

LaTeX 科技排版

LaTeX 介绍

潘建瑜

华东师范大学数学系

2016

内容提要

- ① T_EX 与 L^AT_EX
- ② L^AT_EX 排版基础

TeX 介绍

TeX 是一个 **功能强大** 且 **十分灵活** 的排版语言

- TeX 系统提供了 300 + 600 多条基本的排版命令
- TeX 是目前公认的 **数学公式排版最好的** 排版语言
- TeX 是 **免费** 的
- TeX 的名字来自大写的希腊字母 (τ, ϵ, χ), 意思是 “科技” 和 “艺术”

TeX 的优点

- 高质量的输出: TeX 以排版质量为首要目标
- 超常的稳定性: 系统极少崩溃
- TeX 是可编程的 (宏命令编程语言)
 - 你可以用很少的命令来完成一些复杂的工作
 - 或通过定义新的命令来得到特殊的效果
- TeX 文档是纯文本文件: 占用空间很小
- 良好的通用性: Windows, Linux, MacOS 都支持



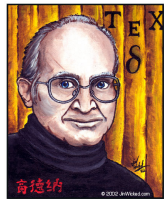
TeX 的缺点

- 命令繁多, 不便记忆
在使用中学习, 通过网络寻求帮助, 人性化的编辑器
- 错误难找: 慢慢积累经验
- 使用不直观
 - 要编译后才能显示效果
 - 每次修改后都要再次编译
 - 目前已有一些所见即所得的扩展, 如: LyX, TeXmacs



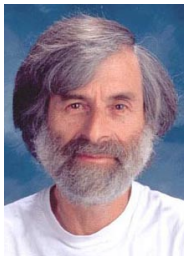
TeX 的发展

- TeX 的第一版于 1978 年面世
70 年代末, 斯坦福大学计算机系教授 D. E. Knuth 在看到其巨著 “The Art of Computer Programming” 第二卷的校样时, 对排版的低质量感到无法忍受, 于是决定开发一个高质量的排版系统, 于是就有了 TeX
- 1982 年版的 TeX 已是一个十分稳定的版本
- 1990 年推出 3.1 版, 并宣布不再更新 (只修正 bug)
- 最新版本号: 3.14159265 (2014 年) $\rightarrow \pi$
- <http://zh.wikipedia.org/wiki/TeX>
- <http://www.cs.stanford.edu/~knuth/>



TeX 扩展 / 排版引擎

- TeX 提供的命令都是一些很底层的命令, 普通用户使用起来不太方便
- TeX 扩展: 在 TeX 基础上, 定义新的命令, 为普通用户排版提供方便
 - Plain TeX: Knuth 开发, 新定义 600 多条命令, 是通常所说的 TeX
 - AMS-TeX: 由美国数学会开发, 适合排版各种复杂的数学公式
 - LaTeX (1984): Lamport 开发, 适合排版论文书籍, 对 TeX 推广贡献巨大
 - ConTeXt (1996): Hans Hagen 开发, 进一步增加 TeX 的易用性



Lamport



Hagen

TeX 扩展 / 排版引擎

- pdfTeX (1997): 由 Han The Thanh 开发, 可直接输出 PDF 文件, 支持多种图片格式
- xeTeX (2005): 由 Jonathan Kew 开发, 支持 Unicode 编码, 可直接访问操作系统字体 <http://xetex.sourceforge.net>
- LuaTeX (2007): 由 Taco Hoekwater 开发, 是 Lua 脚本语言和 TeX 语言的结合体 <http://luatex.org>



