

The **fontscale** package

A user interface for setting document font sizes

Oliver Beery

Version 1.0.2 22 March 2024

1. Introduction

1.1. About

The **fontscale** package provides the following functionalities:

- Set document font sizes via a classic, modular, or musical typographic scale.
- Set arbitrary scales, sizes, and baselineskips for the standard font size commands `\tiny`, `\scriptsize`, `\footnotesize`, `\small`, `\normalsize`, `\large`, `\Large`, `\LARGE`, `\huge`, and `\Huge`.
- Set the font size settings mid-document or within a local group.
- Set the font size by setting the font step, scale, or size.
- Set the font size relative to the current font size.
- Use the current font step, scale, size, and baselineskip directly, for example, for printing or within calculations.
- Print sample text to test different font sizes.
- For advanced users, store the step, scale, size, and baselineskip of each font size command in `expl3` variables.

1.2. Loading the package

This package requires the 10 October 2023 `l3kernel` update. Since this update is fairly recent, you may need to ensure that your \LaTeX installation is up-to-date before using this package.

This package lets the user set arbitrary font sizes. This works with most fonts produced today, but, for historical reasons, the default Computer Modern font is available only in a number of fixed sizes. If you get a warning that Computer Modern is not available in the requested size, you may need to add the code `\RequirePackage{fix-cm}` before `\documentclass` to make Computer Modern available at arbitrary font sizes.

```
\usepackage{fontscale}  
\fontscalesetup{<keys>}
```

Load the `fontscale` package via the usual command in the document preamble. This package has no package options. Instead, the package keys are set up via the command `\fontscalesetup{<keys>}`. `<keys>` must receive a comma-separated list of keys and their associated values. The package keys are documented in §2 and §3.

Many L^AT_EX document classes have a font size option (`10pt`, `11pt`, `12pt`, etc.) which not only changes the set of document font sizes, but also modifies additional settings such as the page layout and vertical spacing which were specifically designed to work with those font sizes. For this reason, you may want to set the document class font size option close to the value of `\normalsize` set by this package.

Internally, this package sets up each font size command from `\tiny` to `\Huge` using the kernel command `\setfontsize`—they have no additional functionality beyond that of `\setfontsize`. Unlike the standard document classes, the font size commands set up by this package do not change the vertical spacing for displayed math and list structures. If the user would like to add this functionality or perhaps other features to the font size commands, they can do so via hooks, which are documented in the L^AT_EX kernel.

1.3. Terminology

When using the `fontscale` package, each font size, including the standard font size commands from `\tiny` to `\Huge`, has a step, scale, size, and baselineskip. The latter two parameters are the familiar first and second arguments to the low-level font command `\fontsize{<size>}{<baselineskip>}`. The font scale is the relative font size; the ratio of the font size to `\normalsize`. The font step is the number of font size gradations from `\normalsize`, which has a step of 0. Table 1 lists the font step of each font size command.

2. Choosing the typographic scale

<code>typographic-scale = classic-10pt, classic-11pt, classic-12pt,</code>	
<code>modular, musical</code>	<code>initial = classic-10pt</code>
<code>modular/ratio = <number></code>	<code>initial = 1.125</code>
<code>musical/ratio = <number></code>	<code>initial = 2</code>
<code>musical/notes = <integer></code>	<code>initial = 5</code>

The key `typographic-scale` sets the font sizes using a classic, modular, or musical typographic scale. These are common methods of choosing a set of document font sizes. The choice of typographic scale initializes to `classic-10pt`.

Table 1: The step of each font size command. These are constants and cannot be changed.

Command	Step
<code>\tiny</code>	-4
<code>\scriptsize</code>	-3
<code>\footnotesize</code>	-2
<code>\small</code>	-1
<code>\normalsize</code>	0
<code>\large</code>	1
<code>\Large</code>	2
<code>\LARGE</code>	3
<code>\huge</code>	4
<code>\Huge</code>	5

The key `typographic-scale` can also be set in a condensed format via the keys `classic-10pt`, `classic-11pt`, `classic-12pt`, `modular`, or `musical`, which set the key `typographic-scale` to the corresponding value.

