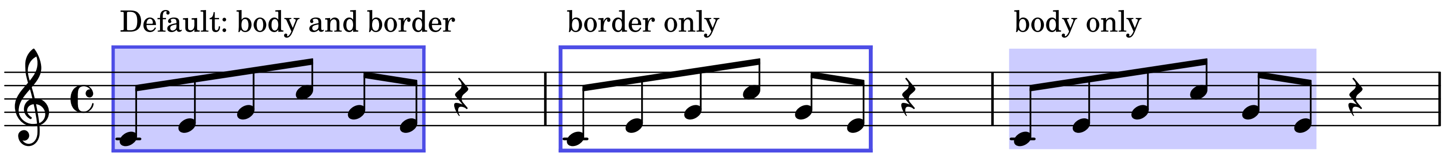
Frames & Rectangles - Showcase

The `frames` module of `anaLYsis` provides numerous functions to draw differently looking frames/rectangles. The basic function is `\genericFrame` which takes and - horizontally - surrounds a music expression as its mandatory argument. The appearance of the frame can be specified either persistently with openLilyLib options in the `\setOption analysis.frames` tree or individually by overriding properties in a `\with {}` block (for details see below).

Generic frames

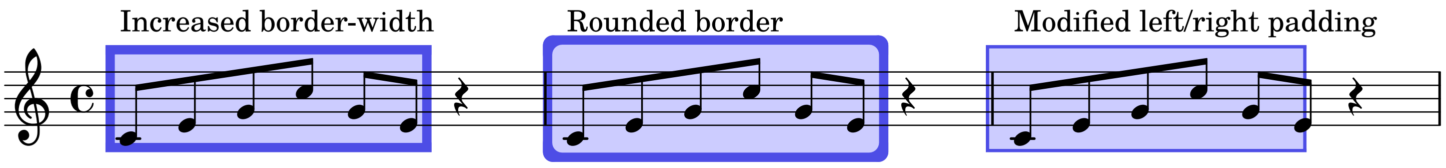
A frame has a body and a border, which by default are printed both.

The color of both elements is controlled with the properties `color` and `border-color` (for example `\setOption analysis.frames.border-color #green`), and they can be made invisible by setting this property to `white` , or they can be completely suppressed by setting it to `#f` .



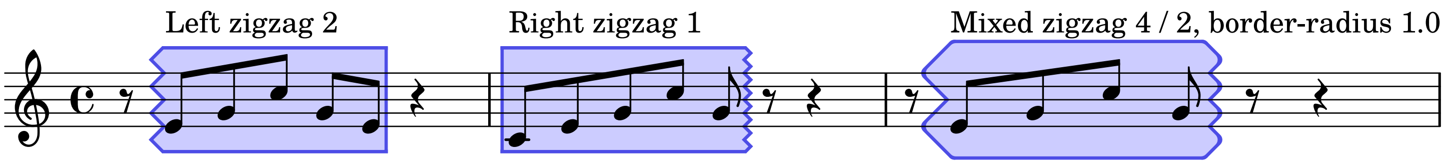
`border-width` (0.25)
`border-radius` (0)
`shorten-pair` #'(0 . 0)

The width (or thickness) of the frame border is controlled with the `border-width` Property. `border-radius` applies a rounded effect on the frame (but will also make it grow outwards). Leaving this at `0` will produce a sharply angled frame. With `shorten-pair` the right and left padding can be modified. Positive values will make the frame narrower while negative values will make it wider. This property should be handled with specific care.



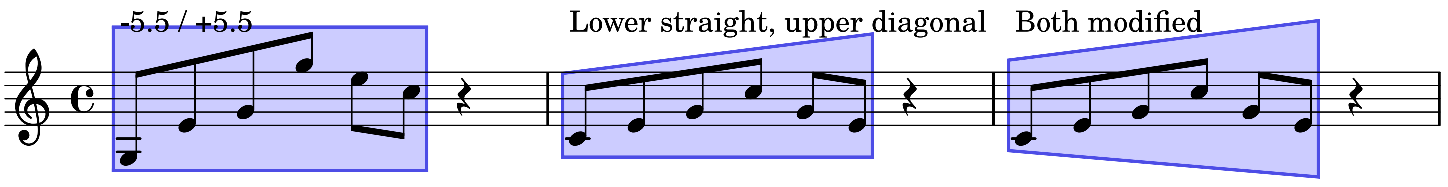
`l-zigzag-width` (0)
`r-zigzag-width` (0)

