

准备文档

1. T_EX 与 L^AT_EX

2. L^AT_EX 排版基础

3. 准备文档

4. 排版细节

5. 数学排版

6. 幻灯片制作

7. 各种模板

3. 准备文档

3.1 文档基本框架

3.2 文档类缺省设置

3.3 章节命令

3.4 标题部分

3.5 摘要与目录

3.6 参考文献

3.7 交叉引用

3.8 索引

准备文档

● 文档 (论文, 书籍) 基本框架

- 标题
- 摘要
- 目录
- 正文: 章, 节, 小节, . . .
- 附录, 参考文献, 索引

● 文档 (论文, 书籍) 基本元素

- 普通文本: 字体, 大小, 脚注
- 浮动对象: 列表, 表格, 插图
- 数学公式, 定义, 定理, 引理, ...
- 排出漂亮文档: 颜色, 链接, ...

文档类缺省设置

```
\documentclass[选项]{文档类}
```

● article 缺省设置

- 标题, 摘要, 正文 (篇, 节, 小节, ...), 参考文献; 不区分奇偶页, 没有章

● book 缺省设置

- 标题页 (书名等, 独占一页), 目录, 正文 (篇, 章, 节, ...), 参考文献, 索引
- 区分奇偶页: 奇数页也称 右页, 偶数页也称 左页, 采用不同的版式

● 自由定制页面布局 (页面宽度, 页面高度, 页眉高度, 页脚高度, 各种边距等等)

☞ 建议使用 `geometry` 宏包 (详细用法参见宏包手册)

● 定制页眉页脚: `\pagestyle`, `\thispagestyle`

☞ 高级定制: `francyhdr` 宏包 (详细用法参见宏包手册)

章节命令

```
\part  
\chapter,  
\section, \subsection, \subsubsection,  
\paragraph, \subparagraph
```

- article 类中没有 `\chapter`
- 章节命令的一般调用方法

```
\章节命令[短标题]{标题}  
\章节命令*{标题}
```

- 短标题: 用于显示在目录和页眉中, 缺省与标题相同
- 带星号的章节命令: 不参与自动编号

章节自动编号

- 每类章节都被赋予一个层次号: 层次号越小, 级别越高

层次号	章节命令	层次号	章节命令
-1	<code>\part</code>	2	<code>\subsection</code>
0	<code>\chapter</code>	3	<code>\subsubsection</code>
1	<code>\section</code>	4	<code>\paragraph</code>
		5	<code>\subparagraph</code>

- `book` 类中的自动编号的章节有:
 - `\part`, `\chapter` → 独立编号
 - `\section`, `\subsection` → 关联编号, 如 2.1, 2.2.1
(节的编号前带有章的编号, 小节的编号前面带有章和节的编号)
- `article` 类中的自动编号的章节有:
 - `\part`, `\section` → 独立编号
 - `\subsection`, `\subsubsection` → 关联编号

调整自动编号的层次

```
\setcounter{secnumdepth}{数}
```

- book 类: 取值范围 -2 至 5, 缺省为 2, 当值为 -2 时取消所有编号
- article 类: 取值范围 -1 至 5, 缺省为 3, 当值为 -1 时取消所有编号

```
\setcounter{secnumdepth}{4}
```

→ subsection 和 paragraph 都自动编号

- 自动编号的章节都有各自所对应的 计数器
 - 计数器从属关系: part → chapter → section → subsection → ...
 - 可以使用 \setcounter 来修改计数器的值

```
\setcounter{section}{3}
```

→ 将计数器 section 的值设为 3

- 每次调用章节命令时, 相应计数器自动加 1, 并同时将从属的计数器 置零

章节举例

Part I

篇 (Part) 的标题

1 节 (Section) 的标题

1.1 小节 (Subsection) 的标题

1.1.1 子节 (Subsubsection) 的标题

article

Chapter 1

章 (chapter) 的标题

1.1 节 (Section) 的标题

1.1.1 小节 (Subsection) 的标题

子节 (Subsubsection) 的标题

book

标题部分: 标题, 作者, 日期, 脚注等

```
\title{标题}  
\author{作者}  
\date{日期}  
\maketitle
```

- `\title`, `\author`, `\date` 只提供生成标题所需的信息, 并不生成标题!
- 生成标题的命令: `\maketitle` → 这个不可省略
- `\date{}` → 留空则不输出日期
如果省略 `\date` 命令, 则自动加上当前日期
- 标题部分的角注: `\thanks{...}`
必需放置在 `\title`, `\author` 和 `\date` 的 参数 中 (即 `{ }` 内), 可出现多次
- 如果标题过长, 可以用 `\\` 强制换行
- 若有多个作者, 可用 `\and` 隔开 (横排), 也可使用 `\\` 隔开 (竖排)

