The titlesec and titletoc Packages*

Javier Bezos[†]

June 14, 2000

Contents

1. Introduction

This package is essentialy a replacement—partial or total—for the LaTeX macros related with sections—namely titles, headers and contents. The goal is to provide new features unavailable in current LaTeX; if you just want a more friendly interface than that of standard LaTeX but without changing the way LaTeX works you may consider using `fancyhdr`, by Piet van Oostrum, `sectsty`, by Rowland McDowell, and `tocloft`, by Peter Wilson, which you can made pretty things with.

Some of the new features provided are:

- Different classes and "shapes" of titles, with tools for very fancy formats. You can define different formats for left and right pages, or numbered and unnumbered titles, measure

[*] The titlesec package is currently at version 2.3.5. © 1998, 1999, 2000 Javier Bezos. The titletoc package is currently at version 1.2.2. © 1999, 2000 Javier Bezos. All Rights Reserved.

[†] Please, send bug reports (with a sample file), comments and suggestions to `jbezos@wanadoo.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: gloss (with José Luis Díaz), accents, polyglot, tensind, esindex.

the width of the title, add a new section level, use graphics, and many more. The Appendix shows a good deal of examples, so jump forward right now!

- Headers and footers defined with no `\...mark` intermediates, and perhaps containing top, first *and* bot marks at time. Top marks correctly syncronized with titles, and without uncompatibilities with the float mechanism. Decorative elements easily added, including picture environments.
- Pretty free form contents, with the possibility of grouping entries of differents levels in a paragraph or changing the format of entries in the middle of a document.

Titlesec works with the standard classes and with many others, including the AMS ones, and it runs smoothly with hyperref.[1]

As usual, firstly load the package in the standard way with `\usepackage`. Then, redefine the sectioning commands with the simple, predefined settings (see section "Quick Reference") or with the provided commands if you want more elaborated formats (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 by the simple settings and should be changed with the help of the "Advanced Interface."

2. Quick Reference

The easiest way to change the format is by means of a set of package options and a couple of commands. If you feel happy with the functionality provided by this set of tools, you needn't go further in this manual. Just read this section and ignore the subsequent ones.

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 use at least an option from them if you want this "easy setup."

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

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

```
big medium small tiny
```

Set the size of titles. 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.

```
raggedleft center raggedright
```

Control the alignment. Don't use `raggedright` if the document itself is already `\raggedright`.

2.2. Spacing

```
compact
```

This option is independent from those above and reduces the spacing above and below the

[1] However, be aware that the AMS classes reimplement the original internal commands. These changes will be lost here. The compatibility with hyperref has been tested with dvips only and it's an unsupported feature. Please, check that your version of hyperref is compatible with titlesec.

titles.

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.

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, etc. This scheme is intended to easy definitions, since in most of cases you will want to modify either spacing or format.[2] Note that that redefines existing sectioning commands, but does not define *new* ones. New sections are defined with `\titleclass`, as described below.

3.1. Format

A set of shapes is provided, which controls the basic distribution of elements in a title. The available shapes 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[3] 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.[4] 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.

[2] 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.)

[3] The label will be slightly displaced to the left if the title is two or more lines long and the `hang` shape is used, except with explicit `\\`.

[4] 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.

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 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. The limitations explained below related to `calcwidth` also applies here.

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

Note, however, that some shapes don't 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*⟩, whose possible values are those described above.
- ⟨*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 suppressed in the table of contents and running heads.
- ⟨*sep*⟩ is the horizontal separation between label and title body and must be a length. This space is vertical in `display` shape; in `frame` it 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 typeset material is in vertical mode with `hang`, `block` and `display`; in horizontal mode with `runin`. Otherwise is ignored.

```
\chaptertitlename
```

It defaults to `\chaptername` except in appendices where it is `\appendixname`. Use it instead of `\chaptername` when defining a chapter.

3.2. Spacing

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

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

- ⟨*left*⟩ increases the left margin, except in the `...margin`, and `drop` shape, where this parameter sets the title width, in `wrap`, the maximum width, and in `runin`, the indentation just before the title. With negative value the title overhangs.[5]

[5] 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 subsequent 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 of increasing the ⟨*right*⟩ margin with this 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.[7] 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 means of `\titleformat`, the simple settings, or `\titleclass`.

3.3. Spacing related 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.

`indentfirst nonindentfirst` (Package options)

By-pass the settings for all of sectioning commands.