The left and right edges of the frame can be decorated with a zig-zag line-thickness. This is achieved by setting `l-zigzag-width` or `r-zigzag-width` to a value greater than zero. In order to avoid strange results the value should be a divisor of the distance between lower and upper corner (see next element). `border-radius` will also make the zigzag curved.



`y-lower` (-4)
`y-upper` (4)

The vertical extent of the frame defaults to -4/+4 (in staff spaces from the center staffline). These values can be modified using The `y-lower` and `y-upper` properties. When given a number they cause a horizontal line at that Y position to be drawn. But using a pair of numbers flexible polygons can be created.



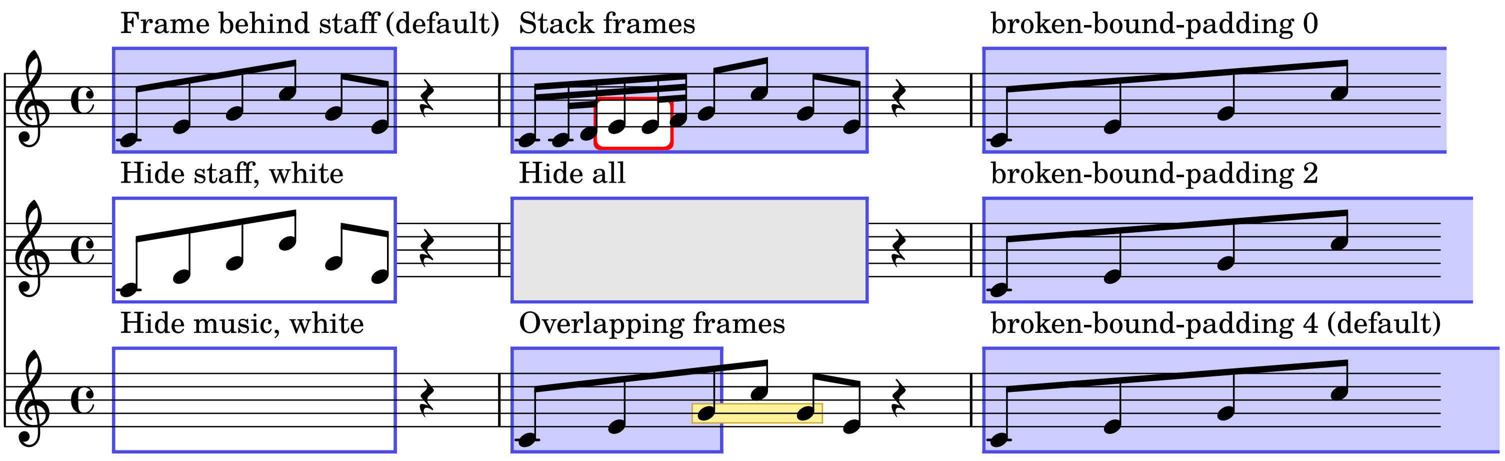
`hide` (none/staff/music/all)
`layer` (-10)
`broken-bound-padding` (4)
`open-on-bottom` (##f)
`open-on-top` (##f)

