

L^AT_EX Learning Notes

Casea

2022/10/20

摘要

This is a Latex learning code!

Following Chen's github repository to learn how to write elegant L^AT_EX.

目录

| | | |
|-----------|----------------------|-----------|
| I | 代码结构 | 10 |
| 1 | 代码结构 | 10 |
| 1.1 | 列表 | 10 |
| 1.2 | 换行 | 10 |
| 1.3 | 颜色 | 11 |
| 1.4 | 纸张方向 | 11 |
| 1.5 | 段落 | 11 |
| 1.6 | 字体 | 11 |
| II | 文本编辑 | 13 |
| 1 | 标题作者日期摘要 | 13 |
| 2 | 创建章节 | 13 |
| 2.1 | 设置章节自动编号深度 | 13 |
| 2.2 | 改变字体样式 | 13 |
| 2.3 | 标题居中 | 14 |
| 3 | 生成目录 | 14 |
| 3.1 | 层次别名 | 14 |
| 3.2 | 引用链接 | 14 |
| 4 | 编辑段落 | 14 |
| 4.1 | 段落首行缩进 | 14 |
| 4.2 | 段落间距调整 | 14 |
| 4.3 | 段落添加文本框 | 15 |
| 4.4 | 段落对齐方式 | 15 |
| 5 | 文本编辑 | 15 |
| 5.1 | 调整字体样式 | 15 |
| 5.2 | 调整字体大小 | 15 |
| 5.3 | 调整字体颜色 | 16 |
| 5.4 | 字体设置 | 17 |
| 5.5 | 下划线与删除线 | 18 |
| 5.6 | 特殊字符 | 19 |

| | | |
|------------|-----------------------|-----------|
| 6 | 创建列表 | 20 |
| 6.1 | 无序列表 | 21 |
| 6.2 | 排序列表 | 23 |
| 6.3 | 阐述性列表 | 24 |
| 6.4 | 自定义列表格式 | 25 |
| 6.4.1 | 垂直间距 | 25 |
| 6.4.2 | 水平间距 | 25 |
| 7 | 创建页眉页脚注 | 26 |
| | | |
| III | 公式编辑 | 29 |
| | | |
| 1 | 基本介绍 | 29 |
| 1.1 | 数学公式环境 | 29 |
| 1.1.1 | 美元符号 | 29 |
| 1.1.2 | equation 环境 | 30 |
| 1.1.3 | align 环境 | 30 |
| 1.1.4 | gather 环境 | 31 |
| 1.2 | 基本格式调整 | 32 |
| 1.2.1 | 字符类型 | 32 |
| 1.2.2 | 调整公式大小 | 33 |
| 1.2.3 | 其他格式调整 | 35 |
| 1.3 | 练习题 | 36 |
| | | |
| 2 | 常用运算符号 | 38 |
| 2.1 | 运算符号 | 38 |
| 2.2 | 标记符号 | 39 |
| 2.3 | 各类括号 | 42 |
| 2.4 | 空心符号 | 45 |
| 2.5 | 特殊函数 | 46 |
| | | |
| 3 | 希腊字母 | 47 |
| | | |
| 4 | 微积分 | 49 |
| 4.1 | 极限 | 49 |
| 4.2 | 导数 | 50 |
| 4.3 | 积分 | 52 |
| 4.4 | 练习题 | 54 |

| | | |
|-----------|------------------------------------------------|-----------|
| 5 | 线性代数 | 55 |
| 5.1 | 矩阵 | 55 |
| 5.2 | 符号 | 58 |
| 5.3 | 范数 | 61 |
| 5.4 | 练习题 | 63 |
| 6 | 概率论与数理统计 | 64 |
| 6.1 | 概率论基础 | 64 |
| 6.2 | 概率分布 | 65 |
| 6.3 | 练习题 | 68 |
| 6.4 | 优化理论 | 68 |
| 6.5 | 练习题 | 71 |
| IV | 表格制作 | 72 |
| 1 | 基本介绍 | 72 |
| 1.1 | <code>tabular</code> 环境: 创建表格内容 | 72 |
| 1.2 | <code>table</code> 环境: 自动编号与浮动表格 | 74 |
| 2 | 合并单元格 | 76 |
| 2.1 | 合并不同列的单元格 | 76 |
| 2.2 | 合并不同行的单元格 | 77 |
| 2.3 | 合并多行多列的单元格 | 78 |
| 3 | 插入斜线与标注 | 78 |
| 3.1 | 插入斜线 | 78 |
| 3.2 | 插入表注 | 80 |
| 4 | 调整表格样式 | 81 |
| 4.1 | 表格尺寸 | 81 |
| 4.2 | 单元格自动对齐与换行 | 82 |
| 4.2.1 | 使用 <code>array</code> 宏包实现单元格自动对齐与换行 | 82 |
| 4.2.2 | 使用 <code>tabularx</code> 宏包实现自动换行 | 84 |
| 4.2.3 | 使用 <code>tabulary</code> 宏包实现自动换行 | 85 |
| 4.2.4 | 使用 <code>\parbox</code> 命令实现人工换行 | 86 |
| 4.3 | 小数点对齐 | 86 |
| 4.4 | 行高 | 87 |
| 4.5 | 列宽 | 89 |
| 4.6 | 线宽 | 89 |

| | | |
|-----------|-----------------------------------|------------|
| 4.7 | 表格字体大小 | 90 |
| 4.8 | 文字环绕表格 | 92 |
| 5 | 创建彩色表格 | 94 |
| 6 | 创建三线表格 | 96 |
| 7 | 创建跨页表格 | 97 |
| 8 | 旋转表格 | 100 |
| 9 | 导入表格 | 102 |
| 9.1 | 快速创建表格 | 102 |
| 9.2 | 创建三线表 | 103 |
| 9.3 | 设置表格属性 | 103 |
| V | 图形插入 | 106 |
| 1 | 插入浮动图片 | 106 |
| 2 | 插入非浮动图片 | 108 |
| 3 | 插入图表目录 | 109 |
| 4 | 定制图表标题样式 | 110 |
| 4.1 | 调整标题头部 | 110 |
| 4.2 | 调整编号 | 111 |
| 4.3 | 调整编号分隔符 | 112 |
| 5 | 插入子图 | 113 |
| 5.1 | 基本介绍 | 113 |
| 5.2 | 调整子图间距 | 115 |
| VI | 图形绘制 | 119 |
| 1 | 基本介绍 | 119 |
| 1.1 | 使用 tikzpicture 环境创建图形元素 | 119 |
| 1.2 | 绘制直线 | 120 |
| 1.3 | 图形缩放 | 122 |
| 1.4 | 绘制箭头 | 123 |

| | | |
|-------------|------------------|------------|
| 1.5 | 调整线条粗细 | 124 |
| 1.6 | 虚线 | 126 |
| 1.7 | 颜色 | 126 |
| 2 | 节点介绍 | 127 |
| 2.1 | 节点基本介绍 | 127 |
| 2.2 | 节点样式 | 128 |
| 2.3 | 节点命名 | 129 |
| 2.4 | 基于相对位置绘制节点 | 129 |
| 2.5 | 连接节点 | 130 |
| 3 | 高级功能 | 132 |
| 3.1 | 矩形、圆形、曲线 | 132 |
| 3.2 | 平滑过渡曲线 | 133 |
| 3.3 | 根据函数绘制曲线 | 134 |
| 3.4 | 简单图形的区域填充 | 136 |
| 3.5 | 在图形中填写标签 | 137 |
| 3.6 | 复杂模型实战解析 | 139 |
| VII | 建立索引及文献引用 | 145 |
| 1 | 图表和公式的索引 | 145 |
| 1.1 | 公式的索引 | 145 |
| 1.2 | 图形的索引 | 146 |
| 1.3 | 表格的索引 | 146 |
| 2 | 创建超链接 | 147 |
| 2.1 | 超链接类型 | 147 |
| 2.1.1 | 文本内链接 | 147 |
| 2.1.2 | 网址链接 | 148 |
| 2.1.3 | 本地文件链接 | 149 |
| 2.2 | 超链接格式 | 149 |
| 3 | Bibtex 用法 | 150 |
| 3.1 | 创建参考文献 | 150 |
| 参考文献 | | 151 |
| 3.2 | 使用 BibTeX 文件 | 152 |

| | |
|---------------------------|----------------|
| 4 文献引用格式 | 154 |
| VIII 幻灯片制作 | 155 |
| 1 基本介绍 | 156 |
| 1.1 Beamer 简介 | 156 |
| 1.2 创建幻灯片 | 158 |
| 1.3 创建章节与生成目录 | 159 |
| 1.4 幻灯片分栏 | 163 |
| 2 添加动画效果 | 165 |
| 2.1 \pause 命令 | 165 |
| 2.2 \item<> 命令 | 166 |
| 2.3 其他命令 | 168 |
| 2.4 自动计数 | 170 |
| 3 创建文本框 | 172 |
| 3.1 区块环境 | 172 |
| 3.2 定理类环境 | 173 |
| 3.3 Beamer 中的盒子 | 174 |
| 4 设置主题样式 | 179 |
| 4.1 基本介绍 | 179 |
| 4.2 颜色主题 | 180 |
| 4.3 字体主题 | 181 |
| 4.4 内部主题 | 182 |
| 4.5 外部主题 | 183 |
| 4.6 表格字体大小 | 183 |
| 4.7 样式调整 | 184 |
| 5 插入程序源代码 | 186 |
| 6 添加参考文献 | 188 |
| 7 插入表格 | 191 |
| 8 插入与调整图片 | 192 |
| 8.1 插入图片 | 192 |
| 8.2 编辑图片 | 193 |
| 8.3 插入子图 | 194 |

| | | |
|-------|---------------------|-----|
| 8.4 | 调整排列方式 | 195 |
| 8.4.1 | 横向分布与纵向分布 | 195 |
| 8.4.2 | 并排显示 | 196 |
| 8.4.3 | 设置任意图片位置 | 197 |
| 8.5 | 调整布局方式 | 200 |
| 8.5.1 | 文字环绕图片 | 200 |
| 8.5.2 | 文字浮于图片上方 | 201 |
| 8.6 | 设置背景图片 | 201 |
| 8.7 | 在标题页中插入图片 | 202 |
| 8.8 | 添加动画效果 | 204 |