`rigidchapters rubberchapters` (Package options)

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` where ⟨*aftersep*⟩ is the separation between title and text, as usual. Actually, the name is misleading because it applies not only to the default chapter, but to any title of top class. (More on classes below.)

`bottomtitles nobottomtitles nobottomtitles*` (Package options)

If the `nobottomtitles` is set, titles close to the bottom margin will be moved to the next

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

[7] They stand for n times `1ex plus .3ex minus .06ex` and `1ex plus .1ex`, respectively.

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.

The `nobottomtitles*` option provides more accurate computations but titles of `margin`, `wrap` or `drop` shapes could be badly placed. Usually, you should use this starred version.

```
aftersep largestsep    (Package options)
```

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.

3.4. Rules

The package includes some tools for helping in adding rules and other stuff below or above the title. Since the margins in titles may be modified, these macros take into account the local settings to place rules properly. They also take into account the space used by marginal titles.

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

The `\titleline` command allows inserting a line, which may contain text and other “horizontal” material. 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}`.[8]

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.

```
calcwidth    (Package option)
```

The `wrap` shape has the capability of measuring the lines in the title to format the

[8] The default is the `s` parameter of the `\makebox` command.

paragraph. This capability may be extended to other three shapes—namely `display`, `block` and `hang`—with this package option. The length of the longest line is returned in `\titlewidth`.

As far as TEX is concerned, any box is considered typeset material. If the box has been enlarged with blank space, or if conversely a box with text has been smashed, the value of `\titlewidth` may be wrong (as far as humans are concerned). The `hang` shape, for instance, uses internally such a kind of boxes, but in this case this behaviour is desired when the title is flushed right; otherwise the `block` shape produces better results. In other words, using boxes whose natural width has been overriden may be wrong. Further, some commands may confuse TEX and stop parsing the title. But if you stick to text, `\\` and `\\[...]` (and it's very unlikely you might want something else), there will be no problems. Another important point is that the ⟨*before*⟩, ⟨*label*⟩, ⟨*sep*⟩, and ⟨*title*⟩ parameters (but not ⟨*after*⟩) are evaluated twice at local scope; if you increase *globally* a counter, you are increasing it twice. In most of cases, placing the conflicting assignment in the ⟨*after*⟩ parameter will be ok.

We now describe a variant of `\titleline`.

`\titleline*[`⟨*align*⟩`]{`⟨*horizontal material*⟩`}`

The text will be enclosed first in a box of width `\titlewidth`; this box will be in turn enclosed in the main box with the specified alignment.

3.5. 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,

```
\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 accomplished with:

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

3.6. Other Package Options

`newparttoc oldparttoc` (Package options)

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 only if `\part` has been redefined.)

`clearempty` (Package options)

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

3.7. Extended Settings

The first argument of `\titleformat` has an extended syntax which allows to set different formats depending on the context.[9] This argument can be a list of key/value pairs in the form:

⟨*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`. A valueless key. It's not necessary unless you want to set different numbered (without this key) and unnumbered (with `numberless`) 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 variants, 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`. 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.

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

[9] The keyval package is required for making use of it.

3.8. Creating new levels

While the shapes and the like modifies the behaviour of titles related to the surrounding text, title classes allows to change the generic behaviour of them. With the help of classes you may insert, say, a new `subchapter` level between `chapter` and `section`, or creating a scheme of your own. *Making a consistent scheme and defining all of related stuff like counters, macros, format and spacing is left to the responsability of the user.* There are three classes: `page` is like the book `\part`, in a single page, `top` is like `\chapter`, which begins a page and places the title at the top, and `straight` is intended for titles in the middle of text.[10]

```
\titleclass{⟨name⟩}[⟨top-num⟩]{⟨class⟩}
\titleclass{⟨name⟩}{⟨class⟩}[⟨sup-level⟩]
```

If you don't use the optional arguments, you just change the ⟨*class*⟩ of ⟨*name*⟩. For example:

```
\titleclass{\part}{straight}
```

makes `part` of `straight` class.

When the first form is used in full, the ⟨*name*⟩ title is considered the top level, with ⟨*top-num*⟩ level number. It should be used only when creating sectioning command from scratch. When the second form is used, the level number is the following of ⟨*sup-level*⟩. For example:

