

一份不太简短的 L^AT_EX 2_ε 介绍

或 93 分钟学会 L^AT_EX 2_ε

原版作者: Tobias Oetiker

Hubert Partl, Irene Hyna and Elisabeth Schlegl

原版版本: Version 3.20, 09 August, 2001

中文翻译: 中国 C_T_EX 用户小组

中文版本: 版本 3.20, 二 二年五月

Tobias Oetiker 拥有版权 © 2000 以及 LShort 的所有的发行版本。保留所有权利。

这份文档是免费的；在自由软件组织颁布的 GNU 通用出版许可证的条款下，你可以再版或者修改它。许可证可以是第二版，或者任何后继版本（随你意）。

发布这份文档是希望它会有用，但并不提供任何保障；甚至没有用于商业的或者适用某一特定目的的暗含保证。更多的细节请查看 GNU 通用出版许可证。

你应该随这份文档收到一份 GNU 通用出版许可证的拷贝；如果没有，写信到自由软件组织，地址：675 Mass Ave, Cambridge, MA 02139, USA。

鸣谢！

在这份介绍中使用的许多材料来自一个奥地利人使用德语撰写的 L^AT_EX 2.09 介绍：

Hubert Partl <partl@mail.boku.ac.at>

Zentraler Informatikdienst der Universität für Bodenkultur Wien

Irene Hyna <Irene.Hyna@bmwf.ac.at>

Bundesministerium für Wissenschaft und Forschung Wien

Elisabeth Schlegl <no_email>

in Graz

如果你对德文文档有兴趣，你可以找到一个由 Jörg Knappen 针对 L^AT_EX 2_ε 更新的版本，在 CTAN 的位置是：

CTAN:/tex-archive/info/lshort/german

在准备这份文档时，我向 comp.text.tex 中的专家们进行咨询。我得到了大量的回应。下面的人帮助进行了校正、建议和提供改进材料。他们付出了极大的热情帮助我将这份文档实现成现在这个样子。我对他们所有人表示衷心的感谢。自然地，你在本书中找到地所有错误都是我的失误。甚至你找到每一个拼写正确的单词，那一定是下面列出的这些人之一帮了我一把的结果。

Rosemary Bailey, Friedemann Brauer, Jan Busa, Markus Brühwiler, David Carlisle, José Carlos Santos, Mike Chapman, Christopher Chin, Carl Cerecke, Chris McCormack, Wim van Dam, Jan Dittberner, Michael John Downes, David Dureisseix, Elliot, David Frey, Robin Fairbairns, Jörg— Fischer, Erik Frisk, Frank, Kasper B. Graversen, Alexandre Guimond, Cyril Goutte, Greg Gamble, Neil Hammond, Rasmus Borup Hansen, Joseph Hilferty, Björn Hvittfeldt, Martien Hulsen, Werner Icking, Jakob, Eric Jacoboni, Alan Jeffrey, Byron Jones, David Jones, Johannes-Maria Kaltenbach, Michael Koundouros, Andrzej Kawalec, Alain Kessi, Christian Kern, Jörg Knappen, Kjetil Kjernsmo, Maik Lehardt, Alexander Mai, Martin Maechler, Aleksandar S Milosevic, Claus Malten, Kevin Van Maren, Lenimar Nunes de Andrade, Hubert Partl, John Reffing, Mike Ressler, Brian Ripley, Young U. Ryu, Bernd Rosenlecher, Chris Rowley, Hanspeter Schmid, Craig Schlenter, Christopher Sawtell, Geoffrey Swindale, Josef Tkadlec, Didier Verna, Fabian Wernli, Carl-Gustav Werner, David Woodhouse, Chris York, Fritz Zaucker, Rick Zacccone, and Mikhail Zotov.

本文档的翻译工作由 C_T_EX 版主“经典问题”倡议，历经近十个月才得以完成。期间参与翻译工作的朋友有：

C _T _E X 论坛 ID	翻译章节	源文件名
经典问题	前言	overview.tex
高原之狼	第一章	things.tex
controlong	第二章	typeset.tex
cxterm	第三章	math.tex, lssym.tex
aloft	第四章	spec.tex
ganzhi	第五章	custom.tex

在此特向这些奉献者表示感谢！

前言

L^AT_EX[1] 是一种排版系统，它非常适用于生成高印刷质量的科技和数学类文档。这个系统同样适用于生成从简单的信件到完整书籍的所有其他种类的文档。L^AT_EX 使用 T_EX[2] 作为它的格式化引擎。

这份短小的介绍描述了 L^AT_EX 2_ε 使用，对 L^AT_EX 的大多数应用来说应该是足够了。参考文献 [1, 3] 对 L^AT_EX 系统提供了完整的描述。

对大多数计算机，从个人计算机（PC）和 Mac 到大型的 UNIX 和 VMS 系统，L^AT_EX 都有适用版本。在许多大学的计算机机房，你会发现系统中已经安装了 L^AT_EX 并能够使用。在 *Local Guide* [4] 中应该提供了关于如何存储本地的 L^AT_EX 安装的信息。如果你对如何入门有疑问，向提供这个小册子的人咨询。这份文档的职责不是告诉你如何安装并设置一个 L^AT_EX 系统，而是教你如何编写你的文档，使得它们可以被 L^AT_EX 处理。

这份介绍分成了五个章节：

第一章 告诉你关于 L^AT_EX 2_ε 文档的基本结构。你也会了解一点 L^AT_EX 的历史。阅读这一章后，你应该对 L^AT_EX 有了一个粗略的印象。这个印象仅仅是一个框架，但是它可以使你整合在其他章节中提供的信息成为一个完整的画面。

第二章 探究排版文档的细节。它解释了大部分基本的 L^AT_EX 命令和环境。在阅读完这一章之后，你就能够编写你的第一份文档了。

第三章 解释了如何使用 L^AT_EX 排版公式。同时，大量的例子会帮助你理解如何使用 L^AT_EX 的一个主要能力。在这个章节的结尾，你会找到列写了 L^AT_EX 中所有可用的数学符号的表格。

第四章 解释了索引和参考文件的生成、EPS 图形的插入和一些其他有用的扩展。

第五章 包含一些潜在的危险信息，有关对 L^AT_EX 产生的标准文档布局如何提供可选方法。它会告诉你如何改变一些 L^AT_EX 看起来有点糟糕的东西，得到漂亮的输出。

按照连续的顺序阅读这些章节是很重要的，这本书毕竟不长。一定要认真阅读例子，在这本书中你会找到所有的例子，因为在各种例子里包含了大部分的信息。

如果你需要获得任何 \LaTeX 相关的材料，在任何一个“全面的 \TeX 归档网络”（CTAN）站点中找找看。它的主页是 <http://www.ctan.org>。所有的宏包也可以从 $\text{\textit{ftp}}$ 归档站点 <ftp://www.ctan.org> 和遍布全球的各个镜像站点中获得。即可在 <ftp://ctan.tug.org>（美国），<ftp://ftp.dante.de>（德国），<ftp://ftp.tex.ac.uk>（英国）找到它们。如果你不在这些国家中，那么选择离你最近的归档站点。

在本书中你会找到有关CTAN的其他参考资料。特别地，给出你可能需要下载的软件和文档的指示。这里没有写出完整的 `url`，而是仅仅写了带有你应该去的在CTAN树结构定位的 `CTAN:` 信息。如果你想在自己的计算机上运行 \LaTeX ，请在 `CTAN:/tex-archive/systems` 中看看有什么。

如果你有意在这份文档中增加、删除或者改变一些内容，请通知我。我对从 \LaTeX 初学者的反馈特别感兴趣，关于这份介绍哪些部分很容易理解，哪些部分可能需要更好地解释。

Tobias Oetiker <oetiker@ee.ethz.ch>

Department of Electrical Engineering,
Swiss Federal Institute of Technology

这份文档的最新版本可以在
`CTAN:/tex-archive/info/lshort`
得到。
有关这份文档的最新中文翻译，请到
<http://bbs.ctex.org>
查找最新信息。

目录

鸣谢！	iii
前言	vi
1 基本知识	1
1.1 游戏的名目	1
1.1.1 \TeX	1
1.1.2 \LaTeX	1
1.2 \LaTeX 基础	2
1.2.1 作者、图书设计者和排版者	2
1.2.2 版面设计	3
1.2.3 优势和不足	3
1.3 \LaTeX 源文件	4
1.3.1 空白距离	5
1.3.2 特殊字符	5
1.3.3 \LaTeX 命令	5
1.3.4 注释	6
1.4 源文件的结构	7
1.5 一个典型的命令行过程	8
1.6 文档布局	9
1.6.1 文档类	9
1.6.2 宏包	11
1.7 各类 \LaTeX 文件	11
1.7.1 页面式样	13
1.8 大型文档	13
2 文档排版	15
2.1 文档和语言的结构	15

2.2	断行和分页	17
2.2.1	段落整理	17
2.2.2	断字	18
2.3	学习字符串排版	19
2.4	特殊字符和符号	19
2.4.1	引号	19
2.4.2	破折号和连字号	20
2.4.3	波浪号 (~)	20
2.4.4	度的符号 (°)	20
2.4.5	省略号 (...)	20
2.4.6	连字	21
2.4.7	注音符号和特殊字符	21
2.5	支持使用国际语言	21
2.5.1	对德语的支持	24
2.6	单词的间隔	24
2.7	标题, 章和节	25
2.8	交叉引用	27
2.9	脚注	27
2.10	强调	28
2.11	环境	28
2.11.1	Itemize, Enumerate, and Description	29
2.11.2	Flushleft, Flushright, and Center	29
2.11.3	Quote, Quotation, and Verse	30
2.11.4	逐字打印	30
2.11.5	表格	31
2.12	浮动体	33
2.13	保护脆弱命令	36
3	数学公式	37
3.1	基本知识	37
3.2	数学模式中的分组	39
3.3	建立数学公式模块	39
3.4	数学空格	43
3.5	垂直对齐	44
3.6	幻影	46
3.7	数学字体大小	46

3.8	定理、定义 ...	47
3.9	粗体符号	48
3.10	数学符号表	50
4	特殊功能	57
4.1	包含 EPS 图形	57
4.2	参考文献	59
4.3	索引	60
4.4	定制页眉和页脚	61
4.5	Verbatim 宏包	61
4.6	下载并安装 L ^A T _E X 宏包	62
5	定制 L^AT_EX	64
5.1	建立新的命令、环境和包	64
5.1.1	建立新的命令	65
5.1.2	建立新的环境	66
5.1.3	建立你自己的宏包	66
5.2	字体和尺寸	67
5.2.1	字体变换命令	67
5.2.2	Danger, Will Robinson, Danger	70
5.2.3	建议	70
5.3	文档对象之间的间隔	70
5.3.1	行距	70
5.3.2	段落格式	71
5.3.3	水平距离	72
5.3.4	垂直距离	72
5.4	页面的布局	73
5.5	关于长度的更详细的信息	75
5.6	盒子	76
5.7	标尺和支撑	78
	参考文献	80
	索引	82

图形列表

1.1	T_EX 系统的组成部分	2
1.2	一个简单的 L^AT_EX 源文件	8
1.3	论文类 L^AT_EX 源文件的例子	8
4.1	fancyhdr 设置实例。	62
5.1	Example Package.	67
5.2	页面布局参数	74

表格列表

1.1	文档类	10
1.2	文档类选项	10
1.3	伴随 L ^A T _E X 发布的一些宏包	12
1.4	The Predefined Page Styles of L ^A T _E X.	13
2.1	注音符号和特殊字符	22
2.2	德语特殊字符	24
2.3	浮动允许放置参量。	34
3.1	数学模式重音符	50
3.2	小写希腊字母	50
3.3	大写希腊字母	50
3.4	二元关系符	51
3.5	二元运算符	51
3.6	大尺寸运算符	52
3.7	箭头	52
3.8	定界符	52
3.9	大尺寸定界符	52
3.10	其它符号	53
3.11	非数学符号	53
3.12	AMS 定界符	53
3.13	AMS 希腊和希伯来字母	53
3.14	AMS 二元关系符	54
3.15	AMS 箭头	54
3.16	AMS 二元否定关系符和箭头	55
3.17	AMS 二元运算符	55
3.18	AMS 其它符号	56
3.19	数学字母	56

4.1	graphicx 宏包使用的关键词	58
4.2	索引关键词格式示例。	60
5.1	字体	68
5.2	字体尺寸	68
5.3	标准文档类型中字体的绝对大小	69
5.4	数学字体	69
5.5	TEX的长度单位	73

第一章 基本知识

本章的第一部分简要介绍了 $\text{\LaTeX} 2_{\epsilon}$ 的主要思想和发展历史。第二部分集中说明 \LaTeX 文档的基本结构。在阅读完本章之后，你应该已经对 \LaTeX 的工作原理有了大致的了解，这将有助于你在进一步的阅读中不断将新获得的信息融入一个整体的图象之中。

1.1 游戏的名目

1.1.1 \TeX

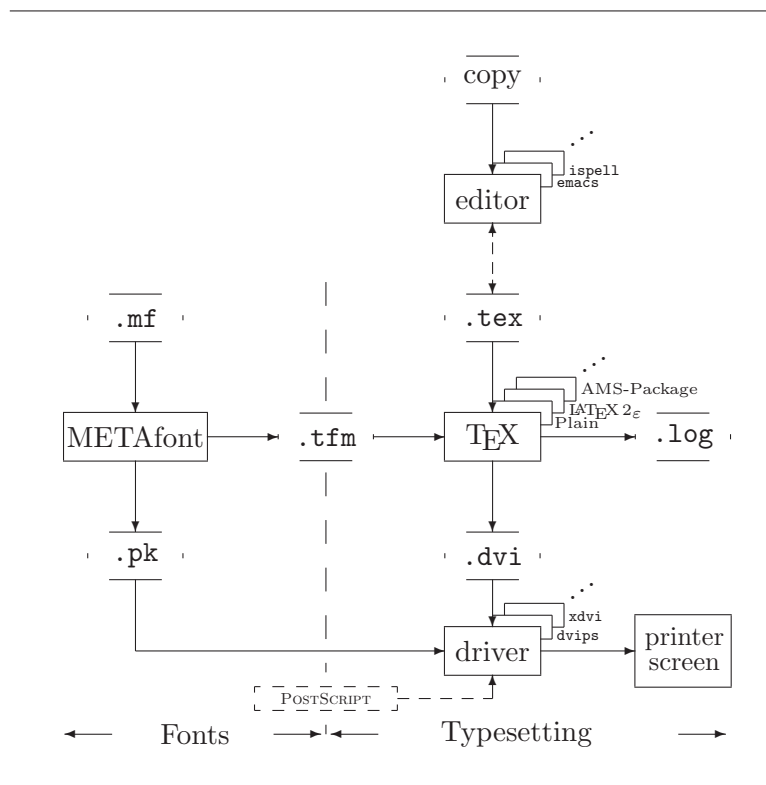
\TeX 是由 Donald E. Knuth[2] 编写的计算机程序，用于文章和数学公式的排版。1977 年 Knuth 开始编写 \TeX 排版系统引擎的时候，是为了探索当时正开始进入出版工业的数字印刷设备的潜力。他特别希望能因此扭转那种排版质量下降的趋势，使自己写的书和文章免受其害。我们现在使用的 \TeX 系统是在 1982 年发布的，1989 年又略作改进，增进了对 8 字节字符和多语言的支持。 \TeX 以具有优异的稳定性，可以在各种不同类型的计算机上运行，以及几乎没有错误而著称。 \TeX 的版本号不断趋近于 π ，现在为 3.14159。

\TeX 的发音为 “Tech”，其中的 “ch” 音和德语 “Ach” 和苏格兰语 “Loch” 中的 “ch” 类似。在 ASCII 文本环境中， \TeX 表示为 \TeX 。

1.1.2 \LaTeX

\LaTeX 是一个宏包，其目的是使作者能够利用一个预先定义好的专业页面设置，从而得以高质量地排版和打印他们的作品。 \LaTeX 最早是由 Leslie Lamport[1] 编写的，并使用 \TeX 作为其排版系统引擎。

1994 年，Frank Mittelbach 领导的 $\text{\LaTeX} 3$ 小组对 \LaTeX 宏包进行了更新，作了一些被期望已久的改进，并且将 $\text{\LaTeX} 2.09$ 发布以来数年间出现的各种不同的补丁重新统一了起来。这个新版本被称作 $\text{\LaTeX} 2_{\epsilon}$ ，以示和旧版本相区别。本书内容就是针对 $\text{\LaTeX} 2_{\epsilon}$ 的。

图 1.1: T_EX 系统的组成部分

L^AT_EX 2_ε 的发音为 “Lay-tech” 或 “Lah-tech”。要在 ASCII 文本环境中表示 L^AT_EX 2_ε，可以用 LaTeX。L^AT_EX 2_ε 的发音为 “Lay-tech two e”，也可写成 LaTeX2e。

图 1.1 显示了 T_EX 和 L^AT_EX 2_ε 是如何共同工作的。该图取自 Kees van der Laan 写的 `wots.tex`。

1.2 L^AT_EX 基础

1.2.1 作者、图书设计者和排版者

出版的第一步是作者将他们的手稿交给出版公司，然后由图书设计者来决定整本书的版面形式（包括栏宽、字体、标题前后的间距……）。图书设计者会把他的排版说明写进手稿里，一起交给排版者，排版者最后根据这些说明完成这本书的排版工作。

一个图书设计者要试图理解作者写作时的意图。他要根据手稿的内容和他自己的职业知识决定章节标题、文献引用、例子、公式、等等。

在 L^AT_EX 环境中，L^AT_EX 代替了图书设计者的角色，T_EX 则是它的排版者。但是 L^AT_EX “仅仅”是一个计算机程序，它需要更多的指导。作者必须在他的作品中提供附加信息，用于描述作品的逻辑结构。这些信息是以“L^AT_EX 命令”的形式写入文章中的。

这种方式和所见即所得（WYSIWYG¹）方式是相当不同的。很多现代文字处理程序都采用所见即所得方式，例如 MS Word 和 Corel Wordperfect。作者可以利用这些应用程序，在向输入文章的同时，通过与计算机互动的方式决定整个文档的排版形式。在整个过程中，作者随时可以在屏幕上看到作品最终打印出来的效果。

在使用 L^AT_EX 的时候，一般来说是不能在输入文章的同时看到最终的输出效果的。但是将文章用 L^AT_EX 处理过之后，是可以在屏幕上预览最终的输出效果的。因此在真正打印文档之前是可以对其进行修正的。

1.2.2 版面设计

排版设计是一门艺术。缺乏经验的作者常常会犯严重的排版格式错误，只因为他们认为图书设计仅仅是个美学问题——“如果排印好的文档从艺术角度来看不错，就说明设计是成功的。”但是既然文档是用来阅读而不是用来挂在画廊里展览的，它的可读性和可理解性就比好看的外观要重要的多。例如：

- 必须为标题选择适当的字体大小和序号，使得读者能够清楚地理解章节结构。
- 行宽既要足够短以避免读者的眼睛疲劳，又要足够长以维持页面美观。

使用所见即所得系统时，作者经常会写出看上去很漂亮，但却缺乏清晰结构或结构不连贯的文档来。为了避免这样的排版格式错误，L^AT_EX 要求作者明确说明其文档的逻辑结构。然后，L^AT_EX 再根据文档结构选择最适当的版面格式。

1.2.3 优势和不足

使用“所见即所得”（WYSIWYG）的人和使用 L^AT_EX 的人在一起时经常谈论的一个话题就是“L^AT_EX 到底比一般的字处理软件强还是弱”。面对这种讨论，你最好是保持低调，因为这样的辩论很容易失去控制。不过，很多时候你也不能逃避……

¹What you see is what you get.

所以，这里给你提供一些武器。L^AT_EX 优于一般的字处理软件之处可以简单归纳为如下几点：

- 提供专业级的排版设计，使你的文档开起来如同印刷好的一样。
- 可以更方便地排版数学公式。
- 用户仅仅需要掌握少数容易理解的，用来说明文档之逻辑结构的命令，而无须对实际的页面设计做胡乱的修补。
- 可以很容易的生成脚注、索引、目录和参考文献等复杂的结构。
- 有大量免费地可添加宏集，协助你完成许多基本的 L^AT_EX 未直接支持的排版任务。例如，支持在文档中插入 POSTSCRIPT 图形的宏集和排版符合各类标准的参考文献的宏集，等等。The L^AT_EX Companion [3] 中给出了大量的宏集说明。
- L^AT_EX 鼓励作者写作具有良好结构的文章，因为 L^AT_EX 就是通过详细地说明文章的结构来进行其排版工作的。
- T_EX 作为 L^AT_EX 的格式化引擎是免费软件，且具有极高的可移植性。因此，他几乎可以在任何硬件平台上运行。

L^AT_EX 当然也有一些不足之处。尽管我可以确定其他人能够列出上百条，我却很难找出一条比较明智的。;-)

- 对于出卖其灵魂的人来说，L^AT_EX 不能很好的工作²。
- 尽管在预先定义好的版面中可以调节一些参数，设计全新的版面还是很困难的，需要耗费大量的时间³。
- L^AT_EX 不适合于排版非结构化的、无序的文档。
- 即使有一个令人振奋的开始，但你也有可能无法完全掌握其精髓。

1.3 L^AT_EX 源文件

L^AT_EX 源文件的格式为普通的 ASCII 文件，你可以使用任何文本编辑器来创建。L^AT_EX 源文件不仅包括你所要排版的文本，还包括 L^AT_EX 所能识别的，如何排版这些文本的命令。

²原文为：“L^AT_EX does not work well for people who have sold their souls ...”，不知译的是否正确。

³传闻这将是未来的 L^AT_EX3 系统的一个重要组成部分。

1.3.1 空白距离

L^AT_EX 将空格和制表符等空白字符视为相同的空白距离 (space)。多个连续的空白字符等同为一个空白字符。在 L^AT_EX 文件中，每行开始的空白字符将被忽略，而单个的回车符被视为一空格。

L^AT_EX 使用空行来结束段落，两行文本中的空行标志上一段落的结束和新段落的开始。如同空格一样，多个空行所起的作用和一个空行的作用是相同的。下面例子中，左边是源文件中的文本，右边则是排版后的输出结果。

```
It does not matter whether you
enter one or several      spaces
after a word.
```

```
An empty line starts a new
paragraph.
```

It does not matter whether you enter one or
several spaces after a word.

