

---

# titlesec 与 titletoc 宏包使用说明

## The titlesec and titletoc Packages

---

ChinaT<sub>E</sub>X Documentation Workshop



Released by ChinaT<sub>E</sub>X Documentation Workshop.

June, 2011

翻译: Gao Hu

# titlesec 和 titletoc 宏包<sup>\*</sup>

宏包作者: Javier Bezos<sup>†</sup>  
翻译: GaoHu

2007-08-12

## 目录

1. 简介 1
2. 速成参考 2
  - 2.1. 格式, 2.—2.2. 空白, 3.—2.3. 工具, 3.
3. 高级接口 4
  - 3.1. 格式, 4.—3.2. 空白, 6.—3.3. 空白相关工具, 6.—3.4. 关于行, 8.—3.5. 页面样式, 10.—3.6. 中断, 10.—3.7. 其他宏包选项, 11.—3.8. 扩展设置, 12.—3.9. 创建新的级别以及改变文档类, 13.
4. 附加须知 14
  - 4.1. 固定宽度标签, 14.—4.2. 带星号的版本, 15.—4.3. 变量, 16.—4.4. 在节标题后包含一个句 = 点号, 16.
5. 页面样式 17
  - 5.1. 定义页面样式, 17.—5.2. 额外的设置, 19.—5.3. 运行带有浮动体的页眉 Running heads with floats, 20.—5.4. 标记, 20.—5.5. A couple of examples, 21.
6. 最后的注释 22
7. 目录: titletoc 宏包 22
  - 7.1. titletoc 10 分钟速成指导, 23.—7.2. 更多内容, 26.—7.3. Partial TOC's, 28.—7.4. 部分列表, 29.—7.5. 示例, 30.—7.6. 在内容中插入图片, 30.—7.7. 使用星号标注条目, 31.
8. titlesec 的原理 31
9. 附录 31
  - 9.1. 一个完整的例子, 35.—9.2. 标准文档, 36.—9.3. 章的例子, 36.

## 1. 简介

这个宏包主要是为了部分程度或者完全替代 L<sup>A</sup>T<sub>E</sub>X 中节相关的宏包（也就是指标题、页眉和内容）。宏包的目的是提供现有 L<sup>A</sup>T<sub>E</sub>X 不具备的新特性；如果

---

<sup>\*</sup>titlesec 宏包现在是 2136 版本。© 1998–2007 Javier Bezos。titletoc 宏包现在是 1.6 版本。© 1999–2007 Javier Bezos。保留所有权利。

<sup>†</sup>请在 <http://www.tex-tipografia.com/contact.html> 报告 bug、提供评论以及建议。英语并不是我的强项，因此如果你发现手册中的错误，请联系我。同作者的其他宏包: gloss (和 José Luis Díaz 合作), enumitem, accents, tensind, esindex, dotlessi。

你希望拥有一个比标准 L<sup>A</sup>T<sub>E</sub>X 更友好的接口，同时又不想改变 L<sup>A</sup>T<sub>E</sub>X 的工作方式，那么你可以考虑使用 Piet van Oostrum 的 fancyhdr，Rowland McDonnellis 的 sectsty，以及 Peter Wilson 的 tocloft 宏包，使用这些宏包，你可以产生相当漂亮的文档。

如下是本宏包所提供的一些新特性：

- 不同的文档类和“形状”的标题，同时还有能够设计精美格式的工具。你可以为左右页面定义不同的格式，以及带编号和不带编号的标题，调整标题的宽度，添加新级别的节，使用图形等等。附录列举了许多很不错的例子，不妨现在就欣赏一下！
- 不需要使用形如 \...mark 的中间形式来定义页眉和页脚，同时也可能包含顶部、首条以及底部的标注。顶部标记可以与标题正确同步，而且不会和浮动体结构产生冲突。可以很轻松的添加装饰元素，甚至是图片环境。
- 相当自由的内容形式，同时提供了组合同一段落段落中不同级别的条目，或者更改在文档中间的条目的格式的可能性。

titlesec 可以与标准文档类以及许多其他内容协同工作，比如 AMS 系列，而且可以与 hyperref 宏包无缝结合。<sup>1</sup>

通常情况下，按照标准的 \usepackage 方式加载这个宏包。然后，使用简单的，预定义的设置（请参照“速成参考”）重新定义节命令，或者使用提供的命令来定义你需要的更加详细的格式（参照“高级接口”）。在第二种情况下，你只需要重新定义你要使用的命令。两种方法可以同时使用，但是因为 \part 通常是以一种非标准的方式实现的，在简单设置模式下我们保持其不变，如果需要，可以借助于“高级接口”来进行调整。

## 2. 速成参考

更改格式的最简单的办法是使用宏包提供的一系列选项和命令。一旦你发现这些工具提供的功能已经可以让你满意，那么就不需要继续深入阅读这本手册。那么，只需阅读本节，忽略后续的内容吧。

### 2.1. 格式

有三组选项用于控制字体、大小、对齐。并不需要设置所有选项，因为每组选项都设置有默认值；尽管如此，如果你希望使用这种“简单设置”，那么至

<sup>1</sup>尽管如此，请注意 AMS 类重新实现了原有的内部命令。这些改变在此处会失效。与 hyperref 的兼容性已经对 dvips, dvipdfm 以及 pdftex 做过测试，但是这些还是不支持的特性。因此，请检查你的 hyperref 版本是否与 titlesec 兼容。

少要使用其中的一个选项。

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

选择相应的字体/衬线/样式。默认为 bf。

```
big medium small tiny
```

设置标题的大小。默认为 big，这是标准文档类的大小。使用 tiny 的话，节（章除外）会按照文本的大小打印。medium 和 small 是介于两者之间的布局。

```
raggedleft center raggedright
```

控制对齐方式。

## 2.2. 空白

```
compact
```

这个选项独立于上述各条，用于减少标题之上和之下部分的空白宽度。

## 2.3. 工具

```
\titlelabel{<label-format>}
```

调整节、小节等的标签格式。label 可选的值为 \thetitle、\thesection、\thesubsection 等。标准类中的默认值为

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

你也可以在简单地在计数器之后添加一个点分隔符：

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

本文就是这么处理的。

```
\titleformat*{<command>}{<format>}
```

这个命令允许变更分节命令的 <format>，比如下面的例子：

```
\titleformat*{\section}{\itshape}
```

### 3. 高级接口

我们提供了两个命令来更改标题的格式。第一个主要针对“内部”格式，比如，样式、字体、标签等等。第二个则定义了“外部”格式，比如，前后的空白、缩进等等。因为大部分情况下你可能想要更改空白或者格式，因此这个模式被设计成可以很简单的进行重定义。<sup>2</sup>这些命令重新定义现有的分节命令，但是并不是创建新的命令。新的分节级别可以通过 `\titleclass` 来增加（下文我们将会讲到这个命令），之后就可以使用这里描述的命令来进行设置。

#### 3.1. 格式

我们提供了一些形状的集合，用于控制标题中元素的基本形态。可用的形状有：

`hang` 默认值。带有一个悬挂标签。（与标准的 `\section` 类似。）

`block` 将整个标题输出在一个块（一个段落）中，不带有额外的格式。对于居中输出的标题<sup>3</sup>和特殊的格式（包括像 `picture`, `pspicture` 等的图片工具）很有用。

