第二章　命令与环境

LaTeX处理程序所需要的文本必须用文本编辑器把它们输入到一个后缀为 .tex 的文件中，任何人都必须这样做。这个文件的内容完全是可以打印的字符，没有在屏幕不可见的字符或命令。然而，为了使得 LaTeX进行某种操作，就必须有一些命令。这些命令在输入文本中是可见的，在输出的结果中就不存在了。因此知道命令与要打印出来的文本的区别，以及命令的作用是相当重要的。

§2.1　命令的名称与参数

字符 # $ & ~_ ^ % { } 表示特殊的命令，它们的含义后面会讲到。如果要以文本的形式打印这些字符，在它们前面必须有 \ (反斜杠)。

大约有二十条命令是由 \ 后接另一个非字母字符的，即它共只由两个字符组成。在这种方式中，\$ 就表示暂时取消命令 $ 的含义，而以文本的方式打印出这个字符。

绝大多数的命令是由 \ 后接一个或多个字母组成的，第一个非字母字符表示命令名称的结束。许多命令可以有参数，或变量来扩展相应的功能。参数有可能是可以省略的，也就是说它们可以有，也可以没有；也有的参数是不能省略的，即至少必须给一个允许的值。这些命令的语法是

\命令名[可省参数]{不可省略的参数}

这里可省参数是放在中括号 [ ] 里，而不可省略参数放在大括号 { } 里。一条命令可以有几个可省略参数，它们就必须按指定顺序位于各自的中括号内。如果没有用可省略参数，方括号可以不写。在命令名称和参数之间可以有任意数目的空格。

对于可省略参数，有可能出现两个问题：如果没有使用任何可省略参数，而接下来的文本中第一个字符是 [，或者一个可选参数的文本中有] ，那么LaTeX就会误解参数。这可以通过使用大括号来遮住的中括号，如 {[} 和 {]} 。

有的命令有几个不可省略的参数。它们中的每一个都必须位于一对大括号内，而且顺序要同命令描述中的顺序一样。例如，

\rule[提升高度]{宽度}{高度}

就会生成一个大小为指定 宽度 和 高度 的黑矩形，并且在当前基线基础上向上提升一定的高度。一个宽 10mm ，高 3mm ，并且位于基线上的矩形，可以如此生成： \rule{10mm}{3mm} 。在这里并没有给出可省参数。参数必须按描述中的顺序出现，不可能交换。

有些命令有两种形式，一种是标准形式，而另一种就是所谓的 *- 形式。后者的标志是命令名称以 * 号结尾，这个 * 号位于成对的相应于可省参数和

不可省参数的 [ ] 和 { } 之前。标准形式与 *- 形式之间的区别，我们会在每条命令中加以解释。

命令名称是在第一个非字母字符之前结束的。如果命令后接可省参数或不可省参数，那么命令名称是在 [或 { 之前结束的，因为这两个字符就为非字母字符。然而有很多命令，其没有参数，只是由名称组成，如生成 LaTeX标志的命令 \LaTeX 。如果这样的命令后面接的是一个如逗号或句号的标点符号，那这显然表示命令的结束。如果它后接一个普通单词，那么命令名称与接下来单词间的空白当做命令结束符对待： \LaTeX logo 生成的结果是 LaTeXlogo ，也就是说，这里的空白只当做命令的结束符，而不当做两个单词的间隔。这是空白的一个特殊规则，在 2.5.1 中有介绍。

为了在只由名称组成的命令之后插入一个空格，那就需要在命令后使用一个空结构 {} 或空格命令（ \ 加空格 ）。因此生成 'The LaTeX logo' 的正确方法是输入 The \LaTeX{} logo 或 The \LaTeX\ logo 。如果不这样做，也可以把命令自身放在大括号内，如 The {\LaTeX} logo ，这样也会得到有空格的正确输出： 'The LaTeX logo' 。顺带说一下，用命令 \LaTeXe 可以得到 LaTeX 2_ε标志。

