排版软件 LaTeX 简介

师维学

南京大学数学系

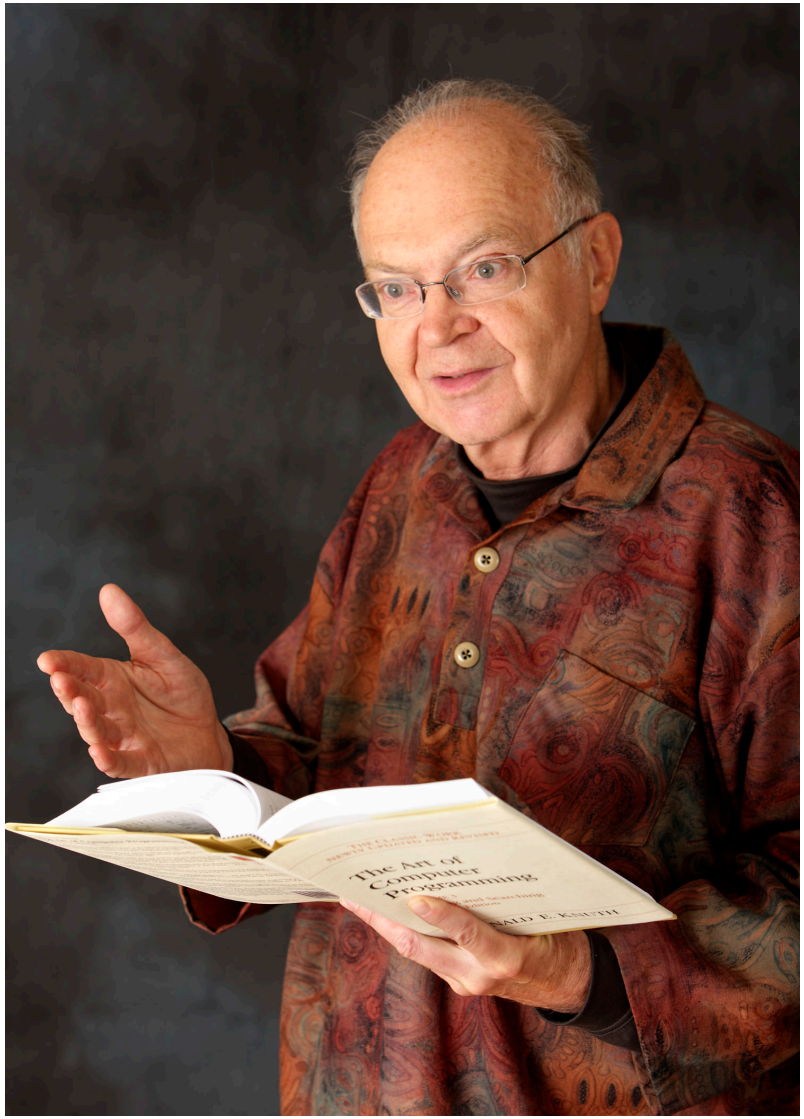
November 19, 2018

TEX 和他的作者

因为 LATEX 是建立在 TEX 基础上的软件，所以要介绍 LATEX, 就必须首先介绍 TEX, 它是诞生于 20 世纪 70 年代末到 80 年代初的一款计算机排版软件，作者是美国 Stanford 大学的高德纳(Donald Ervin Knuth) 教授.

高德纳最早开始自行编写 TEX 的原因是当时十分粗糙的排版水平已经影响到他的巨著《计算机程序设计艺术》（The Art of Computer Programming）的印刷质量。他以典型的黑客思维模式，最终决定自行编写一个排版软件：TEX。他原本以为他只需要半年时间，在 1978 年下半年就能完成，但最终他用了超过十年时间，直到 1989 年 TEX 才最终停止修改。

TEX 和他的作者



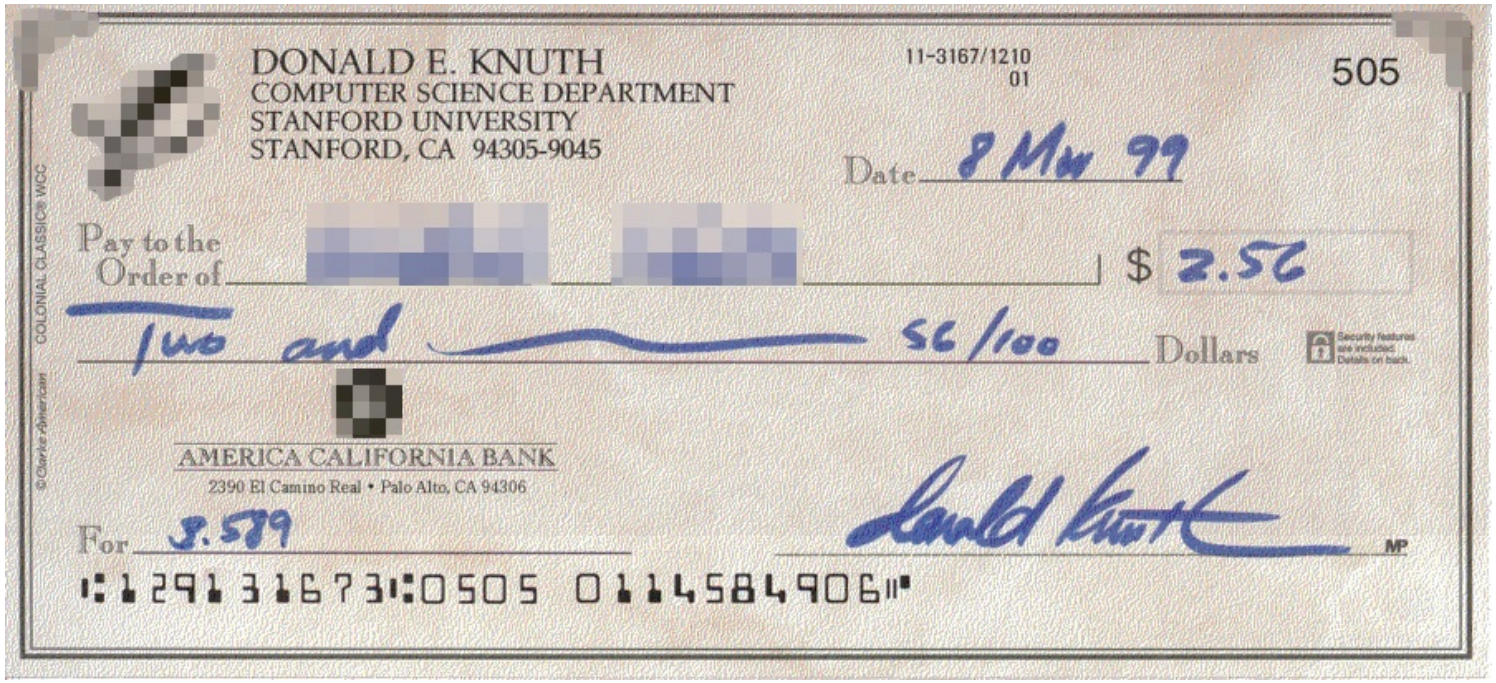
Donald Ervin Knuth

TEX 和他的作者

TEX 功能强大、几近无懈可击，其版本号非自然数列，也非年份，而是不断逼近的圆周率(最新版本为3.1415926)—这等于宣告产品接近完美，已经不可能作大的改进了。高德纳为此设置了悬赏奖金：谁找出 TEX 里的一个 bug, 就付给其 2.56 美元，找出第二个 5.12 美元，第三个 10.24 美元……依此累加。算法大师不可能不明白指数增长的可怕性（传说中的国王就是玩指数游戏输掉了江山），然而直到今天，他也未能为此付出多少钱，而真正发现漏洞的人在获得支票后往往不愿将其兑现。高德纳曾表示“最后一次升级是（于我过世后）将版本数改为 π，那时任何余下的漏洞将被看作程序的功能。”