`display` 在单独的段落中放置标签。（就像标准的 `\chapter` 一样）。

`runin` 一个融合标题。（类似于标准的 `\paragraph`）。

`leftmargin` 在左边缘放置标题。在一页底部的标题会被移动到下一页，并不会溢出到底部边界，因此大的标题有可能产生 `underfull` 页面。<sup>4</sup>这种情况下，你需要使用 `\raggedbottom` 或者使用下文描述的 `nobottomtitles` 选项来增强页面元素的健壮性。因为此处使用的机制独立于边界盒子的机制，因此他们可以重叠使用。不建议采用的 `margin`，虽然它的功能相似。

`rightmargin` 同 `leftmargin` 但是体现在右边界。

`drop` 假如第一个段落比标题长，那么折叠标题周围的文本（否则，就各行其道）。`leftmargin` 中的说明在这里也适用。

`wrap` 与 `drop` 很相似。唯一的区别在于 `drop` 模式下，为标题保留的空白是固定的，在 `warp` 模式下，标题长度自动调整为与最长的行一致。下文中将要叙述的 `calcwidth` 的限制在此处也适用。

<sup>2</sup>信息是从类的分节命令中“抽取”出来的，但是章和部分除外。我们假设标准的 `\startsection` 为：如果没有使用该宏定义节，那么就提供任意的格式和空白，你稍后可以改变这些内容。（不幸的是，没有办法去捕获章或者部分的这种格式，或者是一种标准类使用的相似的格式。）

<sup>3</sup>如果标签长于两行或者使用了 `hang` 来控制形状，那么标签将会轻微的偏左放置，除非强制显式使用了 `\\`

<sup>4</sup>尽管如此，与标题相隔几行之后的浮动体会与此处使用使用的页面中断冲突，因此有时导致标题溢出。

frame 同 display 类似，但是标题会有边框包围。

尽管如此，仍然有一些样式在章和部分中是没有意义的。

```
\titleformat{<command>}[<shape>]{<format>}{<label>}{<sep>}{<before>}{<after>}
```

含义：

- `<command>` 是要重定义的分节命令。比如 `\part`, `\chapter`, `\section`, `\subsection`, `\subsubsection`, `\paragraph` 或者 `\subparagraph`。
- 通过 `<shape>` 来设置段落的形状，可以设置为上文讨论的值。
- `<format>` 是将要应用到整个标题的格式 — 包括标签和文本。这部分可以包含垂直元素（以及某些形式的水平元素），这些元素会打印在标题上方的空白之后。
- 标签在 `<label>` 中定义。如果在那一级别节中没有分节标签，你可以把该值留空，但是需要注意，如果这样做的话，并不会抑制目录以及后续页眉的数字编号。
- `<sep>` 定义标签和标题正文之间的水平距离，必需为一个长度值（不可留空）。在 display 样式中，这个距离为竖直距离；在 frame 样式中，这个距离为文本到边框的距离。如果使用了带有星号的分节命令，那么 `<label>` 和 `<sep>` 都会被忽略。如果你使用 `picture` 或者类似的环境，那么请设置这个参数为 0 pt。
- `<before>` 用于处理标题正文之前的内容。最后一个命令可以带有一个参数，作为标题的文本。<sup>5</sup> 尽管如此，如果使用宏包的 `explicit` 选项，那么需要显式的通过 `#1` 给出标题（见下文）。
- `<after>` 指定跟在标题正文之后的内容。如果使用 `hang`, `block` 或者 `display` 那么打印输出为垂直模式；如果使用 `runin` 以及 `letfmargin` (`[2.7]` 后者，在段落的开始) 的模式，那么打印输出为水平模式。其他的情况会被忽略。

```
\chaptertitlename
```

默认情况下为 `\chaptername`，在附录中为 `\appendixname`。在定义章时，请使用该变量代替 `\chaptername`。

<sup>5</sup> 请记住只有在段落与段落之间改变字号才是安全的，如果在文本中间改变字号，那么请通过分组括号的形式进行本地化小范围的更改；否则前导的文字就会有问题 — 要么过大，要么过小。

### 3.2. 空白

```
\titlespacing*{<command>}{<left>}{<beforesep>}{<aftersep>}{<right>}
```

带星的版本会取消紧跟标题之后的段落的缩进；但是如果是在 drop, wrap 以及 runin 模式下，很可能就无效了。

- `<left>` 增加左缩进。有几种例外，在 ...margin 和 drop 模式下，该参数设置标题的宽度；在 wrap 模式下，设置最大宽度；在 runin 模式下设置标题前的缩进。如果使用一个负值，那么启用标题悬垂。<sup>6</sup>
- `<beforesep>` 标题前的垂直分隔。
- `<aftersep>` 标题和文本之间的分隔。如果使用 hang, block 和 display 为垂直距离；如果使用 runin, drop, wrap 和 ...margin 为水平距离。通过把这个参数设置为负值，你可以定义一个比 `\parskip` 小的有效的空白距离。<sup>7</sup>
- `<right>` 参数是可选的。使用该参数，可以使得 hang, block 和 display 样式拥有增加右边距的可能。

如果你不喜欢输入完整的命令参数，甚至是 plus 和 minus 参数，那么可以使用宏包提供的缩写 `*n`。对于 `<beforesep>` 参数，等价于带有部分扩展性和少量收缩性的 `n ex`。`<aftersep>` 类似于带有部分扩展性（稍小一点）而没有收缩性的 `n ex`。<sup>8</sup>因此，你可以按照如下书写

```
\titlespacing{\section}{0pt}{*4}{*1.5}
```

`\beforetitleunit` 和 `\aftertitleunit` 的长度作为\* 设置中的单元长度，如果你不喜欢预定义的值，那么你可以自己定义他们（比如在排版结构上做轻微的修改。）

注意 `\titlespacing` 不能和 `\chapter` 或者 `\part` 一起工作，除非你使用 `\titleformat`（一种简单的设置方法）或者 `\titleclass` 来更改他们的标题格式。`\titlespacing` 中的参数必需是尺度；`\stretch` 包含了一个命令，因此会引发一个错误。

### 3.3. 空白相关工具

这些命令是作为 `\titleformat` 和 `\titlespacing` 的工具提供的。

<sup>6</sup>这个参数和 `\@startsection` 中的 `<indent>` 不同，后者不能正确的工作。在后者中使用一个负值或者 `<indent>` 比标签的宽度大，那么标题的第一行会从页面外边缘开始（这和我们预期的一样），但是后续的行不会；糟糕的是，这些行会在右边距的附近发生缩短行为。

<sup>7</sup>请参考 Goossens, Mittelbach 和 Samarin 的: The L<sup>A</sup>T<sub>E</sub>X Companion, Addison Wesley, 1993, 第 25 页。

<sup>8</sup>这两个命令分别代表  $n$  次 `1ex + .3ex - .06ex` 以及 `1ex + .1ex`。



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

不同的 `\ragged...` 命令，稍微有一些差异。需要强调的是，`\ragged...` 命令会取消 `\titlespacing` 设置的左空白和右空白。`\fillast` 调整除最后一行之外的段落样式，最后一行会居中显示。<sup>9</sup>这些命令在 `frame` 标签中也同样可以工作。

`\filinner` 和 `\filouter` 以及 `\filleft` 或者 `\filright` 的行为要视具体页面而定。由于  $\text{\TeX}$  的页面中断是异步的，因此这些命令只可以在 `\chapter` 中使用。如果你需要一个根据页面来设置不同格式的通用工具，那么请参考下文的“扩展设置”。

`\wordsep`

当前字体的单词间隔。

`indentafter noindentafter` (Package options)

忽略当前所有分节命令的设置。<sup>10</sup>

`rigidchapters rubberchapters` (Package options)

