

# Tips and tricks for titlesec

Javier Bezos



## What are rubber lengths?

Only in very rare occasions, the material in a page fits exactly in the page height, thus making necessary some adjusting.  $\LaTeX$  allows vertical spaces to *stretch* and *shrink* in order to perform this adjusting. (In fact, both horizontal and vertical spaces can do that, but we will concentrate in the vertical case. We will assume positive values; when negative values are involved there are additional considerations.)

A length may be followed by a `plus <length>` part which is used if the material is shorter than the page. In such a case, the extra space is distributed proportionally to the values after the `plus` parts. For instance, if a page contains the following vertical spaces:

```
3.5 pc plus 1 pc
2 pc
3 pc plus 2 pc
```

and the page is 6 pt larger than the material, 2 pt are added to the first space (i.e., it measures 3.5 pc + 2 pt) and 4 pt to the last, leaving the second one untouched. If the space to be distributed is greater than the total `plus` in a page, it is still distributed in the same fashion but  $\TeX$  will complain with an `Underfull \vbox`.

Instead of giving an absolute length after `plus` you may use a so-called infinite stretch with the `fil` unit. In this case, the `plus` parts containing absolute lengths are *ignored*, and only those based in `fil` are considered in the same way as absolute lengths are, except  $\TeX$  never complains. For instance, if a page contains the following spaces:

```
3.5 pc plus 1 fil
2 pc
2.5 pc plus 1 pc
3 pc plus 2 fil
```

and the page is 9 pc larger than the material, 3 pc are added to the first space and 6 pc to the last, leaving the second and the third ones untouched. There are further orders of infinities. If you use the `fill` unit, the `plus` parts with absolute lengths or `fil` are ignored in the computations; and similarly `filll`, `fillll` and so on.

For the case where the material is larger than the page there is a `minus` counterpart which behaves like `plus` except that the space is subtracted instead of added. However, there is still another difference: if absolute values are in force, they will be the *maximum* values to be subtracted. In other words, if the total `minus` in a page is smaller than the space to be retrieved, the values in the `lminusl` parts are used, thus producing an `Overfull \vbox` with text sticking out into the bottom margin.

The  $\LaTeX$  `\stretch{<num>}` command is just a synonymous with `0pt plus <num>fill` followed by a `\relax`. (This `\relax` makes `\titleformat` to raise an error if `\stretch` is used in the arguments allowing a `*` abbreviation, because they are expecting lengths without any additional stuff.)



## How can I place the chapter title at the very top of the page?

The `\topskip` parameter is added when `\vspace*` is used at the top of the page. Plain TeX has a `\topglue` which corrects the glue in these contexts, but sadly LaTeX has no built-in correction. Thus, if you like the title at the very top of the page, you should use a negative value which you must set by hand ( $\approx \text{\topskip} + \text{font size}$ ). Example (where font size is 10pt):

```
\titlespacing*{\chapter}{30mm}{-20pt}{40pt}
```



## How can I modify a shape?

[To be updated and filled.]

Titlesec provides some shapes which I think cover most of the possible ways to format a sectioning title. However, you can add your own shapes because they are loaded on request by the package. Perhaps you have noticed that titlesec comes with a few files named `block.tss`, `wrap.tss`, etc. These are the shapes not considered basic (the basic shapes are `hang`, `display` and `runin` and are defined in the main file) and they are not loaded except when used—perhaps you have guessed that `tss` means “TitleSec Shape.”

To add a shape you should know the internals of titlesec, which are not always easy to understand, but modifying a shape is a lot easier. The code in `block.tss` follows:

```
\gdef\ttlh@block#1#2#3#4#5#6#7#8{%
  \gdef\ttl@makeline##1{\hspace{#6}##1\hspace{#7}}%
  \setlength\leftskip{#6}%
  \setlength\rightskip{#7}%
  \interlinepenalty\@M
  \ttl@beginlongest
  #1%
  \ttl@glcmds
  \parindent\z@
  \leavevmode
  \ifttl@label
    {#2}\ifdim#3=\z@\else\hspace{#3}\fi
  \fi
  #4{#8}%
  \kern\z@\strut\@@par
  \nobreak\ttl@midlongest#5\@@par
  \ttl@endlongest}
```

Note that the definition must be global and the macro is named with the `ttlh@` prefix followed by the name of the shape. The eight parameters are:

1. Global format of title.
2. Label.
3. Space between label and title.

4. Format of title text.
5. Code below the title (the last optional argument of `\titleformat`).
6. Left space.
7. Right space.
8. Title text

The `\ttl@makeline` command is used by `\titleline`. Two commands are used by the `calwidth` option: `\ttl@beginlongest`, `\ttl@midlongest` and `\ttl@endlongest`; the enclosed code is evaluated twice, except that surrounded by `\ttl@midlongest` and the subsequent `\@@par` (the fifth parameter).

.....

Why `titlesec` does not provide predefined command for the scheme in “Starred sections”?

Because they are so simple... These are basic  $\LaTeX$  macros and introducing new ones with new names (which would be in fact a disguise of the original macros) does not provide further functionality.

.....

How can I collapse a range of sections in the running heads?

[To be filled.]

For that to be accomplished, we will need a  $\TeX$  macro named `\let`, whose meaning can be easily guessed.

.....

How can I change the name of chapters in the middle of a toc (say, to “Appendix”)?

[To be filled.]

.....

A little more on `\titleline`

[To be filled.]