

# The `xfrac` package

## Split-level fractions\*

The L<sup>A</sup>T<sub>E</sub>X3 Project<sup>†</sup>

Released 2012/11/09

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}
```

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

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 T<sub>E</sub>X Book* 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.

---

\*This file describes v4299, last revised 2012/11/09.

<sup>†</sup>E-mail: [latex-team@latex-project.org](mailto:latex-team@latex-project.org)

## 1.2 The Present

The most common way to produce split level fractions within  $\text{\LaTeX} 2_{\epsilon}$  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:  $\frac{1}{2}$ . The package is very simple to use but there are a few issues:

- It uses the virgule instead of the solidus.
- Font size of numerator and denominator is bigger than in the built-in symbol. Compare Palatino:  $\frac{1}{2}$  *vs.*  $\frac{1}{2}$ .
- It doesn't correct for fonts using text figures such as in the `eco` package. Compare  $\frac{1}{2}$  and  $\frac{8}{9}$ .
- 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  $\frac{1}{2}$  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  $\frac{1}{2}$  and  $\frac{8}{9}$  instead of  $\frac{1}{2}$  and  $\frac{8}{9}$ . 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 to 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  $\text{\TeX}$  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.

Another example: If we want `\sfrac` to produce split fractions without doing anything at all, we can choose the collection `plainmath`, which is defined as

```
\DeclareCollectionInstance{plainmath}{xfrac}{mathdefault}{math}
{
  denominator-bot-sep = 0 pt,
  numerator-bot-sep    = 0 pt,
  numerator-top-sep    = \c_max_dim,
  scaling              = false,
  slash-right-mkern    = 0 mu,
  slash-left-mkern     = 0 mu
}
```

This creates an alternative version of the instance `mathdefault` with settings as specified by the keys. In the default math setup **numerator-top-sep** is set to `0 pt`, and here we set **numerator-bot-sep** to `0 pt` as well, so in order to avoid over-specification (and an error message) we must set **numerator-top-sep** to `\c_max_dim`. We activate (obeying normal scoping rules) it with:

`\UseCollection{xfrac}{plainmath}`

Then `$$\sfrac{8}{13}$` produces 8/13 and just typing `$8/13$` gives the same result: 8/13.

## 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.  $\frac{5}{6}$  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:  $-2\mu$
- slash-left-mkern (muskip)** Same as **slash-left-kern** but for math mode only and should be specified in **mu** units.  
Default:  $-1\mu$
- 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

```
1 <*package>
2 <@@=xfrac>
3 \ProvidesExplPackage
4   {\ExplFileName}{\ExplFileDate}{\ExplFileVersion}{\ExplFileDescription}

\l__xfrac_cm_std_bool There is one option to support.
5 \keys_define:nn { xfrac }
6   {
7     cm-recommended .choice:,
8     cm-recommended /
9       false      .code:n      =
10      { \bool_set_true:N \l__xfrac_cm_std_bool },
11     cm-recommended /
12      true      .code:n      =
13      { \bool_set_false:N \l__xfrac_cm_std_bool },
14     cm-recommended .default:n = { true },
15     cm-standard    .bool_set:N = \l__xfrac_cm_std_bool
16   }
17 \ProcessKeysOptions { xfrac }
(End definition for \l__xfrac_cm_std_bool. This variable is documented on page ??.)

\l__xfrac_slash_box In keeping with the LATEX3 philosophy, rather than use generic scratch boxes and get
\l__xfrac_tmp_box   confused, xfrac reserves its own named working space.
18 \box_new:N \l__xfrac_slash_box
19 \box_new:N \l__xfrac_tmp_box
(End definition for \l__xfrac_slash_box. This function is documented on page ??.)

\__xfrac_tmp:w Used for the raised boxes: weird as it does not take an argument but the \raisebox
does.
20 \cs_new:Npn \__xfrac_tmp:w { }
(End 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.

