

LilyPond

The music typesetter

Snippets

LilyPond Snippet Repository contributors

This document shows a selected set of LilyPond snippets from the [LilyPond Snippet Repository](#) (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 may be for a newer version (see below).

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.

This document is for LilyPond version 2.11.39.

Table of Contents

Pitches	1
Rhythms	14
Expressive marks.....	30
Repeats	39
Simultaneous notes.....	42
Staff notation.....	45
Editorial and educational use	56
Text	67
Vocal music.....	84
Chords.....	102
Piano	107
Percussion	119
Guitar	126
Strings.....	128
Ancient notation.....	135
Contexts and engravers	157
Tweaks and overrides.....	171
Paper and layout.....	195
Titles	205
Spacing	208
MIDI.....	217
Templates	221

Pitches

‘adding-ambiti-per-voice.ly’

Ambits can be added per voice. In that case, the ambitus must be moved manually to prevent collisions.

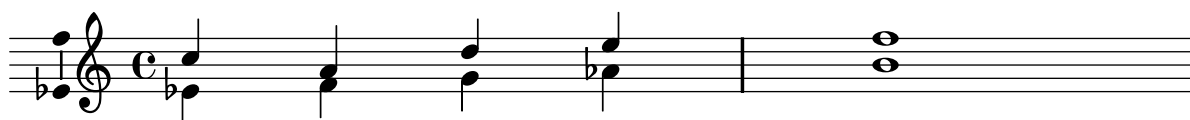
```
\new Staff <<
  \new Voice \with {
    \consists "Ambitus_engraver"
  } \relative c'' {
    \override Ambitus #'X-offset = # 2.0
    \voiceOne
    c4 a d e f1
  }
  \new Voice \with {
    \consists "Ambitus_engraver"
  } \relative c' {
    \voiceTwo
    es4 f g as b1
  }
}>>
```



‘ambiti-multiple-voices.ly’

If you have multiple voices in a single staff and you want a single ambitus per staff rather than per voice, add the `Ambitus_engraver` to the `Staff` context rather than to the `Voice` context.

```
\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
  }
}>>
```



‘applying-noteheads-styles-depending-on-the-step-of-the-scale.ly’

The `shapeNoteStyles` property gives you the ability to define various note heads styles for each step of the scale (as defined by the key signature or the "tonic" property).

This property requires a set of symbols, which can be purely arbitrary (geometrical expressions such as triangle, cross, xcircle etc. are allowed) or based on old American engraving tradition (you can use some latin note names as well).

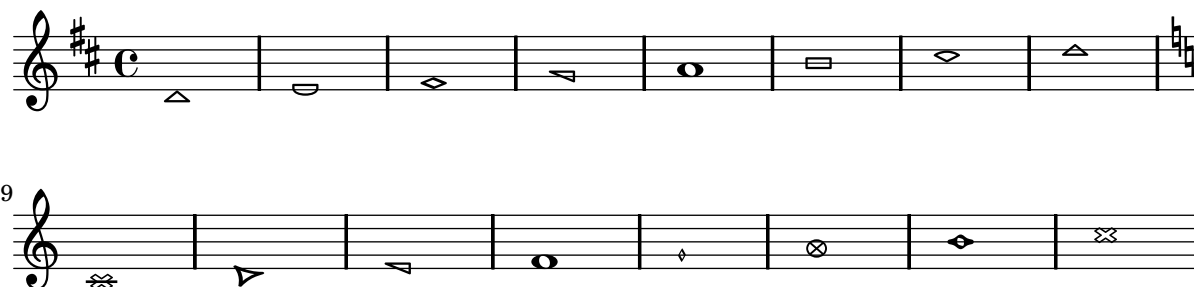
That said, if you’re trying to imitate old American song books, you may also want to try LilyPond’s predefined note heads styles, through shortcut commands such as `\aikenHeads` or `\sacredHarpHeads`.

This example shows different ways to obtain shape note heads, and demonstrates the ability to transpose a melody without losing the correspondance between harmonic functions and note heads styles.

```
fragment = {
  \key c \major
  c1 d e f g a b c
  \break
}

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

    \relative {
      \set shapeNoteStyles = ##(cross triangle fa #f mensural xcircle diamond)
      \fragment
    }
  }
}
```



‘clefs-commonly-tweaked-properties.ly’

The command `\clef "treble_8"` is equivalent to setting `clefGlyph`, `clefPosition` (which controls the Y position of the clef), `middleCPosition` and `clefOctavation`. A clef is printed when any of these properties are changed.

Note that changing the glyph, the position of the clef, or the octavation, does not in itself change the position of subsequent notes on the staff: the position of middle C must also be specified to do this. The positional parameters are relative to the staff centre line, positive

numbers displacing upwards, counting 1 for each line and space. The `clefOctavation` value would normally be set to 7, -7, 15 or -15, but other values are not invalid.

When a clef change takes place at a line break the new clef symbol is printed at both the end of the previous line and the beginning of the new line by default. If the warning clef at the end of the previous line is not required it can be suppressed by setting the `explicitClefVisibility` Staff property to the value `end-of-line-invisible`. The default behaviour can be recovered with `\unset Staff.explicitClefVisibility`.

The following examples show the possibilities when setting these properties manually. On the first line, the manual changes preserve the standard relative positioning of clefs and notes, whereas on the second line, they do not.

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

% The following clef changes do not preserve
% the normal relationship between notes and clefs:

\set Staff.clefGlyph = #"clefs.F"
\set Staff.clefPosition = #2
c'
\set Staff.clefGlyph = #"clefs.G"
c'
\set Staff.clefGlyph = #"clefs.C"
c'
\set Staff.clefOctavation = #7
c'
\set Staff.clefOctavation = #0
\set Staff.clefPosition = #0
c'}
```

```
% Here we go back to the normal clef:

\set Staff.middleCPosition = #4
c'
}
```



‘creating-a-sequence-of-notes-on-various-pitches.ly’

In music that contains many occurrences of the same sequence of notes at different pitches, you can use the following music function. It takes a note, of which the pitch is used. The supporting Scheme functions were borrowed from the Tips and Tricks document in the manual.

This example creates the rhythm used throughout Mars, from The Planets, by Gustav Holst.

```
#(define (make-note-req p d)
  (make-music 'NoteEvent
    'duration d
    'pitch p))

#(define (make-note p d)
  (make-music 'EventChord
    'elements (list (make-note-req p d))))

#(define (seq-music-list elts)
  (make-music 'SequentialMusic
    'elements elts))

#(define (make-triplet elt)
  (make-music 'TimeScaledMusic
    'denominator 3
    'numerator 2
    'element elt))

rhythm = #(define-music-function (parser location note) (ly:music?)
  "Make the rhythm in Mars (the Planets) at the given note's pitch"
  (let* ((p (ly:music-property
    (car (ly:music-property note 'elements))
    'pitch)))
    (seq-music-list (list
      (make-triplet (seq-music-list (list
        (make-note p (ly:make-duration 3 0 2 3))
        (make-note p (ly:make-duration 3 0 2 3))
```

```

        (make-note p (ly:make-duration 3 0 2 3))
      )))
      (make-note p (ly:make-duration 2 0))
      (make-note p (ly:make-duration 2 0))
      (make-note p (ly:make-duration 3 0))
      (make-note p (ly:make-duration 3 0))
      (make-note p (ly:make-duration 2 0))
    ))))

\score {
  \new Staff {
    \time 5/4

    \rhythm c'
    \rhythm c''
    \rhythm g
  }
}

```



`'dodecaphonic-style-accidentals-for-each-note-including-naturals.ly'`

In early XXth century works, starting with Schönberg, Berg and Webern (the "second" Viennese school), every pitch in the twelve-tone scale has to be regarded as equal, without any hierarchy such as the classical (tonal) degrees. Therefore, these composers print one accidental for each note, even at natural pitches, to emphasize their new approach to music theory and language. This snippet shows how to achieve such notation rules with LilyPond.

```

webernAccidentals = {
  % the 5s are just "a value different from any accidental"
  \set Staff.keySignature = #'((0 . 5) (1 . 5) (2 . 5) (3 . 5)
    (4 . 5) (5 . 5) (6 . 5))

  \set Staff.extraNatural = ##f
  #(set-accidental-style 'forget)
}

\layout {
  \context { \Staff \remove Key_engraver }
}

\score {
  {
    \webernAccidentals
    c' dis' cis' cis'
    c' dis' cis' cis'
    c' c' dis' des'
  }
}

```



‘generating-random-notes.ly’

This Scheme-based snippet allows you to generate 24 random notes (or as many as you want), based on the current time (or any randomish number you might wish to specify instead, so you can obtain the same random notes each time): i.e. to get different random notes patterns, just change this number.

```
\score {
{ # (let ((random-state (seed->random-state (current-time))))
  (ly:export
    (make-music 'SequentialMusic 'elements
      (map (lambda x
        (let ((idx (random 12 random-state)))
          (make-music 'EventChord
            'elements (list (make-music 'NoteEvent
              'duration (ly:make-duration 2 0 1 1)
              'pitch (ly:make-pitch (quotient idx 7)
                (remainder idx 7)
                0))))))
      (make-list 24))))))
}
```



‘makam.ly’

Makam is Turkish type of melody that uses 1/9 tone microtonal alterations.

% Define 1/9 alterations.

```
#(define-public KOMA 1/9)
#(define-public BAKIYE 4/9)
#(define-public KUCUK 5/9)
#(define-public BUYUKMUCENNEB 8/9)
```

% Define pitch names

```
makamPitchNames = #` (
  (c . ,(ly:make-pitch -1 0 NATURAL))
  (d . ,(ly:make-pitch -1 1 NATURAL))
  (e . ,(ly:make-pitch -1 2 NATURAL))
  (f . ,(ly:make-pitch -1 3 NATURAL))
  (g . ,(ly:make-pitch -1 4 NATURAL))
  (a . ,(ly:make-pitch -1 5 NATURAL))
  (b . ,(ly:make-pitch -1 6 NATURAL))

  (cc . ,(ly:make-pitch -1 0 KOMA))
  (dc . ,(ly:make-pitch -1 1 KOMA))
```



```

(ec . ,(ly:make-pitch -1 2 KOMA))
(fc . ,(ly:make-pitch -1 3 KOMA))
(gc . ,(ly:make-pitch -1 4 KOMA))
(ac . ,(ly:make-pitch -1 5 KOMA))
(bc . ,(ly:make-pitch -1 6 KOMA))

(cb . ,(ly:make-pitch -1 0 BAKIYE))
(db . ,(ly:make-pitch -1 1 BAKIYE))
(eb . ,(ly:make-pitch -1 2 BAKIYE))
(fb . ,(ly:make-pitch -1 3 BAKIYE))
(gb . ,(ly:make-pitch -1 4 BAKIYE))
(ab . ,(ly:make-pitch -1 5 BAKIYE))
(bb . ,(ly:make-pitch -1 6 BAKIYE))

(ck . ,(ly:make-pitch -1 0 KUCUK))
(dk . ,(ly:make-pitch -1 1 KUCUK))
(ek . ,(ly:make-pitch -1 2 KUCUK))
(fk . ,(ly:make-pitch -1 3 KUCUK))
(gk . ,(ly:make-pitch -1 4 KUCUK))
(ak . ,(ly:make-pitch -1 5 KUCUK))
(bk . ,(ly:make-pitch -1 6 KUCUK))

(cbm . ,(ly:make-pitch -1 0 BUYUKMUCENNEB))
(dbm . ,(ly:make-pitch -1 1 BUYUKMUCENNEB))
(ebm . ,(ly:make-pitch -1 2 BUYUKMUCENNEB))
(fbm . ,(ly:make-pitch -1 3 BUYUKMUCENNEB))
(gbm . ,(ly:make-pitch -1 4 BUYUKMUCENNEB))
(abm . ,(ly:make-pitch -1 5 BUYUKMUCENNEB))
(bbm . ,(ly:make-pitch -1 6 BUYUKMUCENNEB))

;; f for flat.
(cfc . ,(ly:make-pitch -1 0 (- KOMA)))
(dfk . ,(ly:make-pitch -1 1 (- KOMA)))
(efk . ,(ly:make-pitch -1 2 (- KOMA)))
(ffk . ,(ly:make-pitch -1 3 (- KOMA)))
(gfk . ,(ly:make-pitch -1 4 (- KOMA)))
(afk . ,(ly:make-pitch -1 5 (- KOMA)))
(bfk . ,(ly:make-pitch -1 6 (- KOMA)))

(cfb . ,(ly:make-pitch -1 0 (- BAKIYE)))
(dfk . ,(ly:make-pitch -1 1 (- BAKIYE)))
(efk . ,(ly:make-pitch -1 2 (- BAKIYE)))
(ffk . ,(ly:make-pitch -1 3 (- BAKIYE)))
(gfk . ,(ly:make-pitch -1 4 (- BAKIYE)))
(afb . ,(ly:make-pitch -1 5 (- BAKIYE)))
(bfb . ,(ly:make-pitch -1 6 (- BAKIYE)))

(cfk . ,(ly:make-pitch -1 0 (- KUCUK)))
(dfk . ,(ly:make-pitch -1 1 (- KUCUK)))
(efk . ,(ly:make-pitch -1 2 (- KUCUK)))
(ffk . ,(ly:make-pitch -1 3 (- KUCUK)))
(gfk . ,(ly:make-pitch -1 4 (- KUCUK)))

```

```

(afk . ,(ly:make-pitch -1 5 (- KUCUK)))
(bfk . ,(ly:make-pitch -1 6 (- KUCUK)))

(cfbm . ,(ly:make-pitch -1 0 (- BUYUKMUCENNEB)))
(dfbm . ,(ly:make-pitch -1 1 (- BUYUKMUCENNEB)))
(efbm . ,(ly:make-pitch -1 2 (- BUYUKMUCENNEB)))
(ffbm . ,(ly:make-pitch -1 3 (- BUYUKMUCENNEB)))
(gfbm . ,(ly:make-pitch -1 4 (- BUYUKMUCENNEB)))
(afbm . ,(ly:make-pitch -1 5 (- BUYUKMUCENNEB)))
(bfbm . ,(ly:make-pitch -1 6 (- BUYUKMUCENNEB)))

)

%% set pitch names.
pitchnames = \makamPitchNames
#(ly:parser-set-note-names parser makamPitchNames)

makamGlyphs = #'((1 . "accidentals.doublesharp")
  (8/9 . "accidentals.sharp.slashslashslash.stemstem")
  (5/9 . "accidentals.sharp.slashslashslash.stem")
  (4/9 . "accidentals.sharp")
  (1/9 . "accidentals.sharp.slashslash.stem")
  (0 . "accidentals.natural")
  (-1/9 . "accidentals.mirroredflat")
  (-4/9 . "accidentals.flat.slash")
  (-5/9 . "accidentals.flat")
  (-8/9 . "accidentals.flat.slashslash")
  (-1 . "accidentals.flatflat")
)

\relative {

  %{ define alteration <-> symbol mapping. The following glyphs are available.
  accidentals.sharp
  accidentals.sharp.slashslash.stem
  accidentals.sharp.slashslashslash.stemstem
  accidentals.sharp.slashslashslash.stem
  accidentals.sharp.slashslash.stemstemstem
  accidentals.natural
  accidentals.flat
  accidentals.flat.slash
  accidentals.flat.slashslash
  accidentals.mirroredflat.flat
  accidentals.mirroredflat
  accidentals.flatflat
  accidentals.flatflat.slash
  accidentals.doublesharp
  %}

  \override Accidental #'glyph-name-alist = \makamGlyphs

  \override Staff.KeySignature #'glyph-name-alist = \makamGlyphs

```

```
\set Staff.keySignature = #'(
  (3 . 4/9)
  (6 . -1/9))

c cc db fk gbm gfc gfb efk dfbm
}
```



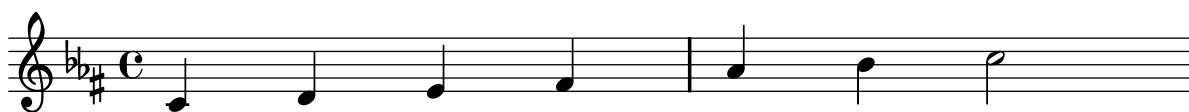
‘non-traditional-key-signatures.ly’

The commonly used `\key` command sets the `keySignature` property, in the `Staff` context. However, non-standard key signatures can be specified by setting this property directly. The format of this command is a list: `\set Staff.keySignature = #'((octave . step) . alter) ((octave . step) . alter) ...` where, for each element in the list, octave specifies the octave (0 being the octave from middle C to the B above), step specifies the note within the octave (0 means C and 6 means B), and alter is `,SHARP`, `,FLAT`, `,DOUBLE-SHARP` etc. (Note the leading comma.)

However, for each item in the list, you can also use the alternative format `(step . alter)`, which specifies that the same alteration should hold in all octaves.

Here is an example of a possible key signature for generating a whole-tone scale:

```
\relative c' {
  \set Staff.keySignature =
    #`(((0 . 3) . ,SHARP) ((0 . 5) . ,FLAT) ((0 . 6) . ,FLAT))
  c d e fis aes bes c2
}
```



‘ottava-text.ly’

Internally, the `set-octavation` function sets the properties `ottavation` (e.g., to `"8va"` or `"8vb"`) and `middleCPosition`. To override the text of the bracket, set `ottavation` after invoking `set-octavation`, like in the following example.

```
{
  #(set-octavation 1)
  \set Staff.ottavation = #"8"
  c''1
  #(set-octavation 0)
  c'1
  #(set-octavation 1)
  \set Staff.ottavation = #"Text"
  c''1
}
```



‘preventing-extra-naturals-from-being-automatically-added.ly’

In accordance with standard typesetting rules, a natural sign is printed before a sharp or flat if a previous accidental on the same note needs to be canceled. To change this behavior, set the `extraNatural` property to "false" in the Staff context.

```
\relative {
  aeses'4 aes ais a
  \set Staff.extraNatural = ##f
  aeses4 aes ais a
}
```



‘preventing-natural-signs-from-being-printed-when-the-key-signature-changes.ly’

When the key signature changes, natural signs are automatically printed to cancel any accidentals from previous key signatures. This may be altered by setting to "false" the `printKeyCancellation` property in the Staff context.

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



‘quoting-another-voice-with-transposition.ly’

Quotations take into account the transposition of both source and target. In this example, all instruments play sounding central C, the target is a instrument in F. The target part may be `\transposed`. In this case, all the pitches (including the quoted ones) will transposed as well.

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

```
\addQuote sax {
```

```

\transposition es'
a8 a a a a a a a
}

quoteTest = {
  \transposition f % french horn

  g'4
  << \quoteDuring #"clarinet" { \skip 4 } s4^"clar" >>
  << \quoteDuring #"sax" { \skip 4 } s4^"sax" >>
}

<< \quoteTest
  \new Staff
  << \transpose c' d' \quoteTest
    s4_"up 1 tone"
  >>
>>
>>

```

The image displays two musical staves. The top staff is labeled 'clar' and 'sax'. The bottom staff is labeled 'clar' and 'sax' and has 'up 1 tone' written below it. Both staves show a sequence of notes: a quarter note, a half note, a quarter note, and a half note, followed by a quarter rest and a half note.

`'transposing-pitches-with-minimum-accidentals-smart-transpose.ly'`

There is a way to enforce enharmonic modifications for notes in order to have the minimum number of accidentals. In that case, “Double accidentals should be removed, as well as E-sharp (-> F), bC (-> B), bF (-> E), B-sharp (-> C).”, as proposed by a request for a new feature. In this manner, the most natural enharmonic notes are chosen in this example.

```

#(define (naturalise-pitch p)
  (let* ((o (ly:pitch-octave p))
        (a (* 4 (ly:pitch-alteration p)))
        ; alteration, a, in quarter tone steps, for historical reasons
        (n (ly:pitch-notename p)))

    (cond
      ((and (> a 1) (or (eq? n 6) (eq? n 2)))
       (set! a (- a 2))
       (set! n (+ n 1)))
      ((and (< a -1) (or (eq? n 0) (eq? 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))))

    (if (< n 0) (begin (set! o (- o 1)) (set! n (+ n 7)))))

```

```

(if (> n 6) (begin (set! o (+ o 1)) (set! n (- n 7))))

(ly:make-pitch o n (/ a 4)))

#(define (naturalise music)
  (let* ((es (ly:music-property music 'elements))
        (e (ly:music-property music 'element))
        (p (ly:music-property music 'pitch)))

    (if (pair? es)
        (ly:music-set-property!
         music 'elements
         (map (lambda (x) (naturalise x)) es)))

        (if (ly:music? e)
            (ly:music-set-property!
             music 'element
             (naturalise e)))

            (if (ly:pitch? p)
                (begin
                 (set! p (naturalise-pitch p))
                 (ly:music-set-property! music 'pitch p)))

                music)))

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

naturaliseMusic =
#(define-music-function (parser location m)
                        (ly:music?)
                        (naturalise m))

\score {
  \new Staff {
    \transpose c ais \music
    \naturaliseMusic \transpose c ais \music
    \break
    \transpose c deses \music
    \naturaliseMusic \transpose c deses \music
  }
  \layout { ragged-right = ##t}
}

```





Rhythms

‘adding-beams,-slurs,-ties-etc.-when-using-tuplet-and-non-tuplet-rythms..ly’

LilyPond syntax can involve many unusual placements for parentheses, brackets etc., which might sometimes have to be interleaved. 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 superpose manual beaming, manual slurs, ties, and phrasing slurs, while using tuplet sections (enclosed with curly braces).

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



‘adding-drum-parts.ly’

LilyPond makes drums input quite easy, with powerful pre-configured tools such as the `\drummode` function and the `DrumStaff` context: drums are placed to their own staff positions (with a special clef symbol) and have note heads according to the drum. You can easily attach an extra symbol to the drum, and restrict the number of lines.

```
drh = \drummode { cymc4.^"crash" hhc16^"h.h." hh \repeat "unfold" 5 {hhc8 hho hhc8 hh16 hh}
drl = \drummode {\repeat "unfold" 3 {bd4 sn8 bd bd4 << bd ss >> } bd8 tommh tommh bd toml t
timb = \drummode { \repeat "unfold" 2 {timh4 ssh timl8 ssh r timh r4 ssh8 timl r4 cb8 cb} }
```

```
\score {
  \repeat "volta" 2 {
    <<
      \new DrumStaff \with {
        drumStyleTable = #timbales-style
        \override StaffSymbol #'line-count = #2
        \override BarLine #'bar-size = #2
      } <<
        \set Staff.instrumentName = "timbales"
        \timb
      >>
      \new DrumStaff <<
        \set Staff.instrumentName = "drums"
        \new DrumVoice {\stemUp \drh }
        \new DrumVoice {\stemDown \drl }
      >>
    >>
  }
  \layout {}
}
```



```

\midi {
  \context {
    \Score
    tempoWholesPerMinute = #(ly:make-moment 120 4)
  }
}

```

‘altering-the-number-of-stems-in-a-beam.ly’

You can alter the number of stems in a beam. In this example, two sets of four 32nds are joined, as if they were 8th notes.

```

\relative {
  #(override-auto-beam-setting '(end * * * *) 1 4)
  f32 g a b b a g f

  f32 g a
  \set stemRightBeamCount = #1 b
  \set stemLeftBeamCount = #1 b
  a g f
}

```

‘automatic-beam-subdivisions.ly’

Lilypond can subdivide beams automatically. Set the property `subdivideBeams`, and beams are subdivided at beat positions (as specified in beat length)

```

\relative{ b'32^"default"[ a g f c' b a g f e d' c b a g f ]
  \set subdivideBeams = ##t
  b32^"subdivision enabled"[ a g f c' b a g f e d' c b a g f ]
  \set Score.beatLength = #(ly:make-moment 1 8)
  b32^"beatLength 1 8"[ a g f c' b a g f e d' c b a g f ]
  \set Score.beatLength = #(ly:make-moment 1 16)
  b32^"beatLength 1 16"[ a g f c' b a g f e d' c b a g f ]
}

```

The image displays four musical staves illustrating different beam settings. The first staff shows two examples: 'default' and 'subdivision enabled'. The second staff shows two examples: 'beatLength 1 8' and 'beatLength 1 16'. The notes are beamed in groups of two, four, and eight, with the 'beatLength 1 16' setting showing a different grouping pattern.

‘automatic-beams-two-per-two-in-4-4-or-2-2-time-signature.ly’

In time signature 2/2 or 4/4 the beam are ----- _ _ Default | | | | I
want | | | |. Use a "macro" with #(override-auto-beam-setting '.....

% Automatic beams two per two in 4/4 or 2/2 time signature

```

%
% -----
% Default   | | | |
%
% - - - -
% I want    | | | |

```

% The good way adapted from David Bobrof

% macro for beamed two per two in 2/2 and 4/4 time signature

```

qbeam={
  #(override-auto-beam-setting '(end 1 8 * *) 1 4 'Staff)
  #(override-auto-beam-setting '(end 1 8 * *) 2 4 'Staff)
  #(override-auto-beam-setting '(end 1 8 * *) 3 4 'Staff)
}

```

% other macros

```
timeFractionstyle={ \override Staff.TimeSignature #'style = #'() }
```

```
textn = ^\markup{ without the macro }
```

```
texty = ^\markup{ with the macro }
```

```
\score {
```

```
<<
```

```
\new Staff << \relative c'' {
```

```
\timeFractionstyle
```

```
\time 4/4
```

```
g8\textn g g g g g g g g g g4 g8 g g
```

```
}
```

```

>>

%Use the macro

\new Staff << \relative c'' {
  \timeFractionstyle
  \time 4/4
  \qbeam
  g8\texty g g g   g g g g   g g g g4   g8 g g
}

>>

>>
\layout{ raggedright = ##t }
}

```



‘beam-across-line-breaks.ly’

By default, beams can’t be printed across line breaks. Here’s a way to force the line break, by setting the #'breakable property. See also in the manual the "Line Breaking" and "Manual beams" sections.

```

\layout { ragged-right= ##t }

\relative c'' {
  \override Score.Beam #'breakable = ##t
  \time 3/16 c16-[ d e \break f-]
}

```



‘changing-time-signatures-inside-a-polymetric-section-using--compressmusic.ly’

The measureLength variable, together with measurePosition, determines when a barline is needed. However, when using \compressMusic, the scaling of durations makes it difficult to change time signatures without making a mess of it.

Therefore, measureLength has to be set manually, using the ly:make-moment callback. The second argument has to be the same as the second argument of \compressMusic.

```

\layout {
  \context { \Score
    \remove "Timing_translator"
    \remove "Default_bar_line_engraver"
  }
  \context {
    \Staff
    \consists "Timing_translator"
    \consists "Default_bar_line_engraver"
  }
}

<<
\new Staff {
  \compressMusic #'( 8 . 5 ) {
    \time 6/8
    \set Timing.measureLength = #(ly:make-moment 3 5)
    b8 b b b b b
    \time 2/4
    \set Timing.measureLength = #(ly:make-moment 2 5)
    b4 b
  }
}
\new Staff {
  \clef bass
  \time 2/4
  c2 d e f }
>>

```



‘chant-or-psalms-notation.ly’

This form of notation is used for the chant of the Psalms, where verses aren’t always the same length.

```

stemon = { \override Staff.Stem #'transparent = ##f }
stemoff = { \override Staff.Stem #'transparent = ##t }

\score {
  \new Staff \with {\remove "Time_signature_engraver" }
  {
    \key g \minor
    \set Score.timing = ##f
    \stemoff a'\breve bes'4 g'4
    \stemon a'2 \bar "||"
    \stemoff a'\breve g'4 a'4
  }
}

```

```

\stemon f'2 \bar "||"
\stemoff a'\breve^\markup { \italic flexe }
\stemon g'2 \bar "||"
}
\layout { raggedright = ##t}
}

```



‘compound-time-signatures.ly’

Odd 20th century time signatures (such as "5/8") can often be played as compound time signatures (e.g. "3/8 + 2/8"), which combine two or more unequal metrics. LilyPond can make such musics quite easy to read and play, by explicitly printing the compound time signatures and adapting the automatic beaming behaviour. (You can even add graphic measure grouping indications, see the appropriate snippet in this database.)

```

#(define (compound-time one two num)
  (markup #:override '(baseline-skip . 0) #:number
    (:#:line ((#:column (one num)) #:vcenter "+" (:#:column (two num))))))

\relative {
  %% compound time signature hack
  \time 5/8
  \override Staff.TimeSignature #'stencil = #ly:text-interface::print
  \override Staff.TimeSignature #'text = #(compound-time "2" "3" "8" )
  #(override-auto-beam-setting '(end 1 8 5 8) 1 4)
  c8 d e fis gis | c fis, gis e d | c8 d e4 gis8
}

```



‘conducting-signs,-measure-grouping-signs.ly’

The Scheme function `set-time-signature`, in combination with the Measure grouping engraver, it will create MeasureGrouping signs. Such signs ease reading rhythmically complex modern music. In the following example, the 9/8 measure is subdivided in 2, 2, 2 and 3. This is passed to `set-time-signature` as the third argument (2 2 2 3)

```

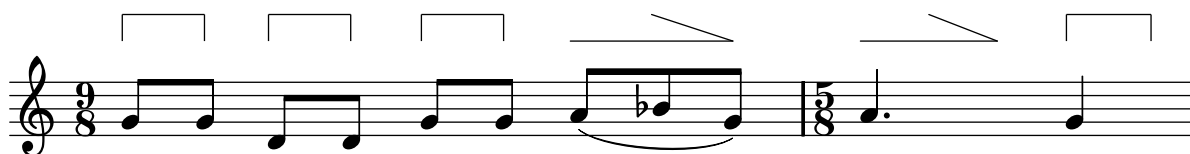
\score {
  \relative c' {
    #(set-time-signature 9 8 '(2 2 2 3))
    g8[ g] d[ d] g[ g] a8[( bes g)] |
    #(set-time-signature 5 8 '(3 2))
    a4. g4
  }
  \layout {
    \context {

```

```

\Staff
\consists "Measure_grouping_engraver"
}
}
}

```



‘controlling-tuplet-bracket-visibility.ly’

Default behaviour 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, you can set the property `TupletBracket #'bracket-visibility` to either `##t` (always print a bracket), `##f` (never print a bracket) or `#'if-no-beam` (only print a bracket if there is no beam).

```

mus = \relative c'' {
  \times 2/3 {c16 [ d e ] f8]
  \times 2/3 {c8 d e }
  \times 2/3 { c4 d e }
}

```

```

\new Voice \relative c'{
  << \mus s4^"default" >>
  \override TupletBracket #'bracket-visibility = #'if-no-beam
  << \mus s4^"if-no-beam" >>
  \override TupletBracket #'bracket-visibility = ##t
  << \mus s4^"##t" >>
  \override TupletBracket #'bracket-visibility = ##f
  << \mus s4^"##f" >>
}

```



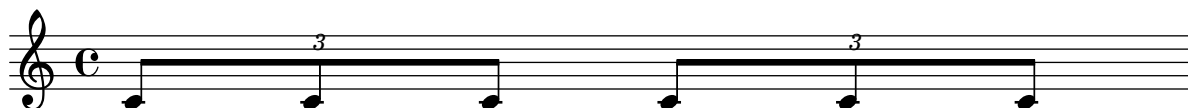
‘entering-several-tuplets-using-only-one---times-command.ly’

This example shows how to specify how long each of the tuplets contained within the bracket after `\times` should last. Many consecutive tuplets can then be contained within a single `\times` { ... }, thus saving typing.

In the example, two triplets are shown, while `\times` was entered only once.

For more information about `make-moment`, see "Time administration".

```
\relative {
  \set tupletSpannerDuration = #(ly:make-moment 1 4)
  \times 2/3 { c8 c c c c c }
}
```



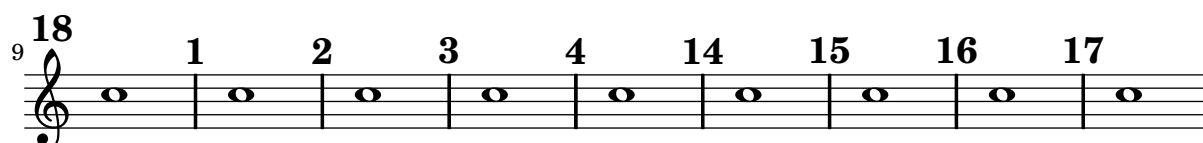
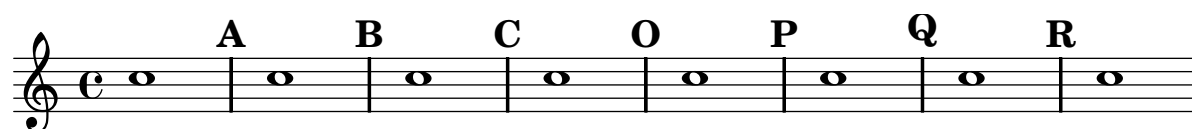
‘forcing-rehearsal-marks-to-start-from-a-given-letter-or-number.ly’

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

```
\relative c''{
  c1 \mark \default
  c1 \mark \default
  c1 \mark \default
  c1 \mark #14
  c1 \mark \default
  c1 \mark \default
  c1 \mark \default
  c1 \mark \default
  c1 \mark \default
}
```

```
\break
```

```
\set Score.markFormatter = #format-mark-numbers
c1 \mark #1
c1 \mark \default
c1 \mark \default
c1 \mark \default
c1 \mark #14
c1 \mark \default
c1 \mark \default
c1 \mark \default
c1 \mark \default
}
```



‘heavily-customized-polymetric-time-signatures.ly’

Though the set-time-signature thing was not the most essential here, it has been included to show the beat of this piece (which is a template of a real balkan song!).

```

#(define (compound-time one two three four five six seven eight nine ten num)
  (markup #:override '(baseline-skip . 0) #:number
    (:line ((#:column (one num)) #:vcenter "+") (:column (two num)) #:vcenter "+") (:column
melody =
{
  \relative c'' {
    \set Staff.instrumentName = "Bb Sop."
    \key g \major \time 25/8
    \override Staff.TimeSignature #'stencil = #ly:text-interface::print
    \override Staff.TimeSignature #'text = #(compound-time "3" "2" "2" "3" "2" "2" "2" "3" "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
      c4. d4 c4 d4. c4 d c2 d4. e4-^ d4 |
      c4. d4 c4 d4. c4 d c2 d4. e4-^ d4 | \break }
}
drum = {
  \new DrumStaff \drummode
  {
    \bar "|:" bd4. ^\markup { "Drums" } sn4 bd \bar ":" sn4. bd4 sn \bar ":"
    bd sn bd4. sn4 bd \bar ":@"
  }
}

{
  \melody
  \drum
}

```

Bb Sop.

2

4

6 Drums

`'making-an-object-invisible-with-the-transparent-property.ly'`

Setting the `transparent` property will cause an object to be printed in 'invisible ink': the object is not printed, but all its other behavior is retained. The object still takes up space, it takes part in collisions, and slurs, and ties and beams can be attached to it.

The 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 c'' <<
{
  \once \override Stem #'transparent = ##t
  b8~ b8\noBeam
} \\\ {
  b[ g8]
}
>>
```

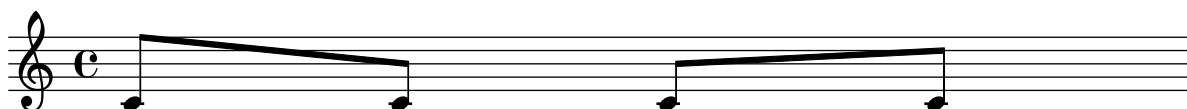


`'manually-controlling-beam-positions.ly'`

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

```
\score {
  \context Voice \relative c {
    %% from upper staffline (position 4) to centre (position 0)
    \override Beam #'positions = #'(2 . 0)
    c'8[ c]

    %% from center to one above centre (position 2)
    \override Beam #'positions = #'(0 . 1)
    c[ c]
  }
}
```



`'printing-music-with-different-time-signatures.ly'`

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

This can be achieved with the `\compressMusic` command, as demonstrated here.

The barlines can't be printed at the `Score` level anymore, so you have to remove the `Barline_engraver` and put it in the `Staff` context.

```
% Thanks to Adam James Wilson for this snippet

\paper {
  indent = #0
  ragged-right = ##t
}

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

\layout {
  \context { \Score
    \remove "Timing_translator"
    \remove "Time_signature_engraver"
    \remove "Default_bar_line_engraver"
    \override SpacingSpanner #'uniform-stretching = ##t
    \override SpacingSpanner #'strict-note-spacing = ##t
    proportionalNotationDuration = #(ly:make-moment 1 64)
  }
  \context { \Staff
    \consists "Timing_translator"
    \consists "Default_bar_line_engraver"
    \consists "Time_signature_engraver"
  }
  \context { \Voice
    \remove Forbid_line_break_engraver
    tupletFullLength = ##t
  }
}

Bassklarinette = \new Staff <<
  \global
  {
    \bar "|"
    \clef treble
    \time 3/8
    d''4.

    \bar "|"
    \time 3/4
    r8 des''2( c''8)

    \bar "|"
    \time 7/8
    r4. ees''2 ~

    \bar "|"
    \time 2/4
    \tupletUp
    \times 2/3 {ees''4 r4 d''4 ~}

    \bar "|"
  }
```

```

\time 3/8
\tupletUp
\times 3/4 {d''4 r4}

\bar "|"
\time 2/4
e''2

\bar "|"
\time 3/8
es''4.
\bar "|"
\time 3/4
r8 d''2 r8
\bar "|"
}

>>
Perkussion = \new StaffGroup <<
\new Staff <<
\global
{
\bar "|"
\clef percussion
\time 3/4
r4 c'2 ~

\bar "|"
c'2.

\bar "|"
R2.

\bar "|"
r2 g'4 ~

\bar "|"
g'2. ~

\bar "|"
g'2.
}
>>
\new Staff <<
\global
{
\bar "|"
\clef percussion
\time 3/4
R2.

\bar "|"
g'2. ~

```

```
\bar "|"
g'2.
```

```
\bar "|"
r4 g'2 ~
```

```
\bar "|"
g'2 r4
```

```
\bar "|"
g'2.
}
```

```
>>
```

```
>>
```

```
\score { << \Bassklarinette \Perkussion >>
```

```
}
```

A musical score for the song "The Rose Tree". The score is written for three parts: a vocal line and two piano accompaniment lines. The vocal line is in treble clef, 3/4 time, and features a melody of quarter notes: C4, D4, E4, F4, G4, A4, B4, A4, G4, F4, E4, D4, C4. The piano accompaniment consists of two staves. The upper piano staff is in treble clef and plays a simple harmonic accompaniment of quarter notes: C4, D4, E4, F4, G4, A4, B4, A4, G4, F4, E4, D4, C4. The lower piano staff is in bass clef and plays a simple harmonic accompaniment of quarter notes: C3, D3, E3, F3, G3, A3, B3, A3, G3, F3, E3, D3, C3. The score is written in a simple, clean style with no dynamics or articulation marks.

Rests may be used in various styles.

```
\layout {
  indent = 0.0
  raggedright = ##t
}

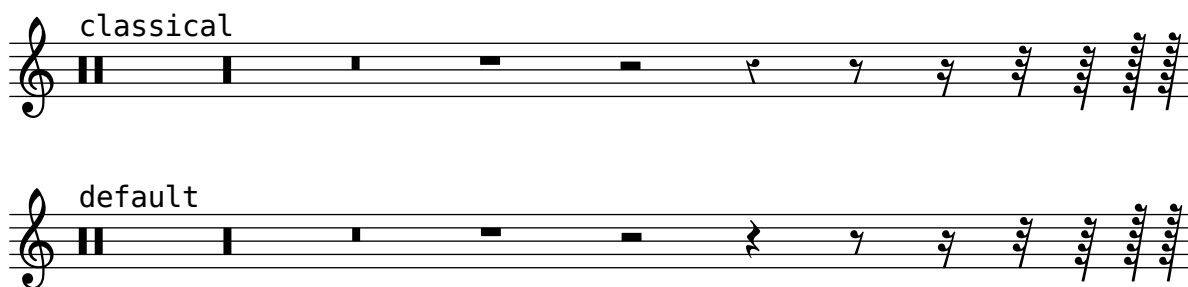
\context Staff \relative c {
  \set Score.timing = ##f
  \override Staff.Rest #'style = #'mensural
  r\maxima^\markup \typewriter { mensural }
  r\longa r\breve r1 r2 r4 r8 r16 r32 r64 r128 r128
  \bar ""

  \override Staff.Rest #'style = #'neomensural
  r\maxima^\markup \typewriter { neomensural }
  r\longa r\breve r1 r2 r4 r8 r16 r32 r64 r128 r128
  \bar ""

  \override Staff.Rest #'style = #'classical
  r\maxima^\markup \typewriter { classical }
  r\longa r\breve r1 r2 r4 r8 r16 r32 r64 r128 r128
  \bar ""

  \override Staff.Rest #'style = #'default
  r\maxima^\markup \typewriter { default }
  r\longa r\breve r1 r2 r4 r8 r16 r32 r64 r128 r128
}
```

The image shows two staves of music. The top staff is labeled 'mensural' and features a treble clef with a common time signature 'C'. The notation consists of vertical stems with horizontal flags indicating pitch and duration. The bottom staff is labeled 'neomensural' and also features a treble clef with a common time signature 'C'. The notation is similar to the mensural staff but uses a different set of symbols for the notes, which are more compact and uniform in shape.



‘rhythmic-slashes.ly’

In "simple" lead-sheets, sometimes no actual notes are written, instead only "rhythmic patterns" and chords above the measures are noted giving the structure of a song. Such a feature is for example useful while creating/transcribing the structure of a song and also when sharing lead sheets with guitarists or jazz musicians.

The standard support for this is described in section "Measure repeats", but then the first beat has to be an ordinary note or rest.

This example shows two solutions to this problem, by redefining ordinary rests to be printed as slashes. (If the duration of each beat is not a quarter note, replace the r4 in the definitions by a rest of the appropriate duration).

% Macro to print single slash

```
rs = {
  \once \override Rest #'stencil = #ly:percent-repeat-item-interface::beat-slash
  \once \override Rest #'thickness = #'0.48
  \once \override Rest #'slope = #'1.7
  r4
}
```

% Function to print a specified number of slashes

```
comp = #(define-music-function (parser location count) (integer?)
  #{
    \override Rest #'stencil = #ly:percent-repeat-item-interface::beat-slash
    \override Rest #'thickness = #'0.48
    \override Rest #'slope = #'1.7
    \repeat unfold $count { r4 }
    \revert Rest #'stencil
  })
```

```
\score{
  \relative c'{
    c d e f | \rs \rs \rs \rs | \comp #4 |
  }
}
```



‘skips-in-lyric-mode-2.ly’

Although you can't use 's' in lyric mode (it's taken to be a literal s, not a space) you can use either "" or ..

So for example:

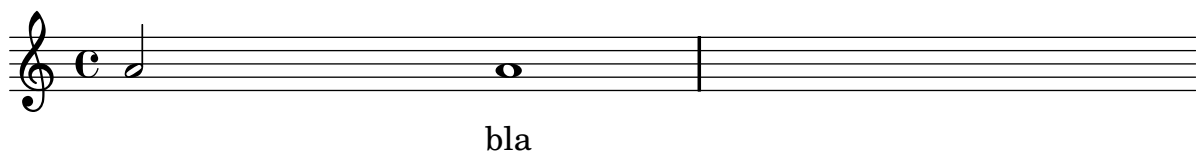
```
<<
\relative c'' { a4 b c d }
\new Lyrics \lyricmode { a4 _2 gap4 }
>>
```



‘skips-in-lyric-mode.ly’

The `s` syntax is only available in note mode and chord mode. In other situations, for example, when entering lyrics, you should use the `\skip` command.

```
<<
\relative { a'2 a1 }
\new Lyrics \lyricmode { \skip 2 bla1 }
>>
```



Expressive marks

‘adding-beams,-slurs,-ties-etc.-when-using-tuplet-and-non-tuplet-rythms..ly’

LilyPond syntax can involve many unusual placements for parentheses, brackets etc., which might sometimes have to be interleaved. 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 superpose manual beaming, manual slurs, ties, and phrasing slurs, while using tuplet sections (enclosed with curly braces).

