

# L<sup>A</sup>T<sub>E</sub>X简介

张伟平

中国科大 统计与金融系

2009年9月14日

# 目录

- 1 T<sub>E</sub>X介绍
- 2 T<sub>E</sub>X的宏包和扩展
- 3 环境集
- 4 L<sup>A</sup>T<sub>E</sub>X命令集
- 5 页面版式命令
- 6 计数器命令
- 7 目录表
- 8 交叉引用和索引
- 9 宏包
- 10 L<sup>A</sup>T<sub>E</sub>X中文化

# 框架

- 1 T<sub>E</sub>X介绍
- 2 T<sub>E</sub>X的宏包和扩展
- 3 环境集
- 4 L<sup>A</sup>T<sub>E</sub>X命令集
- 5 页面版式命令
- 6 计数器命令
- 7 目录表
- 8 交叉引用和索引
- 9 宏包
- 10 L<sup>A</sup>T<sub>E</sub>X中文化

# T<sub>E</sub>X介绍

T<sub>E</sub>X是一款功能强大的排版系统. 它是由Stanford University 的Donald E. Knuth (高德纳) 教授开发的.

◆ 1978年, Knuth教授在看到其巨著“ The Art of Computer Programming ” 的排版时, 对其中的数学公式排版非常不满意, 于是决定自己开发一款用于数学排版的软件, 因此才有了T<sub>E</sub>X这款软件. T<sub>E</sub>X是希腊词 T E X N H 的缩写, 其意指科技和艺术, 其也是英文technical 的词源.

◆ 1993年左右Knuth宣布不再对T<sub>E</sub>X进行更新.

◆ T<sub>E</sub>X目前的版本是3.1415926→ $\pi$ .



# T<sub>E</sub>X的特点

⇒ T<sub>E</sub>X是一款功能强大的排版系统

• T<sub>E</sub>X提供了300多条基本的排版命令

• 用户可以在基本命令的基础上定义自己的命令

• T<sub>E</sub>X是公认的数学排版最好的排版系统, 美国数学学会 (AMS) 鼓励数学家们使用T<sub>E</sub>X系统向它的期刊投稿. 世界上许多一流的出版社如Kluwer, Addison-Wesley, 牛津大学出版社等也利用T<sub>E</sub>X系统出版书籍和期刊.

• T<sub>E</sub>X是免费的!!!

• T<sub>E</sub>X系统以高质量的排版输出为目标, 其非常的稳定. 采用宏命令编程语言, 因此可以使用少数代码实现复杂的结果. 另外, T<sub>E</sub>X的文档为文本文档, 占用空间非常小.

## ⇒ T<sub>E</sub>X的缺点

- T<sub>E</sub>X的命令很多, 不便于记忆. 需要一本参考书, 要多用.
- T<sub>E</sub>X编译出现错误时不容易解决, 需要经验.
- T<sub>E</sub>X不是所见即所得系统.(尽管也有一些商业T<sub>E</sub>X软件是所见即所得)
- T<sub>E</sub>X的宏包书写有一定难度.

# 框架

- 1 T<sub>E</sub>X介绍
- 2 T<sub>E</sub>X的宏包和扩展
- 3 环境集
- 4 L<sup>A</sup>T<sub>E</sub>X命令集
- 5 页面版式命令
- 6 计数器命令
- 7 目录表
- 8 交叉引用和索引
- 9 宏包
- 10 L<sup>A</sup>T<sub>E</sub>X中文化

# T<sub>E</sub>X的宏包

T<sub>E</sub>X的基本命令直接使用起来是比较繁琐的, 因此有必要使用宏来重新定义一些简单的命令, 以实现背后复杂的基本命令.

- **Plain TeX** (执行命令: `tex`) Knuth设计了一个名叫Plain T<sub>E</sub>X的基本格式, 以与低层次的T<sub>E</sub>X互应. 现在已经很少有人使用. 美国数学学会开发了AMS - T<sub>E</sub>X宏包(`amsmath`), 用于数学公式的排版.

- **L<sup>A</sup>T<sub>E</sub>X** (执行命令: `latex`) L<sup>A</sup>T<sub>E</sub>X是现在使用最多的T<sub>E</sub>X宏包, 是由Leslie Lamport 和SRI International 在上世纪80年代早期开发的. 现在的版本是L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>. L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>可以加载`amsmath`宏包.

- **ConTeXt** (执行命令: `texexec`) 和L<sup>A</sup>T<sub>E</sub>X类似功能的宏包.

# TEX扩展

TEX系统诞生于上世纪70年代, 因此缺少现代软件的一些功能和特点. 下面这些程序通过改写TEX的引擎以增加TEX系统对一些新功能的支持:

- **PDFTeX** (执行命令: pdftex) TEX比pdf格式的诞生早了10余年, TEX系统的输出文件格式为.dvi (device independent). 而PDFTeX可以输出dvi 和pdf 两种格式. 在pdf 模式下, 输出的pdf 文档可以使用pdf 文档的各种功能: 比如超链接(hyperlinks), 书签(bookmarks), 等等,

- **XeTeX** (执行命令: xetex) TEX不仅早于现代所有主要的字体技术(比如TrueType, OpenType, PostScript Type 1等), 而且也早于现在操作系统已经包含了字体的做法. 因此, TEX使用自己定义的字体格式, 和别的不兼容. XeTeX 通过增强TEX的字体控制功能使其可以使用操作系统已有的各种字体以解决这个问题

# TeX扩展

• **LuaTeX** (执行命令: `luatex`) TeX使用其专属的,集成的,基于宏的语言. 许多用户抱怨这种语言比较难于掌握, LuaTeX 通过引入一个现代的解释语言, Lua, 来解决这个问题.

