

# 漫谈 $\text{\LaTeX}$ 排版常见概念误区

别把  $\text{\LaTeX}$  当 Word 用!

林莲枝

2018 年 3 月 20 日

这不是新手教程哦 ☆

也不是排版入门教程……

对排版的一些规范、习惯有些了解，有助于学习  $\text{\LaTeX}$

1short 是好东西，请读完！



- 不是文字处理器
- 和 Word, OpenOffice 的功能、需求有区别
- 排版的目的?
  - 良好的阅读体验 (*readability*) 
  - 排版、出版业界有一定的常见规范与考量
  - **L<sup>A</sup>T<sub>E</sub>X 是排版系统，不是文字处理器**

不讨论安装，不过还是提一下……



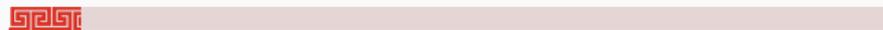
套装发行版

编译器

编辑器

CTeX 套装发行版和 CTeX 宏包/文档类是两回事!

## 关于版面



## ——为什么页距那么大？真是浪费空间。



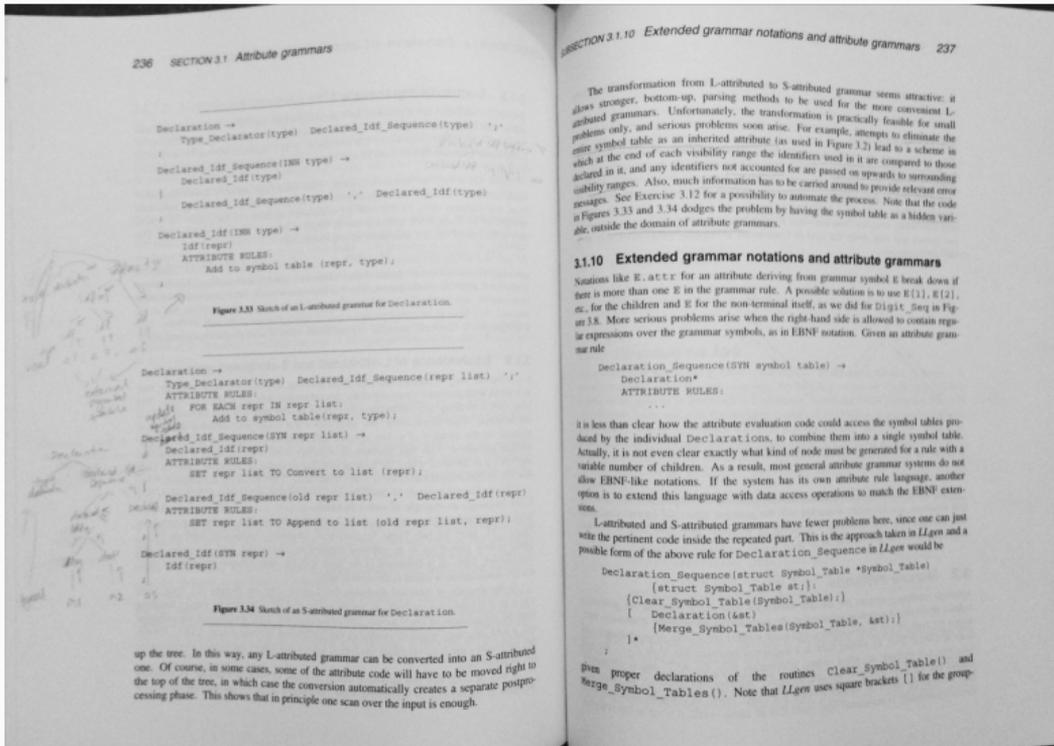
- 把版面挤得满满的，未必利于阅读体验
- 理想行长和字号的关系

*Anything from 45 to 75 characters is widely regarded as a satisfactory length of line for a single-column page set in a serifed text face in a text size. The 66-character line ... is widely regarded as ideal. For multiple-column work, a better average is 40 to 50 characters.*

*—Robert Bringhurst, “The Elements of Typographic Style”*

- Margin notes

# 实例：Margin notes, 双面印刷



## ——为什么排版书籍或者毕业论文模板时，页距忽大忽小？是 bug 吗？



- 排版时要决定单面或者双面印刷
- twoside vs oneseide
- 奇偶数页 ( recto and verso pages )

