

# Chapter 2 Typesetting Text

Donald E. Knuth(高德纳)

2017 年 7 月 24 日

## 目录

<b>1 文章和语言的结构</b>	<b>3</b>
<b>2 断行和断页</b>	<b>3</b>
2.1 合理分段 . . . . .	3
2.2 连字符 . . . . .	4
<b>3 预定义好的字符串</b>	<b>4</b>
<b>4 特殊字符和符号</b>	<b>4</b>
4.1 引号 . . . . .	4
4.2 短划线和连字符 . . . . .	4
4.3 波浪线 . . . . .	5
4.4 斜杠 . . . . .	5
4.5 度 . . . . .	5
4.6 欧元符号 . . . . .	5
4.7 省略号 . . . . .	6
4.8 连字 . . . . .	6
4.9 重音符号和特殊符号 . . . . .	6
<b>5 国际语言支持/中文排版支持</b>	<b>6</b>
5.1 C <sub>T</sub> E <sub>X</sub> 的安装 . . . . .	6
5.2 C <sub>T</sub> E <sub>X</sub> 文档类 . . . . .	7
<b>6 单词间的空格</b>	<b>7</b>
<b>7 标题、章、节</b>	<b>7</b>
<b>8 交叉引用</b>	<b>8</b>
<b>9 脚注</b>	<b>9</b>
<b>10 强调</b>	<b>9</b>

目 录	2
<b>11 环境</b>	<b>9</b>
11.1 Itemize , Enumerate , and Description . . . . .	9
11.2 Flushleft , Flushright , and Center . . . . .	10
11.3 Quote , Quotation , and Verse . . . . .	10
11.4 Abstract . . . . .	10
11.5 Printing Verbatim . . . . .	11
11.6 Tabular . . . . .	11
<b>12 浮动体</b>	<b>13</b>
<b>13 保护脆弱命令</b>	<b>14</b>

## 1 文章和语言的结构

书写一篇文章最重要的一点是把想法、信息、知识传达给读者。 $\text{\LaTeX}$  与其他类型的排版系统不同，你只需要告诉它一个文本的逻辑结构和语义结构。然后根据文档类文件和各种样式的文件中所带有的“规则”来生成文件。

$\text{\LaTeX}$  中最重要的文本单位是段。分段需要在对应的源码中空一行，如果要继续写可以用换行符，用`\`或`\newline`。下面是说明该换行还是该另起一段的三个正确示例：

```
% Example 1
\ldots when Einstein introduced his formula
\begin{equation}
    e = m \cdot c^2 \ ; \ ,
\end{equation}
which is at the same time the most widely known
and the least well understood physical formula.
```

```
% Example 2
\ldots from which follows Kirchhoff' s current law:
\begin{equation}
    \sum_{k=1}^n I_k=0 \ ;
\end{equation}
```

```
Kirchhoff' s voltage law can be derives \ldots
```

```
% Example 3
\ldots which has several advantages.
\begin{equation}
    I_D = I_F - I_R
\end{equation}
is the core of a very different transistor model. \ldots
```

## 2 断行和断页

### 2.1 合理分段

- `\` or `\newline`:

断行但不是另起一段。`\`也在表格、公式等地方用于分行，而 `\newline` 只用于文本段落中。

- `\*`:

断行，但不另起一页和不断页。

- `\newpage` or `\clearpage`:

通常情况下两个命令都能起到另起一页的作用，但有一些区别：一是在双排版中 `\newpage` 只起到另起一栏的作用；二是涉及到浮动体的排版上行为不同。

- `\linebreak[⟨n⟩]` `\nolinebreak[⟨n⟩]` `\pagebreak[⟨n⟩]` `\nopagebreak[⟨n⟩]`:

不满足于  $\text{\LaTeX}$  默认的断行和断页位置，用其高速哪些地方适合断页，哪些地方不合适。 $\langle n \rangle$  代表合适/不合适的程度，取值范围为 0-4，不带可选参数时，缺省为 4。以上命令适合给出优先考虑断行断页\禁止断行断页的位置，但不适合直接拿来断行或断页，使用 `\newline` 或 `\newpage` 等是更好的选择。

## 2.2 连字符

如果遇到很长的英文单词，仅在单词之间的位置断行无法生成宽度匀称的行时，就要考虑从单词中间段开。对于绝大部分单词， $\text{\LaTeX}$  能够找到合适的断词位置，在断开的行尾加上连字符 -。如果一些单词没能自动断词，我们可以在单词内手动使用 `\-` 命令指定断词的位置。另外，也可以使用 `\hyphenation{word list}` 命令来指定使用连字符的位置，例如 `texttt\hyphenation{FORTRAN Hy-phen-a-tion}`，其中的 `word list` 是不区分大小写的。

- `\mbox{text}` 或 `\fbox{text}`:

会避免 `text` 被连字符分开。`\fbox\mbox` 多了个可见的框。

## 3 预定义好的字符串

- `\today` : 打印当天日期
- `\TeX` :  $\text{\TeX}$
- `\LaTeX` :  $\text{\LaTeX}$
- `\LaTeXe` :  $\text{\LaTeX 2}_\epsilon$

## 4 特殊字符和符号

### 4.1 引号

- 双引号: “...tex...”
- 单引号: ‘...tex...’  
“Please press the ‘x’ key.”

### 4.2 短划线和连字符

在  $\text{\LaTeX}$  中有下面四种横杠:

- - : - 连字符，用于连接词语。  
daughter-in-law, X-rated

- -- : – 短破折号，常用于连接数字表示起止范围。

pages 13–67

- --- : — 长破折号，常用于表示意思的转换。

yes—or no?

- \$-\$ : – 减号

0,1 and –1

### 4.3 波浪线

- \~{} : ~

<http://www.rich.edu/~bush>

- \$\sim\$ : ~

<http://www.clever.edu/~demo>

### 4.4 斜杠

- read/write: 不允许用连字符拆分
- read\slash write: 不允许用连字符拆分

### 4.5 度

- \$30\,^{\circ}\mathrm{C}\$ : 30 °C
- 30 \textcelsius: 30 °C
- 86 \textdegree F: 86 °F

### 4.6 欧元符号

首先需要在导言区加载 textcomp 包:

```
\usepackage{textcomp}
```

使用命令输出:

```
\texteuro
```

如果所用的字体不包含欧元符号或者想用别的字体的欧元符号, 导入 eurosym 宏包, 用 gen 来替换 official 参数可以使用和当前字体匹配的欧元符号:

```
\usepackage[official]{eurosym}
```

- \texteuro: €
- \euro: €

## 4.7 省略号

L<sup>A</sup>T<sub>E</sub>X 提供了命令 `\ldots` 来生成省略号，相对于直接输入三个点的方式更为合理。`\ldots` 和 `\dots` 是两个等效的命令。

- `.....`
- `\ldots:...`
- `\dots:...`

## 4.8 连字

有些相邻的字母在排版时会连接起来，可以通过 `\mbox` 命令避免它们相连。

- `ffshfilfluffia`: `ffshfilfluffia`
- `f \mbox{ } fshf \mbox{ } ilf \mbox{ } luf \mbox{ } f \mbox{ } ia`