使用 `rigidchapters` 时，章标题的空白总是一样的，此时 `\titlespacing` 中的 `\aftersep` 不再像上文所描述的表示从标题文字的底部到文本正文之间的空白距离，而是从标题文字的顶部开始算起。比如，`\beforesep` + `\aftersep` 现在就是从页面的顶部到正文的固定的距离。默认的情况下，`\aftersep` 是标题和正文的间距，`rubberchapters` 为其默认值。实际上，这个名字有点误导的嫌疑，因为他并不仅仅作用于默认的章，而作用于顶级类的任何标题。（下文有更多关于文档类的讨论）。

`bottomtitles nobottomtitles nobottomtitles*` (宏包选项)

如果设置了 `nobottomtitles`，那么在页面底部的标题会被移动到下一页，这会使页面边缘看起来不一致。最小的不移动标题的下边距可以通过

`\renewcommand{\bottomtitlespace}{\length}`

来进行设置（大概的精度），默认的宽度值为 `.2\textheight`。前几页中的一个下边界重排的例子是通过设置这个值为 0 pt 得到的。默认值是 `bottomtitles`，这个选项简单的把 `\bottomtitlespace` 设置为一个负值。

<sup>9</sup>显然，这是一个奇怪的名字，但是它很简短啊。

<sup>10</sup>之前是使用 `indentfirst` 以及 `nonindentfirst`，现在不再建议使用。



nobottomtitles\* 提供了更加精确的计算，但是 margin, wrap 或者 drop 的形状可能会被简单的进行处理。通常情况下，你应该使用带星号的版本。

`aftersep largestsep` (Package options)

默认情况下，`<aftersep>` 用于表示两个连续的标题之间的距离。有时候这并不是我们希望的行为，特别是 `<beforesep>` 的空白比起 `<aftersep>` 过大的时候（否则，默认的情况看起来也不错）。使用 `largestsep` 来指定，那么就会使用二者中间的最大值。默认的是 `aftersep`。

`\\ \\*`  
`pageatnewline` (Package option)

**2.6** 宏包的 `pageatnewline` 是为了向后兼容而使用的。在 2.6 版本中，`\\` 不允许一个页面中断，因此等同于 `\\*`。我假设没有任何人希望页面在标题的中间中断，因此设置这个为默认值。如果因为一个奇怪的原因你要这样做，那么可以使用这个宏包选项。

### 3.4. 关于行

这个宏包包含了一些有助于在标题之上（下）的增加行和其他内容的工具。因为标题的边距可能会被修改，这些宏会考虑本地的设置并合理放置行。他们同时也会考虑到页边标题使用的空间。

`\titleline[<align>]{<horizontal material>}`  
`\titlerule[<height>]`  
`\titlerule*[<width>]{<text>}`

`\titleline` 命令允许插入一行，改行可以包含文本或者任何“水平”的元素。这个命令主要的目的是行和前导元素，但是实际上对其他的目的也有用。行有一个固定的宽度，因此必须要进行填充。比如，`\titleline{CHAPTER}` 会产生一个 underfull 的盒子。`<align>`（l, r 或者 c）是一些有帮助的辅助命令，因此你可以简单的输入 `\titleline[c]{CHAPTER}`。<sup>11</sup>

不要在包含垂直元素的地方使用 `\titleline`，否则可能引起奇怪的问题。换句话说，你可以在 `<format>`（总是可以）和 `<after>`（hang, display 以及 block）参数中使用；以及在 `<before>` 和 `<label>` 最开始处的 display 样式中使用。但是请尽量先做下测试，行为很有可能会与预期不同。

`\titlrrule` 命令相对于 `\titleline`，在需要的时候增加了自动包围的功能，因此可以用于构建行线以及填充。不带星的版本会使用 .4pt 高度划线，或者使用 `<height>` 参数（如果存在）。比如下面的例子：

<sup>11</sup>默认是 `\makebox` 命令的 s 参数。

```
\titlerule[.8pt]%
\vspace{1pt}%
\titlerule
```

会绘制两条不同高度的线，两条线之间的分隔是 1 pt。

带星号的版本重复使用 `<text>` 中的内容和他自身的宽度作为盒子的前导。盒子的宽度可以被改变为 `<width>`，但是第一个盒子的宽度还是保留为它自身的宽度，这样 `<text>` 可以对齐到空间填充后的左右边界。

看一个实际的例子，

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

会是正文和节之前的标题产生一段空白距离。

calwidth	(Package option)
----------	------------------

`warp` 样式可以度量标题中的行，以对段落进行格式化。使用这个宏包中的选项，这种能力可以扩展到其他三种样式，`display`、`block`、`hang`。通过 `\titlewidth` 返回最长行的长度。

考虑到在  $\text{T}_\text{E}\text{X}$  中，任何盒子都是排版的一个元素。如果一个盒子使用空白填充因此放大了，或者反过来考虑，一个盒子的文本破碎了，那么 `\titlewidth` 的值就可能错误（照常人的观点考虑）。举个例子，对于 `hang` 形状来说，使用的是内部的一种盒子，但是这种情况的本意是想要把标题右对齐；否则使用 `block` 形状可以取得更好的效果。换句话说，使用一个本身宽度被覆盖了盒子可能会有错误。<sup>12</sup>更进一步来说，有些命令可能会对  $\text{T}_\text{E}\text{X}$  感到疑惑，进而停止处理标题。但是如果你坚持使用文版，那么使用 `\\` 和 `\\[...]`（这也是在你没有其他需求的情况下），就没有什么问题了。另外比较重要的一点是，`<before>`、`<label>`、`<sep>` 以及 `<tilte>` 参数（没有 `<after>`）会在本地区域中被计算两次；如果你增加了一个全局的计数器，那么你就是增加了它两次。在大部分情况下，在 `<after>` 变量后放置一个冲突的声明是没有问题的。<sup>13</sup>

我们现在讨论 `\titleline` 的一个变化形式。

<sup>12</sup>这其中也包括调整了的行，它的内部单词宽度被放大了。

<sup>13</sup>还有两个更进一步的变量，`\titlewidthfirst` 和 `\titlewidthlast`，用于返回第一行和最后一行的长度。不需要特殊的工具来使用它们，但是你可以把他们的值赋给 `\titlewidth`，然后就可以使用 `\titleline*`。

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

文本首先会被封装在一个宽度为 `\titlewidth` 的盒子中；之后这个盒子会按照指定的对齐方式被封装在主要盒子中。没有等价的 `\titlerule` 命令，因此如果你考虑使用 `\titlewidth`，你需要强制使用 `\titleline*` 来做封装。

```
\titleline*[c]{\titlerule[.8pc]}
```

### 3.5. 页面样式

**2.8** 你可以在类的 `top` 和 `page` 级别设置页面样式，就像对于默认设置的章一样，使用如下命令：<sup>14</sup>

```
\assignpagestyle{\command}{\pagestyle}
```

举个例子，如果你想要强制在章中指定页码：

```
\assignpagestyle{\chapter}{empty}
```

### 3.6. 中断

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

使用 `\newcommand` 来定义以上命令，可以在不同级别使用不同的页面中断。对于没有定义的值，会使用文档类提供的内部值（通常是 `-300`）。看一个例子：

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

在分节处开始一个新页面。在一些页面布局中，即使节开始一个新页面，标题上部的空间也会被保留；可以通过如下命令来调整：

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

**2.6** 到现在为止，`\...break` 还只能在 `straight` 类中使用，但是它现在也可以在 `top` 类中使用。比较合适的值是 `\cleardoublepage`（使用 `openright` 时的默认值）和 `\clearpage`（使用 `openany` 时的默认值）。因此，在它的类已改变成 `top` 时，可以通过定义 `\chapterbreak` 为 `\clearpage` 来覆盖 `openright`（在这个例子中，部分可以使用 `openright` 继续）。