## ——为什么模板里会突然有空白页？

- 双面印刷时，新章节的安排
- `openright` vs `openany`
- 默认选项是哪些呢？

文档类	默认选项
<code>article</code> , <code>ctexart</code> , <code>scrartcl...</code>	<code>oneside</code> (没有 <code>chapter</code> )
<code>report</code> , <code>ctexrep</code> , <code>scrreprt...</code>	<code>oneside</code> , <code>openany</code>
<code>book</code> , <code>ctexbook</code> , <code>scrbook</code> , <code>memoir...</code>	<code>twoside</code> , <code>openright</code>

## 关于浮动体



为什么我的图、表格到处乱跑！ $\LaTeX$  这好大的 bug!

这不是 bug，是  $\LaTeX$  排版的精髓之一

有些强迫症宝宝希望保留浮动体的标题以及编号的功能，但是希望浮动体「乖乖待在插入的位置」。

对于这些小朋友，老夫必须说：「这是病，得治」。

## 「浮动体」了解一下

- 排版出来供阅读的文本是主体
- 图、表为辅
- 插在页面的文本段落之间，打断（妨碍）阅读节奏 (reading flow)
- 因此图、表都闪一边去，不要打断我的阅读节奏！  
页面顶端、尾部都可以，甚至是之后的页面！

此为**浮动体**。

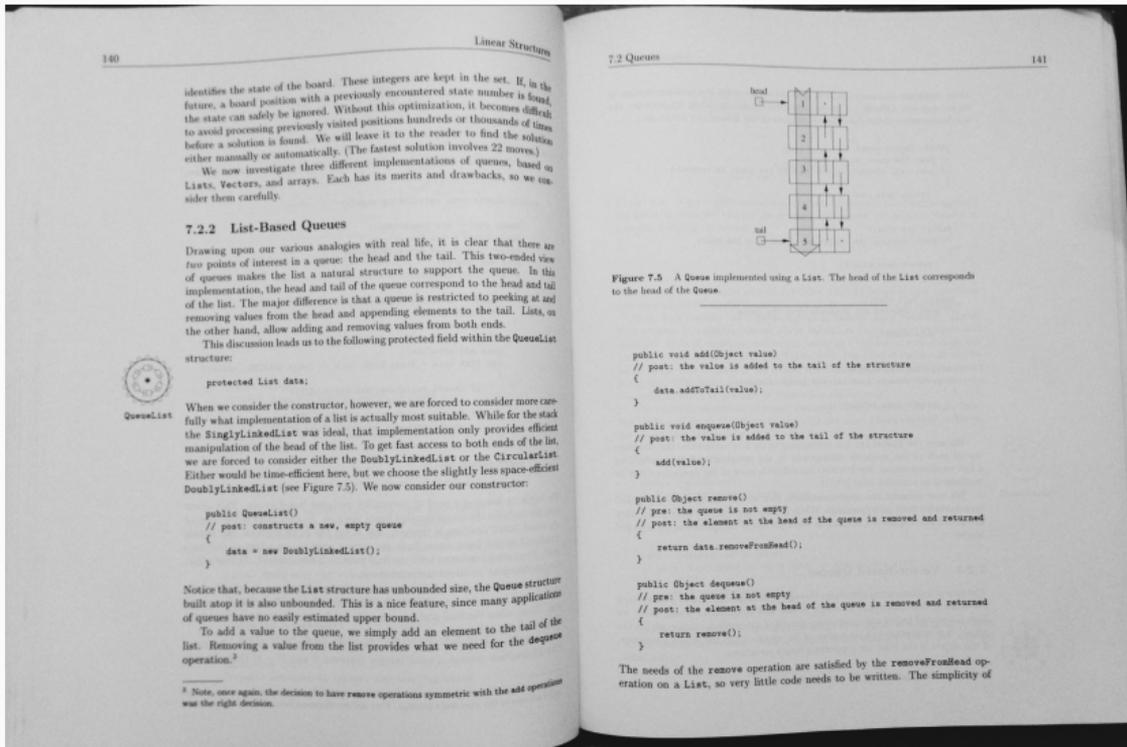
图和说明不在同一页，那读者怎么读得明白啊？  
还说什么增进阅读体验?? 差评!!