## 4.9 重音符号和特殊符号

L<sup>A</sup>T<sub>E</sub>X 支持用命令输入各种西欧语言的特殊符号和重音，重音符号和特殊符号命令列表：

ò	ó	ô	õ
ō	ó	ö	ç
ö	ö	ő	q
q	q	oo	
œ	Œ	æ	Æ
å	Å		
ø	Ø	ı	Ł
ı	ı	ı	ı

# 5 国际语言支持/中文排版支持

L<sup>A</sup>T<sub>E</sub>X 对其他很多语言提供了支持。`babel` 宏包可以用于对各种语言进行适配。排版中文有两种方式，一种是使用 `xeCJK` 宏包，另一种是使用 `CTEX` 宏包和文档类。`CTEX` 宏包和文档类是对 `CJK` 和 `xeCJK` 等宏包的进一步封装。文档类包括 `ctexart`、`ctexrep`、`ctexbook`，分别是对 L<sup>A</sup>T<sub>E</sub>X 的三个标准文档类 `article`、`report`、`book` 的封装，对 L<sup>A</sup>T<sub>E</sub>X 的排版样式做了许多调整，以切合中文排版风格。最新版本的 `CTEX` 宏包/文档类甚至支持自动配置字体。

## 5.1 C<sub>T</sub>E<sub>X</sub> 的安装

`CTEX` 宏集依赖的宏包和宏集已被最常见的 T<sub>E</sub>X 发行版 T<sub>E</sub>XLive 和 MiK<sub>T</sub>E<sub>X</sub> 所收录。如果本地安装的 T<sub>E</sub>XLive 或 MiK<sub>T</sub>E<sub>X</sub> 不是完全版本，就需要通过这两个发行版提供的宏包管理器来安装宏包。

T<sub>E</sub>XLive 的宏包管理器是 `tlmgr`。在 Linux 系统上，一般需要 `sudo` 权限才能正确地执行 `tlmgr` 的功能。

直接使用 `sudo tlmgr [arg]` 时，可能会提示找不到 `tlmgr` 或没有这个命令。`sudo` 有一种内置的保护机制，只会使用安全的环境变量 `PATH`。如果 `TeXLive` 的路径不在 `sudo` 的安全环境变量内，它就找不到相关的命令。可以在终端执行 `sudo gedit /etc/sudoers`，然后将 `TeXLive` 的路径添加到 `sudo` 的 `secure_path` 中。不同路径用：隔开。

问题解决后，在终端中依次执行以下命令，以更新 `tlmgr` 宏包管理器、已安装的所有宏包、安装 `CTeX` 宏集。

```
sudo tlmgr update --self
sudo tlmgr update --all
sudo tlmgr install ctex
```

## 5.2 CTeX 文档类

`CTeX` 宏集提供了四个中文文档类：`ctexart`、`ctexrep`、`ctexbook` 和 `ctexbeamer`，分别对应 `LATeX` 的标准文档类 `article`、`report`、`book` 和 `beamer`。使用它们的时候，需要将涉及到的所有源文件使用 UTF-8 编码保存。

下面是使用 `ctexart` 文档类编写的一个例子：

```
\documentclass[UTF8]{ctexart}
\begin{document}
\end{document}
```

## 6 单词间的空格

`LATeX` 默认句子以句点、问号或者感叹号结尾。但是如果句点跟在一个大写字母后面，它不会认为这是句子结尾，因为大写字母后面跟句点往往是缩略词。

用户可以通过具体的命令来改变上面的默认设定。一个斜杠跟一个空格会产生一个不会被扩大的空格；一个波浪线（~）会产生一个既不能被扩大、也不能从这里断行的空格；在句点前使用 `\@` 命令，不管这个句点是不是跟在大写字母后面，都会指定这个句子到句点就结束。使用 `\frenchspacing` 命令可以强制不在一个句子后面插入多余的空格。如果使用 `\frenchspacing` 命令就没必要再用 `\@` 了。

例子：

```
Mr. Smith was happy to see her
cf. Fig. 5
I like BASIC. What about you?
```

## 7 标题、章、节

文档类中的几种分层次结构命令：

```
\section{...}
\subsection{...}
\subsubsection{...}
\paragraph{...}
\subparagraph{...}
```

如果你想把你的文件分成不同的部分而不影响章节编号的使用：

`\part{...}`

当使用 `report` 和 `book` 类时:

`\chapter{...}`

下面是两个比较特殊的情况:

- `\part` 命令不会影响 `chapter` 或 `section` 的编号。
- `\appendix` 命令没有任何参数, 会把 `chapter` (对于 `report`、`book`) 或 `section` (对于 `article`) 的数字编号转换成字母编号。

`\tableofcontents` 命令可以用于建立目录, 目录就会在这条命令所在的位置生成。一般新写的文档需要编译两次才能正确生成目录, 必要的时候  $\text{\LaTeX}$  也会提示需要编译三次。

以上的分章节的命令都有一个可以在命令名称后加一个星号的版本, 例如 `\section{...}` 命令, 加星号之后的命令为 `\section*{...}`, 不同之处是加星版本的章节命令对应的标题不会显示在目录里, 也不会被编号。

有时候章节标题太长, 这回导致其在目录里显示不佳。可以通过下面的命令在真真的标题前选择添加一个参数, 指定在目录中显示的标题。

```
\chapter[Title for the table of contents]{A long and espexially boring title, show in the text}
```

整个文档的标题是通过 `\maketitle` 命令产生。在调用 `\maketitle` 命令之前文档标题的内容需要由 `\title{...}`、`\author{...}`、`\date{...}` (可选) 等参数指定。在 `\author{...}` 命令的参数中, 可以用 `\and` 来间隔多个作者名字。

$\text{\LaTeX}$  2<sub>ε</sub> 在 `book` 文档有以下三个额外的命令, 可以进行前沿、正文、后记的结构划分。这三个命令还可以和 `\appendix` 命令结合, 生成有前沿、正文、附录、后记四部分的文档。

- `\frontmatter` 前言部分, 放置在文档主体的最开始 (`\begin*{document}`), 他会把页码变成罗马数字, 其后的 `\chapter` 不编号
- `\mainmatter` 正文部分, 页码为阿拉伯数字格式, 从 1 开始计数, 其后的章节编号正常
- `\backmatter` 后记部分, 页码格式不变, 继续正常计数; 其后的 `\chapter` 不编号

## 8 交叉引用

当需要对图片表格等进行引用时, 用下列命令:

- `\label{marher}`
- `\ref{marher}`
- `\pageref{marher}`

其中, `maeker` 是由用户自行定义的标识符。

A refernce to this subsection look like: “see section 8 on page 8.”



## 9 脚注

可以使用`\footnote{...}` 命令来添加脚注，脚注应该紧跟在它注解的词或句子（包括标点符号）后面。由于脚注会分散读者的注意力，所以尽量在文章主体说清楚，少用脚注。

Footnotes<sup>1</sup> are often used by people using L<sup>A</sup>T<sub>E</sub>X.

## 10 强调

在 L<sup>A</sup>T<sub>E</sub>X 中可以通过`\underline{...}` 命令来实现。但在印刷书籍中，一般通过 `\emph{...}` 命令，使用意大利字体进行强调。但并不是绝对的，需要结合具体语境。

*If you use emphasized text, then L<sup>A</sup>T<sub>E</sub>X uses the normal font for emphasizing.*

## 11 环境

环境的典型命令为`\begin{\emph{environment}} text \end{\emph{environment}}` 其中 `environment` 是环境的名字。

环境可以相互嵌套，例如：

```
\begin{aaa}
...
  \begin{bbb}
    ...
    \end{bbb}
  ...
\end{aaa}
```

### 11.1 Itemize , Enumerate , and Description

示例：

1. You can nest the list environments to your taste:

- But it might start to look silly.
- With a dash.

2. Therefore remember:

**Stupid** things will not become smart because they are in a list.

**Smart** things , though, can be presented beautifully in a list.

---

<sup>1</sup>This is a footnote.

