

LaTeX 入门

yunxinyi@gmail.com

2012-10-26

目录

1 LaTeX 入门	5
1.1 LaTeX 是什么?	7
1.1.1 我们如何受益	7
1.1.2 开源的好处	7
1.1.3 形式与内容的分离	7
1.1.4 可移植性	7
1.1.5 成果的保护	8
1.1.6 对比 LaTeX 并及其相比传统字处理软件的优缺点	8
1.1.7 挑战是什么?	8
1.2 安装 LaTeX	9
1.2.1 使用网络安装向导安装 TeX	9
1.2.2 离线安装 TeX Live	9
1.2.3 在其他系统上安装	9
1.3 写第一份 LaTeX 文档	10
1.3.1 Time for action-使用 TeXworks 写我们的第一份文档	10
1.4 总结	11
2 字, 句, 段的格式化	13
2.1 节省事件和精力自定义命令	15
2.1.1 Time for action-creating our first command using it as an abbrevitaion	15
2.1.2 Gentle spacing after commands	15
2.1.3 Time for action-adding intellingent spacing to command ouput	15
2.1.4 creating more universal commands-using arguments	15
2.1.5 Time for action-creating a macro for formating keywords	15
2.1.6 Using optional arguments	15
2.1.7 Time for action-marking keywords with optional formating	15
2.2 使用 box 来限制文字的宽度	16
2.3 Breaking lines and paragraphs	17
2.3.1 提高断字 (Imporving hyphenation)	17
2.3.2 Improving the justification further	17

2.3.3	Breaking lines manually	17
2.3.4	Preventing line breaks	17
2.3.5	Managing line breaks wisely	17
2.3.6	Exploring the fine details	18
2.3.7	Understanding ligatures	18
2.3.8	Choosing the rightdash	18
2.3.9	Setting dots	18
2.3.10	Time for action-using differently spaced dots	18
2.3.11	Time for action-comparing dots to ellipsis	18
2.3.12	Setting accents	18
2.3.13	Time for action-experimenting with accents	18
2.3.14	Using special characters directly in the editor	18
2.3.15	Time for action-using accents directly	18
2.4	Truning off full justification	19
2.4.1	Time for action-justifying a paragraph to the left	19
2.4.2	Creating ragged-left text	19
2.4.3	Time for action-centering a title	19
2.4.4	Using environments for justification	19
2.4.5	Time for action-centering verses	19
2.5	显示引用	20
2.5.1	Time for action-quoting a scientist	20
2.5.2	引用更长的文本	20
2.5.3	Time for action-quoting TeX's benefits	20
3	页面设计	21
3.1	定义全局布局	23
3.1.1	Time for action-写个包含章节的书	23
4	创建列表	25
4.1	项目符号列表	25
4.2	数字列表	25
4.3	定义列表	25
5	创建表格和插入图片	27
5.1	以列的方式排列文本和数据	27
5.2	排版复杂的表格	27
5.3	在文档中插入图片	27
5.4	给表格和图片添加标题	27
5.5	控制图表和表格的布局	27

6	交叉引用	29
6.1	设置引用标签	29
6.2	区间,脚注,列表条目,表格和其他的引用	29
6.3	页码的引用及 <code>ranges</code>	29
6.4	使 LaTeX 对长页面的冗余引用	29
6.5	引用的自动命名	29
6.6	创建到外部文档的引用	29
7	内容和引用的列表	31
7.1	决定要插入 TOC 的标题的级别	31
7.2	创建并个性化图列表 (LOF) 和表格列表 (LOT)	31
7.3	将随意的文本和命令插入到 TOC 和其他列表	31
7.4	引用和参考书目	31
7.5	创建目录	31
7.6	修改所有这些列表的标题	31
8	编辑数学公式	33
8.1	编辑基本公式	33
8.2	文本中插入公式和公式间插入文本	33
8.3	等式的中间对齐并编号	33
8.4	多列等式的对齐	33
8.5	插入数学符号例如:根号,操作符,希腊字母和箭头	33
8.6	分数	33
8.7	表达式的堆叠	33
8.8	矩阵	33
9	字体使用	35
9.1	为整个文档选择其他字体集	35
9.2	为特定的形状加载字体包	35
9.3	如何安装附加字体	35
9.4	多种字体的使用	35
9.5	输入文本的编码和输出的字体	35
10	创作大型文档	37
10.1	一个文档分为几个文件	37
10.2	代码的输入和重用	37
10.3	<code>swap out document</code> 设置	37
10.4	编译文档的一部分	37
10.5	创建一个基于其他 <code>sub-files</code> 的主文档	37
10.6	设计标题页	37

