

Contents

1 第一部分	3
1.1 简介	3
1.1.1 Latex 家族	3
1.1.2 发行版	4
1.2 安装与配置	4
1.2.1 Windows	4
1.2.2 Linux-Ubuntu	5
1.3 基本设置	6
1.4 newcommand 自定义命令	6
1.5 版面设置	7
1.5.1 页面设置	7
1.5.2 页眉页脚	7
1.5.3 首行缩进	7
1.5.4 行间距	7
1.5.5 段间距	8
1.6 表格	8
1.7 数学公式	10
1.7.1 根式 & 分式	10
1.7.2 定界符	11
1.7.3 省略号	11
1.7.4 矩阵	12
1.7.5 多行公式	12
1.7.6 公式组	12
1.7.7 分段函数	13
1.8 代码排版	13
1.9 定制章节样式	14
1.9.1 盒子	15

Latex 学习手记

白岩

March 19, 2017



1

第一部分

文档地址: <http://www.ctex.org/documents/packages/>

这是我正式决定将我的学习笔记写成电子版的第一篇。

纸质版的笔记排版太难受,修改也很麻烦,不如改成电子版的笔记,以后使用起来也很方便。由于写笔记使用的是 **Latex**,所以第一篇笔记就贡献给 **Latex** 了。

Latex 是常用的排版工具,本学习笔记用于自己以后复习和使用查表使用,故有些东西就不用讲的很详细了。**Latex** 在 Windows 和 Linux 上均可以使用,所以双系统的我就都安装配置了。

在此之前,或引用,或自己写,先记录一些基础的入门知识。

简介

Latex 家族

TeX, pdfTeX, XeTeX, LuaTeX, LaTeX, pdfLaTeX, XeLaTeX ...

TeX 是高德纳 (Donald Ervin Knuth, 1938 年 1 月 10 日-) 教授愤世嫉俗 (追求完美) 做出来的排版引擎,同时也是该引擎使用的标记语言 (Markup Lang) 的名称。

LaTeX 则是 L. Lamport (1941 年 2 月 7 日-) 教授开发的基于 **TeX** 的排版系统。实际上 **LaTeX** 利用 **TeX** 的控制命令,定义了许多新的控制命令并封装成一个可执行文件。这个可执行文件会去解释 **LaTeX** 新定义的命令成为 **TeX** 的控制命令,并最终交由 **TeX** 引擎进行排版。

因此在 **TeX** - **LaTeX** 组合中,

最终进行断行、分页等操作的,是 **TeX** 引擎;

LaTeX 实际上是一个工具,它将用户按照它的格式编写的文档解释成 **TeX** 引擎能理解的形式并交付给 **TeX** 引擎处理,再将最终结果返回给用户。

pdfTeX - **pdfLaTeX** **TeX** 系统生成的文件是 **dvi** 格式,虽然可以用其他程序将其转换为例如 **pdf** 等更为常见的格式,但是毕竟不方便。为了解决这个问题,Hàn Thế Thành 博士在他的博士论文中提出了 **pdfTeX** 这个对 **TeX** 引擎的扩展。二者最主要的差别就是 **pdfTeX** 直接输出 **pdf** 格式文档,而 **TeX** 引擎则输出

dvi 格式的文档。

pdfLaTeX 这个程序的主要工作依旧是将 LaTeX 格式的文档进行解释，不过此次是将解释之后的结果交付给 pdfTeX 引擎处理。

XeTeX - XeLaTeX 高德纳教授在实现 TeX 的当初并没有考虑到中日韩等字符的处理，而只支持 ASCII 字符。这并不是说中日韩字符就无法使用 TeX 引擎排版了，事实上 TeX 将每个字符用一个框包括起来（这被称为盒子）然后将一个个的盒子按照一定规则排列起来，因而 TeX 的算法理论上适用于任何字符。ASCII 字符简单理解，就是在半角模式下你的键盘能直接输出的字符。

在 XeTeX 出现之前，为了能让 TeX 系统排版中文，国人曾使用了天元、CCT、CJK 等手段处理中文。其中天元和 CCT 现在已经基本不用，CJK 因为使用时间长且效果相对较好，现在还有人使用。