§2.2 环境

一个环境是用命令 \begin{ 环境 } 来初始化的，最后用 \end{ 环境 } 结束。对于环境的所允许取的值列在有关命令的章节中。

环境的作用是：位于其内的文本要根据环境参数进行不同的处理。有可能（暂时地）改变某一处理特征，如缩进，行宽，字样等等。这些改变只在该环境内有作用。例如，对于 quote 环境，

```
前面的文本
\begin{quote}
文本 1 \small 文本 2 \bfseries 文本 3
\end{quote}
后续文本
```

与前面和后续文本的左边和右边页边相比，环境中的页边界要增大。在这个例子中，对三部分文本： 文本1 、 文本2 和 文本3 的页边界要增大。在 文本1 后面是命令 \small ，这样就把后面文本变成小字样。在 文本2 后又有一命令 \bfseries ，它切换到黑体字样。而所有这些命令的作用到 \end{quote} 时也就结束了。

在 quote 环境中的三部分文本都相比于前面和后面的文本进行了缩进。 文本1 是以正常字样出现的，这同环境外是一样的。而 文本2 和 文本3 则是小号字样，而且 文本3**还是黑体** 。

当 quote 环境结束时，接下来的文本字样同以前的一样。

很多命令的名称与环境名称一样。在这种情况中，所用的命令名是没有前缀 \ 的。例如，命令 \em 切换成强调字样，通常用的是斜体，而相应的环境 \begin{em} 则把所有文本变为斜体，直到 \end{em} 为止。

一个没有名称的环境可以用一对大括号 {...} 来表示。其中所有命令的作用当遇到右大括号时也就结束了。

用户也可以创建另外的环境，见 7.4 节的描述。

§2.3 声明

声明 就是一种命令，它并不打印出任何文本，而是改变某一参数或命令的值或含义。声明一旦出现，其马上就起作用，直到遇到同一类型的另一个声明。然而，如果声明是位于一个环境或 {...} 中，那它的作用也就只能延续到相应的 \end 命令或右大括号 } 为止。在前一节中提到的命令 \em, \bfseries 和 \small 都是这样一种没有直接输出，而是改变当前字样的声明例子。

有一些声明与参数有关，如命令 \setlength 会给长度参数赋值（见 2.4 和 7.2 节）。

例：

`{\bfseries This text appears in bold face}` 这里的 \bfseries 声明改变了字样：**This text appears in bold face**. 这个声明的作用到遇到右大括号 } 时结束。

`\setlength{\parindent}{0.5cm}` 把段落中第一行的缩进设为 0.5cm 。当遇到下一个命令 \setlength{\parindent}，或者最近的一个结束当前环境的 \end 时，这个声明的作用就会终止。

`\pagenumbering{roman}` 用罗马数字来显示页码。

有些声明，如上面最后那个例子，它的作用是全局的，也就是说它并不局限于当前环境。下面的声明都具有这种性质，稍后给出它们的含义：

```
\newcounter     \pagenumbering      \newlength
\setcounter     \thispagestyle      \newsavebox
\addtocounter
```

用这些命令结出的声明，也是马上起作用，而且直到被同类型的一个新声明覆盖其含义为止。在上面最后那个例子中，只有当遇到 \pagenumbering{arabic} 这样的命令时，才不会以罗马数字显示页码。

§2.4 长度

§2.4.1 固定长度

长度是由前面可能有符号（ + 或 - ）的小数，后接一个必需的尺寸单位组成。下面是可允许的单位及缩写名称：

`cm` 厘米，
`mm` 毫米，
`in` 英寸（ 1in = 2.54cm ），
`pt` 点（ 1in = 72.27pt ），
`bp` 大点（ 1in = 72bp ），
`pc` pica（ 1pc = 12pt ），
`dd` didôt 点（ 1157dd = 1238pt ），
`cc` cicero（ 1cc = 12dd ），
`em` 与字体相关的尺寸，表示大写字母 M 的宽度，
`ex` 也是与字体相关的尺寸，表示字母 x 的高度。

