

LilyPond

Das Notensatzprogramm

Snippets

Das LilyPond-Entwicklerteam

This document shows a selected set of LilyPond snippets from the LilyPond Snippet Repository (<https://lsr.di.unimi.it>) (LSR). It is in the public domain.

We would like to address many thanks to Sebastiano Vigna for maintaining LSR web site and database, and the University of Milano for hosting LSR.

Please note that this document is not an exact subset of LSR: some snippets come from input/new LilyPond sources directory, and snippets from LSR are converted through convert-ly, as LSR is based on a stable LilyPond version, and this document is for version 2.25.34.

Snippets are grouped by tags; tags listed in the table of contents match a section of LilyPond notation manual. Snippets may have several tags, and not all LSR tags may appear in this document.

In the HTML version of this document, you can click on the file name or figure for each example to see the corresponding input file.

Zu mehr Information, wie dieses Handbuch unter den anderen Handbüchern positioniert, oder um dieses Handbuch in einem anderen Format zu lesen, besuchen Sie bitte Abschnitt “Handbücher” in *Allgemeine Information*.

Wenn Ihnen Handbücher fehlen, finden Sie die gesamte Dokumentation unter <https://lilypond.org/>.

This document has been placed in the public domain.

Für LilyPond Version 2.25.34

Inhaltsverzeichnis

1	Pitches	1
	Ambitus pro Stimme hinzufügen	1
	Adding an ottava marking to a single voice	1
	Aiken head thin variant noteheads	2
	Altering the length of beamed stems	2
	Ambitus	3
	Ambitus after key signature	3
	Ambitus mit vielen Stimmen	4
	Notenkopfstile basierend auf der Tonleiterstufe erstellen	4
	Automatically changing the stem direction of the middle note based on the melody	5
	Changing ottava text	6
	Changing the ambitus gap	6
	Changing the interval of lines on the stave	7
	Clefs can be transposed by arbitrary amounts	8
	Coloring notes depending on their pitch	8
	Creating a sequence of notes on various pitches	9
	Creating custom key signatures	9
	Direction of merged ‚fa‘ shape note heads	10
	Force a cancellation natural before accidentals	10
	Forcing a clef symbol to be displayed	11
	Generating random notes	11
	Hiding accidentals on tied notes at the start of a new system	11
	Keep change clefs full-sized	12
	Makam-Beispiel	12
	Modifying the ottava spanner slope	13
	Untypische Tonarten	13
	Numbers as easy note heads	14
	Orchester, Chor und Klavier	15
	Verhindern, dass zusätzliche Auflösungszeichen automatisch hinzugefügt werden	20
	Auflösungszeichen nicht setzen, wenn die Tonart wechselt	20
	Eine Stimme mit Transposition zitieren	20
	Separating key cancellations from key signature changes	21
	Transposing pitches with minimum accidentals („smart“ transpose)	22
	Turkish Makam example	24
	Eigenschaften des Schlüssels optimieren	25
	Using \autoChange with more than one voice	26
2	Rhythms	28
	Adding beams, slurs, ties, etc., when using triplet and non-triplet rhythms	28
	Adding drum parts	28
	Adjusting grace note spacing	29
	Taktnummern ausrichten	29
	Alternative breve notes	30
	Appoggiatura or grace note before a bar line	31
	Automatic beam subdivisions	31
	Automatically change durations	32
	Balkenenden auf Score-Ebene	33
	Beam nibs	34

Balken über Zeilenumbrüche	35
Balken für weit auseinander liegende Noten ändern	36
Die Erscheinung von Pausentakten ändern	36
Changing the number of augmentation dots per note	36
Das Tempo ohne Metronom-Angabe verändern	37
Die Zahl der N-tole verändern	37
Changing time signatures inside a polymetric section using <code>\scaleDurations</code>	37
Chant or psalm notation	39
Complex time signatures	39
Dirigierzeichen, Taktgruppenzeichen	39
Controlling tuplet bracket visibility	40
Cow and ride bell example	41
Eine Metronombezeichnung als Textbeschriftung erstellen	41
Bindebögen manuell setzen	42
Engraving tremolos with floating beams	43
Entering several tuplets using only one <code>\tuplet</code> command	43
Forcing rehearsal marks to start from a given letter or number	44
Generating custom flags	44
Schlagrhythmus für Gitarren	45
Heavily customized polymetric time signatures	46
High and low woodblock example	47
Making an object invisible using <code>\hide</code>	48
Legatobögen mit kompliziertem Strichelmuster definieren	49
Manually controlling beam positions	49
Merging multi-measure rests in a polyphonic part	49
Modifying tuplet bracket length	50
Moving dotted notes in polyphony	50
Multi-measure rest length control	51
Textbeschriftung und Mehrtaktpausen	51
Nicht-standard-N-tolennummern	52
Numbering single measure rests	53
Partcombine and <code>\autoBeamOff</code>	53
Percussion example	54
Zeilenumbrüche bei N-tolen mit Balken erlauben	55
Positioning grace note beams at the height of normal note beams	56
Positionierung von Verzierungen mit verschiebbarem Platz	56
Positionierung von Ganztaktpausen	57
Positioning opposing fermatas on a bar line	58
Preventing final mark from removing final tuplet	58
Setzen der Taktnummern in regelmäßigen Intervallen	59
Printing bar numbers for broken measures	59
Setzen von Taktnummern in Kästen oder Kreisen	60
Printing bar numbers using <code>modulo-bar-number-visible</code>	61
Printing bar numbers with changing regular intervals	61
Metronom- und Übungszeichen unter das System setzen	62
Printing music with different time signatures	62
Setzen der Taktnummer für den ersten Takt	64
Printing tuplet brackets on the note head side	65
Globale Umdefinition von Verzierungsnoten	65
Entfernung von Taktnummern in einer Partitur	66
Pausenstile	66
Standard-Balkenwerte rückgängig machen	67

Rhythmic slashes	68
Skips in lyric mode	68
Skips in lyric mode (2)	69
Stemlets	69
Bebalkung nach Taktschlag	70
Subdividing beams	70
Tam-tam example	71
Tambourine example	72
Three-sided box	72
Time signature in brackets	73
Time signature in parentheses	73
Time signature printing only the numerator as a number (instead of the fraction)	74
Veränderung des Layouts von Verzierungen innerhalb der Noten	74
User-defined time signatures	75
Using alternative flag styles	75
Using grace note slashes with normal heads	76
Überbindungen für Arpeggio benutzen	77
3 Expressive marks	78
Adding parentheses around an expressive mark or chordal note	78
Adding timing marks to long glissandi	78
Adjusting slur positions vertically	78
Das Aussehen von unbestimmten Glissandi anpassen	79
Aligning the ends of hairpins to NoteColumn directions	80
Alternative breve notes	80
Asymmetric slurs	80
Breathing signs	81
Broken crescendo hairpin	82
Caesura („railtracks“) with fermata	82
Center text below hairpin dynamics	83
Text und Strecker-Stile für Dynamik-Texte ändern	84
Changing the appearance of a slur from solid to dotted or dashed	85
Das Atemzeichen-Symbol verändern	85
Changing the number of augmentation dots per note	85
Combining dynamics with markup texts	86
Combining dynamics with markup texts (2)	86
Moderne Glissandi	86
Controlling spanner visibility after a line break	87
Position von Fingersatz in Akkorden kontrollieren	88
Die vertikale Anordnung von Beschriftungen kontrollieren	88
Creating „real“ parenthesized dynamics	89
Einen Doppelschlag mit Vorhalt erstellen	89
Arpeggios zwischen unterschiedlichen Stimmen erzeugen	90
Arpeggio über mehrere Systeme in anderen Kontexten	90
Arpeggio zwischen Systemen in einem Klaviersystem erstellen	91
Creating double-digit fingerings	92
Creating slurs across voices	92
Creating text spanners	92
Dynamics spanner with custom text	93
Glissando kann Grobs überspringen	94
Hairpins with different line styles	94

Crescendo-Linien von Dynamik-Texten unterdrücken	95
Horizontally aligning custom dynamics like „più f“	95
Eine Zäsur einfügen	98
Laissez vibrer ties	98
Line arrows	99
Legatobögen mit kompliziertem Strichelmuster definieren	99
Die Standardwerte der Abkürzungen von Artikulationen verändern	100
Moving slur positions vertically	100
Moving the ends of hairpins	101
Positioning arpeggios	102
Textbeschriftung innerhalb von Bögen positionieren	102
Printing hairpins in various styles	102
Crescendo Klammern al niente schreiben	103
Metronom- und Übungszeichen unter das System setzen	103
Das Verhalten von Crescendo-Klammern an Taktlinien beeinflussen	104
Die Mindestlänge von Crescendo-Klammern bestimmen	104
Showing the same articulation above and below a note or chord	104
Snap pizzicato („Bartok“ pizzicato)	105
Using a bracket to clarify divisi	105
Using a tick as the breath mark symbol	106
Doppelte Bögen für Legato-Akkorde benutzen	106
Using the whiteout property	107
Vertical line as a baroque articulation mark	107
Vertically aligning dynamics across multiple notes	107
4 Repeats	109
Changing the default bar lines	109
Controlling the appearance of tremolo slashes	109
Cross-staff tremolos	111
Engraving tremolos with floating beams	111
Isolierte Prozentwiederholungen	112
Measure counters	112
Sichtbarkeit von Prozent-Wiederholungen	113
Prozent-Wiederholungen zählen	114
Positioning segno and coda (with line break)	114
Setting the double repeat default for volte	115
Volta-Klammern verkürzen	115
Unfolding tremolo repeats	116
Wiederholungs-(Volta-)Klammern unterhalb der Akkordsymbole	116
Volta brackets in multiple staves	117
Volta text markup using repeatCommands	118
5 Simultaneous notes	119
Zusätzliche Stimmen, um Zusammenstöße zu vermeiden	119
Changing \partCombine texts	119
Changing a single note's size in a chord	120
Clusters	120
Zwei Stimmen auf einem System kombinieren	121
Displaying complex chords	122
Horizontale Verschiebung von Noten erzwingen	122
Making an object invisible using \hide	123

Moving dotted notes in polyphony	124
Suppressing warnings for clashing note columns	124
Two \partCombine pairs on one staff	124
6 Staff notation	127
Ambitus pro Stimme hinzufügen	127
Adding an extra staff	127
Adding an extra staff at a line break	128
Adding indicators to staves which get split after a break	129
Orchesterstichnoten zu einem Klavierauszug hinzufügen	133
Alternative Takt Nummerierung	135
Ambitus after key signature	136
Changing the default bar lines	137
Changing the number of lines in a staff	137
Changing the staff size	138
Creating blank staves	138
Creating custom key signatures	140
Hälsa über beide Systeme	141
Klammer anzeigen, wenn nur ein System gesetzt wird	142
Displaying a whole GrandStaff system if only one of its staves is alive	142
Extending a trill spanner	144
Extending glissandi across repeats	145
Flat ties	146
Forcing measure width to adapt to a metronome mark's width	148
Glissando kann Grobs überspringen	149
Harmonizing bar line thickness for staves with different sizes	149
Incipit	151
Inserting score fragments above a staff, as markups	156
Let TabStaff print the topmost string at bottom	157
Letter tablature formatting	158
Making glissandi breakable	158
Eine Linie des Notensystems dicker als die anderen machen	160
Measure counters	160
Mensurstriche-Layout (Taktstriche zwischen den Systemen)	161
Modifying the ottava spanner slope	161
Systeme schachteln	162
Untypische Tonarten	163
Orchester, Chor und Klavier	163
Print chord names with same root and different bass as slash and bass note	168
Putting lyrics inside the staff	170
Eine andere Stimme zitieren	170
Eine Stimme mit Transposition zitieren	171
Removing brace on first line of piano score	172
Removing connecting bar lines on StaffGroup, PianoStaff, or GrandStaff	173
Die erste leere Notenzeile auch entfernen	174
Setting system separators	175
Shape individual ties in chords	176
Tick bar lines	179
Time signature in brackets	179
Time signature in parentheses	179
Eigenschaften des Schlüssels optimieren	180

Two \partCombine pairs on one staff	181
Eine eckige Klammer zu Beginn von Systemgruppen benutzen	183
Using \autoChange with more than one voice	183
Using mark lines in a Frenched score	184
Vertically aligned StaffGroups without connecting SystemStartBar	187
Wiederholungs-(Volta-)Klammern unterhalb der Akkordsymbole	193
Volta brackets in multiple staves	194
7 Editorial annotations	195
Adding fingerings to a score	195
Adding links to objects	195
Adding markups in a tablature	197
Fingersatz auch innerhalb des Systems setzen	197
Alternative Taktnummerierung	198
Analysis brackets above the staff	199
Analysis brackets with labels	200
Notenkopfstile basierend auf der Tonleiterstufe erstellen	200
Blanking staff lines using the \whiteout command	201
Changing a single note's size in a chord	202
Changing the appearance of a slur from solid to dotted or dashed	202
Coloring notes depending on their pitch	202
Position von Fingersatz in Akkorden kontrollieren	203
Einen Doppelschlag mit Vorhalt erstellen	204
Creating blank staves	204
Creating double-digit fingerings	206
Standardrichtung für Hälse auf der Mittellinie	206
Different font size settings for instrumentName and shortInstrumentName	207
Drawing boxes around grobs	208
Drawing circles around note heads	209
Drawing circles around various objects	209
Embedding native PostScript in a \markup block	210
Generate special note head shapes	210
Gitternetzlinien: Aussehen verändern	211
Grid lines: emphasizing rhythms and notes synchronization	212
Hammer-on and pull-off	213
Hammer-on and pull-off using chords	213
Hammer-on and pull-off using voices	213
Eine Linie des Notensystems dicker als die anderen machen	214
Marking notes of spoken parts with a cross on the stem (Sprechstimme)	214
Measure counters	215
Measure spanner	216
Positioning fingering indications precisely	217
Textbeschriftung innerhalb von Bögen positionieren	218
Printing text from right to left	218
String number extender lines	218
Using the whiteout property	219
8 Text	220
Adding markups in a tablature	220
Adding the current date to a score	220
Adjusting vertical spacing of lyrics	221

Aligning and centering instrument names.....	221
Aligning syllables with melisma.....	223
Aligning text marks to notes.....	223
Blanking staff lines using the \whiteout command.....	224
Center text below hairpin dynamics.....	224
Changing ottava text.....	226
Changing the default text font family.....	226
Combining dynamics with markup texts.....	227
Combining dynamics with markup texts (2).....	228
Zwei Stimmen auf einem System kombinieren.....	228
Creating „real“ parenthesized dynamics.....	229
Creating text spanners.....	230
Demonstrating all \header fields.....	231
Embedding native PostScript in a \markup block.....	232
Silben im Gesangstext formatieren.....	232
How to put ties between syllables in lyrics.....	233
Ausrichtung von Gesangstext.....	233
Markup list.....	234
Textbeschriftung und Mehrtaktpausen.....	235
Of the ubiquity of markup objects.....	235
Outputting the version number.....	237
Piano template with centered lyrics.....	237
Zeichen über jedem System ausgeben.....	238
Printing text from right to left.....	238
Putting lyrics inside the staff.....	239
Isolierter Text in zwei Spalten.....	239
String number extender lines.....	240
Three-sided box.....	240
UTF-8.....	241
Vorlage für Vokalensemble mit dem Gesangstext über und unter dem System.....	242
Volta text markup using repeatCommands.....	244
9 Vocal music.....	245
Ambitus pro Stimme hinzufügen.....	245
Adding indicators to staves which get split after a break.....	245
Orchesterstichnoten zu einem Klavierauszug hinzufügen.....	249
Adjusting vertical spacing of lyrics.....	251
Aligning syllables with melisma.....	251
Ambitus.....	252
Ambitus after key signature.....	253
Ambitus mit vielen Stimmen.....	253
Ancient notation template – modern transcription of Gregorian music.....	254
Vorlage für anglikanischen Psalm.....	254
Arranging separate lyrics on a single line.....	257
Changing stanza fonts.....	258
Chant or psalm notation.....	259
Forcing hyphens to be shown.....	259
Silben im Gesangstext formatieren.....	260
How to put ties between syllables in lyrics.....	260
Hymnus-Vorlage.....	260
Ausrichtung von Gesangstext.....	262

Marking notes of spoken parts with a cross on the stem (Sprechstimme)	263
Orchester, Chor und Klavier	264
Vorlage für Klavier und Gesangsstimme	269
Putting lyrics inside the staff	270
SATB-Chorvorlage auf vier Systemen	270
Vorlage für eine Notenzeile mit Noten, Text und Akkorden	272
Single-staff template with notes, lyrics, chords, and frets	273
Vorlage für ein Notensystem mit Noten und Gesangstext	274
Skips in lyric mode	274
Skips in lyric mode (2)	275
Using a bracket to clarify divisi	275
Using tags to produce mensural and modern music from the same source	276
Gesangstext und Ossia vertikal ausrichten	278
Vertically aligning stanza numbers of different staves	279
Vertically centered common lyrics	280
Vorlage für Vokalensemble	281
Vocal ensemble template with automatic piano reduction	283
Vorlage für Vokalensemble mit dem Gesangstext über und unter dem System	285
Sologesang und zweistimmiger Refrain	287
10 Chord notation	290
Adding a figured bass above or below the notes	290
Tatkstriche in einen ChordNames-Kontext hinzufügen	290
Adjusting figured bass alteration glyphs	291
Changing a single note's size in a chord	291
Akkordsymboltrenner verändern	291
Positionen von Generalbass-Alterationszeichen verändern	292
Akkordsymbolausnahmen	292
Akkordbezeichnung maj7	293
Chord names alternative	293
Chords with stretched fingering for FretBoards and TabVoice	303
Clusters	303
Position von Fingersatz in Akkorden kontrollieren	304
Cross-staff chords – beaming problems workaround	304
Customizing the chord grid style	305
Customizing the no-chord symbol	306
Display non-English chord names	306
Displaying complex chords	307
Manually break figured bass extenders for only some numbers	308
Print chord names with same root and different bass as slash and bass note	308
Akkordsymbole bei Wechsel anzeigen	311
Ein einfaches Liedblatt	311
Vorlage für eine Notenzeile mit Noten, Text und Akkorden	311
Single-staff template with notes, lyrics, chords, and frets	312
Vorlage für eine Notenzeile mit Akkorden	313
Vertically centering paired figured bass extenders	314
Wiederholungs-(Volta-)Klammern unterhalb der Akkordsymbole	314
11 Keyboard and other multi-staff instruments	316
Accordion register symbols	316
Changing the text for sustain markings	317

Clusters	317
Position von Fingersatz in Akkorden kontrollieren	317
Creating slurs across voices	318
Cross-staff chords – beaming problems workaround	319
Cross-staff tremolos	319
Fine-tuning pedal brackets	320
Indicating cross-staff chords with a bracket	320
Vorlage für Jazz-Combo	321
Laissez vibrer ties	327
Vorlage für einfache Klaviernotation	328
Piano template with centered lyrics	328
Vorlage für Klavier und Gesangsstimme	329
Removing brace on first line of piano score	330
Using \autoChange with more than one voice	331
Vocal ensemble template with automatic piano reduction	332
12 Percussion	335
Adding drum parts	335
Cow and ride bell example	336
Heavily customized polymetric time signatures	336
High and low woodblock example	337
Vorlage für Jazz-Combo	338
Percussion beaters	343
Percussion example	346
Printing music with different time signatures	348
Tam-tam example	350
Tambourine example	350
13 Fretted string instruments	352
Adding fingerings to a score	352
Adding fingerings to tablatures	352
Adding markups in a tablature	352
Fingersatz auch innerhalb des Systems setzen	353
Automatic fretboards barré	353
Changing fret orientations	354
Chord changes for fretboards	354
Akkordglissando in Tabulaturen	355
Chords with stretched fingering for FretBoards and TabVoice	356
Position von Fingersatz in Akkorden kontrollieren	357
Bunddiagramme anpassen	357
Anpassung von Beschriftungs-Bunddiagrammen	359
Fingersatz, Saitennummern und Fingersatz für die rechte Hand	360
Flamenco notation	361
Fret diagrams explained and developed	364
Alternative Bunddiagrammtabellen	371
Flageolett von Bundinstrumenten in einer Tabulatur	373
Guitar slides	374
Schlagrhythmus für Gitarren	375
Hammer-on and pull-off	376
Hammer-on and pull-off using chords	377
Hammer-on and pull-off using voices	377

How to change fret diagram position	377
Vorlage für Jazz-Combo	378
Laissez vibrer ties	384
Let TabStaff print the topmost string at bottom	385
Letter tablature formatting	386
Referenz für Flageolett von offenen Saiten	386
Positionierung von Fingersatz der rechten Hand	388
Polyphonie in einer Tabulatur	389
Setting up predefined fretboards for other instruments	390
Gleiten (Glissando) in Tabulatur	392
Hals- und Balkenverhalten in einer Tabulatur	393
String number extender lines	393
14 Unfretted string instruments	395
Creating slurs across voices	395
Dotted harmonics	395
Snap pizzicato („Bartok“ pizzicato)	396
Vorlage für Streichquartett (einfach)	396
Vorlage für Streichquartett mit einzelnen Stimmen	397
15 Wind instruments	401
Größe von Holzbläserdiagrammen ändern	401
Fingering symbols for wind instruments	401
Flute slap notation	402
Graphische und Text-Holzbläserdiagramme	402
Recorder fingering chart	403
Liste der Löcher für Holzbläserdiagramme	404
Liste der Holzbläserdiagramme	405
16 Ancient notation	407
Adding a figured bass above or below the notes	407
Ancient fonts	407
Ancient notation template – modern transcription of Gregorian music	410
Ancient time signatures	411
Chant or psalm notation	411
Custodes	412
Incipit	413
Mensurstriche-Layout (Taktstriche zwischen den Systemen)	417
Pausenstile	418
Using tags to produce mensural and modern music from the same source	419
Vertical line as a baroque articulation mark	421
17 World music	422
Arabische Improvisation	422
Makam-Beispiel	422
Printing text from right to left	422
Turkish Makam example	423

18	Contexts and engravers	424
	Ambitus pro Stimme hinzufügen	424
	Adding an extra staff	424
	Adding an extra staff at a line break	425
	Tatkstriche in einen ChordNames-Kontext hinzufügen	426
	Ambitus after key signature	426
	Analysis brackets with labels	427
	Automatically changing the stem direction of the middle note based on the melody	427
	Changing MIDI output to one channel per voice	428
	Changing time signatures inside a polymetric section using <code>\scaleDurations</code>	429
	Arpeggios zwischen unterschiedlichen Stimmen erzeugen	430
	Creating blank staves	431
	Arpeggio zwischen Systemen in einem Klaviersystem erstellen	432
	Creating custom key signatures	433
	Hälsa über beide Systeme	434
	Defining an engraver in Scheme: <code>ambitus engraver</code>	435
	Displaying a whole <code>GrandStaff</code> system if only one of its staves is alive	442
	Engravers one by one	443
	Gitternetzlinien: Aussehen verändern	446
	Grid lines: emphasizing rhythms and notes synchronization	447
	Measure counters	448
	Measure spanner	449
	Mensurstriche-Layout (Taktstriche zwischen den Systemen)	450
	Systeme schachteln	451
	Zeilenumbrüche bei N-tolen mit Balken erlauben	452
	Print chord names with same root and different bass as slash and bass note	453
	Zeichen über jedem System ausgeben	455
	Printing music with different time signatures	456
	Entfernung von Taktnummern in einer Partitur	458
	Eine eckige Klammer zu Beginn von Systemgruppen benutzen	459
	Using mark lines in a Frenched score	459
	Using tags to produce mensural and modern music from the same source	462
	Sologesang und zweistimmiger Refrain	464
	Wiederholungs-(Volta-)Klammern unterhalb der Akkordsymbole	466
	Volta brackets in multiple staves	467
19	Tweaks and overrides	468
	Adding an ottava marking to a single voice	468
	Adding links to objects	468
	Adding markups in a tablature	470
	Adding timing marks to long glissandi	471
	Adjusting grace note spacing	471
	Adjusting slur positions vertically	472
	Adjusting vertical spacing of lyrics	473
	Aligning text marks to notes	473
	Altering the length of beamed stems	474
	Alternative Taktnummerierung	474
	Analysis brackets above the staff	476
	Analysis brackets with labels	476
	Asymmetric slurs	477
	Breaking horizontal alignment of dynamics and textscripts	477

Breaking vertical alignment of dynamics and textscripts	478
Caesura („railtracks“) with fermata	479
Changing a single note’s size in a chord	479
Changing beam thickness and spacing	479
Die Erscheinung von Pausentakten ändern	480
Changing properties for individual grobs	480
Text und Strecker-Stile für Dynamik-Texte ändern	481
Changing the default text font family	481
Changing the staff size	482
Das Tempo ohne Metronom-Angabe verändern	483
Changing the text for sustain markings	483
Controlling spanner visibility after a line break	484
Controlling the appearance of tremolo slashes	484
Die vertikale Anordnung von Beschriftungen kontrollieren	486
Controlling tuplet bracket visibility	486
Einen Doppelschlag mit Vorhalt erstellen	487
Creating custom key signatures	488
Creating text spanners	488
Cross-staff chords – beaming problems workaround	489
Hälsa über beide Systeme	490
Custodes	490
Bunddiagramme anpassen	491
Anpassung von Beschriftungs-Bunddiagrammen	493
Klammer anzeigen, wenn nur ein System gesetzt wird	494
Displaying grob ancestry	495
Dotted harmonics	497
Drawing boxes around grobs	497
Drawing circles around various objects	498
Dynamics spanner with custom text	498
Extending a trill spanner	499
Extending glissandi across repeats	499
Fine-tuning pedal brackets	500
Flat ties	501
Force a cancellation natural before accidentals	503
Horizontale Verschiebung von Noten erzwingen	504
Fret diagrams explained and developed	504
Generate special note head shapes	511
Generating custom flags	512
Glissando kann Grobs überspringen	513
Hairpins with different line styles	513
Horizontally aligning custom dynamics like „più f“	514
How to change fret diagram position	517
Eine Zäsur einfügen	518
Keep change clefs full-sized	518
Line arrows	519
Making an object invisible using \hide	519
Making glissandi breakable	520
Manually controlling beam positions	521
Measure-centered bar numbers	522
Mensurstriche-Layout (Taktstriche zwischen den Systemen)	522
Modifying the ottava spanner slope	523
Moving dotted notes in polyphony	524

Moving slur positions vertically	524
Systeme schachteln	525
Overriding articulations by type.....	526
Sichtbarkeit von Prozent-Wiederholungen	527
Positioning arpeggios	528
Positioning fingering indications precisely	528
Positionierung von Ganztaktpausen	529
Textbeschriftung innerhalb von Bögen positionieren	530
Setzen von Taktnummern in Kästen oder Kreisen	530
Metronom- und Übungszeichen unter das System setzen	531
Printing note names with and without an octave marker.....	531
Printing tuplet brackets on the note head side	532
Proportional strict notespacing.....	533
Removing brace on first line of piano score	533
Removing connecting bar lines on StaffGroup, PianoStaff, or GrandStaff.....	534
Die erste leere Notenzeile auch entfernen	535
Pausenstile	536
Rhythmic slashes	537
Separating key cancellations from key signature changes	537
Das Verhalten von Crescendo-Klammern an Taktlinien beeinflussen	539
Setting system separators	539
Shape individual ties in chords.....	540
Showing the same articulation above and below a note or chord	543
String number extender lines.....	543
Suppressing warnings for clashing note columns	544
Time signature in brackets	544
Time signature in parentheses.....	545
Time signature printing only the numerator as a number (instead of the fraction)	545
Tuplet bracket and change staff	545
Eigenschaften des Schlüssels optimieren	546
Veränderung des Layouts von Verzierungen innerhalb der Noten	548
Using alternative flag styles	548
Using ly:grob-object to access grobs with \tweak	550
Using the \tweak command to tweak individual grobs	551
Vertikale Ausrichtung von Dynamik und Textbeschriftung beeinflussen.....	551
Gesangstext und Ossia vertikal ausrichten	552
Vertically aligning stanza numbers of different staves	553
Vertically centering paired figured bass extenders.....	554
20 Paper and layout	555
Aligning and centering instrument names.....	555
Arranging separate lyrics on a single line	556
Book parts	557
Changing the staff size.....	561
Clip systems.....	562
Creating blank staves	563
Demonstrating all \header fields	565
Displaying a whole GrandStaff system if only one of its staves is alive.....	567
Setting system separators	568
Table of contents	569
Vertically aligned StaffGroups without connecting SystemStartBar.....	571

21	Titles	578
	Adding the current date to a score	578
	Aligning and centering instrument names	578
	Demonstrating all \header fields	580
	Outputting the version number	581
22	Spacing	582
	Adjusting vertical spacing of lyrics	582
	Fingersatz auch innerhalb des Systems setzen	582
	Breaking horizontal alignment of dynamics and textscripts	583
	Breaking vertical alignment of dynamics and textscripts	584
	Harmonizing bar line thickness for staves with different sizes	584
	Page label	586
	Proportional strict notespacing	587
	Vertikale Ausrichtung von Dynamik und Textbeschriftung beeinflussen	588
	Gesangstext und Ossia vertikal ausrichten	588
23	MIDI	590
	Changing MIDI output to one channel per voice	590
	Das Tempo ohne Metronom-Angabe verändern	591
	Creating custom dynamics in MIDI output	591
	Demo of MIDI instruments	592
	Replacing default MIDI instrument equalization	595
24	Templates	597
	Ancient notation template – modern transcription of Gregorian music	597
	Vorlage für anglikanischen Psalm	597
	Hymnus-Vorlage	600
	Vorlage für Jazz-Combo	602
	Orchester, Chor und Klavier	608
	Vorlage für einfache Klaviernotation	613
	Piano template with centered lyrics	613
	Vorlage für Klavier und Gesangsstimme	614
	SATB-Chorvorlage auf vier Systemen	615
	Vorlage für eine Notenzeile mit Noten, Text und Akkorden	617
	Single-staff template with notes, lyrics, chords, and frets	618
	Vorlage für eine Notenzeile mit Akkorden	619
	Vorlage für ein Notensystem mit Noten und Gesangstext	619
	Vorlage für ein Notensystem	620
	Vorlage für Streichquartett (einfach)	621
	Vorlage für Streichquartett mit einzelnen Stimmen	622
	Vorlage für Vokalensemble	624
	Vocal ensemble template with automatic piano reduction	626
	Vorlage für Vokalensemble mit dem Gesangstext über und unter dem System	628
	Sologesang und zweistimmiger Refrain	630

1 Pitches

See also Abschnitt “Pitches” in *Notationsreferenz*.

Ambitus pro Stimme hinzufügen

Ambitus können pro Stimme gesetzt werden. In diesem Fall müssen sie manuell verschoben werden, um Zusammenstöße zu verhindern.

```
\new Staff <<
  \new Voice \with {
    \consists "Ambitus_engraver"
  } \relative c' {
    \override Ambitus.X-offset = 2.0
    \voiceOne
    c4 a d e
    f1
  }
  \new Voice \with {
    \consists "Ambitus_engraver"
  } \relative c' {
    \voiceTwo
    es4 f g as
    b1
  }
}>>
```



Adding an ottava marking to a single voice

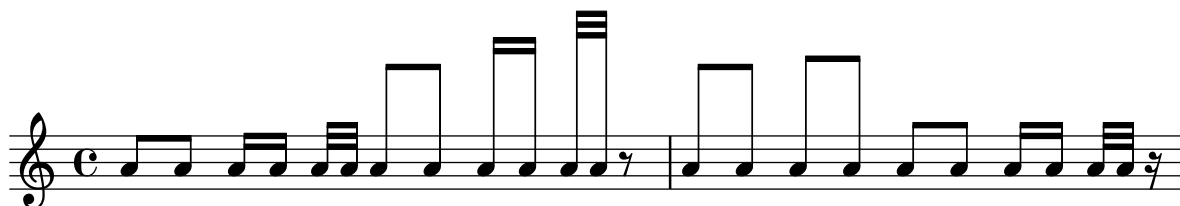
If you have more than one voice on the staff, setting octavation in one voice transposes the position of notes in all voices for the duration of the ottava bracket. If the octavation is only intended to apply to one voice, the *Ottava_spanner_engraver* should be moved to Voice context.

```
\layout {
  \context {
    \Staff
    \remove Ottava_spanner_engraver
  }
  \context {
    \Voice
    \consists Ottava_spanner_engraver
  }
}

{
  \clef bass
  << { <g d'>1~ q2 <c' e'> }
  \\
  {
    r2.
  }
}
```



```
a8[ a] a16[ a] a32[ a] r16 |
}
```



Ambitus

Ambitus indicate pitch ranges for voices.

Accidentals only show up if they are not part of the key signature. `AmbitusNoteHead` grobs also have ledger lines.

```
\layout {
  \context {
    \Voice
    \consists "Ambitus_engraver"
  }
}
```

```
<<
\new Staff {
  \relative c' {
    \time 2/4
    c4 f'
  }
}
\new Staff {
  \relative c' {
    \time 2/4
    \key d \major
    cis4 as'
  }
}
>>
```



Ambitus after key signature

By default, ambitus are positioned at the left of the clef. The `\ambitusAfter` function allows for changing this placement. Syntax is `\ambitusAfter grob-interface`; see Graphical Object Interfaces (<https://lilypond.org/doc/v2.24/Documentation/internals/graphical-object-interfaces>) for a list of possible values for *grob-interface*.

A common use case is printing the ambitus between key signature and time signature.

```
\new Staff \with {
```

```

\consists Ambitus_engraver
} \relative {
  \ambitusAfter key-signature
  \key d \major
  es'8 g bes cis d2
}

```



Ambitus mit vielen Stimmen

Indem man den `Ambitus_engraver` im Staff-Kontext hinzufügt, erhält man einen einzigen Ambitus pro System, auch in dem Fall, dass mehrere Stimmen sich im gleichen System befinden.

```

\new Staff \with {
  \consists "Ambitus_engraver"
}
<<
  \new Voice \relative c'' {
    \voiceOne
    c4 a d e
    f1
  }
  \new Voice \relative c' {
    \voiceTwo
    es4 f g as
    b1
  }
>>

```



Notenkopfstile basierend auf der Tonleiterstufe erstellen

Die `shapeNoteStyles`-(`NotenFormenStile`)-Eigenschaft kann benutzt werden, um verschiedene Notenstile für jeden Schritt der Tonleiter zu definieren (vorgegeben von der Tonart oder der „tonic“ (Tonika)-Eigenschaft. Diese Eigenschaft braucht eine Anzahl von Symbolen, welche beliebig sein können (geometrische Ausdrücke wie `triangle` (Dreieck), `cross` (Kreuz) und `xcircle` (X-Kreis) sind erlaubt) oder basierend auf einer alten amerikanischen Notensatztradition (einige lateinische Notenbezeichnungen sind auch erlaubt).

Um alte amerikanische Liederbücher zu imitieren, gibt es einige vordefinierte Notenstile wie etwa `\aikenHeads` (im Stil von Aiken) oder `\sacredHarpHeads` (im Stil der Sacred Harp-Tradition).

Dieses Beispiel zeigt, wie man unterschiedlich geformte Noten erhält und eine Melodie transponieren kann, ohne dass das Verhältnis zwischen den harmonischen Funktionen und dem Notenstil verloren geht.

```

fragment = {
  \key c \major
  c2 d

```

```

e2 f
g2 a
b2 c
}

\new Staff {
  \transpose c d
  \relative c' {
    \set shapeNoteStyles = ##(do re mi fa
                          #f la ti)

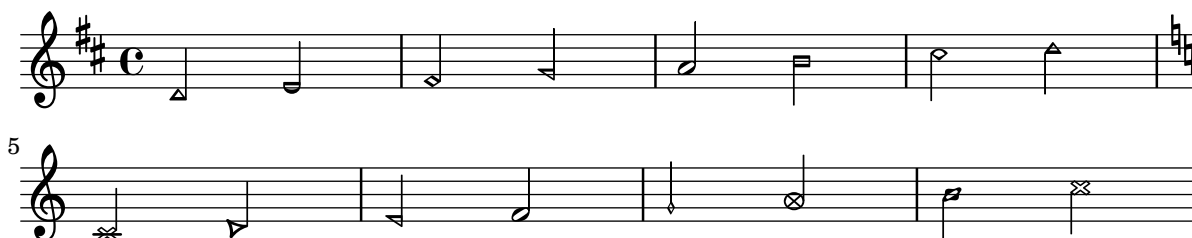
    \fragment
  }

  \break

  \relative c' {
    \set shapeNoteStyles = ##(cross triangle fa #f
                          mensural xcircle diamond)

    \fragment
  }
}

```



Automatically changing the stem direction of the middle note based on the melody

LilyPond can alter the stem direction of the middle note on a staff so that it follows the melody, by adding the `Melody_engraver` to the `Voice` context.

The context property `suspendMelodyDecisions` may be used to turn off this behavior locally.

```

\relative c' {
  \time 3/4
  a8 b g f b g |
  \set suspendMelodyDecisions = ##t
  a b g f b g |
  \unset suspendMelodyDecisions
  c b d c b c |
}

\layout {
  \context {
    \Voice
    \consists "Melody_engraver"
    \autoBeamOff
  }
}

```



Changing ottava text

Internally, `\ottava` sets the properties `ottavation` (for example, to `8va` or `8vb`) and `middleCPosition`. To override the text of the bracket, set `ottavation` after invoking `\ottava`.

Short text is especially useful when a brief ottava is used.

```
{
  c'2
  \ottava 1
  \set Staff.ottavation = "8"
  c''2
  \ottava 0
  c'1
  \ottava 1
  \set Staff.ottavation = "Text"
  c''1
}
```



Changing the ambitus gap

It is possible to change the default gap between the ambitus noteheads and the line joining them.

```
\layout {
  \context {
    \Voice
    \consists "Ambitus_engraver"
  }
}
```

```
\new Staff {
  \time 2/4
  % Default setting
  c'4 g''
}
```

```
\new Staff {
  \time 2/4
  \override AmbitusLine.gap = 0
  c'4 g''
}
```

```
\new Staff {
  \time 2/4
  \override AmbitusLine.gap = 1
  c'4 g''
}
```

```

}

\new Staff {
  \time 2/4
  \override AmbitusLine.gap = 1.5
  c'4 g'
}

```



Changing the interval of lines on the staff

`staffLineLayoutFunction` is used to change the position of notes. This snippet shows setting its value to `ly:pitch-semitones` in order to produce a chromatic scale with the distance between each space and line of the staff equal to one semitone.

```

scale = \relative c' {
  a4 ais b c
  cis4 d dis e
  f4 fis g gis
  a1
}

\new Staff \with {
  \remove "Accidental_engraver"
  staffLineLayoutFunction = #ly:pitch-semitones
}
{
  <<
    \scale
    \context NoteNames {
      \set printOctaveNames = ##f
      \scale
    }
  >>
}

```



Clefs can be transposed by arbitrary amounts, not just by octaves.

The first system of the musical score for 'The Rose Tree' is written in treble clef with a common time signature (C). It consists of four measures. The first measure contains four eighth notes: G4, A4, B4, and C5. The second measure contains four eighth notes: B4, A4, G4, and F#4, with an '8' below the staff. The third measure contains four eighth notes: E4, D4, C4, and B3, with a '5' below the staff. The fourth measure contains a triplet of eighth notes: A3, G3, and F#3, with a '3' above the staff.

It is possible to color note heads depending on their pitch and/or their names: the function used in this example even makes it possible to distinguish enharmonics.

```
% Association list of pitches to colors.
#(define color-mapping
  (list
    (cons (ly:make-pitch 0 0 NATURAL) (x11-color 'red))
    (cons (ly:make-pitch 0 0 SHARP) (x11-color 'green))
    (cons (ly:make-pitch 0 1 FLAT) (x11-color 'green))
    (cons (ly:make-pitch 0 2 NATURAL) (x11-color 'red))
    (cons (ly:make-pitch 0 2 SHARP) (x11-color 'green))
    (cons (ly:make-pitch 0 3 FLAT) (x11-color 'red))
    (cons (ly:make-pitch 0 3 NATURAL) (x11-color 'green))
    (cons (ly:make-pitch 0 4 SHARP) (x11-color 'red))
    (cons (ly:make-pitch 0 5 NATURAL) (x11-color 'green))
    (cons (ly:make-pitch 0 5 FLAT) (x11-color 'red))
    (cons (ly:make-pitch 0 6 SHARP) (x11-color 'red))
    (cons (ly:make-pitch 0 1 NATURAL) (x11-color 'blue))
    (cons (ly:make-pitch 0 3 SHARP) (x11-color 'blue))
    (cons (ly:make-pitch 0 4 FLAT) (x11-color 'blue))
    (cons (ly:make-pitch 0 5 SHARP) (x11-color 'blue))
    (cons (ly:make-pitch 0 6 FLAT) (x11-color 'blue))))

% Compare pitch and alteration (not octave).
#(define (pitch-equals? p1 p2)
  (and
    (= (ly:pitch-alteration p1) (ly:pitch-alteration p2))
    (= (ly:pitch-notename p1) (ly:pitch-notename p2))))

#(define (pitch-to-color pitch)
  (let ((color (assoc pitch color-mapping pitch-equals?)))
    (if color
      (cdr color))))
```

```

#(define (color-notehead grob)
  (pitch-to-color
    (ly:event-property (event-cause grob) 'pitch)))

\score {
  \new Staff \relative c' {
    \override NoteHead.color = #color-notehead
    c8 b d dis ees f g aes
  }
}

```



Creating a sequence of notes on various pitches

In music that contains many occurrences of the same sequence of notes at different pitches, the following music function may prove useful. It takes a note, of which only the pitch is used.

This example creates the rhythm used throughout *Mars*, from Gustav Holst's *The Planets*.

```

rhythm =
#(define-music-function (p) (ly:pitch?)
  "Make the rhythm in Mars (the Planets) at the given pitch"
  #{ \tuplet 3/2 { $p 8 8 8 } 4 4 8 8 4 #})

\new Staff {
  \time 5/4
  \rhythm c'
  \rhythm c''
  \rhythm g
}

```



Creating custom key signatures

LilyPond supports custom key signatures. In this example, print for D minor and D major with an extended range of shown flats.

```

\new Staff \with {
  \override StaffSymbol.line-count = #8
  \override KeySignature.flat-positions = #'((-7 . 6))
  \override KeyCancellation.flat-positions = #'((-7 . 6))
  \override KeySignature.sharp-positions = #'((-6 . 7))
  \override KeyCancellation.sharp-positions = #'((-6 . 7))

  \override Clef.stencil =
    #(\lambda (grob)
      (grob-interpret-markup grob
        #{ \markup\combine

```



```

\musicglyph "clefs.C"
\translate #'(-3 . -2)
\musicglyph "clefs.F"
#})))
clefPosition = #3
middleCPosition = #3
middleCClefPosition = #3
}

{
\key d\minor f bes, f bes, |
\key d\major fis b, fis b, |
}

```



Direction of merged ‚fa‘ shape note heads

Using property `NoteCollision.fa-merge-direction`, the direction of „fa“ shape note heads („fa“, „faThin“, etc.) can be controlled independently of the stem direction if two voices with the same pitch and different stem directions are merged. If this property is not set, the „down“ glyph variant is used.

```

{
\clef bass

<< { \aikenHeads
    f2
    \override Staff.NoteCollision.fa-merge-direction = #UP
    f2 }
\\ { \aikenHeads
    f2
    f2 }
>>
}

```



Force a cancellation natural before accidentals

The following example shows how to force a natural sign before an accidental.

```

\relative c' {
\key es \major
bes c des
\tweak Accidental.restore-first ##t
eis
}

```



Forcing a clef symbol to be displayed

When a clef sign has already been displayed and it has not been changed to a different clef, then repeating the `\clef` command will be ignored by LilyPond, since it is not a change of clef. It is possible to force the clef to be redisplayed using the command `\set Staff.forceClef = ##t`.

```
\relative c' {
  \clef treble
  c1
  \clef treble
  c1
  \set Staff.forceClef = ##t
  c1
  \clef treble
  c1
}
```



Generating random notes

This Scheme-based snippet generates random notes. Use as

```
\randomNotes n from to dur
```

to generate *n* random notes between pitches *from* and *to*, with duration *dur*.

```
randomNotes =
#(define-music-function (n from to dur)
  (integer? ly:pitch? ly:pitch? ly:duration?)
  (let ((from-step (ly:pitch-steps from))
        (to-step (ly:pitch-steps to)))
    (make-sequential-music
     (map (lambda (_)
            (let* ((step (+ from-step
                              (random (- to-step from-step))))
                  (pitch (ly:make-pitch 0 step 0)))
              #{ $pitch $dur #})))
          (iota n))))))

\randomNotes 24 c' g' 8
```

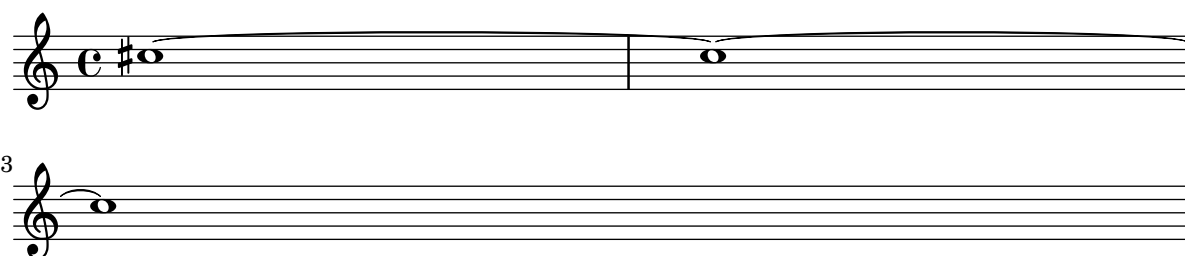


Hiding accidentals on tied notes at the start of a new system

This shows how to hide accidentals on tied notes at the start of a new system.

```
\relative c' {
  \override Accidental.hide-tied-accidental-after-break = ##t
  cis1~ cis~
  \break
  cis
}
```

}



Keep change clefs full-sized

When a clef changes, the clef sign displayed is smaller than the initial clef. This can be overridden by setting the context property `full-size-change` to `##t`.

```
\relative c' {
  \clef "treble"
  c1
  \clef "bass"
  c1
  \clef "treble"
  c1
  \override Staff.Clef.full-size-change = ##t
  \clef "bass"
  c1
  \clef "treble"
  c1
  \revert Staff.Clef.full-size-change
  \clef "bass"
  c1
  \clef "treble"
  c1
}
```



Makam-Beispiel

Makam ist eine türkische Melodie, in der 1/9-Tonabstände eingesetzt werden. Sehen Sie sich die Initialisierungsdatei `makam.ly` für weitere Information zu Tonhöhenbezeichnungen und Alterationen an (siehe Handbuch zum Lernen 2.25.34, 4.6.3 Weitere Information zu Hinweisen, wo diese Datei gespeichert ist).

```
\include "makam.ly"

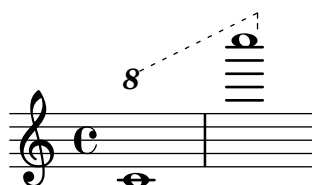
\relative c' {
  \set Staff.keyAlterations = #`((6 . ,(- KOMA)) (3 . ,BAKIYE))
  c4 cc db fk
  gbm4 gfc gfb efk
  fk4 db cc c
}
```



Modifying the ottava spanner slope

It is possible to change the slope of the ottava spanner.

```
\relative c' {
  \override Staff.OttavaBracket.stencil = #ly:line-spanner::print
  \override Staff.OttavaBracket.bound-details =
    #`((left . ((Y . 0)
      (attach-dir . ,LEFT)
      (padding . 0)
      (stencil-align-dir-y . ,CENTER)))
      (right . ((Y . 5.0) ; Change the number here
        (padding . 0)
        (attach-dir . ,RIGHT)
        (text . ,(make-draw-dashed-line-markup
          (cons 0 -1.2))))))
  \override Staff.OttavaBracket.left-bound-info =
    #ly:horizontal-line-spanner::calc-left-bound-info-and-text
  \override Staff.OttavaBracket.right-bound-info =
    #ly:horizontal-line-spanner::calc-right-bound-info
  \ottava 1
  c1
  c'''1
}
```



Untypische Tonarten

Der üblicherweise benutzte `\key`-Befehl setzt die `keySignature`-Eigenschaft im `Staff`-Kontext.

Um untypische Tonartenvorzeichen zu erstellen, muss man diese Eigenschaft direkt setzen. Das Format für den Befehl ist eine Liste: `\set Staff.keySignature = #`(((Oktave . Schritt) . Alteration) ((Oktave . Schritt) . Alteration) ...)` wobei für jedes Element in der Liste Oktave die Oktave angibt (0 ist die Oktave vom eingestrichenen C bis zum eingestrichenen H), Schritt gibt die Note innerhalb der Oktave an (0 heißt C und 6 heißt H), und Alteration ist `,SHARP`, `,FLAT`, `,DOUBLE-SHARP` usw. (Beachte das beginnende Komma.)

Alternativ kann auch jedes Element der Liste mit dem allgemeineren Format `(Schritt . Alteration)` gesetzt werden, wobei dann die Einstellungen für alle Oktaven gelten.

Hier ein Beispiel einer möglichen Tonart für eine Ganztonleiter:

```
\include "arabic.ly"

\relative do' {
  \set Staff.keyAlterations = #`((0 . ,SEMI-FLAT)
    (1 . ,SEMI-FLAT)
```

```

(2 . ,FLAT)
(5 . ,FLAT)
(6 . ,SEMI-FLAT))

% \set Staff.extraNatural = ##f
re reb \down reb resd
dod dob dosd \down dob |
dobsb dodsdo do do |
}

```



Numbers as easy note heads

Easy notation note heads use the `note-names` property of the `NoteHead` object to determine what appears inside the note head. By overriding this property, it is possible to print numbers representing the scale-degree.

A simple engraver can be created to do this for every note head object it sees.

```

#(define Ez_numbers_engraver
  (make-engraver
    (acknowledgers
      ((note-head-interface engraver grob source-engraver)
        (let* ((context (ly:translator-context engraver))
              (tonic-pitch (ly:context-property context 'tonic))
              (tonic-name (ly:pitch-notename tonic-pitch))
              (grob-pitch
                (ly:event-property (event-cause grob) 'pitch))
              (grob-name (ly:pitch-notename grob-pitch))
              (delta (modulo (- grob-name tonic-name) 7)))
              (note-names
                (make-vector 7 (number->string (1+ delta))))))
          (ly:grob-set-property! grob 'note-names note-names))))))

#(set-global-staff-size 30)

\layout {
  ragged-right = ##t
  \context {
    \Voice
    \consists \Ez_numbers_engraver
  }
}

\relative c' {
  \easyHeadsOn
  c4 d e f
  g4 a b c \break

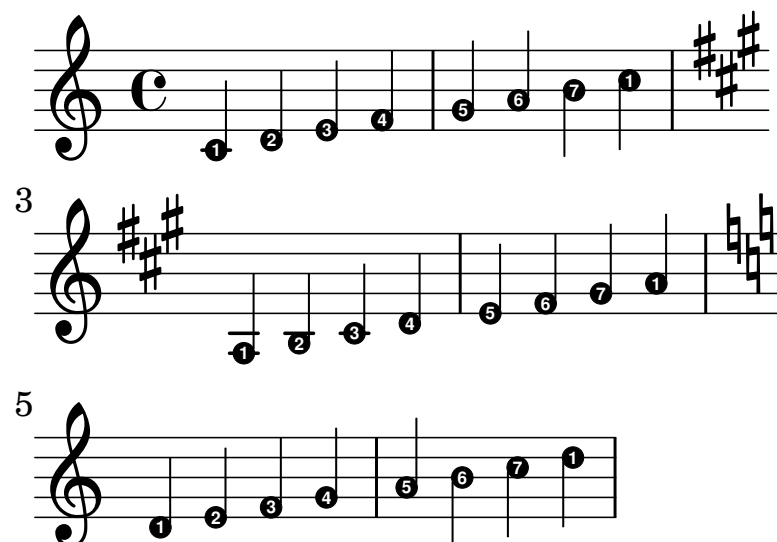
  \key a \major
  a,4 b cis d
  e4 fis gis a \break
}

```

```

\key d \dorian
d,4 e f g
a4 b c d
}

```



Orchester, Chor und Klavier

Diese Vorlage zeigt die Benutzung von geschachtelten `StaffGroup`- und `GrandStaff`-Kontexte, um Instrumente in Untergruppen zu unterteilen, und die Benutzung von `\transpose` für transponierende Instrumente. Alle Noten werden in C geschrieben. Noten können in C eingegeben werden, oder auch in der Tonart des Instrumentes: dann müssen sie zuerst nach C transponiert werden, bevor sie einer Variable zugewiesen werden.

```

#(set-global-staff-size 17)

```

```

\paper {
  indent = 3.0\cm % add space for instrumentName
  short-indent = 1.5\cm % add less space for shortInstrumentName
}

```

```

fluteMusic = \relative c' { \key g \major g'1 b }

```

```

% Pitches as written on a manuscript for Clarinet in A
% are transposed to concert pitch.

```

```

clarinetMusic = \transpose c' a
  \relative c'' { \key bes \major bes1 d }

```

```

trumpetMusic = \relative c { \key g \major g'1 b }

```

```

% Key signature is often omitted for horns

```

```

hornMusic = \transpose c' f
  \relative c { d'1 fis }

```

```

percussionMusic = \relative c { \key g \major g1 b }

```

```

sopranoMusic = \relative c'' { \key g \major g'1 b }

```

```

sopranoLyrics = \lyricmode { Lyr -- ics }

altoIMusic = \relative c' { \key g \major g'1 b }
altoILyrics = \sopranoLyrics
altoIIMusic = \relative c' { \key g \major g'1 b }
altoIILyrics = \lyricmode { Ah -- ah }

tenorMusic = \relative c' { \clef "treble_8" \key g \major g1 b }
tenorLyrics = \sopranoLyrics

pianoRHMus = \relative c { \key g \major g''1 b }
pianoLHMus = \relative c { \clef bass \key g \major g1 b }

violinIMusic = \relative c' { \key g \major g'1 b }
violinIIMusic = \relative c' { \key g \major g'1 b }

violaMusic = \relative c { \clef alto \key g \major g'1 b }

celloMusic = \relative c { \clef bass \key g \major g1 b }

bassMusic = \relative c { \clef "bass_8" \key g \major g,1 b }

\book {
  \score {
    <<
    \new StaffGroup = "StaffGroup_woodwinds" <<
      \new Staff = "Staff_flute" \with { instrumentName = "Flute" }
        \fluteMusic

      \new Staff = "Staff_clarinet" \with {
        instrumentName = \markup { \concat { "Clarinet in B" \flat } }
      }
      % Declare that written Middle C in the music
      % to follow sounds a concert B flat, for
      % output using sounded pitches such as MIDI.
      %\transposition bes

      % Print music for a B-flat clarinet
      \transpose bes c' \clarinetMusic
    >>

    \new StaffGroup = "StaffGroup_brass" <<
      \new Staff = "Staff_hornI" \with {
        instrumentName = "Horn in F"
      }
      % \transposition f
      \transpose f c' \hornMusic

      \new Staff = "Staff_trumpet" \with {
        instrumentName = "Trumpet in C"
      }
      \trumpetMusic
  }
}

```

```

>>

\new RhythmicStaff = "RhythmicStaff_percussion" \with {
  instrumentName = "Percussion"
}
  \percussionMusic

\new PianoStaff \with {
  instrumentName = "Piano"
} <<
  \new Staff { \pianoRHMusik }
  \new Staff { \pianoLHMusik }
>>

\new ChoirStaff = "ChoirStaff_choir" <<
  \new Staff = "Staff_soprano" \with {
    instrumentName = "Soprano"
  }
    \new Voice = "soprano" \sopranoMusic
    \new Lyrics \lyricsto "soprano" { \sopranoLyrics }

  \new GrandStaff = "GrandStaff_altos" \with {
    \accepts Lyrics
  } <<
    \new Staff = "Staff_altoI" \with {
      instrumentName = "Alto I"
    }
      \new Voice = "altoI"
      \altoIMusik
      \new Lyrics \lyricsto "altoI" { \altoILyrics }
    \new Staff = "Staff_altoII" \with {
      instrumentName = "Alto II"
    }
      \new Voice = "altoII"
      \altoIIMusik
      \new Lyrics \lyricsto "altoII" { \altoIILyrics }
  >>

  \new Staff = "Staff_tenor" \with {
    instrumentName = "Tenor"
  }
    \new Voice = "tenor" \tenorMusic
    \new Lyrics \lyricsto "tenor" { \tenorLyrics }
>>

\new StaffGroup = "StaffGroup_strings" <<
  \new GrandStaff = "GrandStaff_violins" <<
    \new Staff = "Staff_violinI" \with {
      instrumentName = "Violin I"
    }
      \violinIMusik
    \new Staff = "Staff_violinII" \with {

```



```
        instrumentName = "Violin II"
    }
    \violinIIMusic
>>

\new Staff = "Staff_viola" \with {
    instrumentName = "Viola"
}
\violaMusic

\new Staff = "Staff_cello" \with {
    instrumentName = "Cello"
}
\celloMusic

\new Staff = "Staff_bass" \with {
    instrumentName = "Double Bass"
}
\bassMusic
>>
>>
}
}
```

Flute

Clarinet in B \flat

Horn in F

Trumpet in C

Percussion

Piano

Soprano

Alto I

Alto II

Tenor

Violin I

Violin II

Viola

Cello

Double Bass

Lyr - ics

Lyr - ics

Ah - ah

Lyr - ics

Verhindern, dass zusätzliche Auflösungszeichen automatisch hinzugefügt werden

Den traditionellen Notensatzregeln zufolge wird ein Auflösungszeichen immer dann vor einem Kreuz oder B gesetzt, wenn ein vorheriges Versetzungszeichen der gleichen Note aufgehoben werden soll. Um dieses Verhalten zu ändern, muss die Eigenschaft `extraNatural` im `Staff`-Kontext auf "false" gesetzt werden.

```
\relative c' {
  aeses4 aes ais a
  \set Staff.extraNatural = ##f
  aeses4 aes ais a
}
```



Auflösungszeichen nicht setzen, wenn die Tonart wechselt

Wenn die Tonart wechselt, werden automatisch Auflösungszeichen ausgegeben, um Versetzungszeichen der vorherigen Tonart aufzulösen. Das kann verhindert werden, indem die `printKeyCancellation`-Eigenschaft im `Staff`-Kontext auf "false" gesetzt wird.

```
\relative c' {
  \key d \major
  a4 b cis d
  \key g \minor
  a4 bes c d
  \set Staff.printKeyCancellation = ##f
  \key d \major
  a4 b cis d
  \key g \minor
  a4 bes c d
}
```



Eine Stimme mit Transposition zitieren

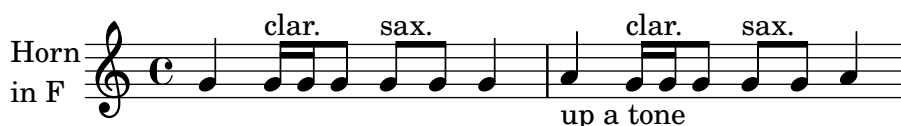
Zitate berücksichtigen sowohl die Transposition der Quelle als auch des Zielinstruments. In diesem Beispiel spielen alle Instrumente klingendes C, das Zielinstrument ist in F. Die Noten für das Zielinstrument können mit `\transpose` transponiert werden, in diesem Fall werden alle Noten (auch die zitierten) transponiert.

```
\addQuote clarinet {
  \transposition bes
  \repeat unfold 8 { d'16 d' d'8 }
}

\addQuote sax {
  \transposition es'
  \repeat unfold 16 { a8 }
}
```

```
quoteTest = {
  % french horn
  \transposition f
  g'4
  << \quoteDuring "clarinet" { \skip 4 } s4^"clar." >>
  << \quoteDuring "sax" { \skip 4 } s4^"sax." >>
  g'4
}

{
  \new Staff \with {
    instrumentName = \markup { \column { Horn "in F" } }
  }
  \quoteTest
  \transpose c' d' << \quoteTest s4_"up a tone" >>
}
```



Separating key cancellations from key signature changes

By default, the accidentals used for key cancellations are placed adjacent to those for key signature changes. This behavior can be changed by overriding the `break-align-orders` property of the `BreakAlignment` grob.

The value of `break-align-orders` is a vector of length 3, with quoted lists of breakable items as elements. Each list describes the default order of prefatory matter at the end, in the middle, and at the beginning of a line, respectively. We are only interested in changing the behaviour in the middle of a line.

If you look up the definition of `break-align-orders` in LilyPond's Internals Reference (<https://lilypond.org/doc/v2.24/Documentation/internals/breakalignment>), you get the following order in the second element:

```
...
staff-bar
key-cancellation
key-signature
...
```

We want to change that, moving `key-cancellation` before `staff-bar`. To make this happen we use the `grob-transformer` function, which gives us access to the original vector as the second argument of the lambda function, here called *orig* (we don't need the first argument, *grob*). We return a new vector, with unchanged first and last elements. For the middle element, we first remove `key-cancellation` from the list, then adding it again before `staff-bar`.

```
#(define (insert-before where what lst)
  (cond
    ((null? lst) ; If the list is empty,
     (list what)) ; return a single-element list.
    ((eq? where (car lst)) ; If we find symbol `where`,
     (cons what lst)) ; insert `what` before curr. position.
    (else ; Otherwise keep building the list by
     (cons (car lst) ; adding the current element and
```

```

                                ; recursing with the next element.
      (insert-before where what (cdr lst))))))

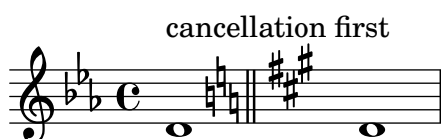
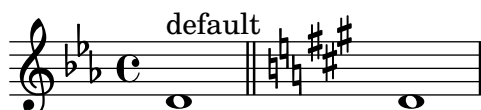
cancellationFirst =
\override Score.BreakAlignment.break-align-orders =
#(grob-transformer
  'break-align-orders
  (lambda (grob orig)
    (let* ((middle (vector-ref orig 1))
           (middle (delq 'key-cancellation middle))
           (middle (insert-before
                        'staff-bar 'key-cancellation middle)))
      (vector
        ;; end of line
        (vector-ref orig 0)
        ;; middle of line
        middle
        ;; beginning of line
        (vector-ref orig 2))))))

music = { \key es \major d'1 \bar "||"
          \key a \major d'1 }

{ <>^\markup "default"
  \music }

{ <>^\markup "cancellation first"
  \cancellationFirst
  \music }

```



Transposing pitches with minimum accidentals („smart“ transpose)

This example uses some Scheme code to enforce enharmonic modifications for notes in order to have the minimum number of accidentals. In this case, the following rules apply:

- double accidentals should be removed
- b sharp → c
- e sharp → f
- c flat → b
- f flat → e

In this manner, the most natural enharmonic notes are chosen.

```
#(define (naturalize-pitch p)
```

```

(let ((o (ly:pitch-octave p))
      ;; `ly:pitch-alteration` returns quarter tone steps.
      (a (* 4 (ly:pitch-alteration p)))
      (n (ly:pitch-notename p)))
  (cond
    ((and (> a 1)
          (or (eqv? n 6) (eqv? n 2))))
    (set! a (- a 2))
    (set! n (+ n 1)))
    ((and (< a -1)
          (or (eqv? n 0) (eqv? n 3))))
    (set! a (+ a 2))
    (set! n (- n 1))))
  (cond
    ((> a 2)
     (set! a (- a 4))
     (set! n (+ n 1)))
    ((< a -2)
     (set! a (+ a 4))
     (set! n (- n 1))))
  (when (< n 0)
    (set! o (- o 1))
    (set! n (+ n 7)))
  (when (> n 6)
    (set! o (+ o 1))
    (set! n (- n 7)))
  (ly:make-pitch o n (/ a 4)))

#(define (naturalize music)
  (let ((es (ly:music-property music 'elements))
        (e (ly:music-property music 'element))
        (p (ly:music-property music 'pitch)))
    (when (pair? es)
      (ly:music-set-property! music 'elements
                              (map naturalize es)))
    (when (ly:music? e)
      (ly:music-set-property! music 'element
                              (naturalize e)))
    (when (ly:pitch? p)
      (set! p (naturalize-pitch p))
      (ly:music-set-property! music 'pitch p))
    music))

naturalizeMusic =
#(define-music-function (m) (ly:music?)
  (naturalize m))

music = \relative c' { c4 d e g }

\new Staff {
  \transpose c ais { \music }
  \naturalizeMusic \transpose c ais { \music }

```

```
\transpose c deses { \music }
\naturalizeMusic \transpose c deses { \music }
}
```



Turkish Makam example

This template uses the start of a well-known Turkish *Saz Semai* that is familiar in the repertoire in order to illustrate some of the elements of Turkish music notation.

```
#{set-default-paper-size "a6" 'landscape)
```

```
\include "turkish-makam.ly"
```

```
\header {
  title = "Hüseyini Saz Semaisi"
  composer = "Lavtacı Andon"
  tagline = ##f
}
```

```
\relative {
  \set Staff.extraNatural = ##f
  \set Staff.autoBeaming = ##f
```

```
\key a \huseyni
\time 10/8
```

```
a'4 g'16[ fb] e8.[ d16] d[ c d e] c[ d c8] bfc |
a16[ bfc a8] bfc c16[ d c8] d16[ e d8] e4 fb8 |
d4 a'8 a16[ g fb e] fb8[ g] a8.[ b16] a16[ g] |
g4 g16[ fb] fb8.[ e16] e[ g fb e] e4 r8 |
```

```
\layout {
  indent = 0
}
```

Hüseyini Saz Semaisi

Lavtacı Andon



Eigenschaften des Schlüssels optimieren

Der Befehl `\clef "treble_8"` ist gleichbedeutend mit einem expliziten Setzen der Eigenschaften von `clefGlyph`, `clefPosition` (welche die vertikale Position des Schlüssels bestimmt), `middleCPosition` und `clefTransposition`. Ein Schlüssel wird ausgegeben, wenn eine der Eigenschaften außer `middleCPosition` sich ändert.

Eine Änderung des Schriftzeichens (Glyph), der Schlüsselposition oder der Oktavierung selber ändert noch nicht die Position der darauf folgenden Noten auf dem System: das geschieht nur, wenn auch die Position des eingestrichenen C (`middleCPosition`) angegeben wird. Die Positionsparameter sind relativ zur Mittellinie des Systems, dabei versetzen positive Zahlen die Position nach oben, jeweils eine Zahl für jede Linie plus Zwischenraum. Der `clefTransposition`-Wert ist normalerweise auf 7, -7, 15 oder -15 gesetzt, aber auch andere Werte sind gültig.

Wenn ein Schlüsselwechsel an einem Zeilenwechsel geschieht, wird das neue Symbol sowohl am Ende der alten Zeilen als auch am Anfang der neuen Zeile ausgegeben. Wenn der Warnungs-Schlüssel am Ende der alten Zeile nicht erforderlich ist, kann er unterdrückt werden, indem die `explicitClefVisibility`-Eigenschaft des Staff-Kontextes auf den Wert `end-of-line-invisible` gesetzt wird. Das Standardverhalten kann mit `\unset Staff.explicitClefVisibility` wieder hergestellt werden.

Die folgenden Beispiele zeigen die Möglichkeiten, wenn man diese Eigenschaften manuell setzt. Auf der ersten Zeile erhalten die manuellen Änderungen die ursprüngliche relative Positionierung von Schlüssel und Noten, auf der zweiten Zeile nicht.

```
{
  % The default treble clef.
  \key f \major
  c'1
  % The standard bass clef
  \set Staff.clefGlyph = "clefs.F"
  \set Staff.clefPosition = 2
  \set Staff.middleCPosition = 6
  \set Staff.middleCClefPosition = 6
  \key g \major
  c'1
  % The baritone clef.
  \set Staff.clefGlyph = "clefs.C"
  \set Staff.clefPosition = 4
  \set Staff.middleCPosition = 4
  \set Staff.middleCClefPosition = 4
  \key f \major
  c'1
  % The standard choral tenor clef.
  \set Staff.clefGlyph = "clefs.G"
  \set Staff.clefPosition = -2
  \set Staff.clefTransposition = -7
  \set Staff.middleCPosition = 1
  \set Staff.middleCClefPosition = 1
  \key f \major
  c'1
  % A non-standard clef.
  \set Staff.clefPosition = 0
  \set Staff.clefTransposition = 0
  \set Staff.middleCPosition = -4
```



```

\set Staff.middleCClefPosition = -4
\key g \major
c'1 \break

% The following clef changes do not preserve
% the normal relationship between notes, key signatures
% and clefs.
\set Staff.clefGlyph = "clefs.F"
\set Staff.clefPosition = 2
c'1
\set Staff.clefGlyph = "clefs.G"
c'1
\set Staff.clefGlyph = "clefs.C"
c'1
\set Staff.clefTransposition = 7
c'1
\set Staff.clefTransposition = 0
\set Staff.clefPosition = 0
c'1

% Return to the normal clef.
\set Staff.middleCPosition = 0
c'1
}

```



Using \autoChange with more than one voice

Here is a demonstration of how to use \autoChange with more than one voice.

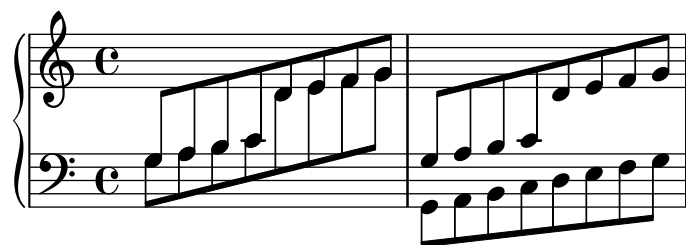
```

\score {
  \new PianoStaff
  <<
    \new Staff = "up" {
      <<
        \set Timing.beamExceptions = #'()
        \set Timing.beatStructure = #'(4)
        \new Voice {
          \voiceOne
          \autoChange
          \relative c' {
            g8 a b c d e f g
            g,8 a b c d e f g
          }
        }
      }
    }
  \new Voice {

```

```
\voiceTwo
\autoChange
\relative c' {
  g8 a b c d e f g
  g,,8 a b c d e f g
}
}
>>
}

\new Staff = "down" {
  \clef bass
}
>>
}
```



2 Rhythms

See also Abschnitt “Rhythms” in *Notationsreferenz*.

Adding beams, slurs, ties, etc., when using tuplet and non-tuplet rhythms

LilyPond primarily uses postfix syntax for inputting parentheses, brackets, etc., which might feel unintuitive for novices.

For example, when entering a manual beam, the left square bracket has to be placed *after* the starting note and its duration, not before. Similarly, the right square bracket should directly follow the note which is to be at the end of the requested beaming, even if this note happens to be inside a tuplet section.

This snippet demonstrates how to combine manual beaming, manual slurs, ties, and phrasing slurs with tuplet sections (enclosed within curly braces).

```
{
  r16[ g16 \tuplet 3/2 { r16 e'8] }
  g16( a \tuplet 3/2 { b d' e' } )
  g8[( a \tuplet 3/2 { b d' ) e']\ ( ~ )
  \time 2/4
  \tuplet 5/4 { e'32 a b d' e' } a'4.\)
}
```



Adding drum parts

Using the powerful pre-configured tools such as the `\drummode` function and the `DrumStaff` context, inputting drum parts is quite easy: drums are placed at their own staff positions (with a special clef symbol) and have note heads according to the drum. Attaching an extra symbol to the drum or restricting the number of lines is possible.

```
drh = \drummode {
  cymc4.^"crash" hhc16^"h.h." hh hhc8 hho hhc8 hh16 hh
  hhc4 r4 r2
}
drl = \drummode {
  bd4 sn8 bd bd4 << bd ss >>
  bd8 tommh tommh bd toml toml bd tomfh16 tomfh
}
timb = \drummode {
  timh4 ssh timl8 ssh r timh r4
  ssh8 timl r4 cb8 cb
}

\score {
  <<
  \new DrumStaff \with {
    instrumentName = "timbales"
    drumStyleTable = #timbales-style
```

```

\override StaffSymbol.line-count = #2
\override BarLine.bar-extent = #'(-1 . 1)
}
<<
\timb
>>
\new DrumStaff \with { instrumentName = "drums" }
<<
\new DrumVoice { \stemUp \drh }
\new DrumVoice { \stemDown \drl }
>>
>>
\layout { }
\midi { \tempo 4 = 120 }
}

```

Adjusting grace note spacing

The space given to grace notes can be adjusted using the spacing-increment property of `Score.GraceSpacing`.

```

graceNotes = {
  \grace { c4 c8 c16 c32 }
  c8
}

\relative c' {
  c8
  \graceNotes
  \override Score.GraceSpacing.spacing-increment = #2.0
  \graceNotes
  \revert Score.GraceSpacing.spacing-increment
  \graceNotes
}

```

Taktnummern ausrichten

Taktnummern sind standardmäßig links an ihrem Ursprungsobjekt ausgerichtet. Das ist normalerweise die linke Ecke einer Linie oder, wenn die Nummern innerhalb einer Zeile gesetzt werden, auf der linken Seite eines Taktstrichs. Die Nummern können auch direkt über dem Taktstrich positioniert werden oder rechts vom Taktstrich gesetzt werden.

```

\relative c' {

```

```

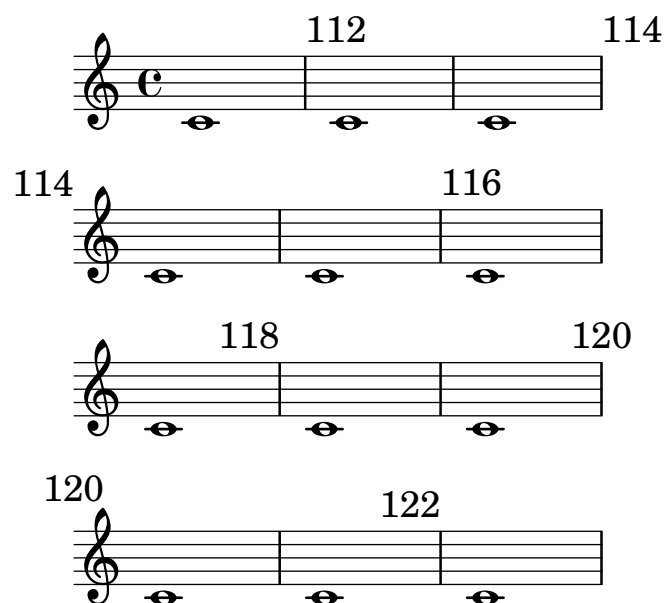
\set Score.currentBarNumber = 111
\override Score.BarNumber.break-visibility = #all-visible
% Increase the size of the bar number by 2
\override Score.BarNumber.font-size = 2
% Print a bar number every second measure
\set Score.barNumberVisibility = #(every-nth-bar-number-visible 2)

c1 | c1 | c1 | \break
c1 | c1 | c1 | \break

\override Score.BarNumber.self-alignment-X =
  #(break-alignment-list CENTER RIGHT CENTER)
c1 | c1 | c1 | \break
c1 | c1 | c1 |
}

\paper {
  line-width = 70\mm
}

```



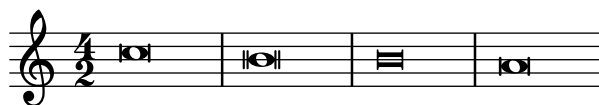
Alternative breve notes

Breve notes are also available with two vertical lines on each side of the notehead instead of one line and in baroque style.

```

\relative c' ' {
  \time 4/2
  c\breve |
  \override Staff.NoteHead.style = #'altdefault
  b\breve
  \override Staff.NoteHead.style = #'baroque
  b\breve
  \revert Staff.NoteHead.style
  a\breve
}

```



Appoggiatura or grace note before a bar line

By default, appoggiaturas and grace notes that occur on the first beat of a measure are printed after the bar line. A possible solution for single staves to print it before the bar line is to add an invisible bar line and then the visible one.

In multi-staff systems, however, adding an invisible bar line distorts the positioning of full-bar rests in other staves; they are no longer centered but slightly shifted to the left. A better solution for such situations is to use the `\afterGrace` command with setting `afterGraceFraction` appropriately.

```
<<
{
  \appoggiatura d''8 c''4 r2. |
  \appoggiatura { \bar "" d''8 \bar "|" } |
  c''4 r2.
}
{ R1 | R1 }
>>

afterGraceFraction = 15/16

<<
{
  \appoggiatura d''8 c''4 \afterGrace r2. d''8( |
  c''4) r2.
}
{ R1 | R1 }
>>
```



Automatic beam subdivisions

Beams can be subdivided automatically. By setting the property `subdivideBeams`, beams are subdivided whenever possible. The intervals and depth of subdivision can be limited with properties `beamMinimumSubdivision` and `beamMaximumSubdivision`, respectively.

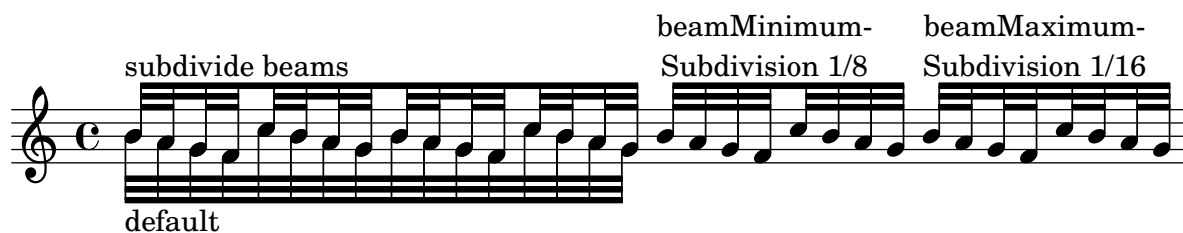
```
\new Staff {
```

```

\relative c'' {
  <<
  {
    \voiceOne
    \set subdivideBeams = ##t
    b32["subdivide beams" a g f c' b a g
    b32 a g f c' b a g]
  }
  \new Voice {
    \voiceTwo
    b32_"default"[ a g f c' b a g
    b32 a g f c' b a g]
  }
  >>
  \oneVoice
  \once \set beamMinimumSubdivision = #1/8
  b32^\markup \center-column { "beamMinimum-"
    "Subdivision 1/8" } [ a g f c' b a g]

  \once \set beamMaximumSubdivision = #1/16
  b32^\markup \center-column { "beamMaximum-"
    "Subdivision 1/16" } [ a g f c' b a g]
}
}

```



Automatically change durations

`shiftDurations` can be used to change the note lengths of a piece of music.

It takes two arguments – the scaling factor as a power of two, and the number of dots to be added as a positive integer.

```
music = \relative c'' { a1 b2 c4 d8 r }
```

```

{
  \time 4/2
  \music
  \time 4/4
  \shiftDurations 1 0 \music
  \time 2/4
  \shiftDurations 2 0 \music
  \time 4/1
  \shiftDurations -1 0 \music
  \time 8/1
  \shiftDurations -2 0 \music
  \time 6/2
  \shiftDurations 0 1 \music
  \time 7/2
}

```

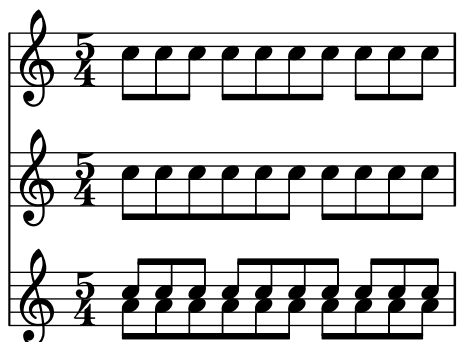
```
\shiftDurations 0 2 \music
}
```



Balkenenden auf Score-Ebene

Balkenenderegeln, die im Score-Kontext definiert werden, wirken sich auf alle Systeme aus, können aber auf Staff- und Voice-Ebene neu verändert werden:

```
\relative c'' {
  \time 5/4
  % Set default beaming for all staves
  \set Score.beatBase = #1/8
  \set Score.beatStructure = 3,4,3
  <<
    \new Staff {
      c8 c c c c c c c c c
    }
    \new Staff {
      % Modify beaming for just this staff
      \set Staff.beatStructure = 6,4
      c8 c c c c c c c c c
    }
    \new Staff {
      % Inherit beaming from Score context
      <<
        {
          \voiceOne
          c8 c c c c c c c c c
        }
        % Modify beaming for this voice only
        \new Voice {
          \voiceTwo
          \set Voice.beatStructure = 6,4
          a8 a a a a a a a a
        }
      >>
    }
  >>
}
```

Beam nibs

Beam nibs at the start and end of beams together with beams attached to solitary notes that look like flat flags are possible with a combination of `stemLeftBeamCount`, `stemRightBeamCount`, and paired `[]` beam indicators.

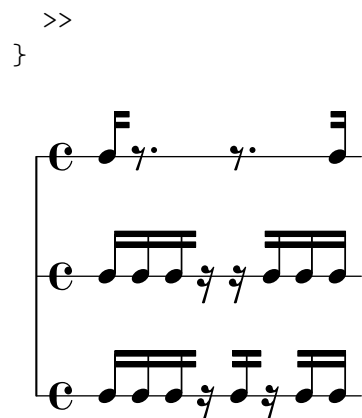
For imitating right-pointing flat flags on lone notes, use paired `[]` beam indicators and set `stemLeftBeamCount` to zero. For imitating left-pointing flat flags on lone notes, set `stemRightBeamCount` to zero instead (line one).

For right-pointing nibs at the end of a run of beamed notes, set `stemRightBeamCount` to a positive value. For left-pointing nibs at the start of a run of beamed notes, set `stemLeftBeamCount` instead (line two).

Sometimes it may make sense for a lone note surrounded by rests to carry both a left- and right-pointing nib. Do this with paired `[]` beam indicators alone (line three).

Note that `\set stemLeftBeamCount` is always equivalent to `\once \set`. In other words, the beam count settings are not „sticky“, so the pair of nibs attached to the lone 16th note in the last example has nothing to do with the `\set` command for the beam before.

```
\score {
  <<
    \new RhythmicStaff {
      \set stemLeftBeamCount = 0
      c16[] r8.
      r8.
      \set stemRightBeamCount = 0
      16[]
    }
    \new RhythmicStaff {
      16 16
      \set stemRightBeamCount = 2
      16 r r
      \set stemLeftBeamCount = 2
      16 16 16
    }
    \new RhythmicStaff {
      16 16
      \set stemRightBeamCount = 2
      16 r16
      16[] r16
      \set stemLeftBeamCount = 2
      16 16
    }
  }
}
```



Balken über Zeilenumbrüche

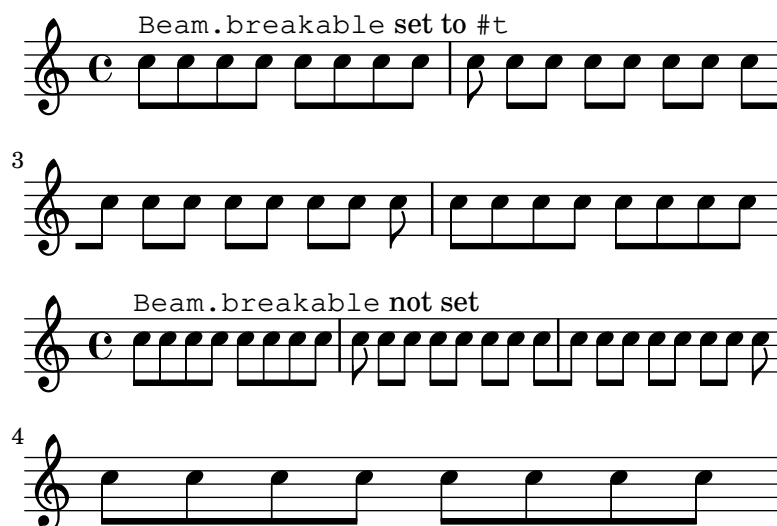
Zeilenumbrüche sind normalerweise während Balken verboten. Das kann geändert werden.

```
music = {
  \repeat unfold 8 c8
  c8 \repeat unfold 7 { c[ c] } c
  \repeat unfold 8 c8
}

\relative c'' {
  <>\markup { \typewriter Beam.breakable set to \typewriter "#t" }
  \override Beam.breakable = ##t
  \music
}

\relative c'' {
  <>\markup { \typewriter Beam.breakable not set }
  \music
}

\paper {
  line-width = 100\mm
}
```



Balken für weit auseinander liegende Noten ändern

Balken mit Hälsen in unterschiedliche Richtungen werden automatisch erstellt, wenn ein großer Sprung zwischen Tonhöhen gefunden wird. Dieses Verhalten kann durch die `auto-knee-gap`-Eigenschaft beeinflusst werden. Ein derartiger Knie-Balken wird erstellt, wenn der Abstand größer ist als der Wert von `auto-knee-gap` plus der Dicke des Balkens (was von der Notendauer und der Neigung des Balkens abhängt). Der Standardwert von `auto-knee-gap` ist 5.5 Notensystemabstände.

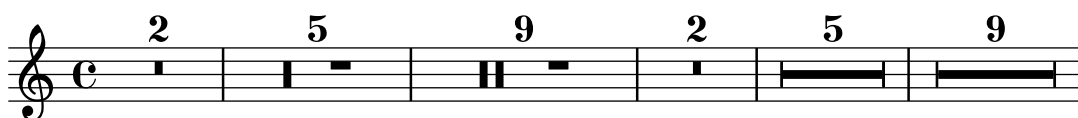
```
{
  f8 f''8 f8 f''8
  \override Beam.auto-knee-gap = #6
  f8 f''8 f8 f''8
}
```



Die Erscheinung von Pausentakten ändern

Wenn zehn oder weniger Pausentakte vorkommen, wird eine Reihe von Longa- und Brevispausen (auch Kirchenpausen genannt) gesetzt, bei mehr Takten wird eine Line mit der Taktanzahl ausgegeben. Der vorgegebene Wert von zehn kann geändert werden, indem man die `expand-limit`-Eigenschaft setzt:

```
\relative c' {
  \compressMMRests {
    R1*2 | R1*5 | R1*9
    \override MultiMeasureRest.expand-limit = 3
    R1*2 | R1*5 | R1*9
  }
}
```



Changing the number of augmentation dots per note

The number of augmentation dots on a single note can be overridden by setting the `dot-count` property of the `Dots` grob.

```
\relative c' {
  c4.. a16 r2 |
  \override Dots.dot-count = 4
  c4.. a16 r2 |
  \override Dots.dot-count = 0
  c4.. a16 r2 |
  \revert Dots.dot-count
  c4.. a16 r2 |
}
```



Das Tempo ohne Metronom-Angabe verändern

Um das Tempo für die MIDI-Ausgabe zu ändern, ohne eine Tempoangabe in den Noten auszugeben, kann die Metronombezeichnung unsichtbar gemacht werden:

```
\score {
  \new Staff \relative c' {
    \tempo 4 = 160
    c4 e g b
    c4 b d c
    \set Score.tempoHideNote = ##t
    \tempo 4 = 96
    d,4 fis a cis
    d4 cis e d
  }
  \layout { }
  \midi { }
}
```



Die Zahl der N-tole verändern

Standardmäßig wird nur der Zähler des N-tolen-Bruchs über der Klammer dargestellt, wie er dem `\times`-Befehl übergeben wird. Man kann aber auch Zähler/Nenner ausgeben lassen, oder die Zahl vollständig unterdrücken.

```
\relative c' {
  \tuplet 3/2 { c8 c c }
  \tuplet 3/2 { c8 c c }
  \override TupletNumber.text = #tuplet-number::calc-fraction-text
  \tuplet 3/2 { c8 c c }
  \omit TupletNumber
  \tuplet 3/2 { c8 c c }
}
```



Changing time signatures inside a polymetric section using \scaleDurations

Flexible polymer with unaligned measures

To support explicit creation of independently measured contexts, remove the `Timing_translator` from `Score` context and define a `TimingStaffGroup` context that has `Timing_translator`. This makes `Timing` an alias for `TimingStaffGroup`, targeting `\time` commands to the enclosing `TimingStaffGroup`.

Unlike LilyPond’s built-in `\enablePerStaffTiming` command, this approach requires the explicit creation of `TimingStaffGroup` contexts; in exchange, it allows creating multiple `Staff` contexts that jointly follow the measure defined in their enclosing `TimingStaffGroup`.

Locally scaled time signatures

Use the unscalable `\time` command to establish a measure of the desired length in Timing, a.k.a. TimingStaffGroup. In this snippet, all staves below TimingStaffGroup use a scaled time signature, so any time signature with the desired measure length is as good as any other. If there were an enclosed context that did not use a scaled time signature, the choice of time signature to set in Timing would matter in that context.

Use the `\polymetric \time` command to set scalable metric properties in contexts below Timing, and use the `\scaleDurations` command to scale both the local meter and the notes to fit the measure.

```
\layout {
  \context {
    \Score
    \remove "Timing_translator"
    \accepts TimingStaffGroup
  }
  \context {
    \StaffGroup
    \name TimingStaffGroup
    \alias StaffGroup
    \consists "Timing_translator"
  }
}

<<
\new TimingStaffGroup <<
  \new Staff {
    \scaleDurations 8/5 {
      \time 6/5 % to set measure length in Timing
      \context Staff \polymetric \time 6/8
      b8 b b b b b
      \time 4/5 % to set measure length in Timing
      \context Staff \polymetric \time 2/4
      b4 b
    }
  }
>>
\new TimingStaffGroup <<
  \new Staff {
    \clef bass
    \time 2/4
    c2 d e f
  }
>>
>>
```



Chant or psalm notation

This form of notation is used for psalm chant, where verses are not always of the same length.

```
stemOff = \hide Staff.Stem
```

```
stemOn = \undo \stemOff
```

```
\score {
  \new Staff \with { \remove "Time_signature_engraver" }
  {
    \key g \minor
    \cadenzaOn
    \stemOff a'\breve bes'4 g'4
    \stemOn a'2 \section
    \stemOff a'\breve g'4 a'4
    \stemOn f'2 \section
    \stemOff a'\breve^\markup { \italic flexe }
    \stemOn g'2 \fine
  }
}
```



Complex time signatures

Odd time signatures (such as „5/8“) can often be played as complex time signatures (e.g. „3/8 + 2/8“), which combine two or more unequal metrics.

LilyPond can make such music quite easy to read and play, by explicitly printing the time signatures and adapting the automatic beaming behavior.

```
\relative c' {
  \time #'((2 . 8) (3 . 8))
  c8 d e fis gis
  c8 fis, gis e d
  c8 d e4 gis8
}
```

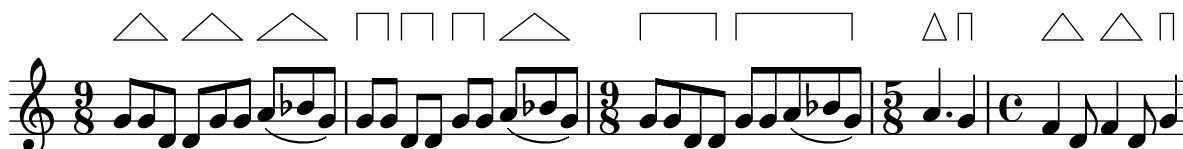


Dirigierzeichen, Taktgruppenzeichen

Optionen, mit denen die Balken in einem Takt gruppiert werden, sind durch die Scheme-Funktion `set-time-signature` erhältlich, die drei Argumente braucht: Die Zahl der

Taktschläge, die Länge des Schläges und die interne gruppieren von Balken in dem Takt. Wenn der `Measure_grouping_engraver` hinzugefügt worden ist, erstellt diese Funktion auch `MeasureGrouping`-(Taktgruppen)-Zeichen. Derartige Zeichen erleichtern das Lesen von rhythmisch komplexer Musik. In dem Beispiel ist der 9/8-Takt in 2, 2, 2 und 3 aufgeteilt. Das wird der `set-time-signature`-Funktion als das dritte Argument mitgegeben: '(2 2 2 3):

```
\score {
  \new Voice \relative c'' {
    \time 9/8
    g8 g d d g g a( bes g) |
    \set Timing.beatStructure = 2,2,2,3
    g8 g d d g g a( bes g) |
    \time 4,5 9/8
    g8 g d d g g a( bes g) |
    \time 5/8
    a4. g4 |
    \time 3,3,2 4/4
    \set Timing.beatBase = #1/8
    f4 d8 f4 d8 g4
  }
  \layout {
    \context {
      \Staff
      \consists "Measure_grouping_engraver"
    }
  }
}
```



Controlling tuplet bracket visibility

The default behavior of tuplet-bracket visibility is to print a bracket unless there is a beam of the same length as the tuplet.

To control the visibility of tuplet brackets, set the property `bracket-visibility` to either `#t` (always print a bracket), `if-no-beam` (only print a bracket if there is no beam) or `#f` (never print a bracket). The latter is in fact equivalent to omitting the `TupletBracket` object altogether from the printed output.

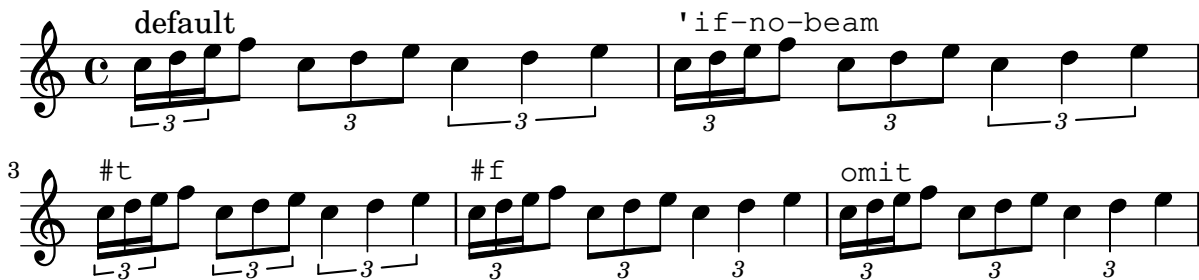
```
music = \relative c'' {
  \tuplet 3/2 { c16[ d e ] f8}
  \tuplet 3/2 { c8 d e }
  \tuplet 3/2 { c4 d e }
}

\new Voice {
  \relative c' {
    \override Score.TextMark.non-musical = ##f
    \textMark "default" \music
    \override TupletBracket.bracket-visibility = #'if-no-beam
    \textMark \markup \typewriter "'if-no-beam" \music
  }
}
```

```

\override TupletBracket.bracket-visibility = ##t
\textMark \markup \typewriter "#t" \music
\override TupletBracket.bracket-visibility = ##f
\textMark \markup \typewriter "#f" \music
\omit TupletBracket
\textMark \markup \typewriter "omit" \music
}
}

```



Cow and ride bell example

Two different bells, entered with ,cb' (cow bell) and ,rb' (ride bell).

```

#(define mydrums '((ridebell default #f 3)
                   (cowbell default #f -2)))

```

```

\new DrumStaff \with { instrumentName = #"Different Bells" }

\drummode {
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)
  \set DrumStaff.clefPosition = 0.5
  \override DrumStaff.StaffSymbol.line-positions = #'(-2 3)
  \override Staff.BarLine.bar-extent = #'(-1.0 . 1.5)

  \time 2/4
  rb8 8 cb8 16 rb16-> ~ |
  16 8 16 cb8 8 |
}

```



Eine Metronombezeichnung als Textbeschriftung erstellen

Neue Metronombezeichnungen können als Textbeschriftung erstellt werden, aber sie ändern nicht das Tempo für die MIDI-Ausgabe.

```

\relative c' {
  \tempo \markup {
    \concat {
      (
        \smaller \general-align #Y #DOWN \note { 16. } #UP
        " = "
        \smaller \general-align #Y #DOWN \note { 8 } #UP
      )
    }
  }
}

```



```

    }
  }
  c1
  c4 c' c,2
}

```



Bindebögen manuell setzen

Überbindungen können manuell gesetzt werden, indem man die `tie-configuration`-Eigenschaft des `TieColumn`-Objekts beeinflusst. Die erste Zahl zeigt den Abstand von der Mitte in Notensystemabständen an, die zweite Zahl zeigt die Richtung an (1 = nach oben, -1 = nach unten).

```

\relative c' {
  <>^"default"
  g'1 ^~ g

  <>^"0"
  \once \override Tie.staff-position = 0
  g1 ^~ g

  <>^"0.0"
  \once \override Tie.staff-position = 0.0
  g1 ^~ g

  <>^"reset"
  \revert Tie.staff-position
  g1 ^~ g
}

\relative c' {
  \override TextScript.outside-staff-priority = ##f
  \override TextScript.padding = 0

  <>^"default"
  <c e g>1~ <c e g>

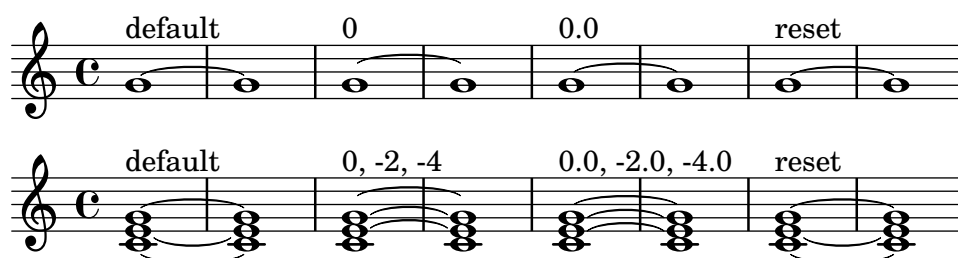
  <>^"0, -2, -4"
  \override TieColumn.tie-configuration =
    #'((0 . 1) (-2 . 1) (-4 . 1))
  <c e g>1~ <c e g>

  <>^"0.0, -2.0, -4.0"
  \override TieColumn.tie-configuration =
    #'((0.0 . 1) (-2.0 . 1) (-4.0 . 1))
  <c e g>1~ <c e g>

  <>^"reset"
  \override TieColumn.tie-configuration = ##f
}

```

```
<c e g>1~ <c e g>
}
```



Engraving tremolos with floating beams

If a tremolo's total duration is less than a quarter-note, or exactly a half note, or between a half note and a whole note, it is normally typeset with all beams touching the stems. Certain engraving styles typeset some of these beams as centered floating beams that do not touch the stems. The number of floating beams in this type of tremolo is controlled with the `gap-count` property of the `Beam` object, and the size of the gaps between beams and stems is set with the `gap` property.

```
\relative c' {
  \repeat tremolo 8 { a32 f }
  \override Beam.gap-count = #1
  \repeat tremolo 8 { a32 f }
  \override Beam.gap-count = #2
  \repeat tremolo 8 { a32 f }
  \override Beam.gap-count = #3
  \repeat tremolo 8 { a32 f }

  \override Beam.gap-count = #3
  \override Beam.gap = #1.33
  \repeat tremolo 8 { a32 f }
  \override Beam.gap = #1
  \repeat tremolo 8 { a32 f }
  \override Beam.gap = #0.67
  \repeat tremolo 8 { a32 f }
  \override Beam.gap = #0.33
  \repeat tremolo 8 { a32 f }
}
```



Entering several tuplets using only one \tuplet command

The property `tupletSpannerDuration` sets how long each of the tuplets contained within the brackets after `\tuplet` should last. Many consecutive tuplets can then be placed within a single `\tuplet` expression, thus saving typing.

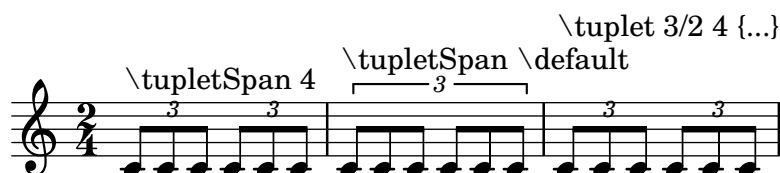
There are ways to set `tupletSpannerDuration` besides using a `\set` command. The command `\tupletSpan` sets it to a given duration, or clears it when instead of a duration `\default` is specified. Another way is to use an optional argument with `\tuplet`.

```
\relative c' {
```

```

\time 2/4
\tupletSpan 4
\tuplet 3/2 { c8~"\tupletSpan 4" c c c c c }
\tupletSpan \default
\tuplet 3/2 { c8~"\tupletSpan \default" c c c c c }
\tuplet 3/2 4 { c8~"\tuplet 3/2 4 {...}" c c c c c }
}

```



Forcing rehearsal marks to start from a given letter or number

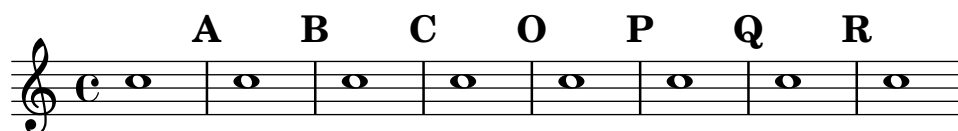
This snippet demonstrates how to obtain automatic ordered rehearsal marks, but from the letter or number desired.

```

\relative c' {
  \override Score.RehearsalMark.Y-offset = #3.5

  c1 \mark \default
  c1 \mark \default
  c1 \mark \default
  c1 \mark #14
  c1 \mark \default
  c1 \mark \default
  c1 \mark \default
  c1
}

```



Generating custom flags

The stencil property of the Flag grob can be set to a custom Scheme function to generate the glyph for the flag.

```

#(define-public (weight-flag grob)
  (let* ((stem-grob (ly:grob-parent grob X))
         (log (- (ly:grob-property stem-grob 'duration-log) 2))
         (is-up? (eqv? (ly:grob-property stem-grob 'direction) UP))
         (yext (if is-up? (cons (* log -0.8) 0) (cons 0 (* log 0.8))))
         (flag-stencil (make-filled-box-stencil '(-0.4 . 0.4) yext))
         (stroke-style (ly:grob-property grob 'stroke-style))
         (stroke-stencil (if (equal? stroke-style "grace")
                              (make-line-stencil 0.2 -0.9 -0.4 0.9 -0.4)
                              empty-stencil)))
    (ly:stencil-add flag-stencil stroke-stencil)))

```

% Create a flag stencil by looking up the glyph from the font

```

#(define (inverted-flag grob)
  (let* ((stem-grob (ly:grob-parent grob X))
        (dir (if (eqv? (ly:grob-property stem-grob 'direction) UP) "d" "u"))
        (flag (retrieve-glyph-flag "" dir "" grob))
        (line-thickness (ly:staff-symbol-line-thickness grob))
        (stem-thickness (ly:grob-property stem-grob 'thickness))
        (stem-width (* line-thickness stem-thickness))
        (stroke-style (ly:grob-property grob 'stroke-style))
        (stencil (if (null? stroke-style)
                     flag
                     (add-stroke-glyph flag stem-grob dir stroke-style "")))
        (rotated-flag (ly:stencil-rotate-absolute stencil 180 0 0)))
    (ly:stencil-translate rotated-flag (cons (- (/ stem-width 2)) 0))))

snippetexamplenotes =
{
  \autoBeamOff c'8 d'16 c'32 d'64 \acciaccatura {c'8} d'64
}

{
  \time 1/4
  <>^"Normal flags"
  \snippetexamplenotes

  <>_"Custom flag: inverted"
  \override Flag.stencil = #inverted-flag
  \snippetexamplenotes

  <>^"Custom flag: weight"
  \override Flag.stencil = #weight-flag
  \snippetexamplenotes

  <>_"Revert to normal"
  \revert Flag.stencil
  \snippetexamplenotes
}

```



Schlagrhythmus für Gitarren

In Gitarrennotation kann neben Melodie, Akkordbezeichnungen und Bunddiagrammen auch der Schlagrhythmus angegeben werden.

```
\include "predefined-guitar-fretboards.ly"
```

```

<<
  \new ChordNames \chordmode {
    c1 | f | g | c
  }

```

```

\new FretBoards \chordmode {
  c1 | f | g | c
}
\new Voice \with {
  \consists "Pitch_squash_engraver"
} \relative c'' {
  \improvisationOn
  c4 c8 c c4 c8 c
  f4 f8 f f4 f8 f
  g4 g8 g g4 g8 g
  c4 c8 c c4 c8 c
}
\new Voice = "melody" \relative c'' {
  c2 e4 e4
  f2. r4
  g2. a4
  e4 c2.
}
\new Lyrics \lyricsto "melody" {
  This is my song.
  I like to sing.
}
>>

```

The image displays a musical score for a song. At the top, four guitar chords are shown: C, F, G, and C. Each chord is represented by a fretboard diagram with fingerings indicated by numbers 1-3. Below the chords is a musical staff with two systems. The first system contains the melody for 'This is my song.' and the second system contains the melody for 'I like to sing.' The lyrics are written below the notes. The melody is in a simple, folk-like style with a mix of eighth and quarter notes.

Heavily customized polymetric time signatures

Though the polymetric time signature shown is not the most essential item here, it has been included to show the beat of this piece (which is the template of a real Balkan song, by the way).

```

melody = \relative c'' {
  \key g \major
  \time #'((3 . 8) (2 . 8) (2 . 8) (3 . 8) (2 . 8) (2 . 8)
           (2 . 8) (2 . 8) (3 . 8) (2 . 8) (2 . 8))
  \set Timing.beamExceptions = #'()
  \set Timing.beatStructure = 3,2,2,3,2,2,2,2,3,2,2
  c8 c c d4 c8 c b c b a4 g fis8 e d c b' c d e4-^ fis8 g \break
  c,4. d4 c4 d4. c4 d c2 d4. e4-^ d4
  c4. d4 c4 d4. c4 d c2 d4. e4-^ d4 \break
}

drum = \new DrumStaff \drummode {
  \repeat volta 2 {

```

```

    bd4.\markup { Drums } sn4 bd \bar ";"
    sn4. bd4 sn \bar ";"
    bd sn bd4. sn4 bd
  }
}

\new Staff {
  \melody
  \drum
}

```

High and low woodblock example

Two Woodblocks, entered with ,wbh' (high woodblock) and ,wbl' (low woodblock). The length of the bar line has been altered with an \override command, otherwise it would be too short. The positions of the two staff lines also have to be explicitly defined.

```

% These lines define the position of the woodblocks in the stave;
% if you like, you can change it or you can use special note heads
% for the woodblocks.

```

```

\define mydrums '((hiwoodblock default #f 3)
                  (lowwoodblock default #f -2)))

```

```

\woodstaff = {
  % This defines a staff with only two lines.
  % It also defines the positions of the two lines.
  \override Staff.StaffSymbol.line-positions = #'(-2 3)

  % This is necessary; if not entered,
  % the barline would be too short!
  \override Staff.BarLine.bar-extent = #'(-1.0 . 1.5)
  % small correction for the clef:
  \set DrumStaff.clefPosition = 0.5
}

```

```

\new DrumStaff {
  % with this you load your new drum style table
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)
}

```

```

\woodstaff

\drummode {
  \time 2/4
  wbh8 16 16 8-> 8 |
  wbl8 16 16-> ~ 16 16 r8 |
}
}

```



Making an object invisible using \hide

Applying `\hide` to a grob causes objects of this type to be printed with „invisible ink“. They are not printed, but all of their other behavior is retained:

- the objects still take up space,
- they take part in collision resolution, and
- slurs, ties, and beams can be attached to them as usual.

This snippet demonstrates how to connect different voices using ties. Normally, ties only connect two notes in the same voice. By introducing a tie in a different voice, and blanking the first up-stem in that voice, the tie appears to cross voices.

```

\relative {
  \time 2/4
  <<
  {
    \once \hide Stem
    \once \override Stem.length = #8
    b'8 ~ 8\noBeam
    \once \hide Stem
    \once \override Stem.length = #8
    g8 ~ 8\noBeam
  }
  \\\
  {
    b8 g g e
  }
  >>
}

```

```

\paper {
  line-width = 40\mm
  ragged-right = ##f
}

```



Legatobögen mit kompliziertem Strichelmuster definieren

Legatobögen können mit einem komplizierten Strichelmuster gesetzt werden, indem die `dash-definition`-Eigenschaft definiert wird. `dash-definition` ist eine Liste bestehend aus `dash-elements`-Elementen. Ein `dash-element` ist eine Liste an Parametern, die das Strichverhalten für einen Abschnitt des Legatobogens definieren.

Der Bogen wird nach dem Bezierparameter `t` definiert, welcher von 0 am linken Ende des Bogens zu 1 am rechten Ende des Bogens reicht. `dash-element` ist eine Liste (`start-t stop-t dash-Unterbrechung dash-Abschnitt`). Die Region des Bogens von `start-t` bis `stop-t` hat eine Unterbrechung von `dash-Unterbrechung` von jedem `dash-Abschnitt-Schwarzabschnitt`. `dash-Abschnitt` ist in Notenlinienzwischenräumen definiert. `dash-Abschnitt` ist auf 1 für einen durchgehenden Bogen gesetzt.

```
\relative c' {
  \once \override
    Slur.dash-definition = #'(( 0 0.3 0.1 0.75)
                              (0.3 0.6 1 1 )
                              (0.65 1.0 0.4 0.75))

  c4( d e f)
  \once \override
    Slur.dash-definition = #'((0 0.25 1 1 )
                              (0.3 0.7 0.4 0.75)
                              (0.75 1.0 1 1 ))

  c4( d e f)
}
```



Manually controlling beam positions

Beam positions may be controlled manually by setting the `positions` property of the `Beam` grob.

```
\relative c' {
  \time 2/4
  % from upper staff-line (position 2) to center (position 0)
  \override Beam.positions = #'(2 . 0)
  c8 c
  % from center to one above center (position 1)
  \override Beam.positions = #'(0 . 1)
  c8 c
}
```



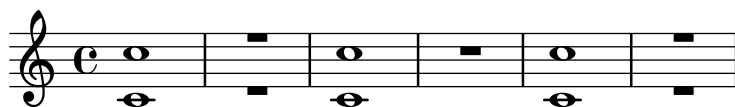
Merging multi-measure rests in a polyphonic part

Multi-measure rests in a polyphonic staff are placed differently depending on the voice they belong to. They can be printed on the same staff line using the setting below. If you omit the `\once` keyword, the change affects all rests in that follow in the given voice.

```
normalPos = \once \revert MultiMeasureRest.direction
```



```
<<
{ c''1 R c'' \normalPos R c'' R } \\
{ c'1 R c' \normalPos R c' R }
>>
```

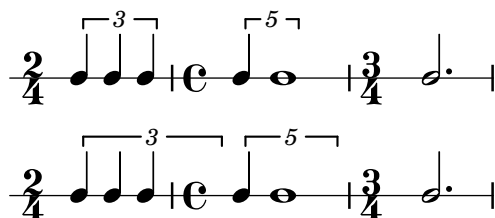


Modifying tuplet bracket length

Tuplet brackets can be made to extend horizontally to prefatory matter or the next note. By default, tuplet brackets end at the right edge of the final note of the tuplet; full-length tuplet brackets extend farther to the right, either to cover all the non-rhythmic notation up to the following note, or to cover only the whitespace before the next item of notation, be that a clef, time signature, key signature, or another note. The example shows how to switch tuplets to full length mode and how to modify what material they cover.

```
\new RhythmicStaff {
  % Defaults.
  \time 2/4 \tuplet 3/2 { c4 4 4 }
  \time 4/4 \tuplet 5/4 { 4 1 }
  \time 3/4 2.
}

\new RhythmicStaff {
  % Set tuplets to be extendable...
  \set tupletFullLength = ##t
  % ...to cover all items up to the next note
  \set tupletFullLengthNote = ##t
  \time 2/4 \tuplet 3/2 { c4 4 4 }
  % ...or to cover just whitespace.
  \set tupletFullLengthNote = ##f
  \time 4/4 \tuplet 5/4 { 4 1 }
  \time 3/4 2.
}
```



Moving dotted notes in polyphony

When a dotted note in the upper voice is moved to avoid a collision with a note in another voice, the default is to move the upper note to the right. This behaviour can be changed setting the prefer-dotted-right property of the NoteCollision grob.

```
\new Staff \relative c' <<
{
  f2. f4
  \override Staff.NoteCollision.prefer-dotted-right = ##f
  f2. f4
}
```

```

\override Staff.NoteCollision.prefer-dotted-right = ##t
f2. f4
}
\\
{ e4 e e e e e e e e e e }
>>

```



Multi-measure rest length control

Multi-measure rests have a length according to their total duration, which is under the control of the space-increment property of the MultiMeasureRest grob; its default value is 2.

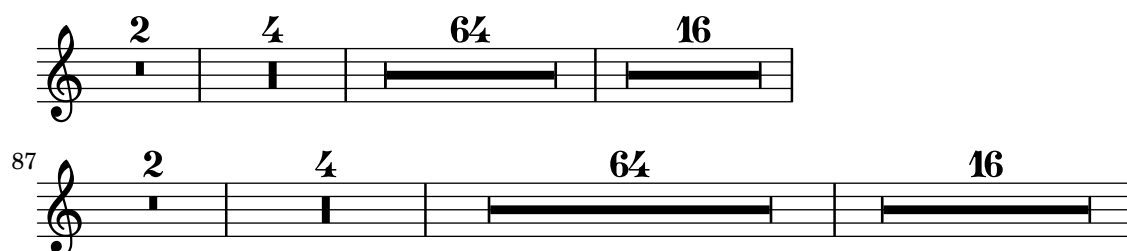
```

\relative c' {
  \omit Staff.TimeSignature
  \compressEmptyMeasures

  R1*2 R1*4 R1*64 R1*16 \break
  \override MultiMeasureRest.space-increment = 4
  R1*2 R1*4 R1*64 R1*16
}

\layout {
  ragged-right = ##t
}

```



Textbeschriftung und Mehrtaktpausen

Textbeschriftungen, die an Mehrtaktpausen gehängt wird, wird über oder unter der Pause zentriert. Lange Beschriftungen lassen den Takt nicht breiter werden. Um eine Mehrtaktpause einer Beschriftung anzupassen, muss eine unsichtbare Pause mit der Beschriftung direkt vor der Mehrtaktpause eingesetzt werden.

Man sollte beachten, dass unsichtbare Pausen automatische Taktstriche nach sich ziehen. Text, der an eine unsichtbare Pause gehängt wird, ist links ausgerichtet an der Position, wo die Pause erscheinen würde. Wenn aber die Länge des Taktes durch die Länge des Textes bestimmt wird, sieht es so aus, als ob der Text zentriert gesetzt ist.

```

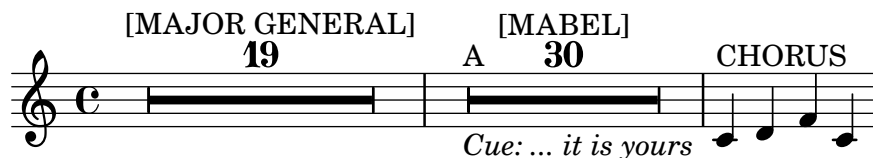
\relative c' {
  \compressMMRests {
    \textLengthOn
    <>^\markup { [MAJOR GENERAL] }
    R1*19
    <>_\markup { \italic { Cue: ... it is yours } }
  }
}

```

```

<>^\markup { A }
R1*30^\markup { [MABEL] }
\textLengthOff
c4^\markup { CHORUS } d f c
}
}

```



Nicht-standard-N-tolennummern

LilyPond stellt auch Formatierungsfunktionen zur Verfügung, mit denen N-tolennummern gesetzt werden können, die sich von dem eigentlichen Bruch unterscheiden. Auch ein Notenwert kann zu Nenner oder Zähler des Bruchs hinzugefügt werden.

```

\relative c'' {
  \once \override TupletNumber.text =
    #(tuplet-number::non-default-tuplet-denominator-text 7)
  \tuplet 3/2 { c4. c4. c4. c4. }
  \once \override TupletNumber.text =
    #(tuplet-number::non-default-tuplet-fraction-text 12 7)
  \tuplet 3/2 { c4. c4. c4. c4. }
  \once \override TupletNumber.text =
    #(tuplet-number::append-note-wrapper
      (tuplet-number::non-default-tuplet-fraction-text 12 7)
      (ly:make-duration 3 0))
  \tuplet 3/2 { c4. c4. c4. c4. }
  \once \override TupletNumber.text =
    #(tuplet-number::append-note-wrapper
      tuplet-number::calc-denominator-text
      (ly:make-duration 2 0))
  \tuplet 3/2 { c8 c8 c8 c8 c8 c8 }
  \once \override TupletNumber.text =
    #(tuplet-number::append-note-wrapper
      tuplet-number::calc-fraction-text
      (ly:make-duration 2 0))
  \tuplet 3/2 { c8 c8 c8 c8 c8 c8 }
  \once \override TupletNumber.text =
    #(tuplet-number::fraction-with-notes
      (ly:make-duration 2 1) (ly:make-duration 3 0))
  \tuplet 3/2 { c4. c4. c4. c4. }
  \once \override TupletNumber.text =
    #(tuplet-number::non-default-fraction-with-notes 12
      (ly:make-duration 3 0) 4 (ly:make-duration 2 0))
  \tuplet 3/2 { c4. c4. c4. c4. }
}

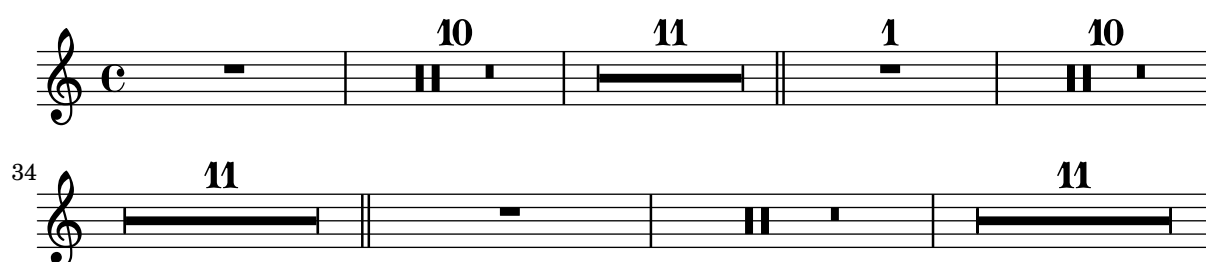
```



Numbering single measure rests

Multi-measure rests show their length by a number except for single measures. This can be changed by setting `restNumberThreshold`.

```
{
  \compressEmptyMeasures
  R1 R1*10 R1*11 \bar "||"
  \set restNumberThreshold = 0
  R1 R1*10 R1*11 \bar "||"
  \set restNumberThreshold = 10
  R1 R1*10 R1*11
}
```



Partcombine and \autoBeamOff

The function of `\autoBeamOff` when used with `\partCombine` can be difficult to understand. It may be preferable to use

```
\set Staff.autoBeaming = ##f
```

instead to ensure that auto-beaming is turned off for the entire staff. Use this at a spot in your score where no beam generated by the auto-beamer is still active.

Internally, `\partCombine` works with four voices – up-stem single, down-stem single, combined, and solo. In order to use `\autoBeamOff` to stop all auto-beaming when used with `\partCombine`, it is necessary to use *four* calls to `\autoBeamOff`.

```
{
  % \set Staff.autoBeaming = ##f % turns off all auto-beaming

  \partCombine {
    \autoBeamOff % applies to split up-stems
    \repeat unfold 4 a'16
    % \autoBeamOff % applies to combined stems
    \repeat unfold 4 a'8
    \repeat unfold 4 a'16
    % \autoBeamOff % applies to solo
    \repeat unfold 4 a'16
    r4
  } {
    % \autoBeamOff % applies to split down-stems
    \repeat unfold 4 f'8
    \repeat unfold 8 f'16 |
    r4
    \repeat unfold 4 a'16
  }
}
```



Percussion example

A short example taken from Stravinsky's *L'histoire du Soldat*.

```
#(define mydrums '((bassdrum   default #f  4)
                   (snare      default #f -4)
                   (tambourine default #f  0)))

U = \stemUp
D = \stemDown

global = {
  \time 3/8 s4.
  \time 2/4 s2*2
  \time 3/8 s4.
  \time 2/4 s2
}

drumsA = {
  \context DrumVoice <<
    \global
    \drummode {
      \autoBeamOff
      \D sn8 \U tamb s |
      sn4 \D sn4 |
      \U tamb8 \D sn \U sn16 \D sn \U sn8 |
      \D sn8 \U tamb s |
      \U sn4 s8 \U tamb
    }
  >>
}

drumsB = \drummode {
  s4 bd8 s2*2 s4 bd8 s4 bd8 s
}

\layout {
  indent = 40\mm
  \context {
    \DrumStaff
    drumStyleTable = #(alist->hash-table mydrums)
  }
}

\score {
  \new StaffGroup <<
    \new DrumStaff \with {
      instrumentName = \markup \center-column {
```

```

    "Tambourine"
    "et"
    "caisse claire s. timbre" }
  } \drumsA
  \new DrumStaff \with {
    instrumentName = "Grosse Caisse"
  } \drumsB
  >>
}

```

Tambourine
et
caisse claire s. timbre

Grosse Caisse

Zeilenumbrüche bei N-tolen mit Balken erlauben

Dieses künstliche Beispiel zeigt, wie sowohl automatische als auch manuelle Zeilenumbrüche innerhalb einer N-tole mit Balken erlaubt werden können. Diese unregelmäßige Bebakung muss allerdings manuell gesetzt werden.

```

\layout {
  \context {
    \Voice
    % Permit automatic line breaks within tuplets.
    \remove "Forbid_line_break_engraver"
    % Allow beams to be broken at line breaks.
    \override Beam.breakable = ##t
  }
}

```

```

\relative c' {
  <>^"manually forced line break"
  a8
  \repeat unfold 5 { \tuplet 3/2 { c8[ b g16 a] } }
  \tuplet 3/2 { c8[ b \break g16 a] }
  \repeat unfold 5 { \tuplet 3/2 { c8[ b g16 a] } }
  c8 \bar "||"
}

```

```

\relative c' {
  <>^"automatic line break"
  \repeat unfold 28 a16
  \tuplet 11/8 { a16[ b c d e f e d c b a] }
  \repeat unfold 28 a16 \bar "||"
}

```

manually forced line break



Positioning grace note beams at the height of normal note beams

When notes are placed on ledger lines, their beams are usually centred on the staff. Grace notes beams are shorter and grace notes on ledger lines may well have beams outside the staff. You can override this beaming for grace notes.

```
\relative c {
  f8[ e]
  \grace {
    f8[ e]
    \override Stem.no-stem-extend = ##f
    f8[ e]
    \revert Stem.no-stem-extend
  }
  f8[ e]
}
```



Positionierung von Verzierungen mit verschiebbarem Platz

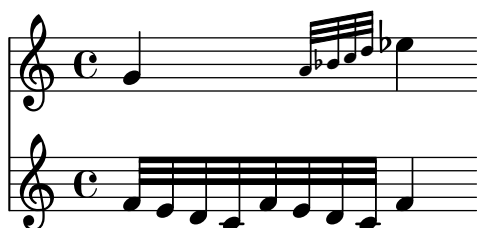
Wenn man die Eigenschaft 'strict-grace-spacing' aktiviert, werden die Verzierungsnoten "fließend" gemacht, d.h. sie sind von den normalen Noten los gekoppelt: Zuerst werden die normalen Noten platziert, dann erst die Verzierungen links von der Hauptnote gesetzt.

```
shiftedGrace =
#(define-music-function (offset music) (number? ly:music?)
  #{
    \override NoteHead.X-offset = #(- offset 0.85)
    \override Stem.X-offset = #offset
    \grace { $music }
    \revert NoteHead.X-offset
    \revert Stem.X-offset
  })

\relative c'' <<
{ g4 \shiftedGrace #-1.3 a32 \shiftedGrace #-0.5 { bes c d } es4 }
```

```
{ f,32 e d c f e d c f4 }
>>
```

```
\layout {
  \context {
    \Score
    \override SpacingSpanner.strict-grace-spacing = ##t
  }
}
```



Positionierung von Ganztaktpausen

Anders als bei normalen Pausen gibt es keinen direkten Befehl, um die vertikale Position von Ganztaktpausen zu beeinflussen, indem man sie an eine Tonhöhe anhängt. In polyphoner Notation wird aber dennoch die Position der Pausen von geraden und ungeraden Stimmen voneinander unterschieden. Die Position von Ganztaktpausen kann wie folgt verändert werden:

```
\relative c' {
  % Multi-measure rests by default are set under the fourth line.
  R1
  % They can be moved using an override or tweak.
  \tweak staff-position -2 R1
  \tweak staff-position 0 R1
  \tweak staff-position 2 R1
  \override MultiMeasureRest.staff-position = 3 R1
  \override MultiMeasureRest.staff-position = 6 R1
  \revert MultiMeasureRest.staff-position
  \break

  % Odd-numbered voices are under the top line.
  << { R1 } \ { a1 } >>
  % Even-numbered voices are under the bottom line.
  << { a1 } \ { R1 } >>
  % Multi-measure rests in both voices remain separate.
  << { R1 } \ { R1 } >>

  % Separating multi-measure rests in more than two voices
  % requires an override or tweak.
  << { R1 } \ { R1 } \ { \tweak staff-position -2 R1 } >>

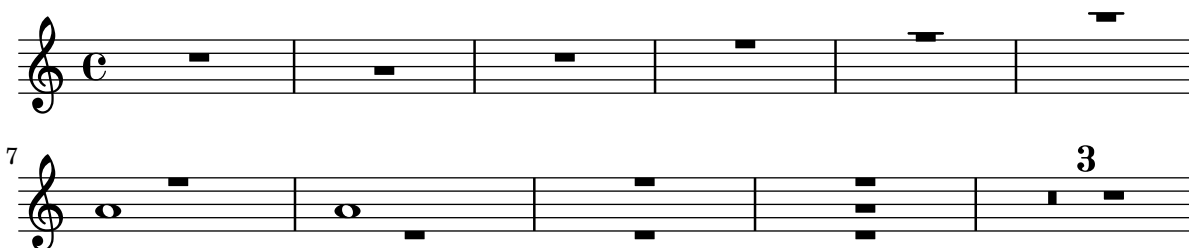
  % Using compressed bars in multiple voices requires another override
  % in all voices to avoid multiple instances being printed.
  \compressMMRests
  <<
    \revert MultiMeasureRest.direction
    { R1*3 } \
```



```

\revert MultiMeasureRest.direction
{ R1*3 }
>>
}

```



Positioning opposing fermatas on a bar line

This snippet demonstrates a command that prints fermatas both above and below a bar line. If there would not otherwise be a bar line, it adds a double bar line. Semantically, the command codes a longer-than-normal caesura, which might be considered misuse depending on the situation.