```
\titleclass{\subchapter}{straight}[\chapter]
\newcounter{subchapter}
\renewcommand{\thesubchapter}{\Alph{subchapter}}
```

creates a level under chapter (some additional code is shown as well). If the chapter level is 0, then the subchapter one is 1; the levels below are increased by one (section is 2, subsection is 3, and so on).

There are two sectioning commands which perform some extra actions depending of its name and ignoring the class:

- `\chapter` logs the string defined in `\chaptertitlename` and the matter is taken into account.
- `\part` does not encapsulates the label in the toc entry, except if you use the `newparttoc` option.

`loadonly` (Package option)

Now, you want create your sectioning commands from scratch. This package option ignores any previous definitions, if any, and hence removes the possibility of using the option described in "Quick Reference." Then you use the `titlesec` tools, and define the corresponding counters and labels.

4. Additional Notes

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

4.1. 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):

[10] There is an further class named `part` to emulate the article `\part`, but you should not use it at all. Use the `straight` class instead.

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

See your LaTeX manual for further reference on boxing commands.

4.2. Starred Versions

The use of sectioning commands in the starred version is strongly discouraged. Instead, you can use a set of 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 further refinement to kill the section number. In titlesec that's accomplished with `\ifthesection` (see below).

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.

4.3. 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{...}
...
\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.

4.4. 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
```

5. Page Styles

As a part of this package a set of commands are provided for one-step headline and footline settings. Those page styles has been designed to work with sectioning related information; you shouldn't use to create headers with the first an last entry of a page in a dictionary, for example (you could, but it's more complicated than with fancyhdr).

It works with standard classes and many others; however, additional LaTeX layout adjustments could be necessary in some of them (for instance the AMS classes).[11] For using them, set the following package option:[12]

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

[12] The way pagestyles works have been reimplemented in full. However, in most of cases you won't notice it at all.

```
pagestyles
```

5.1. Definining Page Styles

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

Defines a new style or redefines an existing one, named ⟨*name*⟩. For convenience, end of lines are ignored, and you needn't "hide" them with `%`.[13]

⟨*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):

```
\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 allowed, too.

```
\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

[13] Even if marks should be reserved to pass information to headers, some classes, like AMS ones, introduce additional code. When redefining the `plain` style of AMS classes, you must add the following line: `\global\topskip\normaltopskip`.

optional arguments are given the odd-page settings applies to both even and odd pages. A couple of starred variant (`\setfoot*`, `\sethead*`) reverses the order of the settings for even pages (obviously, if you use them, the optional arguments are not allowed). In the arguments of `\sethead`/`\setfoot` and because of its one-step mechanism, we must distingish between two groups of commands. The first one are those saved when the marks are sent, with information related to sections, and includes:

- `\thechapter`, `\thesection`...
- `\chaptertitle`, `\sectiontitle`... which print the chapter, section... title.
- `\ifthechapter{`⟨*true*⟩`}{`⟨*false*⟩`}`, `\ifthesection{`⟨*true*⟩`}{`⟨*false*⟩`}`... which expands to ⟨*true*⟩ except if the corresponding title lacks of label or there is no title yet after the superior level (for example, between `\chapter` and a subsequent `\section`).
- Any other command or value "registered" as "mark." (See below.)

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

- `\thepage`.[14]
- Any other command not included in the previous items.

`\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`,
- `\ifthechapter` and `\ifthesection` are valid tests, and
- Marks are updated in `\chapter` and `\section`.

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. 4.2) but if you need them, you may use them as in:

```
\chapter*{My Chapter}
\chaptermark{My Chapter}
```

5.2. Additional Settings

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

Makes the head/foot lines wider. The extra width is asymetrically added and hence the two [four] arguments. Like `\sethead`, the starred version reverses the setting for even pages—for example, `\widenhead*{0pt}{6pc}` is the same as `\widenhead[6pc][0pt]{0pt}{6pc}`.

[14] The old `\usepage` command is no longer necessary, but it's preserved.

5.3. Marks

```
outermarks innermarks topmarks botmarks   (Package options)
```

`innermarks` is the default in LaTeX, with bottom marks in even pages and first marks in odd pages. More convenient is `outermarks`, with top marks in even pages and bottom marks in odd pages; it's the more usual system in tecnical works and it's 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).

```
\bottitlemarks  \toptitlemarks  \firsttitlemarks \nexttoptitlemark
\outertitlemarks  \innertitlemarks
```