TEX 允许自由的再发布及修改，但禁止任何修改版本以 TEX 或任何其他相似的名字命名。

TEX 和他的作者

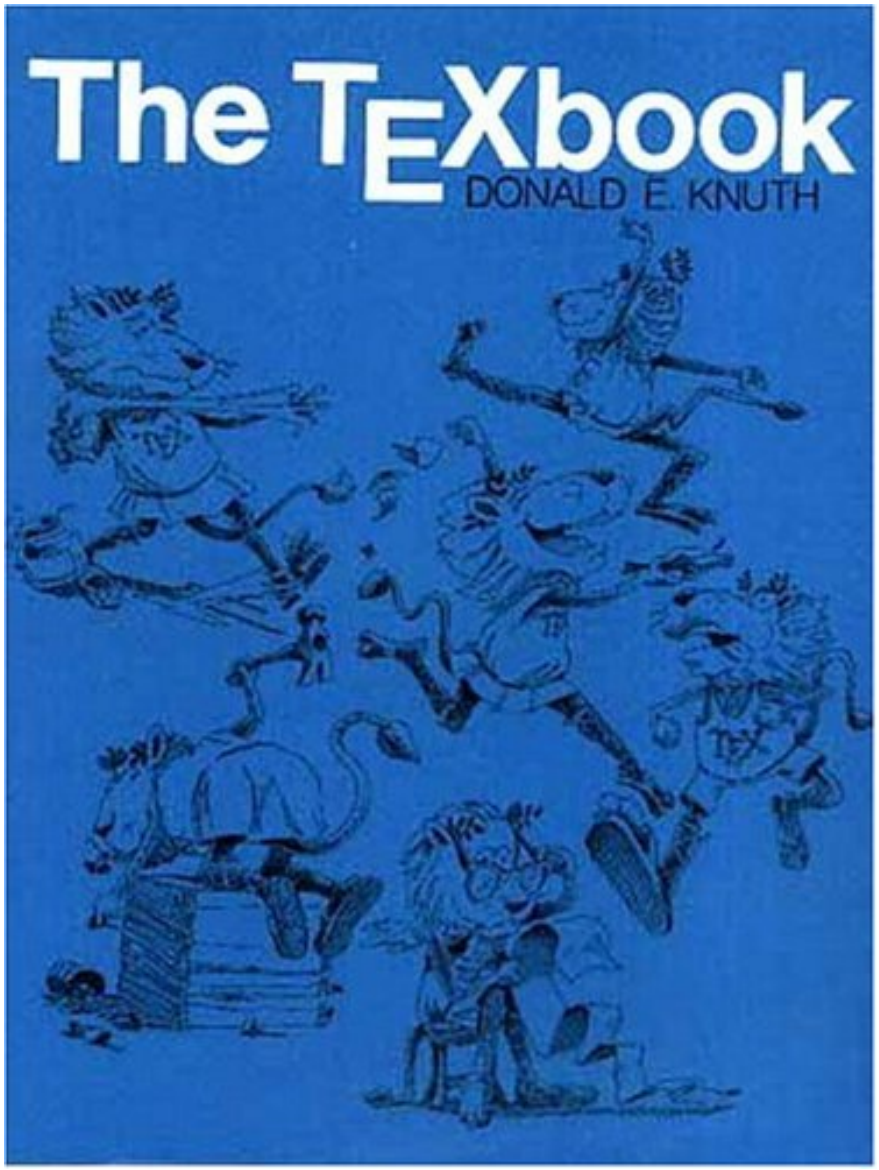

DONALD E. KNUTH
COMPUTER SCIENCE DEPARTMENT
STANFORD UNIVERSITY
STANFORD, CA 94305-9045
11-3167/1210
01
505
Date 8 Mar 99
Pay to the Order of
$ 2.56
Two and 56/100 Dollars
AMERICA CALIFORNIA BANK
2390 El Camino Real • Palo Alto, CA 94306
For 3.589
MP
COLONIAL CLASSIC® WCC

TEX 和他的作者

TEX 这个词的标准发音为 $\tau\epsilon\chi$，其中 χ 相当于普通话“赫”字的声母，或者苏格兰语“loch”一词中“ch”的发音 X 其实是希腊字母χ。音译“泰赫”。在英语和法语中实际通常读作 tεk，音译“泰克”。TEX 这个词来自希腊文中的 $\tau\varepsilon\chi\nu\eta$（TEXNH），希腊文意为“艺术”和“制造”，也是英语中technical（技术）的词源。书写时，三个字母都是大写，字母E应当低于其他两个字母。而不支持下标的系统则只能这样书写：“TeX”。

TEX 的用户喜欢创造一些和 TEX 有关的词汇，例如 TeXnician (与英语单词 technician，技工的发音相近，意为 TEX 用户)，TeXhacker (TEX 程序员，TEX 黑客) 和 TeXnique (与英语单词 technique，技巧的发音相近，意为TeX的使用技巧) 等。有人还给他起了个好听的中文名字“特科(可)爱”.

TEX 和他的作者

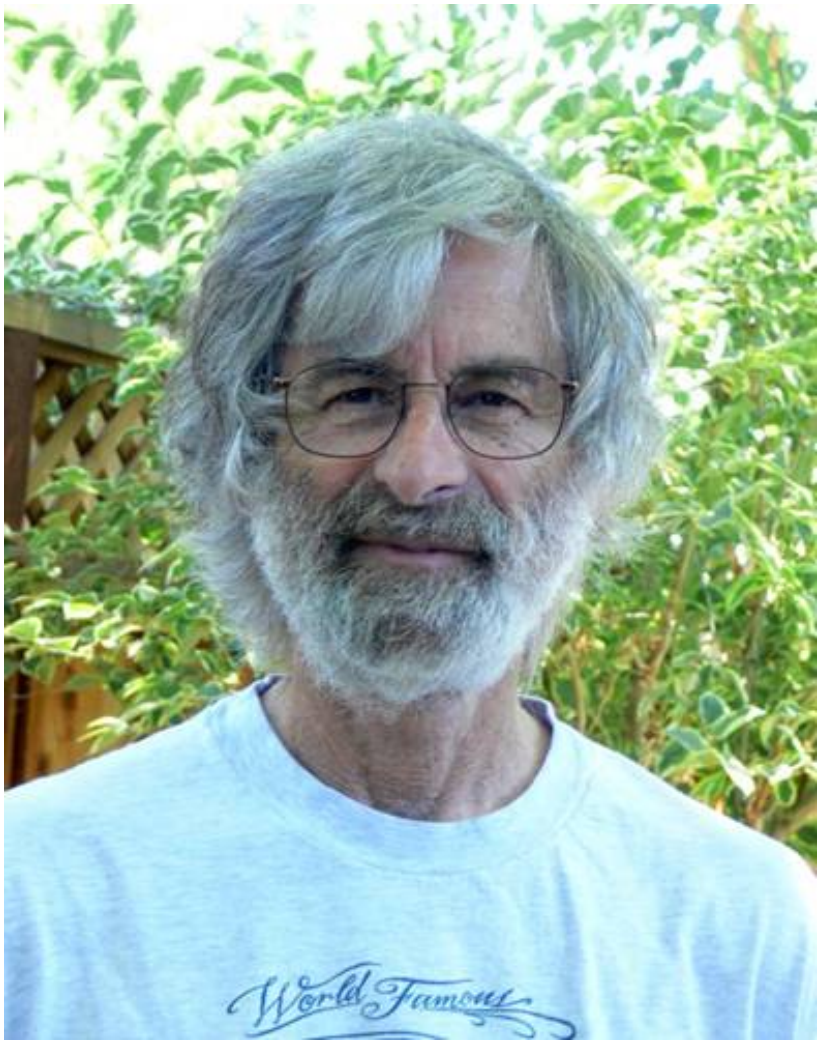


最基本的 TEX 程序只是由一些很原始的命令组成，它们可以完成简单的排版操作和程序设计功能。然而，TEX 也允许用这些原始命令定义一些更复杂的高级命令。这样就可以利用低级的块结构，形成一个用户界面相当友好的环境。

Knuth 设计了一个名叫Plain TEX 的基本格式，以与低层次的原始 TEX 呼应。这种格式是用 TEX 处理文本时相当基本的部分，以致于我们有时都分不清到底哪条指令是真正的处理程序 TEX 的原始命令，哪条是Plain TEX 格式的。大多数声称只使用 TEX 的人，实际上指的是只用Plain TEX。

Leslie Lamport 开发的 LATEX 是当今世界上最流行和使用最为广泛的 TEX 格式。它构筑在Plain TEX 的基础之上，并加进了很多的功能以使得使用者可以更为方便的利用 TEX 的强大功能。使用 LATEX 基本上不需要使用者自己设计命令和宏等，因为 LATEX 已经替你做好了。因此，即使使用者并不是很了解 TEX，也可以在短短的时间内生成高质量的文档。对于生成复杂的数学公式，LATEX 表现的更为出色。

