The titlesec and titletoc Packages*

Javier Bezos†

February 20, 1999

Contents

1. Introduction

This package allows to change in a straightforward way the sectioning format of a document. It works with the standard classes and with many others, including the AMS ones.[1]

The package is loaded by `\usepackage` and then you redefine the sectioning commands. That may be done by the package itself (see section “Simple Interface”) or by you with the provided commands (see section “Advanced Interface.”) In the latter case, you only need to redefine the commands you’ll use. Both methods are available at time, but because `\part` is usually implemented in a non-standard way, it remains untouched and should be changed “by hand” or with the help of the “Advanced Interface.”

[*] The titlesec package is currently at version 2.1. © 1998, 1999 Javier Bezos. All Rights Reserved. The titletoc package is currently at version 1.0. © 1999 Javier Bezos. All Rights Reserved.

[†] Please, send your comments, suggestions and bug reports to jbezos@mx3.redestb.es, or to my postal address: Apartado 116.035, E-28080 Madrid, Spain. English is not my strong point, so contact me when you find mistakes in the manual. Other packages by the same author: accents, polyglot, spanishb, tensind, esindex.

[1] However, note that the AMS classes reimplement the original internal commands. These changes will be lost here.

2. Simple Interface

The easiest way to change the format is by means of a set of package options.

2.1. Format

There are three option groups controlling font, size and align. You need not set all of these groups, since a default is provided for each one; however, you must at least set one of them, or otherwise titlesec performs no formatting and hence you must use the “advanced” commands described below.

```
rm sf tt md bf up it sl sc
```

select the corresponding family/series/shape. Default is `bf`.

Size adjustment is performed by

```
big medium small tiny
```

Default is `big`, which gives the size of standard classes. With `tiny`, sections (except chapters) are typed in the text size. `medium` and `small` are intermediate layouts.

Alignment is controlled by

```
raggedleft center raggedright
```

Don’t use `raggedright` if the document itself is already `\raggedright`.[2]

2.2. Spacing

```
compact
```

This options is independent from the above and reduces the spacing.

2.3. Tools

`\titlelabel{`⟨*label-format*⟩`}`

Changes the label format in sections, subsections, etc. A `\thetitle` command is provided which is respectively `\thesection`, `\thesubsection`, etc. The default value in standard classes is

```
\titlelabel{\thetitle\quad}
```

and you may add a dot after the counter simply with

```
\titlelabel{\thetitle.\quad}
```

That was done in this document.

[2] Note to users of version 1.1: in that version these options where misnamed `flushleft` and `flushright` thus giving the opposite alignment.

3. Advanced Interface

Two commands are provided to change the title format. The first one is used for the "internal" format, i. e., shape, font, label…, the second one defines the "external" format, i. e., spacing before and after, indentation… This scheme is intended to easy definitions, since in most of cases you will want to modify either spacing or format.[3] Note that you can redefine existing sectioning commands, but you cannot define *new* ones.

3.1. Format

A set of shapes is provided, which control the basic distribution of elements in a title. The available shapes in all of sectioning commands are:

hang is the default value, with a hanging label. (Like the standard `\section`.)

block typesets the whole title in a block (a paragraph) without additional formatting. Useful in centered titles[4] and special formatting (including graphic tools such as `picture`, `pspicture`, etc.)

display puts the label in a separate paragraph. (Like the standard `\chapter`.)

runin A run-in title. (Like the standard `\paragraph`.)

leftmargin puts the title at the left margin. Titles at the very end of a page will be moved to the next one and will not stick out in the bottom margin, which means that large titles can lead to underfull pages.[5] In this case you may increase the strechability of the page elements, use `\raggedbottom` or use the package option `nobottomtitles` described below. Since the mechanism used is independent from that of the margin pars, they can overlap. `margin` is a synonymous.

rightmargin is like `leftmargin` but at the right margin.

drop wraps the text around the title, provided that the first paragraph is longer than the title (if not, they overlap). The comments in `leftmargin` also applies here.

wrap is an experimental shape, quite similar to drop. The only difference is that while the space reserved in drop for the title is fixed, in wrap is automatically readjusted to the longest line. Because of the way it is handled, there are two main limitations: you cannot use `\\` (but `\newline` is allowed), and the title has to be ragged right with the `\filright` command described below.

frame similar to display, but the title will be framed.

Note, however, that some shapes does make sense in chapters and parts.

```
\titleformat{⟨command⟩}[⟨shape⟩]{⟨format⟩}{⟨label⟩}{⟨sep⟩}{⟨before⟩}[⟨after⟩]
```

Here

- ⟨*command*⟩ is the sectioning command to be redefined, i. e., `\part`, `\chapter`, `\section`, `\subsection`, `\subsubsection`, `\paragraph` or `\subparagraph`.
- The paragraph shape is set by ⟨*shape*⟩, as described above.

[3] Information is "extracted" from the class sectioning commands, except for chapter and part. Standard definitions with `\@startsection` are presumed, and if an special procedure was followed an arbitary value is provided, which you may change later. (Sadly, there is no way to catch the chapter or part formats and one similar to that of standard classes will be used.) That makes this package compatible with sectsty or fncychap provided they are loaded before and no "simple interface" settings are used.

[4] The label will be slightly displaced to the left if the title is two or more lines long and the `hang` shape is used.

[5] However, floats following the title a couple of lines after will interfere with the page breaking used here and sometimes the title may stick out.

- ⟨*format*⟩ is the format to be applied to the whole title—label and text. This part can contain vertical material which is typesseted just after the space above the title.
- The label is defined in ⟨*label*⟩. You may omit it if there is no section label at that level, but note that by removing it the number is not suppresed in the table of contents and headers. See below for illustrative examples.
- ⟨*sep*⟩ is the separation between label and title body. This space is horizontal in `hang`, `block`, `runin`, `drop`, `wrap` and `...margin` shapes, and vertical in `display` one; in the `frame` shape is the distance from text to frame. Both ⟨*label*⟩ and ⟨*sep*⟩ are ignored in starred versions of sectioning commands. If you are using `picture` and the like, set this parameter to 0 pt.
- ⟨*before*⟩ is code preceding the title body. The very last command can take an argument, which is the title text.
- ⟨*after*⟩ is code following the title body. The typesetted material is in vertical mode in `hang`, `block` and `display` shapes; in horizontal mode in `runin` mode. Otherwise is ignored.