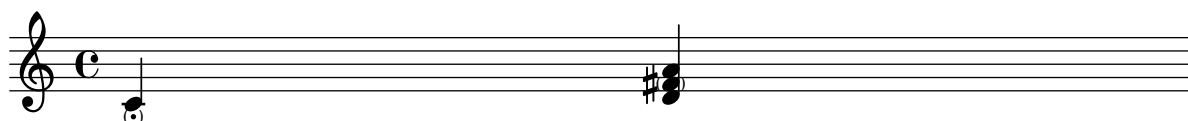
```
{
  r16[ g16 \times 2/3 {r16 e'8} ]
  g16( a \times 2/3 {b d e'}) }
  g8[( a \times 2/3 {b d'}) e'~]]
  \times 4/5 {e'32\ ( a b d' e') a'4.\}
}
```



‘adding-parentheses-around-an-expressive-mark.ly’

The parenthesize function is a special tweak that encloses objects in parentheses. The associated grob is Score.ParenthesesItem.

```
\relative {
  c4 -\parenthesize -.
  \override ParenthesesItem #'padding = #0.1
  <d \parenthesize fis a>
}
```



‘breathing-sign.ly’

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

```
{
  %% Modern notation:
  \new Staff {
    \relative c'' {
      \key es \major \time 3/4

      %% this bar contains no \breathe
      <<
        { g4 as g } \\\
        { es4 bes es }
      >>
    }
  }
}
```



```

>> |

%% 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 =
#(make-musicglyph-markup "scripts.rvarcomma")
<<
  { g4 as g } \\\
  { es4 \breathe bes es }
>> |

%% wedge
\override BreathingSign #'text =
#(make-musicglyph-markup "scripts.upbow")
es8 d es f g8 \breathe f |

%% caesura
\override BreathingSign #'text =
#(make-musicglyph-markup "scripts.caesura.curved")
es8[ d] \breathe es[ f g f] |
es2 r4 \bar "||" \break
}
}
}

```



```

{
  \relative c' {
    << {
      \dynamicUp
      \override DynamicLineSpanner #'staff-padding = #4
      r2 r16 c'8.\pp r4
    } \ {
      \override DynamicLineSpanner #'layer = #0
      des,2~\mf \<
      \override TextScript #'layer = #2
      des16_\markup {
        \postscript {"1.9 -4.5 translate 5 4 scale 1 setgray 0 0 moveto 0 1
          lineto 1 1 lineto 1 0 lineto 0 0 lineto fill" }
        r8. des4~ des16-> \sff
      }
    }
    >>
  }
  \layout { ragged-right = ##t}
}

```



‘caesura-railtracks-with-fermata.ly’

A caesura is sometimes denoted with a double "railtracks" breath mark with a fermata sign positioned over the top of the railtracks. This snippet should present an optically pleasing combination of railtracks and a fermata.

```

{
  \context Voice {
    c''2.
    % use some scheme code to construct the symbol
    \override BreathingSign #'text = #(markup #:line
      (:#musicglyph "scripts.caesura.curved"
        #:translate (cons -1.75 1.6)
        #:musicglyph "scripts.ufermata"
      ))
    \breathe c''4
    % set the breathe mark back to normal
    \revert BreathingSign #'text
    c''2. \breathe c''4
    \bar "|."
  }
}

```



‘center-text-below-hairpin-dynamics.ly’

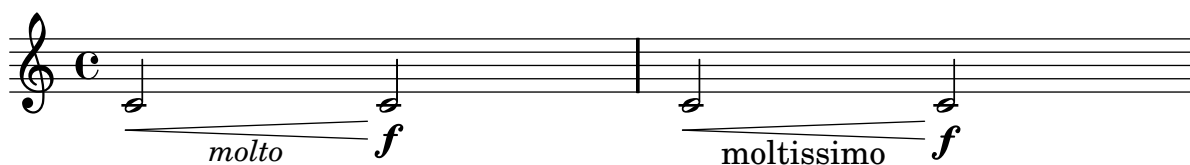
This example provides a function to typeset hairpin (de)crescendo with some additional text below it, such as "molto" or "poco".

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

```
hairpinWithCenteredText = #(define-music-function
                             (parser location text) (markup?)
  #{
    \override Voice.Hairpin #'stencil = #(lambda (grob)
      (ly:stencil-aligned-to
        (ly:stencil-combine-at-edge
          (ly:stencil-aligned-to (ly:hairpin::print grob) X CENTER)
          Y
          DOWN
          (ly:stencil-aligned-to (ly:text-interface::print grob) X CENTER))
        X LEFT))
    \override Voice.Hairpin #'text = $text
  #})

hairpinMolto = \hairpinWithCenteredText \markup {\italic "molto"}
hairpinMore  = \hairpinWithCenteredText \markup {\bigger "moltissimo"}

\new Staff {
  \hairpinMolto c'2\< c'2\f
  \hairpinMore  c'2\< c'2\f
}
```

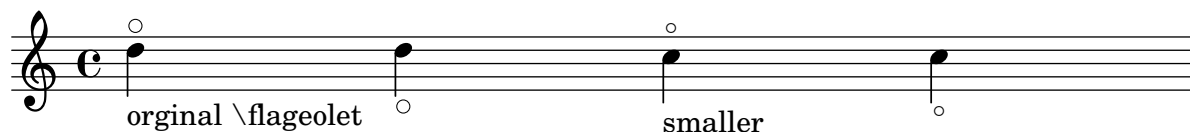


‘changing--flageolet-mark-size.ly’

To make the \flageolet circle smaller you can use the following scheme code (found in the Lilypond-user-archive)

```
F = #(let ((m (make-music 'ArticulationEvent
                          'articulation-type "flageolet"))
  (set! (ly:music-property m 'tweaks)
    (acons 'font-size -3
      (ly:music-property m 'tweaks)))
  m)

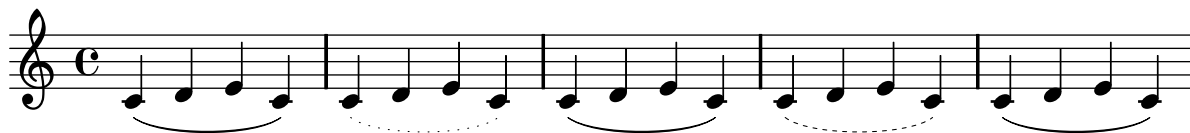
\relative c' { d4^\flageolet_\markup {"original \flageolet "} d4_\flageolet
  c4^F_\markup {smaller } c4_\F
}
```



‘changing-the-appearance-of-a-slur-from-solid-to-dotted-or-dashed.ly’

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

```
\score{
  \relative c'{
    c( d e c) |
    \slurDotted
    c( d e c) |
    \slurSolid
    c( d e c) |
    \slurDashed
    c( d e c) |
    \slurSolid
    c( d e c) |
  }
  \layout{ raggedright=##t }
}
```



‘combining-dynamics-with-markup-texts.ly’

Some dynamics may involve text indications (such as "più forte", "piano subito", etc.). They can be produced using a \markup bloc.

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

```
piuf = \markup { \italic "molto" \dynamic "f" }
```

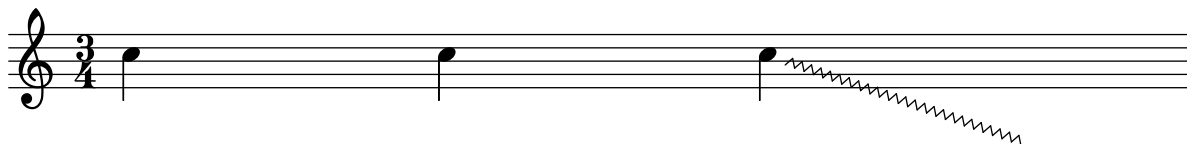
```
\relative c''{
  c-\piuf
  c
  c2\< c2\!

  c2\< c2\!
}
```



‘contemporary-glissando.ly’

Use a glissando without final note. Contemporary glissando !



Use **cadenzaOn** **glissando** **hideNotes**
unHideNotes **cadenzaOff,** **skip and \bar**

‘creating-real-parenthesized-dynamics.ly’

Although the easiest way to add parenthesis to a dynamic mark is to use a `\markup` block, this method has a downside: the created objects will 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 "Markup programmer interface"), combined with the `make-dynamic-script` function. This way, the markup will be regarded as a dynamic, and therefore will remain compatible with commands such as `\dynamicUp` or `\dynamicDown`.

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

```
parenF = #(make-dynamic-script (markup #:line(#:normal-text #:italic
#:fontsize 2 "(" #:hspace -0.8 #:dynamic "f" #:normal-text #:italic
#:fontsize 2 ")") ))
```

```
\score
{
    { c''\parenF c'' c'' \dynamicUp c''\parenF }
}
```



‘creating-text-spanners.ly’

The `\startTextSpan` and `\stopTextSpan` commands give you the ability to create text spanners as easily as pedals indications or octavations. Override some properties of the `TextSpanner` object to modify its output.

```
\relative c''{
    \override TextSpanner #'edge-text = #("bla" . "blu")
    a \startTextSpan
    b c
    a \stopTextSpan

    \override TextSpanner #'dash-period = #2
    \override TextSpanner #'dash-fraction = #0.0
    a \startTextSpan
    b c
```

```

a \stopTextSpan

\revert TextSpanner #'style
\override TextSpanner #'style = #'dashed-line \override TextSpanner #'bound-details #'
\override TextSpanner #'bound-details #'right #'text = \markup { \draw-line #'(0 . -2) }

a \startTextSpan
b c
a \stopTextSpan

\set Staff.middleCPosition = #-13

\override TextSpanner #'dash-period = #10
\override TextSpanner #'dash-fraction = #.5
\override TextSpanner #'thickness = #10
a \startTextSpan
b c
a \stopTextSpan
\set Staff.middleCPosition = #-6
}

```



‘line-arrows.ly’

Arrows can be applied to text-spanners and line-spanners (such as the Glissando)

```

\relative c' {
  \override TextSpanner #'bound-padding = #1.0
  \override TextSpanner #'dash-fraction = #'()
  \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 gis8 a4 b4\glissando
  b,4 | g' c\stopTextSpan c
}

```



`'piano-template-with-centered-dynamics.ly'`

Many piano scores have the dynamics centered between the two staves. This requires a bit of tweaking to implement, but since the template is right here, you don't have to do the tweaking yourself.

```
upper = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

  a b c d
}

lower = \relative c {
  \clef bass
  \key c \major
  \time 4/4

  a2 c
}

dynamics = {
  s2\fff\> s4
  s\!\pp
}

pedal = {
  s2\sustainDown s2\sustainUp
}

\score {
  \new PianoStaff <<
    \new Staff = "upper" \upper
    \new Dynamics = "dynamics" \dynamics
    \new Staff = "lower" <<
      \clef bass
      \lower
    >>
    \new Dynamics = "pedal" \pedal
  >>
  \layout {
    \context {
      \type "Engraver_group"
      \name Dynamics
      \alias Voice % So that \cresc works, for example.
      \consists "Output_property_engraver"

      \override VerticalAxisGroup #'minimum-Y-extent = #'(-1 . 1)
      \override DynamicLineSpanner #'Y-offset = #0
      pedalSustainStrings = #'("Ped." "*Ped." "*")
    }
  }
}
```

```

pedalUnaCordaStrings = #("una corda" "" "tre corde")

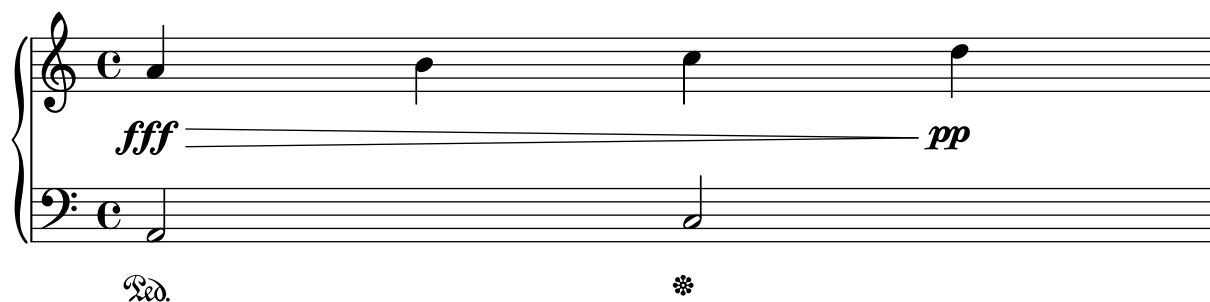
\consists "Piano_pedal_engraver"
\consists "Script_engraver"
\consists "Dynamic_engraver"
\consists "Text_engraver"

\override TextScript #'font-size = #2
\override TextScript #'font-shape = #'italic

\consists "Skip_event_swallow_translator"

\consists "Axis_group_engraver"
}
\context {
  \PianoStaff
  \accepts Dynamics
}
}
\score {
  \new PianoStaff <<
    \new Staff = "upper" << \upper \dynamics >>
    \new Staff = "lower" << \lower \dynamics >>
    \new Dynamics = "pedal" \pedal
  >>
  \midi {
    \context {
      \type "Performer_group"
      \name Dynamics
      \consists "Piano_pedal_performer"
    }
    \context {
      \PianoStaff
      \accepts Dynamics
    }
  }
}

```



Repeats

‘positioning-segno-and-coda-with-line-break.ly’

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's a variation documented in this snippet, where the coda will remain on the same line.

```
{
  \clef treble
  \key g \major
  \time 4/4
  \relative c'' {
    \repeat unfold 2 {
      | c4 c c c
    }

    % Set segno sign as rehearsal mark and adjust size if needed
    % \once \override Score.RehearsalMark #'font-size = #3
    \mark \markup { \musicglyph #"scripts.segno" }
    \repeat unfold 2 {
      | c4 c c c
    }

    % Set coda sign as rehearsal mark and adjust size if needed
    \once \override Score.RehearsalMark #'font-size = #4
    \mark \markup { \musicglyph #"scripts.coda" }
    \repeat unfold 2 {
      | c4 c c c
    }

    % Should Coda be on a new line?
    % Coda NOT on new line: use \nobreak
    % Coda on new line: DON'T use \nobreak
    % \noBreak

    \bar "||"

    % Set segno sign as rehearsal mark and adjust size if needed
    \once \override Score.RehearsalMark #'break-visibility = #begin-of-line-invisible
    % \once \override Score.RehearsalMark #'font-size = #3
    \mark \markup { \musicglyph #"scripts.segno" }

    % Here begins the trickery!
    % \cadenzaOn will suppress the bar count and \stopStaff removes the staff lines.
    \cadenzaOn
    \stopStaff
    % Some examples of possible text-displays

    % text line-aligned
    % =====
    % Move text to the desired position
    % \once \override TextScript #'extra-offset = #'( 2 . -3.5 )
```

```

% | s1*0^\markup { D.S. al Coda } }

% text center-aligned
% =====
% Move text to the desired position
% \once \override TextScript #'extra-offset = #'( 6 . -5.0 )
% | s1*0^\markup { \center-align { D.S. "al Coda" } }

% text and symbols center-aligned
% =====
% Move text to the desired position and tweak spacing for optimum text alignment
\once \override TextScript #'extra-offset = #'( 8 . -5.5 )
\once \override TextScript #'word-space = #1.5
| s1*0^\markup { \center-align { "D.S. al Coda" \line { \musicglyph #"scripts.coda"

% Increasing the unfold counter will expand the staff-free space
\repeat unfold 4 {
  s4 s4 s4 s4
  \bar ""
}
% Resume bar count and show staff lines again
\startStaff
\cadenzaOff

% Should Coda be on new line?
% Coda NOT on new line: DON'T use \break
% Coda on new line: use \break
\break

% Show up, you clef and key!
\once \override Staff.KeySignature #'break-visibility = #end-of-line-invisible
\once \override Staff.Clef #'break-visibility = #end-of-line-invisible

% Set coda sign as rehearsal mark and adjust size and position

% Put the coda sign ontop of the (treble-)clef dependend on coda's line-position

% Coda NOT on new line, use this:
% \once \override Score.RehearsalMark #'extra-offset = #'( -2 . 1.75 )

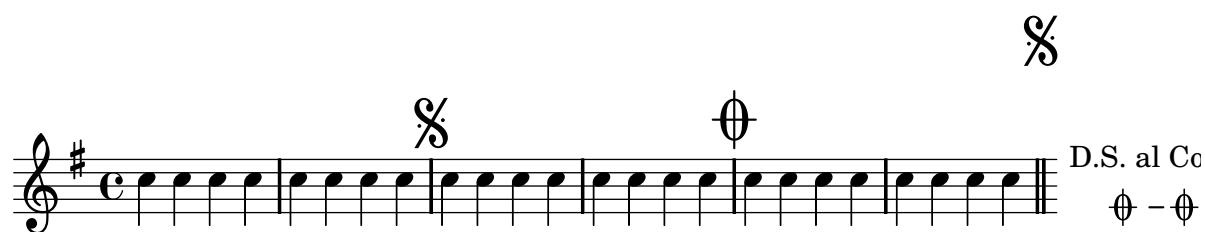
% Coda on new line, use this:
\once \override Score.RehearsalMark #'extra-offset = #'( -8.42 . 1.75 )

\once \override Score.RehearsalMark #'font-size = #5
\mark \markup { \musicglyph #"scripts.coda" }

% The coda
\repeat unfold 5 {
  | c4 c c c
}
\bar"|. "
}

```

}



†

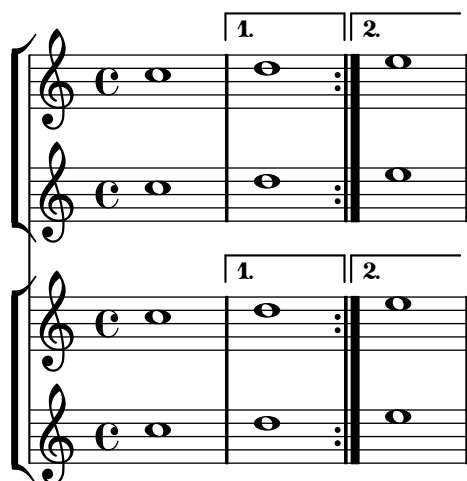


'volta-multi-staff.ly'

By adding `Volta_engraver`, repeat brackets can be put over staves other than the topmost one in a score.

```
vmus = \relative c'' {
  \repeat volta 2 c1 \alternative { d e }
}
```

```
<<
  \new StaffGroup <<
    \context Staff \vmus
    \new Staff \vmus
  >>
  \new StaffGroup <<
    \new Staff \with { \consists Volta_engraver }
    \vmus
    \new Staff \vmus
  >>
>>
```



Simultaneous notes

‘changing-an-individual-notes-size-in-a-chord.ly’

Individual noteheads 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 notehead).

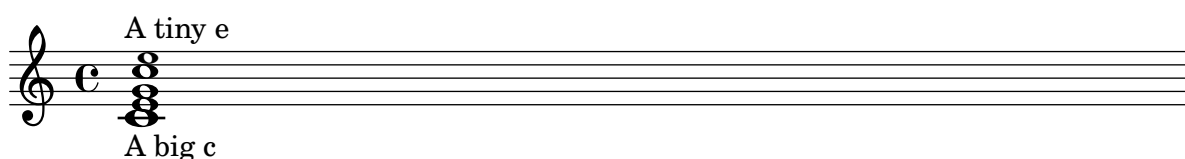
The code for the chord example shown:

```
\header{
  title = "Modify an individual notehead's size in a chord"
}

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

\score{
  \Notes
}
```

Modify an individual notehead's size in a chord



‘clusters.ly’

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

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

fragment = \relative c' {
  c4 f4 <e d'>4
  <g a>8 <e a> a4 c2 <d b>4 e4
  c4
}

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



`'combining-two-parts-on-the-same-staff.ly'`

The part combiner tool (`\partcombine` command) allows you to combine different parts on a same Staff. You can choose whether you want or don't want to add texts such as "solo" or "a2", by defining the `printPartCombineTexts` property.

For vocal scores (hymns), there is no need to add "solo"/"a2" texts, so they should be switched off. However, you'd better not use it if there are any solos, as they won't be indicated. In such cases, you may simply want to use standard LilyPond polyphony.

This snippet presents the three ways two parts can be printed on a same staff : standard polyphony, `\partcombine` whitout texts, and `\partcombine` with texts.

```
musicUp = {
  \time 4/4
  \relative c'' {
    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 a
  }
}

\score{
  \new Staff {
    \set Staff.instrumentName = "Standard polyphony  "
    << \musicUp  \ \musicDown >>
  }

  \layout{
    indent = 6.0\cm
  }
}

\score{
  \context Staff {
    \set Staff.instrumentName = "PartCombine without texts  "
    \partcombine \musicUp \musicDown
  }
  \layout{
    indent = 6.0\cm
    \context {
      \Voice
      printPartCombineTexts = ##f
    }
  }
}
```

```

\score{
  \context Staff {
    \set Staff.instrumentName = "PartCombine with texts "
    \partcombine \musicUp \musicDown
  }
  \layout{
    indent = 6.0\cm
    \context {
      \Voice
      printPartCombineTexts = ##t
    }
  }
}

```

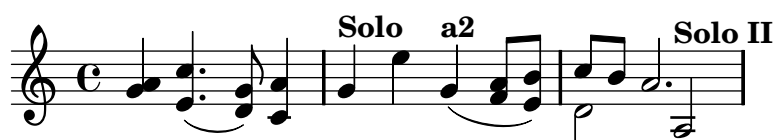
Standard polyphony



PartCombine without texts



PartCombine with texts



Staff notation

‘adding-ambiti-per-voice.ly’

Ambits can be added per voice. In that case, the ambitus must be moved manually to prevent collisions.

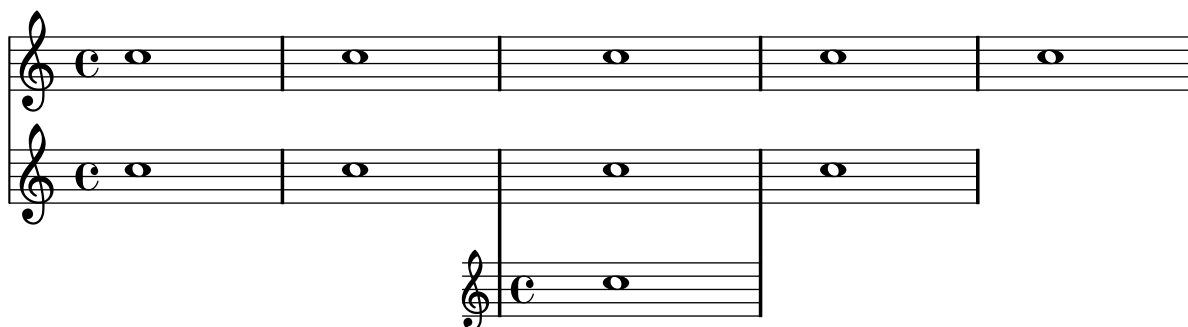
```
\new Staff <<
  \new Voice \with {
    \consists "Ambitus_engraver"
  } \relative c'' {
    \override Ambitus #'X-offset = # 2.0
    \voiceOne
    c4 a d e f1
  }
  \new Voice \with {
    \consists "Ambitus_engraver"
  } \relative c' {
    \voiceTwo
    es4 f g as b1
  }
>>
```



‘adding-an-extra-staff.ly’

You can add (possibly temporarily) an extra staff after the beginning of a piece.

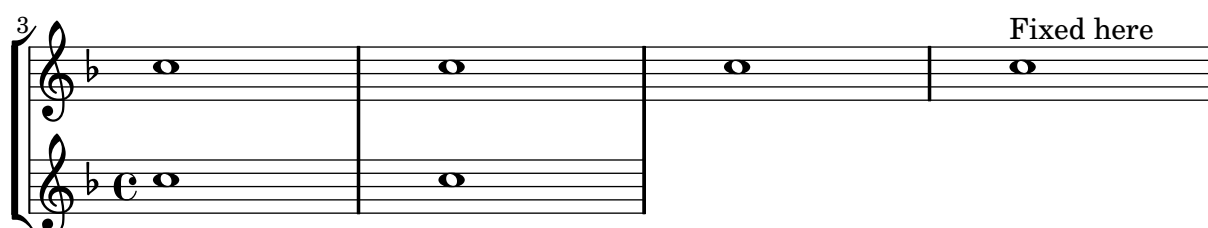
```
\score {
  <<
    \new Staff \relative c'' { c1 c c c c }
    \new StaffGroup \relative c'' {
      \new Staff
      c1 c
      << c1 \new Staff { c1 } >>
      c
    }
  >>
  \layout {
    \context{
      \Score
    }
  }
}
```

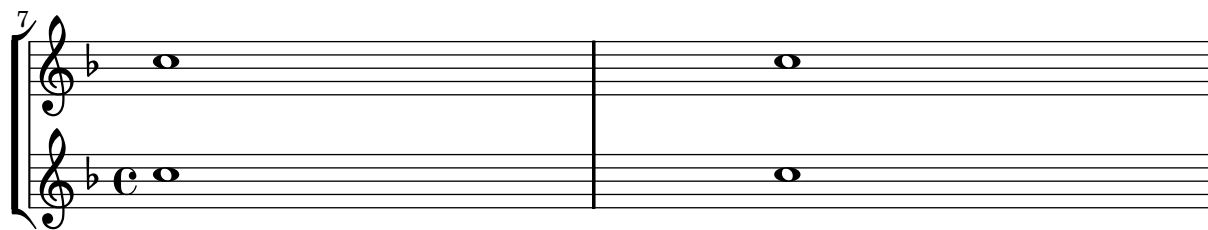


‘adding-and-extra-staff-at-a-line-break.ly’

When adding a new Staff at a line break, LilyPond will unfortunately add some extra space at the end of the line before the break (to fit in a key signature change, which will never be printed anyway). The workaround is to add a setting of `Staff.explicitKeySignatureVisibility` as is shown in the example. In version 2.10 and earlier, you have to do a similar setting for the time signatures, see 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
      c1 c
    }
    >>
    c1 c^"Fixed here" \break
    << { c1 c }
    \new Staff {
      \once \set Staff.explicitKeySignatureVisibility = #end-of-line-invisible
      %The next line is not needed in 2.11.x or later:
      \once \override Staff.TimeSignature #'break-visibility = #end-of-line-invisible
      \key f \major
      c1 c
    }
    >>
  }
}
```





‘changing-the-number-of-lines-in-a-staff.ly’

The number of lines in a staff may be changed by overriding `line-count` in the properties of `StaffSymbol`.

```
upper = \relative c'' {
  c1 d e f
}
```

```
lower = \relative c {
  c1 b a g
}
```

```
\score {
  \context PianoStaff <<
    \new Staff <<
      \upper
    >>
    \new Staff {
      \override Staff.StaffSymbol #'line-count = #4
      \clef bass
      \lower
    }
  >>
}
```



‘changing-the-staff-size.ly’

In order to change staff sizes, both `staff-space` and `fontSize` must be scaled.

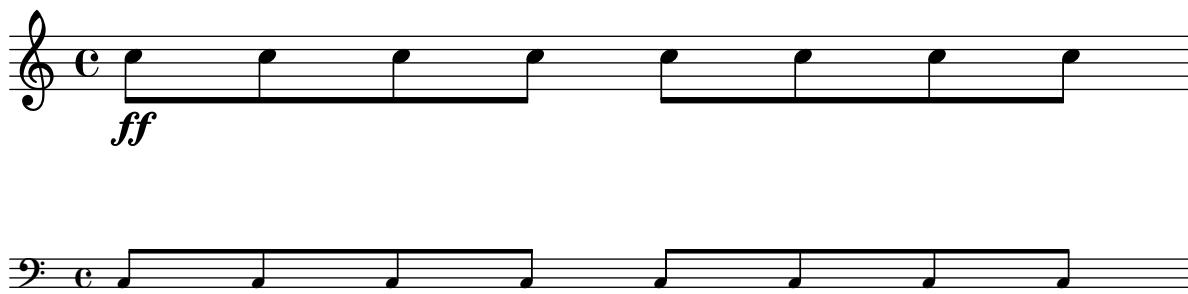
```
{
  \new Staff \relative c'' { \dynamicDown c8 \ff c c c c c c c }
}

{
  \new Staff \with {
    fontSize = #-3
    \override StaffSymbol #'staff-space = #(magstep -3)
  }
}
```

```

\clef bass
c8 c c c c c c c
}
}

```



‘clefs-commonly-tweaked-properties.ly’

The command `\clef "treble_8"` is equivalent to setting `clefGlyph`, `clefPosition` (which controls the Y position of the clef), `middleCPosition` and `clefOctavation`. A clef is printed when any of these properties are changed.

Note that changing the glyph, the position of the clef, or the octavation, does not in itself change the position of subsequent notes on the staff: the position of middle C must also be specified to do this. The positional parameters are relative to the staff centre line, positive numbers displacing upwards, counting 1 for each line and space. The `clefOctavation` value would normally be set to 7, -7, 15 or -15, but other values are not invalid.

When a clef change takes place at a line break the new clef symbol is printed at both the end of the previous line and the beginning of the new line by default. If the warning clef at the end of the previous line is not required it can be suppressed by setting the `explicitClefVisibility` Staff property to the value `end-of-line-invisible`. The default behaviour can be recovered with `\unset Staff.explicitClefVisibility`.

The following examples show the possibilities when setting these properties manually. On the first line, the manual changes preserve the standard relative positioning of clefs and notes, whereas on the second line, they do not.

```

{
% The default treble clef
c'1
% The standard bass clef
\set Staff.clefGlyph = #"clefs.F"
\set Staff.clefPosition = #2
\set Staff.middleCPosition = #6
c'
% The baritone clef
\set Staff.clefGlyph = #"clefs.C"
\set Staff.clefPosition = #4
\set Staff.middleCPosition = #4
c'
% The standard choral tenor clef
\set Staff.clefGlyph = #"clefs.G"
\set Staff.clefPosition = #-2
\set Staff.clefOctavation = #-7

```

```

\set Staff.middleCPosition = #1
c'
% A non-standard clef
\set Staff.clefPosition = #0
\set Staff.clefOctavation = #0
\set Staff.middleCPosition = #-4
c' \break

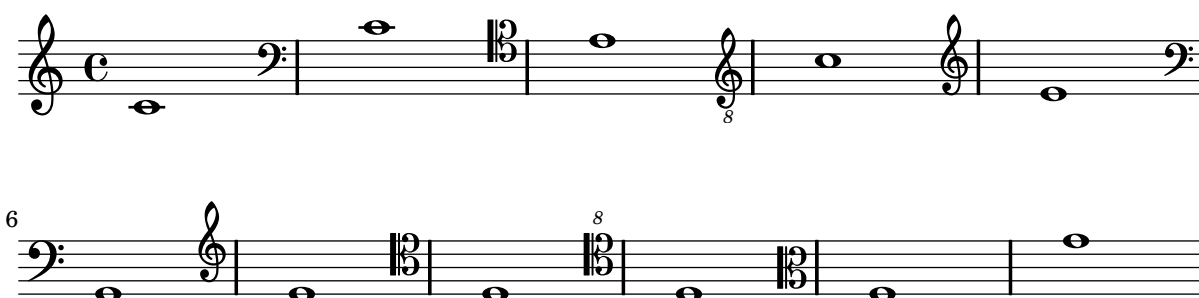
% The following clef changes do not preserve
% the normal relationship between notes and clefs:

\set Staff.clefGlyph = #"clefs.F"
\set Staff.clefPosition = #2
c'
\set Staff.clefGlyph = #"clefs.G"
c'
\set Staff.clefGlyph = #"clefs.C"
c'
\set Staff.clefOctavation = #7
c'
\set Staff.clefOctavation = #0
\set Staff.clefPosition = #0
c'

% Here we go back to the normal clef:

\set Staff.middleCPosition = #4
c'
}

```



‘creating-blank-staves.ly’

To create blank staves, you must generate empty measures, removing also from the `Score` context the `Bar_number_engraver`, and from the `Staff` context the `Time_signature_engraver`, the `Clef_engraver` and the `Bar_engraver`.

```

#(set-global-staff-size 20)

```

```

\score {
  {
    \repeat unfold 12 { s1 \break }
  }
}

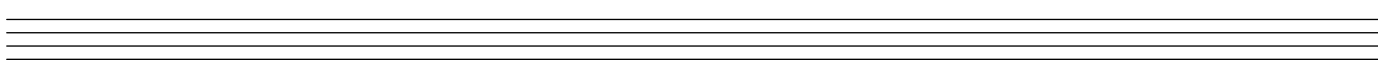
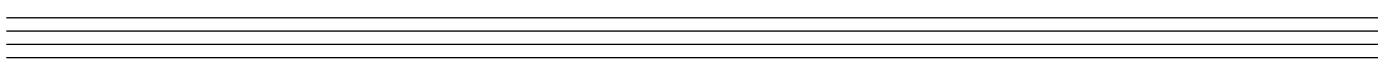
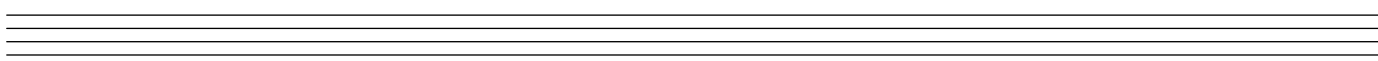
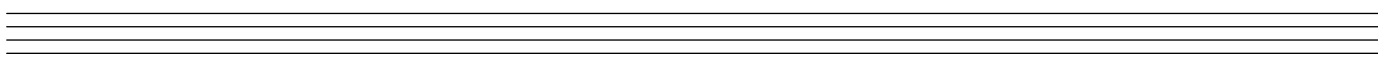
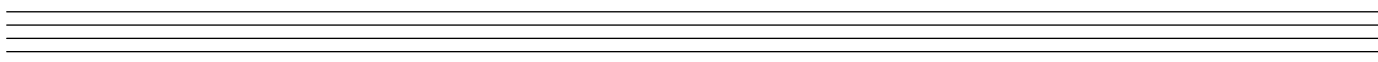
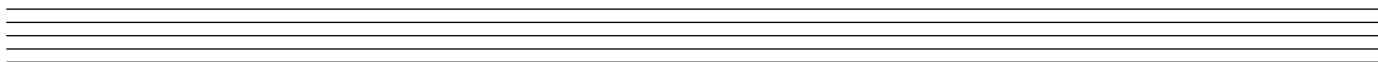
```

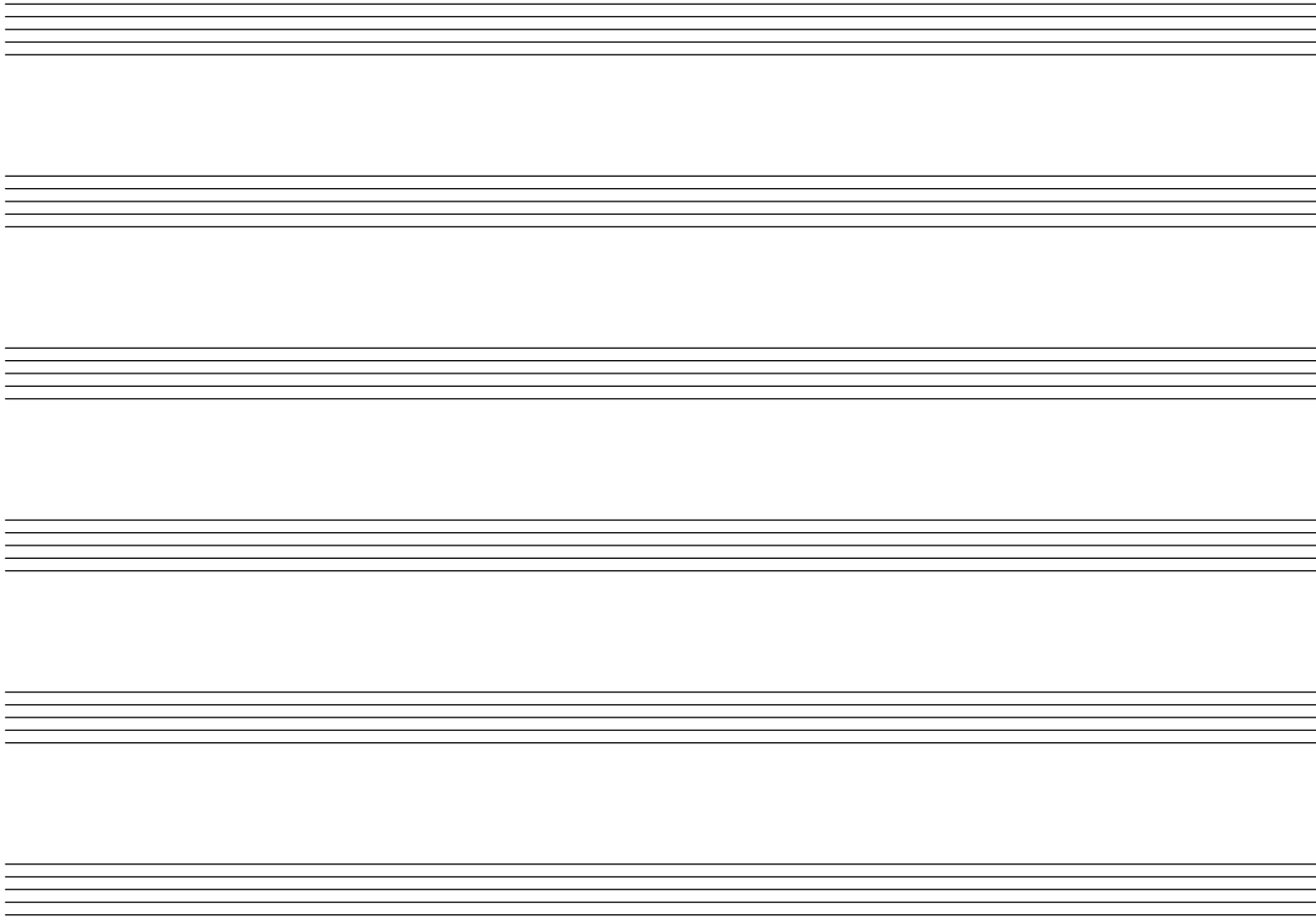
```

\layout {
  indent = 0\in
  \context {
    \Staff
    \remove Time_signature_engraver
    \remove Clef_engraver
    \remove Bar_engraver
  }
  \context {
    \Score
    \remove Bar_number_engraver
  }
}

\paper {
  #(set-paper-size "letter")
  raggedlastbottom = ##f
  linewidth = 7.5\in
  leftmargin = 0.5\in
  bottommargin = 0.25\in
  topmargin = 0.25\in
}

```

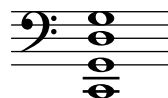




The `\markup` command is quite versatile. In this snippet, it contains a `\score` bloc instead of texts or marks.

```
\relative {
  \time 4/8
  \times 2/3 { c'8 d e } \times 2/3 {c d e}
  \times 2/3 { c8 d e } \times 2/3 {c d e}
  g8 a8 g8 a
  g8 a8 g8 a
}
```

Originalstimmung:



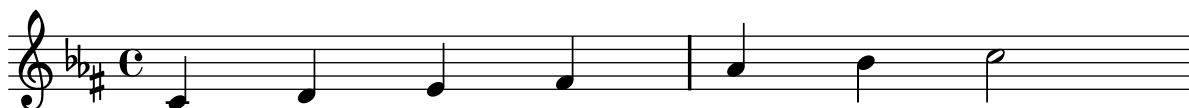
`'non-traditional-key-signatures.ly'`

The commonly used `\key` command sets the `keySignature` property, in the `Staff` context. However, non-standard key signatures can be specified by setting this property directly. The format of this command is a list: `\set Staff.keySignature = #'((octave . step) . alter) ((octave . step) . alter) ...)` where, for each element in the list, octave specifies the octave (0 being the octave from middle C to the B above), step specifies the note within the octave (0 means C and 6 means B), and alter is `,SHARP` `,FLAT` `,DOUBLE-SHARP` etc. (Note the leading comma.)

However, for each item in the list, you can also use the alternative format `(step . alter)`, which specifies that the same alteration should hold in all octaves.

Here is an example of a possible key signature for generating a whole-tone scale:

```
\relative c' {
  \set Staff.keySignature =
    #'(((0 . 3) . ,SHARP) ((0 . 5) . ,FLAT) ((0 . 6) . ,FLAT))
  c d e fis aes bes c2
}
```



`'quoting-another-voice-with-transposition.ly'`

Quotations take into account the transposition of both source and target. In this example, all instruments play sounding central C, the target is an instrument in F. The target part may be `\transposed`. In this case, all the pitches (including the quoted ones) will be transposed as well.

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

\addQuote sax {
  \transposition es'
  a8 a a a a a a
}

quoteTest = {
  \transposition f % french horn

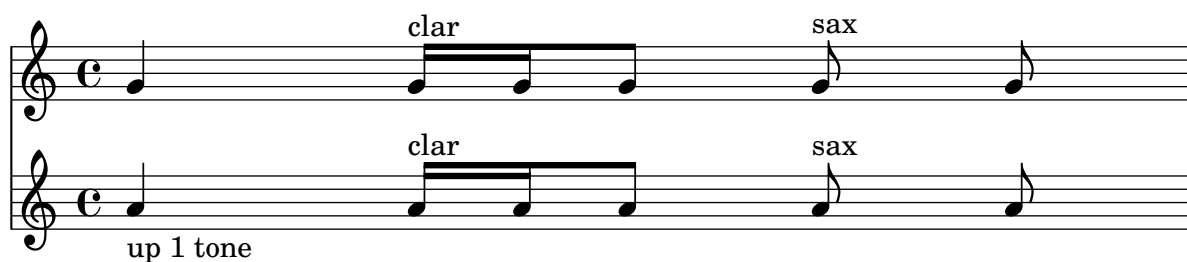
  g'4
  << \quoteDuring #"clarinet" { \skip 4 } s4^"clar" >>
  << \quoteDuring #"sax" { \skip 4 } s4^"sax" >>
}

<< \quoteTest
  \new Staff
  << \transpose c' d' \quoteTest
```

```

s4_"up 1 tone"
>>
>>

```



```

'quoting-another-voice.ly'

```

With `\quote`, fragments of previously entered music may be quoted. `quotedEventTypes` will determines what things are quoted. In this example, a 16th rest is not quoted, since `rest-event` is not in `quotedEventTypes`.

```

quoteMe = \relative c' { fis4 r16 a8.-> b4-\ff c }

```

```

\addQuote quoteMe \quoteMe
original = \relative c'' { c8 d s2 es8 gis8 }

```

```

<<
  \new Staff {
    \set Staff.instrumentName = "quoteMe"
    \quoteMe
  }
  \new Staff {
    \set Staff.instrumentName = "orig"
    \original
  }
  \new Staff \relative c'' <<
    \set Staff.instrumentName = "orig+quote"
    \set Staff.quotedEventTypes = #'(note-event articulation-event)
    \original
    \new Voice {
      s4
      \set fontSize = #-4
      \override Stem #'length-fraction = #(magstep -4)
      \quoteDuring #"quoteMe" { \skip 2. }
    }
  }
>>
>>

```



‘time-signature-in-parentheses.ly’

You may put the time signature in parentheses.

```
tsMarkup = \markup {
  \number {
    \bracket \column { "2" "4" }
  }
}

\score {
  \relative c'' {

% FIXME: Gee, it doesn't work with 2.10 -vv

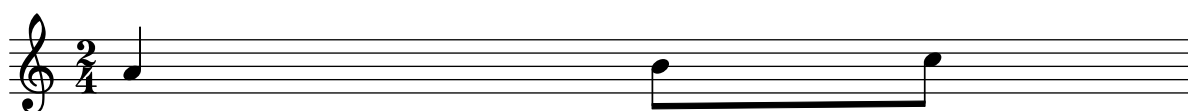
%{

  \override Staff.TimeSignature #'print-function = #Text_interface::print
  \override Staff.TimeSignature #'text = #tsMarkup

%}

  \time 2/4

  a4 b8 c |
}
}
```



‘volta-multi-staff.ly’

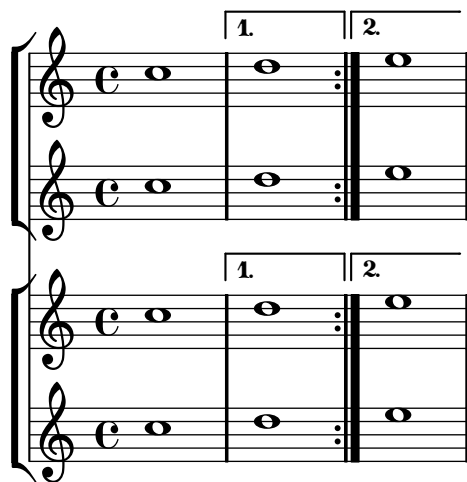
By adding Volta_engraver, repeat brackets can be put over staves other than the topmost one in a score.

```
vmus = \relative c'' {
  \repeat volta 2 c1 \alternative { d e }
}

<<
  \new StaffGroup <<
    \context Staff \vmus
```



```
\new Staff \vmus
>>
\new StaffGroup <<
  \new Staff \with { \consists Volta_engraver }
    \vmus
  \new Staff \vmus
>>
>>
```



Editorial and educational use

‘applying-noteheads-styles-depending-on-the-step-of-the-scale.ly’

The `shapeNoteStyles` property gives you the ability to define various note heads styles for each step of the scale (as defined by the key signature or the "tonic" property).