不同于 CJK 等方式使用 TeX 和 pdfTeX 这两个不直接支持 Unicode 字符的引擎，XeTeX 引擎直接支持 Unicode 字符。也就是说现在不使用 CJK 也能排版中日韩文的文档了，并且这种方式要比之前的方式更加优秀。

XeLaTeX 和 XeTeX 的关系与 pdfLaTeX 和 pdfTeX 的关系类似，这里不再赘述。

使用 XeTeX 引擎需要使用 UTF-8 编码。

所谓编码就是字符在计算机储存时候的对应关系。例如，假设有一种编码，将汉字「你」对应为数字「1」；「好」对应为数字「2」，则含有「你好」的纯文本文件，在计算机中储存为「12」（读取文件的时候，将「12」再转换为「你好」显示在屏幕上或打印出来）。

UTF-8 编码是 Unicode 编码的一种。

发行版

CTeX, MiKTeX, TeX Live 都是被称为「发行」的软件合集。他们包括了上述各种引擎的可执行程序，以及一些文档类、模板、字体文件、辅助程序等等。其中 CTeX 是建立在 MiKTeX 的基础之上的。

TexLive 是跨平台可使用的。故我使用的主要就是 TexLive。

安装与配置

Windows

Windows 下的安装是很简单的。

一般我不使用 Texwork 作为编辑环境，我使用 Texmaker 作为写 LaTeX 文档的编辑器。

TexLive 和 Texmaker 的安装顺序没有要求。

<TexLive>

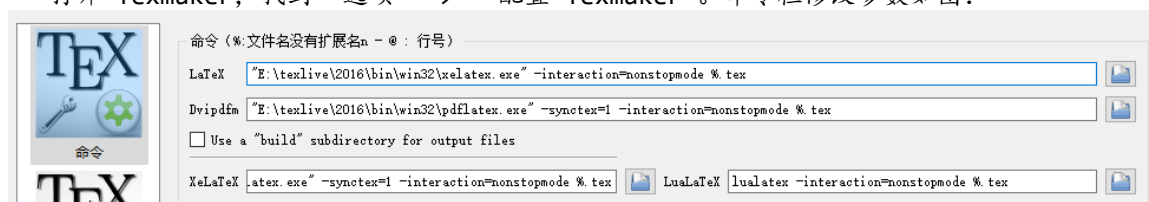
下载 Texlive 的镜像文件，大概 3GB 左右。然后双击安装就可以了。记住安装的文件夹，在安装的时候有些功能可以选择安装或不安装。比如 Texwork 就不用安装了。