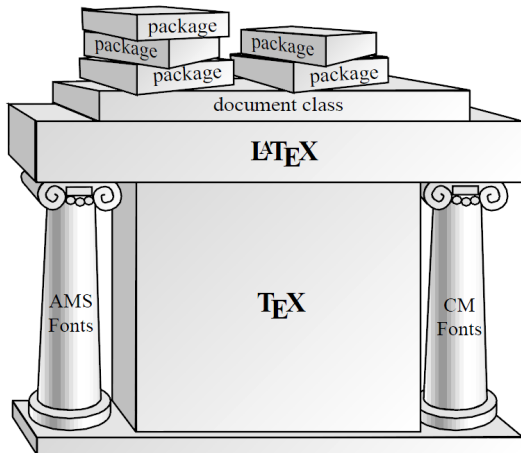
Thanh



Kew



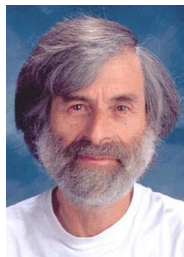
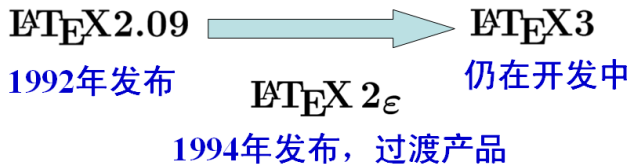
Hoekwater



TeX 结构图

LaTeX 介绍

- 1984 年前后, L^AT_EX 开发了 LaTeX
- LaTeX 提供了一套针对文档逻辑结构的控制序列, 让 TeX 变得更好用了, 普通用户即便不是很了解 TeX, 也可以在很短的时间制作出高质量的排版结果, 因此迅速在计算机科学、数学及相关学科领域得到推广应用



LaTeX 的特点

- 继承了 TeX 的优点和缺点
- 自动编号: 章节、图表、公式定理、文献、...
- 自动生成目录、索引
- 公式、定理、参考文献、插图、页码等可以交叉引用
- 可以通过添加各种宏包扩展其功能, 实现各种特殊要求
-

LaTeX 和 Word

- Word 简单易用, 所见即所得 → 普通办公文档建议用 Word
- LaTeX 输出美观, 质量高 → 科技排版 (特别是数学) 推荐使用 LaTeX
- LaTeX 能实现 Word 的所有功能, 定制性高, 但易用性不如 Word

- TeX 主要用户群: 高校和科研院所的教师和学生, 中学数学教师等
- 主要学科: 数学, 力学, 物理, 计算机等
- 接受 TeX 投稿的出版机构
 - AMS (美国数学会)
 - SIAM (美国工业与应用数学学会)
 - Springer
 - Elsevier
 - IEEE
 - Cambridge University Press
 - Kluwer Academic Publishers
 - Wiley
 - World Scientific
 -

中文 TeX 排版

- CCT → 最早支持简体中文的 TeX 扩展, 由中科院张林波教授开发, 比较符合中国人的习惯和时下中国印刷界的现行标准
- TY → 由华东师范大学肖刚、陈志杰等教授开发
- CJK → 由德国 Lemberg 开发, 可以同时处理中、日、韩三国文字
- ctex 宏集: 面向中文排版的通用 LaTeX 排版框架

中文排版建议: $\text{xeTeX} + \text{ctex}$ 或 $\text{LuaTeX} + \text{ctex}$

TeX 的发行版

- Windows 系统: MikTeX, TeXlive, CTeX
- Unix/Linux 系统: TeXlive
- Mac OS 系统: MacTeX, TeXlive

软件获取和安装

- CTeX 套装: <http://www.ctex.org>
 - 含 MiKTeX, WinEdt, SumatraPDF, TeXFriend 等, 安装后更新 MiKTeX
- Texlive: <http://tug.org/texlive/>
 - 支持在线更新, 大而全, 支持 Windows, Linux, Mac OS
- 编辑器: WinEdt, Texmaker, Texworks, Texstudio, vim, emacs, ...