3.2. Spacing

`\titlespacing*{`⟨*command*⟩`}{`⟨*left*⟩`}{`⟨*beforesep*⟩`}{`⟨*aftersep*⟩`}[`⟨*right*⟩`]`

The starred version supresses the indentation of the paragraph following the title, except in `drop`, `wrap` and `runin` where this alternative does not make sense.

- ⟨*left*⟩ is the “left space” of the title. In the `...margin`, `wrap` and `drop` shapes this parameter sets the title width, in `wrap` the maximum width, in `runin` the indentation, and in the remainder a new left margin with respect to the current one. With negative value the title overhangs. This parameter is not equal to ⟨*indent*⟩ of `\@startsection`, which doesn't work correctly. With a negative value in the latter and if ⟨*indent*⟩ is larger than the label width, the first line of the title will start in the outer margin, as expected, but the second, third… lines will not; worse, those lines will be shortened at the right margin.
- ⟨*beforesep*⟩ is the vertical space before the title.
- ⟨*aftersep*⟩ is the separation between title and text—vertical with `hang`, `block`, and `display`, and horizontal with `runin`, `drop`, `wrap` and `...margin`. By making the value negative, you may define an effective space of less than `\parskip`.[6]
- The `hang`, `display`, `block` and `display` shapes have the possibility to set the ⟨*right*⟩ margin with the optional argument.

If you dislike typing the full skip values, including the `plus` and `minus` parameters, an abreviation `*`n is provided (where n is an integer). In the ⟨*beforesep*⟩ argument this is equivalent to n `ex` with some stretchability and a minute shrinkability. In the ⟨*aftersep*⟩ some strechability (smaller) and no shrinkability. (Actually, they stand for n times `1ex plus .3ex minus .06ex` and `1ex plus .1ex`, respectively.) Thus, you can write

```
\titlespacing{\section}{0pt}{*4}{*1}
```

Note. `\titlespacing` does not work with neither `\chapter` nor `\part` unless you change its title format as well by mean of `\titleformat` (or the simple interface settings).

[6] See Goosens, Mittelbach and Samarin: *The LaTeX Companion*, Reading, Addison Wesley, 1993, p. 25.

3.3. Tools

These commands are provided as tools for `\titleformat` and `\titlespacing`.

```
\filright  \filcenter  \filleft  \fillast  \filinner  \filouter
```

Variants of the `\ragged...` commands, with slight differences. In particular, the `\ragged...` commands cancels the left and right spaces setted by `\titlespacing`. `\fillast` justifies the paragraph, except the last line which is centered. If the document is `\raggedright` you must reset the value of `\rightskip` to 0 pt before, or some irregular format can arise. These commands work in the `frame` label, too.

`\filinner` and `\filouter` are `\filleft` or `\filright` depending on the page. Because of the asyncronous TEX page breaking, these commands can be used in `\chapter` only.

```
\wordsep
```

The inter-word space of the current font. See below for an example.

```
\titleline[⟨align⟩]{⟨horizontal material⟩}
\titlerule[⟨height⟩]
\titlerule*[⟨width⟩]{⟨text⟩}
```

This command allows an horizontal line in vertical mode, where the title ⟨*left*⟩ and ⟨*right*⟩ settings are taken into account, and the margin titles as well. It's intended mainly for rules and leaders but in fact is also useful for other purposes. The line has a fixed width and hence must be filled, i.e., `\titleline{CHAPTER}` produces an underfull box. Here the optional ⟨*align*⟩ (`l`, `r` or `c`) helps, so that you simply type, say, `\titleline[c]{CHAPTER}`.[7]

Using `\titleline` in places where vertical material is not expected can lead to anomalous results. In other words, you can use it in the ⟨*format*⟩ (always) and ⟨*after*⟩ (`hang`, `display` and `block`) arguments; and in the `display` shape at the very beginning of the ⟨*before*⟩ and ⟨*label*⟩ argument as well. But try it out, because it's very likely it works in other places.

The `\titlerule` command, which makes use of `\titleline` if necessary, can be used to build rules and fillers. The unstarred version draws rules of height .4 pt or ⟨*height*⟩, if present. The starred version makes leaders with the ⟨*text*⟩ repeated in boxes of its natural width. The width of the boxes can be changed to ⟨*width*⟩, but the first box remains with its natural width so that the ⟨*text*⟩ is aligned to the left and right edges of the space to be filled.

For instance, with

```
\titleformat{\section}[leftmargin]
  {\titlerule*[1pc]{.}%
   \vspace{1ex}%
   \bfseries}
  {... definition follows
```

leaders spanning over both main text and title precede the section.

3.4. Breaks

```
\sectionbreak    \subsectionbreak    \subsubsectionbreak
\paragraphbreak  \subparagraphbreak
```

By defining these command with `\newcommand` different page breaks could be applied to different levels. In those undefined, a penalty with the internal value provided by the class is used (typically −300). For instance,

[7] The default is the `s` `\makebox` option.

```
\newcommand{\sectionbreak}{\clearpage}
```

makes sections begin a new page. In some stylings, the space above the title is preserved even if the section begins a new page; that's done by:

```
\newcommand{\sectionbreak}{%
  \addpenalty{-300}%
  \vspace*{0pt}}
```

3.5. Package Options

Related package options are:

`indentfirst nonindentfirst`

By-pass the settings for all of sectioning command.

`rigidchapters rubberchapters`

With `rigidchapters` the space for chapter titles is always the same and ⟨*aftersep*⟩ in `\titlespacing` is the total amount reserved to them, i. e., ⟨*beforesep*⟩ + ⟨*aftersep*⟩ is the distance from the top of the page body to the main text. The default is `rubberchapters` which ⟨*aftersep*⟩ is the separation between title and text in, as usual.

`bottomtitles nobottomtitles`

If the `nobottomtitles` is set, titles close to the bottom margin will be moved to the next page and the margin will be ragged. The minimal space required in the bottom margin not to move the title is set (approximately) by

```
\renewcommand{\bottomtitlespace}{⟨length⟩}
```

whose default value is `.3\textheight`. A simple ragged bottom on the page before is obtained with a value of 0 pt. `bottomtitles` is the default, which simply sets `\bottomtitlespace` to a negative value.