<sup>14</sup>在短暂的 2.7 版本中作为 `\titlepagestyle` 命令使用出现。

`\chaptertolists`

**2.6** 如果定义了，那么通常情况下写入到 `lists` 中的空格（比如 `List of Figures` 和 `List of Tables`）就会被这个命令中的代码所替换。如果不想再章的开头出现空格，那么定义其值为空。这个命令不是控制列表空白的通用工具，其仅仅在顶级类的标题中可用，因此对默认的章没有效果，除非你更改了他们的类（换句话说，`\...tolists` 可以用在任何顶级类的标题中。）

### 3.7. 其他宏包选项

`explicit` (Package option)

**2.7** 使用该选项，标题不会默认的跟在 `\before` 之后，需要显示的使用 `#1` 来指定，如下例：

```
\titleformat{\section}
{..}
{\thesection}{..}{#1.}
```

（比较这个例子和 [4.4](#) 节中的例子。）

`newparttoc oldparttoc` (Package options)

标准的 `parts` 会以非标准的方式写入目录的条目编号。你可以通过 `newparttoc` 来做调整，这样 `titletoc` 或者类似的宏包就可以操作条目。（这仅在重定义了 `\part` 的情况下有效。）

`cleareempty` (Package options)

调整 `\cleardoublepage` 的行为，这样空白页面可以使用 `empty` 页面样式。

`toctitles` (Package option)

**2.6** 更改分节标题中可选参数的行为，这样它仅设置运行的页眉，而不是整个 TOC 条目，该选项基于完整标题。

`newlinetospace` (Package option)

**2.6** 替换标题中的所有 `\\` 或者 `\\*` 为一个运行页眉以及 TOC 条目中的一个空格。这样，你不需要重复标题来移除一个格式化命令。

### 3.8. 扩展设置

`\titleformat` 和 `\titlespacing` 的第一个参数都有一个扩展的声明，来允许根据上下文设置不同的格式。<sup>15</sup>这个参数可以是如下形式的键/值对列表：

```
<key>=<value>, <key>=<value>, <key>, <key>,...
```

当前，出了分节命令之外，只考虑到了有页面和未编码的版本。因此，可用的键值为：

- `name`。允许的值为 `\chapter`，`\section`，等等。
- `page`。允许的值为 `odd` 或者 `even`。
- `numberless`。无用键值。除非你要设置不同的编号（不使用这个键值）和不编号的变量（带有 `numberless`），否则不需要该变量。

上文中讨论的带有分节命令的基本形式，也就是

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

实际上是

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

的缩略形式。

假设我们喜欢标题在外边栏的布局样式。我们可以按照如下设置

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

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

```

考虑到页面的信息被写入在 `aux` 文件中，你至少需要执行两次编译才能获得想要的效果。

“编号”版本在生成无编号变量时通常很好，因为极大部分情况下所需的唯一改变就是移除标签，但是如果你需要一些特殊的格式，可以使用 `numberless` 关键字。这个关键字定义了用于替换的没有编号的节的版本（换句话说，就是些在 `secnumdepth` 级别之下的小节，在前言和后记的部分，当然还有加星号的版本）。看如下的例子，

---

<sup>15</sup> 需要 `keyval` 宏包以支持该项设置。

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

`\label` 和 `\sep` 在 `numberless` 变量下都不会被忽略。

这些关键字在 `\titleformat` 和 `\titlespacing` 模式下都可以使用。通过在其中一个（或者两者）使用 `page`，奇数页和偶数页的格式化行为会有所不同。实际上，“奇数”和“偶数”已经是 L<sup>A</sup>T<sub>E</sub>X 中完善建立的项目，但是只不过很容易让人误解。在某一方面打印“奇数”也和“偶数”页的行为是一样的。（参考比较 `\oddsidemargin`）。

如果你想要创建不同的奇数/偶数页面以及不同的编号/无编号版本，我们强烈建议你定义这四个变量。

如果你移除了分节命令中指定的页面指定符，那么你必须移除 `.aux` 文件。

### 3.9. 创建新的级别以及改变文档类

对标题的形状以及类似的修改行为与周围的文本相关，而标题类允许我们变更他们的通用表现。在标题类的帮助下，你可以在 `chapter` 和 `section` 之间插入一个新的 `subchapter` 级别，或者是创建一个你自己的模式。设计一致的模式，并且定义所有相关的元素，比如计数器、宏、格式、空格以及可能的目录、目录格式，这些都是用户的责任。有三种不同的类别：`page` 和 `book` 的 `\part` 相似；在单独的页面下，`top` 和 `\chapter` 相似，这个命令开启一个新页面，并且把标题放置在页面顶部；`straight` 是位于文本中间的标题的扩展。<sup>16</sup>

```
\titleclass{<name>}{<class>}
\titleclass{<name>}{<class>}[<sup-level>]
```

如果你不使用可选的参数，那么就是调整 `<name>` 的 `<class>`。如下例：

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

调整 `part` 为 `straight` 类。

当使用第二种样式的时候，级别的编号是 `<sup-level>` 的后继。举例如下：

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

<sup>16</sup>还有一个更深入的页面类 `part`，目的是效仿 `article` 的 `\part`，但是你完全不应该使用它。可以使用 `straight` 类别来代替。请记得有一些依赖于这些类和那些 `titlesec` 的特性默认不会修改 `\part` 和 `\chapter` 的定义。

上文创建在 `chapter` 之下的一个级别（上文也同时显示了一些额外的代码，但是你添加的时候需要与 `\titleformat` 和 `\titlespacing` 的设置保持一致）。<sup>17</sup>如果章的级别为 0，那么第一个 `subchapter` 的级别为 1；之下的再增加 1（`section` 的为 2，`subsection` 的为 3，依此类推。）

依据他们的名字和要忽略的类，有两个分节命令会做出一些额外的动作：

- `\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)

现在，你希望完全自己创建自己的分节命令。这个宏包选项忽略任何可能存在前文定义，因此，也会移除了使用任何在“快速参考”中提到的选项的可能性。稍后你可以使用 `titlesec` 工具，并且定义相应的计数器和标签。

`\titleclass{<name>}[<top-num>]{<class>}`

此处，`<name>` 标题被视为顶级级别，使用的 `<top-num>` 作为级别编号（通常是 0 或者 -1）。只有在 `loadonly` 的帮助下从头创建分节命令时才应该使用这个，而且应该只有一个（不能多，也不能少）这一类的声明。在此之后，就可以按照上文解释的方法增加剩下的级别。

## 4. 附加须知

这一部分大致介绍一些对定义分节标题有用的  $\text{\LaTeX}$  命令。

### 4.1. 固定宽度标签

`\makebox` 命令允许使用固定宽度的标签，从而使得真实标题（不是标签）的左边栏处在相同的位置。如下的例子（只显示了相关的代码）：

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

请查看你的  $\text{\LaTeX}$  手册以获得关于盒子命令的更深入的参考。

---

<sup>17</sup>考虑到计数器，`remreset`宏包可能比较有用。



## 4.2. 带星号的版本

强烈不推荐在带星号的版本中使用分节命令。作为替代，你可以使用一系列的标记命令，这些命令更易于定义和在需要的时候修改。因此，在选择其中某个之前，你可以测试不同的布局。

首先需要记住，如果你使用

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

那么，分节会被编号，但是不会出现在目录和页眉中。

现在，让我们假设你需要在某个特殊内容中包含一些节；例如，一个（或者多个）带有练习的小节。我们使用一个名为 `exercises` 的环境，它的用法为：

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

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

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

下面的定义强迫除了目录行和页眉之外的地方进行编号：

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

下面的定义添加在目录行的白那红，但是页眉不做变动：

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

下面的定义更新页眉，但是不更新目录行：

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

（我发现在这个特殊例子的字母有一点奇怪；第一个和第二个选项更加智能一点。`\ignorespaces` 不是特别重要，除非在目录中有不希望的空格，否则不需要使用它。）

这些在标准类中可以使用，但是如果你使用 `fancyhdr` 或者 `titlesec` 来定义页眉，那么你需要进一步推敲以除去分节的编号。在 `titlesec` 中，可以通过 `\ifthissection`（参见下文）来完成这项工作。

如你所见，并不存在 `\addcontentsline`，`\markboth`，`\section*`，只有逻辑结构。当然你可以按照你的意愿做调整；例如如果你决定这些标题应该以小字号字体打印，那么可以包含 `\small` 命令，如果你意识到你不喜欢这样，那么可以移除它。

对于简单的文档，标准的 `LATEX` 命令更加简单和直接一些，因此我想上面的这些方法可能在大型文档中使用应该更好一点。

### 4.3. 变量

让我们假设我们希望能够制造一些形如“高级话题”，同时标签后面带有一个标记星号的小节。下面的代码用以完成这项工作：

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

要标记整个节，如下：

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

这样可以标记节，但是不包括小节。如果你希望像标记节一样来标记小节，那么你必须自己相应的定义它们。

### 4.4. 在节标题后包含一个句 = 点号

现在这种样式已经不再使用了，但是之前它是如此的流行。上文我们已经讨论过基本的技巧，现在来看剩余的部分：

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

如果你希望吧这个句号和一些命令（或者是一个下划线）组合起来使用，那么你可以这样：

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

你可能会喜欢 `explicit` 这个宏包选项。

## 5. 页面样式

作为这个宏包的一部分，我提供了一些命令那集合，以方便用户一步化的设置大标题和脚注格式。这些也页面样式被设计为何分节相关的信息一起工作。比如你不需要在字典中使用页面的第一个和最后一个条目创建页眉（你可以，但是这远比使用 `fancyhdr` 复杂）。

他和标准文档类级其他许多东西一起工作；尽管如此，在一些特殊情况下（比如 AMS 文档），可能需要额外的 L<sup>A</sup>T<sub>E</sub>X 布局的调整。<sup>18</sup>为了使用他们，设置如下的宏包选项：<sup>19</sup>

`pagestyle`

一旦完成了这些设置，你可以定义你自己的页面样式，并且激活他们。<sup>20</sup>

### 5.1. 定义页面样式