10.7在标题页使用罗马数字页码	37
10.8add a back matter without sectioning numbers	37
11 进一步美化文档	39
11.1为目录,图标列表和其他的交叉引用加上超链接	39
11.2创建书签的导航条	39
11.3设置标题	39
11.4使用 LaTeX 的颜色	39
12 故障排除	41
12.1理解错去信息并修复	41
12.2处理警告	41
12.3使用 LaTeX 在编译期间产生的所有信息	41
13 使用在线资源	43
.1 列表	45
A 后记	47
B 参考书目	49

Chapter 1

LaTeX 入门

你准备好离开“所见即所得”字处理器进入真实,可靠,高品质的排版世界了吗?好了,让我们一起出发吧!

这是伟大的,如果你决定学 **LaTeX** 的。这本书将引导你前进的道路来帮助你得到最有效地利用它。让我们简单说说关于 **LaTeX** 的利益和面临的挑战,然后我们应准备的工具。在本章中,我们将:

- 知道 **LaTeX** 和谈论字处理器相比的优点和缺点
- **LaTeX** 的软件安装完整的软件包,其中包括一个编辑器
- 写我们的第一个 **LaTeX** 文档

好了,我们开始吧。

1.1 LaTeX 是什么?

LaTeX 是一种文件的排版软件。换句话说,它是一个文件的准备系统。LaTeX 是一个文字处理器,但作为一个文档标记语言。LaTeX 是一个自由的,开放源码软件。它最初是由莱斯利·兰波特,是基于由 Donald Knuth TeX 排版引擎。人们常常把它当作只是 TeX 的,这意味着 LaTeX 的。它有着悠久的历史,你可以读到它在 <http://www.tug.org/whatis.html>。现在,让我们继续寻找我们如何能够物尽其用。

1.1.1 我们如何受益

LaTeX 是特别适合科学和技术文件。其优越的排版数学公式是有口皆碑的。如果你是一个学生或科学家,然后 LaTeX 是最好的选择,即使你不需要它的科研能力,有其他用途 - 它产生的输出质量非常高,它是非常稳定的,并且处理复杂的文档容易不管他们是如何的大。进一步显着的优势,LaTeX 的交叉引用功能,其自动编号和生成的内容,数字和表格,索引,词汇表,和参考书目。这是多国语言特定于语言的功能,它能够使用 PostScript 和 PDF 功能。除了完美的科学家,LaTeX 是非常灵活的,有模板信函,简报,票据,哲学书,法律文本,乐谱,甚至对国际象棋游戏的符号。数以百计的 LaTeX 用户已经写了成千上万的模板,样式,每一个可能的用途有用的工具。这是网上收集和分类的存档服务器。你可以受益于开始与它的默认样式,依靠其令人印象深刻的高品质其智能的格式,但你可以自由定制和修改的一切。人 TeX 的社区已经写了很多的扩展寻址的几乎每一个格式化的需要。

1.1.2 开源的好处

LATEX 的来源是完全免费的,大家都看得到。这使您可以学习和改变一切,从核心的 LaTeX 的最新扩展包。但是,这是什么意思,你作为一个初学者?有一个巨大的 LaTeX 的社区有很多友好,乐于助人的人。即使你不能直接受益于开放的源代码,他们可以阅读源代码,并为您提供帮助。只要加入一个 LaTeX 网络论坛和问您的问题有。助手,如果必要的话,挖到 LaTeX 的来源和在所有的可能性中找到一个解决方案为你,有时往往提供了一个重新定义的一个建议一个合适的包,默认命令。

今天,我们已经从近 30 年的发展中受益的 TeX 的社会。开源的理念是不可能的,因为每个用户都邀请到研究和改进软件和进一步发展。第 13 章,利用网络资源,将指向给社会。