One of the cooler features added to page styles is the posibility of selecting which marks are used in subsequent macros. These commands set from which mark get the values of `\thesection`, `\sectiontitle`, etc. You may use freely in your headers;[15] just for fun, the following header shows the section label from three of these marks:

```
\newpagestyle{funny}{
  \sethead{Top is \toptitlemarks\thesection}
          {First is \firsttitlemarks\thesection}
          {Bot is \bottitlemarks\thesection}}
```

`\outertitlemarks` is the default value when the `outermarks` package option is set, `toptitlemarks` when `topmarks` is set, and so on. `\innertitlemarks` is the default value of LaTeX and of this package if none of those option is used. A further set of marks is provided with `\nexttoptitlemarks` which provides a bottom mark with the values of the top mark of the next page (straight class only); by picking some value from it and comparing with the corresponding value of the `\bottitlemarks` you may know if a section continues in the next page. Of course, these commands will be rarely used.

```
\newtitlemark{⟨macro-name⟩}
\newtitlemark*{⟨variable-name⟩}
```

Add a macro or a variable to the list of saved "marks." The macro must lack of parameters, and the variable should be in TeX form (which is the same in lengths but not in counters).

5.4. A couple of examples

Because of the way marks are handled, special constructions like this are allowed:

```
\newpagestyle{main}[\small\sffamily]{
  \sethead [\textbf{\thepage}]
           [\textsl{\chaptertitle}]
           [[\toptitlemarks\thesection--\bottitlemarks\thesection]
           {\toptitlemarks\thesection--\bottitlemarks\thesection]}
           {\textsl{\sectiontitle}}
           {\textbf{\thepage}}}
```

As you can see, the range of sections in a page `\thesection` is printed in both left and right heads. Of course, this example should be fine tuned to collapse the ranges if there is a single section in a page, but it gives a hint of how gettting stunning headers.

In this document, the following styles are used:

[15] Not exactly: top marks must not be used in the page where a chapter begins, except by overriding the default definition and making it of `top` class.

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

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

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

6. Final comments

- A last minute change —but of course!— introduced a bug in version 2.0 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 was fixed in version 2.1 and hence there is a backward uncompatibility. 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.

- 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 page styles has been reimplemented in full in version 2.3. Most of previous definition will work with the current code, even if many things have changed. In particular, headers using just `\the...` and `\...title` commands, besides `\usepage` and formatting commands still work. Now, pagestyles should be loaded explicitly with the `pagestyles` option, but if it is not given and the three basic commands are used, the page styles macros are loaded with a warning. The new code overcomes the limitations of previous releases, fixes some bugs (the uncompatibility between top marks and floats, for example), and adds new features.

7. Contents: 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, and that you have access to new features. Paragraph format and fonts are set with 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.[16]

7.1. 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):[17]

```
       +------------------------------------+
       |                                    |
       |                                    |
       +------------------------------------+
-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:

```
           +--------------------------------+
        ___|                             ___|
```

[16] But some of these issues are fixed by titlesec.

[17] 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.

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

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

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

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 `\thecontentslabel` (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.[18] You might prefer a `\hspace` instead.

[18] For TeXperts, the default `\xleaders` becomes `\leaders`.

- 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 command 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.

```
\contentsmargin{⟨right⟩}
```

The value set is used in all of sections. If you are wondering 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 using absolute ones (pc, pt, etc.)

7.2. 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 added 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 dot, you should rewrite the `\contentsmargin` command to add a ⟨*correction*⟩ of 6 pt.[19] 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.

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

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

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

```
\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⟩]
            or ...{⟨filler and page⟩}[⟨separator⟩][⟨end⟩]
            or ...{⟨filler and page⟩}[⟨begin⟩][⟨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.

Let's explain how the optional arguments works. First note that the number of them determines their meaning—since there should be a separator between entries this one is always present; on the other hand, ⟨*begin*⟩ is rarely used and hence it has the lowest "preference." The simplest case is when the titles are of the same level; in this case the ⟨*sep*⟩ and the ⟨*end*⟩ parameters (which default to empty) are inserted between consecutive entries and at the end of the block, respectively. ⟨*before*⟩ is executed just once at the very beginnig of the block and its declarations are local to the whole set of entries.

Let's now suppose that we want to group entries of two levels; in this case a nesting principle applies. To fix ideas, we will use section and subsection. When a subsection entry begins after a section one, ⟨*before*⟩ is executed and ⟨*begin*⟩ of subsection is inserted, which should contain text format only. Then subsections are added inserting separators as explained above. When a section arrives, the ending punctuation of subsection and the separator of section is added (except if the block is finished by a subsection, where the ending of section is added instead). We said "after a section" because a subsection never begins a block.[20] The subsection entries are nested inside the section ones, and declarations are again local.

