

# The Responsive Package

Michal Hoftich\*

Version 0.1, 2023-12-15

## **Abstract**

Responsive design aims to display text and other design elements well on variety of outputs, including electronic devices or various paper sizes. It originated on the Web, where it uses Cascading Style Sheets to change design elements.

This package tries to achieve similar result with  $\LaTeX$ . The main aim is to support output of multiple editions of one document with readable font sizes regardless of the physical size of the produced PDF, from smartphones or e-ink readers to printed A4 pages.

---

\*[michal.h21@gmail.com](mailto:michal.h21@gmail.com)

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Features</b>	<b>3</b>
2.1	Fitting Font Size to the Page Size . . . . .	3
2.2	Typographic Scale . . . . .	4
2.3	Line height . . . . .	4
2.4	Media Queries . . . . .	6
<b>3</b>	<b>Package Options</b>	<b>7</b>
<b>4</b>	<b>Commands</b>	<b>8</b>
<b>5</b>	<b>Troubleshooting</b>	<b>8</b>
5.1	Underfull \vbox warning . . . . .	9
<b>6</b>	<b>License</b>	<b>9</b>
<b>7</b>	<b>Changes</b>	<b>9</b>
	<b>References</b>	<b>9</b>

## 1 Introduction

This package aims to facilitate the creation of documents for multiple output formats, such as various page sizes, e-book readers, or even mobile phones.

One of the issues that needs to be addressed is setting the correct font size for readability. The default font size in LaTeX is 10 points, regardless of the page size. This is an appropriate font size for A5 format pages. For A4 format, the font size should be larger, while it may be smaller for smaller displays of e-readers and mobile phones. Similarly, we can adjust the line spacing, which also affects text readability depending on the font size and page size.

Web browsers face a similar problem as they must display text on both large PC monitors and smaller displays of laptops, tablets, and mobile phones. The solution they use is called *responsive design*.

Responsive design is a method of web design that allows flexible and dynamic adaptation of the appearance and layout of page content to different display devices. One of the key elements of responsive design is a flexible structure that enables adjusting the size of elements on the page to the display device.

Another important element is media queries. These allow defining rules based on the properties of the display device, such as screen width and height or output type (paper, display). Thanks to these rules, the same page code can be well-displayed on both large monitors and mobile devices or when printing.

The Responsive package is inspired by these principles. Its main function is to set the font size based on the page size and the approximate number of characters that should fit on the page (see figure 1). It also adjusts the typographic scale (influencing font size, for example, in headings or footnotes), line height, and supports a simple version of media queries.

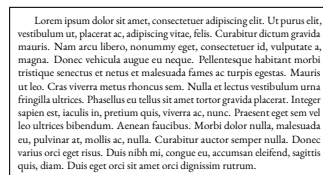
## 2 Features

### 2.1 Fitting Font Size to the Page Size

Regardless of the width of the page, there is a certain range of the number of characters that should be displayed on average per line of text to ensure good readability. With a larger number of characters, the reader may have difficulty quickly moving to the next line, while shorter lines may lead to skipping lines.

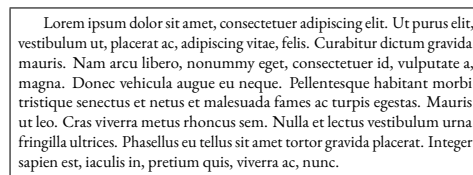
The Responsive package automatically sets the width of the text to a chosen number of characters. Figure 1 shows a difference in the font size in two different text widths.

Figure 1: Example of changing font size based on the width of the text block:



>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

(a) Text width 4 cm



>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc.

(b) Text width 6 cm

## 2.2 Typographic Scale

The typographic scale is a set of predetermined font sizes used to establish a consistent visual style for a document or website. These sizes are typically expressed in point units and progressively increase or decrease by a specific interval along the scale.

A typographic scale may include sizes for headings, footnotes, and body text. Proper use of a typographic scale helps create a visual hierarchy that enhances the readability and aesthetic appeal of the text.

In  $\LaTeX$ , the typographic scale is accessible through commands such as `\large`, `\huge`, or `\scriptsize`. Each of these commands is one interval away from the previous size.

I drew inspiration from Spencer Mortensen’s article (2011), which defines several different typographic scales and provides a method for easily creating additional scales. The default scale in the Responsive package, which closely aligns with the scale used in  $\LaTeX$ , is referred to as *tetratonic*. The package also offers additional scales described in the article, such as the golden scale based on the golden ratio. See Figure 2 for differences between the default scale and scale based on the golden ratio.

The Responsive package includes several predefined typographic scales, but you can also define your own. All scales are defined using the options `number` and `ratio`, specifying how many steps (`number`) the font size increases by the size of `ratio`. For example scale with `number` of 2 and `ratio` of 5, will double the font size over 5 steps. Table 1 lists all predefined scales.

