

L^AT_EX 2_ε 的标准文档类 *

Copyright (C) 1992 by Leslie Lamport
Copyright (C) 1994-2021 by Frank Mittelbach,
Johannes Braams 和 L^AT_EX 项目团队

赣医一附院神经科 黄旭华 翻译

2023/05/17

This file is maintained by the L^AT_EX Project team.
Bug reports can be opened (category `latex`) at
<https://latex-project.org/bugs.html>.

目 录

1	DOCSTRIP 模块	4
2	初始代码	5
3	选项的声明	6
3.1	设定纸张尺寸	6
3.2	选择类型尺寸	7
3.3	双面或单面打印	7
3.4	draft 选项	7
3.5	titlepage 选项	7
3.6	openright 选项	8
3.7	双栏打印	8
3.8	公式编号在左侧	8

* 该文件的版本号为 v1.4n, 最新修订日期为 2023/05/17。

3.9	左对齐显示	8
3.10	打开参考文献	9
4	执行选项	10
5	加载宏包	11
6	文档布局	11
6.1	字体	11
6.2	分段	15
6.3	页面布局	16
6.3.1	垂直间距	17
6.3.2	文本的尺寸	17
6.3.3	边距	20
6.3.4	脚注	23
6.3.5	浮动放置参数	24
6.4	页面样式	27
6.4.1	标记约定	27
6.4.2	定义页面样式	28
7	文档标记	32
7.1	标题	32
7.2	章和节	35
7.2.1	构建块	35
7.2.2	标记命令	37
7.2.3	定义计数器	37
7.2.4	前言、正文和附属资料	38
7.2.5	部分	39
7.2.6	章	43
7.2.7	下级标题	46
7.3	列表	47
7.3.1	普通的列表参数	47
7.3.2	排序列表	50
7.3.3	常规列表	50
7.3.4	解说列表	51
7.4	定义新环境	51

7.4.1	摘要	51
7.4.2	诗歌	52
7.4.3	引用 (quotation)	53
7.4.4	引用 (quote)	53
7.4.5	定理	53
7.4.6	标题页	53
7.4.7	附录	54
7.5	为现有环境设置参数	55
7.5.1	数组 (array) 和表格 (tabular)	55
7.5.2	表格 (tabbing)	56
7.5.3	小页	56
7.5.4	有框的盒子	56
7.5.5	排序单行公式和排序公式组	56
7.6	浮动对象	57
7.6.1	图形	58
7.6.2	表格	58
7.6.3	标题	59
7.7	字体更改	60
8	交叉引用	62
8.1	目录, 等	62
8.1.1	目录	63
8.1.2	图形列表	67
8.1.3	表格列表	68
8.2	参考文献	68
8.3	索引	70
8.4	脚注	70
9	初始化	73
9.1	单词	73
9.2	日期	73
9.3	两栏模式	74
9.4	页面样式	74
9.5	单面或双面打印	74

1 DOCSTRIP 模块

实现 (implementation) 中使用了以下模块 (modules) 来指导 DOCSTRIP 生成外部文件 (external files):

模块	生成的外部文件
article	生成 article 文档类
report	生成 report 文档类
size10	生成 10pt 的类选项
size11	生成 11pt 的类选项
size12	生成 12pt 的类选项
book	生成 book 文档类
bk10	生成 10pt 的 book 类选项
bk11	生成 11pt 的 book 类选项
bk12	生成 12pt 的 book 类选项
driver	生成文档驱动程序 (documentation driver) 文件

2 初始代码

在本节中，我们定义了一些稍后使用的命令。

`\@ptsize` 这个控制序列 (control sequence) 用于存储我们正在排版的点大小 (pointsize) 的第二个数字 (second digit)。因此，通常它的值是 0、1 或 2 中的一个。

```
1 <{*article | report | book}>
2 \newcommand\@ptsize{}
```

`\if@restonecol` 当文档必须打印成两栏 (columns) 时，我们有时不得不临时切换到一栏。这个开关用于记住切换回来。

```
3 \newif\if@restonecol
```

`\if@titlepage` 用于指示是否必须制作标题页 (titlepage) 的开关。对于 article 文档类，默认设置是不创建单独的标题页 (separate titlepage)。

```
4 \newif\if@titlepage
5 <article>\@titlepagefalse
6 <!article>\@titlepagetrue
```

`\if@openright` 用于指示章 (chapters) 是否必须从右页 (right-hand page) 开始的开关。report 类的默认值为 no；book 类的默认值为 yes。

```
7 <!article>\newif\if@openright
```

`\if@mainmatter` `\if@mainmatter` 开关，仅在 book 文档类中可用，表示我们是否正在处理 book 中的主要材料 (main material)。

```
8 <book>\newif\if@mainmatter \@mainmattertrue
```

3 选项的声明

3.1 设定纸张尺寸

变量 `\paperwidth`(纸宽度) 和 `\paperheight`(纸高度) 应反映剪裁 (trimming) 后的实际纸张尺寸 (physical paper size)。对于台式打印机 (desk printer) 输出, 这通常是真正的纸张尺寸, 因为没有后期处理 (post-processing)。真正的书籍制作 (book production) 类 (classes) 可能会添加其他纸张尺寸, 并额外制作裁剪标记 (crop marks for trimming)。在兼容模式 (compatibility mode) 中, 这些 (和一些后续) 选项是禁用的, 因为它们在 $\text{\LaTeX}2.09$ 中不存在。

```
9 \if@compatibility\else
10   \DeclareOption{a4paper}
11     {\setlength\paperheight {297mm}%
12      \setlength\paperwidth  {210mm}}
13   \DeclareOption{a5paper}
14     {\setlength\paperheight {210mm}%
15      \setlength\paperwidth  {148mm}}
16   \DeclareOption{b5paper}
17     {\setlength\paperheight {250mm}%
18      \setlength\paperwidth  {176mm}}
19   \DeclareOption{letterpaper}
20     {\setlength\paperheight {11in}%
21      \setlength\paperwidth  {8.5in}}
22   \DeclareOption{legalpaper}
23     {\setlength\paperheight {14in}%
24      \setlength\paperwidth  {8.5in}}
25   \DeclareOption{executivepaper}
26     {\setlength\paperheight {10.5in}%
27      \setlength\paperwidth  {7.25in}}
```

假设给出了肖像纸 (portrait paper) 的尺寸, 选项 `landscape` 切换 `\paperheight` 和 `\paperwidth` 的值。

```
28   \DeclareOption{landscape}
29     {\setlength\@tempdima {\paperheight}%
30      \setlength\paperheight {\paperwidth}%
31      \setlength\paperwidth {\@tempdima}}
32 \fi
```

3.2 选择类型尺寸

通过定义 `\@ptsize` 来处理类型尺寸选项 (type size options), 它包含所讨论尺寸的最后一位数字, 并在 `\ifcase` 语句 (statements) 上进行分支 (branching)。由于历史原因, 这样做是为了与使用 `\@ptsize` 变量选择特殊操作 (special actions) 的其他包保持兼容 (compatible)。这使得小于 10pt 的尺寸选项 (size options) 的声明变得困难, 尽管可以使用 9 和 8, 前提是类 (class) 不会同时定义 8pt 和 18pt 选项。

```
33 \if@compatibility
34   \renewcommand\@ptsize{0}
35 \else
36   \DeclareOption{10pt}{\renewcommand\@ptsize{0}}
37 \fi
38 \DeclareOption{11pt}{\renewcommand\@ptsize{1}}
39 \DeclareOption{12pt}{\renewcommand\@ptsize{2}}
```

3.3 双面或单面打印

对于双面打印 (two-sided printing), 我们使用开关 `\if@twoside`。此外, 我们还必须设置 `\if@mparswitch`, 以便将任何页边距段落 (margin paragraphs) 放入外部页边距 (outside margin) 中。

```
40 \if@compatibility\else
41   \DeclareOption{oneside}{\@twosidefalse \@mparswitchfalse}
42 \fi
43 \DeclareOption{twoside}{\@twosidetrue \@mparswitchtrue}
```

3.4 draft 选项

如果用户要求 draft(草稿), 我们会显示任何溢框 (overfull boxes)。我们可能会在这个选项中添加一些更有趣的东西。

```
44 \DeclareOption{draft}{\setlength\overfullrule{5pt}}
45 \if@compatibility\else
46   \DeclareOption{final}{\setlength\overfullrule{0pt}}
47 \fi
```

3.5 titlepage 选项

一篇 article(文章) 通常没有单独的标题页 (separate titlepage), 但是用户可以请求一个。

```

48 \DeclareOption{titlepage}{\@titlepagetrue}
49 \if@compatibility\else
50   \DeclareOption{notitlepage}{\@titlepagefalse}
51 \fi

```

3.6 openright 选项

此选项确定章 (chapter) 是否必须从请求的右侧页面 (right-hand page) 开始。

```

52 \!article\if@compatibility
53 \book\@openrighttrue
54 \!article\else
55 \!article\DeclareOption{openright}{\@openrighttrue}
56 \!article\DeclareOption{openany}{\@openrightfalse}
57 \!article\fi

```

3.7 双栏打印

通过该开关再次实现了双栏和单栏打印。

```

58 \if@compatibility\else
59   \DeclareOption{onecolumn}{\@twocolumnfalse}
60 \fi
61 \DeclareOption{twocolumn}{\@twocolumntrue}

```

3.8 公式编号在左侧

leqno 选项可用于获取公式左侧的公式编号 (equation numbers)。它加载在构建格式时从内核文件 (kernel files) 自动生成的代码。如果公式编号确实获得了特殊的格式 (special formatting)，那么类将需要显式地 (explicitly) 提供代码，而不是使用内核文件 (kernel file)。

```

62 \DeclareOption{leqno}{\input{leqno.clo}}

```

3.9 左对齐显示

fleqn 选项重新定义了所显示的数学环境 (math environments)，使其显示为左对齐 (flush left)，并从左边距 (left margin) 缩进 `\mathindent`。它加载在构建格式时从内核文件自动生成的代码。

```

63 \DeclareOption{fleqn}{\input{fleqn.clo}}

```


3.10 打开参考文献

`openbib` 选项生成“open(打开)”参考文献样式 (bibliography style), 其中每个块 (block) 从一个新行开始, 块中的后续行缩进 `\bibindent`。

```
64 \DeclareOption{openbib}{%
```

首先, 在参考文献环境 (bibliography environment) 中添加一些钩子 (hook)。

```
65   \AtEndOfPackage{%  
66     \renewcommand\@openbib@code{%  
67       \advance\leftmargin\bibindent  
68       \itemindent -\bibindent  
69       \listparindent \itemindent  
70       \parsep \z@  
71     }%
```

另外, `\newblock` 的定义被覆盖。

```
72     \renewcommand\newblock{\par}}%  
73 }
```

4 执行选项

在这里，我们执行 (execute) 默认选项来初始化 (initialize) 某些变量。注意，文档类 “book” 总是使用双面打印 (two sided printing)。

```
74 <*article>
75 \ExecuteOptions{letterpaper,10pt,oneside,onecolumn,final}
76 </article>
77 <*report>
78 \ExecuteOptions{letterpaper,10pt,oneside,onecolumn,final,openany}
79 </report>
80 <*book>
81 \ExecuteOptions{letterpaper,10pt,twoside,onecolumn,final,openright}
82 </book>
```

`\ProcessOptions` 命令会为每个声明的 FOO 选项执行代码，用户在他的 `\documentclass` 命令中为其键入了 FOO 选项。对于他键入的每个未声明的 BAR 选项，会被假定为全局选项 (global option)。所有选项都将作为文档选项 (document options) 传递给文档导言中的任何 `\usepackage` 命令。

```
83 \ProcessOptions
```

既然所有选项都已执行，我们就可以加载所选的类选项文件 (class option file)，其中包含所有尺寸相关的代码 (size dependent code)。

```
84 <!book>\input{size1\@ptsize.clo}
85 <book>\input{bk1\@ptsize.clo}
86 </article | report | book>
```

5 加载宏包

标准类文件 (standard class files) 不加载其他宏包 (packages)。

6 文档布局

在本节中，我们最终将讨论令人讨厌的排印细节 (typographical details)。

6.1 字体

L^AT_EX 提供了用户命令 (user commands) 来更改字体的尺寸 (size)，它是相对于“主 (main)”尺寸的。每个相对尺寸变化的 (relative size changing) 命令 `\size` 执行命令 `\@setfontsize\size<font-size><baselineskip>`，其中的：

`<font-size>` 从现在开始使用的字体的绝对尺寸 (absolute size)。

`<baselineskip>` 所选字体尺寸的 `\baselineskip` 的正常值。(实际值为 `\baselinestretch * <baselineskip>`)

L^AT_EX 内核中定义的许多命令缩短了 (shorten) 以下定义，并一直在使用。他们是：

<code>\@vpt</code>	5	<code>\@vipt</code>	6	<code>\@viipt</code>	7
<code>\@viipt</code>	8	<code>\@ixpt</code>	9	<code>\@xpt</code>	10
<code>\@xipt</code>	10.95	<code>\@xiipt</code>	12	<code>\@xivpt</code>	14.4
...					

`\normalsize` 主尺寸 (main size) 的用户级命令 (user level command) 为 `\normalsize`。在内部，当 L^AT_EX 引用主尺寸时，它使用 `\@normalsize`。如果 `\normalsize` 从其默认定义 (default definition) 被重新定义 (仅发出错误消息)，那么 `\@normalsize` 将被定义为 `\normalsize`。否则 `\@normalsize` 只会选择一个 10pt/12pt 的尺寸。

`\normalsize` 宏还为 `\abovedisplayskip`、`\abovedisplayshortskip` 和 `\belowdisplayshortskip` 设置新值。

```
87 <*10pt | 11pt | 12pt>
88 \renewcommand\normalsize{%
89 <*10pt>
90   \@setfontsize\normalsize\@xpt\@xipt
91   \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
```

```

92   \abovedisplayshortskip \z@ \@plus3\p@
93   \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
94 </10pt>
95 < *11pt>
96   \@setfontsize\normalsize\@xipt{13.6}%
97   \abovedisplayskip 11\p@ \@plus3\p@ \@minus6\p@
98   \abovedisplayshortskip \z@ \@plus3\p@
99   \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
100 </11pt>
101 < *12pt>
102   \@setfontsize\normalsize\@xipt{14.5}%
103   \abovedisplayskip 12\p@ \@plus3\p@ \@minus7\p@
104   \abovedisplayshortskip \z@ \@plus3\p@
105   \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
106 </12pt>

```

`\belowdisplayskip` 始终等于 `\abovedisplayskip`。第一级列表 (first level list) 的参数总是由 `\@listI` 给出。

```

107   \belowdisplayskip \abovedisplayskip
108   \let\@listi\@listI}

```

We initially choose the `normalsize` font.

```

109 \normalsize

```

我们使用 `\MakeRobust` 而不是上面的 `\DeclareRobustCommand` 来避免重新定义的日志条目 (log entry)。但如果我们在回滚情况 (rollback situation) 下运行 (2015 年之前), 我们不会触及它。

```

110 \ifx\MakeRobust\@undefined \else
111   \MakeRobust\normalsize
112 \fi

```

`\small` 这类似于 `\normalsize`。

```

113 \DeclareRobustCommand\small{%
114 < *10pt>
115   \@setfontsize\small\@ixpt{11}%
116   \abovedisplayskip 8.5\p@ \@plus3\p@ \@minus4\p@
117   \abovedisplayshortskip \z@ \@plus2\p@
118   \belowdisplayshortskip 4\p@ \@plus2\p@ \@minus2\p@
119   \def\@listi{\leftmargin\leftmargini
120               \topsep 4\p@ \@plus2\p@ \@minus2\p@
121               \parsep 2\p@ \@plus\p@ \@minus\p@

```

```

122             \itemsep \parsep}%
123 </10pt>
124 <*11pt>
125     \@setfontsize\small\@xpt\@xipt
126     \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
127     \abovedisplayshortskip \z@ \@plus3\p@
128     \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
129     \def\@listi{\leftmargin\leftmargini
130             \topsep 6\p@ \@plus2\p@ \@minus2\p@
131             \parsep 3\p@ \@plus2\p@ \@minus\p@
132             \itemsep \parsep}%
133 </11pt>
134 <*12pt>
135     \@setfontsize\small\@xipt{13.6}%
136     \abovedisplayskip 11\p@ \@plus3\p@ \@minus6\p@
137     \abovedisplayshortskip \z@ \@plus3\p@
138     \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
139     \def\@listi{\leftmargin\leftmargini
140             \topsep 9\p@ \@plus3\p@ \@minus5\p@
141             \parsep 4.5\p@ \@plus2\p@ \@minus\p@
142             \itemsep \parsep}%
143 </12pt>
144     \belowdisplayskip \abovedisplayskip
145 }

```

`\footnotesize` 这类似于 `\normalsize`。

```

146 \DeclareRobustCommand\footnotesize{%
147 <*10pt>
148     \@setfontsize\footnotesize\@viipt{9.5}%
149     \abovedisplayskip 6\p@ \@plus2\p@ \@minus4\p@
150     \abovedisplayshortskip \z@ \@plus\p@
151     \belowdisplayshortskip 3\p@ \@plus\p@ \@minus2\p@
152     \def\@listi{\leftmargin\leftmargini
153             \topsep 3\p@ \@plus\p@ \@minus\p@
154             \parsep 2\p@ \@plus\p@ \@minus\p@
155             \itemsep \parsep}%
156 </10pt>
157 <*11pt>
158     \@setfontsize\footnotesize\@ixpt{11}%
159     \abovedisplayskip 8\p@ \@plus2\p@ \@minus4\p@
160     \abovedisplayshortskip \z@ \@plus\p@

```

```

161 \belowdisplayshortskip 4\p@ \@plus2\p@ \@minus2\p@
162 \def\@listi{\leftmargin\leftmargini
163         \topsep 4\p@ \@plus2\p@ \@minus2\p@
164         \parsep 2\p@ \@plus\p@ \@minus\p@
165         \itemsep \parsep}%
166 </11pt>
167 <*12pt>
168 \@setfontsize\footnotesize\@xpt\@xipt
169 \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
170 \abovedisplayshortskip \z@ \@plus3\p@
171 \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
172 \def\@listi{\leftmargin\leftmargini
173         \topsep 6\p@ \@plus2\p@ \@minus2\p@
174         \parsep 3\p@ \@plus2\p@ \@minus\p@
175         \itemsep \parsep}%
176 </12pt>
177 \belowdisplayskip \abovedisplayskip
178 }
179 </10pt | 11pt | 12pt>