<Texmaker>

这个的安装也是十分的简单，首先下载软件，然后双击安装就好了。

<配置 Texmaker>

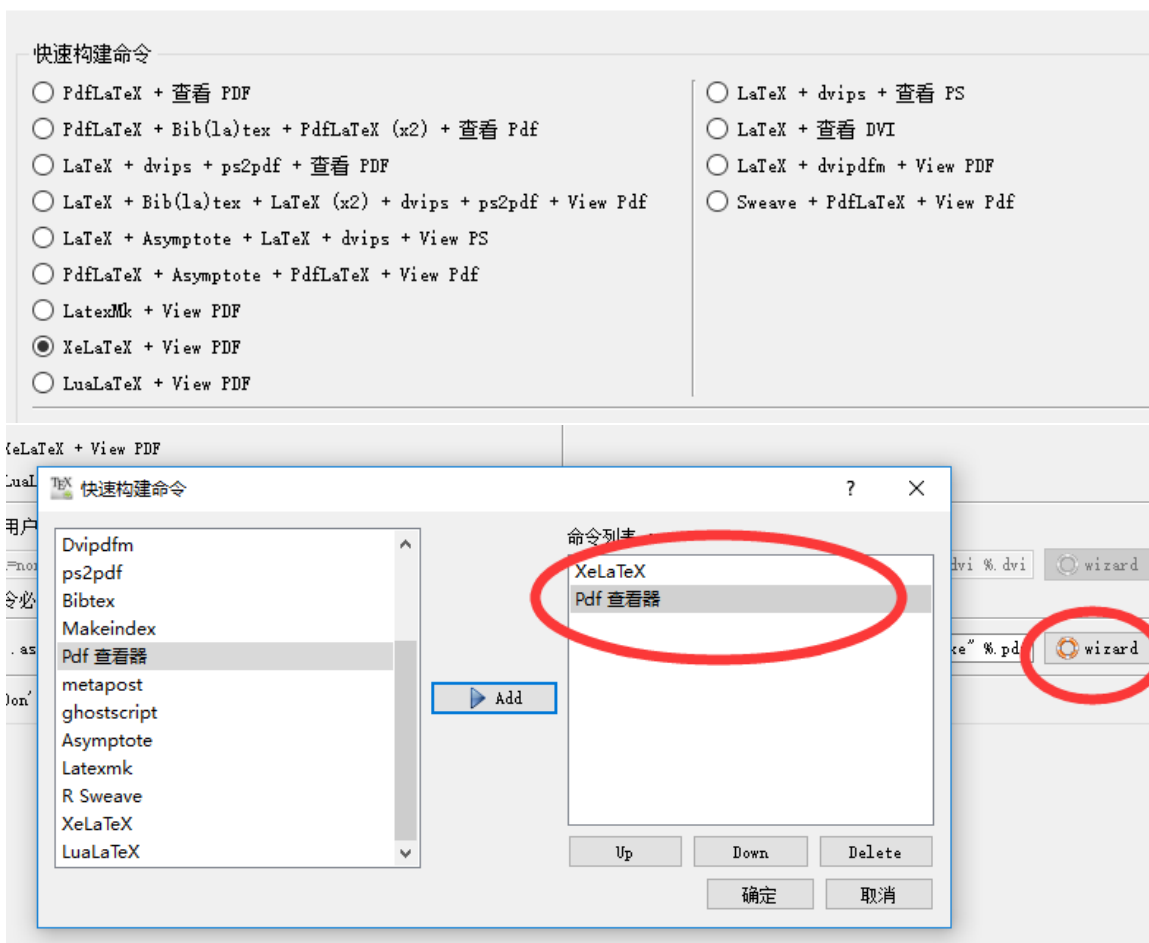
打开 Texmaker，找到“选项”->“配置 Texmaker”。命令栏修改参数如图：



其中双引号中的路径是你的 **TeXLive** 安装的路径，对应相应的可执行文件。需要修改的位置有三处，除了上面两处还有 **xeLaTeX**，内容与 **Latex** 一致。

可以发现，这里我们使用的都是 **xeLaTeX** 来进行编译解析。

下面配置“快速启动”选项。



快速启动需要配置快速启动的组合，和相应的处理程序。如上图，不多说。

Linux-Ubuntu

Linux 下的安装也是比较简单的。

首先使用命令安装

```
1 sudo apt-get install texlive-latex-base
2 sudo apt-get install latex-cjk-all
3 sudo apt-get install texmaker
4
```

这样基本的就安装完毕了，如果在运行的过程中会有写文件找不到，在错误的信息里可以看到缺少 **.sty**，此时就用命令查一下，然后安装就行了。命令如下：

```
1 apt-cache search lastpage (注意不要加.sty文件后缀)
2
3 // 可以看到需要下面的包，以及对这个包的描述：
4 texlive-latex-extra — TeX Live: LaTeX supplementary packages
5
6 // 选择安装即可：
7 sudo apt-get install texlive-latex-extra
8
```

< 配置 Texmaker >

配置的过程几乎是一样的，不同的就是在这里的 `xelatex` 选项不需要配置到相应的文件夹了。将 `Latex` 选项配置为 `xeLatex` 命令就可以了。其余的不变。

此外关于字体，可以使用命令查看现有的系统的字体：‘`fc-list :lang=zh-cn`’

基本设置

这里记录一下导言区常用的包的配置等等。

```
1 常用的导言区会在本文档的源文件中。
2
```

1. 常用的文档类型有：article、report、book 等。其中 article 从 Section 开始，report 从 chapter 开始，Book 从 part 开始。
2. 除了使用 CJK 包来实现中英文混排，还可以直接使用 cTex 来实现。使用 cTex 只需要在 documentclass 的时候指定 `[UTF8]{ctexart}{ctexrep}{ctexbook}`
3. 部分中带星号的属于不进行编号的部分。

newcommand 自定义命令

使用 `newcommand` 可以自己定制命令
语法格式：

```
1  \ newcommand{\yourcommand}[参数个数]{内容}
2
3
4  比如：
5  \ newcommand{\wuhao}{\fontsize{10.5pt}{10.5pt}\selectfont}
6
7  % 说明 \fontsize{}{}与\selectfont是LaTeX提供的字号控制低级命令，供用户自己设置字号大小。
8  % \fontsize{参数1}{参数1}中参数1为字号大小，参数2为行间距，
9  % 只有使用\selectfont命令之后，\fontsize{}{}的设置才能生效。切记
```