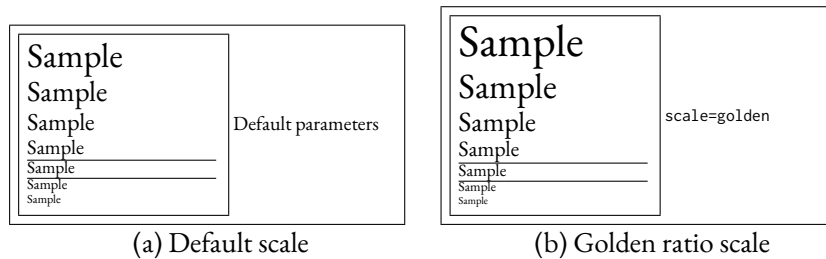
name	number	ratio
heptatonic	7	2
pentatonic	5	2
tetratonic	4	2
tritonic	3	2
golden	2	1.618

Table 1: Predefined typographic scales

## 2.3 Line height

In the default settings of  $\LaTeX$ , the line spacing is set to the font size multiplied by a factor of 1.2. For different fonts and page sizes, a different

Figure 2: Sample of Typographic Scales (the default font size is highlighted with lines)



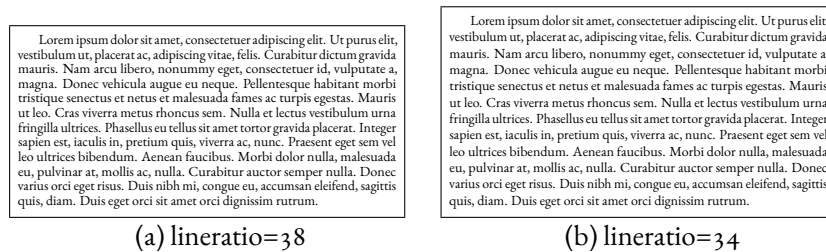
line spacing may be appropriate. Similarly, distinct values may be suitable for the printed and electronic versions of the document.

The Responsive package provides two options which allows you to change the line spacing (Figure 3).

First is the `linratio` option, which bases the line height on the ratio between x-height of the current font and the desired height. I've found this method in article by Edoardo Cavazza in the *Smashing Magazine* (2020). The value of the `linratio` option determines what percentage of the total line height is the x-height. The higher this value, the lower the line will be. For different fonts, the same ratio may lead to different line heights because each font has a different x-height value. The appropriate `linratio` size needs to be determined experimentally. The default value should roughly correspond to a line height of 1.2, the default in  $\LaTeX$ .

Another option is `lineheight`, which is similar to the CSS property `line-height`. Using it, we can choose the line height as a multiple of the font size.

Figure 3: Changing the line spacing by adjusting the value of `linratio`.



## 2.4 Media Queries

Media queries are a technique that allows web developers to dynamically adapt the appearance and behavior of websites based on various device properties, such as screen width and height, device orientation, color support, and more. Thanks to these conditions, responsive and flexible websites can be created that automatically adjust to different types and sizes of devices on which they are displayed.

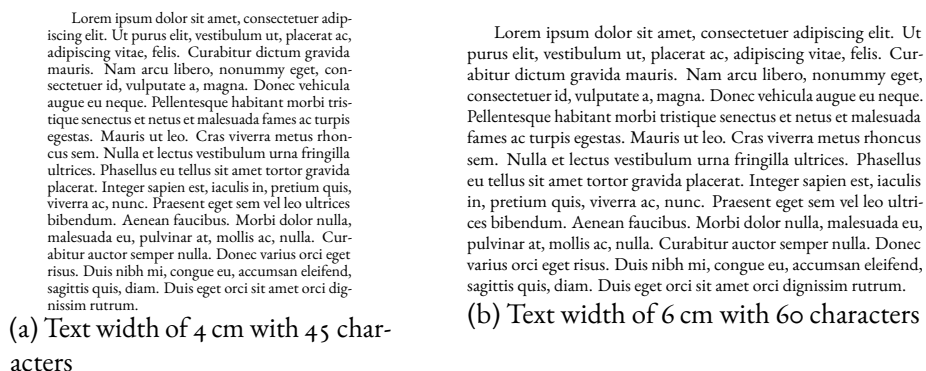
How can this technique be useful for  $\LaTeX$  package authors? They could, for example, set the font size, line spacing, and other elements for specific page dimensions. Once the user chooses the page size based on the device size for which they want to compile the document, these elements are automatically adjusted. For instance, a package author could define that if the width of the text block is smaller than a certain dimension, fewer characters will be displayed on each line compared to longer lines (Figure 4).

The media query used for this figure looks like this:

```
\mediaquery{max-textwidth=4cm}
{\ResponsiveSetup{linratio=38}\setsizes{45}}
\ResponsiveSetup{linratio=34}\setsizes{60}}
```

This media query defines that if the width of the text is smaller than or equal to 4 cm, a smaller number of characters will be displayed per line to prevent the font from being too small and difficult to read. Additionally, a smaller line spacing will be applied.