学习资料

- 📖 [The Not So Short Introduction to LaTeX2e \(in 157 minutes\)](#), Ver. 5.05, 2015
- 📖 [The TeX Book](#), D.E. Knuth, 1993
- 📖 [LaTeX: A Document Preparation System](#) (2nd), L. Lamport, 1994
- 📖 [The Latex Companion](#) (2nd), M. Goossens 等, 2004.
- 📖 [More Math Into LaTeX](#) (4th), G. Gratzer, 2007.

- 📖 [一份不太简短的 LaTeX2e 介绍](#), 基于英文 4.2 版, 2006
- 📖 [LaTeX2e 完全学习手册](#) (2nd), 胡伟, 2013
- 📖 [LaTeX 入门](#), 刘海洋, 2013
- 📖 [LaTeX 入门与提高](#) (2nd), 陈志杰等, 2006

网络资源

- 🔗 <http://www.ctex.org>
- 🔗 TeX@newsrth
- 🔗 <http://www.chinatex.org>
- 🔗 <http://www.latexstudio.net>
- 🔗 LaTeX 编辑部
- 🔗 <http://math.ecnu.edu.cn/~jypan/Teaching/Latex/>
- 🔗 <http://www.tug.org/>
- 🔗 <http://www.ctan.org/>
- 🔗 <http://tex.stackexchange.com/>

① T_EX 与 L^AT_EX

② L^AT_EX 排版基础

L^AT_EX 文稿的排版过程

① 编写源文件:

- tex 源文件为纯文本文件, 以.tex 为扩展名, 如 mypaper.tex
- 可以使用任何文本编辑器编写, 如: EditPlus, Vi, Emacs, ...
- Windows 系统推荐 [WinEdt](#) → 专门针对 tex 开发的文本编辑器, 提供许多便捷功能, 有助于提高排版效率

② 编译 tex 文件:

- 用 latex, pdflatex, xelatex, lualatex 等编译
- 推荐使用 xelatex, lualatex 编译, 直接生成 pdf 文件

LaTeX 源文件的基本框架

```
1 \documentclass{article} % 指定文档类型
2 % 导言区：全局设置，宏包调用等
3 \begin{document} % 正文开始
4
5 Hi, this is my first \LaTeX{} file.
6
7 \end{document} % 结束
```

- 反斜杠开头的字符串: 排版命令(简称命令)
- 注解符: 百分号 %

源文件的基本组成

- LaTeX 源文件: 正文 + 命令 + 注解
- LaTeX 命令: 反斜杠开头的字符串 (只含字母, 区分大小写), 可带参数

```
\command  
\command[option]{arguments}
```

- 方括号中的参数是可选的 (选项), 花括号中的参数是必需的
- 参数可以是一个, 也可以有多个 (若有多个, 用逗号隔开)
- 部分不带参数的命令有时也称为声明

例: 几个常用命令

- `\documentclass`, `\title`, `\author`, `\date`
- `\usepackage`, `\setlength`, `\newcommand`, `\renewcommand`
- `\begin{环境名}`, `\end{环境名}` → 组成一个环境

选择文档类

```
\documentclass[选项]{文档类}
```

- 必须出现在最前面, 用于指定文档的全局版式.
 - 常用的文档类有: `article`, `book`, `beamer`, `ctexart`, `ctexbook`
 - 常用的选项有:
 - `10pt`(缺省值), `11pt`, `12pt` → 指定基本字体的大小
 - `letterpaper`(缺省值), `a4paper`, `a5paper`, ... → 指定纸张的大小

☞ 选项可以组合使用, 如: (相互排斥的除外)

```
1 \documentclass[a4paper]{article}
2 \documentclass[11pt,a4paper]{book}
```

☞ 不同的文档类有不同的选项, 详见后面的介绍

自定义命令

```
\newcommand{新命令}[参数个数][缺省值]{命令内容}
```

- **缺省值**用于设定第一个参数的默认值
- 在使用新命令时, 实际的参数个数可以比指定的**参数个数**少一个, 此时, 第一个参数就是**可选参数**, 使用给定的**缺省值**
- **新命令**中最多只能有一个可选参数, 且必须是第一个参数, 例

```
1 \newcommand{\mycolor}[2][blue]{\textcolor{#1}{#2}}
```