Leslie Lamport 是分布式系统领域的先锋人物，他于 1941 年在纽约出生，1960 年毕业于麻省理工学院数学专业，1963 年获得布兰迪斯大学数学硕士学位，1972 年获得布兰迪斯大学数学博士学位。2001 年Leslie Lamport 进入位于加利福尼亚的微软研究院，任高级研究员，从事分布式计算机系统理论研究，2013 年荣获图灵奖，该奖项是计算机界最高荣誉奖项，有“计算机界的诺贝尔奖”之称，专门奖励那些对计算机事业作出重要贡献的个人。



Leslie Lamport

LaTeX 自从八十年代初问世以来，也在不断的发展。最初的正式版本为 2.09，在经过几年的发展之后，许多新的功能，机制被引入到 LaTeX 中。在享受这些新功能带来的便利的同时，它所伴随的副作用也开始显现，这就是不兼容性。标准的 LaTeX2.09，引入了“新字体选择框架”(NFSS) 的 LaTeX, SLiTEX, $\mathcal{AMS}$-LaTeX等等，相互之间并不兼容。这给使用者和维护者都带来很大的麻烦。为结束这中糟糕的状况，Frank Mittelbach 等人成立了 LaTeX3 项目小组，目标是建立一个最优的，有效的，统一的，标准的命令集合。即得到 LaTeX 的一个新版本 3。这是一个长期目标，向这个目标迈出第一步就是在 1994 年发布的 LaTeX 2ε。LaTeX 2ε采用了NFSS 作为标准，加入了很多新的功能，同时还兼容旧的 LaTeX2.09。LaTeX 2ε每 6 个月更新一次，修正发现的错误并加入一些新的功能。在 LaTeX3 最终完成之前，LaTeX 2ε将是标准的 LaTeX 版本。

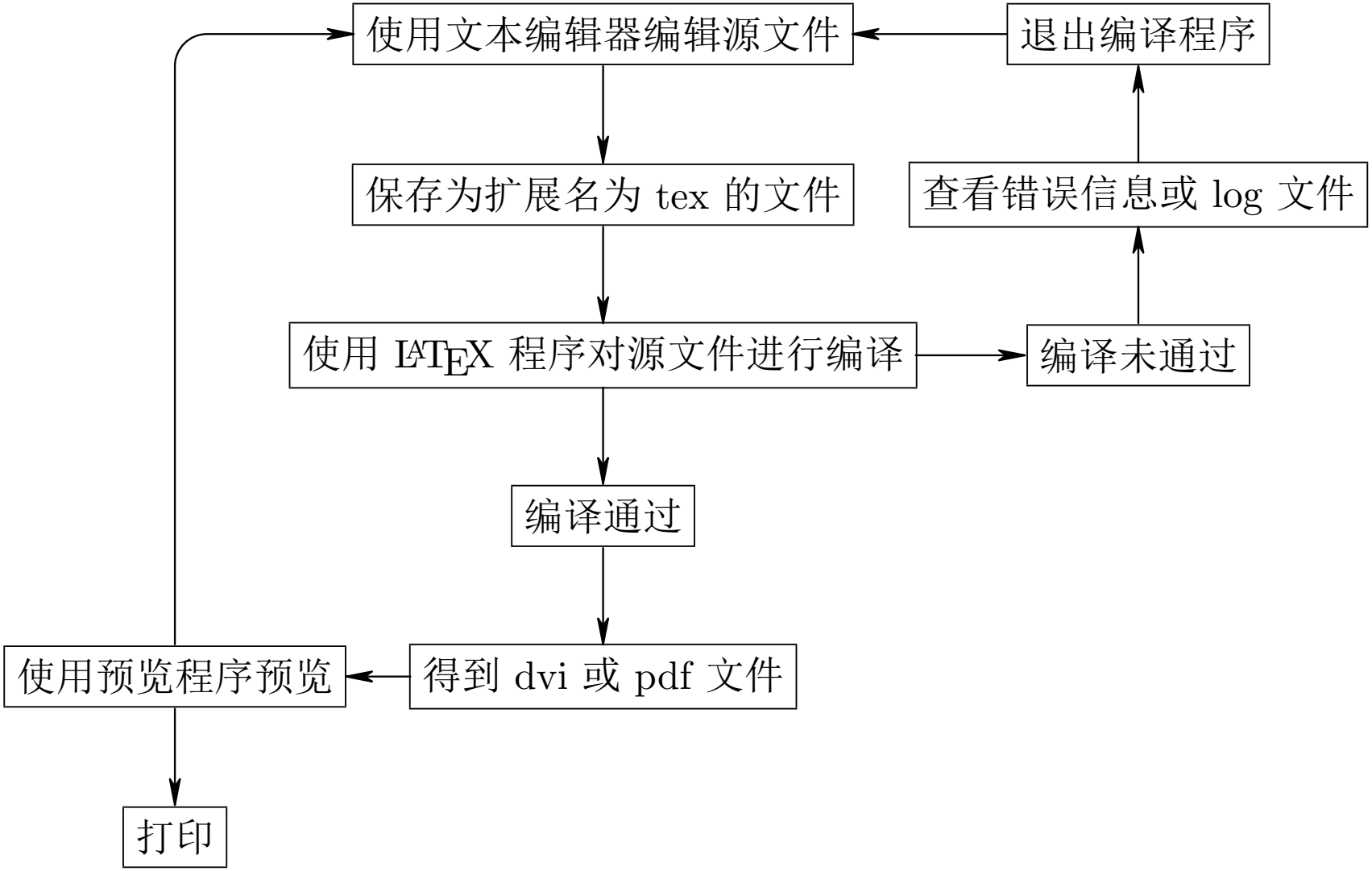
TEX 或 LATEX 的优势

- 高质量的输出
- 超常的稳定性
- TEX 是可编程的
- 高度的灵活性
- 简单方便, TEX 文档是 ASCII 码的文本文件
- 目前为止，TEX 几乎在所有的计算机操作系统平台上得到实现
- TEX 是免费软件，它的源程序也是免费的
- 超级技术支持
- TEX 是一种乐趣

TEX软件的资源

LATEX 的最新版本和有关资源可以在互联网上免费下载：
http://www.ctan.org/
它在世界各地有一百多个镜像站点，在中国大陆的有：
http://mirror.bjtu.edu.cn/CTAN/
http://mirror.hust.edu.cn/CTAN/
http://mirror.neu.edu.cn/CTAN/
http://mirrors.ustc.edu.cn/CTAN/
在 Windows 操作系统下, 推荐使用 Miktex, 可在
https://miktex.org/
下载。之前大家广泛使用的 CTeX 套装已经被 Miktex 包含。

使用 LaTeX 排版的流程



如何准备你的源文件

- 选择一款文本编辑器: WinEdt TeXWorks TeXshop Emacs 等可供选择.
  其中 WinEdt 比较受欢迎, 但是它不是免费的。下载地址: http://www.winedt.com/ 注册授权购买地址: https://item.taobao.com/item.htm?id=551105790596
- 键盘上字符的分类:
  - 普通字符: 将原样显示在排版结果中的字符, 例如, 键盘上的英文字母, 数字, 还有如下的标点符号:

    . : ; , ? ! ‘ ’ ( ) [ ] - / * @
  - 控制符: 只在 LaTeX 命令中使用的字符

    `# $ % & ~ _ ^ \ { }`
  - 数学字符:
    + = | < > 主要在数学环境中使用。
  - 不可见字符: 空格符 换行符
    在 LaTeX 中一串空格符等同于一个空格符, 一个换行符也等同于一个空格符, 而一个空行等同于一个段落的结束。

控制列和环境

控制列: 以反斜杠 \ 开头的一列字母叫做一个控制列 (Control sequence) 也叫“命令”或“指令”, 控制列中不可以包含空格、数字和标点, \ 后面的第一个空格、数字或标点意味着控制列的结尾。控制列是用来指示计算机作排版默认字体和格式以外的事.