An empty line starts a new paragraph.

1.3.2 特殊字符

下面的这些字符是 L^AT_EX 的保留字符，它们或在 L^AT_EX 中有特定的用处，或不一定包含在所有的字库中。如果你直接在文本中使用它们，通常在排版结果中将不会得到这些字符，而且还会导致 L^AT_EX 做一些你不希望发生的事情。

\$ % ^ & _ { } ~ \

当然，这些字符前面加上反斜线，就可以在文本中得到它们。

```
\# \$ \% \^{} \& \_ \{ \} \~{} \\
```

\$ % ^ & _ { } ~

另外一些符号可以由特殊的命令或作为重音命令得到。反斜线 \ 不能够通过在其前添加另外的反斜线来得到，相反的，\\ 是一个用来断行的命令。⁴

1.3.3 L^AT_EX 命令

L^AT_EX 命令 (commands) 是大小写敏感的并有下面两种格式：

⁴试试命令 `\backslash`，它将生成 `\`。

- 以一反斜线 \ 开始，加上只包含字母字符命令名组成。命令名后的空格符、数字或其它非字母字符标志该命令的结束。
- 由一反斜线和一特殊字符组成。

\LaTeX 忽略命令后面的空格。如果你希望在命令后面得到一空格，可以在命令后面加上 {} 和一个空格，或者加上一个特殊的空白距离命令。{} 将阻止 \LaTeX 吞噬掉命令后面的空格。

```
I read that Knuth divides the
people working with \TeX{} into
\TeX{}nicians and \TeX perts.\
Today is \today.
```

```
I read that Knuth divides the people working
with TEX into TEXnicians and TEXperts.
Today is 8th March 2003.
```

许多命令需要一个参数 (parameter) 并用一对大括号 (curly braces) {} 将其括起来置于命令名称的后面。也有一些命令支持用方括号 (square brace) 括起来的可选参数。下面的例子中使用了一些 \LaTeX 命令，并将在以后对他们进行解释。

```
You can \textsl{lean} on me!
```

```
You can lean on me!
```

```
Please, start a new line
right here!\newline
Thank you!
```

```
Please, start a new line right here!
Thank you!
```

1.3.4 注释

当 \LaTeX 在处理源文件时，如果遇到一个百分号字符 %，那么 \LaTeX 将忽略 % 后的该行文本，分行符以及下一行开始的空白字符。这样，我们就可以在源文件中写一些注释，而不会担心他们会出现在最后的排版结果中。

```
This is an % stupid
% Better: instructive <----
example: Supercal%
          ifragilist%
          icexpialidocious
```

```
This is an example: Supercalifragilisticexpi-
alidocious
```

% 也可以用来分割不允许有空格或分行的较长输入文本。

如果需要较长的注释，你可以使用 `verbatim` 宏集所提供的 `comment` 环境。当然，你需要在源文件的导言区里加上命令 `\usepackage{verbatim}`。

```
%This is another
%\begin{comment}
%rather stupid,
%but helpful
%\end{comment}
%example for embedding
%comments in your document.
```

这是另一个

```
\begin{comment}
非常笨拙的，
然而却很有效。
\end{comment}
在你的文档中
使用注释的例子。
```

这是另一个在你的文档中使用注释的例子。

注意这个方法不能使用在像数学环境等一些复杂的环境中。

1.4 源文件的结构

\LaTeX 需要所处理的源文件遵从一定的结构，每个 \LaTeX 文档必须以如下的命令开始：

```
\documentclass{...}
```

这个命令指定了你所写的文档的类别。在此之后，你可以加入控制文档式样的命令，或者使用如下的命令来调入一些宏集，进而为 \LaTeX 系统增添一些新的功能。

```
\usepackage{...}
```

当完成所有的设置⁵后，你可利用如下的命令来开始你的文档：

```
\begin{document}
```

现在你可以输入你所希望排版的文本和所使用的一些 \LaTeX 命令。在文档的最后键入下面的命令来告诉 \LaTeX 你的文档到此结束，从而使 \LaTeX 忽略文档在此命令之后的部分。

⁵`\documentclass` 和 `\begin{document}` 之间的区域称作导言区。

```
\end{document}
```

图 1.2 展示了一个简单的 L^AT_EX 源文件。如图 1.3 所示的源文件则相对复杂一些。

1.5 一个典型的命令行过程

我敢打赌你一定非常渴望尝试第 8 页中短小简洁的 L^AT_EX 源文件。在此，下面的几点可能对你有所帮助。需要注意的是，L^AT_EX 仅仅是一个处理你所提供的源文件的程序，并不具有图形用户界面或精致的按钮。可能有些 L^AT_EX 安装提供了图形用户界面的前端，使得你可以点击按钮来让 L^AT_EX 编译你的源文件，但这并不是 L^AT_EX 真正的工作方式。因此，这里将说明如何在一个基于文本的系统下使得 L^AT_EX 处理你的源文件。当然，前提是你的计算机里已经正确安装了 L^AT_EX。

```
\documentclass{article}
\begin{document}
Small is beautiful.
\end{document}
```

图 1.2: 一个简单的 L^AT_EX 源文件

```
\documentclass[a4paper,11pt]{article}
% define the title
\author{H.~Partl}
\title{Minimalism}
\begin{document}
% generates the title
\maketitle
% insert the table of contents
\tableofcontents
\section{Start}
Well, and here begins my lovely article.
\section{End}
\ldots{} and here it ends.
\end{document}
```

图 1.3: 论文类 L^AT_EX 源文件的例子

1. 创建并编辑你的 \LaTeX 源文件。该文件必须是文本（ASCII）格式。在 Unix 系统下，所有的编辑器都会使用文本格式创建文件。而在 Windows 系统下，你可能需要明确地以 ASCII 或纯文本格式保存你的文件。当选取你的源文件的文件名时，要确保其扩展名为 `.tex`。
2. 运行 \LaTeX 处理你的源文件。如果成功，你将会得到一个扩展名为 `.dvi` 的文件。

```
latex foo.tex
```

3. 现在你可以预览所得到的 DVI 文件。

```
xdvi foo.dvi
```

或将其转换为 PS 文件：

```
dvips -Pcmz foo.dvi -o foo.ps
```

`xdvi` 和 `dvips` 用来处理 `.dvi` 文件的开放源代码工具软件。前者可在 X11 环境下将 DVI 文件显示到屏幕上，后者则将 DVI 文件转换为 PostScript 文件以用来打印。如果你不使用 Unix 操作系统，将会提供其他工具软件来处理 `.dvi` 文件。

1.6 文档布局

1.6.1 文档类

当 \LaTeX 处理源文件时，首先需要知道的是作者所要创建的文档类型。该信息可以通过命令 `\documentclass` 来提供给 \LaTeX 。

`\documentclass[options]{class}`

这里 *class* 指明了所要创建的文档类型。表 1.1 中仅列出了本文所说明的文档类。 \LaTeX 2_ε 发行版本中还包括了像信件、幻灯片等其它文档类。文档类的属性可以通过选项（*options*）来加以调节，不同的选项用逗号隔开。表 1.2 列出了标准文档类最常用的一些选项。

实例：一个 \LaTeX 源文件以下面一行开始

```
\documentclass[11pt,twoside,a4paper]{article}
```

这条命令指定 \LaTeX 使用论文版式，11 磅大小的字体来排版此文档，并且得到适合打印在 A4 纸上的输出结果。

表 1.1: 文档类

article	排版科技期刊、短报告、程序文档、邀请函等。
report	排版多章节的长报告、短篇的书籍、博士论文等。
book	排版书籍。
slides	排版幻灯片。其中使用了较大的 sans serif 字体。也可以考虑使用 FoilT _E X 来得到相同的效果。 ^a

^aCTAN:/tex-archive/macros/latex/contrib/supported/foiltex

表 1.2: 文档类选项

10pt, 11pt, 12pt	设置文档所使用的字体的大小。如果没有声明任何选项，缺省将使用 10pt 字体。
a4paper, letterpaper, ...	定义纸张的大小，缺省的设置为 letterpaper。此外，还可以使用 a5paper, b5paper, executivepaper 和 legalpaper。
fleqn	设置该选项将使数学公式左对齐，而不是中间对齐。
leqno	设置该选项将使数学公式的编号防置于左侧。
titlepage, notitlepage	指定是否在文档标题（document title）后开始一新页。article 文档类缺省不开始新页，而 book 文档类则相反。
onecolumn, twocolumn	指定 L ^A T _E X 以单列（one column）或双列（two column）方式排版文档。
twoside, oneside	指定 L ^A T _E X 排版的文档为双面或单面格式。article 和 report 缺省使用单面格式，而 book 则缺省使用双面格式。需要注意的是该选项仅作用于文档的式样。twoside 选项不会通知你的打印机让以得到双面的打印输出。
openright, openany	此选项决定新的章是仅仅在右边页（奇数页）还是在下一可用页开始。该选项对 article 文档类不起作用，因为该类中并没有定义“章”（Chapter）。report 类中新的一章开始于下一可用页，而 book 类中新的一章总是开始于右边页。

1.6.2 宏包

当你排版文档的时候，你将会发现有很多时候基本的 L^AT_EX 不能够解决你的问题。如果你想插入图形、彩色文本或源代码文件，你需要使用宏包来增强 L^AT_EX 的功能。调入宏包使用如下的命令：

`\usepackage[options]{package}`

这里 *package* 是宏包的名称，*options* 是用来触发宏包中的特殊功能的一组关键词。大部分宏包包含在 L^AT_EX 基本发行版本中（参见表 1.3），另外一些宏包则单独发布。你可以在你所安装的 L^AT_EX 系统中找到更多的有关宏包的信息。*The L^AT_EX Companion* [3] 中提供了关于 L^AT_EX 宏包的主要信息并包括了上百个宏包的详细描述以及如何写作你自己的 L^AT_EX 2_ε 扩展。

1.7 各类 L^AT_EX 文件

当你使用 L^AT_EX 的时候，你可能很快发现你被各种不同扩展名（extensions）的文件搞得昏头昏脑。请注意本表格中所列出的扩展名并不完全，如果你发现其中由你认为比较重要的，最好在其下面划线标明。

.tex L^AT_EX 或 T_EX 源文件。可以用 latex 处理。

.sty L^AT_EX 宏包文件。可使用命令 `\usepackage` 将其加载到你的 L^AT_EX 文件中。

.dtx 文档化 T_EX 文件。这也是 L^AT_EX 宏包发布的主要格式。通过处理一个 .dtx 文件就可以得到该 L^AT_EX 宏包中所包括的宏代码文档。

.ins 为相应的 .dtx 文件的安装文件。如果你在网络上下载了一 L^AT_EX 宏包，你通常会发现会有一个 .dtx 和一个 .ins 文件。使用 L^AT_EX 对 .ins 文件进行处理，可以从 .dtx 文件中提取出宏包。

当你运行 L^AT_EX 处理你的源文件时，会得到下列文件：

.dvi 与设备无关文件。这是 L^AT_EX 编译运行的主要结果。你可以使用 DVI 预览器浏览其内容，或者使用像 dvips 这样的应用程序输出到打印机。

.log 记录了上次编译运行时的详细信息。

- .toc** 存储了所有章节标题。该文件将在下次编译运行时被读入并生成目录。
- .lof** 类似 **.toc** 文件，可生成图形目录。
- .lot** 类似 **.toc** 文件，可生成表格目录。
- .aux** 另一个用来向下次编译运行传递信息的辅助文件。除了其它信息外，**.aux** 文件通常包含交叉引用信息。
- .idx** 如果你的文件中包含有索引， \LaTeX 使用此文件存储所有的索引词条。此文件需要使用 `makeindex` 处理。详见第 4.3 节。

表 1.3: 伴随 \LaTeX 发布的一些宏包

normalfontdoc	排版 \LaTeX 程序文档。 详见 <code>doc.dtx</code> ^a 和 <i>The \LaTeX Companion</i> [3]。
exscale	提供了可缩放版本的数学字体扩展。 详见 <code>ltexscale.dtx</code> 。
normalfontfontenc	声明 \LaTeX 所使用的字体编码 (font encoding)。 详见 <code>ltoutenc.dtx</code> 。
ifthen	提供如下形式的命令： ‘if...then do...otherwise do...’ 详见 <code>ifthen.dtx</code> 和 <i>The \LaTeX Companion</i> [3]。
latexsym	调入此宏包可在文档中使用 \LaTeX 符号。 详见 <code>latexsym.dtx</code> 和 <i>The \LaTeX Companion</i> [3]。
makeidx	提供了生成索引的命令。详见第 4.3 节和 <i>The \LaTeX Companion</i> [3]。
syntonly	仅仅处理文档，但并不对其进行排版。
inputenc	允许声明 ASCII、ISO Latin-1、ISO Latin-2、437/850 IBM 编码页、Apple Macintosh、Next、ANSI-Windows 或自定义的输入编码。详见 <code>inputenc.dtx</code> 。

^a此文件应该已安装在你的系统中，你可以在你有写权限的目录下使用命令 `latex doc.dtx` 得到 `dvi` 文件。对于本表中提及的其它文件同样适用。

.ind 经过处理后的 .idx 文件。可在下次编译运行时加入到你的文档中。

.ilg 运行 makeindex 时生成的记录文件。

1.7.1 页面式样

L^AT_EX 支持三种预定的页眉、页脚 (header/footer) 格式, 称为页面式样 (page styles)。命令

```
\pagestyle{style}
```

中的参数定义了所使用页面式样。表 1.4 列出了预先定义的页面式样。

表 1.4: The Predefined Page Styles of L^AT_EX.

plain	页眉为空, 页脚由居中的页码组成。这是默认的页面式样。
headings	页眉由当前的章节标题和页码组成, 页脚为空。(这是本文档所使用的页面式样)
empty	设置页眉、页脚均为空。

可以使用下面的命令改变当前页的页面式样:

```
\thispagestyle{style}
```

有关如何创建你自己的页眉和页脚的详细说明可参见 *The L^AT_EX Companion* [3] 和第 4.4 节。

1.8 大型文档

当处理大型文档时, 最好将源文件分成几个部分。L^AT_EX 有两条命令来处理这种情况。

```
\include{filename}
```

在文档的正文中使用此命令可将文件名为 *filename.tex* 的内容包括进来。注意 L^AT_EX 在开始处理 *filename.tex* 的内容之前将会开始一新页。

第二个命令可用在文档的导言区，它允许你指导 L^AT_EX 仅仅读入某些 `\included` 文件。

`\includeonly{filename,filename,...}`

这条命令在导言区被 L^AT_EX 读入执行后，在所有的 `\include` 命令中，只有 `\includeonly` 命令参数中列出的文件才会被执行。注意在参数中，文件名和逗号之间不能有空格。

`\include` 命令在新页上排版包括进来的文本，这对于使用 `\includeonly` 命令很有帮助。应为即使一些包括的文件被忽略，分页处也不会变更。如果不想在新页排版包括进来的文本，可使用下面的命令：

`\input{filename}`

该命令只是简单地将指定的文件包括进来，并没有其它限制。

使用 `syntonly` 宏包可以让 L^AT_EX 快速的检查你的文档：L^AT_EX 浏览你的文档，仅仅检查语法和所使用的命令是否正确，不会产生 DVI 输出。在这种模式下，L^AT_EX 运行的非常快，可以节省可观的时间。使用方法非常简单：

```
\usepackage{syntonly}
\syntonly
```

第二章 文档排版

阅读前一章后，就可以了解 \LaTeX 2_ϵ 文档的基本结构了。在这一章我将补充其余的部分，以使你能够生成实用的文档。

2.1 文档和语言的结构

写文档（一些现代 DAAC¹文学除外）的主旨在于向读者传递观点、信息或知识。如果这些观点组织得很好，读者将能更好地理解。如果排版风格反映了内容的逻辑和语义结构，读者就能看见和感觉到文章的这种脉络。

\LaTeX 与其他排版系统的不同在于你只要告诉它文档的逻辑和语义结构，它就按照文档类型和各种格式文件指定的“规则”导出文档的排版风格。

在 \LaTeX （在印刷术）中段落是最重要的文档单位。我们之所以称之为“文档单位”，因为段落是反映一个连贯思想或观点的排版风格形式。在下面的章节中你将学习如何用 `\` 强行断行，通过在源文件中留一空行强行转段。所以，如果开始一个新思想，就另起一段，否则，只应使用断行。如果还犹豫是否应转段，可以把文档想象为观点和思想的传递者。如果旧的思路还在继续，就不应转段。如果同一段中出现了全新的思路，就应该另起一段。

大多数人完全低估恰当设置转段点的重要性。许多人甚至不知道转段点的含义是什么，或者在 \LaTeX 中引进了转段点却浑然不知。如果在文档中使用了公式，后一个错误特别容易发生。看看下面的例子，想想为什么有时在方程前后使用多个空行（转段点），有时却不用。（如果你还不完全理解这些例子中的命令，请阅读本章和下一章，然后重新阅读本节内容。）

% Example 1

¹为标新立异不计代价，译自 Swiss German UVA (Um's Verrecken Anders)。

```
\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 Kirchoff's current law:
\begin{equation}
  \sum_{k=1}^n I_k = 0 \; .
\end{equation}
```

```
Kirchhoff's voltage law can be derived \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
```

下一个较小的文档单位是句子。在英语文档中，在一个句子结尾处句号后的空间大于缩写词句号后的空间。 \LaTeX 试图分辨出你需要那一个。如果 \LaTeX 给错了，你必须告诉 \LaTeX 你到底需要什么。在本章的后面将对此予以解释。

句子也有结构。大多数语言具有非常复杂的标点符号系统，但是在很多语言中（包括德语和英语）只要你记住它表示什么：语言流中的短暂停顿，你就能近乎完美地使用逗号。如果你不肯定在哪里使用逗号，就大声地朗读这个句子，在每个逗号处做一短呼吸。在什么地方如果你感到别扭，就删掉这个逗号；如果在什么地方，你感到必须呼吸（或做了短暂停顿）就插入一个逗号。

最后，通过分成章、节、子节等形式，文档中的段落应该按逻辑在更高的层次进行组织。但是，使用 `\section{The Structure of Text and`

Language} 的排版效果是如此明显，以致如何使用这些高水平的结构几乎是不证自明的。

2.2 断行和分页

2.2.1 段落整理

通常书籍是用等长的行来排版的。为了优化整个段落的内容， \LaTeX 在单词之间插入必要的断行点（`linebreak`）和间隔。如果一行的单词排不下， \LaTeX 也会进行必要的断字。段落如何排版依赖于文档类别。通常，每一段的第一行有缩进，在两段之间没有额外的间隔。更多的信息请参考第 5.3.2 节。

在特殊情形下，有必要命令 \LaTeX 断行

`\ or \newline`

另起一行，而不另起一段。

`*`

在强行断行后，还禁止分页。

`\newpage`

另起一新页。

`\linebreak[n], \nolinebreak[n], \pagebreak[n] and \nopagebreak[n]`

上述命令的效果可以从它们的名称看出来。通过可选参数 n ，作者可以影响这些命令的效果。 n 可以置为 0 和 4 之间的数。如果命令的效果看起来非常差，把 n 取为小于 4 的数，可以让 \LaTeX 选择忽略这个命令。不要这些“break”命令与“new”命令混淆。即使你给出了“break”命令， \LaTeX 仍然试图对齐页面的右边界。如果你真想另起一行，就使用相应的命令。猜猜该是什么命令！

\LaTeX 总是尽可能产生最好的断行效果。如果断行无法达到 \LaTeX 的高标准，就让这一行在段落的右侧溢出。然后在处理输入文件的同时，报告溢出的消息（“overfull hbox”）。这最可能发生在 \LaTeX 找不到合适的地方

断字时候。²你可以使用 `\sloppy` 命令，告诉 \LaTeX 降低一点儿标准。虽然最终的输出结果不是最优的，它通过增加单词之间的间隔，以防止出现过长的行。在这种情况下给出警告（“underfull hbox”）。在大多数情况下得到的结果看起来不会非常好。`\fussy` 命令把 \LaTeX 恢复为缺省状态。

2.2.2 断字

必要时就会出现断字。如果断字算法不能确定正确的断字点，可以使用如下命令告诉 \TeX 如何弥补这个缺憾。

命令

`\hyphenation{word list}`

使列于参量中的单词仅在注有“-”的点断字。命令的参量仅由正常字母构成的单词，或由激活文本中视为正常字母的符号组成。应用于（特殊）语言的已存好，当断字命令出现时，就为激活的语言储存断字可选点。这意味着如果你在文档导言中设置了断字命令，它将影响英文的断字。如果断字命令置于 `\begin{document}` 后面，而且你正使用类似 `babel` 的国际语言支持宏包，那么断字可选点在由 `babel` 激活的语言中就处于活动状态。

下面的例子允许对“hyphenation”和“Hyphenation”进行断字，却根本不允许“FORTRAN”，“Fortran”和“fortran”进行断字。在参量中不允许出现特殊的字符和符号。

例子：

`\hyphenation{FORTRAN Hy-phen-a-tion}`

命令 `\-` 在单词中插入一个自主的断字点。它也就成为这个单词中允许出现的唯一断字点。对于包含特殊字符（注音字符）的单词，这个命令是特别有用的，因为对于包含特殊字符的单词 \LaTeX 不自动断字。

```
I think this is: su\~per\~cal\~%
i\~frag\~i\~lis\~tic\~ex\~pi\~%
al\~i\~do\~cious
```

I think this is: supercalifragilisticexpialidocious

²当发生（盒子溢出）时，虽然 \LaTeX 给出一个警告并显示溢出的那一行，但是不太容易发现溢出的行。如果你在 `\documentclass` 命令中使用选项 `draft`， \LaTeX 就在溢出行的右边标以粗黑线。

命令

`\mbox{text}`

保证把几个单词排在同一行上。在任何情况下，这个命令把它的参量排在一起（同一行上）。

My phone number will change soon.
It will be `\mbox{0116 291 2319}`.

The parameter
`\mbox{\emph{filename}}` should
contain the name of the file.

My phone number will change soon. It will
be 0116 291 2319.

The parameter *filename* should contain the
name of the file.