```
\newpagestyle{<name>}{<global-style>}{<commands>}
\renewpagestyle{<name>}{<global-style>}{<commands>}
```

定义一个新的样式或者重定义已经存在的样式，命名为 `<name>`。方便起见，行的结束符被忽略掉了，但是你不需要使用 `%` 来忽略他们。<sup>21</sup>

`<global-style>` 是任何要同时应用到页脚行和页眉行的样式。允许使用单独的命令。

在 `<commands>` 中，你可以使用（这些命令是相对于本地化页面的，比如在导言中定义它们将不能正常工作，因为页面样式设置会覆盖他们。）：

```
\headrule \footrule
\setheadrule{<length>} \setfootrule{<length>}
```

如果你希望能够一行在页眉线之下，同事在页脚线之上。你可以同时使用 `\setheadrule` 和 `\setfootrule` 来直接设置他们的宽度。（例如，`\setheadrule{.4pt}`，同时这也是默认值。）

<sup>18</sup>双栏布局需要使用 David Carlisle 开发的 `fix2col` 宏包。

<sup>19</sup>页面样式的工作模式在 2.3 发布版本中已经被完全重新实现。尽管如此，大部分情况下完全不需要留意这些东西。

<sup>20</sup>如此定义的页面样式不能通过 `\thispagestyle` 使用，除非他的环绕页面样式也使用 `titlesec` 做了定义。

<sup>21</sup>即使需要保留标记以传递信息给页眉，一些文档类，比如 AMS 系列，引入了额外的代码。当重定义 AMS 文档类的 `plain` 样式的时候，你必须增加如下的行：`\global\topskip\normaltopskip`。

<code>\makeheadrule \makefootrule</code>
--

titlesec 使用这些命令来设置行。如果不存在行，那么两个命令都为空（这个值是默认值。）`\setheadrule{dim}` just stands for

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

除非 dim 设置为 0pt，表示清空 `\makeheadrule` (`\setfootrule` 类似)。

你可以通过 `\linewidth` 获取这个页眉/页脚的宽度，但是围绕着行的盒子实际上是没有尺寸的。因此，你仅仅需要关心行的布局就可以了。它的基线与页眉页脚的基线相同。这意味着，作为行的元素的应该使用例如 `\raisebox` 来突出或者内陷，`\rule` 中的上升参数或者使用 `picture` 中的智能坐标来处理。这使得我们更容易在其上以及其下放置元素。例如，如下的代码创建一个带有其上为黑色加粗线条，其下为红色行线的标题（需要使用 `color` 宏包）：

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

当然，作为行线来使用的元素不仅仅局限于真实的行线；例如图片以及前导，都在允许的范围之内。

<pre>\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⟩}</pre>
--

设置页眉和页脚对应的部分。可选的参数为全部或者全部都不选择。如果没有提供可选参数，那么奇数页的设置就应该在奇偶页面上。一组带有星号的参数 (`\setfoot*`, `\sethead*`) 保留偶数页设置的顺序（很明显，如果使用这些参数，就不能再使用可选的参数）。在 `\sethead`/`\setfoot` 的参数中，以及因为它的一步式机制，我们必须区分两组不同的命令。第一个在发送标记的时候保存，同时带有小节相关的信息，包含：

- `\thechapter`, `\thesection`...
- `\chaptertitle`, `\sectiontitle`... 这些打印章、节... 的标题。
- `\ifthechapter{⟨true⟩}{⟨false⟩}`, `\ifthesection{⟨true⟩}{⟨false⟩}`... 这些展开为 `⟨true⟩` 除非对应的标题缺少标签或者在超级级别之后没有标题。（例如，在 `\chapter` 和后续的 `\section` 之间）。

- 其他任何“注册”为“标记”的命令或者值。（见下文。）

第二组是那些在每页展开的命令，包括：

- `\thepage`。
- 不包含在以上条目中的任何其他命令。

`\setmarks{<primary>}{<secondary>}`

设置要定义的 `\...title` 命令，以及何时更新标记。例如 `\setmarks{chapter}{section}` 意味着：

- `\chaptertitle` 以及 `\sectiontitle` 是页眉中允许的标题，
- `\sectiontitle` 在 `\chapter` 中重置，
- `\ifthechapter` 以及 `\ifthesection` 是合法的测试，以及
- 标记在 `\chapter` 和 `\section` 中进行更新。

默认情况下是 `\setmarks{chapter}{section}`，但是在 `article` 文档类中默认的为 `\setmarks{section}{subsection}`。可以在 `\(re)newpagestyle` 之外使用 `\setmarks` 用以设置新定义或者重定义的页面样式的默认风格。需要用在 `\pagestyle` 之前。

需要注意 `\markboth` 是一个设置 `myheadings` 标记的命令，因此在此处不会有作用。实际上，使用它可能会导致不可预料的结果。无论如何，不鼓励直接使用标记命令（参考 4.2 小节的内容），但是如果你需要他们，可以按照如下的方法使用他们。

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

## 5.2. 额外的设置

`\widenhead[<even-left>][<even-right>]{<odd-left>}{<odd-right>}`  
`\widenhead*{<even-right>/<odd-left>/}{<even-left>/<odd-right>}`

将页眉/页脚线设置的更宽一些。额外的宽度可以不是移植的，因此两个命令都包含了四个参数。像 `\sethead` 一样，带星号的版本会反向设置奇数页面。—例如，`\widenhead*{0pt}{6pc}` 与下面的效果相同 `\widenhead[6pc][0pt]{0pt}{6pc}`。

### 5.3. 运行带有浮动体的页眉 Running heads with floats

psfloats (宏包选项。)

这个宏包选项激活本节中描述的命令。<sup>22</sup>

**2.6** 下面的命令含有新的参数，以允许运行页眉/页脚时进行额外的工作。这些变更是向后兼容的——如果你需要使用老的形式，只需要保留老的名称 floatps，虽然并不推荐这种做法。

