

一份不太简短的 L^AT_EX 2_ε 介绍

或 *112* 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 4.20, May 31, 2006

中文翻译: 中文 T_EX 学会

中文版本: 版本 4.20, 二零零七年九月

Tobias Oetiker 及贡献者拥有版权 © 1995 – 2005。保留所有权利。

这份文档是免费的；在 Free Software Foundation 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 Free Software Foundation，地址：675 Mass Ave, Cambridge, MA 02139, USA。

致谢!

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@bmf.ac.at>
Bundesministerium für Wissenschaft und Forschung Wien

Elisabeth Schlegl <noemail>
in Graz

如果你对德文文档有兴趣, 有一个由 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

下列人士为改进此文提供了校正、建议和素材。他们的不懈努力帮助我把这份文档实现为现在这样子。我对他们所有人表示诚挚的感谢。当然，你在本书中找到的所有错误都是我的失误。而你见到的每一个拼写正确的单词，都一定是由于下面列出的这些人之一通知了我。

Rosemary Bailey, Marc Bevand, Friedemann Brauer, Jan Busa, Markus Brühwiler, Pietro Braione, David Carlisle, José Carlos Santos, Neil Carter, Mike Chapman, Pierre Chardaire, Christopher Chin, Carl Cerecke, Chris McCormack, Wim van Dam, Jan Dittberner, Michael John Downes, Matthias Dreier, David Dureisseix, Elliot, Hans Ehrbar, Daniel Flipo, David Frey, Hans Fugal, Robin Fairbairns, Jörg Fischer, Erik Frisk, Mic Milic Frederickx, Frank, Kasper B. Graversen, Arlo Griffiths, Alexandre Guimond, Andy Goth, Cyril Goutte, Greg Gamble, Frank Fischli, Morten Høgholm, 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, Sander de Kievit, Alain Kessi, Christian Kern, Tobias Klauser, Jörg Knappen, Kjetil Kjernsmo, Maik Lehradt, Rémi Letot, Flori Lambrechts, Axel Liljencrantz, Johan Lundberg, Alexander Mai, Hendrik Maryns, Martin Maechler, Aleksandar S Milosevic, Henrik Mitsch, Claus Malten, Kevin Van Maren, Richard Nagy, Philipp Nagele, Lenimar Nunes de Andrade, Manuel Oetiker, Urs Oswald, Martin Pfister, Demerson Andre Polli, Nikos Pothitos, Maksym Polyakov Hubert Partl, John Reffing, Mike Ressler, Brian Ripley, Young U. Ryu, Bernd Rosenlecher, Chris Rowley, Risto Saarelma, Hanspeter Schmid, Craig Schlenter, Gilles Schintgen, Baron Schwartz, Christopher Sawtell, Miles Spielberg, Geoffrey Swindale, Laszlo Szathmary, Boris Tobotras, Josef Tkadlec, Scott Veirs, Didier Verna, Fabian Wernli, Carl-Gustav Werner, David Woodhouse, Chris York, Fritz Zaucker, Rick Zaccone, and Mikhail Zotov.

4.20 中文版致謝!

中文 $\text{T}_{\text{E}}\text{X}$ 學會啓動的 `lshort-zh-cn` 修正計劃已經完工! 本項計劃歷時八個月, 參加的朋友有:

$\text{C}_{\text{T}}\text{E}_{\text{X}}$ 論壇 ID	執行章節
zpxing	前言、第二章、第五章 1-2.4 & 3、第六章
Frogge	第一章
liwenjun	第三章
lijian605	第四章
gprsnl	第五章 2.5-2.11

haginile 和 Frogge 通讀了全篇, 并寫出了詳盡的勘誤表。blackold 對於第二章亦有所貢獻。最后由 zpxing 統籌全書。

.....

原 3.20 中文版致謝!

本文档的翻译工作由 $\text{C}_{\text{T}}\text{E}_{\text{X}}$ 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:

$\text{C}_{\text{T}}\text{E}_{\text{X}}$ 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

\LaTeX [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信件到完整书籍的所有其他种类的文档。 \LaTeX 使用 \TeX [2] 作为它的格式化引擎。

这份短小的介绍描述了 $\text{\LaTeX} 2_{\epsilon}$ 的使用，对 \LaTeX 的大多数应用来说应该是足够了。参考文献 [1, 3] 对 \LaTeX 系统提供了完整的描述。

这份介绍共有六章：

第一章 告诉你关于 $\text{\LaTeX} 2_{\epsilon}$ 文档的基本结构。你也会从中了解一点 \LaTeX 的历史。阅读这一章后，你应该对 \LaTeX 如何工作有一个大致的理解。

第二章 探究文档排版的细节。它解释了大部分必要的 \LaTeX 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 \LaTeX 排版公式。同时，大量的例子会有助于你理解 \LaTeX 是如何的强大。在这个章节的结尾，你会找到列出 \LaTeX 中所有可用数学符号的表格。

第四章 解释了索引和参考文件的生成、EPS 图形的插入。它介绍了如何使用 pdf \LaTeX 生成 pdf 文档和一些其他有用的扩展宏包。

第五章 演示如何使用 \LaTeX 创建图形。不必使用图形软件画图、存盘并插入 \LaTeX 文档，你可以直接描述图形，然后 \LaTeX 会替你画好它。

第六章 包含一些潜在的危险信息，内容是关于如何改变 \LaTeX 所产生文档的标准布局。它会告诉你如何把 \LaTeX 的输出变得更糟糕，或者更上一层楼，当然这取决于你的能力。

按照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，因为在贯穿全篇的各种例子里包含了很多的信息。

\LaTeX 适用于从 PC 和 Mac 到大型的 UNIX 和 VMS 系统上。许多大学的计算机集群上安装了 \LaTeX ，随时可以使用。*Local Guide* [5] 里应该会介绍如何使用本地安装的 \LaTeX 。如果有问题，就去问给你这本小册子的人。这份文档不会告诉你如何安装一个 \LaTeX 系统，而是教会你编写 \LaTeX 能够处理的文档。

如果你想取得 \LaTeX 的相关材料，请访问“Comprehensive \TeX Archive Network” (CTAN) 站点，主页是 <http://www.ctan.org>。所有的宏包也可以从 ftp 归档站点 <ftp://www.ctan.org> 和遍布全球的各个镜像站点中获得。所有的宏包都可以在 <ftp://ctan.tug.org> 以及它遍布全球的镜像取得。

在本书中你会找到其他引用 CTAN 的地方，尤其是，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`，而仅仅是其在 CTAN: 之后的树状结构中的位置。

请先看看 CTAN:/tex-archive/systems 中有些什么，如果你想在自己的计算机上运行 L^AT_EX。

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 L^AT_EX 初学者的反馈特别感兴趣，尤其是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 Information Technology and
Electrical Engineering,
Swiss Federal Institute of Technology

这份文档的最新版本在
CTAN:/tex-archive/info/lshort

关于这份文档的最新中文翻译，请咨询
<http://bbs.ctex.org>

目录

致谢!	iii
前言	vii
1 基础知识	1
1.1 游戏的名目	1
1.1.1 TeX	1
1.1.2 L ^A T _E X	1
1.2 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	2
1.2.3 优势和不足	2
1.3 L ^A T _E X 源文件	3
1.3.1 空白距离	3
1.3.2 特殊字符	4
1.3.3 L ^A T _E X 命令	4
1.3.4 注释	5
1.4 源文件的结构	5
1.5 一个典型的命令行过程	7
1.6 文档布局	8
1.6.1 文档类	8
1.6.2 宏包	8
1.6.3 页面样式	10
1.7 各类L ^A T _E X 文件	11
1.8 大型项目	12
2 文本排版	13
2.1 文本和语言结构	13
2.2 断行和分页	14
2.2.1 对齐段落	14
2.2.2 断词	15
2.3 内置字符串	16
2.4 特殊字符和符号	16
2.4.1 引号	16
2.4.2 破折号和连字号	17
2.4.3 波浪号(~)	17
2.4.4 度的符号(°)	17
2.4.5 欧元符号(€)	17

2.4.6	省略号(...)	18
2.4.7	连字	18
2.4.8	注音符号和特殊字符	19
2.5	国际语言支持	19
2.5.1	葡萄牙文支持	21
2.5.2	法文支持	22
2.5.3	德文支持	22
2.5.4	朝鲜文支持	23
2.5.5	用希腊文写作	24
2.5.6	斯拉夫文支持	25
2.6	单词间隔	26
2.7	标题、章和节	27
2.8	交叉引用	28
2.9	脚注	29
2.10	强调	29
2.11	环境	30
2.11.1	Itemize、Enumerate 和 Description	30
2.11.2	左对齐、右对齐和居中	30
2.11.3	引用、语录和韵文	31
2.11.4	摘要	31
2.11.5	原文打印	32
2.11.6	表格	32
2.12	浮动体	34
2.13	保护脆弱命令	36
3	数学公式	37
3.1	综述	37
3.2	数学模式的群组	39
3.3	数学公式的基本元素	39
3.4	数学空格	42
3.5	垂直取齐	43
3.6	虚位	45
3.7	数学字体尺寸	45
3.8	定理、定律.....	46
3.9	粗体符号	47
3.10	数学符号表	48
4	专业功能	55
4.1	插入 EPS 图形	55
4.2	参考文献	56
4.3	索引	57
4.4	定制页眉和页脚	58
4.5	Verbatim 宏包	59
4.6	安装额外的宏包	59
4.7	使用 pdfL ^A T _E X	60
4.7.1	发布到网上的 PDF 文档	61
4.7.2	字体	61
4.7.3	使用图形	62
4.7.4	超链接	63
4.7.5	链接的问题	65

4.7.6 书签的问题	65
4.8 创建演示文稿	66
5 数学图形	69
5.1 概述	69
5.2 <code>picture</code> 环境	69
5.2.1 基本命令	70
5.2.2 线段	71
5.2.3 箭头	72
5.2.4 圆	73
5.2.5 文本与公式	74
5.2.6 <code>\multiput</code> 与 <code>\linethickness</code>	74
5.2.7 椭圆	75
5.2.8 重复使用预定义的图形盒子	76
5.2.9 二次 Bézier 曲线	77
5.2.10 悬链线	78
5.2.11 坐标的相对性	79
5.3 <code>Xy-pic</code>	79
6 定制 L^AT_EX	83
6.1 新建命令、环境和宏包	83
6.1.1 新建命令	84
6.1.2 新建环境	84
6.1.3 额外的空白间距	85
6.1.4 命令行的 L ^A T _E X	85
6.1.5 自建宏包	86
6.2 字体和字号	86
6.2.1 字体变换命令	86
6.2.2 战战兢兢，如履薄冰	89
6.2.3 建议	89
6.3 间距	90
6.3.1 行距	90
6.3.2 段落格式	90
6.3.3 水平间距	91
6.3.4 垂直间距	92
6.4 页面布局	92
6.5 更有趣的长度	94
6.6 盒子	95
6.7 标尺和支撑	97
参考文献	99
索引	101

图形清单

1.1 一个简单的L ^A T _E X 源文件。	6
1.2 article 类例子。	6
4.1 fancyhdr 设置实例。	59
4.2 beamer 文档类的范例。	67
6.1 宏包样例。	86
6.2 页面布局参数。	93

表格清单

1.1	文档类。	8
1.2	文档类选项。	9
1.3	随 \LaTeX 一起发行的宏包。	10
1.4	\LaTeX 预定义的页面样式。	10
2.1	欧元符号工具箱。	18
2.2	注音符号和特殊字符。	19
2.3	葡萄牙文所需的导言区。	21
2.4	法文专用命令。	22
2.5	德文专用字符。	23
2.6	希腊文文档所需导言区。	25
2.7	希腊文特殊字符。	25
2.8	保加利亚文、俄文和乌克兰文。	26
2.9	浮动体放置许可。	34
3.1	数学模式重音符号。	48
3.2	希腊字母。	48
3.3	二元关系。	49
3.4	二元运算符。	49
3.5	“大”运算符。	49
3.6	箭头。	50
3.7	定界符。	50
3.8	大定界符。	50
3.9	其他符号。	51
3.10	非数学符号。	51
3.11	AMS 定界符。	51
3.12	AMS 希腊和希伯来字母。	51
3.13	AMS 二元关系。	52
3.14	AMS 箭头。	52
3.15	AMS 二元否定关系符和箭头。	53
3.16	AMS 二元运算符。	53
3.17	AMS 其他符号。	54
3.18	数学字母。	54
4.1	<code>graphicx</code> 宏包使用的关键词。	56
4.2	索引关键词语法示例。	58
6.1	字体。	87

6.2	字号。	87
6.3	标准文档类中的绝对 pt 大小。	88
6.4	数学字体。	88
6.5	TeX 单位。	92

Chapter 1

基础知识

本章的第一部分给出了 $\text{\LaTeX} 2_{\epsilon}$ 原理及历史的简短介绍。第二部分集中讲解 \LaTeX 文档的基本结构。读完本章之后，你应该大致了解 \LaTeX 的工作原理，这对你理解本书的其余部分来说是必须的。

1.1 游戏的名目

1.1.1 \TeX

\TeX 是 Donald E. Knuth 编写的一个以排版文章及数学公式为目标的计算机程序[2]。1977 年，在意识到恶劣的排版质量正在影响自己的著作及文章后，Knuth 开始编写 \TeX 排版系统引擎，探索当时开始进入出版工业的数字印刷设备的潜力，尤为希望能扭转排版质量下滑的这一趋势。我们现在使用的 \TeX 系统发布于 1982 年，在 1989 年又稍做改进，增加了对 8 字节字符及多语言的支持。 \TeX 以其卓越的稳定性、可在不同类型的电脑上运行以及几乎没有缺陷而著称。 \TeX 的版本号不断趋近于 π ，现在为 3.141592。

\TeX 发音为“Tech”，其中“ch”和德语“Ach”¹ 及苏格兰语“Loch”中的“ch”类似。“ch”源自希腊字母，希腊文中，X 是字母“ch”或“chi”。 \TeX 同时也是希腊单词 *technologia* (technology) 的第一个音节。在 ASCII 文本环境中， \TeX 写作 TeX。

1.1.2 \LaTeX

\LaTeX 是一个宏集，它使用一个预先定义好的专业版面，可以使作者们高质量的排版和打印他们的作品。 \LaTeX 最初由 Leslie Lamport 编写[1]，它使用 \TeX 程序作为排版引擎。现在 \LaTeX 由 Frank Mittelbach 负责维护。

\LaTeX 的发音为“Lay-tech”或“Lah-tech”。如果在 ASCII 环境中引用 \LaTeX ，你可以输入 LaTeX。 $\text{\LaTeX} 2_{\epsilon}$ 的发音为“Lay-tech two e”，在 ASCII 环境中写作 LaTeX2e。

¹在德语中，“ch”有两种发音，有的人可能认为“Pech”中较软的“ch”更加合适。被问及这个问题时，Knuth 在德文 Wikipedia 中写道：当人们以他们喜欢的方式来拼读 \TeX 时，我并不感到生气……在德国，更多的人喜欢较软的 ch，因为 X 跟在元音 e 的后面。在俄语中，‘tex’是一个非常普遍的单词，读作‘tyekh’。但我相信最合适的发音来自希腊语，其中 ach 和 Loch 中 ch 的发音稍尖。

1.2 基础

1.2.1 作者、图书设计者和排版者

出版的第一步就是作者把打好字的手稿交给出版公司，然后由图书设计者来决定整个文档的布局（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作者的手稿里，再交给排版者，由排版者根据这些说明来排版全书。

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来决定章节标题、文献引用、例子及公式等等。

在一个 \LaTeX 环境中， \LaTeX 充当了图书设计者的角色，而 \TeX 则是其排版者。但是 \LaTeX “仅仅”是一个程序，因此它需要很多的指导。作者必须提供额外的信息，来描述其著作的逻辑结构。这些信息是以“ \LaTeX 命令”的形式写入文档中的。

这和大多数现代文字处理工具，如 $MS\ Word$ 及 $Corel\ WordPerfect$ 所采用的所见即所得(WYSIWYG²)的方式有很大区别。使用这些工具时，作者在向计算机中输入文档的同时，通过互动的方式确定文章的布局。作者可以从屏幕上看到作品的最终打印效果。

而使用 \LaTeX 时，一般是不能在输入文档的同时看到最终的输出效果的，但是使用 \LaTeX 处理文档之后，便可以在屏幕上预览最终的输出效果。因此在真正打印文档之前还是可以做出改正的。

1.2.2 版面设计

排版设计是一门工艺。不熟练的作者认为书籍设计仅仅是个美学问题，因而经常会犯严重的格式错误——“如果一份文档从艺术的角度看起来不错，那么它的设计就是成功的”。不过作为一份用来阅读而不是挂在画廊里的文档，可读性和可理解性远比漂亮的外观重要。例如：

- 必须选定字号和标题的序号，使读者能清楚的理解章节的结构。
- 每一行既要足够短以避免读者眼睛疲劳，又要足够长以维持页面的美观。

在使用所见即所得系统(WYSIWYG)时，作者经常会写出一些看上去漂亮，但结构欠清晰或不连贯的文章来。 \LaTeX 通过强制作者声明文档的逻辑结构，来避免这些排版格式错误。然后， \LaTeX 再根据文档的结构选择最合适的版面格式。

1.2.3 优势和不足

使用所见即所得(WYSIWYG)的人和使用 \LaTeX 的人遇到一起时，他们经常讨论的话题就是“相比一般文字处理软件， \LaTeX 的优势(advantages of \LaTeX)”或者不足。当这样的讨论开始时，你最好保持低调，因为讨论往往会失控。但有时你也不能逃避……

下面便是一些武器。 \LaTeX 优于一般文字处理软件之处可归纳如下：

- 提供专业的版面设计，可以使一份文档看起来就像“印刷品”一样。
- 可以方便的排版数学公式。
- 用户只需要学一些声明文档逻辑结构的简单易懂的命令，而不必对文档的实际版面修修补补。

²What you see is what you get.

- 可以容易的生成像脚注、引用、目录和参考文献等很多复杂的结构。
- 很多不被基本L^AT_EX支持的排版工作，可以由添加免费的宏包来完成。例如，支持在文件中插入POSTSCRIPT 格式图像的宏包及排版符合各类准确标准的参考文献的宏包等。很多这类宏包在*The L^AT_EX Companion* [3]中都有说明。
- L^AT_EX 鼓励作者按照合理的结构写作，因为L^AT_EX 就是通过指明文档结构来进行排版工作的。
- T_EX，作为L^AT_EX 2_ε的排版引擎，不仅免费，而且具有很高的可移植性，几乎可以在任何硬件平台上运行。

L^AT_EX 也有一些不足之处。尽管我可以确定别人可以列出几百条，我自己却很难找到一些比较理智的;-)

- 没有原则的人不能使用L^AT_EX 很好地工作……
- 尽管可以调节预先定义好的文档版面布局中的一些参数，但设计一个全新的版面还是很困难的，并会耗费大量时间³。
- 很难用L^AT_EX 来写结构不明、组织无序的文档。
- 即使有一个令人鼓舞的开端，你也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 源文件不仅包含了要排版的文本，而且也包含了告诉L^AT_EX 如何排版这些文本内容的命令。

1.3.1 空白距离

空格和制表符等空白字符在L^AT_EX 中被看作相同的空白距离(space)。多个连续的空白字符等同于一个空白字符。在句首的空白距离一般会被忽略，单个空行也被认为是一个“空白距离”。

两行文本间的空白行标志着上段的结束和下段的开始。多个空白行的作用等同于一个空白行。下面便是一个例子，左边是源文件中的文本，右边是排版后的结果。

```
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.

³传闻这将是未来的L^AT_EX3 系统中的一个重要组成部分。

1.3.2 特殊字符

下面的这些字符是 \LaTeX 中的保留字符(reserved characters), 它们或在 \LaTeX 中有特殊的意义, 或不一定存在于所有字库中。如果你直接在文本中输入这些字符, 通常它们不会被输出, 而且还会导致 \LaTeX 做一些你不希望发生的事情。

\$ % ^ & _ { } \

如你看到的, 在这些字符前加上反斜线, 它们就可以正常的输出到文档中。

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

\$ % ^ & _ { }

其他一些特殊符号可以由数学环境中的特殊命令或重音命令得到。反斜线 \ 不能通过在其前面加另一个反斜线得到(\\); 这是一个用来换行的命令⁴。

1.3.3 \LaTeX 命令

\LaTeX 命令(commands) 是大小写敏感的, 有以下两种格式:

- 以一个反斜线(backslash) \ 开始, 命令名只由字母组成。命令名后的空格符、数字或任何非字母的字符都标志着该命令的结束。
- 由一个反斜线和非字母的字符组成。

\LaTeX 忽略命令之后的空白字符。如果你希望在命令后得到一个空格, 可以在命令后加上{} 和一个空格, 或加上一个特殊的空格命令。{} 将阻止 \LaTeX 吃掉命令后的所有空格。

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 \TeX nicians and \TeX perts.
Today is August 2, 2008.

有些命令需要一个参数(parameter), 该参数用花括号(curly braces) {} 括住并写在命令的后面。一些命令支持可选参数(optional parameters), 可选参数可用方括号(square brackets) [] 括住, 然后写在命令的后面。下面的例子中使用了一些 \LaTeX 命令, 不要着急, 后面将解释它们的含义。

You can $\text{\textsl{lean}}$ 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!

⁴试试 $\text{\backslashbackslash}$ 命令, 它将生成一个'\'。

1.3.4 注释

当 \LaTeX 处理一个源文件时，如果遇到一个百分号 $\%$ ， \LaTeX 将忽略 $\%$ 后的该行内容，换行符以及下一行前的空白字符。

我们可以据此在源文件中写一些注释，而且这些注释并不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: Supercalifragilisticex-
pialidocious

符号 $\%$ 也可以用来断开不能含有空白字符或换行符的较长输入内容。

如果注释的内容较长，你可以使用`verbatim` 宏包提供的`comment` 环境。当然，在使用该环境前，你要在文档的导言区（后面将会解释其含义）加上命令`\usepackage{verbatim}`。

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

This is another example for embedding
comments in your document.

需要注意的是以上做法在数学环境等复杂环境中不起作用。

1.4 源文件的结构

当 \LaTeX 2_ϵ 处理源文件时，它希望源文件遵从一定的结构(structure)。因此，每个源文件都要以如下命令开始

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

这条命令指明了你所写的源文档的类型。然后，你就可以加入控制整篇文档样式的命令，或者载入一些为 \LaTeX 增加新特性的宏包(package)。可以用如下命令载入一个宏包

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

当完成所有的设置工作后⁵，你可以用下面的命令开始文档的主体

```
\begin{document}
```

现在你就可以输入带有 \LaTeX 命令的正文了。在文章末尾使用命令

```
\end{document}
```

来告诉 \LaTeX 文档已经结束。 \LaTeX 会忽略此命令后的所有内容。

图1.1 显示的是一个简单的 \LaTeX 2_ϵ 文档的结构。一个较为复杂的源文件(input file) 结构如图1.2 所示。

⁵在`\documentclass` 和`\begin{document}` 之间的部分称作导言区 (preamble)。

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

图 1.1 – 一个简单的 \LaTeX 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{Some Interesting Words}
Well, and here begins my lovely article.
\section{Good Bye World}
\ldots{} and here it ends.
\end{document}
```

图 1.2 – \LaTeX 类 \LaTeX 源文件例子，该例中的所有命令后面都会讲到。

1.5 一个典型的命令行过程

我敢打赌你现在一定非常渴望尝试第6页上短小简洁的 \LaTeX 源文件。下面便是一些帮助： \LaTeX 本身没有图形用户界面或漂亮的按钮，它仅仅是一个处理你提供的源文件的程序。有些 \LaTeX 安装版本提供了一个前端图形界面，你可以通过点击按钮来编译你的源文件。其他的一些系统上可能就要使用命令来编译源文件，下面演示的就是如何在一个基于文本的系统上让 \LaTeX 编译你的源文件。需要注意：以下演示的前提是 \LaTeX 已经正确的安装到了你的电脑中⁶。

1. 创建并编辑你的源文件。源文件必须是普通的 ASCII 格式。在 Unix 系统下，所有的编辑器都可以创建这样的文件。在 Windows 系统下，你必须确保文件以 ASCII 或普通文本格式保存。当选取你源文件的文件名时，确保它的扩展名是 `.tex`。
2. 运行 \LaTeX 编译你的源文件。如果成功的话，你将会得到一个 `.dvi` 文件。为了得到目录和所有的内部引用，可能要多次运行 \LaTeX 。当源文件中存在错误时， \LaTeX 会告诉你错误并停止处理源文件。输入 `ctrl-D` 可以返回到命令行。

```
latex foo.tex
```

3. 现在可以通过几种方法来预览得到的 DVI 文件。你可以使用下列命令将文件显示到屏幕上

```
xdvi foo.dvi &
```

这种方法只适用于安装了 X11 的 Unix 系统。如果你使用的是 Windows 系统，可以使用 `yap` 来预览（或其他预览程序）。

你也可以使用 Ghostscript 将 dvi 文件转换成 POSTSCRIPT 文件来打印或预览。

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

如果你的 \LaTeX 系统中带有 `dvipdf` 工具的话，就可以直接将 `.dvi` 文件转换成 pdf 文件。

```
dvipdf foo.dvi
```