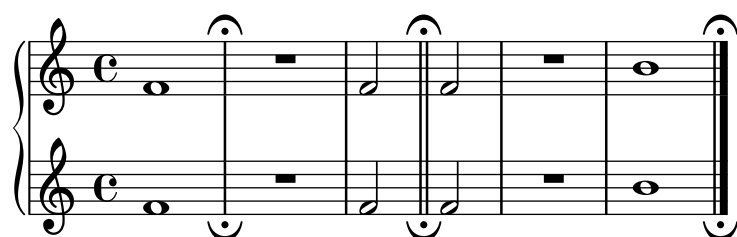
```

twoWayFermata = {
  \once \set Staff.caesuraType = #'((underlying-bar-line . "||"))
  \once \set Staff.caesuraTypeTransform = ##f
  \caesura ~\fermata _\fermata
}

music = {
  f'1 \twoWayFermata
  R1
  f'2 \twoWayFermata f'2
  R1
  b'1 \twoWayFermata \fine
}

\new GrandStaff <<
  \new Staff \music
  \new Staff \music
>>

```



Preventing final mark from removing final tuplet

Due to Issue #2362 (<https://gitlab.com/lilypond/lilypond/-/issues/2362>) the addition of a final mark can result in the loss of a final tuplet marking. This can be overcome by setting `TupletBracket.full-length-to-extent` to `#f`.

```

\new Staff {
  \set tupletFullLength = ##t

```

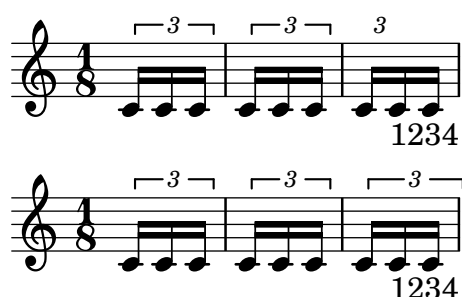
```

\time 1/8
\tuplet 3/2 8 { c'16 c' c' c' c' c' c' c' }
\tweak direction #DOWN \textEndMark "1234"
}

\new Staff {
  \set tupletFullLength = ##t
  \override TupletBracket.full-length-to-extent = ##f

  \time 1/8
  \tuplet 3/2 8 { c'16 c' c' c' c' c' c' c' }
  \tweak direction #DOWN \textEndMark "1234"
}

```



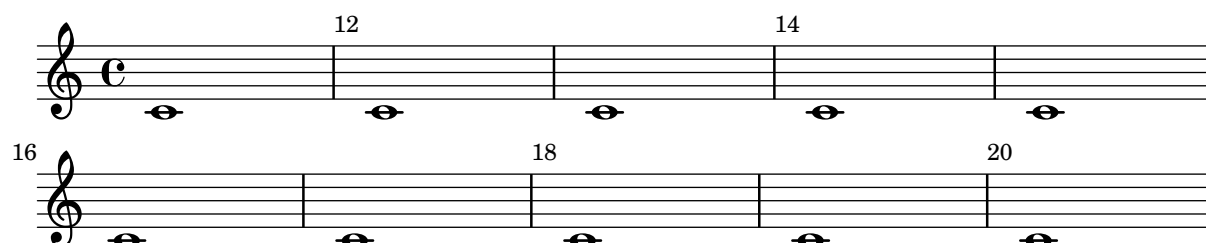
Setzen der Taktnummern in regelmäßigen Intervallen

Taktnummern können in regelmäßigen Intervallen gesetzt werden, indem man die Eigenschaft `barNumberVisibility` definiert. In diesem Beispiel werden die Taktnummern jeden zweiten Takt gesetzt, außer am Ende einer Zeile.

```

\relative c' {
  \override Score.BarNumber.break-visibility = #end-of-line-invisible
  \set Score.currentBarNumber = 11
  % Print a bar number every second measure
  \set Score.barNumberVisibility = #(every-nth-bar-number-visible 2)
  c1 | c | c | c | c |
  \break
  c1 | c | c | c | c |
}

```



Printing bar numbers for broken measures

By default, a bar number of a broken measure is not repeated at the beginning of the new line. Use `first-bar-number-invisible-save-broken-bars` for `barNumberVisibility` to get a parenthesized BarNumber there.

```

\layout {
  \context {

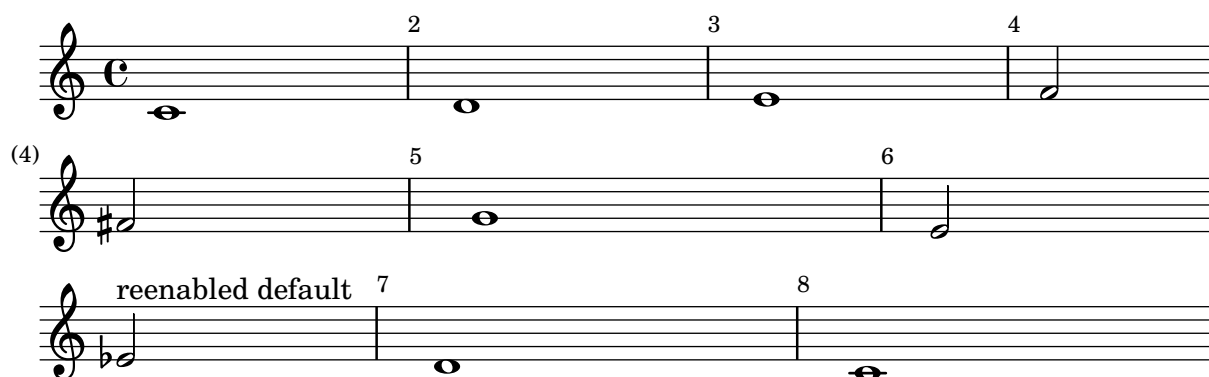
```

```

\Score
  barNumberVisibility = #first-bar-number-invisible-save-broken-bars
  \override BarNumber.break-visibility = ##(#f #t #t)
}
}

\relative c' {
  c1 | d | e | f2 \break
  fis2 | g1 | e2 \break
  <>^"reenabled default"
  % back to default -
  % \unset Score.barNumberVisibility would do so as well
  \set Score.barNumberVisibility =
    #first-bar-number-invisible-and-no-parenthesized-bar-numbers
  es2 | d1 | c
}

```



Setzen von Taktnummern in Kästen oder Kreisen

Taktnummern können auch in Boxen oder Kreisen gesetzt werden.

```

\relative c' {
  % Center bar numbers except at the beginning of a staff.
  \override Score.BarNumber.self-alignment-X =
    #(break-alignment-list CENTER CENTER 0.3)

  % Prevent bar numbers at the end of a line and permit them elsewhere.
  \override Score.BarNumber.break-visibility = #end-of-line-invisible

  \set Score.barNumberVisibility = #(every-nth-bar-number-visible 4)

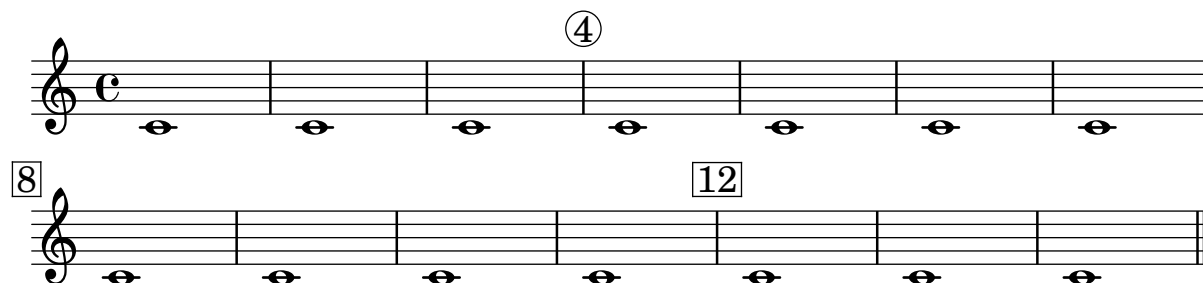
  % Increase the size of the bar number by 2.
  \override Score.BarNumber.font-size = 2

  % Draw a circle round the following bar number(s).
  \override Score.BarNumber.stencil
    = #(make-stencil-circler 0.1 0.25 ly:text-interface::print)
  \repeat unfold 7 { c1 } \break

  % Draw a box round the following bar number(s).
  \override Score.BarNumber.stencil
    = #(make-stencil-boxer 0.1 0.25 ly:text-interface::print)
}

```

```
\repeat unfold 7 { c1 } \bar "|."
}
```



Printing bar numbers using modulo-bar-number-visible

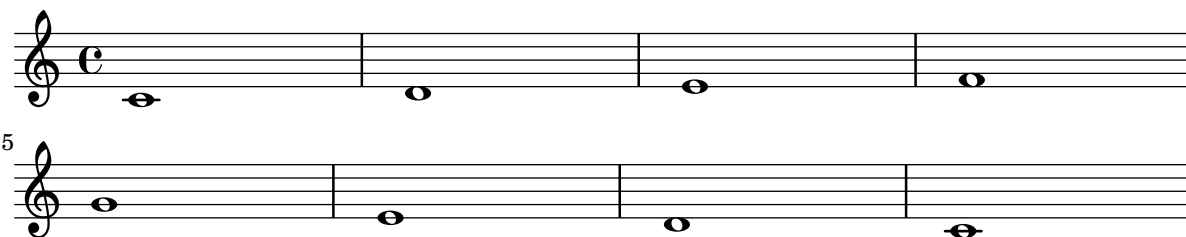
If the remainder of the division of the current bar number by the first argument of `modulo-bar-number-visible` equals its second argument, print a bar number.

This is useful to print the bar number at certain distances. Some examples:

- `(modulo-bar-number-visible 3 2)` → prints 2, 5, 8, ...
- `(modulo-bar-number-visible 4 2)` → prints 2, 6, 10, ...
- `(modulo-bar-number-visible 2 1)` → prints 3, 5, 7, ...
- `(modulo-bar-number-visible 5 0)` → prints 5, 10, 15, ...

```
\layout {
  \context {
    \Score
    \override BarNumber.break-visibility = ##(#f #t #t)
    barNumberVisibility = #(modulo-bar-number-visible 5 0)
  }
}
```

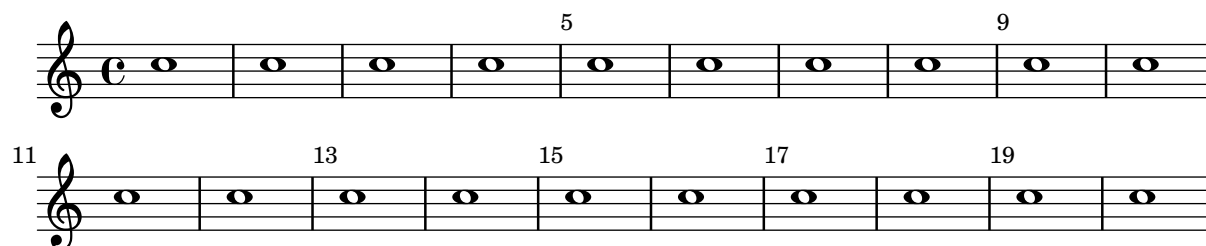
```
\relative c' {
  c1 | d | e | f \break
  g1 | e | d | c
}
```



Printing bar numbers with changing regular intervals

Using the `set-bar-number-visibility` context function, bar number intervals can be changed.

```
\relative c' {
  \override Score.BarNumber.break-visibility = #end-of-line-invisible
  \context Score \applyContext #(set-bar-number-visibility 4)
  \repeat unfold 10 c'1
  \context Score \applyContext #(set-bar-number-visibility 2)
  \repeat unfold 10 c
}
```



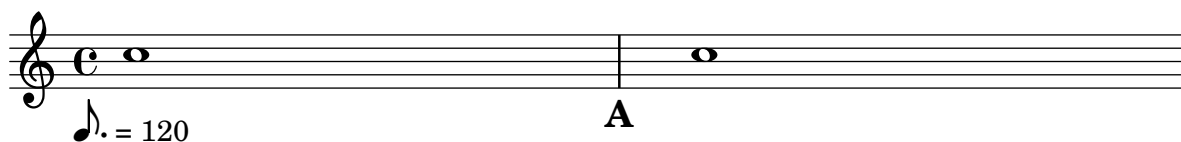
Metronom- und Übungszeichen unter das System setzen

Normalerweise werden Metronom- und Übungszeichen über dem Notensystem ausgegeben. Um sie unter das System zu setzen, muss die `direction`-Eigenschaft von `MetronomeMark` oder `RehearsalMark` entsprechend verändert werden.

```
\layout {
  ragged-right = ##f
}

{
  % Metronome marks below the staff
  \override Score.MetronomeMark.direction = #DOWN
  \tempo 8. = 120
  c''1

  % Rehearsal marks below the staff
  \override Score.RehearsalMark.direction = #DOWN
  \mark \default
  c''1
}
```



Printing music with different time signatures

In the following snippet, two parts have a completely different time signature, yet remain synchronized.

The bar lines can no longer be printed at the `Score` level; to allow independent bar lines in each part, the `Default_barline_engraver` and `Timing_translator` are moved from the `Score` context to the `Staff` context.

If bar numbers are required, the `Bar_number_engraver` should also be moved, since it relies on properties set by the `Timing_translator`; a `\with block` can be used to add bar numbers to the relevant staff.

```
global = {
  \time 3/4 s2.*3 \break
  s2.*3
}
```

```
\layout {
  \context {
    \Score
```

```

\remove "Timing_translator"
\remove "Bar_number_engraver"
\override SpacingSpanner.uniform-stretching = ##t
\override SpacingSpanner.strict-note-spacing = ##t
proportionalNotationDuration = #1/64
}
\context {
  \Staff
  \consists "Timing_translator"
}
\context {
  \Voice
  \remove "Forbid_line_break_engraver"
  tupletFullLength = ##t
}
}

Bassklarinette = \new Staff \with {
  \consists "Bar_number_engraver"
  barNumberVisibility = #(every-nth-bar-number-visible 2)
  \override BarNumber.break-visibility = #end-of-line-invisible
} <<
\global
{
  \clef treble
  \time 3/8 d''4. |
  \time 3/4 r8 des''2( c''8) |
  \time 7/8 r4. ees''2 ~ |
  \time 2/4 \tupletUp \tuplet 3/2 { ees''4 r4 d''4 ~ } |
  \time 3/8 \tupletUp \tuplet 4/3 { d''4 r4 } |
  \time 2/4 e''2 |
  \time 3/8 es''4. |
  \time 3/4 r8 d''2 r8 |
}
>>

Perkussion = \new StaffGroup <<
\new Staff <<
\global
{
  \clef percussion
  \time 3/4 r4 c'2 ~ |
  c'2. |
  R2. |
  r2 g'4 ~ |
  g'2. ~ |
  g'2. |
}
>>
\new Staff <<
\global {
  \clef percussion

```

```

\time 3/4 R2. |
g'2. ~ |
g'2. |
r4 g'2 ~ |
g'2 r4 |
g'2. |
}
>>
>>

```

```

\score {
  <<
    \Bassklarinette
    \Perkussion
  >>
}

```

The musical score is written for two staves: Bass Clarinet and Percussion. The time signature is 3/4. The score is divided into three systems. The first system shows a melody in the Bass Clarinet with a 2-measure rest, followed by a 4-measure rest, and then a 4-measure melody. The Percussion part has a 4-measure rest. The second system starts with a 3-measure rest in the Bass Clarinet, followed by a 4-measure melody. The Percussion part has a 4-measure rest. The third system starts with an 8-measure rest in the Bass Clarinet, followed by a 4-measure melody. The Percussion part has a 4-measure rest.

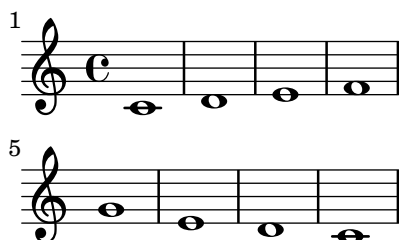
Setzen der Taktnummer für den ersten Takt

Standardmäßig wird die erste Taktzahl einer Partitur nicht gesetzt, wenn sie weniger oder gleich '1' ist. Indem man `barNumberVisibility` auf `all-bar-numbers-visible` setzt, kann eine belie-

biges Taktzahl für den ersten und die folgenden Takte gesetzt werden. Eine leere Taktlinie muss jedoch vor der ersten Note eingefügt werden, damit das funktioniert.

```
\paper {
  line-width = 50\mm
}

\relative c' {
  \set Score.barNumberVisibility = #all-bar-numbers-visible
  c1 | d | e | f \break
  g1 | e | d | c
}
```



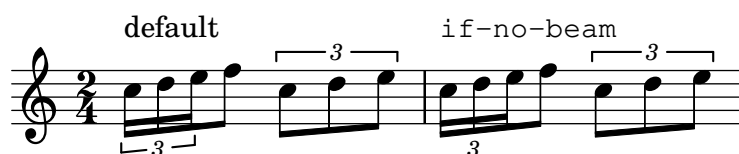
Printing tuplet brackets on the note head side

Whichever option you choose for controlling the tuplet bracket visibility, it will show or hide the tuplet bracket irrespectively of tuplet bracket placement (stem side or note head side). However, when placing the tuplet bracket on the note head side some authors recommend always printing the tuplet bracket. The option `visible-over-note-heads` can be used to achieve this.

```
music = \relative c'' {
  \tupletNeutral \tuplet 3/2 { c16[ d e ] f8]
  \tupletUp \tuplet 3/2 { c8 d e }
}

\new Voice {
  \relative c' {
    \override TextScript.staff-padding = #2.5

    \time 2/4
    \override TupletBracket.visible-over-note-heads = ##t
    \override Score.TextMark.non-musical = ##f
    <>\markup "default" \music
    \override TupletBracket.bracket-visibility = #'if-no-beam
    <>\markup \typewriter "if-no-beam" \music
  }
}
```



Globale Umdefinition von Verzierungsnoten

Die globalen Standardeinstellungen für Verzierungsnoten werden in den Variablen `startGraceMusic`, `stopGraceMusic`, `startAcciaccaturaMusic`,

`stopAcciaccaturaMusic`, `startAppoggiaturaMusic` und `stopAppoggiaturaMusic` gespeichert, die in der Datei `ly/grace-init.ly` definiert sind. Wenn man sie undefiniert, können andere Effekte erreicht werden.

```
startAcciaccaturaMusic = {
  <>(
    \override Flag.stroke-style = "grace"
    \slurDashed
  )
}
```

```
stopAcciaccaturaMusic = {
  \revert Flag.stroke-style
  \slurSolid
  <>
}
```

```
\relative c' {
  \acciaccatura d8 c1
}
```

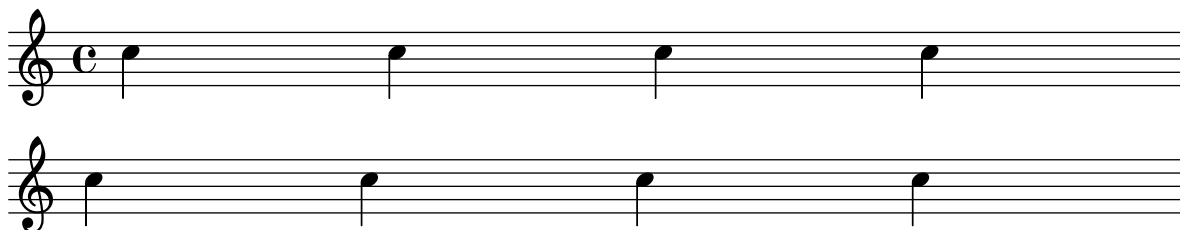


Entfernung von Taktnummern in einer Partitur

Taktnummern können vollkommen aus den Noten entfernt werden, indem man den `Bar_number_engraver` aus dem Score-Kontext entfernt.

```
\layout {
  \context {
    \Score
    \omit BarNumber
    % or:
    % \remove "Bar_number_engraver"
  }
}
```

```
\relative c' {
  c4 c c c \break
  c4 c c c
}
```



Pausenstile

Pausen können in verschiedenen Stilen dargestellt werden.

```
restsA = {
```

```

r\maxima r\longa r\breve r1 r2 r4 r8 r16 s32
s64 s128 s256 s512 s1024 s1024
}
restsB = {
  r\maxima r\longa r\breve r1 r2 r4 r8 r16 r32
  r64 r128 r256 r512 r1024 s1024
}

\new Staff \relative c {
  \omit Score.TimeSignature
  \cadenzaOn

  \override Staff.Rest.style = #'mensural
  <>^\markup \typewriter { mensural } \restsA \bar "" \break

  \override Staff.Rest.style = #'neomensural
  <>^\markup \typewriter { neomensural } \restsA \bar "" \break

  \override Staff.Rest.style = #'classical
  <>^\markup \typewriter { classical } \restsB \bar "" \break

  \override Staff.Rest.style = #'z
  <>^\markup \typewriter { z-style } \restsB \bar "" \break

  \override Staff.Rest.style = #'default
  <>^\markup \typewriter { default } \restsB \bar "" \break
}

```

Standard-Balkenwerte rückgängig machen

Um Balken im 12/8-Takt als 3-4-3-2 zu gruppieren, muss man zuerst die Standardwerte für die Balken im 12/8-Takt rückgängig machen und dann die neuen Werte setzen:

```
\relative c' {
```

```

\time 12/8

% Default beaming
a8 a a a a a a a a a a

% Set new values for beam endings
\set Score.beatStructure = 3,4,3,2
a8 a a a a a a a a a a
}

```



Rhythmic slashes

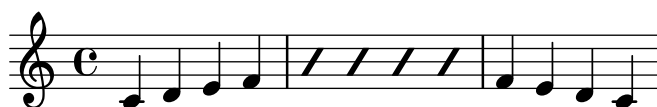
In „simple“ lead-sheets, sometimes no actual notes are written. Instead, only „rhythmic patterns“ and chords above the measures are notated to represent the structure of a song. Such a feature can be useful while creating or transcribing the structure of a song, or when sharing lead sheets with guitarists or jazz musicians.

```

startPat = {
  \improvisationOn
  \omit Stem
}
stopPat = {
  \improvisationOff
  \undo \omit Stem
}

\new Voice \with {
  \consists Pitch_squash_engraver
} {
  c'4 d' e' f' |
  \startPat
  4 4 4 4 |
  \stopPat
  f'4 e' d' c'
}

```



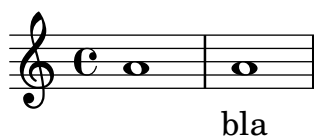
Skips in lyric mode

The ‘s’ syntax for skips is only available in note mode and chord mode. In other situations, for example, when entering lyrics, using the \skip command is recommended.

```

<<
  \relative c'' { a1 | a }
  \new Lyrics \lyricmode { \skip1 bla1 }
>>

```



Skips in lyric mode (2)

Although 's' skips cannot be used in `\lyricmode` (it is taken to be a literal „s“, not a space), double quotes (") or underscores (_) are available.

```
<<
  \relative c'' { a4 b c d }
  \new Lyrics \lyricmode { a4 "" _ gap }
>>
```



Stemlets

In some notational conventions beams are allowed to extend over rests. Depending on preference, these beams may drop 'stemlets' to help the eye appreciate the rhythm better, and in some modern music the rest itself is omitted and only the stemlet remains.

This snippet shows a progression from traditional notation, to beams over the rest, to stemlets over the rest, to stemlets alone. Stemlets are generated by overriding the `stemlet-length` property of `Stem`, and rests are hidden by using `\hide`.

Some `\markup` elements are included in the source to highlight the different notations.

```
\paper {
  ragged-right = ##f
}

{
  c'16^\markup { traditional } d' r f'
  g'16[^\markup \column { "beams" "over rests" } f' r d']

  % N.B. use Score.Stem to set for the whole score.
  \override Staff.Stem.stemlet-length = #0.75

  c'16[^\markup \column { "stemlets" "over rests" } d' r f']
  g'16[^\markup \column { "stemlets" "and no rests" } f'
  \once \hide Rest
  r16 d']
}
```



Bebalkung nach Taktschlag

Sekundäre Balken können in die Richtung gesetzt werden, die ihrer rhythmischen Zugehörigkeit entspricht. Der erste Balken ist zusammengefasst (Standard), der zweite Sechszehntelbalken zeigt den Taktschlag an.

```
\relative c' ' {
  \time 6/8
  a8. a16 a a
  \set strictBeatBeaming = ##t
  a8. a16 a a
}
```



Subdividing beams

The beams of consecutive 16th (or shorter) notes are, by default, not subdivided. That is, the beams of more than two stems stretch over the entire group of notes without a break. This behavior can be modified to subdivide the beams into sub-groups by setting the property `subdivideBeams` to `#t`. When set, beams are subdivided at (rhythmic) intervals to match the metric value of the subdivision.

Using the properties `beamMinimumSubdivision` and `beamMaximumSubdivision` it is possible to configure the limits of automatic beam subdivision, namely the minimum and maximum rhythmic lengths at which beamlets are removed. The default values are 0 for the former and `+inf.0` for the latter, making LilyPond subdivide beams as much as possible.

There are two special cases to consider.

- If the numerator of `beamMaximumSubdivision` is not a power of 2, the rhythmic lengths considered for subdivision are `beamMaximumSubdivision` divided by powers of 2 that stay greater than or equal to `beamMinimumSubdivision`.
- If `beamMaximumSubdivision` is smaller than `beamMinimumSubdivision`, the depth of beam subdivisions is limited by `beamMaximumSubdivision`, but not the frequency and rhythmic intervals, therefore possibly deviating from the correct, expected metric value.

If `respectIncompleteBeams` is set to `#t`, incomplete subdivisions with more than two stems are treated as an 'extension' of the previous subdivision group, i.e., the length of the previous subdivision group gets extended to also cover the incomplete subdivision. If set to `#f` (which is the default), a new subdivision group gets started instead.

```
\relative c' ' {
  \time 1/4

  <>^"default"
  c32 c c c c c c c

  <>^"with subdivision"
  \set subdivideBeams = ##t
  c32 c c c c c c c

  <>^"min 1/8"
  \once \set beamMinimumSubdivision = #1/8
  c32 c c c c c c c
```

```

<>~"max 1/16"
\once \set beamMaximumSubdivision = #1/16
c32 c c c c c c c c

<>~"max 3/8"
\once \set beamMaximumSubdivision = #3/8
\repeat unfold 16 c64

<>~"min 1/32, max 1/64"
% Set maximum beam subdivision interval to 1/64 to limit
% subdivision depth, despite not being metrically correct.
\once \set beamMinimumSubdivision = #1/32
\once \set beamMaximumSubdivision = #1/64
\repeat unfold 32 c128
\break

<>~"beams with incomplete subdivisions"
c32 c c c c c c c r32
c32 c c c c c r16.

<>~\markup { "the same with"
  \typewriter { "respectIncomplete=#t" } }
\set respectIncompleteBeams = ##t
% The incomplete subgroup extends the completed subgroup.
c32 c c c c c c c r32
% No visual change since we have only two stems in the
% incomplete subgroup.
c32 c c c c c r16.
}

```

The image displays three staves of musical notation illustrating different beam settings. The first staff (4/4) shows four measures: 'default' (8 eighth notes), 'with subdivision' (8 eighth notes with stems), 'min 1/8' (8 eighth notes with stems), and 'max 1/16' (8 eighth notes with stems). The second staff (5/4) shows two measures: 'max 3/8' (16 eighth notes) and 'min 1/32, max 1/64' (32 sixteenth notes). The third staff (7/4) shows four measures: 'beams with incomplete subdivisions' (8 eighth notes with stems), 'the same with respectIncomplete=#t' (8 eighth notes with stems), and two more measures showing the same notation with different beam settings.

Tam-tam example

A tam-tam example, entered with ,tt'.

```

#(define mydrums '((tamtam default #f 0)))

\new DrumStaff \with { instrumentName = #"Tamtam" }

\drummode {
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)
}


```

```

\override Staff.StaffSymbol.line-positions = #'( 0 )
\override Staff.BarLine.bar-extent = #'(-1.5 . 1.5)

tt 1 \pp \laissezVibrer
}

```

Tamtam 

Tambourine example

A tambourine example, entered with ,tamb'.

```

#(define mydrums '((tambourine default #f 0)))

\new DrumStaff \with { instrumentName = #"Tambourine" }

\drummode {
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)
  \override Staff.StaffSymbol.line-positions = #'( 0 )
  \override Staff.BarLine.bar-extent = #'(-1.5 . 1.5)

  \time 6/8
  tamb8. 16 8 8 8 8 |
  tamb4. 8 8 8 |
  % The trick with the scaled duration and the shorter rest
  % is neccessary for the correct ending of the trill-span!
  tamb2.*5/6 \startTrillSpan s8 \stopTrillSpan |
}

```

Tambourine 

Three-sided box

This example shows how to add a markup command to get a three-sided box around some text (or other markup).

```

% New command to add a three-sided box, with sides north, west, and south.
% Based on the `box-stencil` command defined in `scm/stencil.scm`.
% Note that ";;" is used to comment a line in Scheme.
#(define-public (NWS-box-stencil stencil thickness padding)
  "Add a box around STENCIL, producing a new stencil."
  (let* ((x-ext (interval-widen (ly:stencil-extent stencil X) padding))
        (y-ext (interval-widen (ly:stencil-extent stencil Y) padding))
        (y-rule (make-filled-box-stencil (cons 0 thickness) y-ext))
        (x-rule (make-filled-box-stencil
                  (interval-widen x-ext thickness) (cons 0 thickness))))
    ;; (set! stencil (ly:stencil-combine-at-edge stencil X 1 y-rule padding))
    (set! stencil (ly:stencil-combine-at-edge stencil X LEFT y-rule padding))
    (set! stencil (ly:stencil-combine-at-edge stencil Y UP x-rule 0.0))
    (set! stencil (ly:stencil-combine-at-edge stencil Y DOWN x-rule 0.0))
    stencil))

```

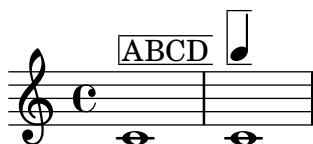
```
% The corresponding markup command, based on the `box` command defined
% in `scm/define-markup-commands.scm`.
```

```
#(define-markup-command (NWS-box layout props arg) (markup?)
  #:properties ((thickness 0.1) (font-size 0) (box-padding 0.2))
  "Draw a box round ARG."
```

Look at THICKNESS, BOX-PADDING, and FONT-SIZE properties to determine line thickness and padding around the markup."

```
(let ((pad (* (magstep font-size) box-padding))
      (m (interpret-markup layout props arg)))
  (NWS-box-stencil m thickness pad)))
```

```
\relative c' {
  c1^\markup { \NWS-box ABCD }
  c1^\markup { \NWS-box \note {4} #1.0 }
}
```



Time signature in brackets

The time signature can be enclosed within brackets.

```
\relative c' {
  \override Staff.TimeSignature.stencil = #(lambda (grob)
    (bracketify-stencil (ly:time-signature::print grob) Y 0.1 0.2 0.1))
  \time 2/4
  a4 b8 c
}
```



Time signature in parentheses

The time signature can be enclosed within parentheses.

```
\relative c' {
  \override Staff.TimeSignature.stencil = #(lambda (grob)
    (parenthesize-stencil (ly:time-signature::print grob) 0.1 0.4 0.4 0.1))
  \time 2/4
  a4 b8 c
}
```



Time signature printing only the numerator as a number (instead of the fraction)

Sometimes, a time signature should not print the whole fraction (for example, 7/4), but only the numerator (digit 7 in this case). This can be easily done by using `\override Staff.TimeSignature.style = #'single-number` to change the style permanently. By using `\revert Staff.TimeSignature.style`, this setting can be reversed. To apply the single-number style to only one time signature, use `\tweak`.

```
\relative c' {
  \time 3/4
  c4 c c
  % Change the style permanently
  \override Staff.TimeSignature.style = #'single-number
  \time 2/4
  c4 c
  \time 3/4
  c4 c c
  % Revert to default style:
  \revert Staff.TimeSignature.style
  \time 2/4
  c4 c
  % single-number style only for the next time signature
  \tweak style #'single-number \time 5/4
  c4 c c c c
  \time 2/4
  c4 c
}
```



Veränderung des Layouts von Verzierungen innerhalb der Noten

Das Layout von Verzierungsausdrücken kann in der Musik verändert werden mit den Funktionen `add-grace-property` und `remove-grace-property`. Das folgende Beispiel definiert die Richtung von Hälsen (Stem) für diese Verzierung, sodass die Hälse nicht immer nach unten zeigen, und ändert den Standardnotenkopf in ein Kreuz.

```
\relative c' {
  \new Staff {
    $(remove-grace-property 'Voice 'Stem 'direction)
    $(add-grace-property 'Voice 'NoteHead 'style 'cross)
    \new Voice {
      \acciaccatura { f16 } g4
      \grace { d16 e } f4
      \appoggiatura { f,32 g a } e2
    }
  }
}
```



User-defined time signatures

New time signature styles can be defined. The time signature in the second measure is printed upside down in both staves.

```
#(add-simple-time-signature-style 'topsy-turvy
  (lambda (fraction)
    (make-rotate-markup 180 (make-compound-meter-markup fraction))))
```

```
<<
\new Staff {
  \time 3/4 f'2.
  \override Score.TimeSignature.style = #'topsy-turvy
  \time 3/4 R2. \bar "|"
}
\new Staff {
  R2. e''
}
>>
```



Using alternative flag styles

Alternative shapes for flags on eighth and shorter notes can be displayed by overriding the stencil property of Flag. LilyPond provides the following functions: `modern-straight-flag`, `old-straight-flag`, and `flat-flag`. Use `\revert` to restore the default shape.

To get stacked (i.e., vertically more compact) flags, call the command `\flagStyleStacked`, which can be reset with `\flagStyleDefault`.

Overriding the Flag stencil does not change how flag elements are positioned vertically. This is especially noticeable for flat flags: LilyPond doesn't dynamically adjust the vertical gaps between flag elements in the same way as it does for beams. A possible solution to harmonize the appearance is to replace flat flags with half beams, as shown in the second staff; however, this can't be done automatically. In the code of this snippet, such half beams are entered with `@` as a prefix, for example `@c8`.

Be aware that half beams are *not* Flag grobs. This means in particular that modifying Flag properties won't have any effect on them (you have to use Beam properties instead), and properties for their associated Stem grob will also behave beam-like.

```
"@" =
#(define-music-function (music) (ly:music?)
  #{ \set stemLeftBeamCount = 0 $music [] #})

testnotes = {
  \autoBeamOff
  c8 d16 e''32 f64 \acciaccatura { g,,,8 } a128 b
}

\relative c' {
```

```

\override TextScript.staff-padding = 6
\time 1/4
  <>^"default" \testnotes
\override Flag.stencil = #modern-straight-flag
  <>_"modern straight" \testnotes
\override Flag.stencil = #old-straight-flag
  <>^"old straight" \testnotes
\override Flag.stencil = #flat-flag
  <>_"flat" \testnotes
\revert Flag.stencil

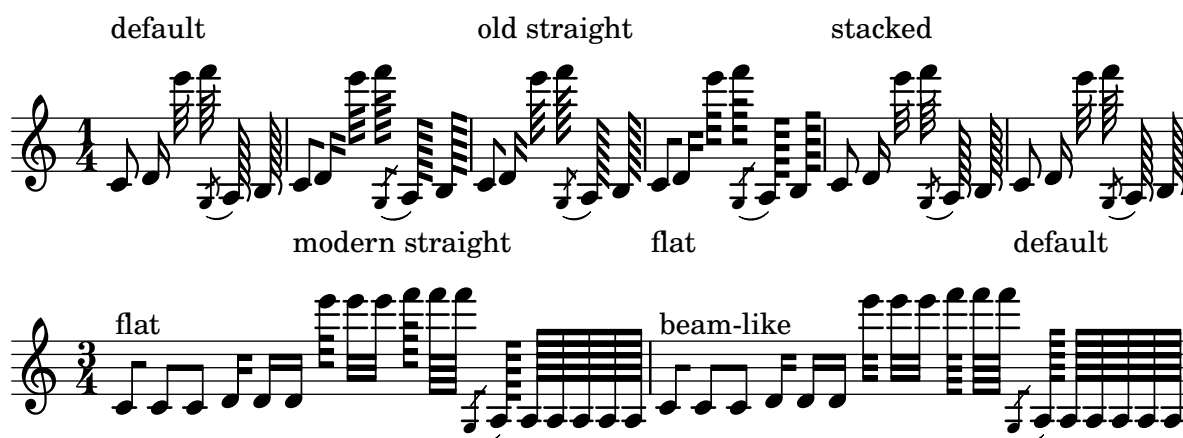
\flagStyleStacked
  <>^"stacked" \testnotes
\flagStyleDefault
  <>_"default" \testnotes
}

\relative c' {
  \time 3/4
  \override Flag.stencil = #flat-flag

  <>^"flat" c8 c[ c] d16 d[ d] e''32 e[ e] f64 f[ f]
    \acciaccatura { g,,,8 } a128 a[ a a a a]
  <>^"beam-like" @c8 c[ c] @d16 d[ d] @e''32 e[ e] @f64 f[ f]
    \acciaccatura { g,,,8 } @a128 a[ a a a a]
}

\layout {
  indent = 0
  \context {
    \Score
    \override NonMusicalPaperColumn.line-break-permission = ##f
  }
}

```



Using grace note slashes with normal heads

The slash through the stem found in acciaccaturas can be applied in other situations.

```

\relative c' {

```

```
\override Flag.stroke-style = "grace"
c8( d2) e8( f4)
}
```



Überbindungen für Arpeggio benutzen

Überbindungen werden teilweise benutzt, um Arpeggios zu notieren. In diesem Fall stehen die übergebundenen Noten nicht unbedingt hintereinander. Das Verhalten kann erreicht werden, indem die `tieWaitForNote`-Eigenschaft auf `#t` gesetzt wird. Diese Funktion ist auch sinnvoll, um etwa ein Tremolo mit einem Akkord zu überbinden, kann aber prinzipiell auch für normale Überbindungen eingesetzt werden

```
\relative c' {
  \set tieWaitForNote = ##t
  \grace { c16[ ~ e ~ g] ~ } <c, e g>2
  \repeat tremolo 8 { c32 ~ c' ~ } <c c,>1
  e8 ~ c ~ a ~ f ~ <e' c a f>2
  \tieUp
  c8 ~ a
  \tieDown
  \tieDotted
  g8 ~ c g2
}
```



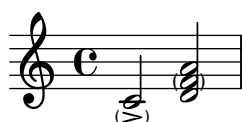
3 Expressive marks

See also Abschnitt “Expressive marks” in *Notationsreferenz*.

Adding parentheses around an expressive mark or chordal note

The `\parenthesize` function is a special tweak that encloses objects in parentheses. The associated grob is `Parentheses`.

```
\relative c' {
  c2-\parenthesize ->
  \override Parentheses.padding = #0.1
  \override Parentheses.font-size = #-4
  <d \parenthesize f a>2
}
```



Adding timing marks to long glissandi

Skipped beats in very long glissandi are sometimes indicated by timing marks consisting of stems without noteheads. Such stems can also be used to carry intermediate expression markings.

If the stems do not align well with the glissando, they may need to be repositioned slightly.

```
glissandoSkipOn = {
  \override NoteColumn.glissando-skip = ##t
  \hide NoteHead
  \override NoteHead.no-ledgers = ##t
}
```

```
glissandoSkipOff = {
  \revert NoteColumn.glissando-skip
  \undo \hide NoteHead
  \revert NoteHead.no-ledgers
}
```

```
\relative c' {
  r8 f8\glissando \glissandoSkipOn f4 g a |
  a8\noBeam \glissandoSkipOff a8
  r8 f8\glissando \glissandoSkipOn g4 a8 \glissandoSkipOff a8 |
  r4 f\glissando\< \glissandoSkipOn a4\> \glissandoSkipOff b8\! r |
}
```



Adjusting slur positions vertically

Using `\override Slur.positions` it is possible to set the vertical position of the start and end points of a slur to absolute values (or rather, forcing LilyPond's slur algorithm to consider these values as desired). In many cases, this means a lot of trial and error until good values are found.

You probably have tried the `\offset` command next just to find out that it doesn't work for slurs, emitting a warning instead.

The code in this snippet allows you to tweak the vertical start and end positions by specifying *relative* changes, similar to `\offset`.

Syntax: `\offsetPositions #'(dy1 . dy2)`

```
offsetPositions =
#(define-music-function (offsets) (number-pair?)
  #{
    \once \override Slur.control-points =
      #(\lambda (grob)
        (match-let (((_ . y1) _ _ (_ . y2))
          (ly:slur::calc-control-points grob))
          ((off1 . off2) offsets))
        (set! (ly:grob-property grob 'positions)
          (cons (+ y1 off1) (+ y2 off2)))
        (ly:slur::calc-control-points grob)))
    #})

\relative c' {
  c4(^"default" c, d2)
  \offsetPositions #'(0 . 1)
  c'4(^"(0 . 1)" c, d2)
  \offsetPositions #'(0 . 2)
  c'4(^"(0 . 2)" c, d2)
  \bar "||"
  g4(^"default" a d'2)
  \offsetPositions #'(1 . 0)
  g,,4(^"(1 . 0)" a d'2)
  \offsetPositions #'(2 . 0)
  g,,4(^"(2 . 0)" a d'2)
}
```



Das Aussehen von unbestimmten Glissandi anpassen

Die `shortest-duration-space`-Eigenschaft kann verändert werden, um das Aussehen von unbestimmten Glissandi anzupassen.

```
\relative c' {
  \override Score.SpacingSpanner.shortest-duration-space = 4.0
  c2-\bendAfter 5
  c2-\bendAfter -4.75
  c2-\bendAfter 8.5
  c2-\bendAfter -6
}
```



Aligning the ends of hairpins to NoteColumn directions

The ends of hairpins may be aligned to the LEFT, CENTER, or RIGHT of NoteColumn grobs by overriding the property `endpoint-alignments`, which is a pair of numbers representing the left and right ends of the hairpin. `endpoint-alignments` are expected to be directions (either -1, 0 or 1). Other values will be transformed with a warning. The right end of a hairpin terminating at a rest is not affected, always ending at the left edge of the rest.

```
{
  c'2\< <c' d'>\! |
  \override Hairpin.endpoint-alignments = #'(1 . -1)
  c'2\< <c' d'>\! |
  \override Hairpin.endpoint-alignments = #'(,LEFT . ,CENTER)
  c'2\< <c' d'>\! |
}
```



Alternative breve notes

Breve notes are also available with two vertical lines on each side of the notehead instead of one line and in baroque style.

```
\relative c' ' {
  \time 4/2
  c\breve |
  \override Staff.NoteHead.style = #'altdefault
  b\breve
  \override Staff.NoteHead.style = #'baroque
  b\breve
  \revert Staff.NoteHead.style
  a\breve
}
```



Asymmetric slurs

Slurs can be made asymmetric to match an asymmetric pattern of notes better.

```
slurNotes = { d,8( a' d f a f' d, a) }
```

```
\relative c' {
  \stemDown
  \slurUp
  \slurNotes
  \once \override Slur.eccentricity = #3.0
  \slurNotes
}
```



Breathing signs

Breathing signs are available in different tastes: commas (default), ticks, vees and „railroad tracks“ (caesura).

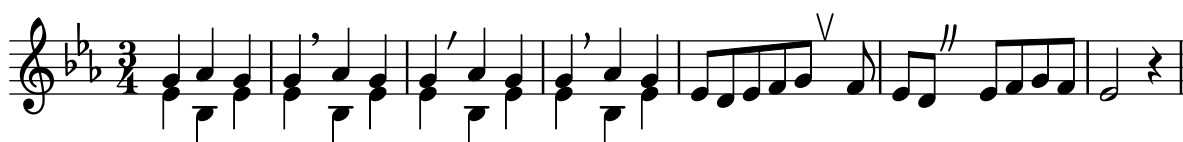
```
\new Staff \relative c'' {
  \key es \major
  \time 3/4
  % this bar contains no \breathe
  << { g4 as g } \ { es4 bes es } >> |
  % Modern notation:
  % by default, \breathe uses the rcomma, just as if saying:
  % \override BreathingSign.text =
  %   #(make-musicglyph-markup "scripts.rcomma")
  << { g4 as g } \ { es4 \breathe bes es } >> |

  % rvarcomma and lvarcomma are variations of the default rcomma
  % and lcomma
  % N.B.: must use Staff context here, since we start a Voice below
  \override Staff.BreathingSign.text =
    \markup { \musicglyph "scripts.rvarcomma" }
  << { g4 as g } \ { es4 \breathe bes es } >> |

  % raltcomma and laltcomma are alternative variations of the
  % default rcomma and lcomma
  \override Staff.BreathingSign.text =
    \markup { \musicglyph "scripts.raltcomma" }
  << { g4 as g } \ { es4 \breathe bes es } >> |

  % vee
  \override BreathingSign.text =
    \markup { \musicglyph "scripts.uupbow" }
  es8[ d es f g] \breathe f |

  % caesura
  \override BreathingSign.text =
    \markup { \musicglyph "scripts.caesura.curved" }
  es8[ d] \breathe es[ f g f] |
  es2 r4 \bar "||"
}
```



Broken crescendo hairpin

In order to make parts of a crescendo hairpin invisible, the following method is used: A white rectangle is drawn on top of the respective part of the crescendo hairpin, making it invisible. The rectangle is defined as a text markup.

The markup command `with-dimensions` tells LilyPond to consider only the bottom edge of the rectangle when spacing it against the hairpin. The property `staff-padding` prevents the rectangle from fitting between the hairpin and staff.

Make sure the hairpin is in a lower layer than the text markup to draw the rectangle over the hairpin.

```
\relative c' {
  <<
  {
    \dynamicUp
    r2 r16 c'8.\pp r4
  }
  \\\
  {
    \override DynamicLineSpanner.layer = #0
    des,2\mf\< ~
    \override TextScript.layer = #2
    \once\override TextScript.staff-padding = #6
    \once\override TextScript.vertical-skylines = #'()
    des16_\markup \with-dimensions #'(2 . 7) #'(0 . 0)
      \with-color #white
      \filled-box #'(2 . 7) #'(0 . 2) #0
    r8. des4 ~ des16->\sff r8.
  }
  >>
}
```



Caesura („railtracks“) with fermata

A caesura is sometimes denoted by a double „railtracks“ breath mark with a fermata sign positioned above. This snippet shows an optically pleasing combination of railtracks and fermata.

```
\relative c' {
  c2.
  % construct the symbol
  \override BreathingSign.text = \markup {
    \override #'(direction . 1)
    \override #'(baseline-skip . 1.8)
    \dir-column {
      \translate #'(0.155 . 0)
      \center-align \musicglyph "scripts.caesura.curved"
      \center-align \musicglyph "scripts.ufermata"
    }
  }
```

```

    }
  }
  \breathe c4
  % set the breath mark back to normal
  \revert BreathingSign.text
  c2. \breathe c4
  \bar "|."
}

```



Center text below hairpin dynamics

This example provides a function to typeset a hairpin (de)crescendo with some additional text below it, such as „molto“ or „poco“. The added text will change the direction according to the direction of the hairpin. The Hairpin is aligned to a DynamicText grob.

The example also illustrates how to modify the way an object is normally printed, using some Scheme code.

```

hairpinWithCenteredText =
#(define-music-function (text) (markup?)
  #{
    \once \override Voice.Hairpin.after-line-breaking =
      #(lambda (grob)
        (let* ((stencil (ly:hairpin::print grob))
              (par-y (ly:grob-parent grob Y))
              (dir (ly:grob-property par-y 'direction))
              (staff-line-thickness
                (ly:output-def-lookup (ly:grob-layout grob)
                                      'line-thickness))
              (new-stencil
                (ly:stencil-aligned-to
                 (ly:stencil-combine-at-edge
                  (ly:stencil-aligned-to stencil X CENTER)
                  Y dir
                  (ly:stencil-aligned-to
                   (grob-interpret-markup
                    grob
                    (make-fontsize-markup
                     (magnification->font-size
                      (+ (ly:staff-symbol-staff-space grob)
                        (/ staff-line-thickness 2))))
                     text))
                  X CENTER))
                 X LEFT))
              (staff-space (ly:output-def-lookup
                            (ly:grob-layout grob) 'staff-space))
              (par-x (ly:grob-parent grob X))
              (dyn-text (grob::has-interface par-x
                                   'dynamic-text-interface))
              (dyn-text-stencil-x-length

```

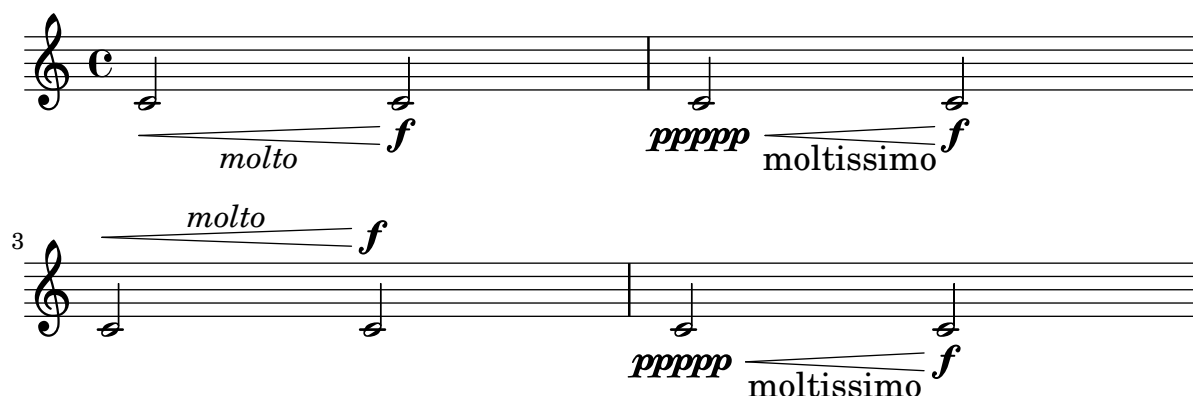
```

      (if dyn-text
        (interval-length
          (ly:stencil-extent
            (ly:grob-property par-x 'stencil) X))
          0))
      (x-shift
        (if dyn-text (- (+ staff-space dyn-text-stencil-x-length)
          (* 0.5 staff-line-thickness))
          0)))
      (ly:grob-set-property! grob 'Y-offset 0)
      (ly:grob-set-property! grob
        'stencil (ly:stencil-translate-axis
          new-stencil
            x-shift X))))
#})

hairpinMolto = \hairpinWithCenteredText \markup { \italic molto }
hairpinMore = \hairpinWithCenteredText \markup { \larger moltissimo }

\relative c' {
  \hairpinMolto c2\< c\f
  \hairpinMore c2\ppppp\< c\f
  \break
  \hairpinMolto c2^\< c\f
  \hairpinMore c2\ppppp\< c\f
}

```



Text und Strecker-Stile für Dynamik-Texte ändern

Der Text, der für Crescendo und Decrescendo gesetzt wird, kann geändert werden, indem man die Eigenschaften `crescendoText` und `decrescendoText` verändert. Der Stil des Streckers kann auch geändert werden, indem die `'style`-Eigenschaft des `DynamicTextSpanner` beeinflusst wird. Der Standardwert ist `'hairpin`, andere Möglichkeiten sind `'line`, `'dashed-line` und `'dotted-line`.

```

\relative c' {
  \set crescendoText = \markup { \italic { cresc. poco } }
  \set crescendoSpanner = #'text
  \override DynamicTextSpanner.style = #'dotted-line
  a2\< a
  a2 a
}

```

```

a2 a
a2 a\mf
}

```



Changing the appearance of a slur from solid to dotted or dashed

The appearance of slurs may be changed from solid to dotted or dashed.

```

\relative c' {
  c4( d e c)
  \slurDotted
  c4( d e c)
  \slurSolid
  c4( d e c)
  \slurDashed
  c4( d e c)
  \slurSolid
  c4( d e c)
}

```



Das Atemzeichen-Symbol verändern

Das Schriftzeichen für das Atemzeichen kann verändert werden, indem die Text-Eigenschaft des BreathingSign-Layoutobjekts mit einer beliebigen Textbeschriftung definiert wird.

```

\relative c' {
  c2
  \override BreathingSign.text =
    \markup { \musicglyph "scripts.rvarcomma" }
  \breathe
  d2
}

```



Changing the number of augmentation dots per note

The number of augmentation dots on a single note can be overridden by setting the dot-count property of the Dots grob.

```

\relative c' {
  c4.. a16 r2 |
  \override Dots.dot-count = 4
  c4.. a16 r2 |
}

```

```
\override Dots.dot-count = 0
c4.. a16 r2 |
\revert Dots.dot-count
c4.. a16 r2 |
}
```



Combining dynamics with markup texts

Some dynamics may involve text indications (such as „più *f*“ or „*p* subito“). These can be produced using a `\markup` block; the resulting object behaves like a `TextScript` grob.

See also „Combining dynamics with markup texts (2)“.

```
piuF = \markup { \italic più \dynamic f }
```

```
\markup \with-true-dimensions % work around a cropping issue
\score {
  \relative c'' {
    c2\f c-\piuF
  }
}
```



Combining dynamics with markup texts (2)

Some dynamics may involve text indications (such as „più *f*“ or „*p* subito“). These can be produced using the `make-dynamic-script` Scheme function; the resulting object behaves like a `DynamicText` grob.

See also „Combining dynamics with markup texts“.

```
piuF = #(make-dynamic-script
  #{ \markup { \normal-text \italic più \dynamic f } #})
```

```
\score {
  \relative c'' {
    c2\f c\piuF
  }
}
```



Moderne Glissandi

Ein modernes Glissando ohne eine Endnote kann gesetzt werden, indem eine Kadenz eingesetzt wird und die Endnote unsichtbar gemacht wird.

```
\relative c'' {
```

```

\time 3/4
\override Glissando.style = #'zigzag
c4 c
\cadenzaOn
c4\glissando
\hideNotes
c,,4
\unhideNotes
\cadenzaOff
\bar "|"
}

```



Controlling spanner visibility after a line break

The visibility of spanners which end on the first note following a line break is controlled by the after-line-breaking callback `ly:spanner::kill-zero-spanned-time`.

For objects such as glissandos and hairpins, the default behaviour is to hide the spanner after a break; disabling the callback will allow the left-broken span to be shown.

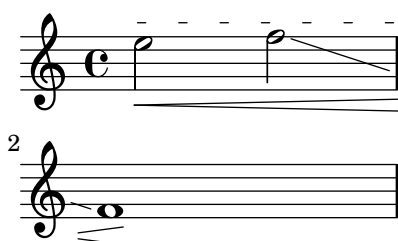
Conversely, spanners which are usually visible, such as text spans, can be hidden by enabling the callback.

```

\paper {
  line-width = 50\mm
}

\relative c' {
  \override Hairpin.to-barline = ##f
  \override Glissando.breakable = ##t
  % show hairpin
  \override Hairpin.after-line-breaking = ##t
  % hide text span
  \override TextSpanner.after-line-breaking =
    #ly:spanner::kill-zero-spanned-time
  e2\<\startTextSpan
  % show glissando
  \override Glissando.after-line-breaking = ##t
  f2\glissando
  \break
  f,1\!\stopTextSpan
}

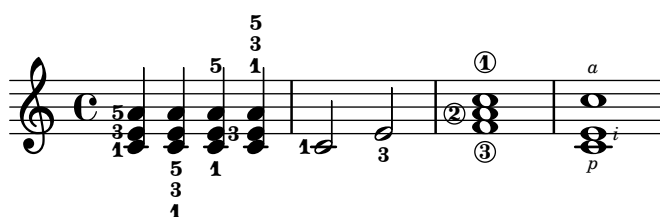
```



Position von Fingersatz in Akkorden kontrollieren

Die Position von Fingersatzzahlen kann exakt kontrolliert werden.

```
\relative c' {
  \set fingeringOrientations = #'(left)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down right up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(left)
  <c-1>2
  \set fingeringOrientations = #'(down)
  <e-3>2
  \set stringNumberOrientations = #'(up left down)
  <f\3 a\2 c\1>1
  \set strokeFingerOrientations = #'(down right up)
  <c\rightHandFinger 1 e\rightHandFinger 2 c'\rightHandFinger 4 >
}
```



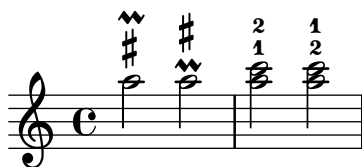
Die vertikale Anordnung von Beschriftungen kontrollieren

Die vertikale Anordnung von Beschriftungen wird mit der 'script-priority-Eigenschaft kontrolliert. Um so kleiner die Zahl, umso näher wird die Beschriftung in Bezug auf die Note gesetzt. In diesem Beispiel hat das TextScript-Objekt (das Kreuz) zuerst die niedrigste Priorität, wird also auch am niedrigsten in dem ersten Beispiel gesetzt. Im zweiten Fall hat der Praller (das Script) die niedrigste Priorität, darum wird er am nächsten zum System gesetzt. Wenn zwei Objekte die gleiche Priorität haben, wird ihre Reihenfolge anhand ihres Auftretens in der Quelldatei entschieden.

```
\relative c''' {
  \once \override TextScript.script-priority = -100
  a2^\prall^\markup { \sharp }

  \once \override Script.script-priority = -100
  a2^\prall^\markup { \sharp }

  \set fingeringOrientations = #'(up)
  <c-2 a-1>2
  <a-1 c\tweak script-priority -100 -2>2
}
```



Creating „real“ parenthesized dynamics

Although the easiest way to add parentheses to a dynamic mark is to use a `\markup` block, this method has a downside: the created objects behave like text markups and not like dynamics.

However, it is possible to create a similar object using the equivalent Scheme code (as described in the Notation Reference), combined with the `make-dynamic-script` function. This way, the markup is regarded as a dynamic and therefore remains compatible with commands such as `\dynamicUp` or `\dynamicDown`.

```
paren =
#(define-event-function (dyn) (ly:event?)
  (make-dynamic-script
    #{ \markup \concat {
      \normal-text \italic \fontsize #2 (
        \pad-x #0.2 #(ly:music-property dyn 'text)
      )
    }
    #}))

\relative c'' {
  c4\paren\f c c \dynamicUp c\paren\p
}
```



Einen Doppelschlag mit Vorhalt erstellen

Einen Doppelschlag mit Vorhalt zu erstellen, wobei die untere Note das Vorzeichen benutzt, erfordert einige Einstellungsänderungen. Die `outside-staff-priority`-Eigenschaft muss auf falsch (`#f`) gesetzt werden, weil sie sonst über die Eigenschaft `avoid-slur` property dominieren würde. Der Wert von `halign` wird benutzt, um den Doppelschlag horizontal zu positionieren.

```
\relative c'' {
  \after 2*2/3 \turn c2( d4) r |
  \after 4 \turn c4.( d8)
  \after 4
  {
    \once \set suggestAccidentals = ##t
    \once \override AccidentalSuggestion.outside-staff-priority = ##f
    \once \override AccidentalSuggestion.avoid-slur = #'inside
    \once \override AccidentalSuggestion.font-size = -3
    \once \override AccidentalSuggestion.script-priority = -1
    \once \hideNotes
    cis8\turn \noBeam
  }
}
```



```
d4.( e8)
}
```



Arpeggios zwischen unterschiedlichen Stimmen erzeugen

Ein Arpeggio kann zwischen Noten aus unterschiedlichen Stimmen auf demselben System gezogen werden, wenn der `Span_arpeggio_engraver` in den `Staff`-Kontext verschoben wird:

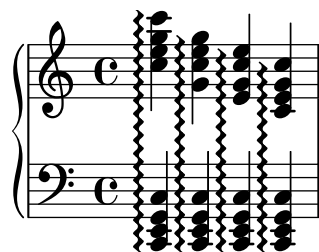
```
\new Staff \with {
  \consists "Span_arpeggio_engraver"
}
\relative c' {
  \set Staff.connectArpeggios = ##t
  <<
    { <e' g>4\arpeggio <d f> <d f>2 }
    \\
    { <d, f>2\arpeggio <g b>2 }
  >>
}
```



Arpeggio über mehrere Systeme in anderen Kontexten

Arpeggio über mehrere Systeme können in anderen Kontexten als dem `PianoStaff` erstellt werden, wenn der `Span_arpeggio_engraver` in den `Score`-Kontext eingefügt wird.

```
\new PianoStaff \relative c' ' <<
  \set PianoStaff.connectArpeggios = ##t
  \new Staff {
    <c e g c>4\arpeggio
    <g c e g>4\arpeggio
    <e g c e>4\arpeggio
    <c e g c>4\arpeggio
  }
  \new Staff {
    \clef bass
    \repeat unfold 4 {
      <c,, e g c>4\arpeggio
    }
  }
}>>
```



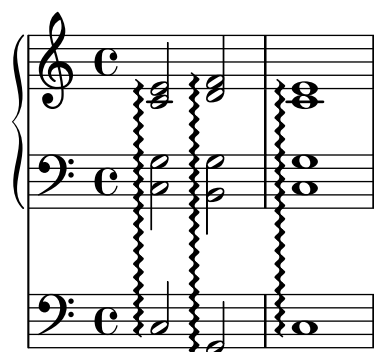
Arpeggio zwischen Systemen in einem Klaviersystem erstellen

In einem Klaviersystem (PianoStaff) ist es möglich, ein Arpeggio zwischen beiden Systemen zu verbinden, indem die `PianoStaff.connectArpeggios`-Eigenschaft gesetzt wird.

```
<<
\new PianoStaff <<
  \new Voice \relative c' {
    <c e>2\arpeggio <d f>2\arpeggio
    <c e>1\arpeggio
  }
  \new Voice \relative c {
    \clef bass
    <c g'>2\arpeggio <b g'>2\arpeggio
    <c g'>1\arpeggio
  }
>>

\new Staff \relative c {
  \set Score.connectArpeggios = ##t
  \clef bass
  c2\arpeggio g\arpeggio
  c1\arpeggio
}
>>

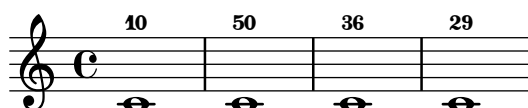
\layout {
  \context {
    \Score
    \consists "Span_arpeggio_engraver"
  }
}
```



Creating double-digit fingerings

Creating fingerings larger than 5 is possible.

```
\relative c' {
  c1-10
  c1-50
  c1-36
  c1-29
}
```



Creating slurs across voices

In some situations it is necessary to create slurs between notes from different voices. The solution is to add invisible notes to one of the voices, using `\hideNotes`.

This example is measure 235 of the Ciaccona from Bach's second partita for solo violin, BWV 1004.

```
\relative c' {
  <<
  {
    d16( a') s a s a[ s a] s a[ s a]
  }
  \\\
  {
    \slurUp
    bes,16[ s e](
    \hideNotes a)
    \unHideNotes f[(
    \hideNotes a)
    \unHideNotes fis](
    \hideNotes a)
    \unHideNotes g[(
    \hideNotes a)
    \unHideNotes gis](
    \hideNotes a)
  }
  >>
}
```



Creating text spanners

The `\startTextSpan` and `\stopTextSpan` commands allow the creation of text spanners as easily as pedal indications or octavations. Override some properties of the `TextSpanner` object to modify its output.

```
\paper { ragged-right = ##f }
```

```

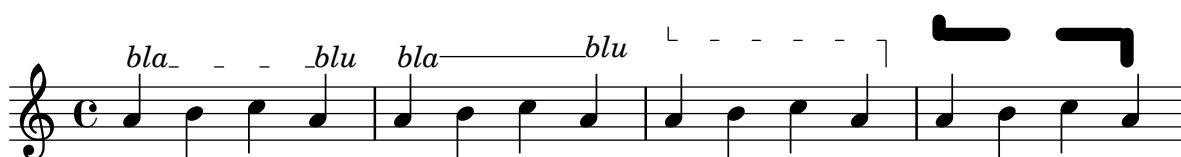
\relative c' {
  \override TextSpanner.bound-details.left.text = #"bla"
  \override TextSpanner.bound-details.right.text = #"blu"
  a4 \startTextSpan
  b4 c
  a4 \stopTextSpan

  \override TextSpanner.style = #'line
  \once \override TextSpanner.bound-details.left.stencil-align-dir-y = #CENTER
  a4 \startTextSpan
  b4 c
  a4 \stopTextSpan

  \override TextSpanner.style = #'dashed-line
  \override TextSpanner.bound-details.left.text =
    \markup { \draw-line #'(0 . 1) }
  \override TextSpanner.bound-details.right.text =
    \markup { \draw-line #'(0 . -2) }
  \once \override TextSpanner.bound-details.right.padding = #-2
  a4 \startTextSpan
  b4 c
  a4 \stopTextSpan

  \override TextSpanner.dash-period = #10
  \override TextSpanner.dash-fraction = #0.5
  \override TextSpanner.thickness = #10
  a4 \startTextSpan
  b4 c
  a4 \stopTextSpan
}

```



Dynamics spanner with custom text

Postfix functions for custom crescendo text spanners. The spanners should start on the first note of the measure. One has to use `-\mycresc`, otherwise the spanner start will rather be assigned to the next note.

*% Two functions for (de)crescendo spanners where you can explicitly
% give the spanner text.*

```

mycresc =
#(define-music-function (mymarkup) (markup?)
  (make-music 'CrescendoEvent
    'span-direction START
    'span-type 'text
    'span-text mymarkup))

mydecresc =
#(define-music-function (mymarkup) (markup?)

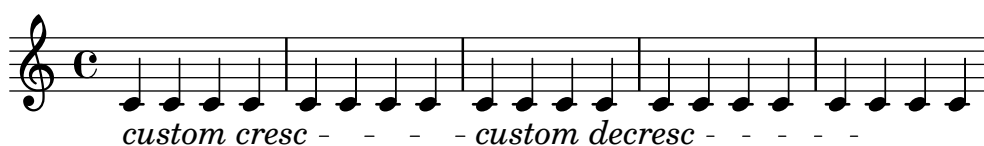
```

```

(make-music 'DecrescendoEvent
  'span-direction START
  'span-type 'text
  'span-text mymarkup))

\relative c' {
  c4-\mycresc "custom cresc" c4 c4 c4 |
  c4 c4 c4 c4 |
  c4-\mydecresc "custom decresc" c4 c4 c4 |
  c4 c4 c4 c4 |
  c4 c4\! c4 c4
}

```



Glissando kann Grobs überspringen

NoteColumn-Grobs können bei Glissandos übersprungen werden.

```

\relative c' {
  a2 \glissando
  \once \override NoteColumn.glissando-skip = ##t
  f''4 d,
}

```



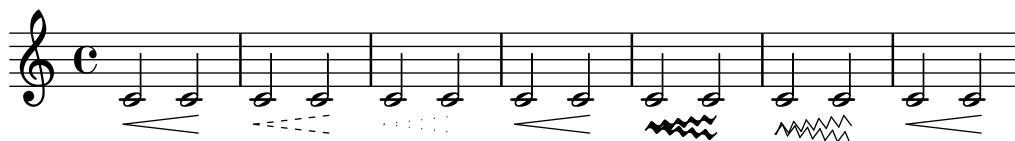
Hairpins with different line styles

Hairpins can take any style from line-interface: dashed-line, dotted-line, line, trill, or zigzag.

```

\relative c' {
  c2\< c\!
  \override Hairpin.style = #'dashed-line
  c2\< c\!
  \override Hairpin.style = #'dotted-line
  c2\< c\!
  \override Hairpin.style = #'line
  c2\< c\!
  \override Hairpin.style = #'trill
  c2\< c\!
  \override Hairpin.style = #'zigzag
  c2\< c\!
  \revert Hairpin.style
  c2\< c\!
}

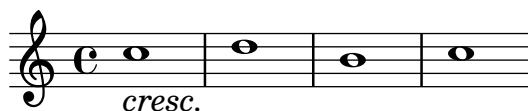
```



Crescendo-Linien von Dynamik-Texten unterdrücken

Dynamik-Texte (wie *cresc.* und *dim.*) werden mit einer gestrichelten Linie gesetzt, die ihre Dauer anzeigt. Diese Linie kann auf folgende Weise unterdrückt werden:

```
\relative c' {
  \override DynamicTextSpanner.style = #'none
  \crescTextCresc
  c1\< | d | b | c\!
}
```



Horizontally aligning custom dynamics like „più f“

Some dynamic expressions involve additional text, like „sempre **pp**“. Since dynamics are usually centered under the note, the `\pp` would be displayed way after the note it applies to.

To correctly align the „sempre **pp**“ horizontally so that it is aligned as if it were only the `\pp`, there are several approaches:

- Simply use `\once \override DynamicText.X-offset = #-9.2` before the note with the dynamics to manually shift it to the correct position. Drawback: This has to be done manually each time you use that dynamic markup...
- Add some padding (`#:hspace 7.1`) into the definition of your custom dynamic mark so that after LilyPond center-aligns it, it is already correctly aligned. Drawback: The padding really takes up that space and does not allow any other markup or dynamics to be shown in that position.
- Shift the dynamic script `\once \overrideX-offset =`. Drawback: `\once \override` is needed for every invocation!
- Set the dimensions of the additional text to 0 (using `#:with-dimensions '(0 . 0)` `'(0 . 0)`). Drawback: For LilyPond, „sempre“ has no extent now. This means it might put other stuff there, causing collisions (which are not detected by LilyPond’s collision detection algorithm!). There also seems to be some spacing, so it is not exactly the same alignment as without the additional text.
- Add an explicit shift directly inside the scheme function for the dynamic script.
- Set an explicit alignment inside the dynamic script. By default, this won’t have any effect, only if one sets `X-offset`! Drawback: One needs to set `DynamicText.X-offset`, which will apply to all dynamic texts! Also, it is aligned at the right edge of the additional text, not at the center of `\pp`.

```
\paper {
  ragged-right = ##f
  indent = 5\cm
}
```

```
% Solution 1: Using a simple markup with a particular halign value
% Drawback: It's a markup, not a dynamic command, so \dynamicDown
```

```

%           etc. will have no effect
semppMarkup = \markup { \halign #1.4 \italic "sempre" \dynamic "pp" }

% Solution 2: Using a dynamic script & shifting with
%           \once \override ...X-offset = ..
% Drawback: \once \override needed for every invocation
semppK =
#(make-dynamic-script
  (markup #:line
    (#:normal-text
      #:italic "sempre"
      #:dynamic "pp"))))

% Solution 3: Padding the dynamic script so the center-alignment
%           puts it at the correct position
% Drawback: the padding really reserves the space, nothing else can be there
semppT =
#(make-dynamic-script
  (markup #:line
    (#:normal-text
      #:italic "sempre"
      #:dynamic "pp"
      #:hspace 7.1)))

% Solution 4: Dynamic, setting the dimensions of the additional text to 0
% Drawback: To lilypond "sempre" has no extent, so it might put
%           other stuff there => collisions
% Drawback: Also, there seems to be some spacing, so it's not exactly the
%           same alignment as without the additional text
semppM =
#(make-dynamic-script
  (markup #:line
    (#:with-dimensions '(0 . 0) '(0 . 0)
      #:right-align
      #:normal-text
      #:italic "sempre"
      #:dynamic "pp"))))

% Solution 5: Dynamic with explicit shifting inside the scheme function
semppG =
#(make-dynamic-script
  (markup #:hspace 0
    #:translate '(-18.85 . 0)
    #:line (#:normal-text
      #:italic "sempre"
      #:dynamic "pp"))))

% Solution 6: Dynamic with explicit alignment. This has only effect
%           if one sets X-offset!
% Drawback: One needs to set DynamicText.X-offset!
% Drawback: Aligned at the right edge of the additional text,
%           not at the center of pp

```

```

semppMII =
#(make-dynamic-script
  (markup #:line (#:right-align
                  #:normal-text
                  #:italic "sempre"
                  #:dynamic "pp")))

\new StaffGroup <<
  \new Staff \with { instrumentName = "standard" }
    \relative c'' {
      \key es \major
      c4\pp c\p c c | c\ff c c\pp c
    }
  \new Staff \with { instrumentName = "normal markup" }
    \relative c'' {
      \key es \major
      c4-\semppMarkup c\p c c | c\ff c c-\semppMarkup c
    }
  \new Staff \with { instrumentName = "explicit shifting" }
    \relative c'' {
      \key es \major
      \once \override DynamicText.X-offset = #-9.2
      c4\semppK c\p c c
      c4\ff c
      \once \override DynamicText.X-offset = #-9.2
      c4\semppK c
    }
  \new Staff \with { instrumentName = "right padding" }
    \relative c'' {
      \key es \major
      c4\semppT c\p c c | c\ff c c\semppT c
    }
  \new Staff \with { instrumentName = "set dimension to zero" }
    \relative c'' {
      \key es \major
      c4\semppM c\p c c | c\ff c c\semppM c
    }
  \new Staff \with { instrumentName = "shift inside dynamics" }
    \relative c'' {
      \key es \major
      c4\semppG c\p c c | c\ff c c\semppG c
    }
  \new Staff \with { instrumentName = "alignment inside dynamics" }
    \relative c'' {
      \key es \major
      \override DynamicText.X-offset = #-1
      c4\semppMII c\p c c | c\ff c c\semppMII c
    }
>>

\layout { \override Staff.InstrumentName.self-alignment-X = #LEFT }

```


standard	
normal markup	
explicit shifting	
right padding	
set dimension to zero	
shift inside dynamics	
alignment inside dynamics	

Eine Zäsur einfügen

Zäsurzeichen können erstellt werden, indem die 'text'-Eigenschaft des BreathingSign-Objektes verändert wird. Ein gekrümmtes Zäsurzeichen ist auch möglich.

```
\relative c' {
  \override BreathingSign.text = \markup {
    \musicglyph "scripts.caesura.straight"
  }
  c8 e4. \breathe g8. e16 c4

  \override BreathingSign.text = \markup {
    \musicglyph "scripts.caesura.curved"
  }
  g8 e'4. \breathe g8. e16 c4
}
```



Laissez vibrer ties

Laissez vibrer ties have a fixed size. Their positioning can be tuned using the tie-configuration property.

See also snippet „Longer laissez vibrer ties“.

```
\relative c' {
  <c e g>4\laissezVibrer r <c f g>\laissezVibrer r
```

```

<c d f g>4\laissezVibrer r <c d f g>4.\laissezVibrer r8

<c d e f>4\laissezVibrer r
\override LaissezVibrerTieColumn.tie-configuration
  = #'((-7 . ,DOWN)
        (-5 . ,DOWN)
        (-3 . ,UP)
        (-1 . ,UP))
<c d e f>4\laissezVibrer r
}

```



Line arrows

Arrows can be applied to text spanners and line spanners (such as glissandi).

```

\relative c' ' {
  \override TextSpanner.bound-padding = #1.0
  \override TextSpanner.style = #'line
  \override TextSpanner.bound-details.right.arrow = ##t
  \override TextSpanner.bound-details.left.text = #"fof"
  \override TextSpanner.bound-details.right.text = #"gag"
  \override TextSpanner.bound-details.right.padding = #0.6

  \override TextSpanner.bound-details.right.stencil-align-dir-y = #CENTER
  \override TextSpanner.bound-details.left.stencil-align-dir-y = #CENTER

  \override Glissando.bound-details.right.arrow = ##t
  \override Glissando.arrow-length = #0.5
  \override Glissando.arrow-width = #0.25

  a8\startTextSpan gis a4 b\glissando b,
  g'4 c\stopTextSpan c2
}

```



Legatobögen mit kompliziertem Strichelmuster definieren

Legatobögen können mit einem komplizierten Strichelmuster gesetzt werden, indem die dash-definition-Eigenschaft definiert wird. dash-definition ist eine Liste bestehend aus dash-elements-Elementen. Ein dash-element ist eine Liste an Parametern, die das Strichverhalten für einen Abschnitt des Legatobogens definieren.

Der Bogen wird nach dem Bezierparameter t definiert, welcher von 0 am linken Ende des Bogens zu 1 am rechten Ende des Bogens reicht. dash-element ist eine Liste (start-t stop-t dash-Unterbrechung dash-Abschnitt). Die Region des Bogens von start-t bis stop-t hat eine Unterbrechung von dash-Unterbrechung von jedem dash-Abschnitt-Schwarzabschnitt.

dash-Abschnitt ist in Notenlinienzwischenräumen definiert. dash-Abschnitt ist auf 1 für einen durchgehenden Bogen gesetzt.

```
\relative c' {
  \once \override
    Slur.dash-definition = #'(( 0 0.3 0.1 0.75)
                               (0.3 0.6 1 1 )
                               (0.65 1.0 0.4 0.75))

  c4( d e f)
  \once \override
    Slur.dash-definition = #'((0 0.25 1 1 )
                               (0.3 0.7 0.4 0.75)
                               (0.75 1.0 1 1 ))

  c4( d e f)
}
```



Die Standardwerte der Abkürzungen von Artikulationen verändern

Die Abkürzungen sind in der Datei `ly/script-init.ly` definiert, wo den Variablen `dashHat`, `dashPlus`, `dashDash`, `dashBar`, `dashLarger`, `dashDot` und `dashUnderscore` Standardwerte zugewiesen werden. Diese Standardwerte können verändert werden. Um zum Beispiel die Abkürzung `~+` (`dashPlus`) mit dem Triller anstatt mit dem `+`-Symbol zu assoziieren, muss der Wert `trill` der Variable `dashPlus` zugewiesen werden:

```
\relative c' ' { c1~+ }
```

```
dashPlus = \trill
```

```
\relative c' ' { c1~+ }
```



Moving slur positions vertically

The vertical position of a slur can be adjusted using the `positions` property of `Slur`. The property has 2 parameters, the first referring to the left end of the slur and the second to the right. The values of the parameters are not used by LilyPond to make an exact movement of the slur – instead it selects what placement of the slur looks best, taking into account the parameter values. Positive values move the slur up, and are appropriate for notes with stems down. Negative values move downward slurs further down.

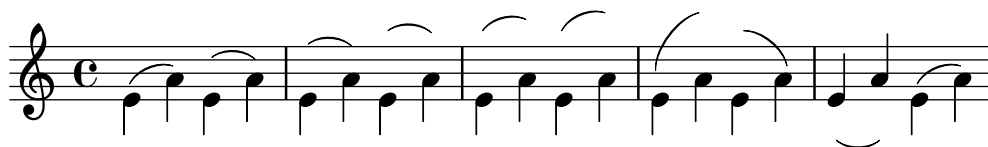
See also snippet „Adjusting slur positions vertically“.

```
\relative c' {
  \stemDown
```

```

e4( a)
\override Slur.positions = #'(1 . 1)
e4( a)
\override Slur.positions = #'(2 . 2)
e4( a)
\override Slur.positions = #'(3 . 3)
e4( a)
\override Slur.positions = #'(4 . 4)
e4( a)
\override Slur.positions = #'(5 . 5)
e4( a)
\override Slur.positions = #'(0 . 5)
e4( a)
\override Slur.positions = #'(5 . 0)
e4( a)
\stemUp
\override Slur.positions = #'(-5 . -5)
e4( a)
\stemDown
\revert Slur.positions
e4( a)
}

```



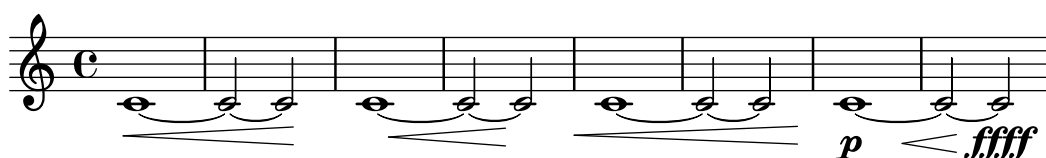
Moving the ends of hairpins

The ends of hairpins may be offset by setting the `shorten-pair` property of the `Hairpin` object. Positive values move endpoints to the right, negative to the left. Unlike the `minimum-length` property, this property only affects the appearance of the hairpin; it does not adjust horizontal spacing (including the position of bounding dynamics). This method is thus suitable for fine-tuning a hairpin within its allotted space.

```

{
  c'1~\<
  c'2~ c'\!
  \once \override Hairpin.shorten-pair = #'(2 . 2)
  c'1~\<
  c'2~ c'\!
  \once \override Hairpin.shorten-pair = #'(-2 . -2)
  c'1~\<
  c'2~ c'\!
  c'1~\p-\tweak shorten-pair #'(2 . 0)\<
  c'2~ c'\ffff
}

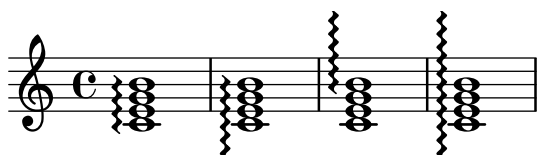
```



Positioning arpeggios

If you need to extend or shorten an arpeggio, you can modify the upper and lower start positions independently.

```
\relative c' {
  <c e g b>1\arpeggio
  \once \override Arpeggio.positions = #'(-5 . 0)
  <c e g b>1\arpeggio
  \once \override Arpeggio.positions = #'(0 . 5)
  <c e g b>1\arpeggio
  \once \override Arpeggio.positions = #'(-5 . 5)
  <c e g b>1\arpeggio
}
```



Textbeschriftung innerhalb von Bögen positionieren

Textbeschriftung kann innerhalb von Bögen gesetzt werden, wenn die `outside-staff-priority`-Eigenschaft auf falsch gesetzt wird.

```
\relative c' {
  \override TextScript.avoid-slur = #'inside
  \override TextScript.outside-staff-priority = ##f
  c2(~\markup { \halign #-10 \natural } d4.) c8
}
```



Printing hairpins in various styles

Hairpin dynamics may be created in a variety of styles.

```
\relative c' {
  \override Hairpin.stencil = #flared-hairpin
  a4< a a a\f
  a4\p< a a a\ff
  a4\s fz< a a a\!
  \override Hairpin.stencil = #constante-hairpin
  a4< a a a\f
  a4\p< a a a\ff
  a4\s fz< a a a\!
  \override Hairpin.stencil = #flared-hairpin
  a4> a a a\f
  a4\p> a a a\ff
  a4\s fz> a a a\!
  \override Hairpin.stencil = #constante-hairpin
  a4> a a a\f
  a4\p> a a a\ff
}
```

```
a4\s fz\> a a a\!
```



Crescendo Klammern al niente schreiben

Crescendo-Klammern können mit einem kleinen Kreis vor der Spitze notiert werden (al niente = bis zum Nichts), indem die `circled-tip`-Eigenschaft des Hairpin-Objekts auf `#t` gesetzt wird.

```
\relative c' {
  \override Hairpin.circled-tip = ##t
  c2\< c\!
  c4\> c\< c2\!
}
```



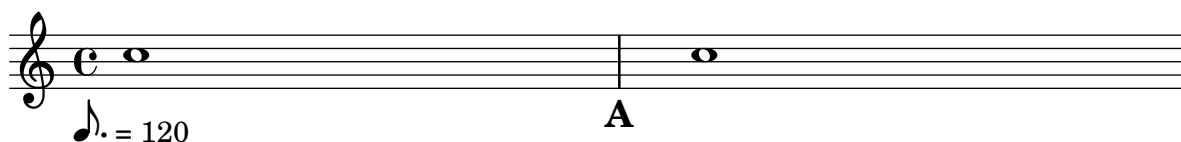
Metronom- und Übungszeichen unter das System setzen

Normalerweise werden Metronom- und Übungszeichen über dem Notensystem ausgegeben. Um sie unter das System zu setzen, muss die `direction`-Eigenschaft von `MetronomeMark` oder `RehearsalMark` entsprechend verändert werden.

```
\layout {
  ragged-right = ##f
}

{
  % Metronome marks below the staff
  \override Score.MetronomeMark.direction = #DOWN
  \tempo 8. = 120
  c''1

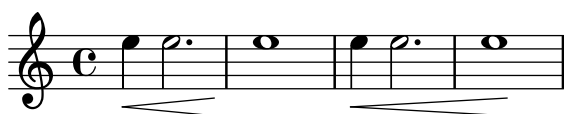
  % Rehearsal marks below the staff
  \override Score.RehearsalMark.direction = #DOWN
  \mark \default
  c''1
}
```



Das Verhalten von Crescendo-Klammern an Taktlinien beeinflussen

Wenn die Note, an welcher eine Crescendo-Klammer endet, die erste Note eines Taktes ist, wird die Klammer an der vorhergehenden Taktlinie beendet. Dieses Verhalten kann auch mit der Eigenschaft `'to-barline` geändert werden:

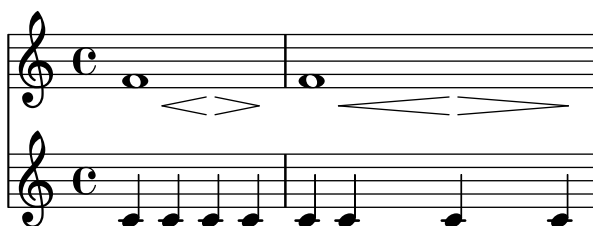
```
\relative c' {
  e4\< e2.
  e1\!
  \override Hairpin.to-barline = ##f
  e4\< e2.
  e1\!
}
```



Die Mindestlänge von Crescendo-Klammern bestimmen

Wenn Crescendo-Klammern zu kurz sind, können sie verlängert werden, indem die `minimum-length`-Eigenschaft des `Hairpin`-Objektes verändert wird.

```
<<
{
  \after 4 \< \after 2 \> \after 2. \! f'1
  \override Hairpin.minimum-length = 8
  \after 4 \< \after 2 \> \after 2. \! f'1
}
{
  \repeat unfold 8 c'4
}
>>
```



Showing the same articulation above and below a note or chord

By default, LilyPond does not allow the same articulation (an accent, a fermata, a flageolet, etc.) to be displayed above and below a note. For example, `c4_\fermata^\fermata` only shows a fermata below. The fermata above gets simply ignored.

However, one can stick scripts (just like fingerings) inside a chord, which means it is possible to have as many articulations as desired. This approach has the advantage that it ignores the stem and positions the articulation relative to the note head. This can be seen in the case of the flageolets in the snippet. To mimic the behaviour of scripts outside a chord, `add-stem-support` would be required.

The solution is thus to write the note as a chord and add the articulations inside of `<...>`, using the direction modifiers `^` and `_` as appropriate.

```
\relative c' {
```

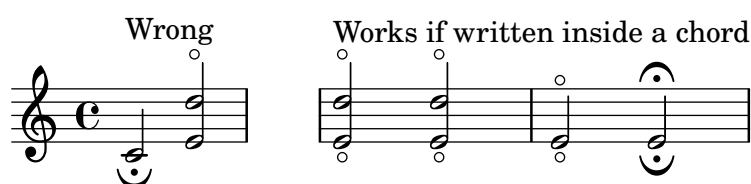
```

<>^"Wrong"
c2_\fermata^\fermata % The second fermata is ignored!
<e d'>2^\flageolet_\flageolet

\stopStaff s1 \startStaff

<>^"Works if written inside a chord"
<e_\flageolet d'^\flageolet>2
<e_\flageolet d'^\flageolet>2
<e_\flageolet^\flageolet>2
<e_\fermata^\fermata>2
}

```



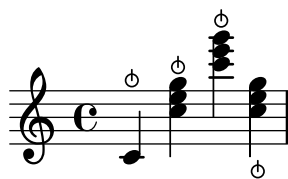
Snap pizzicato („Bartok“ pizzicato)

A snap pizzicato (also known as „Bartok pizzicato“) is a „strong pizzicato where the string is plucked vertically by snapping and rebounds off the fingerboard of the instrument“ (Wikipedia). It is denoted by a circle with a vertical line going from the center upwards outside the circle.

```

\relative c' {
  c4\snappizzicato
  <c' e g>4\snappizzicato
  <c' e g>4^\snappizzicato
  <c, e g>4_\snappizzicato
}

```



Using a bracket to clarify divisi

The `\nonArpeggiato` command can be used to indicate the division of voices where there are no stems to provide the information. This is often seen in choral music.

```

\include "english.ly"

\score {
  \relative c' {
    \key a \major
    \time 2/2
    <<
      \new Voice = "upper" <<
        {
          \voiceOne
          a2( b2
          <b d>1\nonArpeggiato)

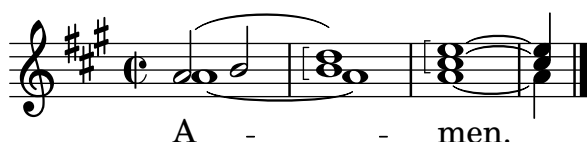
```



```

        <cs e>\nonArpeggiato ~
        <cs e>4
        \fine
    }
    \addlyrics { \lyricmode { A -- men. } }
>>
\new Voice = "lower" {
    \voiceTwo
    a1 ~
    a
    a ~
    a4
    \fine
}
>>
}
}

```



Using a tick as the breath mark symbol

Vocal and wind music frequently uses a tick mark as a breathing sign. This indicates a breath that subtracts a little time from the previous note rather than causing a short pause, which is indicated by the comma breath mark. The mark can be moved up a little to take it away from the staff.

```

\relative c' ' {
    c2
    \breathe
    d2
    \override BreathingSign.Y-offset = #2.6
    \override BreathingSign.text =
        \markup { \musicglyph "scripts.tickmark" }
    c2
    \breathe
    d2
}

```



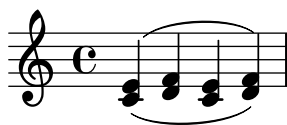
Doppelte Bögen für Legato-Akkorde benutzen

Einige Komponisten schreiben doppelte Bögen, wenn Legato-Akkorde notiert werden. Das kann mit der Eigenschaft `doubleSlurs` erreicht werden.

```

\relative c' {
    \set doubleSlurs = ##t
    <c e>4( <d f> <c e> <d f>)
}

```



Using the whiteout property

Any graphical object can be printed over a white background to mask parts of objects that lie beneath. This can be useful to improve the appearance of collisions in complex situations when repositioning objects is impractical. It is necessary to explicitly set the layer property to control which objects are masked by the white background.

In this example the collision of the tie with the time signature is improved by masking out the part of the tie that crosses the time signature, setting the `whiteout` property of `TimeSignature`. To do this, `TimeSignature` is moved to a layer above `Tie`, which is left in the default layer 1, and `StaffSymbol` is moved to a layer above `TimeSignature` so it is not masked.

```
{
  \override Score.StaffSymbol.layer = 4
  \override Staff.TimeSignature.layer = 3
  b'2 b'~
  \once \override Staff.TimeSignature.whiteout = ##t
  \time 3/4
  b' r4
}
```



Vertical line as a baroque articulation mark

This short vertical line placed above the note is commonly used in baroque music. Its meaning can vary, but generally indicates notes that should be played with more „weight“. The following example demonstrates how to achieve such a notation.

```
upline =
\tweak stencil
#(lambda (grob)
  (grob-interpret-markup grob #{ \markup \draw-line #'(0 . 1) #}))
\stopped

\relative c' {
  a'4~\upline a( c d')_\upline
}
```



Vertically aligning dynamics across multiple notes

Dynamics that occur at, begin on, or end on the same note will be vertically aligned. To ensure that dynamics are aligned when they do not occur on the same note, increase the `staff-padding` property of the `DynamicLineSpanner` object.

```
\relative c' {
```

```
\override DynamicLineSpanner.staff-padding = #4  
c2\p f\mf  
g2\< b4\> c\!  
}
```



4 Repeats

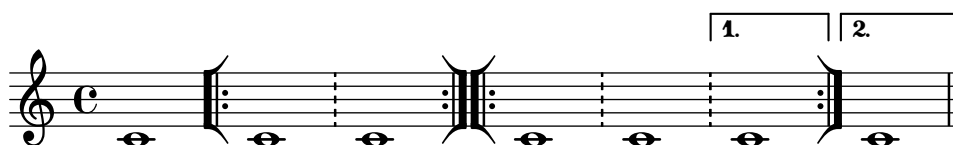
See also Abschnitt “Repeats” in *Notationsreferenz*.

Changing the default bar lines

Default bar lines can be changed when re-defined in a Score context.

```
\layout {
  \context {
    \Score
    % Changing the defaults from engraver-init.ly
    measureBarType = "!"
    startRepeatBarType = "[|:"
    endRepeatBarType = ":|]"
    doubleRepeatBarType = ":||[|:"
  }
}

{
  c'1
  \repeat volta 2 { c' c' }
  \repeat volta 2 { c' c' \alternative { \volta 1 { c' }
                                         \volta 2 { c' } } }
  \bar "|."
}
```



Controlling the appearance of tremolo slashes

Using various properties of the StemTremolo grob it is possible to control the appearance of tremolo slashes.

- Property `slope` sets the slope for tremolo slashes.
- Property `shape` determines whether tremolo slashes look like rectangles (value `rectangle`) or like very small beams (value `beam-like`).
- Property `style` sets both the slope and the shape depending on whether the note has flags, beams, or only a plain stem. This is in contrast to the previous two properties, which change the slope and shape unconditionally. There are two styles defined.
 - `default`: slashes for down-stem flags are longer and more sloped than slashes for up-stem flags; slashes on beamed notes have a rectangular shape and are parallel to the beam.
 - `constant`: all slashes are beam-like and have the same slope except for down-stem flags.

```
music = {
  a''4:32 a':
  e''8: \noBeam e':
  a'':[ a':]
  f':[ g':]
  d':[ d':]
```

```

}

\new Staff {
  <>^\markup "default"
  \music
}

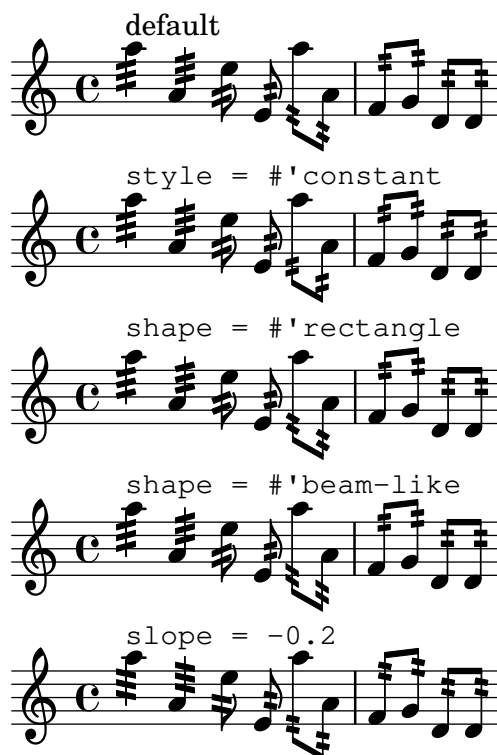
\new Staff {
  <>^\markup \typewriter "style = #'constant"
  \override StemTremolo.style = #'constant
  \music
}

\new Staff {
  <>^\markup \typewriter "shape = #'rectangle"
  \override StemTremolo.shape = #'rectangle
  \music
}

\new Staff {
  <>^\markup \typewriter "shape = #'beam-like"
  \override StemTremolo.shape = #'beam-like
  \music
}

\new Staff {
  <>^\markup \typewriter "slope = -0.2"
  \override StemTremolo.slope = -0.2
  \music
}

```



Cross-staff tremolos

Since `\repeat tremolo` expects exactly two musical arguments for chord tremolos, the note or chord which changes staff within a cross-staff tremolo should be placed inside curly braces together with its `\change Staff` command.

```
\new PianoStaff <<
  \new Staff = "up" \relative c'' {
    \key a \major
    \time 3/8
    s4.
  }
  \new Staff = "down" \relative c'' {
    \key a \major
    \time 3/8
    \voiceOne
    \repeat tremolo 6 {
      <a e'>32
      {
        \change Staff = "up"
        \voiceTwo
        <cis a' dis>32
      }
    }
  }
>>
```



Engraving tremolos with floating beams

If a tremolo's total duration is less than a quarter-note, or exactly a half note, or between a half note and a whole note, it is normally typeset with all beams touching the stems. Certain engraving styles typeset some of these beams as centered floating beams that do not touch the stems. The number of floating beams in this type of tremolo is controlled with the `gap-count` property of the `Beam` object, and the size of the gaps between beams and stems is set with the `gap` property.

```
\relative c'' {
  \repeat tremolo 8 { a32 f }
  \override Beam.gap-count = #1
  \repeat tremolo 8 { a32 f }
  \override Beam.gap-count = #2
  \repeat tremolo 8 { a32 f }
  \override Beam.gap-count = #3
  \repeat tremolo 8 { a32 f }

  \override Beam.gap-count = #3
  \override Beam.gap = #1.33
```

```

\repeat tremolo 8 { a32 f }
\override Beam.gap = #1
\repeat tremolo 8 { a32 f }
\override Beam.gap = #0.67
\repeat tremolo 8 { a32 f }
\override Beam.gap = #0.33
\repeat tremolo 8 { a32 f }
}

```



Isolierte Prozentwiederholungen

Isolierte Prozentwiederholungen können auch ausgegeben werden. Das wird erreicht, indem man eine Ganztaktpause notiert und ihre Ausgabeform ändert:

```

makePercent =
#(define-music-function (note) (ly:music?)
  "Make a percent repeat the same length as NOTE."
  (make-music 'PercentEvent
    'length (ly:music-length note)))

\relative c' {
  \makePercent s1
}

```



Measure counters

This snippet demonstrates the use of the `Measure_counter_engraver` to number groups of successive measures. Any stretch of measures may be numbered, whether consisting of repetitions or not.

The engraver must be added to the appropriate context. Here, a `Staff` context is used; another possibility is a `Dynamics` context.

The counter is begun with `\startMeasureCount` and ended with `\stopMeasureCount`. Numbering will start by default with 1, but this behavior may be modified by overriding the `count-from` property.

When a measure extends across a line break, the number will appear twice, the second time in parentheses.

```

\layout {
  \context {
    \Staff
    \consists #Measure_counter_engraver
  }
}

```

```

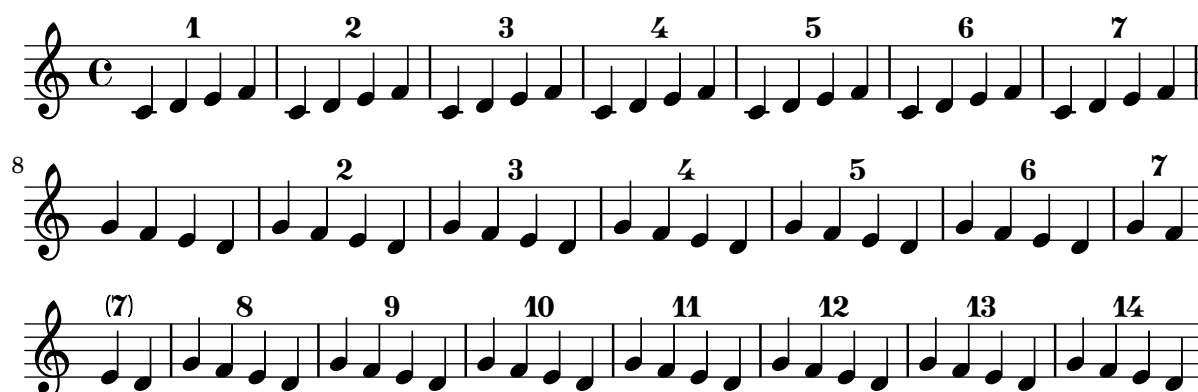
\new Staff {
  \startMeasureCount

```

```

\repeat unfold 7 {
  c'4 d' e' f'
}
\stopMeasureCount
\bar "||"
g'4 f' e' d'
\override Staff.MeasureCounter.count-from = #2
\startMeasureCount
\repeat unfold 5 {
  g'4 f' e' d'
}
g'4 f'
\bar ""
\break
e'4 d'
\repeat unfold 7 {
  g'4 f' e' d'
}
\stopMeasureCount
}

```



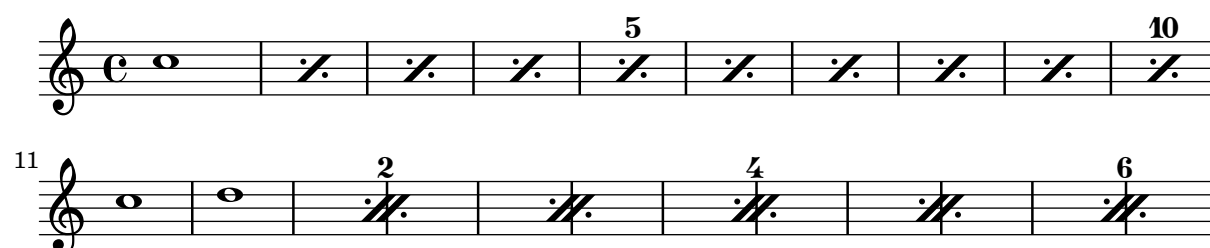
Sichtbarkeit von Prozent-Wiederholungen

Prozentwiederholungszähler können in regelmäßigen Intervallen angezeigt werden, indem man die Eigenschaft `repeatCountVisibility` beeinflusst.

```

\relative c' {
  \set countPercentRepeats = ##t
  \set repeatCountVisibility = #(every-nth-repeat-count-visible 5)
  \repeat percent 10 { c1 } \break
  \set repeatCountVisibility = #(every-nth-repeat-count-visible 2)
  \repeat percent 6 { c1 d1 }
}

```



Prozent-Wiederholungen zählen

Ganztaktwiederholungen mit mehr als zwei Wiederholungen erhalten einen Zähler, wenn man die entsprechende Eigenschaft einsetzt:

```
\relative c' ' {
  \set countPercentRepeats = ##t
  \repeat percent 4 { c1 }
}
```



Positioning segno and coda (with line break)

If you want to place an exiting segno sign and add text like „D.S. al Coda“ next to it where usually the staff lines are you can use this snippet. The coda will resume in a new line. There is a variation documented in this snippet, where the coda will remain on the same line.

```
\relative c' ' {
  c4 c c c | c c c c |
  \repeat segno 2 {
    c4 c c c | c c c c |
    \alternative {
      \volta 1 {
        c4 c c c | c c c c |
        % If you don't use \break at Coda, use \noBreak here
        % and after \bar "" below.
        \noBreak
        \section % double bar line
        \cadenzaOn % pause bar count
        \stopStaff % remove staff lines
        % Increasing the unfold counter will expand the staff-free space
        \repeat unfold 4 {
          s1
          \bar ""
        }
        % Place JumpScript where the staff would normally be.
        \once \override Score.JumpScript.outside-staff-priority = ##f
        \once \override Score.JumpScript.Y-offset = 0
        \startStaff % resume bar count
        \cadenzaOff % show staff lines again
      }
    }
  }
}
\sectionLabel "Coda"
% Show Coda on a new line
\break
\repeat unfold 6 { c4 c c c }
\fine
}
```

*D.S. % al Coda
e poi la Coda*

(7) Coda

Setting the double repeat default for volte

There are different double repeat styles for volte that can be selected using the context property `doubleRepeatBarType`.

```
\relative c' {
  \repeat volta 2 { c1 }
  \set Score.doubleRepeatBarType = ":\dots:"
  \repeat volta 2 { c1 }
  \set Score.doubleRepeatBarType = ":\mid\mid:"
  \repeat volta 2 { c1 }
  \set Score.doubleRepeatBarType = ":\mid\mid:"
  \repeat volta 2 { c1 }
}
```

Volta-Klammern verkürzen

Volta-Klammern werden normalerweise über alle Noten der Klammer gezogen, aber es ist möglich sie zu verkürzen. Hierzu muss `voltaSpannerDuration` definiert werden, in dem Beispiel etwa als 3/4, sodass die Klammer nur einen Takt dauert.

```
\fixed c' {
  \time 3/4
  c4 c c
  \repeat volta 5 {
    d4 d d
    \alternative {
      \volta 1,2,3,4 {
        \once \override Score.VoltaBracket.musical-length =
          \musicLength 2.
        e4 e e
        f4 f f
      }
      \volta 5 {
        g4 g g } } }
  }
}
```

Unfolding tremolo repeats

Currently, `note:duration`, which is more or less a shortcut for `\repeat tremolo`, is not unfolded by `\unfoldRepeats` (this is tracked in Issue #6145 (<https://gitlab.com/lilypond/lilypond/-/issues/6145>)). The function given in this snippet provides a workaround.

```
fixTremolos =
#(define-music-function (music) (ly:music?)
  (music-map
    (lambda (m)
      (let ((event (any (lambda (a)
                          (and (music-is-of-type? a 'tremolo-event)
                              a))
                        (ly:music-property m 'articulations)))))
        (if event
          (let* ((total-tremolo-duration (ly:music-property m
                                                            'duration))
                 (tremolo-type (ly:music-property event
                                                    'tremolo-type))
                 (one-tremolo-note-duration
                  (ly:make-duration (ly:intlog2 tremolo-type)))
                 (tremolo-note-count
                  (/ tremolo-type (expt 2 (ly:duration-log
                                           total-tremolo-duration)))))
            (set! (ly:music-property m 'duration)
                  one-tremolo-note-duration)
            (set! (ly:music-property m 'articulations)
                  (delete! event (ly:music-property m 'articulations)))
            (make-music 'TremoloRepeatedMusic
                        'repeat-count tremolo-note-count
                        'element m))
          m)))
    music))

unfoldRepeats = \unfoldRepeats #'() \fixTremolos \etc

music = { \repeat tremolo 8 c'16 c'2:16 }

{
  \music
  \unfoldRepeats \music
}
```



Wiederholungs-(Volta-)Klammern unterhalb der Akkordsymbole

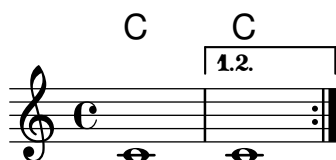
Indem man den `Volta_engraver` zu dem entsprechenden Notensystem hinzufügt, können Wiederholungsklammern unterhalb der Akkorde gesetzt werden.

```
\score {
```

```

<<
  \chords { c1 c1 }
  \new Staff \with { \consists "Volta_engraver" }
  {
    \repeat volta 2 { c'1 \alternative { c' } }
  }
>>
\layout {
  \context {
    \Score
    \remove "Volta_engraver"
  }
}
}

```



Volta brackets in multiple staves

By adding the `Volta_engraver` to the relevant staff, volte can be put over staves other than the topmost one in a score.

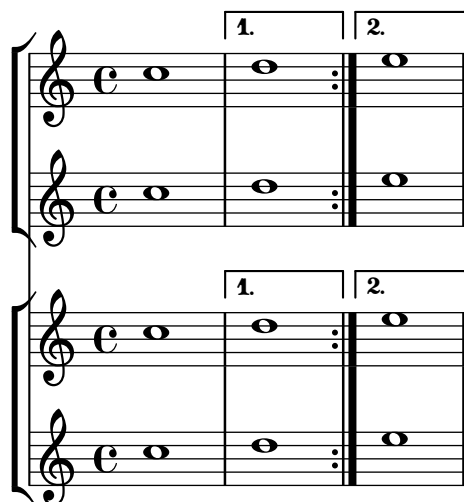
`\repeat` and related commands should be present in all staves.

```

voltaMusic = \relative c'' {
  \repeat volta 2 {
    c1
    \alternative {
      \volta 1 { d1 }
      \volta 2 { e1 }
    }
  }
}

<<
\new StaffGroup <<
  \new Staff \voltaMusic
  \new Staff \voltaMusic
>>
\new StaffGroup <<
  \new Staff \with { \consists "Volta_engraver" }
    \voltaMusic
  \new Staff \voltaMusic
>>
>>

```



Volta text markup using repeatCommands

Though voltes are best specified using `\repeat volta`, the context property `repeatCommands` must be used in cases where the volta text needs more advanced formatting with `\markup`.

Since `repeatCommands` takes a list, the simplest method of including markup is to use an identifier for the text and embed it in the command list using the Scheme syntax `#`((volta ,textIdentifier) ...)` (note the use of the backtick after `#` and the comma before *textIdentifier*). Start- and end-repeat commands can be added as separate list elements:

```
voltaAdLib = \markup { \volta-number { 1. 2. 3... } \italic { ad lib. } }
```

```
\relative c' ' {
  c1
  \set Score.repeatCommands = #`((volta ,voltaAdLib) start-repeat)
  c4 b d e
  \set Score.repeatCommands = #'((volta #f) (volta "4.") end-repeat)
  f1
  \set Score.repeatCommands = #'((volta #f))
}
```



5 Simultaneous notes

See also Abschnitt “Simultaneous notes” in *Notationsreferenz*.

Zusätzliche Stimmen, um Zusammenstöße zu vermeiden

Ein einigen Fällen von sehr komplexer polyphoner Musik sind zusätzliche Stimmen notwendig, um Zusammenstöße zwischen den Noten zu vermeiden. Wenn mehr als vier parallele Stimmen benötigt werden, können zusätzliche Stimmen definiert werden, indem eine Variable mit der Funktion `context-spec-music` definiert wird.

```
voiceFive = #(context-spec-music (make-voice-props-set 4) 'Voice)
```

```
\relative c' ' {
  \time 3/4
  \key d \minor
  \partial 2
  <<
    \new Voice {
      \voiceOne
      a4. a8
      e'4 e4. e8
      f4 d4. c8
    }
    \new Voice {
      \voiceTwo
      d,2
      d4 cis2
      d4 bes2
    }
    \new Voice {
      \voiceThree
      f'2
      bes4 a2
      a4 s2
    }
    \new Voice {
      \voiceFive
      s2
      g4 g2
      f4 f2
    }
  >>
}
```



Changing \partCombine texts

When using the automatic part combining feature, the printed text for the solo and unison sections may be changed.

```

\new Staff <<
  \set Staff.soloText = "girl"
  \set Staff.soloIIText = "boy"
  \set Staff.aDueText = "together"
  \partCombine
    \relative c'' {
      g4 g r r
      a2 g
    }
    \relative c'' {
      r4 r a( b)
      a2 g
    }
  }
>>

```



Changing a single note's size in a chord

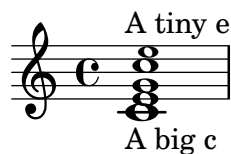
Individual note heads in a chord can be modified with the `\tweak` command inside a chord, by altering the `font-size` property.

Inside the chord (within the brackets `< >`), before the note to be altered, place the `\tweak` command, followed by `font-size` and define the proper size like `#-2` (a tiny note head).

```

\relative c' {
  <\tweak font-size #+2 c e g c
  \tweak font-size #-2 e>1
  ~\markup { A tiny e }_~\markup { A big c }
}

```



Clusters

Clusters are a device to denote that a complete range of notes is to be played.

```

fragment = \relative c' {
  c4 f <e d'>4
  <g a>8 <e a> a4 c2 <d b>4
  e2 c
}

<<
  \new Staff \fragment
  \new Staff \makeClusters \fragment
>>

```



Zwei Stimmen auf einem System kombinieren

Die Funktion, die Stimmen kombiniert (also der `\partCombine`-Befehl) ermöglicht die Kombination unterschiedlicher Stimmen auf einem System. Textanweisungen wie "solo" oder "a2" werden automatisch hinzugefügt. Wenn man sie entfernen will, muss man die Eigenschaft `printPartCombineTexts` auf falsch setzen. Für Klavierauszüge muss natürlich kein "solo"/"a2" usw. hinzugefügt werden, man sollte sie also ausschalten. Wenn aber Solo-Stellen in einem Klavierauszug oder einer Chorpartitur angezeigt werden, ist es besser, normale Polyphonie zu verwenden, weil so die Solostellen angezeigt werden, auch wenn der Text des Stimmenkombinierers ausgeschaltet ist.

Der Schnipsel zeigt drei Möglichkeiten, Stimmen auf einem System zu kombinieren: Standardpolyphonie, `\partCombine` ohne Text und `\partCombine` mit Text.

```
musicUp = \relative c'' {
  \time 4/4
  a4 c4.( g8) a4 |
  g4 e' g,( a8 b) |
  c b a2.
}

musicDown = \relative c'' {
  g4 e4.( d8) c4 |
  r2 g'4( f8 e) |
  d2 \stemDown a
}

\score {
  <<
    \new Staff \with {
      instrumentName = "standard polyphony"
    } << \musicUp \\\musicDown >>

    \new Staff \with {
      instrumentName =
        \markup { \typewriter "\\partCombine" without text}
      printPartCombineTexts = ##f
    } \partCombine \musicUp \musicDown

    \new Staff \with {
      instrumentName =
        \markup { \typewriter "\\partCombine" with text}
    } \partCombine \musicUp \musicDown
  >>

  \layout {
```



```

indent = 6.0\cm
\context {
  \Score
  % Setting this to a large value avoids a bar line at the
  % beginning that would connect the three staves otherwise.
  \override SystemStartBar.collapse-height = 30
}
}
}

```

standard polyphony	
\partCombine without text	
\partCombine with text	

Displaying complex chords

Here is a way to display a chord where the same note is played twice with different accidentals.

```

fixA = {
  \once \override Stem.length = #12
}

fixB = {
  \once \override NoteHead.X-offset = #1.7
  \once \override Stem.length = #7
  \once \override Stem.rotation = #'(45 0 0)
  \once \override Stem.extra-offset = #'(-0.1 . -0.2)
  \once \override Flag.style = #'no-flag
  \once \override Accidental.extra-offset = #'(4 . -.1)
}

\relative c' {
  << { \fixA <b d!>8 } \ { \voiceThree \fixB dis } >> s
}

```



Horizontale Verschiebung von Noten erzwingen

Wenn es zu Zusammenstößen kommt, kann mit folgender Lösung eine andere Position manuell eingestellt werden. Die Einheiten hier sind Notenlinienzwischenräume.

```

\relative c' <<

```

```

{
  <d g>2 <d g>
}
\\
{
  <b f'>2
  \once \override NoteColumn.force-hshift = 1.7
  <b f'>2
}
>>

```



Making an object invisible using \hide

Applying `\hide` to a grob causes objects of this type to be printed with „invisible ink“. They are not printed, but all of their other behavior is retained:

- the objects still take up space,
- they take part in collision resolution, and
- slurs, ties, and beams can be attached to them as usual.

This snippet demonstrates how to connect different voices using ties. Normally, ties only connect two notes in the same voice. By introducing a tie in a different voice, and blanking the first up-stem in that voice, the tie appears to cross voices.

```

\relative {
  \time 2/4
  <<
  {
    \once \hide Stem
    \once \override Stem.length = #8
    b'8 ~ 8\noBeam
    \once \hide Stem
    \once \override Stem.length = #8
    g8 ~ 8\noBeam
  }
  \\
  {
    b8 g g e
  }
  >>
}

```

```

\paper {
  line-width = 40\mm
  ragged-right = ##f
}

```



Moving dotted notes in polyphony

When a dotted note in the upper voice is moved to avoid a collision with a note in another voice, the default is to move the upper note to the right. This behaviour can be changed setting the `prefer-dotted-right` property of the `NoteCollision` grob.

```
\new Staff \relative c' <<
{
  f2. f4
  \override Staff.NoteCollision.prefer-dotted-right = ##f
  f2. f4
  \override Staff.NoteCollision.prefer-dotted-right = ##t
  f2. f4
}
\\
{ e4 e e e e e e e e e e }
>>
```



Suppressing warnings for clashing note columns

If notes from two voices with stems in the same direction are placed at the same position, and both voices have no shift or the same shift specified, the error message „warning: ignoring too many clashing note columns“ appears when compiling the LilyPond file. This message can be suppressed by setting the `ignore-collision` property of the `NoteColumn` object to `#t`. Please note that this does not just suppress warnings but stops LilyPond trying to resolve collisions at all and so may have unintended results unless used with care.

```
ignore = \override NoteColumn.ignore-collision = ##t
```

```
\relative c' {
  \new Staff <<
    \new Voice { \ignore \stemDown f2 g }
    \new Voice { c2 \stemDown c, }
  >>
}
```



Two \partCombine pairs on one staff

The `\partCombine` function takes two music expressions, each containing a part, and distributes them among four `Voice` contexts named „one“, „two“, „solo“, and „shared“, depending on when and how the parts are merged into a common voice.

Variants of `\partCombine` are `\partCombineUp` and `\partCombineDown` to produce up-stem and down-stem merging of two voices, respectively. Combining them to squeeze four parts into a single staff, however, need some special setup, which this snippet defines accordingly.

```
customPartCombineUp =
#(define-music-function (part1 part2) (ly:music? ly:music?)
```

"Make an up-stem `VoiceBox` context that combines PART1 and PART2.

The context is called 'Up'; internally, the function calls
`\\partCombineUp`."

```
{
  \new VoiceBox = "Up" <<
    \context Voice = "one" { \voiceOne }
    \context Voice = "two" { \voiceThree }
    \context Voice = "shared" { \voiceOne }
    \context Voice = "solo" { \voiceOne }
    \context NullVoice = "null" {}
    \partCombine #part1 #part2
  >>
}
```

customPartCombineDown =

```
#(define-music-function (part3 part4) (ly:music? ly:music?)
  "Make a down-stem `VoiceBox` context that combines PART3 and PART4.
```

The context is called 'Down'; internally, the function calls
`\\partCombineDown`."

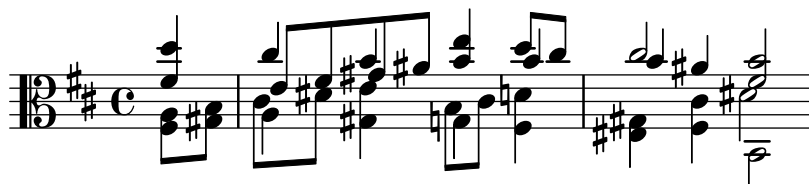
```
{
  \new VoiceBox = "Down" <<
    \set VoiceBox.soloText = #"Solo III"
    \set VoiceBox.soloIIIText = #"Solo IV"
    \context Voice = "one" { \voiceFour }
    \context Voice = "two" { \voiceTwo }
    \context Voice = "shared" { \voiceFour }
    \context Voice = "solo" { \voiceFour }
    \context NullVoice = "null" {}
    \partCombine #part3 #part4
  >>
}
```

```
soprano = { d'4 | cis' b e' d'8 cis' | cis'2 b }
alto = { fis4 | e8 fis gis ais b4 b | b ais fis2 }
tenor = { a8 b | cis' dis' e'4 b8 cis' d'4 | gis cis' dis'2 }
bass = { fis8 gis | a4 gis g fis | eis fis b,2 }
```

```
\new Staff <<
  \key b\minor
  \clef alto
  \partial 4
  \transpose b b' \customPartCombineUp \soprano \alto
  \customPartCombineDown \tenor \bass
>>
```

```
\layout {
  \context {
    \Staff
    \accepts "VoiceBox"
  }
}
```

```
\context {  
  \name "VoiceBox"  
  \type "Engraver_group"  
  \defaultchild "Voice"  
  \accepts "Voice"  
  \accepts "NullVoice"  
}  
}
```



6 Staff notation

See also Abschnitt “Staff notation” in *Notationsreferenz*.

Ambitus pro Stimme hinzufügen

Ambitus können pro Stimme gesetzt werden. In diesem Fall müssen sie manuell verschoben werden, um Zusammenstöße zu verhindern.

```
\new Staff <<
  \new Voice \with {
    \consists "Ambitus_engraver"
  } \relative c'' {
    \override Ambitus.X-offset = 2.0
    \voiceOne
    c4 a d e
    f1
  }
  \new Voice \with {
    \consists "Ambitus_engraver"
  } \relative c' {
    \voiceTwo
    es4 f g as
    b1
  }
}>>
```

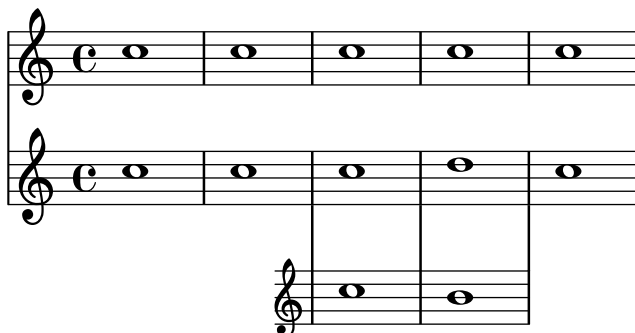


Adding an extra staff

An extra staff can be added (possibly temporarily) after the start of a piece.

```
\score {
  <<
    \new Staff \relative c'' {
      c1 | c | c | c | c
    }
    \new StaffGroup \relative c'' {
      \new Staff {
        c1 | c
      } <<
      { c1 | d }
      \new Staff {
        \once \omit Staff.TimeSignature
        c1 | b
      }
    } >>
  } >>
  c1
}
```

}

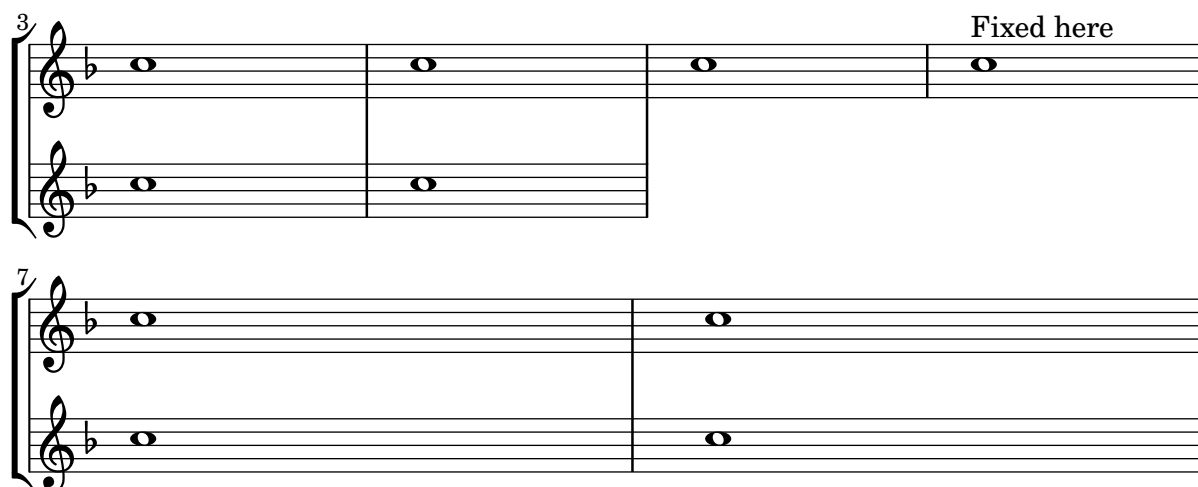


Adding an extra staff at a line break

When adding a new staff at a line break, some extra space is unfortunately added at the end of the line before the break (to fit in a key signature change, which is never printed anyway). The workaround is to set the `explicitKeySignatureVisibility` property of the `Staff` grob as is shown in the example.

```
\score {
  \new StaffGroup \relative c'' {
    \new Staff
    \key f \major
    c1 c^"Unwanted extra space" \break
    << { c1 | c }
    \new Staff {
      \key f \major
      \once \omit Staff.TimeSignature
      c1 | c
    }
  }
  >>
  c1 | c^"Fixed here" \break
  << { c1 | c }
  \new Staff {
    \once \set Staff.explicitKeySignatureVisibility =
      #end-of-line-invisible
    \key f \major
    \once \omit Staff.TimeSignature
    c1 | c
  }
  >>
}
}
```





Adding indicators to staves which get split after a break

This snippet defines the commands `\splitStaffBarLine`, `\convUpStaffBarLine`, and `\convDownStaffBarLine`. These add arrows at a bar line to denote that several voices sharing a staff will each continue on a staff of their own in the next system, or that voices split in this way recombine.