……你以为交叉引用 `\label` 和 `\ref` 的存在意义何在？

## 实例：浮动体与交叉引用



# 实例：浮动体与交叉引用（续）



# 实例：浮动体与交叉引用（续）

3. The usage is consistent. In the alternative design, two of the methods are cursor-relative and depend on a valid cursor. The root-oriented method depends on the cursor being invalid. In our design, the `insert` method inserts new values where a cursor is found to be invalid.

We expect, of course, that there will be a tradeoff. In particular, it seems that when the cursor falls off the tree, the tree must remember how that happened so that any impending insert can be accomplished correctly. Should we decide later to expand the repertoire of insertion routines, it is fairly easy to see that the alternative implementation can be engineered in terms of our single-method insert (see Problem 10.8).

## 10.3 Motivating Example: Expression Trees

Most programming languages involve mathematical expressions that are composed of binary operations applied to values. An example from Java is the simple expression  $R = 1 + (L-1)*2$ . This expression involves four operators (+, \*, -, and \*) and 5 values (R, 1, L, 1, and 2). Languages often represent expressions using binary trees. Each value in the expression appears as a leaf, while the operators are internal nodes that represent the reduction of two values to one (for example,  $L-1$  is reduced to a single value for use on the left side of the multiply). The expression tree associated with the above expression is shown in Figure 10.3a. We might imagine that the following code constructs the tree and prints it:

```
BinaryTreeNode via,v1b,v2,vL,vR,t;  
  
// set up values 1 and 2, and declare variables  
via = new BinaryTreeNode(new value(1));  
v1b = new BinaryTreeNode(new value(1));  
v2 = new BinaryTreeNode(new value(2));  
vL = new BinaryTreeNode(new variable("L",0)); // L=0  
vR = new BinaryTreeNode(new variable("R",0)); // R=0  
  
// set up expression  
t = new BinaryTreeNode(new operator("-",),vL,v1a);  
t = new BinaryTreeNode(new operator("+"),t,v2);  
t = new BinaryTreeNode(new operator("*"),v1b,t);  
t = new BinaryTreeNode(new operator("="),vR,t);  
  
// evaluate and print expression  
System.out.println(eval(t));
```

Once an expression is represented as an expression tree, it may be evaluated by traversing the tree in an agreed-upon manner. Standard rules of mathematical precedence suggest that the parenthesized expression  $(L-1)$  should be evaluated first. (The  $L$  represents a value previously stored in memory.) Once

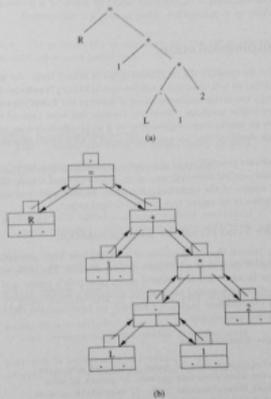


Figure 10.3 Expression trees. In (a), an abstract expression tree representing  $R = 1 + (L - 1) * 2$ . In (b), a possible connectivity of an implementation using references.

- h 就摆这里
- t 页面顶端 OK
- b 页面底端 OK
- p float page, 只有浮动体的页面
- ! 求你了，听听我的请求吧……

- 以上选项不是你说了算，只是给  $\text{\LaTeX}$  排版算法参考一下你的偏好！算法还要考虑很多其它排版规范的！
- 默认选项是 `[btp]`，看，没有 `[h]`
- 真的想把图、表摆在「这里」，`[hbt!]` 很足够了。`float` 宏包的 `[H]` 不建议多用。
- （其实吧，定稿前都不应该纠结这种细节……）

- 注意，有的期刊只准许图、表摆在页面顶端！
- 还有的期刊要求所有图、表都在稿件最后面！

——不行呀我的情形实在不能让它浮动!!!