⁶这是在大部分 Unix 系统下的情况……高手使用 Unix，所以……;-)

1.6 文档布局

1.6.1 文档类

当 \LaTeX 处理源文件时，首先需要知道的就是作者所要创建的文档类型。文档类型可由 \documentclass 命令来指定。

```
 $\text{\documentclass}[options]{class}$ 
```

class 指定想要的文档类型。表1.1给出了一些文档类型的解释。 \LaTeX 2 ϵ 发行版中还提供了其他一些文档类，像信件和幻灯片等。通过 *options* 参数可以定制文档类的属性。不同的选项之间须用逗号隔开。标准文档类的最常用选项如表1.2所示。

例子：一个 \LaTeX 源文件以下面一行开始

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

这条命令会引导 \LaTeX 使用 *article* 格式、**11磅大小的字体**来排版该文档，并得到在A4纸上**双面打印**的效果。

1.6.2 宏包

排版文档时，你可能会发现某些时候基本的 \LaTeX 并不能解决你的问题。如果想插入图形(graphics)、彩色文本(coloured text)或源代码到你的文档中，你就需要使用宏包来增强 \LaTeX 的功能。可使用如下命令调用宏包

```
 $\text{\usepackage}[options]{package}$ 
```

这里 *package* 是宏包的名称，*options* 是用来激活宏包特殊功能的一组关键词。很多宏包随 \LaTeX 基本发行版一起发布(见表1.3)，其他的则单独发布。你可以在所安装的 \LaTeX 系统中找到更多的宏包相关信息。*The \LaTeX Companion* [3]提供了关于宏包的重要信息，它包含了数百个宏包的描述及如何写作自己的 \LaTeX 2 ϵ 扩展的信息。

表 1.1 – 文档类。

article	排版科学期刊、演示文档、短报告、程序文档、邀请函……
proc	一个基于 article 的会议文集类。
minimal	非常小的文档类。只设置了页面尺寸和基本字体。主要用来查错。
report	排版多章节长报告、短篇书籍、博士论文……
book	排版书籍。
slides	排版幻灯片。该文档类使用大号 sans serif 字体。也可以选用 Foil \TeX ^a 来得到相同的效果。

^amacros/latex/contrib/supported/foiltex

表 1.2 – 文档类选项。

<code>10pt, 11pt, 12pt</code>	设置文档中所使用的字体的大小。如果该项没有指定，默认使用10pt 字体。
<code>a4paper, letterpaper, ...</code>	定义纸张的尺寸。缺省设置为letterpaper。此外，还可以使用a5paper, b5paper, executivepaper 以及legalpaper。
<code>fleqn</code>	设置行间公式为左对齐，而不是居中对齐。
<code>leqno</code>	设置行间公式的编号为左对齐，而不是右对齐。
<code>titlepage, notitlepage</code>	指定是否在文档标题(document title) 后另起一页。 <code>article</code> 文档类缺省设置为不开始新页， <code>report</code> 和 <code>book</code> 类则相反。
<code>onecolumn, twocolumn</code>	\LaTeX 以单栏(one column) 或双栏(two column) 的方式来排版文档。
<code>twoside, oneside</code>	指定文档为双面或单面打印格式。 <code>article</code> 和 <code>report</code> 类为单面(single sided) 格式， <code>book</code> 类缺省为双面(double sided) 格式。注意该选项只是作用于文档样式，而不会通知打印机以双面格式打印文档。
<code>landscape</code>	将文档的打印输出布局设置为 landscape 模式。
<code>openright, openany</code>	决定新的一章仅在奇数页开始还是在下一页开始。在文档类型为 <code>article</code> 时该选项不起作用，因为该类中没有定义“章”(chapter)。 <code>report</code> 类默认在下一页开始新一章而 <code>book</code> 类的新一章总是在奇数页开始。

表 1.3 – 随L^AT_EX一起发行的宏包。

<code>doc</code>	排版L ^A T _E X的说明文档。具体描述见 <code>doc.dtx</code> ^a 及 <i>The L^AT_EX Companion</i> [3]。
<code>exscale</code>	提供了按比例伸缩的数学扩展字体。 具体描述见 <code>ltxscale.dtx</code> 。
<code>fontenc</code>	指明使用哪种L ^A T _E X字体编码(font encoding)。 具体描述见 <code>ltoutenc.dtx</code> 。
<code>ifthen</code>	提供如下形式的命令 ‘if ... then do ... otherwise do ...?’ 具体描述见 <code>ifthen.dtx</code> 及 <i>The L^AT_EX Companion</i> [3]。
<code>latexsym</code>	提供L ^A T _E X符号字体。具体描述见 <code>latexsym.dtx</code> 及 <i>The L^AT_EX Companion</i> [3]。
<code>makeidx</code>	提供排版索引的命令。具体描述见第4.3节及 <i>The L^AT_EX Companion</i> [3]。
<code>syntonly</code>	编译文档而不生成dvi文件（常用于查错）。
<code>inputenc</code>	指明使用哪种输入编码，如ASCII, ISO Latin-1, ISO Latin-2, 437/850 IBM code pages, Apple Macintosh, Next, ANSI-Windows 或用户自定义编码。具体描述见 <code>inputenc.dtx</code> 。

^a你的系统中应该安装了该文件，输入命令`latex doc.dtx`处理该文件可得到一个dvi文件。类似的方法适用于本表格中的其他.dtx文件。

现代的T_EX发行版包含了大量免费的宏包。如果你使用的是Unix系统，可以使用命令`texdoc`搜索宏包的说明文档。

1.6.3 页面样式

L^AT_EX支持三种预定义的页眉/页脚(header/footer)样式，称为页面样式(page style)。如下命令

```
\pagestyle{style}
```

中的`style`参数确定了使用哪一种页面样式。表1.4列出了预定义的页面样式。

表 1.4 – L^AT_EX预定义的页面样式。

<code>plain</code>	在页脚正中显示页码。这是页面样式的缺省设置。
<code>headings</code>	在页眉中显示章节名及页码，页脚空白。（本文即采用此样式）
<code>empty</code>	将页眉页脚都设为空白。

可以通过如下命令来改变当前页面的页面样式

```
\thispagestyle{style}
```

如何创建自定义页眉页脚的说明可以参见 *The \LaTeX Companion* [3] 及第58页的第4.4节。

1.7 各类 \LaTeX 文件

使用 \LaTeX 时, 你可能很快发现自己置身于各种不同扩展名(extension) 或毫无线索的文件形成的迷宫之中。下面的列表解释了在使用 \LaTeX 时可能遇到的文件类型。要注意的是, 下表不是所有的扩展名列表, 如果你发现有重要的文件类型没有收录进来, 请通知我。

- .**tex** \LaTeX 或 \TeX 源文件。可以使用`latex` 命令编译。
- .**sty** \LaTeX 宏包文件。可以使用`\usepackage` 命令将宏包文件载入到你的 \LaTeX 文档中。
- .**dtx** 文档化 \TeX 文件。这是 \LaTeX 宏包文件的主要发布格式。如果编译.**dtx** 文档, 将会得到其中包含的 \LaTeX 宏包文件的文档化宏代码。
- .**ins** 对应.**dtx** 文件的安装文件。如果你从网上下载了一个 \LaTeX 的宏包文件, 其中一般会包含一个.**dtx** 文件和一个.**ins** 文件。使用 \LaTeX 处理.**ins** 文件可以解开.**dtx** 文件。
- .**cls** 定义文档外观形式的类文件, 可以通过使用`\documentclass` 命令选取。
- .**fd** 字体描述文件, 可以告诉 \LaTeX 有关新字体的信息。

下面这些文件是使用 \LaTeX 处理源文件时产生的:

- .**dvi** 设备无关文件。这是运行 \LaTeX 编译的主要结果。你可以使用 DVI 预览器预览其内容或使用`dvips` 或其他程序输出到打印机。
- .**log** 记录了上次编译时的详细信息。
- .**toc** 储存了所有的章节标题。下次编译时将读取该文件并生成目录。
- .**lof** 和.**toc** 文件类似, 可生成图形目录。
- .**lot** 和.**toc** 文件类似, 可生成表格目录。
- .**aux** 用来向下次编译传递信息的辅助文件。主要储存交叉引用的相关信息。
- .**idx** 如果文档中包含索引, \LaTeX 将使用该文件存储所有的索引词条。此文件需要使用`makeindex` 处理, 详见位于57页的第4.3节。
- .**ind** 处理过的.**idx** 文件。下次编译时将读入到你的文档中。
- .**ilg** 和.**log** 文件类似, 记录了`makeindex` 命令运行的详细信息。

1.8 大型项目

当处理大型文档时，最好将文档分割成为几部分。 \LaTeX 有两个命令可以帮助你完成这项工作。

```
\include{filename}
```

你可以使用该命令将名为 *filename.tex* 的文档内容插入到当前文档中。需要注意的是，在处理插入的 *filename.tex* 文档前， \LaTeX 会另起一页。

第二个命令只能在导言区使用。它可以让 \LaTeX 仅读入某些 \include 文件。

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

这条命令在文档的导言区执行后，在所有的 \include 命令中，只有文档名出现在 \includeonly 的命令参数中的文档才会被导入。注意文档名和逗号之间不能有空格。

\include 命令会在新的一页上排版载入的文本。当使用 \includeonly 命令时会很有帮助，因为即使一些载入的文本被忽略，分页处也不会发生变化。有些时候可能不希望在新的一页上排版载入的文本，这时可以使用命令

```
\input{filename}
```

\input 命令只是简单的载入指定的文本，没有其他限制。

如果想让 \LaTeX 快速的检查文档中的错误，可以使用 \syntonly 宏包。它可以使 \LaTeX 浏览整个文档，检查语法错误和使用的命令，但并不生成 DVI 输出。在这种模式下， \LaTeX 运行速度很快，可以为你节省宝贵的时间。 \syntonly 宏包的使用非常简单：

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

如果想产生分页，只要注释掉第二行即可(在前面加上一个百分号%)。

Chapter 2

文本排版

阅读了前一章之后，你应该了解关于如何创建一个 \LaTeX 文档的基本知识了。在这一章里，我将补充其余部分，使你能够生成实际文档。

2.1 文本和语言结构

By Hanspeter Schmid <hanspi@schmid-werren.ch>

书写文本的主旨是（某些现代 DAAC¹ 文化除外），向读者传递观点、信息或者知识。如果这些观点被很好地组织起来，那么读者会得到更好的理解。而且，如果排版形式反映内容的逻辑和语义结构，读者就能看到也更喜欢文章的这种脉络。

\LaTeX 不同于其它排版系统之处在于，你必须告诉它文本的逻辑和语义结构。然后它根据类文件和各种样式文件中给定的“规则”生成相应格式的文本。

\LaTeX 最重要的文本单元（印刷术上的）是段落(paragraph)。我们称段落为“文本单元”，因为段落是连续思想或者观点在排版上的反映。在下一节里，你将学会在源代码中如何使用 \backslash 来强迫换行，如何使用空行来分段。因此，一旦开始表达新的思想，就应该另起一段，否则换行就够了。如果无法决定是否分段，想象一下你的文字是观点和思想的载体。如果分段后，原来的思想仍在继续，就应该取消分段。如果有些行在同一段落里阐述了新的思想，那么应该分段。

大部分人完全低估了恰当分段的重要性。许多人甚至不知道分段表示什么，或者，特别是在 \LaTeX 里，设置了分段但却浑然不知。后一错误特别容易发生在文本中使用公式的情况。观察下面的例子并理解为什么有时公式前后都使用空行（分段），而有时不这样。（如果你还不能掌握里面所用的命令以至于无法理解这些例子，请在阅读这一章和下一章后再阅读这一节。）

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

¹为标新立异而不讲成本，译自 the Swiss German UVA (Um's Verrecken Anders).

```

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

Kirchhoff's voltage law can be 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

```

另一个更小的文本单元是句子。在英文文本中，结束句子的句点后面的空格比缩略词的句点后面的空格更长。 \LaTeX 试图判断你需要哪一个，如果 \LaTeX 判断错了，你必须告诉它你需要什么。这将会在下一章里谈到。

文本的结构甚至还包括句子的成份。大部分语言的标点规则非常复杂，但在许多语言（包括德文和英文）中，如果你记住逗号的意思：在语流中的短暂停顿，那么几乎所有的逗号都不会被用错。如果你不确定在什么地方应该使用逗号，大声地朗读句子并在每一个逗号处喘口气。在呼吸别扭的地方删除逗号，而在需要喘口气（或者需要短暂停顿）的地方插入一个逗号。

最后，通过包含段落的章、节和子节等等，段落应该在更高层次被有逻辑地组织起来。然而，使用诸如 $\text{\section{The Structure of Text and Language}}$ 的排版效果，是如此明显以至于如何使用这些高层次的结构是不言而喻的。

2.2 断行和分页

2.2.1 对齐段落

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(line break) 和间隙。如果一行的单词排不下， \LaTeX 也会进行必要的断词。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。更多的内容请参考第6.3.2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。

```
\*
```

在强制断行后，还禁止分页。

```
\newpage
```

另起一页。

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

上述命令的效果可以从它们的名称看出来。通过可选参量 n ，作者可以影响这些命令的效果。 n 可以取为 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 在排版效果不佳的时候选择忽略这个命令。不要把这些“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。然后在处理源文件的同时，报告溢出的消息(“overfull hbox”)。这最有可能发生在 \LaTeX 找不到合适的地方断词的时候²。你可以使用`\sloppy` 命令，告诉 \LaTeX 降低一点儿标准。它通过增加单词之间的间隔，以防止出现过长的行，虽然最终的输出结果不是最优的。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 断词

必要时 \LaTeX 就会断词。如果断词算法不能确定正确的断词点，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“-”的地方断词。命令的参量仅由正常字母构成的单词，或由 \LaTeX 视为正常字母的符号组成。当断词命令出现时，根据正在使用的语言，断词的提示就已经被存好待选了。这意味着如果你在文档导言中设置了断词命令，它将影响英文的断词。如果断词命令置于`\begin{document}` 后面，而且你正使用比方`babel` 的国际语言支持宏包，那么断词提示在由`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”进行断词，却根本不允许“FORTRAN”，“Fortran”和“fortran”进行断词。在参量中不允许出现特殊的字符和符号。

²当发生(Overfull hbox)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。

例子:

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

命令`\-`在单词中插入一个自主的断词点。它也就成为这个单词中允许出现的唯一断词点。对于包含特殊字符(例如: 注音字符)的单词, 这个命令是特别有用的, 因为对于他们, \LaTeX 不会自动断词³。

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

```
I think this is: supercalifragilisticexpialido-
cious
```

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。

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

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

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

```
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 内置字符串

在前面的例子中, 你已经看到用来排版特殊文本字符串的一些非常简单的 \LaTeX 命令了。

命令	例子	描述
<code>\today</code>	August 2, 2008	今日日期
<code>\TeX</code>	\TeX	你最喜爱的排版工具
<code>\LaTeX</code>	\LaTeX	游戏的名目
<code>\LaTeXe</code>	\LaTeX 2_ϵ	现在的化身

2.4 特殊字符和符号

2.4.1 引号

你不能再像在打字机上那样, 把"用作引号(quotation marks)。在印刷中有专门的左引号和右引号。在 \LaTeX 中, 用两个```(重音)产生左引号, 用两个`'`(直立引号)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.”
```

当然我知道这种实现机制不是最理想的, 无论字体如何, 它总是一个反向的勾号或者重音符(```)当左引号, 直立引号(`'`)当右引号。

³除非你正在使用新的 DC 字体(DC font)。

2.4.2 破折号和连字号

L^AT_EX 中有四种短划(dash) 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
```

这些短划线是：‘-’ 连字号(hyphen)，‘-’ 短破折号(en-dash)，‘—’ 长破折号(em-dash) 和‘-’ 减号(minus sign)。

2.4.3 波浪号(~)

波浪号经常和网址用在一起。它在L^AT_EX 中，可用`\~` 产生，但其结果：`~` 却不是你想要。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(°)

下面的例子演示了在L^AT_EX 中如何排版度的符号(degree symbol)：

```
It's $-30\,^{\circ}\mathrm{C}$.
I will soon start to
super-conduct.
```

```
It's -30°C. I will soon start to super-
conduct.
```

`textcomp` 宏包里有另外一个度的符号`\textcelsius`。

2.4.5 欧元符号(€)

现在撰写有关货币的文章，通常需要欧元符号。现有的许多字体都包含它。在你的导言区载入`textcomp` 宏包，

```
\usepackage{textcomp}
```

你就可以使用命令

```
\texteuro
```

来生成欧元符号。

如果你的字体不提供或者你不喜欢它给出的欧元符号，还有两个选择：

首先是eurosym 宏包。它提供了官方的欧元符号：

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

如果你希望得到跟所用字体匹配的欧元符号，使用选项gen 替换official。
marvosym 宏包也提供了很多符号，包括一个名为\EURtm 的欧元符号。它的缺点是没有提供欧元符号的斜体(slanted) 和粗体(bold) 变形。

表 2.1 – 欧元符号工具箱。

LM+textcomp	\texteuro	€	€	€
eurosym	\euro	€	€	€
[gen]eurosym	\euro	€	€	€
marvosym	\EURtm	€	€	€

2.4.6 省略号(...)

在打字机上，逗号(comma) 或句号(period) 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”(ellipsis)，因为间隔划分得不对。有一个专门的命令输出省略号。它被称为

```
\ldots
```

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

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

2.4.7 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(ligature)，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。

```
Not shelfful\\
but shelf\mbox{}ful
```

```
Not shelfful
but shelfful
```

2.4.8 注音符号和特殊字符

L^AT_EX 支持来自许多语言中的注音符号(accent) 和特殊字符(special character)。表2.2 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, ¡Se norita!,
Schönbrunner Schloß Straße
```

表 2.2 – 注音符号和特殊字符。

ò	\'o	ó	\'o	ô	\^o	o	\ o
ō	\=o	ó	\.o	ö	\"o	ç	\c c
ǒ	\u o	ǒ	\v o	ǒ	\H o	q	\c o
o	\d o	o	\b o	ô	\t oo		
œ	\oe	Œ	\OE	æ	\ae	Æ	\AE
å	\aa	Å	\AA				
ø	\o	Ø	\O	l	\l	L	\L
i	\i	j	\j	i	!'	¿	?'

2.5 国际语言支持

如果你需要用英文以外的语文(language) 书写文件，L^AT_EX 有两个地方必须配置好：

1. 所有自动生成的字符串⁴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。
2. 对于一种新语言，L^AT_EX 需要知道它的断词规则。将断词规则输入L^AT_EX 有些难度。这是说为不同断词模式重建格式文件是行得通的。对此 *Local Guide* [5] 给了更多的信息。
3. 特定语言的排版规则。比如法语中，每一个冒号(:) 前面必须留出一定的空白。

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。已经被你的L^AT_EX 系统支持的**语言**列表会在每次编译的时候显示。对于选定的语言，宏包babel 将自动激活适当的断词规则。如果L^AT_EX 的

⁴目录、图形清单……

格式文件不支持在所选择的语言中断词，除了失去断词功能，宏包**babel** 仍起作用，当然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。例如德文(German) 包含很多元音变音(äöü)。利用**babel**，你能用"o 而不是\"o 来输入ö。

如果为 **babel** 指定了多种语言

```
\usepackage[languageA,languageB]{babel}
```

选项中的最后一种语言会被激活(即 languageB)。你可以使用

```
\selectlanguage{languageA}
```

来改变被激活的语言。

大多数现代的计算机系统允许直接从键盘输入某国的字母。为了处理大量不同语系以及/或者计算机平台使用的输入编码，**L^AT_EX** 使用**inputenc** 宏包：

```
\usepackage[encoding]{inputenc}
```

当使用这个宏包时，应该考虑其他人可能因为使用不同的编码，在其计算机上或许不能显示你的源文件。例如，德语元音变音ä 的编码为 132，在一些使用 ISO-LATIN 1 的 Unix 系统上，它的编码就成了 228；但是 Windows 上的 Cyrillic 编码 cp1251 里却根本没有这个字母。所以应小心使用这个功能。根据你使用的系统类型，下列编码可能会派得上用场⁵。

Operating system	encodings	
	western Latin	Cyrillic
Mac	applemac	macukr
Unix	latin1	koi8-ru
Windows	ansinew	cp1251
DOS, OS/2	cp850	cp866nav

如果你有一份多语言文档，其中的编码会有冲突。这时可以使用**ucs** 宏包来选择 unicode。

```
\usepackage{ucs}
\usepackage[utf8x]{inputenc}
```

会让你创建的**L^AT_EX** 文档使用**utf8x**，它是一种多字节的编码，其中每个字符需要最少一个字节，最多 4 个字节。

字体编码是另外一个问题。它定义于一种**T_EX** 字体里每个字母的存放位置。几种不同的输入编码可以被映射到一种字体编码，这样减少了所需的字体集数量。字体编码通过**fontenc** 宏包来处理：

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

其中 *encoding* 是字体编码。可以同时载入几种编码。

⁵要想知道更多基于 Latin 或者 Cyrillic 语言支持的输入编码，请分别阅读**inputenc.dtx** 和**cyinperc.dtx** 的文档。第4.6 节讲到了如何生成宏包文档。

表 2.3 – 葡萄牙文所需的导言区。

```

\usepackage [portuguese]
{babel}
\usepackage [latin1]
{inputenc}
\usepackage[T1]{fontenc}

```

默认的 \LaTeX 字体编码是 OT1 ，Computer Modern \TeX 字体的原有编码。它只包含了 7-bit ASCII 字符集的 128 个字符。需要注音字符的时候， \TeX 把一个正常的字符附上重音符来创建它。虽然输出结果看上去很完美，但这种方法停止了对注音字符的自动断词功能。另外，这种方法不能创建一些拉丁字母，而且对非拉丁字母一筹莫展，比如希腊字母(Greek) 和西里尔字母(Cyrillic)。

为了克服这个缺点，一些 8-bit 的类似 CM 的字体集被打造出来。 T1 编码的 *Extended Cork* (EC) 字体以拉丁语系为基础，包含了支持大部分欧洲语言的字母和标点符号。 LH 字体集包含了排版斯拉夫语系文档必需的字母。因为斯拉夫字母的字形太多，它们被分成四种字体编码 —— T2A , T2B , T2C , 以及 X2 ⁶。希腊文的 LGR 编码字体在 CB 字体集里。

有了这些字体支持，你可以对非英文文本改进或者应用断词了。使用这些新的类似 CM 的字体还有一个好处，它们提供了 CM 字族里各种大小，形状以及比例缩放的字体。

2.5.1 葡萄牙文支持

By Demerson Andre Polli <polli@linux.ime.usp.br>

为了对葡萄牙文(Portuguese) 文档应用断词及各种自动文本，使用命令：

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

或者如果你在巴西的话，替换成**brazilian**。
鉴于葡萄牙文中有许多重音，你可能想要用

```
\usepackage[latin1]{inputenc}
```

来正确的输入它们，并且用

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

来正确的断词。

使用葡萄牙文的文档导言区请参考表2.3。注意我们使用的是 latin1 的输入编码，所以在 Mac 或者 DOS 上会不起作用。请自行选择合适的编码。

⁶这些编码所支持的语言列表可以在[11] 查到。

2.5.2 法文支持

By Daniel Flipo <daniel.flipo@univ-lille1.fr>

一些使用 \LaTeX 创建法文(French) 文档的提示：你可以通过以下命令载入法文支持：

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

请注意，由于历史原因，`babel` 的法文选项或者是`frenchb` 或者是`francais`，而不是`french`。

照此配置，你就可以使用法文的断词了。当然所有的自动文本也都成为法文：`\chapter` 印成 `Chapitre`，`\today` 印成法语里的今天的日期等等。同时也有一些系列的新命令，可以让你更容易的输入法文。请参考表2.4 来获取灵感。

表 2.4 – 法文专用命令。

<code>\og guillemets \fg{}</code>	« guillemets »
<code>M\up{me}, D\up{r}</code>	M ^{me} , D ^r
<code>1\ier{} , 1\iere{} , 1\ieres{} , 2\ieme{} 4\iemes{} , \No 1 , \no 2</code>	1 ^{er} , 1 ^{re} , 1 ^{res} 2 ^e 4 ^{es} N ^o 1, n ^o 2
<code>20 \degres C, 45\degres</code>	20 °C, 45°
<code>\bsc{M. Durand}</code>	M. DURAND
<code>\nombre{1234,56789}</code>	1 234,567 89

你会注意到，切换到法文的时候，列表的版面也改变了。更多关于`babel` 的`frenchb` 选项功能以及如何定制的内容，请对`frenchb.dtx` 运行 \LaTeX 并阅读生成的`frenchb.dvi`。

2.5.3 德文支持

一些使用 \LaTeX 创建德文(German) 文档的提示：你可以通过以下命令来载入德文支持：

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

照此配置，你就可以使用德文的断词了。当然所有的自动文本也都成为德文：例如“Chapter” 印成“Kapitel”。同时也有一些系列的新命令，可以让你更迅速的输入德文，即使你没有使用 `inputenc` 宏包。请参考表2.5 来获取灵感。一旦使用 `inputenc` 宏包，所有这些都不重要了，当然你的文档也被锁定在一个特殊的编码世界里。