IX LaTeX 进阶 205

| | | |
|-----|-------------------------------|-----|
| 1 | 添加程序源代码 | 205 |
| 1.1 | 使用 verbatim 插入程序源代码 | 205 |
| 1.2 | 使用 listings 插入程序源代码 | 206 |
| 2 | 算法伪代码 | 208 |
| 3 | 简历制作 | 209 |
| 3.1 | 使用 article 文档类型制作简历 | 209 |
| 3.2 | 自定义简历格式 | 209 |

Part I

代码结构

1 代码结构

1.1 列表

比较常用的三种文档类型, 包括 `article` (常规文档)、`report` (报告)、`book` (书籍), 其中, `report` 和 `book` 这两种文档类型的文档结构是一致的, 可以使用的结构命令有 `\part{}`、`\chapter{}`、`\section{}`、`\subsection{}`、`\subsubsection{}`、`\paragraph{}`、`\subparagraph{}`, 而 `article` 文档类型中除了没有 `\chapter{}` 这一结构命令之外, 其他都与 `report` 和 `book` 文档类型是一样的。©

```
\begin{itemize}
\item LaTeX is good
\item LaTeX is convenient
\end{itemize}
```

1.2 换行

换行 ‘\’ 只换行无缩进
‘\par’ 自动换行加缩进
使用 `ctex` 包显示中文
需使用间距宏包 `setspace`
行间距设置为 1.3 倍
par 1

```
\begin{spacing}{1.3}
换行 ‘\textbackslash\textbackslash’ 只换行无缩进 \par
‘\textbackslash par’ 自动换行加缩进 \par
使用ctex包显示中文 \par
需使用间距宏包setspace \par
行间距设置为1.3倍 \par
par 1 \par
par 2 \par
\end{spacing}
```

1.3 颜色

Hello, LaTeXers! This is first LaTeX document..

```
\textcolor[rgb]{1,0,0}{Hello , LaTeXers! This is our first LaTeX document.}.
```

1.4 纸张方向

```
\begin{landscape}
  \textcolor[rgb]{1,0,0.5}{Hello , LaTeXers! This is our first LaTeX
    document.}.
\end{landscape}
```

1.5 段落

PA Hello, LaTeXers! This is our first LaTeX document.

pa1 Hello, LaTeXers! This is our first LaTeX document.

Hello, LaTeXers! This is our first LaTeX document.

```
Hello , LaTeXers! This is our first LaTeX document.
\subparagraph{pa1}
Hello , LaTeXers! This is our first LaTeX document.
\subparagraph{}
Hello , LaTeXers! This is our first LaTeX document.
```

1.6 字体

【楷书】

永和九年，岁在癸丑，暮春之初，会于会稽山阴之兰亭，修禊（禊）事也。群贤毕至，少长咸集。此地有崇山峻领（岭），茂林修竹；又有清流激湍，映带左右，引以为流觞曲水，列坐其次。虽无丝竹管弦之盛，一觴一咏，亦足以畅叙幽情。

【黑体】

永和九年，岁在癸丑，暮春之初，会于会稽山阴之兰亭，修禊（禊）事也。群贤毕至，少长咸集。此地有崇山峻领（岭），茂林修竹；又有清流激湍，映带左右，引以为流觞曲水，列坐其次。虽无丝竹管弦之盛，一觴一咏，亦足以畅叙幽情。

{\kaishu 【楷书】

永和九年，岁在癸丑，暮春之初，会于会稽山阴之兰亭，修禊（禊）事也。群贤毕至，少长咸集。此地有崇山峻领（岭），茂林修竹；又有清流激湍，映带左右，引以为流觞曲水，列坐其次。虽无丝竹管弦之盛，一觴一咏，亦足以畅叙幽情。}

{\heiti 【黑体】

永和九年，岁在癸丑，暮春之初，会于会稽山阴之兰亭，修禊（禊）事也。群贤毕至，少长咸集。此地有崇山峻领（岭），茂林修竹；又有清流激湍，映带左右，引以为流觞曲水，列坐其次。虽无丝竹管弦之盛，一觴一咏，亦足以畅叙幽情。}

Part II

文本编辑

1 标题作者日期摘要

摘要

- 使用 `\title{}` 命令设置标题
对于较长的文档标题可以使用 `\\`
对标题内容进行分行
- 使用 `\author{}` 设置作者
作者之间使用 `\and` 进行分隔
- 使用 `\date{}` 设置日期
`{}` 不写任何信息表示默认使用当前日期
- 使用 `\maketitle` 创建标题
- 使用 `\begin{abstract}` 撰写摘要
并在其后使用 `\textbf{}` 设置文档关键词

Keywords: 标题, 作者, 日期, 摘要

2 创建章节

2.1 设置章节自动编号深度

用户也可以通过在导言区使用 `\setcounter{secnumdepth}{}` 计数器命令设置章节自动编号深度, 从而达到批量取消自动编号的效果。在 `{}` 中填写编号深度值, 编号深度值从 0 开始设置,

- `\setcounter{secnumdepth}{0}` 表示自动编号章节层次仅包括 `\part` 和 `chapter`;
- `\setcounter{secnumdepth}{1}` 表示自动编号章节层次深入到 `\section` 级;
- `\setcounter{secnumdepth}{2}` 表示自动编号章节层次深入到 `\subsection` 级;
- `\setcounter{secnumdepth}{3}` 表示自动编号章节层次深入到 `\subsubsection` 级;

2.2 改变字体样式

改变字体样式需要使用 `titlesec` 宏包, 使用宏包命令 `\titleformat*{}{}{}`

2.3 标题居中

调用 `sectsty` 宏包，并用 `\sectionfont {\centering}`

3 生成目录

章节目录一般创建在摘要之后，在 L^AT_EX 中，使用 `\tableofcontents` 命令即可。命令放在哪里，就会在哪里自动创建一个章节目录。默认情况下，该命令会根据用户定义的篇章标题生成文章目录，目录中将包含 `\subsubsection` 及其更高层次的结构标题。但对于带星号的章节命令，其章节标题不会出现在目录中。

3.1 调节章节层次深度

在导言区使用 `\setcounter{tocdepth}{}` 命令指定目录中的章节层次深度我们也可以为每个章节设置不同的目录层次，具体是通过在每个章节创建命令前，使用 `\addtocontents{toc}{\setcounter{tocdepth}{}}` 命令为该章节指定目录层次深度。

3.2 引用链接

如果想要为目录中的章节引用添加链接，使得点击链接后就能跳转到相应章节所在页面，那么只需要在导言区调用 `hyperref` 宏包、并设置 `colorlinks=true` 选项即可，此时文档中章节引用及其它交叉引用均会被自动添加链接（添加了链接的引用将显示为红色）。

4 编辑段落

4.1 段落首行缩进

若想调整段落首行缩进的距离，可以使用 `\setlength{\parindent}{长度}` 命令，在 {长度} 处填写需要设置的距离即可。

如果不想让段落自动首行缩进，在段落前使用命令 `\noindent` 即可。

在段落设置在章节后面时，每一节后的第一段默认是不缩进的，为了使第一段向其他段一样缩进，可以在段落前使用 `\hspace*{\parindent}` 命令，也可以在源文件的前导代码中直接调用宏包 `\usepackage{indentfirst}`。

4.2 段落间距调整

LaTeX 排版时，有时为了使段落与段落之间的区别更加明显，我们可以在段落之间设置一定的间距，最简单的方式是使用 `\smallskip`，`\medskip`，`\bigskip`，等命令

4.3 段落添加文本框

有时因为文档全都是大段大段的文字，版面显得较为单调，这时，我们可以通过给文字加边框来让版面有所变化，不至于过于单调。在 LaTeX 中，我们可以使用 `\fbox{}` 命令对文本新增边框

4.4 段落对齐方式

L^AT_EX 默认的对齐方式是两端对齐，

有时在进行文档排版的过程中，我们为了突出某一段落的内容，会选择将其居中显示，在 LaTeX 中，我们可以使用 `center` 环境对文本进行居中对齐。

另外还有一些出版商要求文档是左对齐或者右对齐，

另外还有一些出版商要求文档是左对齐或者右对齐，这时我们同样可以使用 `flushleft` 环境和 `flushright` 环境将文档设置为左对齐或右对齐

5 文本编辑

文本编辑是制作文档非常重要的一部分，在编辑文本的过程中，主要关注的内容有如何调整字体样式、字体设置、增加下划线、突出文字、调整字体大小、调整对齐格式等。

5.1 调整字体样式

调整文字的样式有很多对应的命令，这些命令包括 `\textit`、`\textbf`、`\textsc`、`\texttt`，在使用的过程中，需要用到花括号 `{}`。具体而言，`\textit` 对应着斜体字，`\textbf` 对应着**粗体字**，`\textsc` 对应着小型大写字母，`\texttt` 对应着打印机字体（即等宽字体）。

如果想对文本中的英文字母进行全部大写，可用 `\uppercase` 和 `\MakeUppercase` 两个命令中的任意一个。

USE UPPERCASE COMMAND TO FORCE ALL UPPERCASE. USE MAKEUPPERCASE COMMAND TO FORCE ALL UPPERCASE.