版面设置

页面设置

这部分直接参见源文件中的设置就可以了，可以设置纸张的大小，各个边的页边距等。

页眉页脚

使用到的宏包是：fancyhdr

在页眉左边写上我的名字，中间写上今天的日期，右边写上我的电话；页脚的正中写上页码；页眉和正文之间有一道宽为 0.4pt 的横线分割，可以在导言区加上如下几行：

```

1  \usepackage{fancyhdr}
2  \pagestyle{fancy}
3  \lhead{\author}
4  \chead{\date}
5  \rhead{152xxxxxxx}
6  \lfoot{}
7  \cfoot{\thepage}
8  \rfoot{}
9  \renewcommand{\headrulewidth}{0.4pt}
10 \renewcommand{\headwidth}{\textwidth}
11 \renewcommand{\footrulewidth}{0pt}
12

```

首行缩进

CTex 已经处理了首行缩进的问题，使用 xeCJK 需要用下面的方法。

```

1  \usepackage[indentfirst]{
2  //就算是这样，首行缩进的长度，仍然不符合中国人的习惯。我们可以在导言区添加这样的控制序列
3  \setlength{\parindent}{\ccwd}
4  // 来调整首行缩进的大小。这里的 \ccwd 是当前字号下一个中文汉字的宽度。
5
6
7

```

行间距

```

1
2  \usepackage{setspace}
3  \onehalfspacing
4
5

```

段间距

```

1
2  \addtolength{\parskip}{.4em}
3  // 相对于原来的加0.4 如果要减小就填负数。
4
5

```

The space between paragraphs, sections, subsections, etc. is determined automatically by LaTeX. If you want to customize the default paragraph spacing, it can be achieved with the following command in the preamble of your document:

```
1 \parskip 7.2pt
```

Additional space between two lines of the same paragraph or within a table is specified with the

```
1 \[length]
```

If you want to add space at the beginning of the document, without anything else written before, then you may use

```
1 { \vspace*[length] }
```

表格

1. 第一种最普通的表格形式

```
1 \begin{tabular}{|l|c|r|}
2 \hline
3 操作系统& 发行版& 编辑器\\
4 \hline
5 Windows & MikTeX & TexMakerX \\
6 \hline
7 Unix/Linux & teTeX & Kile \\
8 \hline
9 Mac OS & MacTeX & TeXShop \\
10 \hline
11 通用& TeX Live & TeXworks \\
12 \hline
13 \end{tabular}
14
15 \begin{tabular}{p{7em} p{23em} } % 用于指定列的宽度。
16 \caption{示例表格} % 用于给表格加标题。
17
```

- * 其中 `{|l|c|r|}` 分别指定每一列的对齐方式。竖线代表竖格线，没有就代表没有格线
- * `hline` 代表横格线
- * `&` 用于列分隔

操作系统	发行版	编辑器
Windows	MikTeX	TexMakerX
Unix/Linux	teTeX	Kile
Mac OS	MacTeX	TeXShop
通用	TeX Live	TeXworks

2. 交替明暗的表格。三线式表格，可以指定每一个格子的颜色。

```
1 \begin{tabular}{>{\sf }lll} %
2 \toprule
3 标签 & 描述 & \\
4 \midrule
```

```

5 <b> & \kai 定义粗体文本\\
6 \rowcolor{mygray} % 下一行的颜色
7 <em> & \kai 定义着重文字\\
8 <i> & \kai 定义斜体文字\\
9 \rowcolor{mygray}
10 <small> & \kai 定义小号文字\\
11 <strong> & \kai 定义大号文字\\
12 \rowcolor{mygray}
13 <sub> & \kai 定义下标 \\
14 <sup> & \kai 定义上标 \\
15 \rowcolor{mygray}
16 \bottomrule
17 \end{tabular}
18

