The xfrac package
Split-level fractions

The LaTeX Project*

Released 2024-06-04

The xfrac package defines a document command `\sfrac` with the following syntax:

`\sfrac[`⟨*instance*⟩`]{`⟨*num*⟩`}[`⟨*sep*⟩`]{`⟨*denom*⟩`}`

Let's show a few examples:

```
\sfrac{1}{2},  $\sfrac{1}{2}$,
$\mathbf{3\times\sfrac{1}{2}}$
\quad \fontfamily{ppl}\selectfont Palatino: \sfrac{1}{2}
\quad \fontfamily{ptm}\selectfont Times: \sfrac{1}{2}
```

½, ½, $\mathbf{3\times\sfrac{1}{2}}$ Palatino: ½ Times: ½

You'll notice something interesting: not only does the `\sfrac` command work as it should in math mode, it also gets the job done for other fonts as well.

1 A Bit of History

1.1 The Past

One of the first exercises in *The TEXBook* is to design a macro for split level fractions. The solution presented is fairly simple, using a *virgule* (a slash) for separating the two components. It looks okay because the text font and math font of Computer Modern look almost identical.

The proper symbol to use instead of the virgule is a *solidus* which does not exist in Computer Modern. It is however available in the European Computer Modern fonts, but I'll get back to that.

1.2 The Present

The most common way to produce split level fractions within LaTeX 2ε is by means of the nicefrac package. Part of the reason it has found widespread use is due to the strange design of the built-in text fractions of the EC fonts, which look like this: ½. The package is very simple to use but there are a few issues:

- It uses the virgule instead of the solidus.

*E-mail: latex-team@latex-project.org

- Font size of numerator and denominator is bigger than in the built-in symbol. Compare Palatino: 1/2 *vs.* ½.
- It doesn't correct for fonts using text figures such as in the eco package. Compare 1/2 and 8/9.
- In math mode, it doesn't always pick up the correct math alphabet.

In short: nicefrac doesn't attempt to be the answer to everything and so this is not a criticism of the package. It works quite well for Computer Modern which was pretty much what was widely available at the time it was developed. Users these days, however, have a choice of many fonts when they write their documents.

1.3 The Future

Fonts are wildly different; one macro that works fine for Computer Modern obviously doesn't work well at all in Palatino. For one we have to make the separator symbol configurable, and we need to take care of several details as well: font scaling of the numerator/denominator pair (ND), font selection of ND, *etc.* If we are to have a single package for this in the future we have to define a totally generic interface for the fraction commands and then adjust parameters depending on the current font. What you see in this prototype implementation of xfrac is just that.

2 Advanced User Interface

2.1 Text mode

The usual problem in text mode has a name: Computer Modern. The solidi of all the Computer Modern fonts leave a lot to be desired, although things are potentially looking better as the Latin Modern fonts are becoming more stable and widespread. As long as the default fonts are Computer Modern variants we must however work around this. One idea that comes to mind is to see what happens when you use a solidus from another font instead. Let's try with Times:

> "You take ½ cup of sugar, . . . "

That looks quite good actually, so it was probably very difficult to obtain that result. Nope, it was extremely easy—if you happen to know about *instances*:

```
\DeclareInstance{xfrac}{cmr}{text}
  {slash-symbol-font = ptm}
```

So we define an instance with the name `cmr` from the template `text` which in turn is of object type `xfrac`. You'll notice the `cmr` is also the name of the font family for Computer Modern Roman and the reasoning behind is that every font family should have it's own settings, and if a document command is to work well in that scheme, letting it use the name of the current font family seems like a good idea. Thus the `\sfrac` command checks to see whether an instance with same name as the current font family exists and uses it if the test is true; otherwise the default setting is used. Here we defined the instance to be used for the font family `cmr` and just told it to use the Times font for typesetting the slash symbol which turns out to be a solidus by default.

The option `cm-recommended` which is loaded by default uses the Times solidus for Computer Modern Roman and Computer Modern Sans Serif and the Palatino solidus for Computer Modern Typewriter Type. This looks quite good. Should you however not want this you can use the option `cm-standard` which produces somewhat acceptable results using Computer Modern exclusively.