在 TEX和 LATEX中的小数，既可以采用英国方式，也可以采取欧洲方式，即小数点可以句号或逗号，也就是说 `12.5cm` 和 `12,5cm` 都可以。

注意 `0` 不是一个合法的长度，因为其没有长度单位。要给出一个零长度，必须用 `0pt` 或 `0cm` 等等。

要想给长度参数赋值，可以用 LATEX命令 `\setlength` ，我们将在第 7.2 节中介绍它和所有其它处理长度的命令。它的语法是：

```
\setlength{\ 长度命令 }{ 已定义的长度 }
```

例如，一行的长度是由一个叫 `\textwidth` 的参数定义的，它通常根据样式和字体尺寸来取默认值。要想把行宽设为 12.5cm ，可以用：

```
\setlength{\textwidth}{12.5cm}
```

§2.4.2 橡皮长度

有些参数的值为橡皮长度，即这个长度可以伸展或收缩给定的量。橡皮长度的语法是：

```
正常值 plus 伸展值 minus 收缩值
```

这里的 `正常值` ， `伸展值` 和 `收缩值` 都是长度。例如：

```
\setlength{\parskip}{1ex plus0.5ex minus0.2ex}
```

其含义为：两段间的行距（称为 `\parskip` ）等于当前字体中 x 的高度，但是该行距可以伸长到 1.5 倍或收缩到 0.8 倍这个长度。

有一个特殊的橡皮长度是 `\fill` 。其正常长度是零，但可以伸展到任何长度。

§2.5 特殊字符

§2.5.1 空格与回车

空格或空白字符具有与通常字符不太一样的性质，我们在前面的 2.1 节中已提到了一些。在处理的过程中，空格是用橡皮长度（ 2.4.2 节）代替的，这样可以使得行恰好填满一行的宽度。因此，如果你不知道下面的规则，就可能出现一些古怪的现象：

- 一个空格同一千个空格是一样的，实际接受的只是第一个空格；
- 在一行开头的空格是被忽略不计的；
- 回车（开始新行）是当做空格对待的。

这些规则的一些结果就是可以在单词间或一行的开头插入随意多的空格（这样会使源文件更好看些），而且一个单词可以恰好位于一行的末尾，而它和后一个单词之间可以没有空格。为了强迫在一般情形中会消失的空格必须出现，那就要用命令 \␣（ \ 后接一个空格键，这里用符号 ␣ 表示）。

而有的时候又有必要避免由于开始一个新行而出现不必要的空格。在这种情形中，该行必须在回车键前插入注释符号 % 。

一行中有两个回车键得到一个空行，这也是开始一个段落的标志。而对于空白字符，一个与一千个是一样的。

§2.5.2 引号

在打字机上可以找到的引号 " 并不会用在书籍印刷中。取而代之的是在开始和结束用不同的符号，如 ‘单引号’ 和 “双引号” 。单引号是用 ‘ 和 ’ 生成的，而双引号是用两个相应的字符表示的： ‘‘ 表示 “， ’’ 表示” 。另外，打字机符号 " 会生成右双引号。

§2.5.3 连字符与破折号

在书籍印刷中，打字机上为 - 的符号有各种不同的长度： -, –, — 。其中最短的是连字符，用在诸如 father-in-law 这样的复合单词中，以及在一行结尾处的单词分割中；中间长短的是短破折号，用于数值范围，如第 33–36 页；而最长的那个是全破折号，用于复合句 — 通常也就称为破折号。可以用输入连字符一次、两次或三次来得到这三种不同长度的横线，即 - 结果为 - ， -- 结果为 – ， --- 结果为 — 。第四种类型的破折号是负号 – ，必须用 $-$ 这样的数学模式来输入（第 5 章）。

§2.5.4 得到命令字符

在 2.1 节中提到，字符 # $ ~ _ % { } 都被当做命令处理。如果要把它们做为文本来显示，必须在它们前面加上字符 \ ：

$ = \$　 &= \&　 %=\%　 #=\#　 _=_　 {=\{　 }=\}.