- 「文章接近完结的地方太多图、表，都漂到文献列表之后了!!」
- 有可能真的是你的文字叙述讨论太少了……
- 考虑一下把图、表都做成附录，摆在 `\appendix` 之后，看起来没那么怪。

——定稿了，文章组织不能再改动了啊……

- 实在不行的话，`\bibliography{...}` 之前加一个 `\clearpage`，可以把至今为止遇到的所有的浮动体都一口气排出来。
- ……是啊那当然会分页啊。☺
- **⚠ 不到定稿，确认别无他法，不建议使用 `placeins`**

```
\usepackage{placeins}
...
\FloatBarrier
\bibliography{...}
```

## 你要的真的是「浮动体」吗?

- 简历里直接 `\includegraphics{...}` 或者 `\begin{tabular}... \end{tabular}` 就可以了, 不需要 `figure` 或 `table` 环境。
- 个别无法使用浮动体 ( 比如 `\marginpar`、使用 `multicols` 的海报 ), 却需要图、表标题的话, 可以用 `caption` 宏包的 `\captionof`。不要滥用!

```
\begin{center}
\includegraphics[width=.9\linewidth]{foo}
\captionof{figure}{This is not a float}
\end{center}
```

## 关于文字与段落



## ——段首怎么没缩进？又是 bug？

- (西文排版) 缩进的意义何在呢？
- 让读者知道，新的一段开始了
- 有的期刊格式或体例 (house style) 认为，`\chapter`，`\section`，`\subsection` 标题已经很清楚的表示「这是新段落」。因此标题过后的第一段，不必再缩进。
- 也有的期刊及体例是要求每一段都缩进的。
- 因此不要自己更改期刊模板的格式。模板已经设定好期刊要求的提交稿件格式。
- 如果是自己的文件，可以用 `indentfirst` 宏包。

——没有呀，我的每一段文字都没有缩进

看看你的代码？

```
\section{Introduction}
```

```
Maecenas urna ac sapien tincidunt lobortis. Nunc feugiat faucibus varius. Ut sed  
purus nunc. Ut eget eros quis lectus mollis pharetra ut in tellus.\\[12pt]
```

```
Pellentesque ultricies velit sed orci pharetra et fermentum lacus imperdiet.
```

```
Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos  
himenaeos. Suspendisse commodo ultrices mauris, condimentum hendrerit lorem  
condimentum et.\\ \\ \\
```

```
Pellentesque urna augue, semper et rutrum ac, consequat id quam. Proin lacinia  
aliquet justo, ut suscipit massa commodo sit amet.\\ \\ \\
```

……额滴个天啊。

——啊知道了，要自己加 `\indent` 或者 `\hspace{1cm}` 是吧。……不，别。

## 「行」和「段」

- 两者是有分别的，并对  $\text{\LaTeX}$  的算法有直接影响。
- 分段：空一行（或多行），或用 `\par`（少用）
- 强制分行：`\\`

```
\section{Introduction}
```

```
Maecenas urna ac sapien tincidunt lobortis. Nunc feugiat faucibus varius. Ut sed purus nunc. Ut eget eros quis lectus mollis pharetra ut in tellus.
```

```
Pellentesque ultricies velit sed orci pharetra et fermentum lacus imperdiet. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Suspendisse commodo ultrices mauris, condimentum hendrerit lorem condimentum et.
```

```
Pellentesque urna augue, semper et rutrum ac, consequat id quam. Proin lacinia aliquet justo, ut suscipit massa commodo sit amet.
```

## ——那样的话段落之间不是没有空间了？

- 段落之间留不留空间 ( paragraph skip ), 也是期刊格式/体例决定的。
- 模板编译出来, 段落之间没有留空, 那是期刊自己的要求模板才会这样的效果, 不要自己去改。
- 如果是自己的文件, 可以用 `parskip` 宏包。
- `memoir`, `scrartcl` 等文档类有自己的机制, 请用心阅读宏包手册。

——用了 `parskip` 宏包，段首又不缩进了！ $\text{\LaTeX}$  真难侍候！

- 再思考一次。（西文排版）缩进的意义何在呢？
- 让读者知道，新的一段开始了。
- `parskip` 宏包一进来，段与段之间已经有空间了，已经可以区别段落了。
- 段首缩进又多此一举了……
- 无可否认有的高校论文要求是要缩进又要段落之后留白啦……
- 如果是自己的文件：

```
\usepackage{parskip}
\setlength{\parindent}{2em}
```

——我用了段落的命令啊，为什么不分段也不缩进？还都给我弄粗体了？

```
\paragraph{We study the effects of factor X on the Y process under circumstances  
of ZZZ. Experiments were carried out to measure...}
```

……快停下……

——哈哈你用错了吧，这样才对！

```
\pararagraph{}We found that...
```

……你也快住手!!!!

