

# L<sup>A</sup>T<sub>E</sub>X(雷泰赫) 简介

---

鸟语花香  
哈尔滨工业大学

2017 年 10 月 7 日

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

## 1 前言

在动笔之前，我们冒昧地对读者做了一个大胆的假设：您刚刚听说 L<sup>A</sup>T<sub>E</sub>X，虽然所知不多，但已决意学习。之所以需要这样的假设，纯粹是因为我们想暂时避开一个非常棘手的问题：为什么要学习 L<sup>A</sup>T<sub>E</sub>X。本文意在简介，只是扮演一个门童的角色，为您说明进入 L<sup>A</sup>T<sub>E</sub>X 殿堂的门径，无意以偏盖全。倘若被某位行家看到错误之处，还望指正。

### 1.1 L<sup>A</sup>T<sub>E</sub>X 的历史

在不了解 L<sup>A</sup>T<sub>E</sub>X 的背景知识之前可以学习它吗？当然可以。但考虑到每个人的学习习惯并不一样，这里提供一些链接，让好奇心更强的读者在学习之前大快朵颐一番。

- 中文：[http://www.ctex.org/documents/shredder/tex\\_frame.html](http://www.ctex.org/documents/shredder/tex_frame.html)
- 英文：<https://en.wikipedia.org/wiki/LaTeX>

第一次就能读准它很重要：L<sup>A</sup>T<sub>E</sub>X——“雷泰赫”。

## 1.2 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。

- <https://en.wikibooks.org/wiki/LaTeX>
- <http://www.tug.org/twg/mactex/tutorials>
- $\text{\LaTeX}$  for Complete Novices: 本地文档，执行命令打开该文档。

```
texdoc dickimaw-novices
```

- $\text{\LaTeX}$  2 $\epsilon$ : An unofficial reference manual

```
texdoc latex2e
```

- The Not So Short Introduction to  $\text{\LaTeX}$  2 $\epsilon$

```
texdoc lshort
```

- 刘海洋.  $\text{\LaTeX}$  入门，电子工业出版社，2013
- TUG: <http://www.tug.org>,  $\text{\TeX}$  用户群 ( $\text{\TeX}$ Users Group, TUG) 网站，全世界  $\text{\TeX}$  用户的组织。这里可以下载到  $\text{\TeX}$ Live CD。
- CTAN: <http://www.ctan.org>, (Comprehensive  $\text{\TeX}$  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。
- $\text{\LaTeX}$ : <http://www.latex-project.org/>,  $\text{\LaTeX}$  官方网站。
- Stackexchange: <http://stackexchange.com>, 专业的技术交流网站。
- $\text{\LaTeX}$  工作室: <http://www.latexstudio.cn>, 提供国内稀有高水准排版服务。还有一个 QQ 社区: 91940767。

## 2 文档结构

用  $\text{\LaTeX}$  写作统共分两步：

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

有人会问：这不和编程一样吗？是的，用  $\text{\LaTeX}$  撰写文档基本上就是一个“用文字编程”的过程。

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。 $\text{\LaTeX}$  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  $\text{\LaTeX}$  命令都是用`\`开头的，它的中文名字叫**反斜杠**。

```
% -----导言区
\documentclass{hitec}    %指定文档所套用的模板
\usepackage[UTF8]{ctex} %引用宏包并加注选项
\usepackage{amsmath}    %引用宏包
\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
```

`xelatex`，以及 `pdflatex` 一般称为`编译引擎`，其实就是个可执行程序。

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。

### 3 基本命令和环境

`命令`和`环境`是构成文档的基本单元。

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]
% 环境体
\end{环境名}
```

下面我们将分别介绍一些基本的命令和环境。

#### 3.1 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}
```