§2.5.5 特殊字符 §, †, ‡, ¶, ©, £

在计算机键盘上并没有这些特殊字符。可以用如下特殊命令来得到它们：

§=\S †=\dag ‡=\ddag ¶=\P ©=\copyright £=\pounds

在第 5 章中描述了如何生成希腊字母和其它数学符号。

§2.5.6 外文字母

在非英文的其它欧洲语言中，还存在一些特殊字符，它们也可以用 TEX得到。这些字符有：

œ={\oe} Œ={\OE} æ={\ae} Æ={\AE} å={\aa} Å={\AA}

¡=!‘ ø={\o} Ø ={\O} ł={\l} Ł ={\L} ß={\ss}

SS={\SS} ¿=?‘

Ångstrøm 可写为 {\AA}ngstr{\o}m, Karlstraße 可用输入 Karlstra{\ss}e 来得到。大写的 \SS 是 LATEX 2ε中的新命令。

§2.5.7 重音

在欧洲语言中，有很多发音记号或重音，用 TEX可以显示出大多数的重音符号：

ò=\‘{o} ó=\’{o} ô=\^{o} ö=\"{o} õ=\~{o}

ō=\={o} ȯ=\.{o} ŏ=\u{o} ǒ=\v{o} ő=\H{o}

o͡o=\t{oo} o̧=\c{o} ọ=\d{o} o̲=\b{o} o̊=\r{o}

（最后的命令 \r 是新出现在 LATEX 2ε中的。）上面的 *o* 只是一个示例，实际上可以用任何字母。而 i 和 j 应该专门提一下，在加上重音时必须去掉点，这只要在这两个字母前面加上 \ 就可以了。命令 \i 和 \j 就会得到 ı 和 ȷ。所以 ǐ 和 ǰ 是用 \u{\i} 和 \H{\j} 来得到的。

对于由非字母组成的重音命令，可以不使用大括号：

ò=\‘o ó=\’o ô=\^o ö=\"o õ=\~o ō=\=o ȯ=\.o

而由字母组成的重音命令就必须用大括号。在 TEX中也可以用其它的记号，但是这些记号很容易被混淆，由此这里也就不提及那些方法了。

§2.5.8 连写

在书籍印籍中，有些特殊组合的字母，并不是单独印出，而是用一个符号来显示，这称为 连写 。在 TEX中对于字母组合 ff, fi, fl, ffi 和 ffl 不是显示为

ff, fi, fl, ffi, ffl ， 而是 ff, fi, fl, ffi, ffl.

在 3.5.1 节中描述了如何强迫 TEX分开显示这些字符，而不是当作连写处理。对于排版象 shelfful 这样的单词，就必须这样做，因为当印刷时象通常那样处理为 ff 连写，那么单词 shelfful 就显得很古怪了。

§2.5.9 日期

在文本中的任何地方，都可以用命令 \today 来显示当前日期。日期的标准形式采用的是美国月，日，年（如 Novermber 15, 1995 ）形式。若要实现其它语言中的日期形式可以借助于 TEX命令 \day ， \month 和 \year ，这三条命令以数字的形式返回当前的参数值。在 C.3.3 节中有一个例子，演示了如何构造这样的一个新 \today 命令。

§2.6 脆弱的命令

有些命令的作用不只是局限于它所处的地方，而对文档的其它地方也有影响。例如，类似于 \chapter{标题} 这样的章节命令，不但只是在调用它的地方生成标题，而且也有可能在后续页面的顶部以不同的字样显示出来，甚到以第三种字样显示在目录中。类似于这样的可以显示在文档不同地方的参数，称为移动参数。

这样的参数从字面意义上看，是被移动重组的。（更精确地说，是未被完全解释的。）如果在移动参数中包含了某些命令，那么这些命令会发生变形，而未发挥其正常作用。有人形象地称之为分离组合。我们称这样的命令为脆弱的，而其它的不是这样易变的，能抵抗这种重组的，称为牢固的命令。

从本质上说，所有包含可省参数的命令，如 \begin 和 \end 都是脆弱的。脆弱命令可以包含在移动参数中，但要前缀命令 \protect ，这样就可以防止其被分离重组了。