So what about old style figures? If you use the eco package you might define an instance similar to this ('cmor' is the name of the roman font activated by eco):

```
\DeclareInstance{xfrac}{cmor}{text}
  {
    slash-symbol-font = ptm,
    numerator-font    = cmr,
    denominator-font  = cmr
  }
```

We also use regular Computer Modern Roman for typesetting ND, so we end up with ½ and ⁸⁄₉ instead of ¹⁄₂ and ⁸⁄₉. Much better.

There are also situations where other tricks are useful. If you don't have the inferior and superior figures available in a font, or the font doesn't have a wider design for small font sizes, you can cheat by manually scaling the ND-pair. I got nice results for Adobe's Stempel Garamond (with small caps and old style figures) with the following setup:

```
\DeclareInstance{xfrac}{pegj}{text}
  {
    numerator-font   = pegx,
    denominator-font = pegx,
    scale-factor     = 0.9,
    h-scale          = 1.1
 }
```

We use the font family `pegx` (Stempel Garamond with real small caps) for typesetting the ND-pair. Additionally the key **scale-factor** specifies that the font size used for the ND-pair should be 0.9 of the height of the solidus, and the key **h-scale** specifies that the ND-pair should be scaled an extra 10 % horizontally.

Should you be so fortunate the have a font with inferior and superior figures like in the Monotype Janson example from Philipp Lehman's excellent *The Font Installation Guide*. In that example Philipp defines the font families `mjn0` for the inferior figures and `mjn1` for the superior. Thus to get the `\sfrac` command to use them on the fly for the font family `mjnj` (Janson, old style figures) we would say

```
\DeclareInstance{xfrac}{mjnj}{text}
  {
    numerator-font      = mjn1,
    denominator-font    = mjn0,
    scaling             = false,
    numerator-bot-sep   = 0 pt,
    denominator-bot-sep = 0 pt
  }
```

I think this example is a very clean way to do it. An alternative approach could be to use the keys **numerator-format** and **denominator-format** to process the arguments and let them determine what to do.

2.2 Math Mode

In math mode the choices are a lot fewer because first of all TEX comes with a built-in limitation of 16 math families. Additionally we will not need a solidus for typesetting split fractions in math, as tradition is to use a virgule instead. We define the basic `mathdefault` instance to simply use the math family in use when the instance is run. So if we're in normal math like `$\sfrac{7}{9}$` we simply get family −1. If we're inside a `\mathbf` we're in family 4 (in the standard setup at least), and so the fraction is typeset with the same math family. Simple, isn't?

You can also declare instances for the math families, but I really don't see why you would. If you do then name them according to the scheme `mathfam`⟨*N*⟩, where ⟨*N*⟩ is the family number, and only do it if you *really* know how to set up math fonts. That is, if `\DeclareMathAlphabet` is unbeknownst to you, then just don't go there.

3 The Template Interface

3.1 The object type 'xfrac'

Arg: 1 The numerator

Arg: 2 The separator

Arg: 3 The denominator

Semantics:

Typesets arguments 1 and 3 separated by argument 2, which in text mode by default is a *solidus*. This is taken from `textcomp` where it is denoted `\textfractionsolidus`. This is the character used for the ready made split level fractions such as ½—except in the (European) Computer Modern fonts. In math mode a *virgule* is used instead as this is more appropriate and it is always available in the math fonts. The solidus is a text symbol only.

3.2 The template 'text' (object type xfrac)

Attributes:

numerator-font (tokenlist) Font family specification to use for the numerator.
Default: `\f@family`

numerator-format (function 1 arg) Action to be taken on the numerator.
Default: Process argument unchanged

slash-symbol (tokenlist) The separator symbol. If not specified the default value will be used instead. Default: Solidus (`\textfractionsolidus`)

slash-symbol-font (tokenlist) Font family specification to use for the separator symbol. Default: `\f@family`

slash-symbol-format (function 1 arg) Action to be taken on the separator symbol.
Default: Process argument unchanged

denominator-font (tokenlist) Font family specification to use for the denominator. Default: `\f@family`

denominator-format (function 1 arg) Action to be taken on the denominator. Default: Process argument unchanged

h-scale (real) Factor by which the numerator and denominator should be horizontally scaled. It should only be used if the real superior and inferior fonts are not available. For instance Stempel Garamond looks excellent if scaled 10 % extra horizontally, *i.e.*, by a factor of 1.1. Default: 1