• **e-TeX** (执行命令: `etex`) e-TeX 为宏包开发者提供了很多有用的命令, 最终用户一般不会用到这些扩展的命令.

• **Omega** 这是一个几乎是完全重新写过的, 支持Unicode的TeX程序. Omega 有很多特征都超出了Knuth对TeX的原始设想. 它采用了与TeX不同的方法来处理字符, 使得可以处理非拉丁语系的文本. 也支持从右到左, 从上到下的排版方式.

# T<sub>E</sub>X发行版

TEX 在不同的硬件和操作系统上有不同的实现版本. 这就像C语言, 在不同的操作系统中有不同的编译系统, 例如Linux 下的gcc, Windows 下的Visual C++ 等. 有时, 一种操作系统里也会有好几种的T<sub>E</sub>X系统. 目前常见的有

✓ **teTEX:** Unix/Linux 下的T<sub>E</sub>X系统.

✓ **MiKTEX, fpTEX:** Windows 平台下的T<sub>E</sub>X系统.

✓ **TeXLive** 由国际T<sub>E</sub>X用户组织开发, 适用于不同的操作系统平台.

# TeX中文支持

## □ 中文预处理程序

✓ **CCT** 中科院张林波教授开发, 在文档格式方面非常符合中文的习惯.

✓ **TY** 华师大肖刚, 陈志杰等教授开发.

□ **CJK** 由德国W. Lemberg 开发, 可以同时处理中文, 日文和韩文. **CJK** 和TeX系统的结合更紧密, 但中文化不如**CCT, TY**.

目前处理中文的流行做法是: **LaTeX + CJK**.

CTEX 指的是CTEX 中文套装的简称, 是把MiKTeX 和一些常用的相关工具, 如GSview, WinEdt 等包装在一起制作的一个简易安装程序, 并对其中的中文支持部分进行了配置, 使得安装后马上就可以使用中文.

# L<sup>A</sup>T<sub>E</sub>X的排版过程

- ① 编写T<sub>E</sub>X源文件, 纯文本, 以`.tex`为扩展名.
  - 👉 可以使用任何文本编辑器.
  - 👉 **WinEdt, WinShell, EditPlus, Vi, Emacs** 等等.
- ② 用LaTeX 编译源文件, 生成.dvi 文件.
- ③ 屏幕打印或者, 转换成.ps 或.pdf 文件.

# TeX中的长度

- ◆ 由十进制小数 加长度单位 表示
- ◆ 常用的长度单位

mm 毫米	pt 点
cm 厘米	em 大写字母M 的宽度
in 英寸	ex 小写字母x 的高度

- ☛  $1\text{in}=2.54\text{cm}=72.27\text{pt}$
- ☛ em 和ex 与当前字体的尺寸有关.
- ◆ 弹性长度, 根据内容需要自动伸缩
- ☛ 正常值 plus 伸展值 minus 缩进值
- ☛ 实际长度可以超过正常值和伸展值之和, 但不能小于正常值和缩小值之差.

# 特殊字符

常见 $\LaTeX$ 扩充文字字符:

符号	\$	&	%	#	\	^	-
命令	\\$	\&	\%	\#	$\backslash$	$\^{\{}$	$\_$
符号	{	}	~	ó	ö	ò	õ
命令	\{	\}	$\sim\{}$	$\' \{o\}$	$\" \{o\}$	$\. \{o\}$	$\H \{o\}$
符号	ø	ò	õ	ö	öö	o	ô
命令	$\d \{o\}$	$\' \{o\}$	$\sim \{o\}$	$\u \{o\}$	$\t \{oo\}$	$\b \{o\}$	$\^ \{o\}$
符号	ō	ö	ø	¶	§	†	‡
命令	$\{o\}$	$\v \{o\}$	$\c \{o\}$	$\P$	$\S$	$\dag$	$\ddag$
符号	£	©	ä	Å	†	Ł	œ
命令	$\pounds$	$\copyright$	$\aa$	$\AA$	$\l$	$\L$	$\oe$
符号	ß	ø	Ø	Œ	ı	ı	Æ
命令	$\ss$	$\o$	$\O$	$\OE$	!'	?'	$\AE$
符号	æ	$\TeX$	$\LaTeX$	$\LaTeX 2\epsilon$	$\AMS-\LaTeX$		
命令	$\ae$	$\TeX$	$\LaTeX$	$\LaTeXe$	$\AmS-\LaTeX$		

1

<sup>1</sup>更多符合可以参看CTEX帮助文档Symbols.pdf或者调用TexFriend.