摘要

```
\begin{abstract}
... ..
\end{abstract}
```

- book 文档类没有 摘要

生成目录

```
\tableofcontents
```

- 设置目录中显示的章节层次: `\setcounter{tocdepth}{层次号}`
- 只有自动编号的章节才会在目录中显示 (带星号的章节不显示)
- ☞ 若需在目录中输出不自动编号的章节, 可使用下面的命令

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

章节名称 可以是 part, chapter, section, ...

第二个命令按原样输出条目内容, 且不显示页码

- 为了生成正确的目录, 有时需编译两次
- 章节标题和目录的高级定制: `titlesec` 和 `titletoc` 宏包 (详细用法参见宏包手册)

参考文献: thebibliography

```
\begin{thebibliography}{编号样本}
\bibitem[编号]{标签} 文献条目
\bibitem[编号]{标签} 文献条目
:
\end{thebibliography}
```

- “编号” → 通常省略, 系统自动按顺序编号, 如 [1], [2], ...
- “编号样本” → 指定用多大地方显示“编号”, 一般为数字, 位数等于最大编号的位数
- 标签 → 文献的 id, 可以由字母, 数字和除逗号外的符号组成
 - 每个文献的 标签 必须唯一 (互不相同)
 - 文献的引用: `\cite{标签}`, `\cite{标签1, 标签2}`
- 文献条目: 论文 (作者, 标题, 期刊, 卷期, 年代, 页码),
书籍 (作者, 书名, 出版社, 年代)
- 参考文献的高级定制: `natbib` 宏包 (详细用法参见宏包手册)

参考文献示例

```
1 \begin{thebibliography}{99}
2
3 \bibitem{AH77}
4 \newblock H. Andrews and B. Hunt,
5 \newblock \emph{Digital Image Restoration},
6 \newblock Prentice-Hall, Englewood Cliffs, NJ, 1977.
7
8 \bibitem{KS95}
9 \newblock T. Kailath and A. H. Sayed,
10 \newblock Displacement Structure: Theory And Applications,
11 \newblock \emph{SIAM Rev.}, 37 (1995), 297--386.
```

交叉引用

- 交叉引用: 章节, 图表, 定理, 引理, 公式, 页码, 参考文献, ...
- L_AT_EX 自动跟踪所有的变动, 保证所有交叉引用与索引记录的正确性
- 相关命令
 - 设置标签: `\label{标签}`
 - 章节, 图表, 定理等的引用: `\ref{标签}`
 - 公式的引用: `\eqref{标签}`
 - 页码的引用: `\pageref{标签}`
 - 参考文献的引用: `\cite{标签}`

索引

- 索引: `makeidx` 宏包

```
\usepackage{makeidx}
```

```
\makeindex → 激活索引功能, 该命令不可缺省
```

- 创建索引条目: 在需要索引的地方插入命令 `\index{索引条目}`
 - 排印索引 → `\printindex`, 通常出现在文档最后
 - 编译过程: 先用 LaTeX 编译, 再用 `makeindex` 编译
 - 索引按 ASCII 码排序
- 多索引问题: `index` 宏包 (详细用法参见宏包手册)