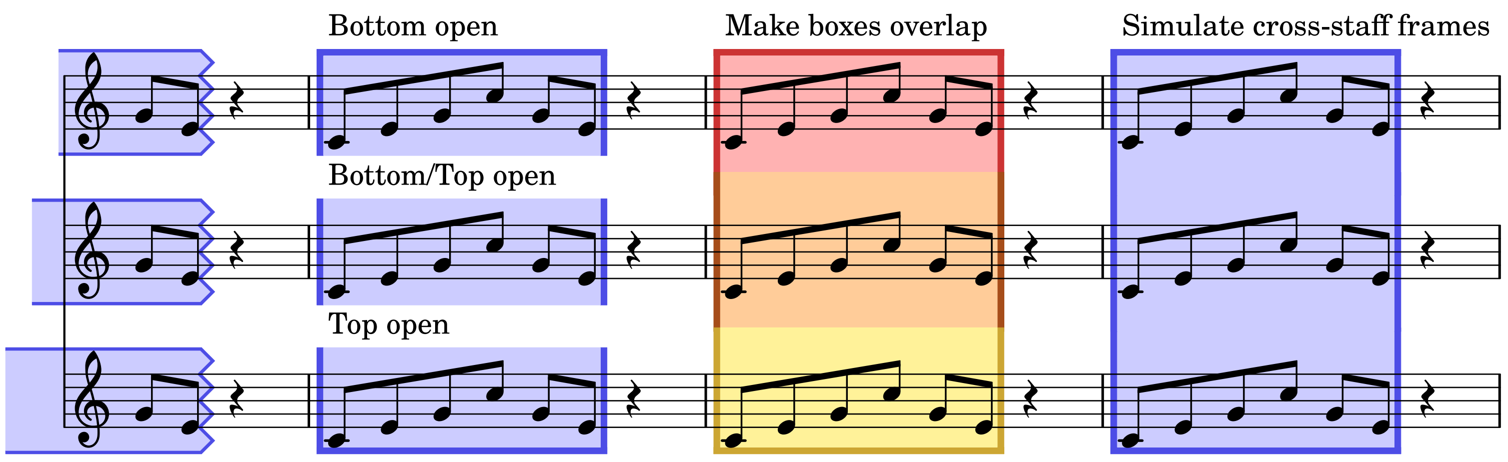
By default frames are printed behind the staff, which is achieved by setting its `layer` property to `-10`. By manually modifying this property the user has detailed control over the stacking of elements. If the `hide` property is set to `staff` the frame is placed between the staff lines and the music. This is achieved by temporarily setting the frame's `layer` to `1` and that of the other score elements to `2`. Setting `hide` to `music` will place the frame in layer `-1` and the music in layer `-2`. So this may interfere with any other `layer` settings the user may have applied otherwise. Setting `hide` to `all` will print the frame in front of everything else (or concretely at the layer `5`. This can be used to reserve space in exercise sheets.

Frames can be nested in a straightforward manner. However, this behaves differently from spanners like manual beams or slurs. In order to print overlapping frames they have to exist in separate voices. **NOTE:** nested frames may behave strangely with other frames acting at the same time. In the example the "hide all" frame makes the stacked frame in the top staff disappear unless the `layer` property is explicitly set to `-1` (or a layer higher than the outer frame's.

If frames are broken around line breaks the broken edge will be printed without border and zigzag lines. The amount of protrusion into the margin can be set with the `broken-bound-padding` property.

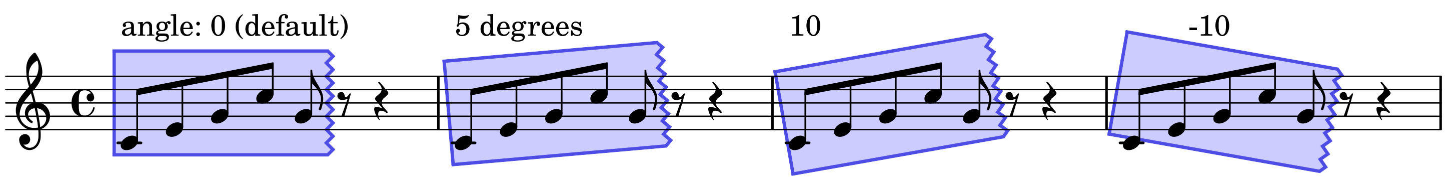
With the boolean `open-on-top` and `open-on-bottom` the top and/or bottom borders can be switched off, which can be used to create fake cross-staff Frames. In that case (and also when using frames broken around line breaks), `border-radius` should be left at `0`.