v-scale (real) Same as **h-scale** only vertically. Probably not of much use but added for completeness. Default: 1

scale-factor (real) Fraction of the size of **slash-symbol**. Used for setting the font size of numerator and denominator. Usually a value of app. 5⁄6 produces fine results. It should only be used if the real superior and inferior fonts are not available. As an example Stempel Garamond looks better if the factor is 0.9. Default: 0.83333

scale-relative (choice) If set to `true` the font size of the numerator and denominator is scaled with respect to the height of the **slash-symbol**. If set to `false` the font is scaled with respect to the total height of the **slash-symbol**. Default: true

scaling (choice) If set to `true` the fonts are allowed to scale. If set to `false` they are not. See the 'Janson' example for an application. Default: true

numerator-top-sep (length) Dimension specifying the space between the top of the **slash-symbol** and the top of the numerator. If not specified, the depth of the solidus will be used, because this value will make the fraction look even. Default: Unspecified

numerator-bot-sep (length) Dimension specifying the lift of the numerator from the baseline. Default: Unspecified

denominator-bot-sep (length) Dimension specifying the lift of the denominator from the baseline. Default: Unspecified

slash-right-kern (length) Dimension specifying the kerning between the **slash-symbol** and the numerator. Default: `0pt`

slash-left-kern (length) Dimension specifying the kerning between the **slash-symbol** and the denominator. Default: `0pt`

math-mode (choice) Are we in math mode or not? Default: false

phantom (tokenlist) A character that suits the common cases. As we would mostly want to use numbers in text mode we choose a "tall" number, while in math it is somewhat different. Default: 8

Semantics & Comments:

This template is also the foundation for the "math" template. The keys **slash-right-mkern** and **slash-left-mkern** can only be used in math mode and are not shown here.

3.3 The template 'math' (object type xfrac)

Attributes:

numerator-font (tokenlist) Font family specification to use for the numerator. Default: `\number\fam`

slash-symbol (tokenlist) The separator symbol. If not specified the default value will be used instead. Default: Virgule (/)

slash-symbol-font (tokenlist) Font family specification to use for the separator symbol. Default: `\number\fam`

denominator-font (tokenlist) Font family specification to use for the denominator. Default: `\number\fam`

scale-factor (real) Fraction of the size of **slash-symbol**. In math mode we cannot rely on the fonts to be able to scale, but giving a default scale of 0.7 fits into the regular size changing scheme—the default scheme has values $(D, T, S, SS) = (1, 1, 0.7, 0.5)$ whereas we with a default **scale-factor** of 0.7 get $(1, 1, 0.7, 0.49)$. That's close enough. Default: 0.7

scale-relative (choice) If set to `true` the font size of the numerator and denominator is scaled with respect to the height of the **slash-symbol**. If set to `false` the font is scaled with respect to the total height of the **slash-symbol**. Default: false

scaling (choice) If set to `true` the fonts are allowed to scale. If set to `false` they are not. See the `plainmath` example for an application. Default: true

numerator-top-sep (length) Dimension specifying the space between the top of the **slash-symbol** and the top of the numerator. If not specified, the depth of the virgule will be used, because this value will make the fraction look even.Default: `0pt`

denominator-bot-sep (length) Dimension specifying the lift of the denominator from the baseline. Default: `0pt`

slash-right-mkern (muskip) Same as **slash-right-kern** but for math mode only and should be specified in `mu` units. Default: `-2mu`

slash-left-mkern (muskip) Same as **slash-left-kern** but for math mode only and should be specified in `mu` units. Default: `-1mu`

math-mode (choice) Are we in math mode or not? Default: true

phantom (tokenlist) A character that suits the common cases. In math we have a high risk of using a parenthesis, so we choose that. Text mode is another story. Default: (

Semantics & Comments:

This template is a restricted version of the `text` template. Only the keys that are different from the `text` template are shown here. Also bear in mind that the attributes **slash-left-kern** and **slash-right-kern** have no meaning in this template.