`aftersep largestsep`

By default, when there are two consecutive titles the ⟨*aftersep*⟩ space from the first one is used between them. Sometimes that is not the desired behaviour, especially when the ⟨*beforesep*⟩ space is much larger than the ⟨*aftersep*⟩ one (otherwise the default seems preferable). With `largestsep` the largest of them is used. `aftersep` is the default.

`newparttoc oldparttoc`

Standard parts write the toc entry number in a non standard way. You may change that with `newparttoc` so that titletoc or a similar package can manipulate the entry. That works if the `\part` has been redefined.

`clearempty`

Modifies the behaviour of `\cleardoublepage` so that the `empty` page style will be used in empty pages.

3.6. Examples

The following examples will be illustrative. In this part, the `\parskip` is 0 pt, and the ragged right has been canceled out.

■

[*4*] This is an example of the section command defined below and, what's more, this is an example of the section command defined below

```
\titleformat{\section}[block]
  {\normalfont\bfseries\filcenter}{\fbox{\itshape\thesection}}{1em}{}
```

■

SECTION 5

A framed title

```
\titleformat{\section}[frame]
  {\normalfont}
  {\filright
   \footnotesize
   \enspace SECTION \thesection\enspace}
  {8pt}
  {\Large\bfseries\filcenter}
```

■

6. A Ruled Title

```
\titleformat{\section}
  {\titlerule
   \vspace{.8ex}%
   \normalfont\itshape}
  {\thesection.}{.5em}{}
```

■

7

Another Ruled Title

```
\titleformat{\section}[block]
  {\normalfont\sffamily}
  {\thesection}{.5em}{\titlerule\\[.8ex]\bfseries}
```

■

SECTION 8

This is an example of the section command defined below and, what's more, this is an example of the section command defined below. Let us repeat it. This is an example of the section command defined below and, what's more, this is an example of the section command defined below

```
\titleformat{\section}[display]
  {\normalfont\fillast}
  {\scshape section \oldstylenums{\thesection}}
  {1ex minus .1ex}
  {\small}
\titlespacing{\section}
  {3pc}{*3}{*2}[3pc]
```

■

This part is the title itself and this part is the section body…

```
\titleformat{\section}[runin]
  {\normalfont\scshape}
  {}{0pt}{}
\titlespacing{\section}
  {\parindent}{*2}{\wordsep}
```

■

9. A Simple Example of the "wrap" Section Shape Which is followed by some text to show the result. Which is followed by some text to show the result. Which is followed by some text to show the result. Which is followed by some text to show the result. Which is followed by some text to show the result. Which is followed by some text to show the result. Which is followed by some text to show the result.

10. And another Note how the text wraps the title and the space reserved to it is readjusted automatically. And that is followed by some text to show the result. Which is followed by some text to show the result.

```
\titleformat{\section}[wrap]
  {\normalfont\fontseries{b}\selectfont\filright}
  {\thesection.}{.5em}{}
\titlespacing{\section}
  {12pc}{1.5ex plus .1ex minus .2ex}{1pc}
```

■

§ 11. Old-fashioned runin title.—Of course, you would prefer just a dot after the title. In this case the optional argument should be `[.]` and the space after a sensible value (1em, for example).

```
\titleformat{\section}[runin]
  {\normalfont\bfseries}
  {\S\ \thesection.}{.5em}{}[.---]
\titlespacing{\section}
  {\parindent}{1.5ex plus .1ex minus .2ex}{0pt}
```

■

Example of margin section

Which is followed by some text to show the result. But don't stop reading, because the following example illustrates how to take advantage of other packages. The last command in the last argument can take an argument, which is the title with no aditional command inside it. We just give the code, but you may try it by yourself. Thus, with the soul package you may say

```
\newcommand{\secformat}[1]{\MakeLowercase{\so{#1}}}
   % \so spaces out letters
\titleformat{\section}[block]
  {\normalfont\scshape\filcenter}
  {\thesection}
  {1em}
  {\secformat}
```

The margin title above was defined:

```
\titleformat{\section}[leftmargin]
  {\normalfont
   \titlerule*[.6em]{\bfseries.}%
   \vspace{6pt}%
   \sffamily\bfseries\filleft}
  {\thesection}{.5em}{}
\titlespacing{\section}
  {4pc}{1.5ex plus .1ex minus .2ex}{1pc}
```

■

The following examples are intended for chapters. However, this document lacks of `\chapter` and are showed using `\sections` with slight changes.

CHAPTER 12

The Title

```
\titleformat{\chapter}[display]
  {\normalfont\Large\filcenter\sffamily}
  {\titlerule[1pt]%
   \vspace{1pt}%
   \titlerule
   \vspace{1pc}%
   \LARGE\MakeUppercase{\chaptername} \thechapter}
  {1pc}
  {\titlerule
   \vspace{1pc}%
   \Huge}
```

■

CHAPTER XIII

The Title

```
\renewcommand{\thechapter}{\Roman{chapter}}
\titleformat{\chapter}[display]
  {\bfseries\Large}
  {\filleft\MakeUppercase{\chaptername} \Huge\thechapter}
  {4ex}
  {\titlerule
   \vspace{2ex}%
   \filright}
  [\vspace{2ex}%
   \titlerule]
```

4. Extended Settings

This is a new feature available from version 2.0.[8] The first argument of `\titleformat` has an extended syntax which allows to set different formats depending on the context. This argument can be a list of key/value pairs in the form:

[8] The keyval package is required to make use of it.

⟨*key*⟩=⟨*value*⟩, ⟨*key*⟩=⟨*value*⟩, ⟨*key*⟩, ⟨*key*⟩,...

Currently, only pages and unnumbered versions are taken care of, besides the sectioning command name. Thus, the available keys are:

- `name`. Allowed values are `\chapter`, `\section`, etc.
- `page`. Allowed values are `odd` or `even`.
- `numberless`. Allowed values are `true` and `false`. By default is false and this key is not necessary unless you want to set different numbered (without this key) and unnumbered (with just `numberless` without value) variants.