命令 `\fbox` 和 `\mbox` 类似，此外它还能围绕内容画一个框。

2.3 学习字符串排版

在前面的例子中，你已经看到用来排特殊字符串的非常简单的 \LaTeX 命令。

Command	Example	Description
<code>\today</code>	8th March 2003	Current date in the current language
<code>\TeX</code>	\TeX	The name of your favorite typesetter
<code>\LaTeX</code>	\LaTeX	The name of the Game
<code>\LaTeXe</code>	$\text{\LaTeX 2}_{\epsilon}$	The current incarnation of \LaTeX

2.4 特殊字符和符号

2.4.1 引号

你不要像在打字机上那样，把 " 用作引号。在印刷中有专门的左引号和右引号。在 \LaTeX 中，用两个 ‘ 产生左引号，用两个 ’ 产生右引号。一个 ‘ 和一个 ’ 产生一个单引号。

“Please press the ‘x’ key.”

“Please press the ‘x’ key.”

2.4.2 破折号和连字号

\LaTeX 中有四种短划标点符号。连续用不同数目的短划，可以得到其中的三种。第四个实际不是标点符号，它是数学中的减号：

```
daughter-in-law, X-rated\\
pages 13--67\\
yes---or no? \\
$0$, $1$ and $-1$
```

```
daughter-in-law, X-rated
pages 13-67
yes—or no?
0, 1 and -1
```

这些短划线的名称是：‘-’ 连字号，‘—’ 短破折号，‘—’ 长破折号和 ‘-’ 减号。

2.4.3 波浪号 (~)

波浪号经常和网址用在一起。它在 \LaTeX 中，可用 \textbackslash~ 产生，但其结果： \textbackslash~ 却不是你真真正想要的。试一下这个：

```
http://www.rich.edu/\~{}bush \\
http://www.clever.edu/\$sim$demo
```

```
http://www.rich.edu/~bush
http://www.clever.edu/~demo
```

2.4.4 度的符号 (°)

在 \LaTeX 中如何排度的符号？

```
Its $-30\text{\textbackslash}\text{\textasciicircum}\text{\textbackslash}\text{\mathrm{C}}$,
I will soon start to
super-conduct.
```

```
Its -30 °C, I will soon start to super-conduct.
```

2.4.5 省略号 (...)

在打字机上，逗号或句号占据的空间和其他字母相等。在书籍印刷中，这些字符仅占据一点儿空间，并且与前一个字母贴得非常紧。所以不能只键入三个点来输出‘省略号’，因为间隔划分得不对。有一个专门的命令输出省略号。它称为

```
\ldots
```

Not like this ... but like this:\\
New York, Tokyo, Budapest, \ldots

Not like this ... but like this:
New York, Tokyo, Budapest, ...

2.4.6 连字

一些字母组合不是简单键入一个个字母得到得的，而实际上用到了一些特殊符号。

效果应为ff fi fl ffi... 而不是 ff fi fl ffi ...

这就是所谓的连字，在两个字母之间插入一个 `\mbox{}`，可以禁止连字。对于由两个词构成的单词，这可能是必要的。

不是%Not
shelfful\\
而是%but
shelf\mbox{ }ful

不是shelfful
而是shelfful

2.4.7 注音符号和特殊字符

\LaTeX 支持来自许多语言中的注音符号和特殊字符。表 2.1 就字母 o 列出了所有的注音符号。对于其他字母也自然有效。

在字母 i 和 j 上标一个注音符号，它的点儿必须去掉。这个可由 `\i` 和 `\j` 做到。

H\^otel, na\"i ve, \'el\'eve,\\
sm\o rrebr\o d, !'Se\~norita!,\\
Sch\"onbrunner Schlo\ss{}
Stra\ss e

Hôtel, naïve, élève,
smørrebrød, ¡Señorita!,
Schönbrunner Schloß Straße

2.5 支持使用国际语言

如果你需要用英语以外的语言书写文件， \LaTeX 有两个地方必须适当配置：

1. 所有自动生成的字符串³必须适用于新语言。对于许多种语言，这个任务可由 Johannes Braams 编的宏包 `babel` 完成。

³表目录, 图目录, ...

2. 对于一种新语言， \LaTeX 需要知道它的断字规则。将断字规则输入 \LaTeX 有些难度。这是说为不同断字模式重建格式文件是行得通的。对此 *Local Guide* [4] 给了更多的信息。

如果你的系统已经配置好了，你可以通过在命令 `\documentclass` 后添加命令

`\usepackage[language]{babel}`

来激活宏包 `babel`。系统支持的语言也应列于 *Local Guide* [4] 中。对于选定的语言，宏包 `babel` 将自动激活适当的断字规则。如果 \LaTeX 的格式文件不支持在所选择的语言中断字，除了失去断字功能，宏包 `babel` 仍起作用，这对于排版效果有很大的负面影响。

对于很多种语言，宏包 `babel` 也提供专门的新命令来简化特殊字符的输入。例如 German，包含很多类似的字符（`äöü`）。利用 `babel`，你能用 `"o` 而不是 `\"o` 来输入 `ö`。

一些计算机系统允许通过键盘直接输入特殊的字符。 \LaTeX 能够处理这些字符。从 1994 年 12 月发布的 $\text{\LaTeX} 2_{\epsilon}$ 开始， $\text{\LaTeX} 2_{\epsilon}$ 的基本版本包括了对几种输入编码的支持。请检查宏包 `inputenc`：

`\usepackage[encoding]{inputenc}`

当使用这宏包时，应该考虑其他人在其计算机上或许不能显示你的输入文件，因为他们使用不同的编码。例如，德语 `ä` 在微机上的编码为 132，

表 2.1: 注音符号和特殊字符.

ò	\‘o	ó	\’o	ô	\^o	õ	\~o
ō	\=o	ò	\.o	ö	\"o	ç	\c c
ö	\u o	ö	\v o	ő	\H o	q	\c o
q	\d o	q	\b o	öo	\t oo		
œ	\oe	Œ	\OE	æ	\ae	Æ	\AE
å	\aa	Å	\AA				
ø	\o	Ø	\O	ı	\l	Ł	\L
ı	\i	ı	\j	ı	!‘	ı	?‘

但是在一些使用 ISO-LATIN 1 的 Unix 系统上，它的编码为 228。所以应小心使用这个功能。根据系统的类型，可能有下列编码。

Operating system	encoding
Mac	applemac
Unix	latin1
Windows	ansinew
OS/2	cp850

字体编码是不同的问题。它定义于一种 $\text{T}_{\text{E}}\text{X}$ -字体的每个字母的所在位置。原始的计算机现代 $\text{T}_{\text{E}}\text{X}$ 字体仅包含旧的 7 - 字节 ASCII 字符集中的 128 个字符。当需要注音字母的时候，由 $\text{T}_{\text{E}}\text{X}$ 利用正常字母和注音符号生成。这个方法停止对含有注音字母的单词进行断字，同时保证输出效果看起来很完美。

幸运的是大多数 $\text{T}_{\text{E}}\text{X}$ 的现代版本带有一套 EC 字体。这些字体看起来像现代计算机字体，但是包含用于欧洲语言大多数注音字母的特殊字符。使用这些字体，可以改进非英语文档中的断字。EC 字体由包含于文档导言中 fontenc 宏包来激活。

```
\usepackage[T1]{fontenc}
```

2.5.1 对德语的支持

用 L^AT_EX 生成德语文档的一些提示。可以用命令

```
\usepackage[german]{babel}
```

加载对德语的支持。

如果 L^AT_EX 系统已经做了相应的配置，就能保证德语单词的断字。同时它把自动文本替换为德语，即把“章”变为“Kapitel”。而且提供了一套新命令使得输入德语文件更加迅速。从表 2.2 中能获得灵感。

表 2.2: 德语特殊字符.

"a	ä	"s	ß
"‘	”	"’	“
"<	«	">	»
\dq	”		

2.6 单词的间隔

为了使输出的右边界对齐，L^AT_EX 在单词间插入不等的间隔。在句子的末尾插入的空间稍多一些，因为这使得文本更具可读性。L^AT_EX 假定句子以句号、问号或惊叹号结尾。如果句号紧跟一个大写字母，它就不视为句子的结尾。因为一般在有缩写地方，才出现句号紧跟大写字母的情况。

作者必须详细说明，这些假设中的任何一个例外。空格前的反斜线符号产生一个不能伸长的空格。波浪字符 ‘~’ 也产生一个不能伸长的空格，并且禁止断行。句号前的命令 \@ 说明这个句号是句子的末尾，即使它紧跟一个大写字母。

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

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

命令

`\frenchspacing`

能禁止在句号后插入额外的空间，它告诉 \LaTeX 在句号后不要插入比正常字母更多的空间。除了参考文献，这在非英语语言中非常普遍。如果使用了 `\frenchspacing`，命令 `\@` 就不必要了。

2.7 标题，章和节

为便于读者理解，应该把文档划分为章，节和子节。 \LaTeX 用特殊的命令支持这个工作，这些命令把节的标题作为参量。你的任务是按正确次序使用它们。

对 `article` 风格的文档，有下列分节命令：

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

对 `report` 和 `book` 风格的文档，还有其他两个分节命令：

```
\part{...}              \chapter{...}
```

因为 `article` 风格的文档不划分为章，所以很容易把它作为一章插入书籍文档中。节之间的间隔，节的序号和标题的字号由 \LaTeX 自动设置。

分节的两个命令有些特殊性：

- 命令 `\part` 不影响章的序号。
- 命令 `\appendix` 不带参量，只把章的序号改用于字母标记。⁴

\LaTeX 在文档编译的最后一个循环中，提取节的标题和页码以生成目录。命令

`\tableofcontents`

在其出现的位置插入目录。为了得到正确的目录内容，一个新文档必须编译两次。有时还要编译三次。届时 \LaTeX 会通知你。

⁴对 `article` 风格的文档改变节的序号。

上面列出的分节命令也以“带星”的形式出现。“带星”的命令通过在命令名称后加 * 来实现。它们生成的节标题既不出现于目录，也不带序号。例如，命令 `\section{Help}` 的“带星”形式为 `\section*{Help}`。

目录出现的标题，一般与输入的文本完全一致。有时这是不可能的，因为标题太长排不进目录。在这种情况下，目录的条目可由真实标题前的可选参量确定。

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

整篇文档的标题由命令

```
\maketitle
```

产生。标题的内容必须在调用 `\maketitle` 以前，由命令

```
\title{...}, \author{...} 和可选的 \date{...}
```

定义。在命令 `\author` 中，可以输入几个用 `\and` 命令分开的名字。

在第 8 页的图 1.3 中，能找到有关上述命令的一个例子。

除了上面解释的分节命令， \LaTeX 2_ϵ 引进了其他三个命令用于 book 风格的文档。它们对划分出版物有用，也能如愿改变章的标题和页码：

`\frontmatter` 应接着命令 `\begin{document}` 使用。它把页码更换为罗马数字。对于正文前的内容普遍使用带星的命令（例如，`\chapter*{Preface}`），以阻止 \LaTeX 对它们排序。

`\mainmatter` 应出现在书的第一章紧前面。它打开阿拉伯页码计数器，并对页码从新计数。

`\appendix` 标志书中附录材料的开始。该命令后的各章序号改用字母标记。

`\backmatter` 应该插入与书中最后一部分内容的紧前面，如参考文献和索引。在标准文档类型中，它对页面没有什么效果。

2.8 交叉引用

在书籍、报告和论文中，需要对图、表和文本的特殊段落进行交叉引用。L^AT_EX 提供了如下交叉引用命令

```
\label{marker}, \ref{marker} and \pageref{marker}
```

其中 *marker* 是使用者选择的标识符。如果在节、子节、图、表或定理后面输入 `\label` 命令，L^AT_EX 把 `\ref` 替换为相应的序号。`\pageref` 命令排印 `\label` 输入处的页码。⁵ 和章节标题一样，使用的序号是前面编译所产生。

A reference to this subsection
`\label{sec:this}` looks like:
“see section~`\ref{sec:this}` on
page~`\pageref{sec:this}`.”

A reference to this subsection looks like: “see
section 12 on page 27.”

2.9 脚注

利用命令

```
\footnote{footnote text}
```

把脚注内容排印于当前页的页脚位置。脚注命令总是置于 (put)⁶ 其指向的单词或句子的后面。脚注是一个句子或句子的一部分，所以应用逗号或句号结尾。⁷

Footnotes\footnote{This is
a footnote.} are often used
by people using \LaTeX.

Footnotes^a are often used by people using
L^AT_EX.

^aThis is a footnote.

⁵注意这些命令对它们指向什么并没有意识。命令 `\label` 只是保存了上一次自动产生的序号。

⁶“put”是最常使用的英语单词之一。

⁷注意，脚注把读者的注意力从文档的正文引开。我们是好奇的动物，每个人都会阅读脚注。所以为什么不把你所说的所有东西都写入正文中。⁸

⁸路标不必走向它指向的地方 :-).

2.10 强调

如果文本是用打字机键入的，用下划线来强调重要的单词。

```
\underline{text}
```

但是在印刷的书中，用一种斜体字体排印要强调的单词。L^AT_EX 提供命令

```
\emph{text}
```

来强调文本。这些命令对其参量的实际作用效果依赖于它的上下文：

```
\emph{If you use
  emphasizing inside a piece
of emphasized text, then
\LaTeX{} uses the
\emph{normal} font for
emphasizing.}
```

If you use emphasizing inside a piece of emphasized text, then L^AT_EX uses the normal font for emphasizing.

请注意要求 L^AT_EX 强调什么和要求它使用不同字体的不同效果：

```
\textit{You can also
  \emph{emphasize} text if
it is set in italics,}
\textsf{in a
  \emph{sans-serif} font,}
\texttt{or in
  \emph{typewriter} style.}
```

You can also emphasize text if it is set in italics, in a sans-serif font, or in typewriter style.

2.11 环境

```
\begin{environment} text \end{environment}
```

其中 *environment* 是环境的名称。只要保持调用顺序，环境可以嵌套几次。

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

在下面的章节中，对所有重要的环境都做了解释。

2.11.1 Itemize, Enumerate, and Description

`itemize` 环境用于简单的列表，`enumerate` 环境用于带序号的列表，`description` 环境用于带描述的列表。

```
\flushleft
\begin{enumerate}
\item You can mix the list
environments to your taste:
\begin{itemize}
\item But it might start to
look silly.
\item[-] With a dash.
\end{itemize}
\item Therefore remember:
\begin{description}
\item[Stupid] things will not
become smart because they are
in a list.
\item[Smart] things, though, can be
presented beautifully in a list.
\end{description}
\end{enumerate}
```

1. You can mix 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.

2.11.2 Flushleft, Flushright, and Center

`flushleft` 和 `flushright` 环境分别产生靠左排列和靠右排列的段落。`center` 环境产生居中的文本。如果你不输入命令 `\` 指定断行点， \LaTeX 将自行决定。

```
\begin{flushleft}
This text is\\ left-aligned.
\LaTeX{} is not trying to make
each line the same length.
\end{flushleft}
```

This text is
left-aligned. \LaTeX is not trying to make
each line the same length.

```
\begin{flushright}
This text is right-\\aligned.
\LaTeX{} is not trying to make
each line the same length.
\end{flushright}
```

This text is right-
aligned. \LaTeX is not trying to make each
line the same length.

```
\begin{center}
At the centre\\of the earth
\end{center}
```

At the centre
of the earth

2.11.3 Quote, Quotation, and Verse

`quote` 环境对重要断语和例子的引用很重要。

```
A typographical rule of thumb
for the line length is:
\begin{quote}
On average, no line should
be longer than 66 characters.
\end{quote}
This is why \LaTeX{} pages have
such large borders by default and
also why multicolumn print is
used in newspapers.
```

A typographical rule of thumb for the line length is:

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

This is why \LaTeX pages have such large borders by default and also why multicolumn print is used in newspapers.

有两个类似的环境：`quotation` 和 `verse` 环境。`quotation` 环境用于超过几段的较长引用，因为它对段落进行缩进。`verse` 环境用于诗歌，在诗歌中断行很重要。在一行的末尾用 `\\` 断行，在每一段后留一空行。

```
I know only one English poem by
heart. It is about Humpty Dumpty.
\begin{flushleft}
\begin{verse}
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.
\end{verse}
\end{flushleft}
```

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.

2.11.4 逐字打印

位于 `\begin{verbatim}` 和 `\end{verbatim}` 之间的文本将直接打印，包括所有的断行和空白，就像在打字机上键入一样，不执行任何 \LaTeX 命令。

在一个段落中，类似的功能可由

```
\verb+text+
```

完成。+ 仅是分隔符的一个例子。除了 * 或空格，可以使用任意一个字符。这个小册子中的许多例子是用这个命令排印的。

The \verb|\ldots| command \ldots

```
\begin{verbatim}
10 PRINT "HELLO WORLD ";
20 GOTO 10
\end{verbatim}
```

```
The \ldots command ...
```

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

```
\begin{verbatim*}
the starred version of
the      verbatim
environment emphasizes
the spaces in the text
\end{verbatim*}
```

```
the_starred_version_of
the_verbatim
environment_emphasizes
the_spaces_in_the_text
```

带星的命令 \verb 能以类似的方式使用：

```
\verb*|like this :-)|
```

```
like_this_:-)|
```

verbatim 环境和 \verb 命令不能在其他命令的参数中使用。

2.11.5 表格

tabular 环境能用来排印带有水平和铅直表线的漂亮表格。L^AT_EX 自动确定每一列的宽度。

命令

```
\begin{tabular}{table spec}
```

的参量 *table spec* 定义了表格的式样。用一个 l 产生左对齐的列，用一个 r 产生右对齐的列，用一个 c 产生居中的列；用 p{宽度值 *width*} 产生相应宽度、包含自动断行文本的列；| 产生铅直表线。

在 `tabular` 环境中，用 `&` 跳入下一列，用 `\\` 开始新的一行，用 `\hline` 插入水平表线。用 `\cline{j-i}` 可添加部分表线，其中 `j` 和 `i` 分别表示表线的起始列和终止列的序号。

```
\begin{tabular}{|r|l|}
\hline
7C0 & hexadecimal \\
3700 & octal \\ \cline{2-2}
11111000000 & binary \\
\hline \hline
1984 & decimal \\
\hline
\end{tabular}
```

7C0	hexadecimal
3700	octal
11111000000	binary
1984	decimal

```
\begin{tabular}{|p{4.7cm}|}
\hline
Welcome to Boxy's paragraph.
We sincerely hope you'll
all enjoy the show.\\
\hline
\end{tabular}
```

Welcome to Boxy's paragraph. We sincerely hope you'll all enjoy the show.
--

表列分隔符可由 `@{...}` 构造。这个命令去掉表列之间的间隔，代之为两个花括号间的任何输入。一个用途在于下面要解释的十进制数对齐问题。另一个可能应用在于用 `@{}` 压缩表列右端空间。

```
\begin{tabular}{@{} l @{}}
\hline
no leading space\\
\hline
\end{tabular}
```

no leading space

```
\begin{tabular}{l}
\hline
leading space left and right\\
\hline
\end{tabular}
```

leading space left and right

由于没有内建的方法使十进制数按小数点对齐，⁹我们可以使用两列“作弊”到达这个目的：整数右对齐，小数左对齐。`\begin{tabular}` 行

⁹如果系统安装了 ‘tools’ 包，请看一下宏包 `dcolumn`。

中的命令 `@{.}` 用一个 “.” 取代了表列间的正常间隔，从而给出了按小数点表列对齐的效果。不要忘记用表列分隔符 (`&`) 取代十进制数的小数点！使用命令 `\multicolumn` 可在数值“表列”上放置一个表列标签。

```
\begin{tabular}{c r @{.} l}
Pi expression      &
\multicolumn{2}{c}{Value} \\
\hline
 $\pi$             & 3.1416 & \\
 $\pi^{\pi}$         & 36.46  & \\
 $(\pi^{\pi})^{\pi}$  & 80662.7 & \\
\end{tabular}
```

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

```
\begin{tabular}{|c|c|}
\hline
\multicolumn{2}{|c|}{Ene} \\
\hline
Mene & Muh! \\
\hline
\end{tabular}
```

Ene	
Mene	Muh!