Note that the implementation in this snippet draws dimensionless arrows into the right margin. For normal printing, this doesn't cause problems. However, it is necessary to increase the bounding box horizontally if you render the code as an image to avoid cropping, as demonstrated below.

```
#(define-markup-command (arrow-at-angle layout props angle-deg length fill)
  (number? number? boolean?)
  (let* ((PI-OVER-180 (/ (atan 1 1) 34))
        (degrees->radians (lambda (degrees) (* degrees PI-OVER-180)))
        (angle-rad (degrees->radians angle-deg))
        (target-x (* length (cos angle-rad)))
        (target-y (* length (sin angle-rad))))
    (interpret-markup layout props
      (markup
        #:translate (cons (/ target-x 2) (/ target-y 2))
        #:rotate angle-deg
        #:translate (cons (/ length -2) 0)
        #:concat (#:draw-line (cons length 0)
                          #:arrow-head X RIGHT fill))))))

splitStaffBarLineMarkup = \markup \with-dimensions #'(0 . 0) #'(0 . 0) {
  \combine
  \arrow-at-angle #45 #(sqrt 8) ##t
  \arrow-at-angle #-45 #(sqrt 8) ##t
}

splitStaffBarLine = {
  \once \override Staff.BarLine.stencil =
  #(lambda (grob)
    (ly:stencil-combine-at-edge
      (ly:bar-line::print grob)
      X RIGHT
```



```

        (grob-interpret-markup grob splitStaffBarLineMarkup)
    0))
\break
}

convDownStaffBarLine = {
  \once \override Staff.BarLine.stencil =
  #(lambda (grob)
    (ly:stencil-combine-at-edge
      (ly:bar-line::print grob)
      X RIGHT
      (grob-interpret-markup grob #{
        \markup\with-dimensions #'(0 . 0) #'(0 . 0) {
          \translate #'(0 . -.13)\arrow-at-angle #-45 #(\sqrt 8) ##t
        }#})
    0))
\break
}

convUpStaffBarLine = {
  \once \override Staff.BarLine.stencil =
  #(lambda (grob)
    (ly:stencil-combine-at-edge
      (ly:bar-line::print grob)
      X RIGHT
      (grob-interpret-markup grob #{
        \markup\with-dimensions #'(0 . 0) #'(0 . 0) {
          \translate #'(0 . .14)\arrow-at-angle #45 #(\sqrt 8) ##t
        }#})
    0))
\break
}

\paper {
  indent = 10\mm
  short-indent = 10\mm
  line-width = 8\cm
}

separateSopranos = {
  \set Staff.instrumentName = "AI AII"
  \set Staff.shortInstrumentName = "AI AII"
  \splitStaffBarLine
  \change Staff = "up"
}

convSopranos = {
  \convDownStaffBarLine
  \change Staff = "shared"
  \set Staff.instrumentName = "S A"
  \set Staff.shortInstrumentName = "S A"
}

```

```

sI = {
  \voiceOne
  \repeat unfold 4 f''2
  \separateSopranos
  \repeat unfold 4 g''2
  \convSopranos
  \repeat unfold 4 c''2
}
sII = {
  s1*2
  \voiceTwo
  \change Staff = "up"
  \repeat unfold 4 d''2
}
aI = {
  \voiceTwo
  \repeat unfold 4 a'2
  \voiceOne
  \repeat unfold 4 b'2
  \convUpStaffBarLine
  \voiceTwo
  \repeat unfold 4 g'2
}
aII = {
  s1*2
  \voiceTwo
  \repeat unfold 4 g'2
}
ten = {
  \voiceOne
  \repeat unfold 4 c'2
  \repeat unfold 4 d'2
  \repeat unfold 4 c'2
}
bas = {
  \voiceTwo
  \repeat unfold 4 f2
  \repeat unfold 4 g2
  \repeat unfold 4 c2
}

\markup \pad-x #3 % avoid cropping
\score {
  <<
  \new ChoirStaff <<
  \new Staff = up \with {
    instrumentName = "SI SII"
    shortInstrumentName = "SI SII"
  } {
    s1*4
  }
}

```

```

\new Staff = shared \with {
  instrumentName = "S A"
  shortInstrumentName = "S A"
} <<
  \new Voice = sopI \sI
  \new Voice = sopII \sII
  \new Voice = altI \aI
  \new Voice = altII \aII
>>
\new Lyrics \with {
  alignBelowContext = up
}
\lyricsto sopII { e f g h }
\new Lyrics \lyricsto altI { a b c d e f g h i j k l }

\new Staff = men \with {
  instrumentName = "T B"
  shortInstrumentName = "T B"
} <<
  \clef F
  \new Voice = ten \ten
  \new Voice = bas \bas
>>
\new Lyrics \lyricsto bas { a b c d e f g h i j k l }
>>
>>

\layout {
  \context {
    \Staff \RemoveEmptyStaves
    \override VerticalAxisGroup.remove-first = ##t
  }
}

```

The image displays three musical systems, each illustrating a different method for adding piano accompaniment to vocal stichnoten (piano stichnoten). Each system consists of a vocal staff and a piano staff.

- System 1:** The vocal staff (Soprano, S A) and piano staff (Tenor, T B) both show notes a, b, c, d. The piano staff is positioned below the vocal staff.
- System 2:** The vocal staff (Soprano I, SI SII) and piano staff (Alto I, AI AII) both show notes e, f, g, h. The piano staff is positioned below the vocal staff.
- System 3:** The vocal staff (Soprano, S A) and piano staff (Tenor, T B) both show notes i, j, k, l. The piano staff is positioned below the vocal staff.

Orchesterstichnoten zu einem Klavierauszug hinzufügen

Hier wird gezeigt, wie man das Hinzufügen von vielen Orchesterstichnoten zu einem Klavierauszug hinzufügen kann. Die musikalische Funktion `\cueWhile` braucht vier Argumente: Die Noten, von denen die Stichnoten formatiert werden sollen, definiert durch `\addQuote`, die Bezeichnung, die mit den Noten angegeben werden soll, dann entweder `#UP` (hoch) oder `#DOWN` (runter) zur Angabe von entweder `\voiceOne` mit der Bezeichnung über dem System oder `\voiceTwo` mit der Bezeichnung unter dem System, und schließlich die Klaviermusik, die parallel zu den Stichnoten gespielt werden soll. Die Bezeichnung des Stichnoteninstruments wird links der Stichnoten positioniert. Viele Abschnitte können zitiert werden, aber sie dürfen sich nicht überlappen.

```
cueWhile =
#(define-music-function
  (instrument name dir music)
  (string? string? ly:dir? ly:music?)
  #{
    \cueDuring $instrument #dir {
      \once \override TextScript.self-alignment-X = #RIGHT
```

```

        \once \override TextScript.direction = $dir
        <>-\markup { \tiny #name }
        $music
    }
    #})

flute = \relative c'' {
    \transposition c'
    s4 s4 e g
}
\addQuote "flute" { \flute }

clarinet = \relative c' {
    \transposition bes
    fis4 d d c
}
\addQuote "clarinet" { \clarinet }

singer = \relative c'' { c4. g8 g4 bes4 }
words = \lyricmode { here's the lyr -- ics }

pianoRH = \relative c'' {
    \transposition c'
    \cueWhile "clarinet" "Clar." #DOWN { c4. g8 }
    \cueWhile "flute" "Flute" #UP { g4 bes4 }
}
pianoLH = \relative c { c4 <c' e> e, <g c> }

\score {
  <<
    \new Staff {
      \new Voice = "singer" {
        \singer
      }
    }
    \new Lyrics {
      \lyricsto "singer"
      \words
    }
    \new PianoStaff <<
      \new Staff {
        \new Voice {
          \pianoRH
        }
      }
      \new Staff {
        \clef "bass"
        \pianoLH
      }
    >>
  >>
}

```



Alternative Taktnummerierung

Zwei alternative Methoden können eingestellt werden, die die Taktnummerierung beeinflussen, insbesondere bei Wiederholungen.

```
music = \relative c' {
  \repeat volta 3 {
    c4 d e f |
    \alternative {
      \volta 1 { c4 d e f | c2 d \break }
      \volta 2 { f4 g a b | f4 g a b | f2 a | \break }
      \volta 3 { c4 d e f | c2 d } } }
  c1 \bar "|"
}

{
  \textMark \markup \large "default"
  \music
}

{
  \textMark \markup \large \typewriter "numbers"
  \set Score.alternativeNumberingStyle = #'numbers
  \music
}

{
  \textMark \markup \large \typewriter "numbers-with-letters"
  \set Score.alternativeNumberingStyle = #'numbers-with-letters
  \music
}

\layout {
  \context {
    \Score
    \override TextMark.Y-offset = #5
  }
}
```

default



numbers



numbers-with-letters



Ambitus after key signature

By default, ambitus are positioned at the left of the clef. The `\ambitusAfter` function allows for changing this placement. Syntax is `\ambitusAfter grob-interface`; see Graphical Object Interfaces (<https://lilypond.org/doc/v2.24/Documentation/internals/graphical-object-interfaces>) for a list of possible values for `grob-interface`.

A common use case is printing the ambitus between key signature and time signature.

```
\new Staff \with {
  \consists Ambitus_engraver
} \relative {
  \ambitusAfter key-signature
  \key d \major
  es'8 g bes cis d2
}
```

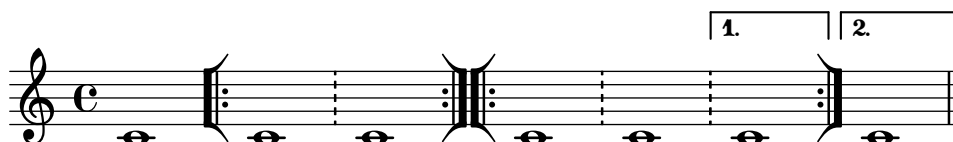


Changing the default bar lines

Default bar lines can be changed when re-defined in a Score context.

```
\layout {
  \context {
    \Score
    % Changing the defaults from engraver-init.ly
    measureBarType = "!"
    startRepeatBarType = "[|:"
    endRepeatBarType = "[:|]"
    doubleRepeatBarType = "[:|][|:"
  }
}

{
  c'1
  \repeat volta 2 { c' c' }
  \repeat volta 2 { c' c' \alternative { \volta 1 { c' }
                                         \volta 2 { c' } } }
  \bar "|."
}
```



Changing the number of lines in a staff

The number of lines in a staff may be changed by overriding the `StaffSymbol` property `line-count`.

```
upper = \relative c'' {
  c4 d e f
}

lower = \relative c {
  \clef bass
  c4 b a g
}

\score {
  \context PianoStaff <<
    \new Staff {
      \upper
    }
    \new Staff {
      \override Staff.StaffSymbol.line-count = #4
      \lower
    }
  >>
```


}



Changing the staff size

The simplest way to resize staves is to use

```
#(set-global-staff-size size)
```

To resize an individual staff's size, you can use the properties `staff-space` and `fontSize`.

```
<<
\new Staff \relative c'' {
  \dynamicDown c8\ff c c c c c c c
}
\new Staff \with {
  fontSize = #-3
  \override StaffSymbol.staff-space = #(magstep -3)
} \relative c {
  \clef bass c8 c c c c\f c c c
}
>>
```



Creating blank staves

To create blank staves, generate empty measures then remove the `Bar_number_engraver` from the `Score` context, and the `Time_signature_engraver`, `Clef_engraver` and `Bar_engraver` from the `Staff` context.

```
#(set-global-staff-size 10) % for the documentation
% #(set-global-staff-size 20) % for letter and A4
```

```
\book {
  \score {
    { \repeat unfold 12 { s1 \break } }

    \layout {
      indent = 0
      \context {
        \Staff
        \remove "Time_signature_engraver"
        \remove "Clef_engraver"
        \remove "Bar_engraver"
      }
    }
  }
}
```

```

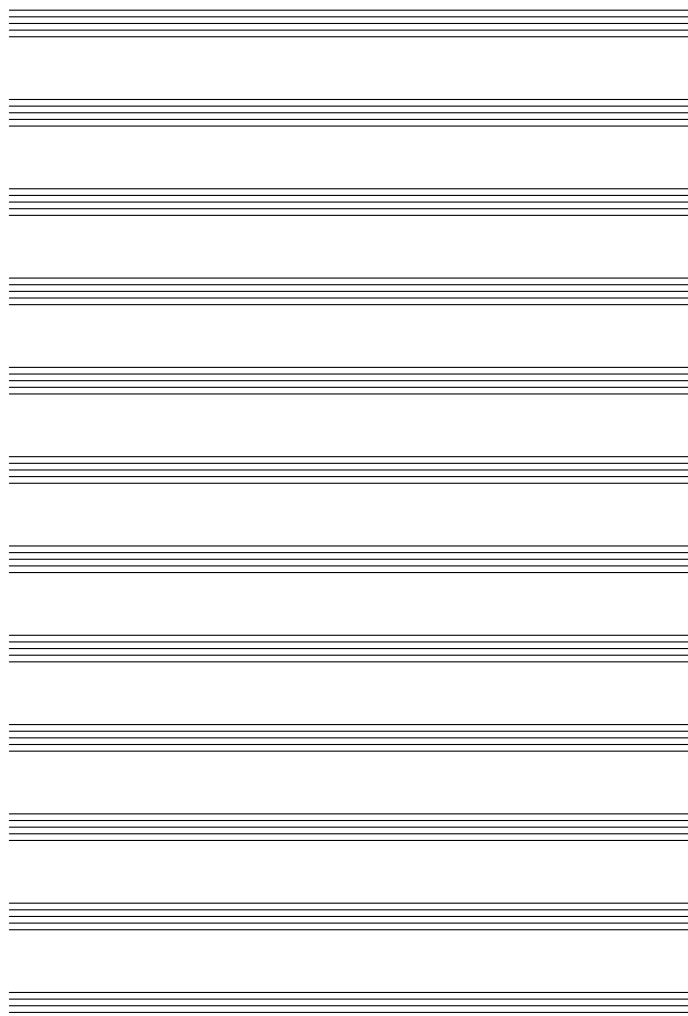
    }
    \context {
      \Score
      \remove "Bar_number_engraver"
    }
  }
}

% for the documentation
\paper {
  #(set-paper-size "a6")
  ragged-last-bottom = ##f
  line-width = 90\mm
  left-margin = 7.5\mm
  bottom-margin = 5\mm
  top-margin = 5\mm
  tagline = ##f
}

% uncomment these lines for "letter" size
%{
\paper {
  #(set-paper-size "letter")
  ragged-last-bottom = ##f
  line-width = 7.5\in
  left-margin = 0.5\in
  bottom-margin = 0.25\in
  top-margin = 0.25\in
  tagline = ##f
}
%}

% uncomment these lines for "A4" size
%{
\paper {
  #(set-paper-size "a4")
  ragged-last-bottom = ##f
  line-width = 180\mm
  left-margin = 15\mm
  bottom-margin = 10\mm
  top-margin = 10\mm
  tagline = ##f
}
%}
}

```



Creating custom key signatures

LilyPond supports custom key signatures. In this example, print for D minor and D major with an extended range of shown flats.

```
\new Staff \with {
  \override StaffSymbol.line-count = #8
  \override KeySignature.flat-positions = #'((-7 . 6))
  \override KeyCancellation.flat-positions = #'((-7 . 6))
  \override KeySignature.sharp-positions = #'((-6 . 7))
  \override KeyCancellation.sharp-positions = #'((-6 . 7))

  \override Clef.stencil =
    #(lambda (grob)
      (grob-interpret-markup grob
        #{ \markup\combine
          \musicglyph "clefs.C"
          \translate #'(-3 . -2)
          \musicglyph "clefs.F"
        #}))
      clefPosition = #3
      middleCPosition = #3
      middleCClefPosition = #3
```

```

}

{
  \key d\minor f bes, f bes, |
  \key d\major fis b, fis b, |
}

```



Hälse über beide Systeme

Dieses Schnipsel zeigt den Einsatz von `Span_stem_engraver` und `\crossStaff`, um Hälse über Systeme hinweg automatisch zu verbinden. Die Länge des Halses muss definiert werden, da der variable Abstand zwischen Notenköpfen und Notensystemen automatisch errechnet wird.

```

\layout {
  \context {
    \PianoStaff
    \consists "Span_stem_engraver"
  }
}

\new PianoStaff <<
  \new Staff {
    <b d'>4 r d'16\> e'8. g8 r\! |
    e'8 f' g'4
    \voiceTwo
    % Down to lower staff
    \crossStaff { e'8 e'8 } e'4 |
  }

  \new Staff {
    \clef bass
    \voiceOne
    % Up to upper staff
    \crossStaff { <e g>4 e, g16 a8. c8 } d |
    g8 f g4 \voiceTwo g8 g g4 |
  }
>>

```

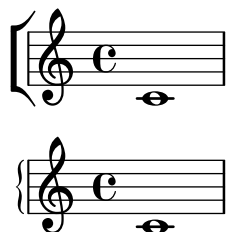


Klammer anzeigen, wenn nur ein System gesetzt wird

Wenn nur ein System einer Systemgruppe vom Typ `ChoirStaff` oder `StaffGroup` angezeigt wird, wird die Klammer zu Beginn normalerweise nicht gesetzt. Das kann verändert werden, indem man die entsprechende Eigenschaft verändert.

Bei Systemen wie `PianoStaff` und `GrandStaff`, die mit einer geschweiften Klammer beginnen, muss eine andere Eigenschaft verändert werden, wie das zweite Beispiel zeigt.

```
\score {
  \new StaffGroup <<
    % Must be lower than the actual number of staff lines
    \override StaffGroup.SystemStartBracket.collapse-height = 4
    \override Score.SystemStartBar.collapse-height = 4
    \new Staff {
      c'1
    }
  >>
}
\score {
  \new PianoStaff <<
    \override PianoStaff.SystemStartBrace.collapse-height = 4
    \override Score.SystemStartBar.collapse-height = 4
    \new Staff {
      c'1
    }
  >>
}
```



Displaying a whole GrandStaff system if only one of its staves is alive

In many orchestral scores it is custom to not show staves for instruments that are silent for a while; this is called a ‚Frenched‘ score. LilyPond provides this functionality via the `\RemoveEmptyStaves` command.

When they play again it is often preferred to show the staves of *all instruments of such a group*. This can be done by adding the `Keep_alive_together_engraver` to the grouping context (e.g., `GrandStaff` or `StaffGroup`).

In the example below the violins are silent in the second system. Only the first violin plays the last measure in the third system but the staff of the second violin is also displayed.

```
\score {
  <<
    \new Staff = "Staff_flute" \with {
      instrumentName = "Flute"
      shortInstrumentName = "Fl"
    } \relative c' {
```

```

\repeat unfold 3 { c'4 c c c | c c c c | c c c c | \break }
}

\new StaffGroup = "StaffGroup_Strings" <<
\new GrandStaff = "GrandStaff_violins" <<
\new Staff = "StaffViolinI" \with {
  instrumentName = "Violin I"
  shortInstrumentName = "Vi I"
} \relative c'' {
  a1 | R1*7 | \repeat unfold 12 a16 a4 |
}
\new Staff = "StaffViolinII" \with {
  instrumentName = "Violin II"
  shortInstrumentName = "Vi II"
} \relative c' {
  e1 | R1*8 |
}
>>

\new Staff = "Staff_cello" \with {
  instrumentName = "Cello"
  shortInstrumentName = "Ce"
} \relative c {
  \clef bass \repeat unfold 9 { c1 } |
}
>>
>>
}

\layout {
  indent = 3.0\cm
  short-indent = 1.5\cm

  \context {
    \GrandStaff
    \consists Keep_alive_together_engraver
  }
  \context {
    \Staff
    \RemoveEmptyStaves
  }
}

```

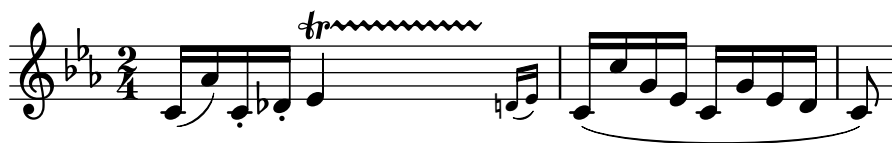
The image displays three musical staves, each representing a different instrument: Flute, Violin I, and Violin II, and Cello. The notation is in 2/4 time, with a key signature of one flat (B-flat). The first staff shows a continuous melody for the Flute. The second and third staves show a trill for Violin I, which is extended in the third staff. The Cello part is shown in the bottom staff of each system, with a single note in the first two systems and a trill in the third system.

Extending a trill spanner

For `TrillSpanner` grobs, the `minimum-length` property becomes effective only if the `set-spacing-rods` procedure is called explicitly.

To do this, the `springs-and-rods` property should be set to `ly:spanner::set-spacing-rods`.

```
\relative c' {
  \key c\minor
  \time 2/4
  c16( as') c,-. des-.
  \once\override TrillSpanner.minimum-length = #15
  \once\override TrillSpanner.springs-and-rods = #ly:spanner::set-spacing-rods
  \afterGrace es4\startTrillSpan { d16[(\stopTrillSpan es)] }
  c( c' g es c g' es d
  c8)
}
```



Extending glissandi across repeats

A glissando that extends into several `\alternative` blocks can be simulated by adding a hidden grace note with a glissando at the start of each `\alternative` block. The grace note should be at the same pitch as the note which starts the initial glissando. This is implemented here with a music function that takes the pitch of the grace note as its argument.

Note that in polyphonic music the grace note must be matched with corresponding grace notes in all other voices.

```
repeatGliss = #(define-music-function (grace)
  (ly:pitch?)
  #{
    % the next two lines ensure the glissando is long enough
    % to be visible
    \once \override Glissando.springs-and-rods
      = #ly:spanner::set-spacing-rods
    \once \override Glissando.minimum-length = 3.5
    \once \hideNotes
    \grace $grace \glissando
  #})

\score {
  \relative c'' {
    \repeat volta 3 { c4 d e f\glissando }
    \alternative {
      { g2 d }
      { \repeatGliss f g2 e }
      { \repeatGliss f e2 d }
    }
  }
}

music = \relative c' {
  \voiceOne
  \repeat volta 2 {
    g a b c\glissando
  }
  \alternative {
    { d1 }
    { \repeatGliss c \once \omit StringNumber e1\2 }
  }
}

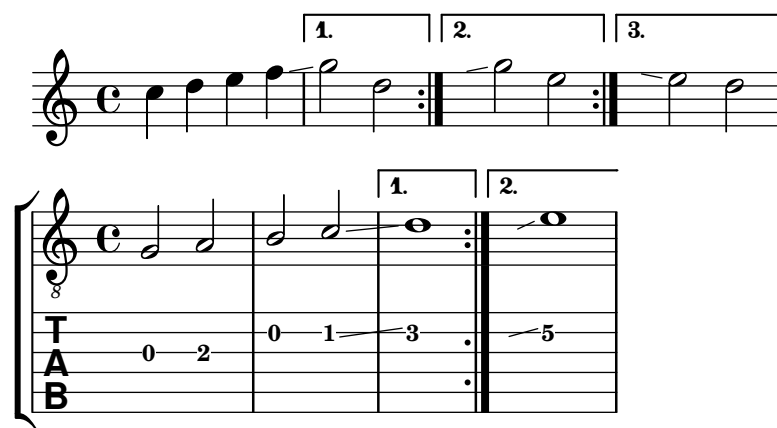
\score {
  \new StaffGroup <<
    \new Staff <<
      \new Voice { \clef "G_8" \music }
    >>
  >>
}
```



```

\new TabStaff <<
  \new TabVoice { \clef "moderntab" \music }
>>
>>
}

```



Flat ties

This snippet provides a function `flared-tie` to draw a tie that consist of straight lines. It is intended as a replacement for the default tie-drawing function (i.e., a replacement argument for the `stencil` property of the `Tie` grob).

The argument of `flared-tie` is a list of coordinate pairs that specify additional points between the first and last point to span up the tie's lines. The first and last point are identical to the original tie's start and end point, respectively. The X and Y coordinate values are multiples of the bounding box length and height of the original tie (also taking care of the tie's direction); consequently, the first point has coordinates (0,0), and the last point (1,0).

The function `flare-tie` defines a shorthand for a flat tie. Further tweaking of the shape is possible by overriding `Tie.details.height-limit` or with `\shape`. It is also possible to change the custom definition on the fly.

```

#(define ((flared-tie coords) grob)
  (define (pair-to-list pair)
    (list (car pair) (cdr pair)))

  (define (normalize-coords goods x y dir)
    (map
      (lambda (coord)
        (cons (* x (car coord)) (* y dir (cdr coord))))
      goods))

  (define (my-c-p-s points thick)
    (make-connected-path-stencil points thick 1.0 1.0 #f #f))

  ;; Calling `ly:tie::print` and assigning its return value to a
  ;; variable in this outer `let` triggers LilyPond to position the
  ;; tie, allowing us to extract its extents. We only proceed,
  ;; however, if the tie doesn't get discarded (for whatever reason).
  (let ((sten (ly:tie::print grob)))
    (if (grob::is-live? grob)
        (let* ((layout (ly:grob-layout grob))

```

```

        (line-thickness (ly:output-def-lookup layout
                                'line-thickness))
        (thickness (ly:grob-property grob 'thickness 0.1))
        (used-thick (* line-thickness thickness))
        (dir (ly:grob-property grob 'direction))
        (xex (ly:stencil-extent sten X))
        (yex (ly:stencil-extent sten Y))
        (lenx (interval-length xex))
        (leny (interval-length yex))
        (xtrans (car xex))
        (ytrans (if (> dir 0)(car yex) (cdr yex)))
        ;; Add last point.
        (coord-list (append coords '((1.0 . 0.0))))
        (uplist
         (map pair-to-list
              (normalize-coords coord-list lenx (* leny 2) dir))))
    (ly:stencil-translate
     (my-c-p-s uplist used-thick)
     (cons xtrans ytrans)))
'()))

% Define a default tie shape consisting of three straight lines.
#(define flare-tie
  (flared-tie '((0.1 . 0.3) (0.9 . 0.3))))

\relative c' {
  a4~ a
  \once \override Tie.stencil = #flare-tie
  a4~ a \break

  <a c e a c e a c e>~ q
  \once \override Tie.stencil = #flare-tie
  q~ q\break

  <>~\markup \small \typewriter "height-limit = 14"
  \override Tie.details.height-limit = 14
  a'4~ a
  \once \override Tie.stencil = #flare-tie
  a4~ a \break

  <>~\markup \small \typewriter "height-limit = 0.5"
  \override Tie.details.height-limit = 0.5
  a4~ a
  \once \override Tie.stencil = #flare-tie
  a4~ a \break

  \revert Tie.details.height-limit

  <>~\markup \small \typewriter
    "\shape #'((0 . 0) (0 . -1) (0 . -1) (0 . 0))"
  \shape #'((0 . 0) (0 . -1) (0 . -1) (0 . 0)) Tie
  a4~ a

```

```

\once \override Tie.stencil = #flare-tie
\shape #'((0 . 0) (0 . -1) (0 . -1) (0 . 0)) Tie
a4~ a \break

<>^\markup \small \typewriter
      "#(flared-tie '((0.2 . 2) (0.5 . -3) (0.8 . 1)))"
\once \override Tie.stencil =
      #(flared-tie '((0.2 . 2) (0.5 . -3) (0.8 . 1)))
a4~ a
<>_ \markup \small \typewriter
      "#(flared-tie '((0.5 . 2)))"
\once \override Tie.stencil = #(flared-tie '((0.5 . 2)))
a'4~ a
}

```

1

2

3 height-limit = 14

4 height-limit = 0.5

5 \shape #'((0 . 0) (0 . -1) (0 . -1) (0 . 0))
#(flared-tie '((0.2 . 2) (0.5 . -3) (0.8 . 1)))

6 #(flared-tie '((0.5 . 2)))

Forcing measure width to adapt to a metronome mark's width

By default, metronome marks do not influence horizontal spacing. This can be solved through a simple override, as shown in the second half of the example.

```

example = {
  R1
  \tempo "Allegro molto" R1*6
  \tempo "poco rit." R1*2
}

```

```

\tempo "a tempo" R1*8 \break
}

{
  \compressMMRests {
    \example
    \override Score.MetronomeMark.extra-spacing-width = #'(-3 . 0)
    \example
  }
}

\layout {
  ragged-right = ##t
}

```

The image shows two staves of musical notation. The first staff starts with a treble clef and a common time signature 'C'. It contains four measures: a whole rest, a quarter note, a half note, and a whole note. Above the staff, tempo markings are placed: 'Allegro molto' above the first measure, 'poco rit.' above the second measure, and 'a tempo' above the third measure. Below the staff, bar line thicknesses are indicated: '6' for the first bar line, '2' for the second, and '8' for the third. The second staff is labeled '18' at the beginning and contains the same sequence of notes and rests as the first staff, with the same tempo markings and bar line thicknesses.

Glissando kann Grobs überspringen

NoteColumn-Grobs können bei Glissandos übersprungen werden.

```

\relative c' {
  a2 \glissando
  \once \override NoteColumn.glissando-skip = ##t
  f''4 d,
}

```

The image shows a single staff of musical notation with a treble clef and a common time signature 'C'. It contains a sequence of notes: a half note 'a', followed by a glissando line (a wavy line) leading to a quarter note 'f''4', which is then followed by a quarter note 'd,'.

Harmonizing bar line thickness for staves with different sizes

When using `\magnifyStaff` only for some staves in a `StaffGroup`, `BarLine` grobs do not align any more due to its changed properties `thick-thickness`, `hair-thickness`, and `kern`.

To fix this, multiple workarounds are available, as demonstrated below.

```

\markuplist {
  % First row.
  \fill-line {
    \score {
      \new StaffGroup <<
        \new Staff \with { \magnifyStaff #1/2 } {
          \textMark \markup \tiny "default"
          b1 b \bar "|."
        }
      }
    }
}

```

```

    \new Staff { b b }
  >>
}
\score {
  \new StaffGroup <<
    \new Staff \with { \magnifyStaff #1/2 } {
      \textMark \markup \tiny \column { "reverting only the"
                                         "final bar line" }

      b1 b
      \revert Staff.BarLine.thick-thickness
      \revert Staff.BarLine.hair-thickness
      \revert Staff.BarLine.kern
      \bar "|."
    }
    \new Staff { b b }
  >>
}
\score {
  \new StaffGroup <<
    \new Staff \with { \magnifyStaff #1/2
                        #(revert-props 'magnifyStaff 0
                                       '((BarLine thick-thickness)
                                         (BarLine hair-thickness)
                                         (BarLine kern))) } {

      \textMark \markup \tiny \column { "cancelling"
                                         \typewriter "\magnifyStaff"
                                         "only for bar lines" }

      b1 b \bar "|."
    }
    \new Staff { b b }
  >>
}
}

\vspace #2

% Second row.
\fill-line {
  \score {
    \new StaffGroup <<
      \new Staff \with { \magnifyStaff #1/2 } {
        \textMark \markup \tiny \column { "mimicking"
                                           \typewriter "\magnifyStaff"
                                           "on the other staves" }

        b1 b \bar "|." }
      \new Staff \with { #(scale-props 'magnifyStaff 1/2 #t
                                       '((BarLine thick-thickness)
                                         (BarLine hair-thickness)
                                         (BarLine kern))) } {

        b b }
    >>
  }
}

```

```

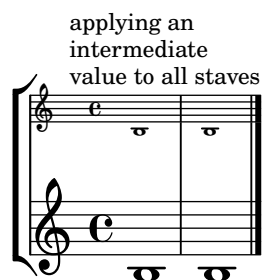
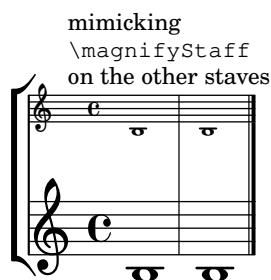
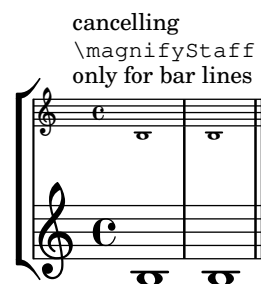
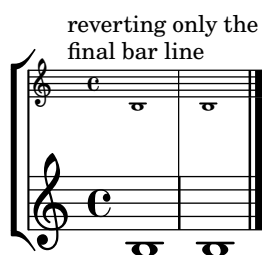
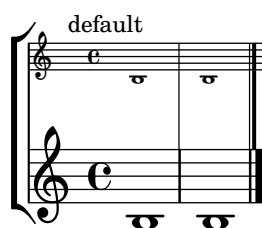
\score {
  \new StaffGroup <<
    \new Staff \with { \magnifyStaff #1/2
      #(scale-props 'magnifyStaff 3/2 #t
        '((BarLine thick-thickness)
          (BarLine hair-thickness)
          (BarLine kern))) } {

      \textMark \markup \tiny \column { "applying an"
        "intermediate"
        "value to all staves" }

      b1 b \bar "|." }
    \new Staff \with { #(scale-props 'magnifyStaff 3/4 #t
      '((BarLine thick-thickness)
        (BarLine hair-thickness)
        (BarLine kern))) } {

      b b }
    >>
  }
  ""
}

```



Incipit

When transcribing mensural music, an incipit at the beginning of the piece is useful to indicate the original key and tempo. While today musicians are used to bar lines in order to faster recognize rhythmic patterns, bar lines were not yet invented during the period of mensural music; in fact, the meter often changed after every few notes. As a compromise, bar lines are often printed between the staves rather than on the staves.

% A short excerpt from the Jubilate Deo by Orlande de Lassus

```
global = {
```

```

\set Score.skipBars = ##t
\key g \major
\time 4/4

% the actual music
\skip 1*8

% let finis bar go through all staves
\override Staff.BarLine.transparent = ##f

% finis bar
\bar "|."
}

discantusIncipit = \new PetrucciStaff {
  \clef "petrucci-c1"
  \key f \major
  \time 2/2
  c'1.
}

discantusNotes = {
  \transpose c' c'' {
    \clef "treble"
    d'2. d'4 |
    b e' d'2 |
    c'4 e'4.( d'8 c' b |
    a4) b a2 |
    b4.( c'8 d'4) c'4 |
    \once \hide NoteHead
    c'1 |
    b\breve |
  }
}

discantusLyrics = \lyricmode {
  Ju -- bi -- la -- te De -- o,
  om -- nis ter -- ra, __ om-
  "...
  -us.
}

altusIncipit = \new PetrucciStaff {
  \clef "petrucci-c3"
  \key f \major
  \time 2/2
  e'1\rest f'1.
}

altusNotes = {
  \transpose c' c'' {
    \clef "treble"

```

```

    r2 g2. e4 fis g |
    a2 g4 e |
    fis g4.( fis16 e fis4) |
    g1 |
    \once \hide NoteHead
    g1 |
    g\breve |
  }
}

altusLyrics = \lyricmode {
  Ju -- bi -- la -- te
  De -- o, om -- nis ter -- ra,
  "...
  -us.
}

tenorIncipit = \new PetrucciStaff {
  \clef "petrucci-c4"
  \key f \major
  \time 2/2
  r\longa
  r\breve
  r1 c'1.
}

tenorNotes = {
  \transpose c' c' {
    \clef "treble_8"
    R1 |
    R1 |
    R1 |
    % two measures
    r2 d'2. d'4 b e' |
    \once \hide NoteHead
    e'1 |
    d'\breve |
  }
}

tenorLyrics = \lyricmode {
  Ju -- bi -- la -- te
  "...
  -us.
}

bassusIncipit = \new PetrucciStaff {
  % The original print shows the b flat
  % for the f major key signature twice.
  \override Staff.KeySignature.flat-positions = #'((-7 . 6))
  \clef "mensural-f"
  \key f\major

```



```

\time 2/2
\tweak Y-offset #1 r\longa \tweak Y-offset #1 r\longa
f1.
}

bassusNotes = {
  \transpose c' c' {
    \clef "bass"
    R1 |
    R1 |
    R1 |
    R1 |
    g2. e4 |
    \once \hide NoteHead
    e1 |
    g\breve |
  }
}

bassusLyrics = \lyricmode {
  Ju -- bi-
  "...
  -us.
}

\score {
  <<
  \new StaffGroup = choirStaff <<
    \new Voice = "discantusNotes" <<
      \set Staff.instrumentName = "Discantus"
      \incipit #1 \discantusIncipit
      \global
      \discantusNotes
    >>
    \new Lyrics \lyricsto discantusNotes { \discantusLyrics }
    \new Voice = "altusNotes" <<
      \set Staff.instrumentName = "Altus"
      \global
      \incipit #1 \altusIncipit
      \altusNotes
    >>
    \new Lyrics \lyricsto altusNotes { \altusLyrics }
    \new Voice = "tenorNotes" <<
      \set Staff.instrumentName = "Tenor"
      \global
      \incipit #1 \tenorIncipit
      \tenorNotes
    >>
    \new Lyrics \lyricsto tenorNotes { \tenorLyrics }
    \new Voice = "bassusNotes" <<
      \set Staff.instrumentName = "Bassus"
      \global

```

```

\incipit #1 \bassusIncipit
\bassusNotes
>>
\new Lyrics \lyricsto bassusNotes { \bassusLyrics }
>>
>>
\layout {
  \context {
    \Score
    %% no bar lines in staves or lyrics
    \hide BarLine
  }
  %% the next two instructions keep the lyrics between the bar lines
  \context {
    \Lyrics
    \consists "Bar_engraver"
    \consists "Separating_line_group_engraver"
  }
  \context {
    \Voice
    %% no slurs
    \hide Slur
    %% Comment in the below "\remove" command to allow line
    %% breaking also at those bar lines where a note overlaps
    %% into the next measure. The command is commented out in this
    %% short example score, but especially for large scores, you
    %% will typically yield better line breaking and thus improve
    %% overall spacing if you comment in the following command.
    %%\remove "Forbid_line_break_engraver"
  }
  indent = 5\cm
  incipit-width = 2.5\cm
}
}

```

The image shows a musical score for a four-part setting of "Jubilate Deo". The parts are labeled on the left: Discantus, Altus, Tenor, and Bassus. The Discantus part is in G major and 4/4 time, featuring a melodic line with a repeat sign. The Altus, Tenor, and Bassus parts are in G major and 4/4 time, featuring a simple harmonic setting of the text "Ju - bi - la - te De - o, om -". The Bassus part is in bass clef and has a repeat sign. The score is presented in a two-measure format, with the first measure showing the beginning of the phrase and the second measure showing the continuation.

4

nis ter - ra, om- ... -us.

nis ter - - ra, ... -us.

8 Ju - bi - la - te ... -us.

Ju - bi- ... -us.

Inserting score fragments above a staff, as markups

The `\markup` command is quite versatile. In this snippet, it contains a `\score` block instead of texts or marks.

```
tuning = \markup \score {
  \new Staff \with { \remove "Time_signature_engraver" }
  {
    \clef bass
    <c, g, d g>1
  }
  \layout {
    indent = 0\cm
  }
}

\header {
  title = "Solo Cello Suites"
  subtitle = "Suite IV"
  subsubtitle = \markup { Originalstimmung: \raise #0.5 \tuning }
  tagline = ##f
}

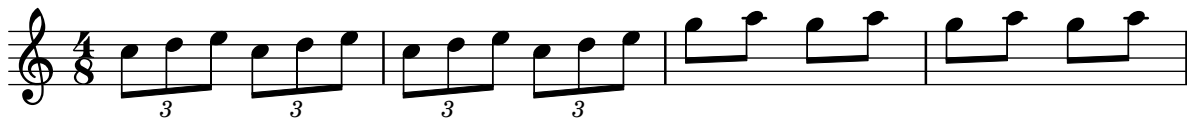
\layout {
  ragged-right = ##f
}

\relative c' {
  \time 4/8
  \tuplet 3/2 { c8 d e } \tuplet 3/2 { c d e }
  \tuplet 3/2 { c8 d e } \tuplet 3/2 { c d e }
  g8 a g a
  g8 a g a
}
```

Solo Cello Suites

Suite IV

Originalstimmung: 



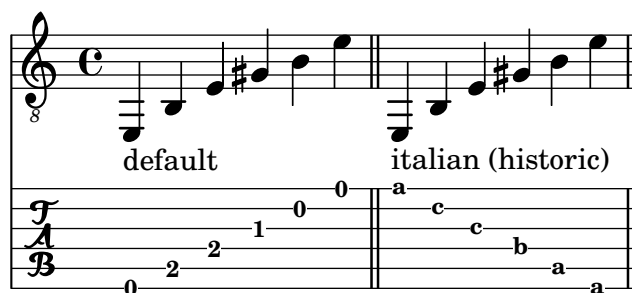
Let TabStaff print the topmost string at bottom

In tablatures, the first string is usually printed topmost. If you want to have it at the bottom, set the `stringOneTopmost` context property to `##f`. For a context-wide setting this could be done in the `\layout` block as well.

```
%\layout {
% \context {
%   \Score
%   stringOneTopmost = ##f
% }
% \context {
%   \TabStaff
%   tablatureFormat = #fret-letter-tablature-format
% }
%}

m = {
  \cadenzaOn
  e, b, e gis! b e'
  \bar "||"
}

<<
\new Staff {
  \clef "G_8"
  <>_"default" \m
  <>_"italian (historic)"\m
}
\new TabStaff
{
  \m
  \set Score.stringOneTopmost = ##f
  \set TabStaff.tablatureFormat = #fret-letter-tablature-format
  \m
}
>>
```

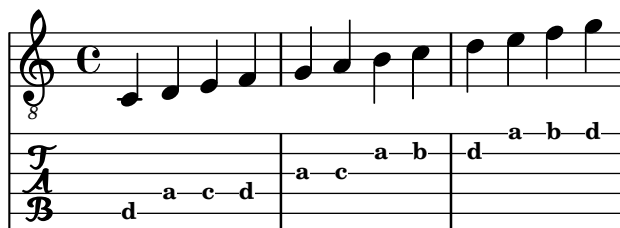


Letter tablature formatting

Tablature can be formatted using letters instead of numbers.

```
music = \relative c {
  c4 d e f
  g4 a b c
  d4 e f g
}

<<
\new Staff {
  \clef "G_8"
  \music
}
\new TabStaff \with {
  tablatureFormat = #fret-letter-tablature-format
} {
  \music
}
>>
```



Making glissandi breakable

Normally, LilyPond refuses to automatically break a line at places where a glissando crosses a bar line. This behavior can be changed by setting the `Glissando.breakable` property to `#t`. Also setting the after-line-breaking property to `#t` makes the glissando line continue after the break.

The breakable property does not affect manual breaks inserted with commands like `\break`.

```
glissandoSkipOn = {
  \override NoteColumn.glissando-skip = ##t
  \hide NoteHead
  \override NoteHead.no-ledgers = ##t
}
```

```
music = {
```

```

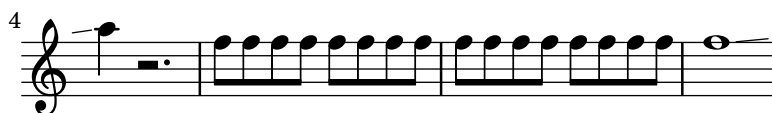
\repeat unfold 16 f8 |
f1\glissando |
a4 r2. |
\repeat unfold 16 f8 |
f1\glissando \once\glissandoSkipOn |
a2 a4 r4 |
\repeat unfold 16 f8
}

\relative c'' {
  <>\markup { \typewriter Glissando.breakable
              set to \typewriter "#t" }
  \override Glissando.breakable = ##t
  \override Glissando.after-line-breaking = ##t
  \music
}

\relative c'' {
  <>\markup { \typewriter Glissando.breakable not set }
  \music
}

\paper {
  line-width = 100\mm
}

```



Eine Linie des Notensystems dicker als die anderen machen

Für den pädagogischen Einsatz kann eine Linie des Notensystems dicker gezeichnet werden (z. B. die Mittellinie, oder um den Schlüssel hervorzuheben). Das ist möglich, indem man zusätzliche Linien sehr nahe an der Linie, die dicker erscheinen soll, einfügt. Dazu wird die `line-positions`-Eigenschaft herangezogen.

```
{
  \override Staff.StaffSymbol.line-positions =
    #'(-4 -2 -0.2 0 0.2 2 4)
  d'4 e' f' g'
}
```



Measure counters

This snippet demonstrates the use of the `Measure_counter_engraver` to number groups of successive measures. Any stretch of measures may be numbered, whether consisting of repetitions or not.

The engraver must be added to the appropriate context. Here, a `Staff` context is used; another possibility is a `Dynamics` context.

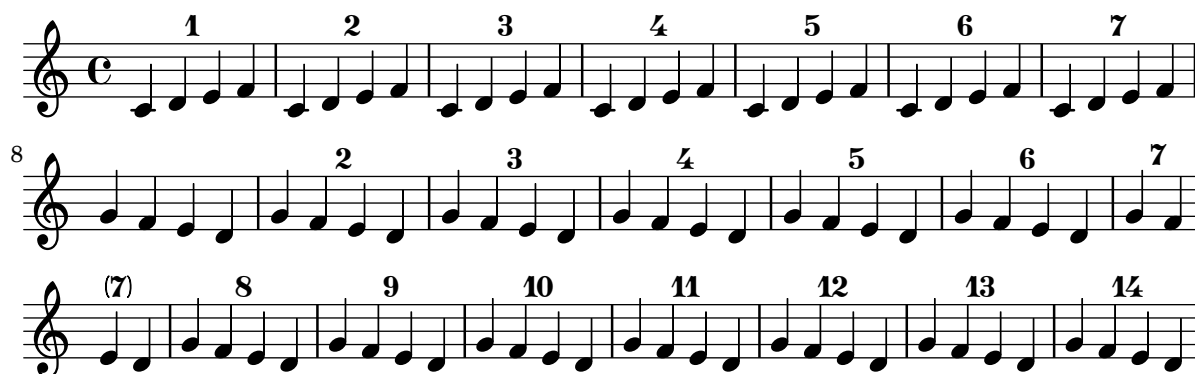
The counter is begun with `\startMeasureCount` and ended with `\stopMeasureCount`. Numbering will start by default with 1, but this behavior may be modified by overriding the `count-from` property.

When a measure extends across a line break, the number will appear twice, the second time in parentheses.

```
\layout {
  \context {
    \Staff
    \consists #Measure_counter_engraver
  }
}

\new Staff {
  \startMeasureCount
  \repeat unfold 7 {
    c'4 d' e' f'
  }
  \stopMeasureCount
  \bar "||"
  g'4 f' e' d'
  \override Staff.MeasureCounter.count-from = #2
  \startMeasureCount
  \repeat unfold 5 {
    g'4 f' e' d'
  }
  g'4 f'
  \bar ""
  \break
  e'4 d'
}
```

```
\repeat unfold 7 {
  g'4 f' e' d'
}
\stopMeasureCount
}
```



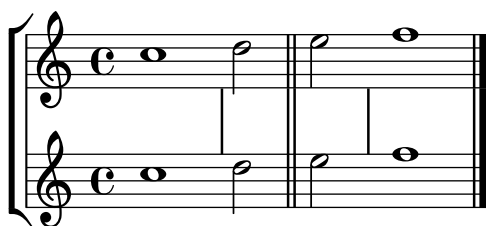
Mensurstriche-Layout (Taktstriche zwischen den Systemen)

Das Mensurstriche-Layout, in welchem die Taktlinien nicht auf den Systemen, sondern zwischen den Systemen gesetzt werden, kann mit einer `StaffGroup` anstelle von `ChoirStaff` erreicht werden. Die Taktlinien auf den Systemen werden mit der `transparent`-Eigenschaft ausgelöscht.

```
\layout {
  \context {
    \Staff
    measureBarType = "-span|"
  }
}
```

```
music = \fixed c'' {
  c1
  d2 \section e2
  f1 \fine
}
```

```
\new StaffGroup <<
  \new Staff \music
  \new Staff \music
>>
```



Modifying the ottava spanner slope

It is possible to change the slope of the ottava spanner.

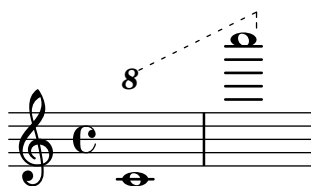
```
\relative c'' {
  \override Staff.OttavaBracket.stencil = #ly:line-spanner::print
```



```

\override Staff.OttavaBracket.bound-details =
  #`((left . ((Y . 0)
    (attach-dir . ,LEFT)
    (padding . 0)
    (stencil-align-dir-y . ,CENTER)))
    (right . ((Y . 5.0) ; Change the number here
    (padding . 0)
    (attach-dir . ,RIGHT)
    (text . ,(make-draw-dashed-line-markup
      (cons 0 -1.2)))))
\override Staff.OttavaBracket.left-bound-info =
  #ly:horizontal-line-spanner::calc-left-bound-info-and-text
\override Staff.OttavaBracket.right-bound-info =
  #ly:horizontal-line-spanner::calc-right-bound-info
\ottava 1
c1
c'''1
}

```



Systeme schachteln

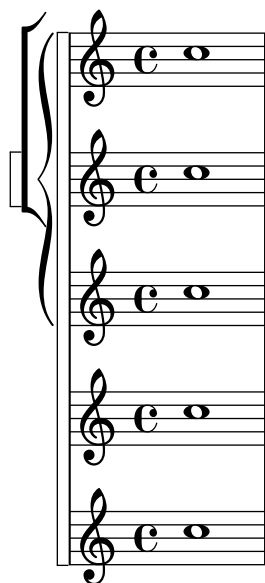
Die Eigenschaft `systemStartDelimiterHierarchy` kann eingesetzt werden, um komplizierte geschachtelte Systemklammern zu erstellen. Der Befehl `\set StaffGroup.systemStartDelimiterHierarchy` nimmt eine Liste mit der Anzahl der Systeme, die ausgegeben werden, auf. Vor jedem System kann eine Systemanfangsklammer angegeben werden. Sie muss in Klammern eingefügt werden und umfasst so viele Systeme, wie die Klammer einschließt. Elemente in der Liste können ausgelassen werden, aber die erste Klammer umfasst immer die gesamte Gruppe. Die Möglichkeiten der Anfangsklammer sind: `SystemStartBar`, `SystemStartBracket`, `SystemStartBrace` und `SystemStartSquare`.

```

\new StaffGroup
\relative c' ' <<
\override StaffGroup.SystemStartSquare.collapse-height = 4
\set StaffGroup.systemStartDelimiterHierarchy
  = #'(SystemStartSquare
    (SystemStartBrace
      (SystemStartBracket a
        (SystemStartSquare b))
      c)
    d)

\new Staff { c1 }
\new Staff { c1 }
\new Staff { c1 }
\new Staff { c1 }
\new Staff { c1 }
>>

```



Untypische Tonarten

Der üblicherweise benutzte `\key`-Befehl setzt die `keySignature`-Eigenschaft im `Staff`-Kontext.

Um untypische Tonartenvorzeichen zu erstellen, muss man diese Eigenschaft direkt setzen. Das Format für den Befehl ist eine Liste: `\set Staff.keySignature = #`(((Oktave . Schritt) . Alteration) ((Oktave . Schritt) . Alteration) ...)` wobei für jedes Element in der Liste Oktave die Oktave angibt (0 ist die Oktave vom eingestrichenen C bis zum eingestrichenen H), Schritt gibt die Note innerhalb der Oktave an (0 heißt C und 6 heißt H), und Alteration ist `,SHARP`, `,FLAT`, `,DOUBLE-SHARP` usw. (Beachte das beginnende Komma.)

Alternativ kann auch jedes Element der Liste mit dem allgemeineren Format `(Schritt . Alteration)` gesetzt werden, wobei dann die Einstellungen für alle Oktaven gelten.

Hier ein Beispiel einer möglichen Tonart für eine Ganztonleiter:

```
\include "arabic.ly"
```

```
\relative do' {
  \set Staff.keyAlterations = #`((0 . ,SEMI-FLAT)
                                (1 . ,SEMI-FLAT)
                                (2 . ,FLAT)
                                (5 . ,FLAT)
                                (6 . ,SEMI-FLAT))

  % \set Staff.extraNatural = ##f
  re reb \dwn reb resd
  dod dob dosd \dwn dob |
  dobsb dods do do |
}
```



Orchester, Chor und Klavier

Diese Vorlage zeigt die Benutzung von geschachtelten `StaffGroup`- und `GrandStaff`-Kontexte, um Instrumente in Untergruppen zu unterteilen, und die Benutzung von `\transpose` für transponierende Instrumente. Alle Noten werden in C geschrieben. Noten können in C eingegeben

werden, oder auch in der Tonart des Instrumentes: dann müssen sie zuerst nach C transponiert werden, bevor sie einer Variable zugewiesen werden.

```

#(set-global-staff-size 17)

\paper {
  indent = 3.0\cm % add space for instrumentName
  short-indent = 1.5\cm % add less space for shortInstrumentName
}

fluteMusic = \relative c' { \key g \major g'1 b }

% Pitches as written on a manuscript for Clarinet in A
% are transposed to concert pitch.
clarinetMusic = \transpose c' a
  \relative c'' { \key bes \major bes1 d }

trumpetMusic = \relative c { \key g \major g''1 b }

% Key signature is often omitted for horns
hornMusic = \transpose c' f
  \relative c { d'1 fis }

percussionMusic = \relative c { \key g \major g1 b }

sopranoMusic = \relative c'' { \key g \major g'1 b }
sopranoLyrics = \lyricmode { Lyr -- ics }

altoIMusic = \relative c' { \key g \major g'1 b }
altoILyrics = \sopranoLyrics
altoIIMusic = \relative c' { \key g \major g'1 b }
altoIILyrics = \lyricmode { Ah -- ah }

tenorMusic = \relative c' { \clef "treble_8" \key g \major g1 b }
tenorLyrics = \sopranoLyrics

pianoRHMus = \relative c { \key g \major g''1 b }
pianoLHMus = \relative c { \clef bass \key g \major g1 b }

violinIMusic = \relative c' { \key g \major g'1 b }
violinIIMusic = \relative c' { \key g \major g'1 b }

violaMusic = \relative c { \clef alto \key g \major g'1 b }

celloMusic = \relative c { \clef bass \key g \major g1 b }

bassMusic = \relative c { \clef "bass_8" \key g \major g,1 b }

\book {
  \score {
    <<
    \new StaffGroup = "StaffGroup_woodwinds" <<

```

```

\new Staff = "Staff_flute" \with { instrumentName = "Flute" }
  \fluteMusic

\new Staff = "Staff_clarinet" \with {
  instrumentName = \markup { \concat { "Clarinet in B" \flat } }
}
% Declare that written Middle C in the music
% to follow sounds a concert B flat, for
% output using sounded pitches such as MIDI.
%\transposition bes

% Print music for a B-flat clarinet
\transpose bes c' \clarinetMusic
>>

\new StaffGroup = "StaffGroup_brass" <<
  \new Staff = "Staff_hornI" \with {
    instrumentName = "Horn in F"
  }
  % \transposition f
  \transpose f c' \hornMusic

  \new Staff = "Staff_trumpet" \with {
    instrumentName = "Trumpet in C"
  }
  \trumpetMusic
>>

\new RhythmicStaff = "RhythmicStaff_percussion" \with {
  instrumentName = "Percussion"
}
  \percussionMusic

\new PianoStaff \with {
  instrumentName = "Piano"
} <<
  \new Staff { \pianoRHMusik }
  \new Staff { \pianoLHMusik }
>>

\new ChoirStaff = "ChoirStaff_choir" <<
  \new Staff = "Staff_soprano" \with {
    instrumentName = "Soprano"
  }
  \new Voice = "soprano" \sopranoMusic
  \new Lyrics \lyricsto "soprano" { \sopranoLyrics }

  \new GrandStaff = "GrandStaff_altos" \with {
    \accepts Lyrics
  } <<
  \new Staff = "Staff_altoI" \with {
    instrumentName = "Alto I"
  }

```

```

    }
    \new Voice = "altoI"
    \altoIMusic
    \new Lyrics \lyricsto "altoI" { \altoILyrics }
    \new Staff = "Staff_altoII" \with {
      instrumentName = "Alto II"
    }
    \new Voice = "altoII"
    \altoIIMusic
    \new Lyrics \lyricsto "altoII" { \altoIILyrics }
  >>

  \new Staff = "Staff_tenor" \with {
    instrumentName = "Tenor"
  }
  \new Voice = "tenor" \tenorMusic
  \new Lyrics \lyricsto "tenor" { \tenorLyrics }
  >>

  \new StaffGroup = "StaffGroup_strings" <<
    \new GrandStaff = "GrandStaff_violins" <<
      \new Staff = "Staff_violinI" \with {
        instrumentName = "Violin I"
      }
      \violinIMusic
      \new Staff = "Staff_violinII" \with {
        instrumentName = "Violin II"
      }
      \violinIIMusic
    >>

    \new Staff = "Staff_viola" \with {
      instrumentName = "Viola"
    }
    \violaMusic

    \new Staff = "Staff_cello" \with {
      instrumentName = "Cello"
    }
    \celloMusic

    \new Staff = "Staff_bass" \with {
      instrumentName = "Double Bass"
    }
    \bassMusic
  >>
  >>
}

```

Flute

Clarinet in B \flat

Horn in F

Trumpet in C

Percussion

Piano

Soprano

Alto I

Alto II

Tenor

Violin I

Violin II

Viola

Cello

Double Bass

Lyr - ics

Lyr - ics

Ah - ah

Lyr - ics

8

8

Detailed description: This is a musical score for a symphony orchestra and vocal soloists. The score is written for a full orchestra, including woodwinds (Flute, Clarinet in B-flat, Horn in F, Trumpet in C), percussion, piano, and strings (Violin I, Violin II, Viola, Cello, Double Bass). There are also four vocal soloists: Soprano, Alto I, Alto II, and Tenor. The music is in 2/4 time, with a key signature of one sharp (F#). The score shows the first two measures of the piece. The woodwinds and strings play a simple harmonic pattern, while the vocalists enter with the lyrics 'Lyr - ics'. The Alto II part has a vocalization 'Ah - ah' in the second measure. The percussion part has a simple rhythmic pattern. The piano part has a simple harmonic pattern. The strings play a simple harmonic pattern. The vocalists enter with the lyrics 'Lyr - ics'.

Print chord names with same root and different bass as slash and bass note

To print subsequent ChordNames only differing in its bass note as slash and bass note, use the Scheme engraver defined in this snippet. The behaviour may be controlled in detail by the chordChanges context property.

```
#(define Bass_changes_equal_root_engraver
  (lambda (ctx)
    "For sequential `ChordNames` with the same root but a different bass,
    the root markup is dropped: D D/C D/B -> D /C /B.
    The behaviour may be controlled by setting the `chordChanges` context
    property."
    (let ((chord-pitches '())
          (last-chord-pitches '())
          (bass-pitch #f))
      (make-engraver
        ((initialize this-engraver)
         (let ((chord-note-namer (ly:context-property ctx
                                                    'chordNoteNamer)))
           ;; Set 'chordNoteNamer, respect user setting if already done
           (ly:context-set-property! ctx 'chordNoteNamer
                                     (if (procedure? chord-note-namer)
                                         chord-note-namer
                                         note-name->markup))))
          (listeners
            ((note-event this-engraver event)
             (let* ((pitch (ly:event-property event 'pitch))
                    (pitch-name (ly:pitch-notename pitch))
                    (pitch-alt (ly:pitch-alteration pitch))
                    (bass (ly:event-property event 'bass #f))
                    (inversion (ly:event-property event 'inversion #f)))
               ;; Collect notes of the chord
               ;; - to compare inversed chords we need to collect the
               ;;   bass note as usual member of the chord, whereas an
               ;;   added bass must be treated separate from the usual
               ;;   chord-notes
               ;; - notes are stored as pairs containing their
               ;;   pitch-name (an integer), i.e. disregarding their
               ;;   octave and their alteration
               (cond (bass (set! bass-pitch pitch))
                     (inversion
                      (set! bass-pitch pitch)
                      (set! chord-pitches
                           (cons (cons pitch-name pitch-alt)
                                  chord-pitches)))
                     (else
                      (set! chord-pitches
                           (cons (cons pitch-name pitch-alt)
                                  chord-pitches)))))))
            (acknowledgers
              ((chord-name-interface this-engraver grob source-engraver)
```

```

(let ((chord-changes (ly:context-property ctx
                                           'chordChanges #f)))
  ;; If subsequent chords are equal apart from their bass,
  ;; reset the 'text-property.
  ;; Equality is done by comparing the sorted lists of this
  ;; chord's elements and the previous chord. Sorting is
  ;; needed because inverted chords may have a different
  ;; order of pitches. `chord-changes` needs to be true.
  (if (and bass-pitch
            chord-changes
            (equal?
             (sort chord-pitches car<)
             (sort last-chord-pitches car<)))
      (ly:grob-set-property!
       grob 'text
       (make-line-markup
        (list
         (ly:context-property ctx 'slashChordSeparator)
         (ly:context-property ctx 'chordNoteNamer)
         bass-pitch
         (ly:context-property ctx
                               'chordNameLowercaseMinor))))))
      (set! last-chord-pitches chord-pitches)
      (set! chord-pitches '())
      (set! bass-pitch #f))))

((finalize this-engraver)
 (set! last-chord-pitches '()))))

myChords = \chordmode {
  % \germanChords

  \set chordChanges = ##t
  d2:m d:m/cis

  d:m/c
  \set chordChanges = ##f
  d:m/b

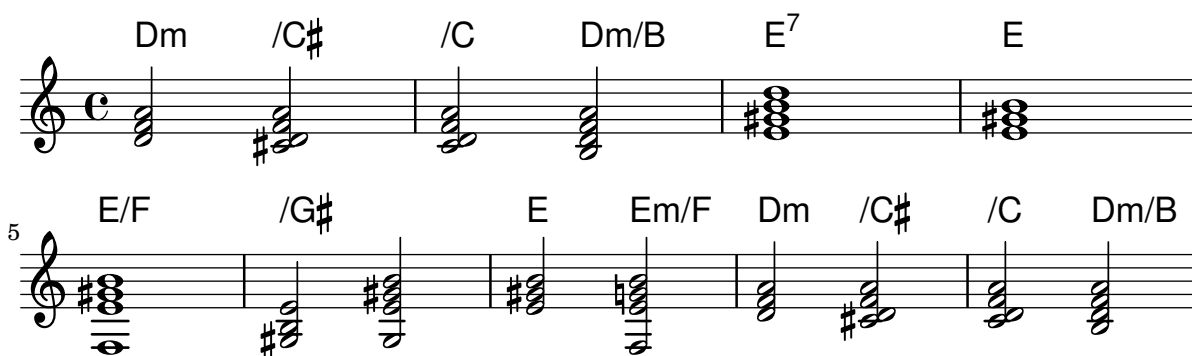
  e1:7
  \set chordChanges = ##t
  e
  \break

  \once \set chordChanges = ##f
  e1/f
  e2/gis e/+gis e e:m/f d:m d:m/cis d:m/c
  \set chordChanges = ##f
  d:m/b
}

```



```
<<
\new ChordNames
  \with { \consists #Bass_changes_equal_root_engraver }
  \myChords
\new Staff \myChords
>>
```



Putting lyrics inside the staff

Lyrics can be moved vertically to place them inside the staff. The lyrics are moved with `\override LyricText.extra-offset = #'(0 . dy)`, and there are similar commands to move the extenders and hyphens. A good value for *dy* must be found by trial and error.

```
<<
\new Staff <<
  \new Voice = "voc" \relative c' { \stemDown a bes c8 b c4 }
>>
\new Lyrics \with {
  \override LyricText.extra-offset = #'(0 . 8.6)
  \override LyricExtender.extra-offset = #'(0 . 8.6)
  \override LyricHyphen.extra-offset = #'(0 . 8.6)
} \lyricsto "voc" { La la -- la __ _ la }
>>
```



Eine andere Stimme zitieren

Die `quotedEventTypes`-Eigenschaft bestimmt die Typen an Musikereignissen, die zitiert werden. Die Standardeinstellung ist `(note-event rest-event)`, womit nur Noten und Pausen der zitierten Stimme für den `\quoteDuring`-Ausdruck übernommen werden. Im Beispiel hier wird die 16-Pause nicht übernommen, weil sich `rest-event` nicht in `quotedEventTypes` befindet.

```
quoteMe = \relative c' {
  fis4 r16 a8.-> b4\ff c
}
\addQuote quoteMe \quoteMe

original = \relative c'' {
  c8 d s2
```

```

\once \override NoteColumn.ignore-collision = ##t
es8 gis8
}

<<
\new Staff \with { instrumentName = "quoteMe" }
\quoteMe

\new Staff \with { instrumentName = "orig" }
\original

\new Staff \with {
  instrumentName = "orig+quote"
  quotedEventTypes = #'(note-event articulation-event)
}
\relative c''
<<
\original
\new Voice {
  s4
  \set fontSize = #-4
  \override Stem.length-fraction = #(magstep -4)
  \quoteDuring "quoteMe" { \skip 2. }
}
>>
>>

```

Eine Stimme mit Transposition zitieren

Zitate berücksichtigen sowohl die Transposition der Quelle als auch des Zielinstruments. In diesem Beispiel spielen alle Instrumente klingendes C, das Zielinstrument ist in F. Die Noten für das Zielinstrument können mit `\transpose` transponiert werden, in diesem Fall werden alle Noten (auch die zitierten) transponiert.

```

\addQuote clarinet {
  \transposition bes
  \repeat unfold 8 { d'16 d' d'8 }
}

\addQuote sax {
  \transposition es'
  \repeat unfold 16 { a8 }
}

```

```
quoteTest = {
  % french horn
  \transposition f
  g'4
  << \quoteDuring "clarinet" { \skip 4 } s4^"clar." >>
  << \quoteDuring "sax" { \skip 4 } s4^"sax." >>
  g'4
}

{
  \new Staff \with {
    instrumentName = \markup { \column { Horn "in F" } }
  }
  \quoteTest
  \transpose c' d' << \quoteTest s4_"up a tone" >>
}
```



Removing brace on first line of piano score

This snippet removes the first brace from a PianoStaff or a GrandStaff, together with the clefs. It may be useful when cutting and pasting the engraved image into existing music.

The code uses `\alterBroken` to hide the brace delimiter at the beginning.

```
someMusic = {
  \once \omit Staff.Clef
  \once \omit Staff.TimeSignature
  \repeat unfold 3 c1 \break
  \repeat unfold 5 c1 \break
  \repeat unfold 5 c1
}

\score {
  \new PianoStaff
  <<
    \new Staff = "right" \relative c' { \someMusic
    \new Staff = "left" \relative c' { \clef F \someMusic }
  >>
  \layout {
    indent=75\mm
    \context {
      \PianoStaff
      \alterBroken transparent #'(#t) SystemStartBrace
    }
  }
}
```

The image shows three musical notation examples. The top example is a single system with three staves, each containing a whole note. The middle example is a system with two staves, each containing a whole note, with a brace on the left and a measure number '4' above the first staff. The bottom example is a system with two staves, each containing a whole note, with a brace on the left and a measure number '9' above the first staff. In all examples, the staves are connected by vertical bar lines.

Removing connecting bar lines on StaffGroup, PianoStaff, or GrandStaff

By default, bar lines in `StaffGroup`, `PianoStaff`, or `GrandStaff` contexts are connected between the staves, i.e., a span bar is printed. This behaviour can be overridden on a staff-by-staff basis.

```
\relative c' {
  \new StaffGroup <<
    \new Staff {
      e1 | e
      \once \override Staff.BarLine.allow-span-bar = ##f
      e1 | e | e
    }
    \new Staff {
      c1 | c | c
      \once \override Staff.BarLine.allow-span-bar = ##f
      c1 | c
    }
    \new Staff {
      a1 | a | a | a | a
    }
  >>
}
```

The image shows a musical notation example of a GrandStaff with three staves. The top two staves are treble clefs, and the bottom staff is a bass clef. The top staff contains five whole notes, the middle staff contains five whole notes, and the bottom staff contains five whole notes. The staves are connected by a brace on the left, but there are no vertical bar lines connecting the staves.

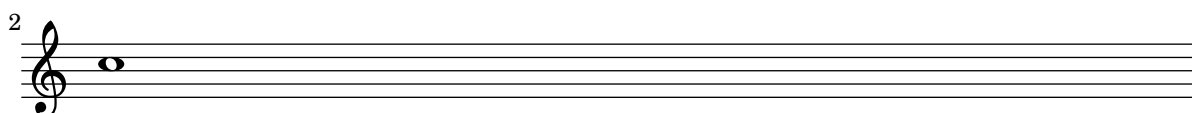
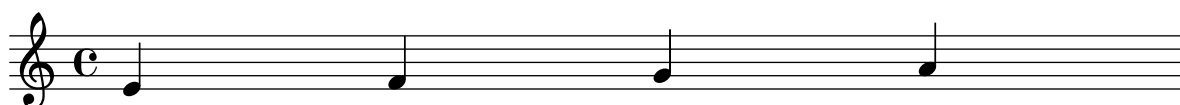
Die erste leere Notenzeile auch entfernen

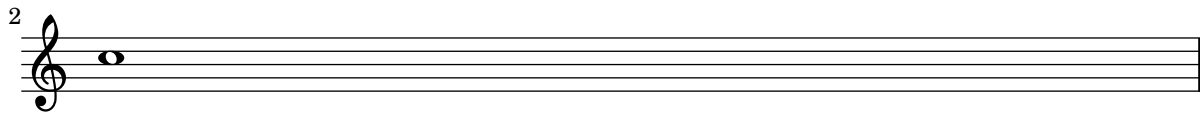
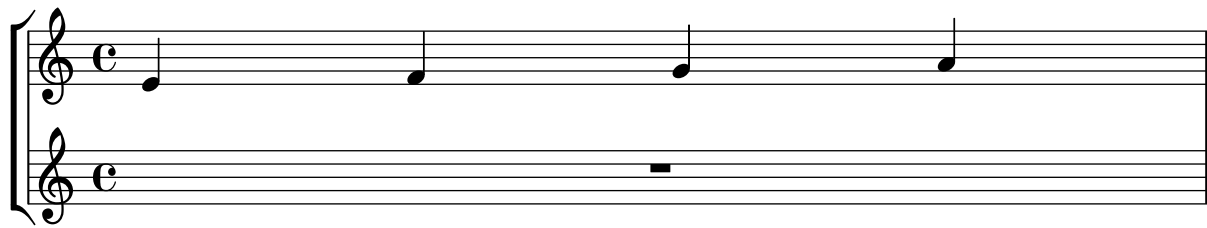
Ein leeres Notensystem kann auch aus der ersten Zeile einer Partitur entfernt werden, indem die Eigenschaft `remove-first` der `VerticalAxisGroup`-Eigenschaft eingesetzt wird. Das kann man global in einer `\layout`-Umgebung oder lokal in dem bestimmten Notensystem machen, das entfernt werden soll. In letzterem Fall muss man den Kontext angeben.

Das untere Notensystem der zweiten Systemgruppe wird nicht entfernt, weil in die Einstellungen in dem Schnipsel nur für das eine Notensystem gültig sind.

```
\layout {
  \context {
    \Staff \RemoveEmptyStaves
    % To use the setting globally, uncomment the following line:
    % \override VerticalAxisGroup.remove-first = ##t
  }
}
\new StaffGroup <<
  \new Staff \relative c' {
    e4 f g a \break
    c1
  }
  \new Staff {
    % To use the setting globally, comment this line,
    % uncomment the line in the \layout block above
    \override Staff.VerticalAxisGroup.remove-first = ##t
    R1 \break
    R
  }
>>

\new StaffGroup <<
  \new Staff \relative c' {
    e4 f g a \break
    c1
  }
  \new Staff {
    R1 \break
    R
  }
>>
```





Setting system separators

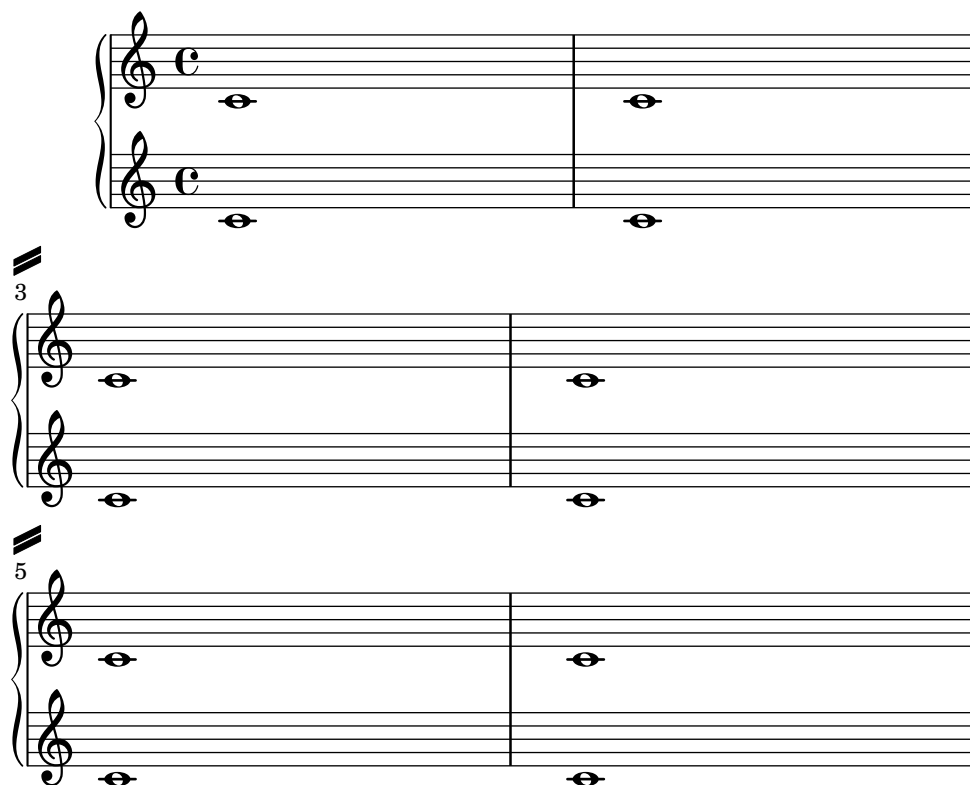
System separators can be inserted between systems. Any markup can be used, but `\slashSeparator` has been provided as a sensible default.

```
#(set-default-paper-size "a5")

\paper {
  system-separator-markup = \slashSeparator
  tagline = ##f
}

notes = \relative c' {
  c1 | c \break
  c1 | c \break
  c1 | c
}

\book {
  \score {
    \new GrandStaff <<
      \new Staff \notes
      \new Staff \notes
    >>
  }
}
```



Shape individual ties in chords

To shape individual ties in chords use the method demonstrated below.

```
{
  \textMark "Chords can be tied note by note."
  <c'~ e'~ g'~ c''~>2 q
}

{
  \textMark \markup \override #'(baseline-skip . 3) \wordwrap {
    Modifying those ties with \typewriter "\\shape" does not succeed,
    because \typewriter TieColumn positions them on its own behalf,
    ignoring \typewriter "\\shape" input more or less. You may
    circumvent this by setting \typewriter positioning-done to
    \typewriter "#t" -- alas, \typewriter positioning-done is an
    internal property, and setting it to \typewriter "#t" means: all
    positioning is done, don't do anything further. The next example
    demonstrates a case where the positioning is not finished: all tie
    directions are down, and the thickness is not accurate.
  }
  <c'~ e'~ g'~ c''~>2
  \once \override TieColumn.positioning-done = ##t
  q
}

{
  \textMark "To fix that, enter ties with explicit direction modifiers."
  <c'_~ e'_~ g'_~ c''^~>2
  \once \override TieColumn.positioning-done = ##t
}
```

```

q
}

{
  \textMark \markup {
    Now you can use \typewriter "\\shape" for each tie as usual. }
  <c'-\shape #'((0 . 0) (0 . -10) (0 . -10) (0 . 0)) _~
  e'-\shape #'((0 . 0) (0 . -5) (0 . -5) (0 . 0)) _~
  g'-\shape #'((0 . 0) (0 . -2) (0 . -2) (0 . 0)) _~
  c''-\shape #'((0 . 0) (0 . 5) (0 . 5) (0 . 0)) ^~
  >2
  \once \override TieColumn.positioning-done = ##t
  q
}

{
  \textMark "This also works at line breaks."
  <c'-\shape #'(((0 . 0) (0 . -10) (0 . -10) (0 . 0))
    ((0 . 0) (0 . -10) (0 . -10) (0 . 0))) _~
  e'-\shape #'(((0 . 0) (0 . -5) (0 . -5) (0 . 0))
    ((0 . 0) (0 . -5) (0 . -5) (0 . 0))) _~
  g'-\shape #'(((0 . 0) (0 . -2) (0 . -2) (0 . 0))
    ((0 . 0) (0 . -2) (0 . -2) (0 . 0))) _~
  c''-\shape #'(((0 . 0) (0 . 5) (0 . 5) (0 . 0))
    ((0 . 0) (0 . 5) (0 . 5) (0 . 0))) ^~
  >2
  \break
  \once \override TieColumn.positioning-done = ##t
  q
}

{
  \textMark \markup {
    It also works with the \typewriter tieWaitForNote property. }
  \set tieWaitForNote = ##t
  c'4-\shape #'((0 . 0) (0 . -10) (0 . -10) (0 . 0)) _~
  e'-\shape #'((0 . 0) (0 . -5) (0 . -5) (0 . 0)) _~
  g'-\shape #'((0 . 0) (0 . -2) (0 . -2) (0 . 0)) _~
  c''-\shape #'((0 . 0) (0 . 5) (0 . 5) (0 . 0)) ^~
  \once \override TieColumn.positioning-done = ##t
  <c' e' g' c''>1
}

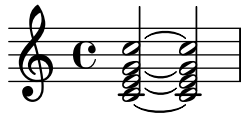
\layout {
  indent = 0
  \context {
    \Score
    \override TextMark.padding = #4
    \override TextMark.break-align-symbols = #'(left-edge)
  }
}

```

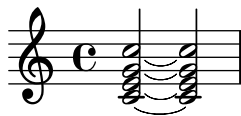


```
\paper {
  score-system-spacing.padding = 3
}
```

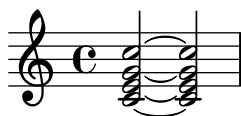
Chords can be tied note by note.



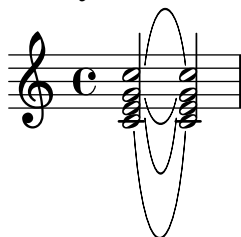
Modifying those ties with `\shape` does not succeed, because `TieColumn` positions them on its own behalf, ignoring `\shape` input more or less. You may circumvent this by setting `positioning-done` to `#t` – alas, `positioning-done` is an internal property, and setting it to `#t` means: all positioning is done, don't do anything further. The next example demonstrates a case where the positioning is not finished: all tie directions are down, and the thickness is not accurate.



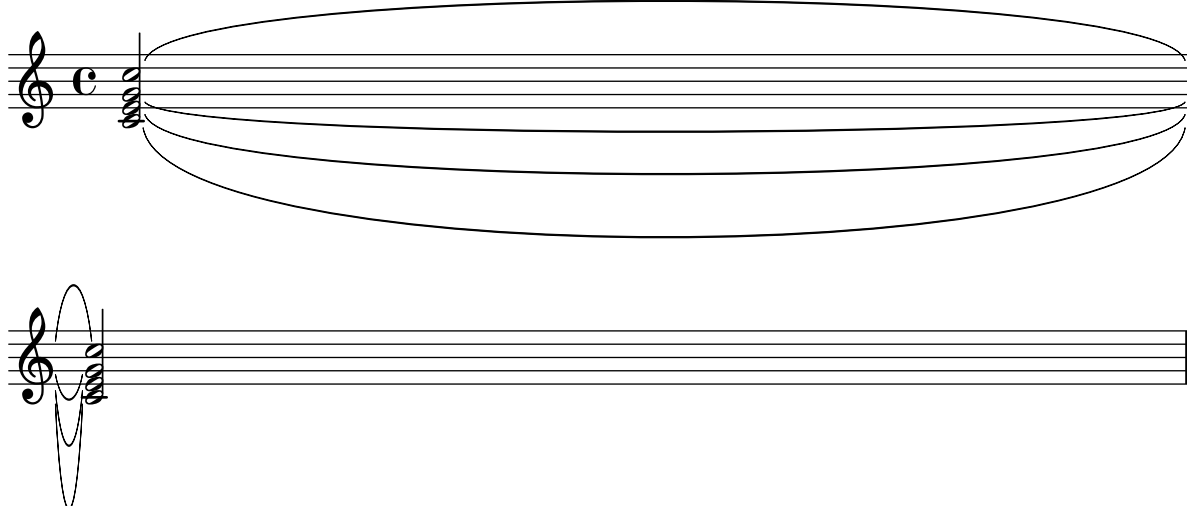
To fix that, enter ties with explicit direction modifiers.



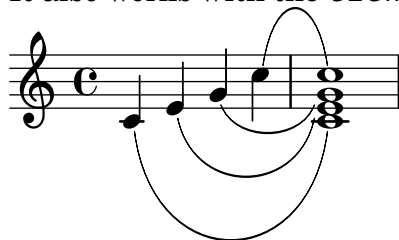
Now you can use `\shape` for each tie as usual.



This also works at line breaks.



It also works with the `tieWaitForNote` property.



Tick bar lines

,Tick' bar lines are often used in music where the bar line is used only for coordination and is not meant to imply any rhythmic stress.

```
\relative c' {
  \set Score.measureBarType = #"'"
  c4 d e f
  g4 f e d
  c4 d e f
  g4 f e d
  \bar "|."
}
```



Time signature in brackets

The time signature can be enclosed within brackets.

```
\relative c' {
  \override Staff.TimeSignature.stencil = #(lambda (grob)
    (bracketify-stencil (ly:time-signature::print grob) Y 0.1 0.2 0.1))
  \time 2/4
  a4 b8 c
}
```



Time signature in parentheses

The time signature can be enclosed within parentheses.

```
\relative c' {
  \override Staff.TimeSignature.stencil = #(lambda (grob)
    (parenthesize-stencil (ly:time-signature::print grob) 0.1 0.4 0.4 0.1))
  \time 2/4
  a4 b8 c
}
```



Eigenschaften des Schlüssels optimieren

Der Befehl `\clef "treble_8"` ist gleichbedeutend mit einem expliziten Setzen der Eigenschaften von `clefGlyph`, `clefPosition` (welche die vertikale Position des Schlüssels bestimmt), `middleCPosition` und `clefTransposition`. Ein Schlüssel wird ausgegeben, wenn eine der Eigenschaften außer `middleCPosition` sich ändert.

Eine Änderung des Schriftzeichens (Glyph), der Schlüsselposition oder der Oktavierung selber ändert noch nicht die Position der darauf folgenden Noten auf dem System: das geschieht nur, wenn auch die Position des eingestrichenen C (`middleCPosition`) angegeben wird. Die Positionsparameter sind relativ zur Mittellinie des Systems, dabei versetzen positive Zahlen die Position nach oben, jeweils eine Zahl für jede Linie plus Zwischenraum. Der `clefTransposition`-Wert ist normalerweise auf 7, -7, 15 oder -15 gesetzt, aber auch andere Werte sind gültig.

Wenn ein Schlüsselwechsel an einem Zeilenwechsel geschieht, wird das neue Symbol sowohl am Ende der alten Zeilen als auch am Anfang der neuen Zeile ausgegeben. Wenn der Warnungs-Schlüssel am Ende der alten Zeile nicht erforderlich ist, kann er unterdrückt werden, indem die `explicitClefVisibility`-Eigenschaft des Staff-Kontextes auf den Wert `end-of-line-invisible` gesetzt wird. Das Standardverhalten kann mit `\unset Staff.explicitClefVisibility` wieder hergestellt werden.

Die folgenden Beispiele zeigen die Möglichkeiten, wenn man diese Eigenschaften manuell setzt. Auf der ersten Zeile erhalten die manuellen Änderungen die ursprüngliche relative Positionierung von Schlüssel und Noten, auf der zweiten Zeile nicht.

```
{
  % The default treble clef.
  \key f \major
  c'1
  % The standard bass clef
  \set Staff.clefGlyph = "clefs.F"
  \set Staff.clefPosition = 2
  \set Staff.middleCPosition = 6
  \set Staff.middleCClefPosition = 6
  \key g \major
  c'1
  % The baritone clef.
  \set Staff.clefGlyph = "clefs.C"
  \set Staff.clefPosition = 4
  \set Staff.middleCPosition = 4
  \set Staff.middleCClefPosition = 4
  \key f \major
  c'1
  % The standard choral tenor clef.
  \set Staff.clefGlyph = "clefs.G"
  \set Staff.clefPosition = -2
  \set Staff.clefTransposition = -7
  \set Staff.middleCPosition = 1
  \set Staff.middleCClefPosition = 1
  \key f \major
  c'1
  % A non-standard clef.
  \set Staff.clefPosition = 0
  \set Staff.clefTransposition = 0
  \set Staff.middleCPosition = -4
```

```

\set Staff.middleCClefPosition = -4
\key g \major
c'1 \break

% The following clef changes do not preserve
% the normal relationship between notes, key signatures
% and clefs.
\set Staff.clefGlyph = "clefs.F"
\set Staff.clefPosition = 2
c'1
\set Staff.clefGlyph = "clefs.G"
c'1
\set Staff.clefGlyph = "clefs.C"
c'1
\set Staff.clefTransposition = 7
c'1
\set Staff.clefTransposition = 0
\set Staff.clefPosition = 0
c'1

% Return to the normal clef.
\set Staff.middleCPosition = 0
c'1
}

```



Two \partCombine pairs on one staff

The `\partCombine` function takes two music expressions, each containing a part, and distributes them among four Voice contexts named „one“, „two“, „solo“, and „shared“, depending on when and how the parts are merged into a common voice.

Variants of `\partCombine` are `\partCombineUp` and `\partCombineDown` to produce up-stem and down-stem merging of two voices, respectively. Combining them to squeeze four parts into a single staff, however, need some special setup, which this snippet defines accordingly.

```

customPartCombineUp =
#(define-music-function (part1 part2) (ly:music? ly:music?)
  "Make an up-stem `VoiceBox` context that combines PART1 and PART2.

```

The context is called 'Up'; internally, the function calls

```

`\partCombineUp`.
#{
  \new VoiceBox = "Up" <<
    \context Voice = "one" { \voiceOne }
    \context Voice = "two" { \voiceThree }
    \context Voice = "shared" { \voiceOne }

```

```

    \context Voice = "solo" { \voiceOne }
    \context NullVoice = "null" {}
    \partCombine #part1 #part2
  >>
#})

```

```

customPartCombineDown =
#(define-music-function (part3 part4) (ly:music? ly:music?)
  "Make a down-stem `VoiceBox` context that combines PART3 and PART4.

```

The context is called 'Down'; internally, the function calls
`\\partCombineDown`.`

```

#{
  \new VoiceBox = "Down" <<
    \set VoiceBox.soloText = #"Solo III"
    \set VoiceBox.soloIIIText = #"Solo IV"
    \context Voice = "one" { \voiceFour }
    \context Voice = "two" { \voiceTwo }
    \context Voice = "shared" { \voiceFour }
    \context Voice = "solo" { \voiceFour }
    \context NullVoice = "null" {}
    \partCombine #part3 #part4
  >>
#})

```

```

soprano = { d'4 | cis' b e' d'8 cis' | cis'2 b }
alto = { fis4 | e8 fis gis ais b4 b | b ais fis2 }
tenor = { a8 b | cis' dis' e'4 b8 cis' d'4 | gis cis' dis'2 }
bass = { fis8 gis | a4 gis g fis | eis fis b,2 }

```

```

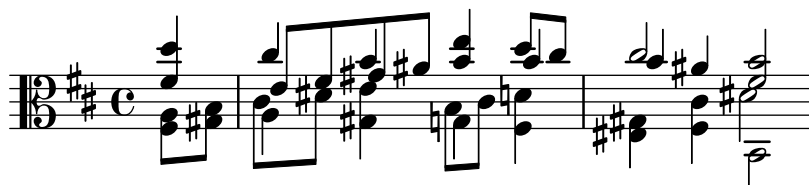
\new Staff <<
  \key b\minor
  \clef alto
  \partial 4
  \transpose b b' \customPartCombineUp \soprano \alto
  \customPartCombineDown \tenor \bass
>>

```

```

\layout {
  \context {
    \Staff
    \accepts "VoiceBox"
  }
  \context {
    \name "VoiceBox"
    \type "Engraver_group"
    \defaultchild "Voice"
    \accepts "Voice"
    \accepts "NullVoice"
  }
}

```



Eine eckige Klammer zu Beginn von Systemgruppen benutzen

Die Klammer zu Beginn von Systemgruppen kann auch in eine eckige Klammer (`SystemStartSquare`) umgewandelt werden, wenn man sie explizit im `StaffGroup`- oder `ChoirStaffGroup`-Kontext setzt.

```
\score {
  \new StaffGroup { <<
    \set StaffGroup.systemStartDelimiter = #'SystemStartSquare
    \new Staff { c'4 d' e' f' }
    \new Staff { c'4 d' e' f' }
  >> }
}
```



Using `\autoChange` with more than one voice

Here is a demonstration of how to use `\autoChange` with more than one voice.

```
\score {
  \new PianoStaff
  <<
    \new Staff = "up" {
      <<
        \set Timing.beamExceptions = #'()
        \set Timing.beatStructure = #'(4)
        \new Voice {
          \voiceOne
          \autoChange
          \relative c' {
            g8 a b c d e f g
            g,,8 a b c d e f g
          }
        }
      >>
    }

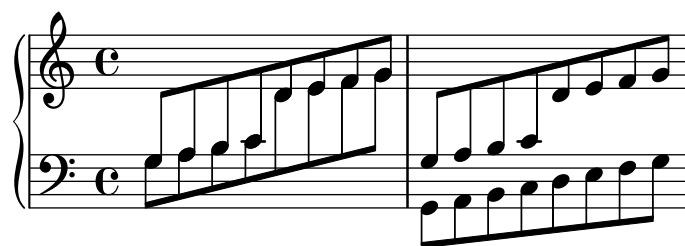
    \new Voice {
      \voiceTwo
      \autoChange
      \relative c' {
        g8 a b c d e f g
        g,,8 a b c d e f g
      }
    }
  >>
}
```

```

    }
  >>
}

\new Staff = "down" {
  \clef bass
}
>>
}

```



Using mark lines in a Frenched score

Using MarkLine contexts (such as in „Placing rehearsal marks other than above the top staff“) in a Frenched score can be problematic if all the staves between two MarkLines are removed in one system. The `Keep_alive_together_engraver` can be used within each StaffGroup to keep the MarkLine alive only as long as the other staves in the group stay alive.

```

bars = {
  \tempo "Allegro" 4=120
  s1*2
  \repeat unfold 5 { \mark \default s1*2 }
  \bar "||"
  \tempo "Adagio" 4=40
  s1*2
  \repeat unfold 8 { \mark \default s1*2 }
  \bar "|."
}

winds = \repeat unfold 120 { c''4 }
trumpet = { \repeat unfold 8 g'2 R1*16 \repeat unfold 4 g'2 R1*8 }
trombone = { \repeat unfold 4 c'1 R1*8 d'1 R1*17 }
strings = \repeat unfold 240 { c''8 }

#(set-global-staff-size 16)
\paper {
  systems-per-page = 5
  ragged-last-bottom = ##f
  tagline = ##f
}

\layout {
  indent = 16\mm
  short-indent = 5\mm
  \context {
    \name MarkLine
    \type Engraver_group

```

```

\consists Output_property_engraver
\consists Axis_group_engraver
\consists Mark_engraver
\consists Metronome_mark_engraver
\consists Staff_collecting_engraver
\override VerticalAxisGroup.remove-empty = ##t
\override VerticalAxisGroup.remove-layer = #'any
\override VerticalAxisGroup.staff-affinity = #DOWN
\override VerticalAxisGroup.nonstaff-relatedstaff-spacing.padding = 1
keepAliveInterfaces = #'()
}
\context {
  \Staff
  \override VerticalAxisGroup.remove-empty = ##t
  \override VerticalAxisGroup.remove-layer = ##f
}
\context {
  \StaffGroup
  \accepts MarkLine
  \consists Keep_alive_together_engraver
}
\context {
  \Score
  \remove Mark_engraver
  \remove Metronome_mark_engraver
  \remove Staff_collecting_engraver
  \override BarNumber.Y-offset = #3
}
}

\score {
  <<
    \new StaffGroup = "winds" \with {
      instrumentName = "Winds"
      shortInstrumentName = "W."
    } <<
    \new MarkLine \bars
    \new Staff \winds
  >>
  \new StaffGroup = "brass" <<
    \new MarkLine \bars
    \new Staff = "trumpet" \with {
      instrumentName = "Trumpet"
      shortInstrumentName = "Tp."
    } \trumpet
    \new Staff = "trombone" \with {
      instrumentName = "Trombone"
      shortInstrumentName = "Tb."
    } \trombone
  >>
  \new StaffGroup = "strings" \with {
    instrumentName = "Strings"

```



```

    shortInstrumentName = "Str."
  } <<
    \new MarkLine \bars
    \new Staff = "strings" { \strings }
  >>
>>
}

```

The musical score is written for four staves: Winds, Trumpet, Trombone, and Strings. The score is divided into sections A, B, C, D, E, F, G, H, J, K, and L. The tempo changes from Allegro (120) to Adagio (40) at measure 11.

Measures 6-10: Winds, Trumpet, and Trombone play a melody in 4/4 time, marked Allegro (120). The Strings play a rhythmic pattern of eighth notes. Section A is measures 6-8, and Section B is measures 9-10.

Measures 11-15: Winds, Trumpet, and Trombone play a melody in 4/4 time, marked Adagio (40). The Strings play a rhythmic pattern of eighth notes. Section C is measures 11-13, and Section D is measures 14-15.

Measures 16-20: Winds, Trumpet, and Trombone play a melody in 4/4 time, marked Adagio (40). The Strings play a rhythmic pattern of eighth notes. Section E is measures 16-18, and Section F is measures 19-20.

Measures 21-25: Winds, Trumpet, and Trombone play a melody in 4/4 time, marked Adagio (40). The Strings play a rhythmic pattern of eighth notes. Section G is measures 21-23, and Section H is measures 24-25.

Measures 26-30: Winds, Trumpet, and Trombone play a melody in 4/4 time, marked Adagio (40). The Strings play a rhythmic pattern of eighth notes. Section J is measures 26-28, Section K is measures 29-30, and Section L is measures 31-33.



Vertically aligned StaffGroups without connecting SystemStartBar

This snippet shows how to achieve vertically aligned StaffGroups with a SystemStartBar for each StaffGroup, but without connecting them.

Note that this only works properly for music that can be printed as a single system.

```

#(set-global-staff-size 15)

```

```

\paper {
  ragged-right = ##f
  print-all-headers = ##t
  tagline = ##f
}

```

```

\layout {
  indent = 0

```

```

  \context {
    \StaffGroup
    \consists Text_mark_engraver
    \consists Staff_collecting_engraver
    systemStartDelimiterHierarchy =
      #'(SystemStartBrace (SystemStartBracket a b))
  }

```

```

  \context {
    \Score
    \remove Text_mark_engraver
    \remove Staff_collecting_engraver
    \override SystemStartBrace.style = #'bar-line
    \omit SystemStartBar
    \override SystemStartBrace.padding = #-0.1
    \override SystemStartBrace.thickness = #1.6
    \override StaffGrouper.staffgroup-staff-spacing.basic-distance = #15
  }
}

```

```

%%% EXAMPLE

```

```

txt =
\lyricmode {
  Wer4 nur den lie -- ben Gott läßt wal2 -- ten4
  und4 hof -- fet auf ihn al -- le Zeit2.
}

```

```

% First StaffGroup "exercise"

eI = \relative c' {
  \textMark \markup {
    \bold Teacher:
    This is a simple setting of the choral. Please improve it. }
  \key a \minor
  \time 4/4
  \voiceOne

  \partial 4 e4
  a b c b
  a b gis2
  e4\fermata g! g f
  e a a gis
  a2.\fermata
  \bar ":|."
}

eII = \relative c' {
  \key a \minor
  \time 4/4
  \voiceTwo
  \partial 4 c4
  e e e gis
  a f e2
  b4 b d d
  c c d d
  c2.
  \bar ":|."
}

eIII = \relative c' {
  \key a \minor
  \time 4/4
  \clef bass
  \voiceOne

  \partial 4 a4
  c b a b
  c d b2
  gis4 g g b
  c a f e
  e2.
}

eIV = \relative c' {
  \key a \minor
  \time 4/4
  \clef bass
  \voiceTwo

```

```

\partial 4 a,4
a' gis a e
a, d e2
e,4\fermata e' b g
c f d e
a,2.\fermata
\bar " : | . "
}

exercise = \new StaffGroup = "exercise" <<
  \new Staff <<
    \new Voice \eI
    \new Voice \eII
  >>

  \new Lyrics \txt

  \new Staff <<
    \new Voice \eIII
    \new Voice \eIV
  >>
>>

% Second StaffGroup "simple Bach"

sbI = \relative c' {
  \textMark \markup { \bold" Pupil:" Here's my version! }
  \key a \minor
  \time 4/4
  \voiceOne

  \partial 4 e4
  a b c b
  a b gis2
  e4\fermata g! g f
  e a a gis
  a2.\fermata
  \bar " : | . "
}

sbII = \relative c' {
  \key a \minor
  \time 4/4
  \voiceTwo
  \partial 4 c8 d
  e4 e e8 f g4
  f f e2
  b4 b8 c d4 d
  e8 d c4 b8 c d4
  c2.
}

```

```

\bar " : | ."
}

sbIII = \relative c' {
  \key a \minor
  \time 4/4
  \clef bass
  \voiceOne

  \partial 4 a8 b
  c4 b a b8 c
  d4 d8 c b2
  gis4 g g8 a b4
  b a8 g f4 e
  e2.
}

sbIV = \relative c' {
  \key a \minor
  \time 4/4
  \clef bass
  \voiceTwo

  \partial 4 a,4
  a' gis a e
  f8 e d4 e2
  e,4\fermata e' b a8 g
  c4 f8 e d4 e
  a,2.\fermata
  \bar " : | ."
}

simpleBach = \new StaffGroup = "simple Bach" <<
  \new Staff <<
    \new Voice \sbI
    \new Voice \sbII
  >>

  \new Lyrics \txt

  \new Staff <<
    \new Voice \sbIII
    \new Voice \sbIV
  >>
>>

% Third StaffGroup "chromatic Bach"

cbI = \relative c' {
  \textMark \markup {
    \bold "Teacher:"
  }
}

```

```

\column {
  "Well, you simply copied and transposed a version of J.S.Bach."
  "Do you know this one?"
}
}
\key a \minor
\time 4/4
\voiceOne

\partial 4 e4
a b c b
a b gis4. fis8
e4\fermata g! g f
e a a8 b gis4
a2.\fermata
\bar " : | ."
}

cbII = \relative c' {
  \key a \minor
  \time 4/4
  \voiceTwo

  \partial 4 c8 d
  e4 e e8 fis gis4
  a8 g! f!4 e2
  b4 e e d
  d8[ cis] d dis e fis e4
  e2.
  \bar " : | ."
}

cbIII = \relative c' {
  \key a \minor
  \time 4/4
  \clef bass
  \voiceOne

  \partial 4 a8 b
  c[ b] a gis8 a4 d,
  e8[ e'] d c b4. a8
  gis4 b c d8 c
  b[ a] a b c b b c16 d
  c2.
}

cbIV = \relative c' {
  \key a \minor
  \time 4/4
  \clef bass
  \voiceTwo

```

```

\partial 4 a4
c, e a, b
c d e2
e4\fermata e a b8 c
gis[ g] fis f e dis e4
a,2.\fermata
\bar " : | . "
}

chromaticBach = \new StaffGroup = "chromatic Bach" <<
  \new Staff <<
    \new Voice \cbI
    \new Voice \cbII
  >>

  \new Lyrics \txt

  \new Staff <<
    \new Voice \cbIII
    \new Voice \cbIV
  >>
>>

% Score

\score {
  <<
    \exercise
    \simpleBach
    \chromaticBach
  >>

  \header {
    title = \markup \column {
      \combine \null \vspace #1
      "Exercise: Improve the given choral"
      " "
    }
  }
}

\layout {
  \context {
    \Lyrics
    \override LyricText.X-offset = #-1
  }
}
}

```

Exercise: Improve the given choral

Teacher: This is a simple setting of the choral. Please improve it.

Wer nur den lie - ben Gott läßt wal - ten und hof-fet auf ihn al - le Zeit

Pupil: Here's my version!

Wer nur den lie - ben Gott läßt wal - ten und hof-fet auf ihn al - le Zeit

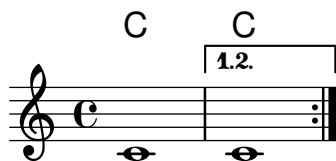
Teacher: Well, you simply copied and transposed a version of J.S.Bach.
Do you know this one?

Wer nur den lie - ben Gott läßt wal - ten und hof-fet auf ihn al - le Zeit

Wiederholungs-(Volta-)Klammern unterhalb der Akkordsymbole

Indem man den `Volta_engraver` zu dem entsprechenden Notensystem hinzufügt, können Wiederholungsklammern unterhalb der Akkorde gesetzt werden.

```
\score {
  <<
    \chords { c1 c1 }
    \new Staff \with { \consists "Volta_engraver" }
    {
      \repeat volta 2 { c'1 \alternative { c' } }
    }
  >>
  \layout {
    \context {
      \Score
      \remove "Volta_engraver"
    }
  }
}
```

Volta brackets in multiple staves

By adding the `Volta_engraver` to the relevant staff, volte can be put over staves other than the topmost one in a score.

`\repeat` and related commands should be present in all staves.

```
voltaMusic = \relative c'' {
  \repeat volta 2 {
    c1
    \alternative {
      \volta 1 { d1 }
      \volta 2 { e1 }
    }
  }
}
```

```
<<
  \new StaffGroup <<
    \new Staff \voltaMusic
    \new Staff \voltaMusic
  >>
  \new StaffGroup <<
    \new Staff \with { \consists "Volta_engraver" }
      \voltaMusic
    \new Staff \voltaMusic
  >>
>>
```



7 Editorial annotations

See also Abschnitt “Editorial annotations” in *Notationsreferenz*.

Adding fingerings to a score

Fingering instructions can be entered using a simple syntax.

```
\relative c' ' {
  c4-1 d-2 f-4 e-3
}
```



Adding links to objects

To add a link to a grob stencil you can use `add-link` as defined here. It works both with `\override` and `\tweak`.

Drawback: `point-and-click` is disturbed for the linked grobs.

Limitation: Works for PDF only.

The linked objects are colored with a separate command.

```
#(define (add-link url-strg)
  (lambda (grob)
    (let* ((stil (ly:grob-property grob 'stencil)))
      (if (ly:stencil? stil)
        (let* ((x-ext (ly:stencil-extent stil X))
              (y-ext (ly:stencil-extent stil Y))
              (url-expr `(url-link ,url-strg ,x-ext ,y-ext))
              (new-stil
                (ly:stencil-add
                 (ly:make-stencil url-expr x-ext y-ext)
                 stil)))
          (ly:grob-set-property! grob 'stencil new-stil))))))
```

```
%%% test
```

```
%% For easier maintenance of this snippet the URL is formatted to use the
%% actually used LilyPond version.
%% Of course a literal URL would work as well.
```

```
#(define major.minor-version
  (string-join (take (string-split (lilypond-version) #\.) 2) "."))

urlI =
#(format #f
  "http://lilypond.org/doc/v~a/Documentation/notation/writing-pitches"
  major.minor-version)

urlII =
#(format #f
  "http://lilypond.org/doc/v~a/Documentation/notation/rhythms"
```

```

major.minor-version)

urlIII =
#(format #f
  "http://lilypond.org/doc/v~a/Documentation/notation/note-heads"
  major.minor-version)

urlIV =
#(format #f
  "http://lilypond.org/doc/v~a/Documentation/notation/beams"
  major.minor-version)

urlV =
#(format #f
  "http://lilypond.org/doc/v~a/Documentation/notation/note-head-styles"
  major.minor-version)

urlVI =
#(format #f
  "http://lilypond.org/doc/v~a/Documentation/notation/writing-pitches"
  major.minor-version)

\relative c' {
  \key cis \minor

  \once \override Staff.Clef.color = #green
  \once \override Staff.Clef.after-line-breaking =
    #(add-link urlI)

  \once \override Staff.TimeSignature.color = #green
  \once \override Staff.TimeSignature.after-line-breaking =
    #(add-link urlII)

  \once \override NoteHead.color = #green
  \once \override NoteHead.after-line-breaking =
    #(add-link urlIII)

  cis'1
  \once \override Beam.color = #green
  \once \override Beam.after-line-breaking =
    #(add-link urlIV)
  cis8 dis e fis gis2
  <gis,
    \tweak Accidental.color #green
    \tweak Accidental.after-line-breaking #(add-link urlVI)
    \tweak color #green
    \tweak after-line-breaking #(add-link urlV)
    \tweak style #'harmonic
  bis
  dis
  fis
  >1

```

```
<cis, cis' e>
}
```



Adding markups in a tablature

By default, markups are not displayed in a tablature.

To make them appear, revert the stencil property of the TextScript grob in the TabStaff context.

```
high = { r4 r8 <g c'> q r8 r4 }
low = { c4 r4 c8 r8 g,8 b, }
pulse = { s8^"1" s8^"&" s8^"2" s8^"&" s8^"3" s8^"&" s8^"4" s8^"&" }
```

```
\score {
  \new TabStaff {
    \repeat unfold 2 << \high \\\ \low \\\ \pulse >>
  }
  \layout {
    \context {
      \TabStaff
      \clef moderntab
      \revert TextScript.stencil
      \override TextScript.font-series = #'bold
      \override TextScript.font-size = #-2
      \override TextScript.color = #red
    }
    \context {
      \Score
      proportionalNotationDuration = #1/8
    }
  }
}
```

	1	&	2	&	3	&	4	&	1	&	2	&	3	&	4	&
T																
A																
B	3				3				2				3			2

Fingersatz auch innerhalb des Systems setzen

Normalerweise werden vertikal orientierte Fingersatzzahlen außerhalb des Systems gesetzt. Das kann aber verändert werden.

```
\relative c' {
  <c-1 e-2 g-3 b-5>2
  \override Fingering.staff-padding = #'()
  <c-1 e-2 g-3 b-5>4 g'-0
  a8[-1 b]-2 g-0 r
  \override Fingering.add-stem-support = ##f
  a[-1 b]-2 g-0 r
  \override Fingering.add-stem-support = ##t
```

```

a[-1 b]-2 g-0 r
\override Fingering.add-stem-support = #only-if-beamed
a[-1 b]-2 g-0 r
}

```



Alternative Taktnummerierung

Zwei alternative Methoden können eingestellt werden, die die Taktnummerierung beeinflussen, insbesondere bei Wiederholungen.

```

music = \relative c' {
  \repeat volta 3 {
    c4 d e f |
    \alternative {
      \volta 1 { c4 d e f | c2 d \break }
      \volta 2 { f4 g a b | f4 g a b | f2 a | \break }
      \volta 3 { c4 d e f | c2 d } } }
  c1 \bar " | ."
}

{
  \textMark \markup \large "default"
  \music
}

{
  \textMark \markup \large \typewriter "numbers"
  \set Score.alternativeNumberingStyle = #'numbers
  \music
}

{
  \textMark \markup \large \typewriter "numbers-with-letters"
  \set Score.alternativeNumberingStyle = #'numbers-with-letters
  \music
}

\layout {
  \context {
    \Score
    \override TextMark.Y-offset = #5
  }
}

```



The image shows a series of musical staves illustrating different ways to place analysis brackets above the staff. The staves are labeled 4, 7, numbers, 2, 2, numbers-with-letters, 2b, and 2c. Each staff shows a sequence of notes with a bracket above it, indicating a specific annotation.

Analysis brackets above the staff

Simple horizontal analysis brackets are added below the staff by default. The following example shows a way to place them above the staff instead.

```
\layout {
  \context {
    \Voice
    \consists "Horizontal_bracket_engraver"
  }
}

\relative c'' {
  \once \override HorizontalBracket.direction = #UP
  c2\startGroup
  d2\stopGroup
}
```



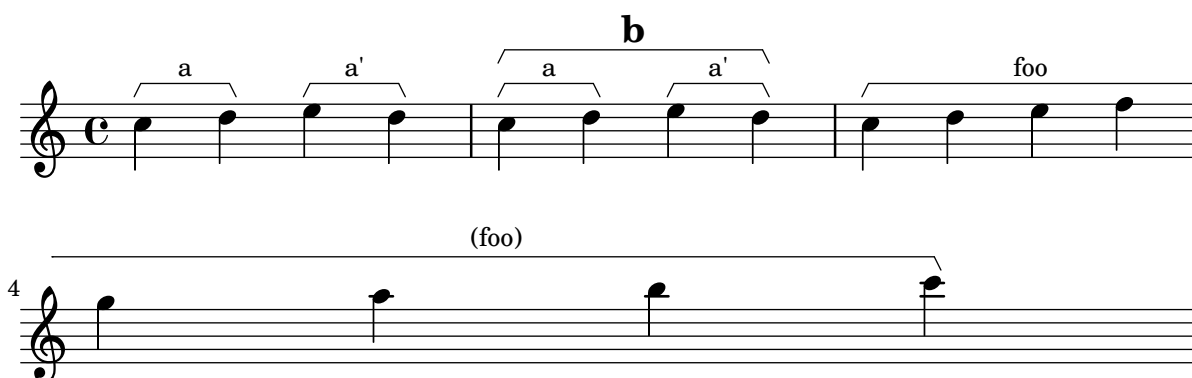
Analysis brackets with labels

Text markup may be added to analysis brackets using the `text` property of the `HorizontalBracketText` grob. Adding different texts to brackets beginning at the same time requires the `\tweak` command.

Bracket text gets parenthesized after a line break. The vertical order of nested brackets can be controlled with the `outside-staff-priority` property.

```
\layout {
  \context {
    \Voice
    \consists "Horizontal_bracket_engraver"
    \override HorizontalBracket.direction = #UP
  }
}

{
  \once\override HorizontalBracketText.text = "a"
  c''\startGroup d''\stopGroup
  \once\override HorizontalBracketText.text = "a'"
  e''\startGroup d''\stopGroup |
  c''-\tweak outside-staff-priority #801
    \tweak HorizontalBracketText.text
      \markup \bold \huge "b" \startGroup
    -\tweak HorizontalBracketText.text "a" \startGroup
    d''\stopGroup
    e''-\tweak HorizontalBracketText.text "a'" \startGroup
    d''\stopGroup\stopGroup |
  c''-\tweak HorizontalBracketText.text foo \startGroup
    d'' e'' f'' | \break
  g'' a'' b'' c'''\stopGroup
}
```



Notenkopfstile basierend auf der Tonleiterstufe erstellen

Die `shapeNoteStyles`-(`NotenFormenStile`)-Eigenschaft kann benutzt werden, um verschiedene Notenstile für jeden Schritt der Tonleiter zu definieren (vorgegeben von der Tonart oder der „tonic“ (Tonika)-Eigenschaft. Diese Eigenschaft braucht eine Anzahl von Symbolen, welche beliebig sein können (geometrische Ausdrücke wie `triangle` (Dreieck), `cross` (Kreuz) und `xcircle` (X-Kreis) sind erlaubt) oder basierend auf einer alten amerikanischen Notensatztradition (einige lateinische Notenbezeichnungen sind auch erlaubt).

Um alte amerikanische Liederbücher zu imitieren, gibt es einige vordefinierte Notenstile wie etwa `\aikenHeads` (im Stil von Aiken) oder `\sacredHarpHeads` (im Stil der Sacred Harp-Tradition).

Dieses Beispiel zeigt, wie man unterschiedlich geformte Noten erhält und eine Melodie transponieren kann, ohne dass das Verhältnis zwischen den harmonischen Funktionen und dem Notenstil verloren geht.

```
fragment = {
  \key c \major
  c2 d
  e2 f
  g2 a
  b2 c
}

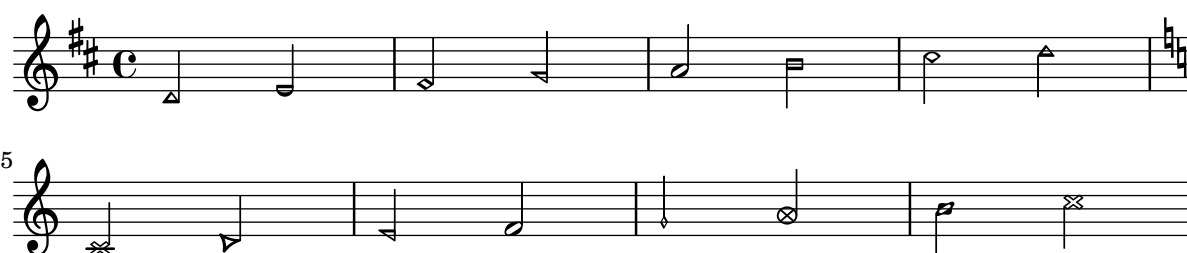
\new Staff {
  \transpose c d
  \relative c' {
    \set shapeNoteStyles = ##(do re mi fa
                          #f la ti)

    \fragment
  }

  \break

  \relative c' {
    \set shapeNoteStyles = ##(cross triangle fa #f
                          mensural xcircle diamond)

    \fragment
  }
}
```

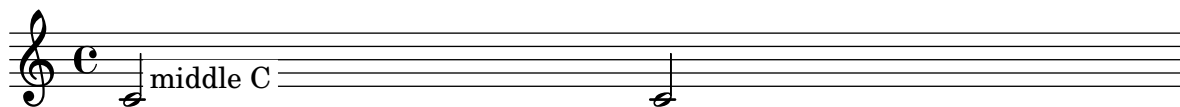


Blanking staff lines using the `\whiteout` command

The `\whiteout` command underlays a markup with a white box. Since staff lines are in a lower layer than most other grobs, this white box will not overlap any other grob.

```
\layout {
  ragged-right = ##f
}

\relative c' {
  \override TextScript.extra-offset = #'(2 . 4)
  c2-\markup { \whiteout \pad-markup #0.5 "middle C" } c
}
```

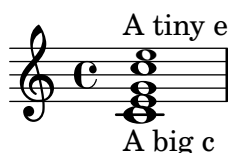



Changing a single note's size in a chord

Individual note heads in a chord can be modified with the `\tweak` command inside a chord, by altering the `font-size` property.

Inside the chord (within the brackets `< >`), before the note to be altered, place the `\tweak` command, followed by `font-size` and define the proper size like `#-2` (a tiny note head).

```
\relative c' {
  <\tweak font-size #+2 c e g c
    \tweak font-size #-2 e>1
  ~\markup { A tiny e }_~\markup { A big c }
}
```



Changing the appearance of a slur from solid to dotted or dashed

The appearance of slurs may be changed from solid to dotted or dashed.

```
\relative c' {
  c4( d e c)
  \slurDotted
  c4( d e c)
  \slurSolid
  c4( d e c)
  \slurDashed
  c4( d e c)
  \slurSolid
  c4( d e c)
}
```



Coloring notes depending on their pitch

It is possible to color note heads depending on their pitch and/or their names: the function used in this example even makes it possible to distinguish enharmonics.

% Association list of pitches to colors.

```
#(define color-mapping
  (list
    (cons (ly:make-pitch 0 0 NATURAL) (x11-color 'red))
    (cons (ly:make-pitch 0 0 SHARP) (x11-color 'green))
    (cons (ly:make-pitch 0 1 FLAT) (x11-color 'green))
```

```

(cons (ly:make-pitch 0 2 NATURAL) (x11-color 'red))
(cons (ly:make-pitch 0 2 SHARP) (x11-color 'green))
(cons (ly:make-pitch 0 3 FLAT) (x11-color 'red))
(cons (ly:make-pitch 0 3 NATURAL) (x11-color 'green))
(cons (ly:make-pitch 0 4 SHARP) (x11-color 'red))
(cons (ly:make-pitch 0 5 NATURAL) (x11-color 'green))
(cons (ly:make-pitch 0 5 FLAT) (x11-color 'red))
(cons (ly:make-pitch 0 6 SHARP) (x11-color 'red))
(cons (ly:make-pitch 0 1 NATURAL) (x11-color 'blue))
(cons (ly:make-pitch 0 3 SHARP) (x11-color 'blue))
(cons (ly:make-pitch 0 4 FLAT) (x11-color 'blue))
(cons (ly:make-pitch 0 5 SHARP) (x11-color 'blue))
(cons (ly:make-pitch 0 6 FLAT) (x11-color 'blue)))

% Compare pitch and alteration (not octave).
#(define (pitch-equals? p1 p2)
  (and
    (= (ly:pitch-alteration p1) (ly:pitch-alteration p2))
    (= (ly:pitch-notename p1) (ly:pitch-notename p2))))

#(define (pitch-to-color pitch)
  (let ((color (assoc pitch color-mapping pitch-equals?)))
    (if color
      (cdr color))))

#(define (color-notehead grob)
  (pitch-to-color
    (ly:event-property (event-cause grob) 'pitch)))

\score {
  \new Staff \relative c' {
    \override NoteHead.color = #color-notehead
    c8 b d dis ees f g aes
  }
}

```



Position von Fingersatz in Akkorden kontrollieren

Die Position von Fingersatzzahlen kann exakt kontrolliert werden.

```

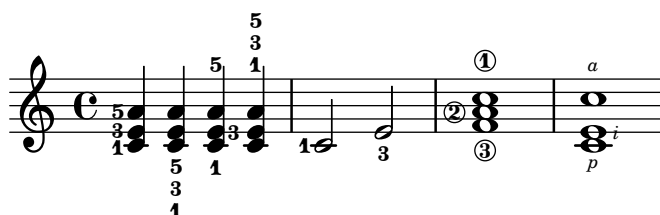
\relative c' {
  \set fingeringOrientations = #'(left)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down right up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(up)
  <c-1 e-3 a-5>4
}

```

```

\set fingeringOrientations = #'(left)
<c-1>2
\set fingeringOrientations = #'(down)
<e-3>2
\set stringNumberOrientations = #'(up left down)
<f\3 a\2 c\1>1
\set strokeFingerOrientations = #'(down right up)
<c\rightHandFinger 1 e\rightHandFinger 2 c'\rightHandFinger 4 >
}

```



Einen Doppelschlag mit Vorhalt erstellen

Einen Doppelschlag mit Vorhalt zu erstellen, wobei die untere Note das Vorzeichen benutzt, erfordert einige Einstellungsänderungen. Die `outside-staff-priority`-Eigenschaft muss auf falsch (`#f`) gesetzt werden, weil sie sonst über die Eigenschaft `avoid-slur` property dominieren würde. Der Wert von `halign` wird benutzt, um den Doppelschlag horizontal zu positionieren.

```

\relative c' {
  \after 2*2/3 \turn c2( d4) r |
  \after 4 \turn c4.( d8)
  \after 4
  {
    \once \set suggestAccidentals = ##t
    \once \override AccidentalSuggestion.outside-staff-priority = ##f
    \once \override AccidentalSuggestion.avoid-slur = #'inside
    \once \override AccidentalSuggestion.font-size = -3
    \once \override AccidentalSuggestion.script-priority = -1
    \once \hideNotes
    cis8\turn \noBeam
  }
  d4.( e8)
}

```



Creating blank staves

To create blank staves, generate empty measures then remove the `Bar_number_engraver` from the `Score` context, and the `Time_signature_engraver`, `Clef_engraver` and `Bar_engraver` from the `Staff` context.

```

#(set-global-staff-size 10) % for the documentation
% #(set-global-staff-size 20) % for letter and A4

```

```

\book {

```

```

\score {
  { \repeat unfold 12 { s1 \break } }

  \layout {
    indent = 0
    \context {
      \Staff
      \remove "Time_signature_engraver"
      \remove "Clef_engraver"
      \remove "Bar_engraver"
    }
    \context {
      \Score
      \remove "Bar_number_engraver"
    }
  }
}

% for the documentation
\paper {
  #(set-paper-size "a6")
  ragged-last-bottom = ##f
  line-width = 90\mm
  left-margin = 7.5\mm
  bottom-margin = 5\mm
  top-margin = 5\mm
  tagline = ##f
}

% uncomment these lines for "letter" size
%{
\paper {
  #(set-paper-size "letter")
  ragged-last-bottom = ##f
  line-width = 7.5\in
  left-margin = 0.5\in
  bottom-margin = 0.25\in
  top-margin = 0.25\in
  tagline = ##f
}
%}

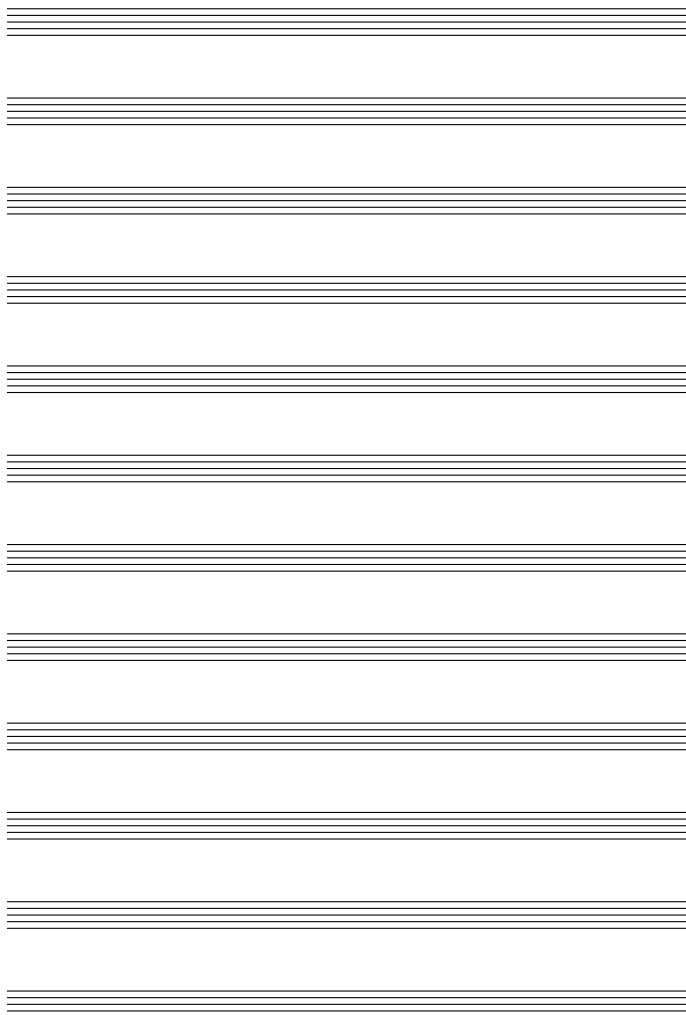
% uncomment these lines for "A4" size
%{
\paper {
  #(set-paper-size "a4")
  ragged-last-bottom = ##f
  line-width = 180\mm
  left-margin = 15\mm
  bottom-margin = 10\mm
  top-margin = 10\mm
  tagline = ##f
}
%}

```

```

}
%}
}

```



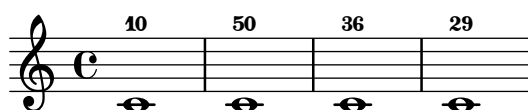
Creating double-digit fingerings

Creating fingerings larger than 5 is possible.

```

\relative c' {
  c1-10
  c1-50
  c1-36
  c1-29
}

```



Standardrichtung für Hälse auf der Mittellinie

Die Richtung von Hälse auf der mittleren Linie kann mit der Stem-Eigenschaft `neutral-direction` gesetzt werden.

```

\relative c' {

```

```

a4 b c b
\override Stem.neutral-direction = #up
a4 b c b
\override Stem.neutral-direction = #down
a4 b c b
}

```



Different font size settings for instrumentName and shortInstrumentName

Choose different font sizes for instrumentName and shortInstrumentName as a context override.

```

InstrumentNameFontSize =
#(define-music-function (font-size-pair) (pair?)
  "Set the font size of `InstrumentName` grobs.