5.2 调整字体大小

字体大小的设置一方面可以在申明文档类型的命令 `\documentclass[]{}{}` 中指定具体的字体大小（如 11pt、12pt）来实现，另一方面也可以通过一些简单的命令来调整。

Produce tiny word

Produce script size word

Produce footnote size word

Produce normal size word

Produce large word

Produce Large word

Produce LARGE word

Produce huge word

Produce Huge word

代码:

Produce {\tiny tiny word}

Produce {\scriptsize script size word}

Produce {\footnotesize footnote size word}

Produce {\normalsize normal size word}

Produce {\large large word}

Produce {\Large Large word}

Produce {\LARGE LARGE word}

Produce {\huge huge word}

Produce {\Huge Huge word}

Produce large word

Produce large word

Produce large word

Produce large word

Produce large word

Produce large word

代码:

Produce \begin{large}large word\end{large}

Produce \begin{Large}large word\end{Large}

Produce \begin{LARGE}large word\end{LARGE}

5.3 调整字体颜色

一般而言, 文本默认的颜色是黑色, 但有时候, 我们可以根据需求改变字体的颜色, 这通过 L^AT_EX 一些拓展的宏包就可以实现, 例如 `xcolor`

使用颜色宏包时, 我们也可以根据需求自定义颜色, 相应的命令为

`\definecolor{A}{B}{C}`, 其中, 位置 A 是颜色标签, 位置 B 是制定颜色系统为 RGB (英文缩写 RGB 是红色、绿色和蓝色三种颜色的英文单词首字母), 位置 C 是具体的 RGB 数值。

This is a simple example for using L^AT_EX.

This is a simple example for using L^AT_EX.

This is a simple example for using L^AT_EX.

This is a simple example for using L^AT_EX.

This is a simple example for using L^AT_EX.

代码

```
\documentclass[12pt]{article}
\usepackage{color}
\definecolor{kugreen}{RGB}{50, 93, 61}
\definecolor{kugreenlys}{RGB}{132, 158, 139}
\definecolor{kugreenlyslys}{RGB}{173, 190, 177}
\definecolor{kugreenlyslyslys}{RGB}{214, 223, 216}
\begin{document}
This is a simple example for using \LaTeX.
{\color{kugreen}This is a simple example for using \LaTeX.}
{\color{kugreenlys}This is a simple example for using \LaTeX.}
{\color{kugreenlyslys}This is a simple example for using \LaTeX.}
{\color{kugreenlyslyslys}This is a simple example for using \LaTeX.}
\end{document}
```

5.4 字体设置

不管是英文还是中文，我们都会用到各种各样的字体，因此，使用 LaTeX 编译出想要的字体对于整个文档也非常重要。对于英文文档的编译，一般会用宏包 `fontspec` 设置具体的字体，调用格式为 `\usepackage{fontspec}`，需要说明的是，这个宏包只能设置英文的字体

```
\setmainfont{Times New Roman}
\setsansfont{DejaVu Sans}
\setmonofont{Latin Modern Mono}
\setsansfont{[foo.ttf]}
```

在 LaTeX 中，编译文档一般默认的英文字体为 `Computer Modern`，如果要将其调整为其他特定类型的字体，可以在前导代码中使用各种字体对应的工具包。

```
\documentclass[a4paper, 12pt]{article}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\begin{document}
Hello, LaTeXers! This is our first LaTeX document.
\end{document}
```

如果文档输入的是中文，首先需要申明文档类型为 ctex 中的 ctexart、ctexrep 之类的。Part I 6 节介绍了部分中文字体的设置方式，宋体、仿宋、隶书、黑体、楷体、幼圆、微软雅黑等字体为目前常用的大部分中文字体，下例将使用 ctexart 给出它们的设置方式。

字体（默认宋体）

字体（仿宋） 字体（宋体） **字体（黑体）**

字体（楷书） 字体（幼圆） 字体（微软雅黑）

代码：

```
\documentclass[12pt,a4paper,utf8]{ctexart}
\begin{document}
\noindent 字体（默认宋体）\\
\fangsong 字体（仿宋） \songti 字体（宋体） \lishu 字体（隶书） \heiti 字体
（黑体）\\
\CJKfamily{zhkai} 字体（楷书） \CJKfamily{zhyou} 字体（幼圆） \CJKfamily{
zhyahai} 字体（微软雅黑）\\
\end{document}
```

5.5 下划线与删除线

有时候，为了突出特定的文本，我们也会使用到各种下划线。最常用的下划线命令是 `\underline`，然而，这个命令存在一个缺陷，即当文本过长，超过页面宽度时，下划线不会自动跳到下一行，因此，我们需要用到一个叫 `ulem` 的宏包，使用这个宏包中的命令 `\uline` 可以增加单下划线，使用 `\uuline` 可以增加双下划线，而使用 `\uwave` 则可以增加波浪线

Generate underlined text.

Generate underlined text.

Generate underlined text.

Generate underlined text.

Generate ~~stridethrough~~ text.

Generate emphasized text.

Generate “quote” text

代码:

```
\documentclass[12pt]{article}
\usepackage{ulem}
\begin{document}
Generate \underline{underlined} text. \\
% 生成带下划线的文本 (使用\underline命令)
Generate \uline{underlined} text. \\
% 生成单下划线的文本 (使用\uline命令)
Generate \uuline{double underlined} text. \\
% 生成单下划线的文本
Generate \uwave{wavy underlined} text. \\
% 生成波浪线的文本
Generate \st{striketrough} text.
\end{document}
```

5.6 特殊字符

在 L^AT_EX 中, 有很多特殊字符的编译需要遵循一定的规则, 例如, 反斜杠 (backslash) 符号是 LaTeX 中定义和使用各类命令的基本符号, 如果要在文档中编译出反斜杠, 可使用 `\textbackslash` 百分号通常用于注释代码, 如果要在文档中编译出百分号, 可使用 `\%` 美元符号通常用于书写公式, 如果要在文档中编译出美元符号, 可使用 `\$`。带圆圈数字可用于各类编号, 我们可以根据需要插入这种特殊符号。在 L^AT_EX 中, 比较常用的一种生成带圆圈数字的方法是使用宏包 `pifont`, 在前导代码中申明使用宏包, 即 `\usepackage{pifont}`, 根据宏包所提供的命令 `\ding{}` 可以生成从 1 到 10 的带圆圈数字, 且圆圈样式也各异。How to write a number in a circle?

Type 1: ①-②-③-④-⑤-⑥-⑦-⑧-⑨-⑩

Type 2: ➊-➋-➌-➍-➎-➏-➐-➑-➒-➓

Type 3: ①-②-③-④-⑤-⑥-⑦-⑧-⑨-⑩

Type 4: ➊-➋-➌-➍-➎-➏-➐-➑-➒-➓

```

\documentclass[12pt]{article}
\usepackage{pifont}
\begin{document}

How to write a number in a circle? \\
Type 1: \ding{172}-\ding{173}-\ding{174}-\ding{175}-\ding{176}-\ding{177}-\ding{178}-\ding{179}-\ding{180}-\ding{181} \\ % 样式1是从172开始
Type 2: \ding{182}-\ding{183}-\ding{184}-\ding{185}-\ding{186}-\ding{187}-\ding{188}-\ding{189}-\ding{190}-\ding{191} \\ % 样式2是从182开始
Type 3: \ding{192}-\ding{193}-\ding{194}-\ding{195}-\ding{196}-\ding{197}-\ding{198}-\ding{199}-\ding{200}-\ding{201} \\ % 样式3是从192开始
Type 4: \ding{202}-\ding{203}-\ding{204}-\ding{205}-\ding{206}-\ding{207}-\ding{208}-\ding{209}-\ding{210}-\ding{211} \\ % 样式4是从202开始

\end{document}

```

6 创建列表

在内容表达上，列表是一种非常有效的方式，它将某一论述内容分成若干个条目进行罗列，能达到简明扼要、醒目直观的表达效果。在论文写作中，列表不失为一种让内容清晰明了的论述方式。按层次分，列表有单层列表和多层列表，多层列表无外乎是最外层列表中嵌套了一层甚至更多层列表。按类型分，列表主要有三种类型，即无序列表、排序列表和阐述性列表，其中，无序列表和排序列表是相对常用的列表类型，L^AT_EX 针对这三种列表提供了一些基本环境