## 其他一些约定

- ◆ 注释符: %
- ◆ 多个连续的空格被视为是一个空格, 英文标点后应空一格.
- ◆ 各种环境的开始和结束最后单独占用一行.
- ◆ 其它的输入: `dvi` → `ps`      `dvi` → `pdf`,      `ps` → `pdf`      `tex` →

`pdf`

- ◆ 换行: 两个回车, `\\`, `\\*`, `\newline` 等(它们的区别参考[这里](#))表示强制换行, 后面还可以加参数, 以增加当前行和新行之间的距离, 如 `\\[3mm]`

- ◆ 强制分页: `\newpage`

# 水平距离和垂直距离

- ◆ `\quad` 插入的空白相当于当前字体尺寸大小
- ◆ `\qquad` `\quad` 的两倍
- ◆ `\,` 大约是 `\quad` 的  $3/18$ , 而减少的距离可以使用 `\!`.
- ◆ `\hspace{宽度大小}` 和 `\hspace*{宽度大小}`
- ◆ `\hfill`: 行充满命令, 如 `\hspace{\hfill}`, 左内容 `\hfill` 右内容
- ◆ `\vspace{高度}` 和 `\vspace*{高度}`
- ◆ `\vfill`: 相当于 `\vspace{\fill}`
- ◆ `\smallskip`: 相当于 `\vspace{\smallskipamount}`
- ◆ `\medskip`: 相当于 `\vspace{\medskipamount}`
- ◆ `\bigskip`: 相当于 `\vspace{\bigskipamount}`

# 段落相关

❖ 段落首行缩进: 由`\parindent`的值决定, 例如`\setlength{\parindent}{2em}`.

❖ `\indent` 和 `\noindent`.

❖ 每节的一段首行不会自动缩进, 解决方法:

➡ `\hspace{\parindent}`

➡ 在导言区加上`\usepackage[indentfirst]`

❖ 段落间距: `\lineskip` + `\parskip`. 可以使用`\setlength`来修改这些距离的值.

❖ `\baselineskip`: 相邻两行基线之间的距离.  
例`\baselineskip=20pt`.

❖ `\baselinestretch`: 行距伸展因子, 例:

`\renewcommand{\baselinestretch}{1.2}`.

# LaTeX源文件例子

一个英文论文格式的LaTeX例子如下: (1.tex)

```
\documentclass[11pt] {article}
\title{First Example}
\date{}
\author{xxx}
\begin{document}
\maketitle
\begin{abstract}
  This is abstract.
\end{abstract}
\section{Introduction}
Hi, This is my first \LaTeX file.
\begin{thebibliography}{10}
  \bibitem{1}R1
  \bibitem{2}R2
\end{thebibliography}
\end{document}
```

❖ 第一行决定文档的版式和字体的大小.

👉 文档的版式有: `article`, `book`, `report`, `letter`, `ctexart`, `ctexbook`, `ctexrep`.

👉 字体的大小可以是12pt,10pt(默认)等.

❖ `\begin{xxx}` 和 `\end{xxx}` 构成一个环境. 一篇文档中只能有一个 `document` 环境.

❖ `\documentclass` 和 `\begin{document}` 之间为导言区, 用于放置一些全局控制命令.

# 支持中文

全部用pdf<sub>l</sub>atex 命令编译:

CJK

```
\documentclass{ctexart}
```

```
\begin{document}
```

中文宏包测试

```
\end{document}
```

或者

```
\documentclass{article}
```

```
\usepackage{ctex}
```

```
\begin{document}
```

中文宏包测试

```
\end{document}
```

CCT

```
\documentclass[CCT]{ctexart}#使用C
```

```
\begin{document}
```

中文宏包测试

```
\end{document}
```

```
\documentclass[CCT,CCTfont]{ctexar
```

```
\begin{document}
```

中文宏包测试

```
\end{document}
```

# 框架

- 1 T<sub>E</sub>X介绍
- 2 T<sub>E</sub>X的宏包和扩展
- 3 环境集**
- 4 L<sup>A</sup>T<sub>E</sub>X命令集
- 5 页面版式命令
- 6 计数器命令
- 7 目录表
- 8 交叉引用和索引
- 9 宏包
- 10 L<sup>A</sup>T<sub>E</sub>X中文化

# L<sup>A</sup>T<sub>E</sub>X环境

L<sup>A</sup>T<sub>E</sub>X环境包括文字版式环境, 图表环境和数学版式环境. 环境引用的格式是

```
\begin{环境名}[选项]
```

...内容...

```
\end{环境名}
```

L<sup>A</sup>T<sub>E</sub>X所有环境都是在`\begin{document}` 和`\end{document}` 环境中使用的.