在德文的书籍里，你会经常发现法文的引号(«guillemets»)。然而德文的打字机里有不同的使用方法。德文书籍中的引号看起来是»this«。在瑞士讲德语的部分，打字机使用«guillemets»，这跟法文一样。

表 2.5 – 德文专用字符。

"a	ä	"s	ß
"‘	”	"’	“
"< or \flqq	«	"> or \frqq	»
\flq	<	\frq	>
\dq	”		

使用类似`\flq`命令的一个主要问题是：如果你用 OT1 字体（这是默认字体），guillemets 看起来就像数学符号“ \ll ”，这令排版者反胃。而 T1 编码的字体含有正确的符号。所以，当你使用这种引号的时候，请确保正在用 T1 编码。（`\usepackage[T1]{fontenc}`）

2.5.4 朝鲜文支持⁷

为了使用 \LaTeX 排版朝鲜文(Korean)，我们需要解决三个问题：

1. 我们要能够编辑朝鲜文的源文件(Korean input files)。朝鲜文源文档必须是普通文本格式的(plain-text format)，但由于朝鲜文使用的字符集迥异于 US-ASCII 指令集，在一般的 ASCII 编辑器里看起来会相当怪异。两个最广为使用的朝鲜文文本文档编码是 EUC-KR 以及 MS-Windows 里它的向上兼容扩展，CP949/Windows-949/UHC。在这些编码里，每一个 US-ASCII 字符代表普通的 ASCII 字符，这跟其他兼容 ASCII 的编码比如 ISO-8859-*x*，EUC-JP，Big5，或者 Shift_JIS 相似。另一方面，从 KS X 1001 字符编码取出的朝鲜语谚文、汉字、朝鲜文字母、平假名、片假名、希腊文和斯拉夫字符以及其他符号和字母都用两个连贯的八位字节来表示。第一种有它的有效位集。直到 1990 年代中期，在非朝鲜文的操作系统上配置朝鲜文兼容环境还是一件费时费力的事。你可以浏览一下有些过时的<http://jshin.net/faq> 来了解那时是如何在非朝鲜文操作系统上使用朝鲜文的。现在，三种主要的操作系统(Mac OS, Unix, Windows) 都具备了相当好的多语言支持和国际化特征，所以在非朝鲜文平台上编辑朝鲜文文档已经不再是一个问题了。
2. \TeX 和 \LaTeX 最初只支持不超过 256 个字符。为了在其他有大量字符的语例如朝鲜文或汉文中让它们工作⁸，开发了一种子字体机制。一个有几千

⁷考虑到朝鲜文 \LaTeX 用户需要处理的大量问题，Karnes KIM 代表韩国 lshort 翻译团队撰写了这一节，并由 SHIN Jungshik 翻译为英文，Tobi Oetiker 作了简化。

⁸朝鲜语谚文是一种由 14 个基本辅音和 10 个基本元音构成的字母书写系统。不同于拉丁或者斯拉夫文字，每一个字符都要被排进跟汉文字符差不多大小的一簇矩形里。每一簇表示一个音节。这样就有限的元音和辅音构成了无限多的音节。但是现代朝鲜文的拼写标准（南、北朝鲜）都对这些簇的构成有严格的限制。因此只有有限个拼写正确的音节存在。朝鲜文字符编码给每一个音节的指定一个代码(KS X 1001:1998 和 KS X 1002:1992)。所以谚文虽然是一种字母文，处理起来却跟汉文和日文这些有几万个表意字符的书写系统差不多。ISO 10646/Unicode 提供了现代朝鲜语谚文的两种表示方法，一种是对相连的谚文字母编码（字母表：<http://www.unicode.org/charts/PDF/U1100.pdf>），另一种对所有拼写规范的现代朝鲜语音节编码(<http://www.unicode.org/charts/PDF/UAC00.pdf>)。使用 \LaTeX 及其相关排版系统处理朝鲜文有一项最令人犯难的挑战，就是对中古朝鲜文——可能会成为未来的朝鲜文——音节的支持，现在还只能用 Unicode 对相连的字母编码来解决。希望未来的 \TeX 引擎如 Ω 和 Λ 会最终提供解决方案，使得朝鲜语言和历史学者丢开 MS Word，虽然它已经

或者几万种字型(glyph)的CJK单字被分割成一组子字体集,每一集合里包含256个字型。对朝鲜文而言,有三个广为使用的宏包:UN Koaunghi开发的 $\text{H}\text{L}\text{A}\text{T}\text{E}\text{X}$,CHA Jaechoon的 $\text{h}\text{L}\text{A}\text{T}\text{E}\text{X}\text{p}$ 以及Werner Lemberg的CJK宏包(CJK package)⁹。 $\text{H}\text{L}\text{A}\text{T}\text{E}\text{X}$ 和 $\text{h}\text{L}\text{A}\text{T}\text{E}\text{X}\text{p}$ 专为朝鲜文设计并且在字体支持之外支持朝鲜文本地化(Korean localization)。对于EUC-KR编码的源文档,它们都可以正确的处理。在使用 Λ 和 Ω 的时候, $\text{H}\text{L}\text{A}\text{T}\text{E}\text{X}$ 还可以处理以CP949/Windows-949/UHC和UTF-8编码的源文档。

CJK宏包不只为朝鲜文提供支持。它还可以处理以UTF-8以及很多CJK编码包括EUC-KR和CP949/Windows-949/UHC的源文档。它支持多种语言内容的文档排版,特别是汉文,日文和朝鲜文。跟 $\text{H}\text{L}\text{A}\text{T}\text{E}\text{X}$ 相比,CJK宏包不提供朝鲜文本地化而且朝鲜文字体也不如 $\text{H}\text{L}\text{A}\text{T}\text{E}\text{X}$ 多。

3. 使用如 TEX 和 $\text{L}\text{A}\text{T}\text{E}\text{X}$ 排版工具的最终目的是用“美学”上令人满意的方式排版文档。可以说,排版中最重要的是优美设计的字体。 $\text{H}\text{L}\text{A}\text{T}\text{E}\text{X}$ 发行版包含10族(family)UHC POSTSCRIPT字体和5族(family)文化部(Munhwabu¹⁰)字体(TrueType)。CJK宏包使用的字体是 $\text{H}\text{L}\text{A}\text{T}\text{E}\text{X}$ 较早版本里的,但它支持Bitstream's cyberbit TrueType字体。

使用 $\text{H}\text{L}\text{A}\text{T}\text{E}\text{X}$ 宏包来输入朝鲜文,只需把

```
\usepackage{hangul}
```

放到你的导言区即可。

这一命令激活了朝鲜文本地化支持。章、节、子节、目录和图表目录都会被转换成相应的朝鲜文,而且使用朝鲜文的习惯来格式化文档。这个宏包还提供了自动的“虚词选择”功能。在朝鲜文里,有大量的这类语法上等价但是形式不同的后缀虚词,哪一个词组组合是正确的依赖于前面的音节是以元音还是以辅音结尾的。(实际情况比这还要复杂,但上述描述足够给你一个大致印象了)以朝鲜文为母语的人选择适合的虚词毫无问题,但是文档编辑中随时改变的参考文献以及其他自动文本就很难确定。每一次你增删参考文献或者改变文档内容的顺序时,手工放置合适的虚词都是一件辛苦的工作。 $\text{H}\text{L}\text{A}\text{T}\text{E}\text{X}$ 的用户就可以从这种烦人而且容易出错的工作中解放出来。

如果你不需要朝鲜文本地化,只是想要排版一些朝鲜文字,可以把放到导言区的命令换成:

```
\usepackage{hfont}
```

更多使用 $\text{H}\text{L}\text{A}\text{T}\text{E}\text{X}$ 排版朝鲜文的信息,请看 $\text{H}\text{L}\text{A}\text{T}\text{E}\text{X}$ Guide。访问Korean TEX User Group (KTUG)的网页<http://www.ktug.or.kr/>。那里也有一份本手册的朝鲜语译本。

2.5.5 用希腊文写作

By Nikolaos Pothitos <pothitos@di.uoa.gr>

使用希腊文(Greek)写作所需的导言内容参见表2.6。它们可以实现希腊文的断词和自动文本¹¹。

对中古朝鲜文有了良好的支持。

⁹这些可以在`language/korean/HLaTeX/`, `language/korean/CJK/`和<http://knot.kaist.ac.kr/htex/>取得。

¹⁰南朝鲜文化部。

¹¹如果对`inputenc`宏包使用了`utf8x`选项,你可以排版希腊文和多声调希腊文的unicode字符。

表 2.6 – 希腊文文档所需导言区。

```
\usepackage[english,greek]{babel}
\usepackage[iso-8859-7]{inputenc}
```

有一组新的命令可以让你更容易地输入希腊文。为了暂时切换为英文或者相反，你可以使用命令`\textlatin{english text}`以及`\textgreek{greek text}`，它们都只有一个参量，可以使用所要求的字体编码排版。或者你也可以使用前面章节说过的命令`\selectlanguage{...}`。表2.7列出了一些希腊文标点符号。对于欧元符号，要使用`\euro`。

表 2.7 – 希腊文特殊字符。

```
; · ? ;
(( « )) »
‘ ‘ ’ ’
```

2.5.6 斯拉夫文支持

By Maksym Polyakov <polyama@myrealbox.com>

版本为 3.7h 的 `babel` 宏包包含了对 T2* 编码以及使用斯拉夫字母排版保加利亚文、俄文和乌克兰文的支持。

斯拉夫文的支持依赖于 L^AT_EX 系统还有 `fontenc` 和 `inputenc` 宏包。但是如果你在数学模式下使用斯拉夫文，就必须在 `inputenc` 之前加载 `mathtext` 宏包¹²：

```
\usepackage{mathtext}
\usepackage[T1,T2A]{fontenc}
\usepackage[koi8-ru]{inputenc}
\usepackage[english,bulgarian,russian,ukranian]{babel}
```

一般情况下，`babel` 会自动选择的默认的字体编码，对于上面三种语文，应该是 T2A。然而，文档不会限制只使用一种字体编码。对于有拉丁语系和斯拉夫语系的多语文文档，应该明确包含拉丁语文字体的编码。在文档中，当选择另外一种语文的时候，`babel` 会控制切换到合适的字体编码。

除了能够断词，翻译自动文本字符串，以及激活一些语文专用的排版规则（比如 `\frenchspacing`），`babel` 还提供了一些命令可以按照保加利亚文、俄文、或者乌克兰文的标准排版。

这三种语言专用的标点符号也被提供了：斯拉夫文本的破折号（它比拉丁语文的破折号略窄，周围有微小的空白）、直接引语用的破折号、引号、以及方便断词的命令，请参考表 2.8。

¹²如果使用了 $\mathcal{A}\mathcal{M}\mathcal{S}\text{-L}\mathcal{A}\text{T}\mathcal{E}\text{X}$ 的宏包，相应的把它们放在 `fontenc` 和 `babel` 之前加载。

表 2.8 – babel 的 Bulgarian、Russian 和 Ukrainian 选项一些额外的定义。

"	当前位置取消连字。
"-	一个明确的断词符号，允许在单词的其他位置断词。
"--	普通斯拉夫文本中的破折号。
"--	合成的姓名（姓）中用的破折号。
"--*	表示直接引语的斯拉夫文破折号。
" "	类似于"-，但是不产生连字号（用于合成词中，比如x-"y 或者或者其他像"enable/disable" 的符号）。
"	没有断开点的合成词标记。
"=	带断开点的合成词标记，允许在构成单词里断词。
"，	短的空白，用于带断开点的姓的首字母。
"‘	用于德文里的左双引号（看起来像„）。
"’	用于德文里的右双引号（看起来像“）。
"<	用于法文的左双引号（看起来像 <<）。
">	用于法文的右双引号（看起来像 >>）。

babel 的 Russian 和 Ukrainian 选项定义了命令\Asbuk 和\asbuk，它们的作用类似于\Alph 和\alph，产生俄文和乌克兰文的大写和小写字母（无论文档的活动语言是哪一个）。babel 的 Bulgarian 选项提供了命令\enumBul 和\enumLat (\enumEng)，它们可以让\Alph 和\alph 产生保加利亚文或者拉丁（英文）字母的大小写，默认为保加利亚文的。

2.6 单词间隔

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。因为一般在有缩写的地方，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白，它告诉L^AT_EX 在句号后不要插入比正常字母更多的空白。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 用专门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{...}
\subsection{...}
\subsubsection{...}
\paragraph{...}
\subparagraph{...}
```

如果想把文档分成几个部分而且不影响章节编号，你可以使用

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

当你使用 `report` 或者 `book` 类的时候，可以用另外一个高层次的分节命令

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

因为 `article` 类的文档不划分为章，所以很容易把它作为一章插入书籍中。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别：

- 命令 `\part` 不影响章的序号。
- 命令 `\appendix` 不带参量，只把章的序号改用为字母标记¹³。

\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(table of contents) 内容，一个新文档必须编译(“ \LaTeX ed”) 两次。有时还要编译第三次。如有必要 \LaTeX 会告诉你。

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，目录的条目可由实际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}
```

¹³对 `article` 类文档改变节的序号。

整篇文档的标题(title) 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` 的参量中, 可以输入几个用`\and` 命令分开的名字。

在第6 页的图1.2 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, 而且章节不计数。当你使用带星的分节命令(例如, `\chapter*{Preface}`) 时, 这些章节就不会出现在目录里。

`\mainmatter` 应出现在书的第一章前面。它启用阿拉伯数字的页码计数器, 并对页码重新计数。

`\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(cross-references)。 \LaTeX 提供了如下交叉引用命令

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

其中`marker` 是用户选择的标识符。如果在节、子节、图、表或定理后面输入`\label` 命令, \LaTeX 把`\ref` 替换为相应的序号。`\pageref` 命令排印`\label` 输入处的页码¹⁴。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 2.8 on page 28.”
```

¹⁴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。

2.9 脚注

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)`¹⁵ 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¹⁶。

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

```
Footnotesa are often used by people using
\LaTeX.
```

```
aThis is a footnote.
```

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。`\LaTeX` 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 em-
phasized text, then \LaTeX uses the normal
font for emphasizing.
```

请注意要求`\LaTeX` 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.
```

¹⁵“put”是最常使用的英文单词之一。

¹⁶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？¹⁷

¹⁷ 路标不必走向它指向的地方:-)。

2.11 环境

为了排版专用的文本， \LaTeX 定义了各种不同格式的环境(environment):

```
\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。

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

下面的章节对所有重要的环境都做了解释。

2.11.1 Itemize、Enumerate 和 Description

`itemize` 环境适用于简单的列表，`enumerate` 环境适用于有排列序号的列表，而 `description` 环境用于带描述列表。

```
\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 左对齐、右对齐和居中

`flushleft` 和 `flushright` 环境分别产生左对齐(left-aligned) 和右对齐(right-aligned) 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 引用、语录和韵文

`quote` 环境可以用于引文、语录和例子。

```
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 摘要

科学出版物惯常以摘要开始，来给读者一个综述或者预期。 \LaTeX 为此提供了 `abstract` 环境。一般 `abstract` 用于 `article` 类文档。

```
\begin{abstract}
The abstract abstract.
\end{abstract}
```

The abstract abstract.

2.11.5 原文打印

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。

2.11.6 表格

`tabular`环境能用来排版带有水平和垂直表线的漂亮表格(table)。 \LaTeX 自动确定每一列的宽度。

命令

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

的参量`table spec`定义了表格的格式。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{width}`产生相应宽度、包含自动断行文本的列；`|`产生垂直表线。

如果一列里的文本太宽， \LaTeX 不会自动折行显示。使用`p{width}`你可以定义如一般段落里折行效果的列。

参量`pos`设定相对于环绕文本基线的垂直位置。使用字母`t`、`b`和`c`来设定表格靠上、靠下或者居中放置。

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}
```

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

表格的列分隔符可由`@{...}`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内容。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}
```

<u>no leading space</u>

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

<u>leading space left and right</u>

由于没有内建机制使十进制数按小数点对齐¹⁸，我们可以使用两列“作弊”达到这个目的：整数向右，小数向左对齐。`\begin{tabular}`行中的命令`@{.}`用一个“.”取代了列间正常间隔，从而给出了按小数点列对齐的效果。不要忘记用列分隔符(&)取代十进制小数点！使用命令`\multicolumn`可在数值“列”上放置一个列标签。

```
\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

¹⁸如果系统安装了‘tools’包，请看一下宏包dcolumn。

```
\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(floating bodies) 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位置。

首先看一下供浮动使用的 \LaTeX 命令：

包含在`figure` 环境或`table` 环境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] 或 \begin{table}[...]
```

称为`placement specifier`，它由浮动许可放置参数写成的字符串组成。请见表2.9。这个参数用于告诉 \LaTeX 浮动体可以被移放的位置。一个`placement specifier` 由一串浮动体许可放置位置 (`float-placing permissions`) 构成。参见表2.9。

一个表格可以由如下命令，例如

```
\begin{table}[!hbp]
```

表 2.9 – 浮动体放置许可。

Spec	浮动体许可放置位置……
h	<i>here</i> 在文本的确切位置上，对于小的浮动体很有用。
t	在页面的顶部(<i>top</i>)
b	在页面的底部(<i>bottom</i>)
p	在一个只有浮动体的专门的页面(<i>page</i>) 上。
!	忽略阻止浮动体放置的大多数内部参数 ^a 。

注意`pt` 和`em` 是 \TeX 单位。请阅读第92 页上表6.5 更多有关的更多内容。

^a例如一页上所允许的浮动体的最大数目。

开始, placement specifier `[!hbp]` 允许 \LaTeX 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]`。

\LaTeX 将按照作者提供的 placement specifier, 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队列中¹⁹。当新的一页开始的时候, \LaTeX 首先检查是否可能用等待队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, 处理等待队列中的第一个浮动体: \LaTeX 重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队列中。对于每一种浮动体, \LaTeX 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 \LaTeX 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等待队列中的某一个。

仅给定单个 placement specifiers 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。不要单独使用参数[h], 在 \LaTeX 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。使用

```
\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 \LaTeX 自动添加。

两个命令

```
\listoffigures 和 \listoftables
```

用起来和`\tableofcontents`命令类似, 分别排版一个图形目录和表格目录。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]{LLLLLoooooonnnngggg}
```

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嵌入的图片保留空间, 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位置(h)²⁰, \LaTeX 尝试得很辛苦(!)。如果这不可能, 它将试图把图片安排在页面的底部(b)。如果不能将图片安排在

¹⁹它们是“先来先走”队列!

²⁰假设图片的等待队列已空。

当前页面，它将决定是否可能开一个浮动页面以放置这张图片或来自表格等待队列中的一些表格。如果没有足够的材料来填充一个专门浮动页面， \LaTeX 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。

在一些情况下，可能需要使用命令

```
\clearpage 或者甚至是\cleardoublepage
```

它命令 \LaTeX 立即放置等待队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 的参量时，一些命令会失效。它们被称为脆弱命令(fragile commands)。 \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}}
```

Chapter 3

数学公式

现在你已经准备好了。那么在这一章里，让我们来着手于 TEX 的强大之处：数学排版。但是，要提醒你的是，本章只是浅尝辄止。可对很多人来说，这里所讲述的内容已很受用，如果你在这里找不到你所需数学排版的解决方案的话，也请不要灰心。极有可能在 $\text{A}\text{M}\text{S-}\text{L}\text{A}\text{T}\text{E}\text{X}$ ¹中找到针对你的问题的某个解决方案。

3.1 综述

$\text{L}\text{A}\text{T}\text{E}\text{X}$ 使用一种特有的模式来排版数学(mathematics)公式。数学公式允许以行间形式排版在一个段落之中，也可以以独立形式排版，此时段落可能会被拆开。处于段内的数学文本要放在 $\backslash($ 与 $\backslash)$ 之间， $\$$ 与 $\$$ 之间，或者 $\backslash\text{begin}\{\text{math}\}$ 与 $\backslash\text{end}\{\text{math}\}$ 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:  $c^2 = a^2 + b^2$ 
```

```
 $\text{T}\text{E}\text{X}$  is pronounced as
 $\backslash(\backslash\tau\backslash\epsilon\text{psilon}\backslash\chi\backslash)\backslash\backslash[6\text{pt}]$ 
100  $\text{m}^3$  of water $\backslash\backslash[6\text{pt}]$ 
This comes from my
 $\backslash\text{begin}\{\text{math}\}\heartsuit\backslash\text{end}\{\text{math}\}$ 
```

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

当你希望把自己的一些较长的数学方程或是公式单独的放在段落之外的时候，那么你最好显示 (display) 它们，而不要拆开此段落。为此，你可以把它们放在 $\backslash[$ 与 $\backslash]$ 之间，或者 $\backslash\text{begin}\{\text{displaymath}\}$ 与 $\backslash\text{end}\{\text{displaymath}\}$ 之间。

¹美国数学学会制作了一个强大的 $\text{L}\text{A}\text{T}\text{E}\text{X}$ 扩展。本章的很多例子都使用了这个扩展。所有最近的 TEX 发行版中都提供了这个扩展。如果你的系统中没有，可以去`macros/latex/required/amslatex`找找看。

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

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

or you can type less with:

```
\[a+b=c\]
```

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

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

or you can type less with:

$$a + b = c$$

如果你希望 LATEX 给你的方程编上号,你可以使用`equation`环境。然后你就可以用`\label`来给一个方程加上标签并在文中的某处用`\ref`或`amsmath`宏包中的`\eqref`命令来引用它。

```
\begin{equation} \label{eq:eps}
\epsilon > 0
\end{equation}
From (\ref{eq:eps}), we gather
\ldots{}From \eqref{eq:eps} we
do the same.
```

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

From (3.1), we gather ... From (3.1) we do the same.

注意一下公式排版样式的不同,前者是行间式样,后者是显示式样:

```
\lim_{n \to \infty}
\sum_{k=1}^n \frac{1}{k^2}
= \frac{\pi^2}{6}
```

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

```
\begin{displaymath}
\lim_{n \to \infty}
\sum_{k=1}^n \frac{1}{k^2}
= \frac{\pi^2}{6}
\end{displaymath}
```

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

数学模式和**文本模式**都有一些不同之处。例如,在**数学模式**中:

1. 大多数的空格和断行没有任何意义,而且所有的空隙要么是从相应数学表达式中自然的生成,要么是用一些专门的命令来指定,如`\,`,`\quad`或`\qquad`。
2. 空白行是不允许的。每个公式只能为一段。
3. 每一个字母都会被认为是一个变量名,且会相应被排版为此种样式。如果你想要在公式中排版普通的文本(直立字体和普通字距),那么你必须要把这些文本放在`\text{rm}{...}`命令中(参阅第45页的第3.7节)。

```
\begin{equation}
\forall x \in \mathbf{R}:
\quad x^2 \geq 0
\end{equation}
```

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

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

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

数学家对于符号的使用总是吹毛求疵：这里习惯上要使用空心粗体(“blackboard bold”)，要包含此字体，得用到`amsfonts` 或是`amssymb` 宏包的`\mathbb` 命令。上面的例子就变成

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

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

3.2 数学模式的群组

大部分数学模式的命令只对其后的一个字符有效，因此，如果你希望一个命令对多个字符起作用，你必须把它们放在一个群组中，使用花括号：`{...}`。

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

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

3.3 数学公式的基本元素

这一节将介绍数学排版中的最重要的一些命令。详细的数学排版符号的命令列表，可参阅第48页第3.10节。

小写希腊字母 (Greek letters) 的输入为`\alpha`、`\beta`、`\gamma`……，大写字母的输入为`\Gamma`、`\Delta` ……²

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

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

指数和下标可以能通过使用`^` 和`_` 两个符号来指定。

```
$a_{1}$ \quad $x^{2}$ \quad $e^{-\alpha t}$ \quad $a_{ij}^3$
$a^{3}_{ij}$ \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) 输入用`\sqrt`； n 次根用`\sqrt[n]` 来得到。根号的大小由 \LaTeX 自动决定。如果仅仅需要根号，可以用`\surd` 得到。

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

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

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

命令`\overline` 和`\underline` 产生水平线，它们会被放在表达式的正上方或是正下方。

² \LaTeX 2_ε 中没有定义大写的 Alpha，因为它外形与罗马字母 A 一样。等到新的数学编码完成后，情形可能会有所更改。

`\overline{m+n}`

$$\overline{m+n}$$

命令`\overbrace`和`\underbrace`可以在一个表达式的上方或下方生成水平括号

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

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

为了给变量增加数学重音符号，如小箭头或是 $\tilde{}$ (tilde)，你可以使用第48页表3.1所列出的命令。覆盖多个字符的宽“帽子”和宽 $\tilde{}$ 号，可以由`\widehat`和`\widetilde`得到。 $'$ 符号则给出了一个撇号(prime)。

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

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

向量可以通过在一个变量上方添加小箭头(arrow symbols)来指定。为此，使用`\vec`命令即可。`\overrightarrow`和`\overleftarrow`这两个命令可以用来表示一个从 A 到 B 的向量。

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

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

通常你没有必要打出一个明显的点号来表明乘法运算；但是有时候也需要它来帮助读者分清一个公式。在这些情况下，你应该使用`\cdot`命令。

```
\begin{displaymath}
v = {\sigma}_1 \cdot {\sigma}_2 \tau_1 \cdot \tau_2
\end{displaymath}
```

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

`\log`等类似的函数名通常是用直立字体，而不是如同变量一样用斜体，因此L^AT_EX提供了以下的命令来排版这些最重要的函数名：

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

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

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

对于取模函数(modulo function)，有两个命令：`\bmod`用于二元运算“ $a \bmod b$ ”，而`\pmod`则用于表达式如“ $x \equiv a \pmod{b}$ ”。


```
$a\bmod b$\
$x\equiv a \pmod{b}$
```

$$a \bmod b$$

$$x \equiv a \pmod{b}$$

一个上下的分式(fraction) 可用`\frac{...}{...}` 命令得到。而其倾斜形式如 $1/2$, 有时是更好的选择, 因为对于简短的分子分母来说, 这看上去更美观。

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

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

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

排版二项式系数或类似的结构, 你可以使用`amsmath` 宏包中的`\binom` 命令。

```
\begin{displaymath}
\binom{n}{k} \quad \mathrm{C}_n^k
\end{displaymath}
```

$$\binom{n}{k} \quad C_n^k$$

对于二元关系, 有时候你需要到把符号互相堆积起来。`\stackrel` 命令会把其第一个参数中的符号以上标大小放在第二个上面, 而第二个符号则以正常的位置摆放。

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

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

积分号 (integral operator) 可以用`\int` 产生, 求和号 (sum operator) 用`\sum` 命令, 而乘积号 (product operator) 要用`\prod` 命令。上限和下限用`^` 和`_` 来指定, 如同上标与下标一样³。

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

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

为了更好的控制一个复杂表达式中指标的放置, `amsmath` 提供了两个额外的工具: `\substack` 命令和`subarray` 环境:

```
\begin{displaymath}
\sum_{\substack{0 < i < n \\ 1 < j < m}}
P(i, j) =
\sum_{\begin{subarray}{l} i \in I \\ 1 < j < m \end{subarray}}
Q(i, j)
\end{displaymath}
```

$$\sum_{\substack{0 < i < n \\ 1 < j < m}} P(i, j) = \sum_{\substack{i \in I \\ 1 < j < m}} Q(i, j)$$

³`AMS-LATEX` 中另有多行的上标/下标。

$\text{T}_{\text{E}}\text{X}$ 提供了各种各样的符号来得到**括号** (braces) 和其他定界符(delimiters) (如: $[\langle \parallel \updownarrow]$)。圆括号和方括号可以由对应的键直接输入而花括号要用 $\{ \}$, 但是所有其它的定界符都要用一定的命令(如: \updownarrow) 生成。所有可用定界符的列表, 请查阅第50 页表3.7。

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

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

如果你在某个左定界符前放一个 \left 命令或是在某个右定界符前放一个 \right 命令, $\text{T}_{\text{E}}\text{X}$ 将会自动决定这对定界符的大小。请注意, 你必须为每个 \left 命令配对相应的 \right 命令, 而且只有在左右定界符被排版在同一行时才会获得正确的大小尺寸。如果你不想使用任何右定界符, 使用看不见的 $\right.$ 即可!