- 无序列表使用方法

```
\begin{itemize}

  \item Item 1 % 条目 1
  \item Item 2 % 条目 2

\end{itemize}
```

- 排序列表使用方法

```
\begin{enumerate}

  \item Item 1 % 条目 1
  \item Item 2 % 条目 2

\end{enumerate}
```

- 阐述性列表

```
\begin{description}

  \item Item 1 % 条目 1
  \item Item 2 % 条目 2

\end{description}
```

6.1 无序列表

LaTeX 中的无序列表环境一般用特定符号（如圆点、星号）作为列表中每个条目的起始标志，以示有别于常规文本。可以忽略主次或者先后顺序关系的条目排列都可以使用无序列表环境来编写，无序列表时很多文档最常用的列表类型，也被称为常规列表。

```
代码
\documentclass[12pt]{article}
\begin{document}
\begin{itemize}
\item Python % 条目 1
\item LaTeX % 条目 2
\item GitHub % 条目 3
\end{itemize}
\end{document}
```

在无序列表环境中，每个条目都是以条目命令 `\item` 开头的，一般默认的起始符号是 `textbullet`，即大圆点符号，当然，也可以根据需要调整起始符号

- Python
- LaTeX
- * GitHub

```
代码
\documentclass[12pt]{article}
\begin{document}
\begin{itemize}

\item Python      % 条目1，起始符号为大圆点
\item LaTeX      % 条目2，起始符号为大圆点
\item[*] GitHub % 条目3，起始符号为星号

\end{itemize}
\end{document}
```

如果想要将所有条目的符号都进行调整，并统一为某一个特定符号，则可以使用 `\renewcommand` 命令进行自定义。其中，命令 `\labelitemi` 是由三部分组成，即 `label` (标签)、`item` (条目)、`i` (一级)，有时候，如果需要创建多级列表，则可以类似这里使用命令 `\labelitemii` (对应于二级列表) 甚至 `\labelitemiii` (对应于三级列表)。

- Python
- LaTeX
- GitHub

```
\documentclass[12pt]{article}
\usepackage{amssymb}
\begin{document}
\begin{itemize}
  \renewcommand{\labelitemi}{\scriptsize$\blacksquare$}
  \item Python % 条目1
  \item LaTeX % 条目2
  \item GitHub % 条目3
\end{itemize}
\end{document}
```

6.2 排序列表

排序列表也被称为编号列表。在排序列表中，每个条目之前都有一个标号，它是由标志和序号两部分组成：序号自上而下，从 1 开始升序排列；标志可以是括号或小圆点等符号。相互之间有密切的关联，通常是按过程顺序或是重要程度排列的条目都可以采用排序列表环境 `enumerate` 以序号作为列表的起始标志，每个条目命令 `item` 将在每个条目之前自动加上一个标号。命令 `item` 生成的默认标号样式为阿拉伯数字加小圆点。排序列表同样可以进行相互嵌套，最多可以达到四层，为了便于区分，不仅每层列表的条目段落都有不同程度的左缩进，二期每层列表中条目的标号也各不相同，其中序号的计数形式与条目所在的层次有关，标志所用的符号除第 2 层是圆括号外，其他各层都是小圆点。

1. pencil
2. calculator
3. rule
4. notebook
 - (a) notes
 - i. note A
 - A. note A
 - ii. note B
 - (b) homework
 - (c) assessments

代码

```
\documentclass[12pt]{article}
\begin{document}
\begin{enumerate}
\item pencil
\item calculator
\item ruler
\item notebook
    \begin{enumerate}
    \item notes
        \begin{enumerate}
        \item note A
            \begin{enumerate}
            \item note a
            \end{enumerate}
        \end{enumerate}
        \item note B
    \end{enumerate}
    \item homework
    \item assessments
    \end{enumerate}
\end{enumerate}
\end{document}
```

6.3 阐述性列表

相比无序列表和排序列表，阐述性列表的使用频率较低，它常用于对一组专业术语进行解释说明。阐述性列表环境为 `description`。在 `description` 环境命令中，每个词条都是需要分别进行阐述的词语，每个阐述可以是一个或多个文本段落。这种形式很像词典，因此诸如名词解释说明之类的列表就可以采用解说列表环境来编写。在阐述性列表环境中，被解释的词条的格式是用 `descriptionlabel` 定义的。

CNN Convolutional Neural Networks

RNN Recurrent Neural Networks

CRNN Convolutional Recurrent Neural Networks

代码

```
\documentclass[12pt]{article}
\begin{document}
\begin{description}
\item [CNN] Convolutional Neural Networks
\item [RNN] Recurrent Neural Network
\item [CRNN] Convolutional Recurrent Neural Network
\end{description}
\end{document}
```

6.4 自定义列表格式

使用系统默认的 L^AT_EX 列表环境排版的列表与上下文之间以及列表条目之间都附加有一段垂直空白，明显有别于列表环境之外的文本格式，通常列表中的条目内容都很简短，这样会造成很多空白，使得列表看起来很稀疏，与前后文本之间的协调性较差。因此，我们需要自定义列表格式。使用 `enumitem` 宏包可以调整 `enumerate` 或 `itemize` 的上下左右缩进间距。

6.4.1 垂直间距

- `topsep` 列表环境与上文之间的距离
- `parsep` 条目里面段落之间的距离
- `itemsep` 条目之间的距离
- `partopsep` 条目与下面段落的距离

6.4.2 水平间距

- `leftmargin` 列表环境左边的空白长度
- `rightmargin` 列表环境右边的空白长度
- `labelsep` 标号与列表环境左侧的距离
- `itemindent` 条目的缩进距离
- `labelwidth` 标号的宽度
- `listparindent` 条目下面段落的缩进距离

Default spacing:

- Python
- LaTeX
- GitHub

Custom Spacing:

- Python
- LaTeX
- GitHub

```
使用enumitem宏包调整无序列表间距
\documentclass[12pt]{article}
\usepackage{enumitem}
\begin{document}
Default spacing:
\begin{itemize}
\item Python % 条目 1
\item LaTeX % 条目 2
\item GitHub % 条目 3
\end{itemize}
Custom Spacing:
\begin{itemize}[itemsep= 15 pt,topsep = 20 pt]
\item Python % 条目 1
\item LaTeX % 条目 2
\item GitHub % 条目 3
\end{itemize}
\end{document}
```

7 创建页眉页脚脚注

在大多数文档中，我们往往需要页眉、页脚及脚注来展示文档的附加信息，例如时间、图形、页码、日期、公司微标、页眉示意图文档标题、文件名或作者姓名等信息。在 L^AT_EX 中，我们常用 `Fancyhdr` 工具包进行页眉、页脚的设置。

```

使用Fancyhdr工具包进行页眉、页脚的设置
\documentclass{article}
\usepackage{fancyhdr}
\pagestyle{fancy}
\lhead{}
\chead{}

\rhead{\bfseries latex-cookbook} %页眉内容
\lfoot{From: Xinyu Chen} %页脚内容
\cfoot{To: Jieling Jin} %页脚内容
\rfoot{\thepage} %在页脚处给出页码
\renewcommand{\headrulewidth}{0.4pt}
\renewcommand{\footrulewidth}{0.4pt}

\begin{document}
This is latex-cookbook!
\end{document}

```

如果想某一页不需要页眉页脚,可以在该页正文内容开始时使用`\thispagestyle{empty}`命令,去除该页页眉页脚。

```

\documentclass{article}
\usepackage{fancyhdr}
\pagestyle{fancy}
\lhead{}
\chead{}

\rhead{\bfseries latex-cookbook}
\lfoot{From: Xinyu Chen}
\cfoot{To: Jieling Jin}
\rfoot{\thepage}
\renewcommand{\headrulewidth}{0.4pt}
\renewcommand{\footrulewidth}{0.4pt}

\begin{document}
This is latex-cookbook!
\newpage
\thispagestyle{empty}
This is latex-cookbook!
\end{document}

```

脚注也是一篇文档中的重要内容，一般用来解释文档中的名词或相关出处。在 L^AT_EX 中，我们常用 `\footnote{}` 命令来添加脚注²。

```
\documentclass{article}
\begin{document}
Caea \footnote{Tian jin university}
\end{document}
```

在论文撰写过程中，我们有时会需要在表格中添加脚注，但是在 table 环境中，`\footnote{}` 命令不起作用，这时我们可以使用 `\minipage{}` 环境来化解。

| CHAPTER | AUTHOR |
|---------------------|--------------------------|
| <i>Introduction</i> | Xinyu Chen ^a |
| <i>Methods</i> | Jieling Jin ^b |
| <i>Case Study</i> | Xinyu Chen |
| <i>Conclusion</i> | Jieling Jin |

^aXinyu Chen is a PhD from the University of Montreal.

^bJieling Jin is a PhD from the Central South University.