This property requires a set of symbols, which can be purely arbitrary (geometrical expressions such as triangle, cross, xcircle etc. are allowed) or based on old American engraving tradition (you can use some latin note names as well).

That said, if you’re trying to imitate old American song books, you may also want to try LilyPond’s predefined note heads styles, through shortcut commands such as `\aikenHeads` or `\sacredHarpHeads`.

This example shows different ways to obtain shape note heads, and demonstrates the ability to transpose a melody without losing the correspondance between harmonic functions and note heads styles.

```
fragment = {
  \key c \major
  c1 d e f g a b c
  \break
}

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

    \relative {
      \set shapeNoteStyles = ##(cross triangle fa #f mensural xcircle diamond)
      \fragment
    }
  }
}
```



‘blanking-staff-lines-using-the--whiteout-command.ly’

The `\whiteout` command underlays a white box under a markup. Since staff lines are in a lower layer than most other grobs, this white box will not overlap any other grob.

```
\paper
{
  ragged-right = ##t
}
{
  \override TextScript #'extra-offset = #'(2 . 4)
  c'4-\markup { \whiteout \pad-markup #0.5 foo } c
}
```



‘changing-an-individual-notes-size-in-a-chord.ly’

Individual noteheads 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 notehead).

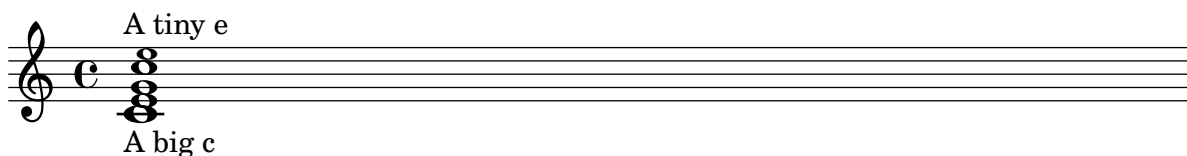
The code for the chord example shown:

```
\header{
  title = "Modify an individual notehead's size in a chord"
}
```

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

```
\score{
  \Notes
}
```

Modify an individual notehead's size in a chord



‘changing-the-appearance-of-a-slur-from-solid-to-dotted-or-dashed.ly’

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

```
\score{
  \relative c'{
    c( d e c) |
    \slurDotted
    c( d e c) |
  }
```

```

\slurSolid
c( d e c) |
\slurDashed
c( d e c) |
\slurSolid
c( d e c) |
}
\layout{ raggedright=###t }
}

```



‘changing-the-default-text-font-family.ly’

The default font families for text can be overridden with `make-pango-font-tree`.

```

\paper {
% change for other default global staff size.
myStaffSize = #20
%{
  run
    lilypond -dshow-available-fonts blabla
    to show all fonts available in the process log.
}%}

#(define fonts
  (make-pango-font-tree "Times New Roman"
                        "Nimbus Sans"
                        "Luxi Mono"
;;                        "Helvetica"
;;                        "Courier"
    (/ myStaffSize 20)))
}

\relative {
  c'^\markup {
    roman: foo \bold bla \italic bar \italic \bold baz
  }
  c'_\markup {
    \override #'(font-family . sans)
    {
      sans: foo \bold bla \italic bar \italic \bold baz
    }
  }
  c'^\markup {
    \override #'(font-family . typewriter)

```

```
{
  mono: foo \bold bla \italic bar \italic \bold baz
}
}
```

The diagram shows a musical staff with three different clef positions. The first clef is a standard treble clef (C-clef on the first line), labeled 'roman: foo **bla** bar *baz*'. The second clef is a C-clef on the second line, labeled 'mono: foo **bla** bar *baz*'. The third clef is a C-clef on the third line, labeled 'sans: foo **bla** bar *baz*'. The text labels are positioned above the staff, and the clef symbols are positioned below the staff.

‘clefs-commonly-tweaked-properties.ly’

The command `\clef "treble_8"` is equivalent to setting `clefGlyph`, `clefPosition` (which controls the Y position of the clef), `middleCPosition` and `clefOctavation`. A clef is printed when any of these properties are changed.

Note that changing the glyph, the position of the clef, or the octavation, does not in itself change the position of subsequent notes on the staff: the position of middle C must also be specified to do this. The positional parameters are relative to the staff centre line, positive numbers displacing upwards, counting 1 for each line and space. The `clefOctavation` value would normally be set to 7, -7, 15 or -15, but other values are not invalid.

When a clef change takes place at a line break the new clef symbol is printed at both the end of the previous line and the beginning of the new line by default. If the warning clef at the end of the previous line is not required it can be suppressed by setting the `explicitClefVisibility` Staff property to the value `end-of-line-invisible`. The default behaviour can be recovered with `\unset Staff.explicitClefVisibility`.

The following examples show the possibilities when setting these properties manually. On the first line, the manual changes preserve the standard relative positioning of clefs and notes, whereas on the second line, they do not.

```
{
% The default treble clef
c'1
% The standard bass clef
\set Staff.clefGlyph = #"clefs.F"
\set Staff.clefPosition = #2
\set Staff.middleCPosition = #6
c'
% The baritone clef
\set Staff.clefGlyph = #"clefs.C"
\set Staff.clefPosition = #4
\set Staff.middleCPosition = #4
c'
% The standard choral tenor clef
\set Staff.clefGlyph = #"clefs.G"
\set Staff.clefPosition = #-2
\set Staff.clefOctavation = #-7
\set Staff.middleCPosition = #1
c'
% A non-standard clef
```

```

\set Staff.clefPosition = #0
\set Staff.clefOctavation = #0
\set Staff.middleCPosition = #-4
c' \break

% The following clef changes do not preserve
% the normal relationship between notes and clefs:

\set Staff.clefGlyph = #"clefs.F"
\set Staff.clefPosition = #2
c'
\set Staff.clefGlyph = #"clefs.G"
c'
\set Staff.clefGlyph = #"clefs.C"
c'
\set Staff.clefOctavation = #7
c'
\set Staff.clefOctavation = #0
\set Staff.clefPosition = #0
c'

% Here we go back to the normal clef:

\set Staff.middleCPosition = #4
c'
}

```



‘coloring-objects.ly’

LilyPond gives you the ability to assign different colors to any grob in your score, such as NoteHeads, Alterations, Beams and so on, by simply overriding the #’color property and choosing your color (over 200 colors are available, see the "List of Colors" Appendix in the Manual).

```

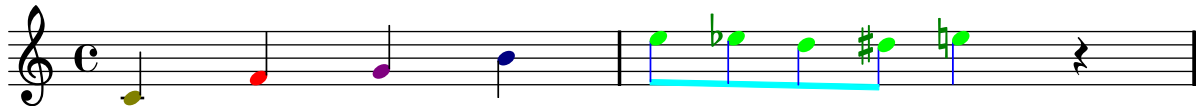
\relative {
  \override Accidental #'color = #darkgreen
  \override Beam #'color = #cyan
  \override NoteHead #'color = #darkyellow
  c4
  \override NoteHead #'color = #red
  f
  \override NoteHead #'color = #darkmagenta

```

```

g
\override NoteHead #'color = #darkblue
b
\override NoteHead #'color = #green
\override Stem #'color = #blue
e8 es d dis e4 r
}

```



‘creating-blank-staves.ly’

To create blank staves, you must generate empty measures, removing also from the `Score` context the `Bar_number_engraver`, and from the `Staff` context the `Time_signature_engraver`, the `Clef_engraver` and the `Bar_engraver`.

```

#(set-global-staff-size 20)

```

```

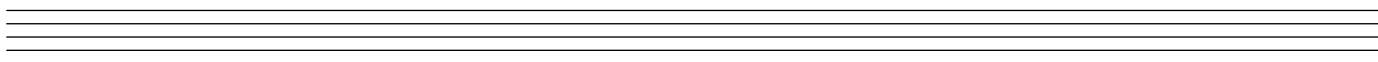
\score {
  {
    \repeat unfold 12 { s1 \break }
  }
  \layout {
    indent = 0\in
    \context {
      \Staff
      \remove Time_signature_engraver
      \remove Clef_engraver
      \remove Bar_engraver
    }
    \context {
      \Score
      \remove Bar_number_engraver
    }
  }
}

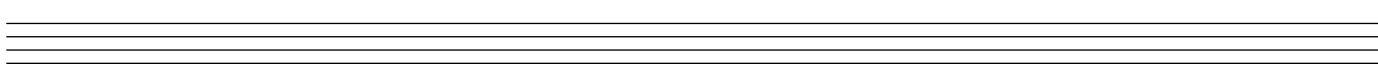
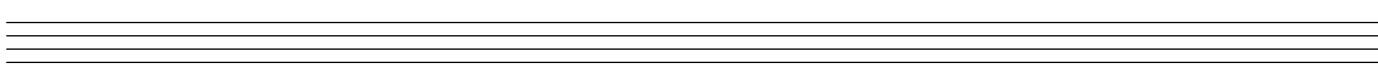
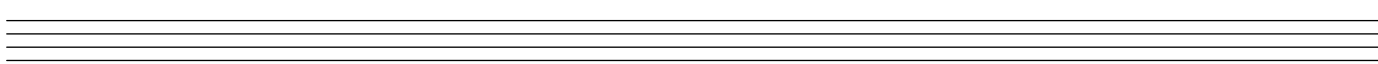
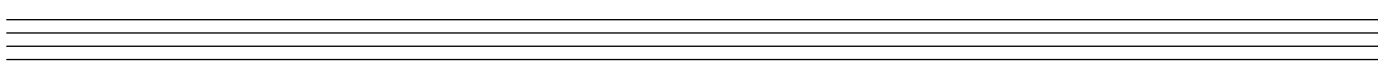
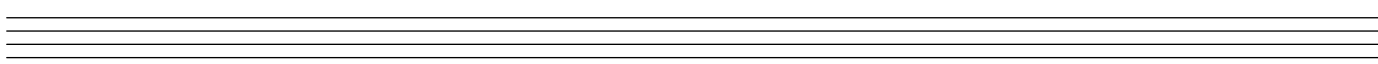
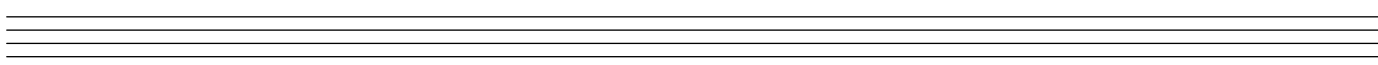
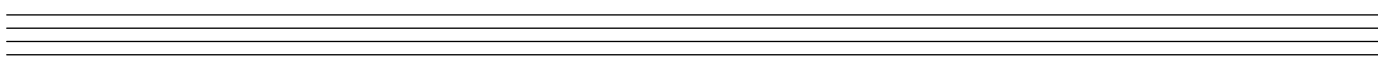
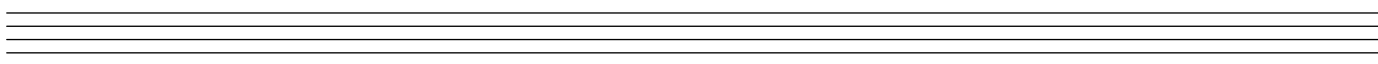
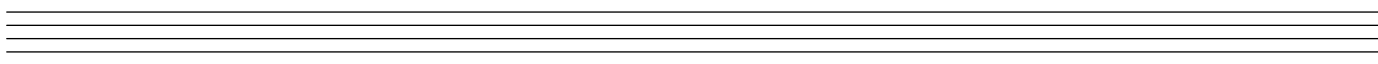
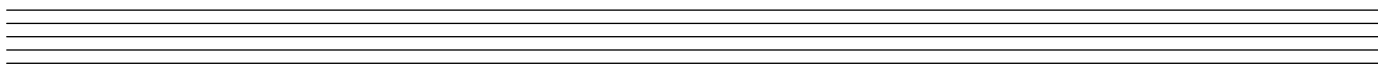
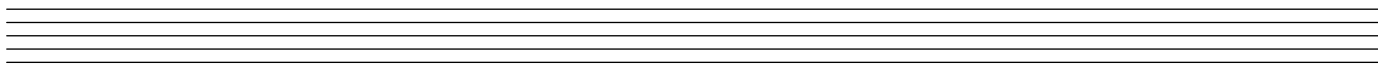
```

```

\paper {
  #(set-paper-size "letter")
  raggedlastbottom = ##f
  linewidth = 7.5\in
  leftmargin = 0.5\in
  bottommargin = 0.25\in
  topmargin = 0.25\in
}

```





‘forcing-rehearsal-marks-to-start-from-a-given-letter-or-number.ly’

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

```
\relative c''{
  c1 \mark \default
  c1 \mark \default
  c1 \mark \default
  c1 \mark #14
  c1 \mark \default
  c1 \mark \default
  c1 \mark \default
  c1 \mark \default

\break

  \set Score.markFormatter = #format-mark-numbers
  c1 \mark #1
  c1 \mark \default
  c1 \mark \default
  c1 \mark \default
  c1 \mark #14
  c1 \mark \default
  c1 \mark \default
  c1 \mark \default
  c1 \mark \default
}
```

The image displays two musical staves. The first staff is a single line of music with a treble clef and a common time signature 'C'. It contains eight measures, each with a single note. Above the notes are the letters A, B, C, O, P, Q, and R, respectively. The second staff is also a single line of music with a treble clef and a common time signature 'C'. It starts at measure 18, indicated by a '9' and '18' at the beginning. It contains eight measures, each with a single note. Above the notes are the numbers 1, 2, 3, 4, 14, 15, 16, and 17, respectively.

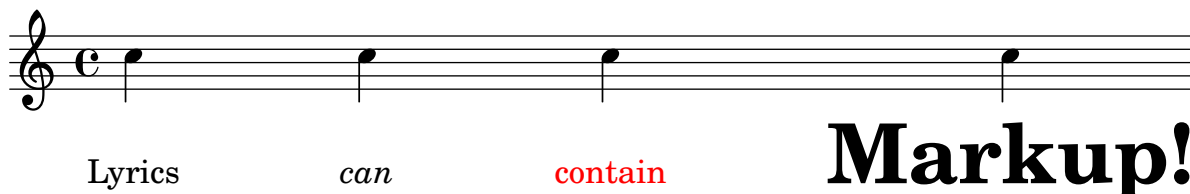
‘formatting-lyrics-syllables.ly’

To format single lyrics syllables, you can simply use \markup { } on these lyrics!

```
melody = \relative c'' { c4 c c c }
lyr = \lyricmode {
  Lyrics \markup { \italic can } \markup {\with-color #red contain }
  \markup {\fontsize #8 \bold "Markup!" }
}

\context Staff <<
  \context Voice = "mel" << \melody >>
  \context Lyrics \lyricsto "mel" \lyr
>>
```

Markup can be used inside lyrics!



Lyrics *can* **contain** **Markup!**

‘inserting-score-fragments-above-the-staff,-as-markups.ly’

The `\markup` command is quite versatile. In this snippet, it contains a `\score` bloc instead of texts or marks.

```
\relative {
  \time 4/8
  \times 2/3 { c'8 d e } \times 2/3 {c d e}
  \times 2/3 { c8 d e } \times 2/3 {c d e}
  g8 a8 g8 a
  g8 a8 g8 a
}
```

Solo Cello Suites

Suite IV

Originalstimmung:



‘rhythmic-slashes.ly’

In "simple" lead-sheets, sometimes no actual notes are written, instead only "rhythmic patterns" and chords above the measures are noted giving the structure of a song. Such a feature is for example useful while creating/transcribing the structure of a song and also when sharing lead sheets with guitarists or jazz musicians.

The standard support for this is described in section "Measure repeats", but then the first beat has to be an ordinary note or rest.

This example shows two solutions to this problem, by redefining ordinary rests to be printed as slashes. (If the duration of each beat is not a quarter note, replace the `r4` in the definitions by a rest of the appropriate duration).

```
% Macro to print single slash
rs = {
  \once \override Rest #'stencil = #ly:percent-repeat-item-interface::beat-slash
  \once \override Rest #'thickness = #'0.48
  \once \override Rest #'slope = #'1.7
  r4
```

```

}

% Function to print a specified number of slashes
comp = #(define-music-function (parser location count) ( integer?)
  #{
    \override Rest #'stencil = #ly:percent-repeat-item-interface::beat-slash
    \override Rest #'thickness = #'0.48
    \override Rest #'slope = #'1.7
    \repeat unfold $count { r4 }
    \revert Rest #'stencil
  }
)

\score{
  \relative c'{
    c d e f | \rs \rs \rs \rs | \comp #4 |
  }
}

```



‘three-sided-box.ly’

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 you use ";" 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 0) padding))
        (y-ext (interval-widen (ly:stencil-extent stencil 1) 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 -1 y-rule padding))
    (set! stencil (ly:stencil-combine-at-edge stencil Y 1 x-rule 0.0))
    (set! stencil (ly:stencil-combine-at-edge stencil Y -1 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?)
  "Draw a box round @var{arg}. Looks at @code{thickness},
@code{box-padding} and @code{font-size} properties to determine line
thickness and padding around the markup."

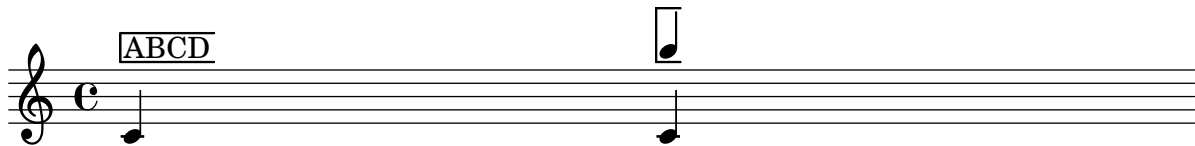
  (let* ((th (chain-assoc-get 'thickness props 0.1))

```

```
(size (chain-assoc-get 'font-size props 0))  
(pad (* (magstep size)  
       (chain-assoc-get 'box-padding props 0.2)))  
(m (interpret-markup layout props arg)))  
(NWS-box-stencil m th pad)))
```

% Test it:

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



Text

‘adjusting-lyrics-vertical-spacing.ly’

This snippets shows you how to bring the lyrics line closer to the Staff.

% Default layout:

```
\score{
  <<
    \new Staff \new Voice = m \relative c'{ c4 d e f g f e d c1}
    \new Lyrics \lyricsto m {aa aa aa aa aa aa aa aa aa }
  >>
}
```

% Reducing the minimum space below the Staff and above the Lyrics:

```
\score {
  <<
    \new Staff \with {
      \override VerticalAxisGroup #'minimum-Y-extent = #'(-1.4)}
    \new Voice = m \relative c'{ c4 d e f g f e d c1 }
    \new Lyrics \with {
      \override VerticalAxisGroup #'minimum-Y-extent = #'(-1.2 . 1)}
    \lyricsto m {aa aa aa aa aa aa aa aa aa }
  >>

  \header {
    tagline = ""
  }
}
```



‘aligning-and-centering-instrument-names.ly’

Instrument names are generally printed at the left side of the staves. To align the names of several different instruments, you can put them in a `\markup` block and use one of the following possibilities:

- * Right-aligned instrument names: this is LilyPond’s default behavior
- * Center-aligned instrument names: with the `\hcenter-in #n` syntax, you can place the instrument names inside a padded box (n being the width of the box)
- * Left-aligned instrument names: you have to print the names on top of an empty box, using the `\combine` command with a `\hspace #n` object.

```

\paper{
  indent = #0
  left-margin = #30
  line-width = #160
}

\new StaffGroup \relative
<<
  \new Staff {
    \set Staff.instrumentName = "blabla"
    c1^"default" c1 \break
    \set Staff.instrumentName = \markup { \hcenter-in #10 "blabla" }
    c1^"centered" c1 \break
    \set Staff.instrumentName = \markup {\combine \hspace #8 "blabla" }
    c1^"left-aligned" c1

  }
  \new Staff {
    \set Staff.instrumentName = "blo"
    c1 c1 \break
    \set Staff.instrumentName = \markup { \hcenter-in #10 "blo" }
    c1 c1 \break
    \set Staff.instrumentName = \markup {\combine \hspace #8 "blo" }
    c1 c1
  }
}
>>

```

default

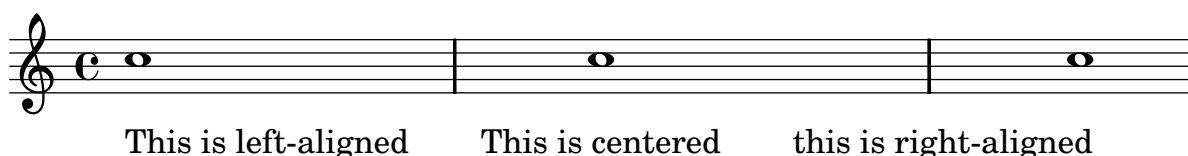
centered

left-aligned

‘aligning-lyrics.ly’

You can specify different horizontal alignment for your lyrics, by overriding the `#'self-alignment-X` property of the `LyricText` object. `#-1` is left, `#0` is center and `#1` is right; however, you can use `#LEFT`, `#CENTER` and `#RIGHT` as well.

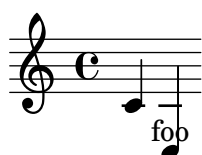
```
\relative c' ' {
  c1 c c
}
\addlyrics {
  \once \override LyricText #'self-alignment-X = #LEFT "This is left-aligned"
  \once \override LyricText #'self-alignment-X = #CENTER "This is centered"
  \once \override LyricText #'self-alignment-X = #1 "this is right-aligned"
}
```



‘blanking-staff-lines-using-the--whiteout-command.ly’

The `\whiteout` command underlays a white box under a markup. Since staff lines are in a lower layer than most other grobs, this white box will not overlap any other grob.

```
\paper
{
  ragged-right = ##t
}
{
  \override TextScript #'extra-offset = #'(2 . 4)
  c'4-\markup { \whiteout \pad-markup #0.5 foo } c
}
```



‘center-text-below-hairpin-dynamics.ly’

This example provides a function to typeset hairpin (de)crescendo with some additional text below it, such as "molto" or "poco".

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

```
hairpinWithCenteredText = #(define-music-function
  (parser location text) (markup?)
  #{
    \override Voice.Hairpin #'stencil = #(lambda (grob)
      (ly:stencil-aligned-to
        (ly:stencil-combine-at-edge
          (ly:stencil-aligned-to (ly:hairpin::print grob) X CENTER)
```

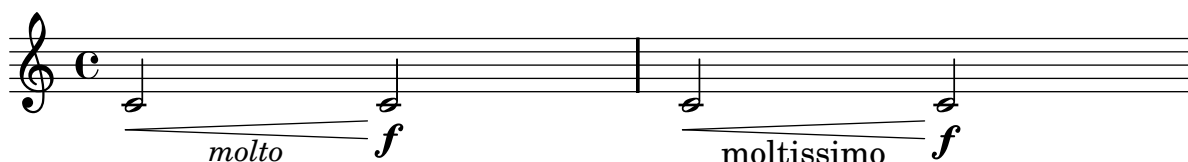
```

Y
DOWN
(ly:stencil-aligned-to (ly:text-interface::print grob) X CENTER))
X LEFT))
\override Voice.Hairpin #'text = $text
#})

hairpinMolto = \hairpinWithCenteredText \markup {\italic "molto"}
hairpinMore  = \hairpinWithCenteredText \markup {\bigger "moltissimo"}

\new Staff {
  \hairpinMolto c'2\< c'2\f
  \hairpinMore  c'2\< c'2\f
}

```



‘changing-the-default-text-font-family.ly’

The default font families for text can be overridden with `make-pango-font-tree`.

```

\paper {
  % change for other default global staff size.
  myStaffSize = #20
  %{
    run
    lilypond -dshow-available-fonts blabla
    to show all fonts available in the process log.
  %}

  #(define fonts
    (make-pango-font-tree "Times New Roman"
                          "Nimbus Sans"
                          "Luxi Mono"
                          "Helvetica"
                          "Courier"
      (/ myStaffSize 20)))
}

\relative {
  c'~\markup {
    roman: foo \bold bla \italic bar \italic \bold baz
  }
  c'_\markup {
    \override #'(font-family . sans)
    {
      sans: foo \bold bla \italic bar \italic \bold baz
    }
  }
}

```



```

    }
  }
  c'\markup {
    \override #'(font-family . typewriter)
    {
      mono: foo \bold bla \italic bar \italic \bold baz
    }
  }
}

```



‘combining-dynamics-with-markup-texts.ly’

Some dynamics may involve text indications (such as "più forte", "piano subito", etc.). They can be produced using a `\markup bloc`.

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

```
piuf = \markup { \italic "molto" \dynamic "f" }
```

```

\relative c''{
  c-\piuf
  c
  c2\< c2\!

  c2\< c2\!
}

```



‘combining-two-parts-on-the-same-staff.ly’

The part combiner tool (`\partcombine` command) allows you to combine different parts on a same Staff. You can choose whether you want or don't want to add texts such as "solo" or "a2", by defining the `printPartCombineTexts` property.

For vocal scores (hymns), there is no need to add "solo"/"a2" texts, so they should be switched off. However, you'd better not use it if there are any solos, as they won't be indicated. In such cases, you may simply want to use standard LilyPond polyphony.

This snippet presents the three ways two parts can be printed on a same staff : standard polyphony, `\partcombine` whitout texts, and `\partcombine` with texts.

```

musicUp = {
  \time 4/4
  \relative c'' {
    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 a
  }
}

\score{
  \new Staff {
    \set Staff.instrumentName = "Standard polyphony  "
    << \musicUp  \ \ \musicDown >>
  }

  \layout{
    indent = 6.0\cm
  }
}

\score{
  \context Staff {
    \set Staff.instrumentName = "PartCombine without texts  "
    \partcombine \musicUp \musicDown
  }
  \layout{
    indent = 6.0\cm
    \context {
      \Voice
      printPartCombineTexts = ##f
    }
  }
}

\score{
  \context Staff {
    \set Staff.instrumentName = "PartCombine with texts  "
    \partcombine \musicUp \musicDown
  }
  \layout{
    indent = 6.0\cm
    \context {
      \Voice
      printPartCombineTexts = ##t
    }
  }
}

```

```

    }
  }
}

```

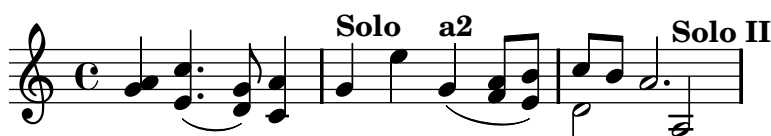
Standard polyphony



PartCombine without texts



PartCombine with texts



‘creating-real-parenthesized-dynamics.ly’

Although the easiest way to add parenthesis to a dynamic mark is to use a `\markup` block, this method has a downside: the created objects will 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 "Markup programmer interface"), combined with the `make-dynamic-script` function. This way, the markup will be regarded as a dynamic, and therefore will remain compatible with commands such as `\dynamicUp` or `\dynamicDown`.

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

```
parenF = #(make-dynamic-script (markup #:line(#:normal-text #:italic
#:fontsize 2 "(" #:hspace -0.8 #:dynamic "f" #:normal-text #:italic
#:fontsize 2 ")") )))
```

```
\score
```

```
{
```

```
  { c''\parenF c'' c'' \dynamicUp c''\parenF }
```

```
}
```



‘creating-text-spanners.ly’

The `<code>\startTextSpan</code>` and `<code>\stopTextSpan</code>` commands give you the ability to create text spanners as easily as pedals indications or octavations. Override some properties of the `<code>TextSpanner</code>` object to modify its output.

```

\relative c''{
  \override TextSpanner #'edge-text = #("bla" . "blu")
  a \startTextSpan
  b c
  a \stopTextSpan

  \override TextSpanner #'dash-period = #2
  \override TextSpanner #'dash-fraction = #0.0
  a \startTextSpan
  b c
  a \stopTextSpan

  \revert TextSpanner #'style
  \override TextSpanner #'style = #'dashed-line \override TextSpanner #'bound-details #'
\override TextSpanner #'bound-details #'right #'text = \markup { \draw-line #'(0 . -2) }

  a \startTextSpan
  b c
  a \stopTextSpan

  \set Staff.middleCPosition = #-13

  \override TextSpanner #'dash-period = #10
  \override TextSpanner #'dash-fraction = #.5
  \override TextSpanner #'thickness = #10
  a \startTextSpan
  b c
  a \stopTextSpan
  \set Staff.middleCPosition = #-6
}

```



‘demonstrating-all-headers.ly’

All header fields with special meanings.

	title	
	subtitle	
poet	instrument	composer
		arranger
piece		opus



localpiece

localopus



‘formatting-lyrics-syllables.ly’

To format single lyrics syllables, you can simply use `\markup { ... }` on these lyrics!

```
melody = \relative c'' { c4 c c c }
```

```
lyr = \lyricmode {
```

```
  Lyrics \markup { \italic can } \markup {\with-color #red contain }
```

```
  \markup {\fontsize #8 \bold "Markup!" }
```

```
}
```

```
\context Staff <<
```

```
  \context Voice = "mel" << \melody >>
```

```
  \context Lyrics \lyricsto "mel" \lyr
```

```
>>
```

Markup can be used inside lyrics!



Lyrics

*can***contain****Markup!**

‘how-to-put-ties-between-syllables-in-lyrics.ly’

This can be achieved by separating those syllables by tildes.

```
\lyrics {
```

```
  wa~o~a
```

```
}
```

wa o a

‘markup-lines.ly’

Text that can spread over pages is entered with the `\markuplines` command.

```
 #(set-default-paper-size "a6")
```

```
 #(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))))
```

```

%% Candide, Voltaire
\markuplines \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 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.
  }
  \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 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.
  }
}

```

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.

‘ottava-text.ly’

Internally, the set-octavation function sets the properties `ottavation` (e.g., to "8va" or "8vb") and `middleCPosition`. To override the text of the bracket, set `ottavation` after invoking `set-octavation`, like in the following example.

```
{
  #(set-octavation 1)
  \set Staff.ottavation = #"8"
  c''1
  #(set-octavation 0)
  c'1
  #(set-octavation 1)
  \set Staff.ottavation = #"Text"
  c''1
}
```



‘outputting-the-version-number.ly’

By putting the output of `lilypond-version` into a lyric or a text markup, it is possible to print the version number of LilyPond in a score, or in a document generated with `lilypond-book`.

```
\score { \context Lyrics {
  \override Score.RehearsalMark #'self-alignment-X = #LEFT
  \mark #(ly:export (string-append "Processed with LilyPond version " (lilypond-version)))
  s2
}
}
```

Processed with LilyPond version 2.11.39

‘piano-template-with-centered-lyrics.ly’

Instead of having a full staff for the melody and lyrics, you can place the lyrics between the piano staff (and omit the separate melody staff).

```
upper = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

  a b c d
}

lower = \relative c {
  \clef bass
  \key c \major
  \time 4/4

  a2 c
}
```



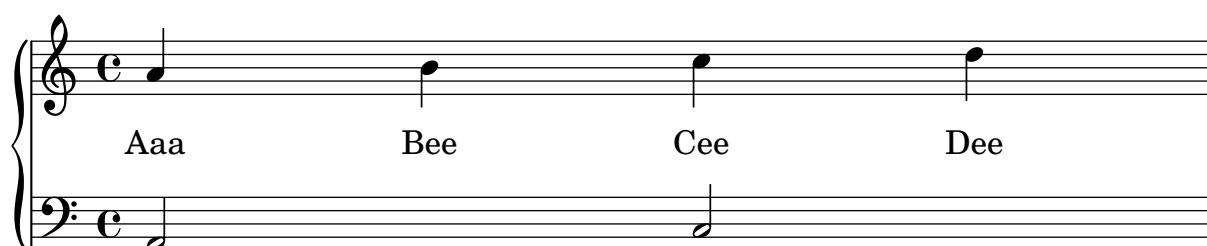
```

}

text = \lyricmode {
  Aaa Bee Cee Dee
}

\score {
  \new GrandStaff <<
    \new Staff = upper { \new Voice = "singer" \upper }
    \new Lyrics \lyricsto "singer" \text
    \new Staff = lower {
      \clef bass
      \lower
    }
  >>
  \layout {
    \context { \GrandStaff \accepts "Lyrics" }
    \context { \Lyrics \consists "Bar_engraver" }
  }
  \midi { }
}

```



‘utf-8.ly’

Various scripts may be used for texts (like titles and lyrics) introduced 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.

% end verbatim - this comment is a hack to prevent texinfo.tex

% from choking on non-European UTF-8 subsets

% Cyrillic font

```
bulgarian = \lyricmode {
```

```
  , , , .
```

```
}
```

```
hebrew = \lyricmode {
```

```
  .
```

```
}
```

```
japanese = \lyricmode {
```

```

}

% "a legal song to you"
portuguese = \lyricmode {
  à vo -- cê uma can -- ção legal
}

\paper {
  ragged-right = ##t
}

\relative {
  c2 d e f g f e
}
\addlyrics { \bulgarian }
\addlyrics { \hebrew }
\addlyrics { \japanese }
\addlyrics { \portuguese }

```



וְהָיָה
 כִּי־יִבְרָא
 סִחֵם
 לְשִׁמּוֹנִי
 אֵיךְ
 חוֹצֵחַ
 , ,
 いろはにほへとちりぬるをわがよたれぞつねならむうみのおくやまけふこえて
 à vo - cê uma can - ção



קרפד
 あさきゆめみじ
 legal

‘vocal-ensemble-template-with-lyrics-aligned-below-and-above-the-staves.ly’

This template is basically the same as the simple "Vocal ensemble" template, with the exception that here all the lyrics lines are placed using `alignAboveContext` and `alignBelowContext`.

```

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
    >>

```

```

\layout {
  \context {
    % a little smaller so lyrics
    % can be closer to the staff
    \Staff
    \override VerticalAxisGroup #'minimum-Y-extent = #'(-3 . 3)
  }
}

\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 \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 \lyricsto basses \bassWords
  >>

  \layout {
    \context {
      % a little smaller so lyrics
      % can be closer to the staff
      \Staff
      \override VerticalAxisGroup #'minimum-Y-extent = #'(-3 . 3)
    }
  }
}

```

hi hi hi hi

ha hu ha hu ha hu ha hu

ho ho ho ho

The first system of a musical score in common time (C). It consists of a treble staff and a bass staff. The treble staff has a key signature of one flat (Bb) and a common time signature (C). The melody is composed of quarter notes, with a slur over the third measure. The lyrics 'hi' are placed above the treble staff, and 'ha hu' are placed below it. The bass staff has a key signature of one flat (Bb) and a common time signature (C). The lyrics 'ho' are placed below the bass staff. The system ends with a double bar line.

hi hi hi hi

ha hu ha hu ha hu ha hu

ho ho ho ho

The second system of a musical score in common time (C). It consists of a treble staff and a bass staff. The treble staff has a key signature of one flat (Bb) and a common time signature (C). The melody is composed of quarter notes, with a slur over the third measure. The lyrics 'hi' are placed above the treble staff, and 'ha hu' are placed below it. The bass staff has a key signature of one flat (Bb) and a common time signature (C). The lyrics 'ho' are placed below the bass staff. The system ends with a double bar line.

Vocal music

‘adding-ambiti-per-voice.ly’

Ambits can be added per voice. In that case, the ambitus must be moved manually to prevent collisions.

```
\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
  }
}>>
```



‘adjusting-lyrics-vertical-spacing.ly’

This snippets shows you how to bring the lyrics line closer to the Staff.

% Default layout:

```
\score{
  <<
    \new Staff \new Voice = m \relative c'{ c4 d e f g f e d c1}
    \new Lyrics \lyricsto m {aa aa aa aa aa aa aa aa aa }
  >>
}
```

% Reducing the minimum space below the Staff and above the Lyrics:

```
\score {
  <<
    \new Staff \with {
      \override VerticalAxisGroup #'minimum-Y-extent = #'(-1. 4)}
    \new Voice = m \relative c'{ c4 d e f g f e d c1 }
    \new Lyrics \with {
      \override VerticalAxisGroup #'minimum-Y-extent = #'(-1.2 . 1)}
    \lyricsto m {aa aa aa aa aa aa aa aa aa }
  >>

  \header {
    tagline = ""
  }
}
```

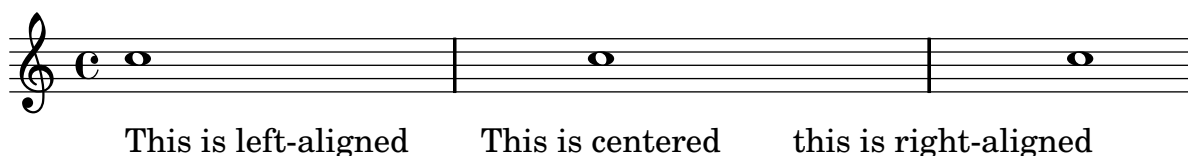
}



‘aligning-lyrics.ly’

You can specify different horizontal alignment for your lyrics, by overriding the `#'self-alignment-X` property of the `LyricText` object. `#-1` is left, `#0` is center and `#1` is right; however, you can use `#LEFT`, `#CENTER` and `#RIGHT` as well.

```
\relative c' {
  c1 c c
}
\addlyrics {
  \once \override LyricText #'self-alignment-X = #LEFT "This is left-aligned"
  \once \override LyricText #'self-alignment-X = #CENTER "This is centered"
  \once \override LyricText #'self-alignment-X = #1 "this is right-aligned"
}
```



‘ambiti-multiple-voices.ly’

If you have multiple voices in a single staff and you want a single ambitus per staff rather than per voice, add the `Ambitus_engraver` to the `Staff` context rather than to the `Voice` context.

```
\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
  }
}>>
```



‘chant-or-psalms-notation.ly’

This form of notation is used for the chant of the Psalms, where verses aren’t always the same length.

```
stemon = { \override Staff.Stem #'transparent = ##f }
stemoff = { \override Staff.Stem #'transparent = ##t }
```

```
\score {
\new Staff \with {\remove "Time_signature_engraver" }
{
    \key g \minor
    \set Score.timing = ##f
    \stemoff a'\breve bes'4 g'4
    \stemon a'2 \bar "||"
    \stemoff a'\breve g'4 a'4
    \stemon f'2 \bar "||"
    \stemoff a'\breve^{\markup { \italic flexe }}
    \stemon g'2 \bar "||"
}
\layout { raggedright = ##t}
}
```



‘demo-midiinstruments.ly’

Problem: How to know which MidiInstrument would be best for your composition? Solution: A Lilypond demo file.

```
\score {
  \new Staff <<
    \new Voice { \melodie
      } %Voice
    >> %Staff
  \layout { }
} %score

\score {
  \new Staff <<
    \tempo 4 = 180
    %\set Staff.instrumentName="S/A"
    %\set Staff.midiMinimumVolume = #0.2
    %\set Staff.midiMaximumVolume = #0.4
    %\set Voice.dynamicAbsoluteVolumeFunction = #0.6
    \new Voice { r \mf
      \set Staff.midiInstrument="acoustic grand" \melodie
      \set Staff.midiInstrument="bright acoustic" \melodie
      \set Staff.midiInstrument="electric grand" \melodie
      \set Staff.midiInstrument="honky-tonk" \melodie
    }
  }
}
```



```
\set Staff.midiInstrument="electric piano 1" \melodie
\set Staff.midiInstrument="electric piano 2" \melodie
\set Staff.midiInstrument="harpsichord" \melodie
\set Staff.midiInstrument="clav" \melodie
\set Staff.midiInstrument="celesta" \melodie
\set Staff.midiInstrument="glockenspiel" \melodie
\set Staff.midiInstrument="music box" \melodie
\set Staff.midiInstrument="vibraphone" \melodie
\set Staff.midiInstrument="marimba" \melodie
\set Staff.midiInstrument="xylophone" \melodie
\set Staff.midiInstrument="tubular bells" \melodie
\set Staff.midiInstrument="dulcimer" \melodie
\set Staff.midiInstrument="drawbar organ" \melodie
\set Staff.midiInstrument="percussive organ" \melodie
\set Staff.midiInstrument="rock organ" \melodie
\set Staff.midiInstrument="church organ" \melodie
\set Staff.midiInstrument="reed organ" \melodie
\set Staff.midiInstrument="accordion" \melodie
\set Staff.midiInstrument="harmonica" \melodie
\set Staff.midiInstrument="concertina" \melodie
\set Staff.midiInstrument="acoustic guitar (nylon)" \melodie
\set Staff.midiInstrument="acoustic guitar (steel)" \melodie
\set Staff.midiInstrument="electric guitar (jazz)" \melodie
\set Staff.midiInstrument="electric guitar (clean)" \melodie
\set Staff.midiInstrument="electric guitar (muted)" \melodie
\set Staff.midiInstrument="overdriven guitar" \melodie
\set Staff.midiInstrument="distorted guitar" \melodie
\set Staff.midiInstrument="acoustic bass" \melodie
\set Staff.midiInstrument="electric bass (finger)" \melodie
\set Staff.midiInstrument="electric bass (pick)" \melodie
\set Staff.midiInstrument="fretless bass" \melodie
\set Staff.midiInstrument="slap bass 1" \melodie
\set Staff.midiInstrument="slap bass 2" \melodie
\set Staff.midiInstrument="synth bass 1" \melodie
\set Staff.midiInstrument="synth bass 2" \melodie
\set Staff.midiInstrument="violin" \melodie
\set Staff.midiInstrument="viola" \melodie
\set Staff.midiInstrument="cello" \melodie
\set Staff.midiInstrument="contrabass" \melodie
\set Staff.midiInstrument="tremolo strings" \melodie
\set Staff.midiInstrument="pizzicato strings" \melodie
\set Staff.midiInstrument="orchestral strings" \melodie
\set Staff.midiInstrument="timpani" \melodie
\set Staff.midiInstrument="string ensemble 1" \melodie
\set Staff.midiInstrument="string ensemble 2" \melodie
\set Staff.midiInstrument="synthstrings 1" \melodie
\set Staff.midiInstrument="synthstrings 2" \melodie
\set Staff.midiInstrument="choir aahs" \melodie
\set Staff.midiInstrument="voice oohs" \melodie
\set Staff.midiInstrument="synth voice" \melodie
\set Staff.midiInstrument="orchestra hit" \melodie
\set Staff.midiInstrument="trumpet" \melodie
```

```
\set Staff.midiInstrument="trombone" \melodie
\set Staff.midiInstrument="tuba" \melodie
\set Staff.midiInstrument="muted trumpet" \melodie
\set Staff.midiInstrument="french horn" \melodie
\set Staff.midiInstrument="brass section" \melodie
\set Staff.midiInstrument="synthbrass 1" \melodie
\set Staff.midiInstrument="synthbrass 2" \melodie
\set Staff.midiInstrument="soprano sax" \melodie
\set Staff.midiInstrument="alto sax" \melodie
\set Staff.midiInstrument="tenor sax" \melodie
\set Staff.midiInstrument="baritone sax" \melodie
\set Staff.midiInstrument="oboe" \melodie
\set Staff.midiInstrument="english horn" \melodie
\set Staff.midiInstrument="bassoon" \melodie
\set Staff.midiInstrument="clarinet" \melodie
\set Staff.midiInstrument="piccolo" \melodie
\set Staff.midiInstrument="flute" \melodie
\set Staff.midiInstrument="recorder" \melodie
\set Staff.midiInstrument="pan flute" \melodie
\set Staff.midiInstrument="blown bottle" \melodie
\set Staff.midiInstrument="shakuhachi" \melodie
\set Staff.midiInstrument="whistle" \melodie
\set Staff.midiInstrument="ocarina" \melodie
\set Staff.midiInstrument="lead 1 (square)" \melodie
\set Staff.midiInstrument="lead 2 (sawtooth)" \melodie
\set Staff.midiInstrument="lead 3 (calliope)" \melodie
\set Staff.midiInstrument="lead 4 (chiff)" \melodie
\set Staff.midiInstrument="lead 5 (charang)" \melodie
\set Staff.midiInstrument="lead 6 (voice)" \melodie
\set Staff.midiInstrument="lead 7 (fifths)" \melodie
\set Staff.midiInstrument="lead 8 (bass+lead)" \melodie
\set Staff.midiInstrument="pad 1 (new age)" \melodie
\set Staff.midiInstrument="pad 2 (warm)" \melodie
\set Staff.midiInstrument="pad 3 (polysynth)" \melodie
\set Staff.midiInstrument="pad 4 (choir)" \melodie
\set Staff.midiInstrument="pad 5 (bowed)" \melodie
\set Staff.midiInstrument="pad 6 (metallic)" \melodie
\set Staff.midiInstrument="pad 7 (halo)" \melodie
\set Staff.midiInstrument="pad 8 (sweep)" \melodie
\set Staff.midiInstrument="fx 1 (rain)" \melodie
\set Staff.midiInstrument="fx 2 (soundtrack)" \melodie
\set Staff.midiInstrument="fx 3 (crystal)" \melodie
\set Staff.midiInstrument="fx 4 (atmosphere)" \melodie
\set Staff.midiInstrument="fx 5 (brightness)" \melodie
\set Staff.midiInstrument="fx 6 (goblins)" \melodie
\set Staff.midiInstrument="fx 7 (echoes)" \melodie
\set Staff.midiInstrument="fx 8 (sci-fi)" \melodie
\set Staff.midiInstrument="sitar" \melodie
\set Staff.midiInstrument="banjo" \melodie
\set Staff.midiInstrument="shamisen" \melodie
\set Staff.midiInstrument="koto" \melodie
\set Staff.midiInstrument="kalimba" \melodie
```

```

\set Staff.midiInstrument="bagpipe" \melodie
\set Staff.midiInstrument="fiddle" \melodie
\set Staff.midiInstrument="shanai" \melodie
\set Staff.midiInstrument="tinkle bell" \melodie
\set Staff.midiInstrument="agogo" \melodie
\set Staff.midiInstrument="steel drums" \melodie
\set Staff.midiInstrument="woodblock" \melodie
\set Staff.midiInstrument="taiko drum" \melodie
\set Staff.midiInstrument="melodic tom" \melodie
\set Staff.midiInstrument="synth drum" \melodie
\set Staff.midiInstrument="reverse cymbal" \melodie
\set Staff.midiInstrument="guitar fret noise" \melodie
\set Staff.midiInstrument="breath noise" \melodie
\set Staff.midiInstrument="seashore" \melodie
\set Staff.midiInstrument="bird tweet" \melodie
\set Staff.midiInstrument="telephone ring" \melodie
\set Staff.midiInstrument="helicopter" \melodie
\set Staff.midiInstrument="applause" \melodie
\set Staff.midiInstrument="gunshot" \melodie
} %Voice
>> %Staff
\midi { }
} %score

```

Demo of all midi sounds

Myself



'formatting-lyrics-syllables.ly'

To format single lyrics syllables, you can simply use \markup { } on these lyrics!

