

使用 L^AT_EX 撰写报告

杨大伟
陈少峰-李大军团队

2015 年 11 月 4 日

本文的最新版本见于<https://github.com/yangdaweihit/latexreport>。

目录

1 前言

随着我们的讨论日渐深入，论述角度和结论也变得多样和细致。如果我们现在回想一下所经历的论证过程，可能已经难以列举出所有细节。我们的研究是猜想后求证，一旦在某个细节上存在偏差，那么就要回到前一个已验证节点处重新做出判断，再继续做验证。为了能够清晰记录我们所经历的研究历程，需要做出及时和准确的报告，这样才能保证我们的研究方式不会随着研究历程的变长而变得低效。现在我恢复了曾经使用过的 L^AT_EX 模板 `hitec.cls` 用于我们日常的报告。该模板来自于 Eli Billauer(2001)(eli_billauer@yahoo.com)。

为了平衡我们可怜的学习成本和所追求的高效率，在未来一段时间内，我会用以 L^AT_EX 的形式撰写报告，同时让小鲍完善细节。各位研究生随后以我报告为示例，撰写自己的报告。在这段过渡时间内，大家可以同时使用 Word 模板和 L^AT_EX 模板，视自己的工作效率而定。但在 11 月之后需全部转为 L^AT_EX 的写作方式。

本文帮助大家度过使用 L^AT_EX 最初比较痛苦的阶段，可以使您熟悉最基本的 L^AT_EX 命令和环境，使用这个模板做出高效和高质量的研究报告。当然，本文假定

你已经看过某人在你面前演示过 \LaTeX 的编译过程，教会了你如何安装 `texlive` 和如何使用某款撰写和编译 \LaTeX 的集成环境，如 `TeXMaker`。

本文包括 \LaTeX 基本命令和环境、自定义命令和环境、自定义符号的方法。为了实现合作写作，我们还做了一些关于文件组织的约定。在模板的使用过程中，我们会发现一些实际需求，比如老师的批注、对其批注的引用等，对于需求我们将定义一些命令或环境，这样有助于我们提升交流的质量。在 \LaTeX 的介绍过程中，文中还提供了相关内容的来源。追溯这些来源，我们就可以慢慢详细了解 \LaTeX 知识。

2 \TeX 和 \LaTeX 的历史

在不了解关于 \TeX 和 \LaTeX 的最基本历史之前可以学习它们吗？当然可以。但如果你的好奇心带给你太多疑问，比如：这东西看起来怎么这么怪？它怎么发音呀？它到底是什么？我为什么要学习它？那最好通过阅读尽快解决这个问题，这里提供两个链接：

- 中文介绍：http://www.ctex.org/documents/shredder/tex_frame.html
- 英文介绍：<https://en.wikipedia.org/wiki/LaTeX>

第一次就能读准它很重要： \TeX ——“泰赫”和 \LaTeX ——“雷泰赫”。

3 自学资料

如果你没兴趣看本文的解释，情愿自己去学习权威教程或获取更丰富的资源。那就给你推荐几个好去处。

3.1 入门教程

- 维基教程：<https://en.wikibooks.org/wiki/LaTeX>
- TUG 官网教程：<http://www.tug.org/twg/mactex/tutorials>
- \LaTeX for Complete Novices: 本地文档，执行命令打开该文档。

```
texdoc dickimaw-novices
```

- L^AT_EX 2_ε: An unofficial reference manual

```
texdoc latex2e
```

- The Not So Short Introduction to L^AT_EX 2_ε

```
texdoc lshort
```

- 刘海洋. L^AT_EX 入门, 电子工业出版社, 2013

3.2 网站

- TUG: <http://www.tug.org>, T_EX 用户群 (T_EXUsers Group, TUG) 网站, 全世界 T_EX 用户的组织。这里可以下载到 TeXLive CD。
- CTAN: <http://www.ctan.org>, (Comprehensive TeX Archive Network, CTAN), 你想要的所有文档、模板都在这里。再强调一下, 是所有的都在这里。
- LaTeX: <http://www.latex-project.org/>, L^AT_EX 官方网站。
- Stackexchange : <http://stackexchange.com>, 专业的技术交流网站。
- L^AT_EX 工作室 : <http://www.latexstudio.cn>, 国内稀有高水准排版服务。