使用 minipage 环境来给表格添加脚注。

```
\documentclass{article}
\begin{document}
\begin{center}
\begin{minipage}{.5\textwidth}
\begin{tabular}{|l|l}
\textsc{Chapter} & \textsc{Author} \\ \hline
\textit{Introduction} & Xinyu Chen\footnote{Xinyu Chen is a PhD
from the University of Montreal.} \\
\textit{Methods} & Jieling Jin\footnote{Jieling Jin is a PhD from
the Central South University.} \\
\textit{Case Study} & Xinyu Chen \\
\textit{Conclusion} & Jieling Jin
\end{tabular}
\end{minipage}
\end{center}
\end{document}
```

²This is a learning course

Part III

公式编辑

1 基本介绍

由于 L^AT_EX 编辑的数学公式颜值非常高，很多理工科研究领域的顶级期刊甚至明确要求投稿论文必须按照给定的 L^AT_EX 模板进行论文排版，这样做一方面能保证论文整体排版的美观程度，另一方面也能让生成出来的数学公式更加规范。一般而言，使用 L^AT_EX 编辑公式的一系列规则与数学公式的编写原则是一致的，例如，在 L^AT_EX 中，我们可以用 `$\frac{\partial f}{\partial x}$` 生成偏导数 $\frac{\partial f}{\partial x}$

1.1 数学公式环境

1.1.1 美元符号

在 L^AT_EX 中生成数学公式也有一些基本规则，插入公式的方式有很多种，最基本的一种方式是使用美元符号，这种方式不仅在 L^AT_EX 适用，在 Markdown 中也是适用的

- 插入行内公式，可以直接在两个美元符号中间编辑需要的公式。
- 插入行间公式，输入四个美元符号，在四个美元符号中间编辑需要的公式，生成的数学公式会自动居中对齐。

$x + y = 2$ is a simple linear equation

$$x + y = 2$$

用美元符号分别在行内和行间生成一条简单的数学公式

```
\documentclass[12pt]{article}
\begin{document}
 $x+y=2$  is a simple linear equation

$$x+y=2$$

\end{document}
```

L^AT_EX 源文件中的美元符号一般都默认为申明数学公式环境，如果想要在文档中编译出美元符号，需要在美元符号前加上一个反斜线，这种做法同样适用于百分号，百分号一般被默认为注释功能。

1.1.2 equation 环境

美元符号可以在行间插入公式，但却没办法对公式进行编号。自动生成带有公式编号的行间公式需要用到数学公式环境 `\begin{equation} \end{equation}`，使用数学公式环境 `\begin{equation} \end{equation}`，L^AT_EX 编译时会自动将公式进行居中对齐。

$$x + y = 2 \tag{1}$$

```
\documentclass[12pt]{article}
\begin{document}
\begin{equation}
x+y=2
\end{equation}
\end{document}
```

1.1.3 align 环境

在 L^AT_EX 中，除了 equation 数学公式环境，还有其他几种数学公式环境可以使用。我们要介绍的第一种是 `\begin{align} \end{align}`，它主要用于数组型的数学表达式，align 环境可以将公式进行自动对齐，它也能对每一条数学表达式分别进行公式编号。

$$x + y = 2 \tag{2}$$

$$2x + y = 3 \tag{3}$$

```
\documentclass[12pt]{article}
\usepackage{amsmath}
\begin{document}
%% 使用 align 环境
\begin{align}
x+y&=2 \\
2x+y&=3
\end{align}
\end{document}
```

```
\documentclass[12pt]{article}
\usepackage{amsmath}
\begin{document}
\begin{align*}
2x+1&=7 & 3y-2&=-5 & -5z+8&=-2 \\
2x&=6 & 3y&=-3 & -5z&=-10 \\
x&=3 & y&=-1 & z&=2
\end{align*}
\end{document}
```

使用`\begin{align}` `\end{align}`编译一个方程组，并且只对第2个方程进行编号。

```
\documentclass[12pt]{article}
\usepackage{amsmath}
\begin{document}
\begin{align}
x+y&=2 \nonumber \\
2x+y&=3
\end{align}
\end{document}
```

1.1.4 gather 环境

我们要介绍的第二种数学公式环境是`\begin{gather}` `\end{gather}`，它既可以将公式进行居中对齐，也能对每一条数学表达式分别进行公式编号。同样的，如果想要移除公式编号，只需要在公式环境中加上星号即可。

$$x + y = 2 \tag{4}$$

$$2x + y = 3 \tag{5}$$

使用 `\begin{gather}` `\end{gather}` 编译一个方程组。

```
\documentclass[12pt]{article}
\usepackage{amsmath}
\begin{document}
\begin{gather}
x+y=2 \\
2x+y=3
\end{gather}
\end{document}
```

1.2 基本格式调整

1.2.1 字符类型

在文本编辑中，我们已经介绍了几种常见的字符类型，实际上，对于数学公式而言，书写时也可以设置不同的字符类型。以 X, Y, Z, x, y, z 为例

- 命令 `\boldsymbol{X,Y,Z,x,y,z}`，编译后的效果为 **X, Y, Z, x, y, z** ，使用之前需申明 `\usepackage{amsmath}`；
- 命令 `\mathrm{X,Y,Z,x,y,z}`，编译后的效果为 X, Y, Z, x, y, z ；
- 命令 `\mathit{X,Y,Z,x,y,z}`，编译后的效果为 X, Y, Z, x, y, z ；
- 命令 `\mathbf{X,Y,Z,x,y,z}`，编译后的效果为 **X, Y, Z, x, y, z** ；
- 命令 `\mathsf{X,Y,Z,x,y,z}`，编译后的效果为 X, Y, Z, x, y, z ；
- 命令 `\mathtt{X,Y,Z,x,y,z}`，编译后的效果为 X, Y, Z, x, y, z ；
- 命令 `\boldmath{X,Y,Z,x,y,z}`，编译后的效果为 X, Y, Z, x, y, z ；依赖于特定工具包，使用之前需申明 `\usepackage{amssymb}`；
- 命令 `\mathcal{X,Y,Z}`，编译后的效果为 $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ ；
- 命令 `\mathbb{X,Y,Z}`，依赖于特定工具包，使用之前需申明 `\usepackage{amssymb, amsfonts}`，编译后的效果为 $\mathbb{X}, \mathbb{Y}, \mathbb{Z}$ ，概率论与数理统计中常见的数学期望符号 E 也是用该命令编译的，即 \mathbb{E} ；
- 命令 `\mathfrak{X,Y,Z,x,y,z}`，依赖于特定工具包，使用之前需申明 `\usepackage{amssymb, amsfonts, eufrak}`，编译后的效果为 $\mathfrak{X}, \mathfrak{Y}, \mathfrak{Z}, \mathfrak{x}, \mathfrak{y}, \mathfrak{z}$ 。

$$x^2 + y^2 - \sin z = 4 \quad (1)$$

$$\boldsymbol{x}^2 + \boldsymbol{y}^2 - \boldsymbol{\sin} z = 4 \quad (2)$$

使用\boldmath和\unboldmath对如下公式进行加粗

```
\documentclass[12pt]{article}
\begin{document}
\begin{equation}
x^2+y^2-\sin z=4
\end{equation}
\boldmath
\begin{equation}
x^2+y^2-\sin z=4
\end{equation}
\unboldmath
\end{document}
```

除此之外, 如果想在公式中插入正常的文本, 可以使用 `\text{}` 命令, 例如 $\$T_{\text{start}}\$$ 为 T_{start} 。

1.2.2 调整公式大小

如果想对单个公式调整公式字符大小, 在美元符号插入的公式中, 可以使用 `\displaystyle`、`\textstyle`、`\scriptstyle` 和 `\scriptscriptstyle` 等申明命令对公式大小进行调整, 公式显示效果依次从小到大, 这些命令一般放在公式前即可。

$$f(x) = \sum_{i=1}^n \frac{1}{x_i}$$

$$f(x) = \sum_{i=1}^n \frac{1}{x_i}$$

$$f(x) = \sum_{i=1}^n \frac{1}{x_i}$$

$$f(x) = \sum_{i=1}^n \frac{1}{x_i}$$

使用`\displaystyle`、`\textstyle`、`\scriptstyle`和`\scriptscriptstyle`这四种命令分别书写函数 $f(x) = \sum_{i=1}^n \frac{1}{x_i}$ 。

```
\documentclass[12pt]{article}
\begin{document}
 $f(x) = \sum_{i=1}^n \frac{1}{x_i},$ 
 $f(x) = \sum_{i=1}^n \frac{1}{x_i},$ 
 $f(x) = \sum_{i=1}^n \frac{1}{x_i},$ 
 $f(x) = \sum_{i=1}^n \frac{1}{x_i}.$ 
\end{document}
```

在各类公式环境（如 `equation`、`align`、`gather`）中，可以外使用一系列字符大小命令进行调整，例如用`\begingroup \endgroup`限定字符区域。

$$x + y = 2 \tag{1}$$

$$2x + y = 3 \tag{2}$$

$$x + y = 2 \tag{3}$$

$$2x + y = 3 \tag{4}$$

在`\begingroup \endgroup`中使用字符大小命令`\small`和`\Large`对公式大小进行调整。

```
\documentclass[12pt]{article}
\usepackage{amsmath}
\begin{document}
%% Small size
\begingroup
\small
\begin{align}
x+y=2 \quad & \\
2x+y=3 \quad & 
\end{align}
\end{group}
%% Large size
\Large
\begin{align}
x+y=2 \quad & \\
2x+y=3 \quad & 
\end{align}
\end{group}
\end{document}
```