- `\paragraph{}`是个章节标题命令，不是「分段」
- `\subsubsection`再下一层就是 `\paragraph{}`，还有 `\subparagraph{}`。这两都很少、很少用到，甚至有的期刊、毕业论文格式不允许用。

——可不可以取消断字机制？看起来好怪，正确的英文不是这样的吧。

- ……误会大了啊。
- 你去拿一本国外出版的、内文两端对齐排版的书，仔细看一看。 [去看看](#)
- $\text{\LaTeX}$  的断字算法是  $\text{\LaTeX}$  的精髓，Knuth 与学生们的心血。
- 从头说起吧。断字的意义何在？

## 断字的意义何在？

Sesame snaps and caramels toffee tart gingerbread. Sugar plum ice cream lemon drops wafer. Dessert dessert marshmallow. Chocolate cake dessert gingerbread pie. Powder carrot cake caramels brownie sugar plum muffin croissant. Cheesecake brownie marshmallow biscuit apple pie muffin lemon drops cotton candy. Carrot cake cake tiramisu chocolate. Tiramisu halvah caramels chocolate. Wafer chupa chups wafer fruitcake cake.

Sesame snaps and caramels toffee tart gingerbread. Sugar plum ice cream lemon drops wafer. Dessert dessert marshmallow. Chocolate cake dessert gingerbread pie. Powder carrot cake caramels brownie sugar plum muffin croissant. Cheesecake brownie marshmallow biscuit apple pie muffin lemon drops cotton candy. Carrot cake cake tiramisu chocolate. Tiramisu halvah caramels chocolate. Wafer chupa chups wafer fruitcake cake.

## 有的体例要求左对齐；左对齐也可以断字

`\raggedright`

Sesame snaps and caramels toffee tart  
gingerbread. Sugar plum ice cream lemon  
drops wafer. Dessert dessert marshmallow.  
Chocolate cake dessert gingerbread pie.  
Powder carrot cake caramels brownie sugar  
plum muffin croissant. Cheesecake brownie  
marshmallow biscuit apple pie muffin lemon  
drops cotton candy. Carrot cake cake  
tiramisu chocolate. Tiramisu halvah  
caramels chocolate. Wafer chupa chups  
wafer fruitcake cake.

`\usepackage{ragged2e}\RaggedRight`

Sesame snaps and caramels toffee tart gin-  
gerbread. Sugar plum ice cream lemon  
drops wafer. Dessert dessert marshmal-  
low. Chocolate cake dessert gingerbread  
pie. Powder carrot cake caramels brownie  
sugar plum muffin croissant. Cheesecake  
brownie marshmallow biscuit apple pie muf-  
fin lemon drops cotton candy. Carrot cake  
cake tiramisu chocolate. Tiramisu halvah  
caramels chocolate. Wafer chupa chups  
wafer fruitcake cake.

不要自己更改模板的设置。

- 算法需要时间计算
- 网页、文字处理器如果全面使用，可能会导致显示缓慢，所以……
- 默认英文断字规则，如果写的是其它语文，记得一定要载入相应的 `babel` 或者 `polyglossia` 语言选项。

**pdf<sub>l</sub>atex, latex** `\usepackage[ngerman]{babel}`

**xelatex, lualatex** `\usepackage{polyglossia} \setmainlanguage{German}`

- 行尾太多断字也不美观，可以载入 microtype 自动微调
  - 在保持排版美观的前提下尽量减少断字
  - 自动实现西文的标点悬挂
  - ( 中文排版没有此一说 📍 )
  - 文中太多不知太长、不知从何断字的词语 ( 比如化学科系的专有名词 )，可能导致编译时间拉长甚至无限循环

- Donald E Knuth and Michael F Plass (1981). “Breaking paragraphs into lines”. In: *Software: Practice and Experience* 11.11, pp. 1119–1184. 
- Michael F Plass (1981). “Optimal pagination techniques for automatic typesetting systems”. PhD thesis. Stanford University. 
- Franklin Mark Liang (1983). “Word Hy-phen-a-tion by Com-put-er”. PhD thesis. Stanford University. 
- Anne Brüggemann-Klein, Rolf Klein, and Stefan Wohlfeil (2003). “On the pagination of complex documents”. In: *Computer Science in Perspective*. Springer, pp. 49–68. 
- Jonathan Fine (2000). “Line breaking and page breaking”. In: *TUGboat* 21.3, pp. 210–221. 