4 xfrac implementation

```
⟨*package⟩
⟨@@=xfrac⟩
```

Load expl3 "up-front" only if required.

```
\@ifundefined{ExplLoaderFileDate}
  {\RequirePackage{expl3}}
  {}
```

Make sure that the version of l3kernel in use is sufficiently new. We use `\ExplFileDate` as `\@ifpackagelater` doesn't work for pre-loaded expl3 in the absence of the package.

```
\@ifl@t@r\ExplLoaderFileDate{2018-02-21}
  {}
  {%
    \PackageError{xfrac}{Support package expl3 too old}
      {%
        You need to update your installation of 'l3kernel'.\MessageBreak
        Loading~xfrac~will~abort!%
      }%
    \endinput
  }%
\ProvidesExplPackage{xfrac}{2024-06-04}{}
  {Split-level fractions}
```

`\IfFormatAtLeastTF` Not present in older kernels: use the LaTeX 2_ε mechanism as this is correct for this case.

```
\providecommand \IfFormatAtLeastTF { \@ifl@t@r \fmtversion }
```

(*End of definition for* `\IfFormatAtLeastTF`*. This function is documented on page* **??**.)

```
\IfFormatAtLeastTF { 2020-02-02 }
  { }
  { \RequirePackage { textcomp } }
\IfFormatAtLeastTF { 2020-10-01 }
  { }
  { \RequirePackage { xparse } }
```

`\l__xfrac_cm_std_bool` There is one option to support.

```
\keys_define:nn { xfrac }
  {
    cm-recommended .choice:,
    cm-recommended /
      false        .code:n     =
        { \bool_set_true:N \l__xfrac_cm_std_bool },
    cm-recommended /
      true         .code:n     =
        { \bool_set_false:N \l__xfrac_cm_std_bool },
    cm-recommended .default:n  = { true },
    cm-standard    .bool_set:N = \l__xfrac_cm_std_bool
  }
```

(*End of definition for* `\l__xfrac_cm_std_bool`.)

```
\IfFormatAtLeastTF { 2022-06-01 }
  { \ProcessKeyOptions }
```

```
  {
    \RequirePackage { l3keys2e }
    \ProcessKeysOptions { xfrac }
  }
\IfFormatAtLeastTF { 2024-06-01 }
  { }
  { \RequirePackage { xtemplate } }
\cs_if_exist:NF \IfInstanceExistsTF
  { \cs_new_eq:NN \IfInstanceExistsTF \IfInstanceExistTF }
```

\l__xfrac_slash_box
\l__xfrac_tmp_box

In keeping with the LaTeX3 philosophy, rather than use generic scratch boxes and get confused, xfrac reserves its own named working space.

```
\box_new:N \l__xfrac_slash_box
\box_new:N \l__xfrac_tmp_box
```

(*End of definition for* `\l__xfrac_slash_box` *and* `\l__xfrac_tmp_box`.)

__xfrac_tmp:w

Used for the raised boxes: weird as it does not take an argument but the `\raisebox` does.

```
\cs_new:Npn \__xfrac_tmp:w { }
```

(*End of definition for* `\__xfrac_tmp:w`.)

4.1 Initialisation of variables

Variables used in templates have to be set up: there is not much to say about these, other than that they must exist.

\l__xfrac_denominator_bot_sep_dim
\l__xfrac_numerator_bot_sep_dim
\l__xfrac_numerator_top_sep_dim
\l__xfrac_slash_left_sep_dim
\l__xfrac_slash_right_sep_dim

Fixed lengths.

```
\dim_new:N \l__xfrac_denominator_bot_sep_dim
\dim_new:N \l__xfrac_numerator_bot_sep_dim
\dim_new:N \l__xfrac_numerator_top_sep_dim
\dim_new:N \l__xfrac_slash_left_sep_dim
\dim_new:N \l__xfrac_slash_right_sep_dim
```

(*End of definition for* `\l__xfrac_denominator_bot_sep_dim` *and others.*)

\l__xfrac_slash_left_muskip
\l__xfrac_slash_right_muskip

Math mode skips.

```
\muskip_new:N \l__xfrac_slash_left_muskip
\muskip_new:N \l__xfrac_slash_right_muskip
```

(*End of definition for* `\l__xfrac_slash_left_muskip` *and* `\l__xfrac_slash_right_muskip`.)