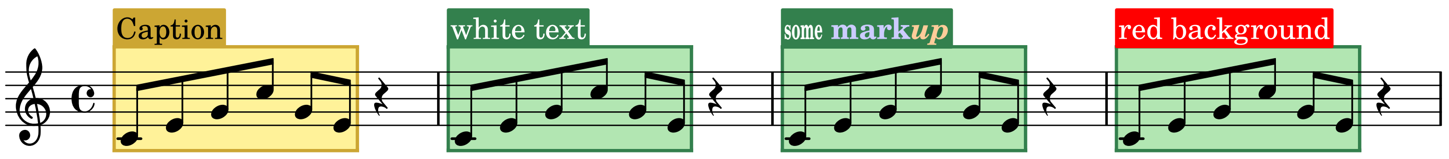
angle (0)

Frames can be rotated couter-clockwise around their center point by setting the `angle` property to a positive value in degrees. Negative values will turn the frame clockwise.



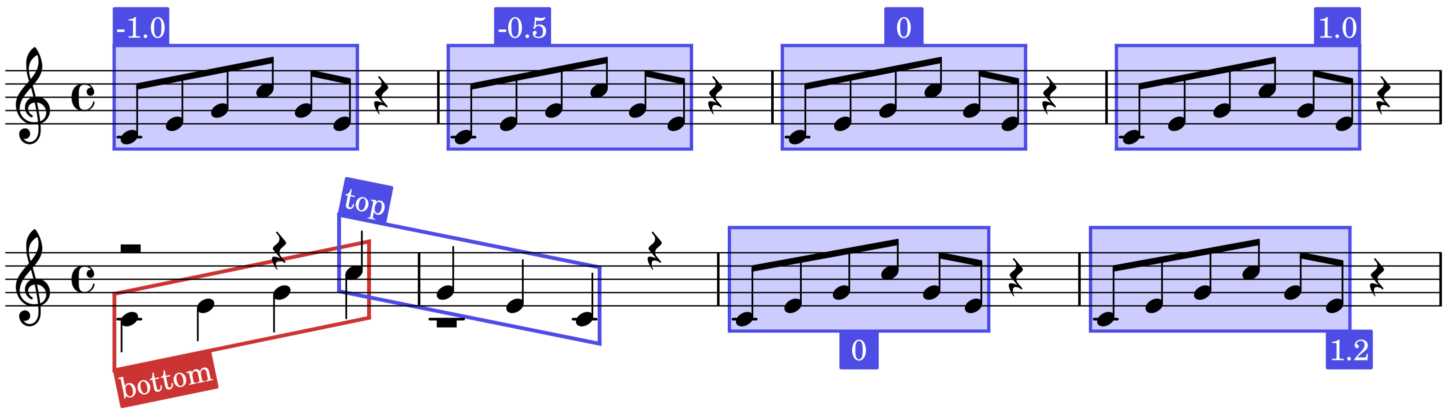
caption (#f)
caption-color (#f)

The `caption` property can be set to a string or markup which will be displayed in a label that is automatically aligned to the frame. By default, the border color will be used for text background. Therefore it can be recommended to choose a light color for the text. Any markup command can be used. Setting the `caption-color` property to a valid color allows to change the caption's background color.



caption-halign (-1)
caption-align-bottom (#f)

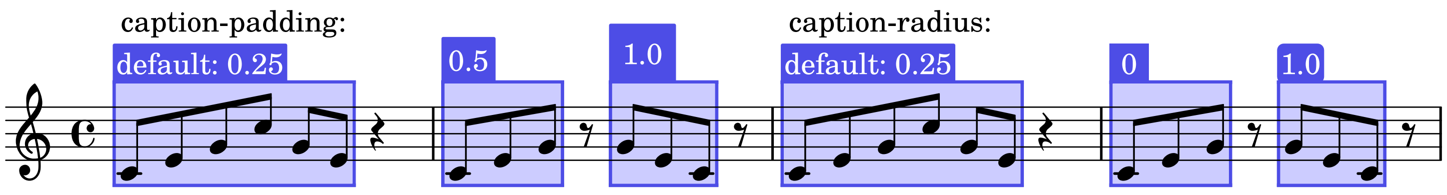
By default, the caption is placed left-aligned which corresponds to the `caption-halign` property being `-1`. `0` will have it centered, `1` will result in a right-aligned caption. Any value in between can be applied, even beyond that range. Setting `caption-align-bottom` to `#t` will move the caption to the bottom edge.