# 文字版式环境

## ❖ **center** 环境

此环境将使位于`\begin{center}` 和`\end{center}` 之间的内容居中排印, 引用格式无选项. 例如要产生如下版式:

L<sup>A</sup>T<sub>E</sub>X for  
Scientists and Engineers  
by  
David J. Bouerger

则可用利用`center` 环境如下输入:

```
\begin{center}
\LaTeX \quad for \\\
Scientists and Engineers\\
by\\
David J. Bouerger
\end{center}
```

# 文字版式环境

## ❖ **flushleft** 环境

此环境将使位于`\begin{flushleft}` 和`\end{flushleft}` 之间的内容靠左边界对齐排印, 而右边界则不一定对齐引用格式无选项. 例如要产生如下邮寄地址:

SPIE

P.O.Box 10

Bellingham, W A 98227-0010

USA

则可用利用`flushleft` 环境如下输入:

```
\begin{flushleft}
SPIE\
P.O.Box 10\
Bellingham, W A 98227-0010\
USA
\end{flushleft}
```

# 文字版式环境

## ❖ **flushright** 环境

此环境与`flushleft`环境相反,产生右边界对齐版式,引用格式无选项.例如要产生如下版式:

Dept. of Stat & Fin.  
Univ. of Sci. & Tech.  
Hefei, Anhui, 230026  
China

则可用利用`flushright`环境如下输入:

```
Dept. of Stat \& Fin.\\
Univ. of Sci. \& Tech.\\
Hefei, Anhui, 230026\\
China
\end{flushright}
```

# 文字版式环境

## ❖ **itemize** 环境

此环境用于条款列举, 被列举条款由一个实心圆标记, 引用格式无选项. 例如要产生如下条款列举

- Optimal information processing
- Digital filtering

则可用利用itemize 环境如下输入:

```
\begin{itemize}
\item Optimal information processing
\item Digital filtering
\end{itemize}
```

# 文字版式环境

## ❖ **enumerate** 环境

此环境也用于条款列举, 与itemize 的差异在于被列举条款的引导标记符不同, 此环境下引导被列举条款的标记符为数字, 引用格式无选项. 例如要产生如下条款列举

1. Optimal information processing
2. Digital filtering

则可用利用enumerate 环境如下输入:

```
\begin{enumerate}
\item Optimal information processing
\item Digital filtering
\end{enumerate}
```

# 文字版式环境

## ❖ **enumerate** 环境

此环境也用于条款列举, 与`itemize` 的差异在于被列举条款的引导标记符不同, 此环境下引导被列举条款的标记符为数字, 引用格式无选项. 例如要产生如下条款列举

1. Optimal information processing
2. Digital filtering

则可用利用`enumerate` 环境如下输入:

```
\begin{enumerate}
\item Optimal information processing
\item Digital filtering
\end{enumerate}
```

`enumerate` 环境可以多重嵌套.

# 文字版式环境

## ❖ **description** 环境

此环境与上述两种环境的差别在于引导列举条款的标志符可以随意选择, 所以特别适合词条的解释: 词条名作标志符, 解释作条款. 例如

**Fox.** A animal being of four legs, simalar to a dog, but cleverer than the dog.

**Food Dash.** Used daily by the dog, underwent possible damage when the fox landed in it.

则可用利用`enumerate` 环境如下输入:

```
\begin{description}
\item[Fox.] A animal being of four legs, simalar to a dog, but cleverer than the dog.
\item[Food Dash.] Used daily by the dog, underwent possible damage when the fox landed in it.
\end{description}
```

# 文字版式环境

## ❖ **minipage** 环境

此环境定义局部段落的宽度小于标准正文宽度, 如摘要部分.

minipage 环境有选项, 格式为

```
\begin{minipage}{size} ... 内容... \end{minipage}
```

其中size 定义了段落的宽度尺寸. 例如要产生如下版面宽度为3 英寸的一段内容

```
This paper describes the use of a neural computational network model for pattern recognition and classification ...
```

则可用利用enumerate 环境如下输入:

```
\begin{center}
\begin{minipage}{3in}
  This paper describes the use of a neural computational network model for
  pattern recognition and classification ...
\end{minipage}
\end{center}
```

# 文字版式环境

## ❖ **quote, quotation** 环境

这两个环境用于引用文献内容或讲话内容的排版, 输出文献段的宽度小于正文的标准宽度, 相当于minipage 环境中某一特定尺寸的输出, 差别在于此环境下输出是自动居中的, 输出内容斜体, 没有引用选项. 输出宽度大约是标准宽度的%90. 例如

*This paper describes the use of a neural computational network model for pattern recognition and classification ...*

则可用如下输入:

```
\begin{quote}
  This paper describes the use of a neural computational network model for
  pattern recognition and classification ...
\end{quote}
```

# 文字版式环境

❖ **verb, verbatim/verbatim\*** 环境这两个环境均用于取消 $\text{\LaTeX}$ 排版, 即位于此环境中的内容将按源文件中输入格式输出,  $\text{\LaTeX}$ 将不对它进行排版. 例如

```
\begin{verbatim}
\[
\bigtriangledown^2=\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}
\]
```

会输出

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

而将输出命令放在`verbatim` 环境中时就会输出源代码, `verbatim*` 环境会标出空格. `verb` 命令用于句中的某些词句的源格式输出, 其使用格式为`\verb|...|` 或者`\verb=...=, \verb\...\.`

# 文字版式环境

❖ **verse** 环境此环境为诗词格式排版环境, 专用于诗词的排版, 引用格式无选项. 例如

```
\begin{center}
{\large\bf Gwendolyn Brooks \\\
The Bean Eaters}
\end{center}
\begin{verse}
They eat beans mostly, this old yellow air. \\\
Dinner is a casual affair. \\\
...
\end{verse}
```