排版细节

1. T_EX 与 L^AT_EX

2. L^AT_EX 排版基础

3. 准备文档

4. 排版细节

5. 数学排版

6. 幻灯片制作

7. 各种模板

4. 排版细节

4.1 排版模式

4.2 西文字体

4.3 中文字体

4.4 表格与列表

4.5 脚注

4.6 计数器

4.7 抄录, 算法, 代码

4.8 彩色显示与超链接

4.9 插图

排版模式: 文字模式与数学模式

- 文字模式
 - 段落模式: 自动分行, 分段, 分页
 - 左到右模式: 将输入字符排成一排, 不换行, 无论长短
 - 数学模式: 排版数学公式
 - 数学公式一定要放在数学环境中
-
- 文本对齐方式 (缺省为左对齐)
 - 左对齐: `\raggedright` 或使用 `flushleft` 环境
 - 右对齐: `\raggedleft` 或使用 `flushright` 环境
 - 文本居中: `\centering` 或使用 `center` 环境

西文字体属性

- 每种字体都包含五种属性: 编码 (code), 族, 系列, 形状, 大小
- 族 (family): 字体的概观样式
 - `\rmfamily`: 罗马 (roman) 字体
 - `\sffamily`: 无衬线 (sans serif) 字体
 - `\ttfamily`: 打字机 (typewriter) 字体
- 系列 (series): 字体的宽度和权重 (黑度)
 - `\mdseries`: 中等权重
 - `\bfseries`: 粗体 (**Bold face**)
- 形状 (shape): 字体的高矮和倾斜
 - `\upshape`: 直立 (upstanding) 字体
 - `\itshape`: 意大利斜体 (*italic*)
 - `\scshape`: 小体大写 (SMALL CAPS) 字体

字体命令

- 前面关于字体的命令都是 **声明**
 - 这些命令在遇到新的同一类命令之前一直起作用
 - 若想限制声明的作用范围, 可使用**分组** 或**环境**
- 每一个字体**声明**都有对应的带参数的**字体命令**
- 字体命令: 只对其参数中的文本起作用, 如:

`{\bfseries 这是黑体 }` \iff `\textbf{这是黑体}`

- 族: `\textrm{...}`, `\textsf{...}`, `\texttt{...}`
 - 形状: `\textup{...}`, `\textit{...}`, `\textsl{...}`, `\textsc{...}`
 - 系列: `\textmd{...}`, `\textbf{...}`
 - 采用缺省值: `\textnormal{...}`, 对应的声明为: `\normalfont`
 - 强调命令: `\emph{...}`, 对应的声明为: `\em`
- 缺省的西文字体: **中等权重, 直立的罗马字体**

字体的尺寸

- 字体的大小

- 基本尺寸: 10pt, 11pt 和 12pt, 缺省为 10pt

```
\documentclass[11pt]{article}
```

- 设定了字体的基本尺寸后, 可使用下面的声明来改变字体大小

<code>\tiny</code>	5pt	<code>\large</code>	12pt
<code>\scriptsize</code>	7pt	<code>\Large</code>	14.4pt
<code>\footnotesize</code>	8pt	<code>\LARGE</code>	17.28pt
<code>\small</code>	9pt	<code>\huge</code>	20.74pt
<code>\normalsize</code>	10pt	<code>\Huge</code>	24.88pt

- ☞ 以上尺寸是以 10pt 为基本尺寸时的大小

中文字体


● 中文字体 (ctex 文档类)

```
\heiti, \songti, \fangsong, \kaishu, \lishu, \youyuan, \yahei
```

● 中文字号 (ctex 文档类)

- 所有的西文字号命令对中文同样起作用, 如 `\huge`, `\LARGE`, `\Large`, ...
- ctex 文档类提供的字号命令: `\zihao{字号}`, 如 `\zihao{3}`, `\zihao{-4}`

0	初号 = 42 磅	3	三号 = 16 磅	6	六号 = 7.5 磅
-0	小初 = 36 磅	-3	小三 = 15 磅	-6	小六 = 6.5 磅
1	一号 = 26 磅	4	四号 = 14 磅	7	七号 = 5.5 磅
-1	小一 = 24 磅	-4	小四 = 12 磅	8	八号 = 5 磅
2	二号 = 22 磅	5	五号 = 10.5 磅		
-2	小二 = 18 磅	-5	小五 = 9 磅		

 1 英寸 = 72 磅 (pt) = 2.54cm

- 详情参见 `ctex` 宏包说明.