1.1.3 形式与内容的分离

的 LaTeX 的一个基本原则是,作者不应该分心太多格式问题。通常情况下,笔者着重逻辑的内容和格式,例如,而不是写一章的标题用大号加粗字体,你只是告诉 LaTeX 的,这是一个章节标题,你可以让 LaTeX 的设计,或者你决定在文件中的的标题看起来像什么,只是一次为整个文档的设置。LaTeX 的使用广泛称为类和包的样式文件,因此很容易设计并修改的外观整个文档和所有它的细节。

1.1.4 可移植性

LaTeX 是可用于几乎所有的操作系统,如 Windows, Linux 和 Mac OS X 中,和等等。它的文件格式是纯文本可读和可编辑的,在所有操作系统上。LaTeX 会在所有系统上都产生相同的输出。虽然有不同的 LaTeX 所谓的 TeX 软件包,我们将重点放在 TeX Live 中,因为这适用于 Windows, Linux 和 Mac OS X 的分布 LaTeX 本身没有图形用户界面,这就是为什么它是如此的原因之一便携。您可以选择任何文本编辑器。有很多的编辑,甚至是专业从事 LaTeX,每一个操作系统。有些编辑器可为多个系统。例如,TeXworks 的运行在 Windows, Linux 和 Mac OS X,这就是原因,我们将使用的原因之一它在我们的书。另外一个很重要的原因是,它可能是最好的,适合初学者。LaTeX 的生成 PDF 输出打印和可读性,在大多数计算机上,看上去是一样的无论操作系统。除了 PDF 格式,它支持 DVI, PostScript 和 HTML 输出,准备地分布在印刷版和网络,电子屏幕上,电子书阅读器或智能手机。综上所述,LaTeX 是便携式三个方面的来源,它的实施和输出。

1.1.5 成果的保护

LaTeX 文档都存储在人类可读的文本格式,而不是在一些不起眼的字处理格式,在不同版本的相同的软件,可能会改变。尝试打开一个 20 岁的书面文件与商业的文字处理器。什么您现代软件说明了什么?即使你可以读取该文件,其外观肯定会比以前不同。LaTeX 的承诺,该文件将永远是可读的,将导致在相同的输出。虽然它的进一步发展,它会保持向后兼容。文字处理,文件可能感染了病毒,恶意宏破坏数据。你有没有听到“隐藏”在一个文本文件中的病毒吗? LaTeX 是不受到病毒的威胁。

1.1.6 对比 LaTeX 并及其相比传统字处理软件的优缺点

我们已经描述的排版系统 LaTeX 的一些优势文字处理软件。LaTeX 的鼓励结构化的写作,其他字处理器可能会迫使你的工作不一致。他们可能隐藏了真正的格式构建和加密你的文件,在一些专有的文件格式。相容性是一个很大的的问题,甚至在相同的软件版本。有一些有趣的文章,可在网上 LaTeX 的其他软件进行比较。当然,他们表达意见。有些岁,因此不包括在最新的软件,但他们讨论重要点,今天仍然有效。你会发现它们列在第 13 章,利用网络资源。

1.1.7 挑战是什么?

可能是陡峭的学习曲线,但是这本书将帮助你掌握它。虽然写的 LaTeX 看起来像编程,不要害怕。你很快就会知道的常用的命令。自动完成和关键字高亮显示的文本编辑器我会支持你的。他们甚至可能会提供给你的命令的菜单和对话框。你现在认为这将需要很长的时间,直到你将学习以实现可信的结果吗? 不要担心,这本书会给你一个快速的开始。您将学习实践有很多实施例。还有更多的例子可以读取和从互联网上下载。在第 13 章中,我们将探讨互联网资源。我们将继续在我们的电脑与安装了 LaTeX,。

1.2 安装 LaTeX

让我们开始安装 LaTeX 的分布 TeX Live 中。这种分布是适用于 Windows, Linux 和 Mac OS X 上, 和其他 Unix-like 的操作系统。TeX Live 中。良好的维护, 它正在积极发展。另一个很好的和用户友好的 Windows 是 LaTeX 的分布 MiKTeX。像任何其他 Windows 应用程序可以很容易地安装, 但它不是可用于其他系统, 如 Linux 或 Mac OS X, 你可以下载从 <http://miktex.org>。