4 文档结构

用 L^AT_EX 写作要经历如下过程：

- 撰写 tex 文档；
- 编译文档并转为特定的阅读格式，一般是 pdf 文档。

我们要做的其实只是撰写文档，其余部分都会通过简单的命令由程序完成。首先介绍 tex 文档的基本结构。L^AT_EX 文档包括导言和主体两部分。

导言部分用于声明文档的类型和引用的宏包。文档类型决定了文档编译后所套用的模板、为文档撰写准备的一些变量设置，本文所套用的模板就是 hitec，作者说要做一个简单的用于高科技写作的模板。

主体结构由一个环境构成`\begin{document} ... \end{document}`。在这两个命令中间便是你撰写文档内容的地方。文档中`%`之后本行内容不被编译，换言之，`%`是注释符。所有 \LaTeX 命令都是用`\`开头的，它的中文名字叫反斜杠。

```
% ----- 引言区
\documentclass{hitec}
\usepackage[UTF8]{ctex} %引用宏包并加注选项
\usepackage{amsmath}
\usepackage{tcolorbox}
\tcbuselibrary{listings}
% ----- 主体
\begin{document}
%您的大作
\end{document}
```

当你写完了文档，就可以编译文档了，比如文档名为 `main.tex`。如果在命令行中操作，是这样的：

```
> latex main.tex
```

但我们多数文档会含有中文，此时应使用 `xelatex` 编译：

```
> xelatex main.tex
```

然而实际上我们很少在命令行中输入命令编译，多数是在一个集成环境中用快捷完成编译。比如在 `TeXMaker` 中的 `F1` 键，在 `Emacs AUCTeX` 中的 `C-c C-c`。总之，在撰写 \LaTeX 文档时，有一款高效的软件是非常必要的。

5 基本命令和环境

命令和环境驱动了整个文档的编译。命令的基本基本结构是`\<命令名>[选项]{参数}`。环境是由一对命令构成的——`\begin{环境名}[选项]`和`\end{环境名}`。`[选项]`分为必选和可选两类，由具体命令和环境的定义决定。

5.1 题目、作者和日期

文章的开头首先是题目和作者、日期。这三个信息都以命令形式定义的。它们都需要在导言区定义，然后在正文中用一条命令`\maketitle`将它们显示出来。

```
...
\title{都闪开，我要用\LaTeX{}写作了}
\author{西门吹牛}
\date{2015年某月某日}
...
\begin{document}
\maketitle % 没有这条命令，前面的定义都烟消云散。
...
\end{document}
```

5.2 中文输入

在导言区只要引用了宏包`ctex`，即可用命令`xelatex`编译中文了。注意，一般要在选项中说明使用编码`UTF8`。

```
\usepackage[UTF8]{ctex}
```

5.3 段落

分段有两种方式：

- 在段落后填加`\\`
- 在段落后填加一个空行。

5.4 章节

章节命令能用到的就三个：

- 章：`\chapter{绪论}`，在我们这个模板中这一级题目是无效的。
- 节：`\section{前言}`
- 次节：`\subsection{背景}`

5.5 列举

```
\begin{itemize}
\item 首先，我要强调。
\item 其次，我还要说明。
\end{itemize}
```

- 首先，我要强调。
- 其次，我还要说明。

5.6 字体

我们在写作时为了强调某些内容，或纯粹为了避免单调，会适度地转变字体。在排版中，字体其实是一个含义丰富的术语。当我们在说一种字体时，实际上是一个属性集合，它包括：

```
\begin{description}
\item[大小] 比如：{\zihao{3}字号三}、{\zihao{5}字号五}、{\zihao{7}字号七}。
\item[字形] 比如：粗体{\textbf{bfseries}}、斜体{\textsl{slant}}、
打印体{\texttt{italic}}。
\item[字族] 比如：\songti{宋体}、\heiti{黑体}、\fangsong{仿宋}。
\end{description}
```