```

`\scriptsize` 这些宏 (macros) 都比以前的宏简单得多, 它们只选择一个新的字体尺寸 (font-size), 但只保留显示 (displays) 和列表 (lists) 的参数。

```

\large 180 <*10pt>
\Large 181 \DeclareRobustCommand\scriptsize{\@setfontsize\scriptsize\@viipt\@viipt}
\large 182 \DeclareRobustCommand\tiny{\@setfontsize\tiny\@vpt\@vpt}
\Large 183 \DeclareRobustCommand\large{\@setfontsize\large\@xiipt{14}}
\huge 184 \DeclareRobustCommand\Large{\@setfontsize\Large\@xivpt{18}}
\Huge 185 \DeclareRobustCommand\LARGE{\@setfontsize\LARGE\@xviipt{22}}
186 \DeclareRobustCommand\huge{\@setfontsize\huge\@xxpt{25}}
187 \DeclareRobustCommand\Huge{\@setfontsize\Huge\@xxvpt{30}}
188 </10pt>
189 <*11pt>
190 \DeclareRobustCommand\scriptsize{\@setfontsize\scriptsize\@viipt{9.5}}
191 \DeclareRobustCommand\tiny{\@setfontsize\tiny\@vpt\@vpt}
192 \DeclareRobustCommand\large{\@setfontsize\large\@xiipt{14}}
193 \DeclareRobustCommand\Large{\@setfontsize\Large\@xivpt{18}}
194 \DeclareRobustCommand\LARGE{\@setfontsize\LARGE\@xviipt{22}}
195 \DeclareRobustCommand\huge{\@setfontsize\huge\@xxpt{25}}
196 \DeclareRobustCommand\Huge{\@setfontsize\Huge\@xxvpt{30}}
197 </11pt>
198 <*12pt>

```

```

199 \DeclareRobustCommand\scriptsize{\@setfontsize\scriptsize\@viipt{9.5}}
200 \DeclareRobustCommand\tiny{\@setfontsize\tiny\@vipt\@viipt}
201 \DeclareRobustCommand\large{\@setfontsize\large\@xivpt{18}}
202 \DeclareRobustCommand\Large{\@setfontsize\Large\@xxviipt{22}}
203 \DeclareRobustCommand\LARGE{\@setfontsize\LARGE\@xxpt{25}}
204 \DeclareRobustCommand\huge{\@setfontsize\huge\@xxvpt{30}}
205 \let\Huge=\huge
206 </12pt>

```

6.2 分段

`\lineskip` 当两行 (lines) 过于接近时, 这些参数控制 $\mathrm{T}_\mathrm{E}\mathrm{X}$ 的行为。

```

\normallineskip 207 < *article | report | book >
                208 \setlength\lineskip{1\p@}
                209 \setlength\normallineskip{1\p@}

```

`\baselinestretch` 这用作 `\baselineskip` 的乘数 (multiplier)。默认设置是不拉伸 (stretch) 基线 (baselines)。请注意, 如果此命令未解析为“空 (empty)”, 则会忽略 `\baselineskip` 规范中的任何 plus 和 minus 部分。

```

210 \renewcommand\baselinestretch{}

```

`\parskip` `\parskip` 在段落之间提供额外的垂直间距 (vertical space), `\parindent` 是段落缩进 (paragraph indentation) 的宽度。`\parindent` 的值取决于我们是否处于双栏模式 (two-column mode)。

```

211 \setlength\parskip{0\p@ \@plus \p@}
212 </article | report | book >
213 < *10pt | 11pt | 12pt >
214 \if@twocolumn
215   \setlength\parindent{1em}
216 \else
217 <10pt>   \setlength\parindent{15\p@}
218 <11pt>   \setlength\parindent{17\p@}
219 <12pt>   \setlength\parindent{1.5em}
220 \fi
221 </10pt | 11pt | 12pt >

```

`\smallskipamount` 这三个参数的值是在 $\mathrm{L}^{\mathrm{A}}\mathrm{T}_\mathrm{E}\mathrm{X}$ 内核中设置的。根据指定的尺寸选项 (size option), `\medskipamount` 它们可能会有所不同。但由于无论尺寸选项如何, 它们始终具有相同的值, 因此 `\bigskipamount` 此我们不会更改它们以保持与 $\mathrm{L}^{\mathrm{A}}\mathrm{T}_\mathrm{E}\mathrm{X}$ 2.09 和旧版本的 $\mathrm{L}^{\mathrm{A}}\mathrm{T}_\mathrm{E}\mathrm{X}_2\epsilon$ 兼容。

```

222 <*10pt | 11pt | 12pt>
223 \setlength\smallskipamount{3\p@ \@plus 1\p@ \@minus 1\p@}
224 \setlength\medskipamount{6\p@ \@plus 2\p@ \@minus 2\p@}
225 \setlength\bigskipamount{12\p@ \@plus 4\p@ \@minus 4\p@}
226 </10pt | 11pt | 12pt>

```

`\@lowpenalty` 命令 `\nopagebreak` 和 `\nolinebreak` 设置了惩罚 (penalty)，以阻止这些中断 (breaks) 在其放置点发生。他们使用 `\@lowpenalty`、`\@medpenalty` 或 `\@highpenalty`，这取决于他们的参数。

```

227 <*article | report | book>
228 \@lowpenalty 51
229 \@medpenalty 151
230 \@highpenalty 301

```

`\clubpenalty` 这些惩罚 (penalty) 用于阻止段末短行 (club lines) 和寡行 (widow lines)¹。因为 `\widowpenalty` 我们使用它们的默认值，所以这里只显示它们，注释掉了。

```

231 % \clubpenalty 150
232 % \widowpenalty 150

```

`\displaywidowpenalty` 在数学显示 (math display) 前阻止寡行 (widow lines)(但不要太多)，禁止在显示前直接中断 (breaking)。允许在显示后中断而不受惩罚 (penalty)。再次使用 `\predisplaypenalty` 默认值，因此我们只在此处显示它们。

```

233 % \displaywidowpenalty 50
234 % \predisplaypenalty 10000
235 % \postdisplaypenalty 0

```

`\interlinepenalty` 允许在段落中间分页 (breaking of a page)。

```

236 % \interlinepenalty 0

```

`\brokenpenalty` 我们允许在用连字符连接的行 (hyphenated line) 之后分页。

```

237 % \brokenpenalty 100
238 </article | report | book>

```

¹[译者注] 寡行 (widow lines): 指的是在新页面顶部单独出现的段落的最后一行。(这一行来自于上一个页面的段落)

6.3 页面布局

所有页边距尺寸 (margin dimensions) 均从距页面顶部和左侧 (lefthand) 一英寸的点开始测量。

6.3.1 垂直间距

`\headheight` `\headheight` 栏外标题 (running head, 即短标题) 所在的盒子的高度。
`\headsep` `\headsep` 是栏外标题的底部和文本顶部之间的距离。`\topskip` 是页面第一行的 `\baselineskip`; 如果值为 0pt, 那么 L^AT_EX 的输出程序 (output routine) 将无法正常工作, 因此不要这样做!

```
239 <*10pt | 11pt | 12pt>
240 \setlength\headheight{12\p@}
241 <!bk> \setlength\headsep {25\p@}
242 <10pt & bk> \setlength\headsep { .25in}
243 <11pt & bk> \setlength\headsep { .275in}
244 <12pt & bk> \setlength\headsep { .275in}
245 <10pt> \setlength\topskip {10\p@}
246 <11pt> \setlength\topskip {11\p@}
247 <12pt> \setlength\topskip {12\p@}
```

`\footskip` 栏外脚注 (running footer) 所在盒子的基线 (baseline) 到最后一行文本基线之间的距离由 `\footskip` 控制。

```
248 <!bk> \setlength\footskip{30\p@}
249 <10pt & bk> \setlength\footskip{ .35in}
250 <11pt & bk> \setlength\footskip{ .38in}
251 <12pt & bk> \setlength\footskip{30\p@}
```

`\maxdepth` T_EX 原语寄存器 (primitive register) `\maxdepth` 有一个类似于 `\topskip` 的函数。`\@maxdepth` 寄存器应该始终包含 `\maxdepth` 的副本。这是通过在 `\begin{document}` 处的内部设置它来实现的。在 plain T_EX 和 L^AT_EX 2.09 中, `\maxdepth` 都有一个固定的值 4pt; 在原生 L^AT_EX 2_ε 模式 (native L^AT_EX 2_ε mode) 中, 我们让值取决于字体尺寸 (typesize)。我们将其设置为 `\maxdepth + \topskip = typesize × 1.5`。碰巧, 在这些类中, `\topskip` 等于 typesize, 因此我们将 `\maxdepth` 设置为 `\topskip` 值的一半。

```
252 \ifcompatibility \setlength\maxdepth{4\p@} \else
253 \setlength\maxdepth{.5\topskip} \fi
```

6.3.2 文本的尺寸

`\textwidth` 当我们处于兼容模式 (compatibility mode) 时, 我们必须确保打印区域的尺寸 (dimensions of the printed area) 与用户习惯看到的尺寸没有差异。

```
254 \if@compatibility
255   \if@twocolumn
256     \setlength\textwidth{410\p@}
257   \else
258     <10pt & !bk>   \setlength\textwidth{345\p@}
259     <11pt & !bk>   \setlength\textwidth{360\p@}
260     <12pt & !bk>   \setlength\textwidth{390\p@}
261     <10pt & bk>    \setlength\textwidth{4.5in}
262     <11pt & bk>    \setlength\textwidth{5in}
263     <12pt & bk>    \setlength\textwidth{5in}
264   \fi
```

当我们不是处于兼容模式时, 我们可以设置一些不同的尺寸 (dimensions), 例如考虑纸张尺寸 (paper size)。

```
265 \else
```

首先, 我们计算所选纸张上允许的最大 `\textwidth`, 并将其存储在 `\@tempdima` 中。然后, 我们在 `\@tempdimb` 中存储大约 60–70 个字符的行长度 (length of a line)。当使用计算机现代字体 (Computer Modern fonts) 时, 给出的值或多或少是合适的。

```
266   \setlength\@tempdima{\paperwidth}
267   \addtolength\@tempdima{-2in}
268   <10pt>   \setlength\@tempdimb{345\p@}
269   <11pt>   \setlength\@tempdimb{360\p@}
270   <12pt>   \setlength\@tempdimb{390\p@}
```

现在我们可以设置 `\textwidth`, 这取决于我们将设置一栏还是两栏。

在双栏模式 (two-column mode) 中, 每一栏的宽度不应超过 `\@tempdimb` (例如, 这可能发生在 A3 纸张上)。

```
271   \if@twocolumn
272     \ifdim\@tempdima>2\@tempdimb\relax
273       \setlength\textwidth{2\@tempdimb}
274     \else
275       \setlength\textwidth{\@tempdima}
276     \fi
```

在单栏模式 (one-column mode) 中, 文本的宽度不应超过最小纸张宽度

(paperwidth)(页边空白为-2 英寸) 和由字符数 (number of characters) 定义的一行的最大长度。

```

277 \else
278   \ifdim\@tempdima>\@tempdimb\relax
279     \setlength\textwidth{\@tempdimb}
280   \else
281     \setlength\textwidth{\@tempdima}
282   \fi
283 \fi
284 \fi

```

这里我们稍微修改一下文本的宽度 (width of the text), 使之成为一个完整的点数 (number of points)。

```

285 \if@compatibility\else
286   \@settopoint\textwidth
287 \fi

```

`\textheight` 现在我们已经计算了文本的宽度, 我们必须考虑高度 (height)。`\textheight` 是文本的高度, 包括脚注 (footnotes) 和图形 (figures), 不包括栏外标题 (running head) 和栏外脚注 (running foot)。

首先确保兼容模式 (compatibility mode) 的尺寸 (dimensions) 与 L^AT_EX2.09 的尺寸相同。行数的计算方法是旧的 `\textheight` 减去 `\topskip` 的下限 (floor), 再除以 `\baselineskip` 得到的 `\normalsize`。旧的 `\textheight` 值是 528pt

```

288 \if@compatibility
289 <10pt & !bk> \setlength\textheight{43\baselineskip}
290 <10pt & bk> \setlength\textheight{41\baselineskip}
291 <11pt> \setlength\textheight{38\baselineskip}
292 <12pt> \setlength\textheight{36\baselineskip}

```

同样, 我们根据纸张尺寸 (papersize) 和使用的基线间距 (baselineskip) 来计算这个值, 以便在页面上有完整的行数 (whole number of lines)。

```

293 \else
294   \setlength\@tempdima{\paperheight}

```

我们在页面的顶部和底部留有至少 1 英寸的边距 (margin)。

```

295   \addtolength\@tempdima{-2in}

```

我们还必须为栏外标题 (running headers) 和栏外页脚 (running footers) 留出空间。

```

296   \addtolength\@tempdima{-1.5in}

```

然后, 我们将结果除以当前的 `\baselineskip`, 并将其存储在计数寄存器 (count register) `\@tempcnta` 中, 该寄存器随后包含适合此页面的行数。

```
297 \divide\@tempdima\baselineskip
298 \@tempcnta=\@tempdima
```

由此我们可以计算文本的高度 (height)。

```
299 \setlength\textheight{\@tempcnta\baselineskip}
300 \fi
```

页面上的第一行的高度为 `\topskip`。

```
301 \addtolength\textheight{\topskip}
```

6.3.3 边距

现在, 这些参数的大多数值都是根据所使用的纸张尺寸 (papersize) 进行计算的。在计算中, 需要考虑 `\marginparsep`, 因此我们首先给它赋值。

`\marginparsep` 主文本 (main text) 和边距注释 (marginal notes) 之间的水平间距 (horizontal space) 由 `\marginparsep` 决定, 两个边距注释间的最小垂直间距 (minimum vertical separation) 由 `\marginparpush` 控制。

```
302 \if@twocolumn
303 \setlength\marginparsep {10\p@}
304 \else
305 <10pt&!bk> \setlength\marginparsep{11\p@}
306 <11pt&!bk> \setlength\marginparsep{10\p@}
307 <12pt&!bk> \setlength\marginparsep{10\p@}
308 <bk> \setlength\marginparsep{7\p@}
309 \fi
310 <10pt | 11pt> \setlength\marginparpush{5\p@}
311 <12pt> \setlength\marginparpush{7\p@}
```

现在我们可以给出其他边距参数 (margin parameters) 的值。对于原生 (native) L^AT_EX 2_ε, 计算这些值。

`\oddsidemargin` 首先, 我们给出兼容模式 (compatibility mode) 的值。

`\evensidemargin` 双面打印 (two-sided printing) 的值:

```
\marginparwidth 312 \if@compatibility
313 <*bk>
314 <10pt> \setlength\oddsidemargin {.5in}
315 <11pt> \setlength\oddsidemargin {.25in}
316 <12pt> \setlength\oddsidemargin {.25in}
```

```

317 <10pt>    \setlength\evensidemargin {1.5in}
318 <11pt>    \setlength\evensidemargin {1.25in}
319 <12pt>    \setlength\evensidemargin {1.25in}
320 <10pt>    \setlength\marginparwidth {.75in}
321 <11pt>    \setlength\marginparwidth {1in}
322 <12pt>    \setlength\marginparwidth {1in}
323 </bk>
324 <*\bk>
325 \if@twoside
326 <10pt>    \setlength\oddsidemargin {44\p@}
327 <11pt>    \setlength\oddsidemargin {36\p@}
328 <12pt>    \setlength\oddsidemargin {21\p@}
329 <10pt>    \setlength\evensidemargin {82\p@}
330 <11pt>    \setlength\evensidemargin {74\p@}
331 <12pt>    \setlength\evensidemargin {59\p@}
332 <10pt>    \setlength\marginparwidth {107\p@}
333 <11pt>    \setlength\marginparwidth {100\p@}
334 <12pt>    \setlength\marginparwidth {85\p@}

```

单面打印 (one-sided printing) 的值:

```

335 \else
336 <10pt>    \setlength\oddsidemargin {63\p@}
337 <11pt>    \setlength\oddsidemargin {54\p@}
338 <12pt>    \setlength\oddsidemargin {39.5\p@}
339 <10pt>    \setlength\evensidemargin {63\p@}
340 <11pt>    \setlength\evensidemargin {54\p@}
341 <12pt>    \setlength\evensidemargin {39.5\p@}
342 <10pt>    \setlength\marginparwidth {90\p@}
343 <11pt>    \setlength\marginparwidth {83\p@}
344 <12pt>    \setlength\marginparwidth {68\p@}
345 \fi
346 </!bk>

```

两栏模式 (two-column mode) 的值:

```

347 \if@twocolumn
348     \setlength\oddsidemargin {30\p@}
349     \setlength\evensidemargin {30\p@}
350     \setlength\marginparwidth {48\p@}
351 \fi

```

当我们不处于兼容模式时, 我们可以考虑所选纸张的尺寸。

`\oddsidemargin` 和 `\marginparwidth` 的值将根据 `\if@twoside` 的状态

设置。

如果 `@twoside` 为 `true`(对于 `book` 总是如此), 我们将使内页边距 (`inner margin`) 小于外页边距 (`outer margin`)。

```
352 \else
353   \if@twoside
354     \setlength\@tempdima      {\paperwidth}
355     \addtolength\@tempdima    {-\textwidth}
356     \setlength\oddsidemargin  {.4\@tempdima}
357     \addtolength\oddsidemargin {-1in}
```

文本的边距宽度 (`width of the margin`) 设置为宽度的余数 (`remainder`), 但宽度为 `0.4in` 英寸的空白 “实际边距 (`real margin`)” 除外。也许应该内置一个检查 (`check`), 以确保 (文本) 边距宽度不会太小!

```
358     \setlength\marginparwidth {.6\@tempdima}
359     \addtolength\marginparwidth {-\marginparsep}
360     \addtolength\marginparwidth {-0.4in}
```

对于单面打印 (`one-sided printing`), 我们通过计算 `\textwidth` 和 `\paperwidth` 之间的差值将文本居中放置在页面上。该差值的一半用于边距 (`margin`)(因此 `\oddsidemargin` 小于 `1` 英寸)。

```
361 \else
362   \setlength\@tempdima      {\paperwidth}
363   \addtolength\@tempdima    {-\textwidth}
364   \setlength\oddsidemargin  {.5\@tempdima}
365   \addtolength\oddsidemargin {-1in}
366   \setlength\marginparwidth {.5\@tempdima}
367   \addtolength\marginparwidth {-\marginparsep}
368   \addtolength\marginparwidth {-0.4in}
369   \addtolength\marginparwidth {-.4in}
370 \fi
```

使用上述算法, `\marginparwidth` 可以变得相当大, 这是我们不希望看到的。

```
371 \ifdim \marginparwidth >2in
372   \setlength\marginparwidth{2in}
373 \fi
```

完成这些计算之后, 我们将它们设置为 `pt` 值。

```
374 \@settopoint\oddsidemargin
375 \@settopoint\marginparwidth
```

现在可以根据上面设置的值计算 `\evensidemargin`。

```
376 \setlength\evensidemargin {\paperwidth}
```

```

377 \addtolength\evensidemargin{-2in}
378 \addtolength\evensidemargin{-\textwidth}
379 \addtolength\evensidemargin{-\oddsidemargin}

```

将 `\evensidemargin` 设置为整点值 (full point value) 可能会产生一个小的误差 (small error)。然而, 该误差在当今技术的双面打印机 (doublesided printer) 能够精确打印的误差范围 (error range) 内。

```

380 \@settopoint\evensidemargin
381 \fi

```

`\topmargin` `\topmargin`(上边距) 是“可打印区域 (printable area)”顶部 (低于纸张顶部 1 英寸) 与栏外标题 (running head) 所在的盒子顶部之间的距离。

现在可以根据上面设置的值计算它。

```

382 \if@compatibility
383 <!bk> \setlength\topmargin{27pt}
384 <10pt & bk> \setlength\topmargin{.75in}
385 <11pt & bk> \setlength\topmargin{.73in}
386 <12pt & bk> \setlength\topmargin{.73in}
387 \else
388 \setlength\topmargin{\paperheight}
389 \addtolength\topmargin{-2in}
390 \addtolength\topmargin{-\headheight}
391 \addtolength\topmargin{-\headsep}
392 \addtolength\topmargin{-\textheight}
393 \addtolength\topmargin{-\footskip} % 这可能是错误的!