The basic form described above with the name of a sectioning command, say

```
\titleformat{\section} ...
```

is an abbreviation for

```
\titleformat{name=\section} ...
```

Let's suppose that we'd like a layout with titles in the outer margin. We may set something like

```
\titleformat{name=\section,page=even}[leftmargin]
  {\filleft\scshape}{\thesection}{.5em}{}

\titleformat{name=\section,page=odd}[rightmargin]
  {\filright\scshape}{\thesection}{.5em}{}
```

Since the page information is written to the `aux` file, at least two runs are necessary to get the desired result.

The "number" version is usually fine when generating unnumbered sections, but if you need some special formatting, there is the `numberless` key which defines an alternative version for sections without numbers (namely, those with level below `secnumdepth`, in the front and back matters and, of course, the starred version). For instance

```
\titleformat{name=\section}{...% The normal definition follows
\titleformat{name=\section,numberless}{...% The unnumbered
% definition follows
```

Neither ⟨*label*⟩ nor ⟨*sep*⟩ are ignored in `numberless` variants.

These keys are available to both `\titleformat` and `\titlespacing`, unless `numberless` which is only available to the former. By using `page` in one (or both) of them, odd and even pages will be formatted differently. Actually, "even" and "odd" are well established LaTeX terms, but misleading. In one side printing the "odd" pages refer to "even" pages as well (cf. `\oddsidemargin`.)

If you intend to create different odd/even *and* different numbered/unnumbered versions, it's recommended defining the four variants.[9]

If you remove the page specifier from a sectioning command you must remove the `.aux` file.

[9] How to synchronize many keys is still under study. By defining the four variants you make sure that future versions of this package will not produce unexpected changes.

5. Page Styles

As a part of this package a set of commands are provided for one-step headline and footline settings. It's not intented for complex headlines which packages like fancyhdr are better suited for, but most of cases can be handled. The main limitation is that you must restrict yourself to pass information to either headlines or footlines, but not to both (by default are passed to headlines); the other may contain text and page numbers, but not section titles, numbers, etc. It works with standard classes and many others, even if titles are not redefined; however, additional LaTeX layout adjustments could be necessary in some of them (for instance the AMS classes).[10] If you use an alternative package, like those above, don't forget the package option

```
nops
```

and all of commands described in this section will be ignored.

5.1. Definining Page Styles

```
\newpagestyle{⟨name⟩}[⟨global-style⟩]{⟨commands⟩}
\renewpagestyle{⟨name⟩}[⟨global-style⟩]{⟨commands⟩}
```

Defines a new style and redefines an existing one, named ⟨*name*⟩. For convenience, end of lines are ignored, and you needn't "hide" them with `%`.

⟨*global-style*⟩ is any command to be applied to both footlines and headlines. Fragile commands are allowed.

In ⟨*commands*⟩ you can use (these commands are local to a page style, i.e., defining them directly in the preamble will not work because the page style settings will override them):

```
\headrule                \footrule
\setheadrule{⟨length⟩}   \setfootrule{⟨length⟩}
```

If you want a rule below the headline and above the footline. You may also set their width directly with the commands `\setheadrule` and `\setfootrule`. (For instance, `\setheadrule{.4pt}` which is, by the way, the default value.)

```
\makeheadrule \makefootrule
```

These commands are used by titlesec to typeset the rules. If there is no rule, both commands are empty (this value is the default). `\setheadrule{d}` just stands for

```
\renewcommand{\makeheadrule}{\rule[-.3\baselineskip]{\linewidth}{d}}
```

unless if `d` is 0 pt, which empties `\makeheadrule` (and similarly `\setfootrule`).

You can access the total head/foot width with `\linewidth` but the box enclosing the rule is in fact dimensionless. Hence, you must be concerned with the rule placement only. Its baseline is the same than that of the head/foot. That means that the material to be used as rule should be raised or lowered with `\raisebox`, the lifting argument in `\rule` or by using sensible coordinates in `picture`, for instance. That makes easy to put material above *and* below. For instance, the following code creates a headline with a black bold line above and a red rule below (the color package is required):

[10] Two columns layout requires the fixmarks package by Joe Pallas and Piet van Oostrum. However, this package does not fix twocolumn + starred sectioning command.

```
\renewcommand{\makeheadrule}{%
    \makebox[0pt][l]{\rule[.7\baselineskip]{\linewidth}{0.8pt}}%
    \color[named]{Red}%
    \rule[-.3\baselineskip]{\linewidth}{0.4pt}}
```

Of course, the material to be used as rule is not restricted to actual rules; pictures and leaders, for instance, are also available.

```
\sethead[⟨even-left⟩][⟨even-center⟩][⟨even-right⟩]
        {⟨odd-left⟩}{⟨odd-center⟩}{⟨odd-right⟩}
\setfoot[⟨even-left⟩][⟨even-center⟩][⟨even-right⟩]
        {⟨odd-left⟩}{⟨odd-center⟩}{⟨odd-right⟩}
```

Set the corresponding part in heads and foots. The optional arguments are all or none. If no optional arguments are given the odd-page settings applies to both even and odd pages. In the arguments of `\sethead` and because of its one-step mechanism, we must distingish between two groups of command. The first one are those expanded when the marks are sent, with information related to sections, and includes:

- `\thechapter`, `\thesection`...
- `\chaptertitle`, `\sectiontitle`... which print the chapter, section... title.
- Any command which is neither fragile nor protected (a command just containing text, for example).

The second group are those expanded in every page, and includes:

- `\usepage`. The page number, but evaluated in every page. (`\thepage` is evaluated when the mark is sent—not a desired behaviour, indeed.)
- Any protected command.

Fragile commands must be protected with `\protect`.

In `\setfoot` the commands are always evaluated in every page.

```
\setmarks{⟨primary⟩}{⟨secondary⟩}
```

Sets which `\...title` commands are to be defined, and when the marks are updated. For example `\setmarks{chapter}{section}` means that:

- `\chaptertitle` and `\sectiontitle` are the allowed titles in heads,
- `\sectiontitle` is reset in `\chapter`, and
- `\markboth{`⟨*even-head*⟩`}{`⟨*odd-head*⟩`}` is executed in `\chapter` and `\section`, where ⟨*even-head*⟩ and ⟨*odd-head*⟩ stand for the internal definitions generated by `\sethead`.
- `\partmark` is defined `\chaptermark{}`.