环境: LaTeX 中事先定义好的一些排版格式用“环境”来界定。环境的用法是:

\begin{环境名}...\end{环境名}

例如: 居中排版的环境是“center”

```
\begin{center} Department of Mathematics,\\
Nanjing University,\\ Nanjing
\end{center}
```

输出的是:

Department of Mathematics,
Nanjing University,
Nanjing

源文件的结构和例子

源文件的例子:

```
\documentclass[12pt]{article}
\usepackage{amsmath}
\title{\LaTeX}
\date{}
\begin{document}
  \maketitle
  \LaTeX{} is a document preparation system for the \TeX{}
  typesetting program. It offers programmable desktop
  publishing features and extensive facilities for
```

源文件的结构和例子

```
  automating most aspects of typesetting and desktop
  publishing, including numbering and cross-referencing,
  tables and figures, page layout, bibliographies, and
  much more. \LaTeX{} was originally written in 1984 by
  Leslie Lamport and has become the dominant method for
  using \TeX; few people write in plain \TeX{} anymore.
  The current version is \LaTeX2e.

    % This is a comment, not shown in final output.
    % The following shows typesetting power of LaTeX:
    \begin{align}
      E_0 &= mc^2                                    \\
      E &= \frac{mc^2}{\sqrt{1-\frac{v^2}{c^2}}}
    \end{align}
\end{document}
```

用 LaTeX 程序编译后得到结果是:

LaTeX

LaTeX is a document preparation system for the TeX typesetting program. It offers programmable desktop publishing features and extensive facilities for automating most aspects of typesetting and desktop publishing, including numbering and cross-referencing, tables and figures, page layout, bibliographies, and much more. LaTeX was originally written in 1984 by Leslie Lamport and has become the dominant method for using TeX; few people write in plain TeX anymore. The current version is LaTeX 2_ε.

$$E_0 = mc^2 \tag{1}$$

$$E = \frac{mc^2}{\sqrt{1 - \frac{v^2}{c^2}}} \tag{2}$$

不要使用旧版 LaTeX 2.09

许多人常常从朋友、同学处拷贝一个源文件, 利用现成的导言区设置来编辑自己的文章, 当然这不失为一种捷径。要注意的是不要使用旧版的即 2.09 版源文件, 旧版源文件的第一行是:

```
\documentstyle[12pt]{article}
```

还有的源文件虽然第一行已经改为

```
\documentclass[12pt]{article}
```

但是在导言区仍然用着为旧版编写的宏包, 例如《中国科学杂志》A 辑的模板的前几行

```
\documentclass[twoside]{article}
%------------- Page Format -------
\usepackage{graphicx,amssymb,
            mathrsfs,amsmath,color,fancyhdr}
\input vatola.sty \input cyracc.def
\input psfig.sty
...
```

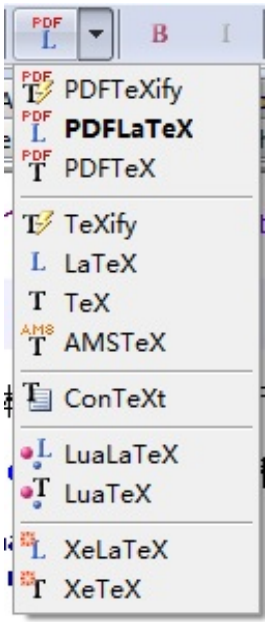
使用 WinEdt 运行 LaTeX

如果使用 WinEdt 作为你的文本编辑器, 请在编辑器新建文档并输入如下几行:

```
\documentclass{article}
\begin{document}
Hello, every one. Here is an example
of source file for \LaTeX.\par
\[\sin^2x+\cos^2x=1,\]
\[ 1+\tan^2x=\frac{1}{\cos^2x}\]
\end{document}
```

使用 WinEdt 运行 LaTeX

保存为 `myfile.tex`。WinEdt 的窗口上方的菜单如下:



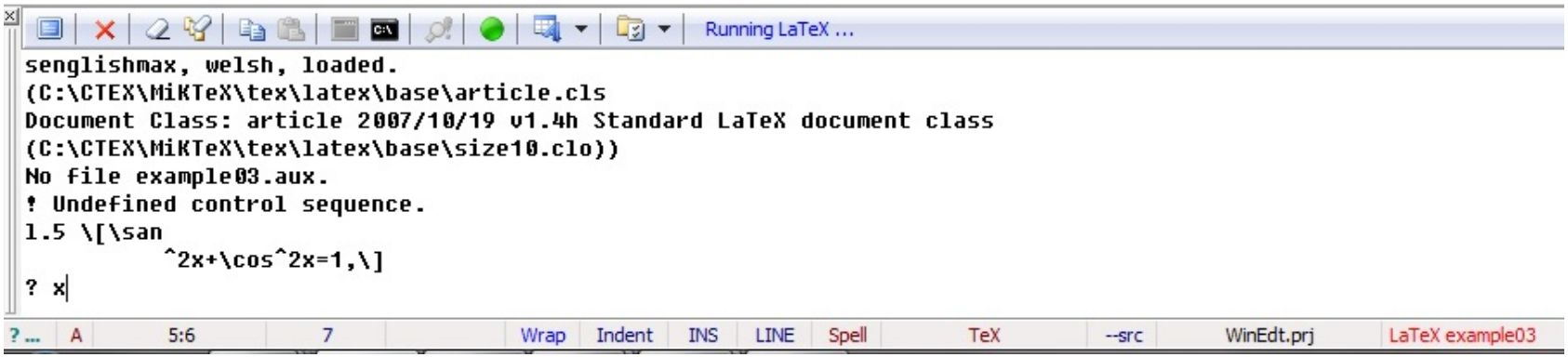
点击 PDFLaTeX 或 LaTeX 按钮编译。

编译后 adobe reader 或 dvi 按钮亮起:

点击 adobe reader 或 dvi 预览。

如果源文件含错误代码, 编译就会在第一个有错误代码的地方中断并给出错误信息。例如在 `myfile.tex` 中把第5行的 `\sin` 改为 `\san` 编译就会得到如下的反馈.

在光标处键入 x 按回车键即可退出编译状态.



一些基本的知识

1. 分段指令: `\par` 或者用空行。
2. 引号: 单引号‘ ’
   双引号‘‘ ’’ 例如:

   ```
   ‘Convention’ dictates that punctuation
   go inside quotes, like ‘‘this,’’ but I
   think it’s better to do ‘‘this’’.
   ```

   结果是:
   ‘Convention’ dictates that punctuation go inside quotes, like “this,” but I think it’s better to do “this”.
   在某些系统的键盘上, 可能缺少‘ 和 ’ 两个符号键, 或者两个键失效. 例如在安装双系统苹果电脑上运行 Windows 时‘ 键会失效. 这时可以使用与‘ 和 ’ 等效的命令 `\lq` 和 `\rq`.
3. 连字符 “-” 的使用: `2--6` 结果是 2–6; `dash---like this` 结果是 dash—like this.

键盘上特殊字符的输出

键盘上已经被 LaTeX 用作控制符的字符, 要在文章中印出时, 可以用以下的方法输出:

命令	\$	\&	\%	\#	_	\{	\}
输出	$	&	%	#	_	{	}

另外三个控制符 ~, ^, \, 一般只在仿键盘输出时才会用到。仿键盘输出, 将在后面介绍。

数学表达式

数学表达式分为两种：一种是混在行文的行中的数学式子，另一种是单列的，也就是独占一行的数学表达式。在源文件中，数学表达式要放在在数学环境中，即，行中的数学公式以 \(开始以 \) 结束，单列的数学表达式以 \[开头以 \] 结束。

数学表达式

```
The formula \(x-3y=7\) is easy to type.
```

结果是:

The formula $x-3y=7$ is easy to type.

在数学环境中空格被无视:

```
Does \(  x +  y  \) always equal \(y+x\)?
```

输出结果是:

Does $x+y$ always equal $y+x$?

数学表达式, 下标与上标

在数学环境中下标用命令 _ 生成, 上标用命令 ^ 生成。例如:

```
\(a_1 > x^{2n}/y^{2n}\)
```

$a_1 > x^{2n}/y^{2n}$

数学环境中的右单引号 ’ 生成“撇” ′ 例如:

```
This proves that \(x'<x''-y'_3<10x'''z\)
```

生成:

This proves that $x' < x'' - y'_3 < 10x'''z$

LaTeX 中 $...$ 和\(...\) 的作用是等同的, 使用 $...$ 可以减少敲击键盘的次数, 但是因为 $ 没有左右之分, 在编辑长数学公式, 出错时不容易查找, 所以建议只在编写短数学公式时使用 $...$ 而在编辑长数学公式时用 \(...\), 在编辑特长公式时你也可以使用 \begin{math}...\end{math} 来代替 \(...\)。

数学表达式, 注释

例如:

```
It is obvious from the definition of $+$ in
$R$ that \(\alpha+\beta\subset\alpha+\gamma\)
```

结果是:

It is obvious from the definition of $+$ in R that $\alpha+\beta\subset\alpha+\gamma$

LaTeX 将无视源文件中的 % 以及 % 所在行位于 % 后面的所有字符，因此可以利用 % 为你的源文件加注释。% 的另一个作用是表示一行的结束, 并与下一行不留空格地对接。

数学表达式, 注释

例如:

```
It is obvious from the defini% 这里可写注释。
tion of $+$ in $R$ that \(\alpha+\beta
\subset\alpha+\gamma\)
```