```

通过改变下一行中的因子 (factor), 整个页面可以垂直移动 (shifted vertically)。

```

394 \addtolength\topmargin{-.5\topmargin}
395 \@settopoint\topmargin
396 \fi

```

6.3.4 脚注

`\footnotesep` `\footnotesep` 是每个脚注开头的支柱 (strut) 的高度。它等于此类中 `\footnotesize` 的支柱的高度, 因此脚注之间没有额外的空间 (extra space)。

```

397 <10pt> \setlength\footnotesep{6.65\p@}
398 <11pt> \setlength\footnotesep{7.7\p@}
399 <12pt> \setlength\footnotesep{8.4\p@}

```

`\footins` `\skip\footins` 是主文本 (main text) 最后一行和第一个脚注顶部之间的空间 (space)。

```
400 <10pt>\setlength{\skip\footins}{9\p@ \@plus 4\p@ \@minus 2\p@}  
401 <11pt>\setlength{\skip\footins}{10\p@ \@plus 4\p@ \@minus 2\p@}  
402 <12pt>\setlength{\skip\footins}{10.8\p@ \@plus 4\p@ \@minus 2\p@}  
403 </10pt | 11pt | 12pt>
```

6.3.5 浮动放置参数

在 L^AT_EX 2_ε 内核中, 所有的浮动参数 (float parameters) 都被赋予默认值。因此, 需要使用 `\renewcommand` 设置非计数器 (not counters) 的参数。

放置浮动对象的限制

`\c@topnumber` *topnumber* 计数器 (counter) 保存可出现在文本页面 (text page) 顶部的最大浮动数 (maximum number of floats)。

```
404 <*article | report | book>  
405 \setcounter{topnumber}{2}
```

`\topfraction` 这表示文本页面 (text page) 顶部的浮动可以占据的最大部分。

```
406 \renewcommand\topfraction{.7}
```

`\c@bottomnumber` *bottomnumber* 计数器 (counter) 保存可出现在文本页底部的最大浮动数 (maximum number of floats)。

```
407 \setcounter{bottomnumber}{1}
```

`\bottomfraction` 这表示文本页面 (text page) 底部的浮动可以占据的最大部分。

```
408 \renewcommand\bottomfraction{.3}
```

`\c@totalnumber` 这表示可以出现在任何文本页面上的最大浮动数。

```
409 \setcounter{totalnumber}{3}
```

`\textfraction` 这表示文本页面必须被文本占据的最小部分。

```
410 \renewcommand\textfraction{.2}
```

`\floatpagefraction` 这表示在生成“浮动页面 (float page)”之前必须被浮动对象 (floating objects) 占据的页面的最小部分 (minimum part of a page)。

```
411 \renewcommand\floatpagefraction{.5}
```


`\c@dbltopnumber` *dbltopnumber* 计数器 (counter) 保存可以出现在两栏文本页面 (two-column text page) 顶部的最大两栏浮点数 (maximum number of two-column floats)。

```
412 \setcounter{dbltopnumber}{2}
```

`\dbltopfraction` 这表示两栏文本页面 (two-column text page) 的最大部分可以被顶部的两栏浮动 (two-column floats) 所占据。

```
413 \renewcommand\dbltopfraction{.7}
```

`\dblfloatpagefraction` 这表示在生成“浮动页面 (float page)”之前，页面的最小部分 (minimum part of a page) 必须由两栏宽浮动对象 (two-column wide floating objects) 占据。

```
414 \renewcommand\dblfloatpagefraction{.5}
```

```
415 </article | report | book>
```

文本页面上的浮动

`\floatsep` 当浮动对象 (floating object) 放置在带有文本的页面上时，这些参数控制浮动对象和页面上其他对象之间的分隔 (separation)。这些参数用于单栏模式 (one-column mode) 和双栏模式 (two-column mode) 中的单栏浮点 (single-column floats)。

`\floatsep` 是移动到文本页面顶部或底部的相邻浮动 (adjacent floats) 之间的间距 (space)。

`\textfloatsep` 是主文本 (main text) 和页面顶部或底部浮动之间的间距 (space)。

`\intextsep` 是文本内浮动 (`\intextsep`) 和文本之间的间距。

```
416 <*10pt>
```

```
417 \setlength\floatsep {12\p@ \@plus 2\p@ \@minus 2\p@}
```

```
418 \setlength\textfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
```

```
419 \setlength\intextsep {12\p@ \@plus 2\p@ \@minus 2\p@}
```

```
420 </10pt>
```

```
421 <*11pt>
```

```
422 \setlength\floatsep {12\p@ \@plus 2\p@ \@minus 2\p@}
```

```
423 \setlength\textfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
```

```
424 \setlength\intextsep {12\p@ \@plus 2\p@ \@minus 2\p@}
```

```
425 </11pt>
```

```
426 <*12pt>
```

```
427 \setlength\floatsep {12\p@ \@plus 2\p@ \@minus 4\p@}
```

```
428 \setlength\textfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
```

```
429 \setlength\intextsep {14\p@ \@plus 4\p@ \@minus 4\p@}
```

```
430 </12pt>
```

`\dblfloatsep` 当处于双栏模式 (two-column mode) 时, 当跨越整个 `\textwidth` (文本宽度) 的浮动对象 (floating objects) 放置在文本页面上时, 浮动和文本之间的分隔由 `\dblfloatsep` 和 `\dbltextfloatsep` 控制。

`\dblfloatsep` 是移动到文本页面 (text page) 顶部或底部的相邻浮动 (adjacent floats) 之间间距 (space)。

`\dbltextfloatsep` 是主文本 (main text) 和页面顶部或底部浮动之间的间距 (space)。

```
431 <*10pt>
432 \setlength\dblfloatsep    {12\p@ \@plus 2\p@ \@minus 2\p@}
433 \setlength\dbltextfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
434 </10pt>
435 <*11pt>
436 \setlength\dblfloatsep    {12\p@ \@plus 2\p@ \@minus 2\p@}
437 \setlength\dbltextfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
438 </11pt>
439 <*12pt>
440 \setlength\dblfloatsep    {14\p@ \@plus 2\p@ \@minus 4\p@}
441 \setlength\dbltextfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
442 </12pt>
```

浮动在自己的页面或栏上

`\@fptop` 当浮动对象 (floating objects) 放置在单独的页面上时, 这些页面的布局由这些
`\@fpsep` 参数控制。在页面顶部插入了 `\@fptop` 的可伸缩空白 (stretchable whitespace),
`\@fpbot` 在页面底部插入了 `\@fpbot` 的可伸缩空白。在相邻的浮动 (adjacent floats) 之间插入 `\@fpsep`。

这些参数用于在单栏模式 (one-column mode) 中放置浮动对象, 或在两栏模式 (two-column mode) 中在单栏浮动 (single-column floats) 中放置浮动对象。

请注意, `\@fptop` 和 `\@fpbot` 两个参数中至少有一个应该包含 `plus ...fil`, 以便填充剩余的空白 (remaining empty space)。

```
443 <*10pt>
444 \setlength\@fptop{0\p@ \@plus 1fil}
445 \setlength\@fpsep{8\p@ \@plus 2fil}
446 \setlength\@fpbot{0\p@ \@plus 1fil}
447 </10pt>
448 <*11pt>
449 \setlength\@fptop{0\p@ \@plus 1fil}
```

```

450 \setlength\@fpsep{8\p@ \@plus 2fil}
451 \setlength\@fpbot{0\p@ \@plus 1fil}
452 </11pt>
453 <*12pt>
454 \setlength\@fptop{0\p@ \@plus 1fil}
455 \setlength\@fpsep{10\p@ \@plus 2fil}
456 \setlength\@fpbot{0\p@ \@plus 1fil}
457 </12pt>

```

`\@dblftop` 双栏模式 (two-column mode) 中的双栏浮动 (double-column floats) 使用类似 `\@dblpsep` 的参数处理。

```

\@dblpbot 458 <*10pt>
459 \setlength\@dblftop{0\p@ \@plus 1fil}
460 \setlength\@dblpsep{8\p@ \@plus 2fil}
461 \setlength\@dblpbot{0\p@ \@plus 1fil}
462 </10pt>
463 <*11pt>
464 \setlength\@dblftop{0\p@ \@plus 1fil}
465 \setlength\@dblpsep{8\p@ \@plus 2fil}
466 \setlength\@dblpbot{0\p@ \@plus 1fil}
467 </11pt>
468 <*12pt>
469 \setlength\@dblftop{0\p@ \@plus 1fil}
470 \setlength\@dblpsep{10\p@ \@plus 2fil}
471 \setlength\@dblpbot{0\p@ \@plus 1fil}
472 </12pt>
473 <*article | report | book>

```

6.4 页面样式

页面样式 (page style) *foo* 是通过定义 `\ps@foo` 命令来定义的。此命令应仅进行局部定义 (local definitions)。定义中不应该夹杂空格 (stray spaces)，因为它们可能导致输出中出现神秘的额外空格 (mysterious extra spaces)(这是应该始终避免的)。

`\@evenhead` `\ps@...` 命令定义 `\@oddhead`、`\@oddfoot`、`\@evenhead`等宏，`\@evenfoot` 用于 `\@oddhead` 定义栏外标题 (running heads) 和流水脚注 (running feet)—例如，`\@oddhead` `\@evenfoot` 是生成奇数页 (odd-numbered pages) 标题盒子 (heading box) 内容的宏。它 `\@oddfoot` 在一个宽度为 `\textwidth` 的 `\hbox` 中被调用。

6.4.1 标记约定

要使标题 (headings) 由分节命令 (sectioning commands) 确定, 页面样式 (page style) 定义了命令 `\chaptermark`、`\sectionmark`、..., 其中 `\chaptermark{<TEXT>}` 被 `\chapter` 调用以设置标记 (mark), 依此类推

在下列宏的帮助下定义了 `\...mark` 命令和 `\...head` 宏。(所有 `\...mark` 命令都应该初始化为 no-ops。)

L^AT_EX 通过使用以下命令生成 “left(左)” 和 “right(右)” 两种标记 (mark), 扩展了 T_EX 的标记功能 (`\mark facility`):

`\markboth{<LEFT>}{<RIGHT>}`: 添加两个标记。

`\markright{<RIGHT>}`: 添加 “right(右)” 标记。

`\leftmark`: 在 `\@oddhead`、`\@oddfoot`、`\@evenhead` 或 `\@evenfoot` 宏中使用时, 它获得当前的 “left(左)” 标记。`\leftmark` 的工作方式类似于 T_EX 的 `\botmark` 命令。

`\rightmark`: 在 `\@oddhead`、`\@oddfoot`、`\@evenhead` 或 `\@evenfoot` 宏中使用时, 它获得当前的 “right(右)” 标记。`\rightmark` 的工作方式类似于 T_EX 的 `\firstmark` 命令。

标记命令 (marking commands) 对于 “编号在” 左标记内 (“numbered within” left marks) 的右标记 (right marks) 工作得相当好—例如, 左标记由 `\chapter` 命令更改, 右标记由 `\section` 命令更改。但是, 如果两个 `\markboth` 同时出现在同一个页面上时, 它确实会产生一些异常的结果。

像 `\tableofcontents` 这样的命令应该使用 `\mkboth` 命令在某些页面样式 (page styles) 中设置标记 (set the marks), 页面样式命令 (pagestyle command)(`\ps@...`) `\let`(让) `\mkboth` 变成 `\markboth` 用于设置标题 (heading), 或者 `\let`(让) `\mkboth` 变成 `\gobbletwo` 设置为不做任何事情。

[译者注] 感觉前一段可能翻译错误! 前一段的原文是: Commands like `\tableofcontents` that should set the marks in some page styles use a `\mkboth` command, which is `\let` by the pagestyle command (`\ps@...`) to `\markboth` for setting the heading or to `\gobbletwo` to do nothing.

6.4.2 定义页面样式

页面样式 (pagestyles) 的 *empty*(空) 和 *plain*(普通) 是在 L^AT_EX 格式中定义的。

`\ps@headings` 双面打印 (two sided printing) 与单面打印 (one sided printing) 的页面样式 (page style) *headings* 的定义必须不同。

```
474 \if@twoside
475   \def\ps@headings{%
```

在这种页面样式中, 流水脚注 (running feet) 是空的 (empty), 栏外标题 (running head, 即短标题) 包含页码 (page number) 和其中一个标记 (marks)。

```
476     \let\@oddfoot\@empty\let\@evenfoot\@empty
477     \def\@evenhead{\thepage\hfil\slshape\leftmark}%
478     \def\@oddhead{\slshape\rightmark}\hfil\thepage}%
```

使用此页面样式时, 栏外标题 (running head, 即短标题) 的内容由章标题 (chapter titles) 和节标题 (section titles) 决定。因此, 我们 `\let` (让) `\@mkboth` 变成 `\markboth`。

```
479     \let\@mkboth\markboth
```

对于 `article` 文档类, 我们定义 `\sectionmark` 来清除右标记 (right mark), 并将节的编号 (当节被编号时) 及其标题放在左标记 (left mark) 中。右标记由 `\subsectionmark` 设置, 以包含小节标题 (subsection titles)。

请注意, `\sectionmark` 命令的参数使用 `##1`, 该参数将在执行 `\ps@headings` 时定义。

```
480 \begin{document}
481   \def\sectionmark##1{%
482     \markboth {\MakeUppercase{%
483       \ifnum \c@secnumdepth >\z@
484         \thesection\quad
485         \fi
486       ##1}}{}}%
487   \def\subsectionmark##1{%
488     \markright {%
489       \ifnum \c@secnumdepth >\@ne
490         \thesubsection\quad
491         \fi
492       ##1}}%
493 \end{document}
```

在 `report` 和 `book` 文档类中, 我们使用 `\chaptermark` 和 `\sectionmark` 宏来填充栏外标题 (running head, 即短标题)。

请注意, `\chaptermark` 命令的参数使用 `##1`, 该参数将在执行 `\ps@headings` 时定义。

```

494 <*report | book>
495     \def\chaptermark##1{%
496         \markboth {\MakeUppercase{%
497             \ifnum \c@secnumdepth >\m@ne
498 <book>             \if@mainmatter
499                 \@chapapp\ \thechapter. \ %
500 <book>             \fi
501                 \fi
502             ##1}}{}}%
503     \def\sectionmark##1{%
504         \markright {\MakeUppercase{%
505             \ifnum \c@secnumdepth >\z@
506                 \thesection. \ %
507             \fi
508             ##1}}}}
509 </report | book>

```

对于单面打印 (one sided printing), \ps@headings 的定义可以简单得多, 因为我们对偶数页和奇数页的处理是一样的。因此, 我们不需要定义 \@even...。

```

510 \else
511     \def\ps@headings{%
512         \let\@oddfoot\@empty
513         \def\@oddhead{\slshape\rightmark\hfil\thepage}%
514         \let\@mkboth\markboth

```

我们现在使用 \markright, 而不是像双面打印那样使用 \markboth。

```

515 <*article>
516     \def\sectionmark##1{%
517         \markright {\MakeUppercase{%
518             \ifnum \c@secnumdepth >\m@ne
519                 \thesection\quad
520             \fi
521             ##1}}}}
522 </article>

523 <*report | book>
524     \def\chaptermark##1{%
525         \markright {\MakeUppercase{%
526             \ifnum \c@secnumdepth >\m@ne
527 <book>             \if@mainmatter
528                 \@chapapp\ \thechapter. \ %

```

```

529 <book>          \fi
530          \fi
531          ##1}}}}
532 </report | book>
533 \fi

```

`\ps@myheadings` 页面样式 (page style) `myheadings` 的定义相当简单, 因为用户可以使用 `\markboth` 和 `\markright` 命令自行确定栏外标题 (running head, 即短标题)。

```

534 \def\ps@myheadings{%
535     \let\@oddfoot\@empty\let\@evenfoot\@empty
536     \def\@evenhead{\thepage\hfil\slshape\leftmark}%
537     \def\@oddhead{\slshape\rightmark}\hfil\thepage}%

```

我们必须确保禁用章节标题所使用的标记命令 (marking commands)。我们这样做, `\let` (让) 它们进入一个吞噬其参数的宏。

```

538     \let\@mkboth\@gobbletwo
539 <!article>      \let\chaptermark\@gobble
540     \let\sectionmark\@gobble
541 <article>       \let\subsectionmark\@gobble
542     }

```

7 文档标记

7.1 标题

`\title` 这三个宏 (macros) 由 L^AT_EX 格式提供, 用于提供有关文档的标题 (title)、作者 `\author` (authors) 和日期 (date) 的信息。信息存储在内部控制序列 (internal control sequences) 中。`\maketitle` 命令的任务是使用提供的信息。这里显示了这些宏的定义以供参考。

```
543 % \DeclareRobustCommand*\title}[1]{\gdef\@title{#1}}
544 % \DeclareRobustCommand*\author}[1]{\gdef\@author{#1}}
545 % \DeclareRobustCommand*\date}[1]{\gdef\@date{#1}}
```

默认情况下, `\date` 宏获取今天的日期。

```
546 % \date{\today}
```

`\maketitle` `\maketitle` 的定义取决于是否创建单独的标题页 (title page)。这是 report 和 book 文档类的默认值, 但是对于 article 类, 它是可选的。

制作标题页时, 我们在局部重新定义 `\footnotesize` 和 `footnoterule`, 以更改 `\thanks` 命令生成的脚注的外观; 这些变化影响到所有脚注。

```
547 \if@titlepage
548   \newcommand\maketitle{\begin{titlepage}%
549     \let\footnotesize\small
550     \let\footnoterule\relax
551     \let \footnote \thanks
```

我们将整个标题垂直居中; 通过添加一个 `\vskip`, 可以稍微调整中心位置。(在兼容模式下, 标题页环境将页码设置为 0, 以保持 L^AT_EX 2.09 样式文件的行为。)

```
552   \null\vfil
553   \vskip 60\p@
```

然后我们用 `\LARGE` 字体设置标题 (title); 留出一点空间 (space), 用 `\large` 字体设置作者 (author)。我们在表格环境 (tabular environment) 中执行此操作, 以便将它们放在单栏 (single column) 中。在 date(日期) 之前, 我们又留了一点空白 (whitespace)。

```
554   \begin{center}%
555     {\LARGE \@title \par}%
556     \vskip 3em%
557     {\large
558       \lineskip .75em%
```



```

559      \begin{tabular}[t]{c}%
560      \@author
561      \end{tabular}\par}%
562      \vskip 1.5em%
563      {\large \@date \par}%      % Set date in \large size.
564      \end{center}\par