```

The first value of FONT-SIZE-PAIR sets the font size of the initial ``instrumentName`` property, the second value sets the font size of ``shortInstrumentName``."

```

;; This code could be changed or extended to set different values
;; for each occurrence of `shortInstrumentName'.
#{
  \override InstrumentName.after-line-breaking =
    #(lambda (grob)
      (let* ((orig (ly:grob-original grob))
             (siblings (if (ly:grob? orig)
                           (ly:spanner-broken-into orig)
                           '())))
        (when (pair? siblings)
          (ly:grob-set-property! (car siblings)
                                'font-size (car font-size-pair))
          (for-each
            (lambda (g)
              (ly:grob-set-property! g
                                    'font-size (cdr font-size-pair)))
            (cdr siblings))))))
  #})

\layout {
  indent = 3\cm
  short-indent = 0.8\cm

  \context {
    \Staff
    \InstrumentNameFontSize #'(6 . -3)
  }
}

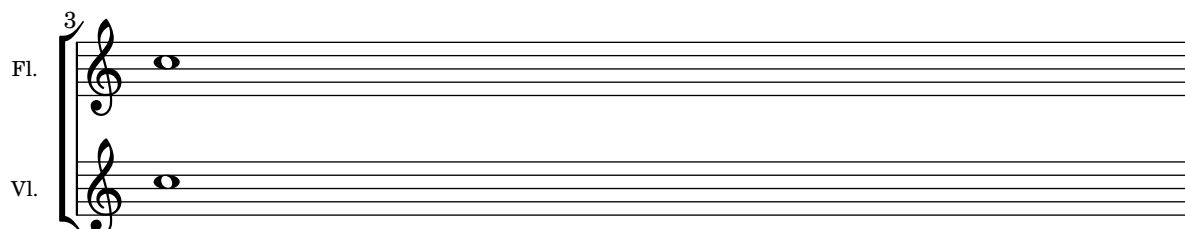
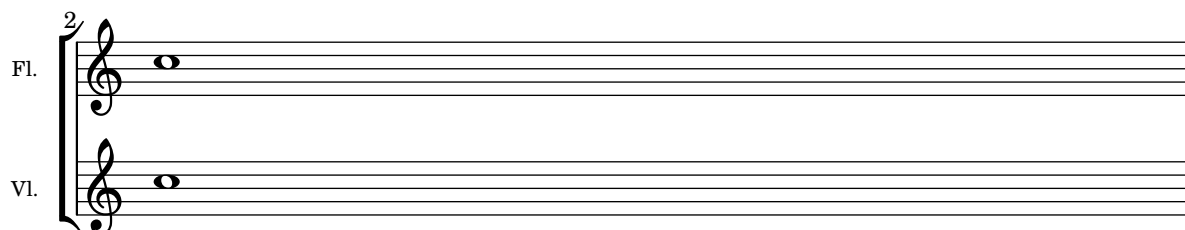
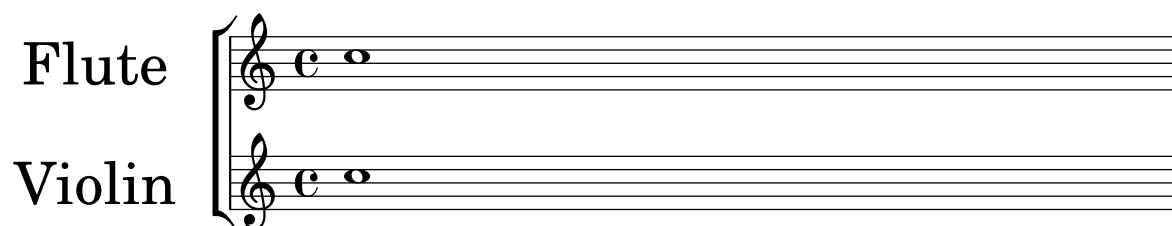
\new StaffGroup <<

```

```

\new Staff \with {
  instrumentName = "Flute"
  shortInstrumentName = "Fl." } {
  c''1 \break c'' \break c'' }
\new Staff \with {
  instrumentName = "Violin"
  shortInstrumentName = "Vl." } {
  c''1 \break c'' \break c'' }
>>

```



Drawing boxes around grobs

The stencil property can be overridden to draw a box around arbitrary grobs, either using `\override` or `\tweak`.

```

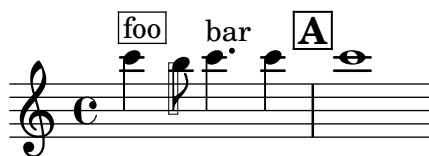
\relative c'' {
  \once \override TextScript.stencil =
    #(make-stencil-boxer 0.1 0.3 ly:text-interface::print)
  c'4^"foo"

  \tweak Stem.stencil
    #(make-stencil-boxer 0.05 0.25 ly:stem::print)
  b8

  c4.^"bar" c4

  \override Score.RehearsalMark.stencil =
    #(make-stencil-boxer 0.15 0.3 ly:text-interface::print)
  \mark \default
  c1
}

```



Drawing circles around note heads

A circle can be drawn around a note head by providing a custom Scheme function to temporarily override the stencil property.

```
circle = \tweak NoteHead.stencil
  #(\lambda (grob)
    (let* ((note (ly:note-head::print grob))
      (combo-stencil (ly:stencil-add
        note
        (circle-stencil note 0.1 0.8))))
      (ly:make-stencil (ly:stencil-expr combo-stencil)
        (ly:stencil-extent note X)
        (ly:stencil-extent note Y))))
  \etc

{ \circle c'' }
```

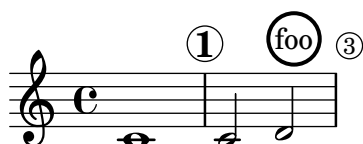


Drawing circles around various objects

The `\circle` command draws circles around `\markup` objects. For other objects, specific tweaks may be required, as demonstrated for rehearsal marks and measure numbers.

```
\relative c' {
  c1
  \set Score.rehearsalMarkFormatter =
    #(\lambda (mark context)
      (make-circle-markup (format-mark-numbers mark context)))
  \mark \default

  c2 d^\markup {
    \override #'(thickness . 3) {
      \circle foo
    }
  }
  \override Score.BarNumber.break-visibility = #all-visible
  \override Score.BarNumber.stencil =
    #(make-stencil-circler 0.1 0.25 ly:text-interface::print)
}
```

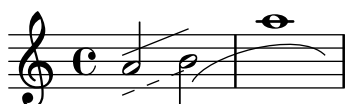


Embedding native PostScript in a `\markup` block

PostScript code can be directly inserted inside a `\markup` block.

In general it is recommended to use LilyPond's native graphical markup commands like `\polygon` instead, which can be used with all LilyPond backends.

```
\relative c' ' {
  a2-\markup \postscript "0 3 moveto
                        5 2 rlineto
                        stroke"
  -\markup \postscript "[1 1] 0 setdash
                        0 0 moveto
                        5 2 rlineto
                        stroke"
  b2-\markup \postscript "1 1 moveto
                        0 0 1 2 8 4 10 2 rcurveto
                        stroke"
  a'1
}
```



Generate special note head shapes

When a note head with a special shape cannot easily be generated with graphic markup, a drawing specification for `ly:make-stencil` can be used to generate the shape. This snippet gives an example for a parallelogram-shaped note head.

Unfortunately, the available commands in a drawing specification are currently not documented (this is tracked in Issue #6874 (<https://gitlab.com/lilypond/lilypond/-/issues/6874>)); in any case, the used path sub-command has the following signature, quite similar to the `make-path-stencil` Scheme function.

```
(path thickness command-list line-cap-style line-join-style fill)
```

The commands in *command-list* resemble PostScript drawing commands but with arguments after the command name.

```
parallelogram =
  #(ly:make-stencil
    '(path 0.1
      (rmoveto 0 0.25
        lineto 1.2 0.75
        lineto 1.2 -0.25
        lineto 0 -0.75
        lineto 0 0.25)
      round
      round
      #t)
    (cons -0.05 1.25)
    (cons -.75 .75))

myNoteHeads = \override NoteHead.stencil = \parallelogram
normalNoteHeads = \revert NoteHead.stencil
```

```
\relative c'' {
  \myNoteHeads
  g4 d'
  \normalNoteHeads
  <f, \tweak stencil \parallelogram b e>4 d
}
```



Gitternetzlinien: Aussehen verändern

Die Erscheinung der Gitternetzlinien kann durch einige Eigenschaften geändert werden.

```
\new ChoirStaff <<
  \new Staff {
    \relative c'' {
      \stemUp
      c'4. d8 e8 f g4
    }
  }
  \new Staff {
    \relative c {
      % this moves them up one staff space from the default position
      \override Score.GridLine.extra-offset = #'(0.0 . 1.0)
      \stemDown
      \clef bass
      \once \override Score.GridLine.thickness = 5.0
      c4
      \once \override Score.GridLine.thickness = 1.0
      g'4
      \once \override Score.GridLine.thickness = 3.0
      f4
      \once \override Score.GridLine.thickness = 5.0
      e4
    }
  }
}>>
```

```
\layout {
  \context {
    \Staff
    % set up grids
    \consists "Grid_point_engraver"
    % set the grid interval to one quarter note
    gridInterval = #1/4
  }
  \context {
    \Score
    \consists "Grid_line_span_engraver"
    % this moves them to the right half a staff space
    \override NoteColumn.X-offset = -0.5
  }
}
```

```
}
}
```



Grid lines: emphasizing rhythms and notes synchronization

Regular vertical lines can be drawn between staves to show note synchronization; however, in case of monophonic music, you may want to make the second staff invisible, and make the lines shorter like in this snippet.

```
\new ChoirStaff {
  \relative c'' <<
  \new Staff {
    \time 12/8
    \stemUp
    c4. d8 e8 f g4 f8 e8. d16 c8
  }
  \new Staff {
    % hides staff and notes so that only the grid lines are visible
    \hideNotes
    \hide Staff.BarLine
    \override Staff.StaffSymbol.line-count = #0
    \hide Staff.TimeSignature
    \hide Staff.Clef

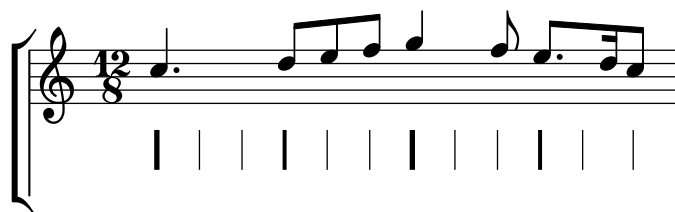
    % dummy notes to force regular note spacing
    \once \override Score.GridLine.thickness = #4.0
    c8 c c
    \once \override Score.GridLine.thickness = #3.0
    c8 c c
    \once \override Score.GridLine.thickness = #4.0
    c8 c c
    \once \override Score.GridLine.thickness = #3.0
    c8 c c
  }
}
>>
}
```

```
\layout {
  \context {
    \Score
    \consists "Grid_line_span_engraver"
    % center grid lines horizontally below note heads
    \override NoteColumn.X-offset = #-0.5
  }
}
```

```

\context {
  \Staff
  \consists "Grid_point_engraver"
  gridInterval = #1/8
  % set line length and positioning:
  % two staff spaces above center line on hidden staff
  % to four spaces below center line on visible staff
  \override GridPoint.Y-extent = #'(2 . -4)
}
}

```



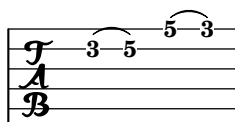
Hammer-on and pull-off

Hammer-on and pull-off can be obtained using slurs.

```

\new TabStaff {
  \relative c' {
    d4( e\2)
    a( g)
  }
}

```



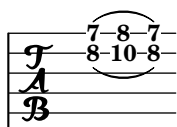
Hammer-on and pull-off using chords

When using hammer-on or pull-off with chorded notes, only a single arc is drawn. However „double arcs“ are possible by setting the doubleSlurs property to #t.

```

\new TabStaff {
  \relative c' {
    % chord hammer-on and pull-off
    \set doubleSlurs = ##t
    <g' b>8( <a c> <g b>)
  }
}

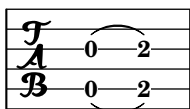
```



Hammer-on and pull-off using voices

The arc of hammer-on and pull-off is upwards in voices one and three and downwards in voices two and four:

```
\new TabStaff {
  \relative c' {
    << { \voiceOne g2( a) }
    \\ { \voiceTwo a,( b) }
    >> \oneVoice
  }
}
```



Eine Linie des Notensystems dicker als die anderen machen

Für den pädagogischen Einsatz kann eine Linie des Notensystems dicker gezeichnet werden (z. B. die Mittellinie, oder um den Schlüssel hervorzuheben). Das ist möglich, indem man zusätzliche Linien sehr nahe an der Linie, die dicker erscheinen soll, einfügt. Dazu wird die `line-positions`-Eigenschaft herangezogen.

```
{
  \override Staff.StaffSymbol.line-positions =
    #'(-4 -2 -0.2 0 0.2 2 4)
  d'4 e' f' g'
}
```



Marking notes of spoken parts with a cross on the stem (Sprechstimme)

This example shows how to put crosses on stems. Mark the beginning of a spoken section with the command `\speakOn` and end it with `\speakOff`.

```
speakOn = \override Stem.stencil =
  #(lambda (grob)
    (let* ((x-parent (ly:grob-parent grob X))
          (is-rest? (ly:grob? (ly:grob-object x-parent 'rest))))
      (if is-rest?
          empty-stencil
          (ly:stencil-combine-at-edge
            (ly:stem::print grob)
            Y
            (- (ly:grob-property grob 'direction))
            (grob-interpret-markup
              grob
              (markup #:center-align #:fontsize -4
                #:musicglyph "noteheads.s2cross"))
            -1.7))))

speakOff = \revert Stem.stencil

\new Staff {
```

```

\relative c'' {
  a4 b a c
  \speakOn
  g4 f r g8 a
  b4 r r8 d e4
  \speakOff
  c4 a g f
}

```



Measure counters

This snippet demonstrates the use of the `Measure_counter_engraver` to number groups of successive measures. Any stretch of measures may be numbered, whether consisting of repetitions or not.

The engraver must be added to the appropriate context. Here, a `Staff` context is used; another possibility is a `Dynamics` context.

The counter is begun with `\startMeasureCount` and ended with `\stopMeasureCount`. Numbering will start by default with 1, but this behavior may be modified by overriding the `count-from` property.

When a measure extends across a line break, the number will appear twice, the second time in parentheses.

```

\layout {
  \context {
    \Staff
    \consists #Measure_counter_engraver
  }
}

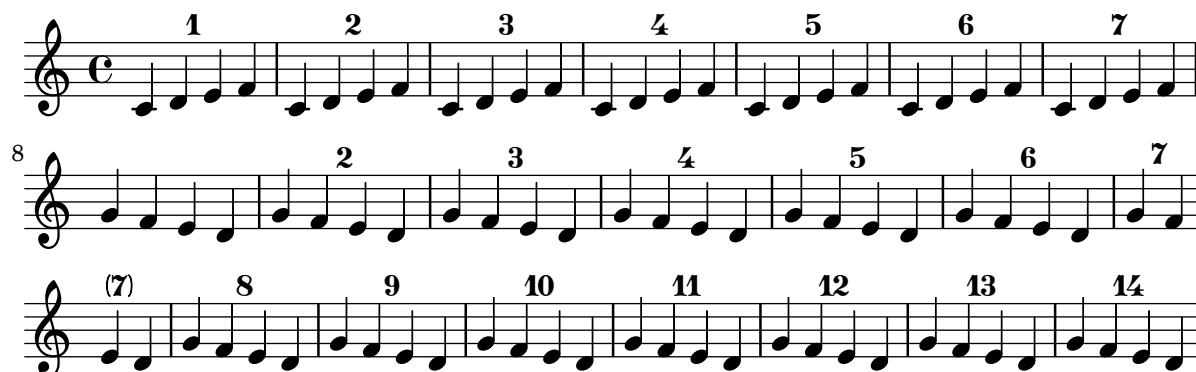
\new Staff {
  \startMeasureCount
  \repeat unfold 7 {
    c'4 d' e' f'
  }
  \stopMeasureCount
  \bar "||"
  g'4 f' e' d'
  \override Staff.MeasureCounter.count-from = #2
  \startMeasureCount
  \repeat unfold 5 {
    g'4 f' e' d'
  }
  g'4 f'
  \bar ""
  \break
  e'4 d'
  \repeat unfold 7 {
    g'4 f' e' d'
  }
}

```

```

}
\stopMeasureCount
}

```



Measure spanner

Measure spanners are an alternate way to print annotated brackets. As opposed to horizontal brackets, they extend between two bar lines rather than two notes. The text is displayed in the center of the bracket.

```

\layout {
  \context {
    \Staff
    \consists Measure_spanner_engraver
  }
}

```

```

<<
\new Staff \relative c'' {
  \key d \minor
  R1*2
  \tweak text "Answer"
  \startMeasureSpanner
  \tuplet 3/2 8 {
    a16[ b c] d[ c b] c[ d e] f[ e d]
  }
  e8 a gis g
  fis f e d~ d c b e
  \stopMeasureSpanner
}
\new Staff \relative c' {
  \key d \minor
  \tweak text "Subject"
  \tweak direction #DOWN
  \startMeasureSpanner
  \tuplet 3/2 8 {
    d16[ e f] g[ f e] f[ g a] bes[ a g]
  }
  a8 d cis c
  b bes a g~ g f e a
  \stopMeasureSpanner
  \tweak text "Counter-subject"
}

```

```

\ tweak direction #DOWN
\ startMeasureSpanner
f8 e a r r16 b, c d e fis g e
a gis a b c fis, b a gis e a4 g8
\ stopMeasureSpanner
}
>>

```

Subject

Answer

Counter-subject

Positioning fingering indications precisely

The semi-automatic positioning of fingering within a chords works fine in most situations. If one of the indications needs to be positioned more precisely the following, tweaks as shown in this snippet may be used. This is particularly useful for correcting the positioning when intervals of a second are involved.

```

\markup \with-true-dimensions % work around a cropping issue
\score {
  \relative c' {
    \set fingeringOrientations = #'(left)
    <c-1 d-2 a'-5>4
    <c-1 d-\tweak extra-offset #'(0 . 0.2)-2 a'-5>

    \set fingeringOrientations = #'(down)
    <c-1 d-2 a'-5>
    <c-\tweak extra-offset #'(0 . -1.1)-1
    d-\tweak extra-offset #'(-1.2 . -1.8)-2 a'-5> |

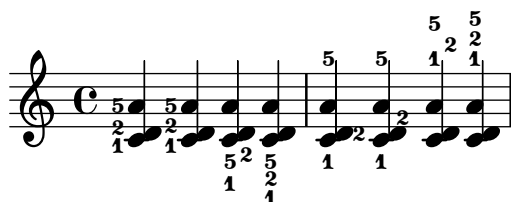
    \set fingeringOrientations = #'(down right up)
    <c-1 d-\tweak extra-offset #'(-0.3 . 0)-2 a'-5>4
    <c-1 d-\tweak extra-offset #'(-1 . 1.2)-2 a'-5>

    \set fingeringOrientations = #'(up)
    <c-1 d-\tweak extra-offset #'(0 . 1.1)-2
    a'-\tweak extra-offset #'(0 . 1)-5>
    <c-1 d-\tweak extra-offset #'(-1.2 . 1.5)-2
    a'-\tweak extra-offset #'(0 . 1.4)-5> |
  }
}

```



```
}
}
```



Textbeschriftung innerhalb von Bögen positionieren

Textbeschriftung kann innerhalb von Bögen gesetzt werden, wenn die `outside-staff-priority`-Eigenschaft auf falsch gesetzt wird.

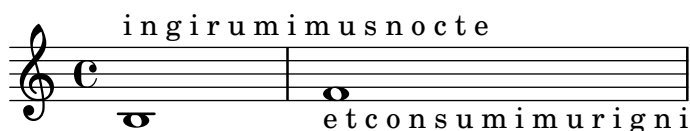
```
\relative c' {
  \override TextScript.avoid-slur = #'inside
  \override TextScript.outside-staff-priority = ##f
  c2(\markup { \halign #-10 \natural } d4.) c8
}
```



Printing text from right to left

It is possible to print text from right to left in a markup object, as demonstrated here.

```
{
  b1^\markup {
    \line { i n g i r u m i m u s n o c t e }
  }
  f'_\markup {
    \override #'(text-direction . -1)
    \line { i n g i r u m i m u s n o c t e }
  }
}
```



String number extender lines

Make an extender line for string number indications, showing that a series of notes is supposed to be played all on the same string.

```
stringNumberSpanner =
  \define-music-function (StringNumber) (string?)
  #{
    \override TextSpanner.style = #'solid
    \override TextSpanner.font-size = #-5
    \override TextSpanner.bound-details.left.stencil-align-dir-y = #CENTER
    \override TextSpanner.bound-details.left.text =
```


8 Text

See also Abschnitt “Text” in *Notationsreferenz*.

Adding markups in a tablature

By default, markups are not displayed in a tablature.

To make them appear, revert the `stencil` property of the `TextScript` grob in the `TabStaff` context.

```
high = { r4 r8 <g c'> q r8 r4 }
low = { c4 r4 c8 r8 g,8 b, }
pulse = { s8^"1" s^"&" s^"2" s^"&" s^"3" s^"&" s^"4" s^"&" }
```

```
\score {
  \new TabStaff {
    \repeat unfold 2 << \high \\\ \low \\\ \pulse >>
  }
  \layout {
    \context {
      \TabStaff
      \clef moderntab
      \revert TextScript.stencil
      \override TextScript.font-series = #'bold
      \override TextScript.font-size = #-2
      \override TextScript.color = #red
    }
    \context {
      \Score
      proportionalNotationDuration = #1/8
    }
  }
}
```

	1	&	2	&	3	&	4	&	1	&	2	&	3	&	4	&
T					1-1								1-1			
A					0-0								0-0			
B	3				3			2	3				3			2
							3								3	

Adding the current date to a score

With a little Scheme code, the current date can easily be added to a score.

```
\paper { tagline = ##f }
```

```
% first, define a variable to hold the formatted date:
date = #(strftime "%d-%m-%Y" (localtime (current-time)))
```

```
% use it in the title block:
```

```
\header {
  title = "Including the date!"
  subtitle = \date
}
```

```
\score {
  \relative c' {
    c4 c c c
  }
}
% and use it in a \markup block:
\markup {
  \date
}
```

Including the date!

21-02-2026



21-02-2026

Adjusting vertical spacing of lyrics

This snippet shows how to bring the lyrics line closer to the staff.

```
music = \relative c' { c4 d e f | g4 f e d | c1 }
text = \lyricmode { aa aa aa aa aa aa aa aa }
```

```
<<
  \new Staff \new Voice = melody \music
  % Default layout:
  \new Lyrics \lyricsto melody \text

  \new Staff \new Voice = melody \music
  % Reducing the minimum space below the staff and above the lyrics.
  \new Lyrics \with {
    \override VerticalAxisGroup.nonstaff-relatedstaff-spacing =
      #'((basic-distance . 1))
  } \lyricsto melody \text
>>
```



Aligning and centering instrument names

The horizontal alignment of instrument names is tweaked by changing the `self-alignment-X` property of the `InstrumentName` grob (usually in the `Staff` context). The `\layout` variables `indent` and `short-indent` define the space in which the instrument names are aligned before the first and the following systems, respectively.

```
\paper {
```

```

    left-margin = 3\cm
}

\new StaffGroup <<
  \new Staff \with {
    \override InstrumentName.self-alignment-X = #LEFT
    instrumentName = \markup \left-column { "Left aligned"
                                             "instrument name" }

    shortInstrumentName = "Left"
  } {
    c'1 \break c'1
  }

  \new Staff \with {
    \override InstrumentName.self-alignment-X = #CENTER
    instrumentName = \markup \center-column { Centered
                                             "instrument name" }

    shortInstrumentName = "Centered"
  } {
    g'1 g'1
  }

  \new Staff \with {
    \override InstrumentName.self-alignment-X = #RIGHT
    instrumentName = \markup \right-column { "Right aligned"
                                             "instrument name" }

    shortInstrumentName = "Right"
  } {
    e'1 e'1
  }
>>


\layout {
  indent = 4\cm
  short-indent = 2\cm
  line-width = 6.5\cm
}

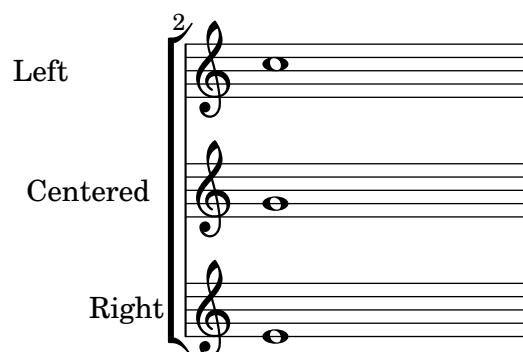
```

Left aligned
instrument name

Centered
instrument name

Right aligned
instrument name



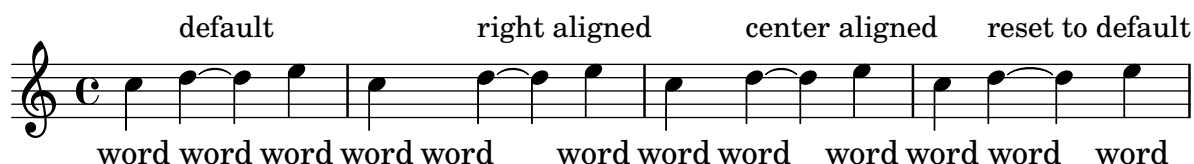


Aligning syllables with melisma

By default, lyrics syllables that start a melisma are left-aligned on their note. The alignment can be altered using the `lyricMelismaAlignment` property.

```
<<
\new Staff {
  \new Voice = "vocal" \relative c' {
    \override TextScript.staff-padding = #2
    c d~\markup default d e
    c d~\markup "right aligned" d e
    c d~\markup "center aligned" d e
    c d~\markup "reset to default" d e
  }
}
\new Lyrics \lyricsto "vocal" {
  word word word
  \set lyricMelismaAlignment = #RIGHT
  word word word
  \set lyricMelismaAlignment = #CENTER
  word word word
  \unset lyricMelismaAlignment
  word word word
}
>>

\layout {
  ragged-right = ##f
}
```



Aligning text marks to notes

By default, `TextMark` objects are aligned to so-called `NonMusicalPaperColumn` grobs, like the left edge of the staff or a bar line. They can be aligned to a note instead by setting the `non-musical` property to `#f`.

```
\layout {
  line-length = 80\mm
}
```

```

}

{
  \textMark "mark a" c'1 |
  \textMark "mark b" c'1 |
  \break
  \override Score.TextMark.non-musical = ##f
  \textMark "mark c" c'1 |
  \textMark "mark d" c'1 |
}

```



Blanking staff lines using the `\whiteout` command

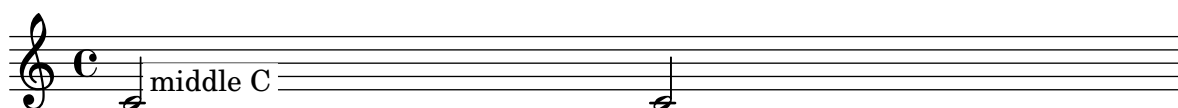
The `\whiteout` command underlays a markup with a white box. Since staff lines are in a lower layer than most other grobs, this white box will not overlap any other grob.

```

\layout {
  ragged-right = ##f
}

\relative c' {
  \override TextScript.extra-offset = #'(2 . 4)
  c2-\markup { \whiteout \pad-markup #0.5 "middle C" } c
}

```



Center text below hairpin dynamics

This example provides a function to typeset a hairpin (de)crescendo with some additional text below it, such as „molto“ or „poco“. The added text will change the direction according to the direction of the hairpin. The Hairpin is aligned to a `DynamicText` grob.

The example also illustrates how to modify the way an object is normally printed, using some Scheme code.

```

hairpinWithCenteredText =
#(define-music-function (text) (markup?)
  #{
    \once \override Voice.Hairpin.after-line-breaking =
      #(lambda (grob)
        (let* ((stencil (ly:hairpin::print grob))
              (par-y (ly:grob-parent grob Y))
              (dir (ly:grob-property par-y 'direction)))

```

```

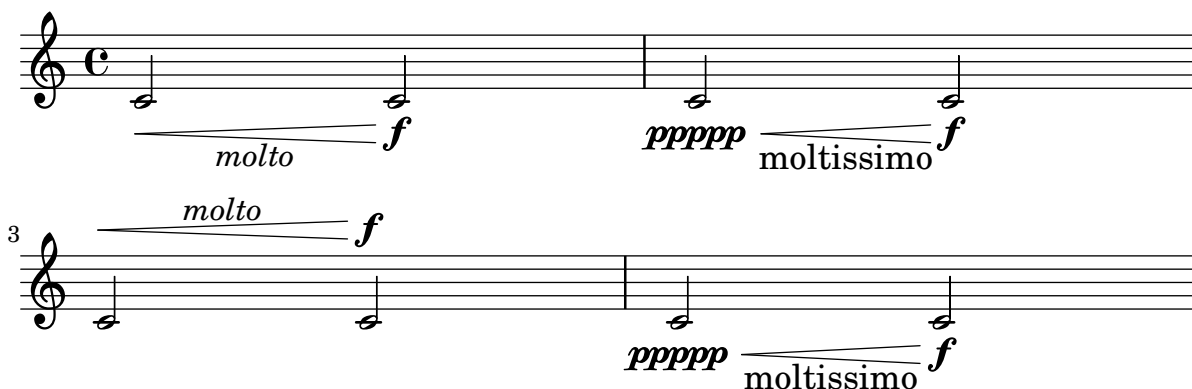
(staff-line-thickness
  (ly:output-def-lookup (ly:grob-layout grob)
                        'line-thickness))
(new-stencil
  (ly:stencil-aligned-to
    (ly:stencil-combine-at-edge
      (ly:stencil-aligned-to stencil X CENTER)
      Y dir
      (ly:stencil-aligned-to
        (grob-interpret-markup
          grob
          (make-fontsize-markup
            (magnification->font-size
              (+ (ly:staff-symbol-staff-space grob)
                (/ staff-line-thickness 2)))
            text))
          X CENTER))
      X LEFT))
  (staff-space (ly:output-def-lookup
    (ly:grob-layout grob) 'staff-space))
  (par-x (ly:grob-parent grob X))
  (dyn-text (grob::has-interface par-x
                                'dynamic-text-interface))

  (dyn-text-stencil-x-length
    (if dyn-text
      (interval-length
        (ly:stencil-extent
          (ly:grob-property par-x 'stencil) X))
      0))
  (x-shift
    (if dyn-text (- (+ staff-space dyn-text-stencil-x-length)
                    (* 0.5 staff-line-thickness))
      0)))
(ly:grob-set-property! grob 'Y-offset 0)
(ly:grob-set-property! grob
  'stencil (ly:stencil-translate-axis
    new-stencil
    x-shift X))))
#})

hairpinMolto = \hairpinWithCenteredText \markup { \italic molto }
hairpinMore = \hairpinWithCenteredText \markup { \larger moltissimo }

\relative c' {
  \hairpinMolto c2\< c\f
  \hairpinMore c2\ppppp\< c\f
  \break
  \hairpinMolto c2^\< c\f
  \hairpinMore c2\ppppp\< c\f
}

```

Changing ottava text

Internally, `\ottava` sets the properties `ottavation` (for example, to 8va or 8vb) and `middleCPosition`. To override the text of the bracket, set `ottavation` after invoking `\ottava`.

Short text is especially useful when a brief ottava is used.

```
{
  c'2
  \ottava 1
  \set Staff.ottavation = "8"
  c''2
  \ottava 0
  c'1
  \ottava 1
  \set Staff.ottavation = "Text"
  c''1
}
```



Changing the default text font family

The default font families for text can be overridden.

```
%{
You may have to install additional fonts.
```

```
Red Hat Fedora: dejavu-fonts-all
```

```
Debian GNU/Linux, Ubuntu: fonts-dejavu-core
                           fonts-dejavu-extra
```

```
%}
```

```
\paper {
  %{
    run
      lilypond -dshow-available-fonts
      to show all fonts available in the process log.
  %}
  property-defaults.fonts.serif = "DejaVu Serif"
```

```

property-defaults.fonts.sans = "DejaVu Sans"
property-defaults.fonts.typewriter = "DejaVu Sans Mono"
}

{
  g'''4^\markup {
    DejaVu Serif: \bold bold
                  \italic italic
                  \italic \bold { bold italic }
  }
  g4_\markup {
    \override #'(font-family . sans) {
      DejaVu Sans: \bold bold
                  \italic italic
                  \italic \bold { bold italic }
    }
  }
  g''2^\markup {
    \override #'(font-family . typewriter) {
      DejaVu Sans Mono: \bold bold
                      \italic italic
                      \italic \bold { bold italic }
    }
  }
}

```



Combining dynamics with markup texts

Some dynamics may involve text indications (such as „più *f*“ or „*p* subito“). These can be produced using a `\markup` block; the resulting object behaves like a `TextScript` grob.

See also „Combining dynamics with markup texts (2)“.

```

piuF = \markup { \italic più \dynamic f }

```

```

\markup \with-true-dimensions % work around a cropping issue
\score {
  \relative c'' {
    c2\f c-\piuF
  }
}

```



Combining dynamics with markup texts (2)

Some dynamics may involve text indications (such as „*più f*“ or „*p subito*“). These can be produced using the `make-dynamic-script` Scheme function; the resulting object behaves like a `DynamicText` grob.

See also „Combining dynamics with markup texts“.

```
piuF = #(make-dynamic-script
        #{ \markup { \normal-text \italic più \dynamic f } #})
```

```
\score {
  \relative c'' {
    c2\f c\piuF
  }
}
```



Zwei Stimmen auf einem System kombinieren

Die Funktion, die Stimmen kombiniert (also der `\partCombine`-Befehl) ermöglicht die Kombination unterschiedlicher Stimmen auf einem System. Textanweisungen wie "solo" oder "a2" werden automatisch hinzugefügt. Wenn man sie entfernen will, muss man die Eigenschaft `printPartCombineTexts` auf falsch setzen. Für Klavierauszüge muss natürlich kein "solo"/"a2" usw. hinzugefügt werden, man sollte sie also ausschalten. Wenn aber Solo-Stellen in einem Klavierauszug oder einer Chorpartitur angezeigt werden, ist es besser, normale Polyphonie zu verwenden, weil so die Solostellen angezeigt werden, auch wenn der Text des Stimmenkombinierers ausgeschaltet ist.

Der Schnipsel zeigt drei Möglichkeiten, Stimmen auf einem System zu kombinieren: Standardpolyphonie, `\partCombine` ohne Text und `\partCombine` mit Text.

```
musicUp = \relative c'' {
  \time 4/4
  a4 c4.( g8) a4 |
  g4 e' g,( a8 b) |
  c b a2.
}
```

```
musicDown = \relative c'' {
  g4 e4.( d8) c4 |
  r2 g'4( f8 e) |
  d2 \stemDown a
}
```

```
\score {
  <<
    \new Staff \with {
      instrumentName = "standard polyphony"
    } << \musicUp \\\musicDown >>

    \new Staff \with {
      instrumentName =
```

```

\markup { \typewriter "\\partCombine" without text}
printPartCombineTexts = ##f
} \partCombine \musicUp \musicDown

\new Staff \with {
  instrumentName =
    \markup { \typewriter "\\partCombine" with text}
} \partCombine \musicUp \musicDown
>>

\layout {
  indent = 6.0\cm
  \context {
    \Score
    % Setting this to a large value avoids a bar line at the
    % beginning that would connect the three staves otherwise.
    \override SystemStartBar.collapse-height = 30
  }
}

```

standard polyphony	
\partCombine without text	
\partCombine with text	

Creating „real“ parenthesized dynamics

Although the easiest way to add parentheses to a dynamic mark is to use a `\markup` block, this method has a downside: the created objects behave like text markups and not like dynamics.

However, it is possible to create a similar object using the equivalent Scheme code (as described in the Notation Reference), combined with the `make-dynamic-script` function. This way, the markup is regarded as a dynamic and therefore remains compatible with commands such as `\dynamicUp` or `\dynamicDown`.

```

paren =
#(define-event-function (dyn) (ly:event?)
  (make-dynamic-script
    #{ \markup \concat {
      \normal-text \italic \fontsize #2 (
        \pad-x #0.2 #(ly:music-property dyn 'text)
        \normal-text \italic \fontsize #2 )
    }
    #}))

\relative c' {

```

```
c4\paren\f c c \dynamicUp c\paren\p
}
```



Creating text spanners

The `\startTextSpan` and `\stopTextSpan` commands allow the creation of text spanners as easily as pedal indications or octavations. Override some properties of the `TextSpanner` object to modify its output.

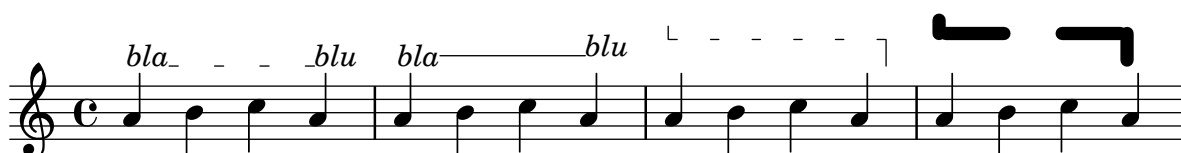
```
\paper { ragged-right = ##f }
```

```
\relative c' {
  \override TextSpanner.bound-details.left.text = #"bla"
  \override TextSpanner.bound-details.right.text = #"blu"
  a4 \startTextSpan
  b4 c
  a4 \stopTextSpan

  \override TextSpanner.style = #'line
  \once \override TextSpanner.bound-details.left.stencil-align-dir-y = #CENTER
  a4 \startTextSpan
  b4 c
  a4 \stopTextSpan

  \override TextSpanner.style = #'dashed-line
  \override TextSpanner.bound-details.left.text =
    \markup { \draw-line #'(0 . 1) }
  \override TextSpanner.bound-details.right.text =
    \markup { \draw-line #'(0 . -2) }
  \once \override TextSpanner.bound-details.right.padding = #-2
  a4 \startTextSpan
  b4 c
  a4 \stopTextSpan

  \override TextSpanner.dash-period = #10
  \override TextSpanner.dash-fraction = #0.5
  \override TextSpanner.thickness = #10
  a4 \startTextSpan
  b4 c
  a4 \stopTextSpan
}
```



Demonstrating all \header fields

A demonstration of all header fields that LilyPond defines by default. Thanks to setting `print-all-headers` to `#t`, much more fields as usual are displayed, indicating the hierarchy of `\header` blocks.


```
\paper {
  #(set-paper-size "a6" 'landscape)
  print-all-headers = ##t
}

\book {
  \header {
    title = "title"
    subtitle = "subtitle"
    composer = "composer"
    arranger = "arranger"
    instrument = "instrument"
    meter = "meter"
    opus = "opus"
    piece = "piece"
    poet = "poet"
    copyright = "copyright"
    tagline = "tagline"
  }

  \bookpart {
    \score {
      \relative c'' { c1 | c | c | c }

      \header {
        title = "localtitle"
        subtitle = "localsubtitle"
        composer = "localcomposer"
        arranger = "localarranger"
        instrument = "localinstrument"
        meter = "localmeter"
        opus = "localopus"
        piece = "localpiece"
        poet = "localpoet"
        copyright = "localcopyright"
        tagline = "localtagline"
      }
    }
  }
}
```

	title	
	subtitle	
poet	instrument	composer
meter		arranger
	localtitle	
	localsubtitle	
localpoet	localinstrument	localcomposer
localmeter		localarranger
localpiece		localopus



	copyright	
	tagline	

Embedding native PostScript in a \markup block

PostScript code can be directly inserted inside a \markup block.

In general it is recommended to use LilyPond's native graphical markup commands like \polygon instead, which can be used with all LilyPond backends.

```
\relative c' ' {
  a2-\markup \postscript "0 3 moveto
                        5 2 rlineto
                        stroke"
  -\markup \postscript "[1 1] 0 setdash
                        0 0 moveto
                        5 2 rlineto
                        stroke"
  b2-\markup \postscript "1 1 moveto
                        0 0 1 2 8 4 10 2 rcurveto
                        stroke"
  a'1
}
```



Silben im Gesangstext formatieren

Textbeschriftungsmodus kann eingesetzt werden, um individuelle Silben im Gesangstext zu formatieren.

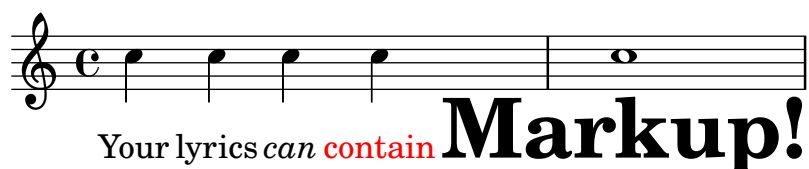
```
mel = \relative c' ' { c4 c c c c1 }
lyr = \lyricmode {
  Your lyrics \markup { \italic can }
  \markup { \with-color #red contain }
  \markup { \fontsize #8 \bold Markup! }
```

```

}

<<
  \new Voice = "melody" \mel
  \new Lyrics \lyricsto "melody" \lyr
>>

```



How to put ties between syllables in lyrics

This can be achieved by separating those syllables by tildes.

```

\lyrics {
  wa~o~a
}

```

waoa

Ausrichtung von Gesangstext

Die horizontale Ausrichtung von Gesangstext kann eingestellt werden, indem man die `self-alignment-X`-Eigenschaft des `LyricText`-Objekts verändert. #-1 bedeutet links, #0 bedeutet mittig und #1 bedeutet rechts, man kann aber genauso gut auch #LEFT, #CENTER und #RIGHT benutzen.

```

\layout {
  ragged-right = ##f
}

```

```

\relative c' {
  c1 c c c
}

```

```

\addlyrics {
  \once \override LyricText.self-alignment-X = #LEFT
  "left-aligned"
  \once \override LyricText.self-alignment-X = #CENTER
  "centered"
  \once \override LyricText.self-alignment-X = 1
  "right-aligned"
  \once \override LyricText.self-alignment-X = #-1.5
  "very right"
}

```



Markup list

Text that can spread over pages is entered with the `\markuplist` command. The `\paragraph` markup command defined in the snippet indents its argument before calling `\justified-lines`.

```
#(set-default-paper-size "a6" 'landscape)

\paper {
  line-width = 11\cm
  tagline = ##f
}

#(define-markup-list-command (paragraph layout props args) (markup-list?)
  (interpret-markup-list layout props
    (make-justified-lines-markup-list (cons (make-hspace-markup 2) args))))

\book { % for correct rendering in the PDF documentation
  % Candide, Voltaire
  \markuplist {
    \override-lines #'(baseline-skip . 2.5) {
      \paragraph {
        Il y avait en Westphalie, dans le château de M. le baron de
        Thunder-ten-tronckh, un jeune garçon à qui la nature avait donné
        les mœurs les plus douces. Sa physionomie annonçait son âme.
        Il avait le jugement assez droit, avec l'esprit le plus
        \concat { simple \hspace #.3 ; }
        c'est, je crois, pour cette raison qu'on le nommait Candide. Les
        anciens domestiques de la maison soupçonnaient qu'il était fils
        de la sœur de monsieur le baron et d'un bon et honnête
        gentilhomme du voisinage, que cette demoiselle ne voulut jamais
        épouser parce qu'il n'avait pu prouver que soixante et onze
        quartiers, et que le reste de son arbre généalogique avait été
        perdu par l'injure du temps.
      }
      \vspace #.3
      \paragraph {
        Monsieur le baron était un des plus puissants seigneurs de la
        Westphalie, car son château avait une porte et des fenêtres. Sa
        grande salle même était ornée d'une tapisserie. Tous les chiens
        de ses basses-cours composaient une meute dans le
        \concat { besoin \hspace #.3 ; }
        ses palefreniers étaient ses
        \concat { piqueurs \hspace #.3 ; }
        le vicaire du village était
        son grand-aumônier. Ils l'appelaient tous monseigneur, et ils
        riaient quand il faisait des contes.
      }
    }
  }
}
```

Il y avait en Westphalie, dans le château de M. le baron de Thunder-ten-tronckh, un jeune garçon à qui la nature avait donné les mœurs les plus douces. Sa physionomie annonçait son âme. Il avait le jugement assez droit, avec l'esprit le plus simple; c'est, je crois, pour cette raison qu'on le nommait Candide. Les anciens domestiques de la maison soupçonnaient qu'il était fils de la sœur de monsieur le baron et d'un bon et honnête gentilhomme du voisinage, que cette demoiselle ne voulut jamais épouser parce qu'il n'avait pu prouver que soixante et onze quartiers, et que le reste de son arbre généalogique avait été perdu par l'injure du temps.

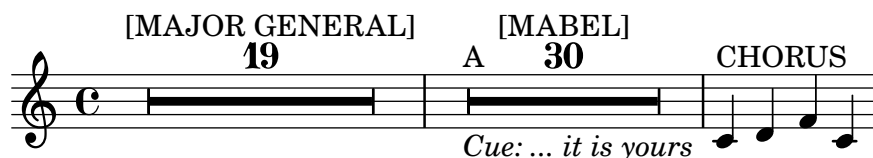
Monsieur le baron était un des plus puissants seigneurs de la Westphalie, car son château avait une porte et des fenêtres. Sa grande salle même était ornée d'une tapisserie. Tous les chiens de ses basses-cours composaient une meute dans le besoin; ses palefreniers étaient ses piqueurs; le vicaire du village était son grand-aumônier. Ils l'appelaient tous monseigneur, et ils riaient quand il faisait des contes.

Textbeschriftung und Mehrtaktpausen

Textbeschriftungen, die an Mehrtaktpausen gehängt wird, wird über oder unter der Pause zentriert. Lange Beschriftungen lassen den Takt nicht breiter werden. Um eine Mehrtaktpause einer Beschriftung anzupassen, muss eine unsichtbare Pause mit der Beschriftung direkt vor der Mehrtaktpause eingesetzt werden.

Man sollte beachten, dass unsichtbare Pausen automatische Taktstriche nach sich ziehen. Text, der an eine unsichtbare Pause gehängt wird, ist links ausgerichtet an der Position, wo die Pause erscheinen würde. Wenn aber die Länge des Taktes durch die Länge des Textes bestimmt wird, sieht es so aus, als ob der Text zentriert gesetzt ist.

```
\relative c' {
  \compressMMRests {
    \textLengthOn
    <>^\markup { [MAJOR GENERAL] }
    R1*19
    <>_\markup { \italic { Cue: ... it is yours } }
    <>^\markup { A }
    R1*30^\markup { [MABEL] }
    \textLengthOff
    c4^\markup { CHORUS } d f c
  }
}
```



Of the ubiquity of markup objects

Text objects are entered either as simple strings between double quotes or as `\markup` blocks that can accept a variety of advanced text formatting and graphical enhancements.

As such, markup blocks may be used:

- in any `TextScript` object (attached to notes with `-`, `^` or `_`),

- in any TextMark introduced with the `\textMark` keyword, or `\textEndMark` command, or other similar objects such as MetronomeMark introduced with `\tempo`,
- as standalone markup blocks, entered at the top level outside of any `\score` block,
- in any definition inside the `\header` block (e.g., title, subtitle, composer) or in some variables defined inside the `\paper` block such as `evenHeaderMarkup` for page numbers.

`\markup` may additionally be used for lyrics, in chord names, and as dynamics. In fact, it is possible to use `\markup` to customize the appearance of virtually any object, as demonstrated in this example using various methods.

```
\paper {
  paper-width = 8\cm
  paper-height = 8\cm
}

\header {
  title = \markup "Title"
  tagline = \markup "(tagline)"
}

\markup "Top-level markup"

dyn = #(make-dynamic-script #{ \markup \serif "DynamicText" #})

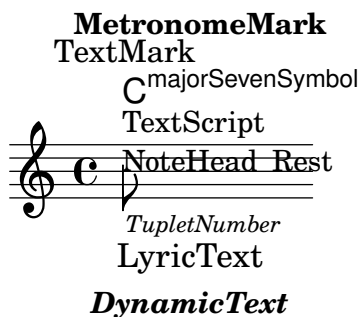
<<
  \new ChordNames \with {
    majorSevenSymbol = \markup "majorSevenSymbol"
  } \chordmode { c1:maj7 }
  \new Staff {
    \tempo \markup "MetronomeMark"
    \textMark \markup "TextMark"

    \once \override TupletNumber.text = \markup "TupletNumber"
    \tuplet 3/2 {
      \once \override NoteHead.stencil = #ly:text-interface::print
      \once \override NoteHead.text = \markup \lower #0.5 "NoteHead"
      c' '8^\markup "TextScript"

      \once \override Rest.stencil = #(lambda (grob)
        (grob-interpret-markup grob #{ \markup "Rest" #}))
      r4
    }
  }
  \new Lyrics \lyricmode { \markup "LyricText" 1 }
  \new Dynamics { s1\dyn }
>>
```

Title

Top-level markup



Outputting the version number

It is possible to print the version number of LilyPond in markup.

```
\markup { Processed with LilyPond version #(lilypond-version) }
```

Processed with LilyPond version 2.25.34

Piano template with centered lyrics

Anstatt ein eigenes System für Melodie und Text zu schreiben, können Sie den Text auch zwischen die beiden Klaviersysteme schreiben (und damit das zusätzliche System für die Gesangsstimme auslassen).

```
upper = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

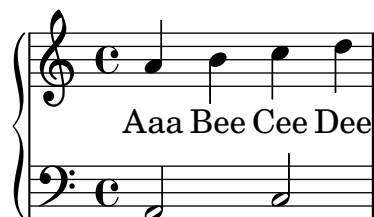
  a4 b c d
}

lower = \relative c {
  \clef bass
  \key c \major
  \time 4/4

  a2 c
}

text = \lyricmode {
  Aaa Bee Cee Dee
}

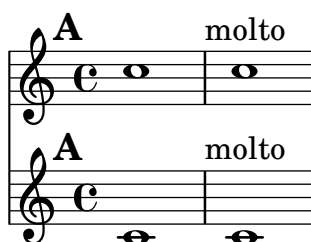
\score {
  \new PianoStaff <<
    \new Staff = upper { \new Voice = "singer" \upper }
    \new Lyrics \lyricsto "singer" \text
    \new Staff = lower { \lower }
  >>
  \layout { }
  \midi { }
}
```



Zeichen über jedem System ausgeben

Normalerweise werden Textzeichen nur über dem obersten Notensystem gesetzt. Sie können aber auch über jedem System ausgegeben werden.

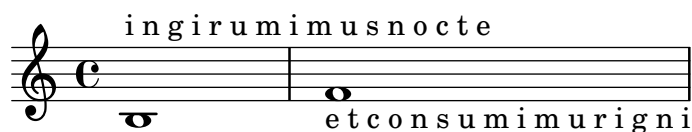
```
\score {
  <<
    \new Staff { \mark \default c''1 \textMark "molto" c'' }
    \new Staff { \mark \default c'1 \textMark "molto" c' }
  >>
  \layout {
    \context {
      \Score
      \remove Mark_engraver
      \remove Text_mark_engraver
      \remove Staff_collecting_engraver
    }
    \context {
      \Staff
      \consists Mark_engraver
      \consists Text_mark_engraver
      \consists Staff_collecting_engraver
    }
  }
}
```



Printing text from right to left

It is possible to print text from right to left in a markup object, as demonstrated here.

```
{
  b1~\markup {
    \line { i n g i r u m i m u s n o c t e }
  }
  f'_~\markup {
    \override #'(text-direction . -1)
    \line { i n g i r u m i m u s n o c t e }
  }
}
```



Putting lyrics inside the staff

Lyrics can be moved vertically to place them inside the staff. The lyrics are moved with `\override LyricText.extra-offset = #'(0 . dy)`, and there are similar commands to move the extenders and hyphens. A good value for *dy* must be found by trial and error.

```
<<
\new Staff <<
  \new Voice = "voc" \relative c' { \stemDown a bes c8 b c4 }
>>
\new Lyrics \with {
  \override LyricText.extra-offset = #'(0 . 8.6)
  \override LyricExtender.extra-offset = #'(0 . 8.6)
  \override LyricHyphen.extra-offset = #'(0 . 8.6)
} \lyricsto "voc" { La la -- la __ _ la }
>>
```



Isolierter Text in zwei Spalten

Isolierter Text kann in mehreren Spalten mit `\markup`-Befehlen angeordnet werden:

```
\markup {
  \fill-line {
    \hspace #1
    \column {
      \line { 0 sacrum convivium }
      \line { in quo Christus sumitur, }
      \line { recolitur memoria passionis ejus, }
      \line { mens impletur gratia, }
      \line { futurae gloriae nobis pignus datur. }
      \line { Amen. }
    }
    \hspace #2
    \column \italic {
      \line { 0 sacred feast }
      \line { in which Christ is received, }
      \line { the memory of His Passion is renewed, }
      \line { the mind is filled with grace, }
      \line { and a pledge of future glory is given to us. }
      \line { Amen. }
    }
  }
  \hspace #1
}
```

O sacrum convivium
 in quo Christus sumitur,
 recolitur memoria passionis ejus,
 mens impletur gratia,
 futurae gloriae nobis pignus datur.
 Amen.

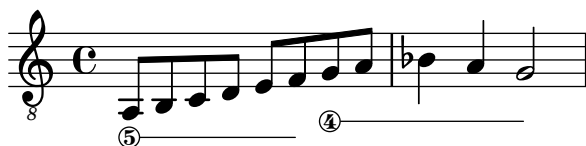
*O sacred feast
 in which Christ is received,
 the memory of His Passion is renewed,
 the mind is filled with grace,
 and a pledge of future glory is given to us.
 Amen.*

String number extender lines

Make an extender line for string number indications, showing that a series of notes is supposed to be played all on the same string.

```
stringNumberSpanner =
  #(define-music-function (StringNumber) (string?)
    #{
      \override TextSpanner.style = #'solid
      \override TextSpanner.font-size = #-5
      \override TextSpanner.bound-details.left.stencil-align-dir-y = #CENTER
      \override TextSpanner.bound-details.left.text =
        \markup { \circle \number $StringNumber }
    #})

\relative c {
  \clef "treble_8"
  \textSpannerDown
  \stringNumberSpanner "5" a8\startTextSpan b c d
  e f\stopTextSpan \stringNumberSpanner "4" g\startTextSpan a |
  bes4 a g2\stopTextSpan
}
```



Three-sided box

This example shows how to add a markup command to get a three-sided box around some text (or other markup).

```
% New command to add a three-sided box, with sides north, west, and south.
% Based on the `box-stencil` command defined in `scm/stencil.scm`.
% Note that ";" is used to comment a line in Scheme.
#(define-public (NWS-box-stencil stencil thickness padding)
  "Add a box around STENCIL, producing a new stencil."
  (let* ((x-ext (interval-widen (ly:stencil-extent stencil X) padding))
        (y-ext (interval-widen (ly:stencil-extent stencil Y) padding))
        (y-rule (make-filled-box-stencil (cons 0 thickness) y-ext))
        (x-rule (make-filled-box-stencil
                  (interval-widen x-ext thickness) (cons 0 thickness))))
    ;; (set! stencil (ly:stencil-combine-at-edge stencil X 1 y-rule padding))
    (set! stencil (ly:stencil-combine-at-edge stencil X LEFT y-rule padding))
    (set! stencil (ly:stencil-combine-at-edge stencil Y UP x-rule 0.0))
    (set! stencil (ly:stencil-combine-at-edge stencil Y DOWN x-rule 0.0))
```

```

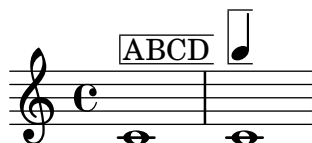
stencil))

% The corresponding markup command, based on the '\box' command defined
% in `scm/define-markup-commands.scm`.
#(define-markup-command (NWS-box layout props arg) (markup?)
  #:properties ((thickness 0.1) (font-size 0) (box-padding 0.2))
  "Draw a box round ARG.

Look at THICKNESS, BOX-PADDING, and FONT-SIZE properties to determine
line thickness and padding around the markup."
  (let ((pad (* (magstep font-size) box-padding))
        (m (interpret-markup layout props arg)))
    (NWS-box-stencil m thickness pad)))

\relative c' {
  c1^\markup { \NWS-box ABCD }
  c1^\markup { \NWS-box \note {4} #1.0 }
}

```



UTF-8

Various scripts may be used for texts (like titles and lyrics) by entering them in UTF-8 encoding, and using a Pango based backend. Depending on the fonts installed, this fragment will render Bulgarian (Cyrillic), Hebrew, Japanese and Portuguese.

```

%{
You may have to install additional fonts.

Red Hat Fedora: linux-libertine-fonts (Latin, Cyrillic, Hebrew)
                google-noto-serif-jp-fonts (Japanese)

Debian GNU/Linux, Ubuntu: fonts-linuxlibertine (Latin, Cyrillic, Hebrew)
                          fonts-noto-cjk (Japanese)
%}

% 'Linux Libertine' fonts also contain Cyrillic and Hebrew glyphs.
\paper {
  property-defaults.fonts.serif =
    "Linux Libertine 0, Noto Serif CJK JP, Noto Serif JP"
}

bulgarian = \lyricmode {
  Жълтата дюля беше щастлива, че пухът, който цъфна, замръзна като гьон.
}

hebrew = \lyricmode {
  .
}

```


}

japanese = \lyricmode {

}

% "a nice song for you"

portuguese = \lyricmode {

à vo -- cê uma can -- ção le -- gal

}

\relative c' {

c2 d

e2 f

g2 f

e2 d

}

\addlyrics { \bulgarian }

\addlyrics { \hebrew }

\addlyrics { \japanese }

\addlyrics { \portuguese }

The image shows a musical score for a vocal ensemble. It consists of two staves of music, each with four measures. The first staff has a treble clef and a common time signature (C). The notes are: quarter, quarter, half, and quarter. The second staff has a treble clef and a common time signature (C). The notes are: quarter, quarter, half, and quarter. The lyrics are written below the notes in four columns, corresponding to the four measures of each staff. The lyrics are in Bulgarian, Hebrew, Japanese, and Portuguese. The first staff has lyrics in Bulgarian, Hebrew, Japanese, and Portuguese. The second staff has lyrics in Bulgarian, Hebrew, Japanese, and Portuguese.

Bulgarian	Hebrew	Japanese	Portuguese
Жълтата	הַי	いろはにほへど	à
дюля	כִּי	ちりぬるを	vo -
беше	סָהָם	わがよたれぞ	- cê
щастлива,	לְשִׁמּוֹעַ	つねならむ	uma
че	וְאֵין	うゐのおくや	can -
пухът,	תִּנְצָה	まけふこえて	- ção
който	קָרַפַּד	あさきゆめみじ	le -
цъфна,	עֵץ	糸ひもせず	- gal

Vorlage für Vokalensemble mit dem Gesangstext über und unter dem System

In diesem Beispiel werden die Texte mit den Befehlen `alignAboveContext` und `alignBelowContext` über und unter dem System angeordnet.

global = {

\key c \major

\time 4/4

}

sopMusic = \relative c' {

```

    c4 c c8[( b)] c4
}
sopWords = \lyricmode {
    hi hi hi hi
}

altoMusic = \relative c' {
    e4 f d e
}
altoWords = \lyricmode {
    ha ha ha ha
}

tenorMusic = \relative c' {
    g4 a f g
}
tenorWords = \lyricmode {
    hu hu hu hu
}

bassMusic = \relative c {
    c4 c g c
}
bassWords = \lyricmode {
    ho ho ho ho
}

\score {
  \new ChoirStaff <<
    \new Staff = "women" <<
      \new Voice = "sopranos" { \voiceOne << \global \sopMusic >> }
      \new Voice = "altos" { \voiceTwo << \global \altoMusic >> }
    >>
    \new Lyrics \with { alignAboveContext = "women" }
      \lyricsto "sopranos" \sopWords
    \new Lyrics \with { alignBelowContext = "women" }
      \lyricsto "altos" \altoWords
    % we could remove the line about this with the line below, since
    % we want the alto lyrics to be below the alto Voice anyway.
    % \new Lyrics \lyricsto "altos" \altoWords

    \new Staff = "men" <<
      \clef bass
      \new Voice = "tenors" { \voiceOne << \global \tenorMusic >> }
      \new Voice = "basses" { \voiceTwo << \global \bassMusic >> }
    >>
    \new Lyrics \with { alignAboveContext = "men" }
      \lyricsto "tenors" \tenorWords
    \new Lyrics \with { alignBelowContext = "men" }
      \lyricsto "basses" \bassWords
    % again, we could replace the line above this with the line below.
    % \new Lyrics \lyricsto "basses" \bassWords
  }

```

```
>>
}
```



Volta text markup using repeatCommands

Though voltes are best specified using `\repeat volta`, the context property `repeatCommands` must be used in cases where the volta text needs more advanced formatting with `\markup`.

Since `repeatCommands` takes a list, the simplest method of including markup is to use an identifier for the text and embed it in the command list using the Scheme syntax `#'((volta ,textIdentifier) ...)` (note the use of the backtick after `#` and the comma before `textIdentifier`). Start- and end-repeat commands can be added as separate list elements:

```
voltaAdLib = \markup { \volta-number { 1. 2. 3... } \italic { ad lib. } }
```

```
\relative c' ' {
  c1
  \set Score.repeatCommands = #'((volta ,voltaAdLib) start-repeat)
  c4 b d e
  \set Score.repeatCommands = #'((volta #f) (volta "4.") end-repeat)
  f1
  \set Score.repeatCommands = #'((volta #f))
}
```



9 Vocal music

See also Abschnitt “Vocal music” in *Notationsreferenz*.

Ambitus pro Stimme hinzufügen

Ambitus können pro Stimme gesetzt werden. In diesem Fall müssen sie manuell verschoben werden, um Zusammenstöße zu verhindern.

```
\new Staff <<
  \new Voice \with {
    \consists "Ambitus_engraver"
  } \relative c' {
    \override Ambitus.X-offset = 2.0
    \voiceOne
    c4 a d e
    f1
  }
  \new Voice \with {
    \consists "Ambitus_engraver"
  } \relative c' {
    \voiceTwo
    es4 f g as
    b1
  }
}>>
```



Adding indicators to staves which get split after a break

This snippet defines the commands `\splitStaffBarLine`, `\convUpStaffBarLine`, and `\convDownStaffBarLine`. These add arrows at a bar line to denote that several voices sharing a staff will each continue on a staff of their own in the next system, or that voices split in this way recombine.

Note that the implementation in this snippet draws dimensionless arrows into the right margin. For normal printing, this doesn't cause problems. However, it is necessary to increase the bounding box horizontally if you render the code as an image to avoid cropping, as demonstrated below.

```
#(define-markup-command (arrow-at-angle layout props angle-deg length fill)
  (number? number? boolean?)
  (let* ((PI-OVER-180 (/ (atan 1 1) 34))
        (degrees->radians (lambda (degrees) (* degrees PI-OVER-180)))
        (angle-rad (degrees->radians angle-deg))
        (target-x (* length (cos angle-rad)))
        (target-y (* length (sin angle-rad))))
    (interpret-markup layout props
      (markup
        #:translate (cons (/ target-x 2) (/ target-y 2))
        #:rotate angle-deg
        #:translate (cons (/ length -2) 0))
```

```

      #:concat (:#draw-line (cons length 0)
        #:arrow-head X RIGHT fill))))))

splitStaffBarLineMarkup = \markup \with-dimensions #'(0 . 0) #'(0 . 0) {
  \combine
  \arrow-at-angle #45 #(sqrt 8) ##t
  \arrow-at-angle #-45 #(sqrt 8) ##t
}

splitStaffBarLine = {
  \once \override Staff.BarLine.stencil =
  #(\lambda (grob)
    (ly:stencil-combine-at-edge
      (ly:bar-line::print grob)
      X RIGHT
      (grob-interpret-markup grob splitStaffBarLineMarkup)
      0))
  \break
}

convDownStaffBarLine = {
  \once \override Staff.BarLine.stencil =
  #(\lambda (grob)
    (ly:stencil-combine-at-edge
      (ly:bar-line::print grob)
      X RIGHT
      (grob-interpret-markup grob #{
        \markup\with-dimensions #'(0 . 0) #'(0 . 0) {
          \translate #'(0 . -.13)\arrow-at-angle #-45 #(sqrt 8) ##t
        }#}))
    0))
  \break
}

convUpStaffBarLine = {
  \once \override Staff.BarLine.stencil =
  #(\lambda (grob)
    (ly:stencil-combine-at-edge
      (ly:bar-line::print grob)
      X RIGHT
      (grob-interpret-markup grob #{
        \markup\with-dimensions #'(0 . 0) #'(0 . 0) {
          \translate #'(0 . .14)\arrow-at-angle #45 #(sqrt 8) ##t
        }#}))
    0))
  \break
}

\paper {
  indent = 10\mm
  short-indent = 10\mm
  line-width = 8\cm
}

```

```

}

separateSopranos = {
  \set Staff.instrumentName = "AI AII"
  \set Staff.shortInstrumentName = "AI AII"
  \splitStaffBarLine
  \change Staff = "up"
}

convSopranos = {
  \convDownStaffBarLine
  \change Staff = "shared"
  \set Staff.instrumentName = "S A"
  \set Staff.shortInstrumentName = "S A"
}

sI = {
  \voiceOne
  \repeat unfold 4 f''2
  \separateSopranos
  \repeat unfold 4 g''2
  \convSopranos
  \repeat unfold 4 c''2
}

sII = {
  s1*2
  \voiceTwo
  \change Staff = "up"
  \repeat unfold 4 d''2
}

aI = {
  \voiceTwo
  \repeat unfold 4 a'2
  \voiceOne
  \repeat unfold 4 b'2
  \convUpStaffBarLine
  \voiceTwo
  \repeat unfold 4 g'2
}

aII = {
  s1*2
  \voiceTwo
  \repeat unfold 4 g'2
}

ten = {
  \voiceOne
  \repeat unfold 4 c'2
  \repeat unfold 4 d'2
  \repeat unfold 4 c'2
}

bas = {
  \voiceTwo
  \repeat unfold 4 f2

```

```

\repeat unfold 4 g2
\repeat unfold 4 c2
}

\markup \pad-x #3 % avoid cropping
\score {
  <<
    \new ChoirStaff <<
      \new Staff = up \with {
        instrumentName = "SI SII"
        shortInstrumentName = "SI SII"
      } {
        s1*4
      }

      \new Staff = shared \with {
        instrumentName = "S A"
        shortInstrumentName = "S A"
      } <<
        \new Voice = sopI \sI
        \new Voice = sopII \sII
        \new Voice = altI \aI
        \new Voice = altII \aII
      >>
      \new Lyrics \with {
        alignBelowContext = up
      }
      \lyricsto sopII { e f g h }
      \new Lyrics \lyricsto altI { a b c d e f g h i j k l }

      \new Staff = men \with {
        instrumentName = "T B"
        shortInstrumentName = "T B"
      } <<
        \clef F
        \new Voice = ten \ten
        \new Voice = bas \bas
      >>
      \new Lyrics \lyricsto bas { a b c d e f g h i j k l }
    >>
  >>

  \layout {
    \context {
      \Staff \RemoveEmptyStaves
      \override VerticalAxisGroup.remove-first = ##t
    }
  }
}

```

The image displays three systems of musical notation. Each system consists of two staves: a vocal staff and a piano accompaniment staff. The vocal staves are labeled S A, SI SII, and AI AII, and the piano staves are labeled T B. The notes are labeled with letters a, b, c, d, e, f, g, h, i, j, k, l. The first system shows a vocal line with notes a, b, c, d and a piano accompaniment with notes a, b, c, d. The second system shows a vocal line with notes e, f, g, h and a piano accompaniment with notes e, f, g, h. The third system shows a vocal line with notes i, j, k, l and a piano accompaniment with notes i, j, k, l. Arrows indicate the continuation of the vocal lines.

Orchesterstichnoten zu einem Klavierauszug hinzufügen

Hier wird gezeigt, wie man das Hinzufügen von vielen Orchesterstichnoten zu einem Klavierauszug hinzufügen kann. Die musikalische Funktion `\cueWhile` braucht vier Argumente: Die Noten, von denen die Stichnoten formatiert werden sollen, definiert durch `\addQuote`, die Bezeichnung, die mit den Noten angegeben werden soll, dann entweder `#UP` (hoch) oder `#DOWN` (runter) zur Angabe von entweder `\voiceOne` mit der Bezeichnung über dem System oder `\voiceTwo` mit der Bezeichnung unter dem System, und schließlich die Klaviermusik, die parallel zu den Stichnoten gespielt werden soll. Die Bezeichnung des Stichnoteninstruments wird links der Stichnoten positioniert. Viele Abschnitte können zitiert werden, aber sie dürfen sich nicht überlappen.

```
cueWhile =
#(define-music-function
  (instrument name dir music)
  (string? string? ly:dir? ly:music?)
  #{
    \cueDuring $instrument #dir {
      \once \override TextScript.self-alignment-X = #RIGHT
```



```

        \once \override TextScript.direction = $dir
        <>-\markup { \tiny #name }
        $music
    }
    #})

flute = \relative c'' {
    \transposition c'
    s4 s4 e g
}
\addQuote "flute" { \flute }

clarinet = \relative c' {
    \transposition bes
    fis4 d d c
}
\addQuote "clarinet" { \clarinet }

singer = \relative c'' { c4. g8 g4 bes4 }
words = \lyricmode { here's the lyr -- ics }

pianoRH = \relative c'' {
    \transposition c'
    \cueWhile "clarinet" "Clar." #DOWN { c4. g8 }
    \cueWhile "flute" "Flute" #UP { g4 bes4 }
}
pianoLH = \relative c { c4 <c' e> e, <g c> }

\score {
    <<
        \new Staff {
            \new Voice = "singer" {
                \singer
            }
        }
        \new Lyrics {
            \lyricsto "singer"
            \words
        }
        \new PianoStaff <<
            \new Staff {
                \new Voice {
                    \pianoRH
                }
            }
            \new Staff {
                \clef "bass"
                \pianoLH
            }
        >>
    >>
}

```



Adjusting vertical spacing of lyrics

This snippet shows how to bring the lyrics line closer to the staff.

```
music = \relative c' { c4 d e f | g4 f e d | c1 }
text = \lyricmode { aa aa aa aa aa aa aa aa }
```

```
<<
\new Staff \new Voice = melody \music
% Default layout:
\new Lyrics \lyricsto melody \text

\new Staff \new Voice = melody \music
% Reducing the minimum space below the staff and above the lyrics.
\new Lyrics \with {
  \override VerticalAxisGroup.nonstaff-relatedstaff-spacing =
    #'((basic-distance . 1))
} \lyricsto melody \text
>>
```



Aligning syllables with melisma

By default, lyrics syllables that start a melisma are left-aligned on their note. The alignment can be altered using the `lyricMelismaAlignment` property.

```
<<
\new Staff {
  \new Voice = "vocal" \relative c' {
    \override TextScript.staff-padding = #2
    c d~\markup default d e
    c d~\markup "right aligned" d e
    c d~\markup "center aligned" d e
    c d~\markup "reset to default" d e
  }
}
```

```

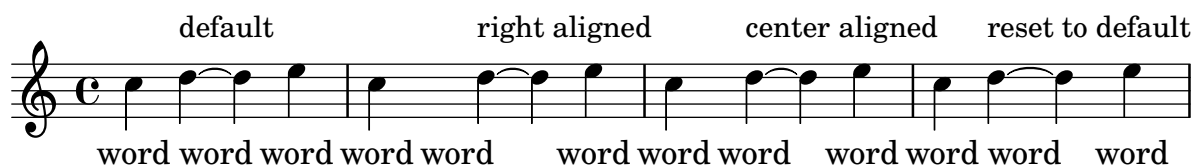
\new Lyrics \lyricsto "vocal" {
  word word word
  \set lyricMelismaAlignment = #RIGHT
  word word word
  \set lyricMelismaAlignment = #CENTER
  word word word
  \unset lyricMelismaAlignment
  word word word
}
>>

```

```

\layout {
  ragged-right = ##f
}

```



Ambitus

Ambitus indicate pitch ranges for voices.

Accidentals only show up if they are not part of the key signature. AmbitusNoteHead grobs also have ledger lines.

```

\layout {
  \context {
    \Voice
    \consists "Ambitus_engraver"
  }
}

```

```

<<
\new Staff {
  \relative c' {
    \time 2/4
    c4 f'
  }
}
\new Staff {
  \relative c' {
    \time 2/4
    \key d \major
    cis4 as'
  }
}
>>

```



Ambitus after key signature

By default, ambitus are positioned at the left of the clef. The `\ambitusAfter` function allows for changing this placement. Syntax is `\ambitusAfter grob-interface`; see Graphical Object Interfaces (<https://lilypond.org/doc/v2.24/Documentation/internals/graphical-object-interfaces>) for a list of possible values for *grob-interface*.

A common use case is printing the ambitus between key signature and time signature.

```
\new Staff \with {
  \consists Ambitus_engraver
} \relative {
  \ambitusAfter key-signature
  \key d \major
  es'8 g bes cis d2
}
```



Ambitus mit vielen Stimmen

Indem man den `Ambitus_engraver` im `Staff`-Kontext hinzufügt, erhält man einen einzigen Ambitus pro System, auch in dem Fall, dass mehrere Stimmen sich im gleichen System befinden.

```
\new Staff \with {
  \consists "Ambitus_engraver"
}
<<
  \new Voice \relative c'' {
    \voiceOne
    c4 a d e
    f1
  }
  \new Voice \relative c' {
    \voiceTwo
    es4 f g as
    b1
  }
}>>
```



Ancient notation template – modern transcription of Gregorian music

This example demonstrates how to do modern transcription of Gregorian music. Gregorian music has no measure, no stems; it uses only half and quarter note heads, and special marks, indicating rests of different length.

```
chant = \relative c' {
  \set Score.timing = ##f
  f4 a2 \divisioMinima
  g4 b a2 f2 \divisioMaior
  g4( f) f( g f) a2 \finalis \break
  f4 a2 \divisioMinima
  g4 b a2 f2 \divisioMaior
  g4( f) f( g a) g2( f) \finalis
}

verba = \lyricmode {
  Lo -- rem ip -- sum do -- lor sit a -- met,
  lo -- rem ip -- sum do -- lor sit a -- met.
}

\score {
  \new GregorianTranscriptionStaff <<
    \new GregorianTranscriptionVoice = "melody" \chant
    \new GregorianTranscriptionLyrics = "one" \lyricsto melody \verba
  >>
}
```



Vorlage für anglikanischen Psalm

Diese Vorlage zeigt eine Art, anglikanische Psalmengesänge zu setzen. Hier wird auch gezeigt, wie Strophen als einfacher Text unter den Noten hinzugefügt werden können. Zwei Strophen sind in unterschiedlicher Weise notiert um mehr Möglichkeiten darzustellen.

```
SopranoMusic = \relative g' {
  g1 | c2 b | a1 | \bar "||"
  a1 | d2 c | c b | c1 | \bar "||"
}

AltoMusic = \relative c' {
  e1 | g2 g | f1 |
  f1 | f2 e | d d | e1 |
}

TenorMusic = \relative a {
```

```

    c1 | c2 c | c1 |
    d1 | g,2 g | g g | g1 |
}

BassMusic = \relative c {
    c1 | e2 e | f1 |
    d1 | b2 c | g' g | c,1 |
}

global = {
    \time 2/2
}

dot = \markup {
    \raise #0.7 \musicglyph "dots.dot"
}

tick = \markup {
    \raise #1 \fontsize #-5 \musicglyph "scripts.rvarcomma"
}

% Use markup to center the chant on the page
\markup \fill-line {
    \score { % centered
        \new ChoirStaff <<
            \new Staff <<
                \global
                \clef "treble"
                \new Voice = "Soprano" <<
                    \voiceOne
                    \SopranoMusic
                >>
                \new Voice = "Alto" <<
                    \voiceTwo
                    \AltoMusic
                >>
            >>
        >>

        \new Staff <<
            \clef "bass"
            \global
            \new Voice = "Tenor" <<
                \voiceOne
                \TenorMusic
            >>
            \new Voice = "Bass" <<
                \voiceTwo
                \BassMusic
            >>
        >>
    >>
}

```

```

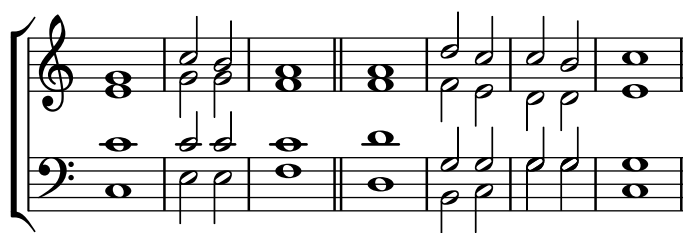
\layout {
  \context {
    \Score
    \override SpacingSpanner.base-shortest-duration =
      \musicLength 2
  }
  \context {
    \Staff
    \remove "Time_signature_engraver"
  }
}
} % End score
} % End markup

\markup \fill-line {
  \left-column {
    \null \null \null
    \line { \fontsize #5 0
      \fontsize #3 come
      let us \bold sing | unto \dot the | Lord : let }
    \line { us heartily \concat { re \bold joice }
      in the | strength of | our }
    \line { sal | vation. }

    \null

    \line { \hspace #2.5 8. Today if ye will hear his voice * }
    \line { \concat { \bold hard en }
      \tick not your \tick hearts : as in the pro- }
    \line { vocation * and as in the \bold day of tempt- \tick }
    \line { -ation \tick in the \tick wilderness. }
  }
}

```



O come let us **sing** | unto • the | Lord : let
us heartily **rejoice** in the | strength of | our
sal | vation.

8. Today if ye will hear his voice *
harden ' not your ' hearts : as in the pro-
vocation * and as in the **day** of tempt- '
-ation ' in the ' wilderness.

Arranging separate lyrics on a single line

Sometimes you may want to put lyrics for different performers on a single line: where there is rapidly alternating text, for example. This snippet shows how it can be done with adjusting the `nonstaff-nonstaff-spacing` property of the `VerticalAxisGroup` grob.

```
\layout {
  \context {
    \Lyrics
    \override VerticalAxisGroup
      .nonstaff-nonstaff-spacing
      .minimum-distance = ##f
  }
}

aliceSings = \markup { \smallCaps "Alice" }
eveSings = \markup { \smallCaps "Eve" }

<<
\new Staff <<
  \new Voice = "alice" {
    f'4^\aliceSings g' r2 |
    s1 |
    f'4^\aliceSings g' r2 |
    s1 | \break
    % ...

    \voiceOne
    s2 a'8^\aliceSings a' b'4 |
    \oneVoice
    g'1
  }
  \new Voice = "eve" {
    s1 |
    a'2^\eveSings g' |
    s1 |
    a'2^\eveSings g'
    % ...

    \voiceTwo
    f'4^\eveSings a'8 g' f'4 e' |
    \oneVoice
    s1
  }
}
>>

\new Lyrics \lyricsto "alice" {
  may -- be
  sec -- ond
  % ...
  Shut up, you fool!
}
```



```

\new Lyrics \lyricsto "eve" {
  that the
  words are
  % ...
  ...and then I was like--
}
>>

```

ALICE EVE ALICE EVE

may - be that the sec - ond words are

5 EVE ALICE

...and then I Shut up, you fool!

was like-

Changing stanza fonts

Fonts can be changed independently for each stanza, including the font used for printing the stanza number.

```

%{
You may have to install additional fonts.

Red Hat Fedora: dejavu-fonts-all

Debian GNU/Linux, Ubuntu: fonts-dejavu-core
                           fonts-dejavu-extra
}%

\relative c' ' {
  \time 3/4
  g2 e4
  a2 f4
  g2.
}
\addlyrics {
  \set stanza = #"1. "
  Hi, my name is Bert.
}
\addlyrics {
  \override StanzaNumber.fonts.serif = "DejaVu Sans"
  \set stanza = #"2. "
  \override LyricText.font-family = #'typewriter
  Oh, ché -- ri, je t'aime
}

```



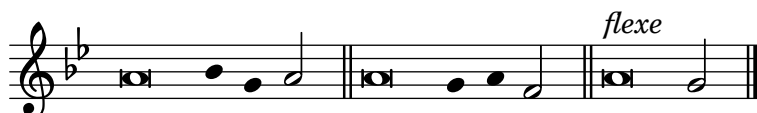
1. Hi, my name is Bert.
2. Oh, ché-ri, jet'aime

Chant or psalm notation

This form of notation is used for psalm chant, where verses are not always of the same length.

```
stemOff = \hide Staff.Stem
stemOn  = \undo \stemOff

\score {
  \new Staff \with { \remove "Time_signature_engraver" }
  {
    \key g \minor
    \cadenzaOn
    \stemOff a'\breve bes'4 g'4
    \stemOn a'2 \section
    \stemOff a'\breve g'4 a'4
    \stemOn f'2 \section
    \stemOff a'\breve^{\markup { \italic flexe }}
    \stemOn g'2 \fine
  }
}
```



Forcing hyphens to be shown

If LilyPond does not think there is space for a hyphen, it will be omitted. This behaviour can be overridden with the minimum-distance property of LyricHyphen.

```
\relative c'' {
  c32 c c c
  c32 c c c
  c32 c c c
  c32 c c c
}

\addlyrics {
  syl -- lab word word
  \override LyricHyphen.minimum-distance = #1.0
  syl -- lab word word
  \override LyricHyphen.minimum-distance = #2.0
  syl -- lab word word
  \revert LyricHyphen.minimum-distance
  syl -- lab word word
}
```

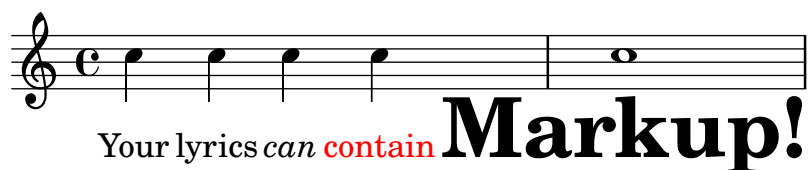


Silben im Gesangstext formatieren

Textbeschriftungsmodus kann eingesetzt werden, um individuelle Silben im Gesangstext zu formatieren.

```
mel = \relative c'' { c4 c c c c1 }
lyr = \lyricmode {
  Your lyrics \markup { \italic can }
  \markup { \with-color #red contain }
  \markup { \fontsize #8 \bold Markup! }
}
```

```
<<
  \new Voice = "melody" \mel
  \new Lyrics \lyricsto "melody" \lyr
>>
```



How to put ties between syllables in lyrics

This can be achieved by separating those syllables by tildes.

```
\lyrics {
  wa~o~a
}
```

wa o a

Hymnus-Vorlage

Dieses Beispiel zeigt eine Möglichkeit, eine Hymnusmelodie zu setzen, in der jede Zeile mit einem Auftakt beginnt und einem unvollständigen Takt abschließt. Es zeigt auch, wie man die Strophen als allein stehenden Text unter die Noten hinzufügt.

```
Timeline = {
  \time 4/4
  \tempo 4=96
  \partial 2
  s2 | s1 | s2 \breathe s2 | s1 | s2 \caesura \break
  s2 | s1 | s2 \breathe s2 | s1 | s2 \fine
}
```

```
SopranoMusic = \relative g' {
  g4 g | g g g g | g g g g | g g g g | g2
  g4 g | g g g g | g g g g | g g g g | g2
}
```

```

AltoMusic = \relative c' {
  d4 d | d d d d | d d d d | d d d d | d2
  d4 d | d d d d | d d d d | d d d d | d2
}

TenorMusic = \relative a {
  b4 b | b b b b | b b b b | b b b b | b2
  b4 b | b b b b | b b b b | b b b b | b2
}

BassMusic = \relative g {
  g4 g | g g g g | g g g g | g g g g | g2
  g4 g | g g g g | g g g g | g g g g | g2
}

global = {
  \key g \major
}

\score { % Start score
  \new PianoStaff << % Start pianostaff
    \new Staff << % Start Staff = RH
      \global
      \clef "treble"
      \new Voice = "Soprano" << % Start Voice = "Soprano"
        \Timeline
        \voiceOne
        \SopranoMusic
      >> % End Voice = "Soprano"
      \new Voice = "Alto" << % Start Voice = "Alto"
        \Timeline
        \voiceTwo
        \AltoMusic
      >> % End Voice = "Alto"
    >> % End Staff = RH

  \new Staff << % Start Staff = LH
    \global
    \clef "bass"
    \new Voice = "Tenor" << % Start Voice = "Tenor"
      \Timeline
      \voiceOne
      \TenorMusic
    >> % End Voice = "Tenor"
    \new Voice = "Bass" << % Start Voice = "Bass"
      \Timeline
      \voiceTwo
      \BassMusic
    >> % End Voice = "Bass"
  >> % End Staff = LH
>> % End pianostaff

```

```

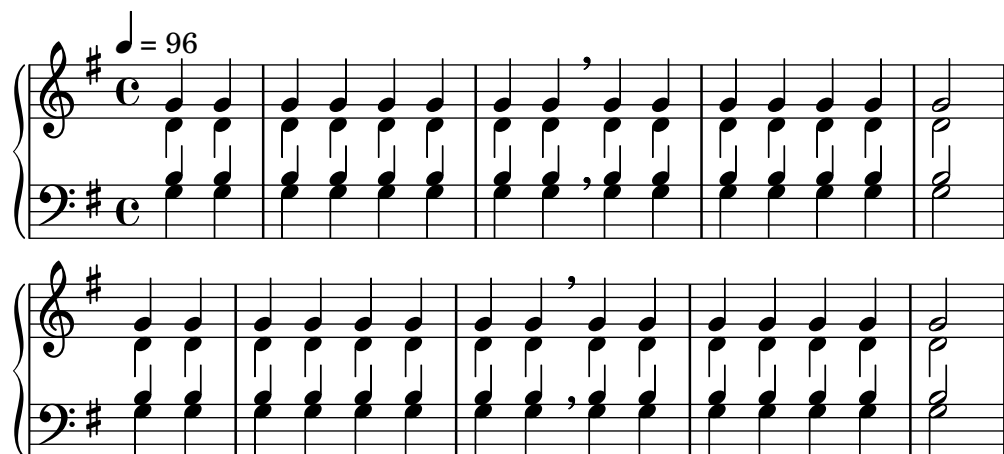
} % End score

\markup \fill-line {
  \left-column {
    "This is line one of the first verse"
    "This is line two of the same"
    \null
    "And here's line one of the second verse"
    "And the next line of the same"
  }
}

\layout {
  \context {
    \Score
    caesuraType = #'((bar-line . "||"))
    fineBarType = "||"
  }
}

\paper { % Start paper block
  indent = 0 % don't indent first system
  line-width = 130 % shorten line length to suit music
  tagline = ##f % Don't print tag line, can be removed
} % End paper block

```



This is line one of the first verse
 This is line two of the same

And here's line one of the second verse
 And the next line of the same

Ausrichtung von Gesangstext

Die horizontale Ausrichtung von Gesangstext kann eingestellt werden, indem man die self-alignment-X-Eigenschaft des LyricText-Objekts verändert. #-1 bedeutet links, #0 bedeutet mittig und #1 bedeutet rechts, man kann aber genauso gut auch #LEFT, #CENTER und #RIGHT benutzen.

```
\layout {
```

```

    ragged-right = ##f
}

\relative c'' {
  c1 c c c
}

\addlyrics {
  \once \override LyricText.self-alignment-X = #LEFT
  "left-aligned"
  \once \override LyricText.self-alignment-X = #CENTER
  "centered"
  \once \override LyricText.self-alignment-X = 1
  "right-aligned"
  \once \override LyricText.self-alignment-X = #-1.5
  "very right"
}

```



Marking notes of spoken parts with a cross on the stem (Sprechstimme)

This example shows how to put crosses on stems. Mark the beginning of a spoken section with the command `\speakOn` and end it with `\speakOff`.

```

speakOn = \override Stem.stencil =
  #(lambda (grob)
    (let* ((x-parent (ly:grob-parent grob X))
      (is-rest? (ly:grob? (ly:grob-object x-parent 'rest))))
      (if is-rest?
        empty-stencil
        (ly:stencil-combine-at-edge
          (ly:stem::print grob)
          Y
          (- (ly:grob-property grob 'direction))
          (grob-interpret-markup
            grob
            (markup #:center-align #:fontsize -4
              #:musicglyph "noteheads.s2cross"))
          -1.7))))))

speakOff = \revert Stem.stencil

\new Staff {
  \relative c'' {
    a4 b a c
    \speakOn
    g4 f r g8 a
    b4 r r8 d e4
  }
}

```

```

\speakingOff
c4 a g f
}
}

```



Orchester, Chor und Klavier

Diese Vorlage zeigt die Benutzung von geschachtelten `StaffGroup`- und `GrandStaff`-Kontexte, um Instrumente in Untergruppen zu unterteilen, und die Benutzung von `\transpose` für transponierende Instrumente. Alle Noten werden in C geschrieben. Noten können in C eingegeben werden, oder auch in der Tonart des Instrumentes: dann müssen sie zuerst nach C transponiert werden, bevor sie einer Variable zugewiesen werden.

```

#(set-global-staff-size 17)

```

```

\paper {
  indent = 3.0\cm % add space for instrumentName
  short-indent = 1.5\cm % add less space for shortInstrumentName
}

```

```

fluteMusic = \relative c' { \key g \major g'1 b }

```

```

% Pitches as written on a manuscript for Clarinet in A
% are transposed to concert pitch.

```

```

clarinetMusic = \transpose c' a
  \relative c'' { \key bes \major bes1 d }

```

```

trumpetMusic = \relative c { \key g \major g'1 b }

```

```

% Key signature is often omitted for horns

```

```

hornMusic = \transpose c' f
  \relative c { d'1 fis }

```

```

percussionMusic = \relative c { \key g \major g1 b }

```

```

sopranoMusic = \relative c'' { \key g \major g'1 b }
sopranoLyrics = \lyricmode { Lyr -- ics }

```

```

altoIMusic = \relative c' { \key g \major g'1 b }
altoILyrics = \sopranoLyrics
altoIIMusic = \relative c' { \key g \major g'1 b }
altoIILyrics = \lyricmode { Ah -- ah }

```

```

tenorMusic = \relative c' { \clef "treble_8" \key g \major g1 b }
tenorLyrics = \sopranoLyrics

```

```

pianoRHMus = \relative c { \key g \major g'1 b }
pianoLHMus = \relative c { \clef bass \key g \major g1 b }

```

```

violinIMusic = \relative c' { \key g \major g'1 b }
violinIIMusic = \relative c' { \key g \major g'1 b }

violaMusic = \relative c { \clef alto \key g \major g'1 b }

celloMusic = \relative c { \clef bass \key g \major g1 b }

bassMusic = \relative c { \clef "bass_8" \key g \major g,1 b }

\book {
  \score {
    <<
      \new StaffGroup = "StaffGroup_woodwinds" <<
        \new Staff = "Staff_flute" \with { instrumentName = "Flute" }
          \fluteMusic

        \new Staff = "Staff_clarinet" \with {
          instrumentName = \markup { \concat { "Clarinet in B" \flat } }
        }
        % Declare that written Middle C in the music
        % to follow sounds a concert B flat, for
        % output using sounded pitches such as MIDI.
        %\transposition bes

        % Print music for a B-flat clarinet
        \transpose bes c' \clarinetMusic
      >>

      \new StaffGroup = "StaffGroup_brass" <<
        \new Staff = "Staff_hornI" \with {
          instrumentName = "Horn in F"
        }
        % \transposition f
        \transpose f c' \hornMusic

        \new Staff = "Staff_trumpet" \with {
          instrumentName = "Trumpet in C"
        }
        \trumpetMusic
      >>

      \new RhythmicStaff = "RhythmicStaff_percussion" \with {
        instrumentName = "Percussion"
      }
      \percussionMusic

      \new PianoStaff \with {
        instrumentName = "Piano"
      } <<
        \new Staff { \pianoRHMusik }
        \new Staff { \pianoLHMusik }
      >>
    }
  }

```



```

\new ChoirStaff = "ChoirStaff_choir" <<
  \new Staff = "Staff_soprano" \with {
    instrumentName = "Soprano"
  }
  \new Voice = "soprano" \sopranoMusic
  \new Lyrics \lyricsto "soprano" { \sopranoLyrics }

\new GrandStaff = "GrandStaff_altos" \with {
  \accepts Lyrics
} <<
  \new Staff = "Staff_altoI" \with {
    instrumentName = "Alto I"
  }
  \new Voice = "altoI"
  \altoIMusic
  \new Lyrics \lyricsto "altoI" { \altoILyrics }
  \new Staff = "Staff_altoII" \with {
    instrumentName = "Alto II"
  }
  \new Voice = "altoII"
  \altoIIMusic
  \new Lyrics \lyricsto "altoII" { \altoIILyrics }
>>

\new Staff = "Staff_tenor" \with {
  instrumentName = "Tenor"
}
\new Voice = "tenor" \tenorMusic
\new Lyrics \lyricsto "tenor" { \tenorLyrics }
>>

\new StaffGroup = "StaffGroup_strings" <<
  \new GrandStaff = "GrandStaff_violins" <<
    \new Staff = "Staff_violinI" \with {
      instrumentName = "Violin I"
    }
    \violinIMusic
    \new Staff = "Staff_violinII" \with {
      instrumentName = "Violin II"
    }
    \violinIIMusic
  >>
>>

\new Staff = "Staff_viola" \with {
  instrumentName = "Viola"
}
\violaMusic

\new Staff = "Staff_cello" \with {
  instrumentName = "Cello"
}

```

```
        \celloMusic

    \new Staff = "Staff_bass" \with {
        instrumentName = "Double Bass"
    }
    \bassMusic
>>
>>
}
}
```

Flute

Clarinet in B \flat

Horn in F

Trumpet in C

Percussion

Piano

Soprano

Alto I

Alto II

Tenor

Violin I

Violin II

Viola

Cello

Double Bass

Lyr - ics

Lyr - ics

Ah - ah

Lyr - ics

8

8

Detailed description: This is a page from a musical score, page 268, titled 'Kapitel 9: Vocal music'. It shows measures 8 and 9 of a piece. The score is for a large ensemble. The instruments listed on the left are: Flute, Clarinet in B-flat, Horn in F, Trumpet in C, Percussion, Piano, Soprano, Alto I, Alto II, Tenor, Violin I, Violin II, Viola, Cello, and Double Bass. The key signature is one sharp (F#), and the time signature is common time (C). The music is written in a grand staff format with multiple staves. The vocal parts (Soprano, Alto I, Alto II, Tenor) have lyrics written below them. The instrumental parts (Flute, Clarinet, Horn, Trumpet, Piano, Violins, Viola, Cello, Double Bass) are written in their respective staves. The percussion part is indicated by a single note on a line. The page number 268 is in the top right corner. The chapter title 'Kapitel 9: Vocal music' is in the top left corner.

Vorlage für Klavier und Gesangsstimme

Das nächste Beispiel ist typisch für ein Lied: Im oberen System die Melodie mit Text, darunter Klavierbegleitung.

```
melody = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

  a b c d
}

text = \lyricmode {
  Aaa Bee Cee Dee
}

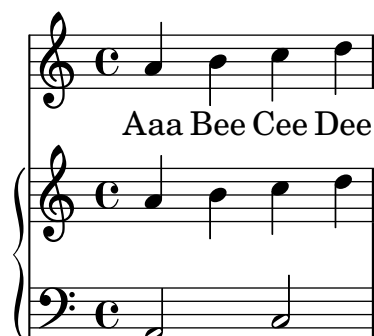
upper = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

  a4 b c d
}

lower = \relative c {
  \clef bass
  \key c \major
  \time 4/4

  a2 c
}

\score {
  <<
    \new Voice = "mel" { \autoBeamOff \melody }
    \new Lyrics \lyricsto mel \text
    \new PianoStaff <<
      \new Staff = "upper" \upper
      \new Staff = "lower" \lower
    >>
  >>
  \layout {
    \context { \Staff \RemoveEmptyStaves }
  }
  \midi { }
}
```



Putting lyrics inside the staff

Lyrics can be moved vertically to place them inside the staff. The lyrics are moved with `\override LyricText.extra-offset = #'(0 . dy)`, and there are similar commands to move the extenders and hyphens. A good value for *dy* must be found by trial and error.

```
<<
\new Staff <<
  \new Voice = "voc" \relative c' { \stemDown a bes c8 b c4 }
>>
\new Lyrics \with {
  \override LyricText.extra-offset = #'(0 . 8.6)
  \override LyricExtender.extra-offset = #'(0 . 8.6)
  \override LyricHyphen.extra-offset = #'(0 . 8.6)
} \lyricsto "voc" { La la -- la __ _ la }
>>
```



SATB-Chorvorlage auf vier Systemen

SATB-Chorvorlage auf vier Systemen

```
global = {
  \key c \major
  \time 4/4
  \dynamicUp
}
sopranonotes = \relative c'' {
  c2 \p \< d c d \f
}
sopranowords = \lyricmode { do do do do }
altonotes = \relative c'' {
  c2\p d c d
}
altowords = \lyricmode { re re re re }
tenornotes = {
  \clef "G_8"
  c2\mp d c d
}
```

```

tenorwords = \lyricmode { mi mi mi mi }
bassnotes = {
  \clef bass
  c2\mf d c d
}
basswords = \lyricmode { mi mi mi mi }

\score {
  \new ChoirStaff <<
    \new Staff <<
      \new Voice = "soprano" <<
        \global
        \sopranonotes
      >>
      \new Lyrics \lyricsto "soprano" \sopranowords
    >>
    \new Staff <<
      \new Voice = "alto" <<
        \global
        \altonotes
      >>
      \new Lyrics \lyricsto "alto" \altowords
    >>
    \new Staff <<
      \new Voice = "tenor" <<
        \global
        \tenornotes
      >>
      \new Lyrics \lyricsto "tenor" \tenorwords
    >>
    \new Staff <<
      \new Voice = "bass" <<
        \global
        \bassnotes
      >>
      \new Lyrics \lyricsto "bass" \basswords
    >>
  >>
}

```

p *f*
 do do do do
p
 re re re re
mp
 mi mi mi mi
mf
 mi mi mi mi

Vorlage für eine Notenzeile mit Noten, Text und Akkorden

Mit diesem Beispiel können Sie einen Song mit Melodie, Text und Akkorden schreiben.

```

melody = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

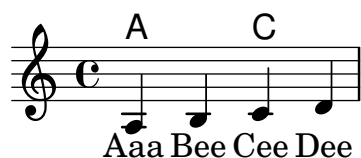
  a4 b c d
}

text = \lyricmode {
  Aaa Bee Cee Dee
}

harmonies = \chordmode {
  a2 c
}

\score {
  <<
    \new ChordNames {
      \set chordChanges = ##t
      \harmonies
    }
    \new Voice = "one" { \autoBeamOff \melody }
    \new Lyrics \lyricsto "one" \text
  >>
  \layout { }
  \midi { }
}

```



Single-staff template with notes, lyrics, chords, and frets

Here is a simple lead sheet template with melody, lyrics, chords, and fret diagrams.

```

verseI = \lyricmode {
  \set stanza = #"1."
  This is the first verse
}

verseII = \lyricmode {
  \set stanza = #"2."
  This is the second verse.
}

theChords = \chordmode {
  % insert chords for chordnames and fretboards here
  c2 g4 c
}

staffMelody = \relative c' {
  \key c \major
  \clef treble
  % Type notes for melody here
  c4 d8 e f4 g
  \bar "|"
}

\score {
  <<
    \context ChordNames { \theChords }
    \context FretBoards { \theChords }
    \new Staff {
      \context Voice = "voiceMelody" { \staffMelody }
    }
    \new Lyrics = "lyricsI" {
      \lyricsto "voiceMelody" \verseI
    }
    \new Lyrics = "lyricsII" {
      \lyricsto "voiceMelody" \verseII
    }
  >>
  \layout { }
  \midi { }
}

```


1. This is the first verse
2. This is the second verse.

Vorlage für ein Notensystem mit Noten und Gesangstext

Das nächste Beispiel zeigt eine einfache Melodie mit Text. Kopieren Sie es in Ihre Datei, fügen Sie Noten und Text hinzu und übersetzen Sie es mit LilyPond. In dem Beispiel wird die automatische Balkenverbindung ausgeschaltet (mit dem Befehl `\autoBeamOff`), wie es für Vokalmusik üblich ist. Wenn Sie die Balken wieder einschalten wollen, müssen Sie die entsprechende Zeile entweder ändern oder auskommentieren.

```
melody = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

  a4 b c d
}

text = \lyricmode {
  Aaa Bee Cee Dee
}

\score{
  <<
    \new Voice = "one" {
      \autoBeamOff
      \melody
    }
    \new Lyrics \lyricsto "one" \text
  >>
  \layout { }
  \midi { }
}
```

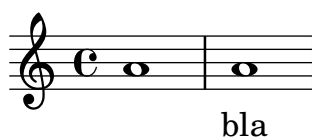
Aaa Bee Cee Dee

Skips in lyric mode

The 's' syntax for skips is only available in note mode and chord mode. In other situations, for example, when entering lyrics, using the `\skip` command is recommended.

```
<<
  \relative c'' { a1 | a }
  \new Lyrics \lyricmode { \skip1 bla1 }
```

>>



Skips in lyric mode (2)

Although 's' skips cannot be used in `\lyricmode` (it is taken to be a literal „s“, not a space), double quotes (") or underscores (_) are available.

<<

```
\relative c'' { a4 b c d }
\new Lyrics \lyricmode { a4 "" _ gap }
```

>>



Using a bracket to clarify divisi

The `\nonArpeggiato` command can be used to indicate the division of voices where there are no stems to provide the information. This is often seen in choral music.

```
\include "english.ly"
```

```
\score {
  \relative c'' {
    \key a \major
    \time 2/2
    <<
    \new Voice = "upper" <<
      {
        \voiceOne
        a2( b2
        <b d>1\nonArpeggiato)
        <cs e>\nonArpeggiato ~
        <cs e>4
        \fine
      }
    \addlyrics { \lyricmode { A -- men. } }
    >>
    \new Voice = "lower" {
      \voiceTwo
      a1 ~
      a
      a ~
      a4
      \fine
    }
  }
  >>
}
```

}



Using tags to produce mensural and modern music from the same source

Using tags it is possible to produce both mensural and modern notation from the same music. In this snippet, a function `\menrest` is introduced, allowing mensural rests to be pitched as in the original, but with modern rests in the standard staff position.

Tags can also be used where other differences are needed: for example using „whole measure rests“ (`R1`, `R\breve`, etc.) in modern music, but normal rests (`r1`, `r\breve`, etc.) in the mensural version. Converting mensural music to its modern equivalent is usually referred to as *transcription*.

The call `c4.\Be c8 c\Am` is the same as `c4.[c8 c]`. However, it suppresses warnings if it starts on a note that can't hold a beam but needs it anyway due to the use of `Completion_heads_engraver`.

[Note that the custos sticks out into the right margin and might be cut off if the LilyPond output gets cropped tightly. The use of `\with-true-dimensions` below avoids this.]

```
\layout {
  line-width = 150\mm
}

menrest = #(define-music-function (note) (ly:music?)
  #{
    \tag #'mens $(make-music 'RestEvent note)
    \tag #'mod $(make-music 'RestEvent note 'pitch '())
  #})

Be = \tag #'mod
  #(begin
    (ly:expect-warning (G_ "stem does not fit in beam"))
    (ly:expect-warning (G_ "beam was started here"))
    (make-span-event 'BeamEvent START))

Am = \tag #'mod ]

MenStyle = {
  \override Score.BarNumber.transparent = ##t
  \override Stem.neutral-direction = #up
  \omit Slur
  \omit Beam
}

finalis = \section

Music = \relative c'' {
  \key f \major
  g1 d'2 \menrest bes4 bes a2 \menrest r4 g4 fis4. fis8 fis4 fis \break
  g e f4.([ g8] a4[ g8 f] g2.\Be fis8 e\Am fis2) g\breve \finalis
}
```

```

}

MenLyr = \lyricmode {
  So farre, deere life, deare life,
  from thy bright beames ab- en- ted,
}
ModLyr = \lyricmode {
  So far, dear life, dear life,
  from your bright beams ab -- sen -- ted, --
}

\markup \with-true-dimensions % work around a cropping issue
\score {
  \keepWithTag #'mens {
    <<
      \new PetrucciStaff {
        \new PetrucciVoice = "Cantus" {
          \clef "petrucci-c1" \time 4/4 \MenStyle \Music
        }
      }
      \new Lyrics \lyricsto "Cantus" \MenLyr
    >>
  }
  \layout {
    \context {
      \PetrucciVoice
      % No longer necessary starting with version 2.25.23.
      \override Flag.style = #'mensural
    }
  }
}

\markup\vspace #1

\score {
  \keepWithTag #'mod {
    \new ChoirStaff <<
      \new Staff {
        \new Voice = "Sop" \with {
          \remove "Note_heads_engraver"
          \consists "Completion_heads_engraver"
          \remove "Rest_engraver"
          \consists "Completion_rest_engraver"
        } \shiftDurations 1 0 { \time 2/4 \autoBeamOff \Music }
      }
      \new Lyrics \lyricsto "Sop" \ModLyr
    >>
  }
}

```

So farre, deere life, deare life, from thy bright
beames ab-fen-ted,

So far, dear life, dear life, from your bright
beams ab - sen - - - - ted,_____

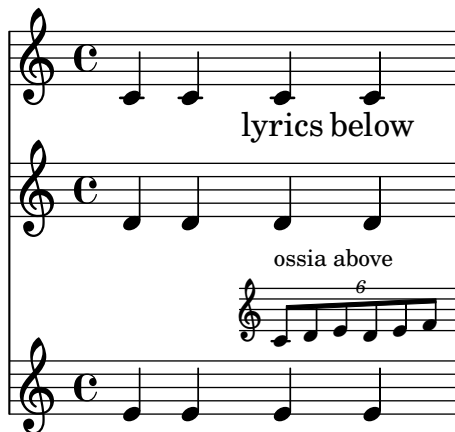
Gesangstext und Ossia vertikal ausrichten

Dieser Schnipsel zeigt, wie man die Kontexteigenschaften `alignBelowContext` und `alignAboveContext` benutzen kann, um die Positionierung von Gesangstext und Ossia-Abschnitten zu kontrollieren.

```
\relative c' <<
\new Staff = "1" { c4 c c c }
\new Staff = "2" { d4 d d d }
\new Staff = "3" { e4 e e e }

{ \skip 2
  <<
    \lyrics {
      \set alignBelowContext = "1"
      lyrics4 below
    }
    \new Staff \with {
      alignAboveContext = "3"
      fontSize = -2
      \override StaffSymbol.staff-space = #(magstep -2)
      \remove "Time_signature_engraver"
      \override VerticalAxisGroup.staff-staff-spacing =
        #'((minimum-distance . 0)
          (basic-distance . 0)
          (padding . 1))
    } {
      \tuplet 6/4 {
        \override TextScript.padding = 2
        c8["^"ossia above" d e d e f]
      }
    }
  }
```

```
>>
}
>>
```



Vertically aligning stanza numbers of different staves

It can happen that stanza numbers don't align vertically if the verses are attached to different staves. To fix that, override the self-alignment-X property of the LyricText grob.

```
\markup { default behavior }
```

```
<<
\new Staff { b b b b }
\lyrics {
  \set stanza = "3."
  a a a a
}
```

```
\new Staff { b b b b }
\lyrics {
  \set stanza = "1."
  aaaaaaaaaa a a a
}
\lyrics {
  \set stanza = "2."
  a a a a
}
```

```
>>
```

```
\markup \vspace #1
\markup {
  using \typewriter "self-alignment-X = #LEFT" }
```

```
<<
\new Staff { b b b b }
\new Lyrics \lyricmode {
  \set stanza = "3."
  a a a a
}
```

```

\new Staff { b b b b }
\new Lyrics \lyricmode {
  \set stanza = "1."
  \once \override LyricText.self-alignment-X = #LEFT
  aaaaaaaaaa a a a
}
\new Lyrics \lyricmode {
  \set stanza = "2."
  a a a a
}
>>

```

default behavior



using self-alignment-X = #LEFT



Vertically centered common lyrics

In a vocal piece where there are several (two, four or more) lines of lyrics and common lyrics for all voices at some point, the common lyrics may be made to appear vertically centered, as shown in the following example:

```

dropLyrics = {
  \override LyricText.extra-offset = #'(0 . -4.5)
  \override LyricHyphen.extra-offset = #'(0 . -4.5)
  \override LyricExtender.extra-offset = #'(0 . -4.5)
  \override StanzaNumber.extra-offset = #'(0 . -4.5)
}

```

```

raiseLyrics = {
  \revert LyricText.extra-offset
  \revert LyricHyphen.extra-offset
  \revert LyricExtender.extra-offset
}

```

```

\revert StanzaNumber.extra-offset
}

skipFour = \repeat unfold 4 { \skip 8 }

lyricsA = \lyricmode {
  The first verse has
  \dropLyrics
  \set stanza = #"  All:"
  the com -- mon __ words
  \raiseLyrics
  used in all four.
}

lyricsB = \lyricmode { In stan -- za two,   \skipFour al -- so ap -- pear. }

lyricsC = \lyricmode { By the third verse, \skipFour are get -- ting dull. }

lyricsD = \lyricmode { Last stan -- za, and \skipFour get used once more. }

melody = \relative c' {
  c4 d e f |
  g f e8( e f) d |
  c4 e d c |
}

\score {
  <<
    \new Voice = m \melody
    \new Lyrics \lyricsto m \lyricsA
    \new Lyrics \lyricsto m \lyricsB
    \new Lyrics \lyricsto m \lyricsC
    \new Lyrics \lyricsto m \lyricsD
  >>
}

```

The first verse has
 In stan - za two,
 By the third verse,
 Last stan - za, and

used in all four.
 al - so ap - pear.
 are get - ting dull.
 get used once more.

Vorlage für Vokalensemble

Dieses Beispiel ist für vierstimmigen Gesang (SATB). Bei größeren Stücken ist es oft sinnvoll, eine allgemeine Variable zu bestimmen, die in allen Stimmen eingefügt wird. Taktart und Vorzeichen etwa sind fast immer gleich in allen Stimmen.

```

\paper {
  top-system-spacing.basic-distance = 10
  score-system-spacing.basic-distance = 20
}

```



```

    system-system-spacing.basic-distance = 20
    last-bottom-spacing.basic-distance = 10
}

global = {
  \key c \major
  \time 4/4
}

sopMusic = \relative {
  c''4 c c8[( b)] c4
}
sopWords = \lyricmode {
  hi hi hi hi
}

altoMusic = \relative {
  e'4 f d e
}
altoWords = \lyricmode {
  ha ha ha ha
}

tenorMusic = \relative {
  g4 a f g
}
tenorWords = \lyricmode {
  hu hu hu hu
}

bassMusic = \relative {
  c4 c g c
}
bassWords = \lyricmode {
  ho ho ho ho
}

\score {
  \new ChoirStaff <<
    \new Lyrics = "sopranos" \with {
      % this is needed for lyrics above a staff
      \override VerticalAxisGroup.staff-affinity = #DOWN
    }
    \new Staff = "women" <<
      \new Voice = "sopranos" {
        \voiceOne
        << \global \sopMusic >>
      }
      \new Voice = "altos" {
        \voiceTwo
        << \global \altoMusic >>
      }
    }
  }

```

```

>>
\new Lyrics = "altos"
\new Lyrics = "tenors" \with {
  % this is needed for lyrics above a staff
  \override VerticalAxisGroup.staff-affinity = #DOWN
}
\new Staff = "men" <<
  \clef bass
  \new Voice = "tenors" {
    \voiceOne
    << \global \tenorMusic >>
  }
  \new Voice = "basses" {
    \voiceTwo << \global \bassMusic >>
  }
>>
\new Lyrics = "basses"
\context Lyrics = "sopranos" \lyricsto "sopranos" \sopWords
\context Lyrics = "altos" \lyricsto "altos" \altoWords
\context Lyrics = "tenors" \lyricsto "tenors" \tenorWords
\context Lyrics = "basses" \lyricsto "basses" \bassWords
>>
}

```



Vocal ensemble template with automatic piano reduction

In diesem Beispiel wird ein automatischer Klavierauszug zu der Chorpartitur hinzugefügt. Das zeigt eine der Stärken von LilyPond – man kann eine Variable mehr als einmal benutzen. Wenn Sie irgendeine Änderung an einer Chorstimme vornehmen, (etwa `tenorMusic`), verändert sich auch der Klavierauszug entsprechend.

```

\paper {
  top-system-spacing.basic-distance = 10
  score-system-spacing.basic-distance = 20
  system-system-spacing.basic-distance = 20
  last-bottom-spacing.basic-distance = 10
}

global = {
  \key c \major
  \time 4/4
}

```

```

sopMusic = \relative {
  c' '4 c c8[( b)] c4
}
sopWords = \lyricmode {
  hi hi hi hi
}

altoMusic = \relative {
  e'4 f d e
}
altoWords = \lyricmode {
  ha ha ha ha
}

tenorMusic = \relative {
  g4 a f g
}
tenorWords = \lyricmode {
  hu hu hu hu
}

bassMusic = \relative {
  c4 c g c
}
bassWords = \lyricmode {
  ho ho ho ho
}

\score {
  <<
    \new ChoirStaff <<
      \new Lyrics = "sopranos" \with {
        % This is needed for lyrics above a staff
        \override VerticalAxisGroup.staff-affinity = #DOWN
      }
      \new Staff = "women" <<
        \new Voice = "sopranos" { \voiceOne << \global \sopMusic >> }
        \new Voice = "altos" { \voiceTwo << \global \altoMusic >> }
      >>
      \new Lyrics = "altos"

      \new Lyrics = "tenors" \with {
        % This is needed for lyrics above a staff
        \override VerticalAxisGroup.staff-affinity = #DOWN
      }
      \new Staff = "men" <<
        \clef bass
        \new Voice = "tenors" { \voiceOne << \global \tenorMusic >> }
        \new Voice = "basses" { \voiceTwo << \global \bassMusic >> }
      >>
      \new Lyrics = "basses"
    >>
  >>
}

```

```

\context Lyrics = "sopranos" \lyricsto "sopranos" \sopWords
\context Lyrics = "altos" \lyricsto "altos" \altoWords
\context Lyrics = "tenors" \lyricsto "tenors" \tenorWords
\context Lyrics = "basses" \lyricsto "basses" \bassWords
>>

\new PianoStaff <<
  \new Staff <<
    \set Staff.printPartCombineTexts = ##f
    \partCombine
    << \global \sopMusic >>
    << \global \altoMusic >>
  >>
  \new Staff <<
    \clef bass
    \set Staff.printPartCombineTexts = ##f
    \partCombine
    << \global \tenorMusic >>
    << \global \bassMusic >>
  >>
>>
>>
>>
}

```

hi hi hi hi

ha ha ha ha

hu hu hu hu

ho ho ho ho

Vorlage für Vokalensemble mit dem Gesangstext über und unter dem System

In diesem Beispiel werden die Texte mit den Befehlen `alignAboveContext` und `alignBelowContext` über und unter dem System angeordnet.

```

global = {
  \key c \major
  \time 4/4

```

```

}

sopMusic = \relative c'' {
  c4 c c8[( b)] c4
}
sopWords = \lyricmode {
  hi hi hi hi
}

altoMusic = \relative c' {
  e4 f d e
}
altoWords = \lyricmode {
  ha ha ha ha
}

tenorMusic = \relative c' {
  g4 a f g
}
tenorWords = \lyricmode {
  hu hu hu hu
}

bassMusic = \relative c {
  c4 c g c
}
bassWords = \lyricmode {
  ho ho ho ho
}

\score {
  \new ChoirStaff <<
    \new Staff = "women" <<
      \new Voice = "sopranos" { \voiceOne << \global \sopMusic >> }
      \new Voice = "altos" { \voiceTwo << \global \altoMusic >> }
    >>
    \new Lyrics \with { alignAboveContext = "women" }
      \lyricsto "sopranos" \sopWords
    \new Lyrics \with { alignBelowContext = "women" }
      \lyricsto "altos" \altoWords
    % we could remove the line about this with the line below, since
    % we want the alto lyrics to be below the alto Voice anyway.
    % \new Lyrics \lyricsto "altos" \altoWords

    \new Staff = "men" <<
      \clef bass
      \new Voice = "tenors" { \voiceOne << \global \tenorMusic >> }
      \new Voice = "basses" { \voiceTwo << \global \bassMusic >> }
    >>
    \new Lyrics \with { alignAboveContext = "men" }
      \lyricsto "tenors" \tenorWords
    \new Lyrics \with { alignBelowContext = "men" }

```

```

\lyricsto "basses" \bassWords
% again, we could replace the line above this with the line below.
% \new Lyrics \lyricsto "basses" \bassWords
>>
}

```



Sologesang und zweistimmiger Refrain

Diese Vorlage erstellt eine Partitur, die mit Sologesang beginnt und einen Refrain für zwei Stimmen enthält. Sie zeigt auch die Benutzung von Platzhalter-Pausen innerhalb der `\global`-Variable, um Taktwechsel (und andere Elemente, die für alle Stimmen gleich sind) für das gesamte Stück zu definieren.

```

global = {
  \key g \major

  % verse
  \time 3/4
  s2.*2
  \break

  % refrain
  \time 2/4
  s2*2
  \bar "|"
}

SoloNotes = \relative g' {
  \clef "treble"

  % verse
  g4 g g |
  b4 b b |

  % refrain
  R2*2 |
}

SoloLyrics = \lyricmode {
  One two three |
  four five six |
}

```

```

SopranoNotes = \relative c'' {
  \clef "treble"

  % verse
  R2.*2 |

  % refrain
  c4 c |
  g4 g |
}

SopranoLyrics = \lyricmode {
  la la |
  la la |
}

BassNotes = \relative c {
  \clef "bass"

  % verse
  R2.*2 |

  % refrain
  c4 e |
  d4 d |
}

BassLyrics = \lyricmode {
  dum dum |
  dum dum |
}

\score {
  <<
    \new Voice = "SoloVoice" << \global \SoloNotes >>
    \new Lyrics \lyricsto "SoloVoice" \SoloLyrics

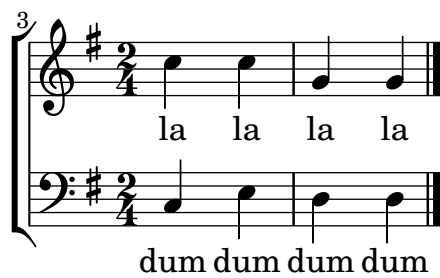
    \new ChoirStaff <<
      \new Voice = "SopranoVoice" << \global \SopranoNotes >>
      \new Lyrics \lyricsto "SopranoVoice" \SopranoLyrics

      \new Voice = "BassVoice" << \global \BassNotes >>
      \new Lyrics \lyricsto "BassVoice" \BassLyrics
    >>
  >>

  \layout {
    ragged-right = ##t
    \context { \Staff
      % these lines prevent empty staves from being printed
      \RemoveEmptyStaves
    }
  }
}

```

```
\override VerticalAxisGroup.remove-first = ##t  
}  
}  
}
```



10 Chord notation

See also Abschnitt “Chord notation” in *Notationsreferenz*.

Adding a figured bass above or below the notes

When writing figured bass, you can place the figures above or below the bass notes by using the commands `\bassFigureStaffAlignmentDown` and `\bassFigureStaffAlignmentUp`. Prepend `\once` to the command if you want to modify only the next figured bass.

The command `\bassFigureStaffAlignmentNeutral` resets the direction of figured bass to the default value.

```
bass = {
  \clef bass
  g4 b, c d |
  e d8 c d2
}

continuo = \figuremode {
  <_>4 <6>4 <5/>4
  \bassFigureStaffAlignmentUp
  <_+>4 <6> |
  \set Staff.useBassFigureExtenders = ##t
  \bassFigureStaffAlignmentDown
  <4>4. <4>8 <_+>4
}

\score {
  <<
    \new Staff = bassStaff \bass
    \context Staff = bassStaff \continuo
  >>
}
```



Taktstriche in einen ChordNames-Kontext hinzufügen

Um Taktstriche in einem ChordNames-Kontext anzeigen zu lassen, muss der `Bar_engraver` hinzugefügt werden.

```
\new ChordNames \with {
  \override BarLine.bar-extent = #'(-1 . 3)
  \consists "Bar_engraver"
}

\chordmode {
  f1:maj7 f:7 bes:7
}
```

F^Δ | F⁷ | B^{b7} |

Adjusting figured bass alteration glyphs

In figured bass, specially designed glyphs for 6\\, 7\\, and 9\\ are used by default. Similarly, specially designed glyphs for symbols 2\\+, 4\\+, and 5\\+ are used by default if plus signs appear after the number.

To change that, pass an alist to `figuredBassPlusStrokedAlist` and set the glyph in question to `#f` (or omit it).

```
#(set-global-staff-size 26)
```

```
\figures {
  \set figuredBassPlusDirection = #RIGHT
  <6\\> <7\\> <9\\> r
  <2\\+> <4\\+> <5\\+> r

  \set figuredBassPlusStrokedAlist =
    #'((2 . "figbass.twoplus")
      ;; (4 . "figbass.fourplus")
      ;; (5 . "figbass.fiveplus")
      (6 . "figbass.sixstroked")
      ;; (7 . "figbass.sevenstroked")
      ;; (9 . "figbass.ninestroked")
    )
  <6\\> <7\\> <9\\> r
  <2\\+> <4\\+> <5\\+> r
}
```

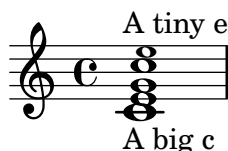
6 7 9 2 4 5+ 6 7 9 2 4+ 5+

Changing a single note's size in a chord

Individual note heads in a chord can be modified with the `\tweak` command inside a chord, by altering the `font-size` property.

Inside the chord (within the brackets `< >`), before the note to be altered, place the `\tweak` command, followed by `font-size` and define the proper size like `#-2` (a tiny note head).

```
\relative c' {
  <\tweak font-size #+2 c e g c
  \tweak font-size #-2 e>1
  ~\markup { A tiny e }_~\markup { A big c }
}
```



Akkordsymboltrenner verändern

Der Trenner zwischen unterschiedlichen Teilen eines Akkordsymbols kann beliebiger Text sein.

```
\chords {
  c:7sus4
  \set chordNameSeparator = \markup { \typewriter | }
  c:7sus4
}
```

}

C⁷ sus4 C⁷ | sus4

Positionen von Generalbass-Alterationszeichen verändern

Versetzungszeichen und Pluszeichen können vor oder nach den Ziffern erscheinen, je nach den Einstellungen der `figuredBassAlterationDirection` und `figuredBassPlusDirection`-Eigenschaften.

```

#(set-global-staff-size 26)

```

```

\figures {
  <5\+> <5+ 4\+> <6 4- 2\+> r
  \set figuredBassAlterationDirection = #RIGHT
  <5\+> <5+ 4\+> <6 4- 2\+> r
  \set figuredBassPlusDirection = #RIGHT
  <5\+> <5+ 4\+> <6 4- 2\+> r
  \set figuredBassAlterationDirection = #LEFT
  <5\+> <5+ 4\+> <6 4- 2\+> r
}

```

$\begin{array}{c} +5 \quad \sharp 5 \quad 6 \\ +\cancel{4} \quad \flat\cancel{4} \\ +2 \end{array}$	$\begin{array}{c} +5 \quad 5\sharp \quad 6 \\ +\cancel{4} \quad \cancel{4}\flat \\ +2 \end{array}$	$\begin{array}{c} 5^+ \quad 5\sharp \quad 6 \\ \cancel{4}^+ \quad \cancel{4}\flat \\ 2_+ \end{array}$	$\begin{array}{c} 5^+ \quad \sharp 5 \quad 6 \\ \cancel{4}^+ \quad \flat\cancel{4} \\ 2_+ \end{array}$
---	--	---	--

Akkordsymbolausnahmen

Die Eigenschaft `chordNameExceptions` kann benutzt werden, um eine Liste an besonderen Notationen für bestimmte Akkorde zu speichern.

```

% Step 1: Define music with chords and markup for maj9 and 6(add9).

```

```

chExceptionMusic = {
  <c e g b d'>-\markup { \super "maj9" }
  <c e g a d'>-\markup { \super "6(add9)" }
}

```

```

% Step 2: Create extended exception list.

```

```

chExceptions =
#(append (sequential-music-to-chord-exceptions chExceptionMusic #t)
  ignatzekExceptions)

```

```

theMusic = \chordmode {
  g1:maj9 g1:6.9
  % Step 3: Register extended exception list.
  \set chordNameExceptions = #chExceptions
  g1:maj9 g1:6.9
}

```

<<

```

  \new ChordNames \theMusic
  \new Voice \theMusic

```

>>

```

\layout {

```

```

line-width = 10\cm
ragged-right = ##f
}

```



Akkordbezeichnung maj7

Das Aussehen des großen Septakkords kann mit `majorSevenSymbol` verändert werden.

```

\chords {
  c:7+
  \set majorSevenSymbol = \markup { j7 }
  c:7+
}

```

C^Δ C^{j7}

Chord names alternative

Chord names are generated from a list of pitches. The functions which construct these names can be customised.

Here are shown chords following Ignatzek (pp. 17-18, 1995), used by default since LilyPond 1.7.20, compared with an alternative Jazz chord notation and Harald Banter's (1987) notation. A smaller font is used in the latter case, as these tend to be overly verbose.