```
melody = \relative c'' { c4 c c c }
```

```
lyr = \lyricmode {
```

```
  Lyrics \markup { \italic can } \markup {\with-color #red contain }
```

```
  \markup {\fontsize #8 \bold "Markup!" }
```

```
}
```

```
\context Staff <<
```

```
  \context Voice = "mel" << \melody >>
```

```
  \context Lyrics \lyricsto "mel" \lyr
```

```
>>
```

Markup can be used inside lyrics!



Lyrics

*can**contain***Markup!**

‘how-to-put-ties-between-syllables-in-lyrics.ly’

This can be achieved by separating those syllables by tildes.

```
\lyrics {
  wa~o~a
}
```

waoa

‘piano-template-with-melody-and-lyrics.ly’

Here is a typical song format: one staff with the melody and lyrics, with piano accompaniment underneath.

```
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

  a 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 { \RemoveEmptyStaffContext }
}
\midi { }
}

```

The image shows a musical score for a vocal melody and piano accompaniment. The vocal line is in treble clef, 4/4 time, with notes A4, B4, C5, and D5. The piano accompaniment consists of two staves: the right hand in treble clef and the left hand in bass clef. The right hand plays A4, B4, and C5, while the left hand plays A3 and C4. The lyrics 'Aaa', 'Bee', 'Cee', and 'Dee' are written below the vocal notes.

‘single-staff-template-with-notes,-lyrics,-and-chords.ly’

This template allows you to prepare a song with melody, words, and chords.

```

melody = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

  a b c d
}

text = \lyricmode {
  Aaa Bee Cee Dee
}

harmonies = \chordmode {
  a2 c2
}

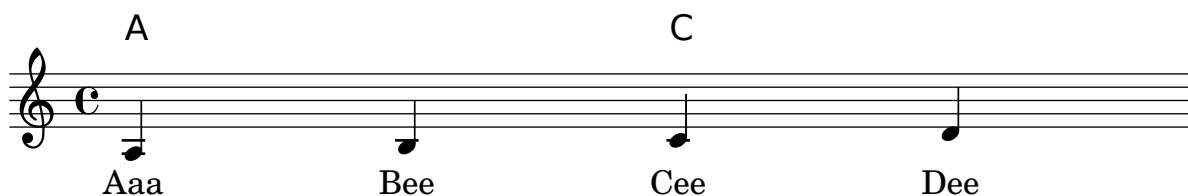
\score {
  <<
    \new ChordNames {
      \set chordChanges = ##t
      \harmonies
    }
    \new Voice = "one" {
      \autoBeamOff

```

```

        \melody
    }
    \new Lyrics \lyricsto "one" \text
    >>
    \layout { }
    \midi { }
}

```



‘single-staff-template-with-notes-and-lyrics.ly’

This small template demonstrates a simple melody with lyrics. Cut and paste, add notes, then words for the lyrics. This example turns off automatic beaming, which is common for vocal parts. If you want to use automatic beaming, you’ll have to change or comment out the relevant line.

```

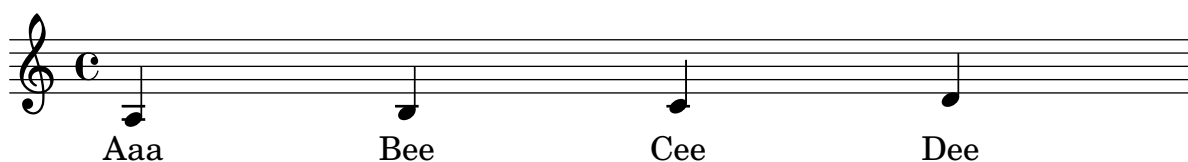
melody = \relative c' {
    \clef treble
    \key c \major
    \time 4/4

    a4 b c d
}

text = \lyricmode {
    Aaa Bee Cee Dee
}

\score{
    <<
        \new Voice = "one" {
            \autoBeamOff
            \melody
        }
        \new Lyrics \lyricsto "one" \text
    >>
    \layout { }
    \midi { }
}

```

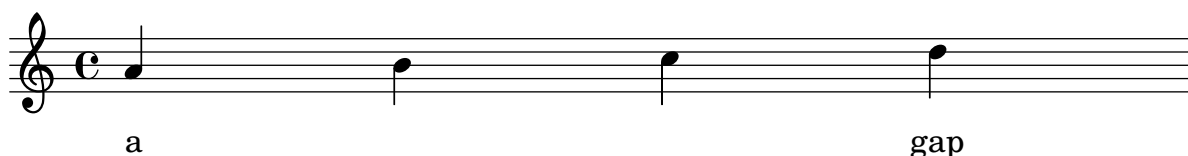


```
'skips-in-lyric-mode-2.ly'
```

Although you can't use 's' in lyric mode (it's taken to be a literal s, not a space) you can use either "" or ..

So for example:

```
<<
\relative c'' { a4 b c d }
\new Lyrics \lyricmode { a4 _2 gap4 }
>>
```



```
'skips-in-lyric-mode.ly'
```

The `s` syntax is only available in note mode and chord mode. In other situations, for example, when entering lyrics, you should use the `\skip` command.

```
<<
  \relative { a'2 a1 }
  \new Lyrics \lyricmode { \skip 2 bla1 }
>>
```



'vertically-aligning-ossias-and-lyrics.ly'

This snippet shows of to use the `alignBelowContext` and `alignAboveContext` properties, which may be needed for text elements (e.g. lyrics) positioning, but also for musical contents such as ossias.

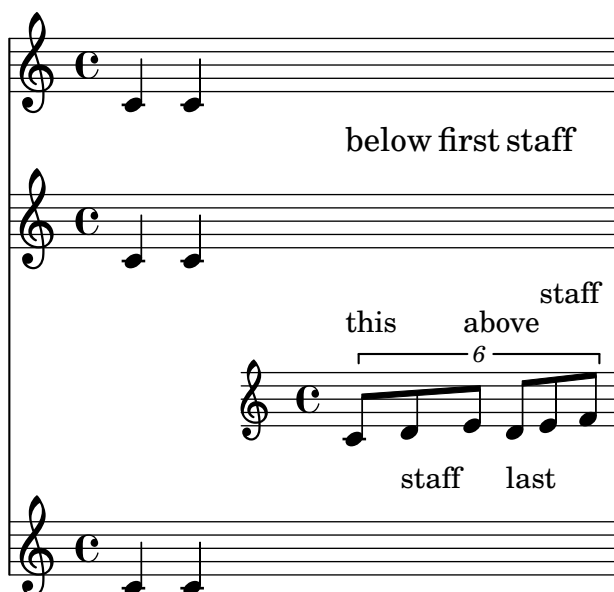
```
\paper {
  ragged-right = ##t
}
```

```
\relative <<
  \new Staff = "1" { c4 c s2 }
  \new Staff = "2" { c4 c s2 }
  \new Staff = "3" { c4 c s2 }
  { \skip 2
    <<
      \lyrics {
        \set alignBelowContext = #"1"
        below8 first staff
      }
      \new Staff {
        \set Staff.alignAboveContext = #"3"
```

```

\times 4/6 {
  \override TextScript #'padding = #3
  c8^"this" d_"staff" e^"above" d_"last" e^"staff" f
}
}
>> }
>>

```



‘vocal-ensemble-template-with-automatic-piano-reduction.ly’

This template adds an automatic piano reduction to the standard SATB vocal score demonstrated in "Vocal ensemble template". This demonstrates one of the strengths of LilyPond – you can use a music definition more than once. If you make any changes to the vocal notes (say, `tenorMusic`), then the changes will also apply to the piano reduction.

```

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 Lyrics = sopranos { s1 }
        \new Staff = women <<
          \new Voice =
            "sopranos" { \voiceOne << \global \sopMusic >> }
          \new Voice =
            "altos" { \voiceTwo << \global \altoMusic >> }
        >>
        \new Lyrics = "altos" { s1 }
        \new Lyrics = "tenors" { s1 }
        \new Staff = men <<
          \clef bass
          \new Voice =
            "tenors" { \voiceOne << \global \tenorMusic >> }
          \new Voice =
            "basses" { \voiceTwo << \global \bassMusic >> }
        >>
        \new Lyrics = basses { s1 }

        \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 >>
        >>
      >>
    >>
  }

```

```

>>
>>
>>
\layout {
  \context {
    % a little smaller so lyrics
    % can be closer to the staff
    \Staff
    \override VerticalAxisGroup #'minimum-Y-extent = #'(-3 . 3)
  }
}

```

The image shows a musical score for a vocal ensemble in 4/4 time, key of C major. It consists of four staves: two for Soprano (Sopranos) and two for Alto (Alto). The lyrics are aligned above and below the staves. The lyrics are: hi, ha, hu, ho. The melody is simple, with a half note for 'hi', a half note for 'ha', a half note for 'hu', and a half note for 'ho'. The Soprano parts are in the upper register, and the Alto parts are in the lower register. The lyrics are aligned with the notes: 'hi' is above the first note, 'ha' is below the second note, 'hu' is below the third note, and 'ho' is below the fourth note. The Soprano parts have a slur over the third and fourth notes, and the Alto parts have a slur over the third and fourth notes.

‘vocal-ensemble-template-with-lyrics-aligned-below-and-above-the-staves.ly’

This template is basically the same as the simple "Vocal ensemble" template, with the exception that here all the lyrics lines are placed using `alignAboveContext` and `alignBelowContext`.

```

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
  >>

  \layout {
    \context {
      % a little smaller so lyrics
      % can be closer to the staff
      \Staff
    }
  }
}

```

```

        \override VerticalAxisGroup #'minimum-Y-extent = #'(-3 . 3)
      }
    }
  }

\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 \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 \lyricsto basses \bassWords
  >>

  \layout {
    \context {
      % a little smaller so lyrics
      % can be closer to the staff
      \Staff
      \override VerticalAxisGroup #'minimum-Y-extent = #'(-3 . 3)
    }
  }
}

```

‘vocal-ensemble-template.ly’

Here is a standard four-part SATB vocal score. With larger ensembles, it’s often useful to include a section which is included in all parts. For example, the time signature and key signatures are almost always the same for all parts. Like in the "Hymn" template, the four voices are regrouped on only two staves.

```
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 Lyrics = sopranos { s1 }
    \new Staff = women <<
      \new Voice =
        "sopranos" { \voiceOne << \global \sopMusic >> }
      \new Voice =
        "altos" { \voiceTwo << \global \altoMusic >> }
    >>
    \new Lyrics = "altos" { s1 }
    \new Lyrics = "tenors" { s1 }
    \new Staff = men <<
      \clef bass
      \new Voice =
        "tenors" { \voiceOne <<\global \tenorMusic >> }
      \new Voice =
        "basses" { \voiceTwo <<\global \bassMusic >> }
    >>
    \new Lyrics = basses { s1 }

    \context Lyrics = sopranos \lyricsto sopranos \sopWords
    \context Lyrics = altos \lyricsto altos \altoWords
    \context Lyrics = tenors \lyricsto tenors \tenorWords
    \context Lyrics = basses \lyricsto basses \bassWords
  >>

  \layout {
    \context {
      % a little smaller so lyrics
      % can be closer to the staff
      \Staff
      \override VerticalAxisGroup #'minimum-Y-extent = #'(-3 . 3)
    }
  }
}

```

A musical score for vocal music, page 101. The score is written for a single voice part on a grand staff (treble and bass clefs) in common time (C). The melody is composed of four measures, each containing a half note. The notes are G4, A4, B4, and A4, with a slur over the last two notes. The lyrics are 'hi ha hu ho' repeated four times. The lyrics are written below the staff, with 'hi' above the first note of each measure, 'ha' above the second, 'hu' above the third, and 'ho' below the fourth. The staff is empty for the rest of the page.

hi hi hi hi

ha ha ha ha

hu hu hu hu

ho ho ho ho

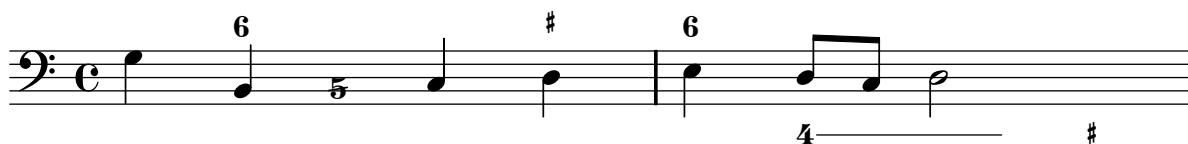
Chords

‘adding-a-figured-bass-above-or-below-the-notes.ly’

When writing a figured bass, here’s a way to specify if you want your figures to be placed above or below the bass notes, by defining the `BassFigureAlignmentPositioning` `#'direction` property (exclusively in a `Staff` context). Choices are `#UP` (or `#1`), `#CENTER` (or `#0`) and `#DOWN` (or `#-1`).

As you can see here, this property can be changed as many times as you wish. Use `\once \override` if you don’t want the tweak to apply to the whole score.

```
bass = { \clef bass g4 b, c d e d8 c d2}
continuo = \figuremode {
  < _ >4 < 6 >8
  \once \override Staff.BassFigureAlignmentPositioning #'direction = #CENTER
  <5/> < _ >4
  \override Staff.BassFigureAlignmentPositioning #'direction = #UP
  < _+ > < 6 >
  \set Staff.useBassFigureExtenders = ##t
  \override Staff.BassFigureAlignmentPositioning #'direction = #DOWN
  < 4 >4. < 4 >8 < _+ >4
}
\score {
  << \new Staff = bassStaff \bass
  \context Staff = bassStaff \continuo >>
}
```



‘changing-the-chord-names-to-german-or-semi-german-notation.ly’

The english naming of chords (default) can be changed to german (`\germanChords` replaces B and Bes to H and B) or semi-german (`\semiGermanChords` replaces B and Bes to H and Bb).

```
music = \chordmode {
  c1/c cis/cis
  b/b bis/bis bes/bes
}
```

```
%% The following is only here to print the names of the
%% chords styles; it can be removed if you do not need to
%% print them.
```

```
\layout {
  \context {\ChordNames \consists Instrument_name_engraver }
}
```

```
<<
\new ChordNames {
  \set ChordNames.instrumentName = #"default"
```



```

\music
}
\new ChordNames {
  \set ChordNames.instrumentName = #"german"
  \germanChords \music }
\new ChordNames {
  \set ChordNames.instrumentName = #"semi-german"
  \semiGermanChords \music }
\context Voice { \music }
>>

```

default	C/C	C [#] /C [#]	B/B	B [#] /B [#]	B ^b /B ^b
german	C/c	C [#] /cis	H/h	H [#] /his	B/b
semi-german	C/c	C [#] /cis	H/h	H [#] /his	B ^b /b

‘chord-name-exceptions.ly’

The property `chordNameExceptions` can be used to store a list of special notations for specific chords.

% 7sus4 denoted with ^7 wahh

```

chExceptionMusic = {
  <c f g bes>1-\markup { \super "7" "wahh" }
}

```

% add to existing exceptions.

```

chExceptions = #(append
  (sequential-music-to-chord-exceptions chExceptionMusic #t)
  ignatzekExceptions)

```

```

theMusic = \chordmode {
  c:7sus4 c:dim7/+f
  \set chordNameExceptions = #chExceptions
  c:7sus4 c:dim7/+f
}

```

```

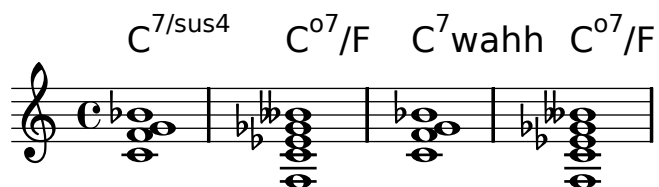
\layout {
  ragged-right = ##t
}

```

```

<< \context ChordNames \theMusic
  \context Voice \theMusic
>>

```



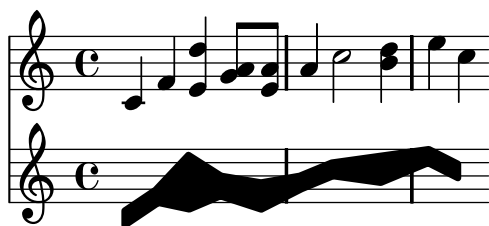
‘clusters.ly’

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

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

fragment = \relative c' {
  c4 f4 <e d'>4
  <g a>8 <e a> a4 c2 <d b>4 e4
  c4
}

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



‘single-staff-template-with-notes,-lyrics,-and-chords.ly’

This template allows you to prepare a song with melody, words, and chords.

```
melody = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

  a b c d
}

text = \lyricmode {
  Aaa Bee Cee Dee
}

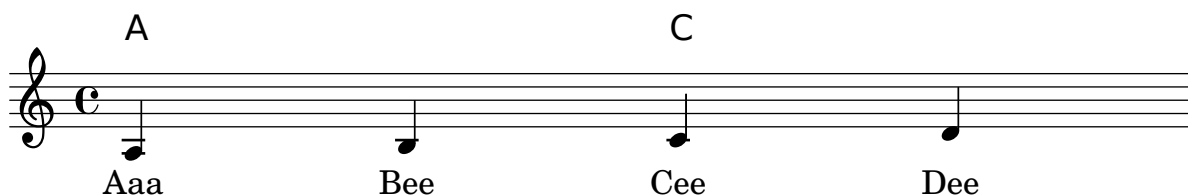
harmonies = \chordmode {
  a2 c2
}

\score {
  <<
    \new ChordNames {
      \set chordChanges = ##t
```

```

        \harmonies
      }
    \new Voice = "one" {
      \autoBeamOff
      \melody
    }
    \new Lyrics \lyricsto "one" \text
  >>
  \layout { }
  \midi { }
}

```



‘single-staff-template-with-notes-and-chords.ly’

Want to prepare a lead sheet with a melody and chords? Look no further!

```

melody = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

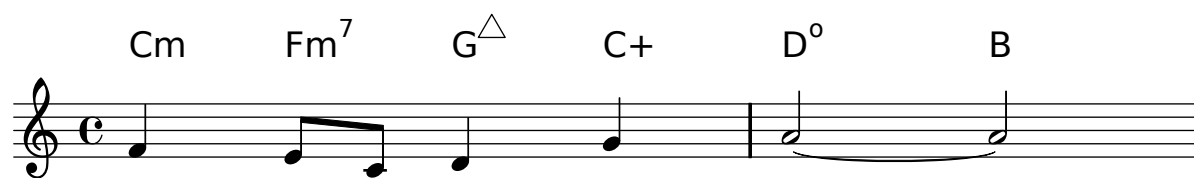
  f4 e8[ c] d4 g |
  a2 ~ a2 |
}

harmonies = \chordmode {
  c4:m f:min7 g:maj c:aug d2:dim b:sus
}

\score {
  <<
    \new ChordNames {
      \set chordChanges = ##t
      \harmonies
    }
    \new Staff \melody
  >>

  \layout{ }
  \midi { }
}

```



Piano

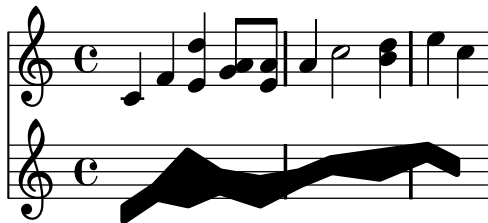
‘clusters.ly’

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

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

fragment = \relative c' {
  c4 f4 <e d'>4
  <g a>8 <e a> a4 c2 <d b>4 e4
  c4
}

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



‘demo-midiinstruments.ly’

Problem: How to know which MidiInstrument would be best for your composition? Solution: A Lilypond demo file.

```
\score {
  \new Staff <<
    \new Voice { \melodie
    } %Voice
  >> %Staff
  \layout { }
} %score

\score {
  \new Staff <<
    \tempo 4 = 180
    %\set Staff.instrumentName="S/A"
    %\set Staff.midiMinimumVolume = #0.2
    %\set Staff.midiMaximumVolume = #0.4
    %\set Voice.dynamicAbsoluteVolumeFunction = #0.6
    \new Voice { r \mf
      \set Staff.midiInstrument="acoustic grand" \melodie
      \set Staff.midiInstrument="bright acoustic" \melodie
      \set Staff.midiInstrument="electric grand" \melodie
      \set Staff.midiInstrument="honky-tonk" \melodie
      \set Staff.midiInstrument="electric piano 1" \melodie
    }
  >>
}
```

```
\set Staff.midiInstrument="electric piano 2" \melodie
\set Staff.midiInstrument="harpsichord" \melodie
\set Staff.midiInstrument="clav" \melodie
\set Staff.midiInstrument="celesta" \melodie
\set Staff.midiInstrument="glockenspiel" \melodie
\set Staff.midiInstrument="music box" \melodie
\set Staff.midiInstrument="vibraphone" \melodie
\set Staff.midiInstrument="marimba" \melodie
\set Staff.midiInstrument="xylophone" \melodie
\set Staff.midiInstrument="tubular bells" \melodie
\set Staff.midiInstrument="dulcimer" \melodie
\set Staff.midiInstrument="drawbar organ" \melodie
\set Staff.midiInstrument="percussive organ" \melodie
\set Staff.midiInstrument="rock organ" \melodie
\set Staff.midiInstrument="church organ" \melodie
\set Staff.midiInstrument="reed organ" \melodie
\set Staff.midiInstrument="accordion" \melodie
\set Staff.midiInstrument="harmonica" \melodie
\set Staff.midiInstrument="concertina" \melodie
\set Staff.midiInstrument="acoustic guitar (nylon)" \melodie
\set Staff.midiInstrument="acoustic guitar (steel)" \melodie
\set Staff.midiInstrument="electric guitar (jazz)" \melodie
\set Staff.midiInstrument="electric guitar (clean)" \melodie
\set Staff.midiInstrument="electric guitar (muted)" \melodie
\set Staff.midiInstrument="overdriven guitar" \melodie
\set Staff.midiInstrument="distorted guitar" \melodie
\set Staff.midiInstrument="acoustic bass" \melodie
\set Staff.midiInstrument="electric bass (finger)" \melodie
\set Staff.midiInstrument="electric bass (pick)" \melodie
\set Staff.midiInstrument="fretless bass" \melodie
\set Staff.midiInstrument="slap bass 1" \melodie
\set Staff.midiInstrument="slap bass 2" \melodie
\set Staff.midiInstrument="synth bass 1" \melodie
\set Staff.midiInstrument="synth bass 2" \melodie
\set Staff.midiInstrument="violin" \melodie
\set Staff.midiInstrument="viola" \melodie
\set Staff.midiInstrument="cello" \melodie
\set Staff.midiInstrument="contrabass" \melodie
\set Staff.midiInstrument="tremolo strings" \melodie
\set Staff.midiInstrument="pizzicato strings" \melodie
\set Staff.midiInstrument="orchestral strings" \melodie
\set Staff.midiInstrument="timpani" \melodie
\set Staff.midiInstrument="string ensemble 1" \melodie
\set Staff.midiInstrument="string ensemble 2" \melodie
\set Staff.midiInstrument="synthstrings 1" \melodie
\set Staff.midiInstrument="synthstrings 2" \melodie
\set Staff.midiInstrument="choir aahs" \melodie
\set Staff.midiInstrument="voice oohs" \melodie
\set Staff.midiInstrument="synth voice" \melodie
\set Staff.midiInstrument="orchestra hit" \melodie
\set Staff.midiInstrument="trumpet" \melodie
\set Staff.midiInstrument="trombone" \melodie
```

```
\set Staff.midiInstrument="tuba" \melodie
\set Staff.midiInstrument="muted trumpet" \melodie
\set Staff.midiInstrument="french horn" \melodie
\set Staff.midiInstrument="brass section" \melodie
\set Staff.midiInstrument="synthbrass 1" \melodie
\set Staff.midiInstrument="synthbrass 2" \melodie
\set Staff.midiInstrument="soprano sax" \melodie
\set Staff.midiInstrument="alto sax" \melodie
\set Staff.midiInstrument="tenor sax" \melodie
\set Staff.midiInstrument="baritone sax" \melodie
\set Staff.midiInstrument="oboe" \melodie
\set Staff.midiInstrument="english horn" \melodie
\set Staff.midiInstrument="bassoon" \melodie
\set Staff.midiInstrument="clarinet" \melodie
\set Staff.midiInstrument="piccolo" \melodie
\set Staff.midiInstrument="flute" \melodie
\set Staff.midiInstrument="recorder" \melodie
\set Staff.midiInstrument="pan flute" \melodie
\set Staff.midiInstrument="blown bottle" \melodie
\set Staff.midiInstrument="shakuhachi" \melodie
\set Staff.midiInstrument="whistle" \melodie
\set Staff.midiInstrument="ocarina" \melodie
\set Staff.midiInstrument="lead 1 (square)" \melodie
\set Staff.midiInstrument="lead 2 (sawtooth)" \melodie
\set Staff.midiInstrument="lead 3 (calliope)" \melodie
\set Staff.midiInstrument="lead 4 (chiff)" \melodie
\set Staff.midiInstrument="lead 5 (charang)" \melodie
\set Staff.midiInstrument="lead 6 (voice)" \melodie
\set Staff.midiInstrument="lead 7 (fifths)" \melodie
\set Staff.midiInstrument="lead 8 (bass+lead)" \melodie
\set Staff.midiInstrument="pad 1 (new age)" \melodie
\set Staff.midiInstrument="pad 2 (warm)" \melodie
\set Staff.midiInstrument="pad 3 (polysynth)" \melodie
\set Staff.midiInstrument="pad 4 (choir)" \melodie
\set Staff.midiInstrument="pad 5 (bowed)" \melodie
\set Staff.midiInstrument="pad 6 (metallic)" \melodie
\set Staff.midiInstrument="pad 7 (halo)" \melodie
\set Staff.midiInstrument="pad 8 (sweep)" \melodie
\set Staff.midiInstrument="fx 1 (rain)" \melodie
\set Staff.midiInstrument="fx 2 (soundtrack)" \melodie
\set Staff.midiInstrument="fx 3 (crystal)" \melodie
\set Staff.midiInstrument="fx 4 (atmosphere)" \melodie
\set Staff.midiInstrument="fx 5 (brightness)" \melodie
\set Staff.midiInstrument="fx 6 (goblins)" \melodie
\set Staff.midiInstrument="fx 7 (echoes)" \melodie
\set Staff.midiInstrument="fx 8 (sci-fi)" \melodie
\set Staff.midiInstrument="sitar" \melodie
\set Staff.midiInstrument="banjo" \melodie
\set Staff.midiInstrument="shamisen" \melodie
\set Staff.midiInstrument="koto" \melodie
\set Staff.midiInstrument="kalimba" \melodie
\set Staff.midiInstrument="bagpipe" \melodie
```

```
\set Staff.midiInstrument="fiddle" \melodie
\set Staff.midiInstrument="shanai" \melodie
\set Staff.midiInstrument="tinkle bell" \melodie
\set Staff.midiInstrument="agogo" \melodie
\set Staff.midiInstrument="steel drums" \melodie
\set Staff.midiInstrument="woodblock" \melodie
\set Staff.midiInstrument="taiko drum" \melodie
\set Staff.midiInstrument="melodic tom" \melodie
\set Staff.midiInstrument="synth drum" \melodie
\set Staff.midiInstrument="reverse cymbal" \melodie
\set Staff.midiInstrument="guitar fret noise" \melodie
\set Staff.midiInstrument="breath noise" \melodie
\set Staff.midiInstrument="seashore" \melodie
\set Staff.midiInstrument="bird tweet" \melodie
\set Staff.midiInstrument="telephone ring" \melodie
\set Staff.midiInstrument="helicopter" \melodie
\set Staff.midiInstrument="applause" \melodie
\set Staff.midiInstrument="gunshot" \melodie
} %Voice
>> %Staff
\midi { }
} %score
```

Demo of all midi sounds

Myself



'jazz-combo-template.ly'

Jazz tune for combo (horns, guitar, piano, bass, drums).

Song
(tune)

Me

moderato

Swing

Trumpet

Alto Sax

Bari Sax

Trombone

Guitar

Piano

Bass

Drums

B^{Δ} Solo $C\sharp m^7$

Cm^{Δ} $D^{\Delta/9}$

'piano-template-simple.ly'

Here is a simple piano staff with some notes.

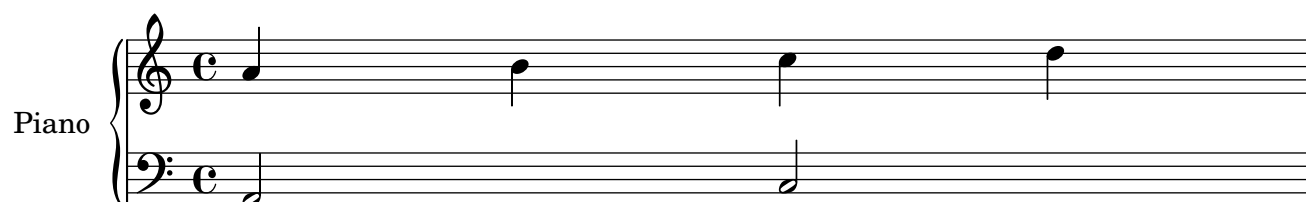
```
upper = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

  a b c d
}

lower = \relative c {
  \clef bass
  \key c \major
  \time 4/4

  a2 c
}

\score {
  \new PianoStaff <<
    \set PianoStaff.instrumentName = "Piano "
    \new Staff = "upper" \upper
    \new Staff = "lower" \lower
  >>
  \layout { }
  \midi { }
}
```



`'piano-template-with-centered-dynamics.ly'`

Many piano scores have the dynamics centered between the two staves. This requires a bit of tweaking to implement, but since the template is right here, you don't have to do the tweaking yourself.

```
upper = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

  a b c d
}

lower = \relative c {
  \clef bass
  \key c \major
  \time 4/4

  a2 c
}

dynamics = {
  s2\fff\> s4
  s\!\pp
}

pedal = {
  s2\sustainDown s2\sustainUp
}

\score {
  \new PianoStaff <<
    \new Staff = "upper" \upper
    \new Dynamics = "dynamics" \dynamics
    \new Staff = "lower" <<
      \clef bass
      \lower
    >>
    \new Dynamics = "pedal" \pedal
  >>
  \layout {
    \context {
      \type "Engraver_group"
```

```

\name Dynamics
\alias Voice % So that \cresc works, for example.
\consists "Output_property_engraver"

\override VerticalAxisGroup #'minimum-Y-extent = #'(-1 . 1)
\override DynamicLineSpanner #'Y-offset = #0
pedalSustainStrings = #'("Ped." "*Ped." "*")
pedalUnaCordaStrings = #'("una corda" "" "tre corde")

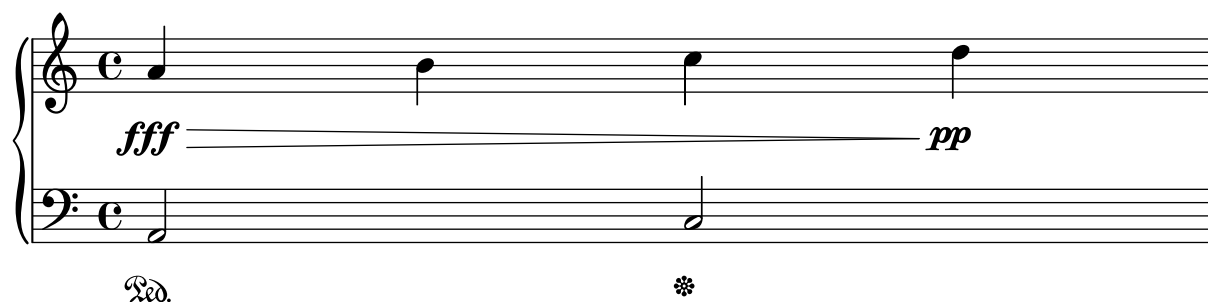
\consists "Piano_pedal_engraver"
\consists "Script_engraver"
\consists "Dynamic_engraver"
\consists "Text_engraver"

\override TextScript #'font-size = #2
\override TextScript #'font-shape = #'italic

\consists "Skip_event_swallow_translator"

\consists "Axis_group_engraver"
}
\context {
  \PianoStaff
  \accepts Dynamics
}
}
\score {
  \new PianoStaff <<
    \new Staff = "upper" << \upper \dynamics >>
    \new Staff = "lower" << \lower \dynamics >>
    \new Dynamics = "pedal" \pedal
  >>
  \midi {
    \context {
      \type "Performer_group"
      \name Dynamics
      \consists "Piano_pedal_performer"
    }
    \context {
      \PianoStaff
      \accepts Dynamics
    }
  }
}
}

```



‘piano-template-with-centered-lyrics.ly’

Instead of having a full staff for the melody and lyrics, you can place the lyrics between the piano staff (and omit the separate melody staff).

```
upper = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

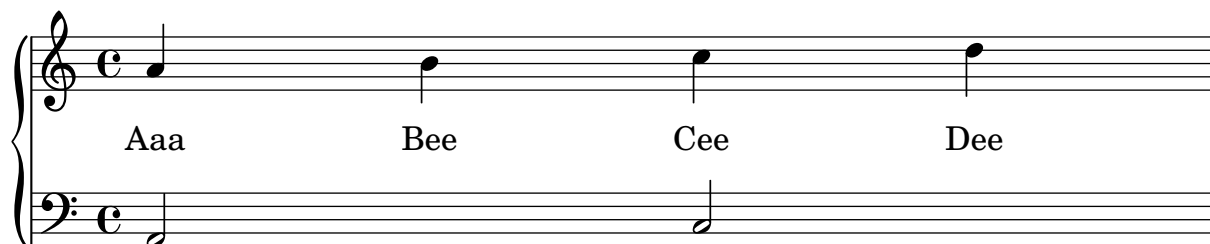
  a b c d
}

lower = \relative c {
  \clef bass
  \key c \major
  \time 4/4

  a2 c
}

text = \lyricmode {
  Aaa Bee Cee Dee
}

\score {
  \new GrandStaff <<
    \new Staff = upper { \new Voice = "singer" \upper }
    \new Lyrics \lyricsto "singer" \text
    \new Staff = lower {
      \clef bass
      \lower
    }
  >>
  \layout {
    \context { \GrandStaff \accepts "Lyrics" }
    \context { \Lyrics \consists "Bar_engraver" }
  }
  \midi { }
}
```



‘piano-template-with-melody-and-lyrics.ly’

Here is a typical song format: one staff with the melody and lyrics, with piano accompaniment underneath.

```
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

    a b c d
}

lower = \relative c {
    \clef bass
    \key c \major
    \time 4/4

    a2 c
}

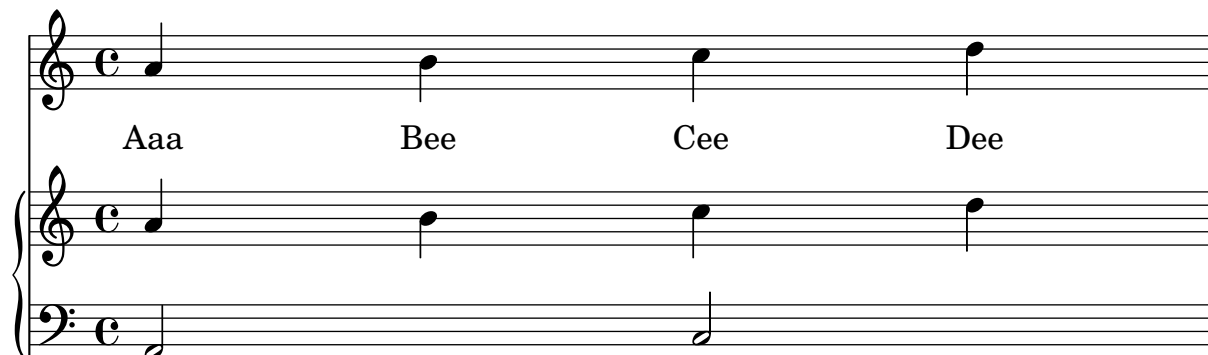
\score {
    <<
        \new Voice = "mel" {
            \autoBeamOff
            \melody
        }
        \new Lyrics \lyricsto mel \text

        \new PianoStaff <<
            \new Staff = "upper" \upper
            \new Staff = "lower" \lower
        >>
    >>
    \layout {
```

```

        \context { \RemoveEmptyStaffContext }
    }
    \midi { }
}

```



‘vocal-ensemble-template-with-automatic-piano-reduction.ly’

This template adds an automatic piano reduction to the standard SATB vocal score demonstrated in "Vocal ensemble template". This demonstrates one of the strengths of LilyPond – you can use a music definition more than once. If you make any changes to the vocal notes (say, `tenorMusic`), then the changes will also apply to the piano reduction.

```

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 Lyrics = sopranos { s1 }
      \new Staff = women <<
        \new Voice =
          "sopranos" { \voiceOne << \global \sopMusic >> }
        \new Voice =
          "altos" { \voiceTwo << \global \altoMusic >> }
      >>
      \new Lyrics = "altos" { s1 }
      \new Lyrics = "tenors" { s1 }
      \new Staff = men <<
        \clef bass
        \new Voice =
          "tenors" { \voiceOne <<\global \tenorMusic >> }
        \new Voice =
          "basses" { \voiceTwo <<\global \bassMusic >> }
      >>
      \new Lyrics = basses { s1 }

      \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 >>
      >>
    >>
  >>
  \layout {
    \context {
      % a little smaller so lyrics
      % can be closer to the staff
      \Staff
    }
  }
}

```

```
\override VerticalAxisGroup #'minimum-Y-extent = #'(-3 . 3)
```

```
}  
}  
}
```

The image shows a musical score for piano, featuring two systems of staves. Each system consists of a grand staff (treble and bass clef) and two vocal staves (treble and bass clef). The time signature is common time (C). The lyrics are: hi, ha, hu, ho. The first system shows the vocal lines and piano accompaniment. The second system shows the vocal lines and piano accompaniment. The lyrics are: hi, ha, hu, ho. The piano accompaniment consists of chords in the bass and single notes in the treble. The vocal lines are simple, with the lyrics written below the notes.

hi ha hu ho

hi ha hu ho

Percussion

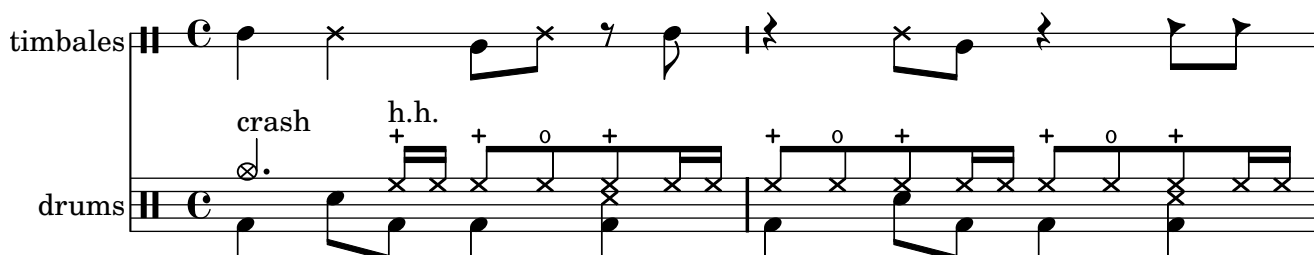
‘adding-drum-parts.ly’

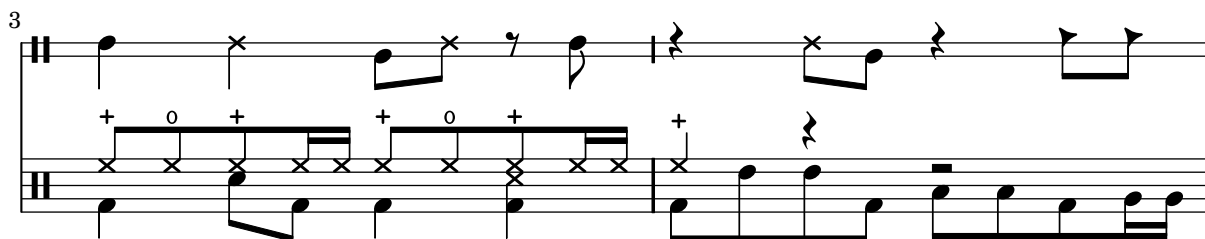
LilyPond makes drums input quite easy, with powerful pre-configured tools such as the `\drummode` function and the `DrumStaff` context: drums are placed to their own staff positions (with a special clef symbol) and have note heads according to the drum. You can easily attach an extra symbol to the drum, and restrict the number of lines.

```
drh = \drummode { cymc4.^"crash" hhc16^"h.h." hh \repeat "unfold" 5 {hhc8 hho hhc8 hh16 hh}
drl = \drummode {\repeat "unfold" 3 {bd4 sn8 bd bd4 << bd ss >> } bd8 tommh tommh bd toml toml}
timb = \drummode { \repeat "unfold" 2 {timh4 ssh timl8 ssh r timh r4 ssh8 timl r4 cb8 cb} }
```

```
\score {
  \repeat "volta" 2 {
    <<
      \new DrumStaff \with {
        drumStyleTable = #timbales-style
        \override StaffSymbol #'line-count = #2
        \override BarLine #'bar-size = #2
      } <<
        \set Staff.instrumentName = "timbales"
        \timb
      >>
      \new DrumStaff <<
        \set Staff.instrumentName = "drums"
        \new DrumVoice {\stemUp \drh }
        \new DrumVoice {\stemDown \drl }
      >>
    >>
  }
  \layout {}

  \midi {
    \context {
      \Score
      tempoWholesPerMinute = #(ly:make-moment 120 4)
    }
  }
}
```





'heavily-customized-polymetric-time-signatures.ly'

Though the set-time-signature thing was not the most essential here, it has been included to show the beat of this piece (which is a template of a real balkan song!).