使用系统中的中文字体

- 使用系统中的中文字体

- 打开一个 `cmd` 窗口, 输入以下命令

```
fc-list :lang=zh-cn > zhfonts.txt
```

→ 打开文件 `zhfonts.txt` 即可看到系统中的中文字体

- 在 `tex` 文件中设置字体 (以 `STZhongsong` 华文中宋为例)

```
\setCJKfamilyfont{STSong}{STZhongsong} → 设置字体
```

```
\newcommand{\STSong}{\CJKfamily{STSong}} → 设置字体命令
```

`\STSong` 就可以作为字体命令使用了, 如 `{\STSong 这是华文中宋}`

表格环境: tabular

```
\begin{tabular}[竖向位置]{列格式}
  first line \\
  :
  last line \\
\end{tabular}
```

- **竖向位置**: 表格在竖直方向与外部文本行的相对位置, 取值有 `t` 或 `b`, 分别表示**上对齐**和**下对齐**, 缺省为**居中对齐**
- **列格式**: 用于指定各列的格式, 常用的参数有: `l`, `c`, `r`, `|`, `||`, ...
- 行与行之间用 `\\` 分隔, 每一行的列与列之间用 `&` 分隔
- 行与行之间的分界线:
 - `\hline`: 与表格同宽的水平线
 - `\cline{m-n}`: 从第 `m` 列开始到第 `n` 列结束的水平线
- 高级表格: `longtable`, `diagbox`, `colortbl`, `booktabs` 等宏包

列表环境: `itemize`, `enumerate`, `description`

`itemize` → 带相同的标签

`enumerate` → 条目标签为自动编号的符号

`description` → 人工指定各条目的标签

● `itemize` 列表环境

```
\begin{itemize}
  \item[标签] 条目内容
  ...
\end{itemize}
```

- 缺省的标签与层数有关, 分别为: ■, -, *, ·
- 也可通过选项**标签**来指定标签
- 不要标签: `\item[]`

列表环境: itemize, enumerate, description (cont.)

● enumerate 列表环境

```
\begin{enumerate}
  \item[标签] 条目内容
  ...
\end{enumerate}
```

○ 缺省标签为自动编号的符号, 与层数有关, 分别为:

- 第一层: 阿拉伯数字后跟圆点: 1. 2.
- 第二层: 圆括号包围的小写拉丁字母: (a) (b)
- 第三层: 小写罗马数字后跟圆点: i. ii.
- 第四层: 大写拉丁字母后跟圆点: A. B.

● 高级列表功能: list 环境, enumitem 宏包

脚注: footnote

`\footnote{脚注文本}` → 自动编号

`\footnote[标记]{脚注文本}` → 手工指定脚注标记

- 该命令应紧接在需要注释的文字后面, 排版后会在所在处显示一个脚注标记, 同时将脚注内容显示在当前页的底部, 并带有相同的脚注标记
- 脚注标记通常为上标形式的数字, 并自动编号

计数器

所有自动编号的条目都有相应的计数器, 如

```
part, chapter, ..., footnote, page, equation, ...
```

- 计数器的关联 (确定从属关系, 该命令需放在导言区)

```
\numberwithin{从计数器}{主计数器}
```

```
1 \numberwithin{equation}{section}}
```

- 高级功能

定义新的计数器, 获取计数器的值, 设置显示方式, 修改计数器等, 参见相关资料

抄录环境: verbatim

```
\begin{verbatim}
... ..
\end{verbatim}
```

```
\begin{verbatim*}
... ..
\end{verbatim*}
```

- 抄录环境: 按原样输出文本, 包括所有 L_AT_EX 命令
- 带星号的环境用 `\` 显示空格

 功能更丰富的抄录: `fancyvrb` 宏包, 提供 `Verbatim` 环境

算法与代码

`\usepackage{algorithm}` → 算法

`\usepackage{algpseudocode}` → 伪代码

`\usepackage{listings}` → 各种语言的代码

```
1 \begin{algorithm}
2 \caption{LU Decomposition}
3 \begin{algorithmic}[1]
4   \For{$k=1$ to $n-1$}
5     \For{$i=k+1$ to $n$}
6       \State  $a_{ik} = a_{ik}/a_{kk}$ 
7       \For{$j=k+1$ to $n$}
8         \State  $a_{ij} = a_{ij} - a_{ik}a_{kj}$ 
9       \EndFor
10    \EndFor
11  \EndFor
12 \end{algorithmic}
13 \end{algorithm}
```