By default it is `\setmarks{chapter}{section}` but `\setmarks{section}{subsection}` in article class.

Note that `\markboth` is a command to set the `myheadings` marks and here does not make sense. In fact, using it can lead to impredictable results. Anyway, using directly the marking commands is discouraged (see sec. 7.2).

5.2. Additional Settings

```
\widenhead[⟨even-left⟩][⟨even-right⟩]{⟨odd-left⟩}{⟨odd-right⟩}
```

Makes the head/foot lines wider. The extra width is asymetrically added and hence the two [four] arguments.[11]

Related package options are:

```
outermarks innermarks topmarks botmarks
```

`innermarks` is the default in LaTeX, with bottom marks in even pages and first marks in odd pages—not useful, not usual and, furthermore, it can lead to undesired results. More convenient is `outermarks`, with top marks in even pages and bottom marks in odd pages; it's the usual system and described in *The TeXbook*, p. 259. Both of them are intended for two side printing; `topmarks`/`botmarks` are intended for one side printing with top/bot marks in every page (they can be used in two side printing as well).

Since `\markboth` is used when the marks are updated, as set by `\setmarks`, special constructions like that are allowed:

```
\newpagestyle{mypagestyle}[\small\sffamily]{
  \sethead [\textbf{\usepage}]
           [\textsl{\chaptertitle}]
           [[\thesection]
           {\thesection]}
           {\textsl{\sectiontitle}}
           {\textbf{\usepage}}}
```

As you can see, `\thesection` is printed in both left and right heads (with right values!)

```
footmarks
```

The sectioning information is received by the footlines instead of the headlines. In this case the `\sethead` arguments are evaluated in every page.

In this document, the following styles are used:

```
\widenhead{2.1pc}{0pc}
\renewpagestyle{plain}[\small\sffamily\slshape]{
  \footrule
  \setfoot{}{\usepage}{}}
\newpagestyle{myps}[\small\sffamily\slshape]{
  \headrule
  \sethead{Titlesec}{\sectiontitle}{\usepage}}
```

The following definitions provide page styles similar to those of Lamport's LaTeX book.

```
\newlength{\headind}
\setlength{\headind}{\marginparsep}
\addtolength{\headind}{\marginparwidth}
\widenhead[\headind][0pt]{0pt}{\headind}
\renewpagestyle{plain}{}
\newpagestyle{latex}[\bfseries]{
  \headrule
  \sethead[\usepage][][\chaptertitle]
          {\thesection\ \sectiontitle}{}{\usepage}}
```

[11] Formerly named `\setheadinded`, now deprecated. That was misleading because it suggests an opposite sign in the values to be used.

6. Examples

6.1. A full example

Now an example of a complete title scheme follows.

```
\documentclass[twoside]{report}
\usepackage[sf,sl,outermarks]{titlesec}

% \chapter, \subsection...: no additional code

\titleformat{\section}
  {\LARGE\sffamily\slshape}
  {\thesection}{1em}{}
\titlespacing{\section}
  {-6pc}{3.5ex plus .1ex minus .2ex}{1.5ex minus .1ex}

\titleformat{\paragraph}[leftmargin]
  {\sffamily\slshape\filright}
  {}{}{}
\titlespacing{\paragraph}
  {5pc}{1.5ex minus .1 ex}{1pc}

% 5+1=6, ie, the negative left margin in section

\widenhead{6pc}{0pc}

\renewpagestyle{plain}{}

\newpagestyle{special}[\small\sffamily]{
   \headrule
   \sethead[\usepage][\textsl{\chaptertitle}][]
           {}{\textsl{\chaptertitle}}{\usepage}}

\newpagestyle{main}[\small\sffamily]{
   \headrule
   \sethead[\usepage][\textsl{\thechapter. \chaptertitle}][]
           {}{\textsl{\thesection. \sectiontitle}}{\usepage}}

\pagestyle{special}

\begin{document}

---TOC

\pagestyle{main}

---Body

\pagestyle{special}

---Index
\end{document}
```

6.2. Standard Classes

Now follows, for your records, how sectioning commands of standard clases could be defined.

```
\titleformat{\chapter}[display]
```

```
  {\normalfont\huge\bfseries}{\chaptername\ \thechapter}{20pt}{\Huge}
\titleformat{\section}
  {\normalfont\Large\bfseries}{\thesection}{1em}{}
\titleformat{\subsection}
  {\normalfont\large\bfseries}{\thesubsection}{1em}{}
\titleformat{\subsubsection}
  {\normalfont\normalsize\bfseries}{\thesubsubsection}{1em}{}
\titleformat{\paragraph}[runin]
  {\normalfont\normalsize\bfseries}{\theparagraph}{1em}{}
\titleformat{\subparagraph}[runin]
  {\normalfont\normalsize\bfseries}{\thesubparagraph}{1em}{}

\titlespacing*{\chapter}      {0pt}{50pt}{40pt}
\titlespacing*{\section}      {0pt}{3.5ex plus 1ex minus .2ex}{2.3ex plus .2ex}
\titlespacing*{\subsection}   {0pt}{3.25ex plus 1ex minus .2ex}{1.5ex plus .2ex}
\titlespacing*{\subsubsection}{0pt}{3.25ex plus 1ex minus .2ex}{1.5ex plus .2ex}
\titlespacing*{\paragraph}    {0pt}{3.25ex plus 1ex minus .2ex}{1em}
\titlespacing*{\subparagraph} {\parindent}{3.25ex plus 1ex minus .2ex}{1em}
```

6.3. Chapter Example

A final example shows how to take advantage of the picture environment for fancy sectioning formats. Even with the simple tools provided by standard LaTeX you may create impressive titles but you may devise more elaborated ones with, for instance, pspicture (PSTricks package) or by including graphics created with the help of external programs.

```
\usepackage[dvips]{color}
\usepackage[rigidchapters]{titlesec}

\DeclareFixedFont{\chapterfont}{T1}{phv}{bx}{n}{11cm}

\newcommand{\chapterformat}[1]{%
  \put(0,190){\parbox[t]{300pt}{\Huge\sffamily\filright#1}}}

\titlespacing{\chapter}{0pt}{0pt}{210pt}
% Most of titles have some depth. The total space is
% a bit larger than the picture box.

\titleformat{\chapter}[block]
  {\begin{picture}(330,200)}
  {\put(450,80){%
     \makebox(0,0)[rb]{%
       \chapterfont\textcolor[named]{SkyBlue}{\thechapter}}}
   \put(0,230){%
     \makebox(0,0)[lb]{%
       \Huge\sffamily\underline{Chapter \thechapter}}}}
  {0pt}
  {\chapterformat}
  [\end{picture}]
```

(The exact values to be used depend on the text area, class, \unitlength, paper size, etc.)