The classic typographic scale consists of the traditional font sizes 6, 7, 8, 9, 10, 11, 12, 14, 16, 18, 21, 24, 36, 48, 60, and 72.¹ They have been used since the sixteenth century and are the default font sizes on most computer software. The keys `classic-10pt`, `classic-11pt`, `classic-12pt` set the size of each font size command to values based on their position in the classic typographic scale. The size of `\normalsize` is the named point size. The sizes of the other font size commands are the adjacent font sizes in the scale. Table 2 lists the size of each font size command when using a classic typographic scale.

The key `modular` sets the font sizes from `\tiny` to `\Huge` using a modular typographic scale. In a modular scale, each font size is a fixed ratio from the adjacent font sizes.² Here, the base font size is the size of `\normalsize`, which can be set via the key `normalsize/size = <size>`, more documented in §3. If not set, `\normalsize` initializes to 10 pt. The modular ratio can be set via the key `modular/ratio`. The initial modular ratio is 1.125.

1. See §3.1 of *The Elements of Typographic Style* by Robert Bringhurst.
2. <https://alistapart.com/article/more-meaningful-typography/>

Table 2: The size of each font size command in pt when using a classic typographic scale.

Command	classic-10pt	classic-11pt	classic-12pt
<code>\tiny</code>	6	7	8
<code>\scriptsize</code>	7	8	9
<code>\footnotesize</code>	8	9	10
<code>\small</code>	9	10	11
<code>\normalsize</code>	10	11	12
<code>\large</code>	11	12	14
<code>\Large</code>	12	14	16
<code>\LARGE</code>	14	16	18
<code>\huge</code>	16	18	21
<code>\Huge</code>	18	21	24

Table 3: The size of each font size command when using a modular or musical typographic scale with the initial settings. The font sizes are in pt and rounded to two decimal places.

Command	modular	musical
<code>\tiny</code>	6.24	5.74
<code>\scriptsize</code>	7.02	6.60
<code>\footnotesize</code>	7.90	7.58
<code>\small</code>	8.89	8.71
<code>\normalsize</code>	10	10
<code>\large</code>	11.25	11.49
<code>\Large</code>	12.66	13.20
<code>\LARGE</code>	14.24	15.16
<code>\huge</code>	16.02	17.41
<code>\Huge</code>	18.02	20

The key `musical` sets the font sizes according to their position in a musical typographic scale. The font size is calculated using the formula:³

$$f_i = f_0 \times r^{i/n}$$

f_0 is the base font size, the size of `\normalsize`. f_i is the font size with font step i . n is the number of notes in the musical scale—the number of font size gradations or steps above `\normalsize`. r is the ratio of the musical scale, the ratio of the highest to the lowest note f_n/f_0 . (The modular typographic scale is actually a special case of the musical typographic scale where $n = 1$.)

The musical ratio r and the number of notes n can be set via the keys `musical/ratio` and `musical/notes`. The initial musical ratio—2—and the initial number of notes—5—are the same as that of the classic typographic scale.⁴ Conveniently, `\Huge` is 5 steps above `\normalsize`. The font size of `\normalsize` f_0 can be set using the key `normalsize/size = <size>`, more documented in §3. When using a musical typographic scale, `\normalsize` initializes to 10pt.⁵

`baselineskip-size-ratio = <number>` `initial = 1.2`

The key `baselineskip-size-ratio` sets the ratio of the font baselineskip to the font size. By default, the font baselineskip of each font size command equals its font size times `baselineskip-size-ratio`. Users should not adjust this value unless they really know what they are doing. When using increased line spacing, users should prefer the low-level font command `\linespread` or the `setspace` package.

3. Overwriting the typographic scale

In some situations, users may need to explicitly set the scale, size, or baselineskip of a font size command, overwriting the value set by the keys in §2. Note that the font steps and the scale of `\normalsize` are constants and cannot be changed.

`/scale = <number>`

The keys `/scale` set the scale of `` to `<number>`, ignoring the value set by the key `typographic-scale`. The key `normalsize/scale` is not defined.

3. I have referenced this article by Spencer Mortensen:

<https://spencermortensen.com/articles/typographic-scale/>

4. Ibid.

5. I have set the initial value to 10pt instead of the 12pt recommended by the Spencer Mortensen article because this matches the default settings of the standard L^AT_EX 2_ε document classes.