This mirrors the mechanism originally used in early LilyPond versions (pre-1.7); not having been properly maintained, however, some features have been lost (mainly chord exception lists) and bugs have been introduced.

```

%%% Legacy chord naming functions (formerly in scm/chord-generic-names.scm)
%%% Copyright (C) 2003--2023 Jan Nieuwenhuizen <janneke@gnu.org>

```

```

#(set-global-staff-size 19.7)

#(define-public (banter-chordnames pitches bass inversion context)
  (old_chord->markup 'banter pitches bass inversion context))

#(define-public (jazz-chordnames pitches bass inversion context)
  (old_chord->markup 'jazz pitches bass inversion context))

#(define (define-translator-property symbol type? description)
  (if (not (and (symbol? symbol)
                (procedure? type?)
                (string? description)))
      (ly:error "error in call of define-translator-property"))
  (if (not (equal? (object-property symbol 'translation-doc) #f))
      (ly:error (G_ "symbol ~S redefined") symbol))

  (set-object-property! symbol 'translation-type? type?)
  (set-object-property! symbol 'translation-doc description)
  symbol)

```

```

#(for-each
  (lambda (x)
    (apply define-translator-property x))
  `((chordNameExceptionsFull ,list? "An alist of full chord
exceptions. Contains @code{(@var{chord} . @var{markup})} entries.")
    (chordNameExceptionsPartial ,list? "An alist of partial chord
exceptions. Contains @code{(@var{chord} . (@var{prefix-markup}
@var{suffix-markup}))} entries.")))

#(define-public (old_chord->markup
                 style pitches bass inversion context)
  "Entry point for @code{Chord_name_engraver}.
@var{pitches}, @var{bass}, and @var{inversion} are lily pitches."
  (define (default-note-namer pitch)
    (note-name->markup pitch #f))

  (define (markup-or-empty-markup markup)
    "Return MARKUP if markup, else empty-markup"
    (if (markup? markup) markup empty-markup))

  (define (accidental->markup alteration)
    "Return accidental markup for ALTERATION."
    (if (= alteration 0)
        (make-line-markup (list empty-markup))
        (conditional-kern-before
         (alteration->text-accidental-markup alteration)
         (= alteration FLAT) 0.094725))))

(define (list-minus a b)
  "Return list of elements in A that are not in B."
  (lset-difference eq? a b))

(define (markup-join markups sep)
  "Return line-markup of MARKUPS, joining them with markup SEP"
  (if (pair? markups)
      (make-line-markup (list-insert-separator markups sep)
                        empty-markup))
      empty-markup))

(define (conditional-kern-before markup bool amount)
  "Add AMOUNT of space before MARKUP if BOOL is true."
  (if bool
      (make-line-markup
       (list (make-hspace-markup amount)
             markup))
      markup))

(define (step-nr pitch)
  (let* ((pitch-nr (+ (* 7 (ly:pitch-octave pitch))
                     (ly:pitch-notename pitch)))
        (root-nr (+ (* 7 (ly:pitch-octave (car pitches))
                     (ly:pitch-notename (car pitches))))))
    ))

```

```

(+ 1 (- pitch-nr root-nr))))

(define (next-third pitch)
  (+ pitch
    (ly:make-pitch 0 2 (if (or (= (step-nr pitch) 3)
                              (= (step-nr pitch) 5))
                          FLAT 0))))

(define (step-alteration pitch)
  (let* ((normalized-pitch (- pitch (car pitches)))
        (alteration (ly:pitch-alteration normalized-pitch)))
    (if (= (step-nr pitch) 7) (+ alteration SEMI-TONE) alteration)))

(define (pitch-unalter pitch)
  (let ((alteration (step-alteration pitch)))
    (if (= alteration 0)
        pitch
        (ly:make-pitch (ly:pitch-octave pitch) (ly:pitch-notename pitch)
                        (- (ly:pitch-alteration pitch) alteration)))))

(define (step-even-or-altered? pitch)
  (let ((nr (step-nr pitch)))
    (if (!= (modulo nr 2) 0)
        (!= (step-alteration pitch) 0)
        #t)))

(define (step->markup-plusminus pitch)
  (let ((alt (step-alteration pitch)))
    (make-line-markup
     (list
      (number->string (step-nr pitch))
      (cond
       ((= alt DOUBLE-FLAT) "--")
       ((= alt FLAT) "-")
       ((= alt NATURAL) "")
       ((= alt SHARP) "+")
       ((= alt DOUBLE-SHARP) "++"))))))

(define (step->markup-accidental pitch)
  (make-line-markup
   (list (accidental->markup (step-alteration pitch))
         (make-simple-markup (number->string (step-nr pitch))))))

(define (step->markup-ignatzek pitch)
  (make-line-markup
   (if (and (= (step-nr pitch) 7)
            (= (step-alteration pitch) 1))
       (list (ly:context-property context 'majorSevenSymbol))
       (list (accidental->markup (step-alteration pitch))
             (make-simple-markup (number->string (step-nr pitch))))))

;; tja, kennok

```

```

(define (make-sub->markup step->markup)
  (lambda (pitch)
    (make-line-markup (list (make-simple-markup "no")
                           (step->markup pitch)))))

(define (step-based-sub->markup step->markup pitch)
  (make-line-markup (list (make-simple-markup "no") (step->markup pitch))))

(define (get-full-list pitch)
  (if (<= (step-nr pitch) (step-nr (last pitches)))
      (cons pitch (get-full-list (next-third pitch)))
      '()))

(define (get-consecutive nr pitches)
  (if (pair? pitches)
      (let* ((pitch-nr (step-nr (car pitches)))
             (next-nr (if (≠ (modulo pitch-nr 2) 0) (+ pitch-nr 2) nr)))
        (if (<= pitch-nr nr)
            (cons (car pitches) (get-consecutive next-nr (cdr pitches)))
            '()))
      '()))

;;; FIXME -- exceptions no longer work. -vv

(define (full-match exceptions)
  (if (pair? exceptions)
      (let* ((e (car exceptions))
             (e-pitches (car e)))
        (if (equal? e-pitches pitches)
            e
            (full-match (cdr exceptions)))))
  #f))

(define (partial-match exceptions)
  (if (pair? exceptions)
      (let* ((e (car exceptions))
             (e-pitches (car e)))
        (if (equal? e-pitches (take pitches (length e-pitches)))
            e
            (partial-match (cdr exceptions)))))
  #f))

;;; FIXME: exceptions don't work anyway.
(if #f (begin
  (write-me "pitches: " pitches))
  (let* ((full-exceptions
    (ly:context-property context 'chordNameExceptionsFull))
    (full-exception (full-match full-exceptions))
    (full-markup (if full-exception (cadr full-exception) '()))
    (partial-exceptions
    (ly:context-property context 'chordNameExceptionsPartial))
    (partial-exception (partial-match partial-exceptions)))

```

```

(partial-pitches (if partial-exception (car partial-exception) '()))
(partial-markup-prefix
  (if partial-exception (markup-or-empty-markup
                        (caddr partial-exception)) empty-markup))
(partial-markup-suffix
  (if (and partial-exception (pair? (caddr partial-exception)))
      (markup-or-empty-markup (caddr partial-exception)) empty-markup))
(root (car pitches))
(full (get-full-list root))
;; kludge alert: replace partial matched lower part of all with
;; 'normal' pitches from full
;; (all pitches)
(all (append (take full (length partial-pitches))
             (drop pitches (length partial-pitches))))

(highest (last all))
(missing (list-minus full (map pitch-unalter all)))
(consecutive (get-consecutive 1 all))
(rest (list-minus all consecutive))
(altered (filter step-even-or-altered? all))
(cons-alt (filter step-even-or-altered? consecutive))
(base (list-minus consecutive altered)))

(if #f (begin
  (write-me "full:" full)
  ;; (write-me "partial-pitches:" partial-pitches)
  (write-me "full-markup:" full-markup)
  (write-me "partial-markup-perfix:" partial-markup-prefix)
  (write-me "partial-markup-suffix:" partial-markup-suffix)
  (write-me "all:" all)
  (write-me "altered:" altered)
  (write-me "missing:" missing)
  (write-me "consecutive:" consecutive)
  (write-me "rest:" rest)
  (write-me "base:" base)))

(case style
  ((banter)
   ;; root
   ;; + steps:altered + (highest all -- if not altered)
   ;; + subs:missing

  (let* ((root->markup default-note-namer)
         (step->markup step->markup-plusminus)
         (sub->markup (lambda (x)
                       (step-based-sub->markup step->markup x)))
         (sep (make-simple-markup "/")))

    (if
      (pair? full-markup)
      (make-line-markup (list (root->markup root) full-markup))

```



```

(make-line-markup
  (list
    (root->markup root)
    partial-markup-prefix
    (make-super-markup
      (markup-join
        (append
          (map step->markup
            (append altered
              (if (and (> (step-nr highest) 5)
                (not
                  (step-even-or-altered? highest)))
                (list highest) '()))))
          (list partial-markup-suffix)
          (map sub->markup missing))
        sep))))))

((jazz)
  ;; root
  ;; + steps:(highest base) + cons-alt
  ;; + 'add'
  ;; + steps:rest
  (let* ((root->markup default-note-namer)
    (step->markup step->markup-ignatzek)
    (sep (make-simple-markup " "))
    (add-prefix (make-simple-markup " add"))))

    (if
      (pair? full-markup)
      (make-line-markup (list (root->markup root) full-markup))

      (make-line-markup
        (list
          (root->markup root)
          partial-markup-prefix
          (make-super-markup
            (make-line-markup
              (list

                ;; kludge alert: omit <= 5
                ;;(markup-join (map step->markup
                ;;              (cons (last base) cons-alt)) sep)

                ;; This fixes:
                ;; c      C5      -> C
                ;; c:2    C5 2    -> C2
                ;; c:3-   Cm5     -> Cm
                ;; c:6.9 C5 6add9 -> C6 add 9 (add?)
                ;; ch = \chords { c c:2 c:3- c:6.9^7 }
                (markup-join (map step->markup

```

```

                                (let ((tb (last base)))
                                  (if (> (step-nr tb) 5)
                                      (cons tb cons-alt)
                                      cons-alt))) sep)

    (if (pair? rest)
        add-prefix
        empty-markup)
    (markup-join (map step->markup rest) sep)
    partial-markup-suffix))))))

(else empty-markup)))

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%% Here begins the actual snippet:

chs = \transpose c' c' {
  <c e g>1
  <c es g> % m = minor triad
  <c e gis>
  <c es ges> \break
  <c e g bes>
  <c es g bes>
  <c e g b> % triangle = maj
  <c es ges beses>
  <c es ges b> \break
  <c e gis bes>
  <c es g b>
  <c e gis b>
  <c es ges bes> \break
  <c e g a> % 6 = major triad with added sixth
  <c es g a> % m6 = minor triad with added sixth
  <c e g bes d'>
  <c es g bes d'> \break
  <c es g bes d' f' a' >
  <c es g bes d' f' >
  <c es ges bes d' >
  <c e g bes des' > \break
  <c e g bes dis'>
  <c e g bes d' f'>
  <c e g bes d' fis'>
  <c e g bes d' f' a'> \break
  <c e g bes d' fis' as'>
  <c e gis bes dis'>
  <c e g bes dis' fis'>
  <c e g bes d' f' as'> \break
  <c e g bes des' f' as'>
  <c e g bes d' fis'>
  <c e g b d'>
  <c e g bes d' f' as'> \break

```

```

<c e g bes des' f' as'>
<c e g bes des' f' a'>
<c e g b d'>
<c e g b d' f' a'> \break
<c e g b d' fis'>
<c e g bes des' f ' a'>
<c f g>
<c f g bes> \break
<c f g bes d'>
<c e g d'> % add9
<c es g f'>
<c e g b fis'> % Lydian
<c e g bes des' ees' fis' aes'> % altered chord
}

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% alternate Jazz notation

```

```

efullmusicJazzAlt = {
  <c e gis>1-\markup { "+" }
  <c e g b>-\markup {
    \normal-size-super
    % \override #'(font-family . math) "N"
    \override #'(font-family . math) "M"
  }
  %%c:3.5.7 = \markup { \override #'(font-family . math) "M" }
  %%c:3.5.7 = \markup { \normal-size-super "maj7" }

  <c es ges>-\markup { \super "o" } % should be $\circ$ ?
  <c es ges bes>-\markup { \super \combine "o" "/" }
  <c es ges beses>-\markup { \super "o7" }
}

```

```

efullJazzAlt = #(sequential-music-to-chord-exceptions efullmusicJazzAlt #f)

```

```

epartialmusicJazzAlt = {
  <c d>1-\markup { \normal-size-super "2" }
  <c es>-\markup { "m" }
  <c f>-\markup { \normal-size-super "sus4" }
  <c g>-\markup { \normal-size-super "5" }
  %% TODO, partial exceptions
  <c es f>-\markup { "m" }-\markup { \normal-size-super "sus4" }
  <c d es>-\markup { "m" }-\markup { \normal-size-super "sus2" }
}

```

```

epartialJazzAlt = #(sequential-music-to-chord-exceptions epartialmusicJazzAlt #f)

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

\score {
  <<

```

```

\new ChordNames {
  %% Already set by default:
  \%set chordNameFunction = #ignatzek-chord-names
  \set instrumentName = "Ignatzek"
  \set shortInstrumentName = "Def"
  \chs
}

\new ChordNames {
  \set chordNameFunction = #jazz-chordnames
  \set majorSevenSymbol = \whiteTriangleMarkup
  \set chordNameSeparator = "/"
  \set chordNameExceptionsFull = \efullJazzAlt
  \set chordNameExceptionsPartial = \epartialJazzAlt
  \set instrumentName = "Alternative"
  \set shortInstrumentName = "Alt"
  \chs
}

%% This is the Banter (1987) style.  It gives exceedingly
%% verbose (wide) names, making the output file take up to 4 pages.

\new ChordNames {
  \set chordNameFunction = #banter-chordnames
  \override ChordName.font-size = -3
  \set instrumentName = "Banter"
  \set shortInstrumentName = "Ban"
  \chs
}

\new Staff \transpose c c' { \chs }
>>
\layout {
  #(layout-set-staff-size 16)
  system-system-spacing.basic-distance = 0
  \context {
    \ChordNames
    \consists "Instrument_name_engraver"
  }
  \context {
    \Score
    \remove "Bar_number_engraver"
  }
}
}

```

Ignatzek	C	Cm	C+	C°
Alternative	C	C ^{b3}	C ^{#5}	C ^{b3 b5}
Banter	C _{no3/no5}	C _{3-//no3/no5}	C _{5+//no3/no5}	C _{3-/5-//no3/no5}

Def	C^7	Cm^7	C^Δ	$C^{\circ 7}$	$Cm^\Delta b5$
Alt	C^7	$C^7 b3$	$C^\#7$	$C^b3 b5 b7$	$C^b3 b5 \#7$
Ban	$C^{7/11/13/15/17}$	$C^{3-/7/11/13/15/17}$	$C^{7+/11/13/15/17}$	$C^{3-/5-/7-/11/13/15/17}$	$C^{3-/5-/7+/11/13/15/17}$
Def	$C^{7 \#5}$	Cm^Δ	$C^\Delta \#5$	C°	
Alt	$C^{7 \#5}$	$C^b3 \#7$	$C^\#5 \#7$	$C^7 b3 b5$	
Ban	$C^{5+/7/11/13/15/17}$	$C^{3-/7+/11/13/15/17}$	$C^{5+/7+/11/13/15/17}$	$C^{3-/5-/7/11/13/15/17}$	
Def	C^6	Cm^6	C^9	Cm^9	
Alt	C^6	$C^b3 6$	C^9	$C^9 b3$	
Ban	$C^{6/11/13/15}$	$C^{3-/6/11/13/15}$	$C^{9/11/13/15/17/19/21}$	$C^{3-/9/11/13/15/17/19/21}$	
Def	Cm^{13}	Cm^{11}	$Cm^7 b5 9$	$C^7 b9$	
Alt	$C^{13 b3}$	$C^{11 b3}$	$C^9 b3 b5$	$C^7 b9$	
Ban	$C^{3-/13/15/17/19/21/23+/25}$	$C^{3-/11/13/15/17/19/21+/23}$	$C^{3-/5-/9/11/13/15/17/19/21}$	$C^{9-/13/15/17/19/21/23}$	
Def	$C^7 \#9$	C^{11}	$C^7 \#11$	C^{13}	
Alt	$C^7 \#9$	C^{11}	$C^9 \#11$	C^{13}	
Ban	$C^{9+/11/13/15/17/19/21}$	$C^{11/13/15/17/19/21+/23}$	$C^{11+/13/15/17/19/21+/23}$	$C^{13/15/17/19/21/23+/25}$	
Def	$C^7 \#11 b13$	$C^7 \#5 \#9$	$C^7 \#9 \#11$	$C^7 b13$	
Alt	$C^9 \#11 b13$	$C^7 \#5 \#9$	$C^7 \#9 \#11$	$C^{11 b13}$	
Ban	$C^{11+/13-/15/17/19/21+/23+/25}$	$C^{5+/9+/11/13/15/17/19/21}$	$C^{9+/11+/13/15/17/19/21+/23}$	$C^{13-/15/17/19/21+/23+/25}$	
Def	$C^7 b9 b13$	$C^7 \#11$	$C^\Delta 9$	$C^7 b13$	
Alt	$C^{11 b9 b13}$	$C^9 \#11$	$C^9 \#7$	$C^{11 b13}$	
Ban	$C^{9-/13-/15/17/19/21+/23+/25}$	$C^{11+/13/15/17/19/21+/23}$	$C^{7+/9/11/13/15/17/19/21}$	$C^{13-/15/17/19/21+/23+/25}$	
Def	$C^7 b9 b13$	$C^7 b9 13$	$C^\Delta 9$	$C^\Delta 13$	
Alt	$C^{11 b9 b13}$	$C^{13 b9}$	$C^9 \#7$	$C^{13 \#7}$	
Ban	$C^{9-/13-/15/17/19/21+/23+/25}$	$C^{9-/13/15/17/19/21+/23+/25}$	$C^{7+/9/11/13/15/17/19/21}$	$C^{7+/13/15/17/19/21+/23+/25}$	
Def	$C^\Delta \#11$	$C^7 b9 13$	C^{sus4}	$C^7 sus4$	
Alt	$C^9 \#7 \#11$	$C^{13 b9}$	$C^{add4 5}$	$C^{add4 5 7}$	
Ban	$C^{7+/11+/13/15/17/19/21+/23}$	$C^{9-/13/15/17/19/21+/23+/25}$	$C^{4/11/13/15/17}$	$C^{4/7/11/13/15/17}$	

Def	$C^9 \text{ sus4}$	C^{add9}	$C^{\text{m add11}}$	$C^{\Delta \text{ add}\sharp 11}$	C^{alt}
Alt	$C^{\text{add4 5 7 9}}$	C^{add9}	$C^{\flat 3 \text{ add11}}$	$C^{\sharp 7 \text{ add}\sharp 11}$	$C^{7 \flat 9 \flat 10 \sharp 11 \flat 13}$
Ban	$C^{4/9//\text{no3/no5/no7/no9}}$	$C^{9//\text{no3/no5/no7/no9}}$	$C^{3-/11//\text{no3/no5/no7/no9/no11+}}$	$C^{7+/11+//\text{no3/no5/no7/no9/no11+}}$	$C^{9-/10-/11+/13-/ \text{no3/no5/no7/no9/no11+}/\text{no13+}}$

Chords with stretched fingering for FretBoards and TabVoice

Sometimes chords with a stretched fingering are required. If not otherwise specified the context property `maximumFretStretch` is set to value 4, though, resulting in a warning about „No string for pitch ...“, and the note is omitted. You may set `maximumFretStretch` to an appropriate value or explicitly assign string numbers to all notes of a chord to fix that.

```
% The code below prints two warnings for the second chord,
% which may be omitted by uncommenting the following line.
%
% #(for-each (lambda (x) (ly:expect-warning "No string for pitch")) (iota 2))
```

```
mus = {
  <c' bes'>
  <c'\2 bes'>
  \set maximumFretStretch = 5
  <c' bes'>
  <c'\2 bes'\1>
}
```

```
<<
  \new FretBoards \mus
  \new TabVoice \mus
>>
```

Clusters

Clusters are a device to denote that a complete range of notes is to be played.

```
fragment = \relative c' {
  c4 f <e d'>4
  <g a>8 <e a> a4 c2 <d b>4
  e2 c
}

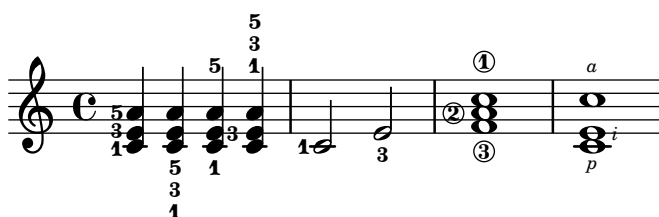
<<
  \new Staff \fragment
  \new Staff \makeClusters \fragment
>>
```



Position von Fingersatz in Akkorden kontrollieren

Die Position von Fingersatzzahlen kann exakt kontrolliert werden.

```
\relative c' {
  \set fingeringOrientations = #'(left)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down right up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(left)
  <c-1>2
  \set fingeringOrientations = #'(down)
  <e-3>2
  \set stringNumberOrientations = #'(up left down)
  <f\3 a\2 c\1>1
  \set strokeFingerOrientations = #'(down right up)
  <c\rightHandFinger 1 e\rightHandFinger 2 c'\rightHandFinger 4 >
}
```



Cross-staff chords – beaming problems workaround

Sometimes it is better to use stems from the ‘other’ staff for creating cross-staff chords to trick LilyPond’s beam collision detector. In the following snippet, if the stems from the lower staff were used instead, it would be necessary to explicitly use

```
\override Staff.Beam.collision-voice-only = ##t
```

so that LilyPond doesn’t move the beams.

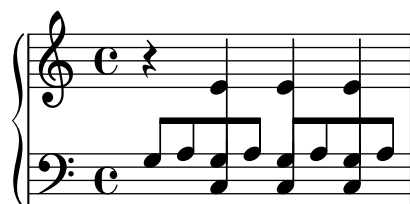
```
\new PianoStaff <<
  \new Staff = up \relative c' <<
  { r4
    \override Stem.cross-staff = ##t
    \override Stem.length = #19 % this is in half-spaces,
      % so it makes stems 9.5 staffspaces long
    \override Stem.Y-offset = #-6 % stems are normally lengthened
      % upwards, so here we must lower the stem by the amount
      % equal to the lengthening - in this case (19 - 7) / 2
```

```

        % (7 is default stem length)
    e e e }
  { s4
    \change Staff = "bottom"
    \override NoteColumn.ignore-collision = ##t
    c, c c
  }
>>

\new Staff = bottom \relative c' {
  \clef bass
  \voiceOne
  g8 a g a g a g a
}
>>

```



Customizing the chord grid style

Custom divisions of chord squares can be defined through the `measure-division-lines-alist` and `measure-division-chord-placement-alist` properties of `ChordSquare`. These are both alists. Their keys are measure divisions, namely lists which give the fraction of the measure that each chord (or rest, or skip) represents. More precisely, a measure division alist is made of positive, exact numbers adding up to 1, for example: '(1/2 1/4 1/4). The exactness requirement means that, e.g., 1/2 is valid but not 0.5.

The values in `measure-division-lines-alist` are lists of lines, which are represented as (x1 y1 x2 y2). The line starts at the point (x1 . y1) and ends at (x2 . y2). Coordinates are expressed in the [-1, 1] scale relative to the extent of the square.

The values in `measure-division-chord-placement-alist` are lists of (x . y) pairs giving the placement of the respective chords.

This example defines a peculiar chord grid style that has a rule for measures divided in three equal parts.

```

\paper {
  line-width = 10\cm
  ragged-right = ##f
}

\new ChordGrid \with {
  \override ChordSquare.measure-division-lines-alist =
    #'(((1) . (0))
      ((1/3 1/3 1/3) . ((-1 -0.4 0 1) (0 -1 1 0.4))))
  \override ChordSquare.measure-division-chord-placement-alist =
    #'(((1) . ((0 . 0)))
      ((1/3 1/3 1/3) . ((-0.7 . 0.5) (0 . 0) (0.7 . -0.5))))
}
\chordmode {

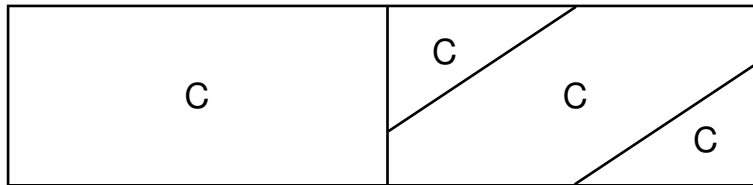
```



```

\time 3/4
c2.
c4 c4 c4
}

```



Customizing the no-chord symbol

By default, rests in a ChordNames context cause the text „N.C.“ to be printed. This markup can be customized by setting the noChordSymbol context property.

```

<<
\chords {
  R1
  \set noChordSymbol = "----"
  R1
  \set noChordSymbol = \markup \italic "Ssh!"
  R1
}
{
  R1*3
}
>>

```



Display non-English chord names

The default English naming of chords can be changed to other languages, as demonstrated in this snippet.

```

scm = \chordmode {
  c1/c | cis/cis
  b1/b | bis/bis | bes/bes
}

\layout {
  indent = 3\cm
  ragged-right = ##f

  \context {
    \ChordNames
    \consists "Instrument_name_engraver"
  }
  \context {
    \Score
    \override InstrumentName.self-alignment-Y = -1.2

```

```

\override InstrumentName.self-alignment-X = #RIGHT
}
}

<<
\new ChordNames {
  \set instrumentName = #"default"
  \scm
}
\new ChordNames {
  \set instrumentName = #"german"
  \germanChords \scm
}
\new ChordNames {
  \set instrumentName = #"semi-german"
  \semiGermanChords \scm
}
\new ChordNames {
  \set instrumentName = #"italian"
  \italianChords \scm
}
\new ChordNames {
  \set instrumentName = #"french"
  \frenchChords \scm
}
\context Voice { \scm }
>>

```

default	C/C	C#/C#	B/B	B#/B#	Bb/Bb
german	C/c	C#/cis	H/h	H#/his	B/b
semi-german	C/c	C#/cis	H/h	H#/his	Bb/b
italian	Do/Do	Do #/Do #	Si/Si	Si #/Si #	Si b/Si b
french	Do/Do	Do #/Do #	Si/Si	Si #/Si #	Si b/Si b

Displaying complex chords

Here is a way to display a chord where the same note is played twice with different accidentals.

```

fixA = {
  \once \override Stem.length = #12
}

fixB = {
  \once \override NoteHead.X-offset = #1.7
  \once \override Stem.length = #7
  \once \override Stem.rotation = #'(45 0 0)
  \once \override Stem.extra-offset = #'(-0.1 . -0.2)
}

```



```

;; Set 'chordNoteNamer, respect user setting if already done
(ly:context-set-property! ctx 'chordNoteNamer
  (if (procedure? chord-note-namer)
      chord-note-namer
      note-name->markup))))

(listeners
  ((note-event this-engraver event)
    (let* ((pitch (ly:event-property event 'pitch))
           (pitch-name (ly:pitch-notename pitch))
           (pitch-alt (ly:pitch-alteration pitch))
           (bass (ly:event-property event 'bass #f))
           (inversion (ly:event-property event 'inversion #f)))
      ;; Collect notes of the chord
      ;; - to compare inversed chords we need to collect the
      ;;   bass note as usual member of the chord, whereas an
      ;;   added bass must be treated separate from the usual
      ;;   chord-notes
      ;; - notes are stored as pairs containing their
      ;;   pitch-name (an integer), i.e. disregarding their
      ;;   octave and their alteration
      (cond (bass (set! bass-pitch pitch))
            (inversion
             (set! bass-pitch pitch)
             (set! chord-pitches
                   (cons (cons pitch-name pitch-alt)
                         chord-pitches)))
            (else
             (set! chord-pitches
                   (cons (cons pitch-name pitch-alt)
                         chord-pitches))))))

(acknowledgers
  ((chord-name-interface this-engraver grob source-engraver)
    (let ((chord-changes (ly:context-property ctx
                                              'chordChanges #f)))
      ;; If subsequent chords are equal apart from their bass,
      ;; reset the 'text-property.
      ;; Equality is done by comparing the sorted lists of this
      ;; chord's elements and the previous chord. Sorting is
      ;; needed because inverted chords may have a different
      ;; order of pitches. `chord-changes` needs to be true.
      (if (and bass-pitch
                chord-changes
                (equal?
                 (sort chord-pitches car<)
                 (sort last-chord-pitches car<)))
          (ly:grob-set-property!
            grob 'text
            (make-line-markup
              (list
                (ly:context-property ctx 'slashChordSeparator)

```

```

        ((ly:context-property ctx 'chordNoteNamer)
         bass-pitch
         (ly:context-property ctx
          'chordNameLowercaseMinor))))))
      (set! last-chord-pitches chord-pitches)
      (set! chord-pitches '())
      (set! bass-pitch #f)))

    ((finalize this-engraver)
     (set! last-chord-pitches '())))))

myChords = \chordmode {
  % \germanChords

  \set chordChanges = ##t
  d2:m d:m/cis

  d:m/c
  \set chordChanges = ##f
  d:m/b

  e1:7
  \set chordChanges = ##t
  e
  \break

  \once \set chordChanges = ##f
  e1/f
  e2/gis e/+gis e e:m/f d:m d:m/cis d:m/c
  \set chordChanges = ##f
  d:m/b
}

<<
  \new ChordNames
    \with { \consists #Bass_changes_equal_root_engraver }
    \myChords
  \new Staff \myChords
>>

```

Dm /C# /C Dm/B E⁷ E
 5 E/F /G# E Em/F Dm /C# /C Dm/B

Akkordsymbole bei Wechsel anzeigen

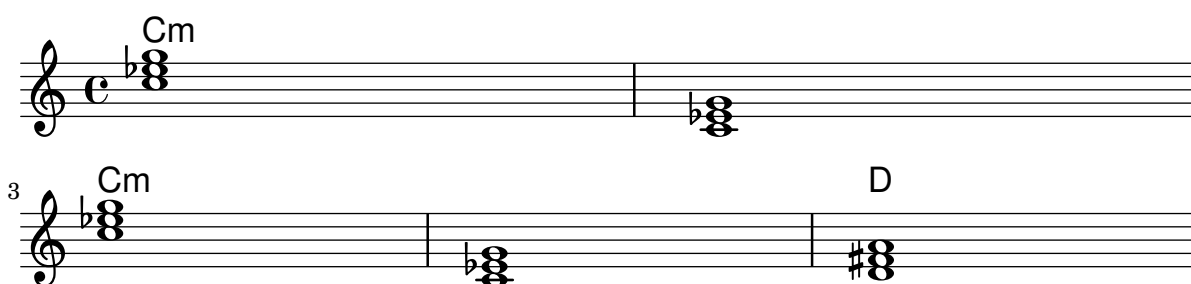
Akkordsymbole können so eingestellt werden, dass sie nur zu Beginn der Zeile und bei Akkordwechseln angezeigt werden.

```

harmonies = \chordmode {
  c'1:m c:m \break
  c'1:m c:m d
}

<<
\new ChordNames {
  \set chordChanges = ##t
  \harmonies
}
\new Staff {
  \harmonies
}
>>

```



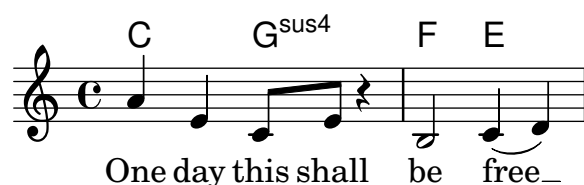
Ein einfaches Liedblatt

Ein Liedblatt besteht aus Akkordbezeichnungen, einer Melodie und dem Liedtext:

```

<<
\chords { c2 g:sus4 f e }
\new Staff \relative c' {
  a4 e c8 e r4
  b2 c4( d)
}
\addlyrics { One day this shall be free __ }
>>

```



Vorlage für eine Notenzeile mit Noten, Text und Akkorden

Mit diesem Beispiel können Sie einen Song mit Melodie, Text und Akkorden schreiben.

```

melody = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

```

```

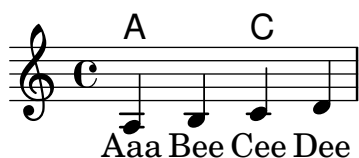
a4 b c d
}

text = \lyricmode {
  Aaa Bee Cee Dee
}

harmonies = \chordmode {
  a2 c
}

\score {
  <<
    \new ChordNames {
      \set chordChanges = ##t
      \harmonies
    }
    \new Voice = "one" { \autoBeamOff \melody }
    \new Lyrics \lyricsto "one" \text
  >>
  \layout { }
  \midi { }
}

```



Single-staff template with notes, lyrics, chords, and frets

Here is a simple lead sheet template with melody, lyrics, chords, and fret diagrams.

```

verseI = \lyricmode {
  \set stanza = #"1."
  This is the first verse
}

verseII = \lyricmode {
  \set stanza = #"2."
  This is the second verse.
}

theChords = \chordmode {
  % insert chords for chordnames and fretboards here
  c2 g4 c
}

staffMelody = \relative c' {
  \key c \major
  \clef treble
  % Type notes for melody here
}

```

```

c4 d8 e f4 g
\bar "|"
}

\score {
  <<
    \context ChordNames { \theChords }
    \context FretBoards { \theChords }
    \new Staff {
      \context Voice = "voiceMelody" { \staffMelody }
    }
    \new Lyrics = "lyricsI" {
      \lyricsto "voiceMelody" \verseI
    }
    \new Lyrics = "lyricsII" {
      \lyricsto "voiceMelody" \verseII
    }
  >>
  \layout { }
  \midi { }
}

```

1. This is the first verse
2. This is the second verse.

Vorlage für eine Notenzeile mit Akkorden

Wollen Sie ein Liedblatt mit Melodie und Akkorden schreiben? Hier ist das richtige Beispiel für Sie!

```

melody = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

  f4 e8[ c] d4 g |
  a2 ~ a
}

harmonies = \chordmode {
  c4:m f:min7 g:maj c:aug |
  d2:dim b4:5 e:sus
}

\score {
  <<
    \new ChordNames {

```



```

\set chordChanges = ##t
\harmonies
}
\new Staff \melody
>>
\layout{ }
\midi { }
}

```



Vertically centering paired figured bass extenders

Where figured bass extender lines are being used by setting `useBassFigureExtenders` to `##t`, pairs of congruent figured bass extender lines are vertically centered if `figuredBassCenterContinuations` is set to `##t`.

```

<<
\relative c' {
  \repeat unfold 3 {
    c8 c b b a a c16 c b b
  }
}
\figures {
  \set useBassFigureExtenders = ##t
  <6+ 4 3>4 <6 4 3>8 r
  <6+ 4 3>4 <6 4 3>8 <4 3+>16 r
  \set figuredBassCenterContinuations = ##t
  <6+ 4 3>4 <6 4 3>8 r
  <6+ 4 3>4 <6 4 3>8 <4 3+>16 r
  \set figuredBassCenterContinuations = ##f
  <6+ 4 3>4 <6 4 3>8 r
  <6+ 4 3>4 <6 4 3>8 <4 3+>16 r
}
>>

```



Wiederholungs-(Volta-)Klammern unterhalb der Akkordsymbole

Indem man den `Volta_engraver` zu dem entsprechenden Notensystem hinzufügt, können Wiederholungsklammern unterhalb der Akkorde gesetzt werden.

```

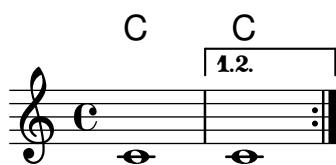
\score {
  <<
  \chords { c1 c1 }

```

```

\new Staff \with { \consists "Volta_engraver" }
{
  \repeat volta 2 { c'1 \alternative { c' } }
}
>>
\layout {
  \context {
    \Score
    \remove "Volta_engraver"
  }
}
}

```



11 Keyboard and other multi-staff instruments

See also Abschnitt “Keyboard and other multi-staff instruments” in *Notationsreferenz*.

Accordion register symbols

Accordion register symbols are available as `\markup` as well as as standalone music events (as register changes tend to occur between actual music events). Bass registers are not overly standardized. The available commands can be found in ‘Discant symbols’ in the Notation Reference (<https://lilypond.org/doc/v2.24/Documentation/notation/accordion#discant-symbols>).

```
#(use-modules (lily accreg))
```

```
\new PianoStaff <<
  \new Staff \relative {
    \clef treble
    \discant "10"
    r8 s32 f'[ bes f] s e[ a e] s d[ g d] s16 e32[ a]
    <<
      { r16 <f bes> r <e a> r <d g> }
      \\
      { d r a r bes r }
    >> |
    <cis e a>1
  }

  \new Staff \relative {
    \clef treble
    \freeBass "1"
    r8 d'32 s16. c32 s16. bes32 s16. a32[ cis] s16
    \clef bass \stdBass "Master"
    <<
      { r16 <f, bes d>^"b" r <e a c>^"am" r <d g bes>^"gm" |
        <e a cis>1^"a" }
      \\
      { d8_"D" c_"C" bes_"B" | a1_"A" }
    >>
  }
>>
```

The image shows a musical score for a piano and an accordion. The piano part is written on two staves (treble and bass), and the accordion part is written on a single staff. The score is in common time (C) and features a sequence of chords and notes. The accordion part uses discant symbols (b, am, gm, a) to indicate specific register changes. The piano part's bass line is labeled with D, C, B, and A, corresponding to the notes in the bass staff. The piano part's treble staff contains a series of eighth and sixteenth notes, while the bass staff contains a series of eighth and sixteenth notes. The accordion part is written in a single staff, with notes and chords that correspond to the piano part's bass line. The score is a demonstration of how to use the `\discant` and `\stdBass` commands in LilyPond.

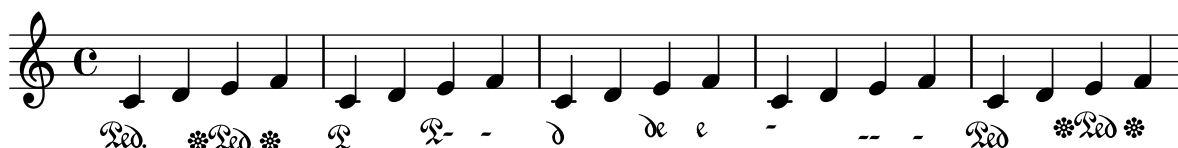
Changing the text for sustain markings

With the `pedalSustainStrings` context property it is possible to set the text used for pedal down and up. Note that the only valid strings are those found in the list of pedal glyphs – the values shown in this snippet constitute an exhaustive list.

```
sustainNotes = { c4\sustainOn d e\sustainOff\sustainOn f\sustainOff }
```

```
\relative c' {
  \sustainNotes
  \set Staff.pedalSustainStrings = #'("P" "P-" "-")
  \sustainNotes
  \set Staff.pedalSustainStrings = #'("d" "de" "e")
  \sustainNotes
  \set Staff.pedalSustainStrings = #'("M" "M-" "-")
  \sustainNotes
  \set Staff.pedalSustainStrings = #'("Ped" "*Ped" "*")
  \sustainNotes
}

\layout {
  ragged-right = ##f
}
```

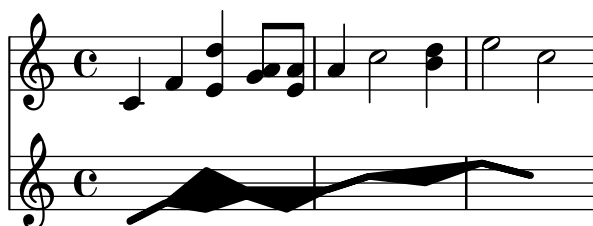


Clusters

Clusters are a device to denote that a complete range of notes is to be played.

```
fragment = \relative c' {
  c4 f <e d'>4
  <g a>8 <e a> a4 c2 <d b>4
  e2 c
}

<<
  \new Staff \fragment
  \new Staff \makeClusters \fragment
>>
```



Position von Fingersatz in Akkorden kontrollieren

Die Position von Fingersatzzahlen kann exakt kontrolliert werden.

```
\relative c' {
```

[illegible]

In some situations it is necessary to create slurs between notes from different voices. The solution is to add invisible notes to one of the voices, using `\hideNotes`.

```
\relative c' {
  <<
  {
    d16( a') s a s a[ s a] s a[ s a]
  }
  \\\
  {
    \slurUp
    bes,16[ s e](
    \hideNotes a)
    \unHideNotes f[(
    \hideNotes a)
    \unHideNotes fis](
    \hideNotes a)
    \unHideNotes g[(
    \hideNotes a)
    \unHideNotes gis](
    \hideNotes a)
  }
  >>
}
```



Cross-staff chords – beaming problems workaround

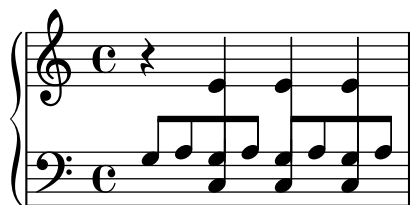
Sometimes it is better to use stems from the ‘other’ staff for creating cross-staff chords to trick LilyPond’s beam collision detector. In the following snippet, if the stems from the lower staff were used instead, it would be necessary to explicitly use

```
\override Staff.Beam.collision-voice-only = ##t
```

so that LilyPond doesn’t move the beams.

```
\new PianoStaff <<
  \new Staff = up \relative c' <<
    { r4
      \override Stem.cross-staff = ##t
      \override Stem.length = #19 % this is in half-spaces,
        % so it makes stems 9.5 staffspaces long
      \override Stem.Y-offset = #-6 % stems are normally lengthened
        % upwards, so here we must lower the stem by the amount
        % equal to the lengthening - in this case (19 - 7) / 2
        % (7 is default stem length)
      e e e }
    { s4
      \change Staff = "bottom"
      \override NoteColumn.ignore-collision = ##t
      c, c c
    }
  >>

  \new Staff = bottom \relative c' {
    \clef bass
    \voiceOne
    g8 a g a g a a
  }
  >>
```



Cross-staff tremolos

Since `\repeat tremolo` expects exactly two musical arguments for chord tremolos, the note or chord which changes staff within a cross-staff tremolo should be placed inside curly braces together with its `\change Staff` command.

```
\new PianoStaff <<
  \new Staff = "up" \relative c' {
    \key a \major
```

```

\time 3/8
s4.
}
\new Staff = "down" \relative c'' {
  \key a \major
  \time 3/8
  \voiceOne
  \repeat tremolo 6 {
    <a e'>32
    {
      \change Staff = "up"
      \voiceTwo
      <cis a' dis>32
    }
  }
}
}
>>

```



Fine-tuning pedal brackets

The appearance of pedal brackets may be altered in different ways.

```

\paper {
  ragged-right = ##f
}

\relative c'' {
  c2\sostenutoOn c
  c2\sostenutoOff c
  c2\tweak shorten-pair #'(-7 . -2) \sostenutoOn c
  c2\sostenutoOff c
  c2\tweak edge-height #'(0 . 3) \sostenutoOn c
  c2\sostenutoOff c
}

```



Indicating cross-staff chords with a bracket

An non-arpeggiato bracket can indicate that notes on two different staves are to be played with the same hand. In order to do this, the PianoStaff must be set to accept cross-staff brackets.

The following example typesets measure 65 of Debussy's prelude *Les collines d'Anacapri*.

```

\new PianoStaff <<

```

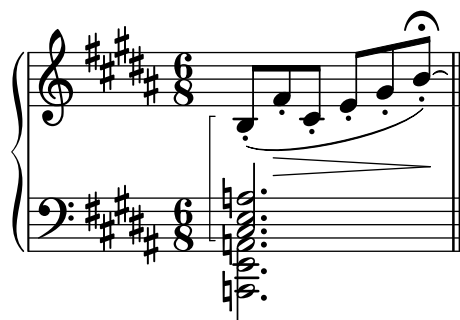
```

\once \set PianoStaff.connectChordBrackets = ##t

\new Staff \relative c' {
  \key b \major
  \time 6/8
  b8-.(\nonArpeggiato fis'-.\> cis-.
    e-. gis-. b-.)\!\fermata^\laissezVibrer
  \section
}

\new Staff \relative c' {
  \clef bass
  \key b \major
  << { <a e cis>2.\nonArpeggiato } \
    { <a, e a,>2. } >>
  \section
}
>>

```



Vorlage für Jazz-Combo

Hier ist ein ziemlich kompliziertes Beispiel für ein Jazz-Ensemble. Achtung: Alle Instrumente sind in `\key c \major` (C-Dur) notiert. Das bezieht sich auf die klingende Musik: LilyPond transponiert die Tonart automatisch, wenn sich die Noten innerhalb eines `ranspose`-Abschnitts befinden.

```

\header {
  title = "Song"
  subtitle = "(tune)"
  composer = "Me"
  meter = "moderato"
  piece = "Swing"
  tagline = \markup \column {
    "LilyPond example file by Amelie Zapf,"
    "Berlin 07/07/2003" }
}

% To make the example display properly in the documentation.
\paper {
  paper-width = 130\mm
  paper-height = 205\mm
}

```



```

% #(set-global-staff-size 16)

\include "english.ly"

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

sl = { \override NoteHead.style = #'slash
       \hide Stem }
nsl = { \revert NoteHead.style
       \undo \hide Stem }
crOn = \override NoteHead.style = #'cross
crOff = \revert NoteHead.style

% Insert chord name style stuff here.

jazzChords = { }

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Keys'n'things %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

global = { \time 4/4 }

Key = { \key c \major }

% ##### Horns #####

% ----- Trumpet -----
trpt = \transpose c d \relative c' {
  \Key
  c1 | c | c |
}
trpHarmony = \transpose c' d {
  \jazzChords
}
trumpet = {
  \global
  \clef treble
  \trpt
}

% ----- Alto Saxophone -----
alto = \transpose c a \relative c' {
  \Key
  c1 | c | c |
}
altoHarmony = \transpose c' a {
  \jazzChords
}
altoSax = {
  \global
  \clef treble

```

```

    \alto
  }

% ----- Baritone Saxophone -----
bari = \transpose c a' \relative c {
  \Key
  c1 | c1 |
  \sl d4^"Solo" d d d \ns1 |
}
bariHarmony = \transpose c' a \chordmode {
  \jazzChords
  s1 | s |
  d2:maj e:m7 |
}
bariSax = {
  \global
  \clef treble
  \bari
}

% ----- Trombone -----
tbone = \relative c {
  \Key
  c1 | c | c |
}
tboneHarmony = \chordmode {
  \jazzChords
}
trombone = {
  \global
  \clef bass
  \tbone
}

% ##### Rhythm Section #####

% ----- Guitar -----
gtr = \relative c'' {
  \Key
  c1 |
  \sl b4 b b b \ns1 |
  c1 |
}
gtrHarmony = \chordmode {
  \jazzChords
  s1 | c2:min7+ d2:maj9 | s1 |
}
guitar = {
  \global
  \clef treble
  \gtr
}

```

```

%% ----- Piano -----
rhUpper = \relative c'' {
  \voiceOne
  \Key
  c1 | c | c |
}
rhLower = \relative c' {
  \voiceTwo
  \Key
  e1 | e | e |
}

lhUpper = \relative c' {
  \voiceOne
  \Key
  g1 | g | g |
}
lhLower = \relative c {
  \voiceTwo
  \Key
  c1 | c | c |
}

PianoRH = {
  \clef treble
  \global
  <<
    \new Voice = "one" \rhUpper
    \new Voice = "two" \rhLower
  >>
}

PianoLH = {
  \clef bass
  \global
  <<
    \new Voice = "one" \lhUpper
    \new Voice = "two" \lhLower
  >>
}

piano = <<
  \new Staff = "upper" \PianoRH
  \new Staff = "lower" \PianoLH
>>

% ----- Bass Guitar -----
Bass = \relative c {
  \Key
  c1 | c | c |
}

```

```

bass = {
  \global
  \clef bass
  \Bass
}

% ----- Drums -----
up = \drummode {
  \voiceOne
  hh4 <hh sn> hh <hh sn> |
  hh4 <hh sn> hh <hh sn> |
  hh4 <hh sn> hh <hh sn> |
}
down = \drummode {
  \voiceTwo
  bd4 s bd s |
  bd4 s bd s |
  bd4 s bd s |
}

drumContents = {
  \global
  <<
    \new DrumVoice \up
    \new DrumVoice \down
  >>
}

%%%%%%%%%% It All Goes Together Here %%%%%%%%%%%

\book { % For the LilyPond documentation.
  \score {
    <<
      \new StaffGroup = "horns" <<
        \new Staff = "trumpet" \with { instrumentName = "Trumpet" }
        \trumpet
        \new Staff = "altosax" \with { instrumentName = "Alto Sax" }
        \altoSax
        \new ChordNames = "barichords" \with { instrumentName = "Bari Sax" }
        \bariHarmony
        \new Staff = "barisax" \with { instrumentName = "Bari Sax" }
        \bariSax
        \new Staff = "trombone" \with { instrumentName = "Trombone" }
        \trombone
      >>

      \new StaffGroup = "rhythm" <<
        \new ChordNames = "chords" \with { instrumentName = "Guitar" }
        \gtrHarmony
        \new Staff = "guitar" \with { instrumentName = "Guitar" }
        \guitar
        \new PianoStaff = "piano" \with {

```

```

        instrumentName = "Piano"
        midiInstrument = "acoustic grand"
    } \piano
    \new Staff = "bass" \with { instrumentName = "Bass" }
        \bass
    \new DrumStaff \with { instrumentName = "Drums" }
        \drumContents
    >>
    >>

\layout {
  \context {
    \Staff
    \RemoveEmptyStaves
  }
  \context {
    \Score
    \override BarNumber.padding = 3
    \override RehearsalMark.padding = 2
    skipBars = ##t
  }
}
\midi { }
}

```

Song

(tune)

Me

moderato
Swing

Trumpet

Alto Sax

Bari Sax

Trombone

Guitar

Piano

Bass

Drums

B^Δ C#m⁷

Cm^Δ D^{Δ9}

LilyPond example file by Amelie Zapf,
Berlin 07/07/2003

Laissez vibrer ties

Laissez vibrer ties have a fixed size. Their positioning can be tuned using the `tie-configuration` property.

See also snippet „Longer laissez vibrer ties“.

```
\relative c' {
  <c e g>4\laissezVibrer r <c f g>\laissezVibrer r
  <c d f g>4\laissezVibrer r <c d f g>4.\laissezVibrer r8

  <c d e f>4\laissezVibrer r
  \override LaissezVibrerTieColumn.tie-configuration
    = #`((-7 . ,DOWN)
```

```

        (-5 . ,DOWN)
        (-3 . ,UP)
        (-1 . ,UP))
    <c d e f>4\laissezVibrer r
}

```



Vorlage für einfache Klaviernotation

Hier ein einfaches Klaviersystem.

```

upper = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

  a4 b c d
}

```

```

lower = \relative c {
  \clef bass
  \key c \major
  \time 4/4

  a2 c
}

```

```

\score {
  \new PianoStaff \with { instrumentName = "Piano" }
  <<
    \new Staff = "upper" \upper
    \new Staff = "lower" \lower
  >>
  \layout { }
  \midi { }
}

```



Piano template with centered lyrics

Anstatt ein eigenes System für Melodie und Text zu schreiben, können Sie den Text auch zwischen die beiden Klaviersysteme schreiben (und damit das zusätzliche System für die Gesangsstimme auslassen).

```

upper = \relative c'' {
  \clef treble

```

```

\key c \major
\time 4/4

a4 b c d
}

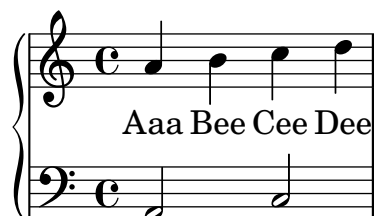
lower = \relative c {
  \clef bass
  \key c \major
  \time 4/4

  a2 c
}

text = \lyricmode {
  Aaa Bee Cee Dee
}

\score {
  \new PianoStaff <<
    \new Staff = upper { \new Voice = "singer" \upper }
    \new Lyrics \lyricsto "singer" \text
    \new Staff = lower { \lower }
  >>
  \layout { }
  \midi { }
}

```



Vorlage für Klavier und Gesangsstimme

Das nächste Beispiel ist typisch für ein Lied: Im oberen System die Melodie mit Text, darunter Klavierbegleitung.

```

melody = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

  a b c d
}

text = \lyricmode {
  Aaa Bee Cee Dee
}

upper = \relative c'' {

```



```

\clef treble
\key c \major
\time 4/4

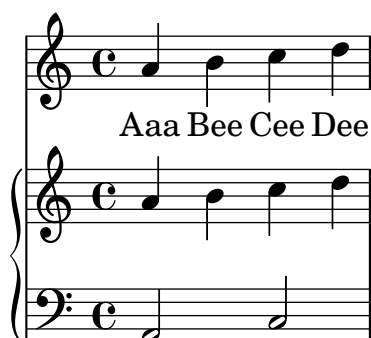
a4 b c d
}

lower = \relative c {
  \clef bass
  \key c \major
  \time 4/4

  a2 c
}

\score {
  <<
    \new Voice = "mel" { \autoBeamOff \melody }
    \new Lyrics \lyricsto mel \text
    \new PianoStaff <<
      \new Staff = "upper" \upper
      \new Staff = "lower" \lower
    >>
  >>
  \layout {
    \context { \Staff \RemoveEmptyStaves }
  }
  \midi { }
}

```



Removing brace on first line of piano score

This snippet removes the first brace from a PianoStaff or a GrandStaff, together with the clefs. It may be useful when cutting and pasting the engraved image into existing music.

The code uses `\alterBroken` to hide the brace delimiter at the beginning.

```

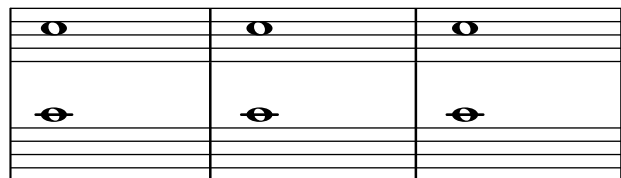
someMusic = {
  \once \omit Staff.Clef
  \once \omit Staff.TimeSignature
  \repeat unfold 3 c1 \break
  \repeat unfold 5 c1 \break
  \repeat unfold 5 c1
}

```

```

\score {
  \new PianoStaff
  <<
    \new Staff = "right" \relative c' { \someMusic
    \new Staff = "left" \relative c' { \clef F \someMusic }
  >>
  \layout {
    indent=75\mm
    \context {
      \PianoStaff
      \alterBroken transparent #'(#t) SystemStartBrace
    }
  }
}

```



Using \autoChange with more than one voice

Here is a demonstration of how to use \autoChange with more than one voice.

```

\score {
  \new PianoStaff
  <<
    \new Staff = "up" {
      <<
        \set Timing.beamExceptions = #'()
        \set Timing.beatStructure = #'(4)
        \new Voice {
          \voiceOne
          \autoChange
          \relative c' {
            g8 a b c d e f g
            g,8 a b c d e f g
          }
        }
      >>
    }
  >>
}

```

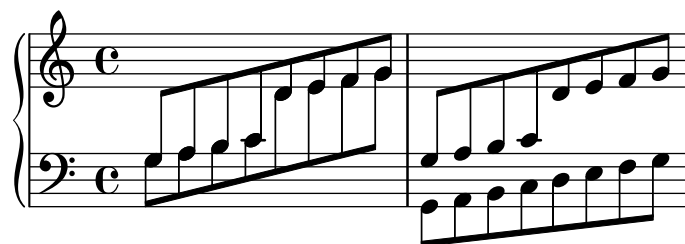
```

    }

    \new Voice {
      \voiceTwo
      \autoChange
      \relative c' {
        g8 a b c d e f g
        g,,8 a b c d e f g
      }
    }
  >>
}

\new Staff = "down" {
  \clef bass
}
>>
}

```



Vocal ensemble template with automatic piano reduction

In diesem Beispiel wird ein automatischer Klavierauszug zu der Chorpartitur hinzugefügt. Das zeigt eine der Stärken von LilyPond – man kann eine Variable mehr als einmal benutzen. Wenn Sie irgendeine Änderung an einer Chorstimme vornehmen, (etwa `tenorMusic`), verändert sich auch der Klavierauszug entsprechend.

```

\paper {
  top-system-spacing.basic-distance = 10
  score-system-spacing.basic-distance = 20
  system-system-spacing.basic-distance = 20
  last-bottom-spacing.basic-distance = 10
}

global = {
  \key c \major
  \time 4/4
}

sopMusic = \relative {
  c''4 c c8[( b)] c4
}

sopWords = \lyricmode {
  hi hi hi hi
}

```

```

altoMusic = \relative {
  e'4 f d e
}
altoWords = \lyricmode {
  ha ha ha ha
}

tenorMusic = \relative {
  g4 a f g
}
tenorWords = \lyricmode {
  hu hu hu hu
}

bassMusic = \relative {
  c4 c g c
}
bassWords = \lyricmode {
  ho ho ho ho
}

\score {
  <<
    \new ChoirStaff <<
      \new Lyrics = "sopranos" \with {
        % This is needed for lyrics above a staff
        \override VerticalAxisGroup.staff-affinity = #DOWN
      }
      \new Staff = "women" <<
        \new Voice = "sopranos" { \voiceOne << \global \sopMusic >> }
        \new Voice = "altos" { \voiceTwo << \global \altoMusic >> }
      >>
      \new Lyrics = "altos"

      \new Lyrics = "tenors" \with {
        % This is needed for lyrics above a staff
        \override VerticalAxisGroup.staff-affinity = #DOWN
      }
      \new Staff = "men" <<
        \clef bass
        \new Voice = "tenors" { \voiceOne << \global \tenorMusic >> }
        \new Voice = "basses" { \voiceTwo << \global \bassMusic >> }
      >>
      \new Lyrics = "basses"

      \context Lyrics = "sopranos" \lyricsto "sopranos" \sopWords
      \context Lyrics = "altos" \lyricsto "altos" \altoWords
      \context Lyrics = "tenors" \lyricsto "tenors" \tenorWords
      \context Lyrics = "basses" \lyricsto "basses" \bassWords
    >>

    \new PianoStaff <<

```

```

\new Staff <<
  \set Staff.printPartCombineTexts = ##f
  \partCombine
  << \global \sopMusic >>
  << \global \altoMusic >>
>>
\new Staff <<
  \clef bass
  \set Staff.printPartCombineTexts = ##f
  \partCombine
  << \global \tenorMusic >>
  << \global \bassMusic >>
>>
>>
>>
}

```

The image displays a musical score for a multi-staff instrument, likely a keyboard or a multi-staff vocal instrument. The score is written in common time (C) and consists of four staves. The first two staves are grouped together by a brace on the left, and the last two staves are also grouped together by a brace on the left. The lyrics are 'hi hi hi hi', 'ha ha ha ha', 'hu hu hu hu', and 'ho ho ho ho'.

The first staff (treble clef) contains the notes: C4, D4, E4, F4, G4, A4, B4, C5. The second staff (bass clef) contains the notes: C3, D3, E3, F3, G3, A3, B3, C4. The third staff (treble clef) contains the notes: C4, D4, E4, F4, G4, A4, B4, C5. The fourth staff (bass clef) contains the notes: C3, D3, E3, F3, G3, A3, B3, C4.

12 Percussion

See also Abschnitt “Percussion” in *Notationsreferenz*.

Adding drum parts

Using the powerful pre-configured tools such as the `\drummode` function and the `DrumStaff` context, inputting drum parts is quite easy: drums are placed at their own staff positions (with a special clef symbol) and have note heads according to the drum. Attaching an extra symbol to the drum or restricting the number of lines is possible.

```

drh = \drummode {
  cymc4.^"crash" hhc16^"h.h." hh hhc8 hho hhc8 hh16 hh
  hhc4 r4 r2
}
drl = \drummode {
  bd4 sn8 bd bd4 << bd ss >>
  bd8 tommh tommh bd toml toml bd tomfh16 tomfh
}
timb = \drummode {
  timh4 ssh timl8 ssh r timh r4
  ssh8 timl r4 cb8 cb
}

\score {
  <<
    \new DrumStaff \with {
      instrumentName = "timbales"
      drumStyleTable = #timbales-style
      \override StaffSymbol.line-count = #2
      \override BarLine.bar-extent = #'(-1 . 1)
    }
    <<
      \timb
    >>
    \new DrumStaff \with { instrumentName = "drums" }
    <<
      \new DrumVoice { \stemUp \drh }
      \new DrumVoice { \stemDown \drl }
    >>
  >>
  \layout { }
  \midi { \tempo 4 = 120 }
}

```

The image shows a musical score for two percussion parts: timbales and drums. The timbales staff is on top, with a C-clef and a common time signature. It contains notes for 'crash' (marked with a cross), 'h.h.' (marked with a plus), and other rhythmic patterns. The drums staff is on the bottom, also with a C-clef and a common time signature. It contains notes for 'bd' (bass drum), 'sn' (snare), 'tom' (tom), and 'cb' (conga), with various rhythmic patterns and stems.

Cow and ride bell example

Two different bells, entered with ,cb‘ (cow bell) and ,rb‘ (ride bell).

```

#(define mydrums '((ridebell default #f 3)
                    (cowbell default #f -2)))

\new DrumStaff \with { instrumentName = #"Different Bells" }

\drummode {
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)
  \set DrumStaff.clefPosition = 0.5
  \override DrumStaff.StaffSymbol.line-positions = #'(-2 3)
  \override Staff.BarLine.bar-extent = #'(-1.0 . 1.5)

  \time 2/4
  rb8 8 cb8 16 rb16-> ~ |
  16 8 16 cb8 8 |
}

```



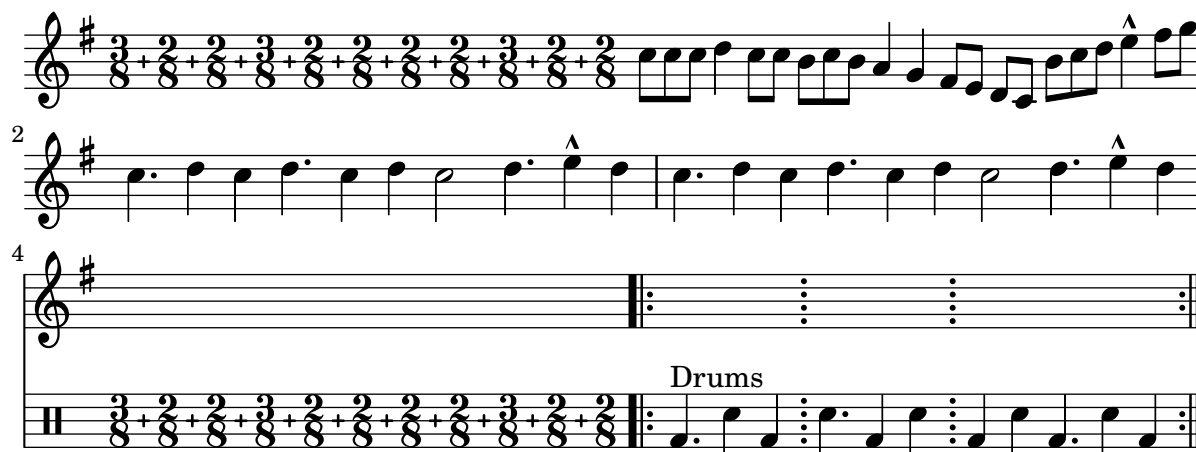
Heavily customized polymetric time signatures

Though the polymetric time signature shown is not the most essential item here, it has been included to show the beat of this piece (which is the template of a real Balkan song, by the way).

```
melody = \relative c'' {
  \key g \major
  \time #'((3 . 8) (2 . 8) (2 . 8) (3 . 8) (2 . 8) (2 . 8)
    (2 . 8) (2 . 8) (3 . 8) (2 . 8) (2 . 8))
  \set Timing.beamExceptions = #'()
  \set Timing.beatStructure = 3,2,2,3,2,2,2,2,3,2,2
  c8 c c d4 c8 c b c b a4 g fis8 e d c b' c d e4-^ fis8 g \break
  c,4. d4 c4 d4. c4 d c2 d4. e4-^ d4
  c4. d4 c4 d4. c4 d c2 d4. e4-^ d4 \break
}

drum = \new DrumStaff \drummode {
  \repeat volta 2 {
    bd4.^{\markup { Drums } sn4 bd \bar "}
    sn4. bd4 sn \bar "
    bd sn bd4. sn4 bd
  }
}

\new Staff {
  \melody
  \drum
}
```



High and low woodblock example

Two Woodblocks, entered with ,wbh' (high woodblock) and ,wbl' (low woodblock). The length of the bar line has been altered with an `\override` command, otherwise it would be too short. The positions of the two staff lines also have to be explicitly defined.

```
% These lines define the position of the woodblocks in the stave;
% if you like, you can change it or you can use special note heads
% for the woodblocks.
```

```
#(define mydrums '((hiwoodblock default #f 3)
                    (lowwoodblock default #f -2)))
```

```
woodstaff = {
  % This defines a staff with only two lines.
  % It also defines the positions of the two lines.
  \override Staff.StaffSymbol.line-positions = #'(-2 3)

  % This is necessary; if not entered,
  % the barline would be too short!
  \override Staff.BarLine.bar-extent = #'(-1.0 . 1.5)
  % small correction for the clef:
  \set DrumStaff.clefPosition = 0.5
}
```

```
\new DrumStaff {
  % with this you load your new drum style table
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)

  \woodstaff

  \drummode {
    \time 2/4
    wbh8 16 16 8-> 8 |
    wbl8 16 16-> ~ 16 16 r8 |
  }
}
```



Vorlage für Jazz-Combo

Hier ist ein ziemlich kompliziertes Beispiel für ein Jazz-Ensemble. Achtung: Alle Instrumente sind in `\key c \major` (C-Dur) notiert. Das bezieht sich auf die klingende Musik: LilyPond transponiert die Tonart automatisch, wenn sich die Noten innerhalb eines `\transpose`-Abschnitts befinden.

```
\header {
  title = "Song"
  subtitle = "(tune)"
  composer = "Me"
  meter = "moderato"
  piece = "Swing"
  tagline = \markup \column {
    "LilyPond example file by Amelie Zapf,"
    "Berlin 07/07/2003" }
}

% To make the example display properly in the documentation.
\paper {
  paper-width = 130\mm
  paper-height = 205\mm
}

% #(set-global-staff-size 16)

\include "english.ly"

%%%%%%%%%%%%%% Some macros %%%%%%%%%%%%%%%

sl = { \override NoteHead.style = #'slash
       \hide Stem }
nsl = { \revert NoteHead.style
       \undo \hide Stem }
crOn = \override NoteHead.style = #'cross
crOff = \revert NoteHead.style

% Insert chord name style stuff here.

jazzChords = { }

%%%%%%%%%%%%%% Keys'n'things %%%%%%%%%%%%%%%

global = { \time 4/4 }

Key = { \key c \major }

% ##### Horns #####

% ----- Trumpet -----
trpt = \transpose c d \relative c' {
```

```

    \Key
    c1 | c | c |
}
trpHarmony = \transpose c' d {
    \jazzChords
}
trumpet = {
    \global
    \clef treble
    \trpt
}

% ----- Alto Saxophone -----
alto = \transpose c a \relative c' {
    \Key
    c1 | c | c |
}
altoHarmony = \transpose c' a {
    \jazzChords
}
altoSax = {
    \global
    \clef treble
    \alto
}

% ----- Baritone Saxophone -----
bari = \transpose c a' \relative c {
    \Key
    c1 | c1 |
    \sl d4^"Solo" d d d \ns1 |
}
bariHarmony = \transpose c' a \chordmode {
    \jazzChords
    s1 | s |
    d2:maj e:m7 |
}
bariSax = {
    \global
    \clef treble
    \bari
}

% ----- Trombone -----
tbone = \relative c {
    \Key
    c1 | c | c |
}
tboneHarmony = \chordmode {
    \jazzChords
}
trombone = {

```

```

\global
\clef bass
\tbone
}

% ##### Rhythm Section #####

% ----- Guitar -----
gtr = \relative c'' {
  \Key
  c1 |
  \sl b4 b b b \ns1 |
  c1 |
}
gtrHarmony = \chordmode {
  \jazzChords
  s1 | c2:min7+ d2:maj9 | s1 |
}
guitar = {
  \global
  \clef treble
  \gtr
}

%% ----- Piano -----
rhUpper = \relative c'' {
  \voiceOne
  \Key
  c1 | c | c |
}
rhLower = \relative c' {
  \voiceTwo
  \Key
  e1 | e | e |
}

lhUpper = \relative c' {
  \voiceOne
  \Key
  g1 | g | g |
}
lhLower = \relative c {
  \voiceTwo
  \Key
  c1 | c | c |
}

PianoRH = {
  \clef treble
  \global
  <<
  \new Voice = "one" \rhUpper

```

```

    \new Voice = "two" \rhLower
  >>
}
PianoLH = {
  \clef bass
  \global
  <<
    \new Voice = "one" \lhUpper
    \new Voice = "two" \lhLower
  >>
}

```

```

piano = <<
  \new Staff = "upper" \PianoRH
  \new Staff = "lower" \PianoLH
>>

```

```

% ----- Bass Guitar -----

```

```

Bass = \relative c {
  \Key
  c1 | c | c |
}
bass = {
  \global
  \clef bass
  \Bass
}

```

```

% ----- Drums -----

```

```

up = \drummode {
  \voiceOne
  hh4 <hh sn> hh <hh sn> |
  hh4 <hh sn> hh <hh sn> |
  hh4 <hh sn> hh <hh sn> |
}
down = \drummode {
  \voiceTwo
  bd4 s bd s |
  bd4 s bd s |
  bd4 s bd s |
}

```

```

drumContents = {
  \global
  <<
    \new DrumVoice \up
    \new DrumVoice \down
  >>
}

```

```

%%%%%%%%%% It All Goes Together Here %%%%%%%%%%%

```

```

\book { % For the LilyPond documentation.
\score {
  <<
    \new StaffGroup = "horns" <<
      \new Staff = "trumpet" \with { instrumentName = "Trumpet" }
      \trumpet
      \new Staff = "altosax" \with { instrumentName = "Alto Sax" }
      \altoSax
      \new ChordNames = "barichords" \with { instrumentName = "Bari Sax" }
      \bariHarmony
      \new Staff = "barisax" \with { instrumentName = "Bari Sax" }
      \bariSax
      \new Staff = "trombone" \with { instrumentName = "Trombone" }
      \trombone
    >>

    \new StaffGroup = "rhythm" <<
      \new ChordNames = "chords" \with { instrumentName = "Guitar" }
      \gtrHarmony
      \new Staff = "guitar" \with { instrumentName = "Guitar" }
      \guitar
      \new PianoStaff = "piano" \with {
        instrumentName = "Piano"
        midiInstrument = "acoustic grand"
      } \piano
      \new Staff = "bass" \with { instrumentName = "Bass" }
      \bass
      \new DrumStaff \with { instrumentName = "Drums" }
      \drumContents
    >>
  >>

\layout {
  \context {
    \Staff
    \RemoveEmptyStaves
  }
  \context {
    \Score
    \override BarNumber.padding = 3
    \override RehearsalMark.padding = 2
    skipBars = ##t
  }
}
\midi { }
}

```

Song

(tune)

Me

moderato
Swing

Trumpet

Alto Sax

Bari Sax

Trombone

Guitar

Piano

Bass

Drums

B^Δ C[#]m⁷

Solo

Cm^Δ D^Δ9

LilyPond example file by Amelie Zapf,
Berlin 07/07/2003

Percussion beaters

Graphic symbols for percussion instruments are not natively supported; however it is possible to include such symbols, either as an external EPS file or as embedded PostScript code inside a markup, as demonstrated in this example.

```
stick = \markup \with-dimensions #'(0.80 . 5.2) #'(0.85 . 5.2) {
  \postscript "
    0 6 translate
    0.8 -0.8 scale
    0 0 0 setrgbcolor
    [] 0 setdash
    1 setlinewidth
```

```

0 setlinejoin
0 setlinecap
gsave [1 0 0 1 0 0] concat
gsave [1 0 0 1 -3.5406095 -199.29342] concat
gsave
0 0 0 setrgbcolor
newpath
7.1434065 200.94354 moveto
7.2109628 200.90454 7.2785188 200.86554 7.3460747 200.82654 curveto
8.2056347 202.31535 9.0651946 203.80414 9.9247546 205.29295 curveto
9.8571989 205.33195 9.7896429 205.37095 9.7220864 205.40996 curveto
8.8625264 203.92115 8.0029664 202.43233 7.1434065 200.94354 curveto
closepath
eofill
grestore
gsave
0 0 0 setrgbcolor
newpath
4.9646672 203.10444 moveto
5.0036707 203.03688 5.0426744 202.96933 5.0816777 202.90176 curveto
6.5704792 203.76133 8.0592809 204.6209 9.5480824 205.48045 curveto
9.5090791 205.54801 9.4700754 205.61556 9.4310717 205.68311 curveto
7.94227 204.82356 6.4534687 203.96399 4.9646672 203.10444 curveto
closepath
eofill
grestore
gsave
<<
/ShadingType 3
/ColorSpace /DeviceRGB
/Coords [113.13708 207.87465 0 113.13708 207.87465 16.162441]
/Extend [true true]
/Domain [0 1]
/Function <<
/FunctionType 3
/Functions
[
<<
/FunctionType 2
/Domain [0 1]
/C0 [1 1 1]
/C1 [0.72941178 0.72941178 0.72941178]
/N 1
>>
]
/Domain [0 1]
/Bounds [ ]
/Encode [ 0 1 ]
>>
>>
newpath
7.6422017 200.76488 moveto

```

```

7.6505696 201.02554 7.3905363 201.24867 7.1341335 201.20075 curveto
6.8759501 201.16916 6.6949602 200.87978 6.7801462 200.63381 curveto
6.8480773 200.39155 7.1438307 200.25377 7.3728389 200.35861 curveto
7.5332399 200.42458 7.6444521 200.59122 7.6422017 200.76488 curveto
closepath
clip
gsave [
  0.052859054 0.063089841 -0.020912282 0.017521108 5.7334261 189.76443
] concat
shfill
grestore
grestore
0 0 0 setrgbcolor
[] 0 setdash
0.027282091 setlinewidth
0 setlinejoin
0 setlinecap
newpath
7.6422017 200.76488 moveto
7.6505696 201.02554 7.3905363 201.24867 7.1341335 201.20075 curveto
6.8759501 201.16916 6.6949602 200.87978 6.7801462 200.63381 curveto
6.8480773 200.39155 7.1438307 200.25377 7.3728389 200.35861 curveto
7.5332399 200.42458 7.6444521 200.59122 7.6422017 200.76488 curveto
closepath
stroke
gsave
<<
/ShadingType 3
/ColorSpace /DeviceRGB
/Coords [113.13708 207.87465 0 113.13708 207.87465 16.162441]
/Extend [true true]
/Domain [0 1]
/Function <<
/FunctionType 3
/Functions
[
<<
/FunctionType 2
/Domain [0 1]
/C0 [1 1 1]
/C1 [0.72941178 0.72941178 0.72941178]
/N 1
>>
]
/Domain [0 1]
/Bounds [ ]
/Encode [ 0 1 ]
>>
>>
newpath
5.2721217 202.83181 moveto
5.2804896 203.09247 5.0204563 203.3156 4.7640539 203.26768 curveto

```

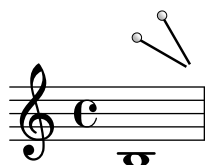


```

4.5058701 203.23609 4.3248803 202.94671 4.4100662 202.70074 curveto
4.4779975 202.45848 4.7737511 202.3207 5.0027593 202.42554 curveto
5.1631598 202.49149 5.2743721 202.65813 5.2721217 202.83181 curveto
closepath
clip
gsave [
  0.052859054 0.063089841 -0.020912282 0.017521108 3.363346 191.83136
] concat
shfill
grestore
grestore
0 0 0 setrgbcolor
[] 0 setdash
0.027282091 setlinewidth
0 setlinejoin
0 setlinecap
newpath
5.2721217 202.83181 moveto
5.2804896 203.09247 5.0204563 203.3156 4.7640539 203.26768 curveto
4.5058701 203.23609 4.3248803 202.94671 4.4100662 202.70074 curveto
4.4779975 202.45848 4.7737511 202.3207 5.0027593 202.42554 curveto
5.1631598 202.49149 5.2743721 202.65813 5.2721217 202.83181 curveto
closepath
stroke
grestore
grestore
"
}

\score {
  b1^\stick
}

```



Percussion example

A short example taken from Stravinsky's *L'histoire du Soldat*.

```

#(define mydrums '((bassdrum default #f 4)
                    (snare      default #f -4)
                    (tambourine default #f 0)))

```

```

U = \stemUp
D = \stemDown

```

```

global = {
  \time 3/8 s4.
  \time 2/4 s2*2
  \time 3/8 s4.
}

```

```

\time 2/4 s2
}

drumsA = {
  \context DrumVoice <<
    \global
    \drummode {
      \autoBeamOff
      \D sn8 \U tamb s |
      sn4 \D sn4 |
      \U tamb8 \D sn \U sn16 \D sn \U sn8 |
      \D sn8 \U tamb s |
      \U sn4 s8 \U tamb
    }
  >>
}

drumsB = \drummode {
  s4 bd8 s2*2 s4 bd8 s4 bd8 s
}

\layout {
  indent = 40\mm
  \context {
    \DrumStaff
    drumStyleTable = #(alist->hash-table mydrums)
  }
}

\score {
  \new StaffGroup <<
    \new DrumStaff \with {
      instrumentName = \markup \center-column {
        "Tambourine"
        "et"
        "caisse claire s. timbre" }
    } \drumsA
    \new DrumStaff \with {
      instrumentName = "Grosse Caisse"
    } \drumsB
  >>
}

```

Tambourine
et
caisse claire s. timbre

Grosse Caisse

Printing music with different time signatures

In the following snippet, two parts have a completely different time signature, yet remain synchronized.

The bar lines can no longer be printed at the Score level; to allow independent bar lines in each part, the `Default_barline_engraver` and `Timing_translator` are moved from the Score context to the Staff context.

If bar numbers are required, the `Bar_number_engraver` should also be moved, since it relies on properties set by the `Timing_translator`; a `\with` block can be used to add bar numbers to the relevant staff.

```
global = {
  \time 3/4 s2.*3 \break
  s2.*3
}

\layout {
  \context {
    \Score
    \remove "Timing_translator"
    \remove "Bar_number_engraver"
    \override SpacingSpanner.uniform-stretching = ##t
    \override SpacingSpanner.strict-note-spacing = ##t
    \proportionalNotationDuration = #1/64
  }
  \context {
    \Staff
    \consists "Timing_translator"
  }
  \context {
    \Voice
    \remove "Forbid_line_break_engraver"
    \tupletFullLength = ##t
  }
}

Bassklarinette = \new Staff \with {
  \consists "Bar_number_engraver"
  \barNumberVisibility = #(every-nth-bar-number-visible 2)
  \override BarNumber.break-visibility = #end-of-line-invisible
} <<
\global
{
  \clef treble
  \time 3/8 d''4. |
  \time 3/4 r8 des''2( c''8) |
  \time 7/8 r4. ees''2 ~ |
  \time 2/4 \tupletUp \tuplet 3/2 { ees''4 r4 d''4 ~ } |
  \time 3/8 \tupletUp \tuplet 4/3 { d''4 r4 } |
  \time 2/4 e''2 |
  \time 3/8 es''4. |
  \time 3/4 r8 d''2 r8 |
}
```

>>

```
Perkussion = \new StaffGroup <<
```

```
  \new Staff <<
```

```
    \global
```

```
    {
```

```
      \clef percussion
```

```
      \time 3/4 r4 c'2 ~ |
```

```
      c'2. |
```

```
      R2. |
```

```
      r2 g'4 ~ |
```

```
      g'2. ~ |
```

```
      g'2. |
```

```
    }
```

>>

```
  \new Staff <<
```

```
    \global {
```

```
      \clef percussion
```

```
      \time 3/4 R2. |
```

```
      g'2. ~ |
```

```
      g'2. |
```

```
      r4 g'2 ~ |
```

```
      g'2 r4 |
```

```
      g'2. |
```

```
    }
```

>>

>>

```
\score {
```

```
  <<
```

```
    \Bassklarinette
```

```
    \Perkussion
```

>>

```
}
```

The image shows a musical score for two parts: Bass Clarinet and Percussion. The Bass Clarinet part is written on a single staff with a treble clef and a 3/4 time signature. It features a melody with slurs and fingerings (2, 4, 3). The Percussion part is written on two staves with a common time signature (C) and a 3/4 time signature. It features a rhythmic pattern with slurs and rests.

(4)

3 4 6

8

Tam-tam example

A tam-tam example, entered with ,tt'.

```
#(define mydrums '((tamtam default #f 0)))
```

```
\new DrumStaff \with { instrumentName = #"Tamtam" }
```

```
\drummode {
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)
  \override Staff.StaffSymbol.line-positions = #'( 0 )
  \override Staff.BarLine.bar-extent = #'(-1.5 . 1.5)
```

```
tt 1 \pp \laissezVibrer
}
```

Tamtam *pp*

Tambourine example

A tambourine example, entered with ,tamb'.

```
#(define mydrums '((tambourine default #f 0)))
```

```
\new DrumStaff \with { instrumentName = #"Tambourine" }
```

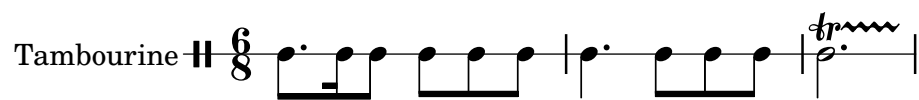
```
\drummode {
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)
  \override Staff.StaffSymbol.line-positions = #'( 0 )
  \override Staff.BarLine.bar-extent = #'(-1.5 . 1.5)
```

```
\time 6/8
```

```

tamb8. 16 8 8 8 8 |
tamb4. 8 8 8 |
% The trick with the scaled duration and the shorter rest
% is neccessary for the correct ending of the trill-span!
tamb2.*5/6 \startTrillSpan s8 \stopTrillSpan |
}

```



13 Fretted string instruments

See also Abschnitt “Fretted string instruments” in *Notationsreferenz*.

Adding fingerings to a score

Fingering instructions can be entered using a simple syntax.

```
\relative c' ' {
  c4-1 d-2 f-4 e-3
}
```

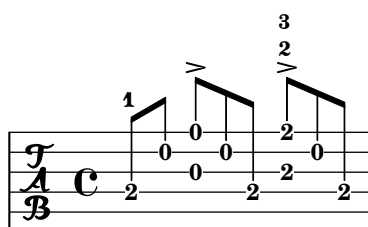


Adding fingerings to tablatures

To add fingerings to tablatures, use a combination of `\markup` and `\finger`.

```
one = \markup { \finger 1 }
two = \markup { \finger 2 }
threeTwo = \markup {
  \override #'(baseline-skip . 2)
  \column {
    \finger 3
    \finger 2
  }
}

\score {
  \new TabStaff {
    \tabFullNotation
    \stemUp
    e8\4~\one b\2 <g\3 e'\1>~>[ b\2 e\4]
    <a\3 fis'\1>~>~\threeTwo[ b\2 e\4]
  }
}
```



Adding markups in a tablature

By default, markups are not displayed in a tablature.

To make them appear, revert the stencil property of the TextScript grob in the TabStaff context.

```
high = { r4 r8 <g c'> q r8 r4 }
low = { c4 r4 c8 r8 g,8 b, }
pulse = { s8~"1" s~"&" s~"2" s~"&" s~"3" s~"&" s~"4" s~"&" }
```

```

\score {
  \new TabStaff {
    \repeat unfold 2 << \high \ \ \low \ \ \pulse >>
  }
  \layout {
    \context {
      \TabStaff
      \clef moderntab
      \revert TextScript.stencil
      \override TextScript.font-series = #'bold
      \override TextScript.font-size = #-2
      \override TextScript.color = #red
    }
    \context {
      \Score
      proportionalNotationDuration = #1/8
    }
  }
}

```

	1	&	2	&	3	&	4	&	1	&	2	&	3	&	4	&
T					1-1								1-1			
A					0-0								0-0			
B	3					3				3				3		2
							3							3		

Fingersatz auch innerhalb des Systems setzen

Normalerweise werden vertikal orientierte Fingersatzzahlen außerhalb des Systems gesetzt. Das kann aber verändert werden.

```

\relative c' {
  <c-1 e-2 g-3 b-5>2
  \override Fingering.staff-padding = #'()
  <c-1 e-2 g-3 b-5>4 g'-0
  a8[-1 b]-2 g-0 r
  \override Fingering.add-stem-support = ##f
  a[-1 b]-2 g-0 r
  \override Fingering.add-stem-support = ##t
  a[-1 b]-2 g-0 r
  \override Fingering.add-stem-support = #only-if-beamed
  a[-1 b]-2 g-0 r
}

```

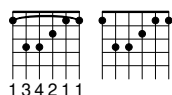


Automatic fretboards barré

When automatic fretboards are used, barré indicators are drawn whenever one finger is responsible for multiple strings.

If no finger indications are given in the chord from which the automatic fretboard is created, no barré indicators are included, because there is no way to identify where barrés should be placed.

```
\new FretBoards {
  <f,-1 c-3 f-4 a-2 c'-1 f'-1>1
  <f, c f a c' f'>1
}
```

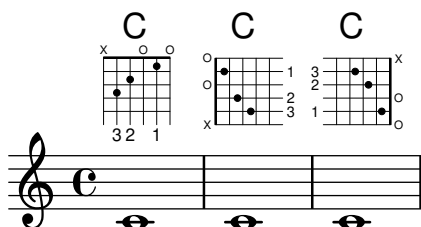


Changing fret orientations

Fret diagrams can be oriented in three ways. By default the top string or fret in the different orientations will be aligned.

```
\include "predefined-guitar-fretboards.ly"
```

```
<<
\chords {
  c1
  c1
  c1
}
\new FretBoards \chordmode {
  c1
  \override FretBoard.fret-diagram-details.orientation =
    #'landscape
  c1
  \override FretBoard.fret-diagram-details.orientation =
    #'opposing-landscape
  c1
}
\new Voice {
  c'1
  c'1
  c'
}
>>
```



Chord changes for fretboards

Fretboards can be set to display only when the chord changes, or at the beginning of a new line.

```
\include "predefined-guitar-fretboards.ly"
```

```
myChords = \chordmode {
```

```

c1 c1 \break
\set chordChanges = ##t
c1 c1 \break
c1 c1
}

<<
\new ChordNames { \myChords }
\new FretBoards { \myChords }
\new Staff { \myChords }
>>

```

The image displays three musical staves, each with a treble clef and a C-clef. Above each staff is a guitar fretboard diagram for a C major chord. The first staff has a '3' above the first measure, the second a '3', and the third a '5'. Each staff shows a glissando from a C major chord to a C major chord, with the notes moving up the fretboard.

Akkordglissando in Tabulaturen

Gleiten von Akkorden kann sowohl im normalen Notensystem als auch in einer Tabulatur notiert werden. Saitennummern werden für Tabulaturen benötigt, weil die automatische Saitenberechnung unterschiedlich für Akkorde und einzelne Noten funktioniert.

```

myMusic = \relative c' {
  <c e g>1 \glissando <f a c>
  <cis, eis gis>1 \glissando <f a c>
  <cis eis gis>1 \glissando <f a c\3>
}

\score {
  <<
    \new Staff {
      \clef "treble_8"
      \omit StringNumber
      \myMusic
    }
    \new TabStaff \myMusic
  }
}

```

```

>>
}

\score {
  <<
    \new Staff {
      \clef "treble_8"
      \omit StringNumber
      \myMusic
    }
    \new TabStaff \with { \override Glissando.style = #'none } {
      \myMusic
    }
  >>
}

```

Chords with stretched fingering for FretBoards and TabVoice

Sometimes chords with a stretched fingering are required. If not otherwise specified the context property `maximumFretStretch` is set to value 4, though, resulting in a warning about „No string for pitch ...“, and the note is omitted. You may set `maximumFretStretch` to an appropriate value or explicitly assign string numbers to all notes of a chord to fix that.

```

% The code below prints two warnings for the second chord,
% which may be omitted by uncommenting the following line.
%
% #(for-each (lambda (x) (ly:expect-warning "No string for pitch")) (iota 2))

```

```

mus = {
  <c' bes'>
  <c'\2 bes'>
  \set maximumFretStretch = 5
  <c' bes'>
  <c'\2 bes'\1>
}

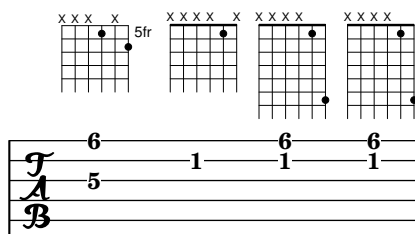
```

```

<<
  \new FretBoards \mus

```

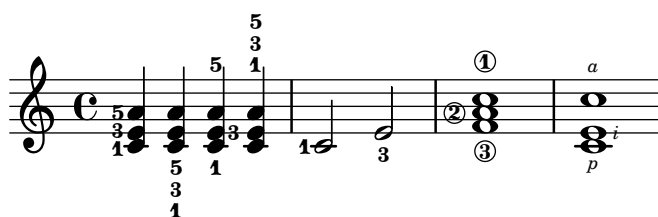
```
\new TabVoice \mus
>>
```



Position von Fingersatz in Akkorden kontrollieren

Die Position von Fingersatzzahlen kann exakt kontrolliert werden.

```
\relative c' {
  \set fingeringOrientations = #'(left)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down right up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(left)
  <c-1>2
  \set fingeringOrientations = #'(down)
  <e-3>2
  \set stringNumberOrientations = #'(up left down)
  <f\3 a\2 c\1>1
  \set strokeFingerOrientations = #'(down right up)
  <c\rightHandFinger 1 e\rightHandFinger 2 c'\rightHandFinger 4 >
}
```



Bunddiagramme anpassen

Eigenschaften von Bunddiagrammen können in 'fret-diagram-details verändert werden. Einstellungen mit dem `\override`-Befehl werden dem `FretBoards.FretBoard`-Objekt zugewiesen. Genauso wie `Voice` ist auch `FretBoards` ein Kontext der niedrigsten Ebene, weshalb der Kontext auch in dem Befehl weggelassen werden kann.

```
\include "predefined-guitar-fretboards.ly"

\storePredefinedDiagram #default-fret-table \chordmode { c' }
                        #guitar-tuning
                        "x;1-1-(;3-2;3-3;3-4;1-1-);"
```

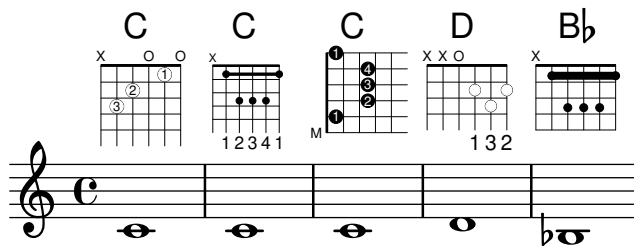
% shorthand

```

oo = #(define-music-function
  (grob-path value)
  (list? scheme?)
  #{ \once \override $grob-path = #value #})

<<
\new ChordNames {
  \chordmode { c1 | c | c | d | bes }
}
\new FretBoards {
  % Set global properties of fret diagram
  \override FretBoards.FretBoard.size = 1.2
  \override FretBoard.fret-diagram-details.finger-code = #'in-dot
  \override FretBoard.fret-diagram-details.dot-color = #'white
  \chordmode {
    c
    \oo FretBoard.size #1.0
    \oo FretBoard.fret-diagram-details.barre-type #'straight
    \oo FretBoard.fret-diagram-details.dot-color #'black
    \oo FretBoard.fret-diagram-details.finger-code #'below-string
    c'
    \oo FretBoard.fret-diagram-details.barre-type #'none
    \oo FretBoard.fret-diagram-details.number-type #'arabic
    \oo FretBoard.fret-diagram-details.orientation #'landscape
    \oo FretBoard.fret-diagram-details.mute-string "M"
    \oo FretBoard.fret-diagram-details.label-dir #LEFT
    \oo FretBoard.fret-diagram-details.dot-color #'black
    c'
    \oo FretBoard.fret-diagram-details.finger-code #'below-string
    \oo FretBoard.fret-diagram-details.dot-radius #0.35
    \oo FretBoard.fret-diagram-details.dot-position #0.5
    \oo FretBoard.fret-diagram-details.fret-count #3
    d
    \oo FretBoard.fret-diagram-details.barre-type #'straight
    \oo FretBoard.fret-diagram-details.finger-code #'none
    \oo FretBoard.fret-diagram-details.dot-radius #0.25
    \oo FretBoard.fret-diagram-details.dot-color #'black
    \oo FretBoard.fret-diagram-details.string-overhang #0.
    \oo FretBoard.fret-diagram-details.barre-thickness #2.
    bes
  }
}
\new Voice {
  c'1 | c' | c' | d' | bes
}
>>

```



Anpassung von Beschriftungs-Bunddiagrammen

Bunddiagramme können mit der Eigenschaft 'fret-diagram-details' angepasst werden. Bunddiagramme, die als Textbeschriftung eingefügt werden, können Veränderungen im Voice.TextScript-Objekt oder direkt in der Beschriftung vorgenommen werden.

<<

```
\chords { c1 | c | c | d }

\new Voice = "mel" {
  \textLengthOn
  % Set global properties of fret diagram
  \override TextScript.size = 1.2
  \override TextScript.fret-diagram-details.finger-code = #'in-dot
  \override TextScript.fret-diagram-details.dot-color = #'white

  %% C major for guitar, no barre, using defaults
  % terse style
  c'1~\markup { \fret-diagram-terse "x;3-3;2-2;o;1-1;o;" }

  %% C major for guitar, barred on third fret
  % verbose style
  % size 1.0
  % roman fret label, finger labels below string, straight barre
  c'1~\markup {
    % standard size
    \override #'(size . 1.0) {
      \override #'(fret-diagram-details . (
        (number-type . roman-lower)
        (finger-code . in-dot)
        (barre-type . straight))) {
        \fret-diagram-verbose #'((mute 6)
          (place-fret 5 3 1)
          (place-fret 4 5 2)
          (place-fret 3 5 3)
          (place-fret 2 5 4)
          (place-fret 1 3 1)
          (barre 5 1 3))
      }
    }
  }

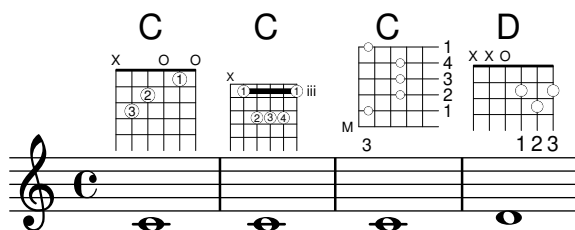
  %% C major for guitar, barred on third fret
  % verbose style
  % landscape orientation, arabic numbers, M for mute string
```

```

% no barre, fret label down or left, small mute label font
c'1^\markup {
  \override #'(fret-diagram-details . (
    (finger-code . below-string)
    (number-type . arabic)
    (label-dir . -1)
    (mute-string . "M")
    (orientation . landscape)
    (barre-type . none)
    (xo-font-magnification . 0.4)
    (xo-padding . 0.3))) {
    \fret-diagram-verbose #'((mute 6)
      (place-fret 5 3 1)
      (place-fret 4 5 2)
      (place-fret 3 5 3)
      (place-fret 2 5 4)
      (place-fret 1 3 1)
      (barre 5 1 3))
  }
}

%% simple D chord
% terse style
% larger dots, centered dots, fewer frets
% label below string
d'1^\markup {
  \override #'(fret-diagram-details . (
    (finger-code . below-string)
    (dot-radius . 0.35)
    (dot-position . 0.5)
    (fret-count . 3))) {
    \fret-diagram-terse "x;x;o;2-1;3-2;2-3;"
  }
}
}
>>

```



Fingersatz, Saitennummern und Fingersatz für die rechte Hand

Dieses Beispiel kombiniert Fingersatz für die linke Hand, Saitennummern und Fingersatz für die rechte Hand.

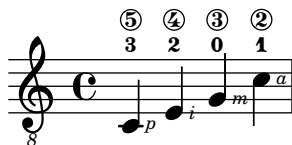
```
#(define RH rightHandFinger)
```

```
\relative c {
  \clef "treble_8"
```

```

<c-3\5\RH 1 >4
<e-2\4\RH 2 >4
<g-0\3\RH 3 >4
<c-1\2\RH 4 >4
}

```



Flamenco notation

For flamenco guitar, some special notation is used.

- A *golpe* symbol indicates a slap on the guitar body with the nail of the ring finger.
- An arrow indicates (the direction of) strokes.
- Different letters for fingering are used („p“: thumb, „i“: index finger, „m“: middle finger, „a“: ring finger and „x“: little finger).
- Marking 3- and 4-finger *rasgueados*: stroke upwards with all fingers, ending with an up- and down using the index finger.
- *Abanicos* are strokes (in tuples) with thumb (down), little and index finger (both up). There's also an *abanico 2* where middle and ring finger are used instead of the little finger.
- *Alza pua* indicates fast playing with the thumb.

Most figures use arrows in combination with fingering; with abanicos and rasgueados, note heads are printed only for the first chord.

This snippet contains some header-like code that can be copied as `flamenco.ly` and included in source files.

```
%%%%%%%% Cut here ----- Start of `flamenco.ly`.
```

```
% Text indicators.
```

```
abanico = ^\markup \small { \italic Abanico }
rasgueado = ^\markup \small { \italic Ras. }
alzapua = ^\markup \small { \italic Alzapua }
```

```
% Finger stroke symbols.
```

```
strokeUp = \markup {
  \combine
    \override #'(thickness . 1.3) \draw-line #'(0 . 2)
    \raise #2 \arrow-head #Y #UP ##f }
strokeDown = \markup {
  \combine
    \arrow-head #Y #DOWN ##f
    \override #'(thickness . 1.3) \draw-line #'(0 . 2) }
```

```
% Golpe symbol.
```

```
golpe = \markup {
  \filled-box #'(0 . 1) #'(0 . 1) #0
  \hspace #-1.6
  \with-color #white
  \filled-box #'(0.15 . 0.85) #'(0.15 . 0.85) #0
}
```



```

}

% Strokes, fingers, and golpe command.
RHp = \rightHandFinger #1
RHi = \rightHandFinger #2
RHm = \rightHandFinger #3
RHa = \rightHandFinger #4
RHx = \rightHandFinger #5
RHu = \rightHandFinger \strokeUp
RHd = \rightHandFinger \strokeDown
RHg = \rightHandFinger \golpe

% Various shorthands.
tupletOff = {
  \once \omit TupletNumber
  \once \omit TupletBracket
}

tupletsOff = {
  \omit TupletNumber
  \override TupletBracket.bracket-visibility = #'if-no-beam
}

tupletsOn = {
  \override TupletBracket.bracket-visibility = #'default
  \undo \omit TupletNumber
}

headsOff = {
  \hide TabNoteHead
  \hide NoteHead
  \override NoteHead.no-ledgers = ##t
}

headsOn = {
  \override TabNoteHead.transparent = ##f
  \override NoteHead.transparent = ##f
  \override NoteHead.no-ledgers = ##f
}

%%%%%%%% Cut here ----- End of `flamenco.ly`.

part = \relative c' {
  \set strokeFingerOrientations = #'(up)
  \key a\major

  <a, e' a cis e\RHu\RHi>8
    <a e' a cis e\RHd\RHi>8
    r4
    r2^\markup\golpe |
  <a e' a cis e\RHu\RHi>8

```

```

    <a e' a cis e\R Hd\R Hi>8
    <a e' a cis e\R Hu\R Hi\R Hg>8
    <a e' a cis e\R Hd\R Hi>8
    r2 |
    <a e' a cis e\R Hu\R Ha>16\rasgueado
    \headsOff
    <a e' a cis e\R Hu\R Hm>
    <a e' a cis e\R Hu\R Hi>
    <a e' a cis e\R Hd\R Hi>~
    \headsOn
    <a e' a cis e>2
    r4 |
    \tupletOff
    \tuplet 5/4 {
      <a e' a cis e\R Hu\R Hx>16\rasgueado
      \headsOff
      <a e' a cis e\R Hu\R Ha>
      <a e' a cis e\R Hu\R Hm>
      <a e' a cis e\R Hu\R Hi>
      <a e' a cis e\R Hd\R Hi>~
      \headsOn
    }
    <a e' a cis e>2
    r4 |
    <>\abanico
    \tupletsOff
    \repeat unfold 4 {
      \tuplet 3/2 {
        <a e' a cis e\R Hd\R Hp>8
        \headsOff
        <a e' a cis e\R Hu\R Hx>
        <a e' a cis e\R Hu\R Hi>
        \headsOn
      }
    }
    \tupletsOff |
    <>\alzapua
    \override Beam.positions = #'(2 . 2)
    \repeat unfold 4 {
      \tuplet 3/2 {
        a8\R Hp
        <e' a\R Hu\R Hg>
        <e a\R Hd>
      }
    }
    \tupletsOn |
    <a, e' a\R Hu\R Hm>1 \bar "|."
  }

\score {
  \new StaffGroup <<
    \context Staff = "part" {

```

```

\clef "G_8"
\part
}
\context TabStaff {
  \part
}
}
>>
\layout {
  ragged-right = ##t
}
}

```

The image displays two musical staves with fret diagrams for guitar. The top staff is in G major (one sharp) and 8/8 time, featuring chords with fingerings (i, x, i) and rasgueado (Ras.) markings. The bottom staff is in G major (one sharp) and 5/8 time, featuring chords with fingerings (p, x, i) and rasgueado (Ras.) markings. Both staves have four strings shown with fret numbers (0, 2, 2, 0) and fingerings (i, x, i) or (p, x, i).

Fret diagrams explained and developed

This snippet shows many possibilities for obtaining and tweaking fret diagrams.

```

<<
\chords {
  a1 a \bar "||" \break
  \repeat unfold 3 {
    c c c d d \bar "||" \break
  }
}

\new Voice {
  % Set global properties of fret diagram
  \override TextScript.size = 1.2
  \override TextScript.fret-diagram-details
    .finger-code = #'below-string
  \override TextScript.fret-diagram-details
    .dot-color = #'black
}

```

```

% 1
%
% A chord for ukulele.
a'1^\markup
  \override #'(fret-diagram-details
    . ((string-count . 4)
      (dot-color . white)
      (finger-code . in-dot)))
  \fret-diagram "4-2-2;3-1-1;2-o;1-o;"

% 2
%
% A chord for ukulele, with formatting defined in definition
% string: 1.2 * size, 4 strings, 4 frets, fingerings below,
% string dot radius .35 of fret spacing, dot position 0.55 of
% fret spacing.
a'1^\markup
  \override #'(fret-diagram-details
    . ((dot-color . white)
      (open-string . "o")))
  \fret-diagram
    "s:1.2;w:4;h:3;f:2;d:0.35;p:0.55;4-2-2;3-1-1;2-o;1-o;"

%%
%% These chords will be in normal orientation
%%

% 3
%
% C major for guitar, barred on third fret: verbose style,
% roman fret label, finger labels below string, straight barre.
c'1^\markup
  % 110% of default size
  \override #'(size . 1.1)
  \override #'(fret-diagram-details
    . ((number-type . roman-lower)
      (finger-code . below-string)
      (barre-type . straight)))
  \fret-diagram-verbose #'((mute 6)
    (place-fret 5 3 1)
    (place-fret 4 5 2)
    (place-fret 3 5 3)
    (place-fret 2 5 4)
    (place-fret 1 3 1)
    (barre 5 1 3))

% 4
%
% C major for guitar, barred on third fret: double barre used

```

```

% to test barre function, verbose style.
c'1^\markup
% 110% of default size
\override #'(size . 1.1)
\override #'(fret-diagram-details
. ((number-type . arabic)
(dot-label-font-mag . 0.9)
(finger-code . in-dot)
(fret-label-font-mag . 0.6)
(fret-label-vertical-offset . 0)
(label-dir . -1)
(mute-string . "M")
(xo-font-magnification . 0.4)
(xo-padding . 0.3)))
\fret-diagram-verbose #'((mute 6)
(place-fret 5 3 1)
(place-fret 4 5 2)
(place-fret 3 5 3)
(place-fret 2 5 4)
(place-fret 1 3 1)
(barre 4 2 5)
(barre 5 1 3))

% 5
%
% C major for guitar, with capo on third fret: verbose style.
c'1^\markup
% 110% of default size
\override #'(size . 1.1)
\override #'(fret-diagram-details
. ((number-type . roman-upper)
(dot-label-font-mag . 0.9)
(finger-code . none)
(fret-label-vertical-offset . 0.5)
(xo-font-magnification . 0.4)
(xo-padding . 0.3)))
\fret-diagram-verbose #'((mute 6)
(capo 3)
(open 5)
(place-fret 4 5 1)
(place-fret 3 5 2)
(place-fret 2 5 3)
(open 1))

% 6
%
% Simple D chord.
d'1^\markup
\override #'(fret-diagram-details
. ((finger-code . below-string)
(dot-radius . 0.35)
(string-thickness-factor . 0.3)

```

```

        (dot-position . 0.5)
        (fret-count . 3)))
\fret-diagram-terse "x;x;o;2-1;3-2;2-3;"

% 7
%
% Simple D chord, large top fret thickness.
d'1^\markup
  \override #'(fret-diagram-details
    . ((finger-code . below-string)
      (dot-radius . 0.35)
      (dot-position . 0.5)
      (top-fret-thickness . 7)
      (fret-count . 3)))
  \fret-diagram-terse "x;x;o;2-1;3-2;2-3;"

%%
%% These chords will be in landscape orientation
%%
\override TextScript.fret-diagram-details
  .orientation = #'landscape

% 8
%
% C major for guitar, barred on third fret: verbose style,
% roman fret label, finger labels below string, straight
% barre.
c'1^\markup
  % 110% of default size
  \override #'(size . 1.1)
  \override #'(fret-diagram-details
    . ((number-type . roman-lower)
      (finger-code . below-string)
      (barre-type . straight)))
  \fret-diagram-verbose #'(mute 6)
    (place-fret 5 3 1)
    (place-fret 4 5 2)
    (place-fret 3 5 3)
    (place-fret 2 5 4)
    (place-fret 1 3 1)
    (barre 5 1 3))

% 9
%
% C major for guitar, barred on third fret: Double barre
% used to test barre function, verbose style.
c'1^\markup
  % 110% of default size
  \override #'(size . 1.1)
  \override #'(fret-diagram-details

```

```

        . ((number-type . arabic)
          (dot-label-font-mag . 0.9)
          (finger-code . in-dot)
          (fret-label-font-mag . 0.6)
          (fret-label-vertical-offset . 0)
          (label-dir . -1)
          (mute-string . "M")
          (xo-font-magnification . 0.4)
          (xo-padding . 0.3)))
\ fret-diagram-verbose #'((mute 6)
                          (place-fret 5 3 1)
                          (place-fret 4 5 2)
                          (place-fret 3 5 3)
                          (place-fret 2 5 4)
                          (place-fret 1 3 1)
                          (barre 4 2 5)
                          (barre 5 1 3))

% 10
%
% C major for guitar, with capo on third fret: verbose style.
c'1~\markup
% 110% of default size
\override #'(size . 1.1)
\override #'(fret-diagram-details
  . ((number-type . roman-upper)
    (dot-label-font-mag . 0.9)
    (finger-code . none)
    (fret-label-vertical-offset . 0.5)
    (xo-font-magnification . 0.4)
    (xo-padding . 0.3)))
\ fret-diagram-verbose #'((mute 6)
                          (capo 3)
                          (open 5)
                          (place-fret 4 5 1)
                          (place-fret 3 5 2)
                          (place-fret 2 5 3)
                          (open 1))

% 11
%
% Simple D chord.
d'1~\markup
\override #'(fret-diagram-details
  . ((finger-code . below-string)
    (dot-radius . 0.35)
    (dot-position . 0.5)
    (fret-count . 3)))
\ fret-diagram-terse "x;x;o;2-1;3-2;2-3;"

% 12
%
```

```

% Simple D chord, large top fret thickness.
d'1~\markup
  \override #'(fret-diagram-details
    . ((finger-code . below-string)
      (dot-radius . 0.35)
      (dot-position . 0.5)
      (top-fret-thickness . 7)
      (fret-count . 3)))
  \fret-diagram-terse "x;x;o;2-1;3-2;2-3;"

%%
%% These chords will be in opposing-landscape orientation.
%%
\override TextScript.fret-diagram-details
  .orientation = #'opposing-landscape

% 13
%
% C major for guitar, barred on third fret: verbose style,
% roman fret label, finger labels below string, straight
% barre.
c'1~\markup
  % 110% of default size
  \override #'(size . 1.1)
  \override #'(fret-diagram-details
    . ((number-type . roman-lower)
      (finger-code . below-string)
      (barre-type . straight)))
  \fret-diagram-verbose #'((mute 6)
    (place-fret 5 3 1)
    (place-fret 4 5 2)
    (place-fret 3 5 3)
    (place-fret 2 5 4)
    (place-fret 1 3 1)
    (barre 5 1 3))

% 14
%
% C major for guitar, barred on third fret: double barre
% used to test barre function, verbose style.
c'1~\markup
  % 110% of default size
  \override #'(size . 1.1)
  \override #'(fret-diagram-details
    . ((number-type . arabic)
      (dot-label-font-mag . 0.9)
      (finger-code . in-dot)
      (fret-label-font-mag . 0.6)
      (fret-label-vertical-offset . 0)
      (label-dir . -1)

```



```

        (mute-string . "M")
        (xo-font-magnification . 0.4)
        (xo-padding . 0.3)))
\ fret-diagram-verbose #'((mute 6)
                          (place-fret 5 3 1)
                          (place-fret 4 5 2)
                          (place-fret 3 5 3)
                          (place-fret 2 5 4)
                          (place-fret 1 3 1)
                          (barre 4 2 5)
                          (barre 5 1 3))

% 15
%
% C major for guitar, with capo on third fret: verbose style.
c'1^\markup
% 110% of default size
\override #'(size . 1.1)
\override #'(fret-diagram-details
             . ((number-type . roman-upper)
               (dot-label-font-mag . 0.9)
               (finger-code . none)
               (fret-label-vertical-offset . 0.5)
               (xo-font-magnification . 0.4)
               (xo-padding . 0.3)))
\ fret-diagram-verbose #'((mute 6)
                          (capo 3)
                          (open 5)
                          (place-fret 4 5 1)
                          (place-fret 3 5 2)
                          (place-fret 2 5 3)
                          (open 1))

% 16
%
% Simple D chord.
d'1^\markup
\override #'(fret-diagram-details
             . ((finger-code . below-string)
               (dot-radius . 0.35)
               (dot-position . 0.5)
               (fret-count . 3)))
\ fret-diagram-terse "x;x;o;2-1;3-2;2-3;"

% 17
%
% Simple D chord, large top fret thickness.
d'1^\markup
\override #'(fret-diagram-details
             . ((finger-code . below-string)
               (dot-radius . 0.35)
               (dot-position . 0.5)

```

```

        (top-fret-thickness . 7)
        (fret-count . 3)))
    \fret-diagram-terse "x;x;o;2-1;3-2;2-3;"
  }
>>

\paper {
  ragged-right = ##t
  system-system-spacing.basic-distance = 20
}

\layout {
  \context {
    \Score
    \override SpacingSpanner.spacing-increment = 3
  }
}

```

The image displays a musical score with three systems of staves, each featuring fret diagrams for chords A, C, and D. The first system shows two A chords (labeled 'A') and a C chord (labeled 'C'). The second system shows three C chords (labeled 'C') and two D chords (labeled 'D'). The third system shows three C chords (labeled 'C') and two D chords (labeled 'D'). Each fret diagram is a 6-string grid with fingerings indicated by numbers 1-4 and 'x' for muted strings. The score is written in treble clef with a key signature of one sharp (F#).

Alternative Bunddiagrammtabellen

Alternative Bunddiagrammtabellen können erstellt werden. Sie können benutzt werden, um alternative Bunddiagramme für einen bestimmten Akkord zu haben.

Damit eine alternative Bunddiagrammentabelle benutzt werden kann, muss die Tabelle zuerst erstellt werden. Dann werden die Bunddiagramme zur Tabelle hinzugefügt.

Die erstellte Bunddiagrammtabelle kann auch leer sein, oder sie kann aus einer existierenden Tabelle kopiert werden.

Die Tabelle, die eingesetzt wird, um vordefinierte Bunddiagramme anzuzeigen, wird mit der Eigenschaft `\predefinedDiagramTable` ausgewählt.

```
\include "predefined-guitar-fretboards.ly"

% Make a blank new fretboard table.
#(define custom-fretboard-table-one
  (make-fretboard-table))

% Make a new fretboard table as a copy of `default-fret-table`.
#(define custom-fretboard-table-two
  (make-fretboard-table default-fret-table))

% Add a chord to `custom-fretboard-table-one`.
\storePredefinedDiagram #custom-fretboard-table-one
  \chordmode {c}
  #guitar-tuning
  "3-(;3;5;5;5;3-);"

% Add a chord to `custom-fretboard-table-two`.
\storePredefinedDiagram #custom-fretboard-table-two
  \chordmode {c}
  #guitar-tuning
  "x;3;5;5;5;o;"

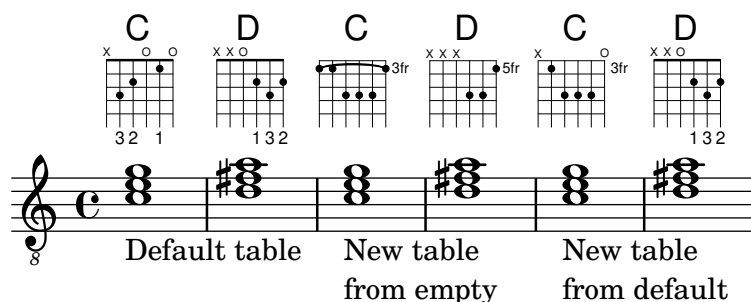
<<
\chords {
  c1 | d1 |
  c1 | d1 |
  c1 | d1 |
}
\new FretBoards {
  \chordmode {
    \set predefinedDiagramTable = #default-fret-table
    c1 | d1 |
    \set predefinedDiagramTable = #custom-fretboard-table-one
    c1 | d1 |
    \set predefinedDiagramTable = #custom-fretboard-table-two
    c1 | d1 |
  }
}
\new Staff {
  \clef "treble_8"
  <<
    \chordmode {
      c1 | d1 |
      c1 | d1 |
      c1 | d1 |
    }
  {

```

```

s1_\markup "Default table" | s1 |
s1_\markup \column { "New table" "from empty" } | s1 |
s1_\markup \column { "New table" "from default" } | s1 |
}
>>
}
>>

```



Flageolett von Bundinstrumenten in einer Tabulatur

Flageolett für Bundinstrumente:

```

pinchedHarmonics = {
  \textSpannerDown
  \override TextSpanner.bound-details.left.text =
    \markup { \halign #-0.5 \teeny "PH" }
  \override TextSpanner.style = #'dashed-line
  \override TextSpanner.dash-period = 0.6
  \override TextSpanner.bound-details.right.attach-dir = 1
  \override TextSpanner.bound-details.right.text =
    \markup { \draw-line #'(0 . 1) }
  \override TextSpanner.bound-details.right.padding = -0.5
}

harmonics = {
  % artificial harmonics (AH)
  \textLengthOn
  <\parenthesize b b'\harmonic>4_\markup { \teeny "AH 16" }
  <\parenthesize g g'\harmonic>4_\markup { \teeny "AH 17" }
  <\parenthesize d' d'\harmonic>2_\markup { \teeny "AH 19" }

  % pinched harmonics (PH)
  \pinchedHarmonics
  <a'\harmonic>2\startTextSpan
  <d'\harmonic>4
  <e'\harmonic>4\stopTextSpan

  % tapped harmonics (TH)
  <\parenthesize g\4 g'\harmonic>4_\markup { \teeny "TH 17" }
  <\parenthesize a\4 a'\harmonic>4_\markup { \teeny "TH 19" }
  <\parenthesize c'\3 c'\harmonic>2_\markup { \teeny "TH 17" }

  % touch harmonics (TCH)
  a4( <e'\harmonic>2. )_\markup { \teeny "TCH" }

```

```

}

frettedStrings = {
  % artificial harmonics (AH)
  \harmonicByFret 4 g4\3
  \harmonicByFret 5 d4\4
  \harmonicByFret 7 g2\3

  % pinched harmonics (PH)
  \harmonicByFret 7 d2\4
  \harmonicByFret 5 d4\4
  \harmonicByFret 7 a4\5

  % tapped harmonics (TH)
  \harmonicByFret 5 d4\4
  \harmonicByFret 7 d4\4
  \harmonicByFret 5 g2\3

  % touch harmonics (TCH)
  a4 \harmonicByFret 9 g2.\3
}

\score {
  <<
    \new Staff
    \with { \omit StringNumber } {
      \new Voice {
        \clef "treble_8"
        \harmonics
      }
    }
    \new TabStaff {
      \new TabVoice {
        \frettedStrings
      }
    }
  >>
}

```

Guitar slides

Unlike glissandos, slides may go from an imprecise point of the fretboard to a specific fret. A good way to do this is to add a hidden grace note before the note which is actually played, as demonstrated in the following example.

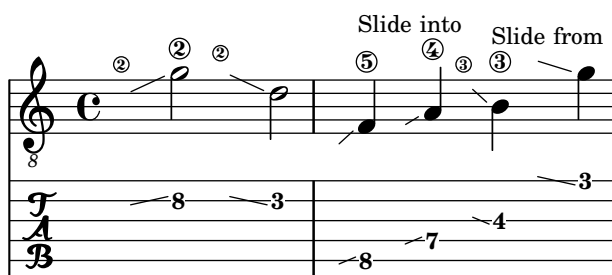
*% Hide fret number: useful to draw slide into/from a casual point of
% the fretboard.*

```
hideFretNumber = {
  \once \hide TabNoteHead
  \once \hide NoteHead
  \once \omit Stem
  \once \omit Flag
  \once \override NoteHead.no-ledgers = ##t
  \once \override Glissando.bound-details.left.padding = #0.3
}

music= \relative c' {
  \grace { \hideFretNumber d8\2 \glissando s2 } g2\2
  \grace { \hideFretNumber g8\2 \glissando s2 } d2 |

  \grace { \hideFretNumber c,8 \glissando s }
    f4\5~\markup \tiny { Slide into }
  \grace { \hideFretNumber f8 \glissando s } a4\4
  \grace { \hideFretNumber e'8\3 \glissando s }
    b4\3~\markup \tiny { Slide from }
  \grace { \hideFretNumber b'8 \glissando s2 } g4 |
}

\score {
  <<
    \new Staff {
      \clef "G_8"
      \music
    }
    \new TabStaff {
      \music
    }
  >>
}
```



Schlagrhythmus für Gitarren

In Gitarrennotation kann neben Melodie, Akkordbezeichnungen und Bunddiagrammen auch der Schlagrhythmus angegeben werden.

```
\include "predefined-guitar-fretboards.ly"
```

```
<<
  \new ChordNames \chordmode {
```

```

    c1 | f | g | c
  }
  \new FretBoards \chordmode {
    c1 | f | g | c
  }
  \new Voice \with {
    \consists "Pitch_squash_engraver"
  } \relative c'' {
    \improvisationOn
    c4 c8 c c4 c8 c
    f4 f8 f f4 f8 f
    g4 g8 g g4 g8 g
    c4 c8 c c4 c8 c
  }
  \new Voice = "melody" \relative c'' {
    c2 e4 e4
    f2. r4
    g2. a4
    e4 c2.
  }
  \new Lyrics \lyricsto "melody" {
    This is my song.
    I like to sing.
  }
}
>>

```

The image displays a musical score for a song. The top staff features four guitar chord diagrams: C (x, 0, 0, 3, 2, 1), F (1, 3, 4, 2, 1, 1), G (2, 1, 0, 0, 0, 3), and C (x, 0, 0, 3, 2, 1). The bottom staff shows the melody: 'This is my song. I like to sing.' with lyrics underneath.

Hammer-on and pull-off

Hammer-on and pull-off can be obtained using slurs.

```

\new TabStaff {
  \relative c' {
    d4( e\2)
    a( g)
  }
}

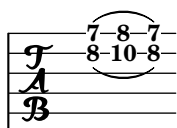
```

The image shows a musical notation for a hammer-on and pull-off. The staff displays a slur over the notes 3, 5, 5, 3, indicating a hammer-on from 3 to 5 and a pull-off from 5 to 3.

Hammer-on and pull-off using chords

When using hammer-on or pull-off with chorded notes, only a single arc is drawn. However „double arcs“ are possible by setting the `doubleSlurs` property to `#t`.

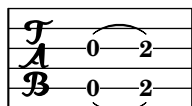
```
\new TabStaff {
  \relative c' {
    % chord hammer-on and pull-off
    \set doubleSlurs = ##t
    <g' b>8( <a c> <g b>)
  }
}
```



Hammer-on and pull-off using voices

The arc of hammer-on and pull-off is upwards in voices one and three and downwards in voices two and four:

```
\new TabStaff {
  \relative c' {
    << { \voiceOne g2( a) }
    \\ { \voiceTwo a,( b) }
    >> \oneVoice
  }
}
```



How to change fret diagram position

If you want to move the position of a fret diagram, for example, to avoid collision, or to place it between two notes, you have various possibilities.

- 1) Modify the value of the padding or extra-offset property (as shown in the first line).
- 2) You can add an invisible voice and attach the fret diagrams to the invisible notes in that voice (as shown in the second line).

If you need to move the fret according with a rythmic position inside the bar (in the example, the third beat of the measure) the second example is better, because the fret is aligned with the third beat itself.

```
harmonies = \chordmode
{
  a8:13
  \once \override ChordNames.ChordName.extra-offset = #'(10 . 0)
  b8:13 s4. |
  s2 b2:13
}

\score {
  <<
```



```

\new ChordNames \harmonies
\new Staff {
  % Method 1.
  a8~\markup \fret-diagram "6-x;5-0;4-2;3-0;2-0;1-2;"
  \once \override TextScript.extra-offset = #'(10 . 0)
  b4.~\markup \fret-diagram "6-x;5-2;4-4;3-2;2-2;1-4;"
  b4. a8 | \break

  % Method 2.
  <<
    { a8 b4.~ b4. a8 }
    { s2 s2~\markup \fret-diagram "6-x;5-2;4-4;3-2;2-2;1-4;" }
  >> |
}
>>
}

```

The image displays musical notation for two staves. The first staff features two fret diagrams: one for A⁹ add13 and another for B⁹ add13. Below these, a melodic line is shown with a slur connecting two notes. The second staff, labeled with a '2' at the beginning, shows a B⁹ add13 fret diagram and a similar melodic line with a slur.

Vorlage für Jazz-Combo

Hier ist ein ziemlich kompliziertes Beispiel für ein Jazz-Ensemble. Achtung: Alle Instrumente sind in `\key c \major` (C-Dur) notiert. Das bezieht sich auf die klingende Musik: LilyPond transponiert die Tonart automatisch, wenn sich die Noten innerhalb eines `ranspose`-Abschnitts befinden.

```

\header {
  title = "Song"
  subtitle = "(tune)"
  composer = "Me"
  meter = "moderato"
  piece = "Swing"
  tagline = \markup \column {
    "LilyPond example file by Amelie Zapf,"
    "Berlin 07/07/2003" }
}

```

% To make the example display properly in the documentation.

```

\paper {
  paper-width = 130\mm
  paper-height = 205\mm
}

```

```

% #(set-global-staff-size 16)

\include "english.ly"

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

sl = { \override NoteHead.style = #'slash
       \hide Stem }
nsl = { \revert NoteHead.style
       \undo \hide Stem }
crOn = \override NoteHead.style = #'cross
crOff = \revert NoteHead.style

% Insert chord name style stuff here.

jazzChords = { }

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

global = { \time 4/4 }

Key = { \key c \major }

% ##### Horns #####

% ----- Trumpet -----
trpt = \transpose c d \relative c' {
  \Key
  c1 | c | c |
}
trpHarmony = \transpose c' d {
  \jazzChords
}
trumpet = {
  \global
  \clef treble
  \trpt
}

% ----- Alto Saxophone -----
alto = \transpose c a \relative c' {
  \Key
  c1 | c | c |
}
altoHarmony = \transpose c' a {
  \jazzChords
}
altoSax = {
  \global

```

```

\clef treble
\alto
}

% ----- Baritone Saxophone -----
bari = \transpose c a' \relative c {
  \Key
  c1 | c1 |
  \sl d4^"Solo" d d d \ns1 |
}
bariHarmony = \transpose c' a \chordmode {
  \jazzChords
  s1 | s |
  d2:maj e:m7 |
}
bariSax = {
  \global
  \clef treble
  \bari
}

% ----- Trombone -----
tbone = \relative c {
  \Key
  c1 | c | c |
}
tboneHarmony = \chordmode {
  \jazzChords
}
trombone = {
  \global
  \clef bass
  \tbone
}

% ##### Rhythm Section #####

% ----- Guitar -----
gtr = \relative c'' {
  \Key
  c1 |
  \sl b4 b b b \ns1 |
  c1 |
}
gtrHarmony = \chordmode {
  \jazzChords
  s1 | c2:min7+ d2:maj9 | s1 |
}
guitar = {
  \global
  \clef treble
  \gtr

```

```

}

%% ----- Piano -----
rhUpper = \relative c'' {
  \voiceOne
  \Key
  c1 | c | c |
}
rhLower = \relative c' {
  \voiceTwo
  \Key
  e1 | e | e |
}

lhUpper = \relative c' {
  \voiceOne
  \Key
  g1 | g | g |
}
lhLower = \relative c {
  \voiceTwo
  \Key
  c1 | c | c |
}

PianoRH = {
  \clef treble
  \global
  <<
    \new Voice = "one" \rhUpper
    \new Voice = "two" \rhLower
  >>
}

PianoLH = {
  \clef bass
  \global
  <<
    \new Voice = "one" \lhUpper
    \new Voice = "two" \lhLower
  >>
}

piano = <<
  \new Staff = "upper" \PianoRH
  \new Staff = "lower" \PianoLH
>>

% ----- Bass Guitar -----
Bass = \relative c {
  \Key
  c1 | c | c |
}

```

```

}
bass = {
  \global
  \clef bass
  \Bass
}

% ----- Drums -----
up = \drummode {
  \voiceOne
  hh4 <hh sn> hh <hh sn> |
  hh4 <hh sn> hh <hh sn> |
  hh4 <hh sn> hh <hh sn> |
}
down = \drummode {
  \voiceTwo
  bd4 s bd s |
  bd4 s bd s |
  bd4 s bd s |
}

drumContents = {
  \global
  <<
    \new DrumVoice \up
    \new DrumVoice \down
  >>
}

%%%%%%%%%% It All Goes Together Here %%%%%%%%%%%

\book { % For the LilyPond documentation.
  \score {
    <<
      \new StaffGroup = "horns" <<
        \new Staff = "trumpet" \with { instrumentName = "Trumpet" }
        \trumpet
        \new Staff = "altosax" \with { instrumentName = "Alto Sax" }
        \altoSax
        \new ChordNames = "barichords" \with { instrumentName = "Bari Sax" }
        \bariHarmony
        \new Staff = "barisax" \with { instrumentName = "Bari Sax" }
        \bariSax
        \new Staff = "trombone" \with { instrumentName = "Trombone" }
        \trombone
      >>

      \new StaffGroup = "rhythm" <<
        \new ChordNames = "chords" \with { instrumentName = "Guitar" }
        \gtrHarmony
        \new Staff = "guitar" \with { instrumentName = "Guitar" }
        \guitar
    >>
  }
}

```

```

\new PianoStaff = "piano" \with {
  instrumentName = "Piano"
  midiInstrument = "acoustic grand"
} \piano
\new Staff = "bass" \with { instrumentName = "Bass" }
  \bass
\new DrumStaff \with { instrumentName = "Drums" }
  \drumContents
>>
>>

\layout {
  \context {
    \Staff
    \RemoveEmptyStaves
  }
  \context {
    \Score
    \override BarNumber.padding = 3
    \override RehearsalMark.padding = 2
    skipBars = ##t
  }
}
\midi { }
}

```

Song

(tune)

Me

moderato
Swing

Trumpet

Alto Sax

Bari Sax

Trombone

Guitar

Piano

Bass

Drums

B^Δ C[♯]m⁷

Solo

Cm^Δ D^Δ9

LilyPond example file by Amelie Zapf,
Berlin 07/07/2003

Laissez vibrer ties

Laissez vibrer ties have a fixed size. Their positioning can be tuned using the `tie-configuration` property.