## 11.2 Flushleft , Flushright , and Center

flushleft、flushright 和 center 环境分别会使段落左对齐、右对齐和居中。

示例：

flushleft:

This text is

left-aligned. L<sup>A</sup>T<sub>E</sub>X is not trying to make each line the same length.

flushright:

This text is right-  
aligned. L<sup>A</sup>T<sub>E</sub>X is not trying to make each line the same length.

center:

At the centre  
of the earth

## 11.3 Quote , Quotation , and Verse

quote 环境适合引用一些名言、重要的词句、示例等。

示例代码：

A typographical rule of thumb for the line length is:

On average, no line should be longer than 66 characters.

This is why L<sup>A</sup>T<sub>E</sub>X pages have such large why multicolumn print is used in newspapers.

quotation 环境 verse 环境很像。但由于 quotation 环境对每一段都会缩进，所以适合引用比较长的、一般有几段的内容；而 verse 环境很适合引用诗歌，利用\\或空行来分行。

示例：

I know only one English poem by heart.It is about Humpty Dumpty.

Humpty Dumpty sat on a wall:

Humpty Dumpty had a great fall.

All the King's horses and all the King's men

Couldn't put Humpty together again.

## 11.4 Abstract

在科学刊物中，一般会以一段摘要开头，让读者对这篇文章有一个整体认知，这是一个惯例。L<sup>A</sup>T<sub>E</sub>X 的 abstract 环境就是用于写摘要的，一般用于 article 文档类。

示例：

摘要

This abstract abstract.

## 11.5 Printing Verbatim

在 `verbatim` 和 `verbatim` 之间的所有文本都会原封不动地打印出来，文本中的  $\LaTeX$  命令不会被执行，所以可以用 `verbatim` 环境来插入一段代码。

如果要插入行间代码，可以用 `\verb+text+` 命令。其中的 `+` 只是分隔符，可以由用户虽已制定，但不能用字母、`*` 或空格，习惯上用 `|`。

示例：

The `\ldots` command ...

```
10 PRINT "HELLO WORLD ";
20 GOTO 10
```

`verbatim` 环境和 `verb` 命令都各自有一个加 `*` 的版本，区别是会把空格显示成 `_`。

示例：

```
the_starred_version_of
the_verbatim
environment_emphasizes
the_spaces_in_the_text

like_this:-)
```

`verbatim` 环境和 `\verb` 命令对符号的处理比较复杂，一般不能用在其他命令的参数里，否则多半会出错。

## 11.6 Tabular

`tabular` 环境可以用于排班表格， $\LaTeX$  会自动地调整表格每一列的宽度。具体命令如下：

`\{tabular}[pos]{table spec}`

其中，`table spec` 参数决定了表格的格式，`l`、`r`、`c` 分别会使单元格内容左对齐、右对齐、居中，不拆行；`|` 会绘制竖线；`p{width}` 会使单元格固定宽度为 `width`，可以自动拆行；可以通过 `@{...}` 来在单元格前后插入任意的文本，但同时它会使单元格前后额外添加的间距。

表格中每行的单元格数目不能多于列格式里 `l\c\r\p\` 的总数（可以少于这个总数），否则出错。

`pos` 参数用于指定表格相对于环绕为基线的垂直位置，可以为 `t`、`b` 和 `c`，分别代表顶部、底部和中间。

在 `tabular` 环境内，`&` 会跳到下一列，`\\` 会新起一行，`\hline` 会插入一条水平线。使用 `\cline{i-j}` 会插入一部分水平线，其中 `i` 和 `j` 是列的编号。

示例 1：

7C0	hexadecimal
3700	octal
11111000000	binary
1984	decimal

示例 2：

Welcome to Boxy's para- graph. We Sincerely hope you'll all enjoy the show.
---

如果想让表格的某一列按小数点对齐，又不使用额外的宏包 (dcolumn)，可以采用一种比较折中的想法：把该列的所有小数部分，各作为一列，然后在 tabular 环境中使用 @{\dots} 作为整数列和小数列的分隔符。整数部分和小数部分之间不要忘了用 & 隔开，因为它们现在是两列了。为了在这两列的表头仅显示一个标签，可以使用 \multicolumn 命令。

示例 3:

Pi expression	Value
$\pi$	3.1416
$\pi^\pi$	36.46
$(\pi^\pi)^\pi$	80662.7

示例 4:

Ene	
Mene	Muh!