`/size = <size>`

The keys `/size` set the size of `` to `<size>`, ignoring the values set by the keys `typographic-scale` and `/scale`. `<size>` supplies a default unit of pt.

The package will issue a warning if the user sets the keys `/size` and `/scale` for the same `` since the latter key will be ignored.

`/baselineskip = <skip>`

The keys `/baselineskip` set the font baselineskip of `` to `<skip>` instead of calculating it from the size of `` and `baselineskip-size-ratio`. `<skip>` supplies a default unit of pt.

` = <size>`

` = <size>/<baselineskip>`

The keys `` set the keys `/size` or `/baselineskip` in a condensed format. The latter format resembles traditional typographic notation where the size and leading of a text can be expressed as, for example, 10/12 or 11/13. The number before/after the slash is the size/leading in pts.

`ignore-order = true, false`

`initial = false`

The user should take care to ensure that the font sizes and font baselineskips remain in the correct order when directly setting the scales, sizes, or baselineskips of the different font size commands. The magnitudes of the sizes and baselineskips should be ordered from `\tiny` to `\normalsize` to `\Huge`. This is important for typographic and syntactic consistency. This package will issue a warning if they are not in the correct order. The user can disable this warning via the key `ignore-order = true` or just `ignore-order`.

4. User commands

4.1. Expandable font parameters

`\CurrentFontStep`

`\CurrentFontScale`

`\CurrentFontSize`

`\CurrentFontBaselineskip`

These are fully expandable commands that expand to the current font step, scale, size, or baselineskip. They can be used for printing or within calculations. Caution: `\CurrentFontStep` will expand to nothing if the current step is undefined.

Internally, `\CurrentFontStep`, `\CurrentFontScale`, and `\CurrentFontSize` use the kernel command `\f@size`. `\CurrentFontSize` expands to `f@size pt`. `\CurrentFontBaselineskip` expands to the kernel command `\f@baselineskip`. Note that the low-level font command `\fontsize{<size>}{<baselineskip>}` sets `\f@size` before expanding its second argument. This means that `\CurrentFontStep`, `\CurrentFontScale`, and `\CurrentFontSize` may need to be first expanded if used in `<baselineskip>`. This package provides the command `\SetFontSizeBaselineskip` as an alternative to `\fontsize + \selectfont` that avoids this issue (§4.2).

4.2. More font sizing commands

The `fontscale` package provides the commands `\SetFontStep`, `\SetFontScale`, `\SetFontSize`, `\ScaleFont`, and `\SetFontSizeBaselineskip` which set the font size in different ways. The first three commands have a starred version which sets the font size relative to the current font size. Internally, these commands all use the low-level font commands `\fontsize{<size>}{<baselineskip>}` and `\selectfont`. These commands should be used only in special cases. Generally, users should prefer the font size commands from `\tiny` to `\Huge` for typographic and syntactic consistency, although `\SetFontStep*{<integer>}` can use a font size command directly.

```
\SetFontStep{<number>}
\SetFontStep*{<number>}
```

`\SetFontStep` sets the font size by setting the font step to `<number>`. The starred version `\SetFontStep*` adds `<number>` to the current font step. If the choice of typographic scale is not modular or musical, then the only valid values for `<number>` are the integers from -4 to 5 , corresponding to the font size commands from `\tiny` to `\Huge`. `\SetFontStep` will issue an error if the step is invalid. `\SetFontStep` sets the font baselineskip to the calculated font size \times `baselineskip-size-ratio`. If the step matches the step of any font size command, then that font size command will be used directly.

Examples:

- For example, `\SetFontStep{-2}` and `\SetFontStep{2}` use `\footnotesize` and `\Large`. Note that it is typically better to use the equivalent font size command in place of `\SetFontStep{<integer>}`.

- If the current font size is `\normalsize`, `\SetFontStep*{1}` and `\SetFontStep*{-1}` use `\large` and `\small`. This is useful for moving up or down a note on the typographic scale.
- If the choice of typographic scale is modular or musical, `\SetFontStep*{-0.5}` sets the font size and font baselineskip to values between that of `\normalsize` and `\small`. Otherwise, it produces an error.