在移动参数中未用 \protect 进行安全保护的脆弱命令，也不是一定要被重组。事实上，只有在极少数情形下才会发生分解。在本书的德语第一版中，只有章节标题中的元音变音（如 ä, ö 和 ü 这样的两点）需要 \protect\" ，以确保其正确地出现在目录中。（在 LATEX的后来版本中，重音命令不再是脆弱命令，而且在 LATEX 2ε中，它变得更牢固。）

对于脆弱命令， LATEX是不能正确处理的，因此会在屏幕上显示出一长串的错误信息。只要按一下回车键，用户可尝试继续处理下去，而不理睬对该命令的不正确对待。有可能还会出现更多的错误，但最终只要按足够多的回车， LATEX通常是能进行后面的处理的，除非那么被损坏的命令导致不可能进行下面的处理，而使程序停止运行。

只有下列命令包含移动参数：

- 所有传递文本信息给目录表的命令.
  它们是章节命令（ 3.3.3 节），\addtocontents, \addcontentsline（ 3.4.3 节）和 \caption （ 6.6.4 节）;
- \typein 和 \typeout 命令（ 8.1.3 节）;
- \markboth 和 \markright 命令（ 3.3.1 节）;

- 标题页上的 \thanks 命令（ 3.3.1 节);
- @- 表达式（ 4.8.1 节);
- \bibitem 的可省参数（ 4.3.6 节);
- 调用了 \makelabels 命令后的 \begin{letter} 命令（ A.1 节)。

只有当脆弱命令出现在上述命令的参数中时，才有必要用 \protect 命令来保护它们。在移动参数中，对于大多数命令而言，只要它是脆弱的，在其前面加上 \protect ，对处理没有什么影响。这个规则的例外就是长度命令（ 7.2 节) 和记数器命令（ 7.1.4 节)，如 \arabic 和 \value 。在这些命令前从来不要用 \protect 。

§2.7 练习

作为一部成功的自学教材的一部分，必须要用练习题。除了正文丰富的例子，读者应自己进行尝试外，在本书的教学中也应给出相当数量的进一步的练习，建议把它们当做必须完成的家庭作业。

所有的练习都是针对于 LaTeX 2$_\varepsilon$的，如果你用的还是 LaTeX2.09 ，那你就必须在开头用 \documentclass 的地方用 \documentstyle 。

练习 2.1本练习用一小段文本来测试一下对运行 LaTeX程序的基本步骤的掌握。同时也包含了几个简单的命令。用计算机的编辑器输入如下文本，并把它存到一个叫 exer.tex 的文件中：

```
\documentclass{article}
\begin{document}
Today (\today) the rate of exchange between the British
pound and American dollar is \pounds 1 =\$1.63, an
increase of 1\% over yesterday.
\end{document}
```

虽然对 LaTeX程序的执行在不同的计算机上可能不一样，但这里我们假定利用命令 latex 来调用它，那么可如下处理文件：

```
latex exer
```

注意：虽然文件名是 exer.tex ，但在调用 LaTeX时只需要给出基本名 exer 。

如果处理过程中没有出现任何错误信息，那么就会成功地生成 .dvi 文件 exer.dvi ，进而可以借助于打印机驱动程序对它进行转化。转化程序的名称与计算机系统有关。那么最后的打印结果应该如下 (日期应是当前日期)：

Today (August 15, 1999) the rate of exchange between the British pound and American dollar is £1 =$1.63, an increase of 1% over yesterday.

对这里所用的命令，用几点要注意：

- 在 \today 后面不必要加上空格，因为) 就完全可以终止它；

- 在 \pounds 后面的空格也可以省略，而且不会打印在结果中；
- 命令 \$ 和 \% 并不需要空格来终止它；如果用了空格，其将会打印在结果中。

练习 2.2: 找本书或期刊，摘录约有 3/4 页的文本，把它输入到 LaTeX文件中。注意用空行来分段。使用与练习 2.1 相同的命令集，即把文本放在命令 \begin{document}...\end{document} 之间，重复上述过程，以得到打印结果。

注意：所选文本中不应包含特殊结构，如缩进块，不同字样，居中文本，列举，数学公式，表格，等等。我们将在后续章节中介绍这些结构。