会输出

**Gwendolyn Brooks**  
**The Bean Eaters**

*They eat beans mostly, this old yellow air.*  
*Dinner is a casual affair.*

# 图表环境

❖ **tabbing** 环境用于制作无线框表格, 引用格式无选项. 例如要制作如下格式的表

No.	Description	Sale Price
1.	Disk Drive	\$ 55
2.	Monitor	\$ 90

则可以利用此环境, 输入

```
\begin{tabbing}
No. \= Description \= Sale Price\\
1. \> Disk Drive \> \$ 55\\
2. \> Monitor \> \$ 90
\end{tabbing}
```

# 图表环境

❖ **tabular** 环境用于制作各种复杂的表格, 引用中有选项, 命令格式为

```
\begin{tabular}{对齐方式}...内容...\end{tabular}
```

对齐内容为每列对齐选项, 可选c(居中), l(左对齐)和r(右对齐).

例如

No.	Description	Sale Price
1.	Disk Drive	\$ 55
2.	Monitor	\$ 90

则可以利用此环境, 输入

```
\begin{tabular}{|c|c|c|}  
  \hline  
  % after \\: \hline or \cline{col1-col2} \cline{col3-col4} ...  
  No. & Description & Sale Price \\  
  \hline  
  1. & Disk Drive & \$ 55 \\
```

# 图表环境

## ❖ **table, figure** 环境

用于制作各种复杂的表格, 引用中有选项, 命令格式为

```
\begin{table/figure}{定位参数}  
\caption{图表标题}  
...内容...  
\end{table/figure}
```

对齐内容为每列对齐选项, 可选c(居中), l(左对齐)和r(右对齐).

定位参数选择有: b: 图表定位于页底; h: 图表定位于当前行; t: 图表定位于页顶; p: 图表单独占一页.

# 图表环境

例如

Table: 记账表

No.	Description	Sale Price
1.	Disk Drive	\$ 55
2.	Monitor	\$ 90

则可以利用此环境, 输入

```
\begin{table}[h]
\caption{记账表}
\begin{tabular}{|c|c|c|}
\hline
% after \: \hline or \cline{col1-col2} \cline{col3-col4} ...
No. & Description & Sale Price \\
\hline
1. & Disk Drive & \$ 55 \\
2. & Monitor & \$ 90 \\
\hline
\end{tabular}
\end{table}
```

# 图表环境

而要生产如下的浮动对象插图



Figure: USTC校徽

则可以利用此环境, 输入

```
\begin{figure}[h]
  % Requires \usepackage{graphicx}
  \includegraphics[width=3cm]{ustc.jpg}
  \caption{USTC校徽}\label{1}
\end{figure}
```

# 数学环境

## ❖ **math** 环境

此环境用于文中公式和数学符号的产生. 此环境下的公式不编号. 使用格式为`$...$` 或者`\(...\)` 或者`\begin{math}...\end{math}`. 例如

This is an example of a formula in text line,  $x^2 + y^2 = z^2$ .

可以使用

This is an example of a formula in text line, `$x^2+y^2=z^2$`.

或者

This is an example of a formula in text line, `\(x^2+y^2=z^2\)`.

或者

This is an example of a formula in text line, `\begin{math}x^2+y^2=z^2\end{math}`.

# 数学环境

## ❖ **displaymath** 环境

此环境用于单独占行, 但不编号的数学公式排版. 使用格式为`$$...$$` 或者`\[...\]` 或者`\begin{displaymath}...\end{displaymath}`. 例如

$$x^2 + y^2 = z^2.$$

可以使用

```
$$x^2+y^2=z^2.$$
```

或者

```
\[x^2+y^2=z^2.\]
```

或者

```
\begin{displaymath}
x^2+y^2=z^2.
\end{displaymath}
```

## ❖ **equation** 环境

此环境用于单独占行, 并且编号的数学公式排版. 使用格式为`\begin{equation}...\end{equation}`. 例如

$$x^2 + y^2 = z^2. \tag{1}$$

可以使用

```
\begin{equation}
x^2+y^2=z^2.
\end{equation}
```

## ❖ `array` 环境

此环境用于排版复杂的数学公式, 矩阵和行列式等, 使用格式为 `\begin{array}{对齐方式}...公式内容...\end{array}`. 例如对齐方式包括居中(c), 左对齐(l) 和右对齐(r).

$$\left( \begin{array}{ccc} a & b & c \\ d & e & f \\ & g & h \\ & & x \end{array} \right)$$

## 可以使用

```
\[  
\left(\begin{array}{c}  
\left[  
\begin{array}{ccc}  
a & b & c\\  
d & e & f  
\end{array}  
\end{array}  
\right)\\  
\begin{array}{cc}  
g & h  
\end{array} \\  
x  
\end{array}  
\right)  
\]
```

# 数学环境

## ❖ `eqnarray` 环境

此环境兼有`array` 和`equation` 两环境的特点, 特别适合排版复杂的多行公式和方程组. 产生的方程组自动编号. 使用格式为`\begin{eqnarray/eqnarray*}...公式内容...\end{eqnarray/eqnarray*}`. 加\*后将不自动编号.