```
#(define (compound-time one two three four five six seven eight nine ten num)
```

```
(markup #:override '(baseline-skip . 0) #:number
```

```
(#:line ((#:column (one num)) #:vcenter "+" (#:column (two num)) #:vcenter "+" (#:column
```

```
melody =
```

```
{
  \relative c'' {
    \set Staff.instrumentName = "Bb Sop."
    \key g \major \time 25/8
    \override Staff.TimeSignature #'stencil = #ly:text-interface::print
    \override Staff.TimeSignature #'text = #(compound-time "3" "2" "2" "3" "2" "2" "2" "3" "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
      c4. d4 c4 d4. c4 d c2 d4. e4-^ d4 |
      c4. d4 c4 d4. c4 d c2 d4. e4-^ d4 | \break }
}
```

```
drum = {
  \new DrumStaff \drummode
  {
    \bar "|:" bd4. ^\markup { "Drums" } sn4 bd \bar ":" sn4. bd4 sn \bar ":"
    bd sn bd4. sn4 bd \bar ":@"
  }
}
```

```
{
  \melody
  \drum
}
```



6

Drums



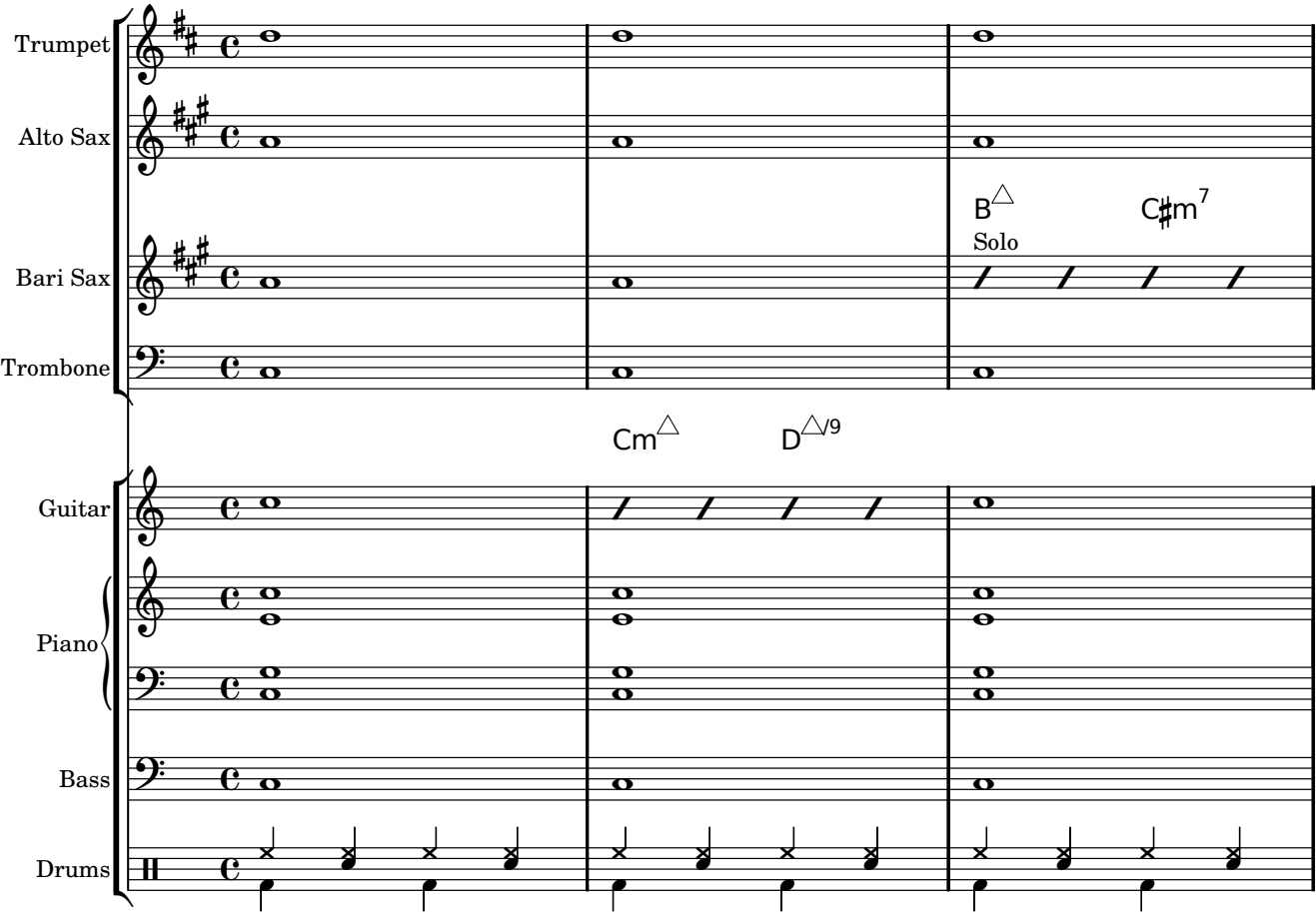
`'jazz-combo-template.ly'`
Jazz tune for combo (horns, guitar, piano, bass, drums).

Song
(tune)

Me

moderato

Swing



`'printing-music-with-different-time-signatures.ly'`

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

This can be achieved with the `\compressMusic` command, as demonstrated here.

The barlines can't be printed at the Score level anymore, so you have to remove the `Barline_engraver` and put it in the Staff context.

% Thanks to Adam James Wilson for this snippet

```
\paper {
  indent = #0
  ragged-right = ##t
}

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

\layout {
  \context { \Score
    \remove "Timing_translator"
    \remove "Time_signature_engraver"
    \remove "Default_bar_line_engraver"
    \override SpacingSpanner #'uniform-stretching = ##t
    \override SpacingSpanner #'strict-note-spacing = ##t
    proportionalNotationDuration = #(ly:make-moment 1 64)
  }
  \context { \Staff
    \consists "Timing_translator"
    \consists "Default_bar_line_engraver"
    \consists "Time_signature_engraver"
  }
  \context { \Voice
    \remove Forbid_line_break_engraver
    tupletFullLength = ##t
  }
}
```

```
Bassklarinette = \new Staff <<
  \global
  {
    \bar "|"
    \clef treble
    \time 3/8
    d''4.

    \bar "|"
    \time 3/4
    r8 des''2( c''8)

    \bar "|"
    \time 7/8
    r4. ees''2 ~

    \bar "|"
    \time 2/4
    \tupletUp
    \times 2/3 {ees''4 r4 d''4 ~}

    \bar "|"
```

```

\time 3/8
\tupletUp
\times 3/4 {d''4 r4}

\bar "|"
\time 2/4
e''2

\bar "|"
\time 3/8
es''4.
\bar "|"
\time 3/4
r8 d''2 r8
\bar "|"
}

>>
Percussion = \new StaffGroup <<
\new Staff <<
\global
{
\bar "|"
\clef percussion
\time 3/4
r4 c'2 ~

\bar "|"
c'2.

\bar "|"
R2.

\bar "|"
r2 g'4 ~

\bar "|"
g'2. ~

\bar "|"
g'2.
}
>>
\new Staff <<
\global
{
\bar "|"
\clef percussion
\time 3/4
R2.

\bar "|"
g'2. ~

```

```
\bar "|"
g'2.
```

```
\bar "|"
r4 g'2 ~
```

```
\bar "|"
g'2 r4
```

```
\bar "|"
g'2.
}
```

```
>>
```

```
>>
```

```
\score { << \Bassklarinette \Percussion >>
```

```
}
```

Musical score for Percussion, page 125. The score consists of three staves. The top staff is a single treble clef staff in 3/4 time, containing a quarter rest, a quarter note, and a quarter rest. The bottom two staves are grouped by a brace on the left and each begins with a double bar line (H), indicating they are for a different instrument or part. Each of these two staves contains a half note.

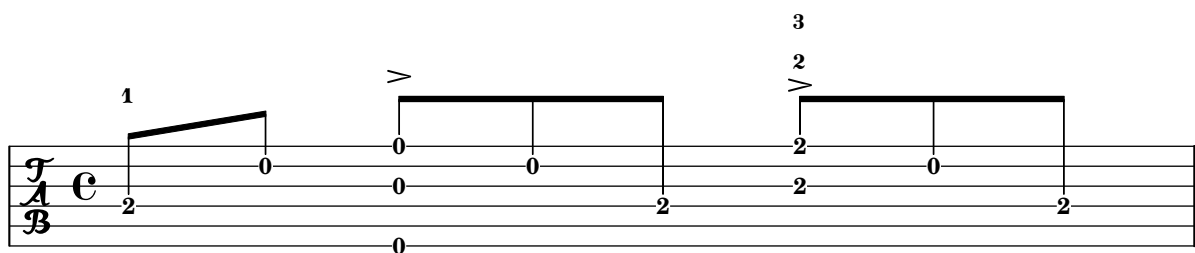
Guitar

‘adding-fingerings-to-tablatures.ly’

To add fingerings to tablatures, you can use a combination of `\markup` and `\finger`.

```
one = \markup{ \finger "1" }
two = \markup{ \finger "2" }
threetwo = \markup{ \column {\finger "3" \finger "2"} }
threefour = \markup{ \column {\finger "3" \finger "4"} }

\score {
  \context TabStaff {
    \stemUp
    e8\4^\one b\2 < e, g\3 e'\1 >^\threetwo[ b\2 e\4]
    < a\3 fis'\1 >^\threefour[ b\2 e\4]
  }
}
```



‘jazz-combo-template.ly’

Jazz tune for combo (horns, guitar, piano, bass, drums).

Song
(tune)

Me

moderato

Swing

Trumpet

Alto Sax

Bari Sax

Trombone

Guitar

Piano

Bass

Drums

B^{Δ} Solo

$C\sharp m^7$

Cm^{Δ}

$D^{\Delta/9}$

'letter-tablature-formatting.ly'

You can format a tablature with letters instead of numbers - so that 0->a, 1->b, 2->c, etc.

BROKEN IN 2.11, COMMENTED OUT.

%% Do not edit this file; it is auto-generated from input/new

%% This file is in the public domain.

\version "2.11.38"

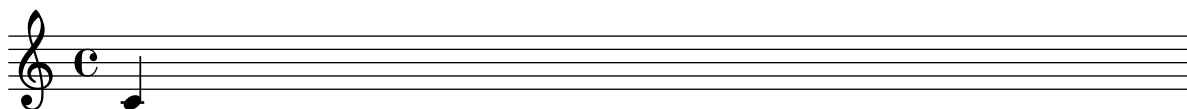
```
\header {
  doctitle = "Letter tablature formatting"
  lsrtags = "guitar"
  texidoc = "
```

You can format a tablature with letters instead of numbers - so that
0->a, 1->b, 2->c, etc.

BROKEN IN 2.11, COMMENTED OUT."

}

{ c'4 }



Strings

‘demo-midiinstruments.ly’

Problem: How to know which MidiInstrument would be best for your composition? Solution: A Lilypond demo file.

```
\score {
  \new Staff <<
    \new Voice { \melodie
    } %Voice
  >> %Staff
  \layout { }
} %score
```

```
\score {
  \new Staff <<
    \tempo 4 = 180
    %\set Staff.instrumentName="S/A"
    %\set Staff.midiMinimumVolume = #0.2
    %\set Staff.midiMaximumVolume = #0.4
    %\set Voice.dynamicAbsoluteVolumeFunction = #0.6
    \new Voice { r \mf
      \set Staff.midiInstrument="acoustic grand" \melodie
      \set Staff.midiInstrument="bright acoustic" \melodie
      \set Staff.midiInstrument="electric grand" \melodie
      \set Staff.midiInstrument="honky-tonk" \melodie
      \set Staff.midiInstrument="electric piano 1" \melodie
      \set Staff.midiInstrument="electric piano 2" \melodie
      \set Staff.midiInstrument="harpsichord" \melodie
      \set Staff.midiInstrument="clav" \melodie
      \set Staff.midiInstrument="celesta" \melodie
      \set Staff.midiInstrument="glockenspiel" \melodie
      \set Staff.midiInstrument="music box" \melodie
      \set Staff.midiInstrument="vibraphone" \melodie
      \set Staff.midiInstrument="marimba" \melodie
      \set Staff.midiInstrument="xylophone" \melodie
      \set Staff.midiInstrument="tubular bells" \melodie
      \set Staff.midiInstrument="dulcimer" \melodie
      \set Staff.midiInstrument="drawbar organ" \melodie
      \set Staff.midiInstrument="percussive organ" \melodie
      \set Staff.midiInstrument="rock organ" \melodie
      \set Staff.midiInstrument="church organ" \melodie
      \set Staff.midiInstrument="reed organ" \melodie
      \set Staff.midiInstrument="accordion" \melodie
      \set Staff.midiInstrument="harmonica" \melodie
      \set Staff.midiInstrument="concertina" \melodie
      \set Staff.midiInstrument="acoustic guitar (nylon)" \melodie
      \set Staff.midiInstrument="acoustic guitar (steel)" \melodie
      \set Staff.midiInstrument="electric guitar (jazz)" \melodie
      \set Staff.midiInstrument="electric guitar (clean)" \melodie
      \set Staff.midiInstrument="electric guitar (muted)" \melodie
      \set Staff.midiInstrument="overdriven guitar" \melodie
    }
  >>
}
```

```
\set Staff.midiInstrument="distorted guitar" \melodie
\set Staff.midiInstrument="acoustic bass" \melodie
\set Staff.midiInstrument="electric bass (finger)" \melodie
\set Staff.midiInstrument="electric bass (pick)" \melodie
\set Staff.midiInstrument="fretless bass" \melodie
\set Staff.midiInstrument="slap bass 1" \melodie
\set Staff.midiInstrument="slap bass 2" \melodie
\set Staff.midiInstrument="synth bass 1" \melodie
\set Staff.midiInstrument="synth bass 2" \melodie
\set Staff.midiInstrument="violin" \melodie
\set Staff.midiInstrument="viola" \melodie
\set Staff.midiInstrument="cello" \melodie
\set Staff.midiInstrument="contrabass" \melodie
\set Staff.midiInstrument="tremolo strings" \melodie
\set Staff.midiInstrument="pizzicato strings" \melodie
\set Staff.midiInstrument="orchestral strings" \melodie
\set Staff.midiInstrument="timpani" \melodie
\set Staff.midiInstrument="string ensemble 1" \melodie
\set Staff.midiInstrument="string ensemble 2" \melodie
\set Staff.midiInstrument="synthstrings 1" \melodie
\set Staff.midiInstrument="synthstrings 2" \melodie
\set Staff.midiInstrument="choir aahs" \melodie
\set Staff.midiInstrument="voice oohs" \melodie
\set Staff.midiInstrument="synth voice" \melodie
\set Staff.midiInstrument="orchestra hit" \melodie
\set Staff.midiInstrument="trumpet" \melodie
\set Staff.midiInstrument="trombone" \melodie
\set Staff.midiInstrument="tuba" \melodie
\set Staff.midiInstrument="muted trumpet" \melodie
\set Staff.midiInstrument="french horn" \melodie
\set Staff.midiInstrument="brass section" \melodie
\set Staff.midiInstrument="synthbrass 1" \melodie
\set Staff.midiInstrument="synthbrass 2" \melodie
\set Staff.midiInstrument="soprano sax" \melodie
\set Staff.midiInstrument="alto sax" \melodie
\set Staff.midiInstrument="tenor sax" \melodie
\set Staff.midiInstrument="baritone sax" \melodie
\set Staff.midiInstrument="oboe" \melodie
\set Staff.midiInstrument="english horn" \melodie
\set Staff.midiInstrument="bassoon" \melodie
\set Staff.midiInstrument="clarinet" \melodie
\set Staff.midiInstrument="piccolo" \melodie
\set Staff.midiInstrument="flute" \melodie
\set Staff.midiInstrument="recorder" \melodie
\set Staff.midiInstrument="pan flute" \melodie
\set Staff.midiInstrument="blown bottle" \melodie
\set Staff.midiInstrument="shakuhachi" \melodie
\set Staff.midiInstrument="whistle" \melodie
\set Staff.midiInstrument="ocarina" \melodie
\set Staff.midiInstrument="lead 1 (square)" \melodie
\set Staff.midiInstrument="lead 2 (sawtooth)" \melodie
\set Staff.midiInstrument="lead 3 (calliope)" \melodie
```

```

\set Staff.midiInstrument="lead 4 (chiff)" \melodie
\set Staff.midiInstrument="lead 5 (charang)" \melodie
\set Staff.midiInstrument="lead 6 (voice)" \melodie
\set Staff.midiInstrument="lead 7 (fifths)" \melodie
\set Staff.midiInstrument="lead 8 (bass+lead)" \melodie
\set Staff.midiInstrument="pad 1 (new age)" \melodie
\set Staff.midiInstrument="pad 2 (warm)" \melodie
\set Staff.midiInstrument="pad 3 (polysynth)" \melodie
\set Staff.midiInstrument="pad 4 (choir)" \melodie
\set Staff.midiInstrument="pad 5 (bowed)" \melodie
\set Staff.midiInstrument="pad 6 (metallic)" \melodie
\set Staff.midiInstrument="pad 7 (halo)" \melodie
\set Staff.midiInstrument="pad 8 (sweep)" \melodie
\set Staff.midiInstrument="fx 1 (rain)" \melodie
\set Staff.midiInstrument="fx 2 (soundtrack)" \melodie
\set Staff.midiInstrument="fx 3 (crystal)" \melodie
\set Staff.midiInstrument="fx 4 (atmosphere)" \melodie
\set Staff.midiInstrument="fx 5 (brightness)" \melodie
\set Staff.midiInstrument="fx 6 (goblins)" \melodie
\set Staff.midiInstrument="fx 7 (echoes)" \melodie
\set Staff.midiInstrument="fx 8 (sci-fi)" \melodie
\set Staff.midiInstrument="sitar" \melodie
\set Staff.midiInstrument="banjo" \melodie
\set Staff.midiInstrument="shamisen" \melodie
\set Staff.midiInstrument="koto" \melodie
\set Staff.midiInstrument="kalimba" \melodie
\set Staff.midiInstrument="bagpipe" \melodie
\set Staff.midiInstrument="fiddle" \melodie
\set Staff.midiInstrument="shanai" \melodie
\set Staff.midiInstrument="tinkle bell" \melodie
\set Staff.midiInstrument="agogo" \melodie
\set Staff.midiInstrument="steel drums" \melodie
\set Staff.midiInstrument="woodblock" \melodie
\set Staff.midiInstrument="taiko drum" \melodie
\set Staff.midiInstrument="melodic tom" \melodie
\set Staff.midiInstrument="synth drum" \melodie
\set Staff.midiInstrument="reverse cymbal" \melodie
\set Staff.midiInstrument="guitar fret noise" \melodie
\set Staff.midiInstrument="breath noise" \melodie
\set Staff.midiInstrument="seashore" \melodie
\set Staff.midiInstrument="bird tweet" \melodie
\set Staff.midiInstrument="telephone ring" \melodie
\set Staff.midiInstrument="helicopter" \melodie
\set Staff.midiInstrument="applause" \melodie
\set Staff.midiInstrument="gunshot" \melodie
} %Voice
>> %Staff
\midi { }
} %score

```

Demo of all midi sounds

Myself



‘string-quartet-template-simple.ly’

This template demonstrates a simple string quartet. It also uses a `\global` section for time and key signatures

```
global= {
  \time 4/4
  \key c \major
}

violinOne = \new Voice { \relative c''{
  \set Staff.instrumentName = "Violin 1 "

  c2 d e1

  \bar "|" }}
violinTwo = \new Voice { \relative c''{
  \set Staff.instrumentName = "Violin 2 "

  g2 f e1

  \bar "|" }}
viola = \new Voice { \relative c' {
  \set Staff.instrumentName = "Viola "
  \clef alto

  e2 d c1

  \bar "|" }}
cello = \new Voice { \relative c' {
  \set Staff.instrumentName = "Cello "
  \clef bass

  c2 b a1

  \bar "|" }}

\score {
  \new StaffGroup <<
    \new Staff << \global \violinOne >>
    \new Staff << \global \violinTwo >>
    \new Staff << \global \viola >>
    \new Staff << \global \cello >>
  >>
```

```

\layout { }
\midi { }
}

```

‘string-quartet-template-with-separate-parts.ly’

The "String quartet template" snippet produces a nice string quartet, but what if you needed to print parts? This new template demonstrates how to use the `\tag` feature to easily split a piece into individual parts.

You need to split this template into separate files; the filenames are contained in comments at the beginning of each file. `piece.ly` contains all the music definitions. The other files – `score.ly`, `vn1.ly`, `vn2.ly`, `vla.ly`, and `vlc.ly` – produce the appropriate part.

Do not forget to remove specified comments when using separate files!

```

%% piece.ly
%% (This is the global definitions file)

global= {
  \time 4/4
  \key c \major
}

Violinone = \new Voice { \relative c' {
  \set Staff.instrumentName = "Violin 1 "

  c2 d e1

  \bar "|" } } %*****
Violintwo = \new Voice { \relative c' {
  \set Staff.instrumentName = "Violin 2 "

  g2 f e1

  \bar "|" } } %*****
Viola = \new Voice { \relative c' {
  \set Staff.instrumentName = "Viola "
  \clef alto
}

```

```

e2 d c1

\bar "|" } } %*****
Cello = \new Voice { \relative c' {
  \set Staff.instrumentName = "Cello      "
  \clef bass

c2 b a1

\bar "|" } } %*****

music = {
  <<
    \tag #'score \tag #'vn1 \new Staff { << \global \Violinone >> }
    \tag #'score \tag #'vn2 \new Staff { << \global \Violintwo>> }
    \tag #'score \tag #'vla \new Staff { << \global \Viola>> }
    \tag #'score \tag #'vlc \new Staff { << \global \Cello>> }
  >>
}

%%% These are the other files you need to save on your computer

%%%%% score.ly
%%%%% (This is the main file)

%\include "piece.ly"          %%% uncomment this line when using a separate file
#(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

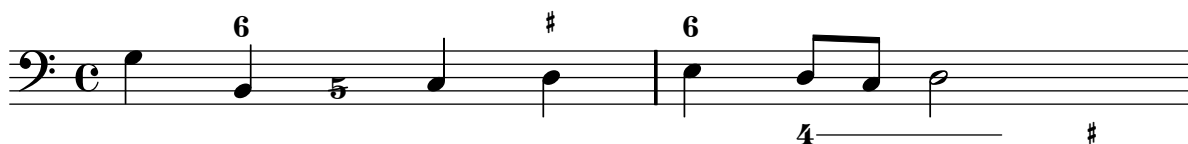
Ancient notation

‘adding-a-figured-bass-above-or-below-the-notes.ly’

When writing a figured bass, here’s a way to specify if you want your figures to be placed above or below the bass notes, by defining the `BassFigureAlignmentPositioning` `#'direction` property (exclusively in a `Staff` context). Choices are `#UP` (or `#1`), `#CENTER` (or `#0`) and `#DOWN` (or `#-1`).

As you can see here, this property can be changed as many times as you wish. Use `\once \override` if you don’t want the tweak to apply to the whole score.

```
bass = { \clef bass g4 b, c d e d8 c d2}
continuo = \figuremode {
  < _ >4 < 6 >8
  \once \override Staff.BassFigureAlignmentPositioning #'direction = #CENTER
  <5/> < _ >4
  \override Staff.BassFigureAlignmentPositioning #'direction = #UP
  < _+ > < 6 >
  \set Staff.useBassFigureExtenders = ##t
  \override Staff.BassFigureAlignmentPositioning #'direction = #DOWN
  < 4 >4. < 4 >8 < _+ >4
}
\score {
  << \new Staff = bassStaff \bass
  \context Staff = bassStaff \continuo >>
}
```



‘ancient-fonts.ly’

Here are shown many (all?) of the symbols that are included in LilyPond’s support of ancient notation.

```
upperStaff = \context GregorianStaff = "upperStaff" <<
  \context GregorianVoice <<
    \set Score.timing = ##f
%   \set Score.forceAccidental = ##t %%%%%%%%%% FIXME: what happened to this property?

  \override Staff.StaffSymbol #'line-count = #4

  \transpose c c {
    \override Staff.KeySignature #'glyph-name-alist = #alteration-vaticana-glyph-name-a-
    \override Staff.Accidental #'glyph-name-alist = #alteration-vaticana-glyph-name-ali
    \override NoteHead #'style = #'vaticana.punctum
    \key es \major
    \clef "vaticana-fa2"
    c!1 des! e! f! ges!

    \override NoteHead #'style = #'vaticana.inclinatum
```

```

a! b! ces'
\override Staff.BarLine #'bar-size = #3.0 \bar "|"
% \break % 1 (8*1)

\override NoteHead #'style = #'vaticana.quilisma
b! des'! ges! fes!
\breath
\clef "vaticana-fa1"
\override NoteHead #'style = #'vaticana.plica
es d
\override NoteHead #'style = #'vaticana.reverse-plica
c d
\override Staff.BarLine #'bar-size = #3.0 \bar "|"
% \break %2 (8*1)

\override NoteHead #'style = #'vaticana.punctum-cavum
es f
\override NoteHead #'style = #'vaticana.lpes
g as
\override NoteHead #'style = #'vaticana.upes
bes as
\override NoteHead #'style = #'vaticana.vupes
g f
\override NoteHead #'style = #'vaticana.linea-punctum
\override Staff.BarLine #'bar-size = #2.0 \bar "|"
% \break % 3 (8*1)

es d
\override NoteHead #'style = #'vaticana.epiphonus
c d
\override NoteHead #'style = #'vaticana.cephalicus
es f

\override Staff.KeySignature #'glyph-name-alist = #alteration-medicaea-glyph-name-alist
\override Staff.Accidental #'glyph-name-alist = #alteration-medicaea-glyph-name-alist
\override Staff.Custos #'style = #'medicaea
\override NoteHead #'style = #'medicaea.punctum
\clef "medicaea-fa2"
ces! des!
\override Staff.BarLine #'bar-size = #3.0 \bar "|"
% \break % 4 (8*1)

e! f! ges!
\clef "medicaea-do2"
\override NoteHead #'style = #'medicaea.inclinatum
a! b! ces'!
\override NoteHead #'style = #'medicaea.virga
b! a!
\override Staff.BarLine #'bar-size = #3.0 \bar "|"
% \break % 5 (8*1)

ges! fes!

```

```

\clef "medicaea-fa1"
\override NoteHead #'style = #'medicaea.rvirga
e! des! ces!

\override Staff.KeySignature #'glyph-name-alist = #alteration-hufnagel-glyph-name-a
\override Staff.Accidental #'glyph-name-alist = #alteration-hufnagel-glyph-name-alis
\override Staff.Custos #'style = #'hufnagel
\override NoteHead #'style = #'hufnagel.punctum
\clef "hufnagel-fa2"
ces! des! es!
\override Staff.BarLine #'bar-size = #3.0 \bar "|"
% \break % 6 (8*1)

fes! ges!
\clef "hufnagel-do2"
\override NoteHead #'style = #'hufnagel.lpes
as! bes! ces!
\override NoteHead #'style = #'hufnagel.virga
bes! as!
\override Staff.BarLine #'bar-size = #3.0 \bar "|"
% \break % 7 (8*1)

ges! fes!
\clef "hufnagel-do-fa"
\override NoteHead #'style = #'hufnagel.punctum
es! des! ces! des! es! fes!
\bar "||"
% \break % 8 (8*1)

s32*1
% \break % 12 (32*1)
}
>>
>>

lowerStaff = \context MensuralStaff = "lowerStaff" <<
\context MensuralVoice <<

% this is broken until further notice -- see refman
% \override Staff.StaffSymbol #'line-count = #5
\applyOutput #'Staff #(outputproperty-compatibility (make-type-checker 'staff-symbol-int

\transpose c c {
  \set autoBeaming = ##f
  \override NoteHead #'style = #'neomensural
  \override Rest #'style = #'neomensural
  \key a \major

% FIXME: lily crashes on some (invalid?) ligatures with:
% ERROR: In procedure gh_scm2int:
% ERROR: Wrong type argument in position 1: ()

```

% FIXME: lily emits "Programming error: Infinity or NaN encountered"
 % on many ligatures such as BB.

```

      cis'1 d'\breve gis'\breve e'\breve \[ e'\longa fis'\longa \]
      \set Staff.forceClef = ##t
      \clef "neomensural-c2"
      cis1
      \bar "|"
%      \break % 2 (16*1)

      \[ g\breve dis''\longa \]
      b\breve \[ a\longa d\longa \]
      \clef "petrucci-c2"
%      \break % 4 (16*1)

      fis1 ces1
      \clef "petrucci-c2"
      r\longa
      \set Staff.forceClef = ##t
      \clef "mensural-c2"
      r\breve
      \bar "|"
%      \break % 5 (8*1)

      r2
      \clef "mensural-g"
      r4 r8 r16 r16
      \override NoteHead #'style = #'mensural
      \override Stem #'flag-style = #'mensural
      \override Stem #'thickness = #1.0
      \override Rest #'style = #'mensural
      \clef "petrucci-f"
      c8 b, c16 b, c32 b, c64 b, c64 b,
      d8 e d16 e d32 e d64 e d64 e
      r\longa
      \set Staff.forceClef = ##t
      \clef "petrucci-f"
      r\breve
      \bar "|"
%      \break % 6 (8*1)

      r\breve
      \clef "mensural-f"
      % FIXME: must set Stem flag-style to #'neomensural to avoid
      % segmentation fault on r8/r16/r32. (Strange: what has
      % Stem flag-style to do with mensural rests?)
      \override Stem #'flag-style = #'neomensural
      % FIXME: produces warnings about "flag `neomensurald4' (or 3) not found".
      r2 r4 r8 r16 r16
      \override Stem #'flag-style = #'mensural
      \set Staff.forceClef = ##t
      \clef "mensural-f"

```

```

e\breve f g a1
\clef "mensural-g"
% \break % 7 (8*1)

\[[ bes'!\longa a'!\longa c'!\longa \]
e'1 d' c' d' \bar "|"
\bar "|"
% \break % 9 (16*1)

bes'!\longa fis'!1 as'!1 ges'!\longa % lig
\set Staff.forceClef = ##t
\clef "mensural-g"
e'2 d' c' \bar "|"
% \break % 11 (16*1)

\set Staff.forceClef = ##t
\clef "petrucci-g"
c'2 d' e' f'
\clef "petrucci-g"
g' as'! bes'! cis'!
bes'! as'! gis'! fis'!
\set Staff.forceClef = ##t
\clef "mensural-g"
es'! des'! cis'!1 \bar "||"
% \break % 12 (8*1)
}
>>
>>

\paper {
  line-thickness = #(/ staff-space 5.0)
}

\score {
  \context Score <<
    \upperStaff
    \lowerStaff
  >>
  \layout {
% do we want to keep these settings? -gp
    line-width = 17.25\cm
    textheight = 26.0\cm
    indent = 0.0
    \context {
      \Score
      \accepts MensuralStaff
      \accepts GregorianStaff
%      timing = ##f %%%%% FIXME: this has no effect
    }
    \context {
      \Voice
      \name MensuralVoice

```

```

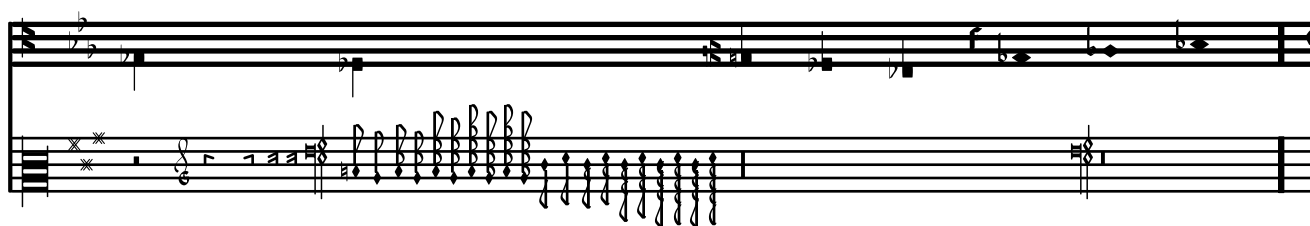
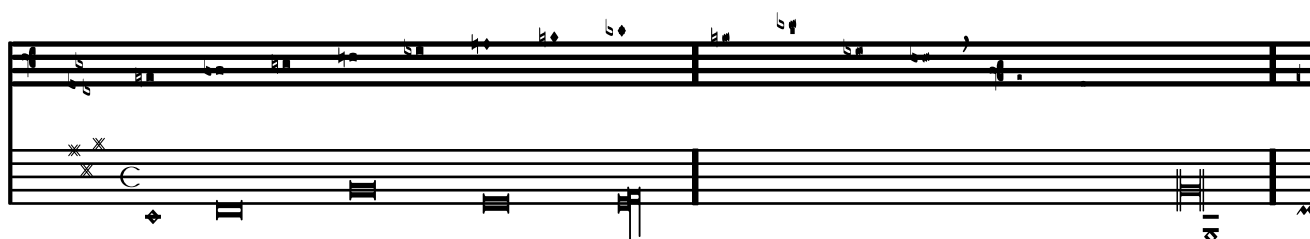
\alias Voice
\remove Ligature_bracket_engraver
\consists Mensural_ligature_engraver
\override NoteHead #'style = #'mensural
% \override Stem #'flag-style = #'mensural %%%%%%%%% FIXME: this core dumps
\override Stem #'thickness = #1.0
\override Rest #'style = #'mensural
autoBeaming = ##f
}
\context {
  \Voice
  \name GregorianVoice
  \alias Voice
  \remove Ligature_bracket_engraver
% \consists Gregorian_ligature_engraver %%%%%%%%% TODO: not yet implemented
\override NoteHead #'style = #'vaticana.punctum
autoBeaming = ##f
}
\context {
  \Staff
  \name MensuralStaff
  \alias Staff
  \accepts MensuralVoice
  \consists Custos_engraver
  \override TimeSignature #'style = #'mensural
  \override KeySignature #'glyph-name-alist = #alteration-mensural-glyph-name-alist
  \override Accidental #'glyph-name-alist = #alteration-mensural-glyph-name-alist
  \override Custos #'style = #'mensural
  \override Custos #'neutral-position = #3
  \override Custos #'neutral-direction = #-1
  clefGlyph = #"clefs.petrucchi-c2"
  clefPosition = #-2
  clefOctavation = #0
}
\context {
  \Staff
  \name GregorianStaff
  \alias Staff
  \accepts GregorianVoice
  \consists Custos_engraver
  \remove Time_signature_engraver
  \override StaffSymbol #'thickness = #2.0
  \override StaffSymbol #'line-count = #4
  \override KeySignature #'glyph-name-alist = #alteration-vaticana-glyph-name-alist
  \override Accidental #'glyph-name-alist = #alteration-vaticana-glyph-name-alist
  \override Custos #'style = #'vaticana
  \override Custos #'neutral-position = #4
  \override Custos #'neutral-direction = #-1
  clefGlyph = #"clefs.vaticana-do"
  clefPosition = #1
  clefOctavation = #0
}

```

```

\context {
  \RemoveEmptyStaffContext
  \accepts MensuralVoice
  \accepts GregorianVoice
}
}
}

```





`'ancient-notation-template----modern-transcription-of-gregorian-music.ly'`

This example demonstrates how to do modern transcription of Gregorian music. Gregorian music has no measure, no stems; it uses only half and quarter noteheads, and special marks, indicating rests of different length.

```
\include "gregorian-init.ly"
```

```
chant = \relative c' {
  \set Score.timing = ##f
  f4 a2 \divisioMinima
  g4 b a2 f2 \divisioMaior
  g4( f) f( g) a2 \finalis
}

verba = \lyricmode {
  Lo -- rem ip -- sum do -- lor sit a -- met
}

\score {
  \new Staff <<
    \new Voice = "melody" {
      \chant
    }
    \new Lyrics = "one" \lyricsto melody \verba
  >>

  \layout {
    \context {
      \Staff
      \remove "Time_signature_engraver"
      \remove "Bar_engraver"
      \override Stem #'transparent = ##t
    }
    \context {
      \Voice
      \override Stem #'length = #0
    }
    \context {
      \Score
      barAlways = ##t
    }
  }
}
```



Lorem ipsum dolor sit a-met


```
'ancient-notation-template---modern-transcription-of-mensural-music.ly'
```

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.

```
global = {
  \set Score.skipBars = ##t

  % incipit
  \once \override Score.SystemStartBracket #'transparent = ##t
  \override Score.SpacingSpanner #'spacing-increment = #1.0 % tight spacing
  \key f \major
  \time 2/2
  \once \override Staff.TimeSignature #'style = #'neomensural
  \override Voice.NoteHead #'style = #'neomensural
  \override Voice.Rest #'style = #'neomensural
  \set Staff.printKeyCancellation = ##f
  \cadenzaOn % turn off bar lines
  \skip 1*10
  \once \override Staff.BarLine #'transparent = ##f
  \bar "||"
  \skip 1*1 % need this extra \skip such that clef change comes
             % after bar line
  \bar ""

  % main
  \revert Score.SpacingSpanner #'spacing-increment % CHECK: no effect?
  \cadenzaOff % turn bar lines on again
  \once \override Staff.Clef #'full-size-change = ##t
  \set Staff.forceClef = ##t
  \key g \major
  \time 4/4
  \override Voice.NoteHead #'style = #'default
  \override Voice.Rest #'style = #'default

  % FIXME: setting printKeyCancellation back to #t must not
  % occur in the first bar after the incipit. Dto. for forceClef.
  % Therefore, we need an extra \skip.
  \skip 1*1
  \set Staff.printKeyCancellation = ##t
  \set Staff.forceClef = ##f

  \skip 1*7 % the actual music

  % let finis bar go through all staves
  \override Staff.BarLine #'transparent = ##f

  % finis bar
  \bar "|."
}
```

```

discantusNotes = {
  \transpose c' c'' {
    \set Staff.instrumentName = "Discantus  "

    % incipit
    \clef "neomensural-c1"
    c'1. s2  % two bars
    \skip 1*8 % eight bars
    \skip 1*1 % one bar

    % main
    \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 \override NoteHead #'transparent = ##t c'1 |
    b\breve |
  }
}

```

```

discantusLyrics = \lyricmode {
  % incipit
  IV-

  % main
  Ju -- bi -- |
  la -- te De -- |
  o, om --
  nis ter -- |
  ra, __ om- |
  "... " |
  -us. |
}

```

```

altusNotes = {
  \transpose c' c'' {
    \set Staff.instrumentName = "Altus  "

    % incipit
    \clef "neomensural-c3"
    r1          % one bar
    f1. s2      % two bars
    \skip 1*7 % seven bars
    \skip 1*1 % one bar

    % main
    \clef "treble"
    r2 g2. e4 fis g | % two bars
    a2 g4 e |
  }
}

```

```

        fis g4.( fis16 e fis4) |
        g1 |
        \once \override NoteHead #'transparent = ##t g1 |
        g\breve |
    }
}

altusLyrics = \lyricmode {
    % incipit
    IV-

    % main
    Ju -- bi -- la -- te | % two bars
    De -- o, om -- |
    nis ter -- ra, |
    "... " |
    -us. |
}

tenorNotes = {
    \transpose c' c' {
        \set Staff.instrumentName = "Tenor "

        % incipit
        \clef "neomensural-c4"
        r\longa    % four bars
        r\breve    % two bars
        r1         % one bar
        c'1. s2    % two bars
        \skip 1*1 % one bar
        \skip 1*1 % one bar

        % main
        \clef "treble_8"
        R1 |
        R1 |
        R1 |
        r2 d'2. d'4 b e' | % two bars
        \once \override NoteHead #'transparent = ##t e'1 |
        d'\breve |
    }
}

tenorLyrics = \lyricmode {
    % incipit
    IV-

    % main
    Ju -- bi -- la -- te | % two bars
    "... " |
    -us. |
}

```

```

bassusNotes = {
  \transpose c' c' {
    \set Staff.instrumentName = "Bassus  "

    % incipit
    \clef "bass"
    r\maxima % eight bars
    f1. s2   % two bars
    \skip 1*1 % one bar

    % main
    \clef "bass"
    R1 |
    R1 |
    R1 |
    R1 |
    g2. e4 |
    \once \override NoteHead #'transparent = ##t e1 |
    g\breve |
  }
}

bassusLyrics = \lyricmode {
  % incipit
  IV-

  % main
  Ju -- bi- |
  "... " |
  -us. |
}

\score {
  \new StaffGroup = choirStaff <<
    \new Voice =
      "discantusNotes" << \global \discantusNotes >>
    \new Lyrics =
      "discantusLyrics" \lyricsto discantusNotes { \discantusLyrics }
    \new Voice =
      "altusNotes" << \global \altusNotes >>
    \new Lyrics =
      "altusLyrics" \lyricsto altusNotes { \altusLyrics }
    \new Voice =
      "tenorNotes" << \global \tenorNotes >>
    \new Lyrics =
      "tenorLyrics" \lyricsto tenorNotes { \tenorLyrics }
    \new Voice =
      "bassusNotes" << \global \bassusNotes >>
    \new Lyrics =
      "bassusLyrics" \lyricsto bassusNotes { \bassusLyrics }
  >>

```

```

\layout {
  \context {
    \Score

    % no bars in staves
    \override BarLine #'transparent = ##t

    % incipit should not start with a start delimiter
    \remove "System_start_delimiter_engraver"
  }
  \context {
    \Voice

    % no slurs
    \override Slur #'transparent = ##t

    % Comment in the below "\remove" command to allow line
    % breaking also at those barlines where a note overlaps
    % into the next bar. 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"
  }
}

```

Discantus

IV-

Altus

IV-

Tenor

IV-

Bassus

IV-

Ju - bi - la - te De -

Ju - bi - la - te

Ju - bi - la - te

Ju - bi - la - te

‘ancient-time-signatures.ly’

Time signatures may also be engraved in an old style.

```
\score {
  {
    \override Staff.TimeSignature #'style = #'neomensural
    s1
  }
  \layout {raggedright = ##t}
}
```

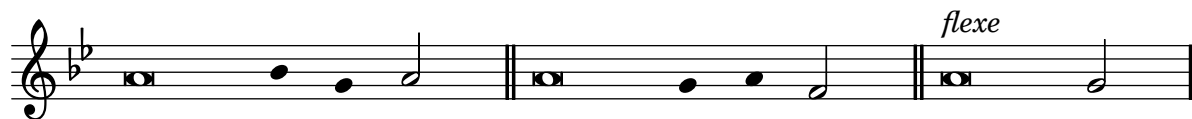
‘chant-or-psalms-notation.ly’

This form of notation is used for the chant of the Psalms, where verses aren’t always the same length.

```
stemon = { \override Staff.Stem #'transparent = ##f }
stemoff = { \override Staff.Stem #'transparent = ##t }
```

```
\score {
  \new Staff \with {\remove "Time_signature_engraver" }
  {
    \key g \minor
    \set Score.timing = ##f
    \stemoff a'\breve bes'4 g'4
    \stemon a'2 \bar "||"
    \stemoff a'\breve g'4 a'4
    \stemon f'2 \bar "||"
    \stemoff a'\breve^{\markup { \italic flexe }}
    \stemon g'2 \bar "||"
  }
}
```

```
\layout { raggedright = ##t}
}
```



```
'custodes.ly'
```

Custodes may be engraved in various styles.

```
\layout {
  \context {
    \Staff
    \consists Custos_engraver
  }
  ragged-right = ##t
}

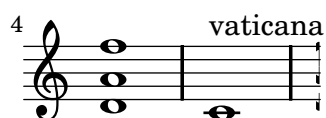
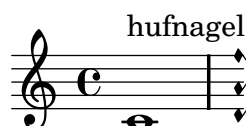
{
  \override Staff.Custos #'neutral-position = #4

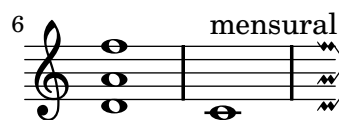
  \override Staff.Custos #'style = #'hufnagel
  c'1^"hufnagel"
  \break < d' a' f' >1

  \override Staff.Custos #'style = #'medicaea
  c'1^"medicaea"
  \break < d' a' f' >1

  \override Staff.Custos #'style = #'vaticana
  c'1^"vaticana"
  \break < d' a' f' >1

  \override Staff.Custos #'style = #'mensural
  c'1^"mensural"
  \break < d' a' f' >1
}
```





'rest-styles.ly'