7. Additional Notes

This part describes briefly some LaTeX commands, useful when defining sectioning titles.

7.1. Fixed Width Labels

The \makebox command allows to use fixed width label, which makes the left margin of the actual title (not the label) to lie in the same place. For instance (only the relevant code is provided):

```
\titleformat{\section}
  {..}
  {\makebox[2em]{\thesection}}{..}{..}
```

See your LaTeX manual for further reference on boxing commands.

7.2. Starred Versions

The use of sectioning commands in the starred version is strongly discouraged. Instead, you can use a markup oriented commands which are easy to define and modify, if necessary. Thus, you can test different layouts before choosing amongst them.

Firstly remember that if you say

```
\setcounter{secnumdepth}{0}
```

sections will be not numbered but they will be included in both toc and headers.

Now, let's suppose that you want to include some sections with a special content; for example, a section (or more) with exercises. We will use an environment named `exercises` whose usage is:

```
\section{A section}
Text of a normal section.

\begin{exercises}
\section{Exercises A}
Some exercises

\section{Exercises B}
Some exercises
\end{exercises}
```

The following definition supresses numbers but neither toc lines nor headers.

```
\newenvironment{exercises}
  {\setcounter{secnumdepth}{0}}
  {\setcounter{secnumdepth}{2}}
```

The following one adds a toc line but headers will remain untouched:

```
\newenvironment{exercises}
 {\setcounter{secnumdepth}{0}%
  \renewcommand\sectionmark[1]{}}
 {\setcounter{secnumdepth}{2}}
```

The following one updates the headers but there will be no toc line:

```
\newenvironment{exercises}
 {\setcounter{secnumdepth}{0}%
  \addtocontents{toc}{\protect\setcounter{tocdepth}{0}\ignorespaces}}
 {\setcounter{secnumdepth}{2}%
  \addtocontents{toc}{\protect\setcounter{tocdepth}{2}\ignorespaces}}
```

(I find the latter a bit odd in this particular example; the first and second options are more sensible. The `\ignorespaces` is not very important, and you needn't it unless there is unwanted space in the toc.)

That works with standard classes, but if you are using fancyhdr or titlesec to define headers you need an extra command. I will name it `\printthesection` and you could define it in the preamble as follows:

```
\newcommand{\printthesection}{\thesection\ }
```

If you define the header with titlesec, for instance:

```
\newpagestyle{myps}{
  \sethead{\printthesection\sectiontitle}{}{\usepage}}
```

then you have to write:

```
\newenvironment{exercises}
  {\setcounter{secnumdepth}{0}%
   \renewcommand{\printthesection}{}}
  {\setcounter{secnumdepth}{2}}
```

and so on.

As you can see, there are no `\addcontentsline`, no `\markboth`, no `\section*`, just logical structure. Of course you may change it as you wish; for example if you decide that these sections should be typeset in small typeface, include `\small`, and if you realize that you don't like that, remove it.

While the standard LaTeX commands are easier and more direct for simple cases, I think the proposed method above is far preferable in large documents.

7.3. Variants

Let's suppose we want to mark some sections as "advanced topics" with an asterisc after the label. The following code does the job:

```
\newcommand{\secmark}{}
\newenvironment{advanced}
  {\renewcommand{\secmark}{*}}
  {}
\titleformat{\section}
  {..}
  {\thesection\secmark\quad}{..}{..}
```

To mark the sections write

```
\begin{advanced}
\section\section{...}
...
\end{advanced}
```

That marks sections but not subsections. If you like being redundand and marking the subsection level as well, you must define it accordingly.

7.4. Rubber Lengths

Only in very rare ocasions, the material in a page fits exactly in the page heigth, 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 that 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 add 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 `minus` parts are used, thus producing an `Overfull \vbox` with text sticking out into the bottom margin.

The LATEX `\strech{`⟨*num*⟩`}` command is just a synonymous with `0pt plus `⟨*num*⟩`fill` followed by a `\relax`. (This `\relax` makes se an error if `\stretch` is used in the arguments allowing a `*` abbreviation, because they are expecting lengths without any additional stuff.)

7.5. Putting a Dot after the Section Title

Today this styling is not used, but formerly it was fairly common. The basic technique was described above, but here is a reminder:

```
\newcommand{\periodafter}[1]{#1.}
\titleformat{\section}
 {..}
 {\thesection}{..}{..\periodafter}
```

If you had to combine that dot with some command (perhaps an underlining), you can say:

```
\newcommand{\periodafter}[2]{#1{#2.}}
\titleformat{\section}
 {..}
 {\thesection}{..}{..\periodafter{\ul}} % \ul from soul package
```

8. Final comments

Version 2.0 had, among others, the following changes:

- The `...margin` shapes now behaves near the bottom margin as you can expect. The significant code has been fully rewritten.

- Some other shapes have been changed as well. For instance, `runin` has been modified to allow a horizontal rule (a `\titleline` to be precise) before the title. These changes could produce slight differences; in particular, spacing is a bit larger because now a strut is inserted automatically.
- The shapes are now in separate files to preserve memory and increase modularity. The `tss` extension means `t`itle`s`ec `s`ection and the `tsc` one `t`itle`s`ec `c`hapter. I apologize to MS-DOS users because of the long file names.
- Added compatibility with AMS classes.