$$x^2 + y^2 + z^2 = 34 \tag{2}$$

$$x + y + z = 10$$

$$x + 10y = 20 \tag{3}$$

则可以输入

```
\begin{eqnarray}
&&x^2+y^2+z^2=34\\
&x+y+z=10\ \nonumber\\
&x+10y=20\end{eqnarray}
```

# 框架

- 1 T<sub>E</sub>X介绍
- 2 T<sub>E</sub>X的宏包和扩展
- 3 环境集
- 4 L<sup>A</sup>T<sub>E</sub>X命令集**
- 5 页面版式命令
- 6 计数器命令
- 7 目录表
- 8 交叉引用和索引
- 9 宏包
- 10 L<sup>A</sup>T<sub>E</sub>X中文化

# 字体命令

❖  $\text{\LaTeX}$ 中任何字体族中的字体一般有三种形式: 字体族, 字体的外形和font series.  $\text{\LaTeX}$ 缺省有三种字体族

字体族	代码	命令
罗马	rm	<code>\rm</code> 或者 <code>\textrm{}</code>
Sans-serif	sf	<code>\sf</code> 或者 <code>\textsf{}</code>
打字体	tt	<code>\tt</code> 或者 <code>\texttt{}</code>

字体外形和font series 命令有

Type Style	命令	例子
意大利体	<code>\textit{words}</code>	<i>words</i>
斜体	<code>\textsl{words}</code>	<i>words</i>
小号大写字体	<code>\textsc{words}</code>	WORDS
粗体	<code>\textbf{words}</code>	<b>words</b>
Sans-serif	<code>\textsf{words}</code>	words
打字体	<code>\texttt{words}</code>	words
强调体	<code>\emph{words}</code>	words

# 字体命令

❖ 字体大小的命令有(从小到大):

`\tiny`, `\scriptsize`, `\footnotesize`, `\small`, `\normalsize`,  
`\large`, `\LARGE`, `\huge`, `\Huge`.

❖ 数学环境下, 公式或者符号的大小命令有(从小到大):

`\scriptscriptstyle`, `\scriptstyle`, `\textstyle`,  
`\displaystyle`, 后两者对于变量的尺寸是一样大的.

# 框架

- 1 T<sub>E</sub>X介绍
- 2 T<sub>E</sub>X的宏包和扩展
- 3 环境集
- 4 L<sup>A</sup>T<sub>E</sub>X命令集
- 5 页面版式命令**
- 6 计数器命令
- 7 目录表
- 8 交叉引用和索引
- 9 宏包
- 10 L<sup>A</sup>T<sub>E</sub>X中文化

# 页面版式命令

## □ 分栏

```
\documentclass[twocolumn]{article}
```

- 默认为`onecolumn`.
- 选用`twocolumn`时可以指定
  - 栏间距`\columnsep`
  - 栏间分割线的粗细`\columnseprule`, 默认为0
  - 栏宽参数`\columnwidth`
  - 通常不用设置
- ♣ 若对整篇文档作修改, 可将设置命令放在导言区.

# 页面版式命令

## □ 单双面选项 `oneside`, `twoside`

```
\documentclass[twocolumn]{article}
```

➤ 单面版式: 所有页码打印位置一致.

双面版式: 页码打印在外侧, 奇右偶左.

➤ 单面版式: 页面左右边距相同

双面版式: 内侧边距相同, 外侧边距相同

➤ `article` 缺省为 `oneside`

`book` 缺省为 `twoside`, `book` 类文档还有选项 `openany`, `openright` 控制每章开始位置, 缺省为 `openright`.

➤ 单面版式: 所有页面均称为右页

双面版式: 奇数页为右页, 偶数页为左页

# 页面版式命令

## □ 控制页面版式的命令

```
\pagestyle{版式}      \thispagestyle{版式}
```

## □ 常用的版式有:

- **plain** 缺省版式, 页眉为空, 页码居中
- **empty** 页眉页脚都是空, 不显示页码
- **headings** 显示页眉页脚(含页码, 页眉标题)
  - 页眉标题由章,节或者小节的标题组成, 自动提取
  - **book** 每一章的第一页和**article** 的首页不显示页眉, 在底部显示页码.
- **myheadings** 与**headings**类似, 但可以使用`\markright`, `\markboth`来定义页眉标题.

# 页面版式命令

正文版式命令用于设置正文的各种参数, 这些命令均位于导言区. 如

```
\setlength{\textheight}{24true cm}      %正文高度
\setlength{\textwidth}{14.5true cm}     %正文宽度
\setlength{\parindent}{20pt}            %段落首行缩进.
\setlength{\parskip}{3pt plus1pt minus2pt} %段落之间的距离
\setlength{\baselineskip}{15pt plus2pt minus1pt}%段落内各行之间的距离.
```

以及

```
\setlength{\topmargin}{-3cm} %距离页面顶部距离
\setlength{\oddsidemargin}{0.6cm} %奇数页左边距
\setlength{\evensidemargin}{0.3cm}%偶数页左边距
```

# 页面版式命令

❖ `\twocolumn`, `\onecolumn` 双列和单列转换命令如

```
\documentclass{article}
\begin{document}
...内容1...
\twocolumn
...内容2...
\onecolumn
...内容3...
\end{document}
```

❖ `\centerline` 行居中命令

# 页面版式命令

❖ `\topskip` 此命令改变由`\textheight` 命令设置的页顶位置, 命令格式为`\topskip size`. 新的页顶部相对于原来的页顶部的偏离距离为`size`, 在此命令之后的新页的页顶将下移`size` 距离, 即新页的高度缩短了`size`. `size` 的默认距离是10pt.