69

## 3.2 中文输入

70

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字体编码为UTF8。

71

72

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

73

## 3.3 段落

74

分段有两种方式：

75

- 在段落后填加`\\`
- 在段落后填加一个空行。

76

77

## 3.4 章节

78

章节命令能用到的就三个：

79

- 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}`

80

81

82

## 3.5 枚举

83

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

84

- 首先，我要强调。
- 其次，我还要说明。

### 3.6 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{eqnarray}`  
`\int_0^1 x^2 \mathrm{d}x = & \dfrac{1}{3} \\`  
`\int_0^1 x^2 \mathrm{d}x = & \sqrt{\dfrac{1}{3}}`  
`\end{eqnarray}`  
 式中：注意 `\mathrm{d}` 是正体，表示微分符号。

$x^2 = 1$ 。行间公式用公式环境，比如：

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

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

式中：注意 `d` 是正体，表示微分符号。

### 3.7 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。 $\text{\LaTeX}$  已经准备好这些字符，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  $\text{\LaTeX}$  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：

### 3.8 颜色

[https://www.sharelatex.com/learn/Using\\_colours\\_in\\_LaTeX](https://www.sharelatex.com/learn/Using_colours_in_LaTeX)

```
\textcolor{red}{红色}

{\color{blue} \rule{\linewidth}{0.5mm}}

\colorbox{BurntOrange}{颜色}
```

红色

颜色

### 3.9 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：

### 3.10 目录

插入目录只需一条命令，位于`\maketitle`之后：

```
\tableofcontents
```

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

## 4 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  $\text{\LaTeX}$  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1。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</code>	<i>emphasized text</i>
<code>\normalfont</code>	<code>\normalfont default text</code>	default font

## 5 表格

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

表格和后面要讲的插图, 在  $\text{\LaTeX}$  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。

134

```
\begin{table}[H]
\caption{tabular 示例}
\label{table:first}
\begin{center}
%\begin{tabular}{rcl}
\begin{tabular}{>\raggedleftp{0.2\linewidth}
>\centeringp{0.2\linewidth} p{0.2\linewidth}}
\toprule
左对齐 & 中间对齐 & 右对齐 \\
\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	中间对齐	右对齐
abcde	o	edcba
123	0	321

计算结果见表2。

135

$\text{\LaTeX}$  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。

136

137

138

139

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。

140

```
\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 & 中间对齐 & 右对齐 \\
\midrule
abcde & o & edcba \\
123 & 0 & 321 \\
\bottomrule
\end{tabularx}
\end{center}
\end{table}
```

表 3: tabularx 示例

左对齐	中间对齐	右对齐
abcde	o	edcba
123	0	321

141

`tabular` 和 `tabularx` 是可以嵌套使用的。

142

```
\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]{l}子类A & ab \\子类B & sadf \\子类C & adf \end{tabular} \\
\end{tabularx}
```

143

```

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

144

## 6 插图

145

参考阅读:[https://en.wikibooks.org/wiki/LaTeX/Floats,\\_Figures\\_and\\_](https://en.wikibooks.org/wiki/LaTeX/Floats,_Figures_and_Captions)  
 Captions

146

147

如前所述, 插图是另一类浮动体环境——figure。如:

148

```

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

149



图 1: L<sup>A</sup>T<sub>E</sub>X 的吉祥物

L<sup>A</sup>T<sub>E</sub>X 的吉祥物是一只可爱的狮子，如图1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。

## 7 自定义

### 7.1 模板

L<sup>A</sup>T<sub>E</sub>X 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<sub>E</sub>X 系统可搜索的目录中，然后刷新 T<sub>E</sub>X 数据库，详见8.1节。

### 7.2 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。

### 7.3 环境

当我们要将特别定制的样式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

## 7.4 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。

## 8 文档组织

### 8.1 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得到这个模板。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.
```

## 8.2 文件分类存放

195

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。

196  
197  
198  
199  
200  
201

## 9 Make

202

撰写 `MakeFile` 编译更方便。

203

## 10 结语

204

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  $\text{\LaTeX}$  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。

205  
206  
207

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。

208  
209

## 待解决问题

210

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		
<code>tabularx</code>	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		