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

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

有些情况下, 有必要手工指定一个数学定界符的正确尺寸, 这可以使用 \big , \Big , \bigg 和 \Bigg 命令, 大多数情况下你只需把它们放在定界符命令的前面⁴。

```
$$\Big( (x+1)(x-1) \Big)^2$$
$\big(\Big(\bigg(\Big\quad
\big\}\Big\}\bigg\}\Big\}\$
\quad
$\big\|\Big\|\bigg\|\Big\|\$
```

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

$$\left(\left(\left(\right) \right) \right) \quad \left\| \right\| \left\| \right\| \left\| \right\| \left\| \right\|$$

有很多命令可以实现在公式中插入**三点列** (three dots)。 \ldots 得到在基线上的点列而 \cdots 是上下居中的点列。另外, 还有 \vdots 命令产生竖直的点列, \ddots 产生对角线的点列。你可以在第3.5 节找到另外一个例子。

```
\begin{displaymath}
x_1, \ldots, x_n \quad \quad \quad x_1 + \cdots + x_n
\end{displaymath}
```

$$x_1, \dots, x_n \quad \quad \quad x_1 + \cdots + x_n$$

3.4 数学空格

如果公式内由 $\text{T}_{\text{E}}\text{X}$ 选择的空格不令人满意, 那么也可以通过插入一些特殊的空格控制命令来调整。有一些命令可以产生小空格: \backslash , 得到 $\frac{3}{18} \text{quad}$ (\backslash), $\backslash:$ 得到 $\frac{4}{18} \text{quad}$ ($\backslash:$) 而 $\backslash;$ 会得到 $\frac{5}{18} \text{quad}$ ($\backslash;$)。转义的空格符 \backslash 产生一个中等大小的空格, 而 \quad ($_$) 和 \qquad ($_$) 产生大的空格。 \quad 的大小与当前字体中字母'M' 的宽度有关。 $\! \$ 命令会产生一个 $-\frac{3}{18} \text{quad}$ ($\!$) 的负空格。

⁴如果使用了某个改变字体大小的命令, 或是指定了11pt 或12pt 参数的话, 这些命令会达不到预期效果。使用 exscale 或 amsmath 宏包可以纠正它。


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

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

对于跨行的长公式或是方程组(equation system), 你可以使用`eqnarray`和`eqnarray*`环境来替代`equation`环境。在`eqnarray`环境中每一行都有一个等式编号。`eqnarray*`则不添加编号。

`eqnarray`和`eqnarray*`环境的用法与一个`{rcl}`形式的3列表格相类似, 这里中间一列可以用来放等号, 不等号, 或者是其他你选择的符号。`\\`命令可以断行。

```
\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, 这里等号两边空白都有些大。`\setlength\arraycolsep{2pt}`可以调小它, 比如在下一个例子里。

长等式不能被分成合适的小段。作者必须指定在哪里断且如何缩进。以下两种方法是最常用的。

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

$$\begin{aligned} \sin x &= x - \frac{x^3}{3!} + \frac{x^5}{5!} - \\ & \quad - \frac{x^7}{7!} + \dots \end{aligned} \quad (3.8)$$

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

$$\begin{aligned} \cos x &= 1 - \frac{x^2}{2!} + \\ & \quad + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots \end{aligned} \quad (3.9)$$

`\nonumber`命令告诉 \LaTeX 不要给这个等式编号。

用这种方法很难让等式正确的垂直对齐; `amsmath`宏包提供了一系列强有力的替代选择(参见`align`, `flalign`, `gather`, `multline`和`split`环境)。

3.6 虚位

我们看不见虚位 (phantom, 也有幻影的意思), 但是在许多人的头脑中它们依然占有一定的位置。L^AT_EX 中也一样。我们可以使用它来实现一些有趣的小技巧。

当使用 `^` 和 `_` 时, L^AT_EX 对文本的垂直对齐有时显得太过于自作多情。使用 `\phantom` 命令你可以给不在最终输出中显示的字符保留位置。理解此意的最好方法是看下面的例子。

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

$${}^{12}_6C \quad \text{versus} \quad {}^{12}_6C$$

```
\begin{displaymath}
\Gamma_{\{ij\}}^{\{\phantom{ij}k\}}
\quad \text{versus} \quad
\Gamma_{ij}^k
\end{displaymath}
```

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

3.7 数学字体尺寸

在数学模式中, T_EX 根据上下文选择字体大小。例如, 上标会排版成较小的字体。如果你想要把等式的一部分排版成罗马字体, 不要用 `\textrm` 命令, 只因 `\textrm` 会暂时切换到文本模式, 而此时字体大小切换机制将不起作用。使用 `\mathrm` 来保持字体大小切换机制的正常。但是要小心, `\mathrm` 只对较短的项有效。空格依然无效而且重音符号也不起作用⁵。

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

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

有时你仍需告诉L^AT_EX 正确的字体大小。在数学模式中, 可用以下四个命令来设定:

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

改变样式也会影响到上下限的显示方式。

```
\begin{displaymath}
\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}
```

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

⁵ $\mathcal{A}\mathcal{M}\mathcal{S}$ -L^AT_EX (amsmath) 宏包可以让 `\textrm` 命令与字体大小切换机制和谐共存。

这个例子中的括号要比`\left[\right]`提供的括号更大些。`\biggl`和`\biggr`命令分别对应于左和右括号。

3.8 定理、定律……

当写数学文档时，你可能需要一种方法来排版“引理”、“定义”、“公理”及其他类似的结构。

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

参量`name`是用来标识“定理”的短关键字。而参数`text`才是真正的“定理”名，它会在最终的文档中被打印出来。

方括号中是可选参量。两者都均用来指定“定理”的编号问题。使用`counter`参数来指定先前声明的“定理”的`name`。则此新的“定理”将与先前定理统一编号。`section`参数让你来指定章节单元，而“定理”会按相应的章节层次来编号。

在你的文档的导言区执行`\newtheorem`命令后，你就可以在文档中使用以下命令了。

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

`amsthm`宏包提供了`\newtheoremstyle{style}`命令，通过从三个预定义样式中选择其一来说明定理的外观，三个样式分别为：`definition`（标题粗体，内容罗马体），`plain`（标题粗体，内容斜体）和`remark`（标题斜体，内容罗马体）。

理论上已经说够多了，下面我们联系一下实践，这个例子希望能够带走你的疑问并让你知道`\newtheorem`环境其实比较复杂且不易理解。

首先定义定理环境

```
\theoremstyle{definition} \newtheorem{law}{Law}
\theoremstyle{plain}      \newtheorem{jury}[law]{Jury}
\theoremstyle{remark}     \newtheorem*{marg}{Margaret}
```

```
\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{marg}No, No, No\end{marg}
```

Law 1. Don't hide in the witness box

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

Margaret. No, No, No

“Jury”定理与“Law”定理共用了同一个计数器，因此它的编号与其他“Law”定理的编号是顺序下来的。方括号中的参量用来指定定理的一个标题或是其他类似的内容。

```
\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”定理有一个与当前章节相联系的编号。你也可以使用其他的单元，如章(chapter)或小节(subsection)。amsthm 还提供了一个proof 环境。

```
\begin{proof}
Trivial, use
\[E=mc^2\]
\end{proof}
```

Proof. Trivial, use

$$E = mc^2$$

□

使用\qedhere 命令你可以移动“证毕”符。“证毕”符默认是在证明结束时单独放于一行。

```
\begin{proof}
Trivial, use
\[E=mc^2 \qedhere\]
\end{proof}
```

Proof. Trivial, use

$$E = mc^2$$

□

3.9 粗体符号

在L^AT_EX 中得到粗体符号相当的不容易；这也许是故意设置的，以防业余水平的排版者过度的使用它们。字体变换命令\mathbf 可得到粗体字母，但是得到的是罗马体（直立的）而数学符号通常要求是斜体。还有一个\boldmath 命令，但是它只能用在数学模式之外。它不仅作用于字母也作用于符号。

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

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

请注意，逗号也成粗体了，这也许不是所需的。

使用amsbsy 宏包（包含在amsmath 中）或 tool 宏包集中的bm 将会便利许多，因为它们包含一个叫\boldsymbol 的命令。

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

$$\mu, M \quad \mu, M$$

3.10 数学符号表

以下表格列出了**数学模式**中的所有常用符号。

要使用表3.11–3.15⁶。必须在导言区先载入`amssymb`宏包而且系统中安装了AMS数学字体。如果系统中没有安装AMS宏包和字体，请查阅`macros/latex/required/amslatex`。更全面的列表可于`info/symbols/comprehensive`处找到。

表 3.1 – 数学模式重音符号。

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

表 3.2 – 希腊字母。

α	<code>\alpha</code>	θ	<code>\theta</code>	o	<code>o</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>		
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>		

⁶这些表格源自于 David Carlisle 的`symbols.tex`，而后在 Josef Tkadlec 的建议下作了较大的改动。

表 3.3 – 二元关系。

你可以在下列符号的相应命令前加上 `\not` 命令，而得到其否定形式。

$<$	<code><</code>	$>$	<code>></code>	$=$	<code>=</code>
\leq	<code>\leq or \le</code>	\geq	<code>\geq or \ge</code>	\equiv	<code>\equiv</code>
\ll	<code>\ll</code>	\gg	<code>\gg</code>	\doteq	<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 ^a	<code>\sqsubset</code>	\sqsupset ^a	<code>\sqsupset</code>	\Join ^a	<code>\Join</code>
\sqsubseteq	<code>\sqsubseteq</code>	\sqsupseteq	<code>\sqsupseteq</code>	\bowtie	<code>\bowtie</code>
\in	<code>\in</code>	\ni , \owns	<code>\ni , \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 or \ne	<code>\neq or \ne</code>

^a 使用 `latexsym` 宏包才能得到这个符号

表 3.4 – 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 , \lor	<code>\vee , \lor</code>	\wedge , \land	<code>\wedge , \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>
\triangleleft	<code>\triangleleft</code>	\bigtriangledown	<code>\bigtriangledown</code>	\ddagger	<code>\ddagger</code>
\lhd ^a	<code>\lhd</code>	\rhd ^a	<code>\rhd</code>	\wr	<code>\wr</code>
\unlhd ^a	<code>\unlhd</code>	\unrhd ^a	<code>\unrhd</code>		

表 3.5 – “大” 运算符。

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

表 3.6 – 箭头。

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

^a 使用 `latexsym` 宏包才能得到这个符号

表 3.7 – 定界符。

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

表 3.8 – 大定界符。

$\left($	<code>\lgroup</code>	$\right)$	<code>\rgroup</code>	$\left\{$	<code>\lmoustache</code>
\uparrow	<code>\arrowvert</code>	\uparrow	<code>\Arrowvert</code>	$\left $	<code>\bracevert</code>
$\right\}$	<code>\rmoustache</code>				

表 3.9 – 其他符号。

\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>	\square	<code>\Box^a</code>	\diamond	<code>\Diamond^a</code>
\perp	<code>\bot</code>	\top	<code>\top</code>	\angle	<code>\angle</code>	\surd	<code>\surd</code>
\diamond	<code>\diamondsuit</code>	\heartsuit	<code>\heartsuit</code>	\clubsuit	<code>\clubsuit</code>	\spadesuit	<code>\spadesuit</code>
\neg	<code>\neg</code> or <code>\lnot</code>	\flat	<code>\flat</code>	\natural	<code>\natural</code>	\sharp	<code>\sharp</code>

^a 使用 `latexsym` 宏包才能得到这个符号

表 3.10 – 非数学符号。

也可以在文本模式中使用这些符号。

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

表 3.11 – 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 3.12 – AMS 希腊和希伯来字母。

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

表 3.13 – AMS 二元关系。

\lessdot	<code>\lessdot</code>	\gtrdot	<code>\gtrdot</code>	\doteqdot	<code>\doteqdot</code>
\leqslant	<code>\leqslant</code>	\geqslant	<code>\geqslant</code>	\risingdotseq	<code>\risingdotseq</code>
\leqslantless	<code>\leqslantless</code>	\geqslantgtr	<code>\geqslantgtr</code>	\fallingdotseq	<code>\fallingdotseq</code>
\leqq	<code>\leqq</code>	\geqq	<code>\geqq</code>	\eqcirc	<code>\eqcirc</code>
\lll or \lllless	<code>\lll</code> or <code>\lllless</code>	\ggg	<code>\ggg</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>	\approx	<code>\approx</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>
\shortparallel	<code>\shortparallel</code>	\Supset	<code>\Supset</code>	\Vvdash	<code>\Vvdash</code>
\blacktriangleleft	<code>\blacktriangleleft</code>	\sqsupset	<code>\sqsupset</code>	\backepsilon	<code>\backepsilon</code>
\vartriangleright	<code>\vartriangleright</code>	\because	<code>\because</code>	\varpropto	<code>\varpropto</code>
\blacktriangleright	<code>\blacktriangleright</code>	\Subset	<code>\Subset</code>	\between	<code>\between</code>
\trianglerighteq	<code>\trianglerighteq</code>	\smallfrown	<code>\smallfrown</code>	\pitchfork	<code>\pitchfork</code>
\vartriangleleft	<code>\vartriangleleft</code>	\shortmid	<code>\shortmid</code>	\smallsmile	<code>\smallsmile</code>
\trianglelefteq	<code>\trianglelefteq</code>	\therefore	<code>\therefore</code>	\sqsubset	<code>\sqsubset</code>

表 3.14 – AMS 箭头。

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

表 3.15 – AMS 二元否定关系符和箭头。

\nless	\ngtr	\varsubsetneqq
\lneq	\gneq	\varsupsetneqq
\nleq	\ngeq	\subsetneqq
\nleqslant	\ngeqslant	\supsetneqq
\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
\subsetseteq	\nsupseteq	\ntrianglelefteq
\subsetsetneqq	\nsupsetneqq	\ntrianglerighteq
\nleftarrow	\rightarrow	\nleftrightarrow
\nLeftarrow	\Rightarrow	\nLeftrightarrow

表 3.16 – AMS 二元运算符。

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

表 3.17 – AMS 其他符号。

\hbar	<code>\hbar</code>	\hbar	<code>\hslash</code>	\mathbb{k}	<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>
\diamond	<code>\lozenge</code>	\blacklozenge	<code>\blacklozenge</code>	\bigstar	<code>\bigstar</code>
\sphericalangle	<code>\angle</code>	\sphericalangle	<code>\measuredangle</code>	\backprime	<code>\backprime</code>
\diagup	<code>\diagup</code>	\diagdown	<code>\diagdown</code>	\varnothing	<code>\varnothing</code>
\nexists	<code>\nexists</code>	\Finv	<code>\Finv</code>	\mho	<code>\mho</code>
\eth	<code>\eth</code>	\sphericalangle	<code>\sphericalangle</code>		

表 3.18 – 数学字母。

实例	命令	所需宏包
ABCDEabcde1234	<code>\mathrm{ABCDE abcde 1234}</code>	
<i>ABCDEabcde1234</i>	<code>\mathit{ABCDE abcde 1234}</code>	
<i>ABCDEF</i>	<code>\mathnormal{ABCDE abcde 1234}</code>	
<i>ABCDEF</i>	<code>\mathcal{ABCDE abcde 1234}</code>	
<i>ABCDEF</i>	<code>\mathscr{ABCDE abcde 1234}</code>	mathrsfs
$\frac{ABCDEF}{abcde1234}$	<code>\mathfrak{ABCDE abcde 1234}</code>	amsfonts or amssymb
$\mathbb{A}\mathbb{B}\mathbb{C}\mathbb{D}\mathbb{E}\mathbb{F}\mathbb{G}\mathbb{H}\mathbb{I}\mathbb{J}$	<code>\mathbb{A}\mathbb{B}\mathbb{C}\mathbb{D}\mathbb{E}\mathbb{F}\mathbb{G}\mathbb{H}\mathbb{I}\mathbb{J}</code>	amsfonts or amssymb

Chapter 4

专业功能

当你整理一个大型文档时， \LaTeX 的一些专门功能，例如自动生成索引、管理参考文献等等，会给你以很大的帮助。详细的关于 \LaTeX 专业功能以及增强功能的描述可以在 *\LaTeX Manual* [1] 和 *The \LaTeX Companion* [3] 找到。

4.1 插入 EPS 图形

\LaTeX 通过 `figure` 和 `table` 环境提供了处理图像图形等浮动体的基本工具。

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(graphics)，第5章中将会介绍其中的几种方法。如果需要这方面的详细信息，请参阅 *The \LaTeX Companion* [3] 和 *\LaTeX Manual* [1]。

在文档中使用图形，一个相对容易的办法就是使用专门的软件包¹生成图形文件，然后将最终的图形文件插入到文档中。 \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` 宏包提供了一套很好的插图命令。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把一幅图片加入你的文档中：

1. 用你的图形软件输出 EPS 格式的文件⁴。
2. 在源文件的导言中加上下面的命令来载入`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 \TeX 中没有规定插入图形的标准，所以 *driver* 的名字是必需的。知道

¹例如 XFig、CorelDraw!、Freehand、Gnuplot……

²另外一个可以用来输出POSTSCRIPT 的工具是GHOSTSCRIPT 程序，它可以从`support/ghostscript` 得到。Windows 和 OS/2 用户可能更喜欢用GSVIEW。

³`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，这个文件会是 EPS 格式的。注意一个 EPS 图片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输出 EPS 格式。

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。然后打印机理解这些信息并能正确的插入 `.eps` 文件。

3. 使用命令

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

来把 *file* 加入你的文档。可选的参数是一系列由逗号隔开的 *keys* 和相应的 (*values*)。 *keys* 可以用来改变插图的宽度、高度以及旋转。表 4.1 列出了最重要的关键词。

表 4.1 – `graphicx` 宏包使用的关键词。

<code>width</code>	把图形调整到指定的宽度
<code>height</code>	把图形调整到指定的高度
<code>angle</code>	逆时针旋转图形
<code>scale</code>	缩放图形

下面的示例代码可以帮助我们理解整个过程:

```
\begin{figure}
\centering
\includegraphics[angle=90,
                 width=0.5\textwidth]{test}
\caption{This is a test.}
\end{figure}
```

这段代码把文件 `test.eps` 中的图形插入到文档里。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段落宽度的 0.5 倍。因为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指定为绝对长度单位。详细的信息可以在第 92 页的表 6.5 中找到。如果你想知道更多这方面的知识, 请阅读文献 [9] 和 [13]。

4.2 参考文献

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

```
\bibitem[label]{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}
```

如果你不使用参数 *label*, 参考文献条目的编号是自动生成的。`\begin{thebibliography}` 命令后的参数设定了最大的编号宽度。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。


```
Part1 \cite{pa} has
proposed that \ldots
\begin{thebibliography}{99}
\bibitem{pa} H. Part1:
\emph{German \TeX},
TUGboat Volume 9, Issue 1
(1988)
\end{thebibliography}
```

Part1 [1] has proposed that ...

