

LaTeX HandBook

Michael Jetson
NCEPU

2023 年 2 月 12 日

目录

I	LaTeX 基础	1
1	LaTeX 基础	2
1.1	LaTeX 的历史	2
1.2	用 LaTeX 书写你的第一篇文章	2
1.3	让文章内容更丰富——使用列表	3
II	LaTeX 进阶	5
2	数学公式	6
2.1	L ^A T _E X 书写数学公式的好处	6
2.2	书写第一个数学公式	6
3	命令	8
3.1	为什么定义命令	8
3.2	定义一个新命令	8
3.3	带参数的命令	9
3.4	默认参数的命令	9

Part I

LaTeX 基础

Chapter 1

LaTeX 基础

1.1 LaTeX 的历史

大家对排版这个词应该不陌生，除去小时候耳熟能详的活字印刷，大家现在应该都搞过论文排版，不论是结课论文或者是其他论文，而且相信大家都用过 Word 排版，不管是微软的 Word 还是金山的 WPS，大家都对 Word 排版不陌生，入门容易，而且功能很不错，但是大家会发现那些排版非常精美的论文版面以及复杂的数学公式，Word 难以制作出来，这是因为这些排版使用了更为强大的排版工具——LaTeX。

不过，我无意于给大家详细介绍 LaTeX 的历史，所以对 LaTeX 历史感兴趣的同学可以自行百度或者维基百科，在这里我们只会简单介绍 LaTeX 的用途以及相比于其他排版系统（如 Word）的优势。

LaTeX（音译“拉泰赫”）是一种基于 TEX 的排版系统，由美国计算机学家莱斯利·兰伯特（Leslie Lamport）在 20 世纪 80 年代初期开发，利用这种格式，即使使用者没有排版和程序设计的知识也可以充分发挥由 TeX 所提供的强大功能，能在几天、甚至几小时内生成很多具有书籍质量的印刷品。对于生成复杂表格和数学公式，这一点表现得尤为突出。因此它非常适用于生成高印刷质量的科技和数学类文档。这个系统同样适用于生成从简单的信件到完整书籍的所有其他种类的文档。

1.2 用 LaTeX 书写你的第一篇文章

我们打开 overleaf 网站，新建一个项目，然后就可以看到项目里面自动生成了一个名为 main.tex 的文件，写过编程语言的同学应该会联想到主函数，与这个有异曲同工之妙，当然没学过编程语言的同学也无需担心，可以理解为所有的功能都在这个里面实现即可。

文件中内容如下

```
1 \documentclass{article}
2 \usepackage[utf8]{inputenc}
3 \title{demo}
4 \author{your name}
5 \date{January 2023}
6 \begin{document}
7 \maketitle
8 \section{Introduction}
```

```
9 \end{document}
```

然后点击右上角的 Recompile 进行编译，就可以看到右边出现了一个 PDF 文档界面
接下来我们就开始逐行解析代码：

```
1 \documentclass{article}
2 %这是声明文件的文档类型为 article（文章类型），百分号%代表注释
3
4 \usepackage[utf8]{inputenc}
5 %字体编码，用来嵌入字体
6
7 \title{demo}
8 %文章标题为“demo”
9
10 \author{your name}
11 %作者名字
12
13 \date{January 2023}
14 %日期
15
16 \begin{document}
17 %代表文档主体部分的开始，用LaTeX的术语来说就是开启了一个新的环境
18
19 \maketitle
20 %生成标题，使标题可以显示
21
22 \section{Introduction}
23 %开始一个章节，章节名为Introduction
24
25 \end{document}
26 %声明文档主体部分的结束
```

然后你可以在环境里面加入你自己的文字，或者更改文章标题和章节名，编译后就可以生成一篇真正的文章了，虽然这个文章很简单，但是不要急，我们会继续学习如何让文章更加精美。

1.3 让文章内容更丰富——使用列表

我们想要往文章里面添加各种东西，让文章更丰富，那应该如何完成呢？用过 Word 的同学都知道，除了正文，还有各种级别的标题啊、列表啊什么的，所以我们在这一节将高速大家如何实现分级标题和列表。

我们将 main.tex 的内容修改为以下代码：

```
1 \documentclass{article}
2 %这是声明文件的文档类型为 article （文章类型）
3 \usepackage[utf8]{ctex}
4 %使用ctex宏包以及UTF8编码，可以支持中文，否则中文内容无法显示
5 \title{demo}
6 %文章标题为 “demo”
7 \author{your name}
8 %作者名字
9 \date{January 2023}
10 %日期
11 \begin{document}
12 %代表文档主体部分的开始，用LaTeX的术语来说就是开启了一个新的环境
13 \maketitle
14 %生成标题，使标题可以显示
15 \section{Introduction}
16 %开始一个章节，章节名为Introduction，可理解为一二级标题
17 \subsection{Introduction}
18 %可理解为二级标题
19 \subsubsection{Introduction}
20 %可理解为二级标题
21 \begin{itemize}
22 %开启一个无序列表环境，每一个\item后空格（可以一个也可以多个，不影响）
23 %，加上文本就可以生成无序列表
24     \item 第一个项目
25     \item 第二个项目
26 \end{itemize}
27 \end{document}
28 %声明文档主体部分的结束
```

接下来我们就会发现，文章中出现了一个无序的列表

- 这是第一项
- 这是第二项

Part II

LaTeX 进阶

Chapter 2

数学公式

2.1 L^AT_EX 书写数学公式的好处

各种涉及数学的专业的同学应该都看到过各种书籍、论文里面排版精美的数学公式，但是大家发现在 Word 里面书写数学公式总有各种各样的局限，尤其是在书写复杂公式和各种数学符号的时候，Word 总是难以完成，但是 LaTeX 不存在这个问题，使用 LaTeX 可以排版出非常精美的公式，实际上大家所看到的论文中的公式，基本上都是由 LaTeX 进行排版的。