用表格环境排印的材料总是呆在同一页上。如果要排印一个长表格，可以看一下 `supertabular` 和 `longtabular` 环境。

2.12 浮动体

今天大多数出版物含有许多图片和表格。由于不能把它们分割在不同的页面上，所以需要专门的处理。如果一个图片或一个表格太大在当前页面排不下，一个解决办法就是每次新开一页。这个方法在页面上留下部分空白，效果看起来很差。

对于在当前排不下的任何一个图片或表格，其解决办法是把它们“浮动”到下一页，与此同时当前页面用正文文本填充。 \LaTeX 提供了两个浮动体环境；一个用于图片，一个用于表格。要充分发挥这两个环境的优越性，应该大致了解 \LaTeX 处理浮动体的内在原理。但是浮动可能成为令人沮丧的主要原因，因为 \LaTeX 总不把浮动体放在你想要的地方。

首先看一下供浮动使用的 \LaTeX 命令：

包围于环境 figure 或环境 table 中的任何材料都将被视为浮动内容。
两个浮动环境都支持可选参数

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

称为放置说明符，它由浮动允许放置参量写成的字符串组成。请见表 2.3。
注意：0pt 和 1.05em 都是 T_EX 长度单位。第 73 页中的表 5.5 含有更多的有关信息。

一个表格可以由如下命令，例如

`\begin{table}[!hbp]`

开始，放置说明符 [!hbp] 允许 L^AT_EX 把表格就放当前页，或放在某页的底部 (b)，或放在一个专门的浮动页上 (p)，严格按照放置说明符放置即使看起来不好 (!)如果没有给定放置说明符，缺省值为 [tbp]。

L^AT_EX 将按照作者提供的放置说明符，安排它遇到的每一个浮动体。如果浮动体在当前页不能安排，就把它寄存在图片或表格等待序列中¹⁰。当新的一页开始的时候，L^AT_EX 首先检查是否可能用等待序列中的浮动体填充一个专门的“浮动”页面。如果这不可能，就像对待刚在文本中出现的浮动体一样，处理等待序列中的第一个浮动体：L^AT_EX 从新尝试按照其相应的放置说明符（除了不再可能的‘h’）来处理它。文本中出现的任何一个新浮动体寄存在相应的等待序列中。对于每一种浮动体，L^AT_EX 保持它们出现的顺序。这就说明了为什么一个不能安排的图片把所有后来的图片都推到文档末尾的原因。所以：

如果 L^AT_EX 没有像你期望的那样安排浮动体，那么经常是仅有一个浮动体堵塞了两个中的某一个等待序列。

¹⁰它们是“先来先走”序列！

表 2.3: 浮动允许放置参量。

放置说明符	浮动体允许放置位置...
h	浮动体就放在当前页面上。这主要用于小浮动体。
t	放在页面顶部
b	放在页面底部
p	放在一专门页面，仅含一个浮动体。
!	忽略阻止浮动体放置的大多数内部参数 ^a 。

^a例如一页中允许的浮动体的最大个数。

仅给定单个放置说明符是允许的，但这会引起问题。如果在指定的位置安排不了，它就会成为障碍，堵住后续的浮动体。不要单独使用参量 [h]，在 L^AT_EX 最近的版本中，它的效果太差了以至于被 [ht] 自动替换。

虽然对浮动体问题已经作了些说明，对 table 和 figure 环境还有些内容要交代。利用

```
\caption{caption text}
```

命令，可以给浮动体定义一个标题。序号和字符串“图”或“表”将由 L^AT_EX 自动添加。

两个命令

```
\listoffigures and \listoftables
```

用起来和 \tableofcontents 命令类似，分别排印一个图目录和表目录。在这些目录中，所有的标题都将重复。如果打算使用长标题，就必须准备一个能放进目录的，较短版本的标题。即在 \caption 命令后面的括号内输入较短版本的标题。

```
\caption[Short]{LLLLLoooooonnnnnnggggg}
```

利用 \label 和 \ref，在文本中可以为浮动体创建交叉引用。

下面的例子画一个方形，并将它插入文档。如果想在完成的文档中为你打算粘上的图片保留空间，你可以利用这个例子。

```
Figure~\ref{white} is an example of Pop-Art.
\begin{figure}[!hbp]
\makebox[\textwidth]{\framebox[5cm]{\rule{0pt}{5cm}}}
\caption{Five by Five in Centimetres.} \label{white}
\end{figure}
```

在上面的例子中，为了把图片就放在当前页面 (h)¹¹，L^AT_EX 尝试得真辛苦 (!) 如果这不可能，它将试图图片安排在页面的底部 (b)。如果不能将图片安排在当前页面，它将决定是否可能开一个新页面以安排这张图片，或来自表格等待序列中的一些表格。如果没有足够的材料填充一个专门浮动页面，L^AT_EX 就开一个新页，像对文本中刚出现的图片一样，再一次处理这个图片。

¹¹假设图片等待序列已空。

在一定情况下，可能需要使用命令

`\clearpage` or even the `\cleardoublepage`

command. 它命令 \LaTeX 立即安排等待序列中所有剩下的浮动体，并且开一新页。命令 `\cleardoublepage` 甚至会命令 \LaTeX 新开奇数页面。

在本书的后面，将介绍如何在 \LaTeX 2_ϵ 文档中插入 PostScript 图形。

2.13 保护脆弱命令

作为命令（如 `\caption` 或 `\section`）参量的文本，可能在文档中出现多次（例如，在文档的目录和正文中）。当用于类似 `\section` 的命令参量时，一些命令会失效。它们被称为脆弱命令。`\footnote` 或 `\phantom` 是脆弱命令的例子。这些脆弱命令需要的，正是保护。把 `\protect` 命令放在它们前面，就能保护它们。

`\protect` 仅仅保护紧跟其右侧的命令，连它的参量也不惠及。在大多数情形下，过多的 `\protect` 并不碍事。

```
\section{I am considerate
      \protect\footnote{and protect my footnotes}}
```

第三章 数学公式

准备好了！接下来我们就要领略到 $\text{T}_{\text{E}}\text{X}$ 强大之所在：数学符号和公式的排版。本章所介绍的内容基本可以满足大部分人的需要。即便如此，也只是对此项功能的概括性的描述。如果不能在此章中找到你所需要的排版数学公式的方法，那么你可以在 $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{T}_{\text{E}}\text{X}$ ¹ 或其它宏集中找到答案。

3.1 基本知识

$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 使用一种特殊的模式来排版数学符号和公式（`mathematics`）。段落中的数学表达式应该置于 `\(` 和 `\)`，`\$` 和 `\$` 或者 `\begin{math}` 和 `\end{math}` 之间。

Add `\$a\$` squared and `\$b\$` squared
to get `\$c\$` squared. Or, using
a more mathematical approach:
`\$c^{\{2\}}=a^{\{2\}}+b^{\{2\}}\$`

Add a squared and b squared to get c squared.
Or, using a more mathematical approach:
$$c^2 = a^2 + b^2$$

`\TeX{\}` is pronounced as
`\$ \tau \epsilon \chi \$`. `\[6pt]`
`100~m\$^{\{3\}}\$` of water `\[6pt]`
This comes from my `\$ \heartsuit \$`

$\text{T}_{\text{E}}\text{X}$ is pronounced as $\tau\epsilon\chi$.
 100 m^3 of water
This comes from my ♥

对于较大的数学式子，最好的方法是使用显示式样来排版：将它们放置于 `\[` 和 `\]` 或 `\begin{displaymath}` 和 `\end{displaymath}` 之间。这样排版出的公式是没有编号的。如果你希望 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 对其添加编号的话，可以使用 `equation` 环境来达到这一目的。

¹CTAN:/tex-archive/macros/latex/required/amslatex

Add a squared and b squared
to get c squared. Or, using
a more mathematical approach:
$$c^2 = a^2 + b^2$$

And just one more line.

Add a squared and b squared to get c squared.
Or, using a more mathematical approach:

$$c^2 = a^2 + b^2$$

And just one more line.

利用 `\label` 和 `\ref` 对公式加以引用。

$$\epsilon > 0$$

From [\(3.1\)](#), we gather
$$\dots$$

$$\epsilon > 0 \quad (3.1)$$

From [\(3.1\)](#), we gather ...

对比一下用不同式样排版所得到的结果：

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$$

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$$

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$$

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$$

数学模式和文本模式有很多不同之处。例如在数学模式中：

1. 空格和分行都将被忽略。所有的空格或是由数学表达式逻辑的衍生，或是由特殊的命令如 `\,`，`\quad` 或 `\qquad` 来得到。
2. 不允许有空行，每个公式中只能有一个段落。
3. 每个字符都将被看作是一个变量名并以此来排版。如果你希望在公式中出现普通的文本（使用正体字并可以有空格），那么你必须使用命令 `\text{rm}{...}` 来输入这些文本。

$$\forall x \in \mathbf{R} : x^2 \geq 0$$

$$\forall x \in \mathbf{R} : x^2 \geq 0 \quad (3.2)$$

```
\begin{equation}
x^2 \geq 0 \qquad
\text{for all } x \in \mathbf{R}
\end{equation}
```

$$x^2 \geq 0 \quad \text{for all } x \in \mathbf{R} \quad (3.3)$$

数学家们通常对使用什么样的符号非常挑剔：习惯上使用“空心粗体”（blackboard bold）来表示实数集合。这种字体可用 `amsfonts` 或 `amssymb` 宏包中的命令 `\mathbb` 来得到。上面的例子变为：

```
\begin{displaymath}
x^2 \geq 0 \qquad
\text{for all } x \in \mathbb{R}
\end{displaymath}
```

$$x^2 \geq 0 \quad \text{for all } x \in \mathbb{R}$$

3.2 数学模式中的分组

数学模式中的命令仅对其后面第一个字符起作用。所以，如果你希望某一命令作用于多个字符的话，那么你就必须将它们放置于括号中：`\{...\}`。

```
\begin{equation}
a^{x+y} \neq a^{x+y}
\end{equation}
```

$$a^x + y \neq a^{x+y} \quad (3.4)$$

3.3 建立数学公式模块

在这一节中将介绍排版数学符号和公式的最重要的命令。详细的命令列表可参考第 50 页第 3.10 节。

小写希腊字母（Lowercase Greek letters）的输入命令为：`\alpha`, `\beta`, `\gamma`, ...，相应地，大写形式的输入命令为：`\Gamma`, `\Delta`, ...。²

```
\lambda, \xi, \pi, \mu, \Phi, \Omega
```

$$\lambda, \xi, \pi, \mu, \Phi, \Omega$$

指数和下标可用 `^` 和 `_` 后加相应的字符来实现。

²LaTeX 2_ε 没有定义 Alpha 的大写形式，因为它和普通的罗马字体 A 很像。也许新的数学编码完成后会有所变化。

`a_{1} \quad x^{2} \quad`
`$e^{-\alpha t}$ \quad`
`$a^{\{3\}}_{ij}$ \quad`
`e^{x^2} \neq {e^x}^2$`

$$a_1 \quad x^2 \quad e^{-\alpha t} \quad a_{ij}^3$$

$$e^{x^2} \neq e^{x^2}$$

平方根 (square root) 的输入命令为：`\sqrt`， n 次方根相应地为：`\sqrt[n]`。方根符号的大小由 L^AT_EX 自动加以调整。也可用 `\surd` 仅给出符号。

`\sqrt{x} \quad`
`$\sqrt{x^2+\sqrt{y}}$ \quad`
`\quad $\sqrt[3]{2}$ \quad`
`$\surd[x^2+y^2]$`

$$\sqrt{x} \quad \sqrt{x^2 + \sqrt{y}} \quad \sqrt[3]{2}$$

$$\sqrt{x^2 + y^2}$$

命令 `\overline` 和 `\underline` 在表达式的上、下方画出水平线。

`$\overline{m+n}$ \quad`
`$\underline{m+n}$`

$$\overline{m+n} \quad \underline{m+n}$$

命令 `\overbrace` 和 `\underbrace` 在表达式的上、下方给出一水平的大括号。

`$\underbrace{a+b+\cdots+z}_{26}$`

$$\underbrace{a+b+\cdots+z}_{26}$$

数学重音符号如小箭头和~ (tilde) 等的输入命令可参考第 50 页中的表 3.1。可覆盖多个字符的宽重音符号可由 `\widetilde` 和 `\widehat` 等得到。字符 ' 将生成 ' (prime)。

`\begin{displaymath}`
`y=x^2 \quad y'=2x \quad y''=2`
`\end{displaymath}`

$$y = x^2 \quad y' = 2x \quad y'' = 2$$

向量 (Vectors) 通常用上方有小箭头 (arrow symbols) 的变量表示。这可由 `\vec` 得到。另两个命令 `\overrightarrow` 和 `\overleftarrow` 在定义从 A 到 B 的向量时非常有用。

`\begin{displaymath}`
`\vec a \quad \overrightarrow{AB}`
`\end{displaymath}`

$$\vec{a} \quad \overrightarrow{AB}$$

一般情况下，乘法算式中的圆点符可以省略。然而有时为了帮助读者解读复杂的公式，也有必要用命令 `\cdot` 将圆点符表示出来。

```
\begin{displaymath}
v = {\sigma}_1 \cdot {\sigma}_2
{\tau}_1 \cdot {\tau}_2
\end{displaymath}
```

$$v = \sigma_1 \cdot \sigma_2 \tau_1 \cdot \tau_2$$

函数名通常用罗马字体正体排版，而不是像变量名一样用意大利体排版。因此， \LaTeX 提供下述命令来排版最重要的一些函数名。

```
\arccos \cos \csc \exp \ker \limsup \min
\arcsin \cosh \deg \gcd \lg \ln \Pr
\arctan \cot \det \hom \lim \log \sec
\arg \coth \dim \inf \liminf \max \sin
\sinh \sup \tan \tanh
```

```
\[\lim_{x \rightarrow 0}
\frac{\sin x}{x}=1\]
```

$$\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$$

排版模函数 (modulo function) 有两个命令：`\bmod` 用于二元运算符“ $a \bmod b$ ”，`\pmod` 用于表达式，例如“ $x \equiv a \pmod{b}$ ”。

分数 (fraction) 使用 `\frac{...}{...}` 排版。一般来说， $1/2$ 这种形式更受欢迎，因为对于少量的分式，它看起来更好些。

```
$1\frac{1}{2}$~hours
\begin{displaymath}
\frac{x^2}{k+1} \qquad
x^{\frac{2}{k+1}} \qquad
x^{1/2}
\end{displaymath}
```

$$1\frac{1}{2} \text{ hours}$$

$$\frac{x^2}{k+1} \quad x^{\frac{2}{k+1}} \quad x^{1/2}$$

排版二项系数或类似的结构可以使用命令 `{... \choose ...}` 或 `{... \atop ...}`。第二个命令与第一个命令的输出相同，只是没有括号³。

```
\begin{displaymath}
{n \choose k} \qquad {x \atop y+2}
\end{displaymath}
```

$$\binom{n}{k} \quad \frac{x}{y+2}$$

对于二元关系，将符号堆在一起可能更有用。`\stackrel` 将第一项中的符号以上标大小放在处于正常位置的第二项上。

³注意这些旧命令在 `amsmath` 宏集中禁止使用，而是用 `\binom` 和 `\genfrac` 来代替。后者是所有相关结构的超集，例如可以通过 `\newcommand{\newatop}[2]\genfrac{}{}{0pt}{1}{#1}{#2}` 来得到 `\atop` 的一个类似结构

```
\begin{displaymath}
\int f_N(x) \stackrel{!}{=} 1
\end{displaymath}
```

$$\int f_N(x) \stackrel{!}{=} 1$$

积分运算符 (integral operator) 用 `\int` 来生成。求和运算符 (sum operator) 由 `\sum` 生成。乘积运算符 (product operator) 由 `\prod` 生成。上限和下限用 `^` 和 `_` 来生成, 类似于上标和下标⁴。

```
\begin{displaymath}
\sum_{i=1}^n \int_0^{\frac{\pi}{2}} \prod_{\epsilon}
\prod_{\epsilon} \epsilon
\end{displaymath}
```

$$\sum_{i=1}^n \int_0^{\frac{\pi}{2}} \prod_{\epsilon}$$

对于括号 (braces) 和其它分隔符 (delimiters), 在 $\mathrm{T}_\mathrm{E}\mathrm{X}$ 中有各种各样的符号 (例如 $\langle \parallel \updownarrow$)。圆括号和方括号可以用相应的键输入。花括号用 `\{`。其它的分隔符用专门命令 (例如 `\updownarrow`) 来生成。所有可用的分隔符列表可以参考第 52 页中的表 3.8。

```
\begin{displaymath}
a, b, c \neq \{a, b, c\}
\end{displaymath}
```

$$a, b, c \neq \{a, b, c\}$$

如果将命令 `\left` 放在开分隔符前, $\mathrm{T}_\mathrm{E}\mathrm{X}$ 会自动决定分隔符的正确大小。注意必须用对应的右分隔符 `\right` 来关闭每一个左分隔符 `\left`, 并且只有当这两个分隔符排在同一行时大小才会被正确确定。如果不想在右边放任何东西, 使用隐藏的 `\right.'`!

```
\begin{displaymath}
1 + \left( \frac{1}{1-x^2} \right)^3
\end{displaymath}
```

$$1 + \left(\frac{1}{1-x^2} \right)^3$$

某些情况下有必要手工指出数学分隔符的正确大小, 这可以使用命令 `\big`, `\Big`, `\bigg` 及 `\Bigg` 作为大多数分隔符命令的前缀⁵。

⁴ $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathrm{L}\mathrm{A}\mathrm{T}_\mathrm{E}\mathrm{X}$ 另外有多行上标/下标。

⁵ 如果使用了改变大小的命令或者指定了 11pt 或 12pt 选项, 这些命令并不像预料的那样起作用。使用 `exscale` 或 `amsmath` 宏集来修正这种行为。

$$\frac{(x+1)(x-1)}{(x^2-1)^2}$$

$$\left((x+1)(x-1) \right)^2$$

将三个圆点（**three dots**）输入公式可以使用几种命令。`\ldots` 将点排在基线上。`\cdots` 将它们设置为居中。除此之外，可用 `\vdots` 命令使其垂直，而用 `\ddots` 将得到对角型（diagonal dots）。第 3.5 节中还有其它例子。

```
\begin{displaymath}
x_{\{1\}}, \ldots, x_{\{n\}} \quad \quad \quad
x_{\{1\}} + \cdots + x_{\{n\}}
\end{displaymath}
```

$$x_1, \dots, x_n \qquad x_1 + \dots + x_n$$

3.4 数学空格

如果公式中由 $\text{T}_{\text{E}}\text{X}$ 选择的空格不令人满意，可以通过插入特殊的空格命令来进行调节。有几个命令用于小空格： \backslash ，对应于 $\frac{3}{18}$ quad (U)， \backslash ；对应于 $\frac{4}{18}$ quad (U)， \backslash ；对应于 $\frac{5}{18}$ quad (U)。脱离的空格符号 $\backslash_$ 生成中等大小的空格。 \backslash quad (U) 和 \backslash qqquad (U) 产生大空格。 \backslash quad 的大小对应于目前字体中字符 ‘M’ 的宽度。 $\backslash!$ 命令生成负空格 $-\frac{3}{18}$ quad (U)。

```
\newcommand{\ud}{\mathrm{d}}
\begin{displaymath}
\int\!\!\!\int\!\!\!\int_{\mathcal{D}} g(x,y)
\quad \mathrm{d}x, \quad \mathrm{d}y
\end{displaymath}
instead of
\begin{displaymath}
\int\!\!\!\int\!\!\!\int_{\mathcal{D}} g(x,y)\mathrm{d}x \mathrm{d}y
\end{displaymath}
```

instead of

$$\iint_D g(x, y) \mathrm{d}x \mathrm{d}y$$

注意微分中的‘d’按惯例设为罗马字体。

$\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ 提供微调多重积分符号间空格的另一种方式，也就是 `\iint`, `\iiint`, `\iiiiiint` 和 `\idotsint` 等命令。使用 `amsmath` 宏集，以上的例子可以用下面这种方式来排版。

```
\newcommand{\ud}{\mathrm{d}}
\begin{displaymath}
\iint_D \, \, \, \ud x \, \, \, \ud y
\end{displaymath}
```

$$\iint_D dx dy$$

更详细的介绍请参考电子文件 testmath.tex (包括在 $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ 中) 或 “The LaTeX Companion” 中的第八章。

3.5 垂直对齐

排版 arrays 使用 array 环境来排版数组 (arrays)。它有些类似于 tabular 环境, 使用 \\ 命令来分行。

```
\begin{displaymath}
\mathbf{X} =
\left( \begin{array}{ccc}
x_{11} & x_{12} & \ldots \\
x_{21} & x_{22} & \ldots \\
\vdots & \vdots & \ddots
\end{array} \right)
\end{displaymath}
```

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \dots \\ x_{21} & x_{22} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

array 环境也可以使用 “.” 作为隐藏右分隔符来排版只有一个大分隔符的表达式。

```
\begin{displaymath}
y = \left\{ \begin{array}{l}
a & \text{if } d > c \\
b+x & \text{in the morning} \\
l & \text{all day long}
\end{array} \right.
\end{displaymath}
```

$$y = \begin{cases} a & \text{if } d > c \\ b+x & \text{in the morning} \\ l & \text{all day long} \end{cases}$$

像在 tabular 环境中一样, 也可以在 array 环境中画线。例如分隔矩阵中的元素。

```
\begin{displaymath}
\left( \begin{array}{c|c}
1 & 2 \\ \hline
3 & 4
\end{array} \right)
\end{displaymath}
```

$$\left(\begin{array}{c|c} 1 & 2 \\ \hline 3 & 4 \end{array} \right)$$

对于分布于几行的公式或者方程组 (equation system)，可以使用 `eqnarray` 和 `eqnarray*` 环境来代替 `equation`。在 `eqnarray` 中，每一行都会有一个方程编号。`eqnarray*` 不对方程进行编号。

`eqnarray` 和 `eqnarray*` 环境类似于 `{rcl}` 形式的三列表格。中间的一列可以用作等号或不等号，或者其它看起来适合的符号。使用 `\\` 命令分行。

```
\begin{eqnarray}
f(x) &= & \cos x & \\
f'(x) &= & -\sin x & \\
\int_0^x f(y)dy &= & \sin x & \\
\end{eqnarray}
```

$$\begin{aligned} f(x) &= \cos x & (3.5) \\ f'(x) &= -\sin x & (3.6) \\ \int_0^x f(y)dy &= \sin x & (3.7) \end{aligned}$$

注意等号每边的空格都很大。可通过设定 `\setlength\arraycolsep{2pt}` 来减小空格，如下一个例子中所示。

长方程不会自动地分割成小的。作者必须指定在哪里分割以及缩进多少。以下是最常使用的两种方法。

```
{\setlength\arraycolsep{2pt}
\begin{eqnarray}
\sin x &= & x - \frac{x^3}{3!} + \frac{x^5}{5!} - \\
&& \frac{x^7}{7!} + \cdots & \\
&& {}-\frac{x^7}{7!}+{} \cdots & \\
\end{eqnarray}}
```

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots \quad (3.8)$$

```
\begin{eqnarray}
\lefteqn{ \cos x = 1 } \\
&& -\frac{x^2}{2!} + \\
&& \frac{x^4}{4!} - \\
&& \frac{x^6}{6!} + \cdots \\
\end{eqnarray}
```

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots \quad (3.9)$$

`\nonumber` 命令将阻止 L^AT_EX 为此方程生成一个编号。

使用这些方法很难得到垂直对齐的方程。`amsmath` 提供了一些更有用的方法 (见 `split` 和 `align` 环境)。

3.6 幻影

我们无法看到幻影 (phantom)，但是它们在许多人印象中仍然会占据一些空间。L^AT_EX 没有不同，我们也可以使用这些作一些有趣的空格技巧。

垂直对齐文本时使用 `^` 和 `_`。L^AT_EX 有时只是有一点帮助。使用 `\phantom` 命令可以为不在最终输出中出现的字符预留空间。最好参考下面的例子。

```
\begin{displaymath}
{}^{\{12\}}_{\{\phantom{1}6\}}\mathrm{C}
\quad \mathrm{versus} \quad
{}^{\{12\}}_{\{6\}}\mathrm{C}
\end{displaymath}
```

$${}^{12}_6\mathrm{C} \quad \text{versus} \quad {}^{12}_6\mathrm{C}$$

```
\begin{displaymath}
\Gamma_{ij}^{\{\phantom{i}j\}k}
\quad \mathrm{versus} \quad
\Gamma_{ij}^{\{j\}k}
\end{displaymath}
```

$$\Gamma_{ij}^{k} \quad \text{versus} \quad \Gamma_{ij}^k$$

3.7 数学字体大小

在数学模式中，T_EX 根据上下文选择字体大小。例如，使用较小的字体排版上标。如果想用罗马字体排版方程中的一部分，不要使用 `\textrm` 命令，因为当 `\textrm` 暂时脱离文本模式时字体大小交换机制不起作用。这时可以使用 `\mathrm` 来确保字体大小交换机制起作用。但是需要注意的是，`\mathrm` 只对于较短的项才起作用。空格仍然不起作用，并且重音字符也不起作用⁶。

```
\begin{equation}
2^{\{\textrm{nd}\}} \quad \quad
2^{\{\mathrm{nd}\}}
\end{equation}
```

$$2^{\mathrm{nd}} \quad 2^{\mathrm{nd}} \quad (3.10)$$

尽管如此，有时必须告诉 L^AT_EX 正确的字体大小。在数学模式中，字体大小用四个命令来设定：

⁶ $\mathcal{A}\mathcal{M}\mathcal{S}$ -L^AT_EX 宏集使得 `\textrm` 命令与字体大小改变相兼容。

`\displaystyle (123)`, `\textstyle (123)`, `\scriptstyle (123)` and `\scriptscriptstyle (123)`.