```

然后，我们调用 `\@thanks` 打印进入脚注中的信息并完成页面。

```

565      \@thanks
566      \vfil\null
567      \end{titlepage}%

```

我们重置 *footnote* 计数器 (counter)，禁用 `\thanks` 和 `\maketitle`，并通过清空内部信息宏 (internal information macros) 来节省一些存储空间 (storage space)。

```

568      \setcounter{footnote}{0}%
569      \global\let\thanks\relax
570      \global\let\maketitle\relax
571      \global\let\@thanks\@empty
572      \global\let\@author\@empty
573      \global\let\@date\@empty
574      \global\let\@title\@empty

```

设置标题后，声明命令 `\title` 等可能会消失 (vanish)。`\and` 的定义只有在 `\author` 的参数中才有意义，所以也可以这样做。

```

575      \global\let\title\relax
576      \global\let\author\relax
577      \global\let\date\relax
578      \global\let\and\relax
579      }

```

当标题不在自己的页面上时，标题的布局略有不同。我们用符号 (symbols) 来标记脚注 (footnotes)，我们必须处理两栏文档 (two-column documents)。

因此，我们首先启动一个新组 (new group)，以保持局部更改 (changes local)。然后我们重新定义 `\thefootnote` 以使用 `\fnsymbol`；并更改 `\@makefnmark` 使脚注标记 (footnotemarks) 的宽度为零 (以使作者姓名居中看起来更好)。

```

580 \else
581   \newcommand\maketitle{\par
582     \begingroup
583       \renewcommand\thefootnote{\fnsymbol\c@footnote}%
584       \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
585       \long\def\@makefntext##1{\parindent 1em\noindent

```

```

586          \hb@xt@1.8em{%
587          \hss\@textsuperscript{\normalfont\@thefnmark}}##1}%

```

如果这是一个两栏文档 (two-column document), 我们将以两栏模式 (two-column mode) 开始一个新页面, 标题设置为文本的全宽。标题信息的实际打印由 \@maketitle 完成。

```

588      \if@twocolumn
589      \ifnum \col@number=\@ne
590      \@maketitle
591      \else
592      \twocolumn[\@maketitle]%
593      \fi
594      \else

```

当这不是一个两栏文档时, 我们只需开始一个新页面, 防止浮动对象 (floating objects) 出现在此页面顶部并打印标题信息 (title information)。

```

595      \newpage
596      \global\@topnum\z@    % Prevents figures from going at top of page.
597      \@maketitle
598      \fi

```

此页面有一个 *plain* 的布局。我们称 \@thanks 生成脚注。

```

599      \thispagestyle{plain}\@thanks

```

现在, 我们可以关闭组 (group), 重置 *footnote* 计数器, 禁用 \@thanks、\maketitle 和 \@maketitle, 并通过清空内部信息宏 (internal information macros) 来节省一些存储空间 (storage space)。

```

600      \endgroup
601      \setcounter{footnote}{0}%
602      \global\let\thanks\relax
603      \global\let\maketitle\relax
604      \global\let\@maketitle\relax
605      \global\let\@thanks\@empty
606      \global\let\@author\@empty
607      \global\let\@date\@empty
608      \global\let\@title\@empty
609      \global\let\title\relax
610      \global\let\author\relax
611      \global\let\date\relax
612      \global\let\and\relax
613      }

```

`\@maketitle` 当我们没有单独的标题页时，这个宏负责格式化标题信息。

我们总是开始一个新的页面，留下一些空白 (white space) 并将信息居中。标题以 `\LARGE` 字体设置，作者姓名 (author names) 和日期 (date) 以 `\large` 字体设置。

```
614 \def\@maketitle{%
615   \newpage
616   \null
617   \vskip 2em%
618   \begin{center}%
619   \let \footnote \thanks
620   {\LARGE \@title \par}%
621   \vskip 1.5em%
622   {\large
623     \lineskip .5em%
624     \begin{tabular}[t]{c}%
625       \@author
626     \end{tabular}\par}%
627   \vskip 1em%
628   {\large \@date}%
629   \end{center}%
630   \par
631   \vskip 1.5em}
632 \fi
```

7.2 章和节

7.2.1 构建块

类文件 (class file) 这一部分中的定义使用了两个内部宏 (internal macros) `\@startsection` 和 `\secdef`。为了理解这里发生了什么，我们描述了它们的语法。

`\@startsection` 宏有 6 个必需参数 (required arguments)，可以选择后跟一个 *、可选参数 (optional argument) 和必需参数：

```
\@startsection<name><level><indent><beforeskip><afterskip><style> optional
*
```

[<altheading>]<heading>

这是一个用于启动节 (section) 的通用命令 (generic command)，参数具有以下含义：

⟨*name*⟩ 用户级命令的名称, 例如 “section”。

⟨*level*⟩ 表示节深度的数字—例如, chapter=1、section = 2 等。当且仅当 ⟨*level*⟩
≤ *secnumdepth* 计数器的值时, 将打印节编号 (section number)。

⟨*indent*⟩ 标题从左边距的缩进。

⟨*beforeskip*⟩ 此参数的绝对值 (absolute value) 允许在标题 (heading) 上方放置垂直空间 (skip)。如果为负值 (negative), 则紧跟标题后面的文本段落缩进 (indent) 将被抑制。

⟨*afterskip*⟩ 如果为正值 (positive), 则允许在标题 (heading) 下方放置垂直空间 (skip)。否则允许在参选标题 (run-in heading) 的右侧放置垂直空间 (skip)。

⟨*style*⟩ 设置标题样式 (style of the heading) 的命令

* 当此项缺失时, 将对标题进行编号, 并递增相应的计数器 (counter)。

⟨*altheading*⟩ 提供一个可在目录 (table of contents) 和栏外标题 (running heads) 中使用的替代标题 (alternative heading)。使用 * 形式时, 不应出现这种情况。

⟨*heading*⟩ 新节 (new section) 的标题 (heading)。

分段命令 (sectioning command) 通常定义为 \@startsection 及其前六个参数。

当不使用 \@startsection 来定义分节命令 (sectioning command) 时, 可以使用宏 \secdef。它有两个参数:

```
\secdef⟨unstarcmds⟩⟨starcmds⟩
```

⟨*unstarcmds*⟩ 用于分节命令 (sectioning command) 的正常形式 (normal form)。

⟨*starcmds*⟩ 用于分节命令 (sectioning command) 的 * 形式。

您可以按如下方式使用 \secdef:

```
\def\chapter { ... \secdef \CMDA \CMDB }  
\def\CMDA    [#1]#2{ ... } % Command to define  
                        % \chapter[...]{...}  
\def\CMDB    #1{ ... }   % Command to define  
                        % \chapter*{...}
```

7.2.2 标记命令

`\chaptermark` `\...mark` 命令的默认初始化 (default initializations)。这些命令用于定义页面样式 (page styles)(见第 6.4.2 节), 其中大多数已经由 L^AT_EX 格式定义, 因此仅在此处显示。

```
\subsubsectionmark 633 <!article>\newcommand*\chaptermark[1]{}
\paragraphmark      634 % \newcommand*\sectionmark[1]{}
\subparagraphmark    635 % \newcommand*\subsectionmark[1]{}
                     636 % \newcommand*\subsubsectionmark[1]{}
                     637 % \newcommand*\paragraphmark[1]{}
                     638 % \newcommand*\subparagraphmark[1]{}

```

7.2.3 定义计数器

`\c@secnumdepth` 计数器 `secnumdepth` 的值给出了用于生成节号 (section numbers) 的最高级别的分节命令 (sectioning command) 的深度 (depth)。

```
639 <article>\setcounter{secnumdepth}{3}
640 <!article>\setcounter{secnumdepth}{2}

```

`\c@part` 这些计数器 (counters) 用于节编号 (section numbers)。宏 `\newcounter{<newctr>}[<oldctr>]`

`\c@chapter` 定义 `<newctr>` 为计数器 (counter), 当计数器 `<oldctr>` 被步进时 (stepped),

`\c@section` `<newctr>` 重置为零 (reset to zero)。`<oldctr>` 必须已定义。

```
\c@subsection 641 \newcounter {part}
\c@subsubsection 642 <article>\newcounter {section}
\c@paragraph    643 <*report | book>
\c@subparagraph 644 \newcounter {chapter}
                 645 \newcounter {section}[chapter]
                 646 </report | book>
                 647 \newcounter {subsection}[section]
                 648 \newcounter {subsubsection}[subsection]
                 649 \newcounter {paragraph}[subsubsection]
                 650 \newcounter {subparagraph}[paragraph]

```

`\thepart` 对于任何计数器 `CTR`, `\theCTR` 是定义计数器 `CTR` 的打印版本 (printed version) 的宏。它是根据以下宏来定义的:

`\thesection` `\arabic{COUNTER}` 将 `COUNTER` 的值打印为阿拉伯数字 (arabic numeral)。

`\thesubsubsection` `\roman{COUNTER}` 将 `COUNTER` 的值打印为小写罗马数字 (lowercase roman numeral)。

`\thesubparagraph`

`\Roman{COUNTER}` 将 *COUNTER* 的值打印为大写罗马数字 (uppercase roman numeral)。

`\alph{COUNTER}` 将 *COUNTER* 的值打印为小写字母 (lowercase letter):
1 = a、2 = b 等。

`\Alph{COUNTER}` 将 *COUNTER* 的值打印为大写字母 (uppercase letter):
1 = A、2 = B 等。

实际上, 为了节省空间, 我们使用了内部计数器表示法 (internal counter representations) 和操作这些表示法的命令。

```
651 \renewcommand \thepart {\@Roman\c@part}
652 \langle article \rangle \renewcommand \thesection {\@arabic\c@section}
653 \langle *report | book \rangle
654 \renewcommand \thechapter {\@arabic\c@chapter}
655 \renewcommand \thesection {\thechapter.\@arabic\c@section}
656 \langle /report | book \rangle
657 \renewcommand \thesubsection {\thesection.\@arabic\c@subsection}
658 \renewcommand \thesubsubsection {\thesubsection.\@arabic\c@subsubsection}
659 \renewcommand \theparagraph {\thesubsubsection.\@arabic\c@paragraph}
660 \renewcommand \thesubparagraph {\theparagraph.\@arabic\c@subparagraph}
```

`\@chapapp` `\@chapapp` 最初定义为 “`\chaptername`”。`\appendix` 命令将其重新定义为 “`\appendixname`”。

```
661 \langle report | book \rangle \newcommand \@chapapp {\chaptername}
```

7.2.4 前言、正文和附属资料

一本书 (book) 包含这三个 (逻辑) 节 (sections)²。如果我们正在处理正文 (main Matter), 则开关 `\@mainmatter` 为 true。当此开关为 false 时, `\chapter` 命令不打印章编号 (chapter numbers)。

在这里, 我们定义了启动这些节 (sections) 的命令。

`\frontmatter` 此命令启动罗马页码编号 (Roman page numbering) 并关闭章编号 (chapter numbering)。由于这会从 1 开始重新启动页面编号, 因此还应确保使用的是右页 (recto page)。

²[译者注] 这三个逻辑节 (logical sections) 就是: 前言 (front Matter)、正文 (main Matter)、附属资料 (back Matter)。前言包括诸如标题页、摘要、目录、序言、注释清单、数字清单和表格清单, 其中一些前题页如标题页传统上是不编号的; 附属资料可能包括词汇表、注释、参考书目和索引等。

```

662 <*book>
663 \newcommand\frontmatter{%
664 % \if@openright
665 \cleardoublepage
666 % \else
667 % \clearpage
668 % \fi
669 \@mainmatterfalse
670 \pagenumbering{roman}}

```

`\mainmatter` 此命令清除页面，开始阿拉伯数字页码编号 (arabic page numbering)，并打开章编号 (chapter numbering)。由于这会从 1 开始重新启动页面编号，因此还应确保使用的是右页 (recto page)。

```

671 \newcommand\mainmatter{%
672 % \if@openright
673 \cleardoublepage
674 % \else
675 % \clearpage
676 % \fi
677 \@mainmattertrue
678 \pagenumbering{arabic}}

```

`\backmatter` 这将清除页面，关闭章编号 (chapter numbering) 并保持页面编号不变。

```

679 \newcommand\backmatter{%
680 \if@openright
681 \cleardoublepage
682 \else
683 \clearpage
684 \fi
685 \@mainmatterfalse}
686 </book>

```

7.2.5 部分

`\part` 启动文档新部分 (new part) 的命令。

在 article 类中，`\part` 的定义相当简单；我们开始一个新段落 (new paragraph)，添加一点空白 (white space)，取消第一段的缩进，并使用 `\secdef`。与其他分节命令 (sectioning commands)($\text{\LaTeX 2}_{\epsilon}$ 内核中的 cf. `\@startsection`) 一样，我们需要检查 `@noskipsec` 开关并强制水平模式 (horizontal mode)(如果设置了)。

```

687 <*article>
688 \newcommand\part{%
689     \if@noskipsec \leavevmode \fi
690     \par
691     \addvspace{4ex}%
692     \@afterindentfalse
693     \secdef\@part\@spart}
694 </article>

```

对于 report 和 book 类，我们的做法有些不同。

我们启动一个新的 (右边的) 页面并使用 *plain* 页面样式 (pagestyle)。

```

695 <*report | book>
696 \newcommand\part{%
697     \if@openright
698         \cleardoublepage
699     \else
700         \clearpage
701     \fi
702     \thispagestyle{plain}%

```

当我们制作两栏文档 (two-column document) 时，这将是一个单栏页面 (one column page)。我们使用 @tempswa 来记住切换回两栏。

```

703     \if@twocolumn
704         \onecolumn
705         \@tempswatrue
706     \else
707         \@tempswafalse
708     \fi

```

We need an empty box to prevent the fil glue from disappearing. 我们需要一个空盒子来防止空白粘连 (fil glue) 消失。

```

709     \null\vfil

```

在这里，我们使用 \secdef 来指示使用哪些命令来创建实际的标题 (actual heading)。

```

710     \secdef\@part\@spart}
711 </report | book>

```

\@part 此宏执行部分的标题 (title of the part) 的实际格式 (actual formatting)。同样，article 文档类的宏定义与 report 和 book 文档类的宏定义不同。

当 article 文档类的 *secnumdepth* 大于 -1 时，我们有一个已编号的部分 (numbered part)，否则它就是未编号的 (unnumbered)。


```

712 <*article>
713 \def\@part[#1]#2{%
714     \ifnum \c@secnumdepth >\m@ne
715         \refstepcounter{part}%
716         \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
717     \else
718         \addcontentsline{toc}{part}{#1}%
719     \fi

```

我们打印 article 类中左对齐的标题 (title flush left)。此外，我们还防止断行和重新设置字体。

```

720     {\parindent \z@ \raggedright
721     \interlinepenalty \@M
722     \normalfont

```

当这是一个已编号的部分 (numbered part) 时，我们必须打印编号 (number)number 和标题 (title)。此处的 \nobreak 应防止分页。

```

723     \ifnum \c@secnumdepth >\m@ne
724         \Large\bfseries \partname\nobreakspace\thepart
725         \par\nobreak
726     \fi
727     \huge \bfseries #2%

```

现在我们清空标记寄存器 (mark registers)，留下一些空白 (white space)，并让 \@afterheading 负责抑制缩进。

```

728     \markboth{}{}\par}%
729     \nobreak
730     \vskip 3ex
731     \@afterheading}
732 </article>

```

对于 report 和 book 文档类，当 secnumdepth 大于 -2，我们有一个已编号的部分 (numbered part)，否则就没有编号 (unnumbered)。

```

733 <*report|book>
734 \def\@part[#1]#2{%
735     \ifnum \c@secnumdepth >-2\relax
736         \refstepcounter{part}%
737         \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
738     \else
739         \addcontentsline{toc}{part}{#1}%
740     \fi

```

我们清空标记寄存器 (mark registers), 并将标题放在 report 和 book 文档类的页面中央。此外, 我们还防止断行和重新设置字体。

```
741 \markboth{}{}%
742 {\centering
743 \interlinepenalty \@M
744 \normalfont
```

当这是一个已编号的部分 (numbered part) 时, 我们必须打印编号 (number)。

```
745 \ifnum \c@secnumdepth >-2\relax
746 \huge\bfseries \partname\nobreakspace\thepart
747 \par
```

在打印标题之前, 我们留出一些空间 (space), 并将收尾工作 (finishing) 留给 \endpart。

```
748 \vskip 20\p@
749 \fi
750 \Huge \bfseries #2\par}%
751 \@endpart}
752 </report | book>
```

\@spart 当使用用户命令的星形形式 (star form) 时, 此宏执行部分标题 (title of the part) 的实际格式 (actual formatting)。在这种情况下, 我们从不打印编号 (number)。否则格式相同。

该宏在 article 文档类中的定义与在 report 和 book 文档类中的定义差异与它们在 \@part 中定义差异相同。

```
753 < *article>
754 \def\@spart#1{%
755 {\parindent \z@ \raggedright
756 \interlinepenalty \@M
757 \normalfont
758 \huge \bfseries #1\par}%
759 \nobreak
760 \vskip 3ex
761 \@afterheading}
762 </article>
763 < *report | book>
764 \def\@spart#1{%
765 {\centering
766 \interlinepenalty \@M
```

```

767      \normalfont
768      \Huge \bfseries #1\par}%
769      \@endpart}
770 \</report | book>

```

\@endpart 此宏完成了 \@part 和 \@spart 的部分页 (part page)。

首先我们填充当前页面 (current page)。

```

771 \<*report | book>
772 \def\@endpart{\vfil\newpage

```

然后, 当我们处于双面模式 (twosided mode) 并且章 (chapters) 应该位于右侧时, 我们会生成一个完全空白的页面 (completely blank page)。

```

773      \if@twoside
774      \if@openright
775      \null
776      \thispagestyle{empty}%
777      \newpage
778      \fi
779      \fi

```

当这是一个两栏文档 (two-column document) 时, 我们必须切换回两栏模式 (two-column mode)。

```

780      \if@tempswa
781      \twocolumn
782      \fi}
783 \</report | book>

```

7.2.6 章

\chapter 章 (chapter) 应该总是从新页面开始, 因此我们首先调用 \clearpage 并将此页面的页面样式 (pagestyle) 设置为 *plain*。

```

784 \<*report | book>
785 \newcommand\chapter{\if@openright\cleardoublepage\else\clearpage\fi
786      \thispagestyle{plain}}%

```

然后我们防止浮动 (floats) 出现在页面顶部, 因为在章标题上方有浮动对象 (floating object) 看起来很奇怪。

```

787      \global\@topnum\z@

```

然后, 我们通过将 \@afterindent 开关设置为 false 来抑制第一段的缩进。我们使用 \secdef 指定用于实际设置章标题 (chapter title) 的宏。

```

788 \afterindentfalse
789 \secdef\@chapter\@schapter}