首先, 我们将参观 TEX Live 的主页, 参与调查的安装可能性。如果深入研究提供的信息有自由探索的网页。打开 TEX Live 的主页在 <http://tug.org/texlive>。

我们将介绍两种安装方式。第一, 将在网上和需要互联网连接。另一种方法是一个巨大的下载开始, 但可以完成离线。让我们来看看两种安装方法。

1.2.1 使用网络安装向导安装 TeX

1.2.2 离线安装 TeX Live

1.2.3 在其他系统上安装

1.3 写第一份 LaTeX 文档

我们已经安装了 TeX 和启动编辑器,现在让我们写在深水的一端跳我们的第一个 LaTeX 文档。

1.3.1 Time for action-使用 TeXworks 写我们的第一份文档

我们的首要目标是创建一个文档,打印出来只有一句话。我们要使用它了解一个 LaTeX 文档的基本结构。

1. 点击桌面上的图标,或打开它的启动 **TeXworks** 的编辑器“开始”菜单。
2. 点击新建按钮
3. 输入以下内容

```
1 \documentclass={article}  
2 \begin{document}  
3 This is our first document.  
4 \end{document}
```

4. 点击保存按钮保存。选择要存储的位置,理想是它自己的文件夹。
5. 在下拉列表中选择 **TeXworks** 工具条,选择 **pdfLaTeX**
6. 点击排版按钮
7. 输出窗口会自动打开。有一个看看吧:

刚刚发生了什么? 您刚才看到一个 LaTeX 文档的生命的最初几分钟。其下的时间天将由编辑,排版,等。不要忘记保存您的记录频繁。宣布,违背经典的字处理软件,你可以看不到效果立即改变,但结果却只是一个点击即可。

一展身手 - 检查先进的 LaTeX 的编辑器

你有经验的工作与复杂的程序呢? 你喜欢使用一个功能丰富的强大的编辑器呢? 然后看看这些胶乳编辑。访问他们的网站,发现截图和了解它们的功能:

- TeXnicCenter a very powerful editor for Windows, <http://texniccenter.org/>
- Kile a user-friendly editor for operating systems with KDE, such as Linux, <http://kile.sourceforge.net/>
- TeXShop an easy-to-use and very popular editor for Mac OS X, <http://pages.uoregon.edu/koch/texshop/>
- Texmaker a cross-platform editor running on Linux, Mac OS X, Unix, and Windows systems, <http://www.xmlmath.net/texmaker/>

上面提到的编辑器是免费的开源软件。

1.4 总结

TeX 的好处。这将是轮到我们使用的美德 LaTeX 来达到最佳的结果。此外,我们介绍了:

- 安装 TeX Live
- 使用 TeXworks 编辑器
- 创建一份 LaTeX 文档并产生输出

现在,我们已经有了一个功能并经过测试的 LaTeX 系统,我们就可以写我们自己的 LaTeX 文档。在下一章中,我们将详细的文本格式。

Chapter 2

字, 句, 段的格式化

在上一章中,我们安装了 **LaTeX** 和 **TeXworks** 的编辑器来编写我们的第一个文档。现在,我们将谈论的文档结构,我们将重点放在文本的细节和它的格式。

在这一章中,我们将:

- 探讨关于逻辑格式化
- 学习如何更改字体,文本的形状和风格
- 使用 **box** 来限制文档的宽度
- 学习如何断行和提高断字
- 探索对齐和格式化段落

过工作的例子,并尝试新的功能,我们将学习 **LaTeX** 一些基本概念。本章的结束,我们将熟悉命令和环境。你甚至可以定义自己的命令。

2.1 节省事件和精力自定义命令

2.1.1 Time for action-creating our first command using it as an abbreviation

2.1.2 Gentle spacing after commands

2.1.3 Time for action-adding intelligent spacing to command output

2.1.4 creating more universal commands-using arguments

2.1.5 Time for action-creating a macro for formatting keywords

2.1.6 Using optional arguments

2.1.7 Time for action-marking keywords with optional formatting

2.2 使用 **box** 来限制文字的宽度

2.3 Breaking lines and paragraphs

通常情况下,当你写的文字,你并不需要关心自动换行。只要输入文本与编辑 **LaTeX** 会使其适合线和照顾的理由。如果你想开始一个新的段落,结果得到一个换行符在输出时,只需插入一个空行,然后再继续你的文字。现在,我们将找出如何控制自动换行。首先,我们将看到如何提高自动断行。然后,我们将学习指令直接插入休息。