改变式样也会影响上下界显示的方式。

```
\begin{displaymath}
\mathop{\mathrm{corr}}(X,Y)=
\frac{\displaystyle
\sum_{i=1}^n(x_i-\overline{x})
(y_i-\overline{y})}
{\displaystyle\biggl[
\sum_{i=1}^n(x_i-\overline{x})^2
\sum_{i=1}^n(y_i-\overline{y})^2
\biggr]^{1/2}}
\end{displaymath}
```

$$\mathrm{corr}(X,Y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\left[\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2 \right]^{1/2}}$$

这个例子中，我们需要比标准的 `\left[\right]` 还要大一些的括号。

3.8 定理、定义 ...

写数学文档时有可能需要一种方式来排版“引理”、“定义”、“公理”以及类似的结构。L^AT_EX 为此提供了下述命令：

```
\newtheorem{name}[counter]{text}[section]
```

name 是短关键字，用于标识“定理”。*text* 定义“定理”的真实名称，会在最终文件中打印出来。

方括号中的选项是任意的，可以用于指定“定理”中使用的标号。*counter* 可以指定先前声明的“定理”的 *name*。然后新“定理”会按同样的顺序编号。*section* 指定“定理”编号所在的章节层次。

在文件的导言中执行 `\newtheorem` 命令之后，在文件中可以使用如下命令。

```
\begin{name}[text]
This is my interesting theorem
\end{name}
```

理论上这是足够的。下面的例子有望尽释前疑，并使人清楚地意识到，`\newtheorem` 环境非常复杂，很难理解。

```
% definitions for the document
% preamble
\newtheorem{law}{Law}
\newtheorem{jury}[law]{Jury}
%in the document
\begin{law} \label{law:box}
Don't hide in the witness box
\end{law}
\begin{jury}[The Twelve]
It could be you! So beware and
see law~\ref{law:box}\end{jury}
\begin{law}No, No, No\end{law}
```

Law 1 *Don't hide in the witness box*

Jury 2 (The Twelve) *It could be you! So beware and see law [1](#)*

Law 3 *No, No, No*

“Jury”定理和“Law”定理使用同一个计数器。因此，编号是顺序排列的。方括号中的选项用于为这个定理指定一个标题或者类似的东西。

```
\flushleft
\newtheorem{mur}{Murphy}[section]
\begin{mur}
If there are two or more
ways to do something, and
one of those ways can result
in a catastrophe, then
someone will do it.\end{mur}
```

Murphy 3.8.1 *If there are two or more ways to do something, and one of those ways can result in a catastrophe, then someone will do it.*

“Murphy”定理的编号与当前节相链接。也可以使用其它单位，例如章或小节。

3.9 粗体符号

在 \LaTeX 中很难得到粗体符号。这也许是故意的，因为业余排版者总是过份使用粗体。字体改变命令 $\text{\texttt{\textbackslash mathbf}}$ 给出粗体字母，但是这些是罗马字体（竖直的），而数学符号通常是斜体。有一个 $\text{\texttt{\textbackslash boldmath}}$ 命令，但是这只能用于数学模式之外。对于符号也是如此。

```
\begin{displaymath}
\mu, M \quad \mathbf{M} \quad \quad
\mbox{\texttt{\textbackslash boldmath $\mu, M$}}
\end{displaymath}
```

μ, M \mathbf{M} $\boldsymbol{\mu, M}$

注意逗号也是粗体，这可能不是所需要的。

使用工具包中的 `amssbsy` (包括在 `amsmath` 中) 和 `bm` 很容易办到这点, 因为它们包含 `\boldsymbol` 命令。

```
\begin{displaymath}
\mu, M \quad \quad
\boldsymbol{\mu}, \boldsymbol{M}
\end{displaymath}
```

$$\mu, M \quad \quad \boldsymbol{\mu}, \boldsymbol{M}$$

3.10 数学符号表

下面的表格中将给出在数学模式中常用的所有符号。使用表 3.12–3.16⁷ 所列出的符号，必须事先安装 AMS 数学字库并且在文档的导言区加载宏包：amssymb。如果你的系统中没有安装 AMS 宏包和数学字库，可去下述地址下载：

CTAN:/tex-archive/macros/latex/required/amslatex

表 3.1: 数学模式重音符

\hat{a}	<code>\hat{a}</code>	\check{a}	<code>\check{a}</code>	\tilde{a}	<code>\tilde{a}</code>	\acute{a}	<code>\acute{a}</code>
\grave{a}	<code>\grave{a}</code>	\dot{a}	<code>\dot{a}</code>	\ddot{a}	<code>\ddot{a}</code>	\breve{a}	<code>\breve{a}</code>
\bar{a}	<code>\bar{a}</code>	\vec{a}	<code>\vec{a}</code>	\widehat{A}	<code>\widehat{A}</code>	\widetilde{A}	<code>\widetilde{A}</code>

表 3.2: 小写希腊字母

α	<code>\alpha</code>	θ	<code>\theta</code>	\omicron	<code>\omicron</code>	υ	<code>\upsilon</code>
β	<code>\beta</code>	ϑ	<code>\vartheta</code>	π	<code>\pi</code>	ϕ	<code>\phi</code>
γ	<code>\gamma</code>	ι	<code>\iota</code>	ϖ	<code>\varpi</code>	φ	<code>\varphi</code>
δ	<code>\delta</code>	κ	<code>\kappa</code>	ρ	<code>\rho</code>	χ	<code>\chi</code>
ϵ	<code>\epsilon</code>	λ	<code>\lambda</code>	ϱ	<code>\varrho</code>	ψ	<code>\psi</code>
ε	<code>\varepsilon</code>	μ	<code>\mu</code>	σ	<code>\sigma</code>	ω	<code>\omega</code>
ζ	<code>\zeta</code>	ν	<code>\nu</code>	ς	<code>\varsigma</code>		
η	<code>\eta</code>	ξ	<code>\xi</code>	τ	<code>\tau</code>		

表 3.3: 大写希腊字母

Γ	<code>\Gamma</code>	Λ	<code>\Lambda</code>	Σ	<code>\Sigma</code>	Ψ	<code>\Psi</code>
Δ	<code>\Delta</code>	Ξ	<code>\Xi</code>	Υ	<code>\Upsilon</code>	Ω	<code>\Omega</code>
Θ	<code>\Theta</code>	Π	<code>\Pi</code>	Φ	<code>\Phi</code>		

⁷这些表格来自 David Carlisle 的 `symbols.tex`，随后根据 Josef Tkadlec 的建议作了较大的改动。

表 3.4: 二元关系符

你可以在下述命令的前面加上 `\not` 来得到其否定形式。

$<$	<code><</code>	$>$	<code>></code>	$=$	<code>=</code>
\leq	<code>\leq</code> or <code>\le</code>	\geq	<code>\geq</code> or <code>\ge</code>	\equiv	<code>\equiv</code>
\ll	<code>\ll</code>	\gg	<code>\gg</code>	$\dot{=}$	<code>\doteq</code>
\prec	<code>\prec</code>	\succ	<code>\succ</code>	\sim	<code>\sim</code>
\preceq	<code>\preceq</code>	\succeq	<code>\succeq</code>	\simeq	<code>\simeq</code>
\subset	<code>\subset</code>	\supset	<code>\supset</code>	\approx	<code>\approx</code>
\subseteq	<code>\subseteq</code>	\supseteq	<code>\supseteq</code>	\cong	<code>\cong</code>
\sqsubset	<code>\sqsubset</code> ^a	\sqsupset	<code>\sqsupset</code> ^a	\Join	<code>\Join</code> ^a
\sqsubseteq	<code>\sqsubseteq</code>	\sqsupseteq	<code>\sqsupseteq</code>	\bowtie	<code>\bowtie</code>
\in	<code>\in</code>	\ni	<code>\ni</code> , <code>\owns</code>	\propto	<code>\propto</code>
\vdash	<code>\vdash</code>	\dashv	<code>\dashv</code>	\models	<code>\models</code>
\mid	<code>\mid</code>	\parallel	<code>\parallel</code>	\perp	<code>\perp</code>
\smile	<code>\smile</code>	\frown	<code>\frown</code>	\asymp	<code>\asymp</code>
$:$	<code>:</code>	\notin	<code>\notin</code>	\neq	<code>\neq</code> or <code>\ne</code>

^a使用宏包 `latexsym` 来得到这个符号

表 3.5: 二元运算符

$+$	<code>+</code>	$-$	<code>-</code>	\triangleleft	<code>\triangleleft</code>
\pm	<code>\pm</code>	\mp	<code>\mp</code>	\triangleright	<code>\triangleright</code>
\cdot	<code>\cdot</code>	\div	<code>\div</code>	\star	<code>\star</code>
\times	<code>\times</code>	\setminus	<code>\setminus</code>	\ast	<code>\ast</code>
\cup	<code>\cup</code>	\cap	<code>\cap</code>	\circ	<code>\circ</code>
\sqcup	<code>\sqcup</code>	\sqcap	<code>\sqcap</code>	\bullet	<code>\bullet</code>
\vee	<code>\vee</code> , <code>\lor</code>	\wedge	<code>\wedge</code> , <code>\land</code>	\diamond	<code>\diamond</code>
\oplus	<code>\oplus</code>	\ominus	<code>\ominus</code>	\uplus	<code>\uplus</code>
\odot	<code>\odot</code>	\oslash	<code>\oslash</code>	\amalg	<code>\amalg</code>
\otimes	<code>\otimes</code>	\bigcirc	<code>\bigcirc</code>	\dagger	<code>\dagger</code>
\triangleup	<code>\triangleup</code>	\bigtriangledown	<code>\bigtriangledown</code>	\ddagger	<code>\ddagger</code>
\lhd	<code>\lhd</code> ^a	\rhd	<code>\rhd</code> ^a	\wr	<code>\wr</code>
\unlhd	<code>\unlhd</code> ^a	\unrhd	<code>\unrhd</code> ^a		

表 3.6: 大尺寸运算符

\sum	<code>\sum</code>	\bigcup	<code>\bigcup</code>	\bigvee	<code>\bigvee</code>	\bigoplus	<code>\bigoplus</code>
\prod	<code>\prod</code>	\bigcap	<code>\bigcap</code>	\bigwedge	<code>\bigwedge</code>	\bigotimes	<code>\bigotimes</code>
\coprod	<code>\coprod</code>	\bigsqcup	<code>\bigsqcup</code>			\bigodot	<code>\bigodot</code>
\int	<code>\int</code>	\oint	<code>\oint</code>			\biguplus	<code>\biguplus</code>

表 3.7: 箭头

\leftarrow	<code>\leftarrow</code> or <code>\gets</code>	\longleftarrow	<code>\longleftarrow</code>	\uparrow	<code>\uparrow</code>
\rightarrow	<code>\rightarrow</code> or <code>\to</code>	\longrightarrow	<code>\longrightarrow</code>	\downarrow	<code>\downarrow</code>
\leftrightarrow	<code>\leftrightarrow</code>	\longleftrightarrow	<code>\longleftrightarrow</code>	\updownarrow	<code>\updownarrow</code>
\Leftarrow	<code>\Leftarrow</code>	\Longleftarrow	<code>\Longleftarrow</code>	\Uparrow	<code>\Uparrow</code>
\Rightarrow	<code>\Rightarrow</code>	\Longrightarrow	<code>\Longrightarrow</code>	\Downarrow	<code>\Downarrow</code>
\Leftrightarrow	<code>\Leftrightarrow</code>	\Longleftrightarrow	<code>\Longleftrightarrow</code>	\Updownarrow	<code>\Updownarrow</code>
\mapsto	<code>\mapsto</code>	\longmapsto	<code>\longmapsto</code>	\nearrow	<code>\nearrow</code>
\hookrightarrow	<code>\hookrightarrow</code>	\hookleftarrow	<code>\hookleftarrow</code>	\searrow	<code>\searrow</code>
\leftharpoonup	<code>\leftharpoonup</code>	\rightharpoonup	<code>\rightharpoonup</code>	\swarrow	<code>\swarrow</code>
\leftharpoondown	<code>\leftharpoondown</code>	\rightharpoondown	<code>\rightharpoondown</code>	\nwarrow	<code>\nwarrow</code>
\rightleftharpoons	<code>\rightleftharpoons</code>	\iff	<code>\iff</code> (bigger spaces)	\leadsto	<code>\leadsto</code> ^a

^a使用宏包 `latexsym` 来得到这个符号

表 3.8: 定界符

$($	<code>(</code>	$)$	<code>)</code>	\uparrow	<code>\uparrow</code>	\Uparrow	<code>\Uparrow</code>
$[$	<code>[</code> or <code>\lbrack</code>	$]$	<code>]</code> or <code>\rbrack</code>	\downarrow	<code>\downarrow</code>	\Downarrow	<code>\Downarrow</code>
$\{$	<code>\{</code> or <code>\lbrace</code>	$\}$	<code>\}</code> or <code>\rbrace</code>	\updownarrow	<code>\updownarrow</code>	\Updownarrow	<code>\Updownarrow</code>
\langle	<code>\langle</code>	\rangle	<code>\rangle</code>	$ $	<code> </code> or <code>\vert</code>	$\ $	<code>\ </code> or <code>\Vert</code>
\lfloor	<code>\lfloor</code>	\rfloor	<code>\rfloor</code>	\lceil	<code>\lceil</code>	\rceil	<code>\rceil</code>
$/$	<code>/</code>	\backslash	<code>\backslash</code>	$.$	<code>.</code> (dual. empty)		

表 3.9: 大尺寸定界符

$\left($	<code>\lgrou</code>	$\right)$	<code>\rgrou</code>	$\left[$	<code>\lmoustache</code>	$\right]$	<code>\rmoustache</code>
\uparrow	<code>\arrowvert</code>	\downarrow	<code>\Arrowvert</code>	$\{$	<code>\bracevert</code>	$\}$	

表 3.10: 其它符号

\dots	<code>\dots</code>	\cdots	<code>\cdots</code>	\vdots	<code>\vdots</code>	\ddots	<code>\ddots</code>
\hbar	<code>\hbar</code>	\imath	<code>\imath</code>	\jmath	<code>\jmath</code>	ℓ	<code>\ell</code>
\Re	<code>\Re</code>	\Im	<code>\Im</code>	\aleph	<code>\aleph</code>	\wp	<code>\wp</code>
\forall	<code>\forall</code>	\exists	<code>\exists</code>	\mho^a	<code>\mho^a</code>	∂	<code>\partial</code>
$'$	<code>'</code>	$'$	<code>\prime</code>	\emptyset	<code>\emptyset</code>	∞	<code>\infty</code>
∇	<code>\nabla</code>	\triangle	<code>\triangle</code>	\Box^a	<code>\Box^a</code>	\diamond	<code>\Diamond^a</code>
\bot	<code>\bot</code>	\top	<code>\top</code>	\angle	<code>\angle</code>	\surd	<code>\surd</code>
\diamondsuit	<code>\diamondsuit</code>	\heartsuit	<code>\heartsuit</code>	\clubsuit	<code>\clubsuit</code>	\spadesuit	<code>\spadesuit</code>
\neg or \nolnot	<code>\neg</code> or <code>\nolnot</code>	\flat	<code>\flat</code>	\natural	<code>\natural</code>	\sharp	<code>\sharp</code>

^a使用宏包 `latexsym` 来得到这个符号

表 3.11: 非数学符号

这些符号也可以在文本模式中使用。

\dagger	<code>\dag</code>	\S	<code>\S</code>	\copyright	<code>\copyright</code>
\ddagger	<code>\ddag</code>	\P	<code>\P</code>	\pounds	<code>\pounds</code>

表 3.12: AMS 定界符

\ulcorner	<code>\ulcorner</code>	\urcorner	<code>\urcorner</code>	\llcorner	<code>\llcorner</code>	\lrcorner	<code>\lrcorner</code>
\lvert	<code>\lvert</code>	\rvert	<code>\rvert</code>	\lVert	<code>\lVert</code>	\rVert	<code>\rVert</code>

表 3.13: AMS 希腊和希伯来字母

\digamma	<code>\digamma</code>	\varkappa	<code>\varkappa</code>	\beth	<code>\beth</code>	\daleth	<code>\daleth</code>	\gimel	<code>\gimel</code>
------------	-----------------------	-------------	------------------------	---------	--------------------	-----------	----------------------	----------	---------------------

表 3.14: AMS 二元关系符

\lessdot	<code>\lessdot</code>	\gtrdot	<code>\gtrdot</code>	\doteqdot or \Doteq	<code>\doteqdot or \Doteq</code>
\leqslant	<code>\leqslant</code>	\geqslant	<code>\geqslant</code>	\risingdotseq	<code>\risingdotseq</code>
\eqslantless	<code>\eqslantless</code>	\eqslantgtr	<code>\eqslantgtr</code>	\fallingdotseq	<code>\fallingdotseq</code>
\leqq	<code>\leqq</code>	\geqq	<code>\geqq</code>	\eqcirc	<code>\eqcirc</code>
\lll or \llless	<code>\lll or \llless</code>	\ggg or \gggtr	<code>\ggg or \gggtr</code>	\circeq	<code>\circeq</code>
\lesssim	<code>\lesssim</code>	\gtrsim	<code>\gtrsim</code>	\triangleq	<code>\triangleq</code>
\lessapprox	<code>\lessapprox</code>	\gtrapprox	<code>\gtrapprox</code>	\bumpeq	<code>\bumpeq</code>
\lessgtr	<code>\lessgtr</code>	\gtrless	<code>\gtrless</code>	\Bumpeq	<code>\Bumpeq</code>
\lesseqgtr	<code>\lesseqgtr</code>	\gtreqless	<code>\gtreqless</code>	\thicksim	<code>\thicksim</code>
\lesseqqgtr	<code>\lesseqqgtr</code>	\gtreqqless	<code>\gtreqqless</code>	\thickapprox	<code>\thickapprox</code>
\preccurlyeq	<code>\preccurlyeq</code>	\succcurlyeq	<code>\succcurlyeq</code>	\approxeq	<code>\approxeq</code>
\curlyeqprec	<code>\curlyeqprec</code>	\curlyeqsucc	<code>\curlyeqsucc</code>	\backsim	<code>\backsim</code>
\precsim	<code>\precsim</code>	\succsim	<code>\succsim</code>	\backsimeq	<code>\backsimeq</code>
\precapprox	<code>\precapprox</code>	\succapprox	<code>\succapprox</code>	\vDash	<code>\vDash</code>
\subseteq	<code>\subseteq</code>	\supseteq	<code>\supseteq</code>	\Vdash	<code>\Vdash</code>
\sqsubset	<code>\sqsubset</code>	\sqsupset	<code>\sqsupset</code>	\Vvdash	<code>\Vvdash</code>
\therefore	<code>\therefore</code>	\because	<code>\because</code>	\backepsilon	<code>\backepsilon</code>
\shortmid	<code>\shortmid</code>	\shortparallel	<code>\shortparallel</code>	\varpropto	<code>\varpropto</code>
\smallsmile	<code>\smallsmile</code>	\smallfrown	<code>\smallfrown</code>	\between	<code>\between</code>
\vartriangleleft	<code>\vartriangleleft</code>	\vartriangleright	<code>\vartriangleright</code>	\pitchfork	<code>\pitchfork</code>
\trianglelefteq	<code>\trianglelefteq</code>	\trianglerighteq	<code>\trianglerighteq</code>	\blacktriangleleft	<code>\blacktriangleleft</code>
				\blacktriangleright	<code>\blacktriangleright</code>

表 3.15: AMS 箭头

\dashleftarrow	<code>\dashleftarrow</code>	\dashrightarrow	<code>\dashrightarrow</code>	\multimap	<code>\multimap</code>
\leftrightsquigarrow	<code>\leftrightsquigarrow</code>	\rightleftarrows	<code>\rightleftarrows</code>	\upuparrows	<code>\upuparrows</code>
\leftrightsquigarrow	<code>\leftrightsquigarrow</code>	\rightleftarrows	<code>\rightleftarrows</code>	\downdownarrows	<code>\downdownarrows</code>
\Lleftarrow	<code>\Lleftarrow</code>	\Rrightarrow	<code>\Rrightarrow</code>	\upharpoonleft	<code>\upharpoonleft</code>
\twoheadleftarrow	<code>\twoheadleftarrow</code>	\twoheadrightarrow	<code>\twoheadrightarrow</code>	\upharpoonright	<code>\upharpoonright</code>
\leftarrowtail	<code>\leftarrowtail</code>	\rightarrowtail	<code>\rightarrowtail</code>	\downharpoonleft	<code>\downharpoonleft</code>
\leftrightharpoons	<code>\leftrightharpoons</code>	\rightleftharpoons	<code>\rightleftharpoons</code>	\downharpoonright	<code>\downharpoonright</code>
\Lsh	<code>\Lsh</code>	\Rsh	<code>\Rsh</code>	\rightsquigarrow	<code>\rightsquigarrow</code>
\looparrowleft	<code>\looparrowleft</code>	\looparrowright	<code>\looparrowright</code>	\leftrightsquigarrow	<code>\leftrightsquigarrow</code>
\curvearrowleft	<code>\curvearrowleft</code>	\curvearrowright	<code>\curvearrowright</code>		
\circlearrowleft	<code>\circlearrowleft</code>	\circlearrowright	<code>\circlearrowright</code>		