1.2.3 其他格式调整

在 `equation`、`align` 等公式环境中，我们也可以通过数组 `array` 环境对数学公式进行对齐

对齐的方式有 `l`（左侧对齐）、`c`（居中对齐）和 `r`（右侧对齐）

当公式过长时，还有一些工具包提供的环境可以让公式进行自动跨行，以工具包 `breqn` 为例，在使用时，用 `\begin{dmath}` `\end{dmath}` 即可

$$\begin{cases} x + y = 2 \\ 2x + y = 3 \end{cases} \quad (1)$$

$$\begin{cases} x + y = 2 \\ 2x + y = 3 \end{cases} \quad (2)$$

使用 `\begin{array}` `\end{array}` 编译公式，并让公式居中对齐。

```
\documentclass[12pt]{article}
\usepackage{amsmath, mathtools}
```

```
\begin{document}
```

```
\begin{equation}
```

```
\begin{array}{c@{\quad}c}
```

```
A = B + C
```

```
\quad\rightarrow
```

```
& D = E - F, \quad \quad
```

```
G = H
```

```
\quad\rightarrow
```

```
& K = P + Q + M.
```

```
\end{array}
```

```
\end{equation}
```

```
\end{document}
```

1.3 练习题

$$\max_{0 \leq x \leq n-1} \prod_{i=1}^c f_i(x)$$

$$\begin{aligned}(a+b)^2 &= a^2 + 2ab + b^2 \\(a-b)^2 &= a^2 - 2ab + b^2 \\(a+b)(a-b) &= a^2 - b^2\end{aligned}\tag{3}$$

$$x + y = 2 \tag{4}$$

$$2x + y = 3 \tag{5}$$

$$f(x) = \begin{cases} x, & x > 0, \\ -x, & x < 0. \end{cases}$$

$$a = b \tag{6a}$$

$$c = d \tag{6b}$$

代 码

```


$$\max_{0 \leq x \leq n-1} \prod_{i=1}^c f_i(x)$$


\begin{align}
(a+b)^2 &= a^2 + 2ab + b^2 & \text{\nonumber} \\
(a-b)^2 &= a^2 - 2ab + b^2 & \\
(a+b)(a-b) &= a^2 - b^2 & \text{\nonumber}
\end{align}

\begin{flalign}
&x+y=2 & \\
&2x+y=3
\end{flalign}

\begin{equation*}
f(x) = \left\{ \begin{array}{c} x, \text{ \& } x > 0, \\ -x, \text{ \& } x < 0. \end{array} \right.
\end{equation*}

\begin{subequations}
\begin{equation}
\text{\label{eq-a}} \\
a = b
\end{equation}
\begin{equation}
\text{\label{eq-b}} \\
c = d
\end{equation}
\end{subequations}

```

2 常用运算符号

常用数学符号包括运算符号、标记符号、各类括号、空心符号及一些特殊函数。

2.1 运算符号

在初等数学中，最基本的运算规则是加减乘除。在 L^AT_EX 中，加法符号和减法符号就是 + 和 -；而乘法符号有两种，第一种是 \times，对应于符号 ×，第二种是 \cdot，对应于符号 ·，除法符号的命令为 \div。

$$3 + 5$$

$$3 - 5$$

$$3 \times 5$$

$$3 \cdot 5$$

$$3 \div 5$$

$$3/5$$

书写加减乘除 $3+5$ 、 $3-5$ 、 3×5 、 $3 \cdot 5$ 、 $3 \div 5$ 和 $3/5$ 。

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

$$$3+5$$$      % 加法
$$$3-5$$$      % 减法
$$$3\times 5$$$ % 乘法
$$$3\cdot 5$$$  % 乘法
$$$3\div 5$$$   % 除法
$$$3/5$$$       % 除法
\end{document}
```

在加减的基础上，命令 \pm（由 plus 和 minus 首字母构成）和 \mp（由 minus 和 plus 首字母构成）分别对应着符号 ± 和 ∓。与加减乘除同样常用的运算符号还有大于号、小于号等对于集合而言，还有一些基本运算符号如 \cap (\cap)、 \cup (\cup)、 \supset (\supset)、 \subset (\subset)、 \supseteq (\supseteq)、 \in (\in)。除此之外，与“属于”命令 \in 相反的“不属于”命令为 \notin，编译效果为 \notin 。

$$x < y$$

$$x > y$$

$$x \leq y$$

$$x \geq y$$

$$x \ll y$$

$$x \gg y$$

书写 $x < y$ 、 $x > y$ 、 $x \leq y$ 、 $x \geq y$ 、 $x \ll y$ 和 $x \gg y$ 。

```
\documentclass[12pt]{article}
\begin{document}
 $x < y$       % 小于
 $x > y$       % 大于
 $x \leq y$     % 小于或等于
 $x \geq y$     % 大于或等于
 $x \ll y$      % 远小于
 $x \gg y$      % 远大于
\end{document}
```

2.2 标记符号

在数学公式的编辑中，还有一些基本数学符号及表达式也非常重要，例如分式、上标、下标。L^AT_EX 中用于书写分数和分式的基本命令为 `\frac{分子}{分母}`，根据场景需要，也可以选用 `\dfrac{分子}{分母}` 和 `\tfrac{分子}{分母}`。

$$\frac{3}{5}$$

$$\frac{x}{y}$$

$$\frac{x}{y}$$

$$\frac{x}{y}$$

$$\frac{x+3}{y+5}$$

书写分数 $\frac{3}{5}$ 与分式 $\frac{xy}{x+3y+5}$ 。

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


$$\frac{3}{5}$$
 % 分数

$$\frac{x}{y}$$
 % 分式1

$$\frac{x+3}{y+5}$$
 % 分式2

\end{document}
```

$$x^3$$

$$x^5$$

$$x^{x+5}$$

$$x^{x^3+5}$$

书写带上标的 x_3 、 x_5 、 x_{x+5} 和 x_{x^3+5} 。

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


$$x^3$$


$$x^5$$


$$x^{x+5}$$


$$x^{x^3+5}$$


\end{document}
```

$$x_3$$

$$x_5$$

$$x_{x+5}$$

$$x_{x^3+5}$$

$$x_1, x_2, \dots, x_n$$

书写带下标的 x_3 、 x_5 、 $xx+5$ 、 $xx3+5$ 和 x_1, x_2, \dots, x_n 。

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

 $x_3$ 
 $x_5$ 
 $xx+5$ 
 $xx_{x_3}+5$ 
 $x_1, x_2, \ldots, x_n$ 

\end{document}
```

$$\hat{x}$$

$$\bar{x}$$

$$\tilde{x}$$

$$\vec{x}$$

$$\dot{x}$$

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

 $\hat{x}$ 
 $\bar{x}$ 
 $\tilde{x}$ 
 $\vec{x}$ 
 $\dot{x}$ 

\end{document}
```

根号同样是数学表达式中的常见符号，在 LaTeX 中，根号的命令为 $\sqrt{\quad}$ ，使用默认设置，生成的表达式为二次方根，如果想要设置为三次方根，则需要调整默认设置，即 $\sqrt[3]{\quad}$ ，以此类推，可以设置四次方根等。

$$\sqrt{3}$$

$$\sqrt[3]{5}$$

$$\sqrt{x+y}$$

$$\sqrt{x^3+y^5}$$

$$\sqrt{1+\sqrt{x}}$$

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


$$\sqrt{3}$$


$$\sqrt[3]{5}$$


$$\sqrt{x+y}$$


$$\sqrt{x^3+y^5}$$


$$\sqrt{1+\sqrt{x}}$$


\end{document}
```

$$\frac{\sqrt{x+3}}{\sqrt{y+5}}$$

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


$$\frac{\sqrt{x+3}}{\sqrt{y+5}}$$


\end{document}
```

2.3 各类括号

在数学表达式中，括号的用处和种类都非常多，例如最常见的小括号、中括号、大括号（即花括号）。

$$x \left(\frac{1}{y} + 1 \right)$$

$$x \left[\frac{1}{y} + 1 \right]$$

$$x \left\{ \frac{1}{y} + 1 \right\}$$

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

$$x\left(\frac{1}{y}+1\right)$$
$$x\left[\frac{1}{y}+1\right]$$
$$x\left\{\frac{1}{y}+1\right\}$$

\end{document}
```

有时候，由于公式过长等原因，我们也可以在需要分行处插入\\将括号内的公式切分成多行。

$$\left(a + \frac{b}{2} + \frac{c}{3} + \frac{d}{4} + \frac{e}{5} + \frac{f}{6} + \frac{g}{7} + \frac{h}{8} \right. \\ \left. + \frac{i}{9} + \frac{j}{10} + \frac{k}{11} + \frac{l}{12} + \frac{m}{13} + \frac{n}{14} + \frac{o}{15} + \cdots \right) \quad (7)$$

```
\documentclass[12pt]{article}
\usepackage{amsmath}
\begin{document}
\begin{equation}
\begin{aligned}
\mathbf{Bigl}(a+\frac{b}{2}+\frac{c}{3}++\frac{d}{4}+\frac{e}{5}++\frac{f}{6}++\frac{g}{7}++\frac{h}{8} \quad \\
+\frac{i}{9}+\frac{j}{10}+\frac{k}{11}++\frac{l}{12}+\frac{m}{13}++\frac{n}{14}++\frac{o}{15}+\cdots\mathbf{Bigr})
\end{aligned}
\end{equation}
\end{document}
```