```

标签	描述
	定义粗体文本
	定义着重文字
<i>	定义斜体文字
<small>	定义小号文字
	定义大号文字
<sub>	定义下标
<sup>	定义上标

3. 为列指定颜色。

```

1 \begin{tabular}{>{\columncolor{mypink}\sf }lll@{}}
2 \toprule
3 \rowcolor{white}
4 & \multicolumn{2}{c}{\bf Specific Heats} \\
5 \cmidrule{2-3}
6 \rowcolor{white}
7 & $c$ (J/kg$\cdots$K) & $C$ (J/mol$\cdots$K) \\
8 \midrule
9 Aluminum & 900 & 24.3 \\
10 Copper & 385 & 24.4 \\
11 Gold & 130 & 25.6 \\
12 Steel/Iron & 450 & 25.0 \\
13 \bottomrule
14 \end{tabular}
15

```

	Specific Heats	
	c (J/kg·K)	C (J/mol·K)
Aluminum	900	24.3
Copper	385	24.4
Gold	130	25.6
Steel/Iron	450	25.0

	Specific Heats	
	c (J/kg·K)	C (J/mol·K)
Aluminum	900	24.3
Copper	385	24.4
Gold	130	25.6
Steel/Iron	450	25.0
Lead	130	26.8
Water	4190	75.4
Ice (−10 °C)	2100	38

LONDON					Price
Sydney	OG4G	Thu Oct 10	Mon Oct 21 or 28	11 or 18 days	999GBP
		Thu Oct 17	Mon Oct 21 or 28	4 or 11 days	999GBP
	OG7A	Sun Oct 13	Mon Oct 21 or 28	8 or 15 days	999GBP
		Sun Oct 20	Mon Oct 28	8 days	999GBP

其他的表格可以参照 2016Penta.tex

数学公式

数学公式有行内公式和独立公式两种；

```

1 行内公式就直接使用两个美元符号夹起来就好啦： $ ... $
2
3 独立公式使用 \[ ... \]
4
5 若要对公式进行编号就使用
6 \begin{equation}
7 ...
8 \end{equation}
9
```

根式 & 分式

```

1 根式 \sqrt{}
2 分式 \frac{}{}
3
```

定界符

```

1 // 小括号
2 \[ \Biggl(\biggl(\Bigl(\bigl((x)\bigr)\Bigr)\biggr)\Biggr) \]
3
4 // 中括号
5 \[ \Biggl[\biggl[\Bigl[\bigl[[x]\bigr]\Bigr]\biggr]\Biggr] \]
6
7 // 大括号
8 \[ \Biggl\{\biggl\{\Bigl\{\bigl\{\{x\}\bigr\}\Bigr\}\biggr\}\Biggr\} \]
9
10 // 尖括号
11 \[ \Biggl\langle\biggl\langle\Bigl\langle\bigl\langle[x]\bigr\rangle\Bigr\rangle\biggr\rangle\Biggr\rangle x
12 \[ \Biggl\rangle\biggl\rangle\Bigl\rangle\bigl\rangle[x]\bigr\rangle\Bigr\rangle\biggr\rangle\Biggr\rangle \]
13
14 \[ \Biggl\lvert\biggl\lvert\Bigl\lvert\bigl\lvert[x]\bigr\rvert\Bigr\rvert\biggr\rvert\Biggr\rvert x
15 \[ \Biggl\rvert\biggl\rvert\Bigl\rvert\bigl\rvert[x]\bigr\rvert\Bigr\rvert\biggr\rvert\Biggr\rvert \]
16
17 \[ \Biggl\lvert\biggl\lvert\Bigl\lvert\bigl\lvert[x]\bigr\rvert\Bigr\rvert\biggr\rvert\Biggr\rvert x
18 \[ \Biggl\rvert\biggl\rvert\Bigl\rvert\bigl\rvert[x]\bigr\rvert\Bigr\rvert\biggr\rvert\Biggr\rvert \]
19
20
```

$$\begin{aligned} & \left(\left(\left(\left((x) \right) \right) \right) \right) \\ & \left[\left[\left[[x] \right] \right] \right] \\ & \left\{ \left\{ \left\{ \{x\} \right\} \right\} \right\} \\ & \langle \langle \langle \langle x \rangle \rangle \rangle \rangle \\ & \| \| \| \| x \| \| \| \| \\ & ||| ||| ||| x ||| ||| ||| \end{aligned}$$