Bibliography

[1] H. Part1: *German T_EX*, TUGboat Volume 9, Issue 1 (1988)

对于大型项目，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 产生的参考文献的外观是基于样式表，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。

4.3 索引

许多书籍的一个非常有用的部分就是它们的索引(index)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` 必须在导言部分被载入：

```
\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 举例解释了参量 *key* 语法。

当 L^AT_EX 处理源文档时，每个 `\index` 命令都会将适当的索引项和当前页码写入一个专门的文件中。这个文件的名称和 L^AT_EX 源文档相同，但具有不同的扩

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

表 4.2 – 索引关键词语法示例。

示例	索引项	注释
<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>	Sam, 2	格式化的索引项
<code>\index{Lin@\textbf{Lin}}</code>	Lin , 7	同上
<code>\index{Jenny textbf}</code>	Jenny, 3	格式化的页码
<code>\index{Joe textit}</code>	Joe, 5	同上
<code>\index{ecole@\'ecole}</code>	école, 4	重音标记

展名后缀(.idx)。这个.idx 文件需要用makeindex 程序来处理。

```
makeindex filename
```

makeindex 程序生成一个与源文件同名的序列化索引文件，这个文件使用.ind 为扩展名。当再次用L^AT_EX 处理源文件时，这个序列化索引文件将被包含到源文件中

```
\printindex
```

命令出现的位置。

L^AT_EX 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。

请注意不谨慎地使用\index 命令，将会影响文档页版面布局。

My Word \index{Word}. As opposed
to Word\index{Word}. Note the
position of the full stop.

My Word . As opposed to Word. Note the
position of the full stop.

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。

定制页眉和页脚时一件棘手的事情就是得到每个页面所属的章节名称。L^AT_EX 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 section 名，使用\leftmark 来代表当前的 chapter 名。这两个命令的值将在处理 chapter 或者 section 命令时被重新赋值。

为了获得最大的灵活性，\chapter 等命令并不对\rightmark 和\leftmark 直接进行重定义，而是间接地通过调用\chaptermark、\sectionmark 或者\subsectionmark 来重新定义\rightmark 和\leftmark。

因此，只需要重新定义\chaptermark 命令，就能修改页眉上显示的章名。

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fancyhdr 宏包所带的文档。

⁶可以在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 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} % 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。

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，并解决了原来的一些限制。这本身并没有什么特别的，但 `verbatim` 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集。如果你想更多地了解这个宏包，可以阅读 [10]。

4.6 安装额外的宏包

大多数的 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 安装都带有大量预装的样式宏包，但更多的宏包可以在网上得到。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 $\text{T}_{\text{E}}\text{X}$ 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 \LaTeX 命令。这将会产生一个 `.sty` 文件。
2. 把 `.sty` 文件移到 \LaTeX 系统能找到的地方。一般是在 `.../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 \LaTeX 系统: `teTeX`, `fpTeX - texhash`; `web2c - maktexlsr`; `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 \LaTeX 命令。这会生成一个 `.dvi` 文件。注意你可能需要多次运行 \LaTeX 命令来正确处理交叉引用。
2. 检查一下 \LaTeX 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 \LaTeX 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 \LaTeX 命令。

4.7 使用 pdf \LaTeX

By Daniel Flipo <Daniel.Flipo@univ-lille1.fr>

PDF 是一种超文本文档(hypertext) 格式。类似于网页, 文档中的某些词可以被超链接标记。它们链接到这个文档的另一个地方, 甚至是另外一个文档。如果你点击这样一个超链接, 你将转调到链接的目的地。这意味着在 \LaTeX 格式的文档中所有 `\ref` 和 `\pageref` 出现的位置都变成超链接。此外, 目录、索引和其它类似的结构变成超链接的集合。

现在大多数的网页都是用 HTML 超文本标记语言编写。在写科技文档的时候, 这种格式有两个严重的缺陷:

1. 数学公式在 HTML 文档中通常都不被支持。尽管对此有一个标准, 但是大多是现在使用的浏览器都不支持, 或者缺少需要的字体。
2. 打印 HTML 文档是比较容易的, 但打印的结果会因系统平台和浏览器的不同而出现差异。这结果与我们在 \LaTeX 世界中期待的高质量相差十万八千里。

有许多将 \LaTeX 转为 HTML 的尝试。其中一些可以说是相当成功的, 它们能将一个标准的 \LaTeX 源文件生成一个合格的网页。但是为了达到目的, 需要去掉一些支持。当你开始使用 \LaTeX 的复杂功能和扩展宏包时, 那些尝试就行不通了。若希望即使是发不到网上也保留文档的高质量, 作者们就要使用 PDF (便

便携式文档格式)，它保留了文档的版式和允许超链接导航。现在大多数浏览器只要带上相应的插件都可以直接显示 PDF 文档。

尽管几乎所有的操作系统都有 DVI 和 PS 格式的阅读器，但你会发现人们更多地使用 Acrobat Reader 和 Xpdf 来阅读 PDF 文档。因而提供 PDF 格式的文档将使得潜在的读者更容易阅读。

4.7.1 发布到网上的 PDF 文档

有了 Hàn Thê Thành 开发的 pdfT_EX 程序，使用 L^AT_EX 源文件来创建 PDF 文件将变得非常简单。一般的 TeX 生成 DVI，而 pdfT_EX 生成 PDF。还有 pdfL^AT_EX 程序将 LaTeX 源文件生成 PDF。

现在大多数的 T_EX 发行版本都自动集成安装了 pdfT_EX 和 pdfL^AT_EX，例如：teT_EX，fpT_EX，MikT_EX，T_EXLive 和 CMacT_EX。

为了生成 PDF 而不是 DVI，只需要将命令 `latex file.tex` 改成 `pdflatex file.tex`。在不是使用命令行的 L^AT_EX 系统中，你可以在 T_EX 控制中心找到一个专门的按钮。

在 L^AT_EX 中，你可以通过 `documentclass` 的参数选项来定义页面的大小，例如：`a4paper` 或 `letterpaper`。这些也对 pdfL^AT_EX 有效，但是首先要让 pdfT_EX 知道这种页面的实际大小来控制 PDF 文件的页面大小。若你使用 `hyperref` 宏包（参见第 63 页），页面的大小将自动调整。否则，你需要手动的将下面两行加入到文档的导言区：

```
\pdfpagewidth=\paperwidth
\pdfpageheight=\paperheight
```

接下来一节将深入介绍 L^AT_EX 和 pdfL^AT_EX 之间的不同。主要的不同有三个方面：采用的字体，包含图像的格式和超链接的手动配置。

4.7.2 字体

pdfL^AT_EX 能处理所有的字体（PK 点阵、TrueType、POSTSCRIPT type 1……），但是作为 L^AT_EX 通常的字体格式，PK 点阵字体在 Acrobat Reader 下的显示效果非常差。为了获得文档更高的显示效果，最好使用 POSTSCRIPT Type 1 字体。高级的 T_EX 安装程序会自动设置好，你最好试一下，如果它正常运作，你就可以跳过这一节。

POSTSCRIPT Type 1 的 Computer Modern 和 AMSFonts 字体由 Blue Sky Research 和 Y&Y, Inc. 制作，后来版权属于美国数学学会 (AMS)。这些字体在 1997 年被开放，并且出现在大多数 T_EX 发行版中。

然而，如果你使用 L^AT_EX 来创建非英文的文档，你可能用到 EC、LH 或 CB 字体（关于 OT1 字体的讨论可参见第 21 页）。Vladimir Volovich 创建了 cm-super 字体包，包含全部 EC/TC、EC Concrete、EC Bright 和 LH 字体集。在 `CTAN:/fonts/ps-type1/cm-super` 可以获得，也被集成进了 T_EXLive7 和 MikT_EX。由 Apostolos Syropoulos 创建类似 type 1 CB 的希腊字体可在 `CTAN:/tex-archive/fonts/greek/cb` 获得。不幸的是，这些字体集跟 Blue Sky/Y&Y 的 Type 1 CM 字体的印刷质量不一样。L^AT_EX 会自动提示，而且文档在屏幕的显示效果也不如 Blue Sky/Y&Y type 1 CM 字体那么整洁。在高分辨率的输出设备下，它们生成的效果跟原来的 EC/LH/CB 点阵字体一样。

如果你使用一种拉丁语系的语文来创建文档，你有其它一些选择。

- 使用 `aeguill` 宏包，也叫 *Almost European Computer Modern with Guillemets*。只需在导言区添加一行 `\usepackage{aeguill}`，来启用 AE 虚拟字体替代 EC 字体。

- 或者使用 `mltex` 宏包，但是它只在 `pdfTeX` 设置了 `mltex` 选项时有效。

AE 虚拟字体集，像 `MITTeX` 系统，在 CM 字体的字符基础上创建全部缺失的字母并按 EC 顺序重新排列，就使得 `TeX` 处理它的时候认为它有一个全部 256 个字符的字体集。这样就可使用大部分系统中优质的 type 1 格式的 cm 字体。这套字体现在为 T1 编码，在拉丁语系的欧洲语文中能很好的运作。唯一的不足就是创建的 AE 字符不支持 Acrobat Reader 的查找功能，因此你不能在 PDF 文档中搜索那些带重音符号的单词。

对于俄文，一个类似的解决办法是使用 C1 虚拟字体，这可以在 `ftp://ftp.vsu.ru/pub/tex/font-packs/c1fonts` 上获得。这套字体集成了标准的 Bluesky CM type 1 字体和 Paradissa 与 BaKoMa 的 CMCYR type 1 字体，这些都可以在 CTAN 上找到。由于 Paradissa 字体只包含俄文字母，C1 字体里就没有其他西里尔字符了。

另一种解决办法是使用其它 `POSTSCRIPT` type 1 字体。事实上，其中一些字体被包含在 Acrobat Reader 的相应语言版本中。由于这些字体有不同的字符大小，页面上的文本格式将会改变。一般地，这类字体占得空间要比 CM 字体大，因为 CM 字体是一种非常节省空间的字体。而且文档可视的一致性也被破坏了，因为 Times、Helvetica 和 Courier 字体（这些是 Acrobat 里基本的可替换字体）没有被设计来在单个文档中和平共处。

为了达到上述目的，有两套字体已经做好：`pxfonts`，它基于作为正文字体的 *Palatino*；另外就是基于 *Times* 的 `txfonts` 宏包。只需要在文档的导言区加入下列几行就可以使用这些字体：

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

注：当你编译的时候，在 `.log` 文件中出现下面的信息：

```
Warning: pdftex (file eurmo10): Font eur... not found
```

这意味着文档使用的某些字体没有被找到。你必须解决这些问题，否则输出的 PDF 文件可能不会显示缺失字符的页面。

上面讨论了如此多的字体问题，特别是缺乏与 type 1 格式的 CM 字体同样高质量的 EC 字体一直困扰大家。最近一套新的被称为 Latin Modern (LM) 的高质量字体已开发完成。这使得前面的担心都是多余的。如果你安装了最新的 `TeX`，你可能已经安装好这套字体，需要做的只是在你的文档的导言区添加

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

就可以创建支持所有拉丁字符的优质 PDF 输出文件。

4.7.3 使用图形

通过 `graphicx` 宏包（参见第 55 页），你可以在文档中插入图形。使用 `pdftex` 作为 `driver` 的选项，这个宏包也可用于 `pdfLATEX`：

```
\usepackage[pdftex]{color,graphicx}
```

上面这个例子中，我还包含了 `color` 宏包，因此网页上彩色的文档会看起来更自然一些。

好消息到此为止。坏消息就是 Encapsulated `POSTSCRIPT` 格式的图形并不被 `PDFLaTeX` 所支持。如果你不在命令 `\includegraphics` 声明加载的文件的

扩展名, 宏包`graphicx` 将依赖于 *driver* 选项的设置自动寻找一个合适的文件。对于 `pdftex`, 它支持的格式有 `.png`, `.pdf`, `.jpg` 和 `.mps` (MetaPost), 但不支持 `.eps`。

一个解决这个问题的简单方法是通过在很多系统中可找到的 `epstopdf` 工具将你的 EPS 文件转化为 PDF 格式。对于矢量图 (画), 这时一个很好的解决办法, 但对于位图 (相片、扫描图), 这并不是很理想, 因为 PDF 格式本来就支持包含 PNG 和 JPEG 图像。PNG 格式适合屏幕截图和其它一些只有较少颜色的图像。JPEG 是一种非常适合于相片的格式, 而且占用空间少。

可能对于画一些特殊的几何图形这也不是很理想, 幸好可以通过一些专门的命令语言来画图形, 例如 MetaPost, 它可以在大多数的 T_EX 发行版本中找到, 并且也包含它的扩展手册。

4.7.4 超链接

`hyperref` 宏包将你的文档中的所有内部引用变成超链接。为此, 一些特殊的设置是必须的, 保证 `\usepackage[pdftex]{hyperref}` 是你的文档导言区的最后一行命令。

有很多选项用于定义 `hyperref` 宏包的效果:

- 或者在 `\usepackage[pdftex]{hyperref}` 的 `pdftex` 选项后用逗号隔开展列出;
- 或者使用单独一行的命令 `\hypersetup{options}`。

`pdftex` 是唯一必须的选项, 其他参数用来改变 `hyperref` 的默认效果⁷。下面的列表中默认值被写成 `upright` 字体。

`bookmarks (=true,false)` 显示或隐藏书签栏

`unicode (=false,true)` 允许在 Acrobat 书签栏使用非拉丁语系的字符

`pdftoolbar (=true,false)` 显示或隐藏 Acrobat 的工具栏

`pdfmenubar (=true,false)` 显示或隐藏 Acrobat 的菜单栏

`pdfwindow (=true,false)` 调整显示 PDF 的初始化放大倍率

`pdftitle (=text)` 定义 Acrobat 显示的文档信息(Document Info)

`pdfauthor (=text)` PDF 文件作者名字

`pdfnewwindow (=true,false)` 定义当超链接到另一个文档时, 是否打开一个新的窗口

`colorlinks (=false,true)` 链接有一个彩色框架(`false`) 或者链接文本的颜色设置(`true`)。链接的颜色通过下面的参数来控制 (默认的颜色已列出)

`linkcolor (=red)` 内部链接的颜色(sections, pages, etc.),

`citecolor (=green)` 引用链接的颜色(bibliography)

`filecolor (=magenta)` 文件链接的颜色

⁷值得注意的是 `hyperref` 宏包并不只是在 pdfT_EX 下可用。它也可以用于将 PDF 特殊信息嵌入到由通常 L^AT_EX 生成的 DVI 文件中。然后通过 `dvips` 转化为 PS 格式, 最后用 Adobe Distiller 将 PS 格式转成 PDF。

`urlcolor (=cyan)` URL 链接的颜色(mail, web)

如果你觉得这些默认值合适, 就直接使用

```
\usepackage[pdftex]{hyperref}
```

为了使书签列表打开和链接为彩色 (=true 不需要写出来):

```
\usepackage[pdftex,bookmarks,colorlinks]{hyperref}
```

当创建 PDF 文档是用来打印的, 使用彩色的链接并不是一件好事, 因为彩色的链接将会被打印成灰色, 从而难以阅读。你可以设置不打印彩色的框架:

```
\usepackage{hyperref}
\hypersetup{colorlinks=false}
```

或者将超链接变成黑色:

```
\usepackage{hyperref}
\hypersetup{colorlinks,%
  citecolor=black,%
  filecolor=black,%
  linkcolor=black,%
  urlcolor=black,%
  pdftex}
```

当你只想提供 PDF 文件的文档信息(Document Info):

```
\usepackage[pdfauthor={Pierre Desproges},%
  pdftitle={Des femmes qui tombent},%
  pdftex]{hyperref}
```

除了自动的交叉引用的超链接, 通过下面的命令可以嵌入明确的链接:

```
\href{url}{text}
```

代码

```
The \href{http://www.ctan.org}{CTAN} website.
```

生成的效果为“CTAN”: 单击词“CTAN” 将把你带到 CTAN 网站。

若链接的目的地不是一个 URL, 而是一个本地文件, 你可以使用\href 命令:

```
The complete document is \href{manual.pdf}{here}
```

生成的效果为“The complete document is here”。单击“here” 将打开文件manual.pdf (文件名跟依赖于当前文档的位置)。

若文章的作者希望读者可以容易地发邮件给她, 只需要在文档的标题页的\author 命令中使用命令\href:

```
\author{Mary Oetiker <\href{mailto:mary@oetiker.ch}%
  {mary@oetiker.ch}>}
```

注意到这个链接不仅显示在链接中还显示在页面上。下面的链接

```
\href{mailto:mary@oetiker.ch}{Mary Oetiker}
```

在 Acrobat 中可正常使用, 但页面被打印时, 邮箱地址将不可见。

4.7.5 链接的问题

当一个计数器被重新初始化后可能出现下面的信息:

```
! pdfTeX warning (ext4): destination with the same
  identifier (name{page.1}) has been already used,
  duplicate ignored
```

例如, 在book类的文档中使用命令`\mainmatter`就可能出现上面的警告。它在书的第一章重设页码计数器为1, 但是书的序言部分也有页码1, 从而链接到“页码1”不再是唯一的, 故出现了“duplicate has been ignored.”

一个解决的办法是将`plainpages=false`加入到`hyperref`选项中。不幸的是这样只对页码计数器有效。或者冒险使用`hypertexnames=false`, 但是这会使得索引中的页码链接失效。

4.7.6 书签的问题

书签中的文本有时并不会像你想象中的那样显示, 因为书签仅仅是“文本”, 其中可使用的字符要比正常L^AT_EX文件的少得多。`Hyperref`经常遇到这类问题而出现下面的警告:

```
Package hyperref Warning:
Token not allowed in a PDFDocEncoded string:
```

现在你可以通过为书签提供一个文本字符串来解决这个问题, 用

```
\texorpdfstring{TEX text}{Bookmark Text}
```

来替换不正常显示的文本。

数学表达式也是这类问题的典型代表:

```
\section{\texorpdfstring{ $E=mc^2$ }%
{ $E=mc^2$ }}
```

而通常在书签中`\section{ $E=mc^2$ }`显示为“E=mc2”。

颜色的改变也在书签显示中出问题:

```
\section{\textcolor{red}{Red !}}
```

显示的结果为“redRed!”。命令`\textcolor`被忽略。而它的参数(red)被输出。

如果你使用

```
\section{\texorpdfstring{\textcolor{red}{Red !}}{Red\ !}}
```

结果将更美观。

如果你写 unicode (统一字符编码) 类型的文档, 就需添加宏包`hyperref`的选项`unicode`, 这样你就可以在书签中使用 unicode 字符。然后使用`\texorpdfstring`命令将让你有较大范围的字符供选择。

L^AT_EX 和 pdfL^AT_EX 的源文件的兼容性

理想的话，你的文档用L^AT_EX 和 pdfL^AT_EX 编译的效果应该一致，这方面的问题主要是包含的图像。一个简单的解决办法是在命令`\includegraphics` 中不设置加载的图像的扩展名，它们会自动在当前目录寻找一个格式适合的文件，你需要做的是创建相应格式的图像文件。L^AT_EX 会寻找`.eps` 格式，而 pdfL^AT_EX 会尝试包含一个扩展名为`.png`，`.pdf`，`.jpg` 或`.mps`（按这个顺序）的文件。

若你想在 PDF 格式的文件中使用不同的代码，只需要简单地在导言区加入宏包`ifpdf`⁸。但前提是你已经安装了这个宏包，如果没有安装而你又使用 MiK_TE_X 的话，它将在你第一次使用的时候自动安装。这个宏包定义了一个特殊的命令`\ifpdf`，利用它你可很容易编写条件代码。例如，考虑到打印费用，用 PostScript 格式仅使用黑白色，但在线查看的 PDF 格式将是彩色的。

```
\RequirePackage{ifpdf} % running on pdfTeX?
\ifpdf
  \documentclass[a4paper,12pt,pdftex]{book}
\else
  \documentclass[a4paper,12pt,dvips]{book}
\fi

\ifpdf
  \usepackage{lmodern}
\fi
\usepackage[bookmarks, % add hyperlinks
             colorlinks,
             plainpages=false]{hyperref}
\usepackage[T1]{fontenc}
\usepackage[latin1]{inputenc}
\usepackage[english]{babel}
\usepackage{graphicx}
...
```

在上面的例子中，在非 PDF 格式中我也包含了`hyperref` 宏包，这样`\href` 命令在所有情形下都有效，这也使得我不用在每个情况下都使用条件声明。

注意到当前的T_EX 发行版本（例如T_EXLive），通常的T_EX 会根据文档类型的设置自动选择输出 PDF 还是 DVI。如果你使用上面的代码，你仍然可以使用`pdflatex` 命令来得到 PDF 格式的输出或使用`latex` 得到 DVI 格式。

4.8 创建演示文稿

By Daniel Flipo <Daniel.Flipo@univ-lille1.fr>

你可以将你的科学工作成果通过黑板、透明片或者在你的笔记本电脑上直接使用演示文稿软件呈现。

pdfL^AT_EX 和`beamer` 文档类允许你创建 PDF 格式的演示文稿，结果跟你用一天时间制作的 PowerPoint 看上去差不多，但更便携因为 Acrobat Reader 支持更多的系统平台。

`beamer` 文档类使用带参数的宏包`graphicx`、`color` 和`hyperref` 来适应屏幕阅读的演示文稿。

⁸如果你想知道为什么要使用这个宏包，可以参见T_EX FAQ 的这个栏目
<http://www.tex.ac.uk/cgi-bin/texfaq2html?label=ifpdf>.

```
\documentclass[10pt]{beamer}
\mode<beamer>{%
  \usetheme[hideothersubsections,
            right,width=22mm]{Goettingen}
}

\title{Simple Presentation}
\author[D. Flipo]{Daniel Flipo}
\institute{U.S.T.L. \& GUTenberg}
\titlegraphic{\includegraphics[width=20mm]{USTL}}
\date{2005}

\begin{document}

\begin{frame}<handout:0>
  \titlepage
\end{frame}

\section{一个例子}

\begin{frame}
  \frametitle{Things to do on a Sunday Afternoon}
  \begin{block}{One could \ldots}
    \begin{itemize}
      \item walk the dog\ldots \pause
      \item read a book\pause
      \item confuse a cat\pause
    \end{itemize}
  \end{block}
  and many other things
\end{frame}
\end{document}
```

图 4.2 – beamer 文档类的范例。

当你用PDFL^AT_EX 编译图4.2 中的代码时，你将得到一个 PDF 文件，第一页为标题页，第二页有几个栏目，但当你单击你的演示文档时，一次显示一条栏目。

beamer 类创建 PDF 文件的一个优点是直接生成可用的文档，而不像prosp^er 需要先通过一个 PostScript 步骤，也不像ppow^er4 宏包需要一个后加工处理才能生成演示文档。

用beamer 类，你可以用一个源文件生成几种版本。可以在源文件的中括弧中加入特定的选项来生成不同的版本。有下面几种版式：

beamer PDF 屏幕阅读版本；

trans 幻灯片版本；

handout PDF 讲义版本。

默认的版本为beamer，你可以通过设置不同的全局选项来修改，例如：用`\documentclass[10pt,handout]{beamer}` 来生成讲义版本。

演示文稿外观依赖于你选择的主题。你可以选择 beamer 类自带的一个主题，也可以自己定义一个新的主题。详情请参见 beamer 类的帮助文档beameruserguide.pdf。

让我们再来仔细分析图4.2 中的代码。

对于屏幕阅读版本的演示文稿`\mode<beamer>`，我们选择了 *Goettingen* 主题，它将目录合成到导航面板。通过选项控制面板的大小（这个例子采用 22 mm），和确定面板的位置（正文右侧）。选项`hideothersubsections` 显示章节的标题，但只显示当前章节的子节标题。对于`\mode<trans>` 和`\mode<handout>` 的设置也是一样的，它们将出现在它们标准的版面上。

命令`\title{}`，`\author{}`，`\institute{}` 和`\titlegraphic{}` 定义标题页的内容。`\title[]{}` 和`\author[]{}` 的选项允许你定义显示在 *Goettingen* 主题的面板上的标题和作者名。

面板中的标题和子标题由frame 环境外面的命令`\section{}` 和`\subsection{}` 来创建。

屏幕底部的一些微型导航图标也可以让你浏览整个文档。它们的出现不依赖于你选择的主题。

每张幻灯片或每版屏幕的内容放在frame 环境中。利用尖括弧（< 和>）里面的选项，用演示文档的一个版式来定义一个特殊的帧。在这个例子中，第一页不会由于参量`<handout:0>` 而显示为讲义模式。

除了幻灯片的标题页，强烈建议通过命令`\frametitle{}` 来重新设置每一张幻灯片的标题。如果需要，使用block 环境可以来定义子标题，在这个例子中也可体现出来。注意到章节命令`\section{}` 和`\subsection{}` 不在幻灯片上产生输出结果。

列表环境中的命令`\pause` 允许你一个接一个地显示列表栏目的内容。命令：`\only`、`\uncover`、`\alt` 和`\temporal`，可以让你获得其他的一些演示效果。很多情况下，你可以通过尖括弧中的内容来定制演示效果。

无论如何，建议你阅读 beamer 类的文档beameruserguide.pdf 来获得一个全面的了解。这个宏包正在活跃地开发中，去它们的网站

<http://latex-beamer.sourceforge.net/> 可获取最新的信息。

Chapter 5

数学图形

大部分人使用 \LaTeX 来排版文本内容。因其不面向内容和结构的特点给写作提供了巨大的方便，我们还可以有办法从文本描述生成图形输出。此外，大量的 \LaTeX 扩展被开发出来以克服种种限制。在本节中，我们将学习其中的一些。

5.1 概述

`picture` 环境可以在 \LaTeX 里直接设计图形。详细的介绍请参考 *\LaTeX Manual* [1]。一方面，这种方法有严重的局限性，比如线段的斜率和圆的半径只能在一个很小的范围内取值。另一方面， $\text{\LaTeX} 2_{\epsilon}$ 的 `picture` 环境提供了 `\qbezier` 命令，“q”表示“quadratic”。许多常用的曲线如圆、椭圆、或者悬链线都可以用二次 Bézier 曲线得到令人满意的近似，虽然这可能需要一些辛苦的数学准备。另外，如果有一种编程语言如 Java 能用来生成 \LaTeX 源文档的`\qbezier` 模块，`picture` 环境会更强大。

虽然直接在 \LaTeX 里设计图形的方法有严重的局限性而且通常比较繁琐，但它还是很有用的。这份文档就是用它才变得体积很小，不需要插入额外的图片。

一些宏包，如`epic` 和`hepic` (*The \LaTeX Companion* [3]里有介绍)，或者`pstricks` 可以排除`picture` 环境的局限，并大大地增强了 \LaTeX 的图形功能。

跟前两个宏包只是加强了`picture` 环境不同，`pstricks` 宏包有自己的绘图环境，`pspicture`。`pstricks` 的强大之处在于它广泛应用了POSTSCRIPT。另外，许多宏包可以用来处理专门的问题。其一是`XY-pic`，本章最后会讲到它。*The \LaTeX Graphics Companion* [4]（勿与*The \LaTeX Companion* [3]混淆）里详细介绍了大量的宏包。

\LaTeX 最强大的图形工具可能是`MetaPost`，Donald E. Knuth 编写的`METAFONT` 的孪生兄弟。`MetaPost` 使用非常强大的数学编程语言：`METAFONT`。与`METAFONT` 生成点阵图片不同，`MetaPost` 生成的是封装的POSTSCRIPT 文件，可以导入 \LaTeX 中。其介绍可以看 *A User's Manual for MetaPost* [15]，或者[17]。

关于 \LaTeX 和 \TeX 图形（以及字体）支持方法的详细讨论请参考 *\TeX Unbound* [16]。

5.2 `picture` 环境

5.2.1 基本命令

一个`picture` 环境¹可以用下面两个命令中的一个来创建

```
\begin{picture}(x,y)...\end{picture}
```

或者

```
\begin{picture}(x,y)(x_0,y_0)...\end{picture}
```

数字 x, y, x_0, y_0 是相对于`\unitlength` 而言的, 任何时候 (除了在`picture` 环境之内以外), 都可以使用命令如

```
\setlength{\unitlength}{1.2cm}
```

来改变。`\unitlength` 的默认值是1 pt。第一个数对, (x, y) , 在文档中为图形保留一个矩形的区域。可选的第二个数对, (x_0, y_0) , 为矩形左下角指派任意的坐标。

大多数的绘图命令是下面两种格式之一