Figure 4: Change number of characters and line spacing depending on the text width using Media queries



The tests available in media queries are listed in Table 2. Those expecting dimensional units also have variants using `min-` and `max-`, which do not

test exact values but rather whether the provided value is less than, greater than, or equal to the tested parameter. Tests without these variants evaluate for an exact match. The test for a two-column layout is special in that it does not expect any testing values. It evaluates to true if the document uses the twocolumn option.

test	possible values
textwidth	dimensional unit
textheight	dimensional unit
paperwidth	dimensional unit
paperheight	dimensional unit
orientation	landscape, portrait
twocolumn	none

Table 2: Tests available in media queries

### 3 Package Options

The following options can be passed as package options used in `\usepackage`, or later in the document using the `\ResponsiveSetup` command.

- `characters` Approximate number of characters that should fit to a text line. Default value is 66.
- `noautomatic` Disable automatic setting of font sizes, line spacing and typographic scale in the document. You can set them later using the `\setsizes` and `\ResponsiveSetup` commands.
- `scale` Select named typographic scale. Default scale is `tetratonic`, which is the most similar to the default scale used by  $\LaTeX$ .
- `number` The number of steps during which the text size is multiplied by the amount specified in the `ratio` option.
- `ratio` The amount by which the font size will increase over a given number of steps.
- `lineratio` The ratio used in calculating the line spacing. Default value is 33.3333, which should produce similar line spacing as the default  $\LaTeX$  setting.
- `lineheight` Calculate line spacing by a fixed ratio, similarly to the CSS `lineheight` property.
- `boxwidth` Width of line that is used for font size calculations in the `\setsizes` com-

mand. Default value is `\textwidth`.

## 4 Commands

`\ResponsiveSetup` `\ResponsiveSetup{<options>}`

It allows you to reset the package options later in the document. You need to call the `\setsizes` command to reset font sizes explicitly if you use this command outside of the document preamble, or when using the `noautomatic` option.

`\setsizes` `\setsizes[<line height ratio>]{<number of characters per line>}`

It sets the text size so that the specified average number of characters fit on a line. If you do not provide the number of characters, the package option `characters` will be used.

`\fonttobox` `\fonttobox[<line height ratio>]{<text width>}{<text>}`

Set the text size so that the specified text in the second parameter has the specified size in the first parameter. Optionally, you can set the line height ratio.

`\mediaquery` `\mediaquery{<media query test>}{<>true code>}{<>false code>}`

It tests the document's properties using media queries and executes code based on the test results. The code can typically use `\ResponsiveSetup`, but you can also execute any other commands.

`\DeclareMediaQueryMatcher` `\DeclareMediaQueryMatcher{<test name>}{<LATEX 3 boolean query test>}`

Declares a new test for a media query. The tests can use L<sup>A</sup>T<sub>E</sub>X 3 boolean tests, and in case of a positive result, it must execute the command `\mediaquerytrue`.

`\fixtextheight` `\fixtextheight`

It adjusts the page height, header, and footer to prevent underflow `\vbox` warnings in the output routine.

## 5 Troubleshooting

Occasionally, you can run into issues caused by the change of font sizes. L<sup>A</sup>T<sub>E</sub>X sets lot of parameters depending on the base font size. We try to recalculate them according to the new font size, but you can still experience some issues, described in following subsections.



## 5.1 Underfull \vbox warning

Occasionally, you can get the following warning:

```
Underfull \vbox (badness 6412) has occurred while \output is active []
```

We calculate the page height that suits for the font size, but often, we would get error when a bigger font is used on the page, for example in chapters. When you set font sizes in the document preamble, text height should be changed automatically. If you reset font sizes later in the document, you need to use the `\fixtextheight` command. It should help in most cases. If not, you can try the following hack. This trick should fix `\vbox` errors in the output routine, but it can lead to an irregular number of lines on pages.

```
\ExplSyntaxOn
\def\@textbottom{\vskip \z@ \@plus \resp_font_size \@minus \resp_font_size}
\ExplSyntaxOff
```

## 6 License

Permission is granted to copy, distribute and/or modify this software under the terms of the L<sup>A</sup>T<sub>E</sub>X Project Public License, version 1.3.

## 7 Changes

- 2023-12-13 Version 0.1 released.  
Finished documentation.
- 2023-12-07 Added the `lineheight` property.  
Use floating point numbers for the `lineratio`.
- 2023-12-06 ChangeLog started.

## References

CAVAZZA, Edoardo, 2020. Modern CSS Techniques To Improve Legibility. *Smashing Magazine* [online] [viewed 2023-03-30]. Available from: <https://www.smashingmagazine.com/2020/07/css-techniques-legibility/>.

MORTENSEN, Spencer, 2011. *The Typographic Scale* [online]. [viewed 2023-03-30]. Available from: <https://spencermortensen.com/articles/typographic-scale/>.