<code>\l_xfrac_denominator_bot_sep_dim</code>	Fixed lengths.
<code>\l_xfrac_numerator_bot_sep_dim</code>	21 <code>\dim_new:N \l_xfrac_denominator_bot_sep_dim</code>
<code>\l_xfrac_numerator_top_sep_dim</code>	22 <code>\dim_new:N \l_xfrac_numerator_bot_sep_dim</code>
<code>\l_xfrac_slash_left_sep_dim</code>	23 <code>\dim_new:N \l_xfrac_numerator_top_sep_dim</code>
<code>\l_xfrac_slash_right_sep_dim</code>	24 <code>\dim_new:N \l_xfrac_slash_left_sep_dim</code>
	25 <code>\dim_new:N \l_xfrac_slash_right_sep_dim</code>
	(End definition for <code>\l_xfrac_denominator_bot_sep_dim</code> . This function is documented on page ??.)
<code>\l_xfrac_slash_left_muskip</code>	Math mode skips.
<code>\l_xfrac_slash_right_muskip</code>	26 <code>\muskip_new:N \l_xfrac_slash_left_muskip</code>
	27 <code>\muskip_new:N \l_xfrac_slash_right_muskip</code>
	(End definition for <code>\l_xfrac_slash_left_muskip</code> . This function is documented on page ??.)
<code>\l_xfrac_hscale_fp</code>	Floating point values.
<code>\l_xfrac_scale_factor_fp</code>	28 <code>\fp_new:N \l_xfrac_hscale_fp</code>
<code>\l_xfrac_vscale_fp</code>	29 <code>\fp_new:N \l_xfrac_scale_factor_fp</code>
	30 <code>\fp_new:N \l_xfrac_vscale_fp</code>
	(End definition for <code>\l_xfrac_hscale_fp</code> . This function is documented on page ??.)
<code>\l_xfrac_denominator_font_tl</code>	Token lists, which include floating-point numbers and math(s) skips.
<code>\l_xfrac_numerator_font_tl</code>	31 <code>\tl_new:N \l_xfrac_denominator_font_tl</code>
<code>\l_xfrac_phantom_tl</code>	32 <code>\tl_new:N \l_xfrac_numerator_font_tl</code>
<code>\l_xfrac_slash_symbol_tl</code>	33 <code>\tl_new:N \l_xfrac_phantom_tl</code>
<code>\l_xfrac_slash_symbol_font_tl</code>	34 <code>\tl_new:N \l_xfrac_slash_symbol_tl</code>
	35 <code>\tl_new:N \l_xfrac_slash_symbol_font_tl</code>
	(End definition for <code>\l_xfrac_denominator_font_tl</code> . This function is documented on page ??.)
<code>\_xfrac_fontscale:</code>	Functions, either things which are calculated “on the fly” (no argument required) or are
<code>\_xfrac_math:n</code>	functions taking one argument in the code.
<code>\_xfrac_denominator_font_change:</code>	36 <code>\cs_new_nopar:Npn \_xfrac_fontscale: { }</code>
<code>\_xfrac_denominator_format:n</code>	37 <code>\cs_new:Npn \_xfrac_math:n #1 { }</code>
<code>\_xfrac_numerator_font_change:</code>	38 <code>\cs_new_nopar:Npn \_xfrac_denominator_font_change: { }</code>
<code>\_xfrac_numerator_format:n</code>	39 <code>\cs_new:Npn \_xfrac_denominator_format:n #1 { }</code>
<code>\_xfrac_relscale:</code>	40 <code>\cs_new_nopar:Npn \_xfrac_numerator_font_change: { }</code>
<code>\_xfrac_slash_symbol_font_change:</code>	41 <code>\cs_new:Npn \_xfrac_numerator_format:n #1 { }</code>
<code>\_xfrac_slash_symbol_format:n</code>	42 <code>\cs_new_nopar:Npn \_xfrac_relscale: { }</code>
<code>\_xfrac_text_or_math:n</code>	43 <code>\cs_new_nopar:Npn \_xfrac_slash_symbol_font_change: { }</code>
	44 <code>\cs_new:Npn \_xfrac_slash_symbol_format:n #1 { }</code>
	45 <code>\cs_new:Npn \_xfrac_text_or_math:n #1 { }</code>
	(End definition for <code>\_xfrac_fontscale:.</code> This function is documented on page ??.)

## 4.2 The template

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

```
46 \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.