省略号

```

1 // 横版省略号
2 \[ x_1,x_2,\cdots ,x_n\quad 1,2,\cdots ,n\quad
3
4 // 竖版省略号
5 \vdots\quad \ddots \]
6
7 % 对应的 \cdot 是点乘, \dot 是在符号上方加点。
8
9
```

$$x_1, x_2, \dots, x_n \quad 1, 2, \dots, n \quad \vdots \quad \ddots$$

连续求和 & 求积

```

1 \sum_{i=1}^n
2
3 %如果是行内公式, 这样的上下标是在右侧的, 如果是独立公式, 是在上下的。
4 %如果要改变位置, 可以使用 \sum\limits_{i=1}^n 和 \sum\nolimits_{i=1}^n
5 % 前者用于行内公式, 后者用于独立公式。
6
```

矩阵

```

1
2 \[ \begin{pmatrix} a&b\\c&d \end{pmatrix} \quad
3
4 \begin{bmatrix} a&b\\c&d \end{bmatrix} \quad
5
6 \begin{Bmatrix} a&b\\c&d \end{Bmatrix} \quad
7
8 \begin{vmatrix} a&b\\c&d \end{vmatrix} \quad
```

```

9
10 \begin{Vmatrix} a&b\\c&d \end{Vmatrix} \]
11
12

```

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad \left\{ \begin{matrix} a & b \\ c & d \end{matrix} \right\} \quad \left| \begin{matrix} a & b \\ c & d \end{matrix} \right| \quad \left\| \begin{matrix} a & b \\ c & d \end{matrix} \right\|$$

多行公式

```

1 // 公式对齐版本
2 \[\begin{aligned}
3 x &= {}& a+b+c+{} \\
4 &d+e+f+g \\
5 \end{aligned}\]
6

```

$$x = a + b + c + \\ d + e + f + g$$

公式组

```

1 // 公式对齐版本
2 \begin{gather}
3 a = b+c+d \\
4 x = y+z \\
5 \end{gather}
6 \begin{align}
7 a &= b+c+d \\
8 x &= y+z \\
9 \end{align}
10

```

$$a = b + c + d \tag{1.1}$$

$$x = y + z \tag{1.2}$$

$$a = b + c + d \tag{1.3}$$

$$x = y + z \tag{1.4}$$

分段函数

```

1 // 用Cases来实现分段
2 \[ y= \begin{cases}
3 -x, \quad x \leq 0 \\
4 x, \quad x > 0 \\
5 \end{cases} \]
6
7
8

```

$$y = \begin{cases} -x, & x \leq 0 \\ x, & x > 0 \end{cases}$$

代码排版

排版代码主要使用的包是: listings

If you just want to write code within your document the package provides the `lstlisting` environment:

```
1 \begin{lstlisting}
2 Put your code here.
3 \end{lstlisting}
```

Another possibility, that is very useful if you created a program on several files and you are still editing it, is to import the code from the source itself. This way, if you modify the source, you just have to recompile the LaTeX code and your document will be updated. The command is:

```
1 \lstinputlisting{source_filename.py}
2
3 \lstinputlisting[language=Python]{source_filename.py}
4
5 \lstinputlisting[language=Python, firstline=37, lastline=45]{source_filename.py}
6
```

Support Language It supports the following programming languages:

ABAP, ACSL, Ada, Algol, Ant, Assembler, Awk, bash, Basic, C#, C++, C, Caml, Clean, Cobol, Comal, csh, Delphi, Eiffel, Elan, erlang, Euphoria, Fortran, GCL, Gnuplot, Haskell, HTML, IDL, inform, Java, JVMIS, ksh, Lisp, Logo, Lua, make, Mathematica, Matlab, Mercury, MetaPost, Miranda, Mizar, ML, Modelica, Modula-2, MuPAD, NASTRAN, Oberon-2, Objective C, OCL, Octave, Oz, Pascal, Perl, PHP, PL/I, Plasm, POV, Prolog, Promela, Python, R, Reduce, Rexx, RSL, Ruby, S, SAS, Scilab, sh, SHELXL, Simula, SQL, tcl, TeX, VBScript, Verilog, VHDL, VRML, XML, XSLT.