大小 比如：**字号三**、**字号五**、**字号七**。

字形 比如：粗体 **bfseries**、斜体 *slant*、打印体 *italic*。

字族 比如：宋体、**黑体**、仿宋。

其它还有一个属性“字族”，因很少涉及，这里暂略去。关于字体还有更丰富的命令，暂时也不详谈。

改变字体一般有两种方式：一种是改变所选择的文字，称为“区域命令”；另一种是改变命令声明以后的字体，称为“声明命令”或“模式命令”。

也许你会在有的地方看过命令，如 `\bf`、`\md`、`it`、`\sl`、`\sc`、`\sf`、`\tt`，或 `\rm` 等。这些命令已经是陈旧命令，不提倡再使用。原因这些命令不允许复合作用，如：

```
\it{\bf{ it and bf format}}
```

it and bf format

显然, `\it` 命令已经被`\bf`屏蔽了。1994 年发布了 $\text{\LaTeX} 2_{\epsilon}$ 取代了老版本的 \LaTeX 的 2.09 版, 使用了新的命令风格`\textxx`, 如:

```
\textit{\textbf{Never mind the rain}}
```

Never mind the rain

改进后的字体命令就可以嵌套作用了。

字体声明命令见表??。字体声明命令还可以作为环境使用。表中命令嵌套使用对临时设置字体时很重要。

表 1: 字体改变声明

Declaration	Example Input	Corresponding output
<code>\rmfamily</code>	<code>\rmfamily roman text</code>	roman text
<code>\sffamily</code>	<code>\sffamily sans serif text</code>	sans serif text
<code>\ttfamily</code>	<code>\ttfamily typewriter text</code>	typewriter text
<code>\mdseries</code>	<code>\mdseries medium text</code>	medium text
<code>\upshape</code>	<code>\upshape upright text</code>	upright text
<code>\itshape</code>	<code>\itshape italic text</code>	<i>italic text</i>
<code>\slshape</code>	<code>\slshape slant text</code>	<i>slant text</i>
<code>\em</code>	<code>\em emphasized text</code>	<i>emphasized text</i>
<code>\normalfont</code>	<code>\normalfont default text</code>	default font

5.7 特殊字符

在文档写作中, 经常涉及希腊字母或其它特殊字符的输入。 \LaTeX 已经状态准备好了这些字符, 只需要通过命令即可实现输入。

```
%\usepackage{amsmath}
 $\alpha$   $\beta$   $\xi$   $\sigma$   $\theta$ 
```

$$\alpha \beta \xi \sigma \theta$$

这些字符已经汇总到了一个文档中供查询—`symbols-a4.pdf`。在字符界面中，输入如下命令即可自动找到该文档并用系统默认的 pdf 浏览器打开：

```
> texdoc symbols-a4
```

还有一些网站专门提供了手写识别 L^AT_EX 符号的功能，只需要用鼠标画出查询的符号，网站就会推荐出可能的符号命令。这里推荐一个：

<http://detexify.kirelabs.org/classify.html>

5.8 数学公式

参考阅读：<https://en.wikibooks.org/wiki/LaTeX/Mathematics>

```
%\usepackage{amsmath}
```

一行文字之间的公式叫行内公式，比如：`$x^2=1$`。行间公式用公式环境，比如：

```
\begin{equation}
```

```
\int_{0}^{1} x^2 \mathrm{d}x = \mathrm{dfrac{1}{3}}
```

```
\end{equation}
```

式中：注意`d`是正体，表示微分符号。

一行文字之间的公式叫行内公式，比如： $x^2 = 1$ 。行间公式用公式环境，比如：

$$\int_0^1 x^2 dx = \frac{1}{3} \quad (1)$$

式中：注意 d 是正体，表示微分符号。

5.9 表格

参考阅读：<https://en.wikibooks.org/wiki/LaTeX/Tables>

表格和后面要讲的插图，在 L^AT_EX 中称为浮动体。各类表格环境声明于浮动体 `table` 之中。浮动体的位置通过选项为控制，一般我们为了使表格处于期望的当前位置，需要引用一个宏包 `float`，然后使用选项 `[H]`。要引用表格，需要在浮动体中用命令`\label`声明该浮动体的名字，再用`\ref`引用即可。


```

\begin{table}[H]
\caption{tabular示例}
\label{table:first}
\begin{center}
%\begin{tabular}{rcl}
\begin{tabular}{>{\raggedleft}p{0.2\linewidth}
               >{\centering}p{0.2\linewidth}
               p{0.2\linewidth}}
\toprule
左对齐列 & 中间对齐列 & 右对齐列 \\
\hline
abcde   & o & edcba \\
123     & 0 & 321 \\
\bottomrule
\end{tabular}
\end{center}
\end{table}