❖ `\leftskip`, `\rightskip` 此命令设置版面的左边距或右边距到新的位置, 使用格式为`\leftskip/rightskip size`. `size` 即为新的左边界或右边界相对于原左边界或右边界的偏离距离值, 它的默认值为0pt.

# 编号格式命令

编号格式用于设置编号(如页号, 章节号等)的表达方式. 编号的表示方式有阿拉伯数字, 罗马数字, 字母等, L<sup>A</sup>T<sub>E</sub>X提供五种编号表示方式, 使用格式为

命令{options}

options 为参数, 它为各种计数器名, 如page, chapter, equation 等.

阿拉伯数	小写罗马数	大写罗马数	小写字母	大写字母
<code>\arabic</code>	<code>\roman</code>	<code>\Roman</code>	<code>\alph</code>	<code>\Alph</code>

❖ `\pagenumbering` 页码格式设置命令, 对页码格式的设置, 除了使用如上命令外, 还可以使用`\pagenumbering`命令, 使用格式为

`\pagenumbering{options}`,

参数 options 可选值包括arabic, roman, Roman, alpha 和Alph

# 框架

- 1 T<sub>E</sub>X介绍
- 2 T<sub>E</sub>X的宏包和扩展
- 3 环境集
- 4 L<sup>A</sup>T<sub>E</sub>X命令集
- 5 页面版式命令
- 6 计数器命令**
- 7 目录表
- 8 交叉引用和索引
- 9 宏包
- 10 L<sup>A</sup>T<sub>E</sub>X中文化

# 计数器命令

L<sup>A</sup>T<sub>E</sub>X中的所有自动编号都是基于计数器的值. 若想认为干预编号顺序, 就必须改变计数器的值. L<sup>A</sup>T<sub>E</sub>X提供了两个改变计数器值的命令和一个自定义新计数器的命令.

❖ `\setcounter` 此命令设置指定计数器的初值为任意给定值, 使用格式为

```
\setcounter{计数器名}{给定值}
```

❖ `\addtocounter` 此命令也用于改变各种计数器的值, 使用格式为

```
\addtocounter{计数器名}{增量}
```

❖ `\newcounter` 此命令用于定义新的计数器, 使用格式为

`\newcounter{新计数器名}` 新定义的计数器名不能与下表中的同名.

# 计数器命令

计数器名	意义
<code>page</code>	页码计数器
<code>chapter</code>	章编号计数器
<code>section</code>	节编号计数器
<code>subsection</code>	子节编号计数器
<code>subsubsection</code>	子子节编号计数器
<code>part</code>	部分计数器
<code>paragraph</code>	段计数器
<code>subparagraph</code>	子段计数器
<code>equation</code>	公式计数器
<code>figure</code>	图计数器
<code>table</code>	表计数器
<code>enumi</code>	<code>enumerate</code> 环境条款编号计数器
<code>enumii</code>	<code>enumerate</code> 环境二级嵌套中列举条款编号
<code>enumiii</code>	<code>enumerate</code> 环境三级嵌套中列举条款编号
<code>enumiv</code>	<code>enumerate</code> 环境四级嵌套中列举条款编号
<code>footnote</code>	脚注计数器
<code>minipage</code>	<code>minipage</code> 环境中脚注编号

# 框架

- 1 T<sub>E</sub>X介绍
- 2 T<sub>E</sub>X的宏包和扩展
- 3 环境集
- 4 L<sup>A</sup>T<sub>E</sub>X命令集
- 5 页面版式命令
- 6 计数器命令
- 7 目录表**
- 8 交叉引用和索引
- 9 宏包
- 10 L<sup>A</sup>T<sub>E</sub>X中文化

# 目录表

## □ 自动生成目录

```
\tableofcontents
```

## ➤ 设置出现在目录中章节层次

```
\setcounter{docdepth}{数}
```

- 数代表层次
- 为了生成目录, 需要编译多次
- 只有自动编号的章节才会出现在目录表中.
- 在目录中输出不编号的章节:

```
\addcontentsline{toc}{章节名称}{条目内容}  
\addtocontents{toc}{条目内容}
```

# 目录表

## □ 插图目录和表格目录

```
\listoffigures, \listoftables
```

- 输出浮动环境`figure`和`table`的目录.
- 图表目录显示的是`caption`里的内容.
- 手工插入其他图标内容

```
\addcontentsline{清单类型}{格式}{条目内容}  
\addtocontents{清单类型}{格式}
```

清单类型: `toc`(目录), `lof`(插图), `lot`(表格).