`\SetFontScale{<number>}`
`\SetFontScale*{<number>}`

`\SetFontScale` sets the font size by setting the font scale to `<number>`. The font baselineskip is set to the calculated font size \times `baselineskip-size-ratio`. The starred version `\SetFontScale*` adds `<number>` to the current font scale. For example, `\SetFontScale{1.25}` makes the font size 25% larger than `\normalsize`. If the current font size is `\normalsize`, `\SetFontScale*{0.25}` is equivalent to `\SetFontScale{1.25}`.

`\SetFontSize{<size>}`
`\SetFontSize*{<size>}`

`\SetFontSize` sets the font size to `<size>`. `<size>` supplies a default unit of pt. The font baselineskip is set to the font size \times `baselineskip-size-ratio`. The starred version `\SetFontSize*` adds `<size>` to the current size. For example, `\SetFontSize*{1}` and `\SetFontSize*{-1pt}` increase and decrease the current font size by a pt.

`\ScaleFont{<number>}`

Scales the current font size and the current font baselineskip by `<number>`. `\ScaleFont` resembles the `\scalefont` command provided by the `scalefnt` package, except that it uses floating point calculations and clears the stretch and shrink components of the font baselineskip. For example, if the current font size and font baselineskip are 10 pt and 12 pt, using `\ScaleFont{0.8}` will reduce them proportionally to 8 pt and 9.6 pt.

`\SetFontSizeBaselineskip{<size>}{<baselineskip>}`

Sets the font size and font baselineskip to `<size>` and `<baselineskip>`. Both `<size>` and `<baselineskip>` supply a default unit of pt. `\SetFontSizeBaselineskip` can be used as an alternative to `\fontsize + \selectfont` as it avoids the issue with `\f@size` discussed in §4.1

4.3. Testing and debugging

The commands documented here are intended primarily for testing and debugging. I have used them often when developing this package so I think users will find them helpful.

```
\PrintFontParameters
\PrintAllFontParameters
```

`\PrintFontParameters` prints the current step, scale, size, and baselineskip in this order in a comma-separated list of key = value pairs. If the current step is undefined (as explained for `\SetFontStep` in §4.2), it will be left blank. `\PrintAllFontParameters` prints the step, scale, size, and baselineskip of each font size command.

```
\PrintSampleText[⟨text⟩]
\PrintSampleText*[⟨text⟩]
```

`\PrintSampleText` prints `⟨text⟩` in each font size ordered from `\tiny` to `\Huge` each followed by a paragraph break. The starred version `\PrintSampleText*` reverses the sizing order. The `⟨text⟩` can be changed via the optional argument, but cannot contain `\par` tokens. The default `⟨text⟩` is “The `fontscale` package: A user interface for setting document font sizes”.

```
\PrintFontSizeCommand
```

`\PrintFontSizeCommand` tests whether the current font size and font baselineskip match that of any font size command. If so, it prints the name of that font size command. If not, it instead prints “UNDEFINED”. One useful way of printing example text is by using `\PrintSampleText` [`\PrintFontSizeCommand`:`\PrintFontParameters`].

4.4. More on `\fontscale` setup

```
\fontscalesetup{⟨keys⟩}
```

Sets and processes the `fontscale` package keys and uses `\normalsize`. `\fontscalesetup` can be applied mid-document or locally within the current `TeX` group. The key `reset` is also available. Its value must be either `initial` or `preamble`. The key `reset = initial` or just `reset` sets all the keys to their initial values. The key `reset = preamble` sets all the keys to their values at the end of the preamble (or does nothing if used before the end of the preamble).

5. Programming

For L^AT_EX3 programmers, the `fontscale` package provides some public `expl3` variables. No public `expl3` functions are provided. The package stores in public variables the step, scale, size, and baselineskip of each font size command. These variables should not be modified directly. Instead, they should be modified only indirectly via the command `\fontscalesetup`.

Each takes the form:

```
\c_fontscale_<font size command>_step_fp  
\l_fontscale_<font size command>_scale_fp  
\l_fontscale_<font size command>_size_dim  
\l_fontscale_<font size command>_baselineskip_skip
```

Exception: `\l_fontscale_normalsize_scale_fp` is not defined.

The scale of `\normalsize` is instead stored as the constant `\c_fontscale_normalsize_scale_fp`, which always equals 1.

The step, scale, size, and baselineskip of each font size command are also stored in property lists, one for each font size command:

```
\l_fontscale_<font size command>_prop
```