```
47 \DeclareTemplateInterface { xfrac } { text } { 3 }
48 {
49   denominator-bot-sep : length      = \c_max_dim          ,
50   denominator-font    : tokenlist  = \f@family           ,
51   denominator-format  : function 1 = #1                  ,
52   h-scale              : real       = 1                   ,
53   math-mode            : choice { false , true }          ,
54                       = false                               ,
55   numerator-font      : tokenlist  = \f@family           ,
56   numerator-format    : function 1 = #1                  ,
57   numerator-bot-sep   : length      = \c_max_dim          ,
58   numerator-top-sep   : length      = \c_max_dim          ,
59   phantom             : tokenlist  = 8                   ,
60   scale-factor        : real       = 0.83333              ,
61   scale-relative      : choice { false , true }          ,
62                       = true                               ,
63   scaling             : choice { false , true }          ,
64                       = true                               ,
65   slash-left-kern     : length      = 0 pt                ,
66   slash-left-mkern    : muskip     = -2 mu               ,
67   slash-right-kern    : length      = 0 pt                ,
68   slash-right-mkern   : muskip     = -1 mu               ,
69   slash-symbol        : tokenlist  = \textfactionsolidus ,
70   slash-symbol-font   : tokenlist  = \f@family           ,
71   slash-symbol-format : function 1 = #1                  ,
72   v-scale             : real       = 1                   ,
73 }
```

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.

```
74 \DeclareTemplateCode { xfrac } { text } { 3 }
75 {
76   denominator-bot-sep = \l__xfrac_denominator_bot_sep_dim ,
77   denominator-font    = \l__xfrac_denominator_font_tl     ,
78   denominator-format  = \__xfrac_denominator_format:n      ,
79   h-scale             = \l__xfrac_hscale_fp                ,
80   math-mode           =
81   {
82     false = \cs_set_eq:NN \__xfrac_math:n \use:n,
83     true  = \cs_set_eq:NN \__xfrac_math:n \ensuremath
84   },
85   numerator-font      = \l__xfrac_numerator_font_tl       ,
86   numerator-format    = \__xfrac_numerator_format:n        ,
```

```

87 numerator-bot-sep = \l__xfrac_numerator_bot_sep_dim ,
88 numerator-top-sep = \l__xfrac_numerator_top_sep_dim ,
89 phantom           = \l__xfrac_phantom_tl           ,
90 scale-factor       = \l__xfrac_scale_factor_fp      ,
91 scale-relative     =
92 {
93   false =
94     \cs_set_nopar:Npn \__xfrac_relscale:
95     {
96       \__dim_eval:w
97       \box_ht:N \l__xfrac_tmp_box + \box_dp:N \l__xfrac_tmp_box
98       \__dim_eval_end:
99     },
100   true =
101     \cs_set_nopar:Npn \__xfrac_relscale:
102     { \box_ht:N \l__xfrac_slash_box }
103 },
104 scaling           =
105 {
106   false = \cs_set_eq:NN \__xfrac_fontscale: \prg_do_nothing:,
107   true =
108     \cs_set_nopar:Npn \__xfrac_fontscale:
109     {
110       \fontsize
111       { \fp_use:N \l__xfrac_scale_factor_fp \__xfrac_relscale: }
112       { \c_zero_dim }
113       \selectfont
114     }
115 },
116 slash-left-kern   = \l__xfrac_slash_left_sep_dim    ,
117 slash-left-mkern   = \l__xfrac_slash_left_muskip     ,
118 slash-right-kern   = \l__xfrac_slash_right_sep_dim   ,
119 slash-right-mkern  = \l__xfrac_slash_right_muskip    ,
120 slash-symbol       = \l__xfrac_slash_symbol_tl       ,
121 slash-symbol-font  = \l__xfrac_slash_symbol_font_tl  ,
122 slash-symbol-format = \__xfrac_slash_symbol_format:n ,
123 v-scale            = \l__xfrac_vscale_fp
124 }

```

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.

```

125 {
126   \AssignTemplateKeys
127   \mode_if_math:TF
128   {
129     \cs_set_eq:NN \__xfrac_text_or_math:n \text
130     \cs_set_nopar:Npx \__xfrac_denominator_font_change:
131     { \tex_fam:D \l__xfrac_denominator_font_tl }
132     \cs_set_nopar:Npx \__xfrac_numerator_font_change:

```