2.3.1 提高断字 (Improving hyphenation)

如果你看一下,你会发现在较长的文本,它是优秀的文字是完全有道理的 **LaTeX** 和如何单词之间的间距被均匀的分布上的线。如果有必要,**LaTeX** 会划分的话,并把连字符的行结束时,为了打破中的行更好的办法。**LaTeX** 的很好的算法已经使用连字符连接的词,但它可能发生它不能找到一个可以接受的方式,将一个字。前面的例子指出了这一个问题:打破了词的缩写,提高产量,但 **LaTeX** 不知道划分。我们将找出如何解决这个问题。

2.3.2 Improving the justification further

如今最流行的 TeX 编译器是 pdfTeX,它直接以 PDF 格式输出。当 xxx 写了 pdfTeX,他使用 microtypographic 的兼容性扩展了 TeX。当我们直接输出 pdf 时,我们实际上是用的 pdfLaTeX,并且我们可以使用 microtype 包来使用它的特性。

2.3.3 Breaking lines manually

我们可能会选择结束压倒一切的全自动行。有几个命令不同的效果。

2.3.4 Preventing line breaks

命令 `\linebreak` 有直接的对应:`\nolinebreak`。此命令在当前位置防止断行。像它的手,它带有一个可选的参数。如果你写 `\nolinebreak[0]`,则建议不断行。使用 `1, 2, 或 3` 使得请求更强大和 `\nolinebreak[4]` 完全禁止它。如果你不提供一个参数,则使用的是后者。已经提到的命令,`\mbox[文字]`,不仅要禁用断字,也避免了换行的完整文本。

LaTeX 可以在字与字之间的空间进行有意义的断行。符号 `\hspace{0.5em}` 标识词间没有空间断行:如果你写 `Dr. Watson`,那么 `Rr.` 绝不会出现在行末。

2.3.5 Managing line breaks wisely

坏断字的文件仍然可以消失的增长,说明一些明智的断字规则将不会做任何伤害,但可能被证明是有用的。但只使用 `\newline`, and `\linebreak` 来调整您的文档的最终版本! 当你还在编辑你的文字,你不需要担心换行。他们仍然在写作过程中可能会改变。难看理由可能有所变动,不干预变得更好。另一方面,如果你手动换行,但后来插入文本之前,其结果可能是不必要的短行。

所以,不要浪费你的能量格式化而你正在写的文本。

2.3.6 Exploring the fine details

印刷字体约定可能需要注意一些小细节,也有不同的破折号,且一个点周围的空间可能会有所不同,这取决于上下文。之后的空间一些字母可能取决于以下,以至于一些字母,甚至可能被接合到一个单一的。这种结构被称为连字。让我们来仔细看看在它们。

2.3.7 Understanding ligatures

2.3.8 Choosing the righthdash

2.3.9 Setting dots

2.3.10 Time for action-using differently spaced dots

2.3.11 Time for action-comparing dots to ellipsis

2.3.12 Setting accents

2.3.13 Time for action-experimenting with accents

2.3.14 Using special characters directly in the editor

2.3.15 Time for action-using accents directly

2.4 Truning off full justification

2.4.1 Time for action-justifying a paragraph to the left

2.4.2 Creating ragged-left text

2.4.3 Time for action-centering a title

2.4.4 Using environments for justification

由于具有对应每个声明都有相对应的环境, 我们可以在之前的文本中使用 `\begin{centering}...` `\end{centering}`。也可以做类似 `ragged-right` 和 `ragged-left` 文本。这里也有一些预定义的环境做类似的事情, 但是同时开始一个新的段落。