在这里，我们可以使用一系列命令代替最常用的\left 和\right 组合，如\bigl 和\bigr 组合、\Bigl 和\Bigr 组合、\biggl 和\biggr 组合、\Biggl 和\Biggr 组合来控制括号大小。

$$(x+y=z)(x+y=z)(x+y=z)(x+y=z)(x+y=z)$$

```
\documentclass[12pt]{article}
\usepackage{amsmath}
\begin{document}
\begin{equation}
\left(x+y=z\right)\left(x+y=z\right)\left(x+y=z\right)\left(x+y=z\right)\left(x+y=z\right)
\end{equation}
\end{document}
```

在数学公式编辑中，除了以上常见的括号，也有一些广义的“括号”。

$$x\left|\frac{1}{y}+1\right|$$

$$x\left\|\frac{1}{y}+1\right\|$$

$$\left\langle\frac{1}{x},\frac{1}{y}\right\rangle$$

$$\left\langle\frac{1}{x},\frac{1}{y}\right\rangle$$

```
\documentclass[12pt]{article}
\begin{document}
$$x\left|\frac{1}{y}+1\right|$$
$$x\left\|\frac{1}{y}+1\right\|$$
$$\left\langle\frac{1}{x},\frac{1}{y}\right\rangle$$
$$\langle\frac{1}{x},\frac{1}{y}\rangle$$
\end{document}
```

$$\left.\frac{dy}{dx}\right|_{x=1}$$

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

$$\left.\frac{dy}{dx}\right|_{x=1}$$

\end{document}
```

2.4 空心符号

在数学表达式中，我们有时候会用到一些约定俗成的空心符号表示集合，这包括：

- 空心 R 符号 \mathbb{R} 表示由所有实数构成的集合
- 空心 Z 符号 \mathbb{Z} 表示由所有整数构成的集合
- 空心 N 符号 \mathbb{N} 表示由所有非负整数构成的集合，如果要表示正整数，使用符号 \mathbb{N}_+ 即可
- 空心 C 符号 \mathbb{C} 表示由所有复数构成的集合

需要注意的是，要想让 L^AT_EX 成功编译出这些空心符号，我们需要调用特定的工具包，即 `\usepackage{amsfonts}`，一般而言，为了保证公式的编译不出现意外，还会用到其他工具包，即 `\usepackage{amsfonts, amssymb, amsmath}`

$$X \in \mathbb{R}^{m \times n}$$

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

\begin{document}

$$X \in \mathbb{R}^m \times \mathbb{R}^n$$

\end{document}
```

$$\mathbb{R}, \mathbb{Z}, \mathbb{N}, \mathbb{C}, \mathbb{Q}$$

使用工具包 `bbold` 中的 `\mathbb` 命令书写空心的 1、2、3、4、5.

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

\begin{document}


$$\mathbb{1}, \mathbb{2}, \mathbb{3}, \mathbb{4}, \mathbb{5}$$


\end{document}
```

2.5 特殊函数

$$y = \log_2 x$$

$$y = \ln x$$

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

$$y = \log_2 x$$


$$y = \ln x$$

\end{document}
```

$$\sum_{i=1}^n x_i$$

$$\prod_{i=1}^n x_i$$

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

$$\sum_{i=1}^n x_i$$


$$\prod_{i=1}^n x_i$$

\end{document}
```

$$y = \sin x$$

$$y = \arcsin x$$

$$y = \cos x$$

$$y = \arccos x$$

$$y = \tan x$$

$$y = \arctan x$$

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

$$y=\sin x$$
$$y=\arcsin x$$
$$y=\cos x$$
$$y=\arccos x$$
$$y=\tan x$$
$$y=\arctan x$$

\end{document}
```

3 希腊字母

我们在初等数学中便已经学习到了一些常用的希腊字母，例如最常见的 π （对应于`\pi`），圆周率 π 约等于 3.14，圆的面积为 πr^2 、周长为 $2\pi r$ 。在几何学中，我们习惯用各种希腊字母表示度数，如 α （对应于`\alpha`）、 β （对应于`\beta`）、 θ （对应于`\theta`）、 ϕ 对应于`\phi`）、 ψ （对应于`\psi`）、 φ （对应于`\varphi`），使用希腊字母既方便，也容易区分于常用的 x, y, z 等其他变量。

实际上，这些希腊字母也可以用来作为变量，在概率论与数理统计中常常出现的变量就包括：

- 正态分布中的 μ （命令为`\mu`）、 σ （命令为`\sigma`）；
- 泊松分布中的 λ （命令为`\lambda`）；
- 通常表示自由度的希腊字母为 ν （命令为`\nu`）

另外，在不等式中经常用到的希腊字母有 δ （命令为`\delta`）和 ϵ （命令为`\epsilon`）。除了这些，希腊字母还有 γ （命令为`\gamma`）、 η （命令为`\eta`）、 κ （命令为`\kappa`）、 ρ

(命令为\rho)、 τ (命令为\tau) 和 ω (命令为\omega)。当然，前面提到的这些希腊字母在用途上并没有严格的界定，很多时候，我们书写数学表达式时可以根据需要选用适当的希腊字母。

$$S = \pi ab$$

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

$$S=\pi ab$$ % 椭圆面积公式

\end{document}
```

$$a^\alpha b^\beta \cdots k^\kappa l^\lambda \leq a\alpha + b\beta + \cdots + k\kappa + l\lambda$$

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

$$a^{\alpha}b^{\beta}\cdots k^{\kappa}l^{\lambda}\leq a\alpha+b\beta+\cdots
+k\kappa+l\lambda$$

\end{document}
```

$$\phi\left(\frac{x_1+x_2+\cdots+x_n}{n}\right)\leq\frac{\phi(x_1)+\phi(x_2)+\cdots+\phi(x_n)}{n}$$

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

$$\phi\left(\frac{x_{1}+x_{2}+\cdots+x_{n}}{n}\right)\leq\frac{\phi\left(x_{1}\right)+\phi\left(x_{2}\right)+\cdots+\phi\left(x_{n}\right)}{n}$$

\end{document}
```

与英文字母类似的是，有些希腊字母不但有小写字母，还有大写字母。有时，根据需要，我们还会对需要斜体的大写字母，具体如下

- 命令\Gamma 对应于希腊字母 Γ ，命令\varGamma 对应于 Γ ；
- 命令\Delta 对应于希腊字母 Δ ，命令\varDelta 对应于 Δ ；

- 命令\Theta 对应于希腊字母 Θ , 命令\varTheta 对应于 Θ ;
- 命令\Lambda 对应于希腊字母 Λ , 命令\varLambda 对应于 Λ ;
- 命令\Pi 对应于希腊字母 Π , 命令\varPi 对应于 Π ;
- 命令\Sigma 对应于希腊字母 Σ , 命令\varSigma 对应于 Σ ;
- 命令\Phi 对应于希腊字母 Φ , 命令\varPhi 对应于 Φ ;
- 命令\Omega 对应于希腊字母 Ω , 命令\varOmega 对应于 Ω 。

从这些大写希腊字母中可以看到：大写希腊字母的命令是将小写希腊字母的命令首字母进行大写，但这些大写希腊字母与小写希腊字母的区别却不仅仅是尺寸不同；当大写希腊字母作为变量时，可以采用斜体字。

$$\Delta x + \Delta y$$

$$(i, j, k) \in \Omega$$

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


$$\Delta x + \Delta y$$


$$(i, j, k) \in \Omega$$


\end{document}
```

4 微积分

事实上，数学公式的范畴极为广泛，我们所熟知的大学数学课程中，微积分、线性代数、概率论与数理统计中数学表达式的符号系统均大不相同。本节将主要介绍如何使用 L^AT_EX 对微积分中的数学表达式进行书写和编译

4.1 极限

求极限是整个微积分中的基石，例如 $\lim_{x \rightarrow 2} x^2$ 对应的 L^AT_EX 代码为 `$\lim_{x \rightarrow 2} x^2$`

$$\lim_{x \rightarrow -\infty} \frac{3x^2 - 2}{3x - 2x^2} = \lim_{x \rightarrow -\infty} \frac{x^2 \left(3 - \frac{2}{x^2}\right)}{x^2 \left(\frac{3}{x} - 2\right)} = \lim_{x \rightarrow -\infty} \frac{3 - \frac{2}{x^2}}{\frac{3}{x} - 2} = -\frac{3}{2}$$

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

$$\lim_{x\rightarrow-\infty}\frac{3x^2-2}{3x-2x^2}=\lim_{x\rightarrow-\infty}\frac{x^2\left(3-\frac{2}{x^2}\right)}{x^2\left(\frac{3}{x}-2\right)}=\lim_{x\rightarrow-\infty}\frac{3-\frac{2}{x^2}}{\frac{3}{x}-2}=-\frac{3}{2}$$