```

计算结果见表\ref{table:first}。

表 2: tabular 示例

左对齐列	中间对齐列	右对齐列
abcde	o	edcba
123	0	321

计算结果见表??。

\LaTeX 中的表格仅有 `tabbing` 和 `tbluar` 两种环境, 为了丰富和提升表格的功能, 大量表格类宏包被开发出来, 如可定制宽度的 `tabularx`、可跨页的 `longtable`、具有更多丰富功能的 `tabu`、以及彩色盒子 `tcolorbox` 等。本文中的所有代码实际上就是被置于 `tcolorbox` 环境之中。

下面我们仅就最常用的 `tabularx` 给出示例, 其余内容见给出的网络链接。

```

\begin{table}[H]
\caption{tabularx示例}
\begin{center}
\begin{tabularx}{0.7\linewidth}{rp{0.2\linewidth}p{0.2\linewidth}}
\toprule
左对齐列 & 中间对齐列 & 右对齐列 \\
\midrule
abcde & o & edcba \\
123 & 0 & 321 \\
\bottomrule
\end{tabularx}
\end{center}
\end{table}

```

表 3: tabularx 示例

左对齐列	中间对齐列	右对齐列
abcde	o	edcba
123	0	321

tabular 和 tabularx 是可以嵌套使用的。

```

\begin{table}[H]
\centering
\caption{嵌套tabular}
\begin{tabularx}{0.8\linewidth}{p{0.3\linewidth}X}
\toprule
分类 & 描述 \\
\midrule
类型一 & \begin{tabular}[t]{ll}
子类A & ab \\
子类B & sadf \\
子类C & adf
\end{tabular}
\end{tabularx}

```

```

类型二 & 较为简单的情况 && \\
\bottomrule
\end{tabularx}
\label{tab:loopbaublar}
\end{table}

```

表 4: 嵌套 tabular

分类	描述
类型一	子类 A ab
	子类 B sadf
	子类 C adf
类型二	较为简单的情况

5.10 插图

参考阅读：https://en.wikibooks.org/wiki/LaTeX/Floats,_Figures_and_Captions

如前所述，插图是另一类浮动体环境——figure。如：

```

\begin{figure}[H]
\begin{center}
\includegraphics[width=.3\linewidth]{ctanlion}
\caption{\LaTeX{}的吉祥物}
\label{fig:lion}
\end{center}
\end{figure}

```

图 1: L^AT_EX 的吉祥物

L^AT_EX 的吉祥物是一只可爱的狮子，如图??所示，由 Duane Bibby 创作。插图默认的文件为 `.eps` 格式，也可以是 `.jpg` 或 `.pdf` 等。关于插图的更多内容参见网络链接。

5.11 参考文献

命令 `\bibliography` 用来引用某个 `bib` 文件，如：

```
\bibliography{./Reference/Reference.bib}
```

在 `bib` 文件中的所有文献都有一个 `Bibtexkey` 项代表该文献，在 `tex` 文件中用 `\cite` 命令即可通过该项值引用文献，如：

文献 `\cite{Wu2010}` 指出了如下观点：

5.12 目录

插入目录只需一条命令，位于 `\maketitle` 之后：

```
\tableofcontents
```

6 自定义

6.1 模板

L^AT_EX 最吸引人的地方就在于我们可以凭借定义好的模板，将全部精力放在写作中，而完全不必担心会出现格式错误。使用模板的方式很简单，即在文档类型中使用模板名即可，如：

```
\documentclass{hitec}
```