表 3.16: AMS 二元否定关系符和箭头

\nless	\ngtr	\varsubsetneqq
\lneq	\gneq	\varsupsetneqq
\nleq	\ngeq	\nsubseteqeq
\nleqslant	\ngeqslant	\nsupseteqeq
\lneqq	\gneqq	\nmid
\lvertneqq	\gvertneqq	\nparallel
\nleqq	\ngeqq	\nshortmid
\lnsim	\gnsim	\nshortparallel
\lnapprox	\gnapprox	\nsim
\nprec	\nsucc	\ncong
\npreceq	\nsucceq	\nvdash
\precneqq	\succneqq	\nvDash
\precnsim	\succnsim	\nVdash
\precnapprox	\succnapprox	\nVDash
\subsetneq	\supsetneq	\ntriangleleft
\varsubsetneq	\varsupsetneq	\ntriangleright
\nsubseteq	\nsupseteq	\ntrianglelefteq
\subseteqeq	\supseteqeq	\ntrianglerighteq
\nleftarrow	\nrightarrow	\nleftrightarrow
\nLeftarrow	\nRightarrow	\nLeftrightarrow

表 3.17: AMS 二元运算符

\dotplus	\centerdot	\intercal
\ltimes	\rtimes	\divideontimes
\Cup or \doublecup	\Cap or \doublecap	\smallsetminus
\veebar	\barwedge	\doublebarwedge
\boxplus	\boxminus	\circleddash
\boxtimes	\boxdot	\circledcirc
\leftthreetimes	\rightthreetimes	\circledast
\curlyvee	\curlywedge	

表 3.18: AMS 其它符号

\hbar	<code>\hbar</code>	\hslash	<code>\hslash</code>	\Bbbk	<code>\Bbbk</code>
\square	<code>\square</code>	\blacksquare	<code>\blacksquare</code>	\textcircled{S}	<code>\circledS</code>
\triangle	<code>\vartriangle</code>	\blacktriangle	<code>\blacktriangle</code>	\complement	<code>\complement</code>
∇	<code>\triangledown</code>	\blacktriangledown	<code>\blacktriangledown</code>	\Game	<code>\Game</code>
\lozenge	<code>\lozenge</code>	\blacklozenge	<code>\blacklozenge</code>	\bigstar	<code>\bigstar</code>
\angle	<code>\angle</code>	\measuredangle	<code>\measuredangle</code>	\sphericalangle	<code>\sphericalangle</code>
\diagup	<code>\diagup</code>	\diagdown	<code>\diagdown</code>	\backprime	<code>\backprime</code>
\nexists	<code>\nexists</code>	\Finv	<code>\Finv</code>	\varnothing	<code>\varnothing</code>
\eth	<code>\eth</code>	\mho	<code>\mho</code>		

表 3.19: 数学字母

例子	命令	所需宏包
ABCdef	<code>\mathrm{ABCdef}</code>	
ABCdef	<code>\mathit{ABCdef}</code>	
\mathnormal{ABCdef}	<code>\mathnormal{ABCdef}</code>	
\mathcal{ABC}	<code>\mathcal{ABC}</code>	
\mathscr{ABC}	<code>\mathscr{ABC}</code>	<code>mathrsfs</code>
\mathcal{ABC}	<code>\mathcal{ABC}</code>	<code>eucal</code> with option: <code>mathcal</code> or
	<code>\mathscr{ABC}</code>	<code>eucal</code> with option: <code>mathscr</code>
\mathfrak{ABCdef}	<code>\mathfrak{ABCdef}</code>	<code>eufrak</code>
\mathbb{ABC}	<code>\mathbb{ABC}</code>	<code>amsfonts</code> or <code>amssymb</code>

第四章 特殊功能

当你处理一个大型文档时， \LaTeX 的一些特殊功能，例如索引的自动生成、参考文献的管理等等，会给你以很大的帮助。详细的关于 \LaTeX 的特殊功能以及可能的增强功能的描述可以在 *\LaTeX Manual* [1] 和 *The \LaTeX Companion* [3] 找到。

4.1 包含 EPS 图形

\LaTeX 通过 `figure` 和 `table` 环境提供了处理图像或者图形等浮动对象的基本能力。

有几种办法可以通过使用基本 \LaTeX 命令或者 \LaTeX 扩展宏包来产生实际的图形。但是大多数用户发现这些命令相当难以理解。因此我们打算在这个手册里深入介绍这些内容。如果需要这方面的详细信息，请参阅 *The \LaTeX Companion* [3] 和 *\LaTeX Manual* [1]。

一个比较简单的在文档中加入图形的办法就是使用特定的软件包¹生成图形文件，然后将最终的图形文件包含到文档中。 \LaTeX 的宏包提供了许多方法来完成这个工作。在这个手册里，我们只讨论 Encapsulated PostScript (EPS) 图形文件的使用。因为它比较简单而且被广泛地使用。为了使用 EPS 格式的图片，你必须有一个 PostScript 打印机²来输出结果。

由 D. P. Carlisle 制作的 `graphicx` 宏包包含了一套很好的命令来包含图形。它是一个叫作 “`graphics`” 的宏包集中的一部分³。

假设你使用的系统安装了 PostScript 打印机和 `graphicx` 宏包，那么你就可以通过下面的步骤一步步地把一幅图片加入你的文档中：

1. 用你的图像软件输出 EPS 格式的图形文件⁴。

¹例如 XFig, CorelDraw!, Freehand, Gnuplot, ...

²另外一个可以用来输出 PostScript 的工具是 GHOSTSCRIPT 软件，它可以从 CTAN:/tex-archive/support/ghostscript 得到。Windows 和 OS/2 用户可能更喜欢用 GSVIEW。

³CTAN:/tex-archive/macros/latex/required/graphics

⁴如果你的软件不能输出 EPS 格式的文件，你可以尝试安装一个 PostScript 打印机驱动

2. 在输入文件的导言中加上下面的命令来装入 graphicx 宏包。

```
\usepackage[driver]{graphicx}
```

这里 *driver* 是你使用的“dvi 到 postscript”的转换程序。最常用的是 dvips。因为 T_EX 中没有规定包含图形的标准，所以程序的名字是必需的。知道了 *driver* 的名字，graphicx 宏包就可以选择合适的方法在 .dvi 文件中插入关于图形的信息。这样打印机才能理解这些信息并正确的包含这些 .eps 文件。

3. 使用命令

```
\includegraphics[key=value, ...]{file}
```

来把文件 *file* 加入你的文档。可选的参数是一系列由逗号隔开的关键词 *keys* 和相应的值 *values*。关键词 *keys* 可以用来改变包含的图形的宽度、高度以及旋转角度等等。表 4.1 列出了最主要的几个关键词。

表 4.1: graphicx 宏包使用的关键词

width	把图形缩放到指定的宽度
height	把图形缩放到指定的高度
angle	逆时针旋转图形
scale	缩放图形

下面这个例子可以帮助我们理解整个过程：

```
\begin{figure}
\begin{center}
\includegraphics[angle=90, width=0.5\textwidth]{test}
\end{center}
\end{figure}
```

这段代码把存储在文件 test.eps 中的图形包含到文档里。首先 图形被旋转 90 度，然后进行缩放使得图形的宽度等于标准页面宽度的 0.5 倍。因为

程序（例如 Apple LaserWriter），然后将你的图形通过这个驱动程序打印到文件。运气好的话，这个文件可能会是 EPS 格式的。注意一个 EPS 文件不能包含超过一页的内容。一些打印机驱动程序可以明确地指定为输出 EPS 文件。

没有指定图形的高度，图形的高宽变化的比例是 1.0，也就是保持原来的高宽比。高度和宽度参数也可以用绝对长度单位来指定。详细的信息可以在第 73 页的表 5.5 中找到。如果你想知道更多这方面的知识，请阅读文献 [8] 和 [11]。

4.2 参考文献

你可以通过 thebibliography 环境来产生一个参考文献。每个参考文献的条目以如下的命令开头

```
\bibitem{marker}
```

然后使用 *marker* 在正文中引用这本书、这篇文章或者论文。

```
\cite{marker}
```

参考文献条目的编号是自动生成的。 `\begin{thebibliography}` 命令后的参数设置了最大的编号宽度。在下面的例子中，{99} 告诉 L^AT_EX 参考文献条目的编号不会比数字 99 更宽。

```
Partl~\cite{pa} has
proposed that \ldots
\begin{thebibliography}{99}
\bibitem{pa} H.~Partl:
\emph{German \TeX},
TUGboat Volume~9, Issue~1 (1988)
\end{thebibliography}
```

Partl [1] has proposed that ...

参考文献

[1] H. Partl: *German T_EX*, TUGboat Volume 9, Issue 1 (1988)

对于大型的项目，你也许需要使用 BibT_EX 程序。BibT_EX 包含在大多数的 T_EX 发行版本中。它能够让你维护一个参考文献数据库，并从中生成你的论文引用到的文献条目。BibT_EX 对参考文献的描述是基于一种样式表的概念，它可以让你按照大量预先设计好的格式来创建你的参考文献。

表 4.2: 索引关键词格式示例。

示例	索引项	注解
<code>\index{hello}</code>	hello, 1	普通格式的索引项
<code>\index{hello!Peter}</code>	Peter, 3	'hello'下的子项
<code>\index{Sam@\textsl{Sam}}</code>	<i>Sam</i> , 2	定义索引项的格式
<code>\index{Lin@\textbf{Lin}}</code>	Lin , 7	同上
<code>\index{Jenny textbf}</code>	Jenny, 3	定义页码的格式
<code>\index{Joe textit}</code>	Joe, 5	同上

4.3 索引

许多书籍最有用的部分莫过于它们的索引了。使用 L^AT_EX 和辅助工具 `makeindex`⁵，我们能够很容易的生成索引。在这个手册里，只介绍了最基本的索引生成命令。更进一步的了解请参考 *The L^AT_EX Companion* [3]。

为了使用 L^AT_EX 的索引功能，宏包 `makeidx` 必须在导言部分被引入：

`\usepackage{makeidx}`

然后在导言中使用

`\makeindex`

激活索引命令。

索引的内容通过命令

`\index{key}`

指定，这里 *key* 是索引项的关键词。你可以在需要被索引的地方加入这条命令。表 4.2 举例解释了 *key* 参数用法。

当 L^AT_EX 处理输入文档时，每个 `\index` 命令都会将适当的索引项和当前页码写入一个特殊的文件中。这个文件的名字和 L^AT_EX 输入文档相同，但具有不同的扩展名后缀 (`.idx`)。这个 `.idx` 文件需要用 `makeindex` 程序来

⁵在文件名不允许超过 8 个字符的操作系统上，这个程序被命名为 `makeidx`。

处理。

`makeindex filename`

`makeindex` 程序生成一个与源文件同名的排序索引文件，这个文件使用 `.ind` 为扩展名。当再次用 \LaTeX 处理源文件时，这个排序的索引文件将被包含到源文件中

`\printindex`

命令出现的位置。

\LaTeX 2 ϵ 附带的宏包 `showidx` 可以在正文的左边打印出索引项。这个功能在校对文档和索引项时十分有用。

4.4 定制页眉和页脚

Piet van Oostrum 编写的 `fancyhdr` 宏包⁶，提供了一些简单的命令使得我们可以定制文档的页眉和页脚。看一眼本页的顶部，你就能发现这个宏包的用处。

定制页眉和页脚时最棘手的事情莫过于得到每个页面所属的章节名称了。 \LaTeX 通过两个步骤来完成这个任务。在定义页眉和页脚时，你可以使用 `\rightmark` 命令来代表当前的节名，使用 `\leftmark` 来代表当前的章名。这两个命令的值将在处理 `chapter` 或者 `section` 命令时被赋值。

为了获得最大的灵活性，`\chapter` 等命令并不直接对 `\rightmark` 和 `\leftmark` 进行重新定义，而是通过调用 `\chaptermark` 命令或者 `\sectionmark`、`\subsectionmark` 来定义 `\rightmark` 和 `\leftmark`。

因此，如果你想修改页眉上显示的章名的话，只需要简单地重新定义 `\chaptermark` 命令。

图 4.1 显示了如何配置 `fancyhdr` 来得到和本文相似的页眉。无论如何我还是建议你先阅读一下宏包所带的文档。

4.5 Verbatim 宏包

在本文的前面部分你已经知道了 `verbatim` 环境。在这一节中，你将学会使用 `verbatim` 宏包。`verbatim` 宏包重新实现了 `verbatim` 环境，并解决了原来的 `verbatim` 环境的一些限制。这本身并没有什么特别的，但 `verbatim`

⁶可以在CTAN:/tex-archive/macros/latex/contrib/supported/fancyhdr得到。

```

\documentclass{book}
\usepackage{fancyhdr}
\pagestyle{fancy}
% with this we ensure that the chapter and section
% headings are in lowercase.
\renewcommand{\chaptermark}[1]{\markboth{#1}{}}
\renewcommand{\sectionmark}[1]{\markright{\thesection\ #1}}
\fancyhf{} % delete current setting for header and footer
\fancyhead[LE,RO]{\bfseries\thepage}
\fancyhead[LO]{\bfseries\rightmark}
\fancyhead[RE]{\bfseries\leftmark}
\renewcommand{\headrulewidth}{0.5pt}
\renewcommand{\footrulewidth}{0pt}
\addtolength{\headheight}{0.5pt} % make space for the rule
\fancypagestyle{plain}{%
  \fancyhead{} % get rid of headers on plain pages
  \renewcommand{\headrulewidth}{0pt} % and the line
}

```

图 4.1: fancyhdr 设置实例。

宏包还实现了一些新增的功能，这才是我在这里提到这个宏包的原因。
`verbatim` 宏包提供了

`\verbatiminput{filename}`

命令，这个命令允许你把一个 ASCII 码的文本文件包含到你的文档中来，就好像它们是在 `verbatim` 环境中一样。

`verbatim` 宏包是 ‘tools’ 宏包集的一部分，大多数的系统中都预装了这个宏包。如果你想更多地了解这个宏包，可以阅读 [9]。

4.6 下载并安装 L^AT_EX 宏包

大多数的 L^AT_EX 安装都带有大量预装的样式宏包，但还有很多可以在网上得到。在互联网寻找样式宏包的一个主要的地方就是 CTAN(<http://www.ctan.org/>)。

各种宏包的源文件，例如 `geometry`，`hyphenat` 等等，一般来说都包含两个文件：一个扩展名为 `.ins`，另一个扩展名为 `.dtx`。此外，通常会有一个 `readme.txt` 对宏包进行简要的说明。你应该先阅读这个文件。

无论如何，一旦你得到了宏包的源文件，你还要对它们进行处理使得

(a) 你的 T_EX 系统知道这个新的宏包，(b) 生成说明文档。下面是第一部分的步骤：

1. 对 .ins 文件运行 L^AT_EX 命令。这将会产生一个 .sty 文件。
2. 把 .sty 文件移到系统能找到的地方。通常是在 `.../localtexmf/tex/latex` 子目录下（Windows 或者 OS/2 用户应该改变斜线为反斜线）。
3. 刷新系统的文件名数据库。具体的命令取决于你使用的 L^AT_EX 系统：
teTeX, fpTeX – texhash; web2c – maketexlsr; MikTeX – initexmf
-update-fndb 或者使用图形界面。

现在你可以从 .dtx 文件生成说明文档：

1. 对 .dtx 文件运行 L^AT_EX 命令。这会生成一个 .dvi 文件。注意你可能需要多次运行 L^AT_EX 命令来正确处理交叉引用。
2. 检查一下 L^AT_EX 命令是否产生了 .idx 文件。如果没发现这个文件，你就可以执行第 5 步了。
3. 为了生成索引，敲入命令：

`makeindex -s gind.ist name`

(这里 *name* 表示不带扩展名的主文件名)。
4. 再次对 .dtx 文件运行 L^AT_EX 命令。
5. 最后一步但不是必需的，生成 .ps 文件或者 .pdf 文件以方便阅读。

有时你会看见生成了一个 .glo(glossary) 文件。在第 4 步和第 5 步之间运行下面的命令：

```
makeindex -s gglo.ist -o name.gls name.glo
```

确认在执行第 5 步前最后对 .dtx 文件运行一遍 L^AT_EX 命令。

第五章 定制 \LaTeX

到目前为止，运用你所学过的命令可以制作出能被绝大多数读者接受的文档。尽管这些文档看上去不够奇妙，但它们遵循了好的出版系统必须遵守的基本规则。这些规则可以使得文档容易被阅读，同时看起来也非常舒适。

然而在一些情况下， \LaTeX 也许并没有提供适合你需要的命令或者环境，或者利用已有的命令产生的输出和你想要的不同。

在这章中，我将尝试给出一些新的技术，运用这些技术可以教会 \LaTeX 玩一些新的把戏，或者也可以使得 \LaTeX 产生与众不同的输出。

5.1 建立新的命令、环境和包

你也许已经注意到我在这本书中介绍的所有命令都被包含在一个长方形框中，并且在文章最后的索引中还有他们的标记。我并没有直接采用标准的 \LaTeX 命令来实现这个，我创建了一个所谓的宏包（package），并在其中定义了我所需要的命令和环境。因此我可以简化我的文档为：

```
\begin{lscommand}  
  \ci{dum}  
\end{lscommand}
```



\dum

在这个例子中，我使用了一个新的环境 – `lscommand`。这个环境负责在命令的周围画出一个矩形框。同时我还使用了一个命令：`\ci`，这个命令负责输出宏包的名字，并且在索引中添加相应的条目。你可以在文章最后的索引中查找命令 `\dum`，然后你会发现有一个 `\dum` 的条目，这个条目中列出了包含有 `\dum` 命令的所有页的页码。

一旦我觉得这个命令在一个矩形框中进行排版的行为不再适合，我可以轻松的改变 `lscommand` 环境的定义，使得命令的输出具有新的外观。这种做法比起浏览整个文档逐个进行格式修改的方法要轻松得多。而如果我

们使用 L^AT_EX 原有的命令来做这件事情（在矩形框中显示命令并加入到索引中），那唯一的做法就是浏览整个并手动修改。

5.1.1 建立新的命令

为了增加你自己的命令，可以使用如下的命令：

```
\newcommand{name}[num]{definition}
```

基本上，这个命令需要两个参数，第一个参数 *name* 是你想要建立的命令的名称，第二个参数 *definition* 是命令的定义。第三个参数 *num* 是可选的，用于指定命令所需的参数数目（命令最多可以有9个参数）。如果不给出这个参数，那么新建的命令将不接受任何参数。

接下来的两个例子将帮助你理解这种机制。第一个例子定义了一个新的命令 – \tnss。这个命令是句子 “The Not So Short Introduction to L^AT_EX 2_ε” 的缩写。如果你需要在文档中多次使用本书的名称，那么定义这个命令将是非常方便的。

```
\newcommand{\tnss}{The not
  so Short Introduction to
  \LaTeXe}
This is ‘‘\tnss’’ \ldots{}
‘‘\tnss’’
```

This is “The not so Short Introduction to L^AT_EX 2_ε” ... “The not so Short Introduction to L^AT_EX 2_ε”

下一个例子演示了如何建立一个接受唯一参数的命令。在命令的定义中，标记 #1 将被你指定的参数所代替。如果你想使用多个参数，那么可以依次使用 #2、...、#9 等标记。

```
\newcommand{\txsit}[1]
{This is the \emph{#1} Short
  Introduction to \LaTeXe}
% in the document body:
\begin{itemize}
\item \txsit{not so}
\item \txsit{very}
\end{itemize}
```

- This is the *not so* Short Introduction to L^AT_EX 2_ε
- This is the *very* Short Introduction to L^AT_EX 2_ε

L^AT_EX 不允许你用 \newcommand 新建一个与原有命令重名的命令。有一个特殊的命令专门用于处理这种情况：\renewcommand。它使用与命令 \newcommand 相同的语法。

在某些情况之下，你可能会需要使用 `\providecommand` 命令。它完成与 `\newcommand` 命令相同的工作。但如果命令已经存在，L^AT_EX 2_ε 将会忽略这个命令。

处理 L^AT_EX 命令后尾随的空格有一些要注意的事项，参看第 6 页可以获得更多这方面的信息。

5.1.2 建立新的环境

与 `\newcommand` 命令类似，有一个命令用于建立新的环境。这个命令就是 `\newenvironment`，它的语法如下所示：

```
\newenvironment{name}[num]{before}{after}
```

与 `\newcommand` 命令类似，你可以为 `\newenvironment` 命令提供一个可选的参数。在参数 *before* 中提供的内容将在被命令包含的文本之前处理，而在参数 *after* 中提供的内容将恰好在 `\end{name}` 的前面处理。

下面的例子演示了 `\newenvironment` 命令的用法：

```
\newenvironment{king}
{\rule{1ex}{1ex}%
 \hspace{\stretch{1}}}
{\hspace{\stretch{1}}%
 \rule{1ex}{1ex}}
\begin{king}
My humble subjects \ldots
\end{king}
```

■ My humble subjects ... ■

参数 *num* 使用的方式与 `\newcommand` 命令相同。L^AT_EX 还同样保证你不会不小心改变了已有环境的定义。如果你确实希望改变一个存在的环境，你可以使用命令 `\renewenvironment`，它使用和命令 `\newenvironment` 相同的语法。

在这个例子中用到一些命令将在随后解释：`\rule` 命令的解释可以参看第 78 页，`\stretch` 命令的解释可以参看第 72 页，关于 `\hspace` 的信息可以在第 72 页找到。

5.1.3 建立你自己的宏包

如果你建立了很多自己的环境和命令，你的文档的导言部分将变得很长，在这种情况下，建立一个新的 L^AT_EX 包来存放所有你自己定义的命令

和环境将是一个好的处理方式。你可以在文档中使用 `\usepackage` 命令来引入自定义宏包中的环境和命令。

```
% Demo Package by Tobias Oetiker
\ProvidesPackage{demopack}
\newcommand{\tnss}{The not so Short Introduction to \LaTeXe}
\newcommand{\txsit}[1]{The \emph{#1} Short
    Introduction to \LaTeXe}
\newenvironment{king}{\begin{quote}}{\end{quote}}
```

图 5.1: Example Package.