`\mycolor{math}` 就等价于 `\textcolor{blue}{math}`

`\mycolor{red}{math}` 等价于 `\textcolor{red}{math}`

自定义命令

- 用户也可以修改已有的命令

```
\renewcommand{已有命令}[参数个数][缺省值]{命令内容}
```

```
1 \renewcommand{\baselinestretch}{1.2}
```

分组

- 有些命令只对其参数或紧跟其后的一个字符起作用
- 有些命令对后面所有的文本都起作用, 这些名称通常也称为**声明**
- 可以利用大括号 (即**分组**) 来**扩展**或**限制**命令的作用范围

```
1 This is \textbf bold face style.\\
2 This is \textbf{bold face} style.\\
3 This is \bfseries bold face style.\\
4 This is {\bfseries bold face} style.
```

输出结果是

This is **bold** face style. → `\textbf` 只对后面一个字符起作用
This is **bold face** style. → 用分组扩展 `\textbf` 的作用
This is **bold face style**. → `\bfseries` 对后面所有的文本起作用
This is **bold face** style. → 用分组限制 `\bfseries` 的作用

环境

环境是 L^AT_EX 中的一个重要概念

- 在 L^AT_EX 中, 为了排版某些具有特定格式的文本, 需要把它们放在相应的环境中, 如表格, 列表, 数学公式等. 基本语法为:

```
\begin{环境名}  
:  
\end{环境名}
```

- 开始和结束的**环境名**必须一致
- 环境可以嵌套, 但不能交叉
- 环境中可包含其它命令, 通常这些命令只在该环境中起作用
- **document** 环境是 L^AT_EX 的一个最基本的环境, 一篇文档有且只能有一个 **document** 环境, 正文的所有内容都必须放在 **document** 环境中

导言区

导言区: `\documentclass` 和 `\begin{document}` 之间的区域

- 导言区用于放置**全局控制命令**, 如: 调用宏包, 设置页面大小, ...
- 放在导言区的命令对整个文档都起作用
- 有些命令只能放在导言区, 如: 调用宏包, 设置页面大小
- 有些命令不能放在导言区, 如: 章节命令
- 有些命令既可以放在导言区, 也可以放在正文中, 如: 定义新命令

LaTeX 举例

```
1 \documentclass[11pt,a4paper]{article}
2 \usepackage{amsmath} % AMS 数学公式宏包
3 \usepackage{amssymb} % AMS 数学符号
4 \usepackage{amsfonts} % AMS 数学字体
5 \begin{document}
6   The Euler equation is given by
7   $$ e^{ix} \triangleq \cos(x) + i\sin(x). $$
8 \end{document}
```

宏包

宏包 是对 L^AT_EX 功能的扩展

- 宏包调用方法 (只能出现在导言区)

```
\usepackage[选项]{宏包名}
```

- 如果宏包不带选项, 则可以多个一起调用, 如:

```
\usepackage{amsmath, amssymb}
```

- 常用宏包:
 - CJK, ctex, geometry, fancyhdr, natbib, float, caption
 - amsmath, amssymb, amsfons, amsthm, ntheorem, bm
 - xcolor, graphicx, subfigure, picins, wallpaper, epstopdf
 - longtable, colortbl, tcolorbox, mdframed
 - algorithm, algpseudocode, listings