See also snippet „Longer laissez vibrer ties“.

```
\relative c' {
  <c e g>4\laissezVibrer r <c f g>\laissezVibrer r
  <c d f g>4\laissezVibrer r <c d f g>4.\laissezVibrer r8

  <c d e f>4\laissezVibrer r
  \override LaissezVibrerTieColumn.tie-configuration
    = #`((-7 . ,DOWN)
```

```

        (-5 . ,DOWN)
        (-3 . ,UP)
        (-1 . ,UP))
    <c d e f>4\laissezVibrer r
}

```



Let TabStaff print the topmost string at bottom

In tablatures, the first string is usually printed topmost. If you want to have it at the bottom, set the `stringOneTopmost` context property to `#f`. For a context-wide setting this could be done in the `\layout` block as well.

```

%\layout {
%  \context {
%    \Score
%      stringOneTopmost = ##f
%  }
%  \context {
%    \TabStaff
%      tablatureFormat = #fret-letter-tablature-format
%  }
%}

m = {
  \cadenzaOn
  e, b, e gis! b e'
  \bar "||"
}

<<
  \new Staff {
    \clef "G_8"
    <>_"default" \m
    <>_"italian (historic)"\m
  }
  \new TabStaff
  {
    \m
    \set Score.stringOneTopmost = ##f
    \set TabStaff.tablatureFormat = #fret-letter-tablature-format
    \m
  }
>>

```


Letter tablature formatting

Tablature can be formatted using letters instead of numbers.

```
music = \relative c {
  c4 d e f
  g4 a b c
  d4 e f g
}

<<
\new Staff {
  \clef "G_8"
  \music
}
\new TabStaff \with {
  tablatureFormat = #fret-letter-tablature-format
} {
  \music
}
>>
```

Referenz für Flageolett von offenen Saiten

Referenz für Flageolett von offenen Saiten:

```
openStringHarmonics = {
  \textSpannerDown
  \override TextSpanner.staff-padding = 3
  \override TextSpanner.dash-fraction = 0.3
  \override TextSpanner.dash-period = 1

  % first harmonic
  \override TextSpanner.bound-details.left.text =
    \markup\small "1st harm. "
  \harmonicByFret 12 e,2\6\startTextSpan
  \harmonicByRatio #1/2 e,\6\stopTextSpan
```

```
% second harmonic
\override TextSpanner.bound-details.left.text =
  \markup\small "2nd harm. "
\harmonicByFret 7 e,\6\startTextSpan
\harmonicByRatio #1/3 e,\6
\harmonicByFret 19 e,\6
\harmonicByRatio #2/3 e,\6\stopTextSpan
%\harmonicByFret 19 < e,\6 a,\5 d\4 >
%\harmonicByRatio #2/3 < e,\6 a,\5 d\4 >
```

```
% third harmonic
\override TextSpanner.bound-details.left.text =
  \markup\small "3rd harm. "
\harmonicByFret 5 e,\6\startTextSpan
\harmonicByRatio #1/4 e,\6
\harmonicByFret 24 e,\6
\harmonicByRatio #3/4 e,\6\stopTextSpan
\break
```

```
% fourth harmonic
\override TextSpanner.bound-details.left.text =
  \markup\small "4th harm. "
\harmonicByFret 4 e,\6\startTextSpan
\harmonicByRatio #1/5 e,\6
\harmonicByFret 9 e,\6
\harmonicByRatio #2/5 e,\6
\harmonicByFret 16 e,\6
\harmonicByRatio #3/5 e,\6\stopTextSpan
```

```
% fifth harmonic
\override TextSpanner.bound-details.left.text =
  \markup\small "5th harm. "
\harmonicByFret 3 e,\6\startTextSpan
\harmonicByRatio #1/6 e,\6\stopTextSpan
\break
```

```
% sixth harmonic
\override TextSpanner.bound-details.left.text =
  \markup\small "6th harm. "
\harmonicByFret 2.7 e,\6\startTextSpan
\harmonicByRatio #1/7 e,\6\stopTextSpan
```

```
% seventh harmonic
\override TextSpanner.bound-details.left.text =
  \markup\small "7th harm. "
\harmonicByFret 2.3 e,\6\startTextSpan
\harmonicByRatio #1/8 e,\6\stopTextSpan
```

```
% eighth harmonic
\override TextSpanner.bound-details.left.text =
  \markup\small "8th harm. "
\harmonicByFret 2 e,\6\startTextSpan
```

```

\harmonicByRatio #1/9 e,\6\stopTextSpan
}

\score {
  <<
    \new Staff \with { \omit StringNumber } {
      \new Voice {
        \clef "treble_8"
        \openStringHarmonics
      }
    }
    \new TabStaff {
      \new TabVoice {
        \openStringHarmonics
      }
    }
  >>
}

```

The image displays a musical score for the first six harmonics of the open string 'e' (E4) on a fretted string instrument. The score is organized into three systems, each featuring a treble staff and a corresponding tablature staff.

- System 1:** Shows the 1st, 2nd, and 3rd harmonics. The treble staff has a key signature of one sharp (F#) and a common time signature (C). The tablature staff shows fret numbers: (12), (12), (7), (7), (19), (19), (5), (5), (24), (24).
- System 2:** Shows the 4th and 5th harmonics. The treble staff has a key signature of two sharps (F#, C#) and a common time signature (C). The tablature staff shows fret numbers: (4), (4), (9), (9), (16), (16), (3), (3).
- System 3:** Shows the 6th, 7th, and 8th harmonics. The treble staff has a key signature of two sharps (F#, C#) and a common time signature (C). The tablature staff shows fret numbers: (2.7), (2.7), (2.3), (2.3), (2), (2).

Positionierung von Fingersatz der rechten Hand

Man kann die Positionierung von Fingersatz der rechten Hand besser kontrollieren, wenn eine bestimmte Eigenschaft gesetzt wird, wie das folgende Beispiel zeigt:

```

\define RH rightHandFinger

```

```

\relative c {
  \clef "treble_8"

```

```

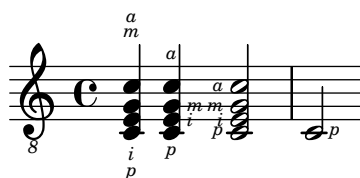
\set strokeFingerOrientations = #'(up down)
<c\RH 1 e\RH 2 g\RH 3 c\RH 4 >4

\set strokeFingerOrientations = #'(up right down)
<c\RH 1 e\RH 2 g\RH 3 c\RH 4 >4

\set strokeFingerOrientations = #'(left)
<c\RH 1 e\RH 2 g\RH 3 c\RH 4 >2

\set strokeFingerOrientations = #'(right)
c\RH 1
}

```



Polyphonie in einer Tabulatur

Polyphonie kann in einer Tabulatur (TabStaff) genauso wie in einem normalen Notensystem erstellt werden.

```

upper = \relative c' {
  \time 12/8
  \key e \minor
  \voiceOne
  r4. r8 e, fis g16 b g e e' b c b a g fis e
}

lower = \relative c {
  \key e \minor
  \voiceTwo
  r16 e d c b a g4 fis8 e fis g a b c
}

\score {
  \new StaffGroup = "tab with traditional" <<
    \new Staff = "guitar traditional" <<
      \clef "treble_8"
      \new Voice = "upper" \upper
      \new Voice = "lower" \lower
    >>

    \new TabStaff = "guitar tab" <<
      \new TabVoice = "upper" \upper
      \new TabVoice = "lower" \lower
    >>
  >>
}

```

Setting up predefined fretboards for other instruments

Predefined fret diagrams can be added for new instruments in addition to the standard diagrams used for guitar. This file shows how this is done by defining a new string tuning and a few predefined fretboards for the Venezuelan *cuatro*.

This file also shows how fingerings can be included in the chords used as reference points for the chord lookup, and displayed in the fret diagram and the TabStaff, but not the music.

These fretboards are not transposable because they contain string information. This is planned to be corrected in the future.

```
% Add fretboards for the cuatro.
%
% Note: This section could be put into a separate file
%       `predefined-cuatro-fretboards.ly`
%       and be \included into each of your compositions.
```

```
cuatroTuning = #`(,(ly:make-pitch 0 6 0)
                  ,(ly:make-pitch 1 3 SHARP)
                  ,(ly:make-pitch 1 1 0)
                  ,(ly:make-pitch 0 5 0))
```

```
dSix = { <a\4 b\1 d\3 fis\2> }
dMajor = { <a\4 d\1 d\3 fis \2> }
aMajSeven = { <a\4 cis\1 e\3 g\2> }
dMajSeven = { <a\4 c\1 d\3 fis\2> }
gMajor = { <b\4 b\1 d\3 g\2> }
```

```
\storePredefinedDiagram #default-fret-table \dSix
                        #cuatroTuning
                        "o;o;o;o;"
\storePredefinedDiagram #default-fret-table \dMajor
                        #cuatroTuning
                        "o;o;o;3-3;"
\storePredefinedDiagram #default-fret-table \aMajSeven
                        #cuatroTuning
                        "o;2-2;1-1;2-3;"
\storePredefinedDiagram #default-fret-table \dMajSeven
                        #cuatroTuning
                        "o;o;o;1-1;"
\storePredefinedDiagram #default-fret-table \gMajor
                        #cuatroTuning
                        "2-2;o;1-1;o;"
```

```

% End of potential include file `predefined-cuatro-fretboards.ly`.

#(set-global-staff-size 16)

primerosNames = \chordmode {
  d:6 d a:maj7 d:maj7
  g
}
primeros = {
  \dSix \dMajor \aMajSeven \dMajSeven
  \gMajor
}

\score {
  <<
    \new ChordNames {
      \set chordChanges = ##t
      \primerosNames
    }

    \new Staff {
      \new Voice \with {
        \remove "New_fingering_engraver"
      }
      \relative c'' {
        \primeros
      }
    }

    \new FretBoards {
      \set Staff.stringTunings = #cuatroTuning
%      \override FretBoard
%      #'(fret-diagram-details string-count) = 4
      \override FretBoard.fret-diagram-details.finger-code = #'in-dot
      \primeros
    }

    \new TabStaff \relative c'' {
      \set TabStaff.stringTunings = #cuatroTuning
      \primeros
    }

  >>

  \layout {
    \context {
      \Score
      \override SpacingSpanner.base-shortest-duration =
        \musicLength 16
    }
  }
}

```

```
\midi { }
}
```

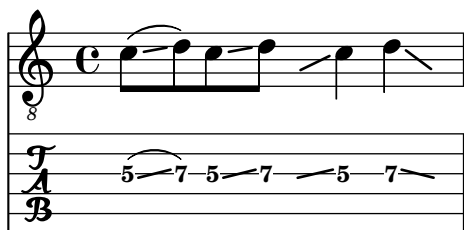
Gleiten (Glissando) in Tabulatur

Gleiten kann sowohl in normalem Notensystem als auch in Tabaturen notiert werden:

```
slides = {
  c'8\3(\glissando d'8\3)
  c'8\3\glissando d'8\3
  \hideNotes
  \grace { g16\glissando }
  \unHideNotes
  c'4\3
  \afterGrace d'4\3\glissando {
    \stemDown \hideNotes
    g16 }
  \unHideNotes
}

\score {
  <<
    \new Staff { \clef "treble_8" \slides }
    \new TabStaff { \slides }
  >>

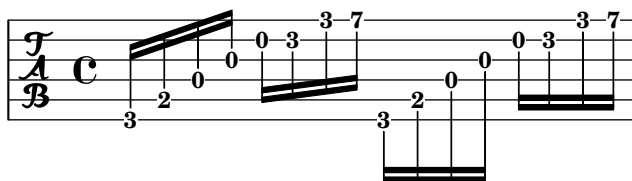
  \layout {
    \context {
      \Score
      \override Glissando.minimum-length = 4
      \override Glissando.springs-and-rods =
        #ly:spanner::set-spacing-rods
      \override Glissando.thickness = 2
      \omit StringNumber
      % or:
      %\override StringNumber.stencil = ##f
    }
  }
}
```



Hals- und Balkenverhalten in einer Tabulatur

Die Richtung von Hälsen wird in Tabulaturen genauso wie in normaler Notation eingestellt. Balken können horizontal eingestellt werden, wie das Beispiel zeigt.

```
\new TabStaff {
  \relative c {
    \tabFullNotation
    g16 b d g b d g b
    \stemDown
    \override Beam.concaveness = 10000
    g,,16 b d g b d g b
  }
}
```

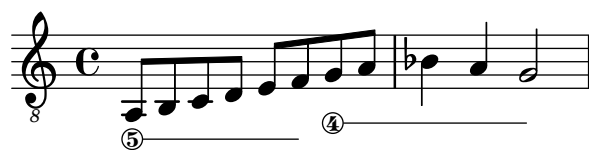


String number extender lines

Make an extender line for string number indications, showing that a series of notes is supposed to be played all on the same string.

```
stringNumberSpanner =
  #(define-music-function (StringNumber) (string?)
    #{
      \override TextSpanner.style = #'solid
      \override TextSpanner.font-size = #-5
      \override TextSpanner.bound-details.left.stencil-align-dir-y = #CENTER
      \override TextSpanner.bound-details.left.text =
        \markup { \circle \number $StringNumber }
    #})

\relative c {
  \clef "treble_8"
  \textSpannerDown
  \stringNumberSpanner "5" a8\startTextSpan b c d
  e f\stopTextSpan \stringNumberSpanner "4" g\startTextSpan a |
  bes4 a g2\stopTextSpan
}
```

14 Unfretted string instruments

See also Abschnitt “Unfretted string instruments” in *Notationsreferenz*.

Creating slurs across voices

In some situations it is necessary to create slurs between notes from different voices. The solution is to add invisible notes to one of the voices, using `\hideNotes`.

This example is measure 235 of the Ciaccona from Bach’s second partita for solo violin, BWV 1004.

```
\relative c' {
  <<
    {
      d16( a') s a s a[ s a] s a[ s a]
    }
    \\\
    {
      \slurUp
      bes,16[ s e](
      \hideNotes a)
      \unHideNotes f[(
      \hideNotes a)
      \unHideNotes fis](
      \hideNotes a)
      \unHideNotes g[(
      \hideNotes a)
      \unHideNotes gis](
      \hideNotes a)
    }
  >>
}
```



Dotted harmonics

Artificial harmonics using `\harmonic` do not show dots. To override this behavior, set the context property `harmonicDots`.

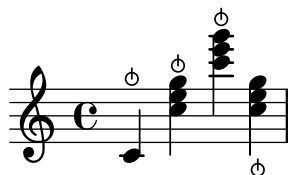
```
\relative c' '' {
  \time 3/4
  \key f \major
  \set harmonicDots = ##t
  <bes f'\harmonic>2. ~
  <bes f'\harmonic>4. <a e'\harmonic>8( <gis dis'\harmonic> <g d'\harmonic>)
  <fis cis'\harmonic>2.
  <bes f'\harmonic>2.
}
```



Snap pizzicato („Bartok“ pizzicato)

A snap pizzicato (also known as „Bartok pizzicato“) is a „strong pizzicato where the string is plucked vertically by snapping and rebounds off the fingerboard of the instrument“ (Wikipedia). It is denoted by a circle with a vertical line going from the center upwards outside the circle.

```
\relative c' {
  c4\snappizzicato
  <c' e g>4\snappizzicato
  <c' e g>4^\snappizzicato
  <c, e g>4_\snappizzicato
}
```



Vorlage für Streichquartett (einfach)

Dieses Beispiel demonstriert die Partitur für ein Streichquartett. Hier wird auch eine „\global“-Variable für Taktart und Vorzeichen benutzt.

```
global= {
  \time 4/4
  \key c \major
}

violinOne = \new Voice \relative c' {
  c2 d
  e1
  \bar "|."
}

violinTwo = \new Voice \relative c' {
  g2 f
  e1
  \bar "|."
}

viola = \new Voice \relative c' {
  \clef alto
  e2 d
  c1
  \bar "|."
}

cello = \new Voice \relative c' {
```

```

\clef bass
c2 b
a1
\bar "|."
}

\score {
  \new StaffGroup <<
    \new Staff \with { instrumentName = "Violin 1" }
      << \global \violinOne >>
    \new Staff \with { instrumentName = "Violin 2" }
      << \global \violinTwo >>
    \new Staff \with { instrumentName = "Viola" }
      << \global \viola >>
    \new Staff \with { instrumentName = "Cello" }
      << \global \cello >>
  >>
  \layout { }
  \midi { }
}

```

Vorlage für Streichquartett mit einzelnen Stimmen

Mit diesem Beispiel können Sie ein schönes Streichquartett notieren, aber wie gehen Sie vor, wenn Sie Stimmen brauchen? Das Beispiel oben hat gezeigt, wie Sie mit Variablen einzelne Abschnitte getrennt voneinander notieren können. Im nächsten Beispiel wird nun gezeigt, wie Sie mit diesen Variablen einzelne Stimmen erstellen.

Sie müssen das Beispiel in einzelne Dateien aufteilen; die Dateinamen sind in den Kommentaren am Anfang jeder Datei enthalten. `piece.ly` enthält die Noten. Die anderen Dateien – `score.ly`, `vn1.ly`, `vn2.ly`, `vla.ly` und `vlc.ly` – erstellen daraus die entsprechenden Stimmen bzw. die Partitur (`score.ly`). Mit `ag` wird den Stimmen ein Name zugewiesen, auf den zurückgegriffen werden kann.

```

% piece.ly
% (This is the global definitions file.)

```

```

global= {
  \time 4/4
  \key c \major

```

```

}

Violinone = \new Voice \relative c' {
  c2 d e1
  \bar "|."
}

Violintwo = \new Voice \relative c' {
  g2 g e1
  \bar "|."
}

Viola = \new Voice \relative c' {
  \clef alto
  e2 d c1
  \bar "|."
}

Cello = \new Voice \relative c' {
  \clef bass
  c2 b a1
  \bar "|."
}

music = <<
  \tag #'score \tag #'vn1
  \new Staff \with { instrumentName = "Violin 1" }
    << \global \Violinone >>

  \tag #'score \tag #'vn2
  \new Staff \with { instrumentName = "Violin 2" }
    << \global \Violintwo >>

  \tag #'score \tag #'vla
  \new Staff \with { instrumentName = "Viola" }
    << \global \Viola >>

  \tag #'score \tag #'vlc
  \new Staff \with { instrumentName = "Cello" }
    << \global \Cello >>
>>

% These are the other files you need to save on your computer

% score.ly
% (This is the main file.)

% Uncomment the line below when using a separate file.
% \include "piece.ly"

#(set-global-staff-size 14)

```

```

\score {
  \new StaffGroup \keepWithTag #'score \music
  \layout { }
  \midi { }
}

%{ Uncomment this block when using separate files.

% vn1.ly
% (This is the Violin 1 part file.)

\include "piece.ly"
\score {
  \keepWithTag #'vn1 \music
  \layout { }
}

% vn2.ly
% (This is the Violin 2 part file.)

\include "piece.ly"
\score {
  \keepWithTag #'vn2 \music
  \layout { }
}

% vla.ly
% (This is the Viola part file.)

\include "piece.ly"
\score {
  \keepWithTag #'vla \music
  \layout { }
}

% vlc.ly
% (This is the Cello part file.)

\include "piece.ly"
\score {
  \keepWithTag #'vlc \music
  \layout { }
}

%}

```

Violin 1

Violin 2

Viola

Cello

The image shows a musical score for four string instruments: Violin 1, Violin 2, Viola, and Cello. The score is written in common time (C) and consists of two measures. Violin 1 and Violin 2 are in treble clef, Viola is in alto clef, and Cello is in bass clef. The notes are as follows:

Instrument	Measure 1	Measure 2
Violin 1	C4 (quarter), D4 (quarter)	E4 (half)
Violin 2	B3 (quarter), A3 (quarter)	G3 (half)
Viola	C3 (quarter), D3 (quarter)	E3 (half)
Cello	C2 (quarter), D2 (quarter)	E2 (half)

15 Wind instruments

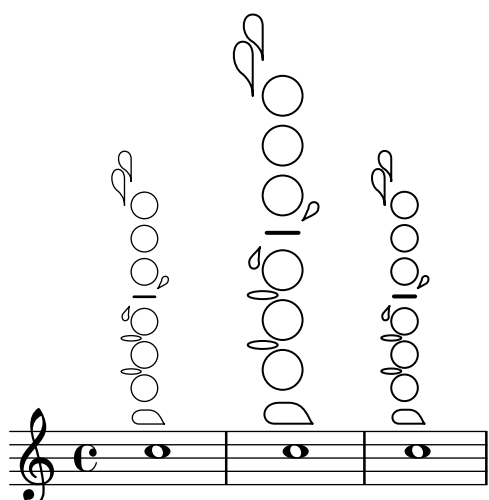
See also Abschnitt “Wind instruments” in *Notationsreferenz*.

Größe von Holzbläserdiagrammen ändern

Die Größe und Dicke der Holzbläserdiagramme kann geändert werden.

```
\relative c' {
  \textLengthOn
  c1^\markup
    \woodwind-diagram #'piccolo #'()

  c^\markup \override #'(size . 1.5)
    \woodwind-diagram #'piccolo #'()
  c^\markup \override #'(thickness . 0.15)
    \woodwind-diagram #'piccolo #'()
}
```



Fingering symbols for wind instruments

Special symbols can be achieved by combining existing glyphs, which is useful for wind instruments.

```
lineup =
  \tweak outside-staff-padding #0
  \tweak staff-padding #0
  \tweak padding #0.2
  \tweak parent-alignment-X #CENTER
  \tweak self-alignment-X #CENTER
  \etc

\relative c' {
  g\open
  g\lineup ^\markup \combine
    \musicglyph "scripts.open"
    \musicglyph "scripts.tenuto"
  g\lineup ^\markup \combine
    \musicglyph "scripts.open"
```



```

\musicglyph "scripts.stopped"
g\stopped
}

```



Flute slap notation

It is possible to indicate special articulation techniques such as a flute „tongue slap“ by replacing the note head with the appropriate glyph. For that we can draw the accent-like note head with `\markup`.

```

slap =
#(define-music-function (music) (ly:music?)
  #{
    \temporary \override NoteHead.stencil =
      #ly:text-interface::print
    \temporary \override NoteHead.text =
      \markup
        \translate #'(1 . 0)
        \override #'(thickness . 1.4)
        \overlay { \draw-line #'(-1.2 . 0.4)
                   \draw-line #'(-1.2 . -0.4) }
    \temporary \override NoteHead.stem-attachment =
      #(lambda (grob)
        (let* ((stem (ly:grob-object grob 'stem))
              (dir (ly:grob-property stem 'direction UP))
              (is-up (eqv? dir UP)))
          (cons dir (if is-up 0 -0.8)))))
    #music
    \revert NoteHead.stencil
    \revert NoteHead.text
    \revert NoteHead.stem-attachment
  })

\relative c' {
  c4 \slap c d r
  \slap { g4 a } b r
}

```



Graphische und Text-Holzbläserdiagramme

In vielen Fällen können die nicht in der mittleren Reihe befindlichen Löcher dargestellt werden, indem man die Lochbezeichnung oder graphische Zeichen benutzt.

```

\relative c' {
  \textLengthOn
  c1^\markup

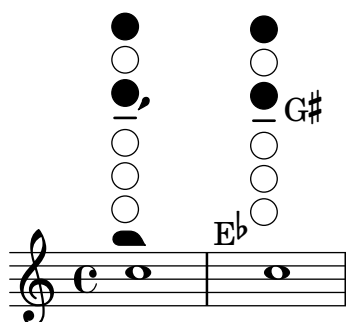
```

```

\woodwind-diagram #'piccolo
      #'((cc . (one three))
        (lh . (gis))
        (rh . (ees)))

c^\markup
  \override #'(graphical . #f)
  \woodwind-diagram #'piccolo
      #'((cc . (one three))
        (lh . (gis))
        (rh . (ees)))
}

```



Recorder fingering chart

The following example demonstrates how fingering charts for wind instruments can be realized.

% range chart for paetzold contrabass recorder

```

centermarkup = {
  \once \override TextScript.self-alignment-X = #CENTER
  \once \override TextScript.X-offset = #(\lambda (g)
    (+ (ly:self-alignment-interface::centered-on-x-parent g)
      (ly:self-alignment-interface::x-aligned-on-self g)))
}

\new Staff \with {
  \remove "Time_signature_engraver"
  \omit Stem
  \omit Flag
  \consists "Horizontal_bracket_engraver"
} {
  \clef bass
  \set Score.timing = ##f

  f,1*1/4 \glissando

  \clef violin
  gis'1*1/4

  a'4^\markup "1)"

  \centermarkup
}

```

```

\once \override TextScript.padding = 2
bes'1*1/4_\markup \override #'(baseline-skip . 1.7) \column {
  \fontsize #-5
  \slashed-digit #0 \finger 1 \finger 2
  \finger 3 \finger 4 \finger 5 \finger 6 \finger 7 }

b'1*1/4

c''4^\markup "1)"

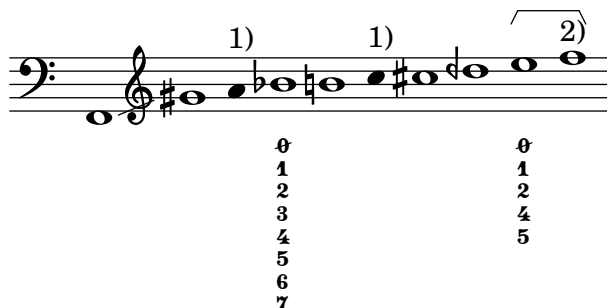
cis''1*1/4

deh''1*1/4

\centermarkup
\once \override TextScript.padding = 2
\once \override Staff.HorizontalBracket.direction = #UP
e''1*1/4_\markup \override #'(baseline-skip . 1.7) \column {
  \fontsize #-5
  \slashed-digit #0 \finger 1 \finger 2
  \finger 4 \finger 5 } \startGroup

f''1*1/4^\markup "2)" \stopGroup
}

```



Liste der Löcher für Holzbläserdiagramme

Dieses Schnipsel erzeugt eine Liste aller möglichen Löcher und Locheinstellungen für Holzbläserdiagramme, wie sie in der Datei `scm/define-woodwind-diagrams.scm` definiert sind. Die Liste wird in der Log-Datei angezeigt, nicht in den Noten. Wenn Ausgabe auf der Kommandozeile gewünscht ist, muss `(current-error-port)` von den Befehlen weggelassen werden.

```

#(print-keys-verbose 'piccolo (current-error-port))
#(print-keys-verbose 'flute (current-error-port))
#(print-keys-verbose 'flute-b-extension (current-error-port))
#(print-keys-verbose 'tin-whistle (current-error-port))
#(print-keys-verbose 'oboe (current-error-port))
#(print-keys-verbose 'clarinet (current-error-port))
#(print-keys-verbose 'bass-clarinet (current-error-port))
#(print-keys-verbose 'low-bass-clarinet (current-error-port))
#(print-keys-verbose 'saxophone (current-error-port))
#(print-keys-verbose 'soprano-saxophone (current-error-port))
#(print-keys-verbose 'alto-saxophone (current-error-port))

```

```

#(print-keys-verbose 'tenor-saxophone (current-error-port))
#(print-keys-verbose 'baritone-saxophone (current-error-port))
#(print-keys-verbose 'bassoon (current-error-port))
#(print-keys-verbose 'contrabassoon (current-error-port))

\score {c''1}

```



Liste der Holzbläserdiagramme

Folgende Noten zeige alle Holzbläserdiagramme, die für LilyPond definiert sind.

```

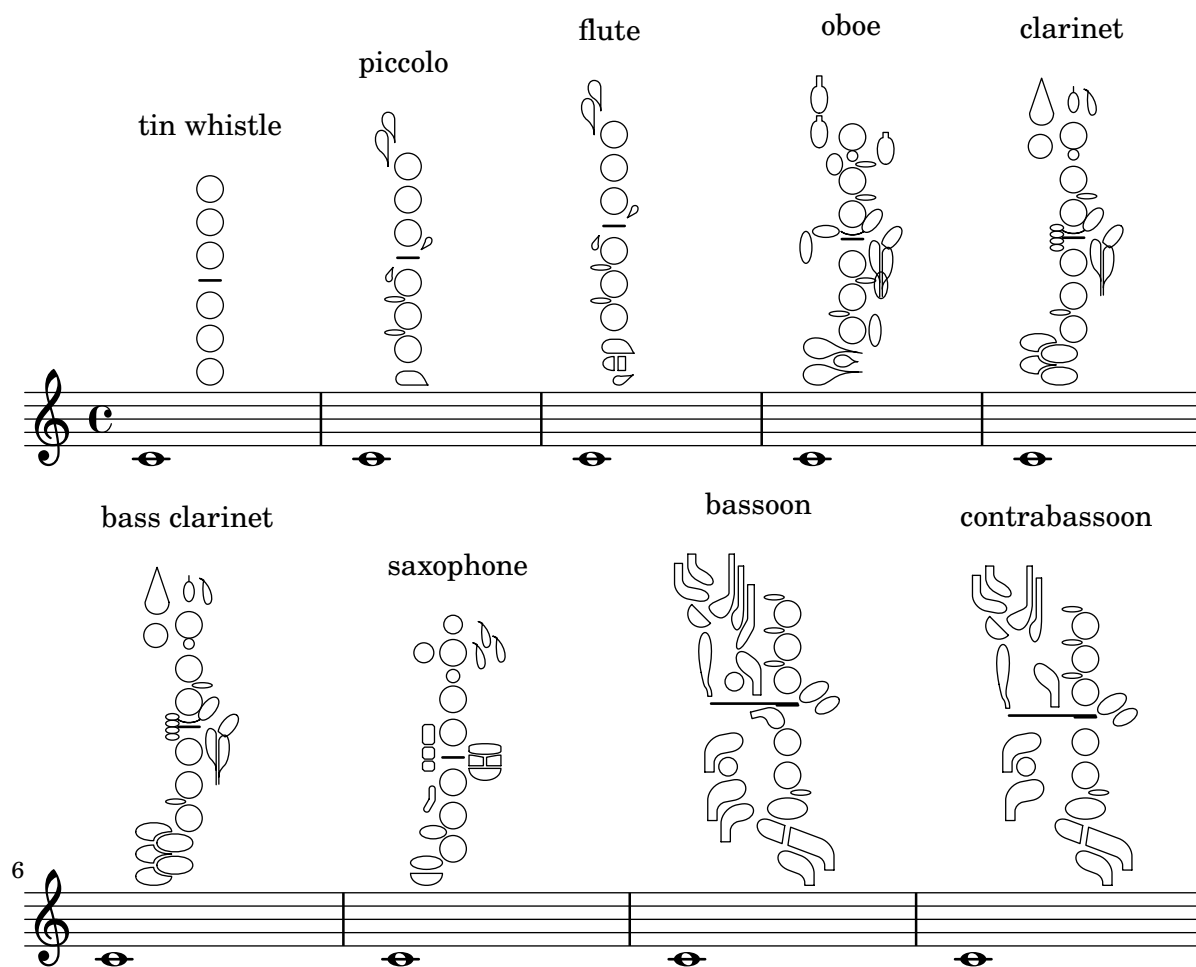
\relative c' {
  \textLengthOn
  c1^\markup \center-column { "tin whistle"
    " "
    \woodwind-diagram #'tin-whistle #'() }
  c1^\markup \center-column { "piccolo"
    " "
    \woodwind-diagram #'piccolo #'() }
  c1^\markup \center-column { "flute"
    " "
    \woodwind-diagram #'flute #'() }
  c1^\markup \center-column { "oboe"
    " "
    \woodwind-diagram #'oboe #'() }
  c1^\markup \center-column { "clarinet"
    " "
    \woodwind-diagram #'clarinet #'() }

  \break

  c1^\markup \center-column { "bass clarinet"
    " "
    \woodwind-diagram #'bass-clarinet #'() }
  c1^\markup \center-column { "saxophone"
    " "
    \woodwind-diagram #'saxophone #'() }
  c1^\markup \center-column { "bassoon"
    " "
    \woodwind-diagram #'bassoon #'() }
  c1^\markup \center-column { "contrabassoon"
    " "
    \woodwind-diagram #'contrabassoon #'() }
}

\paper {
  system-system-spacing.padding = 5
}

```



16 Ancient notation

See also Abschnitt “Ancient notation” in *Notationsreferenz*.

Adding a figured bass above or below the notes

When writing figured bass, you can place the figures above or below the bass notes by using the commands `\bassFigureStaffAlignmentDown` and `\bassFigureStaffAlignmentUp`. Prepend `\once` to the command if you want to modify only the next figured bass.

The command `\bassFigureStaffAlignmentNeutral` resets the direction of figured bass to the default value.

```
bass = {
  \clef bass
  g4 b, c d |
  e d8 c d2
}

continuo = \figuremode {
  <_>4 <6>4 <5/>4
  \bassFigureStaffAlignmentUp
  <_+>4 <6> |
  \set Staff.useBassFigureExtenders = ##t
  \bassFigureStaffAlignmentDown
  <4>4. <4>8 <_+>4
}

\score {
  <<
    \new Staff = bassStaff \bass
    \context Staff = bassStaff \continuo
  >>
}
```



Ancient fonts

This snippets shows many of the symbols contained in the Emmentaler font that are used by LilyPond for typesetting ancient notation.

```
m = { c1 e f ges cis' \bar "||" }
```

```
\markup \with-true-dimensions % work around a cropping issue
\score {
  \new VaticanaVoice {
    \clef "vaticana-fa2"
    \key es \major
    \textMark \markup \rounded-box "Vaticana clefs, custos and note heads"

    \override NoteHead.style = #'vaticana.punctum
```

```

<>^"vaticana.punctum" \m

\override NoteHead.style = #'vaticana.inclinatum
<>^"vaticana.inclinatum" \m

\override NoteHead.style = #'vaticana.quilisma
<>^"vaticana.quilisma" \m

\clef "vaticana-fa1"
\override NoteHead.style = #'vaticana.plica
<>^"vaticana.plica" \m

\override NoteHead.style = #'vaticana.reverse.plica
<>^"vaticana.reverse.plica" \m

\override NoteHead.style = #'vaticana.punctum.cavum
<>^"vaticana.punctum.cavum" \m

\override NoteHead.style = #'vaticana.lpes
<>^"vaticana.punctum.lpes" \m

\override NoteHead.style = #'vaticana.upes
<>^"vaticana.punctum.upes" \m

\override NoteHead.style = #'vaticana.vupes
<>^"vaticana.punctum.vupes" \m

\override NoteHead.style = #'vaticana.linea.punctum
<>^"vaticana.punctum.linea" \m

\override NoteHead.style = #'vaticana.epiphonus
<>^"vaticana.punctum.epiphonus" \m

\override NoteHead.style = #'vaticana.cephalicus
<>^"vaticana.punctum.cephalicus" \m

\break

\textMark \markup \rounded-box "Medicaea clefs, custos and note heads"
\set VaticanaStaff.alterationGlyphs =
  #alteration-medicaea-glyph-name-alist
\override VaticanaStaff.Custos.style = #'medicaea

\clef "medicaea-fa2"
\override NoteHead.style = #'medicaea.punctum
<>^"medicaea.punctum" \m

\clef "medicaea-do2"
\override NoteHead.style = #'medicaea.inclinatum
<>^"medicaea.inclinatum" \m

\override NoteHead.style = #'medicaea.virga

```

```

<>^"medicaea.virga" \m

\clef "medicaea-fa1"
\override NoteHead.style = #'medicaea.rvirga
<>^"medicaea.rvirga" \m

\break

\textMark \markup \rounded-box "Hufnagel clefs, custos and note heads"
\set Staff.alterationGlyphs =
  #alteration-hufnagel-glyph-name-alist
\override VaticanaStaff.Custos.style = #'hufnagel
\clef "hufnagel-fa2"

\break

\override NoteHead.style = #'hufnagel.punctum
<>^"hufnagel.punctum" \m

\clef "hufnagel-do2"
\override NoteHead.style = #'hufnagel.lpes
<>^"hufnagel.lpes" \m

\clef "hufnagel-do-fa"
\override NoteHead.style = #'hufnagel.virga
<>^"hufnagel.virga" \m
}

\layout {
  % Compensate `with-true-dimensions` for PDF output.
  line-width = 159\mm

  \context {
    \Score
    \override TextScript.font-size = #-2
    \override TextMark.break-align-symbols = #'(left-edge clef staff-bar)
    \override TextMark.padding = 4
    \omit BarNumber
  }
  \context {
    \VaticanaStaff
    alterationGlyphs =
      #alteration-vaticana-glyph-name-alist
  }
}
}

```


Vaticana clefs, custos and note heads

Examples of Vaticana notation on a four-line staff:

- vaticana.punctum**: Shows a punctum (a small square) with a custos (a small square with a cross) above it.
- vaticana.inclinatum**: Shows an inclinatum (a small square with a diagonal line) with a custos (a small square with a cross) above it.
- vaticana.quilisma**: Shows a quilisma (a small square with a cross) with a custos (a small square with a cross) above it.
- vaticana.plica**: Shows a plica (a small square with a cross) with a custos (a small square with a cross) above it.
- vaticana.reverse.plica**: Shows a reverse plica (a small square with a cross) with a custos (a small square with a cross) above it.
- vaticana.punctum.cavum**: Shows a punctum (a small square) with a custos (a small square with a cross) above it.
- vaticana.punctum.lpes**: Shows a punctum (a small square) with a custos (a small square with a cross) above it.
- vaticana.punctum.upes**: Shows a punctum (a small square) with a custos (a small square with a cross) above it.
- vaticana.punctum.vupes**: Shows a punctum (a small square) with a custos (a small square with a cross) above it.
- vaticana.punctum.linea**: Shows a punctum (a small square) with a custos (a small square with a cross) above it.
- vaticana.punctum.epiphonus**: Shows a punctum (a small square) with a custos (a small square with a cross) above it.
- vaticana.punctum.cephalicus**: Shows a punctum (a small square) with a custos (a small square with a cross) above it.

Medicaea clefs, custos and note heads

Examples of Medicaea notation on a four-line staff:

- medicaea.punctum**: Shows a punctum (a small square) with a custos (a small square with a cross) above it.
- medicaea.inclinatum**: Shows an inclinatum (a small square with a diagonal line) with a custos (a small square with a cross) above it.
- medicaea.virga**: Shows a virga (a small square with a cross) with a custos (a small square with a cross) above it.
- medicaea.rvirga**: Shows a rvirga (a small square with a cross) with a custos (a small square with a cross) above it.

Hufnagel clefs, custos and note heads

Examples of Hufnagel notation on a four-line staff:

- hufnagel.punctum**: Shows a punctum (a small square) with a custos (a small square with a cross) above it.
- hufnagel.lpes**: Shows a lpes (a small square with a cross) with a custos (a small square with a cross) above it.
- hufnagel.virga**: Shows a virga (a small square with a cross) with a custos (a small square with a cross) above it.

Ancient notation template – modern transcription of Gregorian music

This example demonstrates how to do modern transcription of Gregorian music. Gregorian music has no measure, no stems; it uses only half and quarter note heads, and special marks, indicating rests of different length.

```
chant = \relative c' {
  \set Score.timing = ##f
  f4 a2 \divisioMinima
  g4 b a2 f2 \divisioMaior
  g4( f) f( g f) a2 \finalis \break
  f4 a2 \divisioMinima
  g4 b a2 f2 \divisioMaior
  g4( f) f( g a) g2( f) \finalis
}

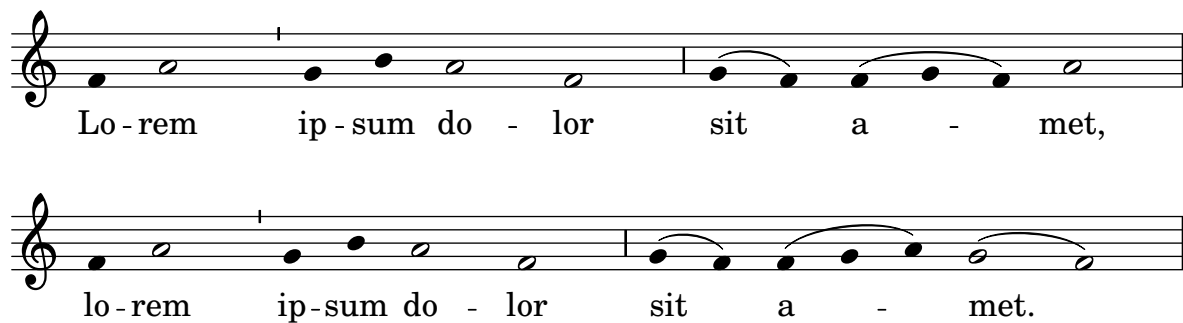
verba = \lyricmode {
  Lo -- rem ip -- sum do -- lor sit a -- met,
  lo -- rem ip -- sum do -- lor sit a -- met.
}
```

```
\score {
  \new GregorianTranscriptionStaff <<
```

```

\new GregorianTranscriptionVoice = "melody" \chant
\new GregorianTranscriptionLyrics = "one" \lyricsto melody \verba
>>
}

```



Ancient time signatures

Time signatures may also be engraved in an old style.

```

{
  \override Staff.TimeSignature.style = #'neomensural
  s1
}

```



Chant or psalm notation

This form of notation is used for psalm chant, where verses are not always of the same length.

```

stemOff = \hide Staff.Stem
stemOn  = \undo \stemOff

\score {
  \new Staff \with { \remove "Time_signature_engraver" }
  {
    \key g \minor
    \cadenzaOn
    \stemOff a'\breve bes'4 g'4
    \stemOn a'2 \section
    \stemOff a'\breve g'4 a'4
    \stemOn f'2 \section
    \stemOff a'\breve~\markup { \italic flexe }
    \stemOn g'2 \fine
  }
}

```



Custodes

Custodes may be engraved in various styles.

```
\layout {
  ragged-right = ##t
}

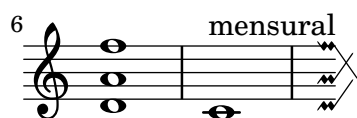
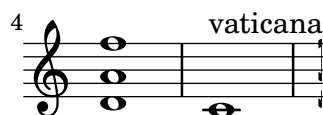
\markup \with-true-dimensions % work around a cropping issue
\score {
  \new Staff \with { \consists "Custos_engraver" } \relative c' {
    \override Staff.Custos.neutral-position = #4

    \override Staff.Custos.style = #'hufnagel
    c1^"hufnagel" \break
    <d a' f'>1

    \override Staff.Custos.style = #'medicaea
    c1^"medicaea" \break
    <d a' f'>1

    \override Staff.Custos.style = #'vaticana
    c1^"vaticana" \break
    <d a' f'>1

    \override Staff.Custos.style = #'mensural
    c1^"mensural" \break
    <d a' f'>1
  }
}
```



Incipit

When transcribing mensural music, an incipit at the beginning of the piece is useful to indicate the original key and tempo. While today musicians are used to bar lines in order to faster recognize rhythmic patterns, bar lines were not yet invented during the period of mensural music; in fact, the meter often changed after every few notes. As a compromise, bar lines are often printed between the staves rather than on the staves.

% A short excerpt from the Jubilate Deo by Orlande de Lassus

```
global = {
  \set Score.skipBars = ##t
  \key g \major
  \time 4/4

  % the actual music
  \skip 1*8

  % let finis bar go through all staves
  \override Staff.BarLine.transparent = ##f

  % finis bar
  \bar "|."
}

discantusIncipit = \new PetrucciStaff {
  \clef "petrucci-c1"
  \key f \major
  \time 2/2
  c''1.
}

discantusNotes = {
  \transpose c' c'' {
    \clef "treble"
    d'2. d'4 |
    b e' d'2 |
    c'4 e'4.( d'8 c' b |
    a4) b a2 |
    b4.( c'8 d'4) c'4 |
    \once \hide NoteHead
    c'1 |
    b\breve |
  }
}

discantusLyrics = \lyricmode {
  Ju -- bi -- la -- te De -- o,
  om -- nis ter -- ra, __ om-
  "...
  -us.
}
```

```
altusIncipit = \new PetrucciStaff {
  \clef "petrucci-c3"
  \key f \major
  \time 2/2
  e'1\rest f'1.
}
```

```
altusNotes = {
  \transpose c' c'' {
    \clef "treble"
    r2 g2. e4 fis g |
    a2 g4 e |
    fis g4.( fis16 e fis4) |
    g1 |
    \once \hide NoteHead
    g1 |
    g\breve |
  }
}
```

```
altusLyrics = \lyricmode {
  Ju -- bi -- la -- te
  De -- o, om -- nis ter -- ra,
  "...
  -us.
}
```

```
tenorIncipit = \new PetrucciStaff {
  \clef "petrucci-c4"
  \key f \major
  \time 2/2
  r\longa
  r\breve
  r1 c'1.
}
```

```
tenorNotes = {
  \transpose c' c' {
    \clef "treble_8"
    R1 |
    R1 |
    R1 |
    % two measures
    r2 d'2. d'4 b e' |
    \once \hide NoteHead
    e'1 |
    d'\breve |
  }
}
```

```
tenorLyrics = \lyricmode {
  Ju -- bi -- la -- te
```

```

    "...
    -us.
}

bassusIncipit = \new PetrucciStaff {
  % The original print shows the b flat
  % for the f major key signature twice.
  \override Staff.KeySignature.flat-positions = #'((-7 . 6))
  \clef "mensural-f"
  \key f\major
  \time 2/2
  \tweak Y-offset #1 r\longa \tweak Y-offset #1 r\longa
  f1.
}

bassusNotes = {
  \transpose c' c' {
    \clef "bass"
    R1 |
    R1 |
    R1 |
    R1 |
    g2. e4 |
    \once \hide NoteHead
    e1 |
    g\breve |
  }
}

bassusLyrics = \lyricmode {
  Ju -- bi-
  "...
  -us.
}

\score {
  <<
  \new StaffGroup = choirStaff <<
  \new Voice = "discantusNotes" <<
    \set Staff.instrumentName = "Discantus"
    \incipit #1 \discantusIncipit
    \global
    \discantusNotes
  >>
  \new Lyrics \lyricsto discantusNotes { \discantusLyrics }
  \new Voice = "altusNotes" <<
    \set Staff.instrumentName = "Altus"
    \global
    \incipit #1 \altusIncipit
    \altusNotes
  >>
  \new Lyrics \lyricsto altusNotes { \altusLyrics }
}

```

```

\new Voice = "tenorNotes" <<
  \set Staff.instrumentName = "Tenor"
  \global
  \incipit #1 \tenorIncipit
  \tenorNotes
>>
\new Lyrics \lyricsto tenorNotes { \tenorLyrics }
\new Voice = "bassusNotes" <<
  \set Staff.instrumentName = "Bassus"
  \global
  \incipit #1 \bassusIncipit
  \bassusNotes
>>
\new Lyrics \lyricsto bassusNotes { \bassusLyrics }
>>
>>
\layout {
  \context {
    \Score
    %% no bar lines in staves or lyrics
    \hide BarLine
  }
  %% the next two instructions keep the lyrics between the bar lines
  \context {
    \Lyrics
    \consists "Bar_engraver"
    \consists "Separating_line_group_engraver"
  }
  \context {
    \Voice
    %% no slurs
    \hide Slur
    %% Comment in the below "\remove" command to allow line
    %% breaking also at those bar lines where a note overlaps
    %% into the next measure. The command is commented out in this
    %% short example score, but especially for large scores, you
    %% will typically yield better line breaking and thus improve
    %% overall spacing if you comment in the following command.
    %%\remove "Forbid_line_break_engraver"
  }
  indent = 5\cm
  incipit-width = 2.5\cm
}
}

```

Discantus

Altus

Tenor

Bassus

- nis ter - ra, om- ... -us.

nis ter - ra, ... -us.

Ju - bi - la - te ... -us.

Ju - bi - ... -us.

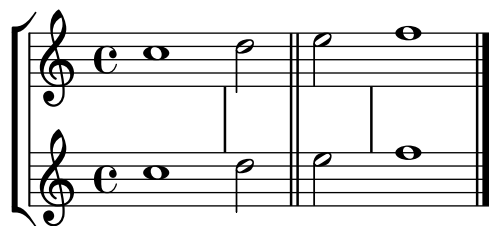
Mensurstriche-Layout (Taktstriche zwischen den Systemen)

Das Mensurstriche-Layout, in welchem die Taktlinien nicht auf den Systemen, sondern zwischen den Systemen gesetzt werden, kann mit einer `StaffGroup` anstelle von `ChoirStaff` erreicht werden. Die Taktlinien auf den Systemen werden mit der `transparent`-Eigenschaft ausgelöscht.

```
\layout {
  \context {
    \Staff
    measureBarType = "-span|"
  }
}
```

```
music = \fixed c'' {
  c1
  d2 \section e2
  f1 \fine
}
```

```
\new StaffGroup <<
  \new Staff \music
  \new Staff \music
>>
```

Pausenstile

Pausen können in verschiedenen Stilen dargestellt werden.

```
restsA = {
  r\maxima r\longa r\breve r1 r2 r4 r8 r16 s32
  s64 s128 s256 s512 s1024 s1024
}
restsB = {
  r\maxima r\longa r\breve r1 r2 r4 r8 r16 r32
  r64 r128 r256 r512 r1024 s1024
}

\new Staff \relative c {
  \omit Score.TimeSignature
  \cadenzaOn

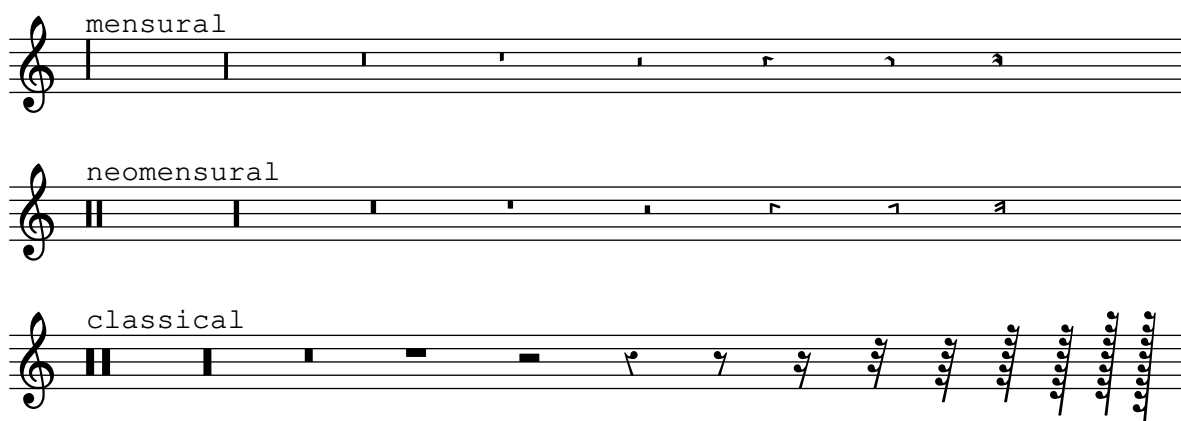
  \override Staff.Rest.style = #'mensural
  <>\markup \typewriter { mensural } \restsA \bar "" \break

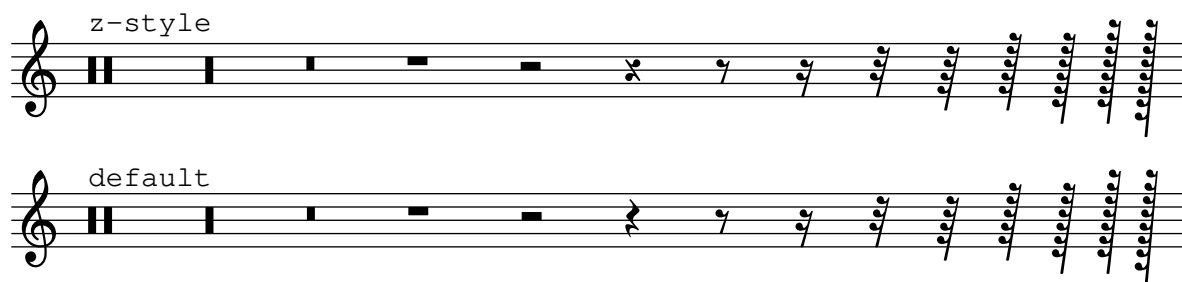
  \override Staff.Rest.style = #'neomensural
  <>\markup \typewriter { neomensural } \restsA \bar "" \break

  \override Staff.Rest.style = #'classical
  <>\markup \typewriter { classical } \restsB \bar "" \break

  \override Staff.Rest.style = #'z
  <>\markup \typewriter { z-style } \restsB \bar "" \break

  \override Staff.Rest.style = #'default
  <>\markup \typewriter { default } \restsB \bar "" \break
}
```





Using tags to produce mensural and modern music from the same source

Using tags it is possible to produce both mensural and modern notation from the same music. In this snippet, a function `\menrest` is introduced, allowing mensural rests to be pitched as in the original, but with modern rests in the standard staff position.

Tags can also be used where other differences are needed: for example using „whole measure rests“ (`R1`, `R\breve`, etc.) in modern music, but normal rests (`r1`, `r\breve`, etc.) in the mensural version. Converting mensural music to its modern equivalent is usually referred to as *transcription*.

The call `c4.\Be c8 c\Am` is the same as `c4.[c8 c]`. However, it suppresses warnings if it starts on a note that can't hold a beam but needs it anyway due to the use of `Completion_heads_engraver`.

[Note that the custos sticks out into the right margin and might be cut off if the LilyPond output gets cropped tightly. The use of `\with-true-dimensions` below avoids this.]

```
\layout {
  line-width = 150\mm
}

menrest = #(define-music-function (note) (ly:music?)
  #{
    \tag #'mens $(make-music 'RestEvent note)
    \tag #'mod $(make-music 'RestEvent note 'pitch '())
  #})

Be = \tag #'mod
  #(begin
    (ly:expect-warning (G_ "stem does not fit in beam"))
    (ly:expect-warning (G_ "beam was started here"))
    (make-span-event 'BeamEvent START))

Am = \tag #'mod ]

MenStyle = {
  \override Score.BarNumber.transparent = ##t
  \override Stem.neutral-direction = #up
  \omit Slur
  \omit Beam
}

finalis = \section

Music = \relative c'' {
  \key f \major
```

```

g1 d'2 \menrest bes4 bes a2 \menrest r4 g4 fis4. fis8 fis4 fis \break
g e f4.([ g8] a4[ g8 f] g2.\Be fis8 e\Am fis2) g\breve \finalis
}

```

```

MenLyr = \lyricmode {
  So farre, deere life, deare life,
  from thy bright beames ab- en- ted,
}

```

```

ModLyr = \lyricmode {
  So far, dear life, dear life,
  from your bright beams ab -- sen -- ted, __
}

```

```

\markup \with-true-dimensions % work around a cropping issue

```

```

\score {
  \keepWithTag #'mens {
    <<
      \new PetrucciStaff {
        \new PetrucciVoice = "Cantus" {
          \clef "petrucci-c1" \time 4/4 \MenStyle \Music
        }
      }
      \new Lyrics \lyricsto "Cantus" \MenLyr
    >>
  }
  \layout {
    \context {
      \PetrucciVoice
      % No longer necessary starting with version 2.25.23.
      \override Flag.style = #'mensural
    }
  }
}

```

```

\markup\vspace #1

```

```

\score {
  \keepWithTag #'mod {
    \new ChoirStaff <<
      \new Staff {
        \new Voice = "Sop" \with {
          \remove "Note_heads_engraver"
          \consists "Completion_heads_engraver"
          \remove "Rest_engraver"
          \consists "Completion_rest_engraver"
        } \shiftDurations 1 0 { \time 2/4 \autoBeamOff \Music }
      }
      \new Lyrics \lyricsto "Sop" \ModLyr
    >>
  }
}

```

So farre, deere life, deare life, from thy bright
beames ab- fen- ted,

So far, dear life, dear life, from your bright
beams ab - sen - - - ted,_____

Vertical line as a baroque articulation mark

This short vertical line placed above the note is commonly used in baroque music. Its meaning can vary, but generally indicates notes that should be played with more „weight“. The following example demonstrates how to achieve such a notation.

```

upline =
\tweak stencil
#(lambda (grob)
  (grob-interpret-markup grob #{ \markup \draw-line #'(0 . 1) #}))
\stopped

\relative c' {
  a'4^\upline a( c d')_\upline
}

```

17 World music

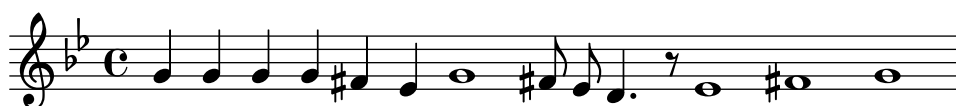
See also Abschnitt “World music” in *Notationsreferenz*.

Arabische Improvisation

Bei Improvisation oder *taqasim*, die zeitlich frei gespielt werden, kann die Taktart ausgelassen werden und `\cadenzaOn` kann eingesetzt werden. Es kann nötig sein, den Versetzungszeichenstil anzupassen, weil sonst die Versetzungszeichen nur einmal ausgegeben werden, da keine Taktlinien gesetzt sind. Hier ein Beispiel, wie der Beginn einer *hijaz*-Improvisation aussehen könnte:

```
\include "arabic.ly"
```

```
\relative sol' {
  \key re \kurd
  \accidentalStyle forget
  \cadenzaOn
  sol4 sol sol sol fad mib sol1 fad8 mib re4. r8 mib1 fad sol
}
```



Makam-Beispiel

Makam ist eine türkische Melodie, in der 1/9-Tonabstände eingesetzt werden. Sehen Sie sich die Initialisierungsdatei `makam.ly` für weitere Information zu Tonhöhenbezeichnungen und Alterationen an (siehe Handbuch zum Lernen 2.25.34, 4.6.3 Weitere Information zu Hinweisen, wo diese Datei gespeichert ist).

```
\include "makam.ly"
```

```
\relative c' {
  \set Staff.keyAlterations = #`((6 . ,(- KOMA)) (3 . ,BAKIYE))
  c4 cc db fk
  gbm4 gfc gfb efk
  fk4 db cc c
}
```

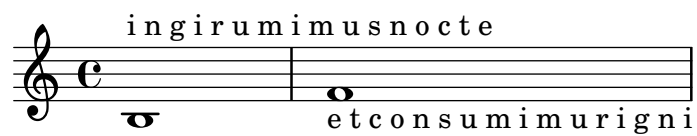


Printing text from right to left

It is possible to print text from right to left in a markup object, as demonstrated here.

```
{
  b1~\markup {
    \line { i n g i r u m i m u s n o c t e }
  }
  f'~\markup {
    \override #'(text-direction . -1)
    \line { i n g i r u m i m u s n o c t e }
  }
}
```

}



Turkish Makam example

This template uses the start of a well-known Turkish *Saz Semai* that is familiar in the repertoire in order to illustrate some of the elements of Turkish music notation.

```

\set-default-paper-size "a6" 'landscape)

```

```

\include "turkish-makam.ly"

```

```

\header {
  title = "Hüseyin Saz Semaisi"
  composer = "Lavtacı Andon"
  tagline = ##f
}

```

```

\relative {
  \set Staff.extraNatural = ##f
  \set Staff.autoBeaming = ##f

```

```

  \key a \huseyni
  \time 10/8

```

```

  a'4 g'16[ fb] e8.[ d16] d[ c d e] c[ d c8] bfc |
  a16[ bfc a8] bfc c16[ d c8] d16[ e d8] e4 fb8 |
  d4 a'8 a16[ g fb e] fb8[ g] a8.[ b16] a16[ g] |
  g4 g16[ fb] fb8.[ e16] e[ g fb e] e4 r8 |

```

}

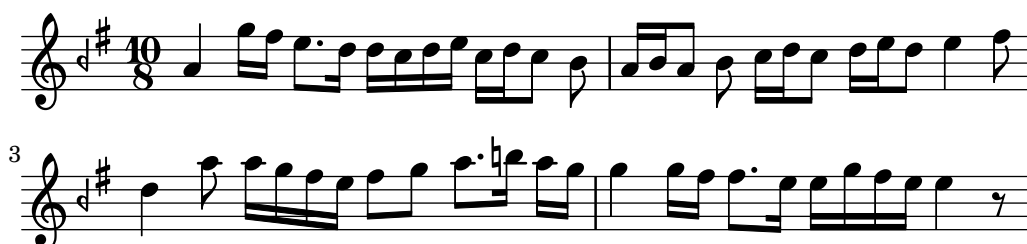
```

\layout {
  indent = 0
}

```

Hüseyin Saz Semaisi

Lavtacı Andon



18 Contexts and engravers

See also Abschnitt “Changing defaults” in *Notationsreferenz* and Abschnitt “Contexts and engravers” in *Handbuch zum Lernen*.

Ambitus pro Stimme hinzufügen

Ambitus können pro Stimme gesetzt werden. In diesem Fall müssen sie manuell verschoben werden, um Zusammenstöße zu verhindern.

```
\new Staff <<
  \new Voice \with {
    \consists "Ambitus_engraver"
  } \relative c'' {
    \override Ambitus.X-offset = 2.0
    \voiceOne
    c4 a d e
    f1
  }
  \new Voice \with {
    \consists "Ambitus_engraver"
  } \relative c' {
    \voiceTwo
    es4 f g as
    b1
  }
}>>
```

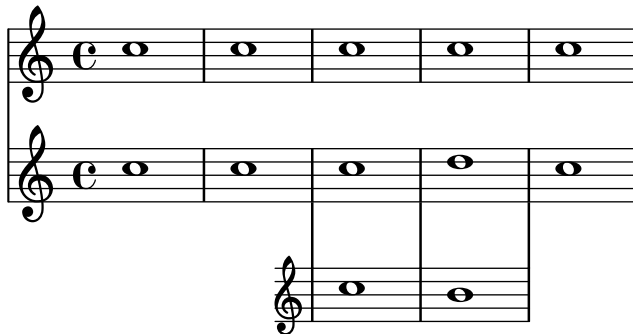


Adding an extra staff

An extra staff can be added (possibly temporarily) after the start of a piece.

```
\score {
  <<
    \new Staff \relative c'' {
      c1 | c | c | c | c
    }
    \new StaffGroup \relative c'' {
      \new Staff {
        c1 | c
      } <<
      { c1 | d }
      \new Staff {
        \once \omit Staff.TimeSignature
        c1 | b
      }
    }
  >>
  c1
}
}
```

```
>>
}
```

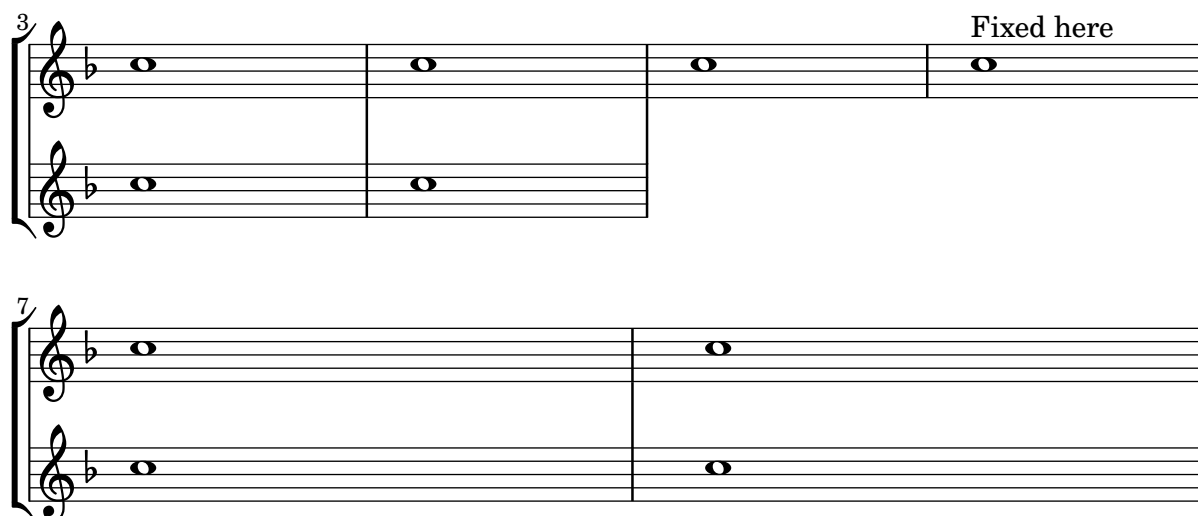


Adding an extra staff at a line break

When adding a new staff at a line break, some extra space is unfortunately added at the end of the line before the break (to fit in a key signature change, which is never printed anyway). The workaround is to set the `explicitKeySignatureVisibility` property of the `Staff` grob as is shown in the example.

```
\score {
  \new StaffGroup \relative c'' {
    \new Staff
    \key f \major
    c1 c^"Unwanted extra space" \break
    << { c1 | c }
    \new Staff {
      \key f \major
      \once \omit Staff.TimeSignature
      c1 | c
    }
  }
  >>
  c1 | c^"Fixed here" \break
  << { c1 | c }
  \new Staff {
    \once \set Staff.explicitKeySignatureVisibility =
      #end-of-line-invisible
    \key f \major
    \once \omit Staff.TimeSignature
    c1 | c
  }
  >>
}
}
```





Taktstriche in einen ChordNames-Kontext hinzufügen

Um Taktstriche in einem ChordNames-Kontext anzeigen zu lassen, muss der `Bar_engraver` hinzugefügt werden.

```
\new ChordNames \with {
  \override BarLine.bar-extent = #'(-1 . 3)
  \consists "Bar_engraver"
}

\chordmode {
  f1:maj7 f:7 bes:7
}
```

F^Δ | F⁷ | B^b7 |

Ambitus after key signature

By default, ambitus are positioned at the left of the clef. The `\ambitusAfter` function allows for changing this placement. Syntax is `\ambitusAfter grob-interface`; see Graphical Object Interfaces (<https://lilypond.org/doc/v2.24/Documentation/internals/graphical-object-interfaces>) for a list of possible values for `grob-interface`.

A common use case is printing the ambitus between key signature and time signature.

```
\new Staff \with {
  \consists Ambitus_engraver
} \relative {
  \ambitusAfter key-signature
  \key d \major
  es'8 g bes cis d2
}
```



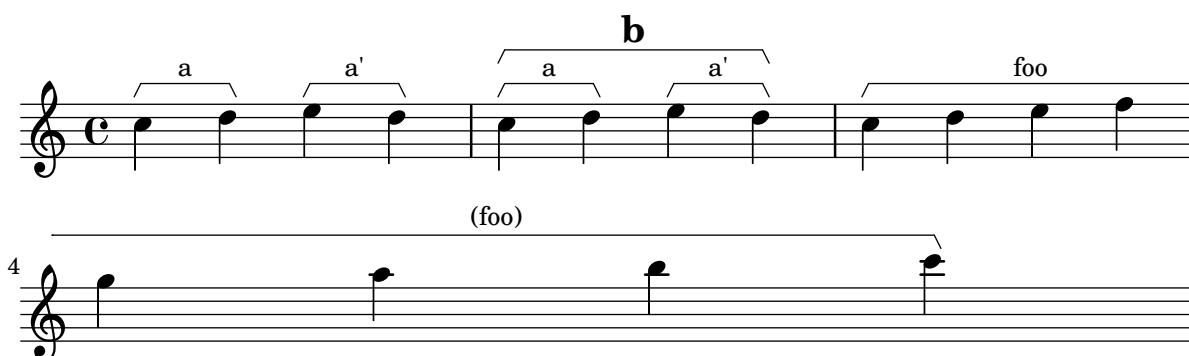
Analysis brackets with labels

Text markup may be added to analysis brackets using the `text` property of the `HorizontalBracketText` grob. Adding different texts to brackets beginning at the same time requires the `\tweak` command.

Bracket text gets parenthesized after a line break. The vertical order of nested brackets can be controlled with the `outside-staff-priority` property.

```
\layout {
  \context {
    \Voice
    \consists "Horizontal_bracket_engraver"
    \override HorizontalBracket.direction = #UP
  }
}

{
  \once\override HorizontalBracketText.text = "a"
  c''\startGroup d''\stopGroup
  \once\override HorizontalBracketText.text = "a'"
  e''\startGroup d''\stopGroup |
  c''-\tweak outside-staff-priority #801
    \tweak HorizontalBracketText.text
    \markup \bold \huge "b" \startGroup
    -\tweak HorizontalBracketText.text "a" \startGroup
    d''\stopGroup
    e''-\tweak HorizontalBracketText.text "a'" \startGroup
    d''\stopGroup\stopGroup |
  c''-\tweak HorizontalBracketText.text foo \startGroup
    d'' e'' f'' | \break
  g'' a'' b'' c'''\stopGroup
}
```



Automatically changing the stem direction of the middle note based on the melody

LilyPond can alter the stem direction of the middle note on a staff so that it follows the melody, by adding the `Melody_engraver` to the `Voice` context.

The context property `suspendMelodyDecisions` may be used to turn off this behavior locally.

```
\relative c'' {
  \time 3/4
  a8 b g f b g |
```

```

\set suspendMelodyDecisions = ##t
a b g f b g |
\unset suspendMelodyDecisions
c b d c b c |
}

\layout {
  \context {
    \Voice
    \consists "Melody_engraver"
    \autoBeamOff
  }
}

```



Changing MIDI output to one channel per voice

When outputting MIDI, the default behavior is for each staff to represent one MIDI channel, with all the voices on a staff amalgamated. This minimizes the risk of running out of MIDI channels, since there are only 16 available per track.

However, by moving the `Staff_performer` to the `Voice` context, each voice on a staff can have its own MIDI channel, as is demonstrated by the following example: despite being on the same staff, two MIDI channels are created, each with a different `midiInstrument`.

```

\score {
  \new Staff <<
    \new Voice \relative c''' {
      \set midiInstrument = "flute"
      \voiceOne
      \key g \major
      \time 2/2
      r2 g-"Flute" ~
      g fis ~
      fis4 g8 fis e2 ~
      e4 d8 cis d2
    }
    \new Voice \relative c'' {
      \set midiInstrument = "clarinet"
      \voiceTwo
      b1-"Clarinet"
      a2. b8 a
      g2. fis8 e
      fis2 r
    }
  >>
  \layout { }
  \midi {
    \context {
      \Staff
      \remove "Staff_performer"
    }
  }
}

```

```

    }
    \context {
      \Voice
      \consists "Staff_performer"
    }
    \tempo 2 = 72
  }
}

```



Changing time signatures inside a polymetric section using `\scaleDurations`

Flexible polymeric with unaligned measures

To support explicit creation of independently measured contexts, remove the `Timing_translator` from `Score` context and define a `TimingStaffGroup` context that has `Timing_translator`. This makes `Timing` an alias for `TimingStaffGroup`, targeting `\time` commands to the enclosing `TimingStaffGroup`.

Unlike LilyPond's built-in `\enablePerStaffTiming` command, this approach requires the explicit creation of `TimingStaffGroup` contexts; in exchange, it allows creating multiple `Staff` contexts that jointly follow the measure defined in their enclosing `TimingStaffGroup`.

Locally scaled time signatures

Use the unscalable `\time` command to establish a measure of the desired length in `Timing`, a.k.a. `TimingStaffGroup`. In this snippet, all staves below `TimingStaffGroup` use a scaled time signature, so any time signature with the desired measure length is as good as any other. If there were an enclosed context that did not use a scaled time signature, the choice of time signature to set in `Timing` would matter in that context.

Use the `\polymeric \time` command to set scalable metric properties in contexts below `Timing`, and use the `\scaleDurations` command to scale both the local meter and the notes to fit the measure.

```

\layout {
  \context {
    \Score
    \remove "Timing_translator"
    \accepts TimingStaffGroup
  }
  \context {
    \StaffGroup
    \name TimingStaffGroup
    \alias StaffGroup
    \consists "Timing_translator"
  }
}

```

```

\new TimingStaffGroup <<
  \new Staff {
    \scaleDurations 8/5 {
      \time 6/5 % to set measure length in Timing
      \context Staff \polymetric \time 6/8
      b8 b b b b b
      \time 4/5 % to set measure length in Timing
      \context Staff \polymetric \time 2/4
      b4 b
    }
  }
>>
\new TimingStaffGroup <<
  \new Staff {
    \clef bass
    \time 2/4
    c2 d e f
  }
>>
>>

```



Arpeggios zwischen unterschiedlichen Stimmen erzeugen

Ein Arpeggio kann zwischen Noten aus unterschiedlichen Stimmen auf demselben System gezogen werden, wenn der `Span_arpeggio_engraver` in den `Staff`-Kontext verschoben wird:

```

\new Staff \with {
  \consists "Span_arpeggio_engraver"
}
\relative c' {
  \set Staff.connectArpeggios = ##t
  <<
    { <e' g>4\arpeggio <d f> <d f>2 }
    \\
    { <d, f>2\arpeggio <g b>2 }
  >>
}

```



Creating blank staves

To create blank staves, generate empty measures then remove the `Bar_number_engraver` from the `Score` context, and the `Time_signature_engraver`, `Clef_engraver` and `Bar_engraver` from the `Staff` context.

```

#(set-global-staff-size 10)  % for the documentation
% #(set-global-staff-size 20) % for letter and A4

```

```

\book {
  \score {
    { \repeat unfold 12 { s1 \break } }

    \layout {
      indent = 0
      \context {
        \Staff
        \remove "Time_signature_engraver"
        \remove "Clef_engraver"
        \remove "Bar_engraver"
      }
      \context {
        \Score
        \remove "Bar_number_engraver"
      }
    }
  }
}

```

```

% for the documentation

```

```

\paper {
  #(set-paper-size "a6")
  ragged-last-bottom = ##f
  line-width = 90\mm
  left-margin = 7.5\mm
  bottom-margin = 5\mm
  top-margin = 5\mm
  tagline = ##f
}

```

```

% uncomment these lines for "letter" size

```

```

%{
\paper {
  #(set-paper-size "letter")
  ragged-last-bottom = ##f
  line-width = 7.5\in
  left-margin = 0.5\in
  bottom-margin = 0.25\in
  top-margin = 0.25\in
  tagline = ##f
}
}%

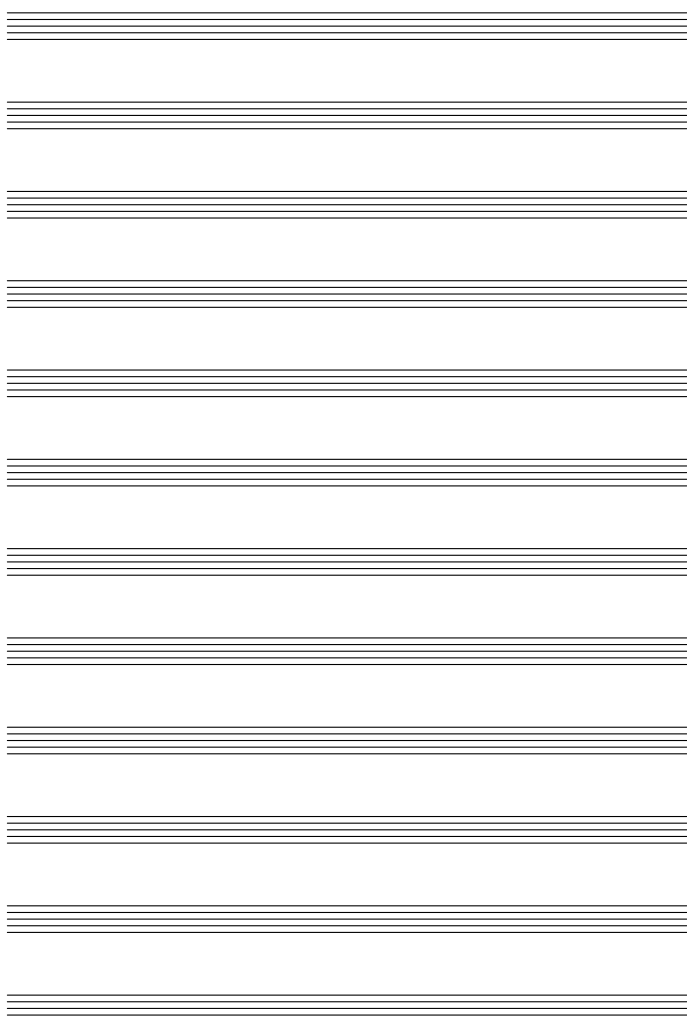
```

```

% uncomment these lines for "A4" size

```

```
%{
\paper {
  #(set-paper-size "a4")
  ragged-last-bottom = ##f
  line-width = 180\mm
  left-margin = 15\mm
  bottom-margin = 10\mm
  top-margin = 10\mm
  tagline = ##f
}
}%
}
```



Arpeggio zwischen Systemen in einem Klaviersystem erstellen

In einem Klaviersystem (`PianoStaff`) ist es möglich, ein Arpeggio zwischen beiden Systemen zu verbinden, indem die `PianoStaff.connectArpeggios`-Eigenschaft gesetzt wird.

```
<<
\new PianoStaff <<
  \new Voice \relative c' {
    <c e>2\arpeggio <d f>2\arpeggio
    <c e>1\arpeggio
```

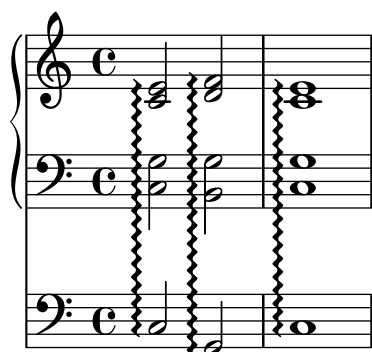
```

    }
    \new Voice \relative c {
      \clef bass
      <c g'>2\arpeggio <b g'>2\arpeggio
      <c g'>1\arpeggio
    }
  >>

  \new Staff \relative c {
    \set Score.connectArpeggios = ##t
    \clef bass
    c2\arpeggio g\arpeggio
    c1\arpeggio
  }
  >>

  \layout {
    \context {
      \Score
      \consists "Span_arpeggio_engraver"
    }
  }
}

```



Creating custom key signatures

LilyPond supports custom key signatures. In this example, print for D minor and D major with an extended range of shown flats.

```

\new Staff \with {
  \override StaffSymbol.line-count = #8
  \override KeySignature.flat-positions = #'((-7 . 6))
  \override KeyCancellation.flat-positions = #'((-7 . 6))
  \override KeySignature.sharp-positions = #'((-6 . 7))
  \override KeyCancellation.sharp-positions = #'((-6 . 7))

  \override Clef.stencil =
    #(lambda (grob)
      (grob-interpret-markup grob
        #{ \markup\combine
          \musicglyph "clefs.C"
          \translate #'(-3 . -2)
          \musicglyph "clefs.F"

```



```

        #}}))
    clefPosition = #3
    middleCPosition = #3
    middleCClefPosition = #3
}

{
  \key d\minor f bes, f bes, |
  \key d\major fis b, fis b, |
}

```



Hälsa über beide Systeme

Dieses Schnipsel zeigt den Einsatz von `Span_stem_engraver` und `\crossStaff`, um Hälsa über Systeme hinweg automatisch zu verbinden. Die Länge des Halses muss definiert werden, da der variable Abstand zwischen Notenköpfen und Notensystemen automatisch errechnet wird.

```

\layout {
  \context {
    \PianoStaff
    \consists "Span_stem_engraver"
  }
}

\new PianoStaff <<
  \new Staff {
    <b d'>4 r d'16\> e'8. g8 r\! |
    e'8 f' g'4
    \voiceTwo
    % Down to lower staff
    \crossStaff { e'8 e'8 } e'4 |
  }

  \new Staff {
    \clef bass
    \voiceOne
    % Up to upper staff
    \crossStaff { <e g>4 e, g16 a8. c8 } d |
    g8 f g4 \voiceTwo g8 g g4 |
  }
}
>>

```



Defining an engraver in Scheme: ambitus engraver

This example demonstrates how the ambitus engraver may be defined on the user side, with a Scheme engraver. This is basically a rewrite in Scheme of the code from `lily/ambitus-engraver.cc`.

```
#(use-modules (oop goops))
```

```
%%%
%%% Grob utilities
%%%
%%% These are literal rewrites of some C++ methods used by the ambitus
%%% engraver.
```

```
#(define (ly:separation-item::add-conditional-item grob grob-item)
  "Add GROB-ITEM to the array of conditional elements of GROB.
```

This is a rewrite of function ``Separation_item::add_conditional_item`` from file ``lily/separation-item.cc``."

```
  (ly:pointer-group-interface::add-grob
    grob 'conditional-elements grob-item))
```

```
#(define (ly:accidental-placement::accidental-pitch accidental-grob)
  "Get the pitch from the grob cause of ACCIDENTAL-GROB.
```

This is a rewrite of function ``accidental_pitch`` from file ``lily/accidental-placement.cc``."

```
  (ly:event-property (ly:grob-property
    (ly:grob-parent accidental-grob Y) 'cause)
    'pitch))
```

```
#(define (ly:accidental-placement::add-accidental grob accidental-grob)
  "Add ACCIDENTAL-GROB to the list of accidentals grobs of GROB.
  ACCIDENTAL-GROB is an `Accidental` grob; GROB is an `AccidentalPlacement`
  grob.
```

This is a rewrite of function ``Accidental_placement::add_accidental`` from file ``lily/accidental-placement.cc``."

```
  (let ((pitch (ly:accidental-placement::accidental-pitch
    accidental-grob)))
    (set! (ly:grob-parent accidental-grob X) grob)
    (let* ((accidentals (ly:grob-object grob 'accidental-grobs))
      (handle (assq (ly:pitch-notename pitch) accidentals))
      (entry (if handle (cdr handle) '()))))
      (set! (ly:grob-object grob 'accidental-grobs)
        (assq-set! accidentals
          (ly:pitch-notename pitch)
          (cons accidental-grob entry))))))
```

```
%%%
%%% Ambitus data structures.
%%%
```

```

%%% The <ambitus> class holds the various grobs that are created to
%%% print an ambitus:
%%%
%%% - `ambitus-group`: the grob that groups all the components of an
%%%   ambitus (`Ambitus` grob);
%%% - `ambitus-line`: the vertical line between the upper and lower
%%%   ambitus notes (`AmbitusLine` grob);
%%% - `ambitus-up-note` and `ambitus-down-note`: the note head and
%%%   accidental for the lower and upper note of the ambitus (see
%%%   `` class below).
%%%
%%% The other slots define the key and clef context of the engraver:
%%%
%%% - `start-c0`: position of middle c at the beginning of the piece.
%%%   It is used to place the ambitus notes according to their pitch;
%%% - `start-key-sig`: the key signature at the beginning of the
%%%   piece. It is used to determine whether accidentals shall be
%%%   printed next to ambitus notes.

#(define-class <ambitus> ()
  (ambitus-group #:accessor ambitus-group)
  (ambitus-line #:accessor ambitus-line)
  (ambitus-up-note #:getter ambitus-up-note
    #:init-form (make <ambitus-note>))
  (ambitus-down-note #:getter ambitus-down-note
    #:init-form (make <ambitus-note>))
  (start-c0 #:accessor ambitus-start-c0
    #:init-value #f)
  (start-key-sig #:accessor ambitus-start-key-sig
    #:init-value '()))

%%% Accessor for the lower and upper note data of an ambitus.
#(define-method (ambitus-note (ambitus <ambitus>) direction)
  "Return lower or upper note of AMBITUS depending on DIRECTION."
  (if (= direction UP)
    (ambitus-up-note ambitus)
    (ambitus-down-note ambitus)))

%%% The `` class holds the grobs that are specific to
%%% ambitus (lower and upper) notes:
%%%
%%% - `head`: an `AmbitusNoteHead` grob;
%%% - `accidental`: an `AmbitusAccidental` grob, to be possibly
%%%   printed next to the ambitus note head.
%%%
%%% Moreover,
%%%
%%% - `pitch` is the absolute pitch of the note;
%%% - `cause` is the note event that causes this ambitus note, i.e.,
%%%   the lower or upper note of the considered music sequence.

#(define-class <ambitus-note> ()

```

```

(head #:accessor ambitus-note-head
  #:init-value #f)
(accidental #:accessor ambitus-note-accidental
  #:init-value #f)
(cause #:accessor ambitus-note-cause
  #:init-value #f)
(pitch #:accessor ambitus-note-pitch
  #:init-value #f))

%%%
%%% Ambitus engraving logic.
%%%
%%% This is rewrite of the code from file `lily/ambitus-engraver.cc`.

```

```

#(define (make-ambitus translator)
  "Build an ambitus object: initialize all the grobs and their
relations."

```

The `Ambitus` grob contains all other grobs:

```

Ambitus
|- AmbitusLine
|- AmbitusNoteHead    for upper note
|- AmbitusAccidental  for upper note
|- AmbitusNoteHead    for lower note
|- AmbitusAccidental  for lower note

```

The parent of an accidental is the corresponding note head, and the accidental is set as the `accidental-grob` property of the note head so that is printed by the function that prints notes."

```

;; Make the ambitus object.
(let ((ambitus (make <ambitus>)))
  ;; Build the `Ambitus` grob, which will contain all other grobs.
  (set! (ambitus-group ambitus)
    (ly:engraver-make-grob translator 'Ambitus '()))
  ;; Build the `AmbitusLine` grob (the line between lower and upper
  ;; note).
  (set! (ambitus-line ambitus)
    (ly:engraver-make-grob translator 'AmbitusLine '()))
  ;; Build the upper and lower `AmbitusNoteHead` and
  ;; `AmbitusAccidental`.
  (for-each
    (lambda (direction)
      (let ((head (ly:engraver-make-grob translator
        'AmbitusNoteHead '()))
            (accidental (ly:engraver-make-grob translator
              'AmbitusAccidental '())))
        (group (ambitus-group ambitus)))
      ;; The parent of the `AmbitusAccidental` grob is the
      ;; `AmbitusNoteHead` grob.
      (set! (ly:grob-parent accidental Y) head)
      ;; The `AmbitusAccidental` grob is set as the

```

```

;; `accidental-grob` object of `AmbitusNoteHead`. This is
;; later used by the function that prints notes.
(set! (ly:grob-object head 'accidental-grob) accidental)
;; Both the note head and the accidental grobs are added to
;; the main ambitus grob.
(ly:axis-group-interface::add-element group head)
(ly:axis-group-interface::add-element group accidental)
;; The note head and the accidental grobs are added to the
;; ambitus object.
(set! (ambitus-note-head (ambitus-note ambitus direction))
      head)
(set! (ambitus-note-accidental (ambitus-note ambitus direction))
      accidental)))
(list DOWN UP))

;; The parent of the ambitus line is the lower ambitus note head.
(set! (ly:grob-parent (ambitus-line ambitus) X)
      (ambitus-note-head (ambitus-note ambitus DOWN)))
;; The ambitus line is added to the ambitus main grob.
(ly:axis-group-interface::add-element (ambitus-group ambitus)
                                       (ambitus-line ambitus))
ambitus))

#(define-method (initialize-ambitus-state
                 (ambitus <ambitus>) translator)
  "Initialize the state of AMBITUS by getting the starting position of
  middle C and key signature from TRANSLATOR's context."
  (if (not (ambitus-start-c0 ambitus))
      (begin
        (set! (ambitus-start-c0 ambitus)
              (ly:context-property (ly:translator-context translator)
                                  'middleCPosition 0))
        (set! (ambitus-start-key-sig ambitus)
              (ly:context-property (ly:translator-context translator)
                                  'keyAlterations))))))

#(define-method (update-ambitus-notes (ambitus <ambitus>) note-grob)
  "Update upper and lower ambitus pitches of AMBITUS using NOTE-GROB."
  ;; Get the event that caused the `note-grob` creation and check
  ;; that it is a `note-event`.
  (let ((note-event (ly:grob-property note-grob 'cause)))
    (if (ly:in-event-class? note-event 'note-event)
        ;; Get the pitch from the note event.
        (let ((pitch (ly:event-property note-event 'pitch)))
          ;; If this pitch is lower than the current ambitus' lower
          ;; note pitch (or it has not been initialized yet), then
          ;; this pitch is the new ambitus' lower pitch. The same is
          ;; done for the upper pitch (but in the opposite
          ;; direction).
          (for-each
            (lambda (direction pitch-compare)
              (if (or (not (ambitus-note-pitch

```

```
(ambitus-note ambitus direction)))
      (pitch-compare
        pitch (ambitus-note-pitch
          (ambitus-note ambitus direction))))))
(begin
  (set! (ambitus-note-pitch
    (ambitus-note ambitus direction))
    pitch)
  (set! (ambitus-note-cause
    (ambitus-note ambitus direction))
    note-event))))
(list DOWN UP)
(list ly:pitch<?
  (lambda (p1 p2) (ly:pitch<? p2 p1))))))

#(define-method (typeset-ambitus (ambitus <ambitus>) translator)
  "Typeset AMBITUS.

- Place the lower and upper ambitus notes according to their pitch and
the position of the middle C.
- Typeset or delete the note accidentals, according to the key
signature. An accidental, if it is to be printed, is added to an
`AccidentalPlacement` grob (a grob dedicated to the placement of
accidentals near a chord).
- Both note heads are added to the ambitus line grob so that a line
gets printed between them."
  ;; Check whether there are lower and upper pitches.
  (if (and (ambitus-note-pitch (ambitus-note ambitus UP))
    (ambitus-note-pitch (ambitus-note ambitus DOWN)))
    ;; Make an `AccidentalPlacement` grob, for placement of note
    ;; accidentals.
    (let ((accidental-placement
      (ly:engraver-make-grob
        translator
        'AccidentalPlacement (ambitus-note-accidental
          (ambitus-note ambitus DOWN)))))
      ;; For lower and upper ambitus notes.
      (for-each
        (lambda (direction)
          (let ((pitch (ambitus-note-pitch
            (ambitus-note ambitus direction))))
            ;; Set the cause and the staff position of the ambitus
            ;; note according to the associated pitch.
            (set! (ly:grob-property
              (ambitus-note-head (ambitus-note ambitus direction))
              'cause)
              (ambitus-note-cause (ambitus-note ambitus direction)))
              (set! (ly:grob-property
                (ambitus-note-head (ambitus-note ambitus direction))
                'staff-position)
                (+ (ambitus-start-c0 ambitus)
                  (ly:pitch-steps pitch))))
```



```

        (list DOWN UP))
      (ly:grob-suicide! ambitus-line))))

%%%
%%% Ambitus engraver definition.
%%%
#(define ambitus-engraver
  (lambda (context)
    (let ((ambitus #f))
      ;; When music is processed, make the ambitus object if not
      ;; already built.
      (make-engraver
        ((process-music translator)
         (if (not ambitus)
             (set! ambitus (make-ambitus translator))))

        ;; Set the ambitus clef and key signature state.
        ((stop-translation-timestep translator)
         (if ambitus
             (initialize-ambitus-state ambitus translator)))

        ;; When a note head grob is built, update the ambitus notes.
        (acknowledgers
         ((note-head-interface engraver grob source-engraver)
          (if ambitus
              (update-ambitus-notes ambitus grob)))))

      ;; Finally, typeset the ambitus according to its upper and
      ;; lower notes (if any).
      ((finalize translator)
       (if ambitus
           (typeset-ambitus ambitus translator)))))))

%%%
%%% Example
%%%

\score {
  \new StaffGroup <<
    \new Staff { c'4 des' e' fis' gis' }
    \new Staff { \clef "bass" c4 des ~ des ees b, }
  >>
  \layout { \context { \Staff \consists #ambitus-engraver } }
}

```



Displaying a whole GrandStaff system if only one of its staves is alive

In many orchestral scores it is custom to not show staves for instruments that are silent for a while; this is called a ‚Frenched‘ score. LilyPond provides this functionality via the `\RemoveEmptyStaves` command.

When they play again it is often preferred to show the staves of *all instruments of such a group*. This can be done by adding the `Keep_alive_together_engraver` to the grouping context (e.g., `GrandStaff` or `StaffGroup`).

In the example below the violins are silent in the second system. Only the first violin plays the last measure in the third system but the staff of the second violin is also displayed.

```
\score {
  <<
    \new Staff = "Staff_flute" \with {
      instrumentName = "Flute"
      shortInstrumentName = "Fl"
    } \relative c' {
      \repeat unfold 3 { c'4 c c c | c c c c | c c c c | \break }
    }

    \new StaffGroup = "StaffGroup_Strings" <<
      \new GrandStaff = "GrandStaff_violins" <<
        \new Staff = "StaffViolinI" \with {
          instrumentName = "Violin I"
          shortInstrumentName = "Vi I"
        } \relative c'' {
          a1 | R1*7 | \repeat unfold 12 a16 a4 |
        }
        \new Staff = "StaffViolinII" \with {
          instrumentName = "Violin II"
          shortInstrumentName = "Vi II"
        } \relative c' {
          e1 | R1*8 |
        }
      >>
    >>

    \new Staff = "Staff_cello" \with {
      instrumentName = "Cello"
      shortInstrumentName = "Ce"
    } \relative c {
      \clef bass \repeat unfold 9 { c1 } |
    }
  >>
}

\layout {
  indent = 3.0\cm
  short-indent = 1.5\cm

  \context {
    \GrandStaff
    \consists Keep_alive_together_engraver
```

```

}
\context {
  \Staff
  \RemoveEmptyStaves
}
}

```

The image displays three musical score excerpts, each showing the effect of switching engravers one by one. The first excerpt shows a Flute, Violin I, Violin II, and Cello. The second excerpt shows Flute and Cello. The third excerpt shows Flute, Violin I, Violin II, and Cello, with Violin I playing a sixteenth-note pattern in the third measure.

Engravers one by one

LilyPond handles the various elements necessary to typeset a score with plugins. Each plugin is called an *engraver*. In this example, (some) engravers are switched on one by one, in the following order:

- note heads,
- staff symbol,
- clef,
- stem,

- beams, slurs, accents,
- accidentals, bar lines, time signature, and key signature.

Engravers are grouped. For example, note heads, slurs, beams, etc., form a Voice context. Engravers for key signature, accidentals, bar line, etc., form a Staff context.

```

topVoice = \relative c' {
  \key d \major
  es8([ g] a[ fis])
  b4
  b16[-. b-. b-. cis-.]
  d4->
}

% empty staff and voice contexts
MyStaff = \context {
  \type Engraver_group
  \name Staff
  \accepts Voice
  \defaultchild Voice
}

MyVoice = \context {
  \type Engraver_group
  \name Voice
}

% add note heads
MyVoice = \context {
  \MyVoice
  \consists Note_heads_engraver
}

\score {
  \topVoice
  \layout {
    \context { \MyStaff }
    \context { \MyVoice }
  }
}

% add staff
MyStaff = \context {
  \MyStaff
  \consists Staff_symbol_engraver
}

\score {
  \topVoice
  \layout {
    \context { \MyStaff }
    \context { \MyVoice }
  }
}

% add clef

```

```

MyStaff = \context {
  \MyStaff
  \consists Clef_engraver
}

\score {
  \topVoice
  \layout {
    \context { \MyStaff }
    \context { \MyVoice }
  }
}

% add stems
MyVoice = \context {
  \MyVoice
  \consists Stem_engraver
}

\score {
  \topVoice
  \layout {
    \context { \MyStaff }
    \context { \MyVoice }
  }
}

% add beams, slurs, and accents
MyVoice = \context {
  \MyVoice
  \consists Beam_engraver
  \consists Slur_engraver
  \consists Script_engraver
  \consists Rhythmic_column_engraver
}

\score {
  \topVoice
  \layout {
    \context { \MyStaff }
    \context { \MyVoice }
  }
}

% add accidentals, bar, time signature, and key signature
MyStaff = \context {
  \MyStaff
  \consists Accidental_engraver
  \consists Bar_engraver
  \consists Time_signature_engraver
  \consists Key_engraver
}

\score {
  \topVoice
  \layout {

```

```

\context { \MyStaff }
\context { \MyVoice }
}
}

```



Gitternetzlinien: Aussehen verändern

Die Erscheinung der Gitternetzlinien kann durch einige Eigenschaften geändert werden.

```

\new ChoirStaff <<
  \new Staff {
    \relative c' ' {
      \stemUp
      c'4. d8 e8 f g4
    }
  }
  \new Staff {
    \relative c {
      % this moves them up one staff space from the default position
      \override Score.GridLine.extra-offset = #'(0.0 . 1.0)
      \stemDown
      \clef bass
      \once \override Score.GridLine.thickness = 5.0
      c4
      \once \override Score.GridLine.thickness = 1.0
      g'4
      \once \override Score.GridLine.thickness = 3.0
      f4
      \once \override Score.GridLine.thickness = 5.0
      e4
    }
  }
}
>>

```

```

\layout {
  \context {
    \Staff
    % set up grids
    \consists "Grid_point_engraver"
    % set the grid interval to one quarter note
    gridInterval = #1/4
  }
  \context {
    \Score
    \consists "Grid_line_span_engraver"
    % this moves them to the right half a staff space
    \override NoteColumn.X-offset = -0.5
  }
}

```



Grid lines: emphasizing rhythms and notes synchronization

Regular vertical lines can be drawn between staves to show note synchronization; however, in case of monophonic music, you may want to make the second staff invisible, and make the lines shorter like in this snippet.

```

\new ChoirStaff {
  \relative c'' <<
  \new Staff {
    \time 12/8
    \stemUp
    c4. d8 e8 f g4 f8 e8. d16 c8
  }
  \new Staff {
    % hides staff and notes so that only the grid lines are visible
    \hideNotes
    \hide Staff.BarLine
    \override Staff.StaffSymbol.line-count = #0
    \hide Staff.TimeSignature
    \hide Staff.Clef

    % dummy notes to force regular note spacing
    \once \override Score.GridLine.thickness = #4.0
    c8 c c
    \once \override Score.GridLine.thickness = #3.0
    c8 c c
    \once \override Score.GridLine.thickness = #4.0
    c8 c c
  }
}

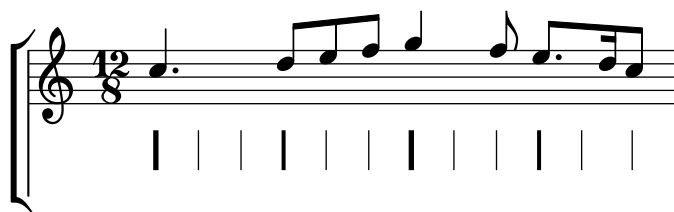
```

```

        \once \override Score.GridLine.thickness = #3.0
        c8 c c
    }
    >>
}

\layout {
  \context {
    \Score
    \consists "Grid_line_span_engraver"
    % center grid lines horizontally below note heads
    \override NoteColumn.X-offset = #-0.5
  }
  \context {
    \Staff
    \consists "Grid_point_engraver"
    gridInterval = #1/8
    % set line length and positioning:
    % two staff spaces above center line on hidden staff
    % to four spaces below center line on visible staff
    \override GridPoint.Y-extent = #'(2 . -4)
  }
}

```



Measure counters

This snippet demonstrates the use of the `Measure_counter_engraver` to number groups of successive measures. Any stretch of measures may be numbered, whether consisting of repetitions or not.

The engraver must be added to the appropriate context. Here, a `Staff` context is used; another possibility is a `Dynamics` context.

The counter is begun with `\startMeasureCount` and ended with `\stopMeasureCount`. Numbering will start by default with 1, but this behavior may be modified by overriding the `count-from` property.

When a measure extends across a line break, the number will appear twice, the second time in parentheses.

```

\layout {
  \context {
    \Staff
    \consists #Measure_counter_engraver
  }
}

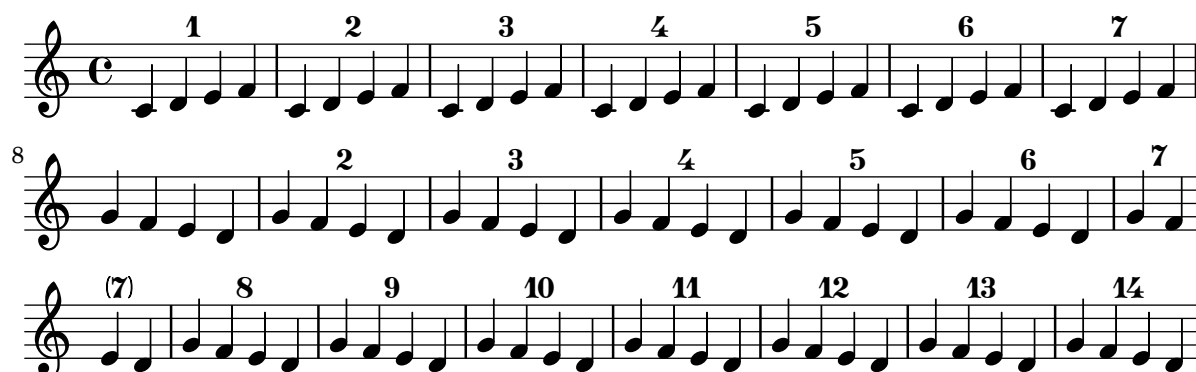
\new Staff {

```

```

\startMeasureCount
\repeat unfold 7 {
  c'4 d' e' f'
}
\stopMeasureCount
\bar "||"
g'4 f' e' d'
\override Staff.MeasureCounter.count-from = #2
\startMeasureCount
\repeat unfold 5 {
  g'4 f' e' d'
}
g'4 f'
\bar ""
\break
e'4 d'
\repeat unfold 7 {
  g'4 f' e' d'
}
\stopMeasureCount
}

```



Measure spanner

Measure spanners are an alternate way to print annotated brackets. As opposed to horizontal brackets, they extend between two bar lines rather than two notes. The text is displayed in the center of the bracket.

```

\layout {
  \context {
    \Staff
    \consists Measure_spanner_engraver
  }
}

```

```

<<
\new Staff \relative c'' {
  \key d \minor
  R1*2
  \tweak text "Answer"
  \startMeasureSpanner
  \tuplet 3/2 8 {

```



```

    a16[ b c] d[ c b]  c[ d e] f[ e d]
  }
  e8 a gis g
  fis f e d~ d c b e
  \stopMeasureSpanner
}
\new Staff \relative c' {
  \key d \minor
  \tweak text "Subject"
  \tweak direction #DOWN
  \startMeasureSpanner
  \tuplet 3/2 8 {
    d16[ e f] g[ f e] f[ g a] bes[ a g]
  }
  a8 d cis c
  b bes a g~ g f e a
  \stopMeasureSpanner
  \tweak text "Counter-subject"
  \tweak direction #DOWN
  \startMeasureSpanner
  f8 e a r r16 b, c d e fis g e
  a gis a b c fis, b a gis e a4 g8
  \stopMeasureSpanner
}
>>

```

The image displays a musical score for three systems. The first system consists of a single staff with a treble clef and a key signature of one flat (B-flat). The second system is a two-staff system with a treble clef and a key signature of one flat. The third system is a two-staff system with a treble clef and a key signature of one flat. The first system is labeled 'Subject' and the second system is labeled 'Answer'. The third system is labeled 'Counter-subject'.

Mensurstriche-Layout (Taktstriche zwischen den Systemen)

Das Mensurstriche-Layout, in welchem die Taktlinien nicht auf den Systemen, sondern zwischen den Systemen gesetzt werden, kann mit einer `StaffGroup` anstelle von `ChoirStaff` erreicht werden. Die Taktlinien auf den Systemen werden mit der `transparent`-Eigenschaft ausgelöscht.

```

\layout {
  \context {
    \Staff
    measureBarType = "-span|"
  }
}

```

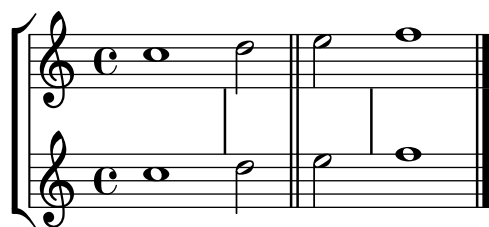
```

    }
  }

  music = \fixed c'' {
    c1
    d2 \section e2
    f1 \fine
  }

  \new StaffGroup <<
    \new Staff \music
    \new Staff \music
  >>

```



Systeme schachteln

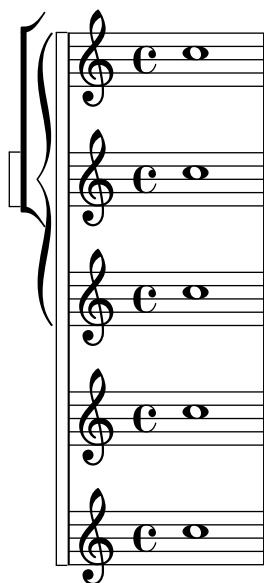
Die Eigenschaft `systemStartDelimiterHierarchy` kann eingesetzt werden, um komplizierte geschachtelte Systemklammern zu erstellen. Der Befehl `\set StaffGroup.systemStartDelimiterHierarchy` nimmt eine Liste mit der Anzahl der Systeme, die ausgegeben werden, auf. Vor jedem System kann eine Systemanfangsklammer angegeben werden. Sie muss in Klammern eingefügt werden und umfasst so viele Systeme, wie die Klammer einschließt. Elemente in der Liste können ausgelassen werden, aber die erste Klammer umfasst immer die gesamte Gruppe. Die Möglichkeiten der Anfangsklammer sind: `SystemStartBar`, `SystemStartBracket`, `SystemStartBrace` und `SystemStartSquare`.

```

\new StaffGroup
\relative c'' <<
  \override StaffGroup.SystemStartSquare.collapse-height = 4
  \set StaffGroup.systemStartDelimiterHierarchy
    = #'(SystemStartSquare
          (SystemStartBrace
            (SystemStartBracket a
              (SystemStartSquare b))
            c)
          d)

  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }
>>

```



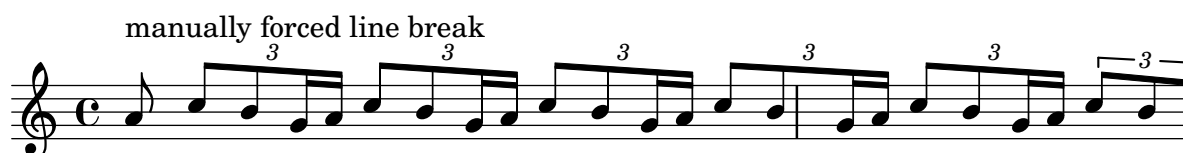
Zeilenumbrüche bei N-tolen mit Balken erlauben

Dieses künstliche Beispiel zeigt, wie sowohl automatische als auch manuelle Zeilenumbrüche innerhalb einer N-tole mit Balken erlaubt werden können. Diese unregelmäßige Bebalkung muss allerdings manuell gesetzt werden.

```
\layout {
  \context {
    \Voice
    % Permit automatic line breaks within tuplets.
    \remove "Forbid_line_break_engraver"
    % Allow beams to be broken at line breaks.
    \override Beam.breakable = ##t
  }
}
```

```
\relative c' {
  <>^"manually forced line break"
  a8
  \repeat unfold 5 { \tuplet 3/2 { c8[ b g16 a] } }
  \tuplet 3/2 { c8[ b \break g16 a] }
  \repeat unfold 5 { \tuplet 3/2 { c8[ b g16 a] } }
  c8 \bar "||"
}
```

```
\relative c' {
  <>^"automatic line break"
  \repeat unfold 28 a16
  \tuplet 11/8 { a16[ b c d e f e d c b a] }
  \repeat unfold 28 a16 \bar "||"
}
```





Print chord names with same root and different bass as slash and bass note

To print subsequent ChordNames only differing in its bass note as slash and bass note, use the Scheme engraver defined in this snippet. The behaviour may be controlled in detail by the chordChanges context property.

```
#(define Bass_changes_equal_root_engraver
  (lambda (ctx)
    "For sequential `ChordNames` with the same root but a different bass,
    the root markup is dropped: D D/C D/B -> D /C /B.
    The behaviour may be controlled by setting the `chordChanges` context
    property."
    (let ((chord-pitches '())
          (last-chord-pitches '())
          (bass-pitch #f))
      (make-engraver
        ((initialize this-engraver)
         (let ((chord-note-namer (ly:context-property ctx
                                                    'chordNoteNamer)))
           ;; Set 'chordNoteNamer, respect user setting if already done
           (ly:context-set-property! ctx 'chordNoteNamer
                                     (if (procedure? chord-note-namer)
                                         chord-note-namer
                                         note-name->markup))))
        (listeners
         ((note-event this-engraver event)
          (let* ((pitch (ly:event-property event 'pitch))
                 (pitch-name (ly:pitch-notename pitch))
                 (pitch-alt (ly:pitch-alteration pitch))
                 (bass (ly:event-property event 'bass #f))
                 (inversion (ly:event-property event 'inversion #f)))
            ;; Collect notes of the chord
            ;; - to compare inversed chords we need to collect the
            ;;   bass note as usual member of the chord, whereas an
            ;;   added bass must be treated separate from the usual
            ;;   chord-notes
            ;; - notes are stored as pairs containing their
            ;;   pitch-name (an integer), i.e. disregarding their
```

```

;; octave and their alteration
(cond (bass (set! bass-pitch pitch))
      (inversion
       (set! bass-pitch pitch)
       (set! chord-pitches
        (cons (cons pitch-name pitch-alt)
              chord-pitches)))
      (else
       (set! chord-pitches
        (cons (cons pitch-name pitch-alt)
              chord-pitches))))))

(acknowledgers
 ((chord-name-interface this-engraver grob source-engraver)
  (let ((chord-changes (ly:context-property ctx
                                           'chordChanges #f)))
    ;; If subsequent chords are equal apart from their bass,
    ;; reset the 'text-property.
    ;; Equality is done by comparing the sorted lists of this
    ;; chord's elements and the previous chord. Sorting is
    ;; needed because inverted chords may have a different
    ;; order of pitches. `chord-changes` needs to be true.
    (if (and bass-pitch
              chord-changes
              (equal?
               (sort chord-pitches car<)
               (sort last-chord-pitches car<)))
        (ly:grob-set-property!
         grob 'text
         (make-line-markup
          (list
           (ly:context-property ctx 'slashChordSeparator)
           ((ly:context-property ctx 'chordNoteNamer)
            bass-pitch
            (ly:context-property ctx
                                'chordNameLowercaseMinor))))))
        (set! last-chord-pitches chord-pitches)
        (set! chord-pitches '())
        (set! bass-pitch #f))))

((finalize this-engraver)
 (set! last-chord-pitches '()))))

myChords = \chordmode {
  % \germanChords

  \set chordChanges = ##t
  d2:m d:m/cis

  d:m/c
  \set chordChanges = ##f

```

```

d:m/b

e1:7
\set chordChanges = ##t
e
\break

\once \set chordChanges = ##f
e1/f
e2/gis e/+gis e e:m/f d:m d:m/cis d:m/c
\set chordChanges = ##f
d:m/b
}

<<
\new ChordNames
  \with { \consists #Bass_changes_equal_root_engraver }
  \myChords
\new Staff \myChords
>>

```

Zeichen über jedem System ausgeben

Normalerweise werden Textzeichen nur über dem obersten Notensystem gesetzt. Sie können aber auch über jedem System ausgegeben werden.

```

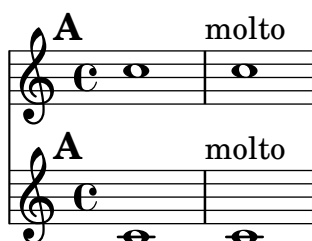
\score {
  <<
    \new Staff { \mark \default c''1 \textMark "molto" c'' }
    \new Staff { \mark \default c'1 \textMark "molto" c' }
  >>
  \layout {
    \context {
      \Score
      \remove Mark_engraver
      \remove Text_mark_engraver
      \remove Staff_collecting_engraver
    }
    \context {
      \Staff
      \consists Mark_engraver
      \consists Text_mark_engraver
    }
  }
}

```

```

    \consists Staff_collecting_engraver
  }
}
}

```



Printing music with different time signatures

In the following snippet, two parts have a completely different time signature, yet remain synchronized.

The bar lines can no longer be printed at the Score level; to allow independent bar lines in each part, the `Default_barline_engraver` and `Timing_translator` are moved from the `Score` context to the `Staff` context.

If bar numbers are required, the `Bar_number_engraver` should also be moved, since it relies on properties set by the `Timing_translator`; a `\with` block can be used to add bar numbers to the relevant staff.

```

global = {
  \time 3/4 s2.*3 \break
  s2.*3
}

\layout {
  \context {
    \Score
    \remove "Timing_translator"
    \remove "Bar_number_engraver"
    \override SpacingSpanner.uniform-stretching = ##t
    \override SpacingSpanner.strict-note-spacing = ##t
    proportionalNotationDuration = #1/64
  }
  \context {
    \Staff
    \consists "Timing_translator"
  }
  \context {
    \Voice
    \remove "Forbid_line_break_engraver"
    tupletFullLength = ##t
  }
}

Bassklarinette = \new Staff \with {
  \consists "Bar_number_engraver"
  barNumberVisibility = #(every-nth-bar-number-visible 2)
  \override BarNumber.break-visibility = #end-of-line-invisible
}

```

```

} <<
\global
{
  \clef treble
  \time 3/8 d''4. |
  \time 3/4 r8 des''2( c''8) |
  \time 7/8 r4. ees''2 ~ |
  \time 2/4 \tupletUp \tuplet 3/2 { ees''4 r4 d''4 ~ } |
  \time 3/8 \tupletUp \tuplet 4/3 { d''4 r4 } |
  \time 2/4 e''2 |
  \time 3/8 es''4. |
  \time 3/4 r8 d''2 r8 |
}
>>

Perkussion = \new StaffGroup <<
\new Staff <<
\global
{
  \clef percussion
  \time 3/4 r4 c'2 ~ |
  c'2. |
  R2. |
  r2 g'4 ~ |
  g'2. ~ |
  g'2. |
}
>>
\new Staff <<
\global {
  \clef percussion
  \time 3/4 R2. |
  g'2. ~ |
  g'2. |
  r4 g'2 ~ |
  g'2 r4 |
  g'2. |
}
>>
>>

\score {
  <<
    \Bassklarinette
    \Perkussion
  >>
}

```


The image shows three systems of musical notation. Each system consists of a treble staff and a grand staff (two staves). The first system has measures with various time signatures (3/8, 3/4, 7/8, 2/4) and fingerings (2, 4, 3). The second system is marked with a (4) and has measures with time signatures (3/8, 3/4, 3/8, 3/4) and fingerings (3, 4, 6). The third system is marked with an 8 and has measures with time signatures (3/4, 3/4).

Entfernung von Taktnummern in einer Partitur

Taktnummern können vollkommen aus den Noten entfernt werden, indem man den `Bar_number_engraver` aus dem `Score`-Kontext entfernt.

```
\layout {
  \context {
    \Score
    \omit BarNumber
    % or:
    % \remove "Bar_number_engraver"
  }
}

\relative c' {
  c4 c c c \break
  c4 c c c
}
```

The image shows a single staff of music with a treble clef and a common time signature (C). The staff contains four measures, each with a quarter note (c4).



Eine eckige Klammer zu Beginn von Systemgruppen benutzen

Die Klammer zu Beginn von Systemgruppen kann auch in eine eckige Klammer (SystemStartSquare) umgewandelt werden, wenn man sie explizit im StaffGroup- oder ChoirStaffGroup-Kontext setzt.

```
\score {
  \new StaffGroup { <<
    \set StaffGroup.systemStartDelimiter = #'SystemStartSquare
    \new Staff { c'4 d' e' f' }
    \new Staff { c'4 d' e' f' }
  >> }
}
```



Using mark lines in a Frenched score

Using MarkLine contexts (such as in „Placing rehearsal marks other than above the top staff“) in a Frenched score can be problematic if all the staves between two MarkLines are removed in one system. The Keep_alive_together_engraver can be used within each StaffGroup to keep the MarkLine alive only as long as the other staves in the group stay alive.

```
bars = {
  \tempo "Allegro" 4=120
  s1*2
  \repeat unfold 5 { \mark \default s1*2 }
  \bar "||"
  \tempo "Adagio" 4=40
  s1*2
  \repeat unfold 8 { \mark \default s1*2 }
  \bar "|."
}

winds = \repeat unfold 120 { c''4 }
trumpet = { \repeat unfold 8 g'2 R1*16 \repeat unfold 4 g'2 R1*8 }
trombone = { \repeat unfold 4 c'1 R1*8 d'1 R1*17 }
strings = \repeat unfold 240 { c''8 }

#(set-global-staff-size 16)
\paper {
  systems-per-page = 5
  ragged-last-bottom = ##f
  tagline = ##f
}
```

```

\layout {
  indent = 16\mm
  short-indent = 5\mm
  \context {
    \name MarkLine
    \type Engraver_group
    \consists Output_property_engraver
    \consists Axis_group_engraver
    \consists Mark_engraver
    \consists Metronome_mark_engraver
    \consists Staff_collecting_engraver
    \override VerticalAxisGroup.remove-empty = ##t
    \override VerticalAxisGroup.remove-layer = #'any
    \override VerticalAxisGroup.staff-affinity = #DOWN
    \override VerticalAxisGroup.nonstaff-relatedstaff-spacing.padding = 1
    keepAliveInterfaces = #'()
  }
  \context {
    \Staff
    \override VerticalAxisGroup.remove-empty = ##t
    \override VerticalAxisGroup.remove-layer = ##f
  }
  \context {
    \StaffGroup
    \accepts MarkLine
    \consists Keep_alive_together_engraver
  }
  \context {
    \Score
    \remove Mark_engraver
    \remove Metronome_mark_engraver
    \remove Staff_collecting_engraver
    \override BarNumber.Y-offset = #3
  }
}

\score {
  <<
  \new StaffGroup = "winds" \with {
    instrumentName = "Winds"
    shortInstrumentName = "W."
  } <<
  \new MarkLine \bars
  \new Staff \winds
  >>
  \new StaffGroup = "brass" <<
  \new MarkLine \bars
  \new Staff = "trumpet" \with {
    instrumentName = "Trumpet"
    shortInstrumentName = "Tp."
  } \trumpet
  \new Staff = "trombone" \with {

```

```

    instrumentName = "Trombone"
    shortInstrumentName = "Tb."
  } \trombone
>>
\new StaffGroup = "strings" \with {
  instrumentName = "Strings"
  shortInstrumentName = "Str."
} <<
  \new MarkLine \bars
  \new Staff = "strings" { \strings }
>>
>>
}

```

The musical score is divided into sections A through H. The tempo is **Allegro** (♩ = 120) for sections A, B, C, D, and E. The tempo changes to **Adagio** (♩ = 40) for sections F, G, and H. The score is written for Winds, Trumpet, Trombone, and Strings. The strings play a continuous eighth-note pattern throughout the piece.

Section A (Measures 1-5): Winds and Trumpet play a melody, while Trombone and Strings play a continuous eighth-note pattern.

Section B (Measures 6-10): Winds and Trumpet play a melody, while Trombone and Strings play a continuous eighth-note pattern.

Section C (Measures 11-15): Winds and Trumpet play a melody, while Trombone and Strings play a continuous eighth-note pattern.

Section D (Measures 16-20): Winds and Trumpet play a melody, while Trombone and Strings play a continuous eighth-note pattern.

Section E (Measures 21-25): Winds and Trumpet play a melody, while Trombone and Strings play a continuous eighth-note pattern.

Section F (Measures 26-30): Winds and Trumpet play a melody, while Trombone and Strings play a continuous eighth-note pattern.

Section G (Measures 31-35): Winds and Trumpet play a melody, while Trombone and Strings play a continuous eighth-note pattern.

Section H (Measures 36-40): Winds and Trumpet play a melody, while Trombone and Strings play a continuous eighth-note pattern.

The image shows a musical score snippet with two systems of staves. The first system (measures 21-25) includes staves for W. (Waldhorn), Tp. (Trumpet), and Str. (Strings). The second system (measures 26-30) includes staves for W. (Waldhorn) and Str. (Strings). The notation is a mix of mensural (labeled J, K, L, M, N) and modern (labeled M, N) styles.

Using tags to produce mensural and modern music from the same source

Using tags it is possible to produce both mensural and modern notation from the same music. In this snippet, a function `\menrest` is introduced, allowing mensural rests to be pitched as in the original, but with modern rests in the standard staff position.

Tags can also be used where other differences are needed: for example using „whole measure rests“ (`R1`, `R\breve`, etc.) in modern music, but normal rests (`r1`, `r\breve`, etc.) in the mensural version. Converting mensural music to its modern equivalent is usually referred to as *transcription*.

The call `c4.\Be c8 c\Am` is the same as `c4.[c8 c]`. However, it suppresses warnings if it starts on a note that can't hold a beam but needs it anyway due to the use of `Completion_heads_engraver`.

[Note that the custos sticks out into the right margin and might be cut off if the LilyPond output gets cropped tightly. The use of `\with-true-dimensions` below avoids this.]

```
\layout {
  line-width = 150\mm
}

menrest = #(define-music-function (note) (ly:music?)
  #{
    \tag #'mens $(make-music 'RestEvent note)
    \tag #'mod $(make-music 'RestEvent note 'pitch '())
  #})

Be = \tag #'mod
  #(begin
    (ly:expect-warning (G_ "stem does not fit in beam"))
    (ly:expect-warning (G_ "beam was started here"))
    (make-span-event 'BeamEvent START))

Am = \tag #'mod ]

MenStyle = {
  \override Score.BarNumber.transparent = ##t
  \override Stem.neutral-direction = #up
```

```

\omit Slur
\omit Beam
}

finalis = \section

Music = \relative c'' {
  \key f \major
  g1 d'2 \menrest bes4 bes a2 \menrest r4 g4 fis4. fis8 fis4 fis \break
  g e f4.([ g8] a4[ g8 f] g2.\Be fis8 e\Am fis2) g\breve \finalis
}

MenLyr = \lyricmode {
  So farre, deere life, deare life,
  from thy bright beames ab- en- ted,
}

ModLyr = \lyricmode {
  So far, dear life, dear life,
  from your bright beams ab -- sen -- ted, __
}

\markup \with-true-dimensions % work around a cropping issue
\score {
  \keepWithTag #'mens {
    <<
    \new PetrucciStaff {
      \new PetrucciVoice = "Cantus" {
        \clef "petrucci-c1" \time 4/4 \MenStyle \Music
      }
    }
    \new Lyrics \lyricsto "Cantus" \MenLyr
  } >>
}
\layout {
  \context {
    \PetrucciVoice
    % No longer necessary starting with version 2.25.23.
    \override Flag.style = #'mensural
  }
}

\markup\vspace #1

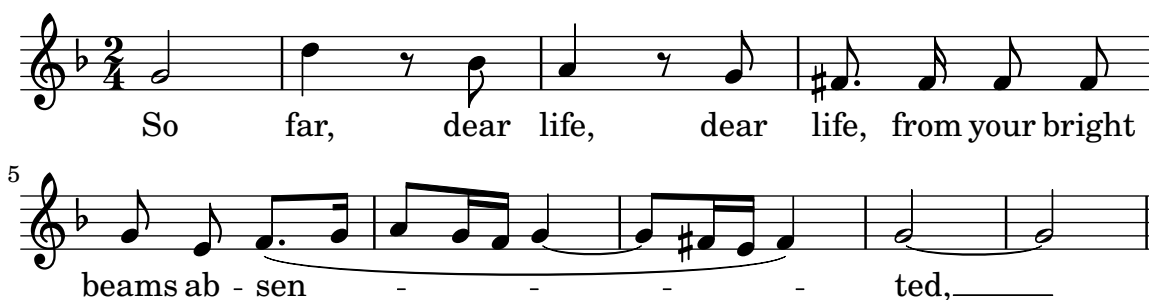
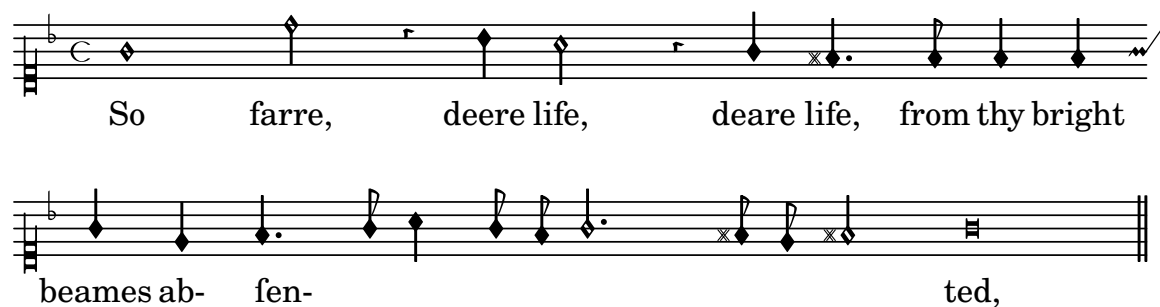
\score {
  \keepWithTag #'mod {
    \new ChoirStaff <<
    \new Staff {
      \new Voice = "Sop" \with {
        \remove "Note_heads_engraver"
        \consists "Completion_heads_engraver"
        \remove "Rest_engraver"
      }
    }
  } >>
}

```

```

\consists "Completion_rest_engraver"
} \shiftDurations 1 0 { \time 2/4 \autoBeamOff \Music }
}
\new Lyrics \lyricsto "Sop" \ModLyr
>>
}
}

```



Sologesang und zweistimmiger Refrain

Diese Vorlage erstellt eine Partitur, die mit Sologesang beginnt und einen Refrain für zwei Stimmen enthält. Sie zeigt auch die Benutzung von Platzhalter-Pausen innerhalb der `\global`-Variable, um Taktwechsel (und andere Elemente, die für alle Stimmen gleich sind) für das gesamte Stück zu definieren.

```

global = {
  \key g \major

```

```

% verse
\time 3/4
s2.*2
\break

```

```

% refrain
\time 2/4
s2*2
\bar "|."