写一个宏包的基本工作就是将你原本很长的文档导言拷贝到一个分离的文件中去，这个文件需要以 `.sty` 结尾。你还需要使用一个专用的命令：

`\ProvidesPackage{package name}`

这个命令应该在你的包的最前面使用。`\ProvidesPackage` 用于告诉 \LaTeX 宏包的名称从而允许 \LaTeX 在你尝试两次引入同一个宏包的时候给出一个良好的错误信息，图 5.1 给出了一个小的包，其中包含了我们之前定义的一些命令。

5.2 字体和尺寸

5.2.1 字体变换命令

\LaTeX 根据文档的逻辑结构（章节、脚注、……）来选择合适的字体和字体大小。在某些情况下，你可能会想要手工改变文档使用的字体及其大小。为了完成这个目的，你可以使用表 5.1 和表 5.2 中列出的那些命令。每个字体的实际尺寸是一个设计问题，并且它依赖于文档所使用的文档类。表 5.3 列出了这些字体变换命令在标准文档类中的绝对尺寸。

```
{\small The small and
\textbf{bold} Romans ruled}
{\Large all of great big
\textit{Italy}.}
```

The small and **bold** Romans ruled all of great big *Italy*.

\LaTeX 2_ϵ 的一个重要特征是字体的各种属性是相互独立的，这意味着你可以改变字体的大小而仍然保留字体原有的粗体或者斜体的特性。

在数学模式中你可以使用字体变换命令来暂时的退出数学模式，然后输入一些一般的文字。如果你希望改变数学公式本身所使用的字体， \LaTeX 提供了另外一套命令。参看表 5.4。

使用字体命令的时候，大括号 (curly braces) 扮演了一个重要角色。它们被用于建立所谓的组。组限制了大多数 \LaTeX 命令的作用范畴。

He likes $\{\backslash\text{\texttt{LARGE}}\text{\texttt{large}}\text{\texttt{and}}\backslash\text{\texttt{small}}\text{\texttt{small}}\}\text{\texttt{letters}}\text{\texttt{.}}$

He likes large and small letters.

如果段落在字体的作用范畴中结束，那么字体尺寸命令还将改变段落中行的距离。因此用于分组的反向大括号 } 不应该太早结束。注意随后 $\backslash\text{\texttt{par}}$ 命令的两个例子¹。

¹ $\backslash\text{\texttt{par}}$ 相当于一个空行

表 5.1: 字体

$\backslash\text{\texttt{term}}\{\dots\}$	roman	$\backslash\text{\texttt{textsf}}\{\dots\}$	sans serif
$\backslash\text{\texttt{ttt}}\{\dots\}$	typewriter		
$\backslash\text{\texttt{tmd}}\{\dots\}$	medium	$\backslash\text{\texttt{textbf}}\{\dots\}$	bold face
$\backslash\text{\texttt{tup}}\{\dots\}$	upright	$\backslash\text{\texttt{textit}}\{\dots\}$	<i>italic</i>
$\backslash\text{\texttt{tsl}}\{\dots\}$	<i>slanted</i>	$\backslash\text{\texttt{textsc}}\{\dots\}$	SMALL CAPS
$\backslash\text{\texttt{emph}}\{\dots\}$	<i>emphasized</i>	$\backslash\text{\texttt{textnormal}}\{\dots\}$	document font

表 5.2: 字体尺寸

$\backslash\text{\texttt{tiny}}$	tiny font	$\backslash\text{\texttt{Large}}$	larger font
$\backslash\text{\texttt{scriptsize}}$	very small font	$\backslash\text{\texttt{LARGE}}$	very large font
$\backslash\text{\texttt{footnotesize}}$	quite small font		
$\backslash\text{\texttt{small}}$	small font	$\backslash\text{\texttt{huge}}$	huge
$\backslash\text{\texttt{normalsize}}$	normal font		
$\backslash\text{\texttt{large}}$	large font	$\backslash\text{\texttt{Huge}}$	largest

表 5.3: 标准文档类型中字体的绝对大小

size	10pt (default)	11pt option	12pt option
<code>\tiny</code>	5pt	6pt	6pt
<code>\scriptsize</code>	7pt	8pt	8pt
<code>\footnotesize</code>	8pt	9pt	10pt
<code>\small</code>	9pt	10pt	11pt
<code>\normalsize</code>	10pt	11pt	12pt
<code>\large</code>	12pt	12pt	14pt
<code>\Large</code>	14pt	14pt	17pt
<code>\LARGE</code>	17pt	17pt	20pt
<code>\huge</code>	20pt	20pt	25pt
<code>\Huge</code>	25pt	25pt	25pt

`{\Large Don't read this! It is not true. You can believe me!}\par`

Don't read this! It is not true.
You can believe me!

`{\Large This is not true either. But remember I am a liar.}\par`

This is not true either. But remember I am a liar.

如果你希望改变整段甚至更多文本的字体，你可能应该使用变换字体的一些环境。

表 5.4: 数学字体

Command	Example	Output
<code>\mathcal{...}</code>	<code>\$_\mathcal{B}=c\$</code>	$\mathcal{B} = c$
<code>\mathrm{...}</code>	<code>\$_\mathrm{K}_2\$</code>	K_2
<code>\mathbf{...}</code>	<code>\$_\sum x=\mathbf{v}\$</code>	$\sum x = \mathbf{v}$
<code>\mathsf{...}</code>	<code>\$_\mathsf{G\times R}\$</code>	$G \times R$
<code>\mathtt{...}</code>	<code>\$_\mathtt{L}(b,c)\$</code>	$L(b, c)$
<code>\mathnormal{...}</code>	<code>\$_\mathnormal{R_{19}}\backslash neq R_{19}\$</code>	$R_{19} \neq R_{19}$
<code>\mathit{...}</code>	<code>\$_\mathit{ffi}\backslash neq ffi\$</code>	$\mathit{ffi} \neq ffi$

```
\begin{Large}
This is not true.
But then again, what is these
days \ldots
\end{Large}
```

This is not true. But then again,
what is these days ...

这将使你从众多的大括号中解脱出来。

5.2.2 Danger, Will Robinson, Danger

正如本章开头曾经说过的那样，在你的文档中运用这些明确的命令修改格式是非常危险的事情，因为这种方式 and L^AT_EX 的基础理念矛盾。在编写 L^AT_EX 文档的时候，要始终注意分离文章的逻辑结构和实际的格式。这也就意味着如果你在文章的多个地方采用了某种特殊的格式来修饰一个经常使用的信息，你就应该使用 `\newcommand` 来定义一个逻辑封装命令，并通过这个命令来修改信息的表达格式。

```
\newcommand{\oops}[1]{\textbf{#1}}
Do not \oops{enter} this room,
it's occupied by a \oops{machine}
of unknown origin and purpose.
```

Do not **enter** this room, it's occupied by a
machine of unknown origin and purpose.

这种方法具有一个明显的优点，如果你以后决定采用不同于 `\textbf` 的格式来表达危险信息，那么你不需要在你的整个文章中查找 `\textbf` 命令，也不需要将用于表达危险信息的 `\textbf` 命令与表达其他信息用到的同一个命令区分开来。

5.2.3 建议

总结这一章中关于字体和字体尺寸的命令，下面是一个简短的建议：

Remember! *The M^ORE fonts you use in a document, the more READABLE and beautiful it becomes.* 你在文章中使用的字体越多，文章看起来就越美观。

5.3 文档对象之间的间隔

5.3.1 行距

如果你想在文档中使用更大的行距，你可以在导言中使用如下命令进

行设定：

`\linespread{factor}`

例如命令 `\linespread{1.3}` 将产生 1.5 倍行距，而命令 `\linespread{1.6}` 则产生双倍行距。缺省情况下的行距为 1。

5.3.2 段落格式

在 \LaTeX 中，有两个命令可以影响段落的布局。在文档的导言部分，可以通过如下的定义来改变段落的布局。

```
\setlength{\parindent}{0pt}
\setlength{\parskip}{1ex plus 0.5ex minus 0.2ex}
```

这两个命令增加了段落之间的距离，并将首行缩进设置为 0。

例子中，长度设定中的 `plus` 和 `minus` 部分将使得 \TeX 在一个范围之内设定行距。为了使得段落正确的显示在页面之上， \TeX 将在 `0.8ex` 到 `1.5ex` 之间调整段落之间的距离。

在欧洲，段落通常用一些空格分隔并且一般不进行首行缩进。值得注意的是，上面提到的命令对于表格中的内容也会造成影响。表格的行距会变得非常疏松，这通常不是想要的结果。为了避免这种情况，你可能需要将上面的命令从导言中移到文档中适合的位置。或者根本不要使用这些，因为一般来说专业的书籍都是用缩进并且通常不用空格来分离段落。

如果你想缩进一个本来没有缩进的段落²，可以在段落的开始使用命令：

`\indent`

当然，这个命令只有在 `\parindent` 不为零的情况下才有效果。

为了创建一个不缩进的段落，你可以在段落的开始部分使用命令：

`\noindent`

²为了缩进章节标题之后的第一个段落，可以使用 `indentfirst` 包。

5.3.3 水平距离

L^AT_EX 系统自动决定单词和句子之间的距离。为了增加水平距离，使用命令：

```
\hspace{length}
```

如果这个水平距离在行首或者行末应该消失的话，用命令 `\hspace*` 代替 `\hspace`。命令的 *length* 参数在简单的情况下只是一个带有单位的数字。较为重要的长度单元在表 5.5 中列了出来。

这是 `\hspace{1.5cm}` 一段长为
1.5 厘米的空白。

这是 一段长为1.5 厘米的空白。

下面的命令将产生一个特殊的橡皮长度：

```
\stretch{n}
```

它将产生一个将一行的宽度充满的长度。如果两个 `\hspace{\stretch{n}}` 命令位于同一行，那么它们将根据伸缩因子分配空间。

```
x\hspace{\stretch{1}}  
x\hspace{\stretch{3}}x
```

x x x

5.3.4 垂直距离

在段落、节、小节……之间的距离是由 L^AT_EX 系统自动决定的。如果必要的话，可以在两段之间增加额外的距离，使用的命令如下所示：

```
\vspace{length}
```

这个命令通常用于两个空行之间。如果这个额外的行距不应该存在于页的顶部和末尾，那么使用命令 `\vspace*` 来代替 `\vspace`。

命令 `\stretch` 和 `\pagebreak` 结合使用可以在页的最后一行输出文本，也可以用来保证文本在页面上垂直居中。

Some text \ldots

```
\vspace{\stretch{1}}
```

这一行将出现在页的最后。 `\pagebreak`

表 5.5: T_EX 的长度单位

mm	millimetre $\approx 1/25$ inch	□
cm	centimetre = 10 mm	□
in	inch = 25.4 mm	□
pt	point $\approx 1/72$ inch $\approx \frac{1}{3}$ mm	□
em	approx width of an ‘M’ in the current font	□
ex	approx height of an ‘x’ in the current font	□

同一段或同一个表格中两行之间的距离可以用如下命令来指定：

`\[length]`

使用命令 `\bigskip` 和 `\smallskip` 你可以获得一个预定义的垂直距离。

5.4 页面的布局

L^AT_EX 2_ε 允许你在 `\documentclass` 命令中指定页面尺寸 (paper size)。然后它将自动的选择其它参数。但有些时候你可能不满意 L^AT_EX 的预设值，这个时候你可以自己改变这些参数。图 5.2 中显示了所有能改变的页面参数。这个图是用 layout 宏包获得的³。

先等等！... 在你开始“让这个狭窄的页面看起来宽一点”之前，先花一些时间想想。和 L^AT_EX 中的大多数规定一样，缺省的页面布局是有其内在原因的。

确实，相对于你的 MS Word 页面来说，它看上去非常的狭窄。但是看看你喜欢的书籍⁴并且统计每个标准文本行的字符数目。你会发现每行的字符不超过 66 个。现在你的 L^AT_EX 页面也是如此。经验显示，如果在一行中塞入更多的字符，阅读将变得困难。这是因为眼睛从行的开始移动到行的结束变得困难了。这也是报纸为何要排版成多栏形式的原因。

因此如果你决定增加文本的宽度，头脑中要明白你正在使你的作品变得难以阅读。警告已经说的够多了，接下来我将告诉你如何去做。

L^AT_EX 提供了两个命令来改变这些参数。他们通常在文章的导言部分使用。

³CTAN:/tex-archive/macros/latex/required/tools

⁴我说的是卓有声誉的出版商正式出版的书籍

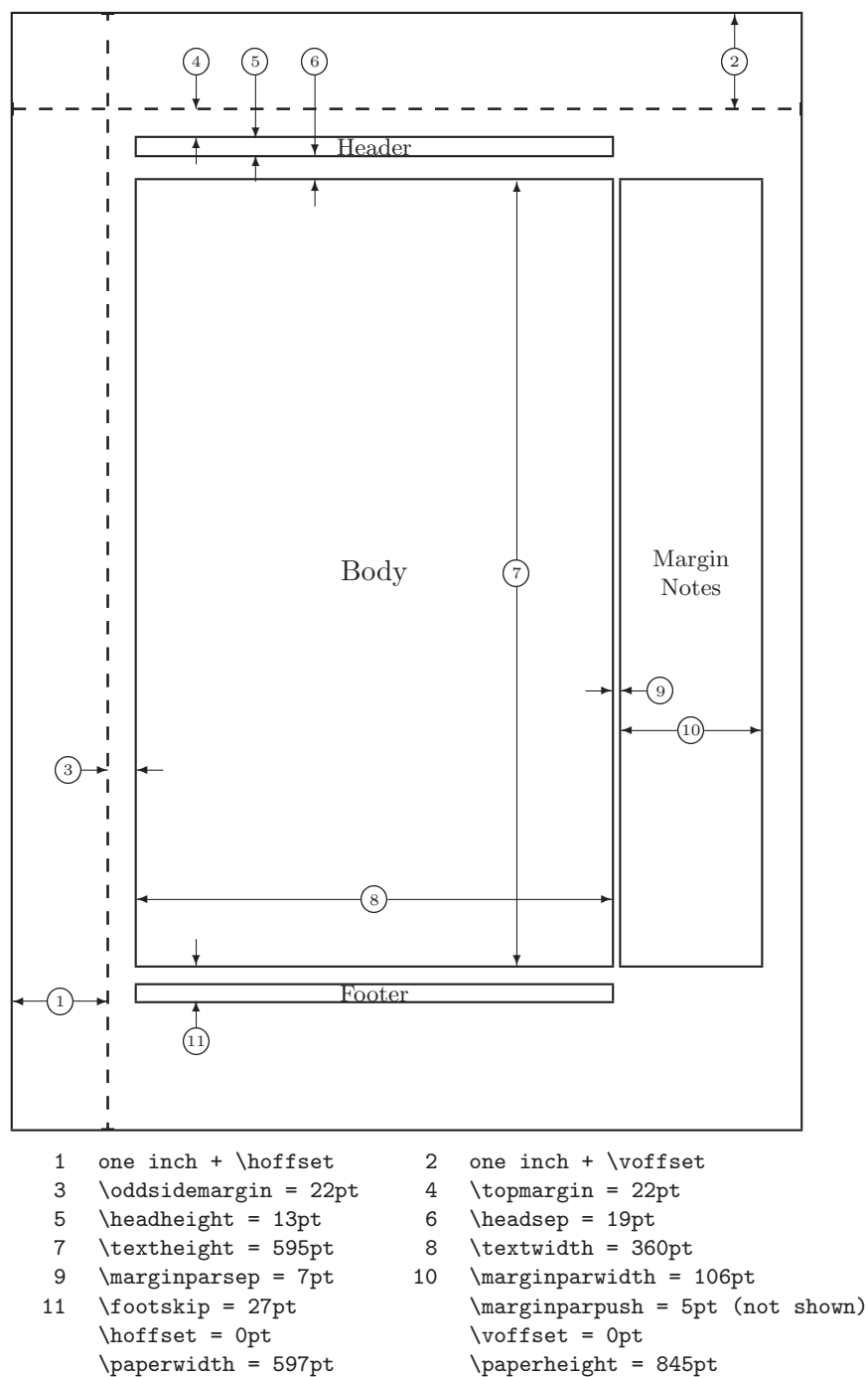


图 5.2: 页面布局参数

第一个命令给某个参数一个固定的值：

```
\setlength{parameter}{length}
```

第二个命令给某个参数增加一个固定的大小。

```
\addtolength{parameter}{length}
```

第二个命令实际上比 `\setlength` 命令更为实用，因为你可以通过修改已有的设置来获得所需的结果。为了给文本的宽度增加1厘米，我将如下的命令放置到导言部分。

```
\addtolength{\hoffset}{-0.5cm}  
\addtolength{\textwidth}{1cm}
```

在进行这个工作的时候，你可能会想要看看`calc`包，它允许你在`setlength`的参数中进行算术运算。它可以运用到任何用数值作为参数的函数中。

5.5 关于长度的更详细的信息

只要有可能，我通常都避免在 \LaTeX 文档中使用绝对长度。相反的，我一般通过页面中其它元素的宽度或高度来指定长度。比如一个图形，我指定 `\textwidth` 作为它的宽度从而使得图形恰好充满整个页面。

下面的三个命令允许你获得一个文本串的宽度、高度以及深度。

```
\settoheight{lscommand}{text}  
\settodepth{lscommand}{text}  
\settowidth{lscommand}{text}
```

下面的例子显示了这些命令的一个可能的用途：

```

\flushleft
\newenvironment{vardesc}[1]{%
  \settowidth{\parindent}{\#1:\ }
  \makebox[0pt][r]{\#1:\ }}{}

\begin{displaymath}
a^2+b^2=c^2
\end{displaymath}

\begin{vardesc}{Where}$a$,
$b$ -- are adjunct to the right
angle of a right-angled triangle.

$c$ -- is the hypotenuse of
the triangle and feels lonely.

$d$ -- finally does not show up
here at all. Isn't that puzzling?
\end{vardesc}

```

$$a^2 + b^2 = c^2$$

Where: a , b – are adjunct to the right angle of a right-angled triangle.

c – is the hypotenuse of the triangle and feels lonely.

d – finally does not show up here at all. Isn't that puzzling?

5.6 盒子

\LaTeX 通过盒子来建立整个文档的布局。首先，每个字符都是一个小的盒子，这些盒子连接起来构成单词，单词本身连接起来构成一行。值得注意的是，单词之间的连接是一个橡皮连接， \LaTeX 将自动进行调整使得单词将恰好构成一行。

我承认，这里的描述是实际情况的一个简化了许多的版本，但重点在于 \TeX 主要操作盒子和连接。不是只有一个字母才能成为一个连接，你几乎可以把任何可见元素（包括盒子自身）放到一个盒子中。然后 \LaTeX 将会像处理单个字母一样处理这个盒子。

在过去的章节中你已经遇到了一些盒子的使用，尽管我并没有明确的说出来。例如 `tabular` 环境和 `\includegraphics` 命令就都构造了一个盒子。这就意味着你可以轻松的将多个表格和图像并列排列。你唯一需要保证的就是它们宽度的总和不大于文本宽度。

使用如下命令可以把一个段落放置到盒子中：

```
\parbox[pos]{width}{text}
```

也可以用下面这个命令完成同样的事情：

```
\begin{minipage}[pos]{width} text \end{minipage}
```

参数 `pos` 可以取以下字符中的一个 `c`, `t` 或 `b`, 这个参数用于控制盒子相对环绕文本基线的垂直位置。参数 `width` 用于调整盒子的宽度。这两个环境的主要不同之处在于你可能无法在一个 `parbox` 环境中使用某些命令或者环境, 而几乎任何东西都可以在 `minipage` 环境中使用。

`\parbox` 中可以包含整个段落, 完成分行在内的几乎所有事情。与之不同, `LATEX` 中还存在另外一类盒子用于处理水平对齐的材料。我们已经知道其中的一个 – `\mbox`, 它简单的将其它盒子包含成为一个盒子, 可以用于防止 `LATEX` 断开两个单词。因为盒子中可以包含盒子, 因此它可以给予作者强大的灵活性。

```
\makebox[width][pos]{text}
```

参数 `width` 定义了生成的盒子从外部观察的宽度⁵。在进行参数设置的时候, 你也可以传递 `\width`、`\height`、`\depth` 和 `\totalheight` 给 `width` 参数。这几个值是测量盒子内部文本来获得的。参数 `pos` 接受一个字符: `c` – 居中、`l` – 靠左、`r` – 靠右和 `s` – 将文本均匀分布到整个盒子中。

命令 `\framebox` 和 `\makebox` 完成同样的工作, 不同之处在于它在内部文本的周围画出一个矩形框。

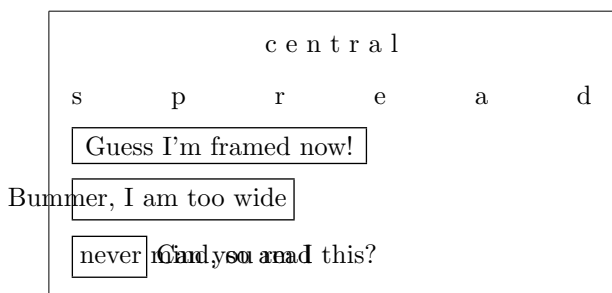
下面的例子演示了你使用命令 `\makebox` 和 `\framebox` 能完成的工作：

⁵这意味着在盒子内部看来, 盒子的宽度可能会小一些, 你甚至可以将盒子的宽度设置为 `0pt`, 这样可以使得盒子中的内容不影响盒子外部的布局。

```

\makebox[\textwidth]{%
  c e n t r a l}\par
\makebox[\textwidth][s]{%
  s p r e a d}\par
\framebox[1.1\width]{Guess I'm
  framed now!} \par
\framebox[0.8\width][r]{Bummer,
  I am too wide} \par
\framebox[1cm][l]{never
  mind, so am I}
Can you read this?