结果仍然是:

It is obvious from the definition of $+$ in R that $\alpha+\beta\subset\alpha+\gamma$

数学表达式, 单列

当数学表达式较长, 难以放在行中, 或为了强调或为了给公式编号以便后面引用时, 就需要将公式单列一行。此时可用 displaymath 环境和 equation 环境, 就是利用 \[...\] 或 \begin{equation}...\end{equation} 来得到单列的数学公式。例如:

```
Here is an example of an unnumbered displayed equation:
\[x'+y^2 = z_i^2 \]
and here is the same equation numbered:
\begin{equation}
x'+y^2 = z_i^2
\end{equation}
```

结果是:

Here is an example of an unnumbered displayed equation:

$$x' + y^2 = z_i^2$$

and here is the same equation numbered:

$$x' + y^2 = z_i^2 \tag{1}$$

`\[...\]` 也可用 `\begin{displaymath}...\end{displaymath}` 代替。

在数学公式中插入文字

有时我们需要在数学公式(特别是单列的数学公式)中插入文字, 例如:

$$ax + by + cz = d, \text{ where } a^2 + b^2 + c^2 \neq 0$$

其代码为:

```
\[ax+by+cz=d,\mbox{ where } a^2+b^2+c^2\ne 0\]
```

即在数学公式中要临时插入文字要使用命令 `\mbox{...}` 或使用 `amsmath` 格式包时用 `\text{...}`.

文档类

LaTeX 的源文件都是以 `\documentclass[选项]{类}` 开头的。花括号中要填入要调用的“文档类”的文件名, 即扩展名为 cls 的文件名, 类文件是事先设计好的文章整体格式的宏包。LaTeX 系统的标准文档类有 `article`, `book`, `report`, `letter`, `slides` 等, 其中 `article` 最为常用。

`article`	用于论文等不太长的文档
`book`	用于排版书籍
`report`	用于排版报告
`letter`	用于排版西文信件
`slides`	用于排版讲演材料

可选择的选项

文档类 `article` 的选项可以是

`11pt`, `12pt` 用以指定字体大小;

`twoside` 适合两面印刷;

`twocolumn` 双列排版;

`a4paper`, `a5paper`, `b5paper` 用以指定纸张尺寸, 默认尺寸为 `letterpaper` 8.5in×11in;

`draft` 为草稿模式输出, 对于溢出边界的行以黑方块标出;

`titlepage` 将生成单独一页的标题页, 并将摘要也单独放在一页;

`leqno` 使单列的数学公式编号在左侧显示;

`fleqn` 使单列的数学公式左对齐。

选项支持多选, 选项之间用逗号隔开, 不留空格。

文档类预设的文章结构

文档类 article 预设的文章结构:

\title	标题	\author	作者
\date	日期	\maketitle	生成标题(页)
\section	节标题	\subsection	小节标题
\subsubsection	次小节标题	\paragraph	段标题
\subparagraph	小段标题	\part	部分的标题
\appendix	附录标题	\tableofcontents	生成目录
\listoffigures	插图目录	\listoftables	表格目录

还有生成摘要和参考文献的环境:

\begin{abstract}...\end{abstract} 生成摘要;

\begin{thebibliography}{99}...\end{thebibliography} 生成参考文献, 参考文献条目用 \bibitem{标签} 列出。

其它的文档类

`amsart`	美国数学会期刊论文
`elsarticle`	Elsevier 出版的期刊论文
`beamer`	一款很流行的讲演材料排版模板
`kluwer`	Kluwer 科学出版公司期刊模板
`ctexart`	中文论文排版

LaTeX 中的长度单位

在 LaTeX 中可以使用的长度单位有: in, pt, mm, cm 等等，下面列出的是这些长度单位以及他们之间的换算关系:

`cm`	厘米
`mm`	毫米
`in`	英寸(1in = 2.54cm)
`pt`	点(1in = 72.27pt)
`bp`	大点(1in = 72bp)
`pc`	pica(1pc = 12pt)
`dd`	didôt 点(1157dd = 1238pt)
`cc`	cicero(1cc = 12dd)
`em`	与字体相关的尺寸, 表示大写字母 M 的宽度;
`ex`	与字体相关的尺寸, 表示小写字母 x 的高度

如何生成空白, 横向

当我们需要在文章中留出空白时, 因为 LaTeX 总是把源文件中, 若干个空格等同于一个空格, 所以要在文章中插入空白, 必须通过使用一些命令来实现。

~ 产生一个英文小写字母的空白, 并禁止在此处断行;

\␣ 产生一个英文小写字母的空白, 在普通行文和数学环境中都可以使用;

\, 产生半个英文小写字母的空白, 在普通行文和数学环境中都可以使用;

\! 产生一个负空白, 即把空白缩小半个英文字母, 只能在数学环境中使用;

\quad 产生一个当前字体大小的空白, 例如, 当前字体大小是 10pt 则 \quad 产生一个 10pt 的空白;

\qquad 产生一个是 \quad 二倍大小的空白;

如何生成空白, 横向

	正常空隙||
~	空 隙, | |
\␣	空 隙, | |
\,	空 隙, ||
\!	‖
\quad	空　隙, |　|
\qquad	空　　隙, |　　|

如何生成空白, 横向

横向任意指定长度空白使用命令: `\hspace{长度}` 或 `\hspace*{长度}` 没有 * 的形式将在换行点忽略这个空白指令, 而带 * 的形式则在任何情况下都插入空白。

横向弹性空白 `\hfill` 将插入空白使得该行被填满, 但是行的开头, 结尾无效, 要在行头, 行尾使用时, 可以使用 `\hspace*{\fill}` 的形式。例如:

```
left\hfill right \\
left\hfill center \hfill right\\
\hspace*{\fill} center \hspace*{\fill}
```

得到的是:

left right
left center right
center

如何生成空白, 竖向

增加段落之间的距离的命令有:

`\smallskip`, `\medskip`, `\bigskip`

竖向指定长度的空白命令是: `\vspace{长度}` 和 `\vspace*{长度}`

竖向弹性空白命令: `\vfill` 和 `\vspace*{\fill}`

他们的用法与横向空白对应的命令类似, 不过必须在竖向模式下有效。

断行与分页

另起一行的命令是: \\[长度] 或 \newline; 强制断行的命令是 \linebreak。\\[长度] 中的 [长度] 用于调整断行后与下一行的距离，缺省时与下一行的距离是默认行距。

例如:

```
\LaTeX{} is a document preparation system
for the \TeX{} typesetting program. It offers
programmable desktop\\
publishing features and extensive facilities for
automating most aspects of typesetting and desktop
publishing, including numbering and\linebreak
cross-referencing, tables and figures,
page layout, bibliographies, and much more.
```

输出是:

LaTeX is a document preparation system for the TeX typesetting program. It offers programmable desktop
publishing features and extensive facilities for automating most aspects of typesetting and desktop publishing, including numbering and cross-referencing, tables and figures, page layout, bibliographies, and much more.

另起一页的命令是: `\newpage`; 强制断页的命令是 `\pagebreak` 他们的用法和区别是: `\newpage` 使该页在该命令所在的点中断而另起一页; 而`\pagebreak` 使当前页在该命令所在行排满后中断另起一页。

字体

LaTeX 排版中普通英文默认字体是 Computer Modern Roman 字体, 数学公式的默认字体是意大利斜体。

有时为了强调某一段文字, 需要临时改变字体。LaTeX 把字体按整体风格(font family)、形状(shape)、字体的粗细(series)来定义。

英文常见的字体风格:

- Roman family; `\textrm{Roman family}`
- Sans serif family; `\textsf{Sans serif family}`
- `Typewriter family. \texttt{Typewriter family}`

字体

英文常见的字体形状:

- 正体 Upright shape; `\textup{Upright shape}`
- 意大利斜体 *Italic shape*; `\textit{Italic shape}`
- 斜体 *Slanted shape*; `\textsl{Slanted shape}`
- 小型大写 Small capital shape. `\textsc{Small capital shape}`

英文常见的字体粗细:

- 一般 Medium series; `\textmd{Medium series}`
- 粗体 **Boldface series**. `\textbf{Boldface series}`

上述字体命令可以用在数学环境中, 但是对数学公式的字体无效。

例如:

```
\[\mathcal U=\{U\, |\, U \textrm{ is
open in } X \textrm{ and } x\in U\}\]
```

$$\mathcal{U} = \{U \,|\, U \text{ is open in } X \text{ and } x \in U\}$$

```
\textbf{\[\mathcal U=\{U\, |\, U \textrm{ is
open in } X \textrm{ and } x\in U\}\]}
```

$$\mathcal{U} = \{U \,|\, U \textbf{ is open in } X \textbf{ and } x \in U\}$$

改变字体的尺寸

LaTeX 的文档类已经规定了全局字体的标准尺寸。局部改变字体尺寸可以使用如下命令:

`\tiny`	smallest	`\large`	large
`\scriptsize`	very small	`\Large`	larger
`\footnotesize`	smaller	`\LARGE`	even large
`\small`	small	`\huge`	still larger
`\normalsize`	normal	`\Huge`	largest

这组命令的有效范围由 `{...}` 来界定。

数学公式的排版

上标与下标:

x^{2y}	`x^{2y}`	x^{y^2}	`x^{y^2}`	x_1^y	`x^y_1`
x_{2y}	`x_{2y}`	x_{y_1}	`x_{y_1}`	x_1^y	`x_1^y`

分式: 在文字行中的分式:

Multiplying by $n/2$ gives $(m+n)/n$.

```
Multiplying by $n/2$ gives $(m+n)/n$.
```

数学公式的排版

单列数学公式中的分式:

$$x = \frac{y + z/2}{y^2 + 1}$$

```
\[ x= \frac{y+z/2}{y^2+1} \]
```

$$\frac{x + y}{1 + \frac{y}{z+1}}$$

```
\[\frac{x+y}{1+\frac{y}{z+1}}\]
```

在文字行中使用 `\frac`:

Multiplying by $\frac{n}{2}$ gives $\frac{m+n}{n}$.

```
Multiplying by $\frac{n}{2}$ gives $\frac{m+n}{n}$.
```

在文字行中使用 `\displaystyle\frac`:

Multiplying by $\displaystyle\frac{n}{2}$ gives $\displaystyle\frac{m+n}{n}$.

```
Multiplying by $\displaystyle\frac{n}{2}$
gives $\displaystyle\frac{m+n}{n}$.
```

如果使用格式包 `amsmath` 则 `\displaystyle\frac` 可以缩写为 `\dfrac`; 而且还可以在单列公式中使用 `\tfrac` 得到文字行中数学公式的分数。

数学公式的排版

根式:

$\sqrt{x+y}$ and $\sqrt[n]{2}$, `\(\sqrt{x+y}\) and \(\sqrt[n]{2}\)`

省略号:

A low ellipsis: $x_1,\ldots,x_n$.

```
A low ellipsis: $x_1,\ldots,x_n$.
```

A centered ellipsis: $a_1+\cdots+a_n$.

```
A centered ellipsis: $a_1+\cdots+a_n$.
```

竖向的省略号: $\vdots$ `\vdots`; 斜向的省略号: $\ddots$ `\ddots`

数学符号, 希腊字母:

α	\alpha	β	\beta	γ	\gamma	δ	\delta	ϵ	\epsilon
ε	\varepsilon	ζ	\zeta	η	\eta	θ	\theta	ϑ	\vartheta
ι	\iota	κ	\kappa	λ	\lambda	μ	\mu	ν	\nu
ξ	\xi	o	o	π	\pi	ϖ	\varpi	ρ	\rho
ϱ	\varrho	σ	\sigma	ς	\varsigma	τ	\tau	υ	\upsilon
ϕ	\phi	φ	\varphi	χ	\chi	ψ	\psi	ω	\omega
Γ	\Gamma	Δ	\Delta	Θ	\Theta	Λ	\Lambda	Ξ	\Xi
Π	\Pi	Σ	\Sigma	Υ	\Upsilon	Φ	\Phi	Ψ	\Psi
Ω	\Omega								

数学公式的排版

在数学符号上加斜划线表示否定:

If $x \not< y$ then $x \not\leq y-1$.

```
If $x\not<y$ then \(x\not\leq y-1 \).
```

文中和单列数学公式的不同:

Here's how they look when displayed

$$\sum_{i=1}^{n} x_i = \int_0^1 f(x)\,dx$$

and in text: $\sum_{i=1}^{n} x_i = \int_0^1 f(x)\,dx$.

数学公式的排版

```
Here's how they look when displayed
\[\sum_{i=1}^n x_i=\int_0^1 f(x)\, dx\]
and in text:
\(\sum_{i=1}^n x_i=\int_0^1 f(x)\, dx\).
```

如果要在文中的数学公式的上下标位于运算符号的正上下方, 可以使用 `\limits` 来实现:

In text we can change the formula to $\sum_{i=1}^{n} x_i = \int_0^1 f(x)\,dx$.

```
In text we can change the formula to
\( \sum\limits_{i=1}^n x_i=\int_0^1 f(x)\, dx\).
```

数学公式的排版

数学环境下的列表: array 环境
例子:

```
\(\begin{array}{clcr}
a+b+c & uv     & x-y & 27     \\
a+b   & u+v    & z   & 134    \\
a     & 3u+vw  & xyz & 2.978
\end{array}  \)
```

输出为:

$$\begin{array}{clcr} a+b+c & uv & x-y & 27 \\ a+b & u+v & z & 134 \\ a & 3u+vw & xyz & 2.978 \end{array}$$

列表与其他元素的对齐:
例子:

```
\[ x-\begin{array}{c}
a_1     \\ \vdots \\ a_n
\end{array}
- \begin{array}[t]{cl}
    u-v     & 13 \\
    u+v     & \begin{array}[b]{r}
                12     \\ -345
                \end{array}
\end{array}                          \]
```

输出为:

$$x - \boxed{\begin{array}{c} a_1 \\ \vdots \\ a_n \end{array}} - \boxed{\begin{array}{ll} u - v & 13 \\ & \boxed{\begin{array}{r} 12 \\ -345 \end{array}} \\ u + v & \end{array}}$$

定界符:

(	(	)	)	↑	\uparrow
[	[	]	]	↓	\downarrow
{	\{	}	\}	↕	\updownarrow
⌊	\lfloor	⌋	\rfloor	⇑	\Uparrow
⌈	\lceil	⌉	\rceil	⇓	\Downarrow
⟨	\langle	⟩	\rangle	⇕	\Updownarrow
/	/	\	\backslash		
|	|	‖	\\|		

让定界符的大小合适:

例子1:

```
\[  \left(  \begin{array}{c}
            \left| \begin{array}{cc}
                    x_{11} & x_{12} \\
                    x_{21} & x_{22}
                    \end{array} \right| \\
            y \\ z
            \end{array} \right)       \]
```

$$\left(\begin{array}{c} \left| \begin{array}{cc} x_{11} & x_{12} \\ x_{21} & x_{22} \end{array} \right| \\ y \\ z \end{array} \right)$$

让定界符的大小合适:

例子2:

```
\[ \vec{x}+\vec{y}+\vec{z} = \left(  \begin{array}{c}
                                a \\ b
                               \end{array} \right[       \]
```

$$\vec{x}+\vec{y}+\vec{z} = \left(\begin{array}{c} a \\ b \end{array} \right[$$

让定界符的大小合适:

例子3:

```
\[ x = \left\{  \begin{array}{ll}
                y    & \mbox{if $y>0$} \\
                z+y & \mbox{otherwise}
                                \end{array} \right.       \]
```

$$x = \left\{ \begin{array}{ll} y & \text{if } y > 0 \\ z + y & \text{otherwise} \end{array} \right.$$

数学公式的排版

多行数学公式的排版:

当公式太长需要排成多行时, 可以使用环境 eqnarray 或 eqnarray*, 这个环境很像 array 三列的效果:

```
\begin{eqnarray}
  x & = & 17y \\
  y & > & a+b+c+d+e+f+g+h+i+j+ \nonumber  \\
    &   & k+l+m+n+o+p+\int f(x)
\end{eqnarray}
```

$$
\begin{aligned}
x &= 17y && (2)\\
y &> a+b+c+d+e+f+g+h+i+j+ \\
&\quad k+l+m+n+o+p+\int f(x) && (3)
\end{aligned}
$$

使用 * 形式时将没有编号:

```
\begin{eqnarray*}
  x & = & 17y \\
  y & > & a+b+c+d+e+f+g+h+i+j+  \\
    &   & k+l+m+n+o+p+\int f(x)
\end{eqnarray*}
```

$$\begin{aligned} x &= 17y \\ y &> a+b+c+d+e+f+g+h+i+j+ \\ &\quad k+l+m+n+o+p+\int f(x) \end{aligned}$$

当 + 或 − 位于一个公式(或子公式)的开头时, TEX 自动把它看成一个一元运算符, 这时它与后面跟着的字符之间没有空白, 例如 `$+x$` 产生 $+x$; `$\sum -x_i$` 产生 $\sum -x_i$.