Settings 一些设置参数。

```
1 backgroundcolor=\color{white}, % choose the background color; you must add \usepackage{color} or \
2 usepackage{xcolor}; should come as last argument
3 basicstyle=\footnotesize, % the size of the fonts that are used for the code
4 breakatwhitespace=false, % sets if automatic breaks should only happen at whitespace
5 breaklines=true, % sets automatic line breaking
6 captionpos=b, % sets the caption-position to bottom
7 commentstyle=\color{mygreen}, % comment style
8 deletekeywords={...}, % if you want to delete keywords from the given language
9 extendedchars=true, % lets you use non-ASCII characters; for 8-bits encodings only, does not
10 work with UTF-8
11 frame=single, % adds a frame around the code
12 keepspaces=true, % keeps spaces in text, useful for keeping indentation of code (possibly
13 needs columns=flexible)
14 keywordstyle=\color{blue}, % keyword style
15 language=Octave, % the language of the code
```

```

14 morekeywords={*,...},           % if you want to add more keywords to the set
15 numbers=left,                   % where to put the line-numbers; possible values are (none, left, right)
16 numbersep=5pt,                  % how far the line-numbers are from the code
17 numberstyle=\tiny\color{mygray}, % the style that is used for the line-numbers
18 rulecolor=\color{black},         % if not set, the frame-color may be changed on line-breaks within not-
    black text (e.g. comments (green here))
19 showspaces=false,               % show spaces everywhere adding particular underscores; it overrides '
    showstringspaces'
20 showstringspaces=false,         % underline spaces within strings only
21 showtabs=false,                 % show tabs within strings adding particular underscores
22 stepnumber=2,                   % the step between two line-numbers. If it's 1, each line will be
    numbered
23 stringstyle=\color{mymauve},    % string literal style
24 tabsize=2,                       % sets default tabsize to 2 spaces
25 title=\lstname                  % show the filename of files included with \lstinputlisting; also try
    caption instead of title

```

更多详细信息查看: <https://en.wikibooks.org/wiki/LaTeX/SourceCodeListings>

文档: <http://mirror.lzu.edu.cn/CTAN/macros/latex/contrib/listings/listings.pdf>

定制章节样式

使用 `titlesec` 包来定制属于自己的样式。

```

1 \usepackage[explicit]{titlesec}
2
3
4 \definecolor{mybluei}{RGB}{20, 0, 100}
5 \definecolor{myblueii}{RGB}{0, 0, 100}
6 \usepackage{rotating, graphicx}
7
8 \usepackage{tcolorbox}
9 \newcommand\ChapterFont{\rmfamily\selectfont\huge}
10 \newcommand\SectionFont{\bfseries\rmfamily\selectfont\Large}
11
12 %chapter
13 \titleformat{\chapter}[block]
14 {\normalfont\ChapterFont\color{myblueii}}
15 {\tcbset{colframe=mybluei, boxrule=0.8pt, left=0pt, right=0pt, top=0pt, bottom=0pt}
16 \raisebox{-0.48\height}{\rotatebox{90}{\tcbbox[boxsep=4pt, colback= white ]{\color{mybluei}\Large $L^A T_E X$
17 }}}
18 \hskip 0.25em
19 \mbox{\tcbbox[ boxsep=12pt, colback=mybluei, tcbox raise = -35pt]{\color{white}\bfseries\fontsize{70}{70}\
20 selectfont\thechapter}}}
21 {0.5em}
22 {\#1\vskip0.6ex\endgraf\titlerule[1ex]}
23 []
24
25 %section
26 \titleformat{\section}
27 {\normalfont\small\sffamily\SectionFont\color{myblueii}}
28 {\colorbox{mybluei}{%
29 \parbox[c][28pt][c]{40pt}{%
30 \centering\textcolor{white}{\SectionFont\Large\rmfamily\thesection}%
31 }%
32 }%
33 {1em}
34 {\#1}
35 [\vspace{-0.755\baselineskip}%
36 \color{myblueii}\hspace*{\dimexpr40pt+2\fbboxsep\relax}%
37 \rule{\dimexpr\textwidth-40pt-2\fbboxsep\relax}{1pt}%