```
\setfloathead*{.}{.}{.}{\extra}}[\which-floats]
\setfloathead[.][.][.]{.}{.}{\extra}}[\which-floats]
(Similarly \setfloatfoot.)
```

带有句点号的参数与 \sethead 中的类似，最后一个参数表示如果存在制定类型的浮动体，那么就使用页眉（默认情况下，对于页眉为 tp，对于页脚为 bp）。例如：

```
\newpagestyle{main}{%
  \sethead ... your definition
  \setfoot ... your definition
  \headrule
  \footrule
  \setfloathead{}{}{}{\setheadrule{0pt}}[p]
  \setfloatfoot{}{}{}{\setfootrule{0pt}}[p]}
```

将会移除在浮动体页面上包括行线在内的的页眉/页脚。

```
\nextfloathead*{.}{.}{.}{\extra}}[\which-floats]
\nextfloathead[.][.][.]{.}{.}{\extra}}[\which-floats]
(Similarly \nextfloatfoot.)
```

在某个浮动体是这些命令的页面的最顶上（相对与最后最下面一个）一个时，你可以强制页眉（相对与页脚），这时，这些带有句点的参数使用方法与 \sethead 类似，只需要放在相应的浮动体之前。这些命令是自调用的，不需要把他们放置在 \(\re)newpagestyle 的范围中。

### 5.4. 标记

outermarks innermarks topmarks botmarks (宏包选项)

innermarks 是 L<sup>A</sup>T<sub>E</sub>X 中默认的选项，偶数页标记在底部，奇数页标记在顶部。更方便的选项是 outermarks，这个选项设置偶数也的标记在顶部，奇数页的标记在底部；这个技术文档中更常见的设置，在 The T<sub>E</sub>X book 的 259 页有

<sup>22</sup>它重定义了一些内部的 L<sup>A</sup>T<sub>E</sub>X 命令因此可能对于同样修改这些命令的其他的宏包是不兼容的。这也是这些命令仅作为可选项配置的最主要原因。

更详细的描述。这两者的设计都是双面打印的；`topmarks/botmarks` 是用来支持单面打印的，在每页的顶部/底部设置标注（他们也可以在双面打印中使用）。<sup>23</sup>

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

页面样式中新加入的一项比较酷的特性，现在提供了指定在后续的宏中使用哪些标记的可能性。可以使用这些命令集合，用来标记类似 `\thesection`，`\sectiontitle` 等的值。你可以在你的页眉中自由使用他们；<sup>24</sup>轻松一下，下面的页眉显示了这些标记中的三种：

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

当设置了 `outermarks` 宏包选项时，`\outertitlemarks` 是默认值；设置 `topmarks` 时，`\toptitlemarks` 是默认值，依此类推。在  $\text{\LaTeX}$  中，如果没有使用这些选项，`\innertitlemarks` 是默认值。`\nexttoptitlemarks` 提供了关于标记的更多设置选项，它提供了一个带有下一页顶部标记值的底部标记（只在直接使用类时有效）；通过获取它的一些值，并与 `\bottitlemarks` 中对应的值比较，你就可以知道一个小节是否延续到了下一页。当然，这些命令的使用频率不会很高。

```
\newtitlemark{<macro-name>}
\newtitlemark*{<variable-name>}
```

添加一个宏或者变量到将要保存的“标记”列表中。宏定义必须是无参数的，而且变量必须使用  $\text{\TeX}$  形式（长度相同，计数不同——字母采用 `\c@<counter>` 的形式）。

## 5.5. A couple of examples

因为标记处理方式的原因，可以使用如下的特殊结构形式：

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

<sup>23</sup>`outermarks` 在显示使用 `\<section>mark` 命令连接时不能正常工作。顶部的标记机制在显示的页面中断之后会失败。

<sup>24</sup>其实有点言过其实：顶层的标记必须使用在一章开始的页面，除非覆盖了默认的定义并且将其作为 `top` 类。



如你所见，一个页面 `\thesection` 的小节的边界可以同时打印在左页眉和右页眉。当然，还需要优化这个例子，以在页面内只存在一个单独的小节时压缩边界，但是，它最起码在如何获取更好的页面上给了我们一点提示。

在这篇文档中，使用了如下的样式：

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

下面的定义提供了类似于 Lamport 的  $\text{\LaTeX}$  书（使用 `calc` 宏包）中相似的页面样式：

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

## 6. 最后的注释

- 标题中只允许使用 `\footnote` 命令，但是它可以工作。脚注的标记不会从目录条目或者运行的页眉中移除；为了解决这个问题：

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

- 页面样式在版本 2.3 中完全进行了重新实现。即使很多东西都改变了，大部分之前的定义在现在的代码中仍然可以正常工作。值得一提的是，页眉现在只需要使用 `\the...` 和 `\...title` 命令，此外 `\usepage` 和格式化的命令仍然可以工作。现在，需的使用 `pagelstyles` 选项来显式的加载页面样式，如果没有这样做，并且使用三个基本的命令，那么加载页面样式宏的时候就会出现警告。新的代码克服了之前版本的这些限制，修正了一些 bug(例如顶部标注和浮动体的不兼容性)，同时增加了一些新的特性。

## 7. 目录: `titletoc` 宏包

这个宏包是 `titlesec` 宏包的同伴，它提供了对目录条目的处理。他的工作原理同 `titlesec` 类似 —除了处理标准  $\text{\LaTeX}$  和文档类中定义的命令之外，还增

加了用于格式化目录条目的通用的新命令。这意味着你只需要学习两个基本的命令以及一些工具，之后你就可以使用这些新特性。或许新的格式和字体可以通过类似于 `\`、`\markbox`、`\large`、`\itshape` 等等的命令来设置，目录条目也不会局限于某种样式，因为他们现在已经相当自由。

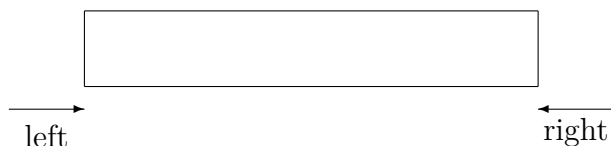
某些情况下，使用 `titletoc` 定义的目录条目的行为与使用标准命令创建的目录条目存在差异。特别是：

- 如果第一个条目比第二个条目的级别高，那么页面不会在两个条目中间中断，比如，在一个节和小节之间。如果两者是相同的级别，那么就允许中断；如果第一个比第二个级别低，那么中断页面或许是一个好做法。
- 前导的符号不会居中，而是向右偏移，这样通常更方便。

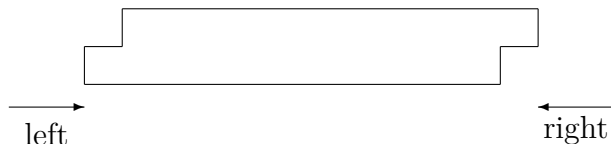
需要注意的是，任何对目录的处理都不可能完成，因为标准 `LATEX` 命令直接定义了一些不能移除的格式化命令。这对于图片和表格列表还有 `\part` 命令及其重要。<sup>25</sup>

### 7.1. titletoc 10 分钟速成指导

目录的条目作为矩形区域对待，填充的内容是文本或者其他填充物。我们绘制出类似的区域（当然，周围的线不会被打印出来）：

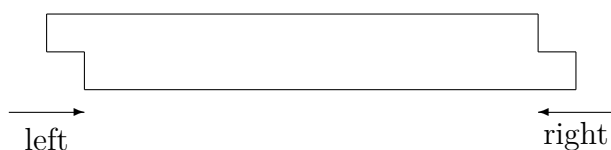


页面左边界和区域的左边线之间的空白命名为 `<left>`；类似的还有 `<right>`。你可以修改第一行的开始或者最后一行的结束。例如，通过设置“接受”两个空白间距为 `\hspace*{2pc}`，得到的区域如下：



通过使用 `\hspace*{-2pc}` 来“清除”两个区域之间的空白，我们得到：

<sup>25</sup>但是其中有些问题已经通过 `titlesec` 得到修正。



如果你已经查看过目录，那么后者对你来说应该很熟悉 – 标签在最开始处，页面在最后：

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

所有你需要做的就是以正确的方式放置他们。如果你使用 `\hspace*{-2pc}` 保留这些内容，只是简单的放置一个包含节标签或者页面的 2 pc 宽度的盒子就可以重新得到这些空间。因为这种布局使用的如此频繁，所以提供了两个专门的命令来做这件事情：

- `\contentslabel{<length>}` 在开头创建空白，并且打印节编号。
- `\contentspage` 在结尾处创建 `<right>` 长度的空白，并且以右对齐打印页码。

现在，我们要介绍的是三个基本的命令：

`\dottedcontents{<section>}[<left>][<above>]  
{<label width>}[<leader width>]`

此处：

- `<section>` 是不带反斜线的节的名称：part, chapter, section 等等。不允许使用 figure 以及 table。（由于只是处理概念上的问题，因为没有反斜线，因为并不是指 `\part`, `\section` 等的宏自身的定义。进一步说，figure 和 table 是环境。）
- `<above>` 是对条目进行全局格式化的代码。允许使用竖直元素。现在假设我们已经知道 `\thecontentslabel` 的值（见下文），这允许你依据此来做决策（使用 ifthen 宏包的帮助）。你可以使用 titlesec `\filleft`, `\fillright`, `\fillcenter`, `\filllast` 命令。
- `<left>` 即使强制使用了括号，这个参数还是使用左页边距设置左边距。
- `<label width>` 是创建的标签空白之间的宽度，参见上文描述。
- `<leader width>` 是包含作为填充器使用的字符的盒子的宽度，参见下文描述。

book 类中节和小节条目的定义大体上等价于:

```
\contentsmargin{2.55em}
\dottedcontents{section}[1.5em]{}{2.3em}{1pc}
\dottedcontents{subsection}[3.8em]{}{3.2em}{1pc}
```

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

此处, `<section>`, `<left>` 和 `<above>` 与前文所述类似, 同时

- `<before with label>` 在水平模式生效, 它在条目之前使用。与在 `\titleformat` 一样, 最后一个命令可以带有一个标题的参数。
- `<before without label>` 与上面类似, 但是没有标签。
- `<filler and page>` 是自解释的。使用 `\titlerule` 命令来创建红包自身和 `titlesec` 共享的填充器。在这里的上下文中使用时, 它的行为有一点变化, 以满足目录前导符的需要。<sup>26</sup> 你可能更喜欢作为替代的 `\hspace` 命令。
- 最后, `<after>` 是跟在条目之后的代码, 也就是水平空白。

在定义条目时, 如果你希望增加水平空白, 那么使用 `\addvspace`, 同时需要使用 `\\*` 来替代 `\\` 作为行中断符号。

这个命令可以在文档的中间使用, 从而在任意地方来改变目录/表格/图表条目的格式。新的格式写入到 `toc` 文件, 因此可能需要运行两次才能看到更改的效果。

```
\contentsmargin{<right>}
```

此处设置的值会在整个分节结构中使用。如果你对此心存疑虑, 那么原因很简单: 大部分情况下, `<right>` 页边距是一个常量。尽管如此, 你可以在 `<before>` 参数中对其进行局部化的更改。注意标准文档类的默认空白并没有留下显示空间来显示超出 100 的粗体页码, 因此你可能希望使用这个命令设置一个大一点的边距。

book 类使用类似于 (但不是完全等同) 如下的方式格式化节的条目:

```
\titlecontents{section}
  [1.5em]
  {}
  {\contentslabel{2.3em}}
  {\hspace*{-2.3em}}
  {\titlerule*[1pc]{.}\contentspage}
```

<sup>26</sup>对于 TeXnicians, 默认的 `\xleaders` 是 `\leaders`。

把这个定义与与上文给出的相比较，你现在应该明白 `\dottedcontents` 是如何定义的。

尽管标准文档类使用字体相对单位（主要是 `em`），但是我还是强烈建议使用绝对单位（`pc`, `pt`, 等等）。

## 7.2. 更多内容

严格的排版规则要求文正的文本不应该超过前导的最后一个句点。观念上看，他们应该是对齐的才对。令人诧异的是， $\text{\TeX}$  中缺少一个自动处理这个问题的工具——当你使用前导句号填充盒子时，他们可以使用 `\cleaders` 原语在盒子中居中，也可以使用 `\xleaders` 来“调整”，或者使用最外边的包围盒子与 `\leaders` 命令来对齐，但是没有办法让他们使用“当前的”的边距对齐。

因此，通过手工方式来获取一个良好的布局是唯一的方法。你可以在 `\contentsmargin` 命令中使用使用可选的参数来达到目的，他的完整声明如下：

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

除了最后一行 `hi` 用于放置前导外，所有的行，`⟨correction⟩` 的长度会加到 `⟨right⟩` 上。例如，如果文本比最后一个句号长了 `6pt`，你可以重写 `\contentsmargin` 命令来添加 `6pt` 的 `\correction`。footnote 有用的一点，许多 `dvi` 预览工具鱼讯获取位置的指向位置的坐标。与标准的  $\text{\LaTeX}$  工具不同，`\titlerule*` 命令已经经过设计，因此 `⟨correction⟩` 会使用尽可能小的值。

```
\thecontentslabel \thecontentspage
```

使用标签和无附加格式的页面包含文版包含文版，除非是文档类书写的部分。

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

如上所述，但是带有不同的 `⟨formats⟩`s。默认情况下只有 `\thecontentslabel` 以及 `\thecontentspage`。

```
\contentspush{⟨text⟩}
```

打印 `⟨text⟩`，并且使用 `⟨text⟩` 的宽度来增加 `⟨left⟩`。类似于悬挂样式的 `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>]
```

这个带型号的版本在一个单独的段落中聚集条目。〈separator〉参数是不同条目之间的分隔符，还有一个带有结束标点的可选参数。例如，如下的文档设置：

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

结果样式如本文档最开始部分的内容。注意，段落的格式必须书写在〈above〉变量中。

现在我们解释下可选参数的工作原理。首先注意擦身农户的个数决定了他们的意义，既然条目之间需要一个分隔符号，那么这个会一直存在。另一方面来说，〈begin〉通常极少用到，因此它是最低级的“特性”。最简单的情况是当标题在同一个级别上时，这种情况下，在连贯的条目和块的结尾处，会插入〈sep〉以及〈end〉参数（默认为空）。〈before〉坠毁在块的开始出执行一次，对于整个条目集而言，它的声明是局部的。