```

133     { \tex_fam:D \l__xfrac_numerator_font_tl }
134 \cs_set_nopar:Npx \__xfrac_slash_symbol_font_change:
135     { \tex_fam:D \l__xfrac_slash_symbol_font_tl }
136 }
137 {
138   \cs_set_eq:NN \__xfrac_text_or_math:n \mbox
139   \cs_set_nopar:Npn \__xfrac_denominator_font_change:
140     {
141       \fontfamily { \l__xfrac_denominator_font_tl }
142       \selectfont
143     }
144   \cs_set_nopar:Npn \__xfrac_numerator_font_change:
145     {
146       \fontfamily { \l__xfrac_numerator_font_tl }
147       \selectfont
148     }
149   \cs_set_nopar:Npn \__xfrac_slash_symbol_font_change:
150     {
151       \fontfamily { \l__xfrac_slash_symbol_font_tl }
152       \selectfont
153     }
154 }

```

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

```

155   \__xfrac_text_or_math:n
156   {
157     \m@th
158     \hbox_set:Nn \l__xfrac_tmp_box
159       { \__xfrac_math:n { \vphantom { ( ) } } }
160     \hbox_set:Nn \l__xfrac_slash_box
161       {
162         \__xfrac_math:n
163         {
164           \__xfrac_slash_symbol_format:n
165           {
166             \__xfrac_math:n
167             {
168               \__xfrac_slash_symbol_font_change:
169               \IfNoValueTF {#2}
170                 { \l__xfrac_slash_symbol_tl } {#2}
171             }
172           }
173         }
174       }

```

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.

```

175   \cs_set_nopar:Npn \__xfrac_tmp:w

```

```

176     {
177         \raisebox
178         {
179             \__dim_eval:w
180             \box_ht:N \l__xfrac_slash_box
181             - \box_dp:N \l__xfrac_slash_box
182             - \height
183             \__dim_eval_end:
184         }
185     }
186 \dim_compare:nNnTF
187 { \l__xfrac_numerator_top_sep_dim } = { \c_max_dim }
188 {
189     \dim_compare:nNnF
190     { \l__xfrac_numerator_bot_sep_dim } = { \c_max_dim }
191     {
192         \cs_set_nopar:Npn \__xfrac_tmp:w
193         {
194             \raisebox
195             { \dim_use:N \l__xfrac_numerator_bot_sep_dim }
196         }
197     }
198 }
199 {
200     \dim_compare:nNnTF
201     { \l__xfrac_numerator_bot_sep_dim } = { \c_max_dim }
202     {
203         \cs_set_nopar:Npn \__xfrac_tmp:w
204         {
205             \raisebox
206             {
207                 \__dim_eval:w
208                 \box_ht:N \l__xfrac_slash_box
209                 - \dim_use:N \l__xfrac_numerator_top_sep_dim
210                 - \height
211                 \__dim_eval_end:
212             }
213         }
214     }
215     {
216         \msg_error:nn { xfrac }
217         { over-specified-numerator-sep }
218     }
219 }

```

Typeset the numerator.

```

220 \__xfrac_tmp:w
221 {
222     \__xfrac_fontscale:
223     \__xfrac_numerator_format:n

```

```

224         {
225             \scalebox
226             { \fp_use:N \l__xfrac_hscale_fp }
227             [ \fp_use:N \l__xfrac_vscale_fp ]
228             {
229                 \__xfrac_math:n
230                 {
231                     \__xfrac_numerator_font_change:
232                     {
233                         \vphantom { \l__xfrac_phantom_tl }
234                         #1
235                     }
236                 }
237             }
238         }
239     }
240     \__xfrac_math:n
241     { % THIS IS JUST WRONG!
242         \scan_align_safe_stop:
243         \mode_if_math:TF
244         { \tex_mskip:D \l__xfrac_slash_right_muskip }
245         { \tex_hskip:D \l__xfrac_slash_right_sep_dim }
246     }

```

Typeset the separator.

```

247     \box_use:N \l__xfrac_slash_box
248     \__xfrac_math:n
249     {
250         \mode_if_math:TF
251         { \tex_mskip:D \l__xfrac_slash_left_muskip }
252         { \tex_hskip:D \l__xfrac_slash_left_sep_dim }
253     }

```

Typeset the denominator.