Rests may be used in various styles.

```
\layout {
  indent = 0.0
  raggedright = ##t
}

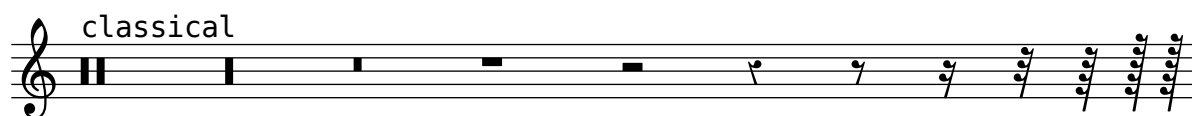
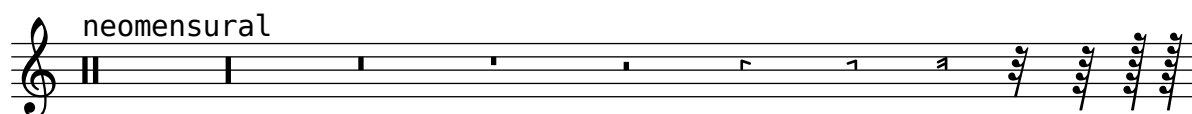
\context Staff \relative c {
  \set Score.timing = ##f
  \override Staff.Rest #'style = #'mensural
  r\maxima\markup \typewriter { mensural }
  r\longa r\breve r1 r2 r4 r8 r16 r32 r64 r128 r128
  \bar ""

  \override Staff.Rest #'style = #'neomensural
  r\maxima\markup \typewriter { neomensural }
  r\longa r\breve r1 r2 r4 r8 r16 r32 r64 r128 r128
  \bar ""

  \override Staff.Rest #'style = #'classical
  r\maxima\markup \typewriter { classical }
  r\longa r\breve r1 r2 r4 r8 r16 r32 r64 r128 r128
  \bar ""

  \override Staff.Rest #'style = #'default
  r\maxima\markup \typewriter { default }
  r\longa r\breve r1 r2 r4 r8 r16 r32 r64 r128 r128
}

```





`'transcription-of-ancient-music-with-incipit.ly'`

As a workaround to get real incipits which are independent from the main score these are included as a markup into the field normally used for the instrument name. As for now lyrics can only be added as a direct markup. It doesn't unfortunately conform with the spacing of the main lyrics.

```
global = {
  \set Score.skipBars = ##t
  \key g \major
  \time 4/4

  %make the staff lines invisible on staves
  \override Staff.BarLine #'transparent = ##t
  \skip 1*8 % the actual music

  % let finis bar go through all staves
  \override Staff.BarLine #'transparent = ##f

  % finis bar
  \bar "|."
}

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 \override NoteHead #'transparent = ##t c'1 |
    b\breve |
  }
}

discantusLyrics = \lyricmode {
  Ju -- bi -- |
  la -- te De -- |
  o, om --
  nis ter -- |
  ra, __ om- |
  "... |
  -us. |
}

altusNotes = {
  \transpose c' c'' {
    \clef "treble"
    r2 g2. e4 fis g | % two bars
  }
}
```

```

    a2 g4 e |
    fis g4.( fis16 e fis4) |
    g1 |
    \once \override NoteHead #'transparent = ##t g1 |
    g\breve |
  }
}

altusLyrics = \lyricmode {
  Ju -- bi -- la -- te | % two bars
  De -- o, om -- |
  nis ter -- ra, |
  "... " |
  -us. |
}

tenorNotes = {
  \transpose c' c' {
    \clef "treble_8"
    R1 |
    R1 |
    R1 |
    r2 d'2. d'4 b e' | % two bars
    \once \override NoteHead #'transparent = ##t e'1 |
    d'\breve |
  }
}

tenorLyrics = \lyricmode {
  Ju -- bi -- la -- te | % two bars
  "... " |
  -us.
}

bassusNotes = {
  \transpose c' c' {
    \clef "bass"
    R1 |
    R1 |
    R1 |
    R1 |
    g2. e4 |
    \once \override NoteHead #'transparent = ##t e1 |
    g\breve |
  }
}

bassusLyrics = \lyricmode {
  Ju -- bi- |
  "... " |
  -us.
}

```

```

incipitDiscantus = \markup{
  \score{
    {
      \set Staff.instrumentName="Discantus "
      \override NoteHead #'style = #'neomensural
      \override Rest #'style = #'neomensural
      \override Staff.TimeSignature #'style = #'neomensural
      \cadenzaOn
      \clef "neomensural-c1"
      \key f \major
      \time 2/2
      c''1._"IV-" s2 %two bars
      \skip 1*8 % eight bars
    }
  }
  \layout {
    \context {\Voice
      \remove Ligature_bracket_engraver
      \consists Mensural_ligature_engraver
    }
    line-width=4.5\cm
  }
}

```

```

incipitAltus = \markup{
  \score{
    {
      \set Staff.instrumentName="Altus "
      \override NoteHead #'style = #'neomensural
      \override Rest #'style = #'neomensural
      \override Staff.TimeSignature #'style = #'neomensural
      \cadenzaOn
      \clef "neomensural-c3"
      \key f \major
      \time 2/2
      r1 % one bar
      f'1._"IV-" s2 % two bars
      \skip 1*7 % seven bars
    }
  }
  \layout {
    \context {\Voice
      \remove Ligature_bracket_engraver
      \consists Mensural_ligature_engraver
    }
    line-width=4.5\cm
  }
}

```

```

incipitTenor = \markup{
  \score{ {

```

```

\set Staff.instrumentName = "Tenor "
\override NoteHead #'style = #'neomensural
  \override Rest #'style = #'neomensural
  \override Staff.TimeSignature #'style = #'neomensural
  \cadenzaOn
  \clef "neomensural-c4"
  \key f \major
  \time 2/2
r\longa % four bars
r\breve % two bars
r1 % one bar
c'1._"IV-" s2 % two bars
\skip 1 % one bar
}
\layout {
  \context {\Voice
    \remove Ligature_bracket_engraver
    \consists Mensural_ligature_engraver
  }
  line-width=4.5\cm
}
}
}

incipitBassus = \markup{
  \score{ {
    \set Staff.instrumentName = "Bassus "
    \override NoteHead #'style = #'neomensural
    \override Rest #'style = #'neomensural
    \override Staff.TimeSignature #'style = #'neomensural
    \cadenzaOn
    \clef "bass"
    \key f \major
    \time 2/2
% incipit
r\maxima % eight bars
f1._"IV-" s2 % two bars
}
  \layout {
    \context {\Voice
      \remove Ligature_bracket_engraver
      \consists Mensural_ligature_engraver
    }
    line-width=4.5\cm
  }
}
}

%StaffGroup is used instead of ChoirStaff to get bar lines between systems
\score {
  <<
  \new StaffGroup = choirStaff <<

```

```

\new Voice =
  "discantusNotes" << \global
  \set Staff.instrumentName=\incipitDiscantus
  \discantusNotes >>
\new Lyrics =
  "discantusLyrics" \lyricsto discantusNotes { \discantusLyrics }

\new Voice =
  "altusNotes" << \global
  \set Staff.instrumentName=\incipitAltus
  \altusNotes >>
\new Lyrics =
  "altusLyrics" \lyricsto altusNotes { \altusLyrics }

\new Voice =
  "tenorNotes" << \global
  \set Staff.instrumentName=\incipitTenor
  \tenorNotes >>
\new Lyrics =
  "tenorLyrics" \lyricsto tenorNotes { \tenorLyrics }

\new Voice =
  "bassusNotes" << \global
  \set Staff.instrumentName=\incipitBassus
  \bassusNotes >>
  >>
\new Lyrics =
  "bassusLyrics" \lyricsto bassusNotes { \bassusLyrics }
%Keep the bass lyrics outside of the staff group to avoid bar lines
%between the lyrics.
>>

\layout {
  \context {
    \Score

    % no bars in staves
    \override BarLine #'transparent = ##t
  }
  % the next three instructions keep the lyrics between the barlines
  \context { \Lyrics
    \consists "Bar_engraver"
    \override BarLine #'transparent = ##t }
  \context { \StaffGroup \consists "Separating_line_group_engraver" }
  \context {
    \Voice

    % no slurs
    \override Slur #'transparent = ##t

    % Comment in the below "\remove" command to allow line
    % breaking also at those barlines where a note overlaps

```

```

% into the next bar. 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
}
}

```

Discantus

Altus

Tenor

Bassus

IV-

IV-

IV-

IV-

Ju - bi - la - te De - o, om -

Ju - bi - la - te De - o, om -

- nis ter - ra, om- ... -us.

nis ter - ra, ... -us.

Ju - bi - la - te ... -us.

Ju - bi - ... -us.

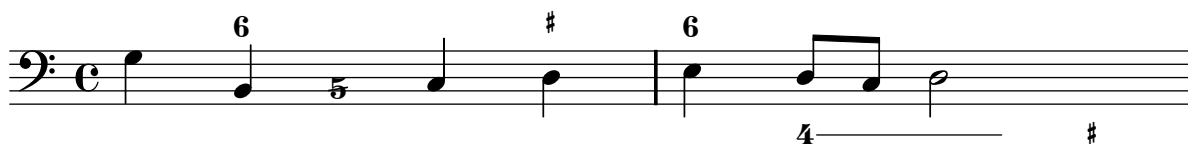
Contexts and engravers

‘adding-a-figured-bass-above-or-below-the-notes.ly’

When writing a figured bass, here’s a way to specify if you want your figures to be placed above or below the bass notes, by defining the `BassFigureAlignmentPositioning` `#'direction` property (exclusively in a `Staff` context). Choices are `#UP` (or `#1`), `#CENTER` (or `#0`) and `#DOWN` (or `#-1`).

As you can see here, this property can be changed as many times as you wish. Use `\once` `\override` if you don’t want the tweak to apply to the whole score.

```
bass = { \clef bass g4 b, c d e d8 c d2}
continuo = \figuremode {
  < _ >4 < 6 >8
  \once \override Staff.BassFigureAlignmentPositioning #'direction = #CENTER
  <5/> < _ >4
  \override Staff.BassFigureAlignmentPositioning #'direction = #UP
  < _+ > < 6 >
  \set Staff.useBassFigureExtenders = ##t
  \override Staff.BassFigureAlignmentPositioning #'direction = #DOWN
  < 4 >4. < 4 >8 < _+ >4
}
\score {
  << \new Staff = bassStaff \bass
  \context Staff = bassStaff \continuo >>
}
```

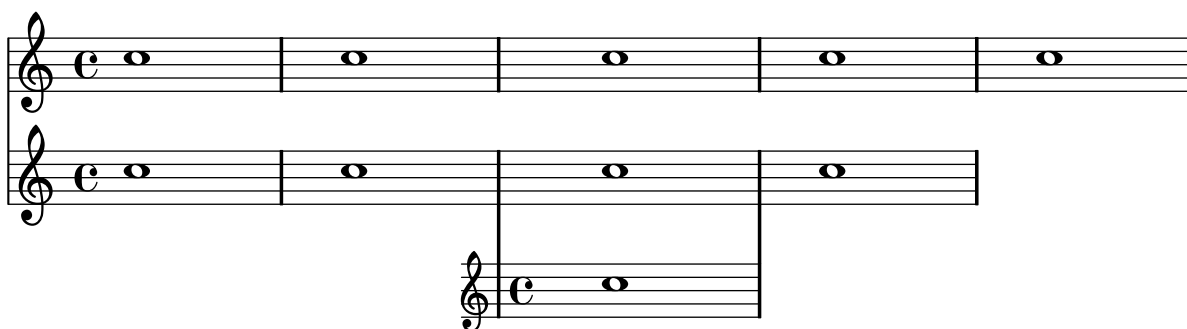


‘adding-an-extra-staff.ly’

You can add (possibly temporarily) an extra staff after the beginning of a piece.

```
\score {
  <<
    \new Staff \relative c''{ c1 c c c c }
    \new StaffGroup \relative c''{
      \new Staff
      c1 c
      << c1 \new Staff { c1 } >>
      c
    }
  >>
  \layout {
    \context{
      \Score
    }
  }
}
```

}

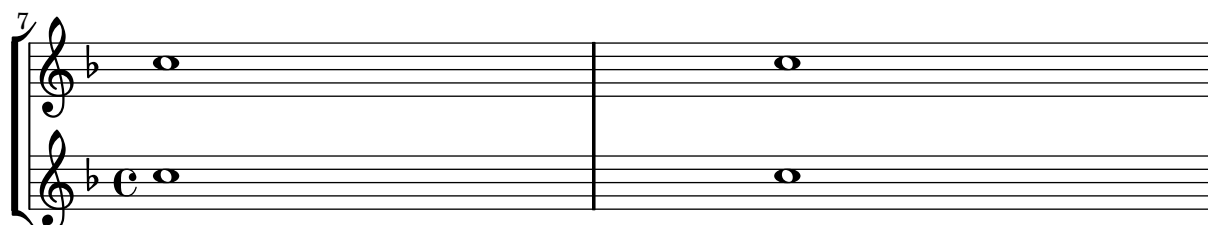
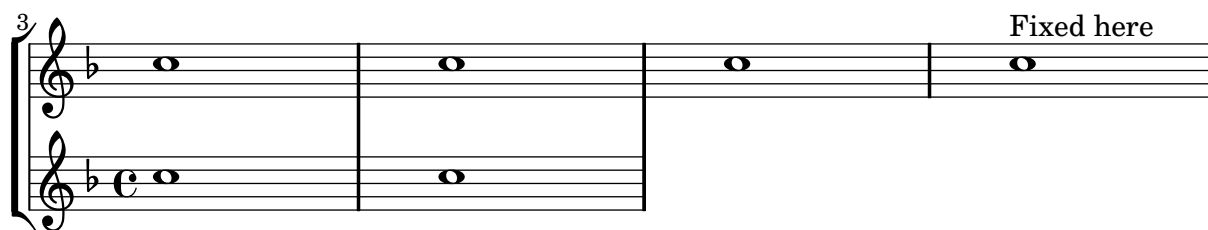


‘adding-and-extra-staff-at-a-line-break.ly’

When adding a new Staff at a line break, LilyPond will unfortunately add some extra space at the end of the line before the break (to fit in a key signature change, which will never be printed anyway). The workaround is to add a setting of `Staff.explicitKeySignatureVisibility` as is shown in the example. In version 2.10 and earlier, you have to do a similar setting for the time signatures, see 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
      c1 c
    }
    >>
    c1 c^"Fixed here" \break
    << { c1 c }
    \new Staff {
      \once \set Staff.explicitKeySignatureVisibility = #end-of-line-invisible
      %The next line is not needed in 2.11.x or later:
      \once \override Staff.TimeSignature #'break-visibility = #end-of-line-invisible
      \key f \major
      c1 c
    }
    >>
  }
}
```





`'changing-time-signatures-inside-a-polymetric-section-using--compressmusic.ly'`

The `measureLength` variable, together with `measurePosition`, determines when a barline is needed. However, when using `\compressMusic`, the scaling of durations makes it difficult to change time signatures without making a mess of it.

Therefore, `measureLength` has to be set manually, using the `ly:make-moment` callback. The second argument has to be the same as the second argument of `\compressMusic`.

```
\layout {
  \context { \Score
    \remove "Timing_translator"
    \remove "Default_bar_line_engraver"
  }
  \context {
    \Staff
    \consists "Timing_translator"
    \consists "Default_bar_line_engraver"
  }
}

<<
\new Staff {
  \compressMusic #'( 8 . 5 ) {
    \time 6/8
    \set Timing.measureLength = #(ly:make-moment 3 5)
    b8 b b b b b
    \time 2/4
    \set Timing.measureLength = #(ly:make-moment 2 5)
    b4 b
  }
}
\new Staff {
  \clef bass
  \time 2/4
  c2 d e f }
>>
```



‘chant-or-psalms-notation.ly’

This form of notation is used for the chant of the Psalms, where verses aren’t always the same length.

```
stemon = { \override Staff.Stem #'transparent = ##f }
stemoff = { \override Staff.Stem #'transparent = ##t }
```

```
\score {
\new Staff \with {\remove "Time_signature_engraver" }
{
    \key g \minor
    \set Score.timing = ##f
    \stemoff a'\breve bes'4 g'4
    \stemon a'2 \bar "||"
    \stemoff a'\breve g'4 a'4
    \stemon f'2 \bar "||"
    \stemoff a'\breve^{\markup { \italic flexe }}
    \stemon g'2 \bar "||"
}
\layout { raggedright = ##t}
}
```



‘creating-blank-staves.ly’

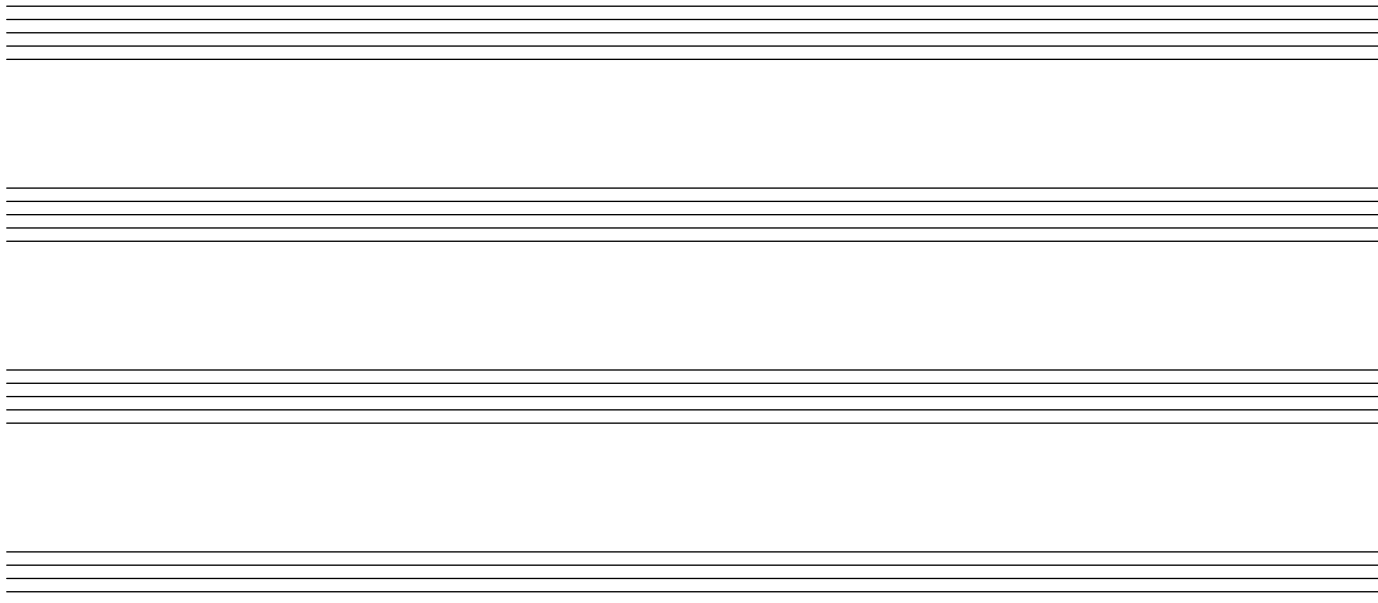
To create blank staves, you must generate empty measures, removing also from the Score context the Bar_number_engraver, and from the Staff context the Time_signature_engraver, the Clef_engraver and the Bar_engraver.

```
#{set-global-staff-size 20}
```

```
\score {
{
    \repeat unfold 12 { s1 \break }
}
\layout {
    indent = 0\in
    \context {
        \Staff
        \remove Time_signature_engraver
        \remove Clef_engraver
        \remove Bar_engraver
    }
    \context {
        \Score
    }
}
```

```
    \remove Bar_number_engraver
  }
}
```

```
\paper {
  #(set-paper-size "letter")
  raggedlastbottom = ##f
  linewidth = 7.5\in
  leftmargin = 0.5\in
  bottommargin = 0.25\in
  topmargin = 0.25\in
}
```



```
'engravers-one-by-one.ly'
```

The notation problem, creating a certain symbol, is handled by plugins. Each plugin is called an Engraver. In this example, 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, accidental, bar, etc. form a Staff context.

You may only see the first example in this document; please download this snippet and run it from your own computer.

```
%% sample music
topVoice = \relative c' {
  \key d\major
  es8([ g] a[ fis])
  b4
  b16[-. b-. b-. cis-.]
  d4->
}
```

```
botVoice = \relative c' {
  \key d\major
  c8([ f] b[ a])
  es4
  es16[-. es-. es-. fis-.]
  b4->
}
```

```
hoom = \relative c {
  \key d \major
```

```

\clef bass
g8-. r
r4
fis8-.
r8
r4
b'4->
}

pah = \relative c' {
  r8 b-.
  r4
  r8 g8-.
  r16 g-. r8
  \clef treble
  fis'4->
}

%
% setup for Request->Element conversion. Guru-only
%

MyStaff =\context {
  \type "Engraver_group"
  \name Staff

  \description "Handles clefs, bar lines, keys, accidentals. It can contain
@code{Voice} contexts."

  \consists "Output_property_engraver"

  \consists "Font_size_engraver"

  \consists "Volta_engraver"
  \consists "Separating_line_group_engraver"
  \consists "Dot_column_engraver"

  \consists "Ottava_spanner_engraver"
  \consists "Rest_collision_engraver"
  \consists "Piano_pedal_engraver"
  \consists "Piano_pedal_align_engraver"
  \consists "Instrument_name_engraver"
  \consists "Grob_pq_engraver"
  \consists "Forbid_line_break_engraver"
  \consists "Axis_group_engraver"

  \consists "Pitch_squash_engraver"

  \override VerticalAxisGroup #'minimum-Y-extent = #'(-6 . 6)
  extraVerticalExtent = ##f
  verticalExtent = ##f

```

```

localKeySignature = #'()

                                % explicitly set instrument, so we don't get
                                % weird effects when doing instrument names for
                                % piano staves

instrumentName = #'()
shortInstrumentName = #'()

\accepts "Voice"
}

MyVoice = \context {
  \type "Engraver_group"
  \name Voice

  \description "
    Corresponds to a voice on a staff. This context handles the
    conversion of dynamic signs, stems, beams, super- and subscripts,
    slurs, ties, and rests.

    You have to instantiate this explicitly if you want to have
    multiple voices on the same staff."

  localKeySignature = #'()
  \consists "Font_size_engraver"

                                % must come before all
  \consists "Output_property_engraver"
  \consists "Arpeggio_engraver"
  \consists "Multi_measure_rest_engraver"
  \consists "Text_spanner_engraver"
  \consists "Grob_pq_engraver"
  \consists "Note_head_line_engraver"
  \consists "Glissando_engraver"
  \consists "Ligature_bracket_engraver"
  \consists "Breathing_sign_engraver"
                                % \consists "Rest_engraver"
  \consists "Grace_beam_engraver"
  \consists "New_fingering_engraver"
  \consists "Chord_tremolo_engraver"
  \consists "Percent_repeat_engraver"
  \consists "Slash_repeat_engraver"

  %{
    Must come before text_engraver, but after note_column engraver.

  %}
  \consists "Text_engraver"
  \consists "Dynamic_engraver"
  \consists "Fingering_engraver"

```

```

\consists "Script_column_engraver"
\consists "Rhythmic_column_engraver"
\consists "Cluster_spanner_engraver"
\consists "Tie_engraver"
\consists "Tie_engraver"
\consists "Tuplet_engraver"
\consists "Note_heads_engraver"
\consists "Rest_engraver"

\consists "Skip_event_swallow_translator"
}

```

```

\score {
  \topVoice
  \layout {
    \context { \MyStaff }
    \context { \MyVoice }
  }
}

```

```

MyStaff = \context {
  \MyStaff
  \consists "Staff_symbol_engraver"
}

```

```

\score {
  \topVoice
  \layout {
    \context { \MyStaff }
    \context { \MyVoice }
  }
}

```

```

MyStaff = \context {
  \MyStaff
  \consists "Clef_engraver"
  \remove "Pitch_squash_engraver"
}

```

```

\score {
  \topVoice
  \layout {
    \context { \MyStaff }
    \context { \MyVoice }
  }
}

```

```

MyVoice = \context {
  \MyVoice
}

```

```

    \consists "Stem_engraver"
}

\score {
  \topVoice
  \layout {
    \context { \MyStaff }
    \context { \MyVoice }
  }
}

MyVoice = \context {
  \MyVoice
  \consists "Beam_engraver"
}

\score {
  \topVoice
  \layout {
    \context { \MyStaff }
    \context { \MyVoice }
  }
}

MyVoice= \context {
  \MyVoice
  \consists "Phrasing_slur_engraver"
  \consists "Slur_engraver"
  \consists "Script_engraver"
}

\score {
  \topVoice
  \layout {
    \context { \MyStaff }
    \context { \MyVoice }
  }
}

MyStaff = \context {
  \MyStaff
  \consists "Bar_engraver"
  \consists "Time_signature_engraver"
}

\score {
  \topVoice
  \layout {
    \context { \MyStaff }
    \context { \MyVoice }
  }
}

```



```
}  
  
MyStaff = \context {  
  \MyStaff  
  \consists "Accidental_engraver"  
  \consists "Key_engraver"  
}  
\score {  
  \topVoice  
  \layout {  
    \context { \MyStaff }  
    \context { \MyVoice }  
  }  
}
```





`'vocal-ensemble-template-with-lyrics-aligned-below-and-above-the-staves.ly'`

This template is basically the same as the simple "Vocal ensemble" template, with the exception that here all the lyrics lines are placed using `alignAboveContext` and `alignBelowContext`.

```
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
  >>

  \layout {
    \context {
      % a little smaller so lyrics
      % can be closer to the staff
      \Staff
      \override VerticalAxisGroup #'minimum-Y-extent = #'(-3 . 3)
    }
  }
}

\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 \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 \lyricsto basses \bassWords
>>

\layout {
  \context {
    % a little smaller so lyrics
    % can be closer to the staff
    \Staff
    \override VerticalAxisGroup #'minimum-Y-extent = #'(-3 . 3)
  }
}

```

hi hi hi hi

ha hu ha hu ha hu ha hu

ho ho ho ho

hi hi hi hi

ha hu ha hu ha hu ha hu

ho ho ho ho

Tweaks and overrides

`'alignment-vertical-spacing.ly'`

By setting properties in `NonMusicalPaperColumn`, vertical spacing of alignments can be adjusted per system.

By setting `alignment-extra-space` or `fixed-alignment-extra-space` an individual system may be stretched vertically.

For technical reasons, `\overrideProperty` has to be used for setting properties on individual objects. `\override` in a `\context` block may still be used for global overrides.

```

#(set-global-staff-size 13)

\relative c''
\new StaffGroup <<
  \new Staff {
    c1\break
    c\break
    c\break
  }
  \new Staff {
    c1 c c
  }
  \new PianoStaff <<
    \new Voice {
      \set PianoStaff.instrumentName = #"piano"
      \set PianoStaff.shortInstrumentName = #"pn"
      c1_"normal"

      \overrideProperty
        #"Score.NonMusicalPaperColumn"
        #'line-break-system-details
        #'((fixed-alignment-extra-space . 15))
      c_"fixed-alignment-extra-space"

      \overrideProperty
        #"Score.NonMusicalPaperColumn"
        #'line-break-system-details
        #'((alignment-extra-space . 15))
      c_"alignment-extra-space"
    }
  { c1 c c }
>>
>>

```

piano

normal

pn

fixed-alignment-extra-space

pn

alignment-extra-space

‘altering-the-number-of-stems-in-a-beam.ly’

You can alter the number of stems in a beam. In this example, two sets of four 32nds are joined, as if they were 8th notes.

```
\relative {
  \#(override-auto-beam-setting '(end * * * *) 1 4)
  f32 g a b b a g f

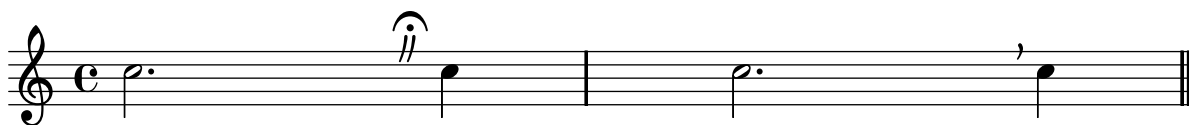
  f32 g a
  \set stemRightBeamCount = #1 b
  \set stemLeftBeamCount = #1 b
  a g f
}
```

‘caesura-railtracks-with-fermata.ly’

A caesura is sometimes denoted with a double "railtracks" breath mark with a fermata sign positioned over the top of the railtracks. This snippet should present an optically pleasing combination of railtracks and a fermata.

```
{
\context Voice {
  c''2.
  % use some scheme code to construct the symbol
  \override BreathingSign #'text = #(markup #:line
    (:musicglyph "scripts.caesura.curved"
    #:translate (cons -1.75 1.6)
    #:musicglyph "scripts.ufermata"
    ))

  \breathe c''4
  % set the breathe mark back to normal
  \revert BreathingSign #'text
  c''2. \breathe c''4
  \bar "|."
}
}
```



‘changing-an-individual-notes-size-in-a-chord.ly’

Individual noteheads 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 notehead).

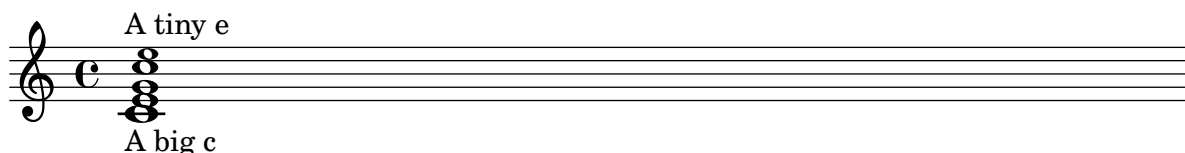
The code for the chord example shown:

```
\header{
  title = "Modify an individual notehead's size in a chord"
}

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

\score{
  \Notes
}
```

Modify an individual notehead's size in a chord



`'changing-properties-for-individual-grobs.ly'`

The `\applyOutput` command gives you the ability to tune any layout object, in any context. It requires a Scheme function with three arguments; advanced users can write it quite easily, whereas new users may want to use pre-defined functions such as this snippet, or the example in the manual.

```
% *****
% ly snippet:
% *****

\layout {
  ragged-right = ##t
}

#(define (mc-squared gr org cur)
  (let*
    (
      (ifs (ly:grob-interfaces gr))
      (sp (ly:grob-property gr 'staff-position))
    )
    (if (memq 'note-head-interface ifs)
      (begin
        (ly:grob-set-property! gr 'stencil ly:text-interface::print)
        (ly:grob-set-property! gr 'font-family 'roman)
        (ly:grob-set-property! gr 'text
          (make-raise-markup -0.5
            (case sp
              ((-5) (make-simple-markup "m"))
              ((-3) (make-simple-markup "c "))
              ((-2) (make-smaller-markup (make-bold-markup "2")))
              (else (make-simple-markup "bla")))
            ))))
      )))
  )))

\context Voice \relative c' {
  \stemUp
  \set autoBeaming = ##f

  { <d f g b>8

    \applyOutput #'Voice #mc-squared
    <d f g b>
  }
}

% *****
% end ly snippet
% *****
```




‘changing-the-default-text-font-family.ly’

The default font families for text can be overridden with `make-pango-font-tree`.

```
\paper {
  % change for other default global staff size.
  myStaffSize = #20
  %{
    run
    lilypond -dshow-available-fonts blabla
    to show all fonts available in the process log.
  %}

  #(define fonts
    (make-pango-font-tree "Times New Roman"
                          "Nimbus Sans"
                          "Luxi Mono"
                          "Helvetica"
                          "Courier"
      (/ myStaffSize 20)))
}

\relative {
  c'\markup {
    roman: foo \bold bla \italic bar \italic \bold baz
  }
  c'\markup {
    \override #'(font-family . sans)
    {
      sans: foo \bold bla \italic bar \italic \bold baz
    }
  }
  c'\markup {
    \override #'(font-family . typewriter)
    {
      mono: foo \bold bla \italic bar \italic \bold baz
    }
  }
}
```

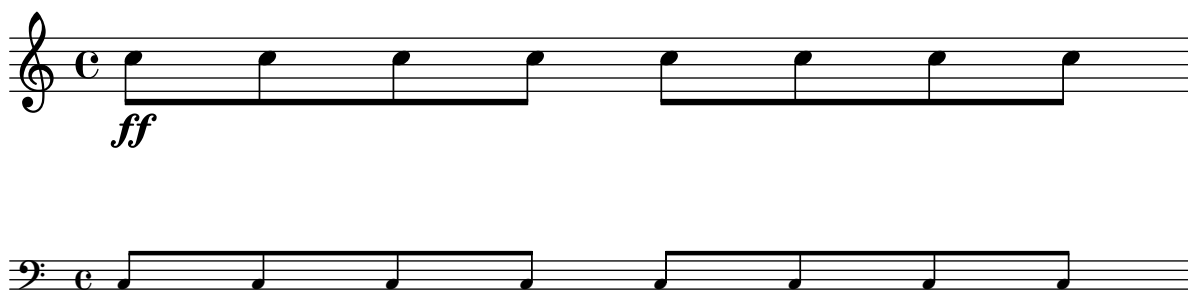


‘changing-the-staff-size.ly’

In order to change staff sizes, both `staff-space` and `fontSize` must be scaled.

```
{
  \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)
  }
  {
    \clef bass
    c8 c c c c c c c
  }
}
```



‘clefs-commonly-tweaked-properties.ly’

The command `\clef "treble_8"` is equivalent to setting `clefGlyph`, `clefPosition` (which controls the Y position of the clef), `middleCPosition` and `clefOctavation`. A clef is printed when any of these properties are changed.

Note that changing the glyph, the position of the clef, or the octavation, does not in itself change the position of subsequent notes on the staff: the position of middle C must also be specified to do this. The positional parameters are relative to the staff centre line, positive numbers displacing upwards, counting 1 for each line and space. The `clefOctavation` value would normally be set to 7, -7, 15 or -15, but other values are not invalid.

When a clef change takes place at a line break the new clef symbol is printed at both the end of the previous line and the beginning of the new line by default. If the warning clef at the end of the previous line is not required it can be suppressed by setting the `explicitClefVisibility` Staff property to the value `end-of-line-invisible`. The default behaviour can be recovered with `\unset Staff.explicitClefVisibility`.

The following examples show the possibilities when setting these properties manually. On the first line, the manual changes preserve the standard relative positioning of clefs and notes, whereas on the second line, they do not.

```
{
  % The default treble clef
  c'1
  % The standard bass clef
  \set Staff.clefGlyph = #"clefs.F"
  \set Staff.clefPosition = #2
  \set Staff.middleCPosition = #6
```

```

c'
% The baritone clef
\set Staff.clefGlyph = #"clefs.C"
\set Staff.clefPosition = #4
\set Staff.middleCPosition = #4
c'
% The standard choral tenor clef
\set Staff.clefGlyph = #"clefs.G"
\set Staff.clefPosition = #-2
\set Staff.clefOctavation = #-7
\set Staff.middleCPosition = #1
c'
% A non-standard clef
\set Staff.clefPosition = #0
\set Staff.clefOctavation = #0
\set Staff.middleCPosition = #-4
c' \break

% The following clef changes do not preserve
% the normal relationship between notes and clefs:

\set Staff.clefGlyph = #"clefs.F"
\set Staff.clefPosition = #2
c'
\set Staff.clefGlyph = #"clefs.G"
c'
\set Staff.clefGlyph = #"clefs.C"
c'
\set Staff.clefOctavation = #7
c'
\set Staff.clefOctavation = #0
\set Staff.clefPosition = #0
c'

% Here we go back to the normal clef:

\set Staff.middleCPosition = #4
c'
}

```



‘coloring-objects.ly’

LilyPond gives you the ability to assign different colors to any grob in your score, such as NoteHeads, Alterations, Beams and so on, by simply overriding the `#'color` property and choosing your color (over 200 colors are available, see the "List of Colors" Appendix in the Manual).

```
\relative {
  \override Accidental #'color = #darkgreen
  \override Beam #'color = #cyan
  \override NoteHead #'color = #darkyellow
  c4
  \override NoteHead #'color = #red
  f
  \override NoteHead #'color = #darkmagenta
  g
  \override NoteHead #'color = #darkblue
  b
  \override NoteHead #'color = #green
  \override Stem #'color = #blue
  e8 es d dis e4 r
}
```



`'controlling-tuplet-bracket-visibility.ly'`

Default behaviour 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, you can set the property `TupletBracket #'bracket-visibility` to either `##t` (always print a bracket), `##f` (never print a bracket) or `#'if-no-beam` (only print a bracket if there is no beam).

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

\new Voice \relative c'{
  << \mus s4^"default" >>
  \override TupletBracket #'bracket-visibility = #'if-no-beam
  << \mus s4^"'if-no-beam" >>
  \override TupletBracket #'bracket-visibility = ##t
  << \mus s4^"#t" >>
  \override TupletBracket #'bracket-visibility = ##f
  << \mus s4^"#f" >>
}
```





‘creating-text-spanners.ly’

The `\startTextSpan` and `\stopTextSpan` commands give you the ability to create text spanners as easily as pedals indications or octavations. Override some properties of the `TextSpanner` object to modify its output.

```
\relative c''{
  \override TextSpanner #'edge-text = #("bla" . "blu")
  a \startTextSpan
  b c
  a \stopTextSpan

  \override TextSpanner #'dash-period = #2
  \override TextSpanner #'dash-fraction = #0.0
  a \startTextSpan
  b c
  a \stopTextSpan

  \revert TextSpanner #'style
  \override TextSpanner #'style = #'dashed-line \override TextSpanner #'bound-details #'
\override TextSpanner #'bound-details #'right #'text = \markup { \draw-line #'(0 . -2) }

  a \startTextSpan
  b c
  a \stopTextSpan

  \set Staff.middleCPosition = #-13

  \override TextSpanner #'dash-period = #10
  \override TextSpanner #'dash-fraction = #.5
  \override TextSpanner #'thickness = #10
  a \startTextSpan
  b c
  a \stopTextSpan
  \set Staff.middleCPosition = #-6
}
```



‘custodes.ly’

Custodes may be engraved in various styles.

```
\layout {
  \context {
    \Staff
    \consists Custos_engraver
```

```

}
ragged-right = ##t
}

{
\override Staff.Custos #'neutral-position = #4

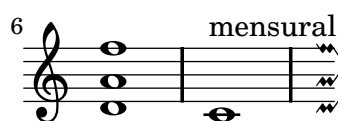
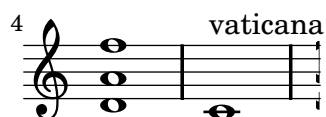
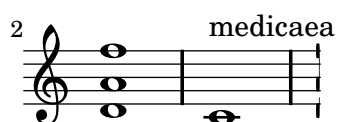
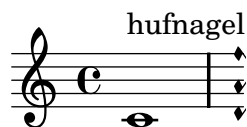
\override Staff.Custos #'style = #'hufnagel
c'1^"hufnagel"
\break < d' a' f''>1

\override Staff.Custos #'style = #'medicaea
c'1^"medicaea"
\break < d' a' f''>1

\override Staff.Custos #'style = #'vaticana
c'1^"vaticana"
\break < d' a' f''>1

\override Staff.Custos #'style = #'mensural
c'1^"mensural"
\break < d' a' f''>1
}

```



‘drawing-boxes-around-grobs.ly’

The `print`-function can be overridden to draw a box around an arbitrary grob.

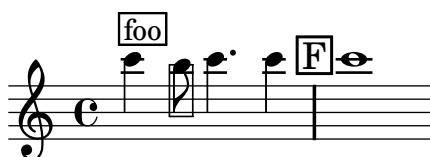
```
\relative c' {
  \override TextScript #'stencil =
  #(make-stencil-boxer 0.1 0.3 ly:text-interface::print)

  c'4^"foo"

  \override Stem #'stencil =
  #(make-stencil-boxer 0.05 0.25 ly:stem::print)

  \override Score.RehearsalMark #'stencil =
  #(make-stencil-boxer 0.15 0.3 ly:text-interface::print)
  b8
  \revert Stem #'stencil

  c4. c4 \mark "F" c1
}
```

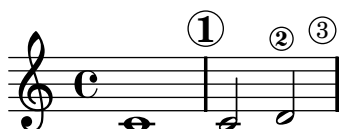


‘drawing-circles-around-various-objects.ly’

The `\circle` command allows you to draw circles around various objects, for example fingering indications. However, some objects require specific tweaks: rehearsal marks depend on the `Score.markFormatter` context, bar numbers on the `Score.BarNumber` context, and so on.

You can tweak the printing of your circles by setting some properties such as `#'thickness`, `#'circle-padding` or `#'font-size`.

```
\relative c'{
  c1
  \set Score.markFormatter
    = #(lambda (mark context)
          (make-circle-markup (format-mark-numbers mark context)))
  \mark \default
  c2 d^{\markup{\circle \finger "2"}}
  \override Score.BarNumber #'break-visibility = #all-visible
  \override Score.BarNumber #'stencil
    = #(make-stencil-circler 0.1 0.25 ly:text-interface::print)
}
```



‘making-an-object-invisible-with-the-transparent-property.ly’

Setting the `transparent` property will cause an object to be printed in ‘invisible ink’: the object is not printed, but all its other behavior is retained. The object still takes up space, it takes part in collisions, and slurs, and ties and beams can be attached to it.

The 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 c'' <<
{
  \once \override Stem #'transparent = ##t
  b8~ b8\noBeam
} \ {
  b[ g8]
}
>>
```

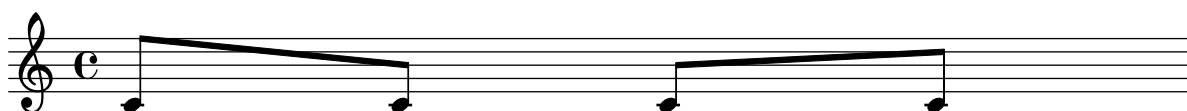


‘manually-controlling-beam-positions.ly’

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

```
\score {
  \context Voice \relative c {
    %% from upper staffline (position 4) to centre (position 0)
    \override Beam #'positions = #'(2 . 0)
    c'8[ c]

    %% from center to one above centre (position 2)
    \override Beam #'positions = #'(0 . 1)
    c[ c]
  }
}
```



‘move-specific-text.ly’

Objects of the same type, like text, can be moved around by using some Scheme code.

```
#(define (make-text-checker text)
  (lambda (grob) (equal? text (ly:grob-property grob 'text))))

\score {
  \relative c''' {
    \stemUp
```



```
\applyOutput #'Voice
#(outputproperty-compatibility
  (make-text-checker (make-simple-markup "m.d.))
  'extra-offset '(-3.5 . -4.5))
a^2^"m.d."
}
\layout { ragged-right = ##t}
}
```

`'proportional-strict-notespacing.ly'`

If `strict-note-spacing` is set spacing of notes is not influenced by bars or clefs part way along the system. Rather, they are put just before the note that occurs at the same time. This may cause collisions.