有时候需要为整列修饰格式，比如整列改变为粗体，如果每个单元格都加上 \bfseries 命令会比较麻烦。array 宏包提供了辅助格式 > 和 <，用于给格式前后加上修饰命令。辅助格式甚至支持插入 \centering 等命令改变 p{width} 列格式的对齐方式，一般还要加额外的命令。辅助格式甚至支持插入 \centering 等命令会破坏表格环境里 \\ 换行命令的定义，\arraybackslash 用来恢复之，如果不加 \arraybackslash 命令，也可以用 \tabularnewline 命令代替原来的 \\ 实现表格换行。

示例 5:

<i>italic*</i>	normal
<i>column*</i>	column

示例 6:

Some center-aligned long text.
-----------------------------------

有时 L<sup>A</sup>T<sub>E</sub>X 默认的表格会比较拥挤，可以通过改变 \arraystretch 和 \tabcolsep 参数来调节。

示例 7:

These lines
are tight

less cramped
table layout

如果只想增加表格中某一行的高度，可以通过添加一条不可见的“支柱”来实现。即使用`\rule` 命令，并把宽度设置为 0。

示例 8:

Pitprop ...
Strut

如果想排版较长的表格，可以使用 `longtable` 环境。在 `booktabs` 宏包中，还有很多其他的命令可以美化表格。

## 12 浮动体

一般的刊物、出版品都会有各种图表，由于这类元素不能跨页，所以需要特殊对待。有一种方法是每当遇到一个图或表在当前页放不开的时候，就新起一页。这种方法会让页面有比较多的留白，看起来会不美观。

一种比较好的解决方法是当前页放不开的时候，就新起一页。这种方法会让页面去，当前页面用正文进行填充。 $\text{\LaTeX}$  提供了两种浮动体环境：`figure` 和 `table`，分别针对图片和表格。任何放在 `figure` 和 `table` 环境中的东西都会被视为是浮动体。

`\{figure}[placement specifier] or \begin{table}[...]`

所有的浮动体环境都会有一个叫 `placement specifier` 的可选的参数，可以用于指定这个浮动体被允许移动到哪里。下面是 `placement specifier` 参数的可选值（默认为 `tbp`，优先级按照 `h-t-b-p` 排列，与顺序无关）：

- `h`，当前位置（代码所处的上下文）
- `t`，顶部，如果是当前页排版可能出现在代码之前
- `b`，底部
- `p`，一个或多个浮动体被放在单独的页面中，这个页面被称为浮动液面（`float page`），与之对应，有文本的页面称为文本页（`text page`）
- `!`，再决定位置时忽略文本页的限制（ $\text{\LaTeX}$  对每个位置的浮动体的总数和占用大小等有一定的限制），只有浮动页的限制（`\floatpagefraction` 和 `\dblfloatpagefraction`）起效。

$\text{\LaTeX}$  会根据用户指定的 `placement specifier` 参数来防止每个浮动体。如果一个浮动体不能被放置在当前页，他就会被加入 `figures` 队列或 `tables` 队列。当新起一页的时候， $\text{\LaTeX}$  会先检查能否把这一页作为一个浮动页来放置队列中的浮动体。如果不行，队列中的第一个浮动体就会被认为是这个新起的一页刚产生的，然后  $\text{\LaTeX}$  就再次根据他们的 `placement spacier` 参数在这个新起的一页进行放置。

$\text{\LaTeX}$  会严格地按照浮动体在队列中的顺序来执行这一过程，这也就是为什么如果 `figures` 队列中的第一