```

254     \dim_compare:nNnTF
255     { \l__xfrac_denominator_bot_sep_dim } = { \c_max_dim }
256     {
257         \cs_set_nopar:Npn \__xfrac_tmp:w
258         { \raisebox { - \box_dp:N \l__xfrac_slash_box } }
259     }
260     {
261         \cs_set_nopar:Npn \__xfrac_tmp:w
262         {
263             \raisebox
264             { \dim_use:N \l__xfrac_denominator_bot_sep_dim }
265         }
266     }
267     \__xfrac_tmp:w
268     {
269         \__xfrac_fontscale:
270         \__xfrac_denominator_format:n

```

```

271         {
272             \scalebox
273             { \fp_use:N \l__xfrac_hscale_fp }
274             [ \fp_use:N \l__xfrac_vscale_fp ]
275             {
276                 \__xfrac_math:n
277                 {
278                     \__xfrac_denominator_font_change:
279                     {
280                         \vphantom { \l__xfrac_phantom_tl }
281                         #3
282                     }
283                 }
284             }
285         }
286     }
287 }
288 }

```

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.

```

289 \DeclareRestrictedTemplate { xfrac } { text } { math }
290 {
291     numerator-font      = \number \fam ,
292     slash-symbol        = / ,
293     slash-symbol-font   = \number \fam ,
294     denominator-font    = \number \fam ,
295     scale-factor        = 0.7 ,
296     scale-relative      = false ,
297     scaling             = true ,
298     numerator-top-sep   = 0 pt ,
299     denominator-bot-sep = 0 pt ,
300     math-mode           = true ,
301     phantom             = ( % )
302 }

```

### 4.3 The standard instances

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

The default “text” instance.

```

303 \DeclareInstance { xfrac } { default } { text } { }

```

The default “math(s)” instance.

```

304 \DeclareInstance { xfrac } { mathdefault } { math } { }

```

```

305 \DeclareCollectionInstance { plainmath } { xfrac } { mathdefault } { math }
306 {
307     denominator-bot-sep = 0 pt ,
308     numerator-bot-sep   = 0 pt ,
309     numerator-top-sep   = \c_max_dim ,

```

```

310     scale-factor      = 1          ,
311     scale-relative    = false      ,
312     scaling           = true       ,
313     slash-right-mkern = 0 mu       ,
314     slash-left-mkern  = 0 mu
315 }

```

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

```

316 \DeclareInstance { xfrac } { cmr } { text }
317 {
318     denominator-bot-sep = 0 pt      ,
319     numerator-top-sep   = 0.2 ex    ,
320     slash-left-kern     = -0.1 em   ,
321     slash-right-kern    = -0.1 em
322 }
323 \DeclareInstance { xfrac } { cmss } { text }
324 {
325     denominator-bot-sep = 0 pt      ,
326     numerator-top-sep   = 0.2 ex    ,
327     slash-left-kern     = -0.1 em   ,
328     slash-right-kern    = -0.1 em
329 }
330 \DeclareInstance { xfrac } { cmtt } { text }
331 {
332     denominator-bot-sep = 0 pt      ,
333     numerator-top-sep   = 0.2 ex    ,
334     slash-left-kern     = -0.1 em   ,
335     slash-right-kern    = -0.1 em
336 }

```

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

```

337 \bool_if:NF \l__xfrac_cm_std_bool
338 {
339     \DeclareInstance { xfrac } { cmr } { text }
340     { slash-symbol-font = ptm }
341     \DeclareInstance { xfrac } { cmss } { text }
342     { slash-symbol-font = ptm }
343     \DeclareInstance { xfrac } { cmtt } { text }
344     { slash-symbol-font = ppl }
345 }

```

Things works slightly better with Latin Modern.

```

346 \DeclareInstance { xfrac } { lmr } { text }
347 {
348     denominator-bot-sep = 0 pt      ,
349     numerator-top-sep   = 0.1 ex    ,
350     slash-left-kern     = -0.15 em  ,
351     slash-right-kern    = -0.15 em
352 }
353 \DeclareInstance { xfrac } { lmss } { text }

```



```

354 {
355     denominator-bot-sep = 0 pt      ,
356     numerator-top-sep   = 0 pt      ,
357     slash-left-kern     = -0.15 em ,
358     slash-right-kern    = -0.15 em
359 }
360 \DeclareInstance { xfrac } { lmtt } { text }
361 {
362     denominator-bot-sep = 0 pt      ,
363     numerator-top-sep   = 0 pt      ,
364     slash-left-kern     = -0.15 em ,
365     slash-right-kern    = -0.15 em
366 }

```

## 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.