```

\@chapter 当我们有一个已编号的章 (numbered chapter) 时, 就会调用这个宏。当 `secnumdepth` 大于 `-1` 时, 并且在 `book` 类中, `\@mainmatter` 为 `true`, 我们显示章编号 (chapter number)。我们还通过向终端 (terminal) 写入消息来通知用户即将排版新章。

```

790 \def\@chapter[#1]#2{\ifnum \c@secnumdepth >\m@ne
791 \book \if@mainmatter
792 \refstepcounter{chapter}%
793 \typeout{\@chapapp\space\thechapter.}%
794 \addcontentsline{toc}{chapter}%
795 {\protect\numberline{\thechapter}#1}%
796 \book}
797 \else
798 \addcontentsline{toc}{chapter}{#1}%
799 \fi
800 \book}
801 \else
802 \addcontentsline{toc}{chapter}{#1}%
803 \fi

```

在将条目 (entry) 写入目录 (table of contents) 后, 我们将本章 (可选) 标题存储为 `\chaptermark`, 并在图 (figures) 和表 (tables) 列表中添加一些空白。

```

804 \chaptermark{#1}%
805 \addtocontents{lof}{\protect\addvspace{10\p@}}%
806 \addtocontents{lot}{\protect\addvspace{10\p@}}%

```

然后我们调用 `\@makechapterhead` 来格式化实际的章标题 (actual chapter title)。当我们处于两栏模式 (two-column mode) 时, 为了让章标题使用整个 `\textwidth` 宽度, 我们必须以一种特殊的方式进行此操作。在单栏模式 (one-column mode) 中, 我们调用 `\@afterheading` 来抑制缩进

```

807 \if@twocolumn
808 \topnewpage[\@makechapterhead{#2}]%
809 \else
810 \@makechapterhead{#2}%
811 \@afterheading
812 \fi}

```

\@makechapterhead 上面的宏使用 `\@makechapterhead<text>` 来格式化章标题 (heading of the chapter)。

我们首先留一些空白 (white space)。我们打开了一个组 (group)，其中段落缩进为 0pt，文本设置为右侧不对齐 (ragged)。我们还重置了字体。

```
813 \def\@makechapterhead#1{%
814   \vspace*{50\p}%
815   {\parindent \z@ \raggedright \normalfont
```

然后我们检查是否必须打印章编号。如果是这样，我们在章编号 (chapter number) 和标题之间留一些空白 (whitespace)。

```
816   \ifnum \c@secnumdepth >\m@ne
817   \if@mainmatter
818     \huge\bfseries \@chapapp\space \thechapter
819     \par\nobreak
820     \vskip 20\p@
821   \book \fi
822   \fi
```

现在我们将标题设置为大号粗体 (large bold) 字体。我们防止分页符 (page-break) 出现在标题的中间或后面。最后，在文本开始之前，我们留下一些空白。

```
823   \interlinepenalty\@M
824   \Huge \bfseries #1\par\nobreak
825   \vskip 40\p@
826 }}
```

\@schapter 当我们有一个未编号的章 (unnumbered chapter) 时，就会调用此宏。它比 **\@chapter** 简单得多，因为它只需要排版章标题 (chapter title)。

```
827 \def\@schapter#1{\if@twocolumn
828   \topnewpage[\@makeschapterhead{#1}]%
829   \else
830     \@makeschapterhead{#1}%
831     \@afterheading
832   \fi}
```

\@makeschapterhead 上面的宏使用 **\@makeschapterhead<text>** 格式化章标题。它与 **\@makechapterhead** 类似，只是它不必打印章编号 (chapter number)。

```
833 \def\@makeschapterhead#1{%
834   \vspace*{50\p}%
835   {\parindent \z@ \raggedright
836     \normalfont
837     \interlinepenalty\@M
```

```

838     \Huge \bfseries #1\par\nobreak
839     \vskip 40\p@
840   }}
841 \</report | book>

```

7.2.7 下级标题

这些命令都使用 \@startsection。

\section 这给出了一个普通的标题 (normal heading), 标题上下都有空白 (space), 标题设置为 \Large\bfseries, 第一段没有缩进。

```

842 \newcommand\section{\@startsection {section}{1}{\z@}%
843                                     {-3.5ex \@plus -1ex \@minus -.2ex}%
844                                     {2.3ex \@plus .2ex}%
845                                     {\normalfont\Large\bfseries}}

```

\subsection 这给出了一个普通的标题 (normal heading), 标题上下都有空白 (space), 标题设置为 \large\bfseries, 第一段没有缩进。

```

846 \newcommand\subsection{\@startsection{subsection}{2}{\z@}%
847                                     {-3.25ex\@plus -1ex \@minus -.2ex}%
848                                     {1.5ex \@plus .2ex}%
849                                     {\normalfont\large\bfseries}}

```

\subsubsection 这将给出一个普通的标题 (normal heading), 标题上方和下方都有空白 (space), 标题设置为 \normalsize\bfseries, 第一段没有缩进。

```

850 \newcommand\subsubsection{\@startsection{subsubsection}{3}{\z@}%
851                                     {-3.25ex\@plus -1ex \@minus -.2ex}%
852                                     {1.5ex \@plus .2ex}%
853                                     {\normalfont\normalsize\bfseries}}

```

\paragraph 这将在标题上方和右侧提供一个带有空白的栏外标题 (run-in heading), 标题设置为 \normalsize\bfseries。

```

854 \newcommand\paragraph{\@startsection{paragraph}{4}{\z@}%
855                                     {3.25ex \@plus 1ex \@minus .2ex}%
856                                     {-1em}%
857                                     {\normalfont\normalsize\bfseries}}

```

\subparagraph 这将在标题上方和右侧显示缩进的带有空白的栏外标题 (run-in heading), 标题设置为 \normalsize\bfseries。

```

858 \newcommand\subparagraph{\@startsection{subparagraph}{5}{\parindent}%
859                                     {3.25ex \@plus1ex \@minus .2ex}%
860                                     {-1em}%
861                                     {\normalfont\normalsize\bfseries}}

```

7.3 列表

7.3.1 普通的列表参数

以下命令用于设置列表环境 (list environment) 参数的默认值。有关参数含义的解释, 请参阅 L^AT_EX 手册。列表环境的默认设置如下。首先, 将 `\rightmargin`、`\listparindent` 和 `\itemindent` 设置为 0pt。然后, 对于第 K 级列表 (Kth level list), 调用命令 `\@listK`, 其中 “K” 表示 “i”、“ii”、...、“vi”。(即, 第三级列表调用 `\@listiii`。) 按照惯例, `\@listK` 应将 `\leftmargin` 设置为 `\leftmarginK`。

```

\leftmargin 当我们处于两栏模式 (two-column mode) 时, 有些页边距设置得稍微小一些。
\leftmarginii 862 \if@twocolumn
\leftmarginiii 863   \setlength\leftmargini {2em}
\leftmarginiiii 864 \else
\leftmarginiv 865   \setlength\leftmargini {2.5em}
\leftmarginv 866 \fi
\leftmarginvi 在这些文件中的所有参数设置合理化之前, 我们需要在此外部级别 (outer
\leftmarginvii level) 设置 \leftmargin 的值。

```

```
867 \leftmargin \leftmargini
```

计算以下三个值, 使其大于 `\labelsep` 和默认标签 (default labels) 宽度之和
默认标签即 “(m)”、“vii.” 和 “M.”。

```

868 \setlength\leftmarginiii {2.2em}
869 \setlength\leftmarginiiii {1.87em}
870 \setlength\leftmarginiv {1.7em}
871 \if@twocolumn
872   \setlength\leftmarginv {.5em}
873   \setlength\leftmarginvi {.5em}
874 \else
875   \setlength\leftmarginv {1em}
876   \setlength\leftmarginvi {1em}
877 \fi

```

`\labelsep` `\labelsep` 是标签 (label) 和项目文本 (text of an item) 之间的距离;
`\labelwidth` `\labelwidth` 是标签的宽度。

```
878 \setlength \labelsep {.5em}
879 \setlength \labelwidth{\leftmargini}
880 \addtolength\labelwidth{-\labelsep}
```

`\partopsep` 当用户在环境前留下空行 (blank line) 时, 除了 `\parskip` 和 `\topsep` 之外, 还会插入一个额外的垂直空间 `\partopsep`。

```
881 </article | report | book>
882 <10pt>\setlength\partopsep{2\p@ \@plus 1\p@ \@minus 1\p@}
883 <11pt>\setlength\partopsep{3\p@ \@plus 1\p@ \@minus 1\p@}
884 <12pt>\setlength\partopsep{3\p@ \@plus 2\p@ \@minus 2\p@}
```

`\@beginparpenalty` 这些惩罚 (penalties) 在列表 (list) 或段落 (paragraph) 环境前后插入。它们被
`\@endparpenalty` 设置为奖金值 (bonus value), 以鼓励在这些点上分页。

`\@itempenalty` 在列表项 (list items) 之间插入此惩罚 (penalty)。

```
885 <*article | report | book>
886 \@beginparpenalty -\@lowpenalty
887 \@endparpenalty -\@lowpenalty
888 \@itempenalty -\@lowpenalty
889 </article | report | book>
```

`\@listi` `\@listi` 为显示在顶层 (top-level) 的列表定义了 `\leftmargin`、`\parsep`、
`\@listI` `\topsep`、`\itemsep` 等的值。它的定义由字体尺寸命令 (font-size commands) 修改, 例如, 在 `\small` 中, 列表参数的值 “更小 (smaller)”。

由于这个原因, `listI` 被定义为保存 `listi` 的一个副本, 以便 `\normalsize` 可以将所有参数切换回来。

```
890 <*10pt | 11pt | 12pt>
891 \def\@listi{\leftmargin\leftmargini
892 <*10pt>
893         \parsep 4\p@ \@plus2\p@ \@minus\p@
894         \topsep 8\p@ \@plus2\p@ \@minus4\p@
895         \itemsep4\p@ \@plus2\p@ \@minus\p@}
896 </10pt>
897 <*11pt>
898         \parsep 4.5\p@ \@plus2\p@ \@minus\p@
899         \topsep 9\p@ \@plus3\p@ \@minus5\p@
900         \itemsep4.5\p@ \@plus2\p@ \@minus\p@}
```



```

901 </11pt>
902 <*12pt>
903         \parsep 5\p@ \@plus2.5\p@ \@minus\p@
904         \topsep 10\p@ \@plus4\p@ \@minus6\p@
905         \itemsep5\p@ \@plus2.5\p@ \@minus\p@}
906 </12pt>
907 \let\@listI\@listi

```

我们对参数进行了初始化 (initialise), 尽管严格来说这是不必要的。

```

908 \@listi

```

`\@listii` 下面是用于更高层级列表的相同宏。请注意, 它们没有保存的版本 (saved versions), 也不会被字体尺寸命令 (font-size commands) 修改。换句话说, 此类假定嵌套列表 (nested lists) 仅以 `\normalsize` 显示, 即主文档尺寸 (document size)。

```

\@listvi 909 \def\@listii {\leftmargin\leftmarginii
910         \labelwidth\leftmarginii
911         \advance\labelwidth-\labelsep
912 <*10pt>
913         \topsep 4\p@ \@plus2\p@ \@minus\p@
914         \parsep 2\p@ \@plus\p@ \@minus\p@
915 </10pt>
916 <*11pt>
917         \topsep 4.5\p@ \@plus2\p@ \@minus\p@
918         \parsep 2\p@ \@plus\p@ \@minus\p@
919 </11pt>
920 <*12pt>
921         \topsep 5\p@ \@plus2.5\p@ \@minus\p@
922         \parsep 2.5\p@ \@plus\p@ \@minus\p@
923 </12pt>
924         \itemsep \parsep}
925 \def\@listiii{\leftmargin\leftmarginiii
926         \labelwidth\leftmarginiii
927         \advance\labelwidth-\labelsep
928 <10pt>         \topsep 2\p@ \@plus\p@ \@minus\p@
929 <11pt>         \topsep 2\p@ \@plus\p@ \@minus\p@
930 <12pt>         \topsep 2.5\p@ \@plus\p@ \@minus\p@
931         \parsep \z@
932         \partopsep \p@ \@plus\z@ \@minus\p@
933         \itemsep \topsep}
934 \def\@listiv {\leftmargin\leftmarginiv

```

```

935             \labelwidth\leftmarginiv
936             \advance\labelwidth-\labelsep}
937 \def\@listv {\leftmargin\leftmarginv
938             \labelwidth\leftmarginv
939             \advance\labelwidth-\labelsep}
940 \def\@listvi {\leftmargin\leftmarginvi
941             \labelwidth\leftmarginvi
942             \advance\labelwidth-\labelsep}
943 </10pt | 11pt | 12pt>

```

7.3.2 排序列表

排序列表环境 (enumerate environment) 使用四个计数器 (counters): *enumi*、*enumii*、*enumiii* 和 *enumiv*, 其中 *enumN* 控制第 N 级排序 (enumeration) 的编号 (numbering)。

`\theenumi` 计数器已定义为 L^AT_EX 格式, 但此处更改了它们的表示形式 (representation)。

```

\theenumii 944 < *article | report | book >
\theenumiii 945 \renewcommand\theenumi{\@arabic\c@enumi}
\theenumiv 946 \renewcommand\theenumii{\@alph\c@enumii}
          947 \renewcommand\theenumiii{\@roman\c@enumiii}
          948 \renewcommand\theenumiv{\@Alph\c@enumiv}

```

`\labelenumi` 每个项目 (item) 的标签 (label) 由 `\labelenumi ... \labelenumiv` 命令生成。

```

\labelenumii 949 \newcommand\labelenumi{\theenumi.}
\labelenumiii 950 \newcommand\labelenumii{(\theenumii)}
\labelenumiv 951 \newcommand\labelenumiii{\theenumiii.}
          952 \newcommand\labelenumiv{\theenumiv.}

```

`\p@enumii` 在引用排序列表的第 N 层级的项目时, `\p@enumN\theenumN` 的展开 (expansion) 定义 `\ref` 命令的输出。

```

\p@enumiv 953 \renewcommand\p@enumii{\theenumi}
          954 \renewcommand\p@enumiii{\theenumi(\theenumii)}
          955 \renewcommand\p@enumiv{\p@enumiii\theenumiii}

```

7.3.3 常规列表

`\labelitemi` 项目 (itemization) 由四个命令控制: `\labelitemi`、`\labelitemii`、`\labelitemiii` 和 `\labelitemiv`, 它们定义了各种项目级别的标签: 使用的符号有黑点 (bullet)

```

\labelitemiii
\labelitemiv

```

let)、粗体短划线 (bold en-dash)、居中星号 (centered asterisk) 和居中的点 (centred dot)。

```
956 \newcommand\labelitemi {\labelitemfont \textbullet}
957 \newcommand\labelitemii {\labelitemfont \bfseries \textendash}
958 \newcommand\labelitemiii{\labelitemfont \textasteriskcentered}
959 \newcommand\labelitemiv {\labelitemfont \textperiodcentered}
```

`\labelitemfont` `\labelitemfont` 的默认定义 (default definition) 是将字体重置为 `\normalfont`, 以便不管周围的条件如何, 始终生成相同的符号 (same symbol)。

可能的替代方案 (alternatives) 是, 例如:

```
\renewcommand\labelitemfont
{\normalfont\fontfamily{lmss}\selectfont}
\renewcommand\labelitemfont
{\rmfamily\normalshape}
```

第一种方法将使用 Latin Modern Sans 的符号, 第二种方法将只允许更改字体序列 (font series), 以便在粗体上下文中的 `itemize` 产生更粗体的符号 (bolder symbols)。

```
960 \newcommand\labelitemfont{\normalfont}
```

7.3.4 解说列表

`description` (*env.*) 这里定义了解说列表环境 (description environment) – 而 `itemize` 和 `enumerate` 环境是以 L^AT_EX 格式定义的。

```
961 \newenvironment{description}
962           {\list{}\labelwidth\z@ \itemindent-\leftmargin
963           \let\makelabel\descriptionlabel}}
964           {\endlist}
```

`\descriptionlabel` 要更改标签 (label) 的格式, 必须重新定义 `\descriptionlabel`。

```
965 \newcommand*\descriptionlabel[1]{\hspace\labelsep
966                               \normalfont\bfseries #1}
```

7.4 定义新环境

7.4.1 摘要

`abstract` (*env.*) 当我们制作单独的标题页 (separate titlepage) 时, 我们也会将摘要 (abstract) 放在它自己的页面上。摘要在页面上垂直居中放置。

请注意，此环境不是为 books 定义的。

```
967 % \changes{v1.3m}{1995/10/23}{Added setting of \cs{beginparpenalty} to
968 %    discourage page break before abstract heading.}
969 \if@article\report
970 \if@titlepage
971   \newenvironment{abstract}{%
972     \titlepage
973     \null\vfil
974     \@beginparpenalty\@lowpenalty
975     \begin{center}%
976       \bfseries \abstractname
977       \@endparpenalty\@M
978     \end{center}}%
979   {\par\vfil\null\endtitlepage}
```

当我们没有创建单独的标题页–article 文档类的默认标题页，我们必须检查是否处于两栏模式 (two-column mode)。在这种情况下，摘要将作为一个\section*，否则将使用引用环境 (quotation environment) 对摘要进行排版。

```
980 \else
981   \newenvironment{abstract}{%
982     \if@twocolumn
983       \section*{\abstractname}%
984     \else
985       \small
986       \begin{center}%
987         {\bfseries \abstractname\vspace{-.5em}\vspace{\z@}}%
988       \end{center}%
989       \quotation
990     \fi}
991   {\if@twocolumn\else\endquotation\fi}
992 \fi
993 \if@article\report
```

7.4.2 诗歌

verse (*env.*) 诗歌环境 (verse environment) 是通过巧妙地使用列表环境 (list environment) 的参数来定义的。用户键入 \\ 结束一行。这是通过 \let \\ 等于 \@centercr 来实现的。

```
994 \newenvironment{verse}
```

```

995          {\let\\\@centercr
996          \list{}\{\itemsep      \z@
997              \itemindent    -1.5em%
998              \listparindent\itemindent
999              \rightmargin \leftmargin
1000              \advance\leftmargin 1.5em}%
1001          \item\relax}
1002      {\endlist}

```

7.4.3 引用 (quotation)

`quotation` (*env.*) 引用环境 (quotation environment) 也可以通过巧妙地使用列表环境 (list environment) 的参数来定义。环境中的线条设置小于 `\textwidth`。此环境中段落的第一行是缩进的。

```

1003 \newenvironment{quotation}
1004     {\list{}\{\listparindent 1.5em%
1005         \itemindent    \listparindent
1006         \rightmargin \leftmargin
1007         \parsep        \z@ \@plus\p@}%
1008     \item\relax}
1009     {\endlist}

```

7.4.4 引用 (quote)

`quote` (*env.*) 引用 (quote) 环境与引用 (quotation) 环境类似，只是段落没有缩进。

```

1010 \newenvironment{quote}
1011     {\list{}\{\rightmargin\leftmargin}%
1012     \item\relax}
1013     {\endlist}

```

7.4.5 定理

这个文档类 (document class) 没有定义自己的定理环境 (theorem environments)，可以使用 L^AT_EX 格式提供的默认值。

7.4.6 标题页

`titlepage` (*env.*) 在普通环境 (normal environments) 中，titlepage 环境只会开始和结束页面，并禁止页码 (page numbers)。当 L^AT_EX 处于两栏模式 (two-column mode) 时，