\l__xfrac_hscale_fp
\l__xfrac_scale_factor_fp
\l__xfrac_vscale_fp

Floating point values.

```
\fp_new:N \l__xfrac_hscale_fp
\fp_new:N \l__xfrac_scale_factor_fp
\fp_new:N \l__xfrac_vscale_fp
```

(*End of definition for* `\l__xfrac_hscale_fp`, `\l__xfrac_scale_factor_fp`, *and* `\l__xfrac_vscale_fp`.)

\l__xfrac_denominator_font_tl
\l__xfrac_numerator_font_tl
\l__xfrac_phantom_tl
\l__xfrac_slash_symbol_tl
\l__xfrac_slash_symbol_font_tl

Token lists, which include floating-point numbers and math(s) skips.

```
\tl_new:N \l__xfrac_denominator_font_tl
\tl_new:N \l__xfrac_numerator_font_tl
\tl_new:N \l__xfrac_phantom_tl
\tl_new:N \l__xfrac_slash_symbol_tl
\tl_new:N \l__xfrac_slash_symbol_font_tl
```

(*End of definition for* `\l__xfrac_denominator_font_tl` *and others.*)

`\__xfrac_fontscale:`
`\__xfrac_math:n`
`\__xfrac_denominator_font_change:`
`\__xfrac_denominator_format:n`
`\__xfrac_numerator_font_change:`
`\__xfrac_numerator_format:n`
`\__xfrac_relscale:`
`\__xfrac_slash_symbol_font_change:`
`\__xfrac_slash_symbol_format:n`
`\__xfrac_text_or_math:n`

Functions, either things which are calculated "on the fly" (no argument required) or are functions taking one argument in the code.

```
\cs_new:Npn \__xfrac_fontscale: { }
\cs_new:Npn \__xfrac_math:n #1 { }
\cs_new:Npn \__xfrac_denominator_font_change: { }
\cs_new:Npn \__xfrac_denominator_format:n #1 { }
\cs_new:Npn \__xfrac_numerator_font_change: { }
\cs_new:Npn \__xfrac_numerator_format:n #1 { }
\cs_new:Npn \__xfrac_relscale: { }
\cs_new:Npn \__xfrac_slash_symbol_font_change: { }
\cs_new:Npn \__xfrac_slash_symbol_format:n #1 { }
\cs_new:Npn \__xfrac_text_or_math:n #1 { }
```

(*End of definition for* `\__xfrac_fontscale:` *and others.*)

4.2 The template

There is only one object type in xfrac, rather unimaginatively named `xfrac`.

```
\IfFormatAtLeastTF { 2024-06-01 }
  { \NewTemplateType }
  { \DeclareObjectType }
    { xfrac } { 3 }
```

A single template interface is used for both text and math(s), which does make a few things a little complex later.

```
\DeclareTemplateInterface { xfrac } { text } { 3 }
  {
    denominator-bot-sep : length     = \c_max_dim          ,
    denominator-font    : tokenlist  = \f@family           ,
    denominator-format  : function 1 = #1                  ,
    h-scale             : real       = 1                   ,
    math-mode           : choice { false , true }
                                     = false               ,
    numerator-font      : tokenlist  = \f@family           ,
    numerator-format    : function 1 = #1                  ,
    numerator-bot-sep   : length     = \c_max_dim          ,
    numerator-top-sep   : length     = \c_max_dim          ,
    phantom             : tokenlist  = 8                   ,
    scale-factor        : real       = 0.83333             ,
    scale-relative      : choice { false , true }
                                     = true                ,
    scaling             : choice { false , true }
                                     = true                ,
    slash-left-kern     : length     = 0 pt                ,
    slash-left-mkern    : muskip     = -2 mu               ,
    slash-right-kern    : length     = 0 pt                ,
    slash-right-mkern   : muskip     = -1 mu               ,
    slash-symbol        : tokenlist  = \textfractionsolidus ,
    slash-symbol-font   : tokenlist  = \f@family           ,
    slash-symbol-format : function 1 = #1                  ,
    v-scale             : real       = 1                   ,
  }
```