```
\paper {
  ragged-right = ##t
  indent = 0
}
\layout {
  \context {
    \Score
  }
}
```

```
\relative c' <<
\override Score.SpacingSpanner #'strict-note-spacing = ##t
\set Score.proportionalNotationDuration = #(ly:make-moment 1 16)
\new Staff {
  c8[ c \clef alto c c \grace { d16 } c8 c] c4 c2
  \grace { c16[ c16] }
  c2 }
\new Staff {
  c2 \times 2/3 { c8 \clef bass cis,, c }
  c4
  c1
}
>>
```

A musical score for the song "The Rose Tree". It consists of two staves, a treble staff and a bass staff, both in common time (C). The treble staff begins with a treble clef and a common time signature. The melody starts on a half note G4, followed by a quarter note A4, and then a series of eighth notes: B4, C5, D5, E5, F5, G5, A5, B5, C6, D6, E6, F6, G6, A6, B6, C7, D7, E7, F7, G7, A7, B7, C8, D8, E8, F8, G8, A8, B8, C9, D9, E9, F9, G9, A9, B9, C10, D10, E10, F10, G10, A10, B10, C11, D11, E11, F11, G11, A11, B11, C12, D12, E12, F12, G12, A12, B12, C13, D13, E13, F13, G13, A13, B13, C14, D14, E14, F14, G14, A14, B14, C15, D15, E15, F15, G15, A15, B15, C16, D16, E16, F16, G16, A16, B16, C17, D17, E17, F17, G17, A17, B17, C18, D18, E18, F18, G18, A18, B18, C19, D19, E19, F19, G19, A19, B19, C20, D20, E20, F20, G20, A20, B20, C21, D21, E21, F21, G21, A21, B21, C22, D22, E22, F22, G22, A22, B22, C23, D23, E23, F23, G23, A23, B23, C24, D24, E24, F24, G24, A24, B24, C25, D25, E25, F25, G25, A25, B25, C26, D26, E26, F26, G26, A26, B26, C27, D27, E27, F27, G27, A27, B27, C28, D28, E28, F28, G28, A28, B28, C29, D29, E29, F29, G29, A29, B29, C30, D30, E30, F30, G30, A30, B30, C31, D31, E31, F31, G31, A31, B31, C32, D32, E32, F32, G32, A32, B32, C33, D33, E33, F33, G33, A33, B33, C34, D34, E34, F34, G34, A34, B34, C35, D35, E35, F35, G35, A35, B35, C36, D36, E36, F36, G36, A36, B36, C37, D37, E37, F37, G37, A37, B37, C38, D38, E38, F38, G38, A38, B38, C39, D39, E39, F39, G39, A39, B39, C40, D40, E40, F40, G40, A40, B40, C41, D41, E41, F41, G41, A41, B41, C42, D42, E42, F42, G42, A42, B42, C43, D43, E43, F43, G43, A43, B43, C44, D44, E44, F44, G44, A44, B44, C45, D45, E45, F45, G45, A45, B45, C46, D46, E46, F46, G46, A46, B46, C47, D47, E47, F47, G47, A47, B47, C48, D48, E48, F48, G48, A48, B48, C49, D49, E49, F49, G49, A49, B49, C50, D50, E50, F50, G50, A50, B50, C51, D51, E51, F51, G51, A51, B51, C52, D52, E52, F52, G52, A52, B52, C53, D53, E53, F53, G53, A53, B53, C54, D54, E54, F54, G54, A54, B54, C55, D55, E55, F55, G55, A55, B55, C56, D56, E56, F56, G56, A56, B56, C57, D57, E57, F57, G57, A57, B57, C58, D58, E58, F58, G58, A58, B58, C59, D59, E59, F59, G59, A59, B59, C60, D60, E60, F60, G60, A60, B60, C61, D61, E61, F61, G61, A61, B61, C62, D62, E62, F62, G62, A62, B62, C63, D63, E63, F63, G63, A63, B63, C64, D64, E64, F64, G64, A64, B64, C65, D65, E65, F65, G65, A65, B65, C66, D66, E66, F66, G66, A66, B66, C67, D67, E67, F67, G67, A67, B67, C68, D68, E68, F68, G68, A68, B68, C69, D69, E69, F69, G69, A69, B69, C70, D70, E70, F70, G70, A70, B70, C71, D71, E71, F71, G71, A71, B71, C72, D72, E72, F72, G72, A72, B72, C73, D73, E73, F73, G73, A73, B73, C74, D74, E74, F74, G74, A74, B74, C75, D75, E75, F75, G75, A75, B75, C76, D76, E76, F76, G76, A76, B76, C77, D77, E77, F77, G77, A77, B77, C78, D78, E78, F78, G78, A78, B78, C79, D79, E79, F79, G79, A79, B79, C80, D80, E80, F80, G80, A80, B80, C81, D81, E81, F81, G81, A81, B81, C82, D82, E82, F82, G82, A82, B82, C83, D83, E83, F83, G83, A83, B83, C84, D84, E84, F84, G84, A84, B84, C85, D85, E85, F85, G85, A85, B85, C86, D86, E86, F86, G86, A86, B86, C87, D87, E87, F87, G87, A87, B87, C88, D88, E88, F88, G88, A88, B88, C89, D89, E89, F89, G89, A89, B89, C90, D90, E90, F90, G90, A90, B90, C91, D91, E91, F91, G91, A91, B91, C92, D92, E92, F92, G92, A92, B92, C93, D93, E93, F93, G93, A93, B93, C94, D94, E94, F94, G94, A94, B94, C95, D95, E95, F95, G95, A95, B95, C96, D96, E96, F96, G96, A96, B96, C97, D97, E97, F97, G97, A97, B97, C98, D98, E98, F98, G98, A98, B98, C99, D99, E99, F99, G99, A99, B99, C100, D100, E100, F100, G100, A100, B100, C101, D101, E101, F101, G101, A101, B101, C102, D102, E102, F102, G102, A102, B102, C103, D103, E103, F103, G103, A103, B103, C104, D104, E104, F104, G104, A104, B104, C105, D105, E105, F105, G105, A105, B105, C106, D106, E106, F106, G106, A106, B106, C107, D107, E107, F107, G107, A107, B107, C108, D108, E108, F108, G108, A108, B108, C109, D109, E109, F109, G109, A109, B109, C110, D110, E110, F110, G110, A110, B110, C111, D111, E111, F111, G111, A111, B111, C112, D112, E112, F112, G112, A112, B112, C113, D113, E113, F113, G113, A113, B113, C114, D114, E114, F114, G114, A114, B114, C115, D115, E115, F115, G115, A115, B115, C116, D116, E116, F116, G116, A116, B116, C117, D117, E117, F117, G117, A117, B117, C118, D118, E118, F118, G118, A118, B118, C119, D119, E119, F119, G119, A119, B119, C120, D120, E120, F120, G120, A120, B120, C121, D121, E121, F121, G121, A121, B121, C122, D122, E122, F122, G122, A122, B122, C123, D123, E123, F123, G123, A123, B123, C124, D124, E124, F124, G124, A124, B124, C125, D125, E125, F125, G125, A125, B125, C126, D126, E126, F126, G126, A126, B126, C127, D127, E127, F127, G127, A127, B127, C128, D128, E128, F128, G128, A128, B128, C129, D129, E129, F129, G129, A129, B129, C130, D130, E130, F130, G130, A130, B130, C131, D131, E131, F131, G131, A131, B131, C132, D132, E132, F132, G132, A132, B132, C133, D133, E133, F133, G133, A133, B133, C134, D134, E134, F134, G134, A134, B134, C135, D135, E135, F135, G135, A135, B135, C136, D136, E136, F136, G136, A136, B136, C137, D137, E137, F137, G137, A137, B137, C138, D138, E138, F138, G138, A138, B138, C139, D139, E

‘rest-styles.ly’

Rests may be used in various styles.

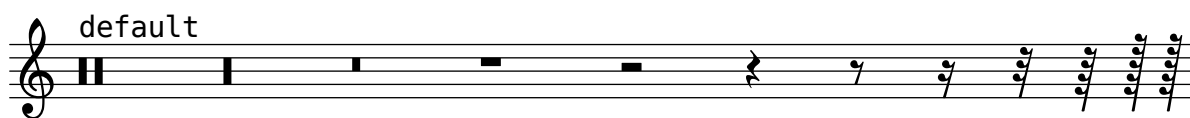
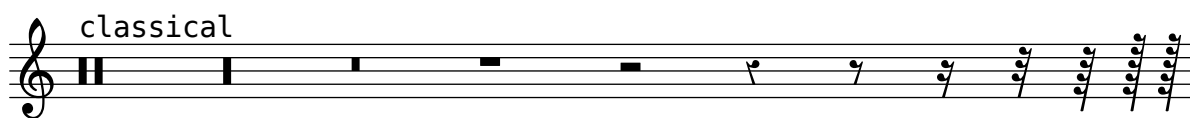
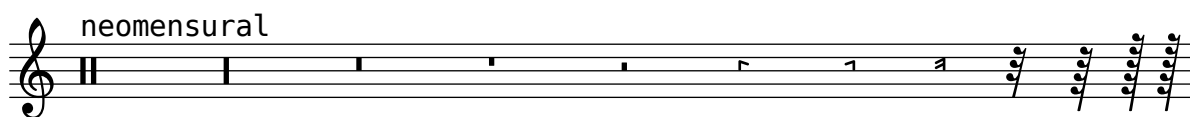
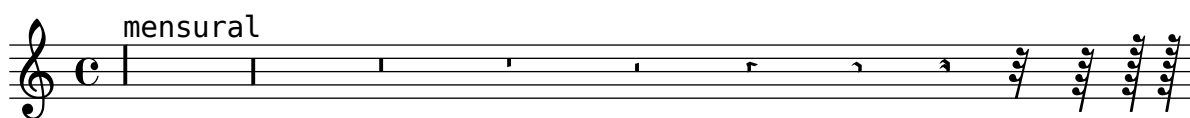
```
\layout {
  indent = 0.0
  raggedright = ##t
}

\context Staff \relative c {
  \set Score.timing = ##f
  \override Staff.Rest #'style = #'mensural
  r\maxima^\markup \typewriter { mensural }
  r\longa r\breve r1 r2 r4 r8 r16 r32 r64 r128 r128
  \bar ""

  \override Staff.Rest #'style = #'neomensural
  r\maxima^\markup \typewriter { neomensural }
  r\longa r\breve r1 r2 r4 r8 r16 r32 r64 r128 r128
  \bar ""

  \override Staff.Rest #'style = #'classical
  r\maxima^\markup \typewriter { classical }
  r\longa r\breve r1 r2 r4 r8 r16 r32 r64 r128 r128
  \bar ""

  \override Staff.Rest #'style = #'default
  r\maxima^\markup \typewriter { default }
  r\longa r\breve r1 r2 r4 r8 r16 r32 r64 r128 r128
}
```



‘rhythmic-slashes.ly’

In "simple" lead-sheets, sometimes no actual notes are written, instead only "rhythmic patterns" and chords above the measures are noted giving the structure of a song. Such a feature is for example useful while creating/transcribing the structure of a song and also when sharing lead sheets with guitarists or jazz musicians.

The standard support for this is described in section "Measure repeats", but then the first beat has to be an ordinary note or rest.

This example shows two solutions to this problem, by redefining ordinary rests to be printed as slashes. (If the duration of each beat is not a quarter note, replace the r4 in the definitions by a rest of the appropriate duration).

```
% Macro to print single slash
rs = {
  \once \override Rest #'stencil = #ly:percent-repeat-item-interface::beat-slash
  \once \override Rest #'thickness = #'0.48
  \once \override Rest #'slope = #'1.7
  r4
}

% Function to print a specified number of slashes
comp = #(define-music-function (parser location count) ( integer?)
  #{
    \override Rest #'stencil = #ly:percent-repeat-item-interface::beat-slash
    \override Rest #'thickness = #'0.48
    \override Rest #'slope = #'1.7
    \repeat unfold $count { r4 }
    \revert Rest #'stencil
  }
)

\score{
  \relative c'{
    c d e f | \rs \rs \rs \rs | \comp #4 |
  }
}
```



‘time-signature-in-parentheses.ly’

You may put the time signature in parentheses.

```
tsMarkup = \markup {
  \number {
    \bracket \column { "2" "4" }
  }
}

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

```
% FIXME: Gee, it doesn't work with 2.10 -vv

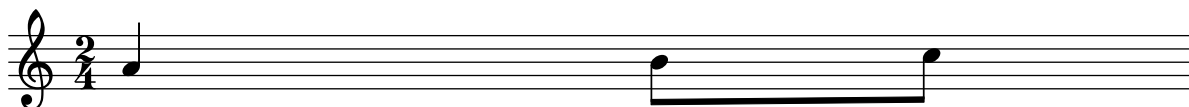
%{

  \override Staff.TimeSignature #'print-function = #Text_interface::print
  \override Staff.TimeSignature #'text = #tsMarkup

%}

\time 2/4

  a4 b8 c |
}
}
```



‘transcription-of-ancient-music-with-incipit.ly’

As a workaround to get real incipits which are independent from the main score these are included as a markup into the field normally used for the instrument name. As for now lyrics can only be added as a direct markup. It doesn’t unfortunately conform with the spacing of the main lyrics.

```
global = {
  \set Score.skipBars = ##t
  \key g \major
  \time 4/4

  %make the staff lines invisible on staves
  \override Staff.BarLine #'transparent = ##t
  \skip 1*8 % the actual music

  % let finis bar go through all staves
  \override Staff.BarLine #'transparent = ##f

  % finis bar
  \bar "|."
}
```

```
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 \override NoteHead #'transparent = ##t c'1 |
        b\breve |
    }
}

discantusLyrics = \lyricmode {
    Ju -- bi -- |
    la -- te De -- |
    o, om --
    nis ter -- |
    ra, __ om- |
    "... " |
    -us. |
}

altusNotes = {
    \transpose c' c' {
        \clef "treble"
        r2 g2. e4 fis g | % two bars
        a2 g4 e |
        fis g4.( fis16 e fis4) |
        g1 |
        \once \override NoteHead #'transparent = ##t g1 |
        g\breve |
    }
}

altusLyrics = \lyricmode {
    Ju -- bi -- la -- te | % two bars
    De -- o, om -- |
    nis ter -- ra, |
    "... " |
    -us. |
}

tenorNotes = {
    \transpose c' c' {
        \clef "treble_8"
        R1 |
        R1 |
        R1 |
        r2 d'2. d'4 b e' | % two bars
        \once \override NoteHead #'transparent = ##t e'1 |
        d'\breve |
    }
}

tenorLyrics = \lyricmode {
    Ju -- bi -- la -- te | % two bars
    "... " |
    -us.
}

```

```

bassusNotes = {
  \transpose c' c' {
    \clef "bass"
    R1 |
    R1 |
    R1 |
    R1 |
    g2. e4 |
    \once \override NoteHead #'transparent = ##t e1 |
    g\breve |
  }
}

bassusLyrics = \lyricmode {
  Ju -- bi- |
  "... " |
  -us.
}

incipitDiscantus = \markup{
  \score{
    {
      \set Staff.instrumentName="Discantus "
      \override NoteHead #'style = #'neomensural
      \override Rest #'style = #'neomensural
      \override Staff.TimeSignature #'style = #'neomensural
      \cadenzaOn
      \clef "neomensural-c1"
      \key f \major
      \time 2/2
      c''1._"IV-" s2 %two bars
      \skip 1*8 % eight bars
    }
  }
  \layout {
    \context {\Voice
      \remove Ligature_bracket_engraver
      \consists Mensural_ligature_engraver
    }
    line-width=4.5\cm
  }
}

incipitAltus = \markup{
  \score{
    {
      \set Staff.instrumentName="Altus "
      \override NoteHead #'style = #'neomensural
      \override Rest #'style = #'neomensural
      \override Staff.TimeSignature #'style = #'neomensural
      \cadenzaOn
    }
  }
}

```

```

        \clef "neomensural-c3"
        \key f \major
        \time 2/2
        r1          % one bar
f'1._"IV-" s2      % two bars
\skip 1*7 % seven bars
    }
\layout {
    \context {\Voice
        \remove Ligature_bracket_engraver
        \consists Mensural_ligature_engraver
    }
    line-width=4.5\cm
}
}

incipitTenor = \markup{
    \score{ {
        \set Staff.instrumentName = "Tenor  "
        \override NoteHead #'style = #'neomensural
        \override Rest #'style = #'neomensural
        \override Staff.TimeSignature #'style = #'neomensural
        \cadenzaOn
        \clef "neomensural-c4"
        \key f \major
        \time 2/2
r\longa % four bars
r\breve % two bars
r1      % one bar
c'1._"IV-" s2 % two bars
\skip 1 % one bar
    }
    \layout {
        \context {\Voice
            \remove Ligature_bracket_engraver
            \consists Mensural_ligature_engraver
        }
        line-width=4.5\cm
    }
}
}

incipitBassus = \markup{
    \score{ {
        \set Staff.instrumentName = "Bassus  "
        \override NoteHead #'style = #'neomensural
        \override Rest #'style = #'neomensural
        \override Staff.TimeSignature #'style = #'neomensural
        \cadenzaOn
        \clef "bass"
        \key f \major

```

```

        \time 2/2
% incipit
r\maxima % eight bars
f1._"IV-" s2 % two bars
}
\layout {
    \context {\Voice
        \remove Ligature_bracket_engraver
        \consists Mensural_ligature_engraver
    }
    line-width=4.5\cm
}
}

%StaffGroup is used instead of ChoirStaff to get bar lines between systems
\score {
  <<
  \new StaffGroup = choirStaff <<
    \new Voice =
      "discantusNotes" << \global
      \set Staff.instrumentName=\incipitDiscantus
      \discantusNotes >>
    \new Lyrics =
      "discantusLyrics" \lyricsto discantusNotes { \discantusLyrics }

    \new Voice =
      "altusNotes" << \global
      \set Staff.instrumentName=\incipitAltus
      \altusNotes >>
    \new Lyrics =
      "altusLyrics" \lyricsto altusNotes { \altusLyrics }

    \new Voice =
      "tenorNotes" << \global
      \set Staff.instrumentName=\incipitTenor
      \tenorNotes >>
    \new Lyrics =
      "tenorLyrics" \lyricsto tenorNotes { \tenorLyrics }

    \new Voice =
      "bassusNotes" << \global
      \set Staff.instrumentName=\incipitBassus
      \bassusNotes >>
    >>
    \new Lyrics =
      "bassusLyrics" \lyricsto bassusNotes { \bassusLyrics }
    %Keep the bass lyrics outside of the staff group to avoid bar lines
    %between the lyrics.
  >>

  \layout {

```



```

\context {
  \Score

  % no bars in staves
  \override BarLine #'transparent = ##t
}
% the next three instructions keep the lyrics between the barlines
  \context { \Lyrics
    \consists "Bar_engraver"
    \override BarLine #'transparent = ##t }
  \context { \StaffGroup \consists "Separating_line_group_engraver" }
\context {
  \Voice

  % no slurs
  \override Slur #'transparent = ##t

  % Comment in the below "\remove" command to allow line
  % breaking also at those barlines where a note overlaps
  % into the next bar. 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
}
}

```

Discantus

Altus

Tenor

Bassus

IV-

IV-

IV-

IV-

4

- nis ter - ra, om- ... -us.

nis ter - - ra, ... -us.

8 Ju - bi - la - te ... -us.

Ju - bi- ... -us.

‘using-the--tweak-command-to-tweak-individual-grobs.ly’

With the `weak` command, you can tune every grob directly. Here are some examples of available tweaks.

```
{
  \set fingeringOrientations = #'(right)
  <
    \tweak #'font-size #3 c
    \tweak #'color #red d-\tweak #'font-size #8 -4
    \tweak #'style #'cross g
    \tweak #'duration-log #1 a
  >4
}
```

‘vertically-aligning-ossias-and-lyrics.ly’

This snippet shows of to use the `alignBelowContext` and `alignAboveContext` properties, which may be needed for text elements (e.g. lyrics) positioning, but also for musical contents such as ossias.

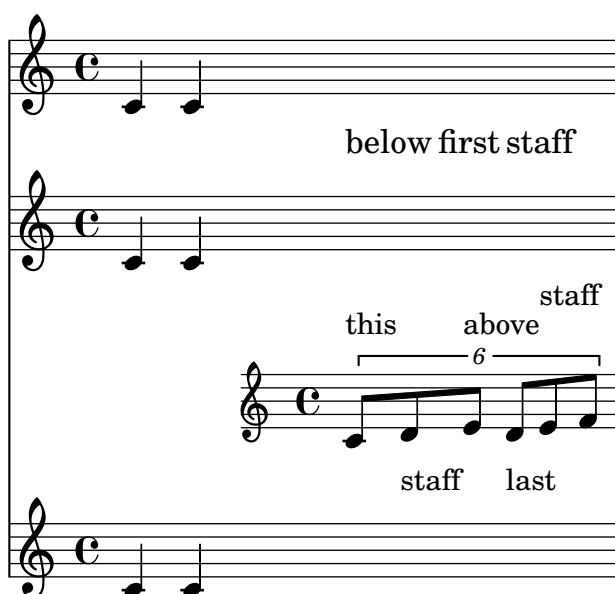
```
\paper {
  ragged-right = ##t
}

\relative <<
  \new Staff = "1" { c4 c s2 }
  \new Staff = "2" { c4 c s2 }
  \new Staff = "3" { c4 c s2 }
  { \skip 2
    <<
      \lyrics {
```

```

\set alignBelowContext = #"1"
below8 first staff
}
\new Staff {
  \set Staff.alignAboveContext = #"3"
  \times 4/6 {
    \override TextScript #'padding = #3
    c8^"this" d_"staff" e^"above" d_"last" e_"staff" f
  }
}
>> }
>>

```



‘vertically-centered-dynamics-and-textscripts.ly’

By setting the Y-extent property to a fixed value (here -1.5 . 1.5), we force LilyPond to align every element of the DynamicLineSpanner (text elements and dynamics) to a common reference point, regardless to the actual extent of these objects. This way, every element will be vertically centered, for a nicer output (you can compare the first and the second line in this example; the trick is only applied on the second line).

The same idea is used to align the text scripts along their baseline.

```

\paper { indent = 0 line-width = 5\in }

music = \relative c''
{
  c2\p^{\markup { "gorgeous" }} c\f^{\markup { "fantastic" }}
  c4\p c \f \> c c \! \p
}

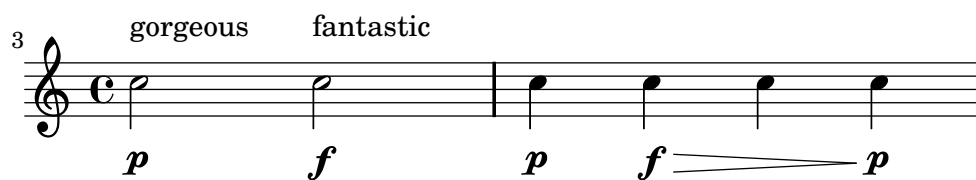
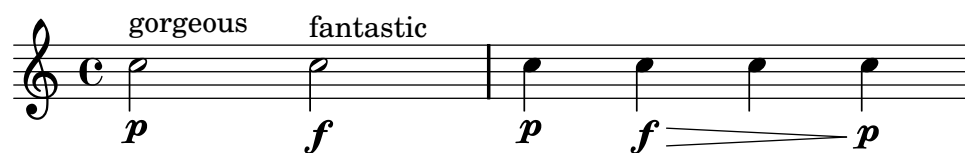
\score
{
  {
    \music \break

```

```

\override DynamicLineSpanner #'staff-padding = #2.0
\override DynamicLineSpanner #'Y-extent = #'(-1.5 . 1.5)
\override TextScript #'Y-extent = #'(-1.5 . 1.5)
\music
}
}

```



Paper and layout

‘aligning-and-centering-instrument-names.ly’

Instrument names are generally printed at the left side of the staves. To align the names of several different instruments, you can put them in a `\markup` block and use one of the following possibilities:

- * Right-aligned instrument names: this is LilyPond’s default behavior
- * Center-aligned instrument names: with the `\hcenter-in #n` syntax, you can place the instrument names inside a padded box (n being the width of the box)
- * Left-aligned instrument names: you have to print the names on top of an empty box, using the `\combine` command with a `\hspace #n` object.

```
\paper{
  indent = #0
  left-margin = #30
  line-width = #160
}

\new StaffGroup \relative
<<
  \new Staff {
    \set Staff.instrumentName = "blabla"
    c1^"default" c1 \break
    \set Staff.instrumentName = \markup { \hcenter-in #10 "blabla" }
    c1^"centered" c1 \break
    \set Staff.instrumentName = \markup { \combine \hspace #8 "blabla" }
    c1^"left-aligned" c1

  }
  \new Staff {
    \set Staff.instrumentName = "blo"
    c1 c1 \break
    \set Staff.instrumentName = \markup { \hcenter-in #10 "blo" }
    c1 c1 \break
    \set Staff.instrumentName = \markup { \combine \hspace #8 "blo" }
    c1 c1
  }
}>>
```

The image shows a musical score snippet with two staves. The top staff is labeled 'blabla' and the bottom staff is labeled 'blo'. Both staves have a treble clef and a common time signature 'C'. The top staff has a note on the first line (G4) with the label 'default' above it. The bottom staff has a note on the first line (G4) with the label 'blo' above it. Both staves have a double bar line after the first measure.

The image shows two musical score examples. The top example is labeled 'centered' and the bottom example is labeled 'left-aligned'. Both examples show two staves, 'blabla' and 'blo', with a brace on the left. The staves are divided into two systems by a vertical line. The notes are represented by stylized 'e' symbols. In the 'centered' example, the notes are centered within the staff lines. In the 'left-aligned' example, the notes are aligned to the left of the staff lines.

‘alignment-vertical-spacing.ly’

By setting properties in `NonMusicalPaperColumn`, vertical spacing of alignments can be adjusted per system.

By setting `alignment-extra-space` or `fixed-alignment-extra-space` an individual system may be stretched vertically.

For technical reasons, `\overrideProperty` has to be used for setting properties on individual objects. `\override` in a `\context` block may still be used for global overrides.

`#(set-global-staff-size 13)`

```
\relative c''
\new StaffGroup <<
  \new Staff {
    c1\break
    c\break
    c\break
  }
  \new Staff {
    c1 c c
  }
\new PianoStaff <<
  \new Voice {
    \set PianoStaff.instrumentName = #"piano"
    \set PianoStaff.shortInstrumentName = #"pn"
    c1_"normal"

    \overrideProperty
      #"Score.NonMusicalPaperColumn"
      #'line-break-system-details
      #'((fixed-alignment-extra-space . 15))
    c_"fixed-alignment-extra-space"

    \overrideProperty
      #"Score.NonMusicalPaperColumn"
      #'line-break-system-details
      #'((alignment-extra-space . 15))
    c_"alignment-extra-space"
  }
}
```

```

    { c1 c c }
  >>
>>

```

piano

normal

pn

fixed-alignment-extra-space

pn

aligment-extra-space

‘changing-the-staff-size.ly’

In order to change staff sizes, both `staff-space` and `fontSize` must be scaled.

```

{
  \new Staff \relative c'' { \dynamicDown c8 \ff c c c c c c c }
}

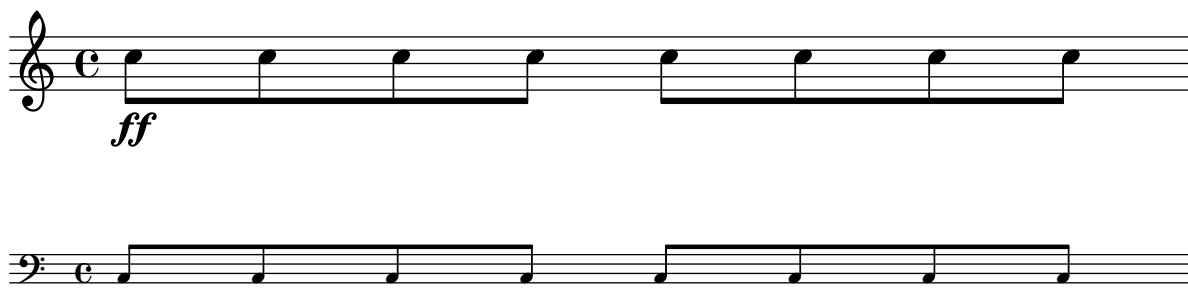
{
  \new Staff \with {
    fontSize = #-3

```

```

\override StaffSymbol #'staff-space = #(magstep -3)
}
{
  \clef bass
  c8 c c c c c c c c
}
}

```



‘clip-systems.ly’

This code shows how to clip (extracts) snippets from a full score.

This file needs to be run separately with `-dclip-systems`; the snippets page may not adequately show the results.

The result will be files named ‘*base-from-start-to-end*[-count].eps’.

- If system starts and ends are included, they include extents of the System grob, eg. 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.

```

#(ly:set-option 'clip-systems)

```

```

#(set! output-count 1)

```

```

origScore = \score{
  \relative {
    \set Staff.instrumentName = #"bla"
    c1
    d
    \grace c16
    e1
    \key d\major

    f
    \break \clef bass
    g,
    fis
  }
}

```

```

\book {
  \score {

```



```

\origScore
\layout {

  %% each clip-region is a (START . END) pair
  %% where both are rhythmic-locations.

  %% (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 4 0 1))

    (cons
      (make-rhythmic-location 0 0 1)
      (make-rhythmic-location 6 0 1))
  )
}
}
}

#(set! output-count 0)
#(ly:set-option 'clip-systems #f)

\book {
  \score { \origScore }
  \markup { \bold \fontsize #6 clips }
  \score {
    \lyrics {
      \markup { 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
    }
  }
}

```



‘creating-blank-staves.ly’

To create blank staves, you must generate empty measures, removing also from the `Score` context the `Bar_number_engraver`, and from the `Staff` context the `Time_signature_engraver`, the `Clef_engraver` and the `Bar_engraver`.

```

#(set-global-staff-size 20)

```

```

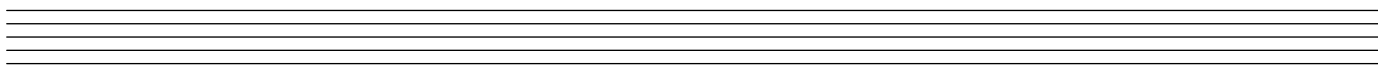
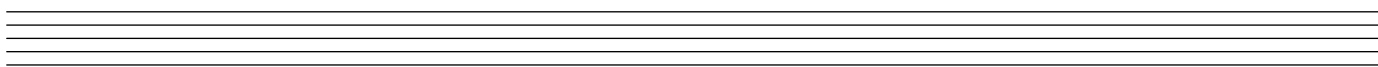
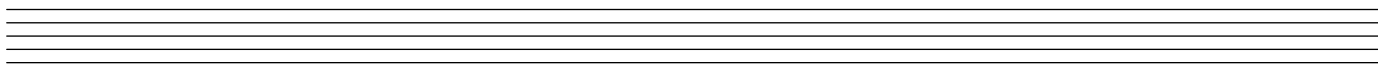
\score {
  {
    \repeat unfold 12 { s1 \break }
  }
  \layout {
    indent = 0\in
    \context {
      \Staff
      \remove Time_signature_engraver
      \remove Clef_engraver
      \remove Bar_engraver
    }
    \context {
      \Score
      \remove Bar_number_engraver
    }
  }
}

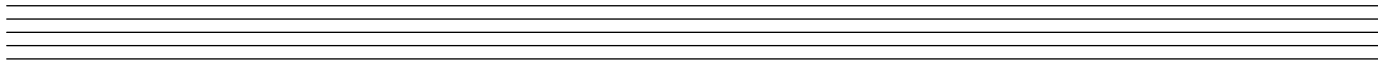
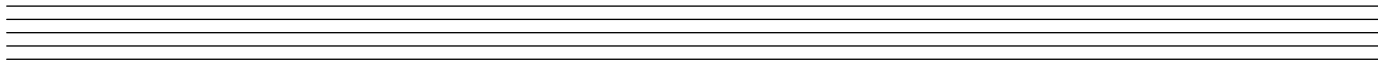
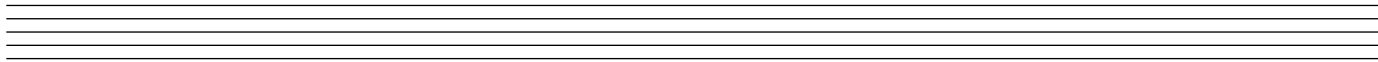
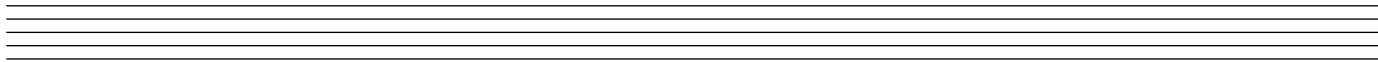
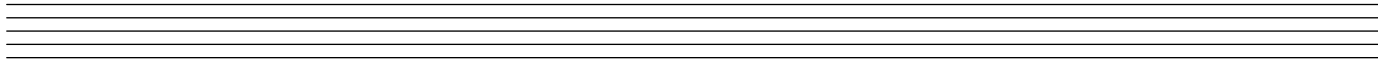
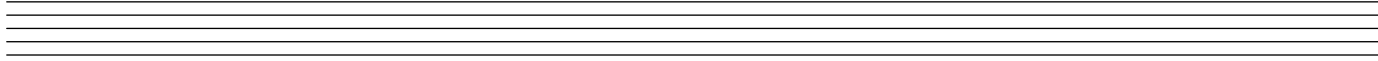
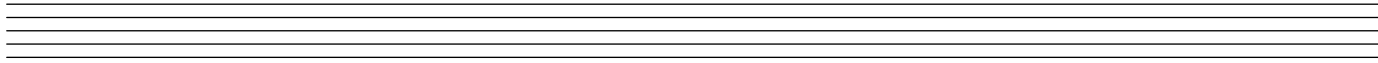
```

```

\paper {
  #(set-paper-size "letter")
  raggedlastbottom = ##f
  linewidth = 7.5\in
  leftmargin = 0.5\in
  bottommargin = 0.25\in
  topmargin = 0.25\in
}

```





`‘demonstrating-all-headers.ly’`

All header fields with special meanings.

title

subtitle

instrument

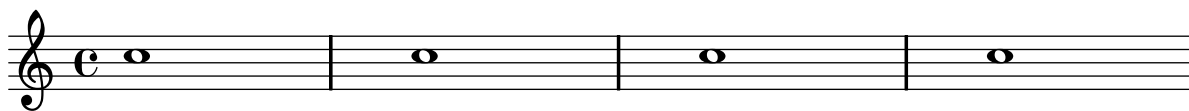
poet

composer

arranger

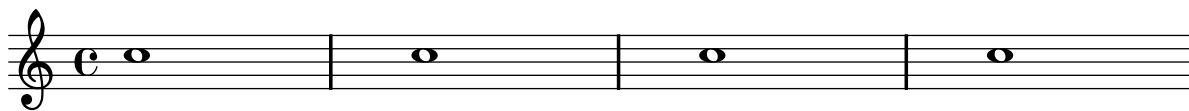
piece

opus



localpiece

localopus



‘table-of-contents.ly’

A table of contents is included using `\markuplines \table-of-contents`. The TOC items are added with the `\tocItem` command.

```

#(set-default-paper-size "a6")

```

```

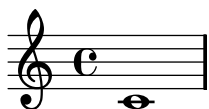
\book {
  \markuplines \table-of-contents
  \pageBreak

  \tocItem \markup "The first score"
  \score {
    {
      c'1 \pageBreak
      \mark "A" \tocItem \markup "Mark A"
      d'
    }
  }
  \pageBreak
  \tocItem \markup "The second score"
  \score {
    { e' }
    \header { piece = "Second score" }
  }
}

```

Table of Contents

The first score	2
Mark A	3
The second score	4



3



4

Second score



Titles

`‘adding-the-current-date-to-a-score.ly’`

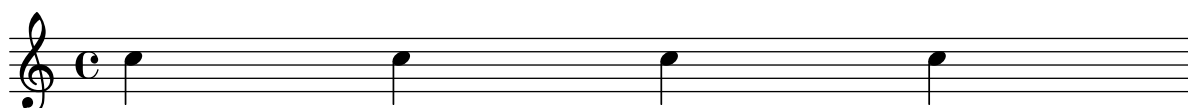
I often find it useful to include a date on printed music, so that I can see if I’m using the latest version, or tell someone else that he should only use the version after a certain date. A simple solution is to enter the date manually to the `.ly` file. But that’s very error prone. It’s easy to forget updating the date. So I thought it would be useful if you can add the date on which the PDF file is generated automatically. I didn’t figure it out myself, but I asked on lilypond-user mailing list. And guess what? Someone came with an excellent solution! So thank you very much Toine Schreurs for sending this solution to the user mailing list. I post it here for future reference.

The solution is to use two scheme functions called `strftime` and `localtime`, as shown in the snippet. It is a very flexible solution, you can format the date just as you like it by adapting the `"%d-%m-%Y"` string. See the Guile documentation for more details on this format string: [Formatting Calendar Time](#).

```
\score {
  \relative c'' {
    c4 c c c
  }
}
% and use it in a \markup block:
\markup {
  \date
}
```

Including the date!

17-02-2008



17-02-2008

`‘aligning-and-centering-instrument-names.ly’`

Instrument names are generally printed at the left side of the staves. To align the names of several different instruments, you can put them in a `\markup` block and use one of the following possibilities:

- * Right-aligned instrument names: this is LilyPond’s default behavior
- * Center-aligned instrument names: with the `\hcenter-in #n` syntax, you can place the instrument names inside a padded box (`n` being the width of the box)
- * Left-aligned instrument names: you have to print the names on top of an empty box, using the `\combine` command with a `\hspace #n` object.

```
\paper{
  indent = #0
  left-margin = #30
  line-width = #160
}
```

```

}

\new StaffGroup \relative
<<
  \new Staff {
    \set Staff.instrumentName = "blabla"
    c1^"default" c1 \break
    \set Staff.instrumentName = \markup { \hcenter-in #10 "blabla" }
    c1^"centered" c1 \break
    \set Staff.instrumentName = \markup {\combine \hspace #8 "blabla" }
    c1^"left-aligned" c1

  }
  \new Staff {
    \set Staff.instrumentName = "blo"
    c1 c1 \break
    \set Staff.instrumentName = \markup { \hcenter-in #10 "blo" }
    c1 c1 \break
    \set Staff.instrumentName = \markup {\combine \hspace #8 "blo" }
    c1 c1

  }
>>

```

‘demonstrating-all-headers.ly’

All header fields with special meanings.

title**subtitle**

poet

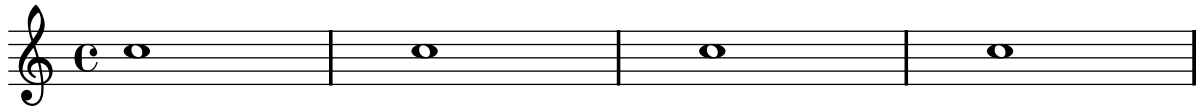
instrument

composer

arranger

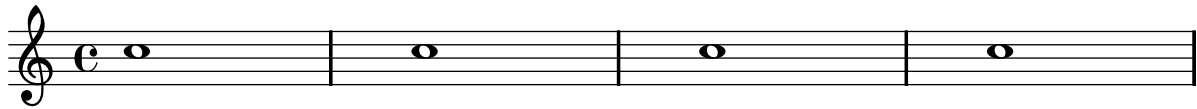
piece

opus



localpiece

localopus



Spacing

‘adjusting-lyrics-vertical-spacing.ly’

This snippets shows you how to bring the lyrics line closer to the Staff.

% Default layout:

```
\score{
  <<
    \new Staff \new Voice = m \relative c'{ c4 d e f g f e d c1}
    \new Lyrics \lyricsto m {aa aa aa aa aa aa aa aa }
  >>
}
```

% Reducing the minimum space below the Staff and above the Lyrics:

```
\score {
  <<
    \new Staff \with {
      \override VerticalAxisGroup #'minimum-Y-extent = #'(-1.4)}
    \new Voice = m \relative c'{ c4 d e f g f e d c1 }
    \new Lyrics \with {
      \override VerticalAxisGroup #'minimum-Y-extent = #'(-1.2 . 1)}
    \lyricsto m {aa aa aa aa aa aa aa aa }
  >>

  \header {
    tagline = ""
  }
}
```



‘alignment-vertical-spacing.ly’

By setting properties in `NonMusicalPaperColumn`, vertical spacing of alignments can be adjusted per system.

By setting `alignment-extra-space` or `fixed-alignment-extra-space` an individual system may be stretched vertically.

For technical reasons, `\overrideProperty` has to be used for setting properties on individual objects. `\override` in a `\context` block may still be used for global overrides.

```
\set-global-staff-size 13)
```

```

\relative c''
\new StaffGroup <<
  \new Staff {
    c1\break
    c\break
    c\break
  }
  \new Staff {
    c1 c c
  }
\new PianoStaff <<
  \new Voice {
    \set PianoStaff.instrumentName = #"piano"
    \set PianoStaff.shortInstrumentName = #"pn"
    c1_"normal"

    \overrideProperty
      #"Score.NonMusicalPaperColumn"
      #'line-break-system-details
      #'((fixed-alignment-extra-space . 15))
    c_"fixed-alignment-extra-space"

    \overrideProperty
      #"Score.NonMusicalPaperColumn"
      #'line-break-system-details
      #'((alignment-extra-space . 15))
    c_"alignment-extra-space"
  }
  { c1 c c }
>>
>>

```

piano

normal

‘page-label.ly’

Page labels may be placed inside music or at top-level, and referred to in markups.

```

#(set-default-paper-size "a6")

```

```

#(define-markup-command (toc-line layout props label text) (symbol? markup?)
  (interpret-markup layout props
    (markup #:fill-line (text #:page-ref label "8" "?")))))

```

```

\book {
  \markup \huge \fill-line { \null "Title Page" \null }

  \pageBreak

  \label #'toc
  \markup \column {
    \large \fill-line { \null "Table of contents" \null }
    \toc-line #'toc "Table of contents"
    \toc-line #'firstScore "First Score"
    \toc-line #'markA "Mark A"
    \toc-line #'markB "Mark B"
    \toc-line #'markC "Mark C"
    \toc-line #'unknown "Unknown label"
  }

  \pageBreak

  \label #'firstScore
  \score {

```

```

{ c'2 c'
  \mark \markup { A (page \concat { \page-ref #'markA "0" "?" ) }} \label #'markA
  c' c'
  \pageBreak
  \mark "B" \label #'markB
  d' d'
  d' d'
  \once \override Score . RehearsalMark #'break-visibility = #begin-of-line-invisible
  \mark "C" \label #'markC
}
\header { piece = "First score" }
}

```

Title Page

2

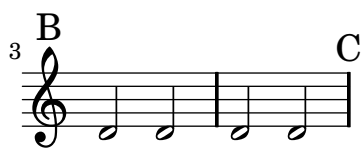
Table of contents

Table of contents	2
First Score	3
Mark A	3
Mark B	4
Mark C	4
Unknown label	?

First score



4



Music engraving by LilyPond 2.11.39—www.lilypond.org

‘proportional-strict-notespacing.ly’

If `strict-note-spacing` is set spacing of notes is not influenced by bars or clefs part way along the system. Rather, they are put just before the note that occurs at the same time. This may cause collisions.

```
\paper {
  ragged-right = ##t
  indent = 0
}
\layout {
  \context {
    \Score
  }
}

\relative c' { <<
  \override Score.SpacingSpanner #'strict-note-spacing = ##t
  \set Score.proportionalNotationDuration = #(ly:make-moment 1 16)
  \new Staff {
    c8[ c \clef alto c c \grace { d16 } c8 c] c4 c2
    \grace { c16[ c16] }
    c2 }
  \new Staff {
    c2 \times 2/3 { c8 \clef bass cis,, c }
    c4
    c1
```

```
}
>>
```

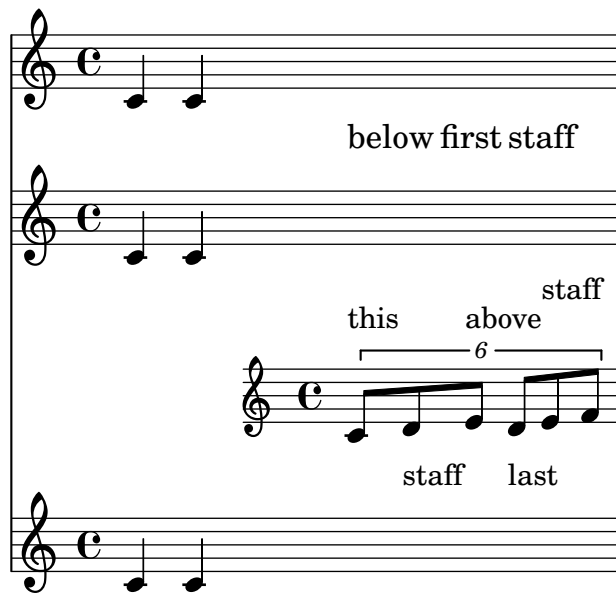


‘vertically-aligning-ossias-and-lyrics.ly’

This snippet shows of to use the `alignBelowContext` and `alignAboveContext` properties, which may be needed for text elements (e.g. lyrics) positioning, but also for musical contents such as ossias.

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

\relative <<
  \new Staff = "1" { c4 c s2 }
  \new Staff = "2" { c4 c s2 }
  \new Staff = "3" { c4 c s2 }
  { \skip 2
    <<
      \lyrics {
        \set alignBelowContext = #"1"
        below8 first staff
      }
      \new Staff {
        \set Staff.alignAboveContext = #"3"
        \times 4/6 {
          \override TextScript #'padding = #3
          c8^"this" d_"staff" e^"above" d_"last" e_"staff" f
        }
      }
    }
  }
  >> }
>>
```

`'vertically-centered-dynamics-and-textscripts.ly'`

By setting the Y-extent property to a fixed value (here $-1.5 \dots 1.5$), we force LilyPond to align every elements of the `DynamicLineSpanner` (text elements and dynamics) to a common reference point, regardless to the actual extent of these objects. This way, every element will be vertically centered, for a nicer output (you can compare the first and the second line in this example; the trick is only applied on the second line).

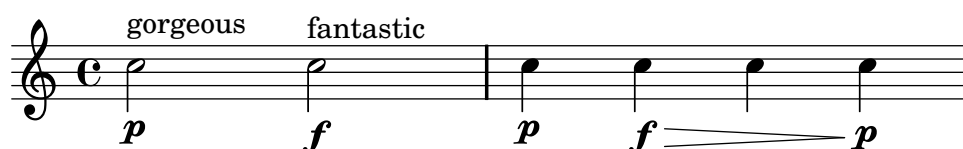
The same idea is used to align the text scripts along their baseline.