```
\put(x,y){object}
```

或者

```
\multiput(x,y)(\Delta x,\Delta y){n}{object}
```

Bézier 曲线是一个例外。它们需要用命令

```
\qbezier(x_1,y_1)(x_2,y_2)(x_3,y_3)
```

来画。

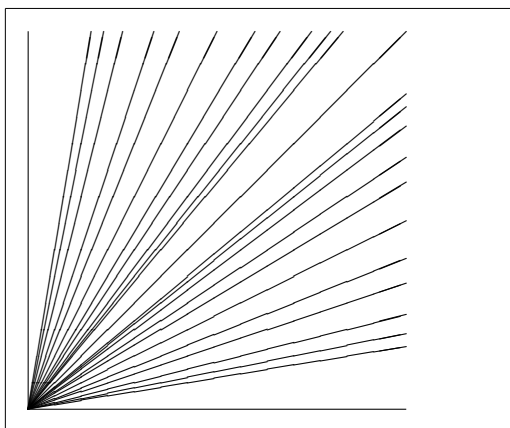
¹信不信由你, `picture` 环境仅需标准的 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X } 2_{\epsilon}$, “开箱即用”, 无需载入宏包。

5.2.2 线段

```

\setlength{\unitlength}{5cm}
\begin{picture}(1,1)
  \put(0,0){\line(0,1){1}}
  \put(0,0){\line(1,0){1}}
  \put(0,0){\line(1,1){1}}
  \put(0,0){\line(1,2){.5}}
  \put(0,0){\line(1,3){.3333}}
  \put(0,0){\line(1,4){.25}}
  \put(0,0){\line(1,5){.2}}
  \put(0,0){\line(1,6){.1667}}
  \put(0,0){\line(2,1){1}}
  \put(0,0){\line(2,3){.6667}}
  \put(0,0){\line(2,5){.4}}
  \put(0,0){\line(3,1){1}}
  \put(0,0){\line(3,2){1}}
  \put(0,0){\line(3,4){.75}}
  \put(0,0){\line(3,5){.6}}
  \put(0,0){\line(4,1){1}}
  \put(0,0){\line(4,3){1}}
  \put(0,0){\line(4,5){.8}}
  \put(0,0){\line(5,1){1}}
  \put(0,0){\line(5,2){1}}
  \put(0,0){\line(5,3){1}}
  \put(0,0){\line(5,4){1}}
  \put(0,0){\line(5,6){.8333}}
  \put(0,0){\line(6,1){1}}
  \put(0,0){\line(6,5){1}}
\end{picture}

```



线段用命令

```
\put(x,y){\line(x1,y1){length}}
```

来画。命令`\line`有两个参量：

1. 一个方向向量，
2. 一个长度。

方向向量需由以下整数构成

$$-6, -5, \dots, 5, 6,$$

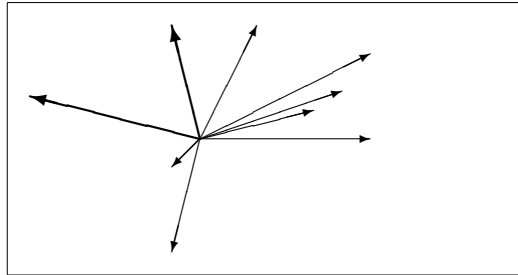
而且它们需要互质（除 1 以外，没有公约数），图形显示了第一象限中所有 25 个可能的斜率值。长度是相对于`\unitlength`来说的。长度的参量当一个垂直线段时是垂直坐标，其他情况都是水平坐标。

5.2.3 箭头

```

\setlength{\unitlength}{0.75mm}
\begin{picture}(60,40)
  \put(30,20){\vector(1,0){30}}
  \put(30,20){\vector(4,1){20}}
  \put(30,20){\vector(3,1){25}}
  \put(30,20){\vector(2,1){30}}
  \put(30,20){\vector(1,2){10}}
  \thicklines
  \put(30,20){\vector(-4,1){30}}
  \put(30,20){\vector(-1,4){5}}
  \thinlines
  \put(30,20){\vector(-1,-1){5}}
  \put(30,20){\vector(-1,-4){5}}
\end{picture}

```



画箭头要用命令

```
\put(x,y){\vector(x1,y1){length}}
```

箭头的方向向量元素比线段的限制更严格，需由以下整数构成

$-4, -3, \dots, 3, 4.$

而且需要互质（除 1 以外，没有公约数）。注意命令 `\thicklines` 对指向左上方的两个箭头产生的效果。

5.2.4 圆

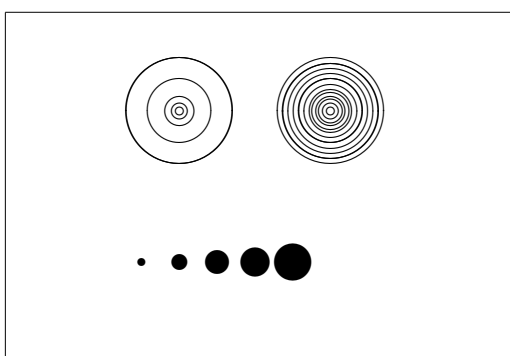
```

\setlength{\unitlength}{1mm}
\begin{picture}(60, 40)
  \put(20,30){\circle{1}}
  \put(20,30){\circle{2}}
  \put(20,30){\circle{4}}
  \put(20,30){\circle{8}}
  \put(20,30){\circle{16}}
  \put(20,30){\circle{32}}

  \put(40,30){\circle{1}}
  \put(40,30){\circle{2}}
  \put(40,30){\circle{3}}
  \put(40,30){\circle{4}}
  \put(40,30){\circle{5}}
  \put(40,30){\circle{6}}
  \put(40,30){\circle{7}}
  \put(40,30){\circle{8}}
  \put(40,30){\circle{9}}
  \put(40,30){\circle{10}}
  \put(40,30){\circle{11}}
  \put(40,30){\circle{12}}
  \put(40,30){\circle{13}}
  \put(40,30){\circle{14}}

  \put(15,10){\circle*{1}}
  \put(20,10){\circle*{2}}
  \put(25,10){\circle*{3}}
  \put(30,10){\circle*{4}}
  \put(35,10){\circle*{5}}
\end{picture}

```



命令

```
\put( $x$ ,  $y$ ){\circle{ $diameter$ }}
```

画了一个圆心在 (x, y) 直径（不是半径）为 $diameter$ 的圆。picture 环境只允许直径最大是 14mm，而且即使在这个限制之下，也不是所有的直径都可获得。命令 `\circle*` 生成圆盘（填充的圆形）。

跟线段的情况一样，你可能需要其他宏包的帮助，比如 `heepic` 或者 `pstricks`。这些宏包的详细说明请参考 *The L^AT_EX Graphics Companion* [4]。

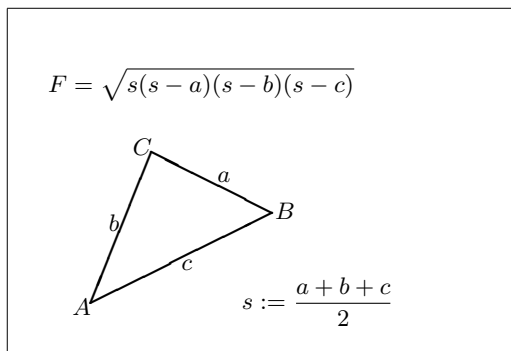
picture 环境还有另外一个可能。如果你不怕麻烦的必要的计算（或者交给一个程序来处理），任意的圆和矩形都可以由二次 Bézier 曲线拼成。请看例子 *Graphics in L^AT_EX 2_ε* [17] 以及 Java 源文件。

5.2.5 文本与公式

```

\setlength{\unitlength}{0.8cm}
\begin{picture}(6,5)
\thicklines
\put(1,0.5){\line(2,1){3}}
\put(4,2){\line(-2,1){2}}
\put(2,3){\line(-2,-5){1}}
\put(0.7,0.3){\A}
\put(4.05,1.9){\B}
\put(1.7,2.95){\C}
\put(3.1,2.5){\a}
\put(1.3,1.7){\b}
\put(2.5,1.05){\c}
\put(0.3,4){F=}
\sqrt{s(s-a)(s-b)(s-c)}
\put(3.5,0.4){\displaystyle
s:=\frac{a+b+c}{2}}
\end{picture}

```



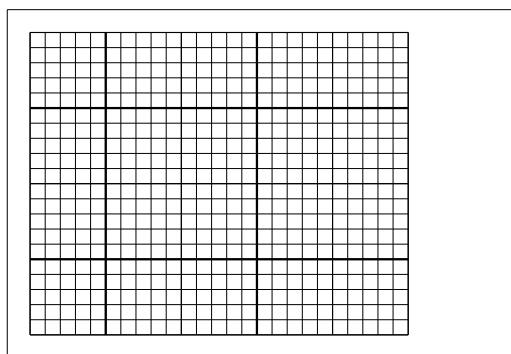
如本例所示，文本与公式可以使用`\put`命令按照正常方式在`picture`环境中使用。

5.2.6 `\multiput`与`\linethickness`

```

\setlength{\unitlength}{2mm}
\begin{picture}(30,20)
\linethickness{0.075mm}
\multiput(0,0)(1,0){26}%
{\line(0,1){20}}
\multiput(0,0)(0,1){21}%
{\line(1,0){25}}
\linethickness{0.15mm}
\multiput(0,0)(5,0){6}%
{\line(0,1){20}}
\multiput(0,0)(0,5){5}%
{\line(1,0){25}}
\linethickness{0.3mm}
\multiput(5,0)(10,0){2}%
{\line(0,1){20}}
\multiput(0,5)(0,10){2}%
{\line(1,0){25}}
\end{picture}

```



命令

```
\multiput(x,y)(\Delta x,\Delta y){n}{object}
```

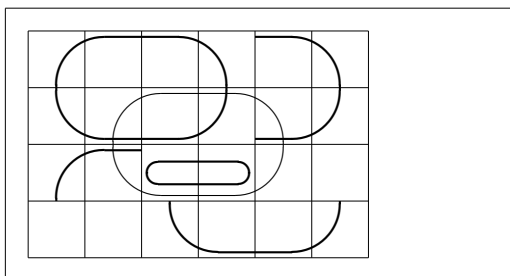
有 4 个参量：初始点，从一个对象到下一个的平移向量，对象的数目和要绘制的对象。命令`\linethickness`可作用于水平和垂直方向的线段，但不能作用于倾斜的线段和圆。然而，该命令可作用于二次 Bézier 曲线。

5.2.7 椭圆

```

\setlength{\unitlength}{0.75cm}
\begin{picture}(6,4)
  \linethickness{0.075mm}
  \multiput(0,0)(1,0){7}%
    {\line(0,1){4}}
  \multiput(0,0)(0,1){5}%
    {\line(1,0){6}}
  \thicklines
  \put(2,3){\oval(3,1.8)}
  \thinlines
  \put(3,2){\oval(3,1.8)}
  \thicklines
  \put(2,1){\oval(3,1.8)[t1]}
  \put(4,1){\oval(3,1.8)[b]}
  \put(4,3){\oval(3,1.8)[r]}
  \put(3,1.5){\oval(1.8,0.4)}
\end{picture}

```



命令

```
\put(x,y){\oval(w,h)}
```

或

```
\put(x,y){\oval(w,h)[position]}
```

可以产生一个中心在 (x, y) 处、宽为 w 高为 h 的椭圆。如本例所示，可选参量 $position$ 可以是 **b**, **t**, **l**, **r**, 分别表示仅绘制椭圆的“下部”、“上部”、“左部”和“右部”，如例所示，这些参数可以进行组合。

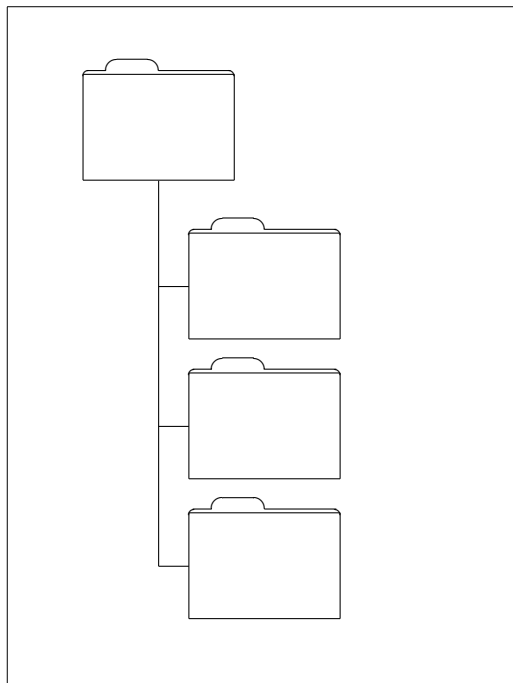
以下两类命令可以控制线宽：一类为 `\linethickness{length}`，另一类为 `\thinlines` 与 `\thicklines`。命令 `\linethickness{length}` 仅对水平和垂直直线（及二次 Bézier 曲线）有作用，`\thinlines` 与 `\thicklines` 则可以作用于倾斜的线段、圆和椭圆。

5.2.8 重复使用预定义的图形盒子

```

\setlength{\unitlength}{0.5mm}
\begin{picture}(120,168)
\newsavebox{\foldera}
\savebox{\foldera}
(40,32)[bl]{% definition
\multiput(0,0)(0,28){2}
{\line(1,0){40}}
\multiput(0,0)(40,0){2}
{\line(0,1){28}}
\put(1,28){\oval(2,2)[t1]}
\put(1,29){\line(1,0){5}}
\put(9,29){\oval(6,6)[t1]}
\put(9,32){\line(1,0){8}}
\put(17,29){\oval(6,6)[tr]}
\put(20,29){\line(1,0){19}}
\put(39,28){\oval(2,2)[tr]}
}
\newsavebox{\folderb}
\savebox{\folderb}
(40,32)[l]{% definition
\put(0,14){\line(1,0){8}}
\put(8,0){\usebox{\foldera}}
}
\put(34,26){\line(0,1){102}}
\put(14,128){\usebox{\foldera}}
\multiput(34,86)(0,-37){3}
{\usebox{\folderb}}
\end{picture}

```



一个图形盒子可以使用命令

```
\newsavebox{name}
```

进行声明，然后使用命令

```
\savebox{name}(width,height)[position]{content}
```

进行定义，最后使用命令

```
\put(x,y)\usebox{name}
```

进行任意次数的重复绘制。

可选参数 $position$ 的作用是定义图形存放盒子的“锚点”。在本例中该参数被设置为 bl ，从而将锚点设置为图形存放盒子的左下角。其他的位置描述有 t 和 r ，分别表示“上”和“右”。

参量 $name$ 指明了 \LaTeX 存储槽，揭示了其命令本质（在本例中指反斜线）。图形盒子可以嵌套：在本例中， \foldera 被用在了 \folderb 的定义中。

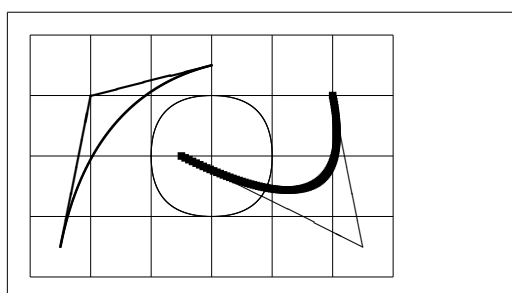
由于命令 \line 在线段长度小于大约 3mm 的时候不能正常工作，所以必须使用命令 \oval 。

5.2.9 二次 Bézier 曲线

```

\setlength{\unitlength}{0.8cm}
\begin{picture}(6,4)
  \linethickness{0.075mm}
  \multiput(0,0)(1,0){7}
    {\line(0,1){4}}
  \multiput(0,0)(0,1){5}
    {\line(1,0){6}}
  \thicklines
  \put(0.5,0.5){\line(1,5){0.5}}
  \put(1,3){\line(4,1){2}}
  \qbezier(0.5,0.5)(1,3)(3,3.5)
  \thinlines
  \put(2.5,2){\line(2,-1){3}}
  \put(5.5,0.5){\line(-1,5){0.5}}
  \linethickness{1mm}
  \qbezier(2.5,2)(5.5,0.5)(5,3)
  \thinlines
  \qbezier(4,2)(4,3)(3,3)
  \qbezier(3,3)(2,3)(2,2)
  \qbezier(2,2)(2,1)(3,1)
  \qbezier(3,1)(4,1)(4,2)
\end{picture}

```



如本例所示，将圆分割为 4 条二次 Bézier 曲线的效果不能令人满意，至少需要 8 条。该图再一次展示了命令 `\linethickness` 对水平或垂直直线以及命令 `\thinlines` 和 `\thicklines` 对倾斜线段的影响。该例同时显示：这两类命令都会影响二次 Bézier 曲线，每一条命令都会覆盖以前所有命令。

令 $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$ 和 m_1, m_2 分别表示一条二次 Bézier 曲线的两个端点及其对应斜率。中间控制点 $S = (x, y)$ 则由下述方程给出

$$\begin{cases} x = \frac{m_2 x_2 - m_1 x_1 - (y_2 - y_1)}{m_2 - m_1}, \\ y = y_i + m_i(x - x_i) \quad (i = 1, 2). \end{cases} \quad (5.1)$$

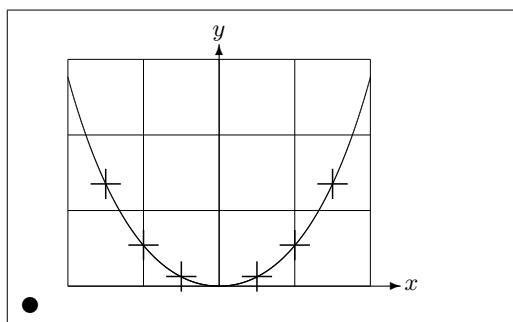
关于生成必要的 `\qbezier` 命令的 Java 程序参见 *Graphics in L^AT_EX 2_ε* [17]。

5.2.10 悬链线

```

\setlength{\unitlength}{1cm}
\begin{picture}(4.3, 3.6)
(-2.5,-0.25)
\put(-2,0){\vector(1,0){4.4}}
\put(2.45,-.05){$x$}
\put(0,0){\vector(0,1){3.2}}
\put(0,3.35){\makebox(0,0){$y$}}
\qBezier(0.0,0.0)(1.2384,0.0)
(2.0,2.7622)
\qBezier(0.0,0.0)(-1.2384,0.0)
(-2.0,2.7622)
\linethickness{.075mm}
\multiput(-2,0)(1,0){5}
{\line(0,1){3}}
\multiput(-2,0)(0,1){4}
{\line(1,0){4}}
\linethickness{.2mm}
\put(.3,.12763){\line(1,0){.4}}
\put(.5,-.07237){\line(0,1){.4}}
\put(-.7,.12763){\line(1,0){.4}}
\put(-.5,-.07237){\line(0,1){.4}}
\put(.8,.54308){\line(1,0){.4}}
\put(1,.34308){\line(0,1){.4}}
\put(-1.2,.54308){\line(1,0){.4}}
\put(-1,.34308){\line(0,1){.4}}
\put(1.3,1.35241){\line(1,0){.4}}
\put(1.5,1.15241){\line(0,1){.4}}
\put(-1.7,1.35241){\line(1,0){.4}}
\put(-1.5,1.15241){\line(0,1){.4}}
\put(-2.5,-0.25){\circle*{0.2}}
\end{picture}

```



在本图中，悬链线 $y = \cosh x - 1$ 对称的两半由二次 Bézier 曲线分别近似地绘成。曲线的右半部分终止于点 $(2, 2.7622)$ ，对应的斜率为 $m = 3.6269$ 。再次使用公式(5.1)，我们可以计算中间控制点。计算结果为 $(1.2384, 0)$ 和 $(-1.2384, 0)$ 。图中的十字为真正的悬链线上的点。误差小于百分之一，很难被发现。

该例指出了命令 `\begin{picture}` 的可选参数的用法。该图通过使用命令

```
\begin{picture}(4.3,3.6)(-2.5,-0.25)
```

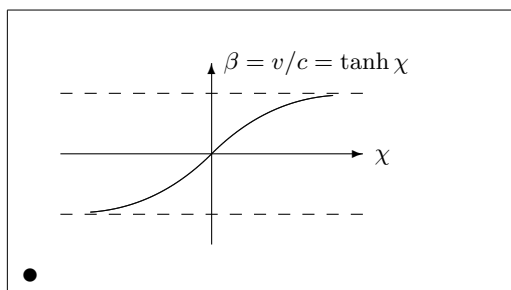
定义了方便的“数学”坐标：左下角（由黑色圆点标出）坐标是 $(-2.5, -0.25)$ 。

5.2.11 坐标的相对性

```

\setlength{\unitlength}{0.8cm}
\begin{picture}(6,4)(-3,-2)
  \put(-2.5,0){\vector(1,0){5}}
  \put(2.7,-0.1){\chi}
  \put(0,-1.5){\vector(0,1){3}}
  \multiput(-2.5,1)(0.4,0){13}
    {\line(1,0){0.2}}
  \multiput(-2.5,-1)(0.4,0){13}
    {\line(1,0){0.2}}
  \put(0.2,1.4)
    {\beta=v/c=\tanh\chi}
  \qbezier(0,0)(0.8853,0.8853)
    (2,0.9640)
  \qbezier(0,0)(-0.8853,-0.8853)
    (-2,-0.9640)
  \put(-3,-2){\circle*{0.2}}
\end{picture}

```



公式(5.1)给出了两条 Bézier 曲线的控制点。正向分支由 $P_1 = (0, 0)$, $m_1 = 1$ 和 $P_2 = (2, \tanh 2)$, $m_2 = 1/\cosh^2 2$ 确定。与前例相同, 本图也定义了在教学上方便的坐标, 左下角的坐标是 $(-3, -2)$ (黑点)。

5.3 Xy-pic

By Alberto Manuel Brandão Simões <albio@alfarrabio.di.uminho.pt>

xy 是绘制流程图的专用宏包。要想使用它, 只需在导言区加上:

```
\usepackage[options]{xy}
```

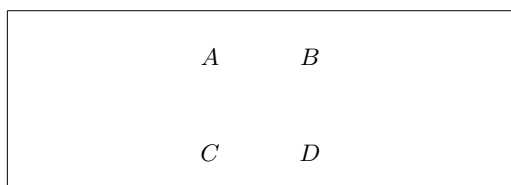
options 列出你需要载入的 Xy-pic 的选项。这些选项基本上被用于调试这个宏包的使用。建议你使用 `all`, 可以让 L^AT_EX 载入 Xy 的所有命令。

Xy-pic 流程图被绘制在一幅以矩阵定位的画布上, 每一个流程图元素被放在矩阵的一个单元中:

```

\begin{displaymath}
\mathrm{xymatrix}{A & B \\
& C & D}
\end{displaymath}

```

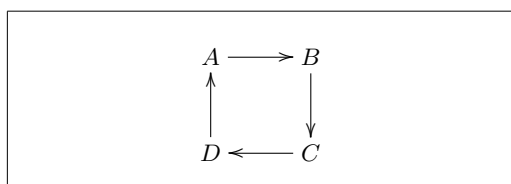


命令 `\xymatrix` 必须置于数学模式中。这里, 我们设定了一个两行两列的矩阵。为了画出流程, 我们只需要使用命令 `\ar` 增加带方向的箭头即可。

```

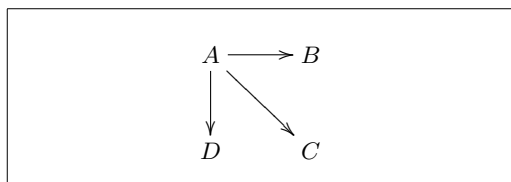
\begin{displaymath}
\mathrm{xymatrix}{A \ar[r] & B \ar[d] \\
D \ar[u] & C \ar[l]}
\end{displaymath}

```



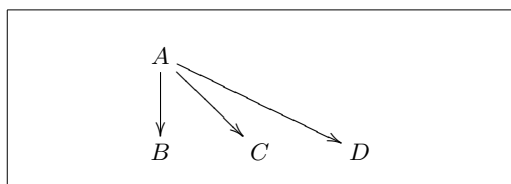
箭头命令要放在其出发的那个单元里。参量是箭头的方向（u: 上, d: 下, r: 右以及l: 左）。

```
\begin{displaymath}
\matrix{
A \ar[d] \ar[dr] \ar[r] & B \\
D & C}
\end{displaymath}
```



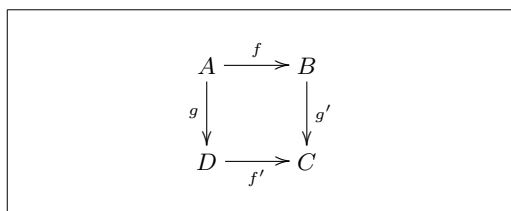
要画对角线，可以指定不只一个方向参量。实际上，你还可以重复同一个方向来得到更大的箭头。

```
\begin{displaymath}
\matrix{
A \ar[d] \ar[dr] \ar[dr] & & \\
B & C & D}
\end{displaymath}
```



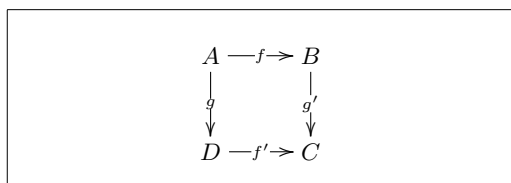
我们还可以绘制一些更有趣的流程图，给箭头加上标签，只需要使用普通的上标和下标。

```
\begin{displaymath}
\matrix{
A \ar[r]^f \ar[d]_g & \\
B \ar[d]^{g'} & \\
D \ar[r]_{f'} & C}
\end{displaymath}
```



如图所示，就像数学模式里一样使用上下标。唯一的区别在于：上标表示放在“箭头的上方”，下标表示放在“箭头的下方”。把文本放到箭头上可以用|。