2.4.5 Time for action-centering verses

让我们重用 fragment of 诗"Annabel Lee"。这次我们将居中所有 verses:

```
1 \documentclass{article}
2 \usepackage{url}
3 \begin{document}
4 \noindent This is the beginning of a poem
5 by Edgar Allan Poe:
6 \begin{center}
7     \emph{Annabel Lee}
8 \end{center}
9 \begin{center}
10     It was many and many a year ago,\\
11     In a kingdom by the sea,\\
12     That a maiden there lived whom you may know\\
13     By the name of Annabel Lee
14 \end{center}
15 The complete poem can be read on
16 \url{http://www.online-literature.com/poe/576/}.
17 \end{document}
```

我们使用 `\` 开始来避免段落缩进。`\{center}` 开始了居中环境。它开始了一个新段落, 留下一些空白 to the preceding text。`\{center}` 结束了环境。我们两次使用环境。第二次, 我们插入了 `\\` 来结束 verses。

在 `center` 环境结束之后, 有一些空白接着, 下一个段落 began at the left margin。

对应于 `ragged-right text` 的环境叫做 `flushleft`, `ragged-left` 的叫做 `flushright`。

2.5 显示引用

想象一下你的作品中有其他作者的引用。如果只是简单的插入到文本中的话将会使得难以阅读。一种常用的提高可读性的方式是：设置这些文本两边缩进。

2.5.1 Time for action-quoting a scientist

我们将引用著名物理学家的思想。

```
1 \documentclass{article}
2 \begin{document}
3 Niels Bohr said: An expert
4 is a person who has made
5 all the mistakes that can be made
6 in a very narrow field.''
7 Albert Einstein said:
8 \begin{quote}
9     Anyone who has never made a mistake
10    has never tried anything new.
11 \end{quote}
12 Errors are inevitable. So,
13 let's be brave trying something new.
14 \end{document}
```

首先我们使用了行内引用。产生了一个左引号;这个字符也称作 **backtick**。产生了个右引号。我们只是键入两个这样的符号来获得双引号。

然后我们使用 **quote** 环境来显示引用。我们并没有开始一个新的段落,因为引用已经设置了缩进。这就是我们在环境中前后不使用空行的原因。

2.5.2 引用更长的文本

当你写短的引用的时候,the **quote environment** 看起来非常好。然而,当你需要引用的文本有好几段的时候,你可能会希望获得与环绕文字相通的缩进格式。**The quotation environment** 将会为你解决问题。

2.5.3 Time for action-quoting TeX's benefits

```
1 \documentclass{article}
2 \usepackage{url}
3 \begin{document}
4 The authors of the CTAN team listed ten good reasons
5 for using \TeX. Among them are:
6 \begin{quotation}
7     \TeX\ has the best output. What you end with,
8     the symbols on the page,
9     is as useable, and beautiful,
10    as a non-professional can produce.
11    \TeX\ knows typesetting. As those
12    plain text samples
13    show, \TeX's has more sophisticated
14    typographical algorithms
15    such as those for making paragraphs
16    and for hyphenating.
17    \TeX\ is fast. On today's machines
18 \end{quotation}
19 \end{document}
```

Chapter 3

页面设计

学过了前一章,格式化文本对我们来说已经很简单了。所以,让我们转到整个页面。在这一章中,我们将学习:

- 调整边框
- 改变行宽
- 区间化文档
- 创建一个内容表格
- 设计页眉和页脚
- 控制分页
- 设置脚注并修改样式

在学习这些的过程中,我们将会对类和包有更深入的认识。

3.1 定义全局布局

我们将写个包含几页的示例文档。我们将在这个文档上练习修改边距,行距,页眉和页脚等和其他更多的东西。

3.1.1 Time for action-写个包含章节的书

我们将开始写一本书。首先,我们将选择一个类,更进一步使用一些填充文本来试验页面布局。

```
1 \documentclass[a4paper,12pt]{book}
2 \usepackage[english]{babel}
3 \usepackage{blindtext}
4 \begin{document}
5 \chapter{Exploring the page layout}
6 In this chapter we will study the layout of pages.
7 \section{Some filler text}
8 \blindtext
9 \section{A lot more filler text}
10 More dummy text will follow.
11 \subsection{Plenty of filler text}
12 \blindtext[10]
13 \end{document}
```