Most of the variable binding is quite simple: of course, the choices are a little more complicated. That is particularly true where these have to set up "on the fly" functions.

```
\DeclareTemplateCode { xfrac } { text } { 3 }
  {
    denominator-bot-sep = \l__xfrac_denominator_bot_sep_dim ,
    denominator-font    = \l__xfrac_denominator_font_tl     ,
    denominator-format  = \__xfrac_denominator_format:n      ,
    h-scale             = \l__xfrac_hscale_fp               ,
    math-mode           =
      {
        false = \cs_set_eq:NN \__xfrac_math:n \use:n,
        true  = \cs_set_eq:NN \__xfrac_math:n \ensuremath
      },
    numerator-font      = \l__xfrac_numerator_font_tl       ,
    numerator-format    = \__xfrac_numerator_format:n        ,
    numerator-bot-sep   = \l__xfrac_numerator_bot_sep_dim   ,
    numerator-top-sep   = \l__xfrac_numerator_top_sep_dim   ,
    phantom             = \l__xfrac_phantom_tl              ,
    scale-factor        = \l__xfrac_scale_factor_fp         ,
    scale-relative      =
      {
        false =
          \cs_set:Npn \__xfrac_relscale:
            {
              \dim_eval:n
                { \box_ht:N \l__xfrac_tmp_box + \box_dp:N \l__xfrac_tmp_box }
            },
        true  =
          \cs_set:Npn \__xfrac_relscale:
            { \dim_eval:n { \box_ht:N \l__xfrac_slash_box } }
      },
    scaling             =
      {
        false = \cs_set_eq:NN \__xfrac_fontscale: \prg_do_nothing:,
        true  =
          \cs_set:Npn \__xfrac_fontscale:
            {
              \fontsize
                { \fp_to_dim:n { \l__xfrac_scale_factor_fp * \__xfrac_relscale: } }
                { \c_zero_dim }
              \selectfont
            }
      },
    slash-left-kern     = \l__xfrac_slash_left_sep_dim      ,
    slash-left-mkern    = \l__xfrac_slash_left_muskip       ,
    slash-right-kern    = \l__xfrac_slash_right_sep_dim     ,
    slash-right-mkern   = \l__xfrac_slash_right_muskip      ,
    slash-symbol        = \l__xfrac_slash_symbol_tl         ,
    slash-symbol-font   = \l__xfrac_slash_symbol_font_tl    ,
    slash-symbol-format = \__xfrac_slash_symbol_format:n     ,
    v-scale             = \l__xfrac_vscale_fp
  }
```

The implementation part starts with applying all of the settings from above. The first

part of the set up is then to determine whether the surroundings are text or math(s), and react accordingly.

```
{
  \mode_if_math:TF
    {
      \cs_set_eq:NN \__xfrac_text_or_math:n \text
      \cs_set:Npe \__xfrac_denominator_font_change:
        { \tex_fam:D \l__xfrac_denominator_font_tl }
      \cs_set:Npe \__xfrac_numerator_font_change:
        { \tex_fam:D \l__xfrac_numerator_font_tl }
      \cs_set:Npe \__xfrac_slash_symbol_font_change:
        { \tex_fam:D \l__xfrac_slash_symbol_font_tl }
    }
    {
      \cs_set_eq:NN \__xfrac_text_or_math:n \mbox
      \cs_set:Npn \__xfrac_denominator_font_change:
        {
          \fontfamily { \l__xfrac_denominator_font_tl }
          \selectfont
        }
      \cs_set:Npn \__xfrac_numerator_font_change:
        {
          \fontfamily { \l__xfrac_numerator_font_tl }
          \selectfont
        }
      \cs_set:Npn \__xfrac_slash_symbol_font_change:
        {
          \fontfamily { \l__xfrac_slash_symbol_font_tl }
          \selectfont
        }
    }
```

Everything is now either inside `\text` or an `\mbox`, depending upon the surroundings. First, there are some boxes to set up.

```
  \__xfrac_text_or_math:n
    {
      \m@th
      \hbox_set:Nn \l__xfrac_tmp_box
        { \__xfrac_math:n { \vphantom { ( ) } } }
      \hbox_set:Nn \l__xfrac_slash_box
        {
          \__xfrac_math:n
            {
              \__xfrac_slash_symbol_format:n
                {
                  \__xfrac_math:n
                    {
                      \__xfrac_slash_symbol_font_change:
                      \IfNoValueTF {#2}
                        { \l__xfrac_slash_symbol_tl } {#2}
                    }
                }
            }
        }
```