## ——为什么有的页面，段落之间、图、表之后的空间特别大？

- 一般的排版规范，要求内容「撑」到主体文字版面的顶部和底部。
- 段落之间、浮动体、行间公式之后的空白其实是可伸缩的。
- 如果某一个页面的文字内容不太足够， $\LaTeX$  会自动调整（增加）这些留白，让最后一行内容可以撑到版面底部。
- 如果内容严重不足，留空太多，版面当然不好看。这时  $\LaTeX$  就会给出警告信息：  
`Underfull \vbox (badness 2573)`

## ——怎么改啊？实在太难看了！

- 没有定稿之前都不必去理它，一般上再多写一些内容就自然会好了。
- 可以尝试避免太多章节标题，却太少主体文字的情况。
- 实在很看不习惯、或者是导师很不喜欢的话，用 `\raggedbottom` 解决。  
⚠ 不到定稿、不到导师不让你毕业的地步，不要滥用。
- ……毕竟搞得和 Word 一模一样的效果的话，还何苦用  $\text{\LaTeX}$  呢……

## 关于字体



- `\bf`, `\it`, `\sc` 不要用了, 没有叠加效果、已过时。
- `\textit` 和 `\emph` 不一样, 有些语系排版「强调」不是用意大利体表现的。
- 试一试: `\emph{This is a \emph{nice} example.}`
- 中文字体与排版没有所谓的「斜体」; Word 那是伪斜体 (可能难看)。
- 「斜体」`\textsl` 和「意大利手写体」`\textit` 不一样。
- 这么一说, *italics* 和楷体有那么一点点的相似概念吧。

- 不是每一个字体都有多个字重。因此有的字体没有**粗体**，比如 EB Garamond。
- 只能用伪粗体（可能难看），中文有时可以用黑体风格来代替。
- pdf<sub>l</sub>atex 编译时，`\usepackage[utf8]{inputenc}\usepackage[T1]{fontenc}` 是好习惯。
- times 和 txfonts 宏包已经过时了，很多 bug，尽量用 mathptmx 或 newtxtext, newtxmath 宏包。

## 关于文献格式



不要擅自更改模板里的引用、文献格式。

## 文献格式由 `\bibliographystyle...` 决定的

- 数字引用 numerical 还是作者—年份 author-year?
- 作者姓氏字母顺序排序 (plain, acm), 还是引用顺序排序 (unsrt, IEEEtran)?
- 有的格式是 >3 作者就 et al. 缩短, 有的是 >4, 有的是 >6。甚至有绝不 et al. 的……都正常, 交给模板和 `\bibliographystyle` 去处理就好。
- 大多 Author-year 体系格式里, 完全同样的作者 (团队) 在同一年刊登的论文, 会自动被标为 2012a, 2012b。
- John Doe, Jane Smith and Foo Bar  $\neq$  John Doe, Foo Bar and Jane Smith.
  - 这两张论文即使都是 2012 年刊出来的, 也不会标为 2012a, 2012b.
  - 反倒可能在引用时列出完整的 (Doe, Smith and Bar, 2012) 及 (Doe, Bar and Smith, 2012), 以免导致引用歧义

# 切记!!

- 供投稿的模板格式未必和最终刊登出来的效果一样，出版社故意的。
- 因此不要自寻烦恼，一直想要把格式调成刊出来的一样。
- 身为作者你应该专注的是内容而非格式，不然何必提供模板？

没有提供  $\text{\LaTeX}$  模板?

快去问清楚, 提交稿件时是不是一定要见到.doc 档!

是的话, 别用  $\text{\LaTeX}$ !