环境会暂时切换到单栏模式 (one-column mode)。在 report 类中, 它还将页码重置为 1, 然后在两栏模式中, 在末尾将其重置为 1。对于 book 类, 环境通过发出 `\cleardoublepage` 命令来确保标题页位于右侧页 (recto page) 上。在兼容模式 (compatibility mode) 中, 它将页码设置为零。这是不正确的, 因为它的结果是使用右侧页面 (right-hand page) 的页面参数 (page parameters), 但它是以前的方式。

首先, 我们给出了兼容模式 (compatibility mode) 的定义。

```

1014 \if@compatibility
1015   \newenvironment{titlepage}
1016     {%
1017 <book>       \cleardoublepage
1018             \if@twocolumn
1019               \@restonecoltrue\onecolumn
1020             \else
1021               \@restonecolfalse\newpage
1022             \fi
1023             \thispagestyle{empty}%
1024             \setcounter{page}\z@
1025           }%
1026   {\if@restonecol\twocolumn \else \newpage \fi
1027   }

```

这是原始的 (native) L^AT_EX 2_ε。

```

1028 \else
1029   \newenvironment{titlepage}
1030     {%
1031 <book>       \cleardoublepage
1032             \if@twocolumn
1033               \@restonecoltrue\onecolumn
1034             \else
1035               \@restonecolfalse\newpage
1036             \fi
1037             \thispagestyle{empty}%
1038             \setcounter{page}\@ne
1039           }%
1040   {\if@restonecol\twocolumn \else \newpage \fi

```

如果我们不是在双面模式 (two-side mode) 中, 标题页 (title page) 后的第一页也应该是第 1 页。

```

1041   \if@twoside\else

```

```

1042         \setcounter{page}\@ne
1043     \fi
1044 }
1045 \fi

```

7.4.7 附录

`\appendix` `\appendix` 命令并不是真正的环境，它是一个宏，它对事情的处理方式做了一些改变。

在 `article` 文档类中，`\appendix` 命令必须执行以下操作：

- 将节 (section) 和子节 (subsection) 计数器重置为零，
- 重新定义 `\thesection` 以产生按字母顺序排列的附录编号 (alphabetic appendix numbers)。这种重新定义是全局进行的，以确保即使在诸如 `multicols` 之类的环境中发出 `\appendix`，它也能够继续存在。

```

1046 ⟨*article⟩
1047 \newcommand\appendix{\par
1048   \setcounter{section}{0}%
1049   \setcounter{subsection}{0}%
1050   \gdef\thesection{\@Alph\c@section}}
1051 ⟨/article⟩

```

在 `report` 和 `book` 文档类中，`\appendix` 命令必须执行以下操作：

- 将章 (chapter) 和节 (section) 计数器重置为零
- 将 `\@chapapp` 设置为 `\appendixname`(用于消息)
- 重新定义章计数器 (chapter counter) 以生成附录编号 (appendix numbers)
- 如果附录名称 (appendix titles) 和附录标题 (appendix headings) 看起来与章名称 (chapter titles) 和章标题 (chapter headings) 不同，可能需要重新定义 `\chapter` 命令。这种重新定义是全局进行的，以确保即使在诸如 `multicols` 之类的环境中发出 `\appendix`，它也能继续存在。

```

1052 ⟨*report | book⟩
1053 \newcommand\appendix{\par
1054   \setcounter{chapter}{0}%
1055   \setcounter{section}{0}%

```

```

1056 \gdef\@chapapp{\appendixname}%
1057 \gdef\thechapter{\@Alph\c@chapter}}
1058 \</report | book>

```

7.5 为现有环境设置参数

7.5.1 数组 (array) 和表格 (tabular)

`\arraycolsep` 数组环境 (array environment) 中的列 (columns) 由 2 个 `\arraycolsep` 分隔。

```
1059 \setlength\arraycolsep{5\p@}
```

`\tabcolsep` 表格环境 (tabular environment) 中的列 (columns) 由 2 个 `\tabcolsep` 分隔。

```
1060 \setlength\tabcolsep{6\p@}
```

`\arrayrulewidth` 数组 (array) 和表格 (tabular) 环境中线段 (rule) 的宽度由 `\arrayrulewidth` 给出。

```
1061 \setlength\arrayrulewidth{.4\p@}
```

`\doublerulesep` 数组 (array) 和表格 (tabular) 环境中相邻线段 (rule) 之间的间距由 `\doublerulesep` 给出。

```
1062 \setlength\doublerulesep{2\p@}
```

7.5.2 表格 (tabbing)

`\tabbingsep` 这将控制 `\'` 命令放入的空间 (space)。(有关说明, 请参阅 L^AT_EX 手册。)

```
1063 \setlength\tabbingsep{\labelsep}
```

7.5.3 小页

`\@minipagerestore` 当 `\@minipagerestore` 宏进入到一个小页环境 (minipage environmen) 时, 它被调用来设置在小页环境中需要以不同方式处理的事情。在当前的类中, 它什么也不做。

`\@mpfootins` 小页 (minipages) 有自己的脚注 (footnotes); `\skip\@mpfootins` 在迷你页面中对脚注的作用与对普通脚注 (ordinary footnotes) 的相同。

```
1064 \skip\@mpfootins = \skip\footins
```


7.5.4 有框的盒子

`\fboxsep` `\fbox` 和 `\framebox` 在框和其中的文本之间留下空白 (space)。

`\fboxrule` 由 `\fbox` 和 `\framebox` 创建的盒子 (box) 中线段 (rules) 的宽度。

```
1065 \setlength\fboxsep{3\p@}
1066 \setlength\fboxrule{.4\p@}
```

7.5.5 排序单行公式 (equation) 和排序公式组 (eqnarray)

`\theequation` 在章内时, 排序单行公式 (equation) 计数器将在新章开始时重置, 排序单行公式 (equation) 编号将以章编号为前缀。

此代码必须遵循 `\chapter` 的定义, 或者更确切地说, 遵循章计数器 (chapter counter) 的定义。

```
1067 <article>\renewcommand \theequation {\@arabic\c@equation}
1068 <*report | book>
1069 \@addtoreset {equation}{chapter}
1070 \renewcommand\theequation
1071   {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@equation}
1072 </report | book>
```

`\jot` `\jot` 是排序公式组 (eqnarray) 环境行之间添加的额外空间 (extra space)。使用默认值。

```
1073 % \setlength\jot{3pt}
```

`\@eqnnum` `\@eqnnum` 宏定义排序单行公式 (equation) 编号在公式中的显示方式。再次使用默认值。

```
1074 % \def\@eqnnum{(\theequation)}
```

7.6 浮动对象

L^AT_EX 格式仅定义了一些可用于定义浮动对象 (floating objects) 的工具。这是在文档类 (document class) 中完成的。它需要为 TYPE 类型 (例如 TYPE=figure) 的每个浮动对象定义以下宏。

`\fps@TYPE` TYPE 类型的浮动的默认位置说明符 (default placement specifier)。

`\ftype@TYPE` TYPE 类型的浮动的类型编号 (type number)。每个 TYPE 都关联了一个唯一的正 TYPE 编号, 它是 2 的幂。例如, 图形 (figures) 可能有

类型编号 1, 表格 (tables) 可能有类型编号 2, 程序 (programs) 可能有类型编号 4, 等等。

`\ext@TYPE` 指示存储浮动类型 `TYPE` 内容列表 (contents list) 的文件的扩展名。例如 `\ext@figure='lof'`。

`\fnum@TYPE` 为标题 (caption) 生成图形编号 (figure number) 的宏。例如 `\fnum@TYPE == 'Figure \thefigure'`。

`\@makecaption<num><text>` 用于制作标题 (caption) 的宏, `<num>` 为标题文本 `\fnum@...` and `<text>` 生成的值。它可以假定它位于适当宽度的 `\parbox` 中。这将用于所有浮动对象。

实现浮动对象 (如图形) 的实际环境是使用 `\@float` 和 `\end@float` 宏定义的, 而这些宏又是在 `LATEX` 格式中定义好的。

实现单栏浮动对象 (single-column floating object) 的环境以 `TYPE` 类型的 `\@float{TYPE}[\<placement>]` 开始, `TYPE` 类型带有 `<placement>` 作为位置说明符 (placement specifier)。`<PLACEMENT>` 的默认值由 `\fps@TYPE` 定义。

环境结束于 `\end@float`, 例如, `\figure == \@floatfigure, \endfigure == \end@float`。

7.6.1 图形

下面是图形环境 (figure environment) 的实现。

`\c@figure` 首先, 我们必须分配一个计数器 (counter) 来对图形 (figures) 进行编号。

在 report 和 book 文档类中, 章 (chapters) 内的图形按章编号。

```
1075 <*article>
1076 \newcounter{figure}
1077 \renewcommand \thefigure {\@arabic\c@figure}
1078 </article>
1079 <*report | book>
1080 \newcounter{figure}[chapter]
1081 \renewcommand \thefigure
1082     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@figure}
1083 </report | book>
```

`\fps@figure` 以下是 “figure(图)” 类型的浮动对象的参数。

```
\ftype@figure 1084 \def\fps@figure{tbp}
\ext@figure 1085 \def\ftype@figure{1}
\num@figure
```

```

1086 \def\ext@figure{lof}
1087 \def\fnun@figure{\figurename\nobreakspace\thefigure}

```

`figure (env.)` 以及实际环境 (actual environment) 的定义。带 * 的表单 (form) 用于双栏图形 (double-column figures)。

```

1088 \newenvironment{figure}
1089         {\@float{figure}}
1090         {\end@float}
1091 \newenvironment{figure*}
1092         {\@dblfloat{figure}}
1093         {\end@dblfloat}

```

7.6.2 表格

下面是表环境 (table environment) 的实现。它与图形环境 (figure environment) 非常相似。

`\c@table` 首先，我们必须分配一个计数器来对表进行编号。

在 report 和 book 文档类中，章内的表格按章编号。

```

1094 \<article>
1095 \newcounter{table}
1096 \renewcommand\thetable{\@arabic\c@table}
1097 \</article>
1098 \<report | book>
1099 \newcounter{table}[chapter]
1100 \renewcommand \thetable
1101     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@table}
1102 \</report | book>

```

`\fps@table` 以下是“table(表格)”类型的浮动对象的参数。

```

\ftype@table 1103 \def\fps@table{tbp}
\ext@table 1104 \def\ftype@table{2}
\num@table 1105 \def\ext@table{lot}
1106 \def\fnun@table{\tablename\nobreakspace\thetable}

```

`table (env.)` 以及实际环境的定义。带 * 的表格用于双栏表格 (double-column tables)。

```

table* (env.) 1107 \newenvironment{table}
1108         {\@float{table}}
1109         {\end@float}
1110 \newenvironment{table*}

```

```

1111          {\@dblfloat{table}}
1112          {\end@dblfloat}

```

7.6.3 标题

`\@makecaption` `\caption` 命令调用 `\@makecaption` 来格式化浮动对象的标题 (caption)。它有两个参数, 第一个 $\langle number \rangle$, 为浮动对象的编号 (number), 第二个 $\langle text \rangle$, 为标题的文本 (text of the caption)。通常 $\langle number \rangle$ 包含一个字符串 (string), 如“图 3.2”。宏可以假定它是在一个宽度为 `\normalsize` 的 `\parbox` 中调用的。

`\abovecaptionskip` 这些长度包含标题上方和下方要保留的空白 (space)。

```

\belowcaptionskip 1113 \newlength\abovecaptionskip
1114 \newlength\belowcaptionskip
1115 \setlength\abovecaptionskip{10\p@}
1116 \setlength\belowcaptionskip{0\p@}

```

此宏的定义是 `\long`, 能在标题 (caption) 中允许多个段落 (paragraph)。

```

1117 \long\def\@makecaption#1#2{%
1118   \vskip\abovecaptionskip

```

我们想看看标题 (caption) 是否适合页面上的一行, 因此我们首先在一个临时盒子 (temporary box) 中排版。

```

1119   \sbox\@tempboxa{#1: #2}%

```

我们可以测量它的宽度。它比当前的 `\hsize` 大, 我们将标题排版为普通段落 (ordinary paragraph)。

```

1120   \ifdim \wd\@tempboxa >\hsize
1121     #1: #2\par

```

如果标题合适, 我们将其居中。因为这在垂直模式 (vertical mode) 中直接使用了一个 `\hbox`, 所以它不会执行 `\everypar` 令牌 (tokens); 这里唯一需要做的就是重置“小页标志 (minipage flag)”, 所以我们显式地这么做。

```

1122   \else
1123     \global \@minipagefalse
1124     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
1125   \fi
1126   \vskip\belowcaptionskip}

```

7.7 字体更改

这里我们提供了 L^AT_EX 2.09 及更早版本中常见的声明性字体更改命令 (declarative font changing commands)。这些命令在文本模式 (text mode) 和数学模式 (math mode) 中工作。它们是为了兼容性而提供的, 但使用时应以 `\text...` 和 `\math...` 命令开始。这些命令使用 `\DeclareTextFontCommand` 定义, 该命令有三个参数: 要定义的用户命令 (user command)、在文本模式中要执行 L^AT_EX 命令、在数学模式中要执行 L^AT_EX 命令。

`\rm` 修改族 (family) 的命令。在兼容模式下, 我们首先选择 “default” 字体, 以获得 L^AT_EX 2.09 行为 (behaviour)。

```
\sf 1127 \DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
      1128 \DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
      1129 \DeclareOldFontCommand{\tt}{\normalfont\ttfamily}{\mathtt}
```

`\bf` 更改为粗体序列 (bold series) 的命令。应该使用 `\mdseries` 显式切换回中等序列 (medium series)。

```
1130 \DeclareOldFontCommand{\bf}{\normalfont\bfseries}{\mathbf}
```

`\sl` 以及更改字体形状 (shape) 的命令。默认情况下, 倾斜 (slanted) 和小型大写 (`\it` (small caps) 形状不可用作数学字母 (math alphabets), 因此这些更改在数学模式 (math mode) 中不起作用。但是, 我们确实警告用户, 这样的选择不会产生任何影响。应该使用 `\upshape` 明确地更改回直立形状 (upright shape)。

```
1131 \DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
1132 \DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
1133 \DeclareOldFontCommand{\sc}{\normalfont\scshape}{\@nomath\sc}
```

`\cal` 命令 `\cal` 和 `\mit` 只能在数学模式中使用, 在数学模式之外, 它们没有任何作用。目前, 新字体选择方案 (New Font Selection Scheme, NFSS) 定义了这些命令来生成警告消息。因此, 我们必须 “手工 (by hand)” 定义它们。

```
1134 \DeclareRobustCommand*\cal{\@fontswitch\relax\mathcal}
1135 \DeclareRobustCommand*\mit{\@fontswitch\relax\mathnormal}
```

8 交叉引用

8.1 目录，等

`\section` 命令将 `\contentsline{section}{<title>}{<page>}` 命令写入 .toc 文件中, 在 .toc 文件中, <title> 包含条目 (entry) 的内容, 而 <page> 是页码 (page number)。如果正在对节进行编号, 则 <title> 格式为 `\numberline{<num>}{<heading>}`, 这里的 <num> 是由 `\thesection` 生成的编号 (number)。其他分节命令 (sectioning commands) 的工作方式类似。

“figure(图形)”环境中的 `\caption` 命令写入

`\contentsline{figure}{\numberline{<num>}{<caption>}}{<page>}`

在 .lof 文件中, 其中的 <num> 是 `\thefigure` 生成的编号 (number), 而 <caption> 是图形标题 (figure caption)。它在 “table(表格)” 环境中的工作方式类似。

命令 `\contentsline{<name>}` 展开为 `\l@<name>`。因此, 要指定目录 (table of contents), 我们必须定义 `\l@chapter`、`\l@section`、`\l@subsection`、... ; 要指定图表目录 (list of figures), 我们必须定义 `\l@figure`; 等等。其中大多数可以用 `\@dottedtocline` 命令来定义, 其工作原理如下:

`\@dottedtocline{<level>}{<indent>}{<numwidth>}{<title>}{<page>}`

<level> 只有当 $\langle level \rangle \leq \text{tocdepth}$ 计数器的值时才生成条目。注意, `\chapter` 是 0 级, `\section` 是 1 级, 等等。

<indent> 从内容行开始处的左外边距 (outer left margin) 的缩进。

<numwidth> 节编号 (section number) 所在盒子的宽度, 如果 <title> 包括一个 `\numberline` 命令。

`\@pnumwidth` 此命令使用以下三个参数, 这三个参数是用 `\newcommand` 命令设置的 (因此 `\@tocrmarg` 可以使用 em 使它们依赖于字体)。

`\@dotsep` `\@pnumwidth` 放置页码的盒子的宽度。

`\@tocrmarg` 多行条目 (multiple line entries) 的右边距 (right margin)。一个条目需要 $\@tocrmarg \geq \@pnumwidth$ 。

`\@dotsep` 点 (dots) 与点之间的距离, 单位是 。应该定义为像 2 或 1.7 这样的数字

```

1136 \newcommand\@pnumwidth{1.55em}
1137 \newcommand\@tocrmarg{2.55em}
1138 \newcommand\@dotsep{4.5}
1139 \langle article \rangle \setcounter{tocdepth}{3}
1140 \langle !article \rangle \setcounter{tocdepth}{2}

```

8.1.1 目录

`\tableofcontents` 此宏用于请求 L^AT_EX 生成目录。在 report 和 book 文档类中，目录、图形等始终以单栏模式 (single-column mode) 设置。

```

1141 \newcommand\tableofcontents{%
1142 \langle *report | book \rangle
1143     \if@twocolumn
1144         \@restonecoltrue\onecolumn
1145     \else
1146         \@restonecolfalse
1147     \fi

```

使用 `\chapter*` 命令设置标题 (title)，确保栏外标题 (running head)–如果需要–包含正确的信息。

```

1148     \chapter*{\contentsname
1149 \rangle /report | book \rangle
1150 \langle article \rangle \section*{\contentsname

```

`\@mkboth` 的代码放在标题 (heading) 内，以避免标题后的垂直间距 (vertical spacing) 受到任何影响 (在某些情况下)。对于其他命令，例如下面的 `\listoffigures`，它与 L^AT_EX 2.09 版本相比有所改变，因为如果在双栏模式 (two-column mode) 中使用，它将产生严重的错误 (请参见 pr/3285)。然而，在这些类中，`\tableofcontents` 总是以单栏模式 (one-column mode) 排版，因此出于兼容性的原因，保留了一些不一致的设置 (inconsistent setting)。

```

1151         \@mkboth{%
1152             \MakeUppercase\contentsname}{\MakeUppercase\contentsname}}%

```

实际的目录是通过调用 `\@starttoc{toc}` 生成的。之后，如有必要，我们将恢复双栏模式 (two-column mode)。

```

1153     \@starttoc{toc}%
1154 \langle !article \rangle \if@restonecol\twocolumn\fi
1155 }

```

`\l@part` 如上所述, 每个分节命令 (sectioning command) 都需要一个附加宏 (additional macro) 来格式化其在目录中的条目 (entry)。部分条目 (entry for parts) 的宏是以特殊方式定义的。