颜色: xcolor 宏包

```
\usepackage{xcolor}
```

- 使用彩色

```
\color{颜色名} → 声明, 对后面的所有文本起作用
```




















```
\textcolor{颜色名}{对象} → 命令
```

- 页面背景色: `\pagecolor{颜色名}`

- 彩色盒子: `\colorbox{颜色名}{对象}`

- 带框的彩色盒子: `\fcolorbox{边框颜色}{背景颜色}{对象}`

● 可以直接使用的颜色

 <i>black</i>	 <i>darkgray</i>	 <i>lime</i>	 <i>pink</i>	 <i>violet</i>
 <i>blue</i>	 <i>gray</i>	 <i>magenta</i>	 <i>purple</i>	 <i>white</i>
 <i>brown</i>	 <i>green</i>	 <i>olive</i>	 <i>red</i>	 <i>yellow</i>
 <i>cyan</i>	 <i>lightgray</i>	 <i>orange</i>	 <i>teal</i>	

● 更多颜色 (需加 `dvipsnames` 选项)

 <i>Apricot</i>	 <i>Cyan</i>	 <i>Mahogany</i>	 <i>ProcessBlue</i>	 <i>SpringGreen</i>
 <i>Aquamarine</i>	 <i>Dandelion</i>	 <i>Maroon</i>	 <i>Purple</i>	 <i>Tan</i>
 <i>Bittersweet</i>	 <i>DarkOrchid</i>	 <i>Melon</i>	 <i>RawSienna</i>	 <i>TealBlue</i>
 <i>Black</i>	 <i>Emerald</i>	 <i>MidnightBlue</i>	 <i>Red</i>	 <i>Thistle</i>
 <i>Blue</i>	 <i>ForestGreen</i>	 <i>Mulberry</i>	 <i>RedOrange</i>	 <i>Turquoise</i>
 <i>BlueGreen</i>	 <i>Fuchsia</i>	 <i>NavyBlue</i>	 <i>RedViolet</i>	 <i>Violet</i>
 <i>BlueViolet</i>	 <i>Goldenrod</i>	 <i>OliveGreen</i>	 <i>Rhodamine</i>	 <i>VioletRed</i>
 <i>BrickRed</i>	 <i>Gray</i>	 <i>Orange</i>	 <i>RoyalBlue</i>	 <i>White</i>
 <i>Brown</i>	 <i>Green</i>	 <i>OrangeRed</i>	 <i>RoyalPurple</i>	 <i>WildStrawberry</i>
 <i>BurntOrange</i>	 <i>GreenYellow</i>	 <i>Orchid</i>	 <i>RubineRed</i>	 <i>Yellow</i>
 <i>CadetBlue</i>	 <i>JungleGreen</i>	 <i>Peach</i>	 <i>Salmon</i>	 <i>YellowGreen</i>
 <i>CarnationPink</i>	 <i>Lavender</i>	 <i>Periwinkle</i>	 <i>SeaGreen</i>	 <i>YellowOrange</i>
 <i>Cerulean</i>	 <i>LimeGreen</i>	 <i>PineGreen</i>	 <i>Sepia</i>	
 <i>CornflowerBlue</i>	 <i>Magenta</i>	 <i>Plum</i>	 <i>SkyBlue</i>	

☞ 更多用法参见 `xcolor` 宏包手册

超链接: hyperref 宏包

```
\usepackage[选项列表]{hyperref}
```

- 在有交叉引用的地方 (如目录, 书签, 参考文献, 公式等) 建立链接
- 提供对外部文件, 互联网网址, 邮件地址的链接
- 常用选项 (也可通过 `\hypersetup{选项列表}` 来设置)
 - `bookmarks` → 创建书签, 缺省为 `true`
 - `CJKbookmarks` → 支持中日韩文字的书签
 - `colorlinks` → 使用彩色显示链接, 缺省为红色方框
 - `linkcolor` → 内部普通链接 (如页码) 的颜色, 缺省为 `red`
 - `citecolor` → 文献引用链接的颜色, 缺省为 `green`
 - `urlcolor` → URL 链接的颜色, 缺省为 `magenta`
 - `breaklinks` → 允许在链接中断行, 缺省不允许