我们选择的文档类型为 **book**。如名字所指示,这个类适合于书一类的文档。书通常是两面的,包含章节 **which usually start at right-hand pages**。它们可以有一个前言具有一个或多个标题页和一个 **back matter with bibliography, index** 等等。**book** 类支持所有这些。

我们加载了 **babel** 类。这对于非英语 **typesetting** 非常有用。

即使对于英语,这里也有几个选项:

USenglish, american, english, UKenglish, british, canadian, australian, and newzealand。显然,一些是相同的,例如 **UKenglish** 和 **british**。

然而, **There are differences in hyphenation rules between USenglish(american, english) and UKenglish (british)**。

现在,我们只需要 **babel** 来加载 **blindtext**: 这个包被开发用来产生填充文本。它需要 **babel** 来探测文档语言。我们向 **babel** 声明语言为 **english**, 意味这 **American English**。

命令 **\chapter** 产生一个大标题。这个命令总是开始于新的一页。

我们已经了解了 **\section** 命令。它是我们的二级标题,产生一个比

\chapter 小一点的标题。它自动为每个章节计数。

接着是 **\blindtext**, 打印一些无意义的文本来填充空间。

最后, **we refined the sectioning with a \subsection** 命令和更多的

Chapter 4

创建列表

4.1 项目符号列表

4.2 数字列表

4.3 定义列表

Chapter 5

创建表格和插入图片

5.1 以列的方式排列文本和数据

5.2 排版复杂的表格

5.3 在文档中插入图片

5.4 给表格和图片添加标题

5.5 控制图表和表格的布局

Chapter 6

交叉引用

6.1 设置引用标签

6.2 区间, 脚注, 列表条目, 表格和其他的引用

6.3 页码的引用及 **ranges**

6.4 使 **LaTeX** 对长页面的冗余引用

6.5 引用的自动命名

6.6 创建到外部文档的引用

Chapter 7

内容和引用的列表

- 7.1 决定要插入 **TOC** 的标题的级别
- 7.2 创建并个性化图列表 (**LOF**) 和表格列表 (**LOT**)
- 7.3 将随意的文本和命令插入到 **TOC** 和其他列表
- 7.4 引用和参考书目
- 7.5 创建目录
- 7.6 修改所有这些列表的标题

Chapter 8

编辑数学公式

8.1 编辑基本公式

8.2 文本中插入公式和公式间插入文本

8.3 等式的中间对齐并编号

8.4 多列等式的对齐

8.5 插入数学符号例如:根号,操作符,希腊字母和箭头

8.6 分数

8.7 表达式的堆叠

8.8 矩阵

Chapter 9

字体使用

9.1 为整个文档选择其他字体集

9.2 为特定的形状加载字体包

9.3 如何安装附加字体

9.4 多种字体的使用

9.5 输入文本的编码和输出的字体

Chapter 10

创作大型文档

10.1 一个文档分为几个文件

10.2 代码的输入和重用

10.3 **swap out document** 设置

10.4 编译文档的一部分

10.5 创建一个基于其他 **sub-files** 的主文档

10.6 设计标题页

10.7 在标题页使用罗马数字页码

10.8 **add a back matter without sectioning numbers**

Chapter 11

进一步美化文档

11.1 为目录, 图标列表和其他的交叉引用加上超链接

11.2 创建书签的导航条

11.3 设置标题

11.4 使用 **LaTeX** 的颜色

Chapter 12

故障排除

12.1 理解错误信息并修复

12.2 处理警告

12.3 使用 **LaTeX** 在编译期间产生的所有信息

Chapter 13

使用在线资源

xxx

.1 列表

1. 概述

enumerate 有序列表,会有数字前缀。此行使用 `hfill` 填充标签与描述之间的空白。

itemize 使用圆点前缀。

description 需要在 `item` 后的方括号中指定标签

5. 标签和计数器 枚举的列表计数可以通过其计数器来改变。`enumerate` 提供了四个计数器 `enumi,enumii,enumiii,enumiv` 对应不同层次的枚举。值的改变可以通过 `setcounter{enumi}{4}` 来实现。则会输出 5. 对于 `itemize` 而言,无需计数。

Appendix A

后记

Appendix B

参考书目