首先, 我们确保如果应该出现分页符 (pagebreak), 它会出现在此条目之前。此外, 还添加了一些空白 (whitespace), 并且一个组 (group) 开始保持局部的更改 (keep changes local)。

```
1156 \newcommand*\l@part[2]{%
1157   \ifnum \c@tocdepth >-2\relax
1158   <article>   \addpenalty\@secpenalty
1159   <!article>  \addpenalty{-\@highpenalty}%
1160   \addvspace{2.25em \@plus\p}%
```

宏 `\numberline` 要求保存部分编号 (part number) 的盒子的宽度存储在 \LaTeX 的暂存寄存器 (scratch register) `\@tempdima` 中。因此, 即使我们没有在内部使用 `\numberline`, 我们也会在那里初始化它, 因为使用的值非常大, 所以像 `\numberline{VIII}` 这样的值仍然可以工作

```
1161   \setlength\@tempdima{3em}%
1162   \begingroup
```

我们将 `\parindent` 设置为 `0pt`, 并使用 `\rightskip` 为页码 (page numbers) 留出足够的空间 (room)。³ 为防止溢框信息 (overfull box messages), `\parfillskip` 设置为负值。

```
1163   \parindent \z@ \rightskip \@pnumwidth
1164   \parfillskip -\@pnumwidth
```

现在我们可以用大粗体字体 (large bold font) 设置条目。我们确保离开垂直模式 (vertical mode), 设置部分标题 (part title) 并添加页码 (page number), 设置右侧对齐 (flush right)。

```
1165   {\leavevmode
1166     \large \bfseries #1\hfil
1167     \hb@xt@\@pnumwidth{\hss #2%

1168     \kern-\p@\kern\p@}}\par
```

在这个条目之后立即防止分页符 (pagebreak), 但是使用 `\everypar` 重置 `\if@nobreak` 开关。最后, 我们关闭组 (close the group)。

³我们应该将 `\rightskip` 设置为 `\@tocrmarg` 而不是 `\@pnumwidth` (\LaTeX 的任何版本都没有这样做), 否则 `\rightskip` 会太小。不幸的是, 这在 \LaTeX 2_ϵ 中无法改变, 因为我们不想创建不同版本的 \LaTeX 2_ϵ , 除非这是绝对必要的, 否则会产生不同的排版输出 (typeset output); 相反, 我们为了 \LaTeX 3 而暂停它。


```

1169      \nobreak
1170 <article>      \if@compatibility
1171      \global\@nobreaktrue
1172      \everypar{\global\@nobreakfalse\everypar{}}%
1173 <article>      \fi
1174      \endgroup
1175      \fi}

```

`\l@chapter` 此宏设置章目录中的条目的格式。它与 `\l@part` 非常相似。

首先，我们确保如果出现分页符 (pagebreak)，它会出现该条目之前。此外，还添加了一点空白 (whitespace)，一个组 (group) 开始保持局部的更改 (keep changes local)。

```

1176 <*report | book>
1177 \newcommand*\l@chapter[2]{%
1178   \ifnum \c@tocdepth >\m@ne
1179     \addpenalty{-\@highpenalty}%
1180     \vskip 1.0em \@plus\p@

```

宏 `\numberline` 要求保存部分编号 (part number) 的盒子的宽度存储在 \LaTeX 的暂存寄存器 (scratch register) `\@tempdima` 中。因此，即使我们没有在内部使用 `\numberline`，我们也会在那里对其进行初始化 (位置和值似乎有问题，但在不产生兼容性问题的情况下无法更改)。我们开始一个组 (group)，并更改一些段落参数 (paragraph parameters) (另请参阅 `\l@part` 关于 `\rightskip`)。

```

1181   \setlength\@tempdima{1.5em}%
1182   \begingroup
1183     \parindent \z@ \rightskip \@pnumwidth
1184     \parfillskip -\@pnumwidth

```

然后我们离开垂直模式 (vertical mode) 并切换到粗体字体 (bold font)。

```

1185     \leavevmode \bfseries

```

因为这里不使用 `\numberline`，所以在设置条目之前，我们需要“手工”进行一些微调。我们不鼓励但不禁止在章条目 (chapter entry) 后立即使用分页符 (pagebreak)。

```

1186     \advance\leftskip\@tempdima
1187     \hskip -\leftskip
1188     #1\nobreak\hfil
1189     \nobreak\hb@xt@\@pnumwidth{\hss #2%
1190

```

```

1191     \penalty\@highpenalty
1192   \endgroup
1193   \fi}
1194 </report | book>

```

`\l@section` 在 article 文档类中，节 (sections) 目录中的条目看起来很像 report 和 book 文档类的章条目 (chapter entries)。

首先，我们确保如果出现分页符 (pagebreak)，它会出现在该条目之前。此外，还添加了一点空白 (whitespace)，一个组 (group) 开始保持局部的更改 (keep changes local)。

```

1195 <*article>
1196 \newcommand*\l@section[2]{%
1197   \ifnum \c@tocdepth >\z@
1198     \addpenalty\@secpenalty
1199     \addvspace{1.0em \@plus\p@}%

```

宏 `\numberline` 要求保存部分编号 (part number) 的盒子的宽度存储在 L^AT_EX 的暂存寄存器 (scratch register) `\@tempdima` 中。因此，我们把它放在那里。我们开始一个组 (group)，并更改一些段落参数 (另请参阅 `\l@part` 关于 `\rightskip`)。

```

1200   \setlength\@tempdima{1.5em}%
1201   \begingroup
1202     \parindent \z@ \rightskip \@pnumwidth
1203     \parfillskip -\@pnumwidth

```

然后我们离开垂直模式 (vertical mode) 并切换到粗体字体 (bold font)。

```

1204     \leavevmode \bfseries

```

因为这里不使用 `\numberline`，所以在设置条目之前，我们需要“手工”进行一些微调。我们不鼓励但不禁止在章条目 (chapter entry) 后立即使用分页符 (pagebreak)。

```

1205     \advance\leftskip\@tempdima
1206     \hskip -\leftskip
1207     #1\nobreak\hfil
1208     \nobreak\hb@xt@\@pnumwidth{\hss #2%

1209                                     \kern-\p@\kern\p@}\par
1210   \endgroup
1211   \fi}
1212 </article>

```

在 report 和 book 文档类中 \l@section 的定义要简单得多。

```
1213 <*report | book>
1214 \newcommand*\l@section{\@dottedtocline{1}{1.5em}{2.3em}}
1215 </report | book>
```

\l@subsection 所有较低层级的条目都是使用宏 \@dottedtocline 来定义的 (请参见上文)。

```
\l@subsubsection 1216 <*article>
\l@paragraph 1217 \newcommand*\l@subsection{\@dottedtocline{2}{1.5em}{2.3em}}
\l@subparagraph 1218 \newcommand*\l@subsubsection{\@dottedtocline{3}{3.8em}{3.2em}}
1219 \newcommand*\l@paragraph{\@dottedtocline{4}{7.0em}{4.1em}}
1220 \newcommand*\l@subparagraph{\@dottedtocline{5}{10em}{5em}}
1221 </article>
1222 <*report | book>
1223 \newcommand*\l@subsection{\@dottedtocline{2}{3.8em}{3.2em}}
1224 \newcommand*\l@subsubsection{\@dottedtocline{3}{7.0em}{4.1em}}
1225 \newcommand*\l@paragraph{\@dottedtocline{4}{10em}{5em}}
1226 \newcommand*\l@subparagraph{\@dottedtocline{5}{12em}{6em}}
1227 </report | book>
```

8.1.2 图形列表

\listoffigures 此宏用于请求 L^AT_EX 生成图形列表 (list of figures)。它与 \tableofcontents 非常相似。

```
1228 \newcommand\listoffigures{%
1229 <*report | book>
1230   \if@twocolumn
1231     \@restonecoltrue\onecolumn
1232   \else
1233     \@restonecolfalse
1234   \fi
1235   \chapter*{\listfigurename}%
1236 </report | book>
1237 <article>   \section*{\listfigurename}%
1238             \@mkboth{\MakeUppercase\listfigurename}%
1239                   {\MakeUppercase\listfigurename}%
1240   \@starttoc{lof}%
1241 <report | book>   \if@restonecol\twocolumn\fi
1242 }
```

\l@figure 此宏在图形列表 (list of figures) 中生成一个条目 (entry)。

```
1243 \newcommand*\l@figure{\@dottedtocline{1}{1.5em}{2.3em}}
```

8.1.3 表格列表

`\listoftables` 此宏用于请求 L^AT_EX 生成表格列表 (list of tables)。它与 `\tableofcontents` 非常相似。

```
1244 \newcommand\listoftables{%
1245   <*report | book>
1246   \if@twocolumn
1247     \@restonecoltrue\onecolumn
1248   \else
1249     \@restonecolfalse
1250   \fi
1251   \chapter*{\listtablename}%
1252 </report | book>
1253 <article> \section*{\listtablename}%
1254   \@mkboth{%
1255     \MakeUppercase\listtablename}%
1256     {\MakeUppercase\listtablename}%
1257   \@starttoc{lot}%
1258 <report | book> \if@restonecol\twocolumn\fi
1259 }
```

`\l@table` 此宏在表格列表 (list of tables) 中生成一个条目 (entry)。

```
1260 \let\l@table\l@figure
```

8.2 参考文献

`\bibindent` “开放 (open)” 参考文献格式 (bibliography format) 使用 `\bibindent` 缩进。

```
1261 \newdimen\bibindent
1262 \setlength\bibindent{1.5em}
```

`thebibliography (env.)` “theobjectory” 环境执行以下命令：

```
\renewcommand{\newblock}{\hskip.11em \@plus.33em \@minus.07em}
```

— 定义 “封闭 (closed)” 格式，其中条目的块 (blocks)(主要信息单元) 一起运行。

`\sloppy` — 使用它是因为在参考文献中很难做断行 (line breaks) 处理，

`\sfcode\.=1000\relax` — 使 “.” (句号) 不产生句末空格。

此环境的实现 (implementation) 基于通用列表环境 (generic list environment)。它在内部使用 `enumiv` 计数器生成列表的标签

当发现一个空的 “thebibliography” 环境时，会发出警告。

```
1263 \newenvironment{thebibliography}[1]
1264 (*article)
1265     {\section*{\refname}%
```

由于至少在 report 和 book (双栏选项) 中有 `\chapter` 的定义，否则会将其吞没，所以 `\mkboth` 被移出了标题参数 (heading argument)。

```
1266     \mkboth{\MakeUppercase\refname}{\MakeUppercase\refname}%
1267 \end{article}
1268 (*!article)
1269     {\chapter*{\bibname}%
1270     \mkboth{\MakeUppercase\bibname}{\MakeUppercase\bibname}%
1271 \end{article}
1272     \list{\@biblabel{\@arabic\c@enumiv}}%
1273         {\settowidth\labelwidth{\@biblabel{#1}}%
1274         \leftmargin\labelwidth
1275         \advance\leftmargin\labelsep
1276         \@openbib@code
1277         \usecounter{enumiv}%
1278         \let\p@enumiv\@empty
1279         \renewcommand\theenumiv{\@arabic\c@enumiv}}%
1280     \sloppy
```

这是为整个环境设置 `\clubpenalty` 的正常 (非无限) 值，因此我们还必须重置其已存储的值 (stored value)。(为什么在下面第二个 4000 之后有 %?)

```
1281     \clubpenalty4000
1282     \@clubpenalty \clubpenalty
1283     \widowpenalty4000%
1284     \sfcode`.\@m}
1285     {\def\@noitemerr
1286     {\@latex@warning{Empty `thebibliography' environment}}}%
1287     \endlist}
```

`\newblock` `\newblock` 的默认定义是生成一个小空间 (small space)。

```
1288 \newcommand\newblock{\hskip .11em\@plus.33em\@minus.07em}
```

`\@openbib@code` `\@openbib@code` 的默认定义就是什么都不做。它将由 openbib 选项更改。

```
1289 \let\@openbib@code\@empty
```

`\@biblabel` `\bibitem[...]` 命令的标签由该宏生成。使用 L^AT_EX 格式默认值。

```
1290 % \renewcommand*{\@biblabel}[1]{[#1]\hfill}
```

`\@cite` `\cite` 命令的输出由该宏生成。使用 L^AT_EX 格式默认值。

```
1291 % \renewcommand*{\@cite}[1]{[#1]}
```

8.3 索引

`theindex (env.)` “theindex” 环境可用于索引 (index)。它创建的索引包含两栏，每个条目都是一个单独的段落。在用户级别，命令 `\item`、`\subitem` 和 `\subsubitem` 用于生成不同层级的索引条目 (index entries)。当遇到字母表 (alphabet) 中的新字母时，可以添加一定数量的 `\indexspace` 空白 (white space)。

```
1292 \newenvironment{theindex}
1293     {\if@twocolumn
1294       \@restonecolfalse
1295     \else
1296       \@restonecoltrue
1297     \fi
1298 <article>          \twocolumn[\section*{\indexname}]{%
1299 <!article>         \twocolumn[\@makeschapterhead{\indexname}]{%
1300                   \@mkboth{\MakeUppercase\indexname}{%
1301                     {\MakeUppercase\indexname}}%
1302                   \thispagestyle{plain}\parindent\z@
```

必须在 `\twocolumn` 操作之后对 `\columnseprule` 和 `\columnsep` 进行参数更改 (parameter changes)。否则，它们可能会影响索引前的最后一页。

```
1303       \parskip\z@ \@plus .3\p@\relax
1304       \columnseprule \z@
1305       \columnsep 35\p@
1306       \let\item\@idxitem}
```

当文档在索引后继续，并且它是一个单栏文档 (one-column document) 时，我们必须切换回索引之后的一栏。

```
1307       {\if@restonecol\onecolumn\else\clearpage\fi}
```

`\@idxitem` 这些宏用于格式化索引中的条目。

```
\subitem 1308 \newcommand\@idxitem{\par\hangindent 40\p@}
```

```
\subsubitem 1309 \newcommand\subitem{\@idxitem \hspace*{20\p@}}
```

```
1310 \newcommand\subsubitem{\@idxitem \hspace*{30\p@}}
```

`\indexspace` 在索引中的“字母块 (letter blocks)”之间插入的空白 (white space)。

```
1311 \newcommand\indexspace{\par \vskip 10\p@ \@plus5\p@ \@minus3\p@\relax}
```

8.4 脚注

`\footnoterule` 通常，脚注 (footnotes) 与正文 (main body of the text) 之间用一条小线段 (rule) 分开。这条线段是由 `\footnoterule` 宏画出来的。我们必须确保该线段不占用垂直空间 (vertical space)(请参阅 `plain.tex`)，因此我们通过添加适量的垂直间距 (vertical skip) 来补偿 0.4pt 线段的自然高度 (natural height)。

为了防止线段 (rule) 与脚注 (footnote) 冲突，我们首先添加了一点负的垂直间距 (negative vertical skip)，然后我们放置线段，并确保我们在开始此操作的同一点结束。

```
1312 \renewcommand\footnoterule{%
1313   \kern-3\p@
1314   \hrule\@width.4\columnwidth
1315   \kern2.6\p@}
```

`\c@footnote` 脚注在报告 (report) 和书籍 (book) 文档类的章中编号。

```
1316 \!article\@addtoreset{footnote}{chapter}
```

`\@makefnmark` L^AT_EX 的脚注机制 (footnote mechanism) 调用宏 `\@makefnmark` 来生成实际的脚注。宏获取脚注的文本作为其参数，并应使用 `\@thefnmark` 作为脚注的标记 (mark)。当 `\@makefnmark` 宏有效地位于宽度为 `\columnwidth` (即使用 `\hsize = \columnwidth`) 的 `\parbox` 内时，将调用该宏。

下面的 T_EX 代码给出了一个可以实现的示例。

```
\newcommand\@makefnmark[1]{%
  \setpar{\@par
    \@tempdima = \hsize
    \advance\@tempdima-10pt
    \parshape \@ne 10pt \@tempdima}%
  \par
  \parindent 1em\noindent
  \hbox to \z@{\hss\@makefnmark\#1}}
```

这个定义的效果是脚注的所有行缩进 10pt，而新段落的第一行缩进 1em。要更改这些尺寸，只需将“10pt” (在两个位置) 或“1em”替换为所需的值 (desired value)。这个记号 (mark) 刚好与脚注对齐。

在这些文档类中，我们使用了一个更简单的宏，其中脚注文本 (footnote text) 设置为普通文本段落 (ordinary text paragraph)，除了段落的第一行和脚注的第一行外，没有缩进。因此，宏必须做的所有事情都是为后续段落设置 `\parindent` 为适当的值，并在标记前放置适当的缩进 (indentation)。

```
1317 \newcommand\@makefnmark[1]{%  
1318     \parindent 1em%  
1319     \noindent  
1320     \hb@xt@1.8em{\hss\@makefnmark}#1}
```

`\@makefnmark` 打印在文本中指向脚注的脚注标记 (footnote markers) 应由 `\@makefnmark` 宏生成。我们使用它的默认定义。

```
1321 %\renewcommand\@makefnmark{\hbox{\@textsuperscript  
1322 %                                     {\normalfont\@thefnmark}}}
```


9 初始化

9.1 单词

此文档类用于以英语编写的文档。为了准备另一种语言的版本，各种各样的英语单词 (English words) 必须被替换。所有需要替换的英语单词在下面的命令名中定义。这些命令可以在用于非英语的 LATEX 定制类或包中重新定义。

```
\contentsname
\listfigurename 1323 \newcommand\contentsname{Contents}
\listtablename 1324 \newcommand\listfigurename{List of Figures}
                  1325 \newcommand\listtablename{List of Tables}

\refname
\bibname 1326 <article>\newcommand\refname{References}
\indexname 1327 <report | book>\newcommand\bibname{Bibliography}
                  1328 \newcommand\indexname{Index}

\figurename
\tablename 1329 \newcommand\figurename{Figure}
                  1330 \newcommand\tablename{Table}

\partname
\chaptername 1331 \newcommand\partname{Part}
\appendixname 1332 <report | book>\newcommand\chaptername{Chapter}
\abstractname 1333 \newcommand\appendixname{Appendix}
                  1334 <!book>\newcommand\abstractname{Abstract}
```

9.2 日期

`\today` 该宏使用 T_EX 原语 `\month`、`\day` 和 `\year` 来提供 L^AT_EX 运行的日期。

在 `\begin{document}` 中此定义将得到优化，从而不会存储所有“错误 (wrong)”月份的名称。这种优化不会在这里完成，因为这将“冻结 (freeze)”在任何特殊用途的格式 (special purpose format) 中的 `\today`，这种特殊用途的格式是通过将类文件 (class file) 加载到格式文件 (format file) 中而生成的格式。

```
1335 \def\today{\ifcase\month\or
1336   January\or February\or March\or April\or May\or June\or
```

```

1337 July\or August\or September\or October\or November\or December\fi
1338 \space\number\day, \number\year}

```

9.3 两栏模式

`\columnsep` 这给出了两栏模式 (two-column mode) 中两栏之间的距离。

```

1339 \setlength\columnsep{10\p@}

```

`\columnseprule` 这给出了两栏模式 (two-column mode) 中两栏之间的线段 (rule) 的宽度。我们没有可见的线段 (visible rule)。

```

1340 \setlength\columnseprule{0\p@}