现在我们假设我们需要组合两个级别的条目。这种情况下需要使用嵌套规则。我们使用节和小节来完成这种想法。当一个小节条目在一个节之后开始时，就会执行〈before〉，同时会插入小节的〈begin〉部分，这部分应该仅仅包含文本格式。之后就添加小节，并且按照上文所述的方法插入分隔符。当遇到节时，就添加小节的结束标点以及节的分隔符（除小节终止了一个块，这种情况下会添加节的终止符）。我们说“在一个节之后”，因为小节从来都不会开始一个块。<sup>27</sup>小节的条目嵌套在节中，因此声明仍然是局部的。

我们举个例子来说明这种情况。

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

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

---

<sup>27</sup>在极少情况下，可能需要这样做。

会产生类似于下文的内容:

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

```
\contentsuse{<name>}{<ext>}
```

需要确保 titletoc 知晓带有 <ext> 扩展名的文件的内容。主要是为了确保他可以把 \contentsfinish 添加到对应的内容的结尾处 (而且这需要手工添加到目录的结尾处)。这个宏包执行

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

```
leftlabels rightlabels (Package options)
```

These package options set how the labels are aligned in 这个宏包选项设置 \contentslabel 中标签的对齐方式。默认为 rightlabels, 使用 \leftlabels 的情况下, \contentslabel 默认的 <format> 为 \thecontentstlabel\enspace。

```
dotinlabels (Package option)
```

使用这个选项, 在 \contentslabel 中的标签之后会增加一个句点。

### 7.3. Partial TOC's

```
\startcontents[<kind>]
```

在使用这个命令的地方, 会生成一个部分目录。<kind> 参数允许不同的目录集合, 默认值为 default。这些设置可以混用, 但是通常是使用嵌套的方式。例如, 你可以设置两种不同的部分目录: 通过 part 以及通过 chapter(当然还有全部目录这种)。当一个 part 开始时, 使用 \startcontents[parts], 使用 chapter 时, 则为 \startcontents[chapters]。通过这种方式, 新的目录条目在每个 part 和 chapter 开始。<sup>28</sup>

```
\stopcontents[<kind>]
\resumecontents[<kind>]
```

停止 <kind> 类型的部分目录, 稍后可以恢复。因为在必要情况下, 部分内容要通过 \startcontents 来终止, 你不应该过多的使用这些宏。

```
\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

<sup>28</sup>所有的部分目录会保存在一个后缀名为 .ptc 的单独的文件中。



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>` parameter 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 `\contentsmargin`, for instance. 打印当前的 `<kind>` 类型的部分目录表。一般情况下使用主目录表条目的格式, 除非使用了 `<prefix>`。这种情况下, 就是用 `<prefix><level>` 的格式, 前提是已经定义了这种格式。例如, 如果前缀是 `l` 并且定义了 `lsection` 的格式, 那么就使用这个格式; 否则, 就使用 `section` 的格式。`<top>` 参数用用设置目录表的最高级别—对于 part 目录表而言, 应该为 0 (chapter), 对于 chapter 目录表为 1 (section), 依此类推。最后, `<init-code>` 是当前目录表的局部代码; 可以用来改变 `tocdepth` 的值或者 `|`) :

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

```
\titlecontents*{<lsection>}[0pt]
  {\small\itshape}{}{}
  {}{\textbullet\ }{.}
```

包含的条目是那些从级别 1 到级别 2 包含的内容。

#### 7.4. 部分列表 2.6

你可能希望创建部分的 LOFs<sup>29</sup> 以及 LOTs<sup>30</sup>。声明的格式类似于部分 TOCs, 而且应用于 TOCs 的内容对他们也适用。命令如下:

```
\startlist[<kind>]{<list>}
\stoplist[<kind>]{<list>}
\resumelist[<kind>]{<list>}
\printlist[<kind>]{<list>}{<prefix>}{<init-code>}
```

此处 `<list>` 为 `lof` 或者 `lot`。需要注意 `\printlist` 没有 `<top>` 参数, 因为图形和表格没有级别。当前情况下 (2.8 版本), 只支持这两种浮动体, 但是后续的版本会添加更多的浮动体类型支持。不幸的是, 许多文档类使用了一些对这些列表格式化的命令 (明确指出, `chapters` 中的 `\addvspaces`); 我现在还不能确定如何能够在不移除其他人需求的功能的条件下移除这些命令, 但是可以使用一点小计谋来操作, 我们可以使用 `\renewcommand\addvspace[1]{}` 在 `<init-code>` 中重定义 `\addvspace`

<sup>29</sup>译者注: List Of Figures

<sup>30</sup>译者注: List Of Tables

## 7.5. 示例

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

这个章的编号在页面边界，使用的字体比标题的字体稍大。如果章缺少编号（有可能是因为是前言或者参考文献），那么就不会加粗。页码跟随在标题之后，不带任何填充，但是中间有 em 左右的空白。

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

章标题居中显示，章的标签在顶部显示。没有页码。

## 7.6. 在内容中插入图片

`\addtocontents` 命令仍然是可用的，你仍然可以用它执行任何特殊的操作，比如在条目之前或者之后插入图片。可悲的是，不允许插入易碎的参数以及使用复杂的代码，那将会导致一片混乱。一种变通的方案是定义一直执行需要操作的命令，比如使用 `\protect` 加以保护。

让我们假设我们需要在条目前插入一个图片。

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

勉强可以工作。

在我们要插入图片的地方：

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

## 7.7. 使用星号标注条目

现在我们讨论一个与 titlesec 有关的问题: 让我们把带有星号的节作为一个“高级话题”来讨论, 除非你想在目录表中输出他。下面是用到的代码:

```
\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. titlesec 的原理

一旦你阅读了本文档, 你就应该明白, 这个宏包不是给那些喜欢使用标准布局但是只是想做一些小改动的人准备的。它是一套工具, 提供给专业的出版商, 他们拥有明确的布局概念, 但是有没有足够的技术来进行处理。请注意, 本宏包不会给你的节格式化意识带来任何提升。

## 9. 附录

下文的例子是使用本宏包的一些说明, \parskip 设置为 0 pt。

■

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}%
\titeline*[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 Which is followed by some text to show the result.  
 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.

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        Which is followed by some text to show the result. But don't stop reading,  
                  of    because the following example illustrates how to take advantage of other packages.  
 margin        The last command in the last argument can take an argument, which is the title  
 section       with no additional 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]

```

## 9.1. 一个完整的例子

下面十一个完整的标题模式的例子。

```

\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

\widenead{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}

```

## 9.2. 标准文档

现在我们来看一下标准文档类的分节命令是如何定义的。

```

\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}

```

## 9.3. 章的例子

最后，我们用这个例子来说明如何使用 `picture` 环境来创建精美的分节格式。即使是只使用标准 `LATEX` 所提供的基本的工具，你也可以创建出激动人心的标题，但是，使用例如 `pspicture` (`PSTricks` 的宏包，或者包含使用外部程序所创建的图形，你可以设计更细致优美的效果。

```

\usepackage[dvips]{color}
\usepackage[rigidchapters,explicit]{titlesec}

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

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

(所使用的精确值依赖于文本的区域、文档类、\unitlength、纸张大小等等。)