长度

长度由十进制数和长度单位表示, 如: `0.5cm, 11pt, ...`

- 常用长度单位

mm 毫米	pt 点 / 磅
cm 厘米	em 大约为大写字母 M 的宽度
in 英寸	ex 大约为小写字母 x 的高度

☞ 1 in = 2.54cm = 72pt, em 和 ex 与当前字体尺寸有关

- 弹性长度: 可根据排版需要自动伸缩, 基本语法为

`正常值 plus 伸展值 minus 收缩值`

☞ 实际长度可以超过正常值+伸展值, 但不能小于正常值-收缩值, 如

`10pt plus 2pt minus 3pt`


- 弹性长度 `\fill` → 表示充满, 正常值为零, 但可以伸展到任何长度
- `mu`: 数学长度单位, 专用于数学公式, $18\mu=1em$
- 长度设置与修改

```
\setlength{长度数据命令}{长度}  
\addtolength{长度数据命令}{长度}
```

- 长度有时也可以是负值, 如: `\setlength{\hoffset}{-5mm}`
- 常用长度数据命令 (变量)
 - `\textwidth`, `\textheight`
 - `\parindent`, `\parskip`
 - `\hoffset`, `\voffset`
 - `\baselineskip`

分段

- **换行**: LaTeX 会自动换行, 若需 **强制换行**, 可使用 `\\` 或 `*` 或 `\newline`
 - `*` 的另外一个作用: 禁止在换行处分页
 - `\\` 后面可以带 **长度**, 以增加当前行与新行之间的距离, 参数可正可负, 如: `\\[3mm]`, `\\[-5pt]`
- **建议换行**: `\linebreak[n]`
 - `n` 可以取 0, 1, 2, 3, 4, 数越大表示建议力度越大
 - `\linebreak[4]` 等价于 `\linebreak`, 表示强制换行, 并撑满整行
 - `\nolinebreak[n]` 建议不分行

 一般情况下, 不建议使用强制换行
- **分段**: `\par` 或 **两个连续回车** (即一个空行)

分页

- 分页: L^AT_EX 会自动分页
若需 强制分页, 可用命令 `\newpage` 或 `\clearpage`
- 建议分页/不分页: `\pagebreak[n]`, `\nopagebreak[n]`
- 增加当前页的高度
 - `\enlargethispage{尺寸}`: 指定可增加的最大高度
 - `\enlargethispage*{尺寸}`: 严格指定增加的高度

行间距

- 行间距 (可以用 `\setlength` 修改)
 - `\baselineskip` → 相邻两行 **基线** 之间的距离
 - `\lineskip` → 前一行下边界与后一行上边界之间的距离
- 行间距伸展因子: `\baselinestretch`
- 行间距伸展命令: `\linespread`

```
1 \renewcommand{\baselinestretch}{1.2}  
2 \linespread{1.2}
```

- 建议用 **伸展因子** 或 **伸展命令** 来修改行间距
- 如果上述命令放在导言区时, 则直接对整篇文档起作用
- 若放在正文中, 则只有当字体尺寸发生改变时才会起作用
若需要立即起作用, 可在其后面加上 `\selectfont`

段间距

- 段落间距: `\lineskip` + `\parskip`

修改段间距:

```
\setlength{\parskip}{长度}
```

- 段落缩进
 - 段落首行缩进的长度 → `\parindent`
 - 每一节的第一段首行不会自动缩进 (英文习惯)
 - 解决方法: 调用 `indentfirst` 宏包: `\usepackage{indentfirst}`
- 几个相关命令
 - `\indent` → 强制缩进 `\parindent` 的距离
 - `\noindent` → 不允许缩进

水平间距

- 强制空格: `_` 或 `~`
- `\quad` → 产生一段宽度为 1em 的水平空白
- `\qqquad` → `\quad` 的两倍
- `\enskip` → `\quad` 的一半
- `\thinspace` 或 `\,` → 大约为 `\quad` 的 3/18
- `\hspace{宽度}` → 产生指定宽度的水平空白
- `\hspace*{宽度}` → 同 `\hspace`, 若需要在行首产生一定的空白, 则需使用此命令
- `\hfill` → `\hspace{\fill}`, 根据排版需要插入空白, 撑满整行
- `\hphantom{文本内容}`: 水平空白的宽度等于文本内容的总宽度
- 导引线: `\dotfill`, `\hrulefill`