```

```

}

```

```

SoloNotes = \relative g' {
  \clef "treble"

```

```

% verse

```

```

    g4 g g |
    b4 b b |

    % refrain
    R2*2 |
}

SoloLyrics = \lyricmode {
    One two three |
    four five six |
}

SopranoNotes = \relative c'' {
    \clef "treble"

    % verse
    R2.*2 |

    % refrain
    c4 c |
    g4 g |
}

SopranoLyrics = \lyricmode {
    la la |
    la la |
}

BassNotes = \relative c {
    \clef "bass"

    % verse
    R2.*2 |

    % refrain
    c4 e |
    d4 d |
}

BassLyrics = \lyricmode {
    dum dum |
    dum dum |
}

\score {
  <<
    \new Voice = "SoloVoice" << \global \SoloNotes >>
    \new Lyrics \lyricsto "SoloVoice" \SoloLyrics

    \new ChoirStaff <<
      \new Voice = "SopranoVoice" << \global \SopranoNotes >>
      \new Lyrics \lyricsto "SopranoVoice" \SopranoLyrics

```



```

\new Voice = "BassVoice" << \global \BassNotes >>
\new Lyrics \lyricsto "BassVoice" \BassLyrics
>>
>>

\layout {
  ragged-right = ##t
  \context { \Staff
    % these lines prevent empty staves from being printed
    \RemoveEmptyStaves
    \override VerticalAxisGroup.remove-first = ##t
  }
}

```



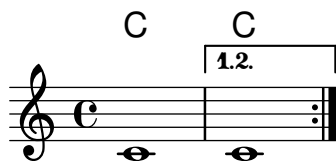
Wiederholungs-(Volta-)Klammern unterhalb der Akkordsymbole

Indem man den `Volta_engraver` zu dem entsprechenden Notensystem hinzufügt, können Wiederholungsklammern unterhalb der Akkorde gesetzt werden.

```

\score {
  <<
    \chords { c1 c1 }
    \new Staff \with { \consists "Volta_engraver" }
    {
      \repeat volta 2 { c'1 \alternative { c' } }
    }
  >>
  \layout {
    \context {
      \Score
      \remove "Volta_engraver"
    }
  }
}

```



Volta brackets in multiple staves

By adding the `Volta_engraver` to the relevant staff, volte can be put over staves other than the topmost one in a score.

`\repeat` and related commands should be present in all staves.

```
voltaMusic = \relative c'' {
  \repeat volta 2 {
    c1
    \alternative {
      \volta 1 { d1 }
      \volta 2 { e1 }
    }
  }
}

<<
  \new StaffGroup <<
    \new Staff \voltaMusic
    \new Staff \voltaMusic
  >>
  \new StaffGroup <<
    \new Staff \with { \consists "Volta_engraver" }
      \voltaMusic
    \new Staff \voltaMusic
  >>
>>
```



19 Tweaks and overrides

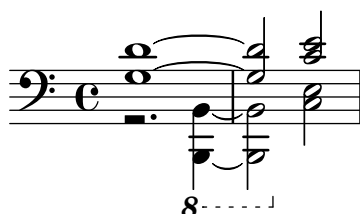
See also Abschnitt “Changing defaults” in *Notationsreferenz* and Abschnitt “Tweaking output” in *Handbuch zum Lernen*.

Adding an ottava marking to a single voice

If you have more than one voice on the staff, setting octavation in one voice transposes the position of notes in all voices for the duration of the ottava bracket. If the octavation is only intended to apply to one voice, the `Ottava_spanner_engraver` should be moved to Voice context.

```
\layout {
  \context {
    \Staff
    \remove Ottava_spanner_engraver
  }
  \context {
    \Voice
    \consists Ottava_spanner_engraver
  }
}

{
  \clef bass
  << { <g d'>1~ q2 <c' e'> }
  \\
  {
    r2.
    \ottava -1
    <b,,, b,,,>4 ~ |
    q2
    \ottava 0
    <c e>2
  }
  >>
}
```



Adding links to objects

To add a link to a grob stencil you can use `add-link` as defined here. It works both with `\override` and `\tweak`.

Drawback: point-and-click is disturbed for the linked grobs.

Limitation: Works for PDF only.

The linked objects are colored with a separate command.

```
#(define (add-link url-strg)
  (lambda (grob)
```

```

(let* ((stil (ly:grob-property grob 'stencil)))
  (if (ly:stencil? stil)
      (let* ((x-ext (ly:stencil-extent stil X))
              (y-ext (ly:stencil-extent stil Y))
              (url-expr `(url-link ,url-strg ,x-ext ,y-ext))
              (new-stil
               (ly:stencil-add
                (ly:make-stencil url-expr x-ext y-ext)
                stil)))
        (ly:grob-set-property! grob 'stencil new-stil))))))

%%% test

%% For easier maintenance of this snippet the URL is formatted to use the
%% actually used LilyPond version.
%% Of course a literal URL would work as well.

#(define major.minor-version
  (string-join (take (string-split (lilypond-version) #\.) 2) "."))

urlI =
#(format #f
  "http://lilypond.org/doc/v~a/Documentation/notation/writing-pitches"
  major.minor-version)

urlIII =
#(format #f
  "http://lilypond.org/doc/v~a/Documentation/notation/rhythms"
  major.minor-version)

urlIIII =
#(format #f
  "http://lilypond.org/doc/v~a/Documentation/notation/note-heads"
  major.minor-version)

urlIV =
#(format #f
  "http://lilypond.org/doc/v~a/Documentation/notation/beams"
  major.minor-version)

urlV =
#(format #f
  "http://lilypond.org/doc/v~a/Documentation/notation/note-head-styles"
  major.minor-version)

urlVI =
#(format #f
  "http://lilypond.org/doc/v~a/Documentation/notation/writing-pitches"
  major.minor-version)

\relative c' {
  \key cis \minor

```

```

\once \override Staff.Clef.color = #green
\once \override Staff.Clef.after-line-breaking =
  #(add-link urlI)

\once \override Staff.TimeSignature.color = #green
\once \override Staff.TimeSignature.after-line-breaking =
  #(add-link urlIII)

\once \override NoteHead.color = #green
\once \override NoteHead.after-line-breaking =
  #(add-link urlIIII)

cis'1
\once \override Beam.color = #green
\once \override Beam.after-line-breaking =
  #(add-link urlIV)
cis8 dis e fis gis2
<gis,
  \tweak Accidental.color #green
  \tweak Accidental.after-line-breaking #(add-link urlVI)
  \tweak color #green
  \tweak after-line-breaking #(add-link urlV)
  \tweak style #'harmonic
  bis
  dis
  fis
>1
<cis, cis' e>
}

```



Adding markups in a tablature

By default, markups are not displayed in a tablature.

To make them appear, revert the stencil property of the TextScript grob in the TabStaff context.

```

high = { r4 r8 <g c'> q r8 r4 }
low = { c4 r4 c8 r8 g,8 b, }
pulse = { s8^"1" s^"&" s^"2" s^"&" s^"3" s^"&" s^"4" s^"&" }

\score {
  \new TabStaff {
    \repeat unfold 2 << \high \ \ \low \ \ \pulse >>
  }
  \layout {
    \context {
      \TabStaff
      \clef moderntab
    }
  }
}

```

```

\revert TextScript.stencil
\override TextScript.font-series = #'bold
\override TextScript.font-size = #-2
\override TextScript.color = #red
}
\context {
  \Score
  proportionalNotationDuration = #1/8
}
}
}

```

	1	&	2	&	3	&	4	&	1	&	2	&	3	&	4	&
T							1-1						1-1			
A							0-0						0-0			
B	3				3				3				3		2	
						3								3		2

Adding timing marks to long glissandi

Skipped beats in very long glissandi are sometimes indicated by timing marks consisting of stems without noteheads. Such stems can also be used to carry intermediate expression markings.

If the stems do not align well with the glissando, they may need to be repositioned slightly.

```

glissandoSkipOn = {
  \override NoteColumn.glissando-skip = ##t
  \hide NoteHead
  \override NoteHead.no-ledgers = ##t
}

glissandoSkipOff = {
  \revert NoteColumn.glissando-skip
  \undo \hide NoteHead
  \revert NoteHead.no-ledgers
}

\relative c' ' {
  r8 f8\glissando \glissandoSkipOn f4 g a |
  a8\noBeam \glissandoSkipOff a8
  r8 f8\glissando \glissandoSkipOn g4 a8 \glissandoSkipOff a8 |
  r4 f\glissando\< \glissandoSkipOn a4\> \glissandoSkipOff b8\! r |
}

```



Adjusting grace note spacing

The space given to grace notes can be adjusted using the `spacing-increment` property of `Score.GraceSpacing`.

```

graceNotes = {
  \grace { c4 c8 c16 c32 }
  c8
}

```

```

}

\relative c' {
  c8
  \graceNotes
  \override Score.GraceSpacing.spacing-increment = #2.0
  \graceNotes
  \revert Score.GraceSpacing.spacing-increment
  \graceNotes
}

```



Adjusting slur positions vertically

Using `\override Slur.positions` it is possible to set the vertical position of the start and end points of a slur to absolute values (or rather, forcing LilyPond's slur algorithm to consider these values as desired). In many cases, this means a lot of trial and error until good values are found. You probably have tried the `\offset` command next just to find out that it doesn't work for slurs, emitting a warning instead.

The code in this snippet allows you to tweak the vertical start and end positions by specifying *relative* changes, similar to `\offset`.

Syntax: `\offsetPositions #'(dy1 . dy2)`

```

offsetPositions =
#(define-music-function (offsets) (number-pair?)
  #{
    \once \override Slur.control-points =
      #(lambda (grob)
          (match-let (((_ . y1) _ _ (_ . y2))
                      (ly:slur::calc-control-points grob))
            ((off1 . off2) offsets))
          (set! (ly:grob-property grob 'positions)
                (cons (+ y1 off1) (+ y2 off2)))
            (ly:slur::calc-control-points grob)))
    #})

\relative c' {
  c4(^"default" c, d2)
  \offsetPositions #'(0 . 1)
  c'4(^"(0 . 1)" c, d2)
  \offsetPositions #'(0 . 2)
  c'4(^"(0 . 2)" c, d2)
  \bar "||"
  g4(^"default" a d'2)
  \offsetPositions #'(1 . 0)
  g,,4(^"(1 . 0)" a d'2)
  \offsetPositions #'(2 . 0)
  g,,4(^"(2 . 0)" a d'2)
}

```



Adjusting vertical spacing of lyrics

This snippet shows how to bring the lyrics line closer to the staff.

```
music = \relative c' { c4 d e f | g4 f e d | c1 }
text = \lyricmode { aa aa aa aa aa aa aa aa }
```

```
<<
\new Staff \new Voice = melody \music
% Default layout:
\new Lyrics \lyricsto melody \text

\new Staff \new Voice = melody \music
% Reducing the minimum space below the staff and above the lyrics.
\new Lyrics \with {
  \override VerticalAxisGroup.nonstaff-relatedstaff-spacing =
    #'((basic-distance . 1))
} \lyricsto melody \text
>>
```



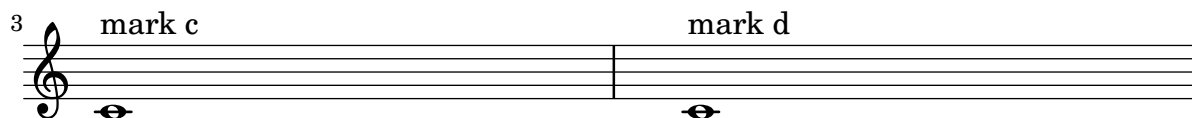
Aligning text marks to notes

By default, TextMark objects are aligned to so-called NonMusicalPaperColumn grobs, like the left edge of the staff or a bar line. They can be aligned to a note instead by setting the non-musical property to #f.

```
\layout {
  line-length = 80\mm
}

{
  \textMark "mark a" c'1 |
  \textMark "mark b" c'1 |
  \break
  \override Score.TextMark.non-musical = ##f
  \textMark "mark c" c'1 |
  \textMark "mark d" c'1 |
}
```

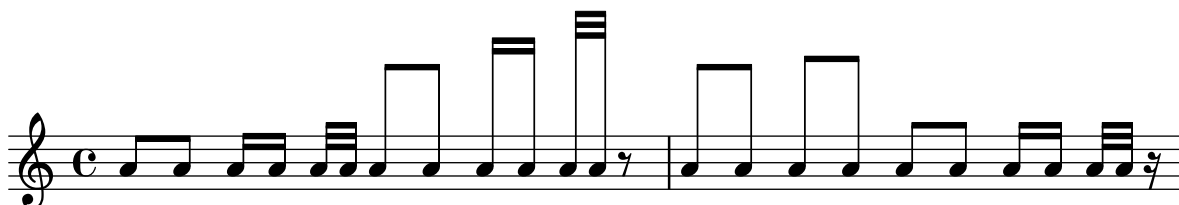




Altering the length of beamed stems

Stem lengths on beamed notes can be varied by overriding the `beamed-lengths` property of the `Stem` details. If a single value is used as an argument, the length applies to all stems. When multiple arguments are used, the first applies to eighth notes, the second to sixteenth notes and so on. The final argument also applies to all notes shorter than the note length of the final argument. Non-integer arguments may also be used.

```
\relative c' {
  \override Stem.details.beamed-lengths = #'(2)
  a8[ a] a16[ a] a32[ a]
  \override Stem.details.beamed-lengths = #'(8 10 12)
  a8[ a] a16[ a] a32[ a] r8 |
  \override Stem.details.beamed-lengths = #'(8)
  a8[ a]
  \override Stem.details.beamed-lengths = #'(8.5)
  a8[ a]
  \revert Stem.details.beamed-lengths
  a8[ a] a16[ a] a32[ a] r16 |
}
```



Alternative Taktnummerierung

Zwei alternative Methoden können eingestellt werden, die die Taktnummerierung beeinflussen, insbesondere bei Wiederholungen.

```
music = \relative c' {
  \repeat volta 3 {
    c4 d e f |
    \alternative {
      \volta 1 { c4 d e f | c2 d \break }
      \volta 2 { f4 g a b | f4 g a b | f2 a | \break }
      \volta 3 { c4 d e f | c2 d } } }
  c1 \bar "|"
}

{
  \textMark \markup \large "default"
  \music
}

{
  \textMark \markup \large \typewriter "numbers"
  \set Score.alternativeNumberingStyle = #'numbers
}
```

```

\music
}

{
  \textMark \markup \large \typewriter "numbers-with-letters"
  \set Score.alternativeNumberingStyle = #'numbers-with-letters
  \music
}

\layout {
  \context {
    \Score
    \override TextMark.Y-offset = #5
  }
}

```

default



numbers



numbers-with-letters



Analysis brackets above the staff

Simple horizontal analysis brackets are added below the staff by default. The following example shows a way to place them above the staff instead.

```
\layout {
  \context {
    \Voice
    \consists "Horizontal_bracket_engraver"
  }
}

\relative c' {
  \once \override HorizontalBracket.direction = #UP
  c2\startGroup
  d2\stopGroup
}
```



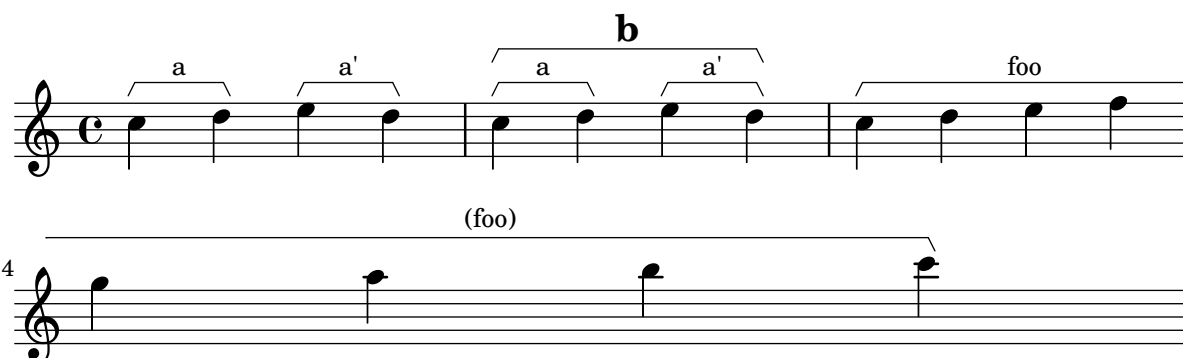
Analysis brackets with labels

Text markup may be added to analysis brackets using the text property of the HorizontalBracketText grob. Adding different texts to brackets beginning at the same time requires the \tweak command.

Bracket text gets parenthesized after a line break. The vertical order of nested brackets can be controlled with the outside-staff-priority property.

```
\layout {
  \context {
    \Voice
    \consists "Horizontal_bracket_engraver"
    \override HorizontalBracket.direction = #UP
  }
}

{
  \once\override HorizontalBracketText.text = "a"
  c''\startGroup d''\stopGroup
  \once\override HorizontalBracketText.text = "a'"
  e''\startGroup d''\stopGroup |
  c''-\tweak outside-staff-priority #801
    \tweak HorizontalBracketText.text
      \markup \bold \huge "b" \startGroup
    -\tweak HorizontalBracketText.text "a" \startGroup
  d''\stopGroup
  e''-\tweak HorizontalBracketText.text "a'" \startGroup
  d''\stopGroup\stopGroup |
  c''-\tweak HorizontalBracketText.text foo \startGroup
  d'' e'' f'' | \break
  g'' a'' b'' c'''\stopGroup
}
```



Asymmetric slurs

Slurs can be made asymmetric to match an asymmetric pattern of notes better.

```
slurNotes = { d,8( a' d f a f' d, a) }
```

```
\relative c' {
  \stemDown
  \slurUp
  \slurNotes
  \once \override Slur.eccentricity = #3.0
  \slurNotes
}
```



Breaking horizontal alignment of dynamics and textscripts

LilyPond uses `DynamicLineSpanner` grobs to horizontally align successive dynamic objects like hairpins and dynamic text, even if they are positioned on different sides of a staff. This connection cannot be broken, contrary to the vertical alignment (see snippet „Breaking vertical alignment of dynamics and textscripts“).

There are two solutions to circumvent the problem.

- Modify the `shorten-pair` property of the `Hairpin` grob to compensate the offset by which the hairpin was moved.
- Put the two dynamic objects into different voices.

Both solutions are demonstrated in this snippet.

```
{
  <>^"default"
  f' _\pp ^\> f' f' f'\!
}

{
  <>^\markup { setting \typewriter shorten-pair }
  f' _\pp \tweak shorten-pair #'(-3 . 0) ^\> f' f' f'\!
}
```

```

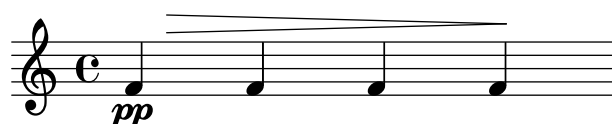
{
  <>^\markup { using another \typewriter Voice context }
  << { f' ^\> f' f' f' \! }
    \new Voice { s4_\pp } >>
}

\layout {
  line-width = 8\cm
  ragged-right = ##f

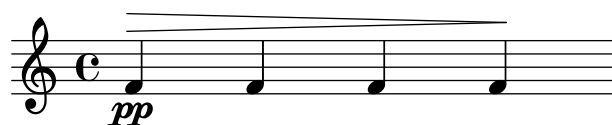
  \context {
    \Voice
    \override TextScript.staff-padding = #3.5
  }
}

```

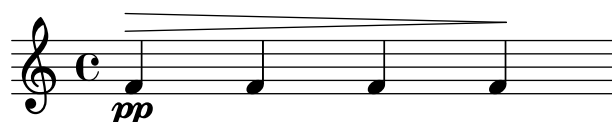
default



setting shorten-pair



using another Voice context



Breaking vertical alignment of dynamics and textscripts

By default, LilyPond uses `DynamicLineSpanner` grobs to vertically align successive dynamic objects like hairpins and dynamic text. However, this is not always wanted. By inserting `\breakDynamicSpan`, which ends the alignment spanner prematurely, this vertical alignment can be avoided.


See also snippet „Breaking horizontal alignment of dynamics and textscripts“.

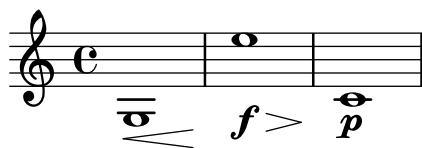
```

{ g1\< |
  e''\f\> |
  c'\p }

{ g1\< |
  e''\breakDynamicSpan\f\> |
  c'\p }

```





Caesura („railtracks“) with fermata

A caesura is sometimes denoted by a double „railtracks“ breath mark with a fermata sign positioned above. This snippet shows an optically pleasing combination of railtracks and fermata.

```
\relative c' {
  c2.
  % construct the symbol
  \override BreathingSign.text = \markup {
    \override #'(direction . 1)
    \override #'(baseline-skip . 1.8)
    \dir-column {
      \translate #'(0.155 . 0)
      \center-align \musicglyph "scripts.caesura.curved"
      \center-align \musicglyph "scripts.ufermata"
    }
  }
  \breathe c4
  % set the breath mark back to normal
  \revert BreathingSign.text
  c2. \breathe c4
  \bar "|."
}
```

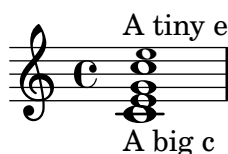


Changing a single note's size in a chord

Individual note heads in a chord can be modified with the `\tweak` command inside a chord, by altering the `font-size` property.

Inside the chord (within the brackets `<>`), before the note to be altered, place the `\tweak` command, followed by `font-size` and define the proper size like `#-2` (a tiny note head).

```
\relative c' {
  <\tweak font-size #+2 c e g c
  \tweak font-size #-2 e>1
  ~\markup { A tiny e }_~\markup { A big c }
}
```



Changing beam thickness and spacing

To make beams thicker or thinner, alter the `beam-thickness` property of the `Beam` grob. To adjust the spacing between beams, alter `length-fraction`.

```

\relative f' {
  \time 1/8
  \override Beam.beam-thickness = #0.4
  \override Beam.length-fraction = #0.8
  c32 c c c
  \revert Beam.beam-thickness % 0.48 is default thickness
  \revert Beam.length-fraction % 1.0 is default spacing
  c32 c c c
  \override Beam.beam-thickness = #0.6
  \override Beam.length-fraction = #1.3
  c32 c c c
}

```



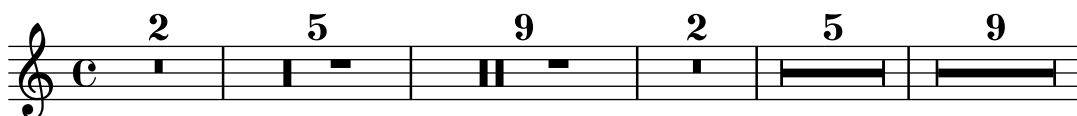
Die Erscheinung von Pausentakten ändern

Wenn zehn oder weniger Pausentakte vorkommen, wird eine Reihe von Longa- und Brevispausen (auch Kirchenpausen genannt) gesetzt, bei mehr Takten wird eine Line mit der Taktanzahl ausgegeben. Der vorgegebene Wert von zehn kann geändert werden, indem man die `expand-limit`-Eigenschaft setzt:

```

\relative c' {
  \compressMMRests {
    R1*2 | R1*5 | R1*9
    \override MultiMeasureRest.expand-limit = 3
    R1*2 | R1*5 | R1*9
  }
}

```



Changing properties for individual grobs

The `\applyOutput` command allows the tuning of any layout object, in any context. It requires a Scheme function with three arguments.

In the example below, function `mc-squared` is executed for all `NoteHead` grobs (within the current `Voice` context) at the current time step; the function modifies the grob's `stencil`, using the `staff-position` property to replace some pitches with markup.

See the 'Extending' manual (<https://lilypond.org/doc/v2.24/Documentation/extending/running-a-function-on-all-layout-objects>) for more information.

```

#(define (mc-squared grob grob-origin context)
  (let ((sp (ly:grob-property grob 'staff-position)))
    (ly:grob-set-property!
      grob 'stencil
      (grob-interpret-markup grob
        #{ \markup \lower #0.5
          #(case sp
            ((-5) "m")

```

```

((-3) "c ")
((-2) #{ \markup \teeny \bold 2 #})
(else "bla")) #}})))

\relative c' {
  <d f g b>2
  \applyOutput Voice.NoteHead #mc-squared
  <d f g b>2
}

```



Text und Strecker-Stile für Dynamik-Texte ändern

Der Text, der für Crescendo und Decrescendo gesetzt wird, kann geändert werden, indem man die Eigenschaften `crescendoText` und `decrescendoText` verändert. Der Stil des Streckers kann auch geändert werden, indem die `'style`-Eigenschaft des `DynamicTextSpanner` beeinflusst wird. Der Standardwert ist `'hairpin`, andere Möglichkeiten sind `'line`, `'dashed-line` und `'dotted-line`.

```

\relative c' {
  \set crescendoText = \markup { \italic { cresc. poco } }
  \set crescendoSpanner = #'text
  \override DynamicTextSpanner.style = #'dotted-line
  a2< a
  a2 a
  a2 a
  a2 a\mf
}

```



Changing the default text font family

The default font families for text can be overridden.

```

%{
You may have to install additional fonts.

Red Hat Fedora: dejavu-fonts-all

Debian GNU/Linux, Ubuntu: fonts-dejavu-core
                           fonts-dejavu-extra
}%

```

```

\paper {
  %{
    run
    lilypond -dshow-available-fonts
    to show all fonts available in the process log.
  }
}

```



```

%}
property-defaults.fonts.serif = "DejaVu Serif"
property-defaults.fonts.sans = "DejaVu Sans"
property-defaults.fonts.typewriter = "DejaVu Sans Mono"
}

{
g'''4^\markup {
  DejaVu Serif: \bold bold
                \italic italic
                \italic \bold { bold italic }
}
g4_\markup {
  \override #'(font-family . sans) {
    DejaVu Sans: \bold bold
                  \italic italic
                  \italic \bold { bold italic }
  }
}
g''2^\markup {
  \override #'(font-family . typewriter) {
    DejaVu Sans Mono: \bold bold
                      \italic italic
                      \italic \bold { bold italic }
  }
}
}
}

```



Changing the staff size

The simplest way to resize staves is to use

```

#(set-global-staff-size size)

```

To resize an individual staff's size, you can use the properties `staff-space` and `fontSize`.

```

<<
\new Staff \relative c'' {
  \dynamicDown c8\ff c c c c c c c
}
\new Staff \with {
  fontSize = #-3
  \override StaffSymbol.staff-space = #(magstep -3)
} \relative c {
  \clef bass c8 c c c c\ff c c c
}
>>

```



Das Tempo ohne Metronom-Angabe verändern

Um das Tempo für die MIDI-Ausgabe zu ändern, ohne eine Tempoangabe in den Noten auszugeben, kann die Metronombezeichnung unsichtbar gemacht werden:

```
\score {
  \new Staff \relative c' {
    \tempo 4 = 160
    c4 e g b
    c4 b d c
    \set Score.tempoHideNote = ##t
    \tempo 4 = 96
    d,4 fis a cis
    d4 cis e d
  }
  \layout { }
  \midi { }
}
```



Changing the text for sustain markings

With the `pedalSustainStrings` context property it is possible to set the text used for pedal down and up. Note that the only valid strings are those found in the list of pedal glyphs – the values shown in this snippet constitute an exhaustive list.

```
sustainNotes = { c4\sustainOn d e\sustainOff\sustainOn f\sustainOff }
```

```
\relative c' {
  \sustainNotes
  \set Staff.pedalSustainStrings = #'("P" "P-" "-")
  \sustainNotes
  \set Staff.pedalSustainStrings = #'("d" "de" "e")
  \sustainNotes
  \set Staff.pedalSustainStrings = #'("M" "M-" "-")
  \sustainNotes
  \set Staff.pedalSustainStrings = #'("Ped" "*Ped" "*")
  \sustainNotes
}

\layout {
  ragged-right = ##f
}
```



Controlling spanner visibility after a line break

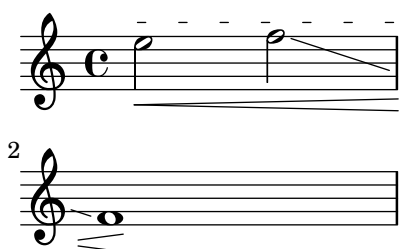
The visibility of spanners which end on the first note following a line break is controlled by the after-line-breaking callback `ly:spanner::kill-zero-spanned-time`.

For objects such as glissandos and hairpins, the default behaviour is to hide the spanner after a break; disabling the callback will allow the left-broken span to be shown.

Conversely, spanners which are usually visible, such as text spans, can be hidden by enabling the callback.

```
\paper {
  line-width = 50\mm
}

\relative c' {
  \override Hairpin.to-barline = ##f
  \override Glissando.breakable = ##t
  % show hairpin
  \override Hairpin.after-line-breaking = ##t
  % hide text span
  \override TextSpanner.after-line-breaking =
    #ly:spanner::kill-zero-spanned-time
  e2\<\startTextSpan
  % show glissando
  \override Glissando.after-line-breaking = ##t
  f2\glissando
  \break
  f,1\!\stopTextSpan
}
```



Controlling the appearance of tremolo slashes

Using various properties of the `StemTremolo` grob it is possible to control the appearance of tremolo slashes.

- Property `slope` sets the slope for tremolo slashes.
- Property `shape` determines whether tremolo slashes look like rectangles (value `rectangle`) or like very small beams (value `beam-like`).
- Property `style` sets both the slope and the shape depending on whether the note has flags, beams, or only a plain stem. This is in contrast to the previous two properties, which change the slope and shape unconditionally. There are two styles defined.

- default: slashes for down-stem flags are longer and more sloped than slashes for up-stem flags; slashes on beamed notes have a rectangular shape and are parallel to the beam.
- constant: all slashes are beam-like and have the same slope except for down-stem flags.

```

music = {
  a''4:32 a':
  e''8: \noBeam e':
  a'':[ a':]
  f':[ g':]
  d':[ d':]
}

\new Staff {
  <>^\markup "default"
  \music
}

\new Staff {
  <>^\markup \typewriter "style = #'constant"
  \override StemTremolo.style = #'constant
  \music
}

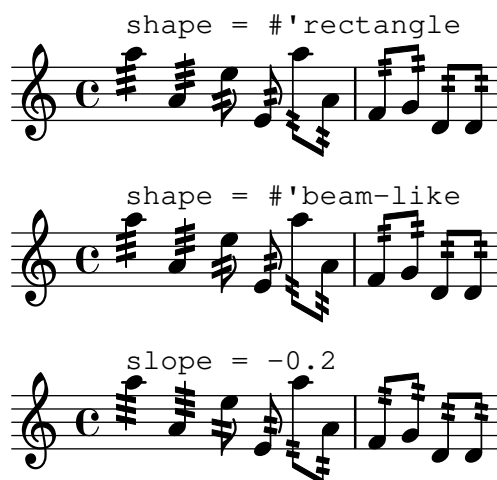
\new Staff {
  <>^\markup \typewriter "shape = #'rectangle"
  \override StemTremolo.shape = #'rectangle
  \music
}

\new Staff {
  <>^\markup \typewriter "shape = #'beam-like"
  \override StemTremolo.shape = #'beam-like
  \music
}

\new Staff {
  <>^\markup \typewriter "slope = -0.2"
  \override StemTremolo.slope = -0.2
  \music
}

```





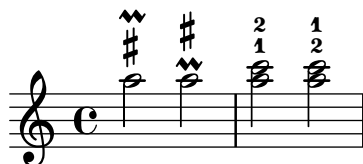
Die vertikale Anordnung von Beschriftungen kontrollieren

Die vertikale Anordnung von Beschriftungen wird mit der 'script-priority-Eigenschaft kontrolliert. Um so kleiner die Zahl, umso näher wird die Beschriftung in Bezug auf die Note gesetzt. In diesem Beispiel hat das TextScript-Objekt (das Kreuz) zuerst die niedrigste Priorität, wird also auch am niedrigsten in dem ersten Beispiel gesetzt. Im zweiten Fall hat der Praller (das Script) die niedrigste Priorität, darum wird er am nächsten zum System gesetzt. Wenn zwei Objekte die gleiche Priorität haben, wird ihre Reihenfolge anhand ihres Auftretens in der Quelldatei entschieden.

```
\relative c'' {
  \once \override TextScript.script-priority = -100
  a2^\prall^\markup { \sharp }

  \once \override Script.script-priority = -100
  a2^\prall^\markup { \sharp }

  \set fingeringOrientations = #'(up)
  <c-2 a-1>2
  <a-1 c>\tweak script-priority -100 -2>2
}
```



Controlling tuplet bracket visibility

The default behavior of tuplet-bracket visibility is to print a bracket unless there is a beam of the same length as the tuplet.

To control the visibility of tuplet brackets, set the property `bracket-visibility` to either `#t` (always print a bracket), `if-no-beam` (only print a bracket if there is no beam) or `#f` (never print a bracket). The latter is in fact equivalent to omitting the `TupletBracket` object altogether from the printed output.

```
music = \relative c'' {
  \tuplet 3/2 { c16[ d e ] f8 }
  \tuplet 3/2 { c8 d e }
```

```

\tuplet 3/2 { c4 d e }
}

\new Voice {
  \relative c' {
    \override Score.TextMark.non-musical = ##f
    \textMark "default" \music
    \override TupletBracket.bracket-visibility = #'if-no-beam
    \textMark \markup \typewriter "'if-no-beam" \music
    \override TupletBracket.bracket-visibility = ##t
    \textMark \markup \typewriter "#t" \music
    \override TupletBracket.bracket-visibility = ##f
    \textMark \markup \typewriter "#f" \music
    \omit TupletBracket
    \textMark \markup \typewriter "omit" \music
  }
}

```



Einen Doppelschlag mit Vorhalt erstellen

Einen Doppelschlag mit Vorhalt zu erstellen, wobei die untere Note das Vorzeichen benutzt, erfordert einige Einstellungsänderungen. Die `outside-staff-priority`-Eigenschaft muss auf falsch (`#f`) gesetzt werden, weil sie sonst über die Eigenschaft `avoid-slur property` dominieren würde. Der Wert von `halign` wird benutzt, um den Doppelschlag horizontal zu positionieren.

```

\relative c' {
  \after 2*2/3 \turn c2( d4) r |
  \after 4 \turn c4.( d8)
  \after 4
  {
    \once \set suggestAccidentals = ##t
    \once \override AccidentalSuggestion.outside-staff-priority = ##f
    \once \override AccidentalSuggestion.avoid-slur = #'inside
    \once \override AccidentalSuggestion.font-size = -3
    \once \override AccidentalSuggestion.script-priority = -1
    \once \hideNotes
    cis8\turn \noBeam
  }
  d4.( e8)
}

```



Creating custom key signatures

LilyPond supports custom key signatures. In this example, print for D minor and D major with an extended range of shown flats.

```
\new Staff \with {
  \override StaffSymbol.line-count = #8
  \override KeySignature.flat-positions = #'((-7 . 6))
  \override KeyCancellation.flat-positions = #'((-7 . 6))
  \override KeySignature.sharp-positions = #'((-6 . 7))
  \override KeyCancellation.sharp-positions = #'((-6 . 7))

  \override Clef.stencil =
    #(lambda (grob)
      (grob-interpret-markup grob
        #{ \markup\combine
          \musicglyph "clefs.C"
          \translate #'(-3 . -2)
          \musicglyph "clefs.F"
        #}))
      clefPosition = #3
      middleCPosition = #3
      middleCClefPosition = #3
    }

{
  \key d\minor f bes, f bes, |
  \key d\major fis b, fis b, |
}
```



Creating text spanners

The `\startTextSpan` and `\stopTextSpan` commands allow the creation of text spanners as easily as pedal indications or octavations. Override some properties of the `TextSpanner` object to modify its output.

```
\paper { ragged-right = ##f }

\relative c' {
  \override TextSpanner.bound-details.left.text = #"bla"
  \override TextSpanner.bound-details.right.text = #"blu"
  a4 \startTextSpan
  b4 c
  a4 \stopTextSpan

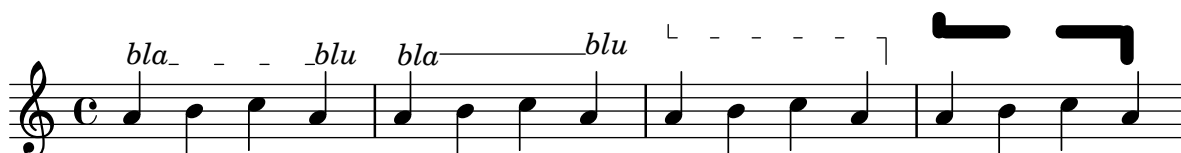
  \override TextSpanner.style = #'line
  \once \override TextSpanner.bound-details.left.stencil-align-dir-y = #CENTER
  a4 \startTextSpan
  b4 c
  a4 \stopTextSpan
}
```

```

\override TextSpanner.style = #'dashed-line
\override TextSpanner.bound-details.left.text =
  \markup { \draw-line #'(0 . 1) }
\override TextSpanner.bound-details.right.text =
  \markup { \draw-line #'(0 . -2) }
\once \override TextSpanner.bound-details.right.padding = #-2
a4 \startTextSpan
b4 c
a4 \stopTextSpan

\override TextSpanner.dash-period = #10
\override TextSpanner.dash-fraction = #0.5
\override TextSpanner.thickness = #10
a4 \startTextSpan
b4 c
a4 \stopTextSpan
}

```



Cross-staff chords – beaming problems workaround

Sometimes it is better to use stems from the ‘other’ staff for creating cross-staff chords to trick LilyPond’s beam collision detector. In the following snippet, if the stems from the lower staff were used instead, it would be necessary to explicitly use

```
\override Staff.Beam.collision-voice-only = ##t
```

so that LilyPond doesn’t move the beams.

```

\new PianoStaff <<
\new Staff = up \relative c' <<
{ r4
  \override Stem.cross-staff = ##t
  \override Stem.length = #19 % this is in half-spaces,
    % so it makes stems 9.5 staffspaces long
  \override Stem.Y-offset = #-6 % stems are normally lengthened
    % upwards, so here we must lower the stem by the amount
    % equal to the lengthening - in this case (19 - 7) / 2
    % (7 is default stem length)
  e e e }
{ s4
  \change Staff = "bottom"
  \override NoteColumn.ignore-collision = ##t
  c, c c
}
>>

```

```

\new Staff = bottom \relative c' {
  \clef bass
  \voiceOne

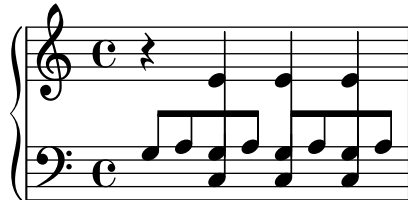
```



```

      g8 a g a g a g a
    }
  >>

```



Hälsе über beide Systeme

Dieses Schnipsel zeigt den Einsatz von `Span_stem_engraver` und `\crossStaff`, um Hälsе über Systeme hinweg automatisch zu verbinden. Die Länge des Halses muss definiert werden, da der variable Abstand zwischen Notenköpfen und Notensystemen automatisch errechnet wird.

```

\layout {
  \context {
    \PianoStaff
    \consists "Span_stem_engraver"
  }
}

```

```

\new PianoStaff <<
  \new Staff {
    <b d'>4 r d'16\> e'8. g8 r\! |
    e'8 f' g'4
    \voiceTwo
    % Down to lower staff
    \crossStaff { e'8 e'8 } e'4 |
  }

```

```

  \new Staff {
    \clef bass
    \voiceOne
    % Up to upper staff
    \crossStaff { <e g>4 e, g16 a8. c8 } d |
    g8 f g4 \voiceTwo g8 g g4 |
  }
  >>

```



Custodes

Custodes may be engraved in various styles.

```

\layout {

```

```

    ragged-right = ##t
}

\markup \with-true-dimensions % work around a cropping issue
\score {
  \new Staff \with { \consists "Custos_engraver" } \relative c' {
    \override Staff.Custos.neutral-position = #4

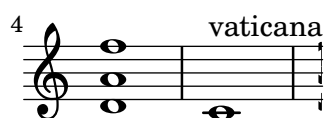
    \override Staff.Custos.style = #'hufnagel
    c1^"hufnagel" \break
    <d a' f'>1

    \override Staff.Custos.style = #'medicaea
    c1^"medicaea" \break
    <d a' f'>1

    \override Staff.Custos.style = #'vaticana
    c1^"vaticana" \break
    <d a' f'>1

    \override Staff.Custos.style = #'mensural
    c1^"mensural" \break
    <d a' f'>1
  }
}

```



Bunddiagramme anpassen

Eigenschaften von Bunddiagrammen können in 'fret-diagram-details verändert werden. Einstellungen mit dem `\override`-Befehl werden dem `FretBoards.FretBoard`-Objekt zugewiesen. Genauso wie `Voice` ist auch `FretBoards` ein Kontext der niedrigsten Ebene, weshalb der Kontext auch in dem Befehl weggelassen werden kann.

```

\include "predefined-guitar-fretboards.ly"

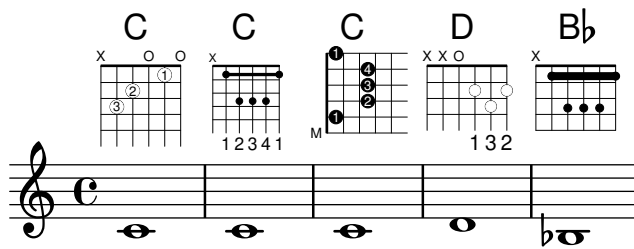
\storePredefinedDiagram #default-fret-table \chordmode { c' }
    #guitar-tuning
    "x;1-1-(;3-2;3-3;3-4;1-1-);"

% shorthand
oo = #(define-music-function
    (grob-path value)
    (list? scheme?)
    #{ \once \override $grob-path = #value #})

<<
\new ChordNames {
    \chordmode { c1 | c | c | d | bes }
}
\new FretBoards {
    % Set global properties of fret diagram
    \override FretBoards.FretBoard.size = 1.2
    \override FretBoard.fret-diagram-details.finger-code = #'in-dot
    \override FretBoard.fret-diagram-details.dot-color = #'white
    \chordmode {
        c
        \oo FretBoard.size #1.0
        \oo FretBoard.fret-diagram-details.barre-type #'straight
        \oo FretBoard.fret-diagram-details.dot-color #'black
        \oo FretBoard.fret-diagram-details.finger-code #'below-string
        c'
        \oo FretBoard.fret-diagram-details.barre-type #'none
        \oo FretBoard.fret-diagram-details.number-type #'arabic
        \oo FretBoard.fret-diagram-details.orientation #'landscape
        \oo FretBoard.fret-diagram-details.mute-string "M"
        \oo FretBoard.fret-diagram-details.label-dir #LEFT
        \oo FretBoard.fret-diagram-details.dot-color #'black
        c'
        \oo FretBoard.fret-diagram-details.finger-code #'below-string
        \oo FretBoard.fret-diagram-details.dot-radius #0.35
        \oo FretBoard.fret-diagram-details.dot-position #0.5
        \oo FretBoard.fret-diagram-details.fret-count #3
        d
        \oo FretBoard.fret-diagram-details.barre-type #'straight
        \oo FretBoard.fret-diagram-details.finger-code #'none
        \oo FretBoard.fret-diagram-details.dot-radius #0.25
        \oo FretBoard.fret-diagram-details.dot-color #'black
        \oo FretBoard.fret-diagram-details.string-overhang #0.
        \oo FretBoard.fret-diagram-details.barre-thickness #2.
        bes
    }
}
\new Voice {
    c'1 | c' | c' | d' | bes
}

```

>>



Anpassung von Beschriftungs-Bunddiagrammen

Bunddiagramme können mit der Eigenschaft 'fret-diagram-details' angepasst werden. Bunddiagramme, die als Textbeschriftung eingefügt werden, können Veränderungen im Voice.TextScript-Objekt oder direkt in der Beschriftung vorgenommen werden.

<<

```
\chords { c1 | c | c | d }

\new Voice = "mel" {
  \textLengthOn
  % Set global properties of fret diagram
  \override TextScript.size = 1.2
  \override TextScript.fret-diagram-details.finger-code = #'in-dot
  \override TextScript.fret-diagram-details.dot-color = #'white

  %% C major for guitar, no barre, using defaults
  % terse style
  c'1~\markup { \fret-diagram-terse "x;3-3;2-2;o;1-1;o;" }

  %% C major for guitar, barred on third fret
  % verbose style
  % size 1.0
  % roman fret label, finger labels below string, straight barre
  c'1~\markup {
    % standard size
    \override #'(size . 1.0) {
      \override #'(fret-diagram-details . (
        (number-type . roman-lower)
        (finger-code . in-dot)
        (barre-type . straight))) {
        \fret-diagram-verbose #'((mute 6)
          (place-fret 5 3 1)
          (place-fret 4 5 2)
          (place-fret 3 5 3)
          (place-fret 2 5 4)
          (place-fret 1 3 1)
          (barre 5 1 3))
      }
    }
  }

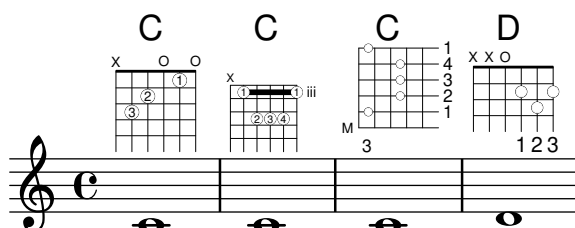
  %% C major for guitar, barred on third fret
```

```

% verbose style
% landscape orientation, arabic numbers, M for mute string
% no barre, fret label down or left, small mute label font
c'1^\markup {
  \override #'(fret-diagram-details . (
    (finger-code . below-string)
    (number-type . arabic)
    (label-dir . -1)
    (mute-string . "M")
    (orientation . landscape)
    (barre-type . none)
    (xo-font-magnification . 0.4)
    (xo-padding . 0.3))) {
    \fret-diagram-verbose #'(mute 6)
      (place-fret 5 3 1)
      (place-fret 4 5 2)
      (place-fret 3 5 3)
      (place-fret 2 5 4)
      (place-fret 1 3 1)
      (barre 5 1 3))
  }
}

%% simple D chord
% terse style
% larger dots, centered dots, fewer frets
% label below string
d'1^\markup {
  \override #'(fret-diagram-details . (
    (finger-code . below-string)
    (dot-radius . 0.35)
    (dot-position . 0.5)
    (fret-count . 3))) {
    \fret-diagram-terse "x;x;o;2-1;3-2;2-3;"
  }
}
}
>>

```



Klammer anzeigen, wenn nur ein System gesetzt wird

Wenn nur ein System einer Systemgruppe vom Typ `ChoirStaff` oder `StaffGroup` angezeigt wird, wird die Klammer zu Beginn normalerweise nicht gesetzt. Das kann verändert werden, indem man die entsprechende Eigenschaft verändert.

Bei Systemen wie `PianoStaff` und `GrandStaff`, die mit einer geschweiften Klammer beginnen, muss eine andere Eigenschaft verändert werden, wie das zweite Beispiel zeigt.

```
\score {
  \new StaffGroup <<
    % Must be lower than the actual number of staff lines
    \override StaffGroup.SystemStartBracket.collapse-height = 4
    \override Score.SystemStartBar.collapse-height = 4
    \new Staff {
      c'1
    }
  >>
}
\score {
  \new PianoStaff <<
    \override PianoStaff.SystemStartBrace.collapse-height = 4
    \override Score.SystemStartBar.collapse-height = 4
    \new Staff {
      c'1
    }
  >>
}
```



Displaying grob ancestry

When working with grob callbacks, it can be helpful to understand a grob's ancestry. Most grobs have parents which influence the positioning of the grob. X- and Y-parents influence the horizontal and vertical positions for the grob, respectively. Additionally, each parent may have parents of its own.

Unfortunately, there are several aspects of a grob's ancestry that can lead to confusion:

- The types of parents a grob has may depend on context.
- For some grobs, the X- and Y-parents are the same.
- A particular *ancestor* may be related to a grob in multiple ways.
- The concept of *generations* is misleading.

For example, the `System` grob can be both parent (on the Y-side) and grandparent (twice on the X-side) to a `VerticalAlignment` grob.

The macro defined in this snippet prints (to the console) a textual representation of a grob's ancestry. For example, the call

```
{
  \once \override NoteHead.before-line-breaking = #display-ancestry
  c
}
```

generates the following output.

```
-----
NoteHead
X,Y: NoteColumn
  X: PaperColumn
    X,Y: System
  Y: VerticalAxisGroup
    X: NonMusicalPaperColumn
      X,Y: System
    Y: VerticalAlignment
      X: NonMusicalPaperColumn
        X,Y: System
      Y: System
```

As a consequence, you have to execute the code in this snippet by yourself, since the generated output file doesn't show the data we are interested in.

```
#(define (get-ancestry grob)
  (if (not (null? (ly:grob-parent grob X)))
      (list (grob::name grob)
            (get-ancestry (ly:grob-parent grob X))
            (get-ancestry (ly:grob-parent grob Y)))
      (grob::name grob)))

#(define (format-ancestry lst padding)
  (string-append
    (symbol->string (car lst)) "\n"
    (let ((X-ancestry (if (list? (cadr lst))
                          (format-ancestry (cadr lst) (+ padding 3))
                          (symbol->string (cadr lst))))
          (Y-ancestry (if (list? (caddr lst))
                          (format-ancestry (caddr lst) (+ padding 3))
                          (symbol->string (caddr lst)))))
      (if (equal? X-ancestry Y-ancestry)
          (string-append (format #f "~&")
                          (make-string padding #\space)
                          "X,Y: "
                          (if (list? (cadr lst))
                              (format-ancestry (cadr lst) (+ padding 5))
                              (symbol->string (cadr lst))))
          (string-append (format #f "~&")
                          (make-string padding #\space)
                          "X: " X-ancestry "\n"
                          (make-string padding #\space)
                          "Y: " Y-ancestry (format #f "~&"))))
    (format #f "~&")))

#(define (display-ancestry grob)
  (format (current-output-port)
    "~2&~a~2%~a~&"
    (make-string 36 #\-)
    (if (ly:grob? grob)
        (format-ancestry (get-ancestry grob) 0)
```

```
(format #f "~a is not a grob" grob))))

\relative c' {
  \once \override NoteHead.before-line-breaking = #display-ancestry
  f4
  \once \override Accidental.before-line-breaking = #display-ancestry
  \once \override Arpeggio.before-line-breaking = #display-ancestry
  <f as c>4\arpeggio
}
```



Dotted harmonics

Artificial harmonics using `\harmonic` do not show dots. To override this behavior, set the context property `harmonicDots`.

```
\relative c' '' {
  \time 3/4
  \key f \major
  \set harmonicDots = ##t
  <bes f'\harmonic>2. ~
  <bes f'\harmonic>4. <a e'\harmonic>8( <gis dis'\harmonic> <g d'\harmonic>)
  <fis cis'\harmonic>2.
  <bes f'\harmonic>2.
}
```



Drawing boxes around grobs

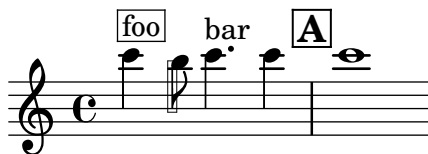
The `stencil` property can be overridden to draw a box around arbitrary grobs, either using `\override` or `\tweak`.

```
\relative c' '' {
  \once \override TextScript.stencil =
    #(make-stencil-boxer 0.1 0.3 ly:text-interface::print)
  c'4^"foo"

  \tweak Stem.stencil
    #(make-stencil-boxer 0.05 0.25 ly:stem::print)
  b8

  c4.^"bar" c4

  \override Score.RehearsalMark.stencil =
    #(make-stencil-boxer 0.15 0.3 ly:text-interface::print)
  \mark \default
  c1
}
```

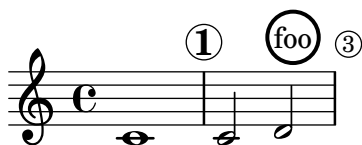



Drawing circles around various objects

The `\circle` command draws circles around `\markup` objects. For other objects, specific tweaks may be required, as demonstrated for rehearsal marks and measure numbers.

```
\relative c' {
  c1
  \set Score.rehearsalMarkFormatter =
    #(lambda (mark context)
      (make-circle-markup (format-mark-numbers mark context)))
  \mark \default

  c2 d^\markup {
    \override #'(thickness . 3) {
      \circle foo
    }
  }
  \override Score.BarNumber.break-visibility = #all-visible
  \override Score.BarNumber.stencil =
    #(make-stencil-circler 0.1 0.25 ly:text-interface::print)
}
```



Dynamics spanner with custom text

Postfix functions for custom crescendo text spanners. The spanners should start on the first note of the measure. One has to use `-\mycresc`, otherwise the spanner start will rather be assigned to the next note.

*% Two functions for (de)crescendo spanners where you can explicitly
% give the spanner text.*

```
mycresc =
#(define-music-function (mymarkup) (markup?)
  (make-music 'CrescendoEvent
    'span-direction START
    'span-type 'text
    'span-text mymarkup))

mydecresc =
#(define-music-function (mymarkup) (markup?)
  (make-music 'DecrescendoEvent
    'span-direction START
    'span-type 'text
    'span-text mymarkup))
```

```
\relative c' {
```

```

c4-\mycresc "custom cresc" c4 c4 c4 |
c4 c4 c4 c4 |
c4-\mydecrec "custom decrec" c4 c4 c4 |
c4 c4 c4 c4 |
c4 c4\! c4 c4
}

```



Extending a trill spanner

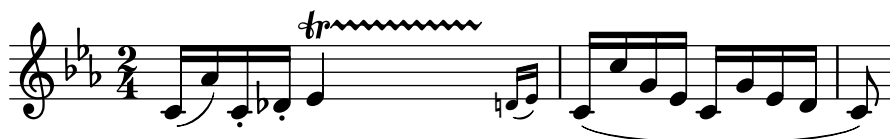
For TrillSpanner grobs, the minimum-length property becomes effective only if the set-spacing-rods procedure is called explicitly.

To do this, the springs-and-rods property should be set to ly:spanner::set-spacing-rods.

```

\relative c' {
  \key c\minor
  \time 2/4
  c16( as') c,-. des-.
  \once\override TrillSpanner.minimum-length = #15
  \once\override TrillSpanner.springs-and-rods = #ly:spanner::set-spacing-rods
  \afterGrace es4\startTrillSpan { d16[(\stopTrillSpan es)] }
  c( c' g es c g' es d
  c8)
}

```



Extending glissandi across repeats

A glissando that extends into several \alternative blocks can be simulated by adding a hidden grace note with a glissando at the start of each \alternative block. The grace note should be at the same pitch as the note which starts the initial glissando. This is implemented here with a music function that takes the pitch of the grace note as its argument.

Note that in polyphonic music the grace note must be matched with corresponding grace notes in all other voices.

```

repeatGliss = #(define-music-function (grace)
  (ly:pitch?)
  #{
    % the next two lines ensure the glissando is long enough
    % to be visible
    \once \override Glissando.springs-and-rods
      = #ly:spanner::set-spacing-rods
    \once \override Glissando.minimum-length = 3.5
    \once \hideNotes
    \grace $grace \glissando
  })

```

```

#})

\score {
  \relative c' {
    \repeat volta 3 { c4 d e f\glissando }
    \alternative {
      { g2 d }
      { \repeatGliss f g2 e }
      { \repeatGliss f e2 d }
    }
  }
}

music = \relative c' {
  \voiceOne
  \repeat volta 2 {
    g a b c\glissando
  }
  \alternative {
    { d1 }
    { \repeatGliss c \once \omit StringNumber e1\2 }
  }
}

\score {
  \new StaffGroup <<
    \new Staff <<
      \new Voice { \clef "G_8" \music }
    >>
    \new TabStaff <<
      \new TabVoice { \clef "moderntab" \music }
    >>
  >>
}

```

The image displays two musical staves. The top staff is a treble clef staff with a C-clef and a 3-measure repeat. The first measure contains the notes C4, D4, E4, and F4. The second measure contains the notes G4, A4, and B4. The third measure contains the notes C5, B4, A4, and G4. The first, second, and third endings are indicated by brackets above the staff. The bottom staff is a guitar tablature staff with a treble clef and a 2-measure repeat. The first measure contains the notes C4, D4, E4, and F4. The second measure contains the notes G4, A4, and B4. The first and second endings are indicated by brackets above the staff. The tablature staff shows fret numbers 0, 2, 3, and 5, and a 2-measure repeat with first and second endings.

Fine-tuning pedal brackets

The appearance of pedal brackets may be altered in different ways.

```
\paper {
```

```

    ragged-right = ##f
  }

  \relative c' {
    c2\sostenutoOn c
    c2\sostenutoOff c
    c2\tweak shorten-pair #'(-7 . -2) \sostenutoOn c
    c2\sostenutoOff c
    c2\tweak edge-height #'(0 . 3) \sostenutoOn c
    c2\sostenutoOff c
  }

```



Flat ties

This snippet provides a function `flared-tie` to draw a tie that consist of straight lines. It is intended as a replacement for the default tie-drawing function (i.e., a replacement argument for the `stencil` property of the `Tie` grob).

The argument of `flared-tie` is a list of coordinate pairs that specify additional points between the first and last point to span up the tie's lines. The first and last point are identical to the original tie's start and end point, respectively. The X and Y coordinate values are multiples of the bounding box length and height of the original tie (also taking care of the tie's direction); consequently, the first point has coordinates (0,0), and the last point (1,0).

The function `flare-tie` defines a shorthand for a flat tie. Further tweaking of the shape is possible by overriding `Tie.details.height-limit` or with `\shape`. It is also possible to change the custom definition on the fly.

```

#(define ((flared-tie coords) grob)
  (define (pair-to-list pair)
    (list (car pair) (cdr pair)))

  (define (normalize-coords goods x y dir)
    (map
      (lambda (coord)
        (cons (* x (car coord)) (* y dir (cdr coord))))
      goods))

  (define (my-c-p-s points thick)
    (make-connected-path-stencil points thick 1.0 1.0 #f #f))

  ;; Calling `ly:tie::print` and assigning its return value to a
  ;; variable in this outer `let` triggers LilyPond to position the
  ;; tie, allowing us to extract its extents. We only proceed,
  ;; however, if the tie doesn't get discarded (for whatever reason).
  (let ((sten (ly:tie::print grob)))
    (if (grob::is-live? grob)
      (let* ((layout (ly:grob-layout grob))
             (line-thickness (ly:output-def-lookup layout

```

```

                                'line-thickness))
  (thickness (ly:grob-property grob 'thickness 0.1))
  (used-thick (* line-thickness thickness))
  (dir (ly:grob-property grob 'direction))
  (xex (ly:stencil-extent sten X))
  (yex (ly:stencil-extent sten Y))
  (lenx (interval-length xex))
  (leny (interval-length yex))
  (xtrans (car xex))
  (ytrans (if (> dir 0)(car yex) (cdr yex)))
  ;; Add last point.
  (coord-list (append coords '((1.0 . 0.0))))
  (uplist
   (map pair-to-list
        (normalize-coords coord-list lenx (* leny 2) dir))))
  (ly:stencil-translate
   (my-c-p-s uplist used-thick)
   (cons xtrans ytrans)))
'()))))

% Define a default tie shape consisting of three straight lines.
#(define flare-tie
  (flared-tie '((0.1 . 0.3) (0.9 . 0.3))))

\relative c' {
  a4~ a
  \once \override Tie.stencil = #flare-tie
  a4~ a \break

  <a c e a c e a c e>~ q
  \once \override Tie.stencil = #flare-tie
  q~ q\break

  <>~\markup \small \typewriter "height-limit = 14"
  \override Tie.details.height-limit = 14
  a'4~ a
  \once \override Tie.stencil = #flare-tie
  a4~ a \break

  <>~\markup \small \typewriter "height-limit = 0.5"
  \override Tie.details.height-limit = 0.5
  a4~ a
  \once \override Tie.stencil = #flare-tie
  a4~ a \break

  \revert Tie.details.height-limit

  <>~\markup \small \typewriter
    "\shape #'((0 . 0) (0 . -1) (0 . -1) (0 . 0))"
  \shape #'((0 . 0) (0 . -1) (0 . -1) (0 . 0)) Tie
  a4~ a
  \once \override Tie.stencil = #flare-tie

```

```

\shape #'((0 . 0) (0 . -1) (0 . -1) (0 . 0)) Tie
a4~ a \break

<>~\markup \small \typewriter
      "#(flared-tie '((0.2 . 2) (0.5 . -3) (0.8 . 1)))"
\once \override Tie.stencil =
      #(flared-tie '((0.2 . 2) (0.5 . -3) (0.8 . 1)))
a4~ a
<>~\markup \small \typewriter
      "#(flared-tie '((0.5 . 2)))"
\once \override Tie.stencil = #(flared-tie '((0.5 . 2)))
a'4~ a
}

```

1

2

3 height-limit = 14

4 height-limit = 0.5

5 \shape #'((0 . 0) (0 . -1) (0 . -1) (0 . 0))
#(flared-tie '((0.2 . 2) (0.5 . -3) (0.8 . 1)))

6 #(flared-tie '((0.5 . 2)))

Force a cancellation natural before accidentals

The following example shows how to force a natural sign before an accidental.

```

\relative c' {
  \key es \major
  bes c des
  \tweak Accidental.restore-first ##t
  eis
}

```



Horizontale Verschiebung von Noten erzwingen

Wenn es zu Zusammenstößen kommt, kann mit folgender Lösung eine andere Position manuell eingestellt werden. Die Einheiten hier sind Notenlinienzwischenräume.

```
\relative c' <<
{
  <d g>2 <d g>
}
\\
{
  <b f'>2
  \once \override NoteColumn.force-hshift = 1.7
  <b f'>2
}
>>
```



Fret diagrams explained and developed

This snippet shows many possibilities for obtaining and tweaking fret diagrams.

```
<<
\chords {
  a1 a \bar "||" \break
  \repeat unfold 3 {
    c c c d d \bar "||" \break
  }
}

\new Voice {
  % Set global properties of fret diagram
  \override TextScript.size = 1.2
  \override TextScript.fret-diagram-details
    .finger-code = #'below-string
  \override TextScript.fret-diagram-details
    .dot-color = #'black

  % 1
  %
  % A chord for ukulele.
  a'1^\markup
    \override #'(fret-diagram-details
      . ((string-count . 4)
        (dot-color . white)
        (finger-code . in-dot)))
```

```

\fret-diagram "4-2-2;3-1-1;2-o;1-o;"

% 2
%
% A chord for ukulele, with formatting defined in definition
% string: 1.2 * size, 4 strings, 4 frets, fingerings below,
% string dot radius .35 of fret spacing, dot position 0.55 of
% fret spacing.
a'1^\markup
  \override #'(fret-diagram-details
    . ((dot-color . white)
      (open-string . "o")))
  \fret-diagram
    "s:1.2;w:4;h:3;f:2;d:0.35;p:0.55;4-2-2;3-1-1;2-o;1-o;"

%%
%% These chords will be in normal orientation
%%

% 3
%
% C major for guitar, barred on third fret: verbose style,
% roman fret label, finger labels below string, straight barre.
c'1^\markup
  % 110% of default size
  \override #'(size . 1.1)
  \override #'(fret-diagram-details
    . ((number-type . roman-lower)
      (finger-code . below-string)
      (barre-type . straight)))
  \fret-diagram-verbose #'((mute 6)
    (place-fret 5 3 1)
    (place-fret 4 5 2)
    (place-fret 3 5 3)
    (place-fret 2 5 4)
    (place-fret 1 3 1)
    (barre 5 1 3))

% 4
%
% C major for guitar, barred on third fret: double barre used
% to test barre function, verbose style.
c'1^\markup
  % 110% of default size
  \override #'(size . 1.1)
  \override #'(fret-diagram-details
    . ((number-type . arabic)
      (dot-label-font-mag . 0.9)
      (finger-code . in-dot)
      (fret-label-font-mag . 0.6)

```



```

        (fret-label-vertical-offset . 0)
        (label-dir . -1)
        (mute-string . "M")
        (xo-font-magnification . 0.4)
        (xo-padding . 0.3)))
\ fret-diagram-verbose #'((mute 6)
                          (place-fret 5 3 1)
                          (place-fret 4 5 2)
                          (place-fret 3 5 3)
                          (place-fret 2 5 4)
                          (place-fret 1 3 1)
                          (barre 4 2 5)
                          (barre 5 1 3))

% 5
%
% C major for guitar, with capo on third fret: verbose style.
c'1~\markup
% 110% of default size
\override #'(size . 1.1)
\override #'(fret-diagram-details
              . ((number-type . roman-upper)
                (dot-label-font-mag . 0.9)
                (finger-code . none)
                (fret-label-vertical-offset . 0.5)
                (xo-font-magnification . 0.4)
                (xo-padding . 0.3)))
\ fret-diagram-verbose #'((mute 6)
                          (capo 3)
                          (open 5)
                          (place-fret 4 5 1)
                          (place-fret 3 5 2)
                          (place-fret 2 5 3)
                          (open 1))

% 6
%
% Simple D chord.
d'1~\markup
\override #'(fret-diagram-details
              . ((finger-code . below-string)
                (dot-radius . 0.35)
                (string-thickness-factor . 0.3)
                (dot-position . 0.5)
                (fret-count . 3)))
\ fret-diagram-terse "x;x;o;2-1;3-2;2-3;"

% 7
%
% Simple D chord, large top fret thickness.
d'1~\markup
\override #'(fret-diagram-details

```

```

        . ((finger-code . below-string)
          (dot-radius . 0.35)
          (dot-position . 0.5)
          (top-fret-thickness . 7)
          (fret-count . 3)))
\fret-diagram-terse "x;x;o;2-1;3-2;2-3;"

%%
%% These chords will be in landscape orientation
%%
\override TextScript.fret-diagram-details
        .orientation = #'landscape

% 8
%
% C major for guitar, barred on third fret: verbose style,
% roman fret label, finger labels below string, straight
% barre.
c'1^\markup
% 110% of default size
\override #'(size . 1.1)
\override #'(fret-diagram-details
        . ((number-type . roman-lower)
          (finger-code . below-string)
          (barre-type . straight)))
\fret-diagram-verbose #'((mute 6)
        (place-fret 5 3 1)
        (place-fret 4 5 2)
        (place-fret 3 5 3)
        (place-fret 2 5 4)
        (place-fret 1 3 1)
        (barre 5 1 3))

% 9
%
% C major for guitar, barred on third fret: Double barre
% used to test barre function, verbose style.
c'1^\markup
% 110% of default size
\override #'(size . 1.1)
\override #'(fret-diagram-details
        . ((number-type . arabic)
          (dot-label-font-mag . 0.9)
          (finger-code . in-dot)
          (fret-label-font-mag . 0.6)
          (fret-label-vertical-offset . 0)
          (label-dir . -1)
          (mute-string . "M")
          (xo-font-magnification . 0.4)
          (xo-padding . 0.3)))

```

```

\ fret-diagram-verbose #'((mute 6)
                        (place-fret 5 3 1)
                        (place-fret 4 5 2)
                        (place-fret 3 5 3)
                        (place-fret 2 5 4)
                        (place-fret 1 3 1)
                        (barre 4 2 5)
                        (barre 5 1 3))

% 10
%
% C major for guitar, with capo on third fret: verbose style.
c'1~\markup
% 110% of default size
\override #'(size . 1.1)
\override #'(fret-diagram-details
. ((number-type . roman-upper)
  (dot-label-font-mag . 0.9)
  (finger-code . none)
  (fret-label-vertical-offset . 0.5)
  (xo-font-magnification . 0.4)
  (xo-padding . 0.3)))
\ fret-diagram-verbose #'((mute 6)
                        (capo 3)
                        (open 5)
                        (place-fret 4 5 1)
                        (place-fret 3 5 2)
                        (place-fret 2 5 3)
                        (open 1))

% 11
%
% Simple D chord.
d'1~\markup
\override #'(fret-diagram-details
. ((finger-code . below-string)
  (dot-radius . 0.35)
  (dot-position . 0.5)
  (fret-count . 3)))
\ fret-diagram-terse "x;x;o;2-1;3-2;2-3;"

% 12
%
% Simple D chord, large top fret thickness.
d'1~\markup
\override #'(fret-diagram-details
. ((finger-code . below-string)
  (dot-radius . 0.35)
  (dot-position . 0.5)
  (top-fret-thickness . 7)
  (fret-count . 3)))
\ fret-diagram-terse "x;x;o;2-1;3-2;2-3;"

```

```

%%
%% These chords will be in opposing-landscape orientation.
%%
\override TextScript.fret-diagram-details
    .orientation = #'opposing-landscape

% 13
%
% C major for guitar, barred on third fret: verbose style,
% roman fret label, finger labels below string, straight
% barre.
c'1^\markup
    % 110% of default size
    \override #'(size . 1.1)
    \override #'(fret-diagram-details
        . ((number-type . roman-lower)
            (finger-code . below-string)
            (barre-type . straight)))
    \fret-diagram-verbose #'((mute 6)
        (place-fret 5 3 1)
        (place-fret 4 5 2)
        (place-fret 3 5 3)
        (place-fret 2 5 4)
        (place-fret 1 3 1)
        (barre 5 1 3))

% 14
%
% C major for guitar, barred on third fret: double barre
% used to test barre function, verbose style.
c'1^\markup
    % 110% of default size
    \override #'(size . 1.1)
    \override #'(fret-diagram-details
        . ((number-type . arabic)
            (dot-label-font-mag . 0.9)
            (finger-code . in-dot)
            (fret-label-font-mag . 0.6)
            (fret-label-vertical-offset . 0)
            (label-dir . -1)
            (mute-string . "M")
            (xo-font-magnification . 0.4)
            (xo-padding . 0.3)))
    \fret-diagram-verbose #'((mute 6)
        (place-fret 5 3 1)
        (place-fret 4 5 2)
        (place-fret 3 5 3)
        (place-fret 2 5 4)
        (place-fret 1 3 1)

```

```

                                (barre 4 2 5)
                                (barre 5 1 3))

% 15
%
% C major for guitar, with capo on third fret: verbose style.
c'1~\markup
% 110% of default size
\override #'(size . 1.1)
\override #'(fret-diagram-details
              . ((number-type . roman-upper)
                 (dot-label-font-mag . 0.9)
                 (finger-code . none)
                 (fret-label-vertical-offset . 0.5)
                 (xo-font-magnification . 0.4)
                 (xo-padding . 0.3)))
\fret-diagram-verbose #'((mute 6)
                        (capo 3)
                        (open 5)
                        (place-fret 4 5 1)
                        (place-fret 3 5 2)
                        (place-fret 2 5 3)
                        (open 1))

% 16
%
% Simple D chord.
d'1~\markup
\override #'(fret-diagram-details
              . ((finger-code . below-string)
                 (dot-radius . 0.35)
                 (dot-position . 0.5)
                 (fret-count . 3)))
\fret-diagram-terse "x;x;o;2-1;3-2;2-3;"

% 17
%
% Simple D chord, large top fret thickness.
d'1~\markup
\override #'(fret-diagram-details
              . ((finger-code . below-string)
                 (dot-radius . 0.35)
                 (dot-position . 0.5)
                 (top-fret-thickness . 7)
                 (fret-count . 3)))
\fret-diagram-terse "x;x;o;2-1;3-2;2-3;"
}
>>

\paper {
  ragged-right = ##t

```

```

system-system-spacing.basic-distance = 20
}

\layout {
  \context {
    \Score
    \override SpacingSpanner.spacing-increment = 3
  }
}

```

The image displays three staves of musical notation, each with five measures. Above each measure is a guitar fretboard diagram illustrating a specific chord and its fingering. The first staff shows two A chords. The second staff shows three C chords and two D chords. The third staff shows three C chords and two D chords. Each chord is accompanied by a fretboard diagram showing fingerings and string numbers.

Generate special note head shapes

When a note head with a special shape cannot easily be generated with graphic markup, a drawing specification for `ly:make-stencil` can be used to generate the shape. This snippet gives an example for a parallelogram-shaped note head.

Unfortunately, the available commands in a drawing specification are currently not documented (this is tracked in Issue #6874 (<https://gitlab.com/lilypond/lilypond/-/issues/6874>)); in any case, the used path sub-command has the following signature, quite similar to the `make-path-stencil` Scheme function.

```
(path thickness command-list line-cap-style line-join-style fill)
```

The commands in *command-list* resemble PostScript drawing commands but with arguments after the command name.

```

parallelogram =
  #(ly:make-stencil

```

```

'(path 0.1
  (rmoveto 0 0.25
    lineto 1.2 0.75
    lineto 1.2 -0.25
    lineto 0 -0.75
    lineto 0 0.25)
  round
  round
  #t)
(cons -0.05 1.25)
(cons -.75 .75))

myNoteHeads = \override NoteHead.stencil = \parallelogram
normalNoteHeads = \revert NoteHead.stencil

\relative c' {
  \myNoteHeads
  g4 d'
  \normalNoteHeads
  <f, \tweak stencil \parallelogram b e>4 d
}

```



Generating custom flags

The stencil property of the Flag grob can be set to a custom Scheme function to generate the glyph for the flag.

```

#(define-public (weight-flag grob)
  (let* ((stem-grob (ly:grob-parent grob X))
        (log (- (ly:grob-property stem-grob 'duration-log) 2))
        (is-up? (eqv? (ly:grob-property stem-grob 'direction) UP))
        (yext (if is-up? (cons (* log -0.8) 0) (cons 0 (* log 0.8))))
        (flag-stencil (make-filled-box-stencil '(-0.4 . 0.4) yext))
        (stroke-style (ly:grob-property grob 'stroke-style))
        (stroke-stencil (if (equal? stroke-style "grace")
                             (make-line-stencil 0.2 -0.9 -0.4 0.9 -0.4)
                             empty-stencil)))
    (ly:stencil-add flag-stencil stroke-stencil)))

% Create a flag stencil by looking up the glyph from the font
#(define (inverted-flag grob)
  (let* ((stem-grob (ly:grob-parent grob X))
        (dir (if (eqv? (ly:grob-property stem-grob 'direction) UP) "d" "u"))
        (flag (retrieve-glyph-flag "" dir "" grob))
        (line-thickness (ly:staff-symbol-line-thickness grob))
        (stem-thickness (ly:grob-property stem-grob 'thickness))
        (stem-width (* line-thickness stem-thickness))
        (stroke-style (ly:grob-property grob 'stroke-style)))
    (ly:stencil-add flag-stencil stroke-stencil)))

```

```

(stencil (if (null? stroke-style)
             flag
             (add-stroke-glyph flag stem-grob dir stroke-style "")))
(rotated-flag (ly:stencil-rotate-absolute stencil 180 0 0))
(ly:stencil-translate rotated-flag (cons (- (/ stem-width 2)) 0)))

snippetexamplenotes =
{
  \autoBeamOff c'8 d'16 c'32 d'64 \acciaccatura {c'8} d'64
}

{
  \time 1/4
  <>^"Normal flags"
  \snippetexamplenotes

  <>_"Custom flag: inverted"
  \override Flag.stencil = #inverted-flag
  \snippetexamplenotes

  <>^"Custom flag: weight"
  \override Flag.stencil = #weight-flag
  \snippetexamplenotes

  <>_"Revert to normal"
  \revert Flag.stencil
  \snippetexamplenotes
}

```



Glissando kann Grobs überspringen

NoteColumn-Grobs können bei Glissandos übersprungen werden.

```

\relative c' {
  a2 \glissando
  \once \override NoteColumn.glissando-skip = ##t
  f''4 d,
}

```



Hairpins with different line styles

Hairpins can take any style from line-interface: dashed-line, dotted-line, line, trill, or zigzag.

```

\relative c' {

```



```

c2\< c\!
\override Hairpin.style = #'dashed-line
c2\< c\!
\override Hairpin.style = #'dotted-line
c2\< c\!
\override Hairpin.style = #'line
c2\< c\!
\override Hairpin.style = #'trill
c2\< c\!
\override Hairpin.style = #'zigzag
c2\< c\!
\revert Hairpin.style
c2\< c\!
}

```



Horizontally aligning custom dynamics like „più f“

Some dynamic expressions involve additional text, like „sempre **pp**“. Since dynamics are usually centered under the note, the `\pp` would be displayed way after the note it applies to.

To correctly align the „sempre **pp**“ horizontally so that it is aligned as if it were only the `\pp`, there are several approaches:

- Simply use `\once \override DynamicText.X-offset = #-9.2` before the note with the dynamics to manually shift it to the correct position. Drawback: This has to be done manually each time you use that dynamic markup...
- Add some padding (`#:hspace 7.1`) into the definition of your custom dynamic mark so that after LilyPond center-aligns it, it is already correctly aligned. Drawback: The padding really takes up that space and does not allow any other markup or dynamics to be shown in that position.
- Shift the dynamic script `\once \overrideX-offset =` Drawback: `\once \override` is needed for every invocation!
- Set the dimensions of the additional text to 0 (using `#:with-dimensions '(0 . 0) '(0 . 0)`). Drawback: For LilyPond, „sempre“ has no extent now. This means it might put other stuff there, causing collisions (which are not detected by LilyPond’s collision detection algorithm!). There also seems to be some spacing, so it is not exactly the same alignment as without the additional text.
- Add an explicit shift directly inside the scheme function for the dynamic script.
- Set an explicit alignment inside the dynamic script. By default, this won’t have any effect, only if one sets `X-offset`! Drawback: One needs to set `DynamicText.X-offset`, which will apply to all dynamic texts! Also, it is aligned at the right edge of the additional text, not at the center of `\pp`.

```

\paper {
  ragged-right = ##f
  indent = 5\cm
}

```

% Solution 1: Using a simple markup with a particular halign value

```

% Drawback: It's a markup, not a dynamic command, so \dynamicDown
%           etc. will have no effect
semppMarkup = \markup { \halign #1.4 \italic "sempre" \dynamic "pp" }

% Solution 2: Using a dynamic script & shifting with
%           \once \override ...X-offset = ..
% Drawback: \once \override needed for every invocation
semppK =
#(make-dynamic-script
  (markup #:line
    (#:normal-text
      #:italic "sempre"
      #:dynamic "pp"))))

% Solution 3: Padding the dynamic script so the center-alignment
%           puts it at the correct position
% Drawback: the padding really reserves the space, nothing else can be there
semppT =
#(make-dynamic-script
  (markup #:line
    (#:normal-text
      #:italic "sempre"
      #:dynamic "pp"
      #:hspace 7.1))))

% Solution 4: Dynamic, setting the dimensions of the additional text to 0
% Drawback: To lilypond "sempre" has no extent, so it might put
%           other stuff there => collisions
% Drawback: Also, there seems to be some spacing, so it's not exactly the
%           same alignment as without the additional text
semppM =
#(make-dynamic-script
  (markup #:line
    (#:with-dimensions '(0 . 0) '(0 . 0)
      #:right-align
      #:normal-text
      #:italic "sempre"
      #:dynamic "pp"))))

% Solution 5: Dynamic with explicit shifting inside the scheme function
semppG =
#(make-dynamic-script
  (markup #:hspace 0
    #:translate '(-18.85 . 0)
    #:line (#:normal-text
      #:italic "sempre"
      #:dynamic "pp"))))

% Solution 6: Dynamic with explicit alignment. This has only effect
%           if one sets X-offset!
% Drawback: One needs to set DynamicText.X-offset!
% Drawback: Aligned at the right edge of the additional text,

```

```

%           not at the center of pp
semppMII =
#(make-dynamic-script
  (markup #:line (#:right-align
                  #:normal-text
                  #:italic "sempre"
                  #:dynamic "pp")))

\new StaffGroup <<
  \new Staff \with { instrumentName = "standard" }
    \relative c'' {
      \key es \major
      c4\pp c\p c c | c\ff c c\pp c
    }
  \new Staff \with { instrumentName = "normal markup" }
    \relative c'' {
      \key es \major
      c4-\semppMarkup c\p c c | c\ff c c-\semppMarkup c
    }
  \new Staff \with { instrumentName = "explicit shifting" }
    \relative c'' {
      \key es \major
      \once \override DynamicText.X-offset = #-9.2
      c4\semppK c\p c c
      c4\ff c
      \once \override DynamicText.X-offset = #-9.2
      c4\semppK c
    }
  \new Staff \with { instrumentName = "right padding" }
    \relative c'' {
      \key es \major
      c4\semppT c\p c c | c\ff c c\semppT c
    }
  \new Staff \with { instrumentName = "set dimension to zero" }
    \relative c'' {
      \key es \major
      c4\semppM c\p c c | c\ff c c\semppM c
    }
  \new Staff \with { instrumentName = "shift inside dynamics" }
    \relative c'' {
      \key es \major
      c4\semppG c\p c c | c\ff c c\semppG c
    }
  \new Staff \with { instrumentName = "alignment inside dynamics" }
    \relative c'' {
      \key es \major
      \override DynamicText.X-offset = #-1
      c4\semppMII c\p c c | c\ff c c\semppMII c
    }
>>

\layout { \override Staff.InstrumentName.self-alignment-X = #LEFT }

```

standard	
normal markup	
explicit shifting	
right padding	
set dimension to zero	
shift inside dynamics	
alignment inside dynamics	

How to change fret diagram position

If you want to move the position of a fret diagram, for example, to avoid collision, or to place it between two notes, you have various possibilities.

- 1) Modify the value of the padding or extra-offset property (as shown in the first line).
- 2) You can add an invisible voice and attach the fret diagrams to the invisible notes in that voice (as shown in the second line).

If you need to move the fret according with a rhythmic position inside the bar (in the example, the third beat of the measure) the second example is better, because the fret is aligned with the third beat itself.

```

harmonies = \chordmode
{
  a8:13
  \once \override ChordNames.ChordName.extra-offset = #'(10 . 0)
  b8:13 s4. |
  s2 b2:13
}

\score {
  <<
    \new ChordNames \harmonies
    \new Staff {
      % Method 1.
      a8~\markup \fret-diagram "6-x;5-0;4-2;3-0;2-0;1-2;"
      \once \override TextScript.extra-offset = #'(10 . 0)
      b4.~\markup \fret-diagram "6-x;5-2;4-4;3-2;2-2;1-4;"
    }
  }

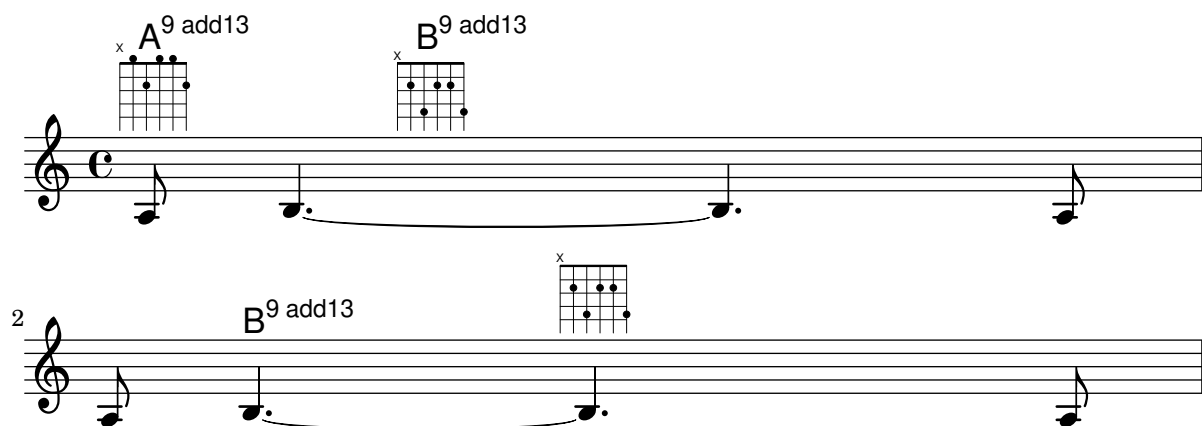
```

```

b4. a8 | \break

% Method 2.
<<
{ a8 b4.~ b4. a8 }
{ s2 s2~\markup \fret-diagram "6-x;5-2;4-4;3-2;2-2;1-4;" }
>> |
}
>>
}

```



Eine Zäsur einfügen

Zäsurzeichen können erstellt werden, indem die 'text'-Eigenschaft des BreathingSign-Objektes verändert wird. Ein gekrümmtes Zäsurzeichen ist auch möglich.

```

\relative c' {
  \override BreathingSign.text = \markup {
    \musicglyph "scripts.caesura.straight"
  }
  c8 e4. \breathe g8. e16 c4

  \override BreathingSign.text = \markup {
    \musicglyph "scripts.caesura.curved"
  }
  g8 e'4. \breathe g8. e16 c4
}

```



Keep change clefs full-sized

When a clef changes, the clef sign displayed is smaller than the initial clef. This can be overridden by setting the context property full-size-change to #t.

```

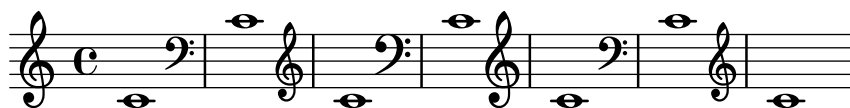
\relative c' {
  \clef "treble"
  c1
  \clef "bass"
  c1
}

```

```

\clef "treble"
c1
\override Staff.Clef.full-size-change = ##t
\clef "bass"
c1
\clef "treble"
c1
\revert Staff.Clef.full-size-change
\clef "bass"
c1
\clef "treble"
c1
}

```



Line arrows

Arrows can be applied to text spanners and line spanners (such as glissandi).

```

\relative c' ' {
  \override TextSpanner.bound-padding = #1.0
  \override TextSpanner.style = #'line
  \override TextSpanner.bound-details.right.arrow = ##t
  \override TextSpanner.bound-details.left.text = #"fof"
  \override TextSpanner.bound-details.right.text = #"gag"
  \override TextSpanner.bound-details.right.padding = #0.6

  \override TextSpanner.bound-details.right.stencil-align-dir-y = #CENTER
  \override TextSpanner.bound-details.left.stencil-align-dir-y = #CENTER

  \override Glissando.bound-details.right.arrow = ##t
  \override Glissando.arrow-length = #0.5
  \override Glissando.arrow-width = #0.25

  a8\startTextSpan gis a4 b\glissando b,
  g'4 c\stopTextSpan c2
}

```



Making an object invisible using \hide

Applying `\hide` to a grob causes objects of this type to be printed with „invisible ink“. They are not printed, but all of their other behavior is retained:

- the objects still take up space,
- they take part in collision resolution, and
- slurs, ties, and beams can be attached to them as usual.

This snippet demonstrates how to connect different voices using ties. Normally, ties only connect two notes in the same voice. By introducing a tie in a different voice, and blanking the first up-stem in that voice, the tie appears to cross voices.

```
\relative {
  \time 2/4
  <<
  {
    \once \hide Stem
    \once \override Stem.length = #8
    b'8 ~ 8\noBeam
    \once \hide Stem
    \once \override Stem.length = #8
    g8 ~ 8\noBeam
  }
  \\\
  {
    b8 g g e
  }
  >>
}

\paper {
  line-width = 40\mm
  ragged-right = ##f
}
```



Making glissandi breakable

Normally, LilyPond refuses to automatically break a line at places where a glissando crosses a bar line. This behavior can be changed by setting the `Glissando.breakable` property to `#t`. Also setting the after-line-breaking property to `#t` makes the glissando line continue after the break.

The breakable property does not affect manual breaks inserted with commands like `\break`.

```
glissandoSkipOn = {
  \override NoteColumn.glissando-skip = ##t
  \hide NoteHead
  \override NoteHead.no-ledgers = ##t
}

music = {
  \repeat unfold 16 f8 |
  f1\glissando |
  a4 r2. |
  \repeat unfold 16 f8 |
  f1\glissando \once\glissandoSkipOn |
  a2 a4 r4 |
  \repeat unfold 16 f8
}
```

}

```
\relative c' {
  <>^\markup { \typewriter Glissando.breakable
               set to \typewriter "#t" }
  \override Glissando.breakable = ##t
  \override Glissando.after-line-breaking = ##t
  \music
}
```

```
\relative c' {
  <>^\markup { \typewriter Glissando.breakable not set }
  \music
}
```

```
\paper {
  line-width = 100\mm
}
```

Glissando.breakable set to #t

Glissando.breakable not set

Manually controlling beam positions

Beam positions may be controlled manually by setting the `positions` property of the `Beam` grob.

```
\relative c' {
  \time 2/4
  % from upper staff-line (position 2) to center (position 0)
  \override Beam.positions = #'(2 . 0)
  c8 c
  % from center to one above center (position 1)
  \override Beam.positions = #'(0 . 1)
  c8 c
}
```


}



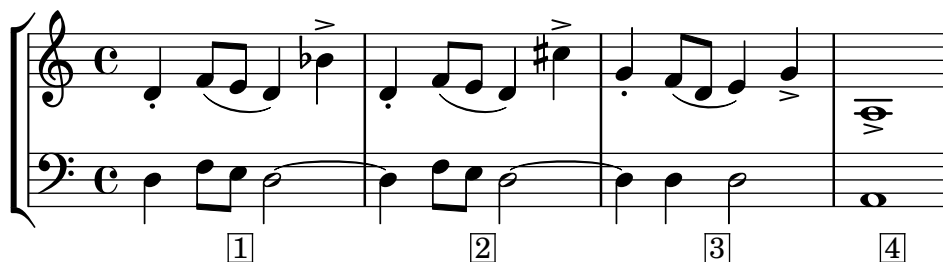
Measure-centered bar numbers

For film scores, a common convention is to center bar numbers within their measure. This is achieved through setting the `centerBarNumbers` context property to `#t`. When this is used, the type of the bar number grobs is `CenteredBarNumber` rather than `BarNumber`.

This example demonstrates a number of settings: the centered bar numbers are boxed and placed below the staves.

```
\layout {
  \context {
    \Score
    centerBarNumbers = ##t
    barNumberVisibility = #all-bar-numbers-visible
    \override CenteredBarNumber.stencil
      = #(make-stencil-boxer 0.1 0.25 ly:text-interface::print)
    \override CenteredBarNumberLineSpanner.direction = #DOWN
  }
}
```

```
\new StaffGroup <<
  \new Staff \relative c' {
    d4-. f8( e d4) bes'-> |
    d,-. f8( e d4) cis'-> |
    g-. f8( d e4) g-> |
    a,1-> |
  }
  \new Staff \relative c {
    \clef bass
    d4 f8 e d2~ |
    4 f8 e d2~ |
    4 4 2 |
    a1 |
  }
>>
```



Mensurstriche-Layout (Taktstriche zwischen den Systemen)

Das Mensurstriche-Layout, in welchem die Taktlinien nicht auf den Systemen, sondern zwischen den Systemen gesetzt werden, kann mit einer `StaffGroup` anstelle von `ChoirStaff` erreicht werden. Die Taktlinien auf den Systemen werden mit der `transparent`-Eigenschaft ausgelöscht.

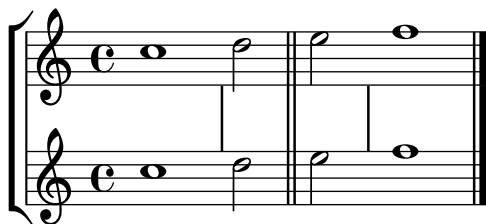
```

\layout {
  \context {
    \Staff
    measureBarType = "-span|"
  }
}

music = \fixed c'' {
  c1
  d2 \section e2
  f1 \fine
}

\new StaffGroup <<
  \new Staff \music
  \new Staff \music
>>

```



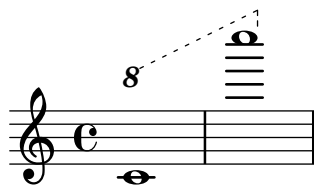
Modifying the ottava spanner slope

It is possible to change the slope of the ottava spanner.

```

\relative c'' {
  \override Staff.OttavaBracket.stencil = #ly:line-spanner::print
  \override Staff.OttavaBracket.bound-details =
    #`((left . ((Y . 0)
      (attach-dir . ,LEFT)
      (padding . 0)
      (stencil-align-dir-y . ,CENTER)))
      (right . ((Y . 5.0) ; Change the number here
      (padding . 0)
      (attach-dir . ,RIGHT)
      (text . ,(make-draw-dashed-line-markup
        (cons 0 -1.2))))))
  \override Staff.OttavaBracket.left-bound-info =
    #ly:horizontal-line-spanner::calc-left-bound-info-and-text
  \override Staff.OttavaBracket.right-bound-info =
    #ly:horizontal-line-spanner::calc-right-bound-info
  \ottava 1
  c1
  c'''1
}

```



Moving dotted notes in polyphony

When a dotted note in the upper voice is moved to avoid a collision with a note in another voice, the default is to move the upper note to the right. This behaviour can be changed setting the `prefer-dotted-right` property of the `NoteCollision` grob.

```
\new Staff \relative c' <<
{
  f2. f4
  \override Staff.NoteCollision.prefer-dotted-right = ##f
  f2. f4
  \override Staff.NoteCollision.prefer-dotted-right = ##t
  f2. f4
}
\\
{ e4 e e e e e e e e e e }
>>
```



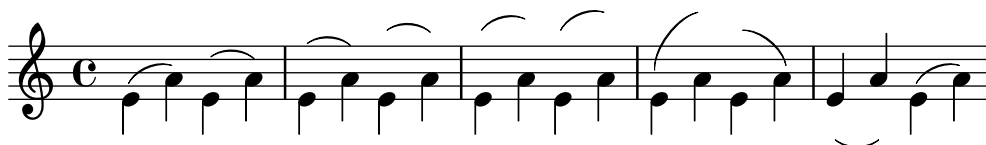
Moving slur positions vertically

The vertical position of a slur can be adjusted using the `positions` property of `Slur`. The property has 2 parameters, the first referring to the left end of the slur and the second to the right. The values of the parameters are not used by LilyPond to make an exact movement of the slur – instead it selects what placement of the slur looks best, taking into account the parameter values. Positive values move the slur up, and are appropriate for notes with stems down. Negative values move downward slurs further down.

See also snippet „Adjusting slur positions vertically“.

```
\relative c' {
  \stemDown
  e4( a)
  \override Slur.positions = #'(1 . 1)
  e4( a)
  \override Slur.positions = #'(2 . 2)
  e4( a)
  \override Slur.positions = #'(3 . 3)
  e4( a)
  \override Slur.positions = #'(4 . 4)
  e4( a)
  \override Slur.positions = #'(5 . 5)
  e4( a)
  \override Slur.positions = #'(0 . 5)
  e4( a)
  \override Slur.positions = #'(5 . 0)
```

```
e4( a )
\stemUp
\override Slur.positions = #'(-5 . -5)
e4( a )
\stemDown
\revert Slur.positions
e4( a )
}
```



Systeme schachteln

Die Eigenschaft `systemStartDelimiterHierarchy` kann eingesetzt werden, um komplizierte geschachtelte Systemklammern zu erstellen. Der Befehl `\setStaffGroup.systemStartDelimiterHierarchy` nimmt eine Liste mit der Anzahl der Systeme, die ausgegeben werden, auf. Vor jedem System kann eine Systemanfangsklammer angegeben werden. Sie muss in Klammern eingefügt werden und umfasst so viele Systeme, wie die Klammer einschließt. Elemente in der Liste können ausgelassen werden, aber die erste Klammer umfasst immer die gesamte Gruppe. Die Möglichkeiten der Anfangsklammer sind: `SystemStartBar`, `SystemStartBracket`, `SystemStartBrace` und `SystemStartSquare`.

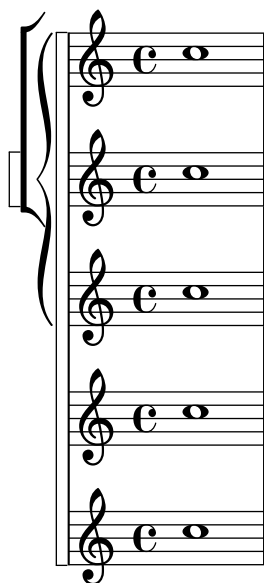
```

\new StaffGroup
\relative c' <<
  \override StaffGroup.SystemStartSquare.collapse-height = 4
  \set StaffGroup.systemStartDelimiterHierarchy
    = #'(SystemStartSquare
      (SystemStartBrace
        (SystemStartBracket a
          (SystemStartSquare b))
        c)
      d)

\new Staff { c1 }
\new Staff { c1 }
\new Staff { c1 }
\new Staff { c1 }
\new Staff { c1 }

>>

```



Overriding articulations by type

Sometimes you may want to affect a single articulation type. Although it is always possible to use `\tweak`, it might become tedious to do so for every single sign of a whole score. The following shows how to tweak articulations with a list of custom settings. One use-case might be to create a style sheet.

```
#(define (custom-script-tweaks ls)
  (lambda (grob)
    (let* ((type (ly:event-property (ly:grob-property grob 'cause)
                                     'articulation-type))
           (tweaks (assoc-ref ls type)))
      (when tweaks
        (for-each
         (lambda (x) (ly:grob-set-property! grob (car x) (cdr x)))
         tweaks))))))
```

```
customScripts =
#(define-music-function (settings) (list?)
  #{
    \override Script.before-line-breaking =
      #(custom-script-tweaks settings)
  })
revertCustomScripts = \revert Script.before-line-breaking
```

% Example

% Predefine two sets of desired tweaks.

```
#(define my-settings-1
  '((accent . ((font-size . 0)
                  (color . (1 0 0))))
    (segno . ((font-size . 0)
               (color . (1 0 0))))
    (staccato . ((color . (1 0 0)))))
```

```

        (padding . 0.5)))
    (staccatissimo . ((padding . 1)
                      (color . (1 0 0))))
    (tenuto . ((color . (1 0 0))
              (rotation . (45 0 0))
              (padding . 2)
              (font-size . 10)))
  ))

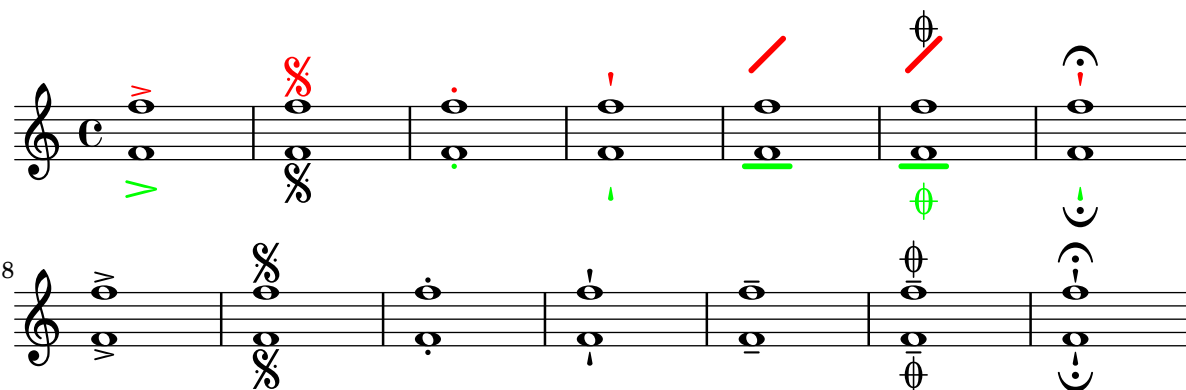
#(define my-settings-2
  '((accent . ((font-size . 4)
                (color . (0 1 0))
                (padding . 1.5)))
    (coda . ((color . (0 1 0))
             (padding . 1)))
    (staccato . ((color . (0 1 0))))
    (staccatissimo . ((padding . 2)
                      (color . (0 1 0))))
    (tenuto . ((color . (0 1 0))
              (font-size . 10)))
  ))

music = { f1-> | f\segno | f-. | f-! | f-- | f--\coda | f-!\fermata | }

block = {
  \music
  \break
  \revertCustomScripts \music
}

\new Staff <<
  \new Voice \with { \customScripts #my-settings-1 }
  \relative c'' { \voiceOne \block }
  \new Voice \with { \customScripts #my-settings-2 }
  \relative c' { \voiceTwo \block }
>>

```



Sichtbarkeit von Prozent-Wiederholungen

Prozentwiederholungszähler können in regelmäßigen Intervallen angezeigt werden, indem man die Eigenschaft `repeatCountVisibility` beeinflusst.

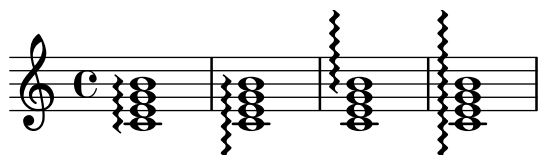
```
\relative c' {
  \set countPercentRepeats = ##t
  \set repeatCountVisibility = #(every-nth-repeat-count-visible 5)
  \repeat percent 10 { c1 } \break
  \set repeatCountVisibility = #(every-nth-repeat-count-visible 2)
  \repeat percent 6 { c1 d1 }
}
```



Positioning arpeggios

If you need to extend or shorten an arpeggio, you can modify the upper and lower start positions independently.

```
\relative c' {
  <c e g b>1\arpeggio
  \once \override Arpeggio.positions = #'(-5 . 0)
  <c e g b>1\arpeggio
  \once \override Arpeggio.positions = #'(0 . 5)
  <c e g b>1\arpeggio
  \once \override Arpeggio.positions = #'(-5 . 5)
  <c e g b>1\arpeggio
}
```



Positioning fingering indications precisely

The semi-automatic positioning of fingering within a chords works fine in most situations. If one of the indications needs to be positioned more precisely the following, tweaks as shown in this snippet may be used. This is particularly useful for correcting the positioning when intervals of a second are involved.

```
\markup \with-true-dimensions % work around a cropping issue
\score {
  \relative c' {
    \set fingeringOrientations = #'(left)
    <c-1 d-2 a'-5>4
    <c-1 d-\tweak extra-offset #'(0 . 0.2)-2 a'-5>

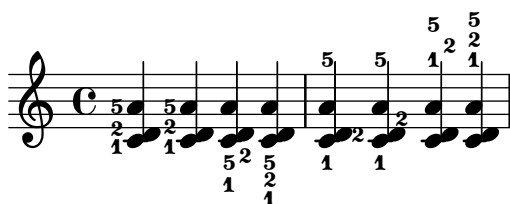
    \set fingeringOrientations = #'(down)
    <c-1 d-2 a'-5>
    <c-\tweak extra-offset #'(0 . -1.1)-1
    d-\tweak extra-offset #'(-1.2 . -1.8)-2 a'-5> |
  }
}
```

```

\set fingeringOrientations = #'(down right up)
<c-1 d-\tweak extra-offset #'(-0.3 . 0)-2 a'-5>4
<c-1 d-\tweak extra-offset #'(-1 . 1.2)-2 a'-5>

\set fingeringOrientations = #'(up)
<c-1 d-\tweak extra-offset #'(0 . 1.1)-2
  a'-\tweak extra-offset #'(0 . 1)-5>
<c-1 d-\tweak extra-offset #'(-1.2 . 1.5)-2
  a'-\tweak extra-offset #'(0 . 1.4)-5> |
}
}

```



Positionierung von Ganztaktpausen

Anders als bei normalen Pausen gibt es keinen direkten Befehl, um die vertikale Position von Ganztaktpausen zu beeinflussen, indem man sie an eine Tonhöhe anhängt. In polyphoner Notation wird aber dennoch die Position der Pausen von geraden und ungeraden Stimmen voneinander unterschieden. Die Position von Ganztaktpausen kann wie folgt verändert werden:

```

\relative c'' {
  % Multi-measure rests by default are set under the fourth line.
  R1
  % They can be moved using an override or tweak.
  \tweak staff-position -2 R1
  \tweak staff-position 0 R1
  \tweak staff-position 2 R1
  \override MultiMeasureRest.staff-position = 3 R1
  \override MultiMeasureRest.staff-position = 6 R1
  \revert MultiMeasureRest.staff-position
  \break

  % Odd-numbered voices are under the top line.
  << { R1 } \ { a1 } >>
  % Even-numbered voices are under the bottom line.
  << { a1 } \ { R1 } >>
  % Multi-measure rests in both voices remain separate.
  << { R1 } \ { R1 } >>

  % Separating multi-measure rests in more than two voices
  % requires an override or tweak.
  << { R1 } \ { R1 } \ { \tweak staff-position -2 R1 } >>

  % Using compressed bars in multiple voices requires another override
  % in all voices to avoid multiple instances being printed.
  \compressMMRests
  <<
    \revert MultiMeasureRest.direction

```



```

{ R1*3 } \\  

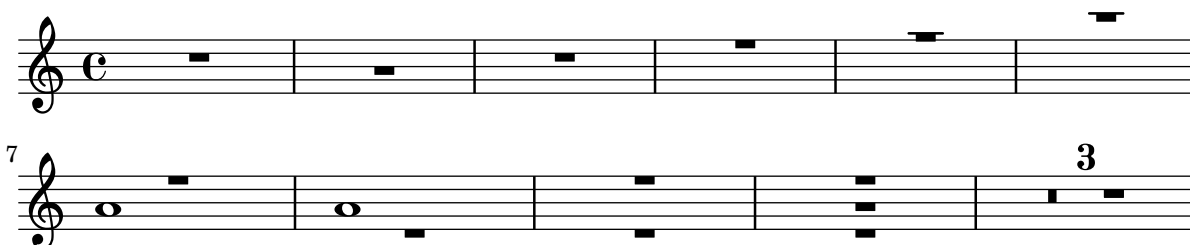
\revert MultiMeasureRest.direction  

{ R1*3 }  

>>  

}

```



Textbeschriftung innerhalb von Bögen positionieren

Textbeschriftung kann innerhalb von Bögen gesetzt werden, wenn die `outside-staff-priority`-Eigenschaft auf falsch gesetzt wird.

```

\relative c' {  

  \override TextScript.avoid-slur = #'inside  

  \override TextScript.outside-staff-priority = ##f  

  c2(~\markup { \halign #-10 \natural } d4.) c8  

}

```



Setzen von Taktnummern in Kästen oder Kreisen

Taktnummern können auch in Boxen oder Kreisen gesetzt werden.

```

\relative c' {  

  % Center bar numbers except at the beginning of a staff.  

  \override Score.BarNumber.self-alignment-X =  

    #(break-alignment-list CENTER CENTER 0.3)  
  

  % Prevent bar numbers at the end of a line and permit them elsewhere.  

  \override Score.BarNumber.break-visibility = #end-of-line-invisible  
  

  \set Score.barNumberVisibility = #(every-nth-bar-number-visible 4)  
  

  % Increase the size of the bar number by 2.  

  \override Score.BarNumber.font-size = 2  
  

  % Draw a circle round the following bar number(s).  

  \override Score.BarNumber.stencil  

    = #(make-stencil-circler 0.1 0.25 ly:text-interface::print)  

  \repeat unfold 7 { c1 } \break  
  

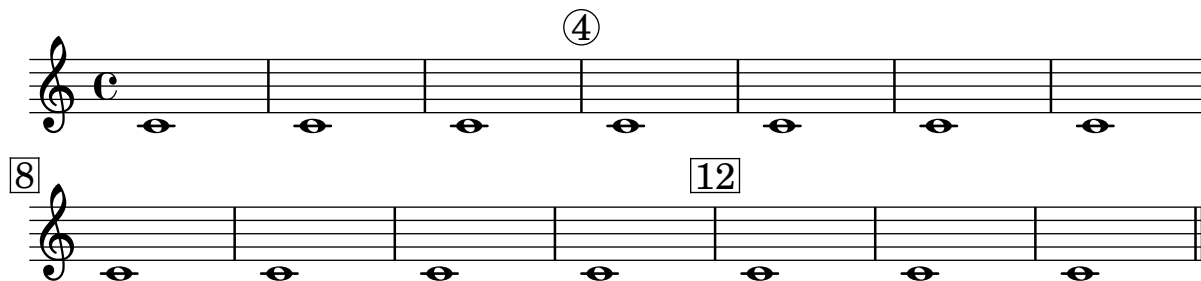
  % Draw a box round the following bar number(s).  

  \override Score.BarNumber.stencil  

    = #(make-stencil-boxer 0.1 0.25 ly:text-interface::print)

```

```
\repeat unfold 7 { c1 } \bar "|."
}
```



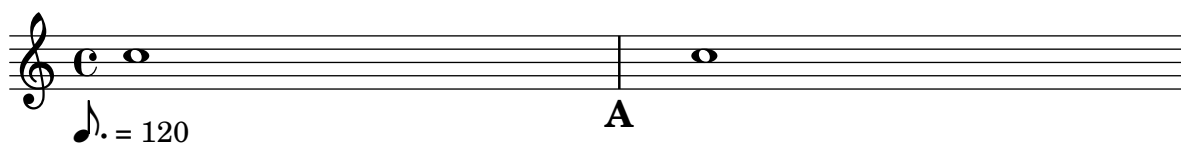
Metronom- und Übungszeichen unter das System setzen

Normalerweise werden Metronom- und Übungszeichen über dem Notensystem ausgegeben. Um sie unter das System zu setzen, muss die `direction`-Eigenschaft von `MetronomeMark` oder `RehearsalMark` entsprechend verändert werden.

```
\layout {
  ragged-right = ##f
}

{
  % Metronome marks below the staff
  \override Score.MetronomeMark.direction = #DOWN
  \tempo 8. = 120
  c''1

  % Rehearsal marks below the staff
  \override Score.RehearsalMark.direction = #DOWN
  \mark \default
  c''1
}
```



Printing note names with and without an octave marker

The `NoteNames` context can be used to print the text value of notes. The `printOctaveNames` property turns on or off the representation of the octave of the note.

```
scale = \relative c' {
  a4 b c d
  e4 f g a
}

\new Staff {
  <<
  \scale
  \context NoteNames {
    \set printOctaveNames = ##f
  }
  \scale
```

```

    }
  >>
  R1
  <<
    \scale
    \context NoteNames {
      \set printOctaveNames = ##t
      \scale
    }
  >>
}

\layout {
  \context {
    \NoteNames
    % Allow vertical overlapping of different `NoteNames` contexts
    % to make them appear as if they were a single line.
    \override VerticalAxisGroup
      .nonstaff-nonstaff-spacing
      .minimum-distance = ##f
  }
}

```



Printing tuplet brackets on the note head side

Whichever option you choose for controlling the tuplet bracket visibility, it will show or hide the tuplet bracket irrespectively of tuplet bracket placement (stem side or note head side). However, when placing the tuplet bracket on the note head side some authors recommend always printing the tuplet bracket. The option `visible-over-note-heads` can be used to achieve this.

```

music = \relative c'' {
  \tupletNeutral \tuplet 3/2 { c16[ d e ] f8]
  \tupletUp \tuplet 3/2 { c8 d e }
}

\new Voice {
  \relative c' {
    \override TextScript.staff-padding = #2.5

    \time 2/4
    \override TupletBracket.visible-over-note-heads = ##t
    \override Score.TextMark.non-musical = ##f
    <>~\markup "default" \music
    \override TupletBracket.bracket-visibility = #'if-no-beam
    <>~\markup \typewriter "if-no-beam" \music
  }
}

```

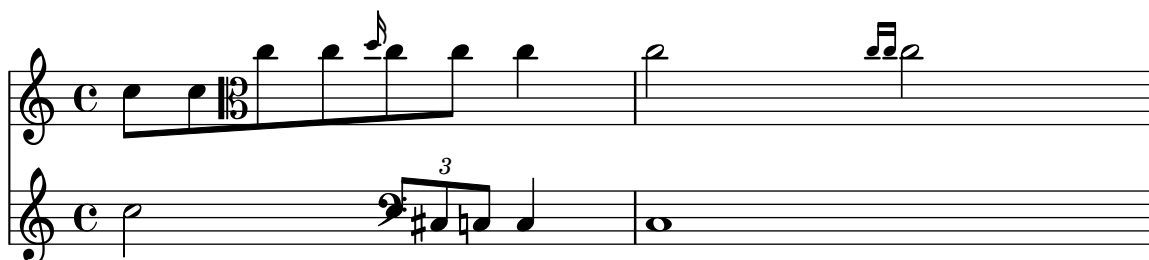


Proportional strict notespacing

If the `strict-note-spacing` property of the `SpacingSpanner` grob is set to `#t`, spacing of notes is not influenced by bars or clefs within a system. Rather, they are placed just before the note that occurs at the same time. This may cause collisions.

```
\relative c' ' <<
  \override Score.SpacingSpanner.strict-note-spacing = ##t
  \set Score.proportionalNotationDuration = #1/16

  \new Staff {
    c8[ c \clef alto c c \grace { d16 } c8 c] c4
    c2 \grace { c16[ c16] } c2
  }
  \new Staff {
    c2 \tuplet 3/2 { c8 \clef bass cis,, c } c4
    c1
  }
>>
```



Removing brace on first line of piano score

This snippet removes the first brace from a `PianoStaff` or a `GrandStaff`, together with the clefs. It may be useful when cutting and pasting the engraved image into existing music.

The code uses `\alterBroken` to hide the brace delimiter at the beginning.

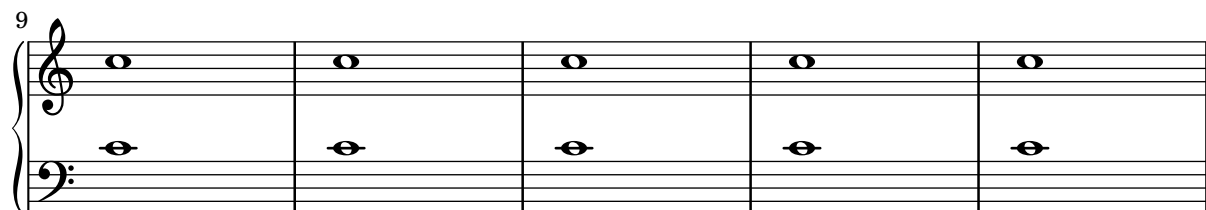
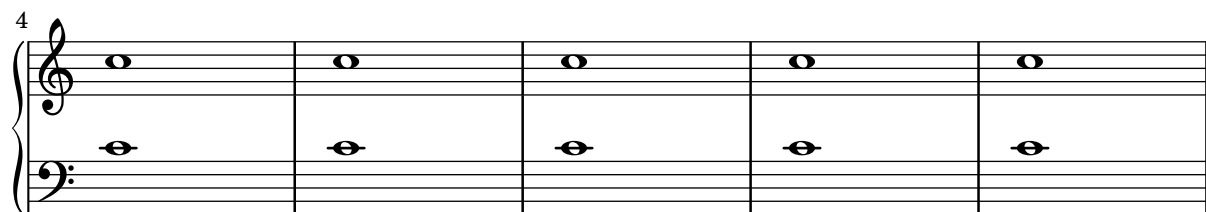
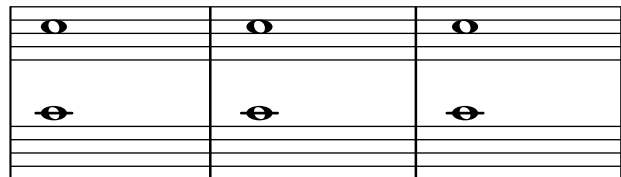
```
someMusic = {
  \once \omit Staff.Clef
  \once \omit Staff.TimeSignature
  \repeat unfold 3 c1 \break
  \repeat unfold 5 c1 \break
  \repeat unfold 5 c1
}

\score {
  \new PianoStaff
  <<
    \new Staff = "right" \relative c' ' \someMusic
    \new Staff = "left" \relative c' { \clef F \someMusic }
  >>
  \layout {
```

```

    indent=75\mm
    \context {
      \PianoStaff
      \alterBroken transparent #'(#t) SystemStartBrace
    }
  }
}

```



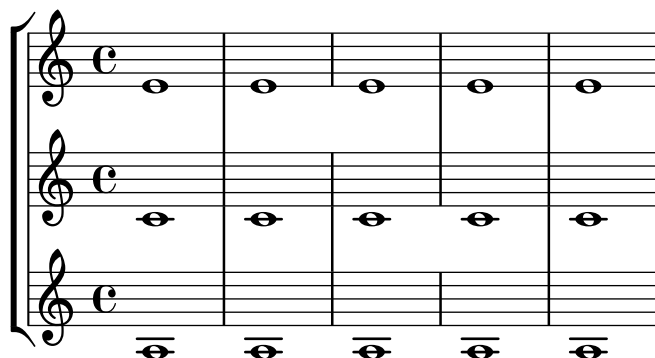
Removing connecting bar lines on StaffGroup, PianoStaff, or GrandStaff

By default, bar lines in StaffGroup, PianoStaff, or GrandStaff contexts are connected between the staves, i.e., a span bar is printed. This behaviour can be overridden on a staff-by-staff basis.

```

\relative c' {
  \new StaffGroup <<
    \new Staff {
      e1 | e
      \once \override Staff.BarLine.allow-span-bar = ##f
      e1 | e | e
    }
    \new Staff {
      c1 | c | c
      \once \override Staff.BarLine.allow-span-bar = ##f
      c1 | c
    }
    \new Staff {
      a1 | a | a | a | a
    }
  >>
}

```



Die erste leere Notenzeile auch entfernen

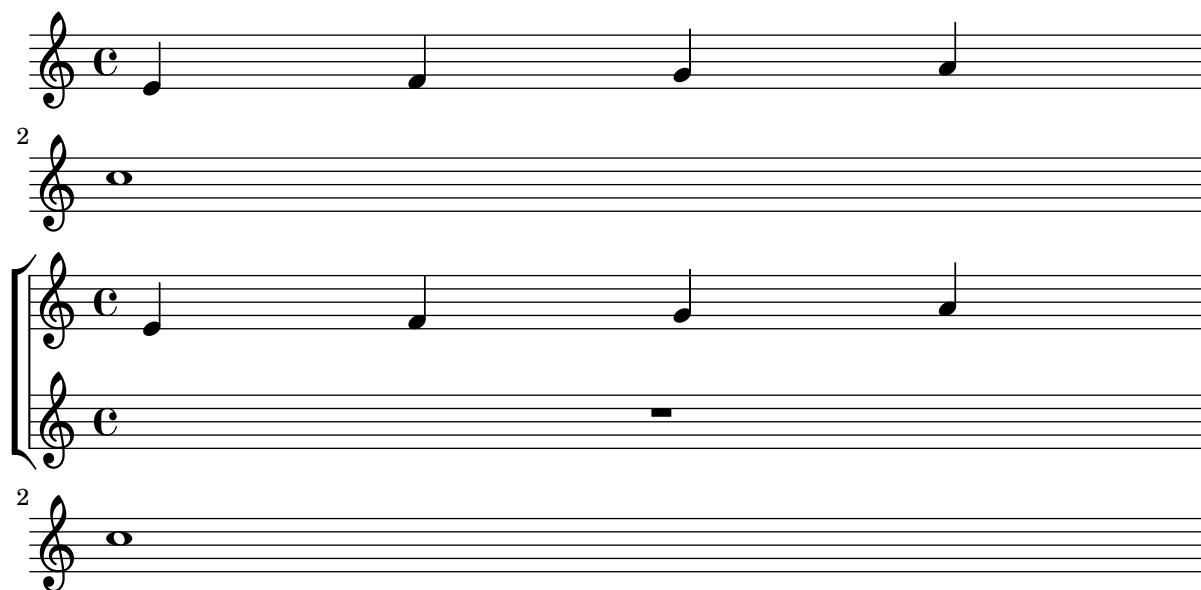
Ein leeres Notensystem kann auch aus der ersten Zeile einer Partitur entfernt werden, indem die Eigenschaft `remove-first` der `VerticalAxisGroup`-Eigenschaft eingesetzt wird. Das kann man global in einer `\layout`-Umgebung oder lokal in dem bestimmten Notensystem machen, das entfernt werden soll. In letzterem Fall muss man den Kontext angeben.

Das untere Notensystem der zweiten Systemgruppe wird nicht entfernt, weil in die Einstellungen in dem Schnipsel nur für das eine Notensystem gültig sind.

```
\layout {
  \context {
    \Staff \RemoveEmptyStaves
    % To use the setting globally, uncomment the following line:
    % \override VerticalAxisGroup.remove-first = ##t
  }
}

\new StaffGroup <<
  \new Staff \relative c' {
    e4 f g a \break
    c1
  }
  \new Staff {
    % To use the setting globally, comment this line,
    % uncomment the line in the \layout block above
    \override Staff.VerticalAxisGroup.remove-first = ##t
    R1 \break
    R
  }
}
>>
```

```
\new StaffGroup <<
  \new Staff \relative c' {
    e4 f g a \break
    c1
  }
  \new Staff {
    R1 \break
    R
  }
}
>>
```



Pausenstile

Pausen können in verschiedenen Stilen dargestellt werden.

```
restsA = {
  r\maxima r\longa r\breve r1 r2 r4 r8 r16 s32
  s64 s128 s256 s512 s1024 s1024
}

restsB = {
  r\maxima r\longa r\breve r1 r2 r4 r8 r16 r32
  r64 r128 r256 r512 r1024 s1024
}

\new Staff \relative c {
  \omit Score.TimeSignature
  \cadenzaOn

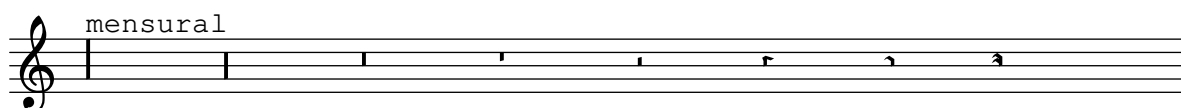
  \override Staff.Rest.style = #'mensural
  <>^\markup \typewriter { mensural } \restsA \bar "" \break

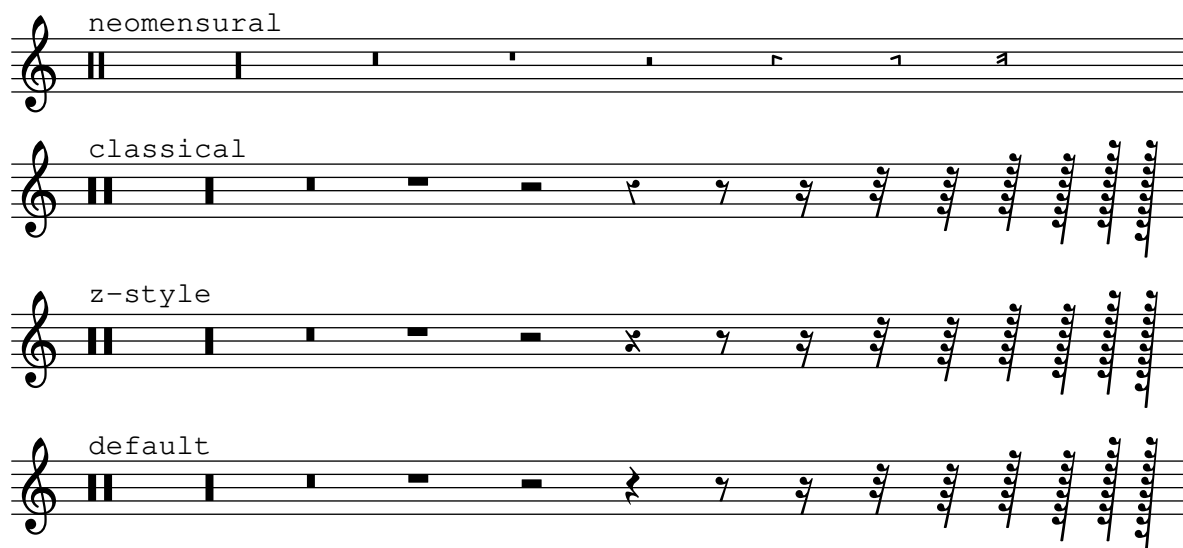
  \override Staff.Rest.style = #'neomensural
  <>^\markup \typewriter { neomensural } \restsA \bar "" \break

  \override Staff.Rest.style = #'classical
  <>^\markup \typewriter { classical } \restsB \bar "" \break

  \override Staff.Rest.style = #'z
  <>^\markup \typewriter { z-style } \restsB \bar "" \break

  \override Staff.Rest.style = #'default
  <>^\markup \typewriter { default } \restsB \bar "" \break
}
```



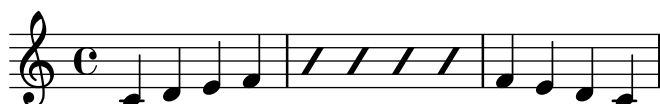


Rhythmic slashes

In „simple“ lead-sheets, sometimes no actual notes are written. Instead, only „rhythmic patterns“ and chords above the measures are notated to represent the structure of a song. Such a feature can be useful while creating or transcribing the structure of a song, or when sharing lead sheets with guitarists or jazz musicians.

```
startPat = {
  \improvisationOn
  \omit Stem
}
stopPat = {
  \improvisationOff
  \undo \omit Stem
}

\new Voice \with {
  \consists Pitch_squash_engraver
} {
  c'4 d' e' f' |
  \startPat
  4 4 4 4 |
  \stopPat
  f'4 e' d' c'
}
```



Separating key cancellations from key signature changes

By default, the accidentals used for key cancellations are placed adjacent to those for key signature changes. This behavior can be changed by overriding the `break-align-orders` property of the `BreakAlignment` grob.