```
\begin{displaymath}
\matrix{
A \ar[r]|f \ar[d]|g & \\
B \ar[d]|g' & \\
D \ar[r]|f' & C}
\end{displaymath}
```



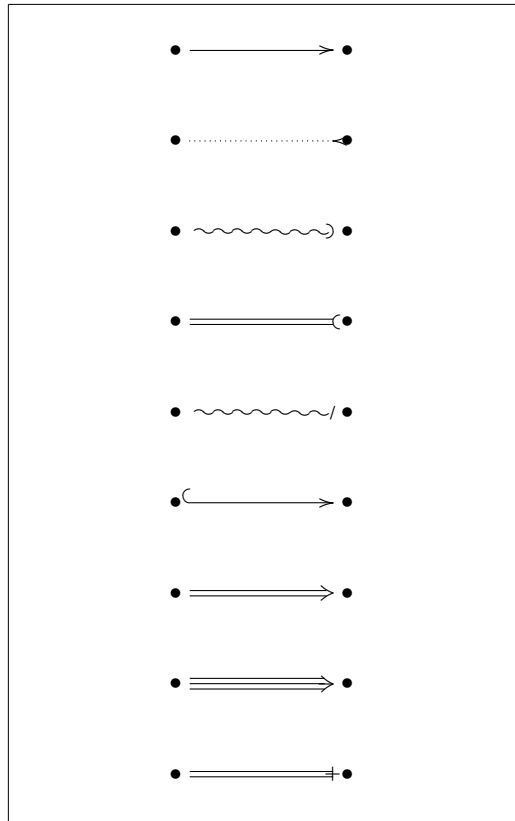
绘制空心箭头的命令是`\ar[...]\hole`。

某些情况下，需要区分不同类型的箭头。可以给它们标上标签，或者使用不同的外观来实现：


```

\shorthandoff{"}
\begin{displaymath}
\ymatrix{
\bullet\ar@{->}[rr] && \bullet\\
\bullet\ar@{.<}[rr] && \bullet\\
\bullet\ar@{~}[rr] && \bullet\\
\bullet\ar@{=}([rr] && \bullet\\
\bullet\ar@{~/}[rr] && \bullet\\
\bullet\ar@{^{}(->)[rr] && \bullet\\
\bullet\ar@2{->}[rr] && \bullet\\
\bullet\ar@3{->}[rr] && \bullet\\
\bullet\ar@{=+}[rr] && \bullet
}
\end{displaymath}
\shorthandon{"}

```

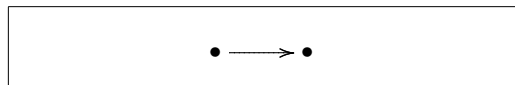


注意下面两幅流程图的区别:

```

\begin{displaymath}
\ymatrix{
\bullet \ar[r] & \\
& \bullet \ar@{.>}[r] & \\
\bullet & & \\
}
\end{displaymath}

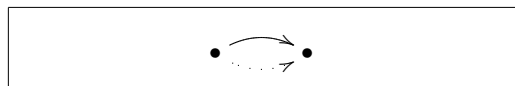
```



```

\begin{displaymath}
\ymatrix{
\bullet \ar@{~/}[r] & \\
& \bullet \ar@/_@{.>}[r] & \\
\bullet & & \\
}
\end{displaymath}

```



两条斜线间的修饰元素决定了曲线应该如何被画出。Xy-pic 提供了很多办法来改变曲线的形状；更详细的内容请参考Xy-pic 的文档。

Chapter 6

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，但它们遵循了高质量排版的已有规则，这些规则可以使得文档易读，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，或者现有的命令产生的输出和你想要的不同。

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

6.1 新建命令、环境和宏包

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矩形框里，并且出现在本书最后的索引中。我并没有直接使用所需的 \LaTeX 命令来实现这个，而是创建了一个宏包(package)，并在其中定义了我所需要的命令和环境。现在我可以简单的写：

```
\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`，这个命令负责输出命令的名字，并且在索引中添加相应的条目。你可以在本书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。

一旦我不再喜欢在一个矩形框中排版命令，我可以轻松的改变`lscommand` 环境的定义，来创建新的外观。跟追踪并修改所有使用原始的 \LaTeX 命令在文字周围画框的地方相比，这种做法容易得多。

6.1.1 新建命令

为了建立你自己的命令，可以使用如下的命令：

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

基本上，这个命令有两个参量，第一个 *name* 是你想要建立的命令的名称，第二个 *definition* 是命令的定义。方括号里的参数 *num* 是可选的，用于指定新命令所需的参量数目（最多 9 个）。如果不给出这个参数，默认就是 0，也就是新建的命令不要任何参量。

接下来的两个例子有助你的理解。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”的简写。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
LATEX 2ε” ... “The not so Short Introduc-
tion to LATEX 2ε”
```

下一个例子演示了如何建立一个接受单一参数的命令。在命令的定义中，标记 `#1` 将被你指定的参量所代替。如果你想使用多个参量，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 不允许你新建一个与现有命令重名的命令。如果你确实需要这么做，有一个专门的命令用于处理这种情况：`\renewcommand`。它使用与命令 `\newcommand` 相同的语法。

在某些情况之下，你可能会希望使用 `\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 将会悄悄忽略原有的那个。

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 4 页可以获得更多这方面的信息。

6.1.2 新建环境

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}
```

同样地，`\newenvironment` 命有一个可选的参量。在 *before* 中的内容将在此环境包含的文本之前处理，而在 *after* 中的内容将在遇到 `\end{name}` 命令时处理。

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 ... ■

参量`num`的使用方式与`\newcommand`命令相同。 \LaTeX 还同样保证你不会不小心新建重名的环境。如果你确实希望改变一个现有的环境,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97页,`\stretch`命令的解释可以参看第91页,关于`\hspace`的信息可以在第91页找到。

6.1.3 额外的空白间距

当创建新的环境时,你或许会为遇到额外的空白间距而烦恼,这些间距可能产生严重的后果。比如当你建立一个标题环境,既不要自身的缩进也不要紧接着的下一段缩进时,在`begin`中加入命令`\ignorespaces`会使新环境忽略执行`begin`之后遇到的一切空白间距,而`end`就需要耍个小花招,因为我们要等到环境结束后才开始处理。使用`\ignorespacesafterend`, \LaTeX 会在`end`处理完毕后,产生一个`\ignorespaces`。

```
\newenvironment{simple}%
{\noindent}%
{\par\noindent}

\begin{simple}
See the space\to the left.
\end{simple}
Same\here.
```

See the space
to the left.

Same
here.

```
\newenvironment{correct}%
{\noindent\ignorespaces}%
{\par\noindent%
 \ignorespacesafterend}

\begin{correct}
No space\to the left.
\end{correct}
Same\here.
```

No space
to the left.

Same
here.

6.1.4 命令行的 \LaTeX

如果使用类 Unix 的操作系统工作,你可能会在使用 Makefiles 建立你的 \LaTeX 项目。那样的话,用命令行参数操控 \LaTeX 来创建同一份文档的不同版本可是十

分有趣的。如果把下列配置写入你的文档：

```
\usepackage{ifthen}
\ifthenelse{\equal{\blackandwhite}{true}}{
  % "black and white" mode; do something..
}{
  % "color" mode; do something different..
}
```

现在，你可以像这样来操作 L^AT_EX：

```
latex '\newcommand{\blackandwhite}{true}\input{test.tex}'
```

首先，定义命令 `\blackandwhite`，然后使用 `input` 来读入实际的文档。要创建彩色版本文档，需要设定 `\blackandwhite` 为 `false`。

6.1.5 自建宏包

如果你定义了很多新的环境和命令，你的文档的导言部分将变得相当长，在这种情况下，建立一个新的 L^AT_EX 宏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。然后你可以在文档中使用 `\usepackage` 命令来引入自定义宏包。

```
% 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 6.1 – 宏包样例。

写一个宏包的基本工作就是将你原本很长的文档导言内容拷贝到另外一个的文件中去，这个文件需要以 `.sty` 结尾。你还加入一个专用的命令：

```
\ProvidesPackage{package name}
```

这个命令应该放在你的宏包的最前面。`\ProvidesPackage` 告诉 L^AT_EX 宏包的名称从而让 L^AT_EX 在你尝试两次引入同一个宏包的时候给出一个明显的错误信息，图 6.1 给出了一个小的宏包示例，其中包含了我们之前定义的一些命令。

6.2 字体和字号

6.2.1 字体变换命令

L^AT_EX 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，你可能会想要手动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 6.1 和表 6.2 中列出的那些命令。每个字体的实际大小是一个设计问题，并且它依赖于文档所使用的文档类及其选项。表 6.3 列出了这些命令在标准文档类中的绝对 pt 大小。

```
{\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*.

L^AT_EX 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退出**数学模式**，然后输入一些正常的文字。如果你希望改变数学公式本身所使用的字体，L^AT_EX 提供了另外一套命令。参看表6.4。

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** (group)。组限制了大多数L^AT_EX 命令的作用范围。

```
He likes {\LARGE large and
\small small} letters}.
```

He likes large and small letters.

如果段落在字体的作用范围中结束，那么字号命令还将改变段落中行距。因此用于分组的反向大括号} 不应该太早出现。注意下面两个例子中\par 命令的位置¹。

¹\par 相当于一个空行

表 6.1 – 字体。

<code>\textrm{...}</code>	roman	<code>\textsf{...}</code>	sans serif
<code>\texttt{...}</code>	typewriter		
<code>\textmd{...}</code>	medium	<code>\textbf{...}</code>	bold face
<code>\textup{...}</code>	upright	<code>\textit{...}</code>	<i>italic</i>
<code>\textsl{...}</code>	<i>slanted</i>	<code>\textsc{...}</code>	SMALL CAPS
<code>\emph{...}</code>	<i>emphasized</i>	<code>\textnormal{...}</code>	document font

表 6.2 – 字号。

<code>\tiny</code>	tiny font	<code>\Large</code>	larger font
<code>\scriptsize</code>	very small font	<code>\LARGE</code>	very large font
<code>\footnotesize</code>	quite small font	<code>\huge</code>	huge
<code>\small</code>	small font	<code>\Huge</code>	largest
<code>\normalsize</code>	normal font		
<code>\large</code>	large font		

表 6.3 – 标准文档类中的绝对 pt 大小。

大小	10pt (默认)	11pt 选项	12pt 选项
<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

表 6.4 – 数学字体。

<code>\mathrm{...}</code>	Roman Font
<code>\mathbf{...}</code>	Boldface Font
<code>\mathsf{...}</code>	Sans Serif Font
<code>\mathtt{...}</code>	Typewriter Font
<code>\mathit{...}</code>	<i>Italic Font</i>
<code>\mathcal{...}</code>	CALIGRAPHIC FONT
<code>\mathnormal{...}</code>	<i>Normal Font</i>


```
{\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，你可能应该使用字体变换命令的环境语法。

```
\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 ...

这将使你从一堆大括号中解脱出来。

6.2.2 战战兢兢，如履薄冰

正如本章开头曾经说过的那样，在你的文档中直接运用这些命令来修改格式是非常危险的事情，因为这种做法和 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 的基础理念相反。在编写 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 文档的时候，要始终注意文章逻辑标记和样式标识的分离。也就是如果你在文章的多个地方采用某种特殊的格式来排版一类经常使用的内容，就应该使用 $\backslash\text{newcommand}$ 来定义一个逻辑封装命令，并通过这个命令来修改相应的表现格式。

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

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

这种方法具有一个明显的优点，你可以在以后决定使用一些不是很有把握实现的特别外观并使之不同于 $\backslash\text{textbf}$ ，那时你就不需要遍历你的整篇文章来找出所有 $\backslash\text{textbf}$ 的地方，然后一个一个地确定是不是要改成没有把握的外观。

6.2.3 建议

总结这一章中关于字体和字号的命令，下面是一个简短的建议：

Remember! *The **MORE** fonts **YOU** use in a document, the more READABLE and beautiful it becomes.*
记住！你使用的字体越多，文章看起来就越易读越美观。

6.3 间距

6.3.1 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}
```

如`\linespread{1.3}`产生 1.5 倍行距，而`\linespread{1.6}`则产生双倍行距。缺省情况下的行距为 1。

注意`\linespread`的效果相当夸张而且不适合出版工作。因此如果你很想改变行距可能会希望使用如下的命令：

```
\setlength{\baselineskip}{1.5\baselineskip}
```

```
{\setlength{\baselineskip}%
  {1.5\baselineskip}
This paragraph is typeset with
the baseline skip set to 1.5 of
what it was before. Note the par
command at the end of the
paragraph.\par}
```

This paragraph has a clear purpose, it shows that after the curly brace has been closed, everything is back to normal.

This paragraph is typeset with the baseline skip set to 1.5 of what it was before. Note the par command at the end of the paragraph.

This paragraph has a clear purpose, it shows that after the curly brace has been closed, everything is back to normal.

6.3.2 段落格式

在L^AT_EX 中，有两个参数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间距，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T_EX 按照指定大小压缩和伸展段落间距。为了使得段落正确的显示在页面之上，T_EX 将在 0.8ex 到 1.5ex 之间调整段落间距。

在欧洲大陆，段落通常用一些空白分隔并且一般首行不缩进。但是值得注意的是，这也会影响目录。目录的行距也会变得非常疏松。为了避免这种情况，你可能需要将上面的两个命令从导言中移到文档中`\tableofcontents` 以下适合的位置，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白来分离段落。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

²为了缩进章节标题之后的第一个段落，可以使用`indentfirst`包。

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
```

当文档以正文而不是章节命令开始的时候，这个命令会提供方便。

6.3.3 水平间距

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}
```

如果这个水平间距即使在行首或者行末也应该保持的话，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。最重要的长度单位在表6.5中列了出来。

```
This\hspace{1.5cm}is a space  
of 1.5 cm.
```

```
This          is a space of 1.5 cm.
```

下面的命令

```
\stretch{n}
```

将产生一个特殊的橡皮长度：一个能把行内剩余所有空隙填满的空白。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
```

当在正文中使用水平间距的时候，相对于字号来调整间距大小会更有道理。这可以通过使用与文本有关的单位`em`和`ex`来实现：

```
{\Large{}big\hspace{1em}y}\  
{\tiny{}tin\hspace{1em}y}
```

```
big y  
tin y
```

表 6.5 – T_EX 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	□

6.3.4 垂直间距

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应该在于页的顶部和末尾也保留下来，那么使用这个命令的星号版本 `\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}}
```

```
This goes onto the last line of the page.\pagebreak
```

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间距。

6.4 页面布局

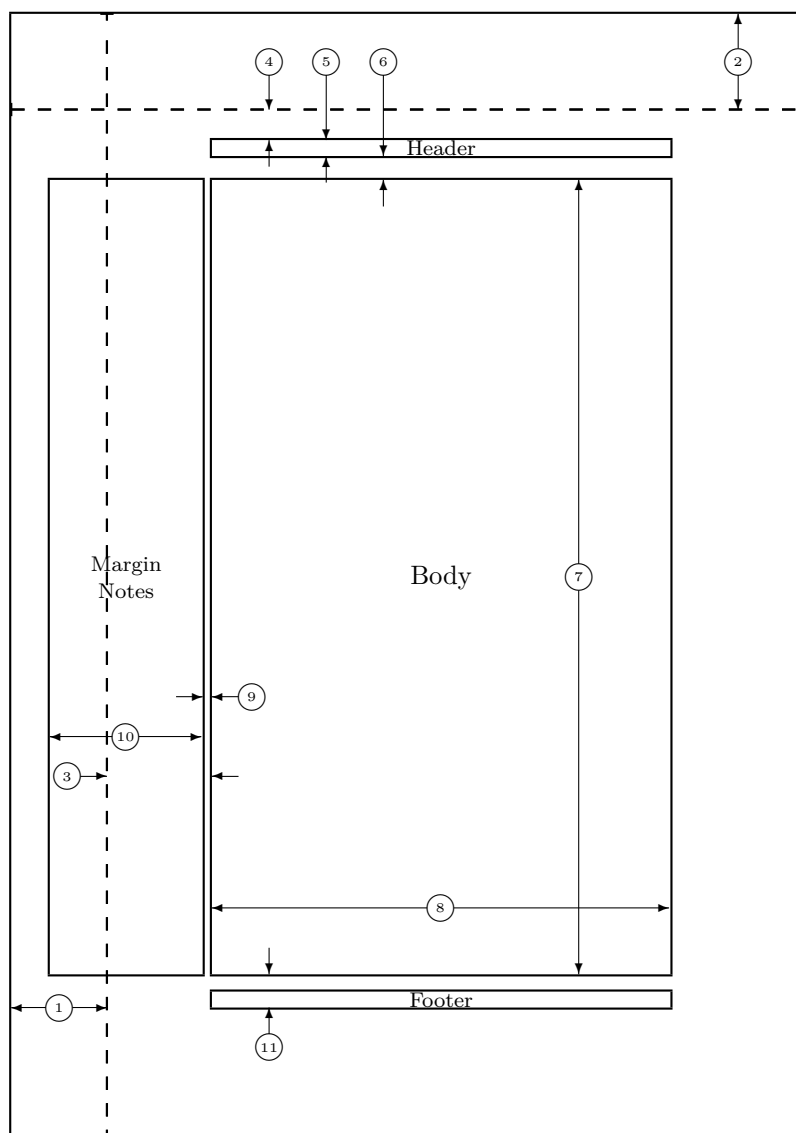
L^AT_EX 2_ε 允许你在 `\documentclass` 命令中指定纸张尺寸 (paper size)。然后它将自动的选择合适的页边距。但有些时候你可能不满意 L^AT_EX 的预设值，这个时候你可以自己改变这些参数。图 6.2 中显示了所有能改变的页面参数。这个图是用 `layout` 宏包产生的³。

先等等！……在你开始幻想“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合理原因的。

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 页面也正是如此。经验显示，如果在一行中塞入更多的

³`macros/latex/required/tools`

⁴我说的是卓有声誉的出版商正式出版的书籍。



1	one inch + \hoffset	2	one inch + \voffset
3	\oddsidemargin = 28pt or \evensidemargin	4	\topmargin = 23pt
5	\headheight = 12pt	6	\headsep = 18pt
7	\textheight = 598pt	8	\textwidth = 345pt
9	\marginparsep = 7pt	10	\marginparwidth = 115pt \marginparpush = 5pt (not shown)
11	\footskip = 25pt \hoffset = 0pt \paperwidth = 597pt		\voffset = 0pt \paperheight = 845pt

图 6.2 – 页面布局参数。

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版芯的宽度，头脑中要明白你正在使给你的读者制造困难。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。

第一个命令给某些参数一个固定的值：

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

第二个命令给某些参数增加一个长度：

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

第二个命令实际上比`\setlength`命令更为实用，因为你可以相对于现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文档导言：

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

这时候，你可能会想要看看`calc`包，它允许你在`\setlength`的参量中进行算术运算。它也可以运用到任何用数值作为函数参量的地方。

6.5 更有趣的长度

只要可能，就应该避免在L^AT_EX文档中使用绝对长度。我更愿意通过页面中其他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{variable}{text}
\settodepth{variable}{text}
\settowidth{variable}{text}
```

下面的例子显示了如何应用这些命令：

```

\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 adjoin 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}

```

$a^2 + b^2 = c^2$
<p>Where: a, b – are adjoin to the right angle of a right-angled triangle.</p> <p style="padding-left: 40px;">c – is the hypotenuse of the triangle and feels lonely.</p> <p style="padding-left: 40px;">d – finally does not show up here at all. Isn't that puzzling?</p>

6.6 盒子

\LaTeX 使用盒子来建立页面。首先，每个字符都是一个小的盒子，这些盒子粘起来构成单词，单词粘起来构成一行。值得注意的是，单词之间粘的是一种特殊的“胶水” (glue)，它是有弹性的，可以使 \LaTeX 压缩或者延伸使得单词将恰好构成页面的一行。

我承认，这里的描述是实际情况一个极度简化了的版本，但关键在于 \TeX 对盒子和胶水进行操作。不是只有字母才能成为盒子，你几乎可以把任何东西包括其他盒子放到一个盒子中。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，尽管我并没有明确的说出来，你已经遇到了一些盒子。例如`tabular` 环境和`\includegraphics` 命令就都产生了一个盒子。这就意味着你可以轻松的将两个表格或图像并列。你唯一需要保证的就是它们宽度的总和不大文本宽度。

使用如下命令可以把一个段落放置到盒子中：

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

或者用下面这个环境完成同样的事情：

```
\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周围文本基线的垂直方向对齐。`width` 是一个长度参量用于调整盒子的宽度。`minipage` 和`\parbox` 的区别在于你可能无法在一个`\parbox` 中使用所有的命令或者环境，而几乎任何东西都可以在`minipage` 中使用。

虽然`\parbox` 可以打包整个段落，完成分行在内的几乎所有事情， \LaTeX 中还存在与此不同的另外一类盒子命令用于处理水平对齐的东西。我们已经知道其中的一个 —— `\mbox`，它只是简单地将其它盒子包含在一个盒子里，从而防

止L^AT_EX 断开两个单词。因为盒子中可以包含盒子，它可以给予作者几乎无限的灵活性。

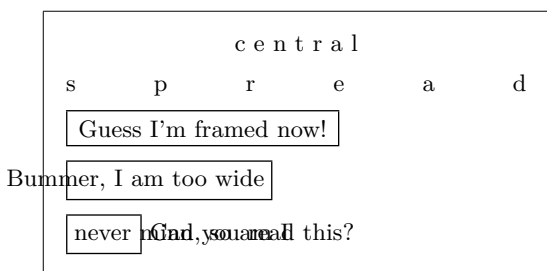
```
\makebox[width][pos]{text}
```

`width` 定义了生成的盒子的外部宽度⁵。除了长度表达式，你也可以传递`\width`、`\height`、`\depth` 和`\totalheight` 给`width`。这几个值是测量盒子内部文本来获得的。参数`pos` 接受一个字符值：`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` 能完成的一些工作：

```
\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，轻而易举。命令

```
\raisebox{lift}[extend-above-baseline][extend-below-baseline]{text}
```

让你能够定义一个盒子在垂直方向的属性。在前面的三个参数中，你可以使用`\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其外部的布局。

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……

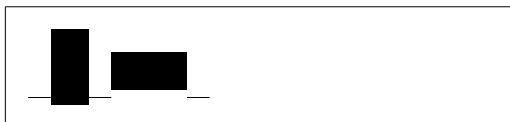
6.7 标尺和支撑

很多页之前你可能注意到这样的命令：

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

通常它被用来输出一个简单的黑色盒子。

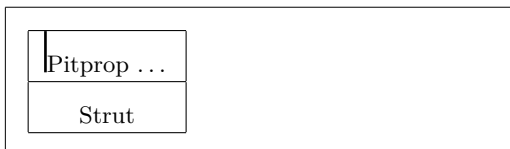
```
\rule{3mm}{.1pt}%
\rule[-1mm]{5mm}{1cm}%
\rule{3mm}{.1pt}%
\rule[1mm]{1cm}{5mm}%
\rule{3mm}{.1pt}
```



这个命令可以用来产生水平方向和垂直方向的线条。例如标题页上的直线就是用一个 `\rule` 命令创建的。

一种特殊的例子是没有宽度只有高度的标尺。在专业的出版中，这被称为支撑 (Struts)。它被用来保证页面的某个元素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。

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



全篇结束。

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] Frank Mittelbach, Michel Goossens, Johannes Braams, David Carlisle, Chris Rowley. *The L^AT_EX Companion, (2nd Edition)*. Addison-Wesley, Reading, Massachusetts, 2004, ISBN 0-201-36299-6.
- [4] Michel Goossens, Sebastian Rahtz and Frank Mittelbach. *The L^AT_EX Graphics Companion*. Addison-Wesley, Reading, Massachusetts, 1997, ISBN 0-201-85469-4.
- [5] Each L^AT_EX installation should provide a so-called *L^AT_EX Local Guide*, which explains the things that 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.
- [6] L^AT_EX3 Project Team. *L^AT_EX 2_ε for authors*. Comes with the L^AT_EX 2_ε distribution as `usrguide.tex`.
- [7] 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`.
- [8] L^AT_EX3 Project Team. *L^AT_EX 2_ε Font selection*. Comes with the L^AT_EX 2_ε distribution as `fntguide.tex`.
- [9] 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.
- [10] 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.
- [11] Vladimir Volovich, Werner Lemberg and L^AT_EX3 Project Team. *Cyrillic languages support in L^AT_EX*. Comes with the L^AT_EX 2_ε distribution as `cyrguide.tex`.