An example will illustrate that.

```
\titlecontents*{section}[0pt]
  {\small\itshape}{}{}
  {}[ \textbullet\ ][.]

\titlecontents*{subsection}[0pt]
  {\upshape}{}{}
  {, \thecontentspage}[ (][. ][)]
```

[20] In rare cases that could be necessary, yet.

produces something similar to:

The first section • *The socond one* • *The third one* (A subsection in it, 1. Another, 2) • *A fourth section* (A subsection in it, 1. Another, 2).

`\contentsuse{⟨name⟩}{⟨ext⟩}`

Makes titletoc aware of the existence of a contents file with ⟨*ext*⟩ extension. The package performs

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

`leftlabels rightlabels` (Package options)

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` (Package option)

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

7.3. Partial TOC's

`\startcontents[⟨kind⟩]`

At the point where this command is used, a partial toc begins. The ⟨*kind*⟩ argument allows differents sets of tocs and it defaults to `default`. These sets may be intermingled, but usually will be nested. For example, you may want two kinds of partial tocs: by part and by chapter (besides the full toc, of course). When a part begins, write `\startcontents[parts]`, and when a chapter `\startcontents[chapters]`. This way a new toc is started at each part and chapter.[21]

`\stopcontents[⟨kind⟩]`
`\resumecontents[⟨kind⟩]`

Stops the partial toc of ⟨*kind*⟩ kind, which may be resumed. Since partial contents are stopped by `\startcontents` if necessary, you won't use those macros very often.

`\printcontents[⟨kind⟩]{⟨prefix⟩}{⟨top⟩}{⟨init-code⟩}`

Print the current partial toc of ⟨*kind*⟩ kind. The format of the main toc entries are used, except if there is a ⟨*prefix*⟩. In such a case, the format of ⟨*prefix*⟩⟨*level*⟩ is used, provided it is defined. For example, if prefix is `l` and the format of `lsection` is defined, then this definition will be used; otherwise, the format is that of `section`. The ⟨*top*⟩ paremeter sets the top level of the tocs—for a part toc it would be `0` (chapter), for a chapter toc `1` (section), and so on. Finally, ⟨*init-code*⟩ is local code for the current toc; it may be used to change the `tocdepth` value or the `\contentsmargin`, for instance.

A simple usage might look like (provided you are using titlesec as well):

[21] *All* partial tocs are stored in a single file with extension `.ptc`.

```
\titleformat{\chapter}[display]
  {...}{...}{...}  % Your definitions come here
  [\vspace*{4pc}%
   \startcontents
   \printcontents{l}{1}{\setcounter{tocdepth}{2}}]
```

The included entries are those in the level from 1 to 2 inclusive.

7.4. 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.

7.5. Inserting a figure in the contents

The `\addtocontents` command is still available and you may use it to perform special operation, like inserting a figure just before or after 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 in turn is 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}}
```

7.6. 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}}{..}{..}
```

8. The titlesec philosophy

Once you have read the documentation it should be clear that this is not a package for the casual 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. No attempt is made to improve your taste in section formatting.

9. Appendix

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

■

$\boxed{\textit{10}}$ 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 11

A framed title

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

■

---

12. A Ruled Title

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

■

13

Another Ruled Title

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

■

• •

14 The length of the “rule” above is that of the longest line in this title increased by two picas

• • • • • • • • • • • • • • • • • • •

15 This one is shorter

```
\titleformat{\section}[block]
  {\filcenter\large
   \addtolength{\titlewidth}{2pc}%
   \titleline*[c]{\titlerule*[.6pc]{\tiny\textbullet}}%
   \addvspace{6pt}%
   \normalfont\sffamily}
  {\thesection}{1em}{}
\titlespacing{\section}
  {5pc}{*2}{*2}[5pc]
```

■

SECTION 16

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}
```

■

17. 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.

18. 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}
```

■

§ 19. 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 20

The Title

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

■

CHAPTER XXI

The Title

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

3.1. A full example

Now an example of a complete title scheme follows. (The page style commands, namely `\widenhead` and `\(re)newpagestyle`, are described below, under “Pages Styles”.)

```
\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}
```

3.2. Standard Classes

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

```
\titleformat{\chapter}[display]
  {\normalfont\huge\bfseries}{\chaptertitlename\ \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}
```

3.3. 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.)