格式: 章节名称, `figure`, `table`

# 目录表

## □ 目录标题中文化

```
\renewcommand{\contentsname}{\hei 目\quad 录}  
\renewcommand{\listfigurename}{\hei 插图目录}  
\renewcommand{\listtablename}{\hei 表格目录}
```

# 框架

- 1 T<sub>E</sub>X介绍
- 2 T<sub>E</sub>X的宏包和扩展
- 3 环境集
- 4 L<sup>A</sup>T<sub>E</sub>X命令集
- 5 页面版式命令
- 6 计数器命令
- 7 目录表
- 8 交叉引用和索引**
- 9 宏包
- 10 L<sup>A</sup>T<sub>E</sub>X中文化

# 交叉引用

- 交叉引用:  $\text{\LaTeX}$ 非常实用的一个功能
  - 章节, 表格, 插图, 公式, 页码, 参考文献等.
  - $\text{\LaTeX}$ 自动跟踪所有变动, 保证交叉引用于索引记录的正确性.
- `\label{标记}` 设置标记.
- `\ref{标记}` 显示标记所对应对象的编号. `\pageref{标记}` 显示标记对象所在页面的页码.

# 索引

- 索引: 在导言区加入

```
\usepackage{makeidx}  
\makeindex
```

- `\index{索引条目}` 标记索引条目.
- `\printindex` 输出索引, 一般放置在文末.
  - 索引标题 `\indexname`
  - 索引按ASCII 码排序, 中文索引处理起来要麻烦一些, 参考一些教材资料.

# 框架

- 1 T<sub>E</sub>X介绍
- 2 T<sub>E</sub>X的宏包和扩展
- 3 环境集
- 4 L<sup>A</sup>T<sub>E</sub>X命令集
- 5 页面版式命令
- 6 计数器命令
- 7 目录表
- 8 交叉引用和索引
- 9 宏包**
- 10 L<sup>A</sup>T<sub>E</sub>X中文化

针对不同的目的, L<sup>A</sup>T<sub>E</sub>X有许多宏包来扩充标准L<sup>A</sup>T<sub>E</sub>X的功能, 在  
导言区加入

```
\usepackage{宏包名称}
```

常用的宏包有:

amsmath, amssymb, mathrsfs, amsthm, amsfonts, graphics, fancybox,  
fancyhdr

更多宏包的介绍参考CTEX 网站在线文档: [TeX/LaTeX 常用宏包介绍](#) .

# 框架

- 1 T<sub>E</sub>X介绍
- 2 T<sub>E</sub>X的宏包和扩展
- 3 环境集
- 4 L<sup>A</sup>T<sub>E</sub>X命令集
- 5 页面版式命令
- 6 计数器命令
- 7 目录表
- 8 交叉引用和索引
- 9 宏包
- 10 L<sup>A</sup>T<sub>E</sub>X中文化**

TeX系统最初没有考虑到支持多国语言, 因此TeX系统对中文(日文, 韩文)一直支持的不太好. 目前TeX的中文化支持已经有几种方案: CCT, TY和CJK.

□ CCT 前面已经说过, CCT 对中文特别是中文标点的支持很好. CCT早期的版本中以扩展名`.ctx`来定义CCT中文LaTeX文档, 编译此类文档时使用`CCT + LaTeX`来编译. 后期的版本都以`.tex`作为扩展名. CCT 文档包括`cctart`, `cctbook`, 例如

```
\documentclass[10pt]{cctart}
\begin{document}
...内容...
\end{document}
```

# L<sup>A</sup>T<sub>E</sub>X中文化

- CJK 使用英文的文档类, 但是在导言区加上`\usepackage{CJK}` 来获得支持中文的能力.
- CJK 定义了两种环境: CJK 和CJK\*: 中文必须放在CJK环境中.

```
\begin{CJK或者CJK*}{编码}{字体族}
...内容...
\end{CJK或者CJK*}
```

两种环境的差异在于CJK\*忽略汉字后面的空格.

- 编码: GB, GBK, Big5, Gbt
- 字体族: kai, hei, song, fs,li, you

❑ CTEX 2.7 以后对这些不同的中文支持方法进行了统一封装, 中文的文档类有 `ctexart`, `ctexbook`, `ctexrep`, 文档的扩展名为 `.tex`.

❑ CTEX 文档类默认使用 `CJK` 宏包, 因此以下两种方式等价

<code>\documentclass{article}</code>	<code>\documentclass{ctexart}</code>
<code>\begin{document}</code>	<code>\begin{document}</code>
<code>\begin{CJK}{GBK}{song}</code>	...内容...
...内容...	<code>\end{document}</code>
<code>\end{CJK}</code>	
<code>\end{document}</code>	

# L<sup>A</sup>T<sub>E</sub>X 中文化

□ CTEX 宏包可以配合CCT使用, 因此以下两种方式等价

<code>\documentclass{cctart}</code>	<code>\documentclass[CCT,CCTfont]{ctexart}</code>
<code>\begin{document}</code>	<code>\begin{document}</code>
...内容...	...内容...
<code>\end{document}</code>	<code>\end{document}</code>

更多的中文化支持文档请参考

- CTEX 说明文档 `$CTEX\CTeX\doc\latex\ctex`
- CCT 说明文档 `$CTEX\CTeX\cct\doc`
- CJK 说明文档 `$CTEX\MiKTeX\doc\latex\cjk` 下的CJK.txt

和commands.txt.