```
\paper { indent = 0 line-width = 5\in }

music = \relative c''
{
  c2\p^\markup { "gorgeous" } c\f^\markup { "fantastic" }
  c4\p c \f \> c c \! \p
}

\score
{
  {
    \music \break

    \override DynamicLineSpanner #'staff-padding = #2.0
    \override DynamicLineSpanner #'Y-extent = #'(-1.5 . 1.5)
    \override TextScript #'Y-extent = #'(-1.5 . 1.5)
    \music
  }
}
```



3 gorgeous fantastic

p *f* *p* *f* *p*

MIDI

‘demo-midiinstruments.ly’

Problem: How to know which MidiInstrument would be best for your composition? Solution: A Lilypond demo file.

```
\score {
  \new Staff <<
    \new Voice { \melodie
    } %Voice
  >> %Staff
  \layout { }
} %score

\score {
  \new Staff <<
    \tempo 4 = 180
    %\set Staff.instrumentName="S/A"
    %\set Staff.midiMinimumVolume = #0.2
    %\set Staff.midiMaximumVolume = #0.4
    %\set Voice.dynamicAbsoluteVolumeFunction = #0.6
    \new Voice { r \mf
      \set Staff.midiInstrument="acoustic grand" \melodie
      \set Staff.midiInstrument="bright acoustic" \melodie
      \set Staff.midiInstrument="electric grand" \melodie
      \set Staff.midiInstrument="honky-tonk" \melodie
      \set Staff.midiInstrument="electric piano 1" \melodie
      \set Staff.midiInstrument="electric piano 2" \melodie
      \set Staff.midiInstrument="harpsichord" \melodie
      \set Staff.midiInstrument="clav" \melodie
      \set Staff.midiInstrument="celesta" \melodie
      \set Staff.midiInstrument="glockenspiel" \melodie
      \set Staff.midiInstrument="music box" \melodie
      \set Staff.midiInstrument="vibraphone" \melodie
      \set Staff.midiInstrument="marimba" \melodie
      \set Staff.midiInstrument="xylophone" \melodie
      \set Staff.midiInstrument="tubular bells" \melodie
      \set Staff.midiInstrument="dulcimer" \melodie
      \set Staff.midiInstrument="drawbar organ" \melodie
      \set Staff.midiInstrument="percussive organ" \melodie
      \set Staff.midiInstrument="rock organ" \melodie
      \set Staff.midiInstrument="church organ" \melodie
      \set Staff.midiInstrument="reed organ" \melodie
      \set Staff.midiInstrument="accordion" \melodie
      \set Staff.midiInstrument="harmonica" \melodie
      \set Staff.midiInstrument="concertina" \melodie
      \set Staff.midiInstrument="acoustic guitar (nylon)" \melodie
      \set Staff.midiInstrument="acoustic guitar (steel)" \melodie
      \set Staff.midiInstrument="electric guitar (jazz)" \melodie
      \set Staff.midiInstrument="electric guitar (clean)" \melodie
      \set Staff.midiInstrument="electric guitar (muted)" \melodie
      \set Staff.midiInstrument="overdriven guitar" \melodie
    }
  >>
}
```

```
\set Staff.midiInstrument="distorted guitar" \melodie
\set Staff.midiInstrument="acoustic bass" \melodie
\set Staff.midiInstrument="electric bass (finger)" \melodie
\set Staff.midiInstrument="electric bass (pick)" \melodie
\set Staff.midiInstrument="fretless bass" \melodie
\set Staff.midiInstrument="slap bass 1" \melodie
\set Staff.midiInstrument="slap bass 2" \melodie
\set Staff.midiInstrument="synth bass 1" \melodie
\set Staff.midiInstrument="synth bass 2" \melodie
\set Staff.midiInstrument="violin" \melodie
\set Staff.midiInstrument="viola" \melodie
\set Staff.midiInstrument="cello" \melodie
\set Staff.midiInstrument="contrabass" \melodie
\set Staff.midiInstrument="tremolo strings" \melodie
\set Staff.midiInstrument="pizzicato strings" \melodie
\set Staff.midiInstrument="orchestral strings" \melodie
\set Staff.midiInstrument="timpani" \melodie
\set Staff.midiInstrument="string ensemble 1" \melodie
\set Staff.midiInstrument="string ensemble 2" \melodie
\set Staff.midiInstrument="synthstrings 1" \melodie
\set Staff.midiInstrument="synthstrings 2" \melodie
\set Staff.midiInstrument="choir aahs" \melodie
\set Staff.midiInstrument="voice oohs" \melodie
\set Staff.midiInstrument="synth voice" \melodie
\set Staff.midiInstrument="orchestra hit" \melodie
\set Staff.midiInstrument="trumpet" \melodie
\set Staff.midiInstrument="trombone" \melodie
\set Staff.midiInstrument="tuba" \melodie
\set Staff.midiInstrument="muted trumpet" \melodie
\set Staff.midiInstrument="french horn" \melodie
\set Staff.midiInstrument="brass section" \melodie
\set Staff.midiInstrument="synthbrass 1" \melodie
\set Staff.midiInstrument="synthbrass 2" \melodie
\set Staff.midiInstrument="soprano sax" \melodie
\set Staff.midiInstrument="alto sax" \melodie
\set Staff.midiInstrument="tenor sax" \melodie
\set Staff.midiInstrument="baritone sax" \melodie
\set Staff.midiInstrument="oboe" \melodie
\set Staff.midiInstrument="english horn" \melodie
\set Staff.midiInstrument="bassoon" \melodie
\set Staff.midiInstrument="clarinet" \melodie
\set Staff.midiInstrument="piccolo" \melodie
\set Staff.midiInstrument="flute" \melodie
\set Staff.midiInstrument="recorder" \melodie
\set Staff.midiInstrument="pan flute" \melodie
\set Staff.midiInstrument="blown bottle" \melodie
\set Staff.midiInstrument="shakuhachi" \melodie
\set Staff.midiInstrument="whistle" \melodie
\set Staff.midiInstrument="ocarina" \melodie
\set Staff.midiInstrument="lead 1 (square)" \melodie
\set Staff.midiInstrument="lead 2 (sawtooth)" \melodie
\set Staff.midiInstrument="lead 3 (calliope)" \melodie
```

```

\set Staff.midiInstrument="lead 4 (chiff)" \melodie
\set Staff.midiInstrument="lead 5 (charang)" \melodie
\set Staff.midiInstrument="lead 6 (voice)" \melodie
\set Staff.midiInstrument="lead 7 (fifths)" \melodie
\set Staff.midiInstrument="lead 8 (bass+lead)" \melodie
\set Staff.midiInstrument="pad 1 (new age)" \melodie
\set Staff.midiInstrument="pad 2 (warm)" \melodie
\set Staff.midiInstrument="pad 3 (polysynth)" \melodie
\set Staff.midiInstrument="pad 4 (choir)" \melodie
\set Staff.midiInstrument="pad 5 (bowed)" \melodie
\set Staff.midiInstrument="pad 6 (metallic)" \melodie
\set Staff.midiInstrument="pad 7 (halo)" \melodie
\set Staff.midiInstrument="pad 8 (sweep)" \melodie
\set Staff.midiInstrument="fx 1 (rain)" \melodie
\set Staff.midiInstrument="fx 2 (soundtrack)" \melodie
\set Staff.midiInstrument="fx 3 (crystal)" \melodie
\set Staff.midiInstrument="fx 4 (atmosphere)" \melodie
\set Staff.midiInstrument="fx 5 (brightness)" \melodie
\set Staff.midiInstrument="fx 6 (goblins)" \melodie
\set Staff.midiInstrument="fx 7 (echoes)" \melodie
\set Staff.midiInstrument="fx 8 (sci-fi)" \melodie
\set Staff.midiInstrument="sitar" \melodie
\set Staff.midiInstrument="banjo" \melodie
\set Staff.midiInstrument="shamisen" \melodie
\set Staff.midiInstrument="koto" \melodie
\set Staff.midiInstrument="kalimba" \melodie
\set Staff.midiInstrument="bagpipe" \melodie
\set Staff.midiInstrument="fiddle" \melodie
\set Staff.midiInstrument="shanai" \melodie
\set Staff.midiInstrument="tinkle bell" \melodie
\set Staff.midiInstrument="agogo" \melodie
\set Staff.midiInstrument="steel drums" \melodie
\set Staff.midiInstrument="woodblock" \melodie
\set Staff.midiInstrument="taiko drum" \melodie
\set Staff.midiInstrument="melodic tom" \melodie
\set Staff.midiInstrument="synth drum" \melodie
\set Staff.midiInstrument="reverse cymbal" \melodie
\set Staff.midiInstrument="guitar fret noise" \melodie
\set Staff.midiInstrument="breath noise" \melodie
\set Staff.midiInstrument="seashore" \melodie
\set Staff.midiInstrument="bird tweet" \melodie
\set Staff.midiInstrument="telephone ring" \melodie
\set Staff.midiInstrument="helicopter" \melodie
\set Staff.midiInstrument="applause" \melodie
\set Staff.midiInstrument="gunshot" \melodie
} %Voice
>> %Staff
\midi { }
} %score

```

Demo of all midi sounds

Myself




```
'ancient-notation-template---modern-transcription-of-mensural-music.ly'
```

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.

```
global = {
  \set Score.skipBars = ##t

  % incipit
  \once \override Score.SystemStartBracket #'transparent = ##t
  \override Score.SpacingSpanner #'spacing-increment = #1.0 % tight spacing
  \key f \major
  \time 2/2
  \once \override Staff.TimeSignature #'style = #'neomensural
  \override Voice.NoteHead #'style = #'neomensural
  \override Voice.Rest #'style = #'neomensural
  \set Staff.printKeyCancellation = ##f
  \cadenzaOn % turn off bar lines
  \skip 1*10
  \once \override Staff.BarLine #'transparent = ##f
  \bar "||"
  \skip 1*1 % need this extra \skip such that clef change comes
             % after bar line
  \bar ""

  % main
  \revert Score.SpacingSpanner #'spacing-increment % CHECK: no effect?
  \cadenzaOff % turn bar lines on again
  \once \override Staff.Clef #'full-size-change = ##t
  \set Staff.forceClef = ##t
  \key g \major
  \time 4/4
  \override Voice.NoteHead #'style = #'default
  \override Voice.Rest #'style = #'default

  % FIXME: setting printKeyCancellation back to #t must not
  % occur in the first bar after the incipit. Dto. for forceClef.
  % Therefore, we need an extra \skip.
  \skip 1*1
  \set Staff.printKeyCancellation = ##t
  \set Staff.forceClef = ##f

  \skip 1*7 % the actual music

  % let finis bar go through all staves
  \override Staff.BarLine #'transparent = ##f

  % finis bar
  \bar "|."
}
```



```

discantusNotes = {
  \transpose c' c'' {
    \set Staff.instrumentName = "Discantus  "

    % incipit
    \clef "neomensural-c1"
    c'1. s2  % two bars
    \skip 1*8 % eight bars
    \skip 1*1 % one bar

    % main
    \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 \override NoteHead #'transparent = ##t c'1 |
    b\breve |
  }
}

```

```

discantusLyrics = \lyricmode {
  % incipit
  IV-

  % main
  Ju -- bi -- |
  la -- te De -- |
  o, om --
  nis ter -- |
  ra, __ om- |
  "... " |
  -us. |
}

```

```

altusNotes = {
  \transpose c' c'' {
    \set Staff.instrumentName = "Altus  "

    % incipit
    \clef "neomensural-c3"
    r1          % one bar
    f1. s2      % two bars
    \skip 1*7 % seven bars
    \skip 1*1 % one bar

    % main
    \clef "treble"
    r2 g2. e4 fis g | % two bars
    a2 g4 e |
  }
}

```

```

        fis g4.( fis16 e fis4) |
        g1 |
        \once \override NoteHead #'transparent = ##t g1 |
        g\breve |
    }
}

altusLyrics = \lyricmode {
    % incipit
    IV-

    % main
    Ju -- bi -- la -- te | % two bars
    De -- o, om -- |
    nis ter -- ra, |
    "... " |
    -us. |
}

tenorNotes = {
    \transpose c' c' {
        \set Staff.instrumentName = "Tenor "

        % incipit
        \clef "neomensural-c4"
        r\longa % four bars
        r\breve % two bars
        r1 % one bar
        c'1. s2 % two bars
        \skip 1*1 % one bar
        \skip 1*1 % one bar

        % main
        \clef "treble_8"
        R1 |
        R1 |
        R1 |
        r2 d'2. d'4 b e' | % two bars
        \once \override NoteHead #'transparent = ##t e'1 |
        d'\breve |
    }
}

tenorLyrics = \lyricmode {
    % incipit
    IV-

    % main
    Ju -- bi -- la -- te | % two bars
    "... " |
    -us. |
}

```

```

bassusNotes = {
  \transpose c' c' {
    \set Staff.instrumentName = "Bassus  "

    % incipit
    \clef "bass"
    r\maxima % eight bars
    f1. s2   % two bars
    \skip 1*1 % one bar

    % main
    \clef "bass"
    R1 |
    R1 |
    R1 |
    R1 |
    g2. e4 |
    \once \override NoteHead #'transparent = ##t e1 |
    g\breve |
  }
}

bassusLyrics = \lyricmode {
  % incipit
  IV-

  % main
  Ju -- bi- |
  "... " |
  -us. |
}

\score {
  \new StaffGroup = choirStaff <<
    \new Voice =
      "discantusNotes" << \global \discantusNotes >>
    \new Lyrics =
      "discantusLyrics" \lyricsto discantusNotes { \discantusLyrics }
    \new Voice =
      "altusNotes" << \global \altusNotes >>
    \new Lyrics =
      "altusLyrics" \lyricsto altusNotes { \altusLyrics }
    \new Voice =
      "tenorNotes" << \global \tenorNotes >>
    \new Lyrics =
      "tenorLyrics" \lyricsto tenorNotes { \tenorLyrics }
    \new Voice =
      "bassusNotes" << \global \bassusNotes >>
    \new Lyrics =
      "bassusLyrics" \lyricsto bassusNotes { \bassusLyrics }
  >>

```

```

\layout {
  \context {
    \Score

    % no bars in staves
    \override BarLine #'transparent = ##t

    % incipit should not start with a start delimiter
    \remove "System_start_delimiter_engraver"
  }
  \context {
    \Voice

    % no slurs
    \override Slur #'transparent = ##t

    % Comment in the below "\remove" command to allow line
    % breaking also at those barlines where a note overlaps
    % into the next bar. 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"
  }
}

```

Discantus

IV-

Altus

IV-

Tenor

IV-

Bassus

IV-

Ju - bi - la - te De -

Ju - bi - la - te

Ju - bi - la - te

IV-

3

o, om - nis ter - ra, — om - ... -us.

De - o, om - nis ter - ra, ... -us.

8

Ju - bi - la - te ... -us.

Ju - bi - ... -us.

'jazz-combo-template.ly'

Jazz tune for combo (horns, guitar, piano, bass, drums).

Song
(tune)

Me

moderato

Swing

Trumpet

Alto Sax

Bari Sax

Trombone

Guitar

Piano

Bass

Drums

B Δ Solo C#m⁷

Cm Δ D Δ /9

‘piano-template-simple.ly’

Here is a simple piano staff with some notes.

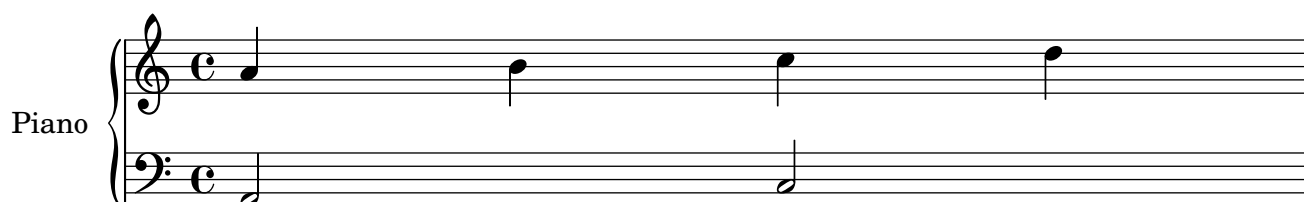
```
upper = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

  a b c d
}

lower = \relative c {
  \clef bass
  \key c \major
  \time 4/4

  a2 c
}

\score {
  \new PianoStaff <<
    \set PianoStaff.instrumentName = "Piano  "
    \new Staff = "upper" \upper
    \new Staff = "lower" \lower
  >>
  \layout { }
  \midi { }
}
```



`'piano-template-with-centered-dynamics.ly'`

Many piano scores have the dynamics centered between the two staves. This requires a bit of tweaking to implement, but since the template is right here, you don't have to do the tweaking yourself.

```
upper = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

  a b c d
}

lower = \relative c {
  \clef bass
  \key c \major
  \time 4/4

  a2 c
}

dynamics = {
  s2\fff\> s4
  s\!\pp
}

pedal = {
  s2\sustainDown s2\sustainUp
}

\score {
  \new PianoStaff <<
    \new Staff = "upper" \upper
    \new Dynamics = "dynamics" \dynamics
    \new Staff = "lower" <<
      \clef bass
      \lower
    >>
    \new Dynamics = "pedal" \pedal
  >>
  \layout {
    \context {
      \type "Engraver_group"
```

```

\name Dynamics
\alias Voice % So that \cresc works, for example.
\consists "Output_property_engraver"

\override VerticalAxisGroup #'minimum-Y-extent = #'(-1 . 1)
\override DynamicLineSpanner #'Y-offset = #0
pedalSustainStrings = #'("Ped." "*Ped." "*")
pedalUnaCordaStrings = #'("una corda" "" "tre corde")

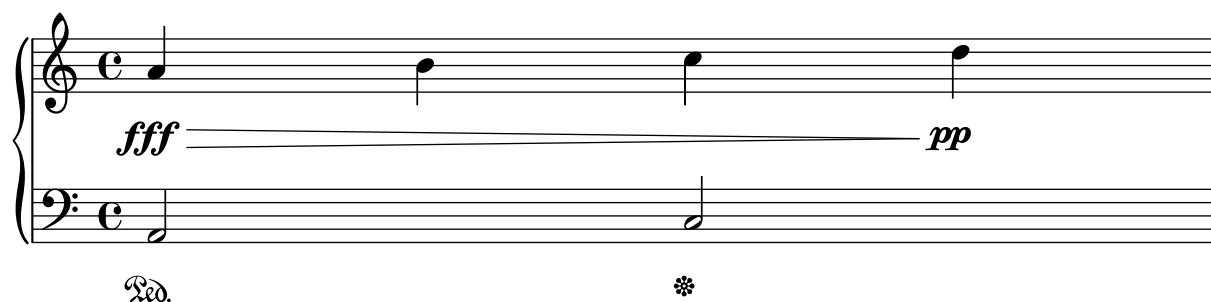
\consists "Piano_pedal_engraver"
\consists "Script_engraver"
\consists "Dynamic_engraver"
\consists "Text_engraver"

\override TextScript #'font-size = #2
\override TextScript #'font-shape = #'italic

\consists "Skip_event_swallow_translator"

\consists "Axis_group_engraver"
}
\context {
  \PianoStaff
  \accepts Dynamics
}
}
\score {
  \new PianoStaff <<
    \new Staff = "upper" << \upper \dynamics >>
    \new Staff = "lower" << \lower \dynamics >>
    \new Dynamics = "pedal" \pedal
  >>
  \midi {
    \context {
      \type "Performer_group"
      \name Dynamics
      \consists "Piano_pedal_performer"
    }
    \context {
      \PianoStaff
      \accepts Dynamics
    }
  }
}
}

```

‘piano-template-with-centered-lyrics.ly’

Instead of having a full staff for the melody and lyrics, you can place the lyrics between the piano staff (and omit the separate melody staff).

```
upper = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

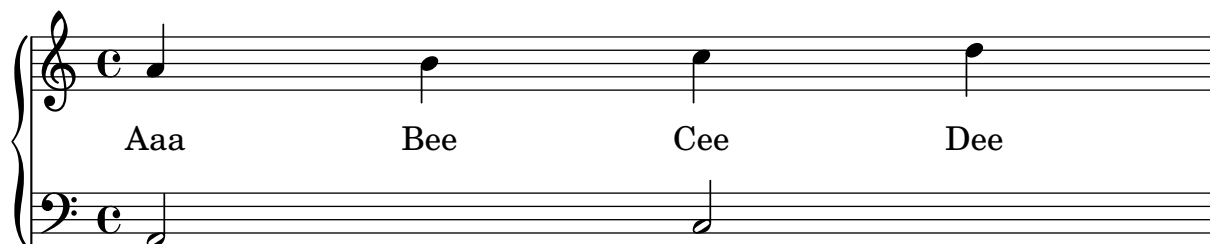
  a b c d
}

lower = \relative c {
  \clef bass
  \key c \major
  \time 4/4

  a2 c
}

text = \lyricmode {
  Aaa Bee Cee Dee
}

\score {
  \new GrandStaff <<
    \new Staff = upper { \new Voice = "singer" \upper }
    \new Lyrics \lyricsto "singer" \text
    \new Staff = lower {
      \clef bass
      \lower
    }
  >>
  \layout {
    \context { \GrandStaff \accepts "Lyrics" }
    \context { \Lyrics \consists "Bar_engraver" }
  }
  \midi { }
}
```



‘piano-template-with-melody-and-lyrics.ly’

Here is a typical song format: one staff with the melody and lyrics, with piano accompaniment underneath.

```
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

    a b c d
}

lower = \relative c {
    \clef bass
    \key c \major
    \time 4/4

    a2 c
}

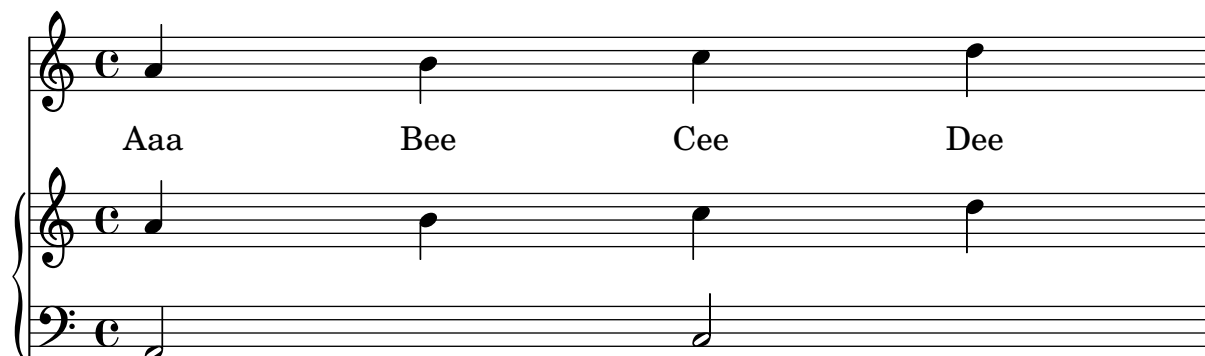
\score {
  <<
    \new Voice = "mel" {
      \autoBeamOff
      \melody
    }
    \new Lyrics \lyricsto mel \text

    \new PianoStaff <<
      \new Staff = "upper" \upper
      \new Staff = "lower" \lower
    >>
  >>
  \layout {
```

```

\context { \RemoveEmptyStaffContext }
}
\midi { }
}

```



`'score-for-diatonic-accordion.ly'`

A template to write score for a diatonic accordion.

- There is a horizontal staff indicating if the accordion must be pushed (thick line) or pulled (thin line)

- There is a small rhythmic staff with lyrics that describes the bass buttons to press. The bar-lines are made of gridlines (Gridlines, not a really satisfying solution, but the best I found)

- The tabulator staff for diatonic accordions shows the geographic position of the buttons and not (as for every other instrument) the pitch of the tones. The keys on the melody-side of the accordion are placed in three columns and about 12 rows. In the tabulator staff notation the - most outer column is described with notes between lines - most inner column is described with notes between lines with a cross as accidental - middle column is described with notes on a line, whereby the row in the middle is represented on the middle line in the staff.

Some words to transpose piano note to the diatonic accordion. —————

1. Every diatonic accordion is built for some keys only. For example for the key of C-major and F-major. So it is important to transpose a piano melody to match one of these keys. Transpose the source code, not only the output because you need this code later on to translate it once more to the tabulator staff. This can be done with the command `'displayLilyMusic'`.

2. You have to alternate the push and pull-direction of the accordion regularly. If the player has a too long part to pull the accordion gets broken. At the other hand some harmonies are only available in one direction (push or pull) Considering this decide which parts of the melody are the push-parts and whic the pull-parts.

3. For each pull- / or push-part translate the piano notes to the according tabulatur representation.

This snippet comes with a useful optional macro for the jEdit text editor.

% Created on Sat Aug 17 2007 by ak

```
verse= \lyricmode { Wie gross bist du! Wie gross bist du! }
```

```

harmonies = \new ChordNames \chordmode {
  \germanChords \set chordChanges = ##t
  bes8 bes8 bes8 es2 f bes1
}

```

```

}

NoStem = \override Stem #'transparent = ##t
NoNoteHead= \override NoteHead #'transparent = ##t
ZeroBeam = \override Beam #'positions = #'(0 . 0)

staffTabLine = \new Staff \with { \remove "Time_signature_engraver" \remove "Clef_engraver"
  \override Staff.StaffSymbol #'line-positions = #'( 0 )
% Shows one horizontal line. The vertical line (simulating a bar-line) is simulated with a g
  \set Staff.midiInstrument="choir aahs"
  \key c \major
  \relative c''
  {
    % disable the following line to see the the noteheads while writing
    \NoNoteHead
    \override NoteHead #'no-ledgers = ##t

    % The beam between 8th-notes is used to draw the push-line
    %How to fast write the push-lines:
    %      1. write repeatedly 'c c c c c c c c |' for the whole length
    %      2. uncomment the line \NoNoteHead
    %      3. compile
    %      4. Mark the positions on which push/pull changes.
    %          In the score-picture click on the position the push- or
    %          (on the noteHead, the cursor will change to a hand-icon)
    %          The cursor in the source code will jump just at this position
    %      a) If a push-part starts there, replace the 'c' by an 'e[
    %      b) If a pull-part starts there, replace the 'c' by an 's'
    %      5. Switch into 'overwrite-mode' by pressing the 'ins' key.
    %      6. For the pull-parts overwrite the 'c' with 's'
    %      7. For every push-part replace the last 'c' with 'e]'
    %      8. Switch into 'insert-mode' again
    %      9. At last it should look lik e.g. (s s e[ c | c c c c c c
    %      10. re-enable the line \NoNoteHead
    \autoBeamOff
    \ZeroBeam
    s8 s s | e[ c c c c c c e] | s s s s s
  }
}

%{
notePush= {      e      f      fis      g      a      c'      c'      d'      ees'      e'
tabPush=  {      g      f      e      b      a      d'      c'      bisis  disis'  f'

notePull= {      g      aes      bes      b      c'      cis'      d'      ees'      e'      f'
tabPull=  {      g      e      f      b      a      disis'  d'      bisis  c'      f'
}%}

% Accordion melody in tabulator score
% 1. Place a copy of the piano melody below
% 2. Separate piano melody into pull- and push-parts according to the staffTabLine you've a
% 3. For each line: Double the line. Remark the 1st one (Keeps unchanged as reference) and t

```

```
% or the macros 'conv2diaton push.bsh' and 'conv2diaton pull.bsh'
% Tips:
% - In jEdit Search & Replace mark the Option 'Keep Dialog'
```

```
AccordionTabTwoCBesDur= {
  % pull 1
  %<f' bes'>8 <f' a'>8 <d' bes'>8 |
  <g' a'>8 <g' b'>8 <e' a'>8 |
  % push 2
  %<g' c'>4 <f' d'> <g' ees'> <f' a'> |
  <g' a'>4 <d' eisis'> <g' bisis'> <d' f'> |
  % pull 3
  % <f' bes'>2 r8 }
  <g' a'>2 r8 }
```

```
  AccordionTab= { \dynamicUp
% 1. Place a copy of the piano melody above
% 2. Separate piano melody into pull- and push-parts according to the staffTabLine you've a
% 3. For each line: Double the line. Remark the 1st one (Keeps unchanged as reference) and t
% change the second line using the transformation paper
% Tips:
% - In jEdit Search & Replace mark the Option 'Keep Dialog'
% -
\AccordionTabTwoCBesDur
}
```

```
\layout {
\context {
  \Staff
  \consists "Grid_point_engraver"

  gridInterval = #(ly:make-moment 4 4) % 4/4 - tact. How many beats per bar

  % The following line has to be adjusted O-F-T-E-N.
  \override GridPoint #'Y-extent = #'(-2 . -21)
}
\context {
  \ChoirStaff
  \remove "System_start_delimiter_engraver"
}
}
```

```
staffVoice = \new Staff=astaffvoice {
  \time 4/4
  \set Staff.instrumentName="Voice"
  \set Staff.midiInstrument="voice oohs"
  \key bes \major
  \partial 8*3
  \clef treble
  {
```

```

        \context Voice = "melodyVoi"
        { <' bes'>8 <' a'>8 <d' bes'>8 | <' c''>4 <' d''> <' es''> <' a'> | <
\bar "|."
}

}

staffAccordionMel = \new Staff \with { \remove "Clef_engraver" } {
    #(set-accidental-style 'forget) %Set the accidentals (Vorzeichen) for each note,
                                %do not remember them for the rest of the measure.
    \time 4/4
    \set Staff.instrumentName="Accordion"
    \set Staff.midiInstrument="voice oohs"
    \key c \major
    \clef treble
    { \AccordionTab \bar "|." }
}

AltOn = #(define-music-function (parser location mag) (number?)
    #{ \override Stem #'length = #$( * 7.0 mag)
      \override NoteHead #'font-size =
        #$(inexact->exact ( * (/ 6.0 (log 2.0)) (log mag))) #})

AltOff = {
    \revert Stem #'length
    \revert NoteHead #'font-size
}

BassRhytm = {s4 s8 | c2 c2 | c2 s8 }
LyricBassRhythmI= \lyricmode { c b | c }

staffBassRhytm = \new Staff=staffbass \with { \remove "Clef_engraver" } {
    % This is not a RhythmicStaff because it must be possible to append lyrics.

    \override Score.GridLine #'extra-offset = #'( 13.0 . 0.0 ) % x.y
    \override Staff.StaffSymbol #'line-positions = #'( 0 )
    % Shows one horizontal line. The vertical line (simulating a bar-line) is simulated
    % Search for 'grid' in this page to find all related functions
    \time 4/4
    {
        \context Voice = "VoiceBassRhytm"
        \stemDown \AltOn #0.6
        \relative c''
        {
            \BassRhytm
        }
        \AltOff
    }
    \bar "|."
}

}

\new Score
\with {

```

```

\consists "Grid_line_span_engraver" %The vertical line (simulating a bar-line) in the staff
}
\new ChoirStaff
  <<
    \harmonies
    \staffVoice
    \context Lyrics = "lmelodyVoi" \with {alignBelowContext=astaffvoice} { \lyric
    \staffAccordionMel
    \staffTabLine
    \staffBassRhythm
    \context Lyrics = "lBassRhythmAboveI" \with {alignAboveContext=staffbass} \lyricsto V
  >>

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                APPENDIX                                %
%      macro 'macro_conv2diaton_push.bsh' for jedit editor              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%{
// original saved in 'Handorg_Adria_Diaton_III.xls'
// Save this buffer, to the other recorded macros in the jedit editor
// and the macro should appear in the
// Macros menu.

// /ak 17.8.07 This macro from converts lilypond piano notation into
// lilypond tabulator notation for the push-part (at the bottom the pull-part) of a diatonic
// It replaces the piano notes of the line where the cursor is by the accordion notation

// Known issues: 1) A note at the end of line is not replaced

textArea.goToEndOfWhiteSpace(false);
textArea.goToStartOfWhiteSpace(true);

String firstName, lastName;

int ReplaceTextInSelection(String sfind, String sreplace)
{
//MsgConcat = new StringBuffer(512);
//MsgConcat.append("Ha");

//Macros.message(view, "On that line replace \"" + sfind + "\" by \"" + sreplace+ "\"");
SearchAndReplace.setSearchString(sfind.toString());
SearchAndReplace.setReplaceString(sreplace.toString());
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setIgnoreCase(true);
SearchAndReplace.replace(view);
SearchAndReplace.setRegexp(true);
return 1;
}

```


[illegible]

[illegible]

```
smainfind="(\\s|~|<|\\{)(tmp)()"; smainrepl="$1$3"; ReplaceTextInSelection( smainfind, smainrepl, 0);
```

 $\ast/$ $\% \}$

Voice

Wie gross bist du! Wie gross bist du!

Accordion

B E^b F B

c c b c

‘single-staff-template-with-notes,-lyrics,-and-chords.ly’

This template allows you to prepare a song with melody, words, and chords.

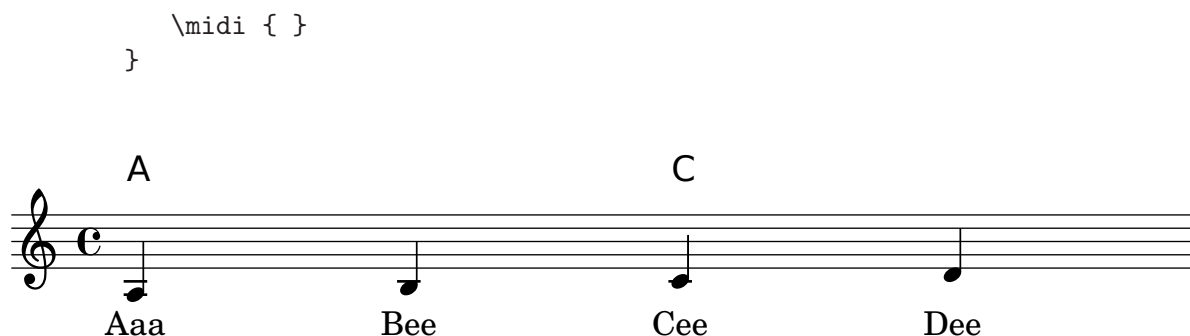
```
melody = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

  a b c d
}

text = \lyricmode {
  Aaa Bee Cee Dee
}

harmonies = \chordmode {
  a2 c2
}

\score {
  <<
    \new ChordNames {
      \set chordChanges = ##t
      \harmonies
    }
    \new Voice = "one" {
      \autoBeamOff
      \melody
    }
    \new Lyrics \lyricsto "one" \text
  >>
  \layout { }
```



‘single-staff-template-with-notes-and-chords.ly’

Want to prepare a lead sheet with a melody and chords? Look no further!

```

melody = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

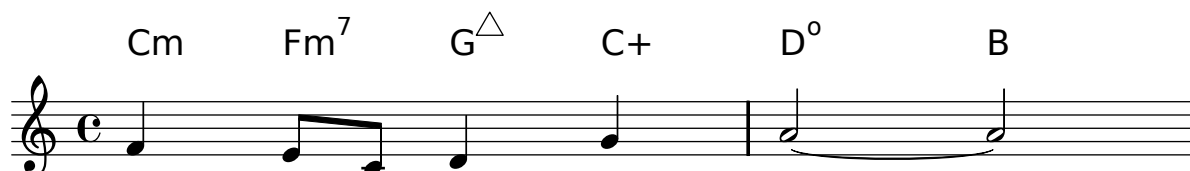
  f4 e8[ c] d4 g |
  a2 ~ a2 |
}

harmonies = \chordmode {
  c4:m f:min7 g:maj c:aug d2:dim b:sus
}

\score {
  <<
    \new ChordNames {
      \set chordChanges = ##t
      \harmonies
    }
    \new Staff \melody
  >>

  \layout{ }
  \midi { }
}

```



‘single-staff-template-with-notes-and-lyrics.ly’

This small template demonstrates a simple melody with lyrics. Cut and paste, add notes, then words for the lyrics. This example turns off automatic beaming, which is common for vocal parts. If you want to use automatic beaming, you’ll have to change or comment out the relevant line.

```

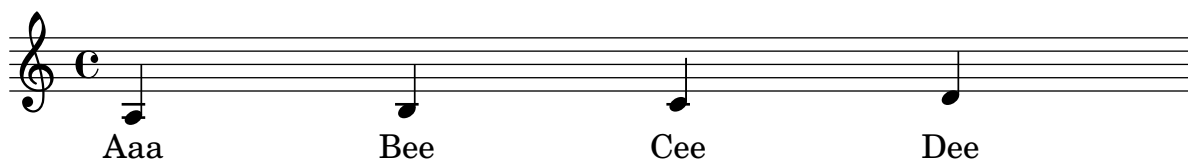
melody = \relative c' {
    \clef treble
    \key c \major
    \time 4/4

    a4 b c d
}

text = \lyricmode {
    Aaa Bee Cee Dee
}

\score{
    <<
        \new Voice = "one" {
            \autoBeamOff
            \melody
        }
        \new Lyrics \lyricsto "one" \text
    >>
    \layout { }
    \midi { }
}

```



‘single-staff-template-with-only-notes.ly’

This very simple template gives you a staff with notes, suitable for a solo instrument or a melodic fragment. Cut and paste this into a file, add notes, and you’re finished!

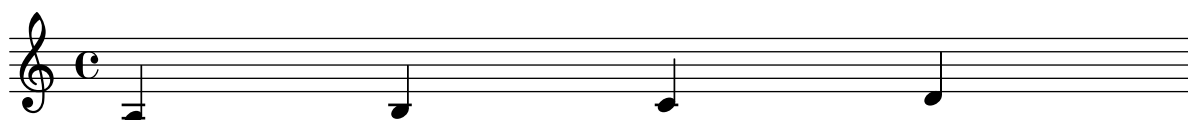
```

melody = \relative c' {
    \clef treble
    \key c \major
    \time 4/4

    a4 b c d
}

\score {
    \new Staff \melody
    \layout { }
    \midi {}
}

```



‘string-quartet-template-simple.ly’

This template demonstrates a simple string quartet. It also uses a `\global` section for time and key signatures

```
global= {
  \time 4/4
  \key c \major
}

violinOne = \new Voice { \relative c''{
  \set Staff.instrumentName = "Violin 1 "

  c2 d e1

  \bar "|" }}
violinTwo = \new Voice { \relative c''{
  \set Staff.instrumentName = "Violin 2 "

  g2 f e1

  \bar "|" }}
viola = \new Voice { \relative c' {
  \set Staff.instrumentName = "Viola "
  \clef alto

  e2 d c1

  \bar "|" }}
cello = \new Voice { \relative c' {
  \set Staff.instrumentName = "Cello      "
  \clef bass

  c2 b a1

  \bar "|" }}

\score {
  \new StaffGroup <<
    \new Staff << \global \violinOne >>
    \new Staff << \global \violinTwo >>
    \new Staff << \global \viola >>
    \new Staff << \global \cello >>
  >>
  \layout { }
  \midi { }
}
```

`'string-quartet-template-with-separate-parts.ly'`

The "String quartet template" snippet produces a nice string quartet, but what if you needed to print parts? This new template demonstrates how to use the `\tag` feature to easily split a piece into individual parts.

You need to split this template into separate files; the filenames are contained in comments at the beginning of each file. `piece.ly` contains all the music definitions. The other files – `score.ly`, `vn1.ly`, `vn2.ly`, `vla.ly`, and `vlc.ly` – produce the appropriate part.

Do not forget to remove specified comments when using separate files!

```

%%%%% piece.ly
%%%%% (This is the global definitions file)

global= {
  \time 4/4
  \key c \major
}

Violinone = \new Voice { \relative c' {
  \set Staff.instrumentName = "Violin 1 "

  c2 d e1

\bar "|" } } %*****
Violintwo = \new Voice { \relative c' {
  \set Staff.instrumentName = "Violin 2 "

  g2 f e1

\bar "|" } } %*****
Viola = \new Voice { \relative c' {
  \set Staff.instrumentName = "Viola "
  \clef alto

  e2 d c1

\bar "|" } } %*****
Cello = \new Voice { \relative c' {
  \set Staff.instrumentName = "Cello "
  \clef bass

```

```

c2 b a1

\bar "|" ."} } %*****

music = {
  <<
    \tag #'score \tag #'vn1 \new Staff { << \global \Violinone >> }
    \tag #'score \tag #'vn2 \new Staff { << \global \Violintwo>> }
    \tag #'score \tag #'vla \new Staff { << \global \Viola>> }
    \tag #'score \tag #'vlc \new Staff { << \global \Cello>> }
  >>
}

%%% These are the other files you need to save on your computer

%%%%% score.ly
%%%%% (This is the main file)

%\include "piece.ly" %%% uncomment this line when using a separate file
#(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

‘vocal-ensemble-template-with-automatic-piano-reduction.ly’

This template adds an automatic piano reduction to the standard SATB vocal score demonstrated in "Vocal ensemble template". This demonstrates one of the strengths of LilyPond – you can use a music definition more than once. If you make any changes to the vocal notes (say, `tenorMusic`), then the changes will also apply to the piano reduction.

```

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 Lyrics = sopranos { s1 }
      \new Staff = women <<
        \new Voice =
          "sopranos" { \voiceOne << \global \sopMusic >> }
        \new Voice =
          "altos" { \voiceTwo << \global \altoMusic >> }
      >>
      \new Lyrics = "altos" { s1 }
      \new Lyrics = "tenors" { s1 }
      \new Staff = men <<
        \clef bass
        \new Voice =
          "tenors" { \voiceOne << \global \tenorMusic >> }
        \new Voice =
          "basses" { \voiceTwo << \global \bassMusic >> }
      >>
      \new Lyrics = basses { s1 }

      \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 >>
>>
>>
>>
\layout {
  \context {
    % a little smaller so lyrics
    % can be closer to the staff
    \Staff
    \override VerticalAxisGroup #'minimum-Y-extent = #'(-3 . 3)
  }
}

```

hi hi hi hi

ha ha ha ha

hu hu hu hu

ho ho ho ho

‘vocal-ensemble-template-with-lyrics-aligned-below-and-above-the-staves.ly’

This template is basically the same as the simple "Vocal ensemble" template, with the exception that here all the lyrics lines are placed using `alignAboveContext` and `alignBelowContext`.

```

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
    >>

  \layout {
    \context {

```

```

        % a little smaller so lyrics
        % can be closer to the staff
        \Staff
        \override VerticalAxisGroup #'minimum-Y-extent = #'(-3 . 3)
    }
}
}

\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 \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 \lyricsto basses \bassWords
  >>

  \layout {
    \context {
      % a little smaller so lyrics
      % can be closer to the staff
      \Staff
      \override VerticalAxisGroup #'minimum-Y-extent = #'(-3 . 3)
    }
  }
}

```

‘vocal-ensemble-template.ly’

Here is a standard four-part SATB vocal score. With larger ensembles, it's often useful to include a section which is included in all parts. For example, the time signature and key signatures are almost always the same for all parts. Like in the "Hymn" template, the four voices are regrouped on only two staves.

```
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 Lyrics = sopranos { s1 }
    \new Staff = women <<
      \new Voice =
        "sopranos" { \voiceOne << \global \sopMusic >> }
      \new Voice =
        "altos" { \voiceTwo << \global \altoMusic >> }
    >>
    \new Lyrics = "altos" { s1 }
    \new Lyrics = "tenors" { s1 }
    \new Staff = men <<
      \clef bass
      \new Voice =
        "tenors" { \voiceOne <<\global \tenorMusic >> }
      \new Voice =
        "basses" { \voiceTwo <<\global \bassMusic >> }
    >>
    \new Lyrics = basses { s1 }

    \context Lyrics = sopranos \lyricsto sopranos \sopWords
    \context Lyrics = altos \lyricsto altos \altoWords
    \context Lyrics = tenors \lyricsto tenors \tenorWords
    \context Lyrics = basses \lyricsto basses \bassWords
  >>

  \layout {
    \context {
      % a little smaller so lyrics
      % can be closer to the staff
      \Staff
      \override VerticalAxisGroup #'minimum-Y-extent = #'(-3 . 3)
    }
  }
}

```

hi hi hi hi

ha ha ha ha

hu hu hu hu

ho ho ho ho