在长公式因换行而使二元运算符 + 或 − 处于行首时, 需要告诉 TEX 它们是二元运算符, 方法是在 + 或 - 的前面加上 `\mbox{}`。

```
\begin{eqnarray*}
   y & = & a+b+c+d+e+f+g+h+i+j  \\
     &   & \mbox{}+k+l+m+n+o+p
\end{eqnarray*}
```

$$\begin{aligned} y &= a+b+c+d+e+f+g+h+i+j \\ &\quad +k+l+m+n+o+p \end{aligned}$$

在使用 eqnarray 或 eqnarray* 时, 可以在其中使用命令 \lefteqn, 它的作用是让 TEX 把其作用下的公式的长度看作零, 从而使 eqnarray 或 eqnarray* 中最左侧的列的宽度适当缩小。

```
\begin{eqnarray*}
\lefteqn{w+x+y+z=  } \\
     &   & a+b+c+d+e+f+g+h+i+j+  \\
     &   & k+l+m+n+o+p
\end{eqnarray*}
```

$$\begin{aligned} w+x+y+z= & \\ & a+b+c+d+e+f+g+h+i+j+ \\ & k+l+m+n+o+p \end{aligned}$$

数学公式的排版, 叠加

- 上画线 `\overline`:

  ```
  You can have nested overlining:
  \(\overline{\overline{x}^2 +1} \).
  ```

  You can have nested overlining: $\overline{\overline{x}^2 + 1}$

- 下画线 `\underline` 在正文和数学环境中都可以用:

  ```
  \underline{The} value is $\underline{3x}$.
  ```

  $\underline{\text{The}}$ value is $\underline{3x}$.

- 横向括号(上下) `\overbrace` 和`\underbrace`:
  ```
  \(\overbrace{a + \underbrace{b+c} + d}\)
  ```

$$\overbrace{a + \underbrace{b + c} + d}$$

数学公式的排版, 叠加

- 横向括号(上下) \overbrace 和\underbrace:
  在单列数学公式中还可以加入标签:

```
\[\underbrace{a + \overbrace{b
+\cdots +y}^{24} +z}_{26} \]
```

$$\underbrace{a+\overbrace{b+\cdots+y}^{24}+z}_{26}$$

数学公式的排版, 叠加

- 数学环境中的重音符号:

$\hat{a}$	\hat{a}	$\acute{a}$	\acute{a}	$\bar{a}$	\bar{a}	$\dot{a}$	\dot{a}
$\check{a}$	\check{a}	$\grave{a}$	\grave{a}	$\vec{a}$	\vec{a}	$\ddot{a}$	\ddot{a}
$\breve{a}$	\breve{a}	$\tilde{a}$	\tilde{a}				

- 跨几个字符之上的箭头等:

$\widehat{abc}$	\widehat{abc}	$\widetilde{abc}$	\widetilde{abc}
$\overrightarrow{abc}$	\overrightarrow{abc}	$\overleftarrow{abc}$	\overleftarrow{abc}

数学公式的排版, 叠加

- 数学符号的堆叠 \stackrel:

```
\(A \stackrel{a'}{\rightarrow}
B \stackrel{b'}{\rightarrow} C \)
```

$A \stackrel{a'}{\rightarrow} B \stackrel{b'}{\rightarrow} C$

```
\(\vec{x} \stackrel{\mathrm{def}}{=\!=}
(x_1,\ldots,x_n) \)
```

$\vec{x} \stackrel{\mathrm{def}}{=\!=} (x_1, \ldots, x_n)$

数学公式的排版, 空白

数学公式中的空白可以使用如下命令:

| |	`\,`	thin space	| |	`\:`	medium space
| |	`\␣`	interword space	‖	`\!`	negative space
| |	`\;`	thick space			

`\,` 和 `\!` 常常用来对数学表达式做精细微调:

$\sqrt{2}\,x$	`\sqrt{2} \, x`
$\sqrt{2}x$	`\sqrt{2} x`
$n/\log n$	`n / \! \log n`
$n/\,\log n$	`n / \log n`

数学公式的排版, 空白

$$\iint z\,dx\,dy$$

```
\int\!\!\int z\, dx\, dy
```

$$\int \int zdxdy$$

```
\int \int z dx dy
```

$$\frac{1}{2}$$

```
\frac{\, 1\, }{2}
```

$$\frac{1}{2}$$

```
\frac{ 1 }{2}
```

数学公式的排版, 字体

在数学环境中改变字体的命令如下:

$italic + 2^{ft} \Psi \log[\psi]$

```
$\mathit{italic + 2^{ft} \Psi \log[\psi]}$
```

$\mathrm{roman} + 2^{\mathrm{ft}} \Psi \log[\psi]$

```
$\mathrm{roman + 2^{ft} \Psi \log[\psi]}$
```

$\mathbf{bold} + \mathbf{2^{ft} \Psi} \log[\psi]$

```
$\mathbf{bold + 2^{ft} \Psi \log[\psi]}$
```

$\mathsf{sans\ serif} + 2^{\mathsf{ft}} \Psi \log[\psi]$

```
$\mathsf{sans\ serif + 2^{ft} \Psi \log[\psi]}$
```

$\mathtt{typewriter} + 2^{\mathtt{ft}} \Psi \log[\psi]$

```
$\mathtt{typerwriter + 2^{ft} \Psi \log[\psi]}$
```

$\mathcal{ABC}$

```
$\mathcal{ABC}$
```

数学公式的排版, 字体

在数学环境中默认字体是意大利斜体, 请注意默认字体和 `\mathit` 的不同:

Is $different$ any $\mathit{different}$ from dif^2e^2rnt?

```
Is $different$ any $\mathit{different}$
from $dif^2e^2rnt$?
```

从此例可以注意到, 不能使用 `$...$` 来生成斜体词来表示强调, 而应该使用 `\emph{...}` 或 `\textit{...}`。

如果要使数学公式中所有字符都变为黑体, 则要使用 `\boldmath` 和 `\unboldmath`:

$\boldsymbol{diff + 2^{ft} \Psi \log[\psi]}$

```
\boldmath $diff + 2^{ft} \Psi \log[\psi]$ \unboldmath
```

数学公式的排版, 字体

还要注意 \boldmath 不能在数学环境内使用。如果要在一个数学公式中, 只把其中一部分变为黑体, 则需要使用 \mbox:

$x + \boldsymbol{\nabla f} = 0$ \(x + \mbox{\boldmath ∇f} =0\)

如果调用 amsmath, 可以使用命令 \boldsymbol 使数学公式中一部分变为黑体:

$x + \boldsymbol{\nabla f} = 0$ \(x + \boldsymbol{\nabla f} =0\)

数学公式中字体大小的控制有如下四条命令:

```
\displaystyle      \textstyle
\scriptstyle       \scriptscriptstyle
```

定理或与之类似的文档结构

在科技排版中常常会有定理、定律、命题、猜想和推论等结构, 他们在文档中一般有固定的表达格式。LaTeX中用 `\newtheorem` 来定义一个类似于定理格式的环境:

```
\newtheorem{环境名}{定理显示名}[选项]
```

这行代码要放在源文件的导言区, 例如:

```
\newtheorem{thm}{Theorem}
```

在正文区输入:

```
\begin{thm}
This is an example of Theorem.
\end{thm}
```

例子另举.

参考文献的排版

在科技排版中一般还要在结尾列出参考文献。LaTeX中用环境 thebibliography 来排版参考文献:

```
\begin{thebibliography}{99}
\bibitem{a} this the first item of reference.
\bibitem{b} this the second item of reference.
\bibitem{c} this the third item of reference.
......
\bibitem{k} this the eleventh item of reference.
\end{thebibliography}
```

输出结果另举.

表格的排版

排表格的环境是 tabular.