2.2 书写第一个数学公式

我们在文档里面输入以下内容

```
1 $$x^2$$
```

然后就可以看到对应位置有以下输出：

$$x^2$$

接下来我们来解析一下：首先，我们要使用 `$$` 将我们要显示的数学公式包起来，这样告诉编译器，我们要书写数学公式了，如果不这样，字母可以显示，但是很多数学符号就无法显示，还会报错，而使用 `$$` 将公式包起来，会让数学公式独占一行显示（也叫行间显示），如果不想单独一行显示，可以用 `$` 将公式括起来

```
1 $x^2$
```

这样可以在行内显示，比如说这样： x^2 。

然后 `x` 是字母，`^` 表示上标，`^` 跟在 `x` 后面，并且 `^` 后面带有一个 `2` 表示上标的内容。

如果能让上标包含的内容更多，就需要将上标想显示的内容括起来

```
1 $$x^{\{1234\}}$$
```

```
2 $$x^{1234}$$
```

$$x^{1234}$$

$$x^1234$$

这是因为上标符号 $^$ 只会将后面第一个字符认定为上标内容，如果普通的多个字符（如 ABC，123 等），则只会将第一个字符变为上标，所以我们需要使用或者其他类似的命令，将一串字符变为一个整体，就可以让上标显示更多内容，同理，在其他数学符号里面也是如此。

Chapter 3

命令

3.1 为什么定义命令

对于命令，大家之前也了解过，一个 `\` 加上相应的单词，就可以让编译器进行相应的操作，这就是命令，但是如果我们想对某一种类似的操作进行重复使用，那么直接进行各种命令的叠加会非常麻烦，所以我们需要定义自己命令来进行各种操作的集成，这就是我们为什么要进行定义新命令的原因。

3.2 定义一个新命令

\LaTeX 的命令相当于其他编程语言中的函数，我们定义新命令的过程跟定义函数的过程也类似，首先我们来定义一个最简单的命令

```
1 \newcommand{\MyCMD}{\textbf{这是我们自定义的命令}}
```

其中，`\newcommand` 是标识符，告诉编译器我们要定义一个新命令，然后 `\MyCMD` 是我们定义的新命令的标识符，类似于函数名，我们在文档中定义这个标识符之后，就可以进行使用，然后第二个花括号里面是新命令的内容，也就是我们使用这个新命令之后会进行什么操作，在这里我们定义这个指令的操作是输出加粗的文字“这是往年自定义的命令”，使用方法及效果如下

```
1 \MyCMD
```

这是我们自定义的命令

当然，第二个括号里面的内容也可以由大家自己设计，可以是任意命令的合法组合，从而实现各种操作。

此外，定义新命令的代码也可以这样写

```
1 \newcommand\MyCMD{\textbf{这是我们自定义的命令}}
```

这里两种写法没有任何实质性区别。

注意，命令的命名只能是大小写字母，不能包括数字下划线等，否则会报错。

这样，我们就可以把各种操作合并为一个简短的命令里面，从而大大提高我们的排版效率。

3.3 带参数的命令

但是，大家思考一下，上面的命令有什么问题呢？是不是过于死板了，只能执行既定的操作，除非你改变命令内部，否则一个命令只能执行这些操作，那我们在处理一些灵活任务的时候就会受到限制，所以我们就需要带参数的命令，就跟需要传入参数的函数一样，功能更强大。首先我们定义一个带有一个参数的命令：

```
1 \newcommand{\MyCMD1}[1]{\textbf{#1}}
```

大家可以发现，这里定义新命令的时候，多了一个中括号，这个括号里面还有一个数字，这个数字就代表参数的个数，但是注意一下，一个命令的参数个数最多是 9 个，然后我们展示一下用法与效果

```
1 \MyCMDone{hello}
```

hello

在这里我们向命令传入了一个参数“hello”，然后根据我们定义的操作，将其加粗展示，在命令里面 #1 就是代表第一个参数，大家也可以进行其他操作。

如果我们想同时处理多个参数，那么可以这样定义，比如我们定义一个三参数的命令：

```
1 \newcommand{\MyCMDtwo}[3]{(#1+#2)^#3}
2 $\MyCMDtwo{x}{y}{2}$
```

使用的时候，有几个参数就要用几个花括号括起来

$(x + y)^2$

这样我们就实现了多参数命令的定义与使用。

3.4 默认参数的命令

除去以上用法，我们还可以定义带有默认参数的命令，如果我们不传入特定参数，那么就会使用默认的参数进行操作

```
1 \newcommand{\MyCMDthree}[3][x]{(#1+#2)^#3}
2 $\MyCMDthree{y}{2}$
3
4 $\MyCMDthree[z]{y}{2}$
```

$(x + y)^2$

$(z + y)^2$

大家可以看到，在这个命令里面，当我们只传入的两个参数的时候，但是输出却包含三个参数，这就是第一个参数是默认参数导致的，在定义环节，我们在定义参数个数地方的后面使用方括号定义默认参数，如果我们想传入一个参数来代替默认的参数，那就在使用命令的时候，用方括号传入，同时要注意，默认参数的个数只有一个，并且只能是第一个参数。

如果我们想传入一个空参数，那么可以这样

```
1 $\MyCMDthree[] {y}{2}$
```

显示效果如下：

$(+y)^2$

可以看到本来应该显示 x 的地方消失了，这是因为我们传入了一个空参数，所以导致显示第一个参数的位置显示为空。