```



```

37 ]
38
39 % subsection
40 \titleformat{\subsection}
41 {\normalfont\small\sffamily\SectionFont\color{myblueii}}
42 {\colorbox{mybluei}{%
43   \parbox[c][16pt][c]{50pt}{%
44     \centering\textcolor{white}{\SectionFont\Large\rmfamily\thesubsection}%
45   }}%
46 }%
47 }
48 {1em}
49 {#1}
50 [\vspace{-0.755\baselineskip}%
51 \color{myblueii}\hspace*{\dimexpr40pt+2\fbboxsep\relax}%
52 \rule{\dimexpr\textwidth-40pt-2\fbboxsep\relax}{0pt}%
53 ]
54
55 具体的用法事实上我也不是很清楚。从上面的代码中我们可以窥探一些用法。
56

```

具体的用法事实上我也不是很清楚。从上面的代码中我们可以窥探一些用法。

`colorbox` 这个是用来画一个颜色块。颜色块内可以放文字或表格。之前在做表格的时候我们使用过这个元素。

基本语法：

```
1 \colorbox{\color{blue}} {Text}
```

两个参数，分别是颜色、内容。

在使用这里的内容的时候我们也可以窥探一些其他东西。这里就是 Latx 中很重要的盒子。

盒子

所谓盒子就是一部分文本，`tex` 把它当作一个整体，如同单个字符一样处理！

实际上，`tex` 内部格式化的方式也是如此，单个字符构成一个盒子，然后把它放到水平的行盒子里面，在单词间插入橡皮长度。行盒子竖直堆积，构成段落盒子，在行之间也要插入橡皮长度，然后把它们放到页面主体盒子中，连同页眉和页脚盒子构成页面盒子！

在 `latex` 里面有三种类型的盒子，LR 盒子，段落盒子，标尺盒子。

- LR 盒子是由水平的从左到右的有序材料组成。
- 段落盒子是竖直堆积的行组成。
- 标尺盒子是一个用黑色填充的实心矩形。通常用来画水平或竖直线！

LR 盒子：

```

1 \mbox{文本} 和 \makebox[宽度][位置]{文本}
2 \fbox{文本} 和 \framebox[宽度][位置]{文本}

```

左边两条命令生成一个 LR 盒子，其宽度恰好是在 `{ }` 中给出的文本的宽度！`fbox` 有外框！

右面命令宽度是由可省的长度参数来预先确定的，另一个可省参数位置定义文本在盒子的位置，如果不给出的，文本时居中排列的，位置参数可以取如下值：`l r s`

文本的自然宽度要比所定义的宽度大的时候，就会出现突出！

`makebox` 命令很有用，可在 `picture` 环境中生成一个居中的或者左右对齐的文本，也可以使部分文本重叠！如 `makebox[0pt][1]{/}S`，结果是一条斜线穿过 `S`。长度定义必须含有单位。

```
1 \newsavebox{\boxname}
```

初始化之后，使用命令

```
1 \sbox{\boxname}{文本}或者 \savebox[宽度][位置]{文本}
```

就会把文本的内容保存起来以备后用。使用时

```
1 \usebox{\boxname} 插入保存的文本。
```

LR 盒子的内容也可以用如下环境保存

```
1 \begin{lrbox}{\boxname} 文本 \end{lrbox}
```

LR 盒子的竖直移位

```
1 \raisebox{上移量}[高度][深度]{文本}
```

子段盒子和小页：

生成给定宽度的盒子，其中的文本行如通常的段落模式一样彼此堆积，命令如下

```
1 \parbox[位置]{宽度}{文本}或者 \begin{minipage}[位置]{宽度} 文本\end{minipage}
```

可省参数位置可取值：**b t**，默认是居中排列。

竖直摆放的问题，对齐两个小页，需要加入 **mbox** 的空行

具有指定高度的段落盒子

命令是 **tex** 命令的扩展。

标尺盒子：

```
1 \rule[提示]{宽度}{高度}
```

也可以生成一个宽度为零的标尺盒子，这样得到一个不可见的具有给定高度的盒子，如此的构造称为一个支撑，如此可以强迫所在的水平盒子具有不同于其所包含内容需要的高度和深度！

嵌套盒子：

盒子的样式参数，对于有框盒子 **fboxrule** 确定方框线的粗细 **fboxsep** 设置方框与被包围文本之间的距离大小。