```

367 \NewDocumentCommand \sfrac { o m o m }
368 {
369     \mode_if_math:TF
370     {
371         \IfInstanceExistTF { xfrac } { mathfam \number \fam }
372         { \UseInstance { xfrac } { mathfam \number \fam } }
373         { \UseInstance { xfrac } { mathdefault } }
374         {#2} {#3} {#4}
375     }
376     {
377         \IfInstanceExistTF { xfrac } {#1}
378         { \UseInstance { xfrac } {#1} }
379         {
380             \IfInstanceExistTF { xfrac } { \f@family }
381             { \UseInstance { xfrac } { \f@family } }
382             { \UseInstance { xfrac } { default } }
383         }
384         {#2} {#3} {#4}
385     }
386 }

```

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

## 4.5 Messages

Just the one.

```

387 \msg_new:nnnn { xfrac } { over-specified-numerator-sep }
388   { You have specified both numerator-top-sep and numerator-bot-sep}
389   {I will pretend that you didn't specify either of them}
390 \</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.

Symbols	
<code>\__dim_eval:w</code> .....	96, 179, 207
<code>\__dim_eval_end:</code> .....	98, 183, 211
<code>\__xfrac_denominator_font_change:</code> ..	
.....	36, 38, 130, 139, 278
<code>\__xfrac_denominator_format:n</code> .....	
.....	36, 39, 78, 270
<code>\__xfrac_fontscale:</code> .....	
.....	36, 36, 106, 108, 222, 269
<code>\__xfrac_math:n</code> .....	36, 37,
	82, 83, 159, 162, 166, 229, 240, 248, 276
<code>\__xfrac_numerator_font_change:</code> .....	
.....	36, 40, 132, 144, 231
<code>\__xfrac_numerator_format:n</code> .....	
.....	36, 41, 86, 223
<code>\__xfrac_relscale:</code> ...	36, 42, 94, 101, 111
<code>\__xfrac_slash_symbol_font_change:</code> ..	
.....	36, 43, 134, 149, 168
<code>\__xfrac_slash_symbol_format:n</code> .....	
.....	36, 44, 122, 164
<code>\__xfrac_text_or_math:n</code> .....	
.....	36, 45, 129, 138, 155
<code>\__xfrac_tmp:w</code> .....	20,
	20, 175, 192, 203, 220, 257, 261, 267
A	
<code>\AssignTemplateKeys</code> .....	126
B	
<code>\bool_if:N</code> .....	337
<code>\bool_set_false:N</code> .....	13
<code>\bool_set_true:N</code> .....	10
<code>\box_dp:N</code> .....	97, 181, 258
<code>\box_ht:N</code> .....	97, 102, 180, 208
<code>\box_new:N</code> .....	18, 19
<code>\box_use:N</code> .....	247
C	
<code>\c_max_dim</code> .....	49, 57, 58, 187, 190, 201, 255, 309
<code>\c_zero_dim</code> .....	112
<code>\cs_new:Npn</code> .....	20, 37, 39, 41, 44, 45
<code>\cs_new_nopar:Npn</code> .....	36, 38, 40, 42, 43
<code>\cs_set_eq:NN</code> .....	82, 83, 106, 129, 138
<code>\cs_set_nopar:Npn</code> .....	94, 101, 108,
	139, 144, 149, 175, 192, 203, 257, 261
<code>\cs_set_nopar:Npx</code> .....	130, 132, 134
D	
<code>\DeclareCollectionInstance</code> .....	305
<code>\DeclareInstance</code> .....	303, 304, 316,
	323, 330, 339, 341, 343, 346, 353, 360
<code>\DeclareObjectType</code> .....	46
<code>\DeclareRestrictedTemplate</code> .....	289
<code>\DeclareTemplateCode</code> .....	74
<code>\DeclareTemplateInterface</code> .....	47
<code>\dim_compare:nNnF</code> .....	189
<code>\dim_compare:nNnTF</code> .....	186, 200, 254
<code>\dim_new:N</code> .....	21, 22, 23, 24, 25
<code>\dim_use:N</code> .....	195, 209, 264
E	
<code>\ensuremath</code> .....	83
<code>\ExplFileDate</code> .....	4
<code>\ExplFileDescription</code> .....	4
<code>\ExplFileName</code> .....	4
<code>\ExplFileVersion</code> .....	4
F	
<code>\f@family</code> .....	50, 55, 70, 380, 381