And this version (2.1):

- A last minute change –but of course!– introduced a bug which added unwanted space before `\titleline`. The manual suggested

  ```
  \titleline{\hrulefill}%
  \vspace{1pt}%
  \titleline{\hrulefill}%
  ```

  as a way to insert to rules spaced by 1 pt, but in fact the space was a lot larger. That has been fixed and hence there is a backward uncompatibilit. As a solution, I strongly recommend to modify the commands to readjust the spacing, but if for some reason the files have to be preserved in the current state, you can create a style file with the following code:

  ```
  \def\ttl@line[#1]#2{%
    \par\hrule height \z@
    \nobreak
    \begingroup
      \parindent\z@
      \leftskip\z@
      \rightskip\z@
      \makebox[\hsize][#1]{\ttl@makeline{#2}}%
      \par
    \endgroup
    \hrule height \z@
    \nobreak}
  ```

  and then add an `\usepackage` loading it. Don't use that in new files.
- The `\part` command is now handled, but in a very restricted way.
- Only a `\footnote` command is allowed in titles, but it will work. However, the footnote mark is not removed from toc entries or running headings; for that use

  ```
  \usepackage[stable]{footmisc}
  ```

- The `wrap` shape should be considered as experimental in the sense that (a) it is not finished and some improvements will be necessary, and (b) part of the code is intended to be included in a new feature which I'd like to include in future releases—the possibility of measuring the longest line in the titles, say, for drawing rules whose length is exactly that. Actually, any new feature in any program should be considered mostly "experimental."
- The code has been tidied up and simplified. The number of auxiliary files has been reduced, and now there are `.tss` files only, shared by sections, chapters and parts. The three basic shapes, namely `display`, `hang` and `runin` has been merged into the main file.
- The code has been changed so that if `\newpage` is used in the ⟨*after*⟩ argument, the page will be correct in the contents.

9. The titletoc package

This package is a companion to the titlesec package and it handles toc entries. However, it's an independent package and you can use it alone. The philosophy is similar to that of titlesec—instead of hooking the commands as defined by standard LaTeX and classes, there are new commands which you can format the toc entries with in a generic way. That means that you have to learn just two new basic command and a couple of tools, no more. Paragraph format and fonts are set with the standard LaTeX commands like `\\`, `\makebox`, `\large`, `\itshape`, and so on, and entries are not shaped in any fashion because they are pretty free form.

The behaviour of entries defined with titletoc are different at some points to that created with the standard commands. In particular:

- Pages are never broken between entries if the first one is of an higher level than the second one as, for instance, between a section and a subsection. If both of them are of the same level, the break is allowed, and if the first is lower than the second, it's considered a good place for a page break.
- The symbols in the leaders are not centered but flushed right. That is usually more convenient.

I would like to note that no attempt to handle tocs can be complete because the standard LaTeX commands write directly some formatting commands which cannot be removed. This is particularly important in lists of figures and tables, and in the `\part` command.

9.1. A ten-minute guide to titletoc

Toc entries are treated as rectangular areas where the text and probably a filler will be written. Let's draw such an area (of course, the lines themselves are not printed):[12]

```
       +-------------------------------------+
       |                                     |
       |                                     |
       +-------------------------------------+
-left--                                       --right-
```

The space between the left page margin and the left edge of the area will be named ⟨*left*⟩; similarly we have ⟨*right*⟩. You are allowed to modify the beginning of the first line and the ending of the last line. For example by "taking up" both places with `\hspace*{2pc}` the area becomes:

```
           +---------------------------------+
        ___|                              ___|
       |                                 |
       +---------------------------------+
-left--                                       --right-
```

And by "clearing" space in both places with `\hspace*{-2pc}` the area becomes:

```
   +-----------------------------------------+
   |___                                      |___
       |                                         |
       +-----------------------------------------+
-left--                                       --right-
```

[12] I regret that the manual is so poorly formatted at the current stage. However, writting English text is for me a very hard task and the documentation takes me a lot; I think that the package availability is more important that a fine layout. In next releases I will improve the documentation.

If you have seen tocs, the latter should be familiar to you– the label at the very beginning, the page at the very end:

```
3.2  This is an example showing that toc
     entries fits in that scheme . . . .   4
```

All you need is to put these elements in the rigth way. If you have reserved the space with `\hspace*{-2pc}`, simply put a box 2 pc width containing the section label or page so that this space will be retrieved; this layout is used so often that two commands are provided which does that for you:

- `\contentslabel{`⟨*length*⟩`}` creates the space at the beginning and prints the section number.
- `\contentspage` creates a space at the end of length ⟨*right*⟩ and prints the page number aligned at the right.

Now, we are about to show the two basic commands:

```
\titlecontents{⟨section⟩}[⟨left⟩]{⟨above⟩}
               {⟨before with label⟩}{⟨before without label⟩}
               {⟨filler and page⟩}[⟨after⟩]
```

Here:

- ⟨*section*⟩ is the section name without backslash: `part`, `chapter`, `section`, etc. `figure` and `table` are allowed, too. (The backlash is ommitted because we are dealing with the concept and not the `\part`, `\section`, etc. macros themselves. Furthermore, `figure` and `table` are environments.)
- ⟨*above*⟩ is code for the global formatting of the entry. Vertical material is allowed. At this point the value of `\thecontsntslabel` (see below) is known which enables you to take decisions depending on its value (with the help of the ifthen package). You may use the titlesec `\filleft`, `\filright`, `\filcenter` and `\fillast` commands.
- ⟨*left*⟩ even if bracketted is currently mandatory and it sets the left margin from the left page margin.
- ⟨*before with label*⟩ is in horizontal mode and it will be used just before the entry title. As in `\titleformat`, the last command can take an argument with the title.
- ⟨*before without label*⟩ is like the above if there is, well, no label.
- ⟨*filler and page*⟩ is self explanatory. Fillers are created with the `\titlerule` command which is shared by that package and titlesec. However, when used in this context its behaviour changes a little to fit the needs of toc leaders.[13]
- And finally ⟨*after*⟩ is code following the entry for, say, vertical space.

When defining entries, use `\addvspace` to add vertical space, and `\\*` instead of `\\` for line breaks.

This commmand can be used in the middle of a document to change the format of toc/lot/lof entries at any point. The new format is written to the toc file and hence two runs are necessary to see the changes.

[13] For TEXperts, the default `\xleaders` becomes `\leaders`.

`\contentsmargin{⟨right⟩}`

The value set is used in all of sections. If you are wonder why, the answer is quite simple: in most of cases the ⟨*right*⟩ margin will be constant. However, you are allowed to change it locally in the ⟨*before*⟩ arguments.