垂直间距

- `\smallskip` → 垂直空白高度为 3pt plus 1pt minus 1pt
- `\medskip` → `\smallskip` 的两倍
- `\bigskip` → `\smallskip` 的四倍
- `\vspace{高度}` → 产生指定高度的垂直空白
- `\vspace*{高度}` → 同 `\vspace`, 主要同在页面的顶部或底部
- `\vfill` → `\vspace{\fill}`
- `\vphantom{文本内容}` → 垂直空白的高度等于文本内容的总高度

特殊字符

- 有 10 个字符被赋予了特殊用途, 需要使用相应的命令才能输出

字符	#	\$	%	{	}	~	_	^	&	\
命令	<code>\#</code>	<code>\\$</code>	<code>\%</code>	<code>\{</code>	<code>\}</code>	<code>\~{}</code>	<code>_{} </code>	<code>\^{} </code>	<code>\&</code>	<code>\textbackslash</code>

- 符号 “>”, “<”, “|” 被定义成数学符号, 只能用在数学模式中, 若要在普通文本中输出, 也需使用相应的命令

字符		<	>
命令	<code>\textbar</code>	<code>\textless</code>	<code>\textgreater</code>

- 引号与连字号

字符	‘	’	“	”	`	-	-	—
命令	<code>\` (倒引号)</code>	<code>'</code>	<code>\`</code>	<code>"</code> 或 <code>'</code>	<code>\`{}</code>	<code>-</code>	<code>--</code>	<code>---</code>

特殊字符

- TeX 标识符, 重音符号 等

字符	TeX	LaTeX	LaTeX 2 _ε	AMS-LaTeX ¹
命令	<code>\TeX</code>	<code>\LaTeX</code>	<code>\LaTeXe</code>	<code>\AmS-\LaTeX</code>

ò	ó	ô	ö	õ	ō	ó	ö
<code>\o</code>	<code>\'o</code>	<code>\^o</code>	<code>\"o</code>	<code>\~o</code>	<code>\=o</code>	<code>\.o</code>	<code>\u{o}</code>
ö	ó	ô	ôo	o	q	q	
<code>\v{o}</code>	<code>\H{o}</code>	<code>\r{o}</code>	<code>\t{oo}</code>	<code>\b{o}</code>	<code>\c{o}</code>	<code>\d{o}</code>	

字符	§	¶	†	‡	©	£
命令	<code>\S</code>	<code>\P</code>	<code>\dag</code>	<code>\ddag</code>	<code>\copyright</code>	<code>\pounds</code>

更多特殊字符参见: [The Comprehensive LATEX Symbol List](#)

可使用 WinEdt 或 TeXFriend 中提供的便捷方式

书写源文件时的注意事项

- 中文和英文之间, 中文与数学公式之间建议留空格
- 各种环境的开始和结束命令最好独占一行
- 分段: 建议使用一个空行
- 单个回车编译时被看成是一个空格
- 多个连续的空格编译时被看成是一个空格
- 数学公式中的标点用英文标点

中文排版

中文排版的两种处理方法:

- CJK 宏包
- `ctex` 文档类 (推荐): `ctexart`, `ctexbook`, `ctexrep`

```
1 \documentclass[a4paper]{ctexart}
2 \usepackage{amsmath,amssymb,amsfonts}
3 \begin{document}
4   欧拉公式是
5   $$ e^{ix} = \cos(x) + i\sin(x). $$
6 \end{document}
```


编译出错处理

- 若编译出错, 会在屏幕上显示问号, 并暂停编译, 等待用户输入命令
 - 回车: 继续处理
 - x: 退出
 - ?: 显示可以使用的命令
 - S: 忽略后续错误
 - R: 执行下去, 出现错误也不停留
 - Q: 不提示任何消息地执行
 - I: 插入一些内容
 - E: 编辑源文件
 - H: 显示帮助