```



现在我们已经知道怎么控制盒子的水平方向长度了，接下来的步骤是学习如何控制垂直方向⁶。对于 L^AT_EX 来说，命令

```
\raisebox{lift}[depth][height]{text}
```

让你能够定义盒子的高度。你可以前面三个参数中使用 `\width`、`\height`、`\depth` 和 `\totalheight`，这样可以使得盒子的参数能够与盒子内部的文本匹配。

```

\raisebox{0pt}[0pt][0pt]{\Large%
\textbf{Aaaa}\raisebox{-0.3ex}{a}%
\raisebox{-0.7ex}{aa}%
\raisebox{-1.2ex}{r}%
\raisebox{-2.2ex}{g}%
\raisebox{-4.5ex}{h}}
he shouted but not even the next
one in line noticed that something
terrible had happened to him.

```

5.7 标尺和支撑

我们之前曾经使用过命令

```
\rule[lift]{width}{height}
```

通常它被用来输出一个黑色的矩形。

⁶全面控制仅仅是水平方向控制和垂直方向控制的同时运用 ...

```

\rule{3mm}{.1pt}%
\rule[-1mm]{5mm}{1cm}%
\rule{3mm}{.1pt}%
\rule[1mm]{1cm}{5mm}%
\rule{3mm}{.1pt}

```



这个命令可以用来产生水平方向和垂直方向的线条。例如扉页上的线条就是用一个 `\rule` 命令创建的。

一种特殊的应用就是没有宽度只有高度的标尺。在专业的出版术语中，这被称为支撑（Struts）。它被用来保证文档的一部分具有一个确定的高度最小值。你可以在 `tabular` 环境中使用支撑来使得某行具有一个特定的高度最小值。你也可以在一个环境中使用它来保证一行具有一个高度最小值。

```

\begin{tabular}{|c|}
\hline
\rule{1pt}{4ex}Pitprop \ldots\
\hline
\rule{0pt}{4ex}Strut\
\hline
\end{tabular}

```



参考文献

- [1] Leslie Lamport. *L^AT_EX: A Document Preparation System*. Addison-Wesley, Reading, Massachusetts, second edition, 1994, ISBN 0-201-52983-1.
- [2] Donald E. Knuth. *The T_EXbook*, Volume A of *Computers and Typesetting*, Addison-Wesley, Reading, Massachusetts, second edition, 1984, ISBN 0-201-13448-9.
- [3] Michel Goossens, Frank Mittelbach and Alexander Samarin. *The L^AT_EX Companion*. Addison-Wesley, Reading, Massachusetts, 1994, ISBN 0-201-54199-8.
- [4] Each L^AT_EX installation should provide a so-called *L^AT_EX Local Guide* which explains the things which are special to the local system. It should be contained in a file called `local.tex`. Unfortunately, some lazy sysops do not provide such a document. In this case, go and ask your local L^AT_EX guru for help.
- [5] L^AT_EX3 Project Team. *L^AT_EX 2_ε for authors*. Comes with the L^AT_EX 2_ε distribution as `usrguide.tex`.
- [6] L^AT_EX3 Project Team. *L^AT_EX 2_ε for Class and Package writers*. Comes with the L^AT_EX 2_ε distribution as `clsguide.tex`.
- [7] L^AT_EX3 Project Team. *L^AT_EX 2_ε Font selection*. Comes with the L^AT_EX 2_ε distribution as `fntguide.tex`.
- [8] D. P. Carlisle. *Packages in the ‘graphics’ bundle*. Comes with the ‘graphics’ bundle as `grfguide.tex`, available from the same source your L^AT_EX distribution came from.

-
- [9] Rainer Schöpf, Bernd Raichle, Chris Rowley. *A New Implementation of L^AT_EX's verbatim Environments*. Comes with the ‘tools’ bundle as `verbatim.dtx`, available from the same source your L^AT_EX distribution came from.
- [10] Graham Williams. *The TeX Catalogue* is a very complete listing of many T_EX and L^AT_EX related packages. Available online from CTAN:/tex-archive/help/Catalogue/catalogue.html
- [11] Keith Reckdahl. *Using EPS Graphics in L^AT_EX 2_ε Documents* which explains everything and much more than you ever wanted to know about EPS files and their use in L^AT_EX documents. Available online from CTAN:/tex-archive/info/epslatex.ps

索引

`\!`, 43
 ", 19
 \$, 37
`\(`, 37
`\)`, 37
`\,`, 38, 43
 -, 19
 —, 19
`\-`, 18
 —, 19
 —, 19
 ., space after, 24
 ..., 20
 .aux, 12
 .dtx, 11
 .dvi, 11
 .idx, 12
 .ilg, 13
 .ind, 13
 .ins, 11
 .lof, 12
 .log, 11
 .lot, 12
 .sty, 11
 .tex, 11
 .toc, 12
`\:`, 43
`\;`, 43
`\@`, 24

`\[`, 37
`\`, 17, 29, 30, 32, 73
`*`, 17
`\]`, 37
 ~, 24

A4 paper, 10
A5 paper, 10
acute, 21
`\addtolength`, 75
 æ, 21
 amsbsy, 49
 amsfons, 39, 56
 amsmath, 41–43, 45, 49
 amssymb, 39, 50
`\and`, 26
`\appendix`, 25, 26
`\arccos`, 41
`\arcsin`, 41
`\arctan`, 41
`\arg`, 41
 array, 44
 arrow symbols, 40
 article class, 10
`\atop`, 41
`\author`, 26

B5 paper, 10
babel, 18, 22
`\backmatter`, 26

- backslash, 5
- `\backslash`, 5
 - base font size, 10
- `\begin`, 28
- `\bibitem`, 59
- `\Big`, 42
- `\big`, 42
- `\Bigg`, 42
- `\bigg`, 42
- `\bigskip`, 73
- `\binom`, 41
 - blackboard bold, 39
 - `bm`, 49
- `\bmod`, 41
 - bold face, 68
 - bold symbols, 48
- `\boldmath`, 48
- `\boldsymbol`, 49
 - book class, 10
- braces, 42
- `calc`, 75
- `\caption`, 35, 36
- `\cdot`, 40
- `\cdots`, 43
 - center, 29
- `\chapter`, 25
- `\chaptermark`, 61
- `\choose`, 41
- `\ci`, 64
- `\cite`, 59
- `\cleardoublepage`, 36
- `\clearpage`, 36
- `\cline`, 32
 - commands, 5
 - `\!`, 43
 - `\C`, 37
 - `\D`, 37
 - `\,`, 38, 43
 - `\-`, 18
 - `\:`, 43
 - `\;`, 43
 - `\@`, 24
 - `\[`, 37
 - `\`, 17, 29, 30, 32, 73
 - `*`, 17
 - `\]`, 37
 - `\addtolength`, 75
 - `\and`, 26
 - `\appendix`, 25, 26
 - `\arccos`, 41
 - `\arcsin`, 41
 - `\arctan`, 41
 - `\arg`, 41
 - `\atop`, 41
 - `\author`, 26
 - `\backmatter`, 26
 - `\backslash`, 5
 - `\begin`, 28
 - `\bibitem`, 59
 - `\Big`, 42
 - `\big`, 42
 - `\Bigg`, 42
 - `\bigg`, 42
 - `\bigskip`, 73
 - `\binom`, 41
 - `\bmod`, 41
 - `\boldmath`, 48
 - `\boldsymbol`, 49
 - `\caption`, 35, 36
 - `\cdot`, 40
 - `\cdots`, 43
 - `\chapter`, 25

- `\chaptermark`, 61
- `\choose`, 41
- `\ci`, 64
- `\cite`, 59
- `\cleardoublepage`, 36
- `\clearpage`, 36
- `\cline`, 32
- `\cos`, 41
- `\cosh`, 41
- `\cot`, 41
- `\coth`, 41
- `\csc`, 41
- `\date`, 26
- `\ddots`, 43
- `\deg`, 41
- `\depth`, 77, 78
- `\det`, 41
- `\dim`, 41
- `\displaystyle`, 46
- `\documentclass`, 9, 17
- `\dq`, 24
- `\dum`, 64
- `\emph`, 28, 68
- `\end`, 28
- `\exp`, 41
- `\fbox`, 19
- `\footnote`, 27, 36
- `\footnotesize`, 68
- `\frac`, 41
- `\framebox`, 77
- `\frenchspacing`, 25
- `\frontmatter`, 26
- `\fussy`, 18
- `\gcd`, 41
- `\genfrac`, 41
- `\height`, 77, 78
- `\hline`, 32
- `\hom`, 41
- `\hspace`, 66, 72
- `\Huge`, 68
- `\huge`, 68
- `\hyphenation`, 18
- `\idotsint`, 43
- `\iiiint`, 43
- `\iiint`, 43
- `\iint`, 43
- `\include`, 13, 14
- `\includegraphics`, 58, 76
- `\includeonly`, 14
- `\indent`, 71
- `\index`, 60
- `\inf`, 41
- `\input`, 14
- `\int`, 42
- `\item`, 29
- `\ker`, 41
- `\label`, 27, 38
- `\LARGE`, 68
- `\Large`, 68
- `\large`, 68
- `\LaTeX`, 19
- `\LaTeXe`, 19
- `\ldots`, 20, 43
- `\left`, 42
- `\leftmark`, 61
- `\lg`, 41
- `\lim`, 41
- `\liminf`, 41
- `\limsup`, 41
- `\linebreak`, 17
- `\linespread`, 71
- `\listoffigures`, 35

\listoftables, 35
\ln, 41
\log, 41
\mainmatter, 26
\makebox, 77
\makeindex, 60
\maketitle, 26
\mathbb, 39
\mathbf, 69
\mathcal, 69
\mathit, 69
\mathnormal, 69
\mathrm, 46, 69
\mathsf, 69
\mathtt, 69
\max, 41
\mbox, 18, 19, 21, 77
\min, 41
\multicolumn, 33
\newcommand, 65, 66
\newenvironment, 66
\newline, 17
\newpage, 17
\newtheorem, 47
\noindent, 71
\nolinebreak, 17
\nonumber, 45
\nopagebreak, 17
\normalsize, 68
\overbrace, 40
\overleftarrow, 40
\overline, 40
\overrightarrow, 40
\pagebreak, 17
\pageref, 27
\pagestyle, 13
\par, 68
\paragraph, 25
\parbox, 77
\parindent, 71
\parskip, 71
\part, 25
\phantom, 36, 46
\pmod, 41
\Pr, 41
\printindex, 61
\prod, 42
\protect, 36
\providecommand, 66
\ProvidesPackage, 67
\qqquad, 38, 43
\quad, 38, 43
\raisebox, 78
\ref, 27, 38
\renewcommand, 65
\renewenvironment, 66
\right, 42
\right., 42
\rightmark, 61
\rule, 66, 78, 79
\scriptscriptstyle, 46
\scriptsize, 68
\scriptstyle, 46
\sec, 41
\section, 25, 36
\sectionmark, 61
\setlength, 71, 75
\settodepth, 75
\settoheight, 75
\settowidth, 75
\sin, 41
\sinh, 41

`\sloppy`, 17
`\small`, 68
`\smallskip`, 73
`\sqrt`, 40
`\stackrel`, 41
`\stretch`, 66, 72
`\subparagraph`, 25
`\subsection`, 25
`\subsectionmark`, 61
`\subsubsection`, 25
`\sum`, 42
`\sup`, 41
`\tableofcontents`, 25
`\tan`, 41
`\tanh`, 41
`\TeX`, 19
`\textbf`, 68
`\textit`, 68
`\textmd`, 68
`\textnormal`, 68
`\textrm`, 46, 68
`\textsc`, 68
`\textsf`, 68
`\textsl`, 68
`\textstyle`, 46
`\texttt`, 68
`\textup`, 68
`\thispagestyle`, 13
`\tiny`, 68
`\title`, 26
`\tnss`, 65
`\today`, 19
`\totalheight`, 77, 78
`\underbrace`, 40
`\underline`, 28, 40
`\usepackage`, 11, 22, 23, 67
`\vdots`, 43
`\vec`, 40
`\verb`, 31
`\verbatiminput`, 62
`\vspace`, 72
`\widehat`, 40
`\widetilde`, 40
`\width`, 77, 78
comment, 6
comments, 6
`\cos`, 41
`\cosh`, 41
`\cot`, 41
`\coth`, 41
`\csc`, 41
curly braces, 6, 68
`\date`, 26
dcolumn, 32
`\ddots`, 43
`\deg`, 41
delimiters, 42
`\depth`, 77, 78
description, 29
`\det`, 41
Deutsch, 24
diagonal dots, 43
`\dim`, 41
dimensions, 72
`\displaymath`, 37
`\displaystyle`, 46
doc, 12
document font size, 10
document title, 10
`\documentclass`, 9, 17
dotless i and j, 21
double line spacing, 71

- `\dq`, 24
- `\dum`, 64
- `dvips`, 9
- `\emph`, 28, 68
- `empty`, 13
- Encapsulated PostScript, 57
- `\end`, 28
- `enumerate`, 29
- environments
 - `array`, 44
 - `center`, 29
 - `comment`, 6
 - `description`, 29
 - `displaymath`, 37
 - `enumerate`, 29
 - `eqnarray`, 45
 - `equation`, 37
 - `figure`, 34, 35
 - `flushleft`, 29
 - `flushright`, 29
 - `itemize`, 29
 - `lscommand`, 64
 - `math`, 37
 - `minipage`, 77
 - `parbox`, 77
 - `quotation`, 30
 - `quote`, 30
 - `table`, 34, 35
 - `tabular`, 31, 76
 - `thebibliography`, 59
 - `verbatim`, 30, 61, 62
 - `verse`, 30
- `eqnarray`, 45
- `equation`, 37
- equation system, 45
- `eucal`, 56
- `eufrak`, 56
- executive paper, 10
- `\exp`, 41
- exponent, 39
- `exscale`, 12, 42
- extension, 11
- `fancyhdr`, 61, 62
- `\fbox`, 19
- `figure`, 34, 35
- `flushleft`, 29
- `flushright`, 29
- `foiltex`, 10
- font, 67
- font encoding, 12
- font size, 67, 68
- `fontenc`, 12, 23
- footer, 13
- `\footnote`, 27, 36
- `\footnotesize`, 68
- formulae, 37
- `\frac`, 41
- fraction, 41
- `\framebox`, 77
- `\frenchspacing`, 25
- `\frontmatter`, 26
- `\fussy`, 18
- `\gcd`, 41
- `\genfrac`, 41
- geometry, 62
- German, 22
- GhostScript, 57
- `graphicx`, 57
- grave, 21
- Greek letters, 39
- grouping, 68

- header, 13
- texttttheadings, 13
- \height, 77, 78
- \hline, 32
- \hom, 41
- horizontal
 - brace, 40
 - line, 40
 - space, 72
- \hspace, 66, 72
- \Huge, 68
- \huge, 68
- hyphenat, 62
- \hyphenation, 18
- \idotsint, 43
- ifthen, 12
- \iiiint, 43
- \iiint, 43
- \iint, 43
- \include, 13, 14
- \includegraphics, 58, 76
- \includeonly, 14
- \indent, 71
- indentfirst, 71
- \index, 60
- \inf, 41
- \input, 14
- input file, 7
- inputenc, 12, 22
- \int, 42
- integral operator, 42
- international, 21
- italic, 68
- \item, 29
- itemize, 29
- \ker, 41
- Knuth, Donald E., 1
- \label, 27, 38
- Lamport, Leslie, 1
- \LARGE, 68
- \Large, 68
- \large, 68
- \LaTeX, 19
- LaTeX 2.09, 1
- LaTeX 2_ε, 1
- LaTeX3, 1
- \LaTeXe, 19
- latexsym, 12
- layout, 73
- \ldots, 20, 43
- \left, 42
- left aligned, 29
- \leftmark, 61
- legal paper, 10
- letter paper, 10
- \lg, 41
- \lim, 41
- \liminf, 41
- \limsup, 41
- line spacing, 70
- linebreak, 17
- \linebreak, 17
- \linespread, 71
- \listoffigures, 35
- \listoftables, 35
- \ln, 41
- \log, 41
- long quotations, 45
- longtabular, 33
- lscommand, 64

- `\mainmatter`, 26
- `\makebox`, 77
 - `makeidx`, 12, 60
 - `makeidx` package, 60
- `\makeindex`, 60
 - `makeindex` program, 60
- `\maketitle`, 26
 - `math`, 37
 - `math` font size, 46
- `\mathbb`, 39
- `\mathbf`, 69
- `\mathcal`, 69
 - mathematical
 - accents, 40
 - functions, 41
 - minus, 19
 - mathematics, 37
- `\mathit`, 69
- `\mathnormal`, 69
- `\mathrm`, 46, 69
 - `mathrsfs`, 56
- `\mathsf`, 69
- `\mathtt`, 69
- `\max`, 41
- `\mbox`, 18, 19, 21, 77
- `\min`, 41
 - `minipage`, 77
 - Mittelbach, Frank, 1
 - modulo function, 41
- `\multicolumn`, 33
- `\newcommand`, 65, 66
- `\newenvironment`, 66
- `\newline`, 17
- `\newpage`, 17
- `\newtheorem`, 47
- `\noindent`, 71
- `\nolinebreak`, 17
- `\nonumber`, 45
- `\nopagebreak`, 17
- `\normalsize`, 68
 - œ, 21
 - one column, 10
 - option, 9
- `\overbrace`, 40
 - overfull hbox, 17
- `\overleftarrow`, 40
- `\overline`, 40
- `\overrightarrow`, 40
 - package, 11, 64
 - packages
 - `amssbsy`, 49
 - `amsfonts`, 39, 56
 - `amsmath`, 41–43, 45, 49
 - `amssymb`, 39, 50
 - `babel`, 18, 22
 - `bm`, 49
 - `calc`, 75
 - `dcolumn`, 32
 - `doc`, 12
 - `eucal`, 56
 - `eufrak`, 56
 - `exscale`, 12, 42
 - `fancyhdr`, 61, 62
 - `fontenc`, 12, 23
 - `geometry`, 62
 - `graphicx`, 57
 - `hyphenat`, 62
 - `ifthen`, 12
 - `indentfirst`, 71
 - `inputenc`, 12, 22
 - `latexsym`, 12

- layout, 73
- longtabular, 33
- makeidx, 12, 60
- mathrsfs, 56
- showidx, 61
- supertabular, 33
- syntonly, 12, 14
- verbatim, 6, 61, 62
- page layout, 73
- page style, 13
 - empty, 13
 - headings, 13
 - plain, 13
- \pagebreak, 17
- \pageref, 27
- \pagestyle, 13
- paper size, 10, 73
- \par, 68
- \paragraph, 25
 - parameter, 6
- \parbox, 77
 - parbox, 77
- \parindent, 71
- \parskip, 71
- \part, 25
- \phantom, 36, 46
 - plain, 13
- \pmod, 41
 - PostScript, 57
- \Pr, 41
 - prime, 40
- \printindex, 61
- \prod, 42
 - product operator, 42
- \protect, 36
- \providecommand, 66
- \ProvidesPackage, 67
- \qqquad, 38, 43
- \quad, 38, 43
 - quotation, 30
 - quote, 30
- \raisebox, 78
- \ref, 27, 38
- \renewcommand, 65
- \renewenvironment, 66
 - report class, 10
- \right, 42
- \right., 42
- \rightmark, 61
 - roman, 68
- \rule, 66, 78, 79
 - sans serif, 68
 - Scandinavian letters, 21
- \scriptscriptstyle, 46
- \scriptsize, 68
- \scriptstyle, 46
- \sec, 41
- \section, 25, 36
- \sectionmark, 61
- \setlength, 71, 75
- \settodepth, 75
- \settoheight, 75
- \settowidth, 75
 - showidx, 61
- \sin, 41
- \sinh, 41
 - slanted, 68
 - slides class, 10
- \sloppy, 17
- \small, 68

- small caps, 68
- `\smallskip`, 73
- space, 5
- `\sqrt`, 40
- square brace, 6
- square root, 40
- `\stackrel`, 41
- `\stretch`, 66, 72
- Struts, 79
- `\subparagraph`, 25
- subscript, 39
- `\subsection`, 25
- `\subsectionmark`, 61
- `\subsubsection`, 25
- `\sum`, 42
- sum operator, 42
- `\sup`, 41
- supertabular, 33
- syntonly, 12, 14
- table, 34, 35
- `\tableofcontents`, 25
- tabular, 31, 76
- `\tan`, 41
- `\tanh`, 41
- `\TeX`, 19
- `\textbf`, 68
- `\textit`, 68
- `\textmd`, 68
- `\textnormal`, 68
- `\textrm`, 46, 68
- `\textsc`, 68
- `\textsf`, 68
- `\textsl`, 68
- `\textstyle`, 46
- `\texttt`, 68
- `\textup`, 68
- `thebibliography`, 59
- `\thispagestyle`, 13
- three dots, 43
- tilde, 20, 40
- tilde (\sim), 24
- `\tiny`, 68
- title, 10
- `\title`, 26
- `\tnss`, 65
- `\today`, 19
- `\totalheight`, 77, 78
- two column, 10
- umlaut, 21
- `\underbrace`, 40
- underfull hbox, 18
- `\underline`, 28, 40
- units, 72, 73
- upright, 68
- URL, 20
- `\usepackage`, 11, 22, 23, 67
- `\vdots`, 43
- `\vec`, 40
- vectors, 40
- `\verb`, 31
- verbatim, 6, 61, 62
- verbatim, 30, 61, 62
- `\verbatiminput`, 62
- verse, 30
- vertical space, 72
- `\vspace`, 72
- whitespace
 - after commands, 6
 - at the start of a line, 5
- `\widehat`, 40

\widetilde, 40

\width, 77, 78

www, 20

WYSIWYG, 3

xdvi, 9

参考文献, 59

长破折号, 20

脆弱命令, 36

德语, 24

逗号, 20

度的符号, 20

短破折号, 20

断行点, 17

放置说明符, 34

浮动体, 33

减号, 20

句号, 20

靠右排列, 29

靠左排列, 29

连字号, 20

目录, 25

省略号, 20

索引, 60

所见即所得, 3

特殊字符, 21

图形, 57

引号, 19

语言, 21

注音符, 21