📖 更多选项参见宏包手册

举例

```
1 \usepackage{hyperref}
2 \hypersetup{colorlinks=true, % 使用彩色链接
3     citecolor=blue, % 引用标记颜色
4     linkcolor=blue, % 内部普通链接的颜色
5     urlcolor=blue, % url 链接的颜色
6     breaklinks=true } % 允许在链接处换行
```

- 创建网页链接: `\url` 和 `\href`

```
\url{网址}
\href{网址}{文本}
```

- `\url` → 生成 [网址](#) 的同时在页面上输出其内容
- `\href` → 生成 [网址](#) 的同时在页面上输出 [文本](#) 的内容

插图: graphicx 宏包

```
\usepackage{graphicx}
\includegraphics[选项]{图形文件名}
```

- 支持的图片格式: pdf, jpg, png (pdfLaTeX 和 xeLaTeX 编译)
 - eps 格式的图片 → epstopdf 宏包, 自动将 eps 转换为 pdf
- 图形文件名 中可以含路径
- 常用 选项 有
 - width, height → 指定图形的宽度和高度 (若只指定宽度或高度, 则按比例缩放)
 - scale → 缩放因子, 如 scale=0.8

```
1 \includegraphics[scale=0.2]{tiger.png}
2 \includegraphics[width=2.5cm]{tiger.png}
3 \includegraphics[width=0.3\textwidth]{tiger.png}
```

旋转和缩放

```
\scalebox{横向缩放因子}[竖向缩放因子]{对象}
```

```
\resizebox{宽度}{高度}{对象}
```

```
\rotatebox[选项]{角度}{对象}
```

- 对象可以是任意盒子 (普通文本, 表格, 图片等)
- `\scalebox`: 若省略 竖向缩放因子, 则表示与横向缩放因子相同
- `\resizebox`: 将对象缩放到指定的大小
- `\rotatebox`: 按逆时针旋转指定角度, 可在选项中指出参考点

```
\rotatebox{45}{\includegraphics[scale=0.1]{tiger.png}}
```



浮动图表: figure, table

```
\begin{figure}[位置]  
    ...  
\end{figure}
```

```
\begin{table}[位置]  
    ...  
\end{table}
```

- 浮动图表: 自动调整图表位置, 避免出现大片的空白
- 位置选项的取值: **h** → here, **t** → top, **b** → bottom, **p** → page
- 优先顺序: **h** → **t** → **b** → **p**
- 缺省值为 **tbp**
- 固定在当前位置: **H** → `\usepackage{float}`

自动编号与图表目录

- 添加标题, 并自动编号 (只有 **浮动图表** 才能自动编号)

```
\caption[短标题]{标题}
```

- 插图目录和表格目录

```
\listoffigures → 打印插图目录
```

```
\listoftables → 打印表格目录
```

- 通常放在 `\tableofcontents` 后面

- 高级功能: `float`, `caption`, `subfigure` 宏包

文档的分割处理: `\input` 与 `\include`

- 文档的分割处理: 将大文件分成几个小文件
 - 主文件 + 子文件
- `\input` 命令: `\input{文件名}`
 - 将指定的文件读入到当前位置, 再进行编译
 - 可以出现在文档的任何部分 (导言区或正文); 可以嵌套
- `\include` 命令: `\include{文件名}`
 - 相当于 `\clearpage\input{文件名}\clearpage`
 - 只能读入 tex 文件; 只能出现在正文部分; 不能嵌套
 - 优点: 可与 `\includeonly{文件清单}` 配合使用, 实现部分编译
 - 编译时, 只读入 `\includeonly` 指定的文件清单
 - `\includeonly` 只能放在导言区
 - 若没有 `\includeonly` 命令, 则读入所有文件

```
1 \documentclass[a4paper]{ctexbook}
2 \input{format.tex}
3
4 \begin{document}
5 \include{cover.tex}
6
7 \frontmatter
8 \tableofcontents
9 \include{preface.tex}
10
11 \mainmatter
12 \include{chap01.tex}
13 \include{chap02.tex}
14   ... ..
15
16 \backmatter
17 \include{reference.tex}
18
19 \end{document}
```