环境 tabular 与环境 array 类似, 但是 array 只能在数学环境中使用, 而 tabular 可以在任何环境中使用.

tabular 的使用方法是:

\begin{tabular}{参数}...\end{tabular}, 其中参数可以是字母 l, c, r 这三个字母的总个数代表表格的列数, l 表示它所对应的列中的内容左对齐, c 所对应的列中的内容居中, r 所对应的列中的内容右对齐; 还可以加入 | 表示生成与表格同高竖线, 命令 \hline 生成一条与表格等宽的横线. 例如:

```
\begin{tabular}{lccr}
left & center & center & right \\
a    & b      & c      & d     \\
1    & 2      & 3      & 4
\end{tabular}
```

表格的排版

输出结果是:

left	center	center	right
a	b	c	d
1	2	3	4

而

```
\begin{tabular}{|l|c|c|r|}
\hline
left & center & center & right \\
\hline
a    & b      & c      & d     \\
\hline
1    & 2      & 3      & 4     \\
\hline
\end{tabular}
```

输出结果是:

left	center	center	right
a	b	c	d
1	2	3	4

有时需要将几个单元格合并, 例如:

```
\begin{tabular}{|l|c|c|r|}
\hline
left & center & center & right \\
\hline
a    & b      & c      & d     \\
\cline{1-1}\cline{3-4}
1    & 2      & 3      & 4     \\
\hline
\end{tabular}
```

结果是:

left	center	center	right
a	b	c	d
1	2	3	4

如果要在上面的表格中去掉第二列的 2, 让 b 居中, 做法是使用命令 `\raisebox{1.5ex}[0ex]{b}`, 即

```
\begin{tabular}{|l|c|c|r|}
\hline
left & center & center & right \\
\hline
a    &        & c      & d     \\
\cline{1-1}\cline{3-4}
1    & \raisebox{1.5ex}[0ex]{b}  & 3      & 4     \\
\hline
\end{tabular}
```

输出是:

left	center	center	right
a	b	c	d
1		3	4

`\raisebox` 的一般格式是 `\raisebox{提升度}[上沿线扩展度][基准线向下扩展度]{文本}`。

表格的排版

再例如:

```
\begin{tabular}{|l|c|c|r|}
\hline
left &
\multicolumn{2}{c|}{ center } & right \\
\hline
a    & b      & c      & d     \\
\hline
1    & 2      & 3      & 4     \\
\hline
\end{tabular}
```

结果是:

left	center		right
a	b	c	d
1	2	3	4

表格的排版

表格的行高和列宽是根据内容自动确定的，例如一行的行高是由这一行的内容高度最大的那个单元格所需高度确定；一列的宽度是由这列中内容最宽的宽度确定。但是有时需要有空行或空列的表格，或需要指定行高，指定列宽。这时可以利用看不见的“支撑”，\strut 是一个与当前字体的高度相同的支撑命令。例如

```
\begin{tabular}{|l|c|c|r|}
\hline
\strut\hspace*{5ex} & \hspace*{5ex}
& \hspace*{5ex} & \hspace*{5ex} \\
\hline
\strut     &       &       &      \\
\hline
\strut     &       &       &      \\
\hline
\end{tabular}
```

给出空白表格：

标尺盒子也可以用于支撑。所谓标尺盒子是完全用黑色填充的矩形，其相应的命令是 \rule[提升度]{宽度}{高度} “提升度”是指矩形的下边与基准线差，例如

```
\rule[0ex]{1cm}{.5pt} example
\rule[1ex]{1cm}{.2cm} \rule[0ex]{1cm}{.2cm}
\rule[-1ex]{1cm}{.2cm} example \rule[0ex]{1cm}{.5pt}
```

给出

如果让矩形的宽度为零，则生成一个不可见的“支撑”，例如

表格的排版

```
\begin{tabular}{|l|c|c|r|}
\hline
left &
\multicolumn{2}{c|}{\hspace*{15pt}center\hspace*{15pt}}
 & right \\
\hline
\rule[-1.5ex]{0pt}{4ex}a    & \hspace*{9pt}b\hspace*{9pt}
     & c       & d      \\
\hline
1    & 2       & 3       & 4      \\
\hline
\end{tabular}
```

输出是:

left	center		right
a	b	c	d
1	2	3	4

交换图的排版

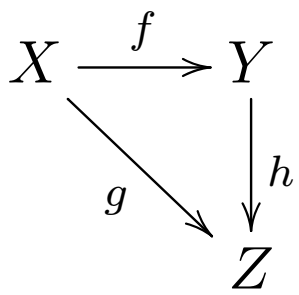
数学文章常常要排版交换图。这需要调用 xy-pic 格式包，即在导言区加入:

```
\usepackage{etex}
\usepackage[all]{xy}
```

然后在正文区排版交换图:

```
\xymatrix{X \ar^{f} [r]
\ar_{g} [dr] & Y \ar^{h} [d] \\
& Z}
```

得到:



几个调整版面的常用命令

- 调整纸张大小的命令:
  `\paperheight` 纸张的高度; `\paperwidth` 纸张的宽度.
- 调整版芯大小的命令:
  `\textheight` 版芯的高度; `\textwidth` 版芯的宽度.

可以在导言区用 `\setlength` 给上述的长度命令赋予新值，例如用:

```
\setlength{\textwidth}{12.5cm}
```

使文本行宽变为 12.5cm。

几个调整版面的常用命令

- 调整版芯位置的命令:
  `\hoffset` 左右平移版芯; `\voffset` 上下平移版芯. 例如用 `\hoffset=-1cm` 使版芯向左平移 1cm。
- 整篇文章调整行距的命令:
  `\renewcommand\baselinestretch{倍数}`.
- 去掉本页页码的命令:
  `\thispagestyle{empty}`.

LaTeX 出现以来, 出现了若干可以处理中文的版本, 在国内曾流行过的有中科院张波林开发的 CCT 系统, 这个系统处理中文需要先对中文的源文件进行预处理, 得到 LaTeX 可以处理的 tex 文件. 现在流行的是使用 Werner Lemberg 设计的 CJK 宏包. 这个宏包可以处理中、日、韩三国的文字而且不需要预处理. 它的使用方法是:

```
\documentclass{article}
\usepackage{CJK}
\begin{document}
\begin{CJK*}{GBK}{song}
\CJKindent
中文
\end{CJK*}
\end{document}
```

其中 song 表示使用宋体字.

可以使用的中文字体有:

fs 仿宋体; kai 楷体字; hei 黑体字; li 隶书体字; you 幼圆体字.

文中临时改变字体(例如改为仿宋体)的命令是:
\CJKfamily{GBK}{fs}.

另一使用中文的方法是使用 CTeX 的文档类:

```
\documentclass{ctexart}
\begin{document}
中文
\end{document}
```

或者调用 CTeX 格式包:

```
\documentclass{article}
\usepackage{ctex}
\begin{document}
中文
\end{document}
```

CTEX 文档类和 CTEX 格式包兼容了 CCT 系统的字体命令和字号命令:

- \songti 宋体.
- \heiti 黑体.
- \kaishu 楷书.
- \fangsong 仿宋.

字号:

\zihao{0}	初号	\zihao{-4}	小四号
\zihao{1}	一号	\zihao{5}	五号
\zihao{2}	二号	\zihao{-5}	小五号
\zihao{3}	三号	\zihao{6}	六号
\zihao{4}	四号	\zihao{7}	七号

使用 CJK 排版中英文混合或中文与数学公式混合的内容时, 汉字结束接着排英文或数学公式, 汉字与英文之间或汉字与数学公式之间没有空格, 例如:

使用 CJK 时, 汉字与英文字之间的空白

```
设 $E$为有界集，当 \(m_*E=m^*E\)
时，称 $E$为 Lebesgue可测.
```

得到的是:

设E为有界集, 当$m_*E = m^*E$时, 称E为Lebesgue可测.

这不符合中英文混合排版的习惯, 看起来也不美观, 所以在编辑源文件时需要在汉字和英文交界处插入空格命令:

```
设~$E$ 为有界集，当~\(m_*E=m^*E\)
时，称~$E$ 为~Lebesgue 可测.
```

得到的是:

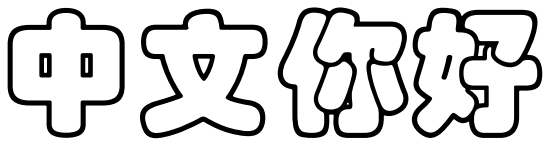
设 E 为有界集, 当 $m_*E = m^*E$ 时, 称 E 为 Lebesgue 可测.

XeLaTeX 简介

XeLaTeX 是近些年来出现的 LaTeX的编译系统. 它的最显著的优点是可以直接调用本机系统字体. 例如:

```
\documentclass{article}
\usepackage{xeCJK}
\setCJKfamilyfont{caiyun}{STCAIYUN.ttf}
\newcommand*{\caiyun}{\CJKfamily{caiyun}}
\input{ctex-xecjk-winfonts.def}
\begin{document}
中文
{\heiti 中文}
{\Huge\caiyun 中文 你好}
\end{document}
```

XeLaTeX 源文件默认编码是 UTF-8 所以源文件保存时注意选择 UTF-8 编码. 得到如下结果:

中文**中文** 

Happy TEXing

谢谢！