The book class formats the section and subsection entries similarly (but not equally) to:

```
\contentsmargin{2.55em}
\titlecontents{section}
              [1.5em]
              {}
              {\contentslabel{2.3em}}
              {\hspace*{-2.3em}}
              {\titlerule*[1pc]{.}\contentspage}
\titlecontents{subsection}
              [3.8em]
              {} % note that 3.8 = 1.5 + 2.3
              {\contentslabel{3.2em}}
              {\hspace*{-3.2em}}
              {\titlerule*[1pc]{.}\contentspage}
```

Although standard classes use font dependent units (mainly em), it's recommended to use absolute ones (pc, pt, etc.)

9.2. And more

Strict tipographical rules state that full text lines shouldn't surpass the last dot of the leaders; ideally they should be aligned. Surprisingly enough, TEX lacks of a tool for doing that automatically—when you fill a box with leading dots, they can be centered in the box with the `\cleaders` primitive , "justified" with `\xleaders` or aligned with the outermost enclosing box with `\leaders`, but there is no way to align them with the "current" margin.

So, the only way to get a fine layout is by hand. To do that, you can use the an optional argument in the `\contentsmargin` command whose syntax in full is the following:

`\contentsmargin[⟨correction⟩]{⟨right⟩}`

The ⟨*correction*⟩ length is add to the ⟨*right*⟩ one in all of lines except the last one, where the leaders are placed. For instance, if the text lines are 6 pt longer than the last dots,[14] you should rewrite the `\contentsmargin` command to add a ⟨*correction*⟩ of 6 pt. Unlike the standard LATEX tools, the `\titlerule*` command has been designed so that the ⟨*correction*⟩ will have the minimum value possible.

`\thecontentslabel  \thecontentspage`

Contains the text with the label and the page with no additional formatting, except that written by the class.

`\contentslabel[⟨format⟩]{⟨space⟩}`
`\contentspage[⟨format⟩]`

As described above, but with different ⟨*format*⟩s. The defaults are just `\thecontentslabel` and `\thecontentspage`, respectively.

[14] Usefully, some dvi previewers allow to get the coordinates of the pointed location.

```
\contentspush{⟨text⟩}
```

Prints the ⟨*text*⟩ and increases ⟨*left*⟩ by the width of ⟨*text*⟩. That is similar to the hang shape of titlesec.

```
\titlecontents*{⟨section⟩}[⟨left⟩]{⟨above⟩}
                {⟨before with label⟩}{⟨before without label⟩}
                {⟨filler and page⟩}[⟨separator⟩][⟨end⟩]
```

This starred version groups the entries in a single paragraph. The ⟨*separator*⟩ argument is the separator between entries, and there is a further optional argument with an ending punctuation. For example, this document sets:

```
\titlecontents*{subsection}[1.5em]
  {\small}
  {\thecontentslabel. }
  {}
  {, \thecontentspage}
  [.---][.]
```

whose result is showed in the contents at the very beginning of this document. Note that the paragraph format must be written in the ⟨*above*⟩ argument.

```
\contentsuse{⟨name⟩}{⟨ext⟩}
```

Defines a new contents file with ⟨*ext*⟩ extension. The package performs

```
\contentsuse{figure}{lof}
\contentsuse{table}{lot}
```

```
leftlabels  rightlabels
```

These package options set how the labels are aligned in `\contentslabel`. Default is `rightlabels`. With `leftlabels` the default ⟨*format*⟩ for `\contentslabel` becomes `\thecontenstlabel\enspace`.

```
dotinlabels
```

With this package option, a dot is added after the label in `\contentslabel`.

9.3. Examples

```
\titlecontents{chapter}
              [0pt]
              {\addvspace{1pc}%
               \itshape}%
              {\contentsmargin{0pt}%
               \bfseries
               \makebox[0pt][r]{\huge\thecontentslabel\enspace}%
               \large}
              {\contentsmargin{0pt}%
               \large}
              {\quad\thepage}
              [\addvspace{.5pc}]
```

The chapter number is out at the edge of the page margin, in a font larger than the font of the title. If the chapter lacks of number (because, say, it is the preface or the bibliography) it is not boldfaced. The page number follows the title without fillers, but after an em-space.

```
\titlecontents{chapter}
             [3pc]
             {\addvspace{1.5pc}%
              \filcenter}
             {CHAPTER \thecontentslabel\\*[.2pc]%
              \huge}
             {\huge}
             {}  % That is, without page number
             [\addvspace{.5pc}]
```

The chapter title is centered with the chapter label on top of it. There is no page number.

9.4. Inserting a figure in the contents

The `\addtocontents` commands is still available and you may use it to perform special operation, like inserting a figure just before or after (or, in fact, in the middle) of an entry. Sadly, fragile arguments are not allowed and writing complex code could be a mess. The trick is to define a command to perform the required operations which is in turn written with `\protect`.

Let's suppose that we want to insert a figure before an entry.

```
\newcommand{\figureintoc}[1]{
  \begin{figure}
    \includegraphics{#1}%
  \end{figure}}
```

makes the dirty work.

In the place where a figure is inserted write:

```
\addtocontents{\protect\figureintoc{myfig}}
```

9.5. Marking entries with asteriscs

Let's now resume a problem explained in relation with titlesec: marking sections with asteriscs to denote an “advanced topic” unless that the star should be printed in the toc as well. Here is the code:

```
\newcommand{\secmark}{}
\newcommand{\marktotoc}[1]{\renewcommand{\secmark}{#1}}
\newenvironment{advanced}
  {\renewcommand{\secmark}{*}%
   \addtocontents{toc}{\protect\marktotoc{*}}}
  {\addtocontents{toc}{\protect\marktotoc{}}}
\titleformat{\section}
  {..}
  {\thesection\secmark}{..}{..}
\titlecontents{section}[..]{..}
  {\contentslabel[\thecontentslabel\secmark]{1.5pc}}{..}{..}
```

10. The titlesec philosophy

Once you have read the documentation it should be clear that this is not a package for the ocassional user who likes the standard layout and wants to make simple changes. This is a tool for the serious typographer that has a clear idea of what layout wants and don't have the skill to get it. If all you need are simple changes, you should consider using the tocloft package by Peter Wilson or the nice sectsty package by Rowland McDonnell. In particular, the latter allows to do simple changes in sections in a simple way without stuffing the package with a large amount of macros.