```

9.4 页面样式

除非用户另有指定，否则文档类 `article` 和 `report` 中都有 *plain* 页面。在“book”文档类中，我们默认使用 *headings* 页面样式。并使用阿拉伯数字页码 (arabic page numbers)。

```

1341 <!book>\pagestyle{plain}
1342 <book>\pagestyle{headings}
1343 \pagenumbering{arabic}

```

9.5 单面或双面打印

当未指定 `twoside` 选项时，我们不会试图使每个页面与其他页面一样长。

```

1344 \if@twoside
1345 \else
1346 \raggedbottom
1347 \fi

```

当指定了 `twocolumn` 选项时，我们调用 `\twocolumn` 来激活此模式。我们试图使每一栏一样长，只需调用 `sloppy` 即可。

```

1348 \if@twocolumn
1349 \twocolumn
1350 \sloppy
1351 \flushbottom

```

通常我们调用 `\onecolumn` 来启动单栏排版。

```

1352 \else
1353 \onecolumn
1354 \fi

```

Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

Symbols	<code>\@evenfoot</code> 474 , 476 , 535	<code>\@makeschapterhead</code>
<code>\@Roman</code> 651	<code>\@evenhead</code> 474 , 477 , 536	828 , 830 , 833 , 1299
<code>\@afterheading</code>	<code>\@fnsymbol</code> 583	<code>\@maketitle</code> ... 590 ,
. 731 , 761 , 811 , 831	<code>\@fontswitch</code> 1134 , 1135	592 , 597 , 604 , 614
<code>\@afterindentfalse</code>	<code>\@fpbot</code> 443	<code>\@medpenalty</code> 227
..... 692 , 788	<code>\@fpsep</code> 443	<code>\@minipagefalse</code> .. 1123
<code>\@author</code> 544 ,	<code>\@fptop</code> 443	<code>\@minipagerestore</code> 1064
560 , 572 , 606 , 625	<code>\@highpenalty</code> .. 227 ,	<code>\@mparswitchfalse</code> .. 41
<code>\@beginparpenalty</code> .	1159 , 1179 , 1191	<code>\@mparswitchtrue</code> ... 43
..... 885 , 974	<code>\@idxitem</code> ... 1306 , 1308	<code>\@mpfootins</code> 1064
<code>\@biblabel</code> 1272 , 1273 , 1290	<code>\@itempenalty</code> 885	<code>\@nobreakfalse</code> ... 1172
<code>\@chapapp</code> . 499 , 528 ,	<code>\@latex@warning</code> .. 1286	<code>\@nobreaktrue</code> 1171
661 , 793 , 818 , 1056	<code>\@listI</code> 108 , 890	<code>\@noitemerr</code> 1285
<code>\@chapter</code> 789 , 790	<code>\@listi</code> 108 ,	<code>\@normalsize</code> 87
<code>\@cite</code> 1291	119 , 129 , 139 ,	<code>\@oddfont</code> 474 , 476 , 512 , 535
<code>\@clubpenalty</code> 1282	152 , 162 , 172 , 890	<code>\@oddfont</code> 474 , 476 , 512 , 535
<code>\@date</code> 545 ,	<code>\@listii</code> 909	<code>\@oddfont</code> 474 , 478 , 513 , 537
563 , 573 , 607 , 628	<code>\@listiii</code> 909	<code>\@openbib@code</code>
<code>\@dblfloat</code> .. 1092 , 1111	<code>\@listiv</code> 909	... 66 , 1276 , 1289
<code>\@dblfpbot</code> 458	<code>\@listv</code> 909	<code>\@openrightfalse</code> ... 56
<code>\@dblpsep</code> 458	<code>\@listvi</code> 909	<code>\@openrighttrue</code> . 53 , 55
<code>\@dblftop</code> 458	<code>\@lowpenalty</code> .. 227 ,	<code>\@part</code> 693 , 710 , 712
<code>\@dotsep</code> 1136	886 , 887 , 888 , 974	<code>\@pnumwidth</code> .. 1136 ,
<code>\@dottedtocline</code> 1214 ,	<code>\@mainmatterfalse</code> .	1163 , 1164 ,
1217 , 1218 , 669 , 685	1167 , 1183 ,
1219 , 1220 ,	<code>\@mainmattertrue</code> 8 , 677	1184 , 1189 ,
1223 , 1224 ,	<code>\@makecaption</code> 1113	1202 , 1203 , 1208
1225 , 1226 , 1243	<code>\@makechapterhead</code> .	<code>\@ptsize</code> 1 , 34 ,
<code>\@endparpenalty</code> 885 , 977 808 , 810 , 813	36 , 38 , 39 , 84 , 85
<code>\@endpart</code> . 751 , 769 , 771	<code>\@makefnmark</code> 584 , 1320 , 1321	<code>\@restonecolfalse</code> .
<code>\@eqnnum</code> 1074	<code>\@makefntext</code> 585 , 1317 1021 ,

1035, 1146, 1233, 1249, 1294	\@restonecoltrue ..	1019, 1033, 1144, 1231, 1247, 1296	\@roman 947	\@schapter ... 789, 827	\@secpenalty 1158, 1198	\@setfontsize	90, 96, 102, 115, 125, 135, 148, 158, 168, 181, 182, 183, 184, 185, 186, 187, 190, 191, 192, 193, 194, 195, 196, 199, 200, 201, 202, 203, 204	\@settopoint .. 286, 374, 375, 380, 395	\@spart ... 693, 710, 753	\@startsection . 842, 846, 850, 854, 858	\@starttoc	1153, 1240, 1257	\@textsuperscript .	584, 587, 1321	\@thanks	565, 571, 599, 605	\@thefnmark	584, 587, 1322	\@title 543, 555, 574, 608, 620	\@titlepagefalse . 5, 50	\@titlepagetrue .. 6, 48	\@tocrmarg 1136	\@topnewpage . 808, 828	\@topnum 596, 787	\@twocolumnfalse ... 59	\@twocolumntrue 61	\@twosidefalse 41	\@twosidetrue 43	\@undefined 110	\@width 1314	A	\abovecaptionskip .	1113, 1118	\abovedisplayshortskip	92, 98, 104, 117, 127, 137, 150, 160, 170	\abovedisplayskip .	91, 97, 103, 107, 116, 126, 136, 144, 149, 159, 169, 177	abstract (env.) 967	\abstractname	976, 983, 987, 1331	\addcontentsline ..	716, 718, 737, 739, 794, 798, 802	\addtocontents 805, 806	\and 578, 612	\appendix 1046	\appendixname 1056, 1331	\arraycolsep 1059	\arrayrulewidth .. 1061	\AtEndOfPackage 65	\author ... 543, 576, 610	B	\backmatter 679	\baselineskip	289, 290, 291, 292, 297, 299	\baselinestretch .. 210	\belowcaptionskip .	1113, 1126	\belowdisplayshortskip	93, 99, 105, 118, 128, 138, 151, 161, 171	\belowdisplayskip .	107, 144, 177	\bf 1130	\bibindent . 67, 68, 1261	\bibname 1269, 1270, 1326	\bigskipamount 222	\bottomfraction ... 408	\brokenpenalty 237	C	\c@bottomnumber ... 407	\c@chapter	641, 654, 1057, 1071, 1082, 1101	\c@dbltopnumber ... 412	\c@enumi 945	\c@enumii 946	\c@enumiii 947	\c@enumiv 948, 1272, 1279	\c@equation . 1067, 1071	\c@figure 1075	\c@footnote . 583, 1316	\c@paragraph . 641, 659	\c@part 641, 651	\c@secnumdepth	483, 489, 497, 505, 518, 526, 639, 714, 723, 735, 745, 790, 816	\c@section	641, 652, 655, 1050	\c@subparagraph 641, 660	\c@subsection . 641, 657	\c@subsubsection ..	641, 658	\c@table 1094	\c@tocdepth	1157, 1178, 1197	\c@topnumber 404
------------------------------	---------------------	------------------------------------	-------------------	-------------------------	-------------------------	---------------------	---	---	---------------------------	--	------------------	------------------	---------------------	----------------	----------------	--------------------	-------------------	----------------	---------------------------------------	--------------------------	--------------------------	-----------------------	-------------------------	-------------------------	-------------------------	-------------------------	-------------------------	------------------------	-----------------------	--------------------	----------	---------------------	------------	------------------------	---	---------------------	--	--------------------------	---------------------	---------------------	---------------------	-----------------------------------	-------------------------	---------------------	----------------------	--------------------------	------------------------	-------------------------	-------------------------	---------------------------	----------	-----------------------	---------------------	------------------------------	-------------------------	---------------------	------------	------------------------	---	---------------------	---------------	----------------	---------------------------	---------------------------	-------------------------	-------------------------	-------------------------	----------	-------------------------	------------------	----------------------------------	-------------------------	--------------------	---------------------	----------------------	---------------------------	--------------------------	----------------------	-------------------------	-------------------------	------------------------	---------------------	---	------------------	---------------------	--------------------------	--------------------------	---------------------	----------	---------------------	-------------------	------------------	------------------------

<code>\if@titlepage</code> 4 , 547 , 970	<code>\l@table</code> 1260	<code>\leftmarginv</code> 862 , 937 , 938
<code>\if@twocolumn</code> .. 214 , 255 , 271 , 302 , 347 , 588 , 703 , 807 , 827 , 862 , 871 , 982 , 991 , 1018 , 1032 , 1143 , 1230 , 1246 , 1293 , 1348	<code>\labelenumi</code> 949 <code>\labelenumii</code> 949 <code>\labelenumiii</code> 949 <code>\labelenumiv</code> 949 <code>\labelitemfont</code> . 956 , 957 , 958 , 959 , 960 <code>\labelitemi</code> 956 <code>\labelitemii</code> 956 <code>\labelitemiii</code> 956 <code>\labelitemiv</code> 956	<code>\leftmarginvi</code> 862 , 940 , 941 <code>\leftskip</code> 1186 , 1187 , 1205 , 1206 <code>\lineskip</code> . 207 , 558 , 623 <code>\listfigurename</code> 1235 , 1237 , 1238 , 1239 , 1323 <code>\listoffigures</code> ... 1228 <code>\listoftables</code> 1244 <code>\listparindent</code> 69 , 998 , 1004 , 1005 <code>\listtablename</code> 1251 , 1253 , 1255 , 1256 , 1323
<code>\if@twoside</code> 325 , 353 , 474 , 773 , 1041 , 1344	<code>\labelsep</code> . 878 , 911 , 927 , 936 , 939 , 942 , 965 , 1063 , 1275	
<code>\indexname</code> 1298 , 1299 , 1300 , 1301 , 1326	<code>\labelwidth</code> ... 878 , 910 , 911 , 926 , 927 , 935 , 936 , 938 , 939 , 941 , 942 , 962 , 1273 , 1274	
<code>\indexspace</code> 1311	<code>\LARGE</code> 180 , 555 , 620 <code>\Large</code> 180 , 724 , 845 <code>\large</code> 180 , 557 , 563 , 622 , 628 , 849 , 1166	
<code>\interlinepenalty</code> . .. 236 , 721 , 743 , 756 , 766 , 823 , 837	<code>\leftmargin</code> 67 , 119 , 129 , 139 , 152 , 162 , 172 , 862 , 891 , 909 , 925 , 934 , 937 , 940 , 962 , 999 , 1000 , 1006 , 1011 , 1274 , 1275	
<code>\intextsep</code> 416	<code>\leftmargini</code> 119 , 129 , 139 , 152 , 162 , 172 , 862 , 879 , 891	
<code>\it</code> 1131	<code>\leftmarginii</code> 862 , 909 , 910	
<code>\item</code> 1001 , 1008 , 1012 , 1306	<code>\leftmarginiii</code> 862 , 925 , 926	
<code>\itemindent</code> . 68 , 69 , 962 , 997 , 998 , 1005	<code>\leftmarginiv</code> 862 , 934 , 935	
<code>\itemsep</code> .. 122 , 132 , 142 , 155 , 165 , 175 , 895 , 900 , 905 , 924 , 933 , 996		
<code>\itshape</code> 1131		
J		
<code>\jot</code> 1073		
L		
<code>\l@chapter</code> 1176		
<code>\l@figure</code> ... 1243 , 1260		
<code>\l@paragraph</code> 1216		
<code>\l@part</code> 1156		
<code>\l@section</code> 1195		
<code>\l@subparagraph</code> .. 1216		
<code>\l@subsection</code> 1216		
<code>\l@subsubsection</code> . 1216		
	<code>\mainmatter</code> 671 <code>\makelabel</code> 963 <code>\MakeRobust</code> .. 110 , 111 <code>\maketitle</code> 547 <code>\MakeUppercase</code> . 482 , 496 , 504 , 517 , 525 , 1152 , 1238 , 1239 , 1255 , 1256 , 1266 , 1270 , 1300 , 1301 <code>\marginparpush</code> 302 <code>\marginparsep</code> 302 , 359 , 367 <code>\marginparwidth</code> ... 312 <code>\mathbf</code> 1130 <code>\mathcal</code> 1134 <code>\mathit</code> 1131 <code>\mathnormal</code> 1135 <code>\mathrm</code> 1127 <code>\mathsf</code> 1128 <code>\mathtt</code> 1129 <code>\maxdepth</code> 252	

<code>\medskipamount</code> 222	24 , 27 , 30 , 31 ,	<code>\section</code> 842 ,
<code>\mit</code> 1134	266 , 354 , 362 , 376	983 , 1150 , 1237 ,
	<code>\paragraph</code> 854	1253 , 1265 , 1298
N	<code>\paragraphmark</code> 633	<code>\sectionmark</code> .. 481 ,
<code>\newblock</code> 72 , 1288	<code>\parindent</code>	503 , 516 , 540 , 633
<code>\newif</code> 3 , 4 , 7 , 8	. 211 , 585 , 720 ,	<code>\sf</code> 1127
<code>\nobreakspace</code>	755 , 815 , 835 ,	<code>\sffamily</code> 1128
724 , 746 , 1087 , 1106	858 , 1163 , 1183 ,	<code>\sl</code> 1131
<code>\normalfont</code> ... 584 ,	1202 , 1302 , 1318	<code>\small</code> 113 , 549 , 985
587 , 722 , 744 ,	<code>\parsep</code> 70 , 121 , 122 ,	<code>\smallskipamount</code> .. 222
757 , 767 , 815 ,	131 , 132 , 141 ,	<code>\subitem</code> 1308
836 , 845 , 849 ,	142 , 154 , 155 ,	<code>\subparagraph</code> 858
853 , 857 , 861 ,	164 , 165 , 174 ,	<code>\subparagraphmark</code> . 633
960 , 966 , 1127 ,	175 , 893 , 898 ,	<code>\subsection</code> 846
1128 , 1129 ,	903 , 914 , 918 ,	<code>\subsectionmark</code> ...
1130 , 1131 ,	922 , 924 , 931 , 1007 487 , 541 , 633
1132 , 1133 , 1322	<code>\parskip</code> 211 , 1303	<code>\subsubitem</code> 1308
<code>\normallineskip</code> ... 207	<code>\part</code> 687	<code>\subsubsection</code> 850
<code>\normalsize</code>	<code>\partname</code> 724 , 746 , 1331	<code>\subsubsectionmark</code> 633
. 87 , 853 , 857 , 861	<code>\partopsep</code> ... 881 , 932	
<code>\num@figure</code> 1084	<code>\postdisplaypenalty</code> 233	T
<code>\num@table</code> 1103	<code>\predisplaypenalty</code> 233	<code>\tabbingsep</code> 1063
	<code>\ps@headings</code> 474	<code>\tabcolsep</code> 1060
O	<code>\ps@myheadings</code> 534	<code>table</code> (env.) 1107
<code>\oddsidemargin</code> 312		<code>table*</code> (env.) 1107
<code>\onecolumn</code> 704 ,	Q	<code>\tablename</code> .. 1106 , 1329
1019 , 1033 ,	<code>\quad</code> 484 , 490 , 519	<code>\tableofcontents</code> . 1141
1144 , 1231 ,	<code>\quotation</code> 989	<code>\textasteriskcentered</code>
1247 , 1307 , 1353	<code>quotation</code> (env.) .. 1003 958
<code>\overfullrule</code> ... 44 , 46	<code>quote</code> (env.) 1010	<code>\textbullet</code> 956
		<code>\textendash</code> 957
P		<code>\textfloatsep</code> 416
<code>\p@enumii</code> 953	R	<code>\textfraction</code> 410
<code>\p@enumiii</code> 953	<code>\refname</code> 1265 , 1266 , 1326	<code>\textheight</code> .. 288 , 392
<code>\p@enumiv</code> ... 953 , 1278	<code>\rm</code> 1127	<code>\textperiodcentered</code> 959
<code>\pagenumbering</code>		<code>\textwidth</code>
... 670 , 678 , 1343	S	. 254 , 355 , 363 , 378
<code>\paperheight</code> ... 11 ,	<code>\sbox</code> 1119	<code>\thanks</code> 551 , 569 , 602 , 619
14 , 17 , 20 , 23 ,	<code>\sc</code> 1131	<code>thebibliography</code>
26 , 29 , 30 , 294 , 388	<code>\scriptsize</code> 180	(env.) 1263
<code>\paperwidth</code>	<code>\scshape</code> 1133	
. 12 , 15 , 18 , 21 ,		

<code>\thechapter</code> ... 499,	716, 724, 737, 746	<code>\topsep</code> 120, 130, 140,
528, 651 , 793,	<code>\thesection</code> ... 484,	153, 163, 173,
795, 818, 1057,	506, 519, 651 , 1050	894, 899, 904,
1071, 1082, 1101	<code>\thesubparagraph</code> .. 651	913, 917, 921,
<code>\theenumi</code>	<code>\thesubsection</code> 490, 651	928, 929, 930, 933
. 944 , 949 , 953 , 954	<code>\thesubsubsection</code> . 651	<code>\topskip</code> .. 239 , 253 , 301
<code>\theenumii</code> 944 , 950 , 954	<code>\thetable</code>	<code>\tt</code>
<code>\theenumiii</code> 944 , 951 , 955	.. 1096, 1100, 1106	<code>\ttfamily</code>
<code>\theenumiv</code> 944 , 952 , 1279	<code>\thispagestyle</code> . 599,	<code>\twocolumn</code>
<code>\theequation</code> 1067 , 1074	702, 776, 786,	592, 781, 1026,
<code>\thefigure</code>	1023, 1037, 1302	1040, 1154,
.. 1077, 1081, 1087	<code>\tiny</code>	1241, 1258,
<code>\thefootnote</code> 583	<code>\title</code> 543 , 575, 609	1298, 1299, 1349
<code>theindex</code> (env.) ... 1292	<code>\titlepage</code>	V
<code>\thepage</code> 477,	<code>titlepage</code> (env.) .. 1014	<code>verse</code> (env.)
478 , 513 , 536 , 537	<code>\today</code> 546 , 1335	994
<code>\theparagraph</code> 651	<code>\topfraction</code> 406	W
<code>\thepart</code> 651 ,	<code>\topmargin</code> 382	<code>\widowpenalty</code> 231 , 1283