The value of `break-align-orders` is a vector of length 3, with quoted lists of breakable items as elements. Each list describes the default order of prefatory matter at the end, in the middle,

and at the beginning of a line, respectively. We are only interested in changing the behaviour in the middle of a line.

If you look up the definition of `break-align-orders` in LilyPond's Internals Reference (<https://lilypond.org/doc/v2.24/Documentation/internals/breakalignment>), you get the following order in the second element:

```
...
staff-bar
key-cancellation
key-signature
...
```

We want to change that, moving `key-cancellation` before `staff-bar`. To make this happen we use the `grob-transformer` function, which gives us access to the original vector as the second argument of the lambda function, here called *orig* (we don't need the first argument, *grob*). We return a new vector, with unchanged first and last elements. For the middle element, we first remove `key-cancellation` from the list, then adding it again before `staff-bar`.

```
#(define (insert-before where what lst)
  (cond
    ((null? lst)           ; If the list is empty,
     (list what))         ; return a single-element list.
    ((eq? where (car lst)) ; If we find symbol `where`,
     (cons what lst))     ; insert `what` before curr. position.
    (else                  ; Otherwise keep building the list by
     (cons (car lst)      ; adding the current element and
           (insert-before ; recursing with the next element.
             where what (cdr lst))))))
```

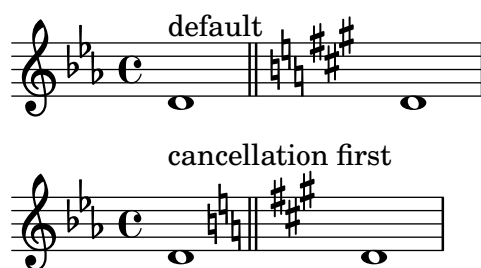
```
cancellationFirst =
\override Score.BreakAlignment.break-align-orders =
#(grob-transformer
  'break-align-orders
  (lambda (grob orig)
    (let* ((middle (vector-ref orig 1))
           (middle (delq 'key-cancellation middle))
           (middle (insert-before
                           'staff-bar 'key-cancellation middle)))
      (vector
        ;; end of line
        (vector-ref orig 0)
        ;; middle of line
        middle
        ;; beginning of line
        (vector-ref orig 2)))))
```

```
music = { \key es \major d'1 \bar "||"
          \key a \major d'1 }
```

```
{ <>^\markup "default"
  \music }
```

```
{ <>^\markup "cancellation first"
  \cancellationFirst
```

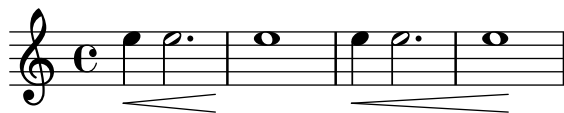
```
\music }
```



Das Verhalten von Crescendo-Klammern an Taktlinien beeinflussen

Wenn die Note, an welcher eine Crescendo-Klammer endet, die erste Note eines Taktes ist, wird die Klammer an der vorhergehenden Taktlinie beendet. Dieses Verhalten kann auch mit der Eigenschaft `'to-barline` geändert werden:

```
\relative c' {
  e4\< e2.
  e1\!
  \override Hairpin.to-barline = ##f
  e4\< e2.
  e1\!
}
```



Setting system separators

System separators can be inserted between systems. Any markup can be used, but `\slashSeparator` has been provided as a sensible default.

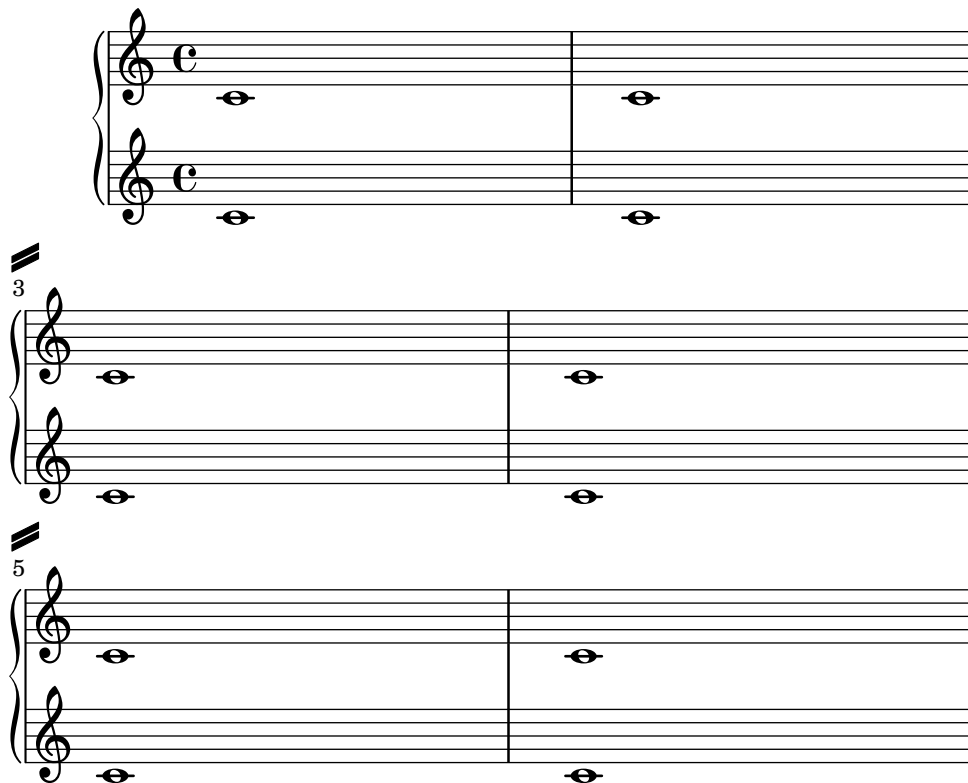
```
#{set-default-paper-size "a5")
```

```
\paper {
  system-separator-markup = \slashSeparator
  tagline = ##f
}
```

```
notes = \relative c' {
  c1 | c \break
  c1 | c \break
  c1 | c
}
```

```
\book {
  \score {
    \new GrandStaff <<
      \new Staff \notes
      \new Staff \notes
    >>
  }
}
```

}



Shape individual ties in chords

To shape individual ties in chords use the method demonstrated below.

```
{
  \textMark "Chords can be tied note by note."
  <c'~ e'~ g'~ c''~>2 q
}

{
  \textMark \markup \override #'(baseline-skip . 3) \wordwrap {
    Modifying those ties with \typewriter "\\shape" does not succeed,
    because \typewriter TieColumn positions them on its own behalf,
    ignoring \typewriter "\\shape" input more or less. You may
    circumvent this by setting \typewriter positioning-done to
    \typewriter "#t" -- alas, \typewriter positioning-done is an
    internal property, and setting it to \typewriter "#t" means: all
    positioning is done, don't do anything further. The next example
    demonstrates a case where the positioning is not finished: all tie
    directions are down, and the thickness is not accurate.
  }
  <c'~ e'~ g'~ c''~>2
  \once \override TieColumn.positioning-done = ##t
  q
}

{
  \textMark "To fix that, enter ties with explicit direction modifiers."
```

```

<c'~ e'~ g'~ c'~>2
\once \override TieColumn.positioning-done = ##t
q
}

{
\textMark \markup {
  Now you can use \typewriter "\\shape" for each tie as usual. }
<c'~\shape #'((0 . 0) (0 . -10) (0 . -10) (0 . 0)) _~
e'~\shape #'((0 . 0) (0 . -5) (0 . -5) (0 . 0)) _~
g'~\shape #'((0 . 0) (0 . -2) (0 . -2) (0 . 0)) _~
c'~\shape #'((0 . 0) (0 . 5) (0 . 5) (0 . 0)) ^~
>2
\once \override TieColumn.positioning-done = ##t
q
}

{
\textMark "This also works at line breaks."
<c'~\shape #'(((0 . 0) (0 . -10) (0 . -10) (0 . 0))
              ((0 . 0) (0 . -10) (0 . -10) (0 . 0))) _~
e'~\shape #'(((0 . 0) (0 . -5) (0 . -5) (0 . 0))
              ((0 . 0) (0 . -5) (0 . -5) (0 . 0))) _~
g'~\shape #'(((0 . 0) (0 . -2) (0 . -2) (0 . 0))
              ((0 . 0) (0 . -2) (0 . -2) (0 . 0))) _~
c'~\shape #'(((0 . 0) (0 . 5) (0 . 5) (0 . 0))
              ((0 . 0) (0 . 5) (0 . 5) (0 . 0))) ^~
>2
\break
\once \override TieColumn.positioning-done = ##t
q
}

{
\textMark \markup {
  It also works with the \typewriter tieWaitForNote property. }
\set tieWaitForNote = ##t
c'4~\shape #'((0 . 0) (0 . -10) (0 . -10) (0 . 0)) _~
e'~\shape #'((0 . 0) (0 . -5) (0 . -5) (0 . 0)) _~
g'~\shape #'((0 . 0) (0 . -2) (0 . -2) (0 . 0)) _~
c'~\shape #'((0 . 0) (0 . 5) (0 . 5) (0 . 0)) ^~
\once \override TieColumn.positioning-done = ##t
<c' e' g' c'>1
}

\layout {
  indent = 0
  \context {
    \Score
    \override TextMark.padding = #4
    \override TextMark.break-align-symbols = #'(left-edge)
  }
}

```

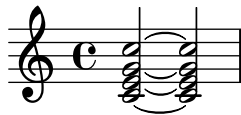
```

}

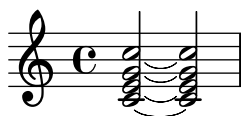
\paper {
  score-system-spacing.padding = 3
}

```

Chords can be tied note by note.



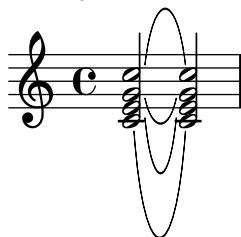
Modifying those ties with `\shape` does not succeed, because `TieColumn` positions them on its own behalf, ignoring `\shape` input more or less. You may circumvent this by setting `positioning-done` to `#t` – alas, `positioning-done` is an internal property, and setting it to `#t` means: all positioning is done, don't do anything further. The next example demonstrates a case where the positioning is not finished: all tie directions are down, and the thickness is not accurate.



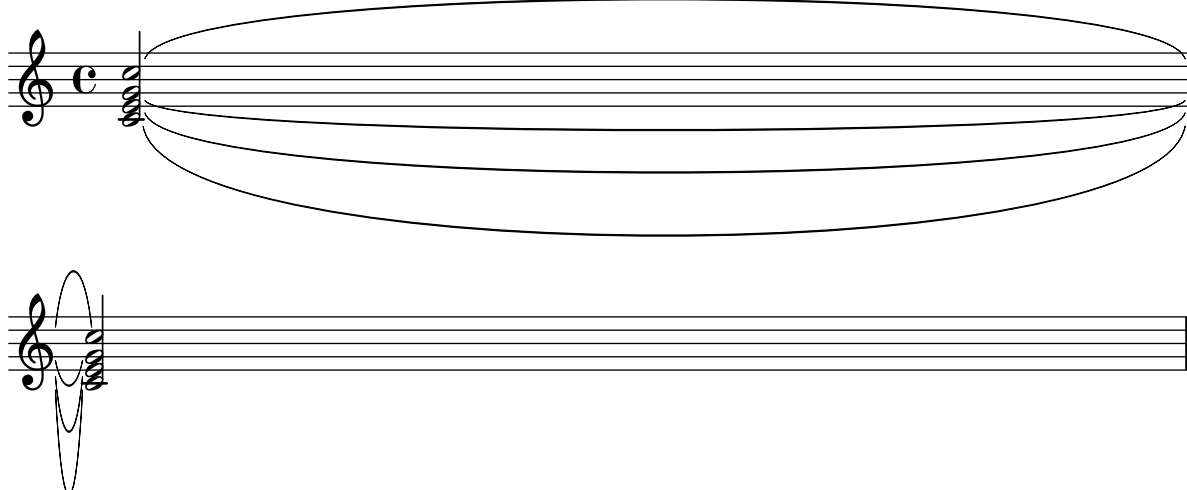
To fix that, enter ties with explicit direction modifiers.



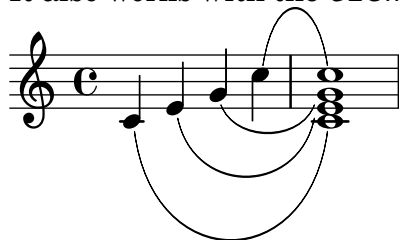
Now you can use `\shape` for each tie as usual.



This also works at line breaks.



It also works with the `tieWaitForNote` property.



Showing the same articulation above and below a note or chord

By default, LilyPond does not allow the same articulation (an accent, a fermata, a flageolet, etc.) to be displayed above and below a note. For example, `c4_\fermata^\fermata` only shows a fermata below. The fermata above gets simply ignored.

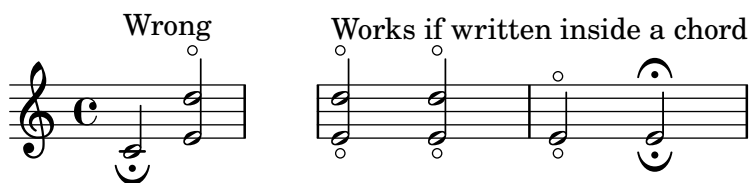
However, one can stick scripts (just like fingerings) inside a chord, which means it is possible to have as many articulations as desired. This approach has the advantage that it ignores the stem and positions the articulation relative to the note head. This can be seen in the case of the flageolets in the snippet. To mimic the behaviour of scripts outside a chord, `add-stem-support` would be required.

The solution is thus to write the note as a chord and add the articulations inside of `<...>`, using the direction modifiers `^` and `_` as appropriate.

```
\relative c' {
  <>^"Wrong"
  c2_\fermata^\fermata % The second fermata is ignored!
  <e d'>2^\flageolet_\flageolet

  \stopStaff s1 \startStaff

  <>^"Works if written inside a chord"
  <e_\flageolet d'^\flageolet>2
  <e_\flageolet d'^\flageolet>2
  <e_\flageolet^\flageolet>2
  <e_\fermata^\fermata>2
}
```



String number extender lines

Make an extender line for string number indications, showing that a series of notes is supposed to be played all on the same string.

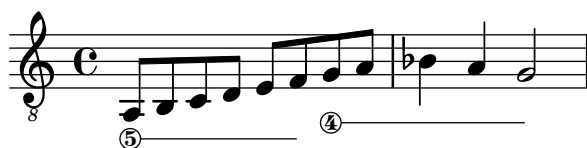
```
stringNumberSpanner =
  #(define-music-function (StringNumber) (string?)
    #{
      \override TextSpanner.style = #'solid
      \override TextSpanner.font-size = #-5
      \override TextSpanner.bound-details.left.stencil-align-dir-y = #CENTER
      \override TextSpanner.bound-details.left.text =
```

```

\markup { \circle \number $StringNumber }
#})

\relative c {
  \clef "treble_8"
  \textSpannerDown
  \stringNumberSpanner "5" a8\startTextSpan b c d
  e f\stopTextSpan \stringNumberSpanner "4" g\startTextSpan a |
  bes4 a g2\stopTextSpan
}

```



Suppressing warnings for clashing note columns

If notes from two voices with stems in the same direction are placed at the same position, and both voices have no shift or the same shift specified, the error message „warning: ignoring too many clashing note columns“ appears when compiling the LilyPond file. This message can be suppressed by setting the `ignore-collision` property of the `NoteColumn` object to `#t`. Please note that this does not just suppress warnings but stops LilyPond trying to resolve collisions at all and so may have unintended results unless used with care.

```
ignore = \override NoteColumn.ignore-collision = ##t
```

```

\relative c' {
  \new Staff <<
    \new Voice { \ignore \stemDown f2 g }
    \new Voice { c2 \stemDown c, }
  >>
}

```



Time signature in brackets

The time signature can be enclosed within brackets.

```

\relative c'' {
  \override Staff.TimeSignature.stencil = #(\lambda (grob)
    (bracketify-stencil (ly:time-signature::print grob) Y 0.1 0.2 0.1))
  \time 2/4
  a4 b8 c
}

```



Time signature in parentheses

The time signature can be enclosed within parentheses.

```
\relative c' ' {
  \override Staff.TimeSignature.stencil = #(lambda (grob)
    (parenthesize-stencil (ly:time-signature::print grob) 0.1 0.4 0.4 0.1))
  \time 2/4
  a4 b8 c
}
```



Time signature printing only the numerator as a number (instead of the fraction)

Sometimes, a time signature should not print the whole fraction (for example, 7/4), but only the numerator (digit 7 in this case). This can be easily done by using `\override Staff.TimeSignature.style = #'single-number` to change the style permanently. By using `\revert Staff.TimeSignature.style`, this setting can be reversed. To apply the single-number style to only one time signature, use `\tweak`.

```
\relative c' ' {
  \time 3/4
  c4 c c
  % Change the style permanently
  \override Staff.TimeSignature.style = #'single-number
  \time 2/4
  c4 c
  \time 3/4
  c4 c c
  % Revert to default style:
  \revert Staff.TimeSignature.style
  \time 2/4
  c4 c
  % single-number style only for the next time signature
  \tweak style #'single-number \time 5/4
  c4 c c c c
  \time 2/4
  c4 c
}
```



Tuplet bracket and change staff

This snippet shows how to set a tuplet starting in a lower staff and finishing in the upper one.

```
aigues = \relative c' {
  \time 6/8
  s4. \stemDown c16[ bes' e] \stemUp g c e \stemDown |
  g8
}
```



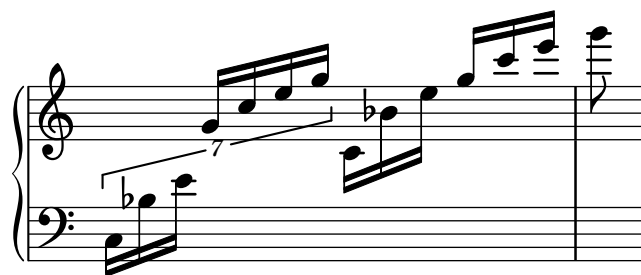
```

}

basses = \relative c {
  \time 3/4
  \clef F
  \tweak positions #'(4.5 . 8.5)
  \tweak edge-height #'(1 . -1)
  \tuplet 7/6 { c16[ bes' e] \change Staff = md \stemUp g[ c e g] } s4. |
s8
}

\new PianoStaff \with { \omit TimeSignature }
<<
  \new Staff = md \aigues
  \new Staff = mg \basses
>>

```



Eigenschaften des Schlüssels optimieren

Der Befehl `\clef "treble_8"` ist gleichbedeutend mit einem expliziten Setzen der Eigenschaften von `clefGlyph`, `clefPosition` (welche die vertikale Position des Schlüssels bestimmt), `middleCPosition` und `clefTransposition`. Ein Schlüssel wird ausgegeben, wenn eine der Eigenschaften außer `middleCPosition` sich ändert.

Eine Änderung des Schriftzeichens (Glyph), der Schlüsselposition oder der Oktavierung selber ändert noch nicht die Position der darauf folgenden Noten auf dem System: das geschieht nur, wenn auch die Position des eingestrichenen C (`middleCPosition`) angegeben wird. Die Positionsparameter sind relativ zur Mittellinie des Systems, dabei versetzen positive Zahlen die Position nach oben, jeweils eine Zahl für jede Linie plus Zwischenraum. Der `clefTransposition`-Wert ist normalerweise auf 7, -7, 15 oder -15 gesetzt, aber auch andere Werte sind gültig.

Wenn ein Schlüsselwechsel an einem Zeilenwechsel geschieht, wird das neue Symbol sowohl am Ende der alten Zeilen als auch am Anfang der neuen Zeile ausgegeben. Wenn der Warnungs-Schlüssel am Ende der alten Zeile nicht erforderlich ist, kann er unterdrückt werden, indem die `explicitClefVisibility`-Eigenschaft des Staff-Kontextes auf den Wert `end-of-line-invisible` gesetzt wird. Das Standardverhalten kann mit `\unset Staff.explicitClefVisibility` wieder hergestellt werden.

Die folgenden Beispiele zeigen die Möglichkeiten, wenn man diese Eigenschaften manuell setzt. Auf der ersten Zeile erhalten die manuellen Änderungen die ursprüngliche relative Positionierung von Schlüssel und Noten, auf der zweiten Zeile nicht.

```

{
  % The default treble clef.
  \key f \major
  c'1
  % The standard bass clef

```

```

\set Staff.clefGlyph = "clefs.F"
\set Staff.clefPosition = 2
\set Staff.middleCPosition = 6
\set Staff.middleCClefPosition = 6
\key g \major
c'1
% The baritone clef.
\set Staff.clefGlyph = "clefs.C"
\set Staff.clefPosition = 4
\set Staff.middleCPosition = 4
\set Staff.middleCClefPosition = 4
\key f \major
c'1
% The standard choral tenor clef.
\set Staff.clefGlyph = "clefs.G"
\set Staff.clefPosition = -2
\set Staff.clefTransposition = -7
\set Staff.middleCPosition = 1
\set Staff.middleCClefPosition = 1
\key f \major
c'1
% A non-standard clef.
\set Staff.clefPosition = 0
\set Staff.clefTransposition = 0
\set Staff.middleCPosition = -4
\set Staff.middleCClefPosition = -4
\key g \major
c'1 \break

% The following clef changes do not preserve
% the normal relationship between notes, key signatures
% and clefs.
\set Staff.clefGlyph = "clefs.F"
\set Staff.clefPosition = 2
c'1
\set Staff.clefGlyph = "clefs.G"
c'1
\set Staff.clefGlyph = "clefs.C"
c'1
\set Staff.clefTransposition = 7
c'1
\set Staff.clefTransposition = 0
\set Staff.clefPosition = 0
c'1

% Return to the normal clef.
\set Staff.middleCPosition = 0
c'1
}

```



Veränderung des Layouts von Verzierungen innerhalb der Noten

Das Layout von Verzierungsausdrücken kann in der Musik verändert werden mit den Funktionen `add-grace-property` und `remove-grace-property`. Das folgende Beispiel definiert die Richtung von Hälzen (Stem) für diese Verzierung, sodass die Hälze nicht immer nach unten zeigen, und ändert den Standardnotenkopf in ein Kreuz.

```
\relative c' {
  \new Staff {
    $(remove-grace-property 'Voice 'Stem 'direction)
    $(add-grace-property 'Voice 'NoteHead 'style 'cross)
    \new Voice {
      \acciaccatura { f16 } g4
      \grace { d16 e } f4
      \appoggiatura { f,32 g a } e2
    }
  }
}
```



Using alternative flag styles

Alternative shapes for flags on eighth and shorter notes can be displayed by overriding the `stencil` property of `Flag`. LilyPond provides the following functions: `modern-straight-flag`, `old-straight-flag`, and `flat-flag`. Use `\revert` to restore the default shape.

To get stacked (i.e., vertically more compact) flags, call the command `\flagStyleStacked`, which can be reset with `\flagStyleDefault`.

Overriding the `Flag` stencil does not change how flag elements are positioned vertically. This is especially noticeable for flat flags: LilyPond doesn't dynamically adjust the vertical gaps between flag elements in the same way as it does for beams. A possible solution to harmonize the appearance is to replace flat flags with half beams, as shown in the second staff; however, this can't be done automatically. In the code of this snippet, such half beams are entered with `@` as a prefix, for example `@c8`.

Be aware that half beams are *not* `Flag` grobs. This means in particular that modifying `Flag` properties won't have any effect on them (you have to use `Beam` properties instead), and properties for their associated `Stem` grob will also behave beam-like.

```
"@" =
#(define-music-function (music) (ly:music?)
  #{ \set stemLeftBeamCount = 0 $music [] #})

testnotes = {
```

```

\autoBeamOff
c8 d16 e''32 f64 \acciaccatura { g,,,8 } a128 b
}

\relative c' {
  \override TextScript.staff-padding = 6
  \time 1/4
  <>^"default" \testnotes
  \override Flag.stencil = #modern-straight-flag
  <>_"modern straight" \testnotes
  \override Flag.stencil = #old-straight-flag
  <>^"old straight" \testnotes
  \override Flag.stencil = #flat-flag
  <>_"flat" \testnotes
  \revert Flag.stencil

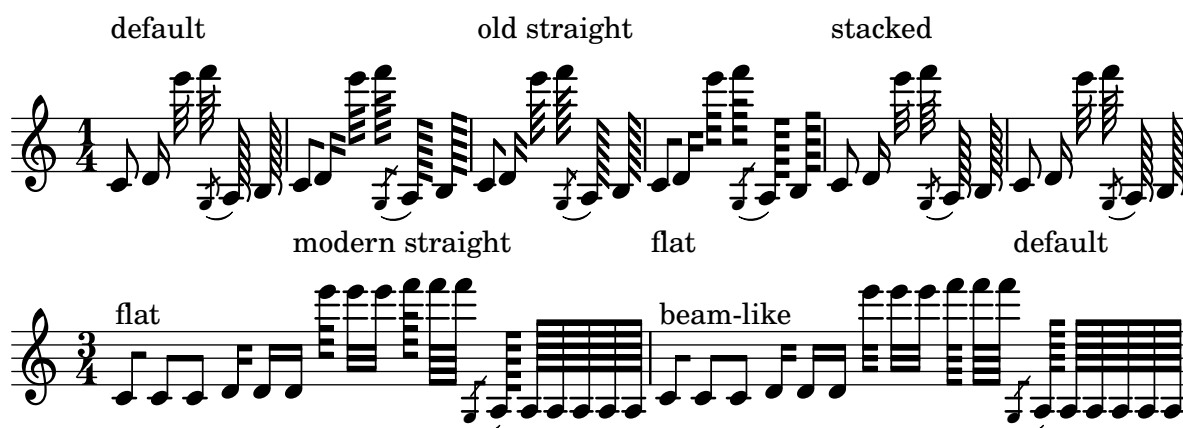
  \flagStyleStacked
  <>^"stacked" \testnotes
  \flagStyleDefault
  <>_"default" \testnotes
}

\relative c' {
  \time 3/4
  \override Flag.stencil = #flat-flag

  <>^"flat" c8 c[ c] d16 d[ d] e''32 e[ e] f64 f[ f]
  \acciaccatura { g,,,8 } a128 a[ a a a a]
  <>^"beam-like" @c8 c[ c] @d16 d[ d] @e''32 e[ e] @f64 f[ f]
  \acciaccatura { g,,,8 } @a128 a[ a a a a]
}

\layout {
  indent = 0
  \context {
    \Score
    \override NonMusicalPaperColumn.line-break-permission = ##f
  }
}

```



Using ly:grob-object to access grobs with \tweak

Some grobs can be accessed „laterally“ from within another grob’s callback. These are usually listed as „layout objects“ in the „Internal properties“ section of a grob interface. The function ly:grob-object is used to access these grobs.

Demonstrated below are some ways of accessing grobs from within a NoteHead callback, but the technique is not limited to NoteHeads. However, the NoteHead callback is particularly important, since it is the implicit callback used by the \tweak command.

The console output of the example function below (display-grobs) is as follows.

```
-----
#<Grob Accidental >
()
#<Grob Stem >
```

It is probably not that useful, but it demonstrates that the grobs are indeed being accessed.

```
#(define (notehead-get-accidental notehead)
  ;; notehead is grob
  (ly:grob-object notehead 'accidental-grob))

#(define (notehead-get-arpeggio notehead)
  ;; notehead is grob
  (let ((notecolumn (notehead-get-notecolumn notehead)))
    (ly:grob-object notecolumn 'arpeggio)))

#(define (notehead-get-notecolumn notehead)
  ;; notehead is grob
  (ly:grob-parent notehead X))

#(define (notehead-get-stem notehead)
  ;; notehead is grob
  (let ((notecolumn (notehead-get-notecolumn notehead)))
    (ly:grob-object notecolumn 'stem)))

#(define (display-grobs notehead)
  ;; notehead is grob
  (let ((accidental (notehead-get-accidental notehead))
        (arpeggio (notehead-get-arpeggio notehead))
        (stem (notehead-get-stem notehead)))
    (format (current-error-port) "~2&~a\n" (make-string 20 #\ -))
    (for-each
     (lambda (x) (format (current-error-port) "~a\n" x))
     (list accidental arpeggio stem))))

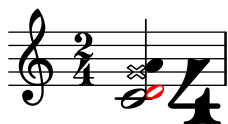
\relative c' {
  %% display grobs for each note head:
  \%override NoteHead.before-line-breaking = #display-grobs
  <c
  %% or just for one:
  \tweak before-line-breaking #display-grobs
  es
  g>1\arpeggio
}
```



Using the `\tweak` command to tweak individual grobs

With the `\tweak` command, every grob can be tuned directly. Here are some examples of available tweaks.

```
\relative c' {
  \time 2/4
  \set fingeringOrientations = #'(right)
  <
    \tweak font-size #3 c
    \tweak color #red d-\tweak font-size #8 -4
    \tweak style #'cross g
    \tweak duration-log #2 a
  >2
}
```



Vertikale Ausrichtung von Dynamik und Textbeschriftung beeinflussen

Indem man die `'Y-extent`-Eigenschaft auf einen passenden Wert setzt, können alle `DynamicLineSpanner`-Objekte (Crescendo-Klammern und Dynamik-Texte) (hairpins and dynamic texts) unabhängig von ihrer wirklichen Ausdehnung an einem gemeinsamen Referenzpunkt ausgerichtet werden. Auf diese Weise ist jedes Element vertikal ausgerichtet und der Notensatz sieht ansprechender aus.

Die gleiche Idee wird benutzt, um Textbeschriftungen an ihrer Grundlinie auszurichten.

```
music = \relative c' {
  a'2\p b\f
  e4\p f\f\> g, b\p
  c2^\markup { \huge gorgeous } c^\markup { \huge fantastic }
}

{
  \music
  \break
  \override DynamicLineSpanner.staff-padding = 3
  \textLengthOn
  \override TextScript.staff-padding = 1
  \music
}
```





Gesangstext und Ossia vertikal ausrichten

Dieser Schnipsel zeigt, wie man die Kontexteigenschaften `alignBelowContext` und `alignAboveContext` benutzen kann, um die Positionierung von Gesangstext und Ossia-Abschnitten zu kontrollieren.

```
\relative c' <<
  \new Staff = "1" { c4 c c c }
  \new Staff = "2" { d4 d d d }
  \new Staff = "3" { e4 e e e }

  { \skip 2
    <<
      \lyrics {
        \set alignBelowContext = "1"
        lyrics4 below
      }
      \new Staff \with {
        alignAboveContext = "3"
        fontSize = -2
        \override StaffSymbol.staff-space = #(magstep -2)
        \remove "Time_signature_engraver"
        \override VerticalAxisGroup.staff-staff-spacing =
          #'((minimum-distance . 0)
            (basic-distance . 0)
            (padding . 1))
      } {
        \tuplet 6/4 {
          \override TextScript.padding = 2
          c8[^"ossia above" d e d e f]
        }
      }
    }
  }
>>
```



Vertically aligning stanza numbers of different staves

It can happen that stanza numbers don't align vertically if the verses are attached to different staves. To fix that, override the `self-alignment-X` property of the `LyricText` grob.

```
\markup { default behavior }

<<
  \new Staff { b b b b }
  \lyrics {
    \set stanza = "3."
    a a a a
  }

  \new Staff { b b b b }
  \lyrics {
    \set stanza = "1."
    aaaaaaaaaa a a a
  }
  \lyrics {
    \set stanza = "2."
    a a a a
  }
>>

\markup \vspace #1
\markup {
  using \typewriter "self-alignment-X = #LEFT" }

<<
  \new Staff { b b b b }
  \new Lyrics \lyricmode {
    \set stanza = "3."
    a a a a
  }

  \new Staff { b b b b }
  \new Lyrics \lyricmode {
    \set stanza = "1."
    \once \override LyricText.self-alignment-X = #LEFT
    aaaaaaaaaa a a a
  }
  \new Lyrics \lyricmode {
    \set stanza = "2."
    a a a a
  }
>>
```

default behavior



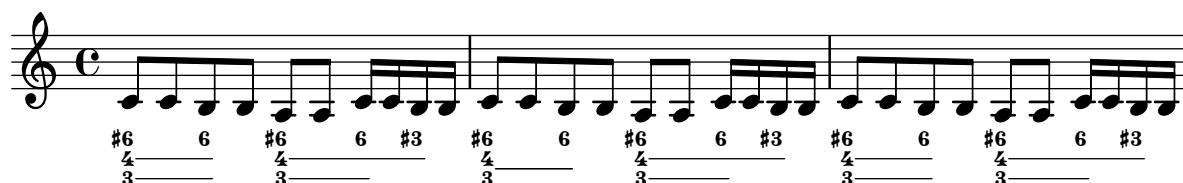
using self-alignment-X = #LEFT



Vertically centering paired figured bass extenders

Where figured bass extender lines are being used by setting `useBassFigureExtenders` to `#t`, pairs of congruent figured bass extender lines are vertically centered if `figuredBassCenterContinuations` is set to `#t`.

```
<<
\relative c' {
  \repeat unfold 3 {
    c8 c b b a a c16 c b b
  }
}
\figures {
  \set useBassFigureExtenders = ##t
  <6+ 4 3>4 <6 4 3>8 r
  <6+ 4 3>4 <6 4 3>8 <4 3+>16 r
  \set figuredBassCenterContinuations = ##t
  <6+ 4 3>4 <6 4 3>8 r
  <6+ 4 3>4 <6 4 3>8 <4 3+>16 r
  \set figuredBassCenterContinuations = ##f
  <6+ 4 3>4 <6 4 3>8 r
  <6+ 4 3>4 <6 4 3>8 <4 3+>16 r
}
>>
```



20 Paper and layout

See also Abschnitt “Spacing issues” in *Notationsreferenz*.

Aligning and centering instrument names

The horizontal alignment of instrument names is tweaked by changing the `self-alignment-X` property of the `InstrumentName` grob (usually in the `Staff` context). The `\layout` variables `indent` and `short-indent` define the space in which the instrument names are aligned before the first and the following systems, respectively.

```
\paper {
  left-margin = 3\cm
}

\new StaffGroup <<
  \new Staff \with {
    \override InstrumentName.self-alignment-X = #LEFT
    instrumentName = \markup \left-column { "Left aligned"
                                           "instrument name" }

    shortInstrumentName = "Left"
  } {
    c'1 \break c'1
  }

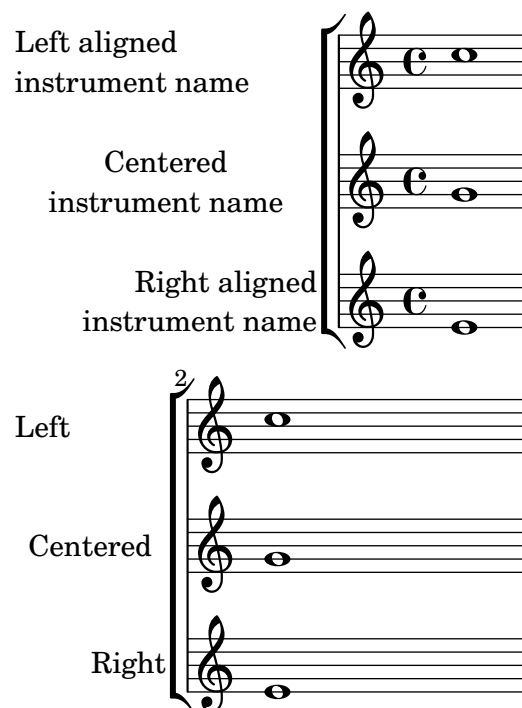
  \new Staff \with {
    \override InstrumentName.self-alignment-X = #CENTER
    instrumentName = \markup \center-column { Centered
                                           "instrument name" }

    shortInstrumentName = "Centered"
  } {
    g'1 g'1
  }

  \new Staff \with {
    \override InstrumentName.self-alignment-X = #RIGHT
    instrumentName = \markup \right-column { "Right aligned"
                                           "instrument name" }

    shortInstrumentName = "Right"
  } {
    e'1 e'1
  }
}
>>

\layout {
  indent = 4\cm
  short-indent = 2\cm
  line-width = 6.5\cm
}
```



Arranging separate lyrics on a single line

Sometimes you may want to put lyrics for different performers on a single line: where there is rapidly alternating text, for example. This snippet shows how it can be done with adjusting the `nonstaff-nonstaff-spacing` property of the `VerticalAxisGroup` grob.

```
\layout {
  \context {
    \Lyrics
    \override VerticalAxisGroup
      .nonstaff-nonstaff-spacing
      .minimum-distance = ##f
  }
}

aliceSings = \markup { \smallCaps "Alice" }
eveSings = \markup { \smallCaps "Eve" }

<<
\new Staff <<
  \new Voice = "alice" {
    f'4^\aliceSings g' r2 |
    s1 |
    f'4^\aliceSings g' r2 |
    s1 | \break
    % ...

    \voiceOne
    s2 a'8^\aliceSings a' b'4 |
    \oneVoice
    g'1
  }
}
```

```

\new Voice = "eve" {
  s1 |
  a'2^\eveSings g' |
  s1 |
  a'2^\eveSings g'
  % ...

  \voiceTwo
  f'4^\eveSings a'8 g' f'4 e' |
  \oneVoice
  s1
}
>>

\new Lyrics \lyricsto "alice" {
  may -- be
  sec -- ond
  % ...
  Shut up, you fool!
}

\new Lyrics \lyricsto "eve" {
  that the
  words are
  % ...
  ...and then I was like--
}
>>

```

The musical score is written on four staves. The first staff contains the lyrics 'may - be' (Alice) and 'that the' (Eve). The second staff contains 'sec - ond' (Alice) and 'words are' (Eve). The third staff contains '...and then I' (Eve) and 'Shut up, you like--' (Alice). The fourth staff contains 'was' (Alice) and 'fool!' (Eve). The lyrics are written below the staves, and the names 'ALICE' and 'EVE' are written above the staves to indicate which character is singing.

Book parts

`\bookpart` can be used to split a book into several parts. Each part last page can be affected by `ragged-last-bottom`. Header and footer markups can detect a part's last page to differentiate with the book's last page.

```
#(set-default-paper-size "a6")
```

```

\book {
  %% book paper, which is inherited by all children bookparts
  \paper {
    ragged-last-bottom = ##t
    %% Page footer: add a different part-tagline at part last page

```

```

oddFooterMarkup = \markup {
  \column {
    \fill-line {
      %% Copyright header field only on book first page.
      \if \on-first-page \fromproperty #'header:copyright
    }
    \fill-line {
      %% Part tagline header field only on each part last page.
      \if \on-last-page-of-part \fromproperty #'header:parttagline
    }
    \fill-line {
      %% Tagline header field only on book last page.
      \if \on-last-page \fromproperty #'header:tagline
    }
  }
}

%% book header, which is inherited by the first bookpart
\header {
  title = "Book title"
  copyright = "Copyright line on book first page"
  parttagline = "Part tagline"
  tagline = "Book tagline"
}

\bookpart {
  %% a different page breaking function may be used on each part
  \paper { page-breaking = #ly:minimal-breaking }
  \header { subtitle = "First part" }
  \markup { The first book part }
  \markup { a page break }
  \pageBreak
  \markup { first part last page }
  \markup \wordwrap {
    with ragged-last-bottom (see the space below this text) }
}

\bookpart {
  \header { subtitle = "Second part" }
  { c'4 }
}

```

Book title

First part

The first book part

a page break

Copyright line on book first page

2

first part last page
with ragged-last-bottom (see the space below this
text)

Part tagline

3

Book title

Second part



Part tagline
Book tagline

Changing the staff size

The simplest way to resize staves is to use

```
 #(set-global-staff-size size)
```

To resize an individual staff's size, you can use the properties `staff-space` and `fontSize`.

```
<<
\new Staff \relative c'' {
  \dynamicDown c8\ff c c c c c c c
}
\new Staff \with {
  fontSize = #-3
  \override StaffSymbol.staff-space = #(magstep -3)
} \relative c {
  \clef bass c8 c c c c\ff c c c
}
>>
```




This code shows how to clip (extract) snippets from a full score.

If system starts and ends are included, they include extents of the System grob, e.g., instrument names.

Grace notes at the end point of the region are not included.

Regions can span multiple systems. In this case, multiple EPS files are generated.

```

#(set-default-paper-size "a6" 'landscape)
\layout {
  indent = 2.4\cm
}

#(ly:set-option 'clip-systems)
#(ly:set-option 'separate-page-formats "ps")
#(define output-suffix "1")

origScore = \new Staff \with { instrumentName = "Instrument" }
\relative c' {
  c1
  d1
  \grace c16 e1
  \key d \major
  f1 \break
  \clef bass
  g,1
  fis1
}

\book {
  \score {
    \origScore
    \layout {
      % Each clip-region is a (START . END) pair
      % where both are rhythmic locations. Syntax:
      %
      % (make-rhythmic-locations BAR-NUMBER NUM DEN)
      %
      % means NUM/DEN whole-notes into bar numbered BAR-NUMBER

      clip-regions = #(list (cons (make-rhythmic-location 2 0 1)
                                (make-rhythmic-location 4 0 1))
                           (cons (make-rhythmic-location 0 0 1)
                                (make-rhythmic-location 2 0 1)))
    }
  }
}

```

```

        (make-rhythmic-location 4 0 1))
      (cons (make-rhythmic-location 0 0 1)
            (make-rhythmic-location 6 0 1)))
    }
  }
}

#(ly:set-option 'clip-systems #f)
#(ly:set-option 'separate-page-formats #f)
#(define output-suffix #f)

\book {
  \score { \origScore }
  \markup { \bold \fontsize #6 clips }
  \score {
    \lyrics {
      "from-2.0.1-to-4.0.1-clip.eps"
      \markup \epsfile #X #30.0
      #(format #f "~a-1-from-2.0.1-to-4.0.1-clip.eps"
        (ly:parser-output-name))
    }
  }
}

```



LilyPond v2.25.34

Creating blank staves

To create blank staves, generate empty measures then remove the `Bar_number_engraver` from the `Score` context, and the `Time_signature_engraver`, `Clef_engraver` and `Bar_engraver` from the `Staff` context.

```

#(set-global-staff-size 10) % for the documentation
% #(set-global-staff-size 20) % for letter and A4

```

```

\book {
  \score {
    { \repeat unfold 12 { s1 \break } }

    \layout {
      indent = 0
      \context {
        \Staff
        \remove "Time_signature_engraver"
        \remove "Clef_engraver"
        \remove "Bar_engraver"
      }
      \context {
        \Score
        \remove "Bar_number_engraver"
      }
    }
  }

  % for the documentation
  \paper {
    #(set-paper-size "a6")
    ragged-last-bottom = ##f
    line-width = 90\mm
    left-margin = 7.5\mm
    bottom-margin = 5\mm
    top-margin = 5\mm
    tagline = ##f
  }

  % uncomment these lines for "letter" size
  %{
  \paper {
    #(set-paper-size "letter")
    ragged-last-bottom = ##f
    line-width = 7.5\in
    left-margin = 0.5\in
    bottom-margin = 0.25\in
    top-margin = 0.25\in
    tagline = ##f
  }
  %}

  % uncomment these lines for "A4" size
  %{
  \paper {
    #(set-paper-size "a4")
    ragged-last-bottom = ##f
    line-width = 180\mm
    left-margin = 15\mm
    bottom-margin = 10\mm
    top-margin = 10\mm
  }
  %}

```



```

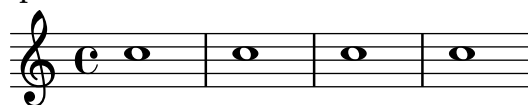
    meter = "meter"
    opus = "opus"
    piece = "piece"
    poet = "poet"
    copyright = "copyright"
    tagline = "tagline"
}

\bookpart {
  \score {
    \relative c'' { c1 | c | c | c }

    \header {
      title = "localtitle"
      subtitle = "localsubtitle"
      composer = "localcomposer"
      arranger = "localarranger"
      instrument = "localinstrument"
      meter = "localmeter"
      opus = "localopus"
      piece = "localpiece"
      poet = "localpoet"
      copyright = "localcopyright"
      tagline = "localtagline"
    }
  }
}

```

	title	
	subtitle	
poet	instrument	composer
meter		arranger
	localtitle	
	localsubtitle	
localpoet	localinstrument	localcomposer
localmeter		localarranger
localpiece		localopus



copyright
tagline

Displaying a whole GrandStaff system if only one of its staves is alive

In many orchestral scores it is custom to not show staves for instruments that are silent for a while; this is called a ‚Frenched‘ score. LilyPond provides this functionality via the `\RemoveEmptyStaves` command.

When they play again it is often preferred to show the staves of *all instruments of such a group*. This can be done by adding the `Keep_alive_together_engraver` to the grouping context (e.g., `GrandStaff` or `StaffGroup`).

In the example below the violins are silent in the second system. Only the first violin plays the last measure in the third system but the staff of the second violin is also displayed.

```
\score {
  <<
    \new Staff = "Staff_flute" \with {
      instrumentName = "Flute"
      shortInstrumentName = "Fl"
    } \relative c' {
      \repeat unfold 3 { c'4 c c c | c c c c | c c c c | \break }
    }

    \new StaffGroup = "StaffGroup_Strings" <<
      \new GrandStaff = "GrandStaff_violins" <<
        \new Staff = "StaffViolinI" \with {
          instrumentName = "Violin I"
          shortInstrumentName = "Vi I"
        } \relative c'' {
          a1 | R1*7 | \repeat unfold 12 a16 a4 |
        }
        \new Staff = "StaffViolinII" \with {
          instrumentName = "Violin II"
          shortInstrumentName = "Vi II"
        } \relative c' {
          e1 | R1*8 |
        }
      >>
    >>

    \new Staff = "Staff_cello" \with {
      instrumentName = "Cello"
      shortInstrumentName = "Ce"
    } \relative c {
      \clef bass \repeat unfold 9 { c1 } |
    }
  >>
}

\layout {
  indent = 3.0\cm
  short-indent = 1.5\cm

  \context {
    \GrandStaff
    \consists Keep_alive_together_engraver
```

```

}
\context {
  \Staff
  \RemoveEmptyStaves
}
}

```

The image displays three musical systems, each with a brace on the left side. The first system includes Flute, Violin I, Violin II, and Cello. The second system includes Flute and Cello, with a system separator (4) between them. The third system includes Flute, Violin I, Violin II, and Cello, with a system separator (7) between them. The Violin I staff in the third system shows a repeat sign.

Setting system separators

System separators can be inserted between systems. Any markup can be used, but `\slashSeparator` has been provided as a sensible default.

```
#(set-default-paper-size "a5")
```

```

\paper {
  system-separator-markup = \slashSeparator
  tagline = ##f
}

```

```

notes = \relative c' {
  c1 | c \break
  c1 | c \break
  c1 | c
}

\book {
  \score {
    \new GrandStaff <<
      \new Staff \notes
      \new Staff \notes
    >>
  }
}

```

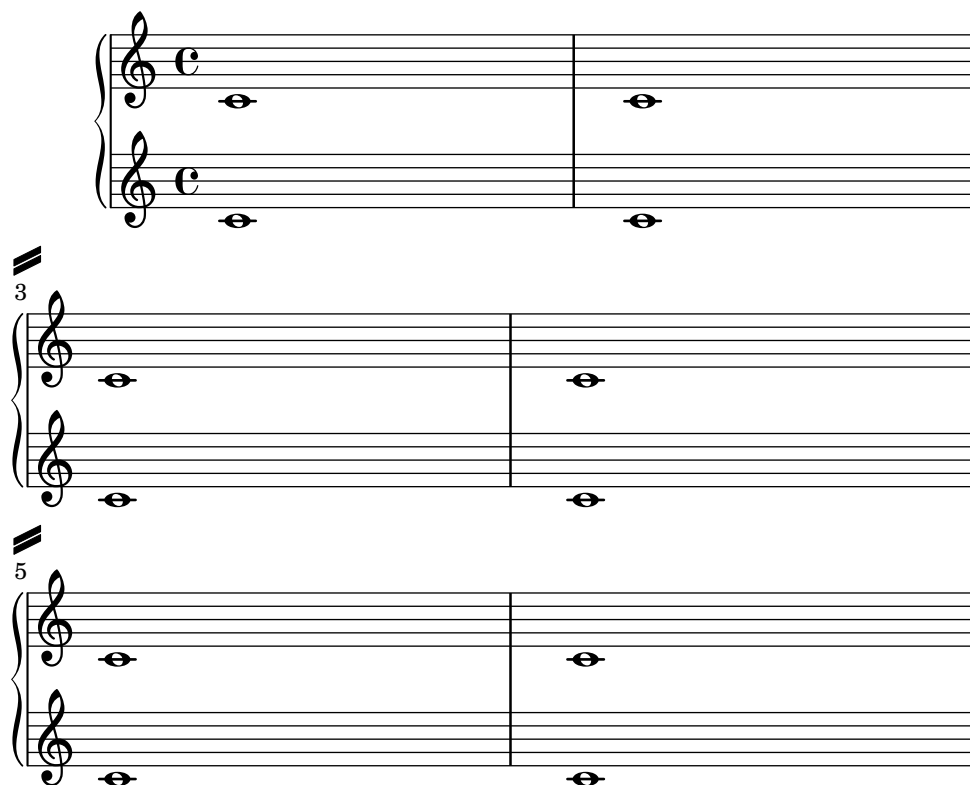


Table of contents

A table of contents is included using `\markuplist \table-of-contents`; its items are added with the `\tocItem` command.

```

#(set-default-paper-size "a7" 'landscape)
#(set-global-staff-size 11)

```

```

\paper {
  print-all-headers = ##t
}

\book {
  \markuplist \table-of-contents

```



```

\pageBreak

\tocItem \markup { The first score }
\score {
  {
    c'1 \pageBreak
    \mark \default \tocItem \markup { Mark A }
    d'1
  }
  \header { title = "First score" }
}
\pageBreak

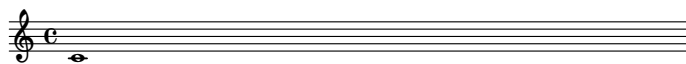
\tocItem \markup { The second score }
\score {
  { e'1 }
  \header { title = "Second score" }
}
}

```

Table of Contents

The first score	2
Mark A	3
The second score	4

2

First score

3



4

Second score

Vertically aligned StaffGroups without connecting SystemStartBar

This snippet shows how to achieve vertically aligned StaffGroups with a SystemStartBar for each StaffGroup, but without connecting them.

Note that this only works properly for music that can be printed as a single system.

```

#(set-global-staff-size 15)

\paper {
  ragged-right = ##f
  print-all-headers = ##t
  tagline = ##f
}

\layout {
  indent = 0

  \context {
    \StaffGroup
    \consists Text_mark_engraver
    \consists Staff_collecting_engraver
    systemStartDelimiterHierarchy =
      #'(SystemStartBrace (SystemStartBracket a b))
  }

  \context {
    \Score
    \remove Text_mark_engraver
    \remove Staff_collecting_engraver
    \override SystemStartBrace.style = #'bar-line
    \omit SystemStartBar
    \override SystemStartBrace.padding = #-0.1
    \override SystemStartBrace.thickness = #1.6
    \override StaffGrouper.staffgroup-staff-spacing.basic-distance = #15
  }
}

%%% EXAMPLE

txt =
\lyricmode {
  Wer4 nur den lie -- ben Gott läßt wal2 -- ten4
  und4 hof -- fet auf ihn al -- le Zeit2.
}

% First StaffGroup "exercise"

eI = \relative c' {
  \textMark \markup {
    \bold Teacher:
    This is a simple setting of the choral. Please improve it. }
  \key a \minor
  \time 4/4

```

```

\voiceOne

\partial 4 e4
a b c b
a b gis2
e4\fermata g! g f
e a a gis
a2.\fermata
\bar " : | ."
}

eII = \relative c' {
  \key a \minor
  \time 4/4
  \voiceTwo
  \partial 4 c4
  e e e gis
  a f e2
  b4 b d d
  c c d d
  c2.
  \bar " : | ."
}

eIII = \relative c' {
  \key a \minor
  \time 4/4
  \clef bass
  \voiceOne

  \partial 4 a4
  c b a b
  c d b2
  gis4 g g b
  c a f e
  e2.
}

eIV = \relative c' {
  \key a \minor
  \time 4/4
  \clef bass
  \voiceTwo

  \partial 4 a,4
  a' gis a e
  a, d e2
  e,4\fermata e' b g
  c f d e
  a,2.\fermata
  \bar " : | ."
}

```

```

exercise = \new StaffGroup = "exercise" <<
  \new Staff <<
    \new Voice \eI
    \new Voice \eII
  >>

  \new Lyrics \txt

  \new Staff <<
    \new Voice \eIII
    \new Voice \eIV
  >>
>>

% Second StaffGroup "simple Bach"

sbI = \relative c' {
  \textMark \markup { \bold" Pupil:" Here's my version! }
  \key a \minor
  \time 4/4
  \voiceOne

  \partial 4 e4
  a b c b
  a b gis2
  e4\fermata g! g f
  e a a gis
  a2.\fermata
  \bar ":|."
}

sbII = \relative c' {
  \key a \minor
  \time 4/4
  \voiceTwo
  \partial 4 c8 d
  e4 e e8 f g4
  f f e2
  b4 b8 c d4 d
  e8 d c4 b8 c d4
  c2.
  \bar ":|."
}

sbIII = \relative c' {
  \key a \minor
  \time 4/4
  \clef bass
  \voiceOne

```

```

\partial 4 a8 b
c4 b a b8 c
d4 d8 c b2
gis4 g g8 a b4
b a8 g f4 e
e2.
}

sbIV = \relative c' {
  \key a \minor
  \time 4/4
  \clef bass
  \voiceTwo

  \partial 4 a,4
  a' gis a e
  f8 e d4 e2
  e,4\fermata e' b a8 g
  c4 f8 e d4 e
  a,2.\fermata
  \bar ":|."
}

simpleBach = \new StaffGroup = "simple Bach" <<
  \new Staff <<
    \new Voice \sbI
    \new Voice \sbII
  >>

  \new Lyrics \txt

  \new Staff <<
    \new Voice \sbIII
    \new Voice \sbIV
  >>
>>

% Third StaffGroup "chromatic Bach"

cbI = \relative c' {
  \textMark \markup {
    \bold "Teacher:"
    \column {
      "Well, you simply copied and transposed a version of J.S.Bach."
      "Do you know this one?"
    }
  }
}
\key a \minor
\time 4/4
\voiceOne

```

```

\partial 4 e4
a b c b
a b gis4. fis8
e4\fermata g! g f
e a a8 b gis4
a2.\fermata
\bar " : | ."
}

cbII = \relative c' {
  \key a \minor
  \time 4/4
  \voiceTwo

  \partial 4 c8 d
  e4 e e8 fis gis4
  a8 g! f!4 e2
  b4 e e d
  d8[ cis] d dis e fis e4
  e2.
  \bar " : | ."
}

cbIII = \relative c' {
  \key a \minor
  \time 4/4
  \clef bass
  \voiceOne

  \partial 4 a8 b
  c[ b] a gis8 a4 d,
  e8[ e'] d c b4. a8
  gis4 b c d8 c
  b[ a] a b c b b c16 d
  c2.
}

cbIV = \relative c' {
  \key a \minor
  \time 4/4
  \clef bass
  \voiceTwo

  \partial 4 a4
  c, e a, b
  c d e2
  e4\fermata e a b8 c
  gis[ g] fis f e dis e4
  a,2.\fermata
  \bar " : | ."
}

```

```

chromaticBach = \new StaffGroup = "chromatic Bach" <<
  \new Staff <<
    \new Voice \cbI
    \new Voice \cbII
  >>

  \new Lyrics \txt

  \new Staff <<
    \new Voice \cbIII
    \new Voice \cbIV
  >>
>>

% Score

\score {
  <<
    \exercise
    \simpleBach
    \chromaticBach
  >>

  \header {
    title = \markup \column {
      \combine \null \vspace #1
      "Exercise: Improve the given choral"
      " "
    }
  }

  \layout {
    \context {
      \Lyrics
      \override LyricText.X-offset = #-1
    }
  }
}

```

Exercise: Improve the given choral

Teacher: This is a simple setting of the choral. Please improve it.

Wer nur den lie - ben Gott läßt wal - ten und hof-fet auf ihn al - le Zeit

Pupil: Here's my version!

Wer nur den lie - ben Gott läßt wal - ten und hof-fet auf ihn al - le Zeit

Teacher: Well, you simply copied and transposed a version of J.S.Bach.
Do you know this one?

Wer nur den lie - ben Gott läßt wal - ten und hof-fet auf ihn al - le Zeit

21 Titles

See also Abschnitt “Titles and headers” in *Notationsreferenz*.

Adding the current date to a score

With a little Scheme code, the current date can easily be added to a score.

```
\paper { tagline = ##f }

% first, define a variable to hold the formatted date:
date = #(strftime "%d-%m-%Y" (localtime (current-time)))

% use it in the title block:
\header {
  title = "Including the date!"
  subtitle = \date
}

\score {
  \relative c' {
    c4 c c c
  }
}

% and use it in a \markup block:
\markup {
  \date
}
```

Including the date!

21-02-2026



21-02-2026

Aligning and centering instrument names

The horizontal alignment of instrument names is tweaked by changing the `self-alignment-X` property of the `InstrumentName` grob (usually in the `Staff` context). The `\layout` variables `indent` and `short-indent` define the space in which the instrument names are aligned before the first and the following systems, respectively.

```
\paper {
  left-margin = 3\cm
}

\new StaffGroup <<
  \new Staff \with {
    \override InstrumentName.self-alignment-X = #LEFT
    instrumentName = \markup \left-column { "Left aligned"
                                              "instrument name" }
    shortInstrumentName = "Left"
```

```

} {
  c''1 \break c''1
}

\new Staff \with {
  \override InstrumentName.self-alignment-X = #CENTER
  instrumentName = \markup \center-column { Centered
                                         "instrument name" }

  shortInstrumentName = "Centered"
} {
  g'1 g'1
}

\new Staff \with {
  \override InstrumentName.self-alignment-X = #RIGHT
  instrumentName = \markup \right-column { "Right aligned"
                                         "instrument name" }

  shortInstrumentName = "Right"
} {
  e'1 e'1
}
>>


\layout {
  indent = 4\cm
  short-indent = 2\cm
  line-width = 6.5\cm
}

```

Left aligned
instrument name

Centered
instrument name

Right aligned
instrument name



Left

Centered

Right



Demonstrating all \header fields

A demonstration of all header fields that LilyPond defines by default. Thanks to setting `print-all-headers` to `#t`, much more fields as usual are displayed, indicating the hierarchy of `\header` blocks.


```
\paper {
  #(set-paper-size "a6" 'landscape)
  print-all-headers = ##t
}

\book {
  \header {
    title = "title"
    subtitle = "subtitle"
    composer = "composer"
    arranger = "arranger"
    instrument = "instrument"
    meter = "meter"
    opus = "opus"
    piece = "piece"
    poet = "poet"
    copyright = "copyright"
    tagline = "tagline"
  }

  \bookpart {
    \score {
      \relative c'' { c1 | c | c | c }

      \header {
        title = "localtitle"
        subtitle = "localsubtitle"
        composer = "localcomposer"
        arranger = "localarranger"
        instrument = "localinstrument"
        meter = "localmeter"
        opus = "localopus"
        piece = "localpiece"
        poet = "localpoet"
        copyright = "localcopyright"
        tagline = "localtagline"
      }
    }
  }
}
```

	title	
	subtitle	
poet	instrument	composer
meter		arranger
	localtitle	
	localsubtitle	
localpoet	localinstrument	localcomposer
localmeter		localarranger
localpiece		localopus



	copyright
	tagline

Outputting the version number

It is possible to print the version number of LilyPond in markup.

```
\markup { Processed with LilyPond version #(lilypond-version) }
```

Processed with LilyPond version 2.25.34

Breaking horizontal alignment of dynamics and textscripts

LilyPond uses `DynamicLineSpanner` grobs to horizontally align successive dynamic objects like hairpins and dynamic text, even if they are positioned on different sides of a staff. This connection cannot be broken, contrary to the vertical alignment (see snippet „Breaking vertical alignment of dynamics and textscripts“).

There are two solutions to circumvent the problem.

- Modify the `shorten-pair` property of the `Hairpin` grob to compensate the offset by which the hairpin was moved.
- Put the two dynamic objects into different voices.

Both solutions are demonstrated in this snippet.

```
{
  <>^"default"
  f' _\pp ^\> f' f' f'\!
}

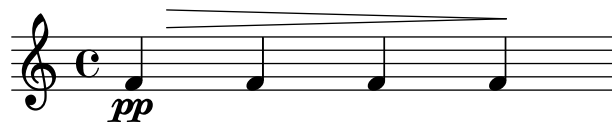
{
  <>^\markup { setting \typewriter shorten-pair }
  f' _\pp \tweak shorten-pair #'(-3 . 0) ^\> f' f' f'\!
}

{
  <>^\markup { using another \typewriter Voice context }
  << { f' ^\> f' f' f'\! }
  \new Voice { s4 _\pp } >>
}

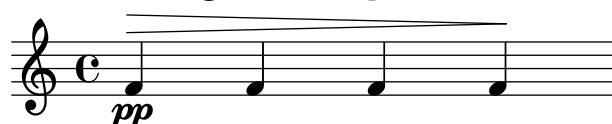
\layout {
  line-width = 8\cm
  ragged-right = ##f

  \context {
    \Voice
    \override TextScript.staff-padding = #3.5
  }
}
```


default



setting shorten-pair



using another Voice context



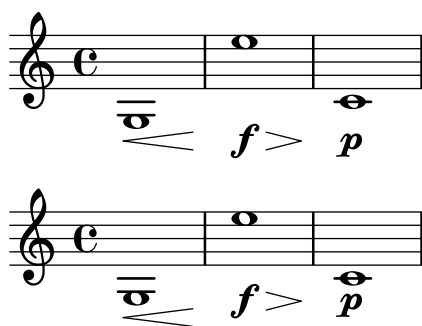
Breaking vertical alignment of dynamics and textscripts

By default, LilyPond uses `DynamicLineSpanner` grobs to vertically align successive dynamic objects like hairpins and dynamic text. However, this is not always wanted. By inserting `\breakDynamicSpan`, which ends the alignment spanner prematurely, this vertical alignment can be avoided.

See also snippet „Breaking horizontal alignment of dynamics and textscripts“.

```
{ g1\< |
  e''\f\> |
  c'\p }
```

```
{ g1\< |
  e''\breakDynamicSpan\f\> |
  c'\p }
```



Harmonizing bar line thickness for staves with different sizes

When using `\magnifyStaff` only for some staves in a `StaffGroup`, `BarLine` grobs do not align any more due to its changed properties `thick-thickness`, `hair-thickness`, and `kern`.

To fix this, multiple workarounds are available, as demonstrated below.

```
\markuplist {
  % First row.
  \fill-line {
    \score {
      \new StaffGroup <<
        \new Staff \with { \magnifyStaff #1/2 } {
          \textMark \markup \tiny "default"
          b1 b \bar "|."
        }
        \new Staff { b b }
      >>
    }
    \score {
      \new StaffGroup <<
        \new Staff \with { \magnifyStaff #1/2 } {
          \textMark \markup \tiny \column { "reverting only the"
                                          "final bar line" }
          b1 b
          \revert Staff.BarLine.thick-thickness
          \revert Staff.BarLine.hair-thickness
          \revert Staff.BarLine.kern
          \bar "|."
        }
      >>
    }
  }
}
```

```

    }
    \new Staff { b b }
  >>
}
\score {
  \new StaffGroup <<
    \new Staff \with { \magnifyStaff #1/2
      #(revert-props 'magnifyStaff 0
        '((BarLine thick-thickness)
          (BarLine hair-thickness)
          (BarLine kern))) } {
      \textMark \markup \tiny \column { "cancelling"
        \typewriter "\magnifyStaff"
        "only for bar lines" }

      b1 b \bar "|."
    }
    \new Staff { b b }
  >>
}
}

\vspace #2

% Second row.
\fill-line {
  \score {
    \new StaffGroup <<
      \new Staff \with { \magnifyStaff #1/2 } {
        \textMark \markup \tiny \column { "mimicking"
          \typewriter "\magnifyStaff"
          "on the other staves" }

        b1 b \bar "|." }
      \new Staff \with { #(scale-props 'magnifyStaff 1/2 #t
        '((BarLine thick-thickness)
          (BarLine hair-thickness)
          (BarLine kern))) } {

        b b }
    >>
  }
  \score {
    \new StaffGroup <<
      \new Staff \with { \magnifyStaff #1/2
        #(scale-props 'magnifyStaff 3/2 #t
          '((BarLine thick-thickness)
            (BarLine hair-thickness)
            (BarLine kern))) } {
        \textMark \markup \tiny \column { "applying an"
          "intermediate"
          "value to all staves" }

        b1 b \bar "|." }
      \new Staff \with { #(scale-props 'magnifyStaff 3/4 #t
        '((BarLine thick-thickness)

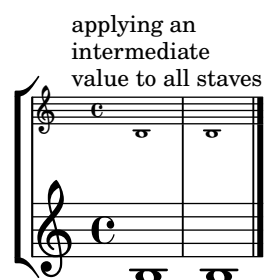
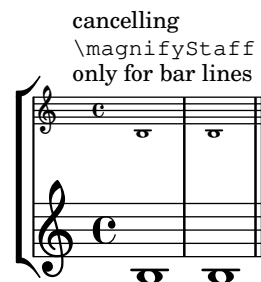
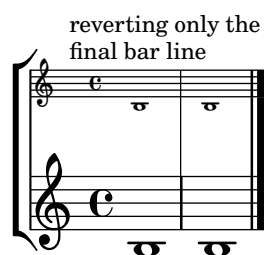
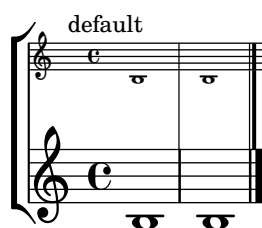
```



```

        (BarLine hair-thickness)
        (BarLine kern))) } {
    b b }
    >>
  }
  ""
}
}

```



Page label

Page labels may be placed inside music or at top-level, and referred to in markups.

```

#(set-default-paper-size "a7" 'landscape)
#(set-global-staff-size 11)

```

```

\label license
\markup \fill-line {
  \center-column {
    "This snippet is available"
    "under the Creative Commons"
    "Public Domain Dedication license." } }

```

```

{
  \repeat volta 2 {
    \label startRepeat
    \repeat unfold 22 { c'2 2 }
    \pageBreak
    \repeat unfold 16 { c'2 2 }
  }
  \textEndMark \markup {
    \with-link #'startRepeat \line {
      To page \page-ref #'startRepeat "0" "?"
    }
  }
}

```

```

    }
  }
}

\markup \fill-line {
  \line {
    See page \page-ref #'license "0" "?" for
    licensing information. } }

```

This snippet is available
under the Creative Commons
Public Domain Dedication license.



See page ? for licensing information.

Proportional strict notespacing

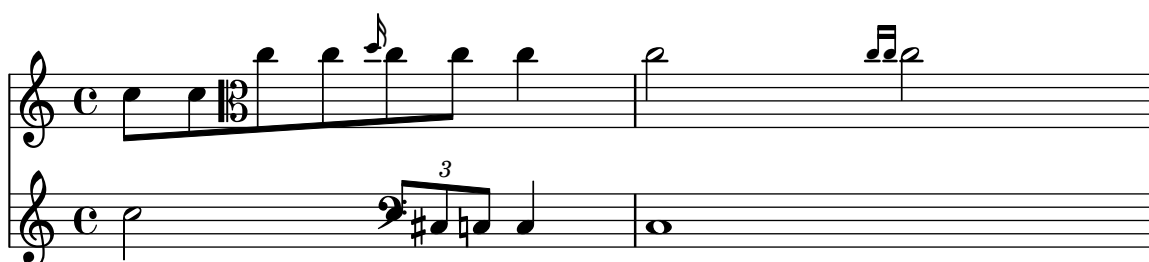
If the `strict-note-spacing` property of the `SpacingSpanner` grob is set to `#t`, spacing of notes is not influenced by bars or clefs within a system. Rather, they are placed just before the note that occurs at the same time. This may cause collisions.

```

\relative c' ' <<
  \override Score.SpacingSpanner.strict-note-spacing = ##t
  \set Score.proportionalNotationDuration = #1/16

  \new Staff {
    c8[ c \clef alto c c \grace { d16 } c8 c] c4
    c2 \grace { c16[ c16] } c2
  }
  \new Staff {
    c2 \tuplet 3/2 { c8 \clef bass cis,, c } c4
    c1
  }
>>

```



Vertikale Ausrichtung von Dynamik und Textbeschriftung beeinflussen

Indem man die 'Y-extent-Eigenschaft auf einen passenden Wert setzt, können alle `DynamicLineSpanner`-Objekte (Crescendo-Klammern und Dynamik-Texte) (hairpins and dynamic texts) unabhängig von ihrer wirklichen Ausdehnung an einem gemeinsamen Referenzpunkt ausgerichtet werden. Auf diese Weise ist jedes Element vertikal ausgerichtet und der Notensatz sieht ansprechender aus.

Die gleiche Idee wird benutzt, um Textbeschriftungen an ihrer Grundlinie auszurichten.

```
music = \relative c' {
  a'2\p b\f
  e4\p f\f\> g, b\p
  c2^\markup { \huge gorgeous } c^\markup { \huge fantastic }
}

{
  \music
  \break
  \override DynamicLineSpanner.staff-padding = 3
  \textLengthOn
  \override TextScript.staff-padding = 1
  \music
}
```



Gesangstext und Ossia vertikal ausrichten

Dieser Schnipsel zeigt, wie man die Kontexteigenschaften `alignBelowContext` und `alignAboveContext` benutzen kann, um die Positionierung von Gesangstext und Ossia-Abschnitten zu kontrollieren.

```
\relative c' <<
  \new Staff = "1" { c4 c c c }
  \new Staff = "2" { d4 d d d }
  \new Staff = "3" { e4 e e e }

  { \skip 2
    <<
      \lyrics {
        \set alignBelowContext = "1"
        lyrics4 below
      }
      \new Staff \with {
        alignAboveContext = "3"
        fontSize = -2
      }
```

```

\override StaffSymbol.staff-space = #(magstep -2)
\remove "Time_signature_engraver"
\override VerticalAxisGroup.staff-staff-spacing =
  #'((minimum-distance . 0)
     (basic-distance . 0)
     (padding . 1))
} {
  \tuplet 6/4 {
    \override TextScript.padding = 2
    c8["^"ossia above" d e d e f]
  }
}
>>
}
>>

```

The image displays a musical score with three staves. The top staff contains five quarter notes. The middle staff contains five quarter notes, with the text "lyrics below" positioned centrally beneath the notes. The bottom staff contains five quarter notes. A sixth staff, positioned between the middle and bottom staves, shows a tuplet of six eighth notes, with the text "ossia above" centered above the notes.

23 MIDI

See also Abschnitt “Creating MIDI output” in *Notationsreferenz*.

Changing MIDI output to one channel per voice

When outputting MIDI, the default behavior is for each staff to represent one MIDI channel, with all the voices on a staff amalgamated. This minimizes the risk of running out of MIDI channels, since there are only 16 available per track.

However, by moving the `Staff_performer` to the `Voice` context, each voice on a staff can have its own MIDI channel, as is demonstrated by the following example: despite being on the same staff, two MIDI channels are created, each with a different `midiInstrument`.

```
\score {
  \new Staff <<
    \new Voice \relative c''' {
      \set midiInstrument = "flute"
      \voiceOne
      \key g \major
      \time 2/2
      r2 g-"Flute" ~
      g fis ~
      fis4 g8 fis e2 ~
      e4 d8 cis d2
    }
    \new Voice \relative c'' {
      \set midiInstrument = "clarinet"
      \voiceTwo
      b1-"Clarinet"
      a2. b8 a
      g2. fis8 e
      fis2 r
    }
  >>
  \layout { }
  \midi {
    \context {
      \Staff
      \remove "Staff_performer"
    }
    \context {
      \Voice
      \consists "Staff_performer"
    }
  }
  \tempo 2 = 72
}
```



Das Tempo ohne Metronom-Angabe verändern

Um das Tempo für die MIDI-Ausgabe zu ändern, ohne eine Tempoangabe in den Noten auszugeben, kann die Metronombezeichnung unsichtbar gemacht werden:

```
\score {
  \new Staff \relative c' {
    \tempo 4 = 160
    c4 e g b
    c4 b d c
    \set Score.tempoHideNote = ##t
    \tempo 4 = 96
    d,4 fis a cis
    d4 cis e d
  }
  \layout { }
  \midi { }
}
```



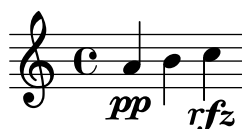
Creating custom dynamics in MIDI output

The following example shows how to create a dynamic marking, not included in the default list, and assign a specific value to it so that it affects MIDI output.

The dynamic mark `\rfz` gets value 0.9.

```
#(define (myDynamics dynamic)
  (if (equal? dynamic "rfz")
      0.9
      (default-dynamic-absolute-volume dynamic)))

\score {
  \new Staff {
    \set Staff.midiInstrument = "cello"
    \set Score.dynamicAbsoluteVolumeFunction = #myDynamics
    \new Voice {
      \relative {
        a'4\pp b c-\rfz
      }
    }
  }
  \layout {}
  \midi {}
}
```



Demo of MIDI instruments

Problem: How to know which `midiInstrument` values would be best for your composition?

Solution: A LilyPond demo file. You have to compile this snippet by yourself and listen to the created MIDI output file.

```
melody = \relative c' {
  \tempo 4 = 150
  c4.\mf g c16 b' c d
  e16 d e f g4 g'4 r
  R1
}

\score {
  \new Voice \melody
  \layout { }
}

\score {
  \new Voice {
    r\mf
    % 1-8 keyboard
    \set Staff.midiInstrument = "acoustic grand" \melody
    \set Staff.midiInstrument = "bright acoustic" \melody
    \set Staff.midiInstrument = "electric grand" \melody
    \set Staff.midiInstrument = "honky-tonk" \melody
    \set Staff.midiInstrument = "electric piano 1" \melody
    \set Staff.midiInstrument = "electric piano 2" \melody
    \set Staff.midiInstrument = "harpsichord" \melody
    \set Staff.midiInstrument = "clav" \melody

    % 9-16 chrom percussion
    \set Staff.midiInstrument = "celesta" \melody
    \set Staff.midiInstrument = "glockenspiel" \melody
    \set Staff.midiInstrument = "music box" \melody
    \set Staff.midiInstrument = "vibraphone" \melody
    \set Staff.midiInstrument = "marimba" \melody
    \set Staff.midiInstrument = "xylophone" \melody
    \set Staff.midiInstrument = "tubular bells" \melody
    \set Staff.midiInstrument = "dulcimer" \melody

    % 17-24 organ
    \set Staff.midiInstrument = "drawbar organ" \melody
    \set Staff.midiInstrument = "percussive organ" \melody
    \set Staff.midiInstrument = "rock organ" \melody
    \set Staff.midiInstrument = "church organ" \melody
    \set Staff.midiInstrument = "reed organ" \melody
    \set Staff.midiInstrument = "accordion" \melody
    \set Staff.midiInstrument = "harmonica" \melody
    \set Staff.midiInstrument = "concertina" \melody

    % 25-32 guitar
    \set Staff.midiInstrument = "acoustic guitar (nylon)" \melody
```

```

\set Staff.midiInstrument = "acoustic guitar (steel)" \melody
\set Staff.midiInstrument = "electric guitar (jazz)" \melody
\set Staff.midiInstrument = "electric guitar (clean)" \melody
\set Staff.midiInstrument = "electric guitar (muted)" \melody
\set Staff.midiInstrument = "overdriven guitar" \melody
\set Staff.midiInstrument = "distorted guitar" \melody
\set Staff.midiInstrument = "guitar harmonics" \melody

```

% 33-40 bass

```

\set Staff.midiInstrument = "acoustic bass" \melody
\set Staff.midiInstrument = "electric bass (finger)" \melody
\set Staff.midiInstrument = "electric bass (pick)" \melody
\set Staff.midiInstrument = "fretless bass" \melody
\set Staff.midiInstrument = "slap bass 1" \melody
\set Staff.midiInstrument = "slap bass 2" \melody
\set Staff.midiInstrument = "synth bass 1" \melody
\set Staff.midiInstrument = "synth bass 2" \melody

```

% 41-48 strings

```

\set Staff.midiInstrument = "violin" \melody
\set Staff.midiInstrument = "viola" \melody
\set Staff.midiInstrument = "cello" \melody
\set Staff.midiInstrument = "contrabass" \melody
\set Staff.midiInstrument = "tremolo strings" \melody
\set Staff.midiInstrument = "pizzicato strings" \melody
\set Staff.midiInstrument = "orchestral harp" \melody
\set Staff.midiInstrument = "timpani" \melody

```

% 49-56 ensemble

```

\set Staff.midiInstrument = "string ensemble 1" \melody
\set Staff.midiInstrument = "string ensemble 2" \melody
\set Staff.midiInstrument = "synthstrings 1" \melody
\set Staff.midiInstrument = "synthstrings 2" \melody
\set Staff.midiInstrument = "choir aahs" \melody
\set Staff.midiInstrument = "voice oohs" \melody
\set Staff.midiInstrument = "synth voice" \melody
\set Staff.midiInstrument = "orchestra hit" \melody

```

% 57-64 brass

```

\set Staff.midiInstrument = "trumpet" \melody
\set Staff.midiInstrument = "trombone" \melody
\set Staff.midiInstrument = "tuba" \melody
\set Staff.midiInstrument = "muted trumpet" \melody
\set Staff.midiInstrument = "french horn" \melody
\set Staff.midiInstrument = "brass section" \melody
\set Staff.midiInstrument = "synthbrass 1" \melody
\set Staff.midiInstrument = "synthbrass 2" \melody

```

% 65-72 reed

```

\set Staff.midiInstrument = "soprano sax" \melody
\set Staff.midiInstrument = "alto sax" \melody
\set Staff.midiInstrument = "tenor sax" \melody

```



```

\set Staff.midiInstrument = "baritone sax" \melody
\set Staff.midiInstrument = "oboe" \melody
\set Staff.midiInstrument = "english horn" \melody
\set Staff.midiInstrument = "bassoon" \melody
\set Staff.midiInstrument = "clarinet" \melody

% 73-80 pipe
\set Staff.midiInstrument = "piccolo" \melody
\set Staff.midiInstrument = "flute" \melody
\set Staff.midiInstrument = "recorder" \melody
\set Staff.midiInstrument = "pan flute" \melody
\set Staff.midiInstrument = "blown bottle" \melody
\set Staff.midiInstrument = "shakuhachi" \melody
\set Staff.midiInstrument = "whistle" \melody
\set Staff.midiInstrument = "ocarina" \melody

% 81-88 synth lead
\set Staff.midiInstrument = "lead 1 (square)" \melody
\set Staff.midiInstrument = "lead 2 (sawtooth)" \melody
\set Staff.midiInstrument = "lead 3 (calliope)" \melody
\set Staff.midiInstrument = "lead 4 (chiff)" \melody
\set Staff.midiInstrument = "lead 5 (charang)" \melody
\set Staff.midiInstrument = "lead 6 (voice)" \melody
\set Staff.midiInstrument = "lead 7 (fifths)" \melody
\set Staff.midiInstrument = "lead 8 (bass+lead)" \melody

% 89-96 synth pad
\set Staff.midiInstrument = "pad 1 (new age)" \melody
\set Staff.midiInstrument = "pad 2 (warm)" \melody
\set Staff.midiInstrument = "pad 3 (polysynth)" \melody
\set Staff.midiInstrument = "pad 4 (choir)" \melody
\set Staff.midiInstrument = "pad 5 (bowed)" \melody
\set Staff.midiInstrument = "pad 6 (metallic)" \melody
\set Staff.midiInstrument = "pad 7 (halo)" \melody
\set Staff.midiInstrument = "pad 8 (sweep)" \melody

% 97-104 synth effects
\set Staff.midiInstrument = "fx 1 (rain)" \melody
\set Staff.midiInstrument = "fx 2 (soundtrack)" \melody
\set Staff.midiInstrument = "fx 3 (crystal)" \melody
\set Staff.midiInstrument = "fx 4 (atmosphere)" \melody
\set Staff.midiInstrument = "fx 5 (brightness)" \melody
\set Staff.midiInstrument = "fx 6 (goblins)" \melody
\set Staff.midiInstrument = "fx 7 (echoes)" \melody
\set Staff.midiInstrument = "fx 8 (sci-fi)" \melody

% 105-112 ethnic
\set Staff.midiInstrument = "sitar" \melody
\set Staff.midiInstrument = "banjo" \melody
\set Staff.midiInstrument = "shamisen" \melody
\set Staff.midiInstrument = "koto" \melody
\set Staff.midiInstrument = "kalimba" \melody

```

```

\set Staff.midiInstrument = "bagpipe" \melody
\set Staff.midiInstrument = "fiddle" \melody
\set Staff.midiInstrument = "shanai" \melody

% 113-120 percussive
\set Staff.midiInstrument = "tinkle bell" \melody
\set Staff.midiInstrument = "agogo" \melody
\set Staff.midiInstrument = "steel drums" \melody
\set Staff.midiInstrument = "woodblock" \melody
\set Staff.midiInstrument = "taiko drum" \melody
\set Staff.midiInstrument = "melodic tom" \melody
\set Staff.midiInstrument = "synth drum" \melody
\set Staff.midiInstrument = "reverse cymbal" \melody

% 121-128 sound effects
\set Staff.midiInstrument = "guitar fret noise" \melody
\set Staff.midiInstrument = "breath noise" \melody
\set Staff.midiInstrument = "seashore" \melody
\set Staff.midiInstrument = "bird tweet" \melody
\set Staff.midiInstrument = "telephone ring" \melody
\set Staff.midiInstrument = "helicopter" \melody
\set Staff.midiInstrument = "applause" \melody
\set Staff.midiInstrument = "gunshot" \melody
}
\midi { }
}

```



Replacing default MIDI instrument equalization

The default MIDI instrument equalizer can be replaced by setting the `instrumentEqualizer` property in the Score context to a user-defined Scheme procedure that uses a MIDI instrument name as its argument along with a pair of fractions indicating the minimum and maximum volumes, respectively, to be applied to that specific instrument.

The following example sets the minimum and maximum volumes for flute and clarinet.

```

#(define my-instrument-equalizer-alist '())

#(set! my-instrument-equalizer-alist
  (append
    '(("flute" . (0.7 . 0.9))
      ("clarinet" . (0.3 . 0.6)))
    my-instrument-equalizer-alist))

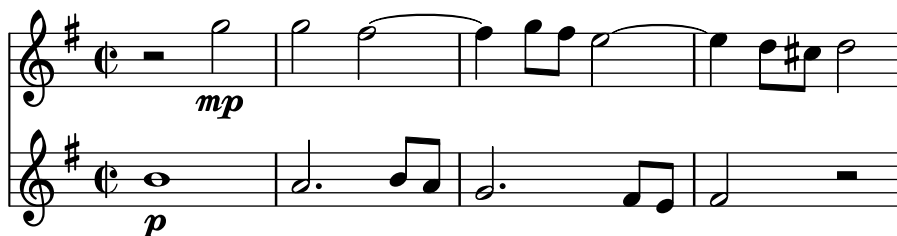
#(define (my-instrument-equalizer s)
  (let ((entry (assoc s my-instrument-equalizer-alist)))
    (if entry
      (cdr entry))))

```

```

\score {
  <<
    \new Staff {
      \key g \major
      \time 2/2
      \set Score.instrumentEqualizer = #my-instrument-equalizer
      \set Staff.midiInstrument = "flute"
      \new Voice \relative {
        r2 g''\mp g fis~
        4 g8 fis e2~
        4 d8 cis d2
      }
    }
    \new Staff {
      \key g \major
      \set Staff.midiInstrument = "clarinet"
      \new Voice \relative {
        b'1\p a2. b8 a
        g2. fis8 e
        fis2 r
      }
    }
  >>
  \layout { }
  \midi { }
}

```



24 Templates

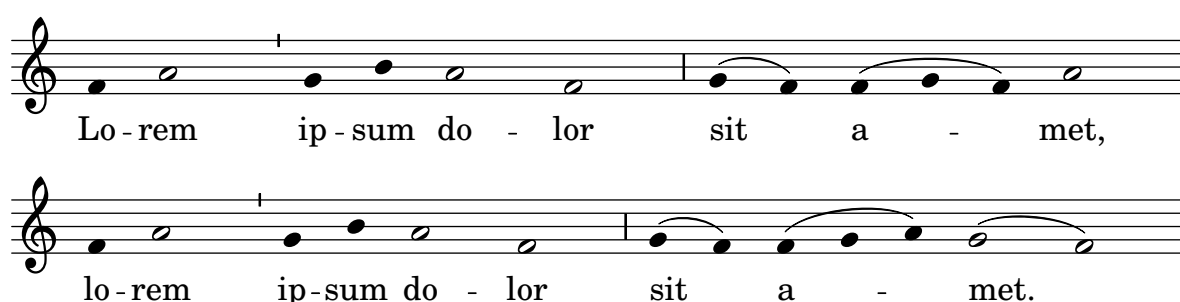
Ancient notation template – modern transcription of Gregorian music

This example demonstrates how to do modern transcription of Gregorian music. Gregorian music has no measure, no stems; it uses only half and quarter note heads, and special marks, indicating rests of different length.

```
chant = \relative c' {
  \set Score.timing = ##f
  f4 a2 \divisioMinima
  g4 b a2 f2 \divisioMaior
  g4( f) f( g f) a2 \finalis \break
  f4 a2 \divisioMinima
  g4 b a2 f2 \divisioMaior
  g4( f) f( g a) g2( f) \finalis
}

verba = \lyricmode {
  Lo -- rem ip -- sum do -- lor sit a -- met,
  lo -- rem ip -- sum do -- lor sit a -- met.
}

\score {
  \new GregorianTranscriptionStaff <<
    \new GregorianTranscriptionVoice = "melody" \chant
    \new GregorianTranscriptionLyrics = "one" \lyricsto melody \verba
  >>
}
```



Vorlage für anglikanischen Psalm

Diese Vorlage zeigt eine Art, anglikanische Psalmengesänge zu setzen. Hier wird auch gezeigt, wie Strophen als einfacher Text unter den Noten hinzugefügt werden können. Zwei Strophen sind in unterschiedlicher Weise notiert um mehr Möglichkeiten darzustellen.

```
SopranoMusic = \relative g' {
  g1 | c2 b | a1 | \bar "||"
  a1 | d2 c | c b | c1 | \bar "||"
}

AltoMusic = \relative c' {
  e1 | g2 g | f1 |
```

```

    f1 | f2 e | d d | e1 |
}

TenorMusic = \relative a {
    c1 | c2 c | c1 |
    d1 | g,2 g | g g | g1 |
}

BassMusic = \relative c {
    c1 | e2 e | f1 |
    d1 | b2 c | g' g | c,1 |
}

global = {
    \time 2/2
}

dot = \markup {
    \raise #0.7 \musicglyph "dots.dot"
}

tick = \markup {
    \raise #1 \fontsize #-5 \musicglyph "scripts.rvarcomma"
}

% Use markup to center the chant on the page
\markup \fill-line {
    \score { % centered
        \new ChoirStaff <<
            \new Staff <<
                \global
                \clef "treble"
                \new Voice = "Soprano" <<
                    \voiceOne
                    \SopranoMusic
                >>
                \new Voice = "Alto" <<
                    \voiceTwo
                    \AltoMusic
                >>
            >>

            \new Staff <<
                \clef "bass"
                \global
                \new Voice = "Tenor" <<
                    \voiceOne
                    \TenorMusic
                >>
                \new Voice = "Bass" <<
                    \voiceTwo
                    \BassMusic
            >>
        >>
    }
}

```

```

>>
>>
>>

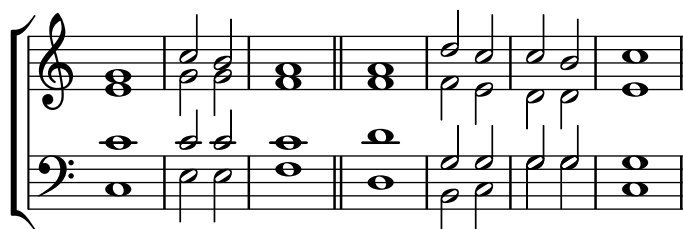
\layout {
  \context {
    \Score
    \override SpacingSpanner.base-shortest-duration =
      \musicLength 2
  }
  \context {
    \Staff
    \remove "Time_signature_engraver"
  }
}
} % End score
} % End markup

\markup \fill-line {
  \left-column {
    \null \null \null
    \line { \fontsize #5 0
      \fontsize #3 come
      let us \bold sing | unto \dot the | Lord : let }
    \line { us heartily \concat { re \bold joyce }
      in the | strength of | our }
    \line { sal | vation. }

    \null

    \line { \hspace #2.5 8. Today if ye will hear his voice * }
    \line { \concat { \bold hard en }
      \tick not your \tick hearts : as in the pro- }
    \line { vocation * and as in the \bold day of tempt- \tick }
    \line { -ation \tick in the \tick wilderness. }
  }
}

```



O come let us **sing** | unto • the | Lord : let
us heartily **rejoice** in the | strength of | our
sal | vation.

8. Today if ye will hear his voice *
harden ' not your ' hearts : as in the pro-
vocation * and as in the **day** of tempt- '
-ation ' in the ' wilderness.

Hymnus-Vorlage

Dieses Beispiel zeigt eine Möglichkeit, eine Hymnusmelodie zu setzen, in der jede Zeile mit einem Auftakt beginnt und einem unvollständigen Takt abschließt. Es zeigt auch, wie man die Strophen als allein stehenden Text unter die Noten hinzufügt.

```
Timeline = {
  \time 4/4
  \tempo 4=96
  \partial 2
  s2 | s1 | s2 \breathe s2 | s1 | s2 \caesura \break
  s2 | s1 | s2 \breathe s2 | s1 | s2 \fine
}

SopranoMusic = \relative g' {
  g4 g | g g g g | g g g g | g g g g | g2
  g4 g | g g g g | g g g g | g g g g | g2
}

AltoMusic = \relative c' {
  d4 d | d d d d | d d d d | d d d d | d2
  d4 d | d d d d | d d d d | d d d d | d2
}

TenorMusic = \relative a {
  b4 b | b b b b | b b b b | b b b b | b2
  b4 b | b b b b | b b b b | b b b b | b2
}

BassMusic = \relative g {
  g4 g | g g g g | g g g g | g g g g | g2
  g4 g | g g g g | g g g g | g g g g | g2
}

global = {
  \key g \major
}

\score { % Start score
  \new PianoStaff << % Start pianostaff
```

```

\new Staff << % Start Staff = RH
  \global
  \clef "treble"
  \new Voice = "Soprano" << % Start Voice = "Soprano"
    \Timeline
    \voiceOne
    \SopranoMusic
  >> % End Voice = "Soprano"
  \new Voice = "Alto" << % Start Voice = "Alto"
    \Timeline
    \voiceTwo
    \AltoMusic
  >> % End Voice = "Alto"
>> % End Staff = RH

\new Staff << % Start Staff = LH
  \global
  \clef "bass"
  \new Voice = "Tenor" << % Start Voice = "Tenor"
    \Timeline
    \voiceOne
    \TenorMusic
  >> % End Voice = "Tenor"
  \new Voice = "Bass" << % Start Voice = "Bass"
    \Timeline
    \voiceTwo
    \BassMusic
  >> % End Voice = "Bass"
>> % End Staff = LH
>> % End pianostaff
} % End score

\markup \fill-line {
  \left-column {
    "This is line one of the first verse"
    "This is line two of the same"
  }
  \null
  "And here's line one of the second verse"
  "And the next line of the same"
}

\layout {
  \context {
    \Score
    caesuraType = #'((bar-line . "||"))
    fineBarType = "||"
  }
}

\paper { % Start paper block
  indent = 0 % don't indent first system

```



```

line-width = 130    % shorten line length to suit music
tagline = ##f       % Don't print tag line, can be removed
} % End paper block

```



This is line one of the first verse
 This is line two of the same

And here's line one of the second verse
 And the next line of the same

Vorlage für Jazz-Combo

Hier ist ein ziemlich kompliziertes Beispiel für ein Jazz-Ensemble. Achtung: Alle Instrumente sind in `\key c \major` (C-Dur) notiert. Das bezieht sich auf die klingende Musik: LilyPond transponiert die Tonart automatisch, wenn sich die Noten innerhalb eines `ranspose`-Abschnitts befinden.

```

\header {
  title = "Song"
  subtitle = "(tune)"
  composer = "Me"
  meter = "moderato"
  piece = "Swing"
  tagline = \markup \column {
    "LilyPond example file by Amelie Zapf,"
    "Berlin 07/07/2003" }
}

% To make the example display properly in the documentation.
\paper {
  paper-width = 130\mm
  paper-height = 205\mm
}

% #(set-global-staff-size 16)

\include "english.ly"

```

```
%%%%%%%%%%%%%% Some macros %%%%%%%%%%%%%%%
```

```
sl = { \override NoteHead.style = #'slash
       \hide Stem }
nsl = { \revert NoteHead.style
        \undo \hide Stem }
crOn = \override NoteHead.style = #'cross
crOff = \revert NoteHead.style
```

```
% Insert chord name style stuff here.
```

```
jazzChords = { }
```

```
%%%%%%%%%%%%%% Keys 'n' thangs %%%%%%%%%%%%%%%
```

```
global = { \time 4/4 }
```

```
Key = { \key c \major }
```

```
% ##### Horns #####
```

```
% ----- Trumpet -----
```

```
trpt = \transpose c d \relative c' {
  \Key
  c1 | c | c |
}
trpHarmony = \transpose c' d {
  \jazzChords
}
trumpet = {
  \global
  \clef treble
  \trpt
}
```

```
% ----- Alto Saxophone -----
```

```
alto = \transpose c a \relative c' {
  \Key
  c1 | c | c |
}
altoHarmony = \transpose c' a {
  \jazzChords
}
altoSax = {
  \global
  \clef treble
  \alto
}
```

```
% ----- Baritone Saxophone -----
```

```
bari = \transpose c a' \relative c {
```

```

    \Key
    c1 | c1 |
    \sl d4^"Solo" d d d \nsl |
}
bariHarmony = \transpose c' a \chordmode {
    \jazzChords
    s1 | s |
    d2:maj e:m7 |
}
bariSax = {
    \global
    \clef treble
    \bari
}

% ----- Trombone -----
tbone = \relative c {
    \Key
    c1 | c | c |
}
tboneHarmony = \chordmode {
    \jazzChords
}
trombone = {
    \global
    \clef bass
    \tbone
}

% ##### Rhythm Section #####

% ----- Guitar -----
gtr = \relative c'' {
    \Key
    c1 |
    \sl b4 b b b \nsl |
    c1 |
}
gtrHarmony = \chordmode {
    \jazzChords
    s1 | c2:min7+ d2:maj9 | s1 |
}
guitar = {
    \global
    \clef treble
    \gtr
}

%% ----- Piano -----
rhUpper = \relative c'' {
    \voiceOne
    \Key

```

```

    c1 | c | c |
}
rhLower = \relative c' {
    \voiceTwo
    \Key
    e1 | e | e |
}

lhUpper = \relative c' {
    \voiceOne
    \Key
    g1 | g | g |
}
lhLower = \relative c {
    \voiceTwo
    \Key
    c1 | c | c |
}

PianoRH = {
    \clef treble
    \global
    <<
        \new Voice = "one" \rhUpper
        \new Voice = "two" \rhLower
    >>
}
PianoLH = {
    \clef bass
    \global
    <<
        \new Voice = "one" \lhUpper
        \new Voice = "two" \lhLower
    >>
}

piano = <<
    \new Staff = "upper" \PianoRH
    \new Staff = "lower" \PianoLH
>>

% ----- Bass Guitar -----
Bass = \relative c {
    \Key
    c1 | c | c |
}
bass = {
    \global
    \clef bass
    \Bass
}

```

```

% ----- Drums -----
up = \drummode {
  \voiceOne
  hh4 <hh sn> hh <hh sn> |
  hh4 <hh sn> hh <hh sn> |
  hh4 <hh sn> hh <hh sn> |
}
down = \drummode {
  \voiceTwo
  bd4 s bd s |
  bd4 s bd s |
  bd4 s bd s |
}

drumContents = {
  \global
  <<
    \new DrumVoice \up
    \new DrumVoice \down
  >>
}

%%%%%%%%%% It All Goes Together Here %%%%%%%%%%%

\book { % For the LilyPond documentation.
  \score {
    <<
      \new StaffGroup = "horns" <<
        \new Staff = "trumpet" \with { instrumentName = "Trumpet" }
        \trumpet
        \new Staff = "altosax" \with { instrumentName = "Alto Sax" }
        \altoSax
        \new ChordNames = "barichords" \with { instrumentName = "Bari Sax" }
        \bariHarmony
        \new Staff = "barisax" \with { instrumentName = "Bari Sax" }
        \bariSax
        \new Staff = "trombone" \with { instrumentName = "Trombone" }
        \trombone
      >>

      \new StaffGroup = "rhythm" <<
        \new ChordNames = "chords" \with { instrumentName = "Guitar" }
        \gtrHarmony
        \new Staff = "guitar" \with { instrumentName = "Guitar" }
        \guitar
        \new PianoStaff = "piano" \with {
          instrumentName = "Piano"
          midiInstrument = "acoustic grand"
        } \piano
        \new Staff = "bass" \with { instrumentName = "Bass" }
        \bass
    >>
  }
}

```

```
\new DrumStaff \with { instrumentName = "Drums" }
  \drumContents
>>
>>

\layout {
  \context {
    \Staff
    \RemoveEmptyStaves
  }
  \context {
    \Score
    \override BarNumber.padding = 3
    \override RehearsalMark.padding = 2
    skipBars = ##t
  }
}
\midi { }
}
```

Song (tune)

Me

moderato
Swing

Trumpet

Alto Sax

Bari Sax

Trombone

Guitar

Piano

Bass

Drums

Cm^Δ D^Δ 9

B^Δ C#m⁷
Solo

LilyPond example file by Amelie Zapf,
Berlin 07/07/2003

Orchester, Chor und Klavier

Diese Vorlage zeigt die Benutzung von geschachtelten `StaffGroup`- und `GrandStaff`-Kontexte, um Instrumente in Untergruppen zu unterteilen, und die Benutzung von `\transpose` für transponierende Instrumente. Alle Noten werden in C geschrieben. Noten können in C eingegeben werden, oder auch in der Tonart des Instrumentes: dann müssen sie zuerst nach C transponiert werden, bevor sie einer Variable zugewiesen werden.

```
#(set-global-staff-size 17)
```

```
\paper {
  indent = 3.0\cm % add space for instrumentName
  short-indent = 1.5\cm % add less space for shortInstrumentName
```

```

}

fluteMusic = \relative c' { \key g \major g'1 b }

% Pitches as written on a manuscript for Clarinet in A
% are transposed to concert pitch.
clarinetMusic = \transpose c' a
  \relative c'' { \key bes \major bes1 d }

trumpetMusic = \relative c { \key g \major g''1 b }

% Key signature is often omitted for horns
hornMusic = \transpose c' f
  \relative c { d'1 fis }

percussionMusic = \relative c { \key g \major g1 b }

sopranoMusic = \relative c'' { \key g \major g'1 b }
sopranoLyrics = \lyricmode { Lyr -- ics }

altoIMusic = \relative c' { \key g \major g'1 b }
altoILyrics = \sopranoLyrics
altoIIMusic = \relative c' { \key g \major g'1 b }
altoIILyrics = \lyricmode { Ah -- ah }

tenorMusic = \relative c' { \clef "treble_8" \key g \major g1 b }
tenorLyrics = \sopranoLyrics

pianoRHMus = \relative c { \key g \major g''1 b }
pianoLHMus = \relative c { \clef bass \key g \major g1 b }

violinIMusic = \relative c' { \key g \major g'1 b }
violinIIMusic = \relative c' { \key g \major g'1 b }

violaMusic = \relative c { \clef alto \key g \major g'1 b }

celloMusic = \relative c { \clef bass \key g \major g1 b }

bassMusic = \relative c { \clef "bass_8" \key g \major g,1 b }

\book {
  \score {
    <<
    \new StaffGroup = "StaffGroup_woodwinds" <<
      \new Staff = "Staff_flute" \with { instrumentName = "Flute" }
        \fluteMusic

      \new Staff = "Staff_clarinet" \with {
        instrumentName = \markup { \concat { "Clarinet in B" \flat } }
      }

      % Declare that written Middle C in the music
      % to follow sounds a concert B flat, for

```



```

% output using sounded pitches such as MIDI.
%\transposition bes

% Print music for a B-flat clarinet
\transpose bes c' \clarinetMusic
>>

\new StaffGroup = "StaffGroup_brass" <<
  \new Staff = "Staff_hornI" \with {
    instrumentName = "Horn in F"
  }
  % \transposition f
  \transpose f c' \hornMusic

  \new Staff = "Staff_trumpet" \with {
    instrumentName = "Trumpet in C"
  }
  \trumpetMusic
>>

\new RhythmicStaff = "RhythmicStaff_percussion" \with {
  instrumentName = "Percussion"
}
\percussionMusic

\new PianoStaff \with {
  instrumentName = "Piano"
} <<
  \new Staff { \pianoRHMusik }
  \new Staff { \pianoLHMusik }
>>

\new ChoirStaff = "ChoirStaff_choir" <<
  \new Staff = "Staff_soprano" \with {
    instrumentName = "Soprano"
  }
  \new Voice = "soprano" \sopranoMusic
  \new Lyrics \lyricsto "soprano" { \sopranoLyrics }

  \new GrandStaff = "GrandStaff_altos" \with {
    \accepts Lyrics
  } <<
    \new Staff = "Staff_altoI" \with {
      instrumentName = "Alto I"
    }
    \new Voice = "altoI"
    \altoIMusic
    \new Lyrics \lyricsto "altoI" { \altoILyrics }
    \new Staff = "Staff_altoII" \with {
      instrumentName = "Alto II"
    }
    \new Voice = "altoII"

```

```

        \altoIIMusic
        \new Lyrics \lyricsto "altoII" { \altoIILyrics }
    >>

    \new Staff = "Staff_tenor" \with {
        instrumentName = "Tenor"
    }
    \new Voice = "tenor" \tenorMusic
    \new Lyrics \lyricsto "tenor" { \tenorLyrics }
>>

\new StaffGroup = "StaffGroup_strings" <<
    \new GrandStaff = "GrandStaff_violins" <<
        \new Staff = "Staff_violinI" \with {
            instrumentName = "Violin I"
        }
        \violinIMusic
        \new Staff = "Staff_violinII" \with {
            instrumentName = "Violin II"
        }
        \violinIIMusic
    >>

    \new Staff = "Staff_viola" \with {
        instrumentName = "Viola"
    }
    \violaMusic

    \new Staff = "Staff_cello" \with {
        instrumentName = "Cello"
    }
    \celloMusic

    \new Staff = "Staff_bass" \with {
        instrumentName = "Double Bass"
    }
    \bassMusic
>>
>>
}
}

```

Flute

Clarinet in B \flat

Horn in F

Trumpet in C

Percussion

Piano

Soprano

Alto I

Alto II

Tenor

Violin I

Violin II

Viola

Cello

Double Bass

Lyr - ics

Lyr - ics

Ah - ah

Lyr - ics

8

8

Vorlage für einfache Klaviernotation

Hier ein einfaches Klaviersystem.

```
upper = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

  a4 b c d
}

lower = \relative c {
  \clef bass
  \key c \major
  \time 4/4

  a2 c
}

\score {
  \new PianoStaff \with { instrumentName = "Piano" }
  <<
    \new Staff = "upper" \upper
    \new Staff = "lower" \lower
  >>
  \layout { }
  \midi { }
}
```



Piano template with centered lyrics

Anstatt ein eigenes System für Melodie und Text zu schreiben, können Sie den Text auch zwischen die beiden Klaviersysteme schreiben (und damit das zusätzliche System für die Gesangsstimme auslassen).

```
upper = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

  a4 b c d
}

lower = \relative c {
  \clef bass
  \key c \major
  \time 4/4
```

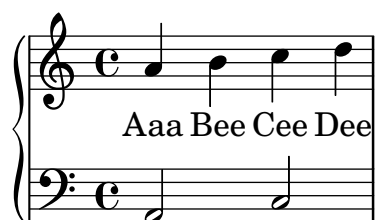
```

a2 c
}

text = \lyricmode {
  Aaa Bee Cee Dee
}

\score {
  \new PianoStaff <<
    \new Staff = upper { \new Voice = "singer" \upper }
    \new Lyrics \lyricsto "singer" \text
    \new Staff = lower { \lower }
  >>
  \layout { }
  \midi { }
}

```



Vorlage für Klavier und Gesangsstimme

Das nächste Beispiel ist typisch für ein Lied: Im oberen System die Melodie mit Text, darunter Klavierbegleitung.

```

melody = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

  a b c d
}

text = \lyricmode {
  Aaa Bee Cee Dee
}

upper = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

  a4 b c d
}

lower = \relative c {
  \clef bass
  \key c \major

```

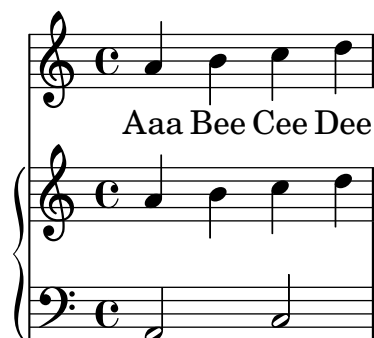
```

\time 4/4

a2 c
}

\score {
  <<
    \new Voice = "mel" { \autoBeamOff \melody }
    \new Lyrics \lyricsto mel \text
    \new PianoStaff <<
      \new Staff = "upper" \upper
      \new Staff = "lower" \lower
    >>
  >>
  \layout {
    \context { \Staff \RemoveEmptyStaves }
  }
  \midi { }
}

```



SATB-Chorvorlage auf vier Systemen

SATB-Chorvorlage auf vier Systemen

```

global = {
  \key c \major
  \time 4/4
  \dynamicUp
}

sopranonotes = \relative c'' {
  c2 \p \< d c d \f
}

sopranowords = \lyricmode { do do do do }
altonotes = \relative c'' {
  c2 \p d c d
}

altowords = \lyricmode { re re re re }
tenornotes = {
  \clef "G_8"
  c2 \mp d c d
}

tenorwords = \lyricmode { mi mi mi mi }
bassnotes = {

```

```

\clef bass
c2\mf d c d
}
basswords = \lyricmode { mi mi mi mi }

\score {
  \new ChoirStaff <<
    \new Staff <<
      \new Voice = "soprano" <<
        \global
        \sopranonotes
      >>
      \new Lyrics \lyricsto "soprano" \sopranowords
    >>
    \new Staff <<
      \new Voice = "alto" <<
        \global
        \altonotes
      >>
      \new Lyrics \lyricsto "alto" \altowords
    >>
    \new Staff <<
      \new Voice = "tenor" <<
        \global
        \tenornotes
      >>
      \new Lyrics \lyricsto "tenor" \tenorwords
    >>
    \new Staff <<
      \new Voice = "bass" <<
        \global
        \bassnotes
      >>
      \new Lyrics \lyricsto "bass" \basswords
    >>
  >>
}

```



Vorlage für eine Notenzeile mit Noten, Text und Akkorden

Mit diesem Beispiel können Sie einen Song mit Melodie, Text und Akkorden schreiben.

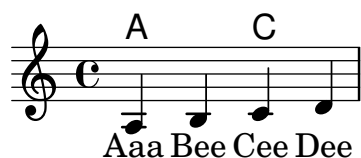
```
melody = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

  a4 b c d
}

text = \lyricmode {
  Aaa Bee Cee Dee
}

harmonies = \chordmode {
  a2 c
}

\score {
  <<
    \new ChordNames {
      \set chordChanges = ##t
      \harmonies
    }
    \new Voice = "one" { \autoBeamOff \melody }
    \new Lyrics \lyricsto "one" \text
  >>
  \layout { }
  \midi { }
}
```

Single-staff template with notes, lyrics, chords, and frets

Here is a simple lead sheet template with melody, lyrics, chords, and fret diagrams.

```

verseI = \lyricmode {
  \set stanza = #"1."
  This is the first verse
}

verseII = \lyricmode {
  \set stanza = #"2."
  This is the second verse.
}

theChords = \chordmode {
  % insert chords for chordnames and fretboards here
  c2 g4 c
}

staffMelody = \relative c' {
  \key c \major
  \clef treble
  % Type notes for melody here
  c4 d8 e f4 g
  \bar "|"
}

\score {
  <<
    \context ChordNames { \theChords }
    \context FretBoards { \theChords }
    \new Staff {
      \context Voice = "voiceMelody" { \staffMelody }
    }
    \new Lyrics = "lyricsI" {
      \lyricsto "voiceMelody" \verseI
    }
    \new Lyrics = "lyricsII" {
      \lyricsto "voiceMelody" \verseII
    }
  >>
  \layout { }
  \midi { }
}

```

1. This is the first verse
2. This is the second verse.

Vorlage für eine Notenzeile mit Akkorden

Wollen Sie ein Liedblatt mit Melodie und Akkorden schreiben? Hier ist das richtige Beispiel für Sie!

```
melody = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

  f4 e8[ c] d4 g |
  a2 ~ a
}

harmonies = \chordmode {
  c4:m f:min7 g:maj c:aug |
  d2:dim b4:5 e:sus
}

\score {
  <<
    \new ChordNames {
      \set chordChanges = ##t
      \harmonies
    }
    \new Staff \melody
  >>
  \layout{ }
  \midi { }
}
```

Vorlage für ein Notensystem mit Noten und Gesangstext

Das nächste Beispiel zeigt eine einfache Melodie mit Text. Kopieren Sie es in Ihre Datei, fügen Sie Noten und Text hinzu und übersetzen Sie es mit LilyPond. In dem Beispiel wird die automatische Balkenverbindung ausgeschaltet (mit dem Befehl `\autoBeamOff`), wie es für Vokalmusik üblich ist. Wenn Sie die Balken wieder einschalten wollen, müssen Sie die entsprechende Zeile entweder ändern oder auskommentieren.

```
melody = \relative c' {
  \clef treble
```

```

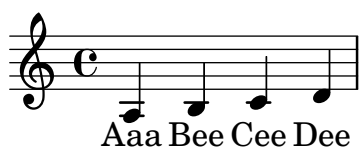
\key c \major
\time 4/4

a4 b c d
}

text = \lyricmode {
  Aaa Bee Cee Dee
}

\score{
  <<
    \new Voice = "one" {
      \autoBeamOff
      \melody
    }
    \new Lyrics \lyricsto "one" \text
  >>
  \layout { }
  \midi { }
}

```



Vorlage für ein Notensystem

Das erste Beispiel zeigt ein Notensystem mit Noten, passend für ein Soloinstrument oder ein Melodiefragment. Kopieren Sie es und fügen Sie es in Ihre Datei ein, schreiben Sie die Noten hinzu, und Sie haben eine vollständige Notationsdatei.

```

melody = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

  a4 b c d
}

\score {
  \new Staff \melody
  \layout { }
  \midi { }
}

```



Vorlage für Streichquartett (einfach)

Dieses Beispiel demonstriert die Partitur für ein Streichquartett. Hier wird auch eine „\global“-Variable für Taktart und Vorzeichen benutzt.

```

global= {
  \time 4/4
  \key c \major
}

violinOne = \new Voice \relative c' {
  c2 d
  e1
  \bar "|."
}

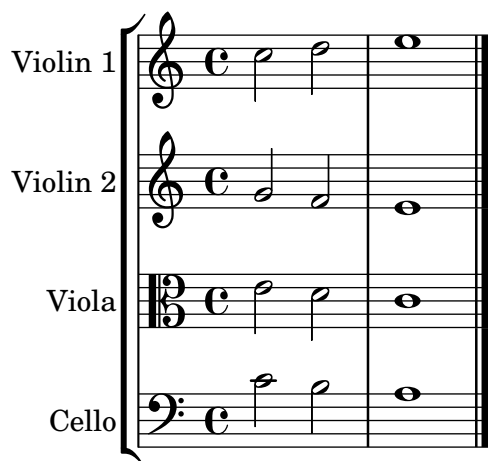
violinTwo = \new Voice \relative c' {
  g2 f
  e1
  \bar "|."
}

viola = \new Voice \relative c' {
  \clef alto
  e2 d
  c1
  \bar "|."
}

cello = \new Voice \relative c' {
  \clef bass
  c2 b
  a1
  \bar "|."
}

\score {
  \new StaffGroup <<
    \new Staff \with { instrumentName = "Violin 1" }
      << \global \violinOne >>
    \new Staff \with { instrumentName = "Violin 2" }
      << \global \violinTwo >>
    \new Staff \with { instrumentName = "Viola" }
      << \global \viola >>
    \new Staff \with { instrumentName = "Cello" }
      << \global \cello >>
  >>
  \layout { }
  \midi { }
}

```



Vorlage für Streichquartett mit einzelnen Stimmen

Mit diesem Beispiel können Sie ein schönes Streichquartett notieren, aber wie gehen Sie vor, wenn Sie Stimmen brauchen? Das Beispiel oben hat gezeigt, wie Sie mit Variablen einzelne Abschnitte getrennt voneinander notieren können. Im nächsten Beispiel wird nun gezeigt, wie Sie mit diesen Variablen einzelne Stimmen erstellen.

Sie müssen das Beispiel in einzelne Dateien aufteilen; die Dateinamen sind in den Kommentaren am Anfang jeder Datei enthalten. `piece.ly` enthält die Noten. Die anderen Dateien – `score.ly`, `vn1.ly`, `vn2.ly`, `vla.ly` und `vlc.ly` – erstellen daraus die entsprechenden Stimmen bzw. die Partitur (`score.ly`). Mit `ag` wird den Stimmen ein Name zugewiesen, auf den zurückgegriffen werden kann.

```
% piece.ly
% (This is the global definitions file.)
```

```
global= {
  \time 4/4
  \key c \major
}
```

```
Violinone = \new Voice \relative c' {
  c2 d e1
  \bar "|."
}
```

```
Violintwo = \new Voice \relative c' {
  g2 g e1
  \bar "|."
}
```

```
Viola = \new Voice \relative c' {
  \clef alto
  e2 d c1
  \bar "|."
}
```

```
Cello = \new Voice \relative c' {
  \clef bass
```

```

    c2 b a1
    \bar "|"
}

music = <<
  \tag #'score \tag #'vn1
  \new Staff \with { instrumentName = "Violin 1" }
    << \global \Violinone >>

  \tag #'score \tag #'vn2
  \new Staff \with { instrumentName = "Violin 2" }
    << \global \Violintwo >>

  \tag #'score \tag #'vla
  \new Staff \with { instrumentName = "Viola" }
    << \global \Viola >>

  \tag #'score \tag #'vlc
  \new Staff \with { instrumentName = "Cello" }
    << \global \Cello >>
>>

% These are the other files you need to save on your computer

% score.ly
% (This is the main file.)

% Uncomment the line below when using a separate file.
% \include "piece.ly"

#(set-global-staff-size 14)

\score {
  \new StaffGroup \keepWithTag #'score \music
  \layout { }
  \midi { }
}

%{ Uncomment this block when using separate files.

% vn1.ly
% (This is the Violin 1 part file.)

\include "piece.ly"
\score {
  \keepWithTag #'vn1 \music
  \layout { }
}

% vn2.ly

```

```
% (This is the Violin 2 part file.)
```

```
\include "piece.ly"
\score {
  \keepWithTag #'vn2 \music
  \layout { }
}
```

```
% vla.ly
% (This is the Viola part file.)
```

```
\include "piece.ly"
\score {
  \keepWithTag #'vla \music
  \layout { }
}
```

```
% vlc.ly
% (This is the Cello part file.)
```

```
\include "piece.ly"
\score {
  \keepWithTag #'vlc \music
  \layout { }
}
```

```
%}
```

Vorlage für Vokalensemble

Dieses Beispiel ist für vierstimmigen Gesang (SATB). Bei größeren Stücken ist es oft sinnvoll, eine allgemeine Variable zu bestimmen, die in allen Stimmen eingefügt wird. Taktart und Vorzeichen etwa sind fast immer gleich in allen Stimmen.

```
\paper {
  top-system-spacing.basic-distance = 10
  score-system-spacing.basic-distance = 20
  system-system-spacing.basic-distance = 20
  last-bottom-spacing.basic-distance = 10
}
```

```
global = {
  \key c \major
}
```

```

\time 4/4
}

sopMusic = \relative {
  c''4 c c8[( b)] c4
}
sopWords = \lyricmode {
  hi hi hi hi
}

altoMusic = \relative {
  e'4 f d e
}
altoWords = \lyricmode {
  ha ha ha ha
}

tenorMusic = \relative {
  g4 a f g
}
tenorWords = \lyricmode {
  hu hu hu hu
}

bassMusic = \relative {
  c4 c g c
}
bassWords = \lyricmode {
  ho ho ho ho
}

\score {
  \new ChoirStaff <<
    \new Lyrics = "sopranos" \with {
      % this is needed for lyrics above a staff
      \override VerticalAxisGroup.staff-affinity = #DOWN
    }
    \new Staff = "women" <<
      \new Voice = "sopranos" {
        \voiceOne
        << \global \sopMusic >>
      }
      \new Voice = "altos" {
        \voiceTwo
        << \global \altoMusic >>
      }
    >>
    \new Lyrics = "altos"
    \new Lyrics = "tenors" \with {
      % this is needed for lyrics above a staff
      \override VerticalAxisGroup.staff-affinity = #DOWN
    }
  }
}

```



```

\new Staff = "men" <<
  \clef bass
  \new Voice = "tenors" {
    \voiceOne
    << \global \tenorMusic >>
  }
  \new Voice = "basses" {
    \voiceTwo << \global \bassMusic >>
  }
>>
\new Lyrics = "basses"
\context Lyrics = "sopranos" \lyricsto "sopranos" \sopWords
\context Lyrics = "altos" \lyricsto "altos" \altoWords
\context Lyrics = "tenors" \lyricsto "tenors" \tenorWords
\context Lyrics = "basses" \lyricsto "basses" \bassWords
>>
}

```



Vocal ensemble template with automatic piano reduction

In diesem Beispiel wird ein automatischer Klavierauszug zu der Chorpartitur hinzugefügt. Das zeigt eine der Stärken von LilyPond – man kann eine Variable mehr als einmal benutzen. Wenn Sie irgendeine Änderung an einer Chorstimme vornehmen, (etwa `tenorMusic`), verändert sich auch der Klavierauszug entsprechend.

```

\paper {
  top-system-spacing.basic-distance = 10
  score-system-spacing.basic-distance = 20
  system-system-spacing.basic-distance = 20
  last-bottom-spacing.basic-distance = 10
}

```

```

global = {
  \key c \major
  \time 4/4
}

```

```

sopMusic = \relative {
  c''4 c c8[( b)] c4
}
sopWords = \lyricmode {
  hi hi hi hi
}

```

```

}

altoMusic = \relative {
  e'4 f d e
}
altoWords = \lyricmode {
  ha ha ha ha
}

tenorMusic = \relative {
  g4 a f g
}
tenorWords = \lyricmode {
  hu hu hu hu
}

bassMusic = \relative {
  c4 c g c
}
bassWords = \lyricmode {
  ho ho ho ho
}

\score {
  <<
    \new ChoirStaff <<
      \new Lyrics = "sopranos" \with {
        % This is needed for lyrics above a staff
        \override VerticalAxisGroup.staff-affinity = #DOWN
      }
      \new Staff = "women" <<
        \new Voice = "sopranos" { \voiceOne << \global \sopMusic >> }
        \new Voice = "altos" { \voiceTwo << \global \altoMusic >> }
      >>
      \new Lyrics = "altos"

      \new Lyrics = "tenors" \with {
        % This is needed for lyrics above a staff
        \override VerticalAxisGroup.staff-affinity = #DOWN
      }
      \new Staff = "men" <<
        \clef bass
        \new Voice = "tenors" { \voiceOne << \global \tenorMusic >> }
        \new Voice = "basses" { \voiceTwo << \global \bassMusic >> }
      >>
      \new Lyrics = "basses"

      \context Lyrics = "sopranos" \lyricsto "sopranos" \sopWords
      \context Lyrics = "altos" \lyricsto "altos" \altoWords
      \context Lyrics = "tenors" \lyricsto "tenors" \tenorWords
      \context Lyrics = "basses" \lyricsto "basses" \bassWords
    >>
  >>
}

```

```

\new PianoStaff <<
  \new Staff <<
    \set Staff.printPartCombineTexts = ##f
    \partCombine
    << \global \sopMusic >>
    << \global \altoMusic >>
  >>
  \new Staff <<
    \clef bass
    \set Staff.printPartCombineTexts = ##f
    \partCombine
    << \global \tenorMusic >>
    << \global \bassMusic >>
  >>
>>
>>
}

```

hi hi hi hi

ha ha ha ha

hu hu hu hu

ho ho ho ho

Vorlage für Vokalensemble mit dem Gesangstext über und unter dem System

In diesem Beispiel werden die Texte mit den Befehlen `alignAboveContext` und `alignBelowContext` über und unter dem System angeordnet.

```

global = {
  \key c \major
  \time 4/4
}

sopMusic = \relative c'' {
  c4 c c8[( b)] c4
}

sopWords = \lyricmode {

```

```

    hi hi hi hi
}

altoMusic = \relative c' {
    e4 f d e
}
altoWords = \lyricmode {
    ha ha ha ha
}

tenorMusic = \relative c' {
    g4 a f g
}
tenorWords = \lyricmode {
    hu hu hu hu
}

bassMusic = \relative c {
    c4 c g c
}
bassWords = \lyricmode {
    ho ho ho ho
}

\score {
  \new ChoirStaff <<
    \new Staff = "women" <<
      \new Voice = "sopranos" { \voiceOne << \global \sopMusic >> }
      \new Voice = "altos" { \voiceTwo << \global \altoMusic >> }
    >>
    \new Lyrics \with { alignAboveContext = "women" }
      \lyricsto "sopranos" \sopWords
    \new Lyrics \with { alignBelowContext = "women" }
      \lyricsto "altos" \altoWords
    % we could remove the line about this with the line below, since
    % we want the alto lyrics to be below the alto Voice anyway.
    % \new Lyrics \lyricsto "altos" \altoWords

    \new Staff = "men" <<
      \clef bass
      \new Voice = "tenors" { \voiceOne << \global \tenorMusic >> }
      \new Voice = "basses" { \voiceTwo << \global \bassMusic >> }
    >>
    \new Lyrics \with { alignAboveContext = "men" }
      \lyricsto "tenors" \tenorWords
    \new Lyrics \with { alignBelowContext = "men" }
      \lyricsto "basses" \bassWords
    % again, we could replace the line above this with the line below.
    % \new Lyrics \lyricsto "basses" \bassWords
  >>
}

```



Sologesang und zweistimmiger Refrain

Diese Vorlage erstellt eine Partitur, die mit Sologesang beginnt und einen Refrain für zwei Stimmen enthält. Sie zeigt auch die Benutzung von Platzhalter-Pausen innerhalb der `\global`-Variable, um Taktwechsel (und andere Elemente, die für alle Stimmen gleich sind) für das gesamte Stück zu definieren.

```
global = {
  \key g \major

  % verse
  \time 3/4
  s2.*2
  \break

  % refrain
  \time 2/4
  s2*2
  \bar "|."
}

SoloNotes = \relative g' {
  \clef "treble"

  % verse
  g4 g g |
  b4 b b |

  % refrain
  R2*2 |
}

SoloLyrics = \lyricmode {
  One two three |
  four five six |
}

SopranoNotes = \relative c'' {
  \clef "treble"

  % verse
```

```

R2.*2 |

% refrain
c4 c |
g4 g |
}

SopranoLyrics = \lyricmode {
  la la |
  la la |
}

BassNotes = \relative c {
  \clef "bass"

  % verse
  R2.*2 |

  % refrain
  c4 e |
  d4 d |
}

BassLyrics = \lyricmode {
  dum dum |
  dum dum |
}

\score {
  <<
    \new Voice = "SoloVoice" << \global \SoloNotes >>
    \new Lyrics \lyricsto "SoloVoice" \SoloLyrics

    \new ChoirStaff <<
      \new Voice = "SopranoVoice" << \global \SopranoNotes >>
      \new Lyrics \lyricsto "SopranoVoice" \SopranoLyrics

      \new Voice = "BassVoice" << \global \BassNotes >>
      \new Lyrics \lyricsto "BassVoice" \BassLyrics
    >>
  >>

  \layout {
    ragged-right = ##t
    \context { \Staff
      % these lines prevent empty staves from being printed
      \RemoveEmptyStaves
      \override VerticalAxisGroup.remove-first = ##t
    }
  }
}

```

One two three four five six

la la la la

dum dum dum dum

The image shows two musical staves. The top staff is a single treble clef in 3/4 time, with a key signature of one sharp (F#). It contains six quarter notes: G4, A4, B4, C5, B4, and A4. Below the staff are the lyrics 'One two three four five six'. The bottom staff is a grand staff (treble and bass clefs) in 3/4 time, with a key signature of one sharp (F#). The treble staff contains four quarter notes: G4, A4, B4, and C5, with lyrics 'la la la la' below. The bass staff contains four quarter notes: G3, A3, B3, and C4, with lyrics 'dum dum dum dum' below. A brace on the left side of the grand staff indicates it is a single musical unit.