Check on the numerator separator dimensions. The code starts with the assumption that neither has been given, as this can then be used to set up a default, which is also used when both values are set erroneously.

```
\cs_set:Npn \__xfrac_tmp:w
  {
    \raisebox
      {
        \dim_eval:n
          {
            \box_ht:N \l__xfrac_slash_box
            - \box_dp:N \l__xfrac_slash_box
            - \height
          }
      }
  }
\dim_compare:nNnTF
  { \l__xfrac_numerator_top_sep_dim } = { \c_max_dim }
  {
    \dim_compare:nNnF
      { \l__xfrac_numerator_bot_sep_dim } = { \c_max_dim }
      {
        \cs_set:Npn \__xfrac_tmp:w
          {
            \raisebox
              { \dim_use:N \l__xfrac_numerator_bot_sep_dim }
          }
      }
  }
  {
    \dim_compare:nNnTF
      { \l__xfrac_numerator_bot_sep_dim } = { \c_max_dim }
        {
          \cs_set:Npn \__xfrac_tmp:w
            {
              \raisebox
                {
                  \dim_eval:n
                    {
                      \box_ht:N \l__xfrac_slash_box
                      - \dim_use:N \l__xfrac_numerator_top_sep_dim
                      - \height
                    }
                }
            }
        }
        {
          \msg_error:nn { xfrac }
            { over-specified-numerator-sep }
        }
  }
```

Typeset the numerator.

```
\__xfrac_tmp:w
  {
```

```
      \__xfrac_fontscale:
      \__xfrac_numerator_format:n
        {
          \scalebox
            { \fp_use:N \l__xfrac_hscale_fp }
            [ \fp_use:N \l__xfrac_vscale_fp ]
            {
              \__xfrac_math:n
                {
                  \__xfrac_numerator_font_change:
                  {
                    \vphantom { \l__xfrac_phantom_tl }
                    #1
                  }
                }
            }
        }
    }
  \__xfrac_math:n
    { % THIS IS JUST WRONG!
      \mode_if_math:TF
        { \tex_mskip:D \l__xfrac_slash_right_muskip }
        { \tex_hskip:D \l__xfrac_slash_right_sep_dim }
    }
```

Typeset the separator.

```
  \box_use:N \l__xfrac_slash_box
  \__xfrac_math:n
    {
      \mode_if_math:TF
        { \tex_mskip:D \l__xfrac_slash_left_muskip }
        { \tex_hskip:D \l__xfrac_slash_left_sep_dim }
    }
```

Typeset the denominator.

```
  \dim_compare:nNnTF
    { \l__xfrac_denominator_bot_sep_dim } = { \c_max_dim }
    {
      \cs_set:Npn \__xfrac_tmp:w
        { \raisebox { - \box_dp:N \l__xfrac_slash_box } }
    }
    {
      \cs_set:Npn \__xfrac_tmp:w
        {
          \raisebox
            { \dim_use:N \l__xfrac_denominator_bot_sep_dim }
        }
    }
  \__xfrac_tmp:w
    {
      \__xfrac_fontscale:
      \__xfrac_denominator_format:n
        {
          \scalebox
            { \fp_use:N \l__xfrac_hscale_fp }
```

```
                  [ \fp_use:N \l__xfrac_vscale_fp ]
                  {
                    \__xfrac_math:n
                      {
                        \__xfrac_denominator_font_change:
                        {
                          \vphantom { \l__xfrac_phantom_tl }
                          #3
                        }
                      }
                  }
              }
          }
      }
  }
```

Since math(s) and text mode are wildly different entities we define a separate template for each. You already saw the "text" template, and here is the "math" template.

```
\IfFormatAtLeastTF { 2024-06-01 }
  {
    \DeclareTemplateCopy { xfrac } { math } { text }
    \EditTemplateDefaults { xfrac } { math }
  }
  { \DeclareRestrictedTemplate { xfrac } { text } { math } }
    {
      numerator-font      = \number \fam ,
      slash-symbol        = /            ,
      slash-symbol-font   = \number \fam ,
      denominator-font    = \number \fam ,
      scale-factor        = 0.7          ,
      scale-relative      = false        ,
      scaling             = true         ,
      denominator-bot-sep = 0 pt         ,
      math-mode           = true         ,
      phantom             = ( % )
    }
```

4.3 The standard instances

For the default instances we just use the relevant templates with the default settings.