模板文件的扩展名为`.cls`，要使用 TeXLive 套件之外的模板，需要将该文件置于 T_EX 系统可搜索的目录中，然后刷新 T_EX 数据库，详见??节。

6.2 命令

为了简化某些固定的输入内容，可以使用`\newcommand` 自定义命令。比如，我们希望在公式中固定单位 MPa。注意，我们要求单位之前留有半个空格，以和数字区分开。但如果我们总是手写输入，难免会在格式上出错。现在自定义一个命令`\MPa`就可以避免这个麻烦。

```
\newcommand{\MPa}{\ \mathrm{MPa}}  
$5\MPa$  
-----  
5 MPa
```

自定义命令还可以带参数，如：

```
\newcommand{\emphtext}[1]{\textcolor{red}{\texttt{#1}}}  
\emphtext{这是一句重要的废话。}  
-----  
这是一句重要的废话。
```

6.3 环境

当我们要将特别定制的风格应用于较多的内容时，将它们作为命令参数被作用就显得不够方便。此时，我们可以自定义环境。自定义环境格式为：

```
\newenvironment{环境名}{环境起始定义}{环境终止定义}
```

一般自定义环境是已有环境的扩展，比如：

```
\newenvironment{centertt}{  
  \begin{center}  
  \fangsong  
  \color{blue}  
}  
{  
  \end{center}  
}  
  
\begin{centertt}  
  居中的蓝色仿宋体环境  
\end{centertt}
```

居中的蓝色仿宋体环境

6.4 自定义文件

当我们自定义的命令或环境越来越多，放在正文中会影响正文的阅读。这时要么将这些定义放在模板中，要么单独放在一个`tex`文件中，比如`symbols.tex`，然后用一条命令就纳入到文档中了：

```
\input{symbols.tex}
```

通常我们只会专注于某一领域的写作，所以随着这个文件的内容越来越多，我们将它放在其它文档中重复使用，以提高我们的写作效率。

7 文档组织

7.1 安装模板

在使用某个模板前，需要将模板加入到 `texlive` 系统中。这样编译文档时才能找到这个模板。把模板装到 `texlive` 中统共分两步：

- 将模板文件 `.cls` 或 `.sty` 拷贝到模板目录。模板目录在 `texlive` 安装目录中的子文件夹 `/texmf-local`。模板拷贝到这个文件夹中的任何子目录即可，但为了便于管理，还是再细致查找到合适的子目录再拷贝为好，比如：`/texmf-local/tex/latex/local`。
- 在字符界面（或称命令行）中 `texhash` 刷新 `texlive` 数据库。

注意，若在 `LINUX` 系统中需要获得管理员权限后运行刷新命令，我们可以看到刷新数据库的路径显示出来：

```
> sudo texhash
texhash: Updating /usr/local/texlive/2015/texmf-config/ls-R...
texhash: Updating /usr/local/texlive/2015/texmf-dist/ls-R...
texhash: Updating /usr/local/texlive/2015/texmf-var/ls-R...
texhash: Updating /usr/local/texlive/texmf-local/ls-R...
texhash: Done.
```

7.2 文件分类存放

通常我们的文档会越写越大，插图、子文档越来越多。为了便于我们的文档管理，需要分类存放文件。如插图放在子文件夹 `figure`，子文档放在子文件夹 `body` 中，而仅令将主文档（一般命名为 `main.tex`）放在一级目录中。在我们技术报告中，经常会反复使用一些数学符号，这时最好的处理方式就是将它们定义为命令，而这些命令定义最好放在一个专门的文件中，我们约定这个文件为 `symbols.tex`，也放在文件夹 `body` 中。

8 结语

基于以上介绍，你已初步了解了撰写一篇文档需要的大部分 `LATEX` 排版命令。真正掌握它们还需要大量的练习。除了网络和电子书籍，请教团队中的师兄和老师也

是非常好的捷径。

本文即是入门介绍，又是一个示例，将在未来的工作中不断补充和丰富，同时也欢迎你也加入到这个文档的写作之中。

待解决问题

表 5: 文档日志

描述	提出日期	解决日期	附注
建立文档	2015.10.03	2015.10.03	
减小表前表后空白	2015.10.03		
tabularx 对齐和设置固度	2015.10.03	2015.10.03	
设计了代码环境	2015.10.14	2015.10.14	
代码环境显示行号	2015.10.14		