`caption-padding` (0.25)

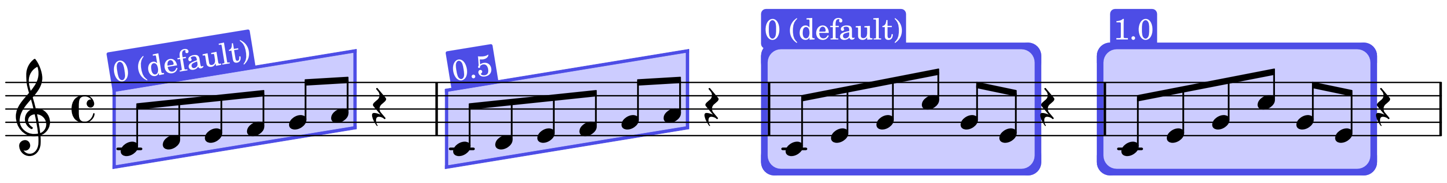
`caption-radius` (0.25)

The distance between the caption text and the label's borders can be controlled with the `caption-padding` property. Setting `caption-radius` to a positive value will apply rounded corners to the caption label.



`caption-translate-x` (0)

In some situations it can look better to manually move the caption away from the frame's corners by a fixed number of spaces. This can be controlled with the `caption-translate-x` property. Positive values will move it to the right, negative values to the left side.



`caption-keep-y` (#f)

The overall height of a markup depends on whether it contains ascenders or descenders:

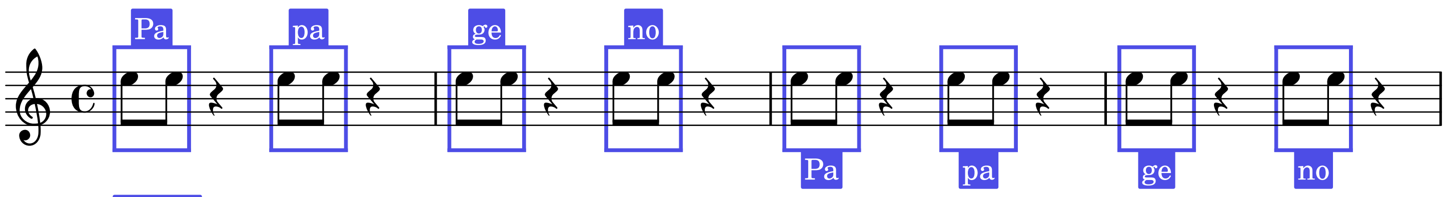
É E e g - Pa pa ge no

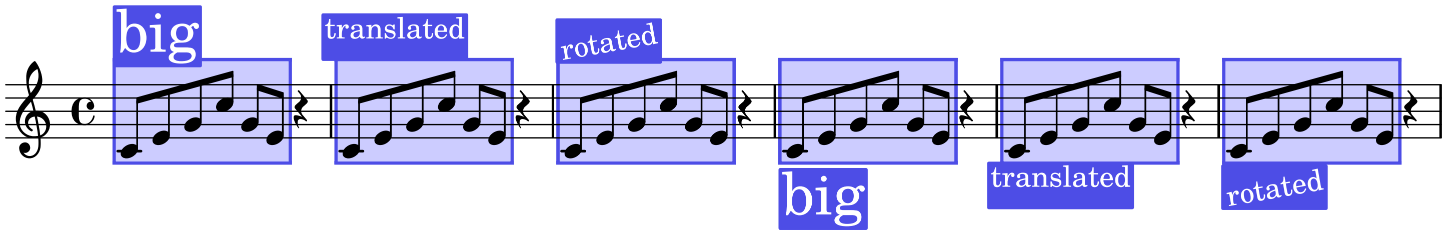
This would make it impossible to always correctly align the labels to the upper or lower frame border. Therefore caption labels are extended to a height that leaves enough space for ascenders and descenders:

É E e g - Pa pa ge no

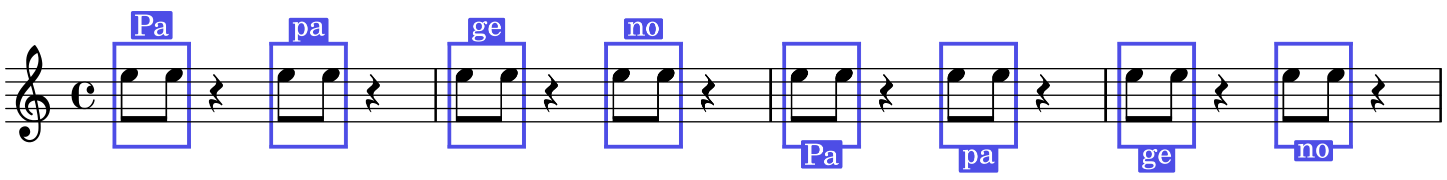
Internally this is done by adding an invisible (and compressed) É and j stencil to every markup:

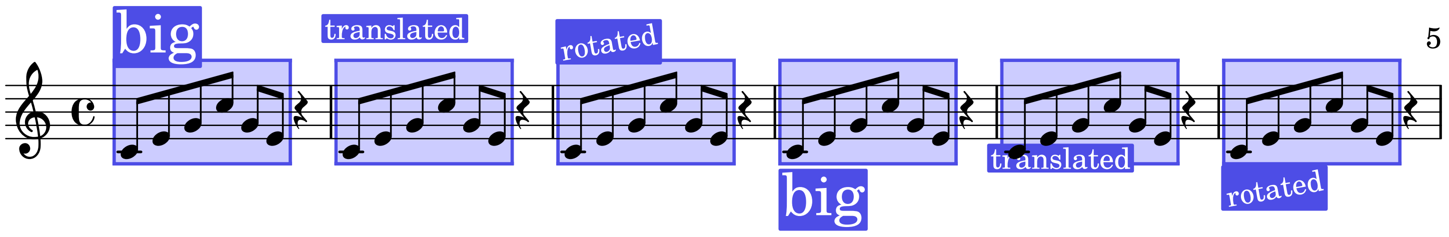
É E e g - Pa pa ge no





This behaviour can lead to unwanted results, e.g. when applying markup commands that change the size or position of the text. It can be turned off by setting the `caption-keep-y` property to `#t`. Then the `\translate` markup command can be used to manually move the caption:





set-top-edge (##f)
set-bottom-edge (##f)
set-left-edge (##f)
set-right-edge (##f)
set-caption-extent (##f)

Frames don't have a horizontal or vertical extent that is "visible" to LilyPond. That means, they are not taken into account by LilyPond's spacing calculations and thus don't consume space of their own. Usually this behavior is preferable because other notation elements are not influenced by the frame's presence.

However, if LilyPond is used to create automatically-cropped images (e.g. via *lilypond-book-preamble.ly*), the frame's borders might be outside the "visible" area and therefore would be missing in the resulting image.

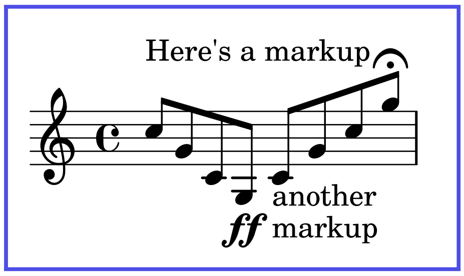
Each of the frame's edges can be made "visible" to LilyPond by adding a hidden null-dimension markup. This behavior can be turned on independently by setting one or more of the properties `set-top-edge`, `set-bottom-edge`, `set-left-edge` and `set-right-edge` to `#t`.

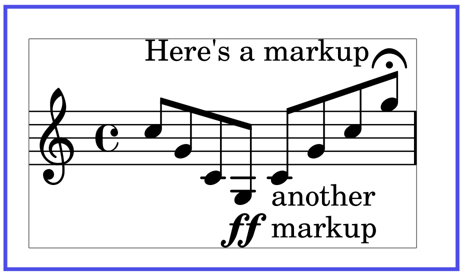
In a similar way, the extent of the caption markup is ignored by default. It can be made "visible" by setting the `set-caption-extent` property to `#t`.