The default "text" instance.

```
\DeclareInstance { xfrac } { default } { text } { }
```

The default "math(s)" instance. We annot set the `numerator-top-sep` in the restricted template above.

```
\DeclareInstance { xfrac } { mathdefault } { math }
  { numerator-top-sep = 0pt }
```

Default Computer Modern setup. Far from optimal, but better than nothing.

```
\DeclareInstance { xfrac } { cmr } { text }
  {
    denominator-bot-sep = 0 pt    ,
    numerator-top-sep   = 0.2 ex  ,
    slash-left-kern     = -0.1 em ,
```

```
    slash-right-kern    = -0.1 em
  }
\DeclareInstance { xfrac } { cmss } { text }
  {
    denominator-bot-sep = 0 pt    ,
    numerator-top-sep   = 0.2 ex  ,
    slash-left-kern     = -0.1 em ,
    slash-right-kern    = -0.1 em
  }
\DeclareInstance { xfrac } { cmtt } { text }
  {
    denominator-bot-sep = 0 pt    ,
    numerator-top-sep   = 0.2 ex  ,
    slash-left-kern     = -0.1 em ,
    slash-right-kern    = -0.1 em
  }
```

We can do better for the Computer Modern fonts. For cmr and cmss we choose Times, and for cmtt use Palatino.

```
\bool_if:NF \l__xfrac_cm_std_bool
  {
    \DeclareInstance { xfrac } { cmr } { text }
      { slash-symbol-font = ptm }
    \DeclareInstance { xfrac } { cmss } { text }
      { slash-symbol-font = ptm }
    \DeclareInstance { xfrac } { cmtt } { text }
      { slash-symbol-font = ppl }
  }
```

Things works slightly better with Latin Modern.

```
\DeclareInstance { xfrac } { lmr } { text }
  {
    denominator-bot-sep = 0 pt     ,
    numerator-top-sep   = 0.1 ex   ,
    slash-left-kern     = -0.15 em ,
    slash-right-kern    = -0.15 em
  }
\DeclareInstance { xfrac } { lmss } { text }
  {
    denominator-bot-sep = 0 pt     ,
    numerator-top-sep   = 0 pt     ,
    slash-left-kern     = -0.15 em ,
    slash-right-kern    = -0.15 em
  }
\DeclareInstance { xfrac } { lmtt } { text }
  {
    denominator-bot-sep = 0 pt     ,
    numerator-top-sep   = 0 pt     ,
    slash-left-kern     = -0.15 em ,
    slash-right-kern    = -0.15 em
  }
```

4.4 The user command

\sfrac Currently there is just a single user command. \sfrac takes two mandatory arguments: numerator and denominator. It can take an optional argument between the mandatory specifying the separator like this:

```
\sfrac{7}[/]{12}
```

It also has an optional argument that comes before the first mandatory argument. If used it will use that instance instead of the auto-detected one, so a user who has defined the instance "cmr2" may use

```
\sfrac[cmr2]{7}{12}
```

and get the settings from "cmr2" instead of the settings of the current font family.

```
\NewDocumentCommand \sfrac { o m o m }
 {
   \mode_if_math:TF
     {
       \IfInstanceExistsTF { xfrac } { mathfam \number \fam }
         { \UseInstance { xfrac } { mathfam \number \fam } }
         { \UseInstance { xfrac } { mathdefault } }
       {#2} {#3} {#4}
     }
     {
       \IfInstanceExistsTF { xfrac } {#1}
         { \UseInstance { xfrac } {#1} }
         {
           \IfInstanceExistsTF { xfrac } { \f@family }
             { \UseInstance { xfrac } { \f@family } }
             { \UseInstance { xfrac } { default } }
         }
       {#2} {#3} {#4}
     }
 }
```

(*End of definition for* \sfrac. *This function is documented on page* ??.)

4.5 Messages

Just the one.

```
\msg_new:nnnn { xfrac } { over-specified-numerator-sep }
  { You~have~specified~both~"numerator-top-sep"~and~"numerator-bot-sep". }
  { I~will~pretend~that~you~didn't~specify~either~of~them. }
```

⟨/package⟩

Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

U

V

X