-
- [12] 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](http://tex-archive/help/Catalogue/catalogue.html)
- [13] 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](http://tex-archive/info/epslatex.ps)
- [14] Kristoffer H. Rose. *X_y-pic User's Guide*. Downloadable from CTAN with X_y-pic distribution
- [15] John D. Hobby. *A User's Manual for MetaPost*. Downloadable from <http://cm.bell-labs.com/who/hobby/>
- [16] Alan Hoenig. *T_EX Unbound*. Oxford University Press, 1998, ISBN 0-19-509685-1; 0-19-509686-X (pbk.)
- [17] Urs Oswald. *Graphics in L^AT_EX 2_ε*, containing some Java source files for generating arbitrary circles and ellipses within the `picture` environment, and *MetaPost - A Tutorial*. Both downloadable from <http://www.ursoswald.ch>

索引

- `\!`, 42
- `"`, 16
- `"'`, 26
- `"-`, 26
- `"---`, 26
- `"<`, 26
- `"=`, 26
- `">`, 26
- `"'`, 26
- `$`, 37
- `\(`, 37
- `\)`, 37
- `\,`, 38, 42
- `-`, 17
- `-`, 17
- `\-`, 16
- `-`, 17
- `—`, 17
- `.,` space after, 26
- `...`, 18
- `\:`, 42
- `\;`, 42
- `\@`, 26
- `\[`, 37
- `\`, 15, 30–32, 92
- `*`, 15
- `\]`, 37
- `,`, 26
- A4 paper, 9
- A5 paper, 9
- å, 19
- abstract, 31
- accent, 19
- Acrobat Reader, 61
- acute, 19
- `\addtolength`, 94
- advantages of L^AT_EX, 2
- æ, 19
- aeguill, 61
- `\Alph`, 26
- `\alph`, 26
- amsbsy, 47
- amsmath, 39, 54
- amsmath, 38, 41–45, 47
- amssymb, 39, 48
- amsthm, 46, 47
- `\and`, 28
- ansinew, 20
- `\appendix`, 27, 28
- applemac, 20
- `\ar`, 79
- `\arccos`, 40
- `\arcsin`, 40
- `\arctan`, 40
- `\arg`, 40
- array, 43
- arrow symbols, 40
- article class, 8
- `\Asbuk`, 26
- `\asbuk`, 26
- `\author`, 28, 64
- B5 paper, 9
- babel, 15, 19, 20, 25, 26
- `\backmatter`, 28
- backslash, 4
- `\backslash`, 4
- base font size, 9
- beamer, 66–68
- `\begin`, 30, 70, 78
- `\bibitem`, 56
- bibliography, 56
- `\Big`, 42
- `\big`, 42
- `\Bigg`, 42
- `\bigg`, 42
- `\biggl`, 46
- `\biggr`, 46
- `\bigskip`, 92

- `\binom`, 41
- blackboard bold, 39
- block, 68
- bm, 47
- `\bmod`, 40
- bold face, 87
- bold symbols, 39, 47
- `\boldmath`, 47
- `\boldsymbol`, 47
- book class, 8
- braces, 42
- brazilian, 21

- calc, 94
- `\caption`, 35, 36
- `\cdot`, 40
- `\cdots`, 42
- center, 30
- `\chapter`, 27
- `\chaptermark`, 58
- `\ci`, 83
- `\circle`, 73
- `\circle*`, 73
- `\cite`, 56
- CJK package, 24
- `\cleardoublepage`, 36
- `\clearpage`, 36
- `\cline`, 32
- color, 66
- coloured text, 8
- comma, 18
- commands, 4
 - `\!`, 42
 - `\(`, 37
 - `\)`, 37
 - `\,`, 38, 42
 - `\-`, 16
 - `\:`, 42
 - `\;`, 42
 - `\@`, 26
 - `\[`, 37
 - `\`, 15, 30–32, 92
 - `*`, 15
 - `\]`, 37
 - `\addtolength`, 94
 - `\Alph`, 26
 - `\alph`, 26
 - `\and`, 28
 - `\appendix`, 27, 28
 - `\ar`, 79
 - `\arccos`, 40
 - `\arcsin`, 40
 - `\arctan`, 40
 - `\arg`, 40
 - `\Asbuk`, 26
 - `\asbuk`, 26
 - `\author`, 28, 64
 - `\backmatter`, 28
 - `\backslash`, 4
 - `\begin`, 30, 70, 78
 - `\bibitem`, 56
 - `\Big`, 42
 - `\big`, 42
 - `\Bigg`, 42
 - `\bigg`, 42
 - `\biggl`, 46
 - `\biggr`, 46
 - `\bigskip`, 92
 - `\binom`, 41
 - `\bmod`, 40
 - `\boldmath`, 47
 - `\boldsymbol`, 47
 - `\caption`, 35, 36
 - `\cdot`, 40
 - `\cdots`, 42
 - `\chapter`, 27
 - `\chaptermark`, 58
 - `\ci`, 83
 - `\circle`, 73
 - `\circle*`, 73
 - `\cite`, 56
 - `\cleardoublepage`, 36
 - `\clearpage`, 36
 - `\cline`, 32
 - `\cos`, 40
 - `\cosh`, 40
 - `\cot`, 40
 - `\coth`, 40
 - `\csc`, 40
 - `\date`, 28
 - `\ddots`, 42
 - `\deg`, 40
 - `\depth`, 96
 - `\det`, 40
 - `\dim`, 40
 - `\displaystyle`, 45
 - `\documentclass`, 8, 11, 15
 - `\dq`, 23
 - `\dum`, 83
 - `\emph`, 29, 87

`\end`, 30, 70
`\enumBul`, 26
`\enumEng`, 26
`\enumLat`, 26
`\eqref`, 38
`\EURtm`, 18
`\exp`, 40
`\fbox`, 16
`\flq`, 23
`\flqq`, 23
`\foldera`, 76
`\folderb`, 76
`\footnote`, 29, 36
`\footskip`, 93
`\frac`, 41
`\framebox`, 96
`\frenchspacing`, 25, 26
`\frontmatter`, 28
`\frq`, 23
`\frqq`, 23
`\fussy`, 15
`\gcd`, 40
`\headheight`, 93
`\headsep`, 93
`\height`, 96
`\hline`, 32
`\hom`, 40
`\href`, 64, 66
`\hspace`, 85, 91
`\hyphenation`, 15
`\idotsint`, 43
`\ifpdf`, 66
`\ignorespaces`, 85
`\ignorespacesafterend`, 85
`\iiiint`, 43
`\iiint`, 43
`\iint`, 43
`\include`, 12
`\includegraphics`, 56, 62, 66, 95
`\includeonly`, 12
`\indent`, 91
`\index`, 57, 58
`\inf`, 40
`\input`, 12
`\int`, 41
`\item`, 30
`\ker`, 40
`\label`, 28, 38
`\LaTeX`, 16
`\LaTeXe`, 16
`\ldots`, 18, 42
`\left`, 42
`\leftmark`, 58
`\lg`, 40
`\lim`, 40
`\liminf`, 40
`\limsup`, 40
`\line`, 71, 76
`\linebreak`, 15
`\linespread`, 90
`\linethickness`, 74, 75, 77
`\listoffigures`, 35
`\listoftables`, 35
`\ln`, 40
`\log`, 40
`\mainmatter`, 28, 65
`\makebox`, 96
`\makeindex`, 57
`\maketitle`, 28
`\marginparpush`, 93
`\marginparsep`, 93
`\marginparwidth`, 93
`\mathbb`, 39
`\mathrm`, 45
`\max`, 40
`\mbox`, 16, 18, 95
`\min`, 40
`\multicolumn`, 33
`\multipt`, 70, 74
`\newcommand`, 84
`\newenvironment`, 84, 85
`\newline`, 15
`\newpage`, 15
`\newsavebox`, 76
`\newtheorem`, 46
`\newtheoremstyle`, 46
`\noindent`, 91
`\nolinebreak`, 15
`\nonumber`, 44
`\nopagebreak`, 15
`\not`, 49
`\oddsidemargin`, 93
`\oval`, 75, 76
`\overbrace`, 40
`\overleftarrow`, 40
`\overline`, 39
`\overrightarrow`, 40
`\pagebreak`, 15
`\pageref`, 28, 60
`\pagestyle`, 10

- `\paperheight`, 93
- `\paperwidth`, 93
- `\par`, 87
- `\paragraph`, 27
- `\parbox`, 95
- `\parindent`, 90
- `\parskip`, 90
- `\part`, 27
- `\phantom`, 36, 45
- `\pmod`, 40
- `\Pr`, 40
- `\printindex`, 58
- `\prod`, 41
- `\protect`, 36
- `\providecommand`, 84
- `\ProvidesPackage`, 86
- `\put`, 70–76
- `\qbezier`, 69, 70, 77
- `\qedhere`, 47
- `\qquad`, 38, 42
- `\quad`, 38, 42
- `\raisebox`, 96
- `\ref`, 28, 38, 60
- `\renewcommand`, 84
- `\renewenvironment`, 85
- `\right`, 42, 43
- `\right.`, 42
- `\rightmark`, 58
- `\rule`, 85, 97
- `\savebox`, 76
- `\scriptscriptstyle`, 45
- `\scriptstyle`, 45
- `\sec`, 40
- `\section`, 27, 36
- `\sectionmark`, 58
- `\selectlanguage`, 20
- `\setlength`, 70, 90, 94
- `\settodepth`, 94
- `\settoheight`, 94
- `\settowidth`, 94
- `\sin`, 40
- `\sinh`, 40
- `\sloppy`, 15
- `\smallskip`, 92
- `\sqrt`, 39
- `\stackrel`, 41
- `\stretch`, 85, 91
- `\subparagraph`, 27
- `\subsection`, 27
- `\subsectionmark`, 58
- `\substack`, 41
- `\subsubsection`, 27
- `\sum`, 41
- `\sup`, 40
- `\tableofcontents`, 27
- `\tan`, 40
- `\tanh`, 40
- `\TeX`, 16
- `\texorpdfstring`, 65
- `\textcelsius`, 17
- `\texteuro`, 17
- `\textheight`, 93
- `\textrm`, 45
- `\textstyle`, 45
- `\textwidth`, 93
- `\thicklines`, 72, 75, 77
- `\thinlines`, 75, 77
- `\thispagestyle`, 11
- `\title`, 28
- `\tnss`, 84
- `\today`, 16
- `\topmargin`, 93
- `\totalheight`, 96
- `\underbrace`, 40
- `\underline`, 29, 39
- `\unitlength`, 70, 71
- `\usebox`, 76
- `\usepackage`, 8, 11, 17–20, 86
- `\vdots`, 42
- `\vec`, 40
- `\vector`, 72
- `\verb`, 32
- `\verbatiminput`, 59
- `\vspace`, 92
- `\widehat`, 40
- `\widetilde`, 40
- `\width`, 96
- `\xymatrix`, 79
- comment, 5
- comments, 5
- `\cos`, 40
- `\cosh`, 40
- `\cot`, 40
- `\coth`, 40
- cp1251, 20
- cp850, 20
- cp866nav, 20
- cross-references, 28
- `\csc`, 40
- curly braces, 4, 87

- dash, 17
- \date, 28
- DC font, 16
- dcolumn, 33
- \ddots, 42
- \deg, 40
- degree symbol, 17
- delimiters, 42
- \depth, 96
- description, 30
- \det, 40
- Deutsch, 22
- \dim, 40
- dimensions, 91
- displaymath, 37
- \displaystyle, 45
- doc, 10
- document font size, 9
- document title, 9
- \documentclass, 8, 11, 15
- dotless i and j, 19
- double line spacing, 90
- double sided, 9
- \dq, 23
- \dum, 83
- epic, 69, 73
- ellipsis, 18
- em-dash, 17
- \emph, 29, 87
- empty, 10
- en-dash, 17
- Encapsulated POSTSCRIPT, 55, 62
- encodings
 - font
 - LGR, 21
 - OT1, 21
 - T1, 21, 25
 - T2*, 25
 - T2A, 21, 25
 - T2B, 21
 - T2C, 21
 - X2, 21
 - input
 - ansinew, 20
 - applemac, 20
 - cp1251, 20
 - cp850, 20
 - cp866nav, 20
 - koi8-ru, 20, 25
 - latin1, 20
 - macukr, 20
 - utf8x, 20
- \end, 30, 70
- \enumBul, 26
- \enumEng, 26
- enumerate, 30
- \enumLat, 26
- environment, 30
- environments
 - abstract, 31
 - array, 43
 - block, 68
 - center, 30
 - comment, 5
 - description, 30
 - displaymath, 37
 - enumerate, 30
 - eqnarray, 44
 - equation, 38
 - figure, 34, 35
 - flushleft, 30
 - flushright, 30
 - frame, 68
 - itemize, 30
 - lscommand, 83
 - math, 37
 - minipage, 95
 - parbox, 95
 - picture, 69, 70, 73, 74
 - proof, 47
 - pspicture, 69
 - quotation, 31
 - quote, 31
 - subarray, 41
 - table, 34, 35
 - tabular, 32, 95
 - thebibliography, 56
 - verbatim, 32, 59
 - verse, 31
- epic, 69
- eqnarray, 44
- \eqref, 38
- equation, 38
- equation system, 44
- eurosym, 18
- \EURtm, 18
- executive paper, 9
- \exp, 40
- exponent, 39

- exscale, 10, 42
- extension, 11
 - .aux, 11
 - .cls, 11
 - .dtx, 11
 - .dvi, 11
 - .fd, 11
 - .idx, 11
 - .ilg, 11
 - .ind, 11
 - .ins, 11
 - .lof, 11
 - .log, 11
 - .lot, 11
 - .sty, 11
 - .tex, 7, 11
 - .toc, 11
- fancyhdr, 58, 59
- \fbox, 16
- figure, 34, 35
- floating bodies, 34
- \flq, 23
- \flqq, 23
- flushleft, 30
- flushright, 30
- foiltex, 8
- \foldera, 76
- \folderb, 76
- font, 86
 - \footnotesize, 87
 - \Huge, 87
 - \huge, 87
 - \LARGE, 87
 - \Large, 87
 - \large, 87
 - \mathbf, 88
 - \mathcal, 88
 - \mathit, 88
 - \mathnormal, 88
 - \mathrm, 88
 - \mathsf, 88
 - \mathtt, 88
 - \normalsize, 87
 - \scriptsize, 87
 - \small, 87
 - \textbf, 87
 - \textit, 87
 - \textmd, 87
 - \textnormal, 87
 - \textrm, 87
 - \textsc, 87
 - \textsf, 87
 - \textsl, 87
 - \texttt, 87
 - \textup, 87
 - \tiny, 87
- font encoding, 10
- font encodings, 20
 - LGR, 21
 - OT1, 21
 - T1, 21, 25
 - T2*, 25
 - T2A, 21, 25
 - T2B, 21
 - T2C, 21
 - X2, 21
- font size, 86, 87
- fontenc, 10, 20, 25
- footer, 10
- \footnote, 29, 36
- \footnotesize, 87
- \footskip, 93
- formulae, 37
- \frac, 41
- fraction, 41
- fragile commands, 36
- frame, 68
- \framebox, 96
- French, 22
- \frenchspacing, 25, 26
- \frontmatter, 28
- \frq, 23
- \frqq, 23
- \fussy, 15
- \gcd, 40
- geometry, 59
- German, 20, 22
- GhostScript, 55
- graphics, 8, 55
- graphicx, 55, 62, 63, 66
- grave, 19
- Greek, 24
- Greek letters, 39
- grouping, 87
- HIAT_EX, 24
- hIAT_EXp, 24
- header, 10

- `\headheight`, 93
- textttheadings, 10
- `\headsep`, 93
- `\height`, 96
- `\hline`, 32
- `\hom`, 40
- horizontal
 - brace, 40
 - dots, 42
 - line, 39
 - space, 91
- `\href`, 64, 66
- `\hspace`, 85, 91
- `\Huge`, 87
- `\huge`, 87
- hyperref, 61, 63, 65, 66
- hypertext, 60
- hyphen, 17
- hyphenat, 59
- `\hyphenation`, 15

- `\idotsint`, 43
- ifpdf, 66
- `\ifpdf`, 66
- ifthen, 10
- `\ignorespaces`, 85
- `\ignorespacesafterend`, 85
- `\iiiint`, 43
- `\iiint`, 43
- `\iint`, 43
- `\include`, 12
- `\includegraphics`, 56, 62, 66, 95
- `\includeonly`, 12
- `\indent`, 91
- indentfirst, 90
- index, 57
- `\index`, 57, 58
- `\inf`, 40
- `\input`, 12
- input encodings
 - ansinew, 20
 - applemac, 20
 - cp1251, 20
 - cp850, 20
 - cp866nav, 20
 - koi8-ru, 20, 25
 - latin1, 20
 - macukr, 20
 - utf8x, 20
- input file, 5
- inputenc, 10, 20, 25
- `\int`, 41
- integral operator, 41
- international, 19
- italic, 87
- `\item`, 30
- itemize, 30

- `\ker`, 40
- Knuth, Donald E., 1
- koi8-ru, 20, 25
- Korean, 23
- Korean font
 - UHC font, 24
- Korean input files, 23

- `\label`, 28, 38
- Lamport, Leslie, 1
- language, 19
- `\LARGE`, 87
- `\Large`, 87
- `\large`, 87
- `\LaTeX`, 16
- LaTeX3, 3
- `\LaTeXe`, 16
- latexsym, 10
- latin1, 20
- layout, 92
- `\ldots`, 18, 42
- `\left`, 42
- left aligned, 30
- `\leftmark`, 58
- legal paper, 9
- letter paper, 9
- `\lg`, 40
- LGR, 21
- ligature, 18
- `\lim`, 40
- `\liminf`, 40
- `\limsup`, 40
- `\line`, 71, 76
- line break, 14
- line spacing, 90
- `\linebreak`, 15
- `\linespread`, 90
- `\linethickness`, 74, 75, 77
- `\listoffigures`, 35
- `\listoftables`, 35
- `\ln`, 40
- `\log`, 40

- long equations, 44
- longtabular, 34
- lscmmand, 83
- macukr, 20
- \mainmatter, 28, 65
- \makebox, 96
- makeidx, 10, 57
- makeidx package, 57
- \makeindex, 57
- makeindex program, 57
- \maketitle, 28
- \marginparpush, 93
- \marginparsep, 93
- \marginparwidth, 93
- marvosym, 18
- math, 37
- math font size, 45
- math spacing, 42
- \mathbb, 39
- \mathbf, 88
- \mathcal, 88
- mathematical
 - accents, 40
 - delimiter, 42
 - functions, 40
 - minus, 17
- mathematics, 37
- \mathit, 88
- \mathnormal, 88
- \mathrm, 45, 88
- mathrsfs, 54
- \mathsf, 88
- mathtext, 25
- \mathtt, 88
- \max, 40
- \mbox, 16, 18, 95
- METAPOST, 63
- \min, 40
- minimal class, 8
- minipage, 95
- minus sign, 17
- Mittelbach, Frank, 1
- mltex, 62
- mltex, 62
- modulo function, 40
- \multicolumn, 33
- \multirow, 70, 74
- \newcommand, 84
- \newenvironment, 84, 85
- \newline, 15
- \newpage, 15
- \newsavebox, 76
- \newtheorem, 46
- \newtheoremstyle, 46
- \noindent, 91
- \nolinebreak, 15
- \nonumber, 44
- \nopagebreak, 15
- \normalsize, 87
- \not, 49
- \oddsidemargin, 93
- œ, 19
- one column, 9
- option, 8
- optional parameters, 4
- OT1, 21
- \oval, 75, 76
- \overbrace, 40
- overfull hbox, 15
- \overleftarrow, 40
- \overline, 39
- \overrightarrow, 40
- package, 5, 8, 83
- packages
 - aeguill, 61
 - amsbsy, 47
 - amsfonts, 39, 54
 - amsmath, 38, 41–45, 47
 - amssymb, 39, 48
 - amsthm, 46, 47
 - babel, 15, 19, 20, 25, 26
 - beamer, 66–68
 - bm, 47
 - calc, 94
 - color, 66
 - dcolumn, 33
 - doc, 10
 - eepic, 69, 73
 - epic, 69
 - eurosym, 18
 - exscale, 10, 42
 - fancyhdr, 58, 59
 - fontenc, 10, 20, 25
 - geometry, 59
 - graphicx, 55, 62, 63, 66
 - hyperref, 61, 63, 65, 66

- hyphenat, 59
- ifpdf, 66
- ifthen, 10
- indentfirst, 90
- inputenc, 10, 20, 25
- latexsym, 10
- layout, 92
- longtabular, 34
- makeidx, 10, 57
- marvosym, 18
- mathrsfs, 54
- mathtext, 25
- mltex, 62
- ppower4, 68
- prosper, 68
- pstricks, 69, 73
- pxfonts, 62
- showidx, 58
- supertabular, 34
- syntonly, 10, 12
- textcomp, 17
- txfonts, 62
- ucs, 20
- verbatim, 5, 59
- xy, 79
- page layout, 92
- page style, 10
 - empty, 10
 - headings, 10
 - plain, 10
- \pagebreak, 15
- \pageref, 28, 60
- \pagestyle, 10
- paper size, 9, 92
- \paperheight, 93
- \paperwidth, 93
- \par, 87
- paragraph, 13
- \paragraph, 27
- parameter, 4
- \parbox, 95
- parbox, 95
- \parindent, 90
- \parskip, 90
- \part, 27
- PDF, 60
- PDF \LaTeX , 68
- pdf \LaTeX , 61, 66
- pdf \LaTeX , 61
- pdf \TeX , 61
- period, 18
- \phantom, 36, 45
- picture, 69, 70, 73, 74
- placement specifier, 35
- plain, 10
- \pmod, 40
- Português, 21
- Portuguese, 21
- POSTSCRIPT, 3, 7, 24, 55, 61, 62, 69
 - Encapsulated, 55, 62
- ppower4, 68
- \Pr, 40
- preamble, 5
- prime, 40
- \printindex, 58
- proc class, 8
- \prod, 41
- product operator, 41
- proof, 47
- prosper, 68
- \protect, 36
- \providecommand, 84
- \ProvidesPackage, 86
- pspicture, 69
- pstricks, 69, 73
- \put, 70–76
- pxfonts, 62
- \qBezier, 69, 70, 77
- \qedhere, 47
- \quad, 38, 42
- \quad, 38, 42
- quotation, 31
- quotation marks, 16
- quote, 31
- \raisebox, 96
- \ref, 28, 38, 60
- \renewcommand, 84
- \renewenvironment, 85
- report class, 8
- reserved characters, 4
- \right, 42, 43
- right-aligned, 30
- \right., 42
- \rightmark, 58
- roman, 87
- \rule, 85, 97
- sans serif, 87

- `\savebox`, 76
- Scandinavian letters, 19
- `\scriptscriptstyle`, 45
- `\scriptsize`, 87
- `\scriptstyle`, 45
- `\sec`, 40
- `\section`, 27, 36
- `\sectionmark`, 58
- `\selectlanguage`, 20
- `\setlength`, 70, 90, 94
- `\settodepth`, 94
- `\settoheight`, 94
- `\settowidth`, 94
- showidx, 58
- `\sin`, 40
- single sided, 9
- `\sinh`, 40
- slanted, 87
- slides class, 8
- `\sloppy`, 15
- `\small`, 87
- Small Caps, 87
- `\smallskip`, 92
- space, 3
- special character, 19
- `\sqrt`, 39
- square brackets, 4
- square root, 39
- `\stackrel`, 41
- `\stretch`, 85, 91
- structure, 5
- Struts, 97
- subarray, 41
- `\subparagraph`, 27
- subscript, 39
- `\subsection`, 27
- `\subsectionmark`, 58
- `\substack`, 41
- `\subsubsection`, 27
- `\sum`, 41
- sum operator, 41
- `\sup`, 40
- superscript, 41
- supertabular, 34
- syntonly, 10, 12

- T1, 21, 25
- T2*, 25
- T2A, 21, 25
- T2B, 21

- T2C, 21
- table, 32
- table, 34, 35
- table of contents, 27
- `\tableofcontents`, 27
- tabular, 32, 95
- `\tan`, 40
- `\tanh`, 40
- `\TeX`, 16
- `\texorpdfstring`, 65
- `\textbf`, 87
- `\textcelsius`, 17
- textcomp, 17
- `\texteuro`, 17
- `\textheight`, 93
- `\textit`, 87
- `\textmd`, 87
- `\textnormal`, 87
- `\textrm`, 45, 87
- `\textsc`, 87
- `\textsf`, 87
- `\textsl`, 87
- `\textstyle`, 45
- `\texttt`, 87
- `\textup`, 87
- `\textwidth`, 93
- thebibliography, 56
- `\thicklines`, 72, 75, 77
- `\thinlines`, 75, 77
- `\thispagestyle`, 11
- three dots, 42
- tilde, 17, 40
- tilde (), 26
- `\tiny`, 87
- title, 9, 28
- `\title`, 28
- `\tnss`, 84
- `\today`, 16
- `\topmargin`, 93
- `\totalheight`, 96
- two column, 9
- txfonts, 62

- ucs, 20
- umlaut, 19
- `\underbrace`, 40
- underfull hbox, 15
- `\underline`, 29, 39
- `\unitlength`, 70, 71
- units, 91, 92

upright, 87
URL, 17
\usebox, 76
\usepackage, 8, 11, 17–20, 86
utf8x, 20

\vdots, 42
\vec, 40
\vector, 72
vectors, 40
\verb, 32
verbatim, 5, 59
verbatim, 32, 59
\verbatiminput, 59
verse, 31
vertical
 dots, 42
vertical space, 92
\vspace, 92

whitespace
 after commands, 4
 at the start of a line, 3

\widehat, 40
\widetilde, 40
\width, 96
Word, 58
www, 17
WYSIWYG, 2

X2, 21
Xpdf, 61
xy, 79
\xymatrix, 79