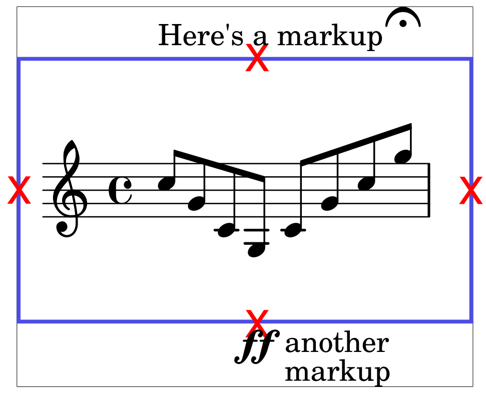
In this example, the entire bar is decorated with a frame that's big enough to enclose all notation elements.

The thin black box shows the "visible" extent of the score. All the frame's borders are outside this area. With automatic cropping, they would disappear.

Here the frame's edges are made "visible" by turning the corresponding properties to `#t`. The red cross marks indicate the position of the invisible null-dimension markups.



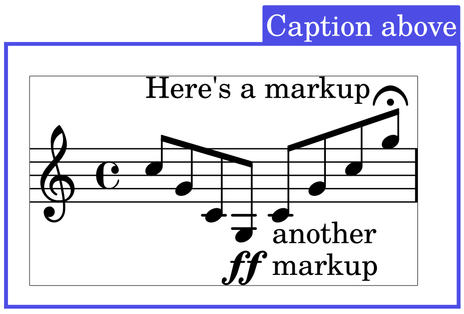




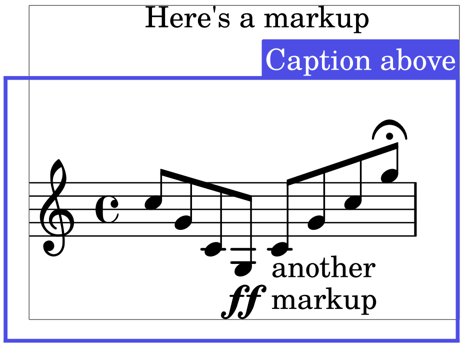
As always, there can be unwanted side effects:

As soon as frames have a "visible" extent, they can push other notation elements outside themselves. The same thing can happen if the caption's extent is no more ignored:

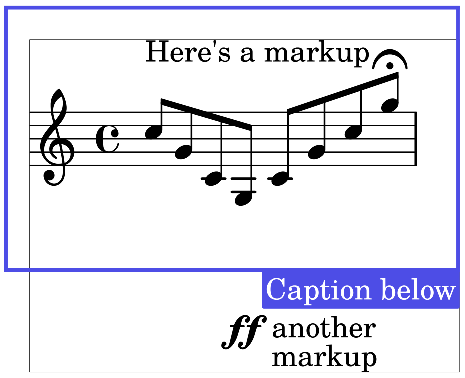
`set-caption-extent` is `#f` (default): Caption has no extent



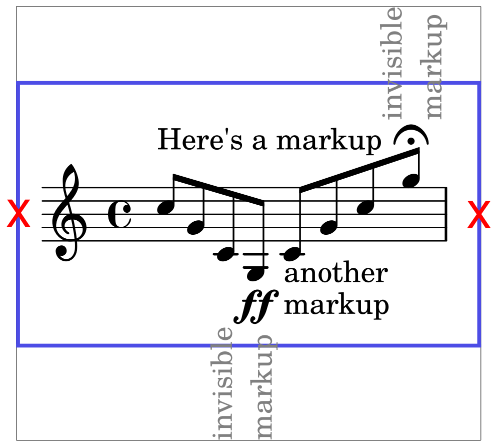
`set-caption-extent` set to `#t`



`set-caption-extent` set to `#t`



In most cases it can be recommended only to set `set-left-edge` and `set-right-edge` to `#t`. Additional extent above or below the staff can be achieved by adding some invisible markup, e.g. with the `\markup \transparent` command.


invisible
markup
Here's a markup
X
X
another
ff markup
invisible
markup