\end{document}
```

$$\lim_{\Delta t \rightarrow 0} \frac{s(t+\Delta t)+s(t)}{\Delta t} \quad \& \quad \lim_{\Delta t \rightarrow 0} \frac{s(t+\Delta t)+s(t)}{\Delta t}$$

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

$\lim_{\Delta t \rightarrow 0} \frac{s(t+\Delta t)+s(t)}{\Delta t}$ \& $\displaystyle \lim_{\Delta t \rightarrow 0} \frac{s(t+\Delta t)+s(t)}{\Delta t}$

\end{document}
```

4.2 导数

在微积分中，给定函数 $f(x)$ 后，我们能够将其导数定义为

$$f'(a) = \lim_{x \rightarrow a} \frac{f(x) - f(a)}{x - a}$$

为了让分数的形式在直观上不显得过大，可以用

$$f'(a) = \lim_{x \rightarrow a} \frac{f(x) - f(a)}{x - a}$$

，其中， \lim 和 \limits 两个命令需要配合使用。需要注意的是， $\hat{f}'(x)$ 中的 \prime 命令是标准写法，有时候也可以写作 $f'(x)$ 。

$$f'(x) = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

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

$$f^{\prime}(x)=\lim_{\Delta x \rightarrow 0} \frac{f(x+\Delta x)-f(x)}{\Delta x}$$

\end{document}
```

$$f'(x) = 15x^4 + 6x^2$$

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

$$f^{\prime}(x)=15x^4+6x^2$$

\end{document}
```

$$\frac{d}{dx}f(x) = 15x^4 + 6x^2$$

$$\frac{d^2}{dx^2}f(x) = 60x^3 + 12x$$

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

$$\frac{d}{dx}f(x)=15x^4+6x^2$$
$$\frac{d^2}{dx^2}f(x)=60x^3+12x$$

\end{document}
```

$$\frac{\partial}{\partial x}f(x, y) = 15x^4y^2 + 6x^2y$$

$$\frac{\partial}{\partial y}f(x, y) = 6x^5y + 2x^3$$

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

$$\frac{\partial}{\partial x}f(x, y)=15x^4y^2+6x^2y$$
$$\frac{\partial}{\partial y}f(x, y)=6x^5y+2x^3$$

\end{document}
```

$$z = \mu \frac{\partial y}{\partial x} \Big|_{x=0}$$

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

$$z=\mu\backslash,\frac{\backslash\textbf{partial }y}{\backslash\textbf{partial }x}\backslash\textbf{bigg}|_{x=0}$$

\end{document}
```

4.3 积分

积分的标准写法为 $\int_a^b f(x) dx$, 代码为 `\int_{\hat{a}}^{\hat{b}}fx\backslash,\mathrm{d}x`, 其中, `\int` 表示积分, 是英文单词 integral 的缩写形式, 使用 `\,` 的目的是引入一个空格。

$$\int \frac{dx}{\sqrt{a^2 + x^2}} = \frac{1}{a} \arcsin \left(\frac{x}{a} \right) + C$$

$$\int \tan^2 x dx = \tan x - x + C$$

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

$$\backslash\textbf{int}\backslash\frac{\backslash\mathrm{d}x}{\backslash\textbf{sqrt}\{a^{\hat{2}}+x^{\hat{2}}\}}=\frac{1}{a}\backslash\textbf{arcsin}\backslash\textbf{left}(\backslash\frac{\{x\}}{\{a\}}\backslash\textbf{right})+C$$
$$\backslash\textbf{int}\backslash\textbf{tan}^{\hat{2}}x\backslash,\mathrm{d}x=\backslash\textbf{tan }x-x+C$$

\end{document}
```

$$\int_a^b [\lambda_1 f_1(x) + \lambda_2 f_2(x)] dx = \lambda_1 \int_a^b f_1(x) dx + \lambda_2 \int_a^b f_2(x) dx$$

$$\int_a^b f(x) dx = \int_a^c f(x) dx + \int_c^b f(x) dx$$

```

\documentclass[12pt]{article}

\begin{document}

\begin{equation}

\begin{aligned}

V&=2\pi\int_0^1 x\left\{1-(x-1)^2\right\}\mathrm{d}x \\\
&=2\pi\int_0^2\left\{-x^3+2x^2\right\}\mathrm{d}x \\\
&=2\pi\left[-\frac{1}{4}x^4+\frac{2}{3}x^3\right]_0^2 \\\
&=8\pi/3

\end{aligned}

\end{equation}

\end{document}

```

53

$$\iint_D f(x, y) \, d\sigma$$

$$\iiint_{\Omega} (x^2 + y^2 + z^2) \, dv$$

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

$$\iint\limits_D f(x,y)\,,\mathrm{d}\sigma$$
$$\iiint\limits_{\Omega} (x^2+y^2+z^2)\mathrm{d}v$$

\end{document}
```

在积分中，有一种特殊的积分符号是在标准的积分符号上加上一个圈，可用来表示计算曲线曲面积分，即 $\oint_C f(x) \, dx + g(y) \, dy$ ，
代码为 `\oint_{C}f(x)\mathrm{d}x+g(y)\mathrm{d}y`。

4.4 练习题

$$f(x) = \frac{f(x_0)}{0!} + \frac{f'(x_0)}{1!}(x - x_0)^2$$

$$+ \cdots + \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n + R_n(x)$$

```
\documentclass[12pt]{article}
\begin{document}
\begin{equation}
\begin{aligned}
f(x) &= \frac{f(x_0)}{0!} + \frac{f^{\prime}(x_0)}{1!}(x-x_0)^2 \\\
&\quad + \dots + \frac{f^{(n)}(x_0)}{n!}(x-x_0)^n + R_n(x) \\\
&\quad \nonumber
\end{aligned}
\end{equation}
\end{document}
```

5 线性代数

5.1 矩阵

使用 L^AT_EX 时, 我们可以用 `\begin{array} \end{array}` 环境来书写矩阵。

$$\begin{array}{ccc} 1 & 2 & 3 \\ 4 & 5 & 6 \end{array}$$

$$\begin{array}{c|cc} 1 & 2 & 3 \\ \hline 4 & 5 & 6 \end{array}$$

```
\documentclass[12pt]{article}
\begin{document}
$$\left[\begin{array}{ccc} 1 & 2 & 3 \\ 4 & 5 & 6 \end{array}\right]_{\text{right}}$$
$$\left[\begin{array}{c|cc} 1 & 2 & 3 \\ \hline 4 & 5 & 6 \end{array}\right]_{\text{right}}$$
\end{document}
```

另外, 除了 `\begin{array} \end{array}`, 我们还可以用 `\begin{matrix} \end{matrix}`、`\begin{pmatrix} \end{pmatrix}`、`\begin{Vmatrix} \end{Vmatrix}` 等一系列环境来书写矩阵。

$$\begin{smallmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{smallmatrix}$$

$$\begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{matrix}$$

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

$$\begin{Bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{Bmatrix}$$

$$\begin{vmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{vmatrix}$$

$$\begin{Vmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{Vmatrix}$$

```
\documentclass[12pt]{article}
\usepackage{mathtools}
\begin{document}

$$\begin{smallmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{smallmatrix}$$


$$\begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{matrix}$$


$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$


$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$


$$\begin{Bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{Bmatrix}$$


$$\begin{vmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{vmatrix}$$


$$\begin{Vmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{Vmatrix}$$

\end{document}
```

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$


```
\documentclass[12pt]{article}
\begin{document}
 $\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$ 
\end{document}
```

当然，有时候，也可以用大写字母直接表示矩阵，用加粗的小写字母表示向量，用小写字母表示标量。需要注意的是，张量作为矩阵的延伸，一般用 \mathcal{X} 或者加粗的 $\boldsymbol{\mathcal{X}}$ 表示，对应的代码分别为 `\mathcal{X}` 和 `\boldsymbol{\mathcal{X}}`

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \begin{bmatrix} 7 \\ 8 \end{bmatrix} = \begin{bmatrix} 1 \times 7 + 2 \times 8 \\ 3 \times 7 + 4 \times 8 \\ 5 \times 7 + 6 \times 8 \end{bmatrix} = 7 \begin{bmatrix} 1 \\ 3 \\ 5 \end{bmatrix} + 8 \begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix} = \begin{bmatrix} 23 \\ 53 \\ 83 \end{bmatrix}$$

```
\documentclass[12pt]{article}
\usepackage{mathtools}
\begin{document}
 $\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \begin{bmatrix} 7 \\ 8 \end{bmatrix} = \begin{bmatrix} 1 \times 7 + 2 \times 8 \\ 3 \times 7 + 4 \times 8 \\ 5 \times 7 + 6 \times 8 \end{bmatrix} = 7 \begin{bmatrix} 1 \\ 3 \\ 5 \end{bmatrix} + 8 \begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix} = \begin{bmatrix} 23 \\ 53 \\ 83 \end{bmatrix}$ 
\end{document}
```

$$\mathbf{y} := \mathbf{y} + \left[A_1 \mid \cdots \mid A_n \right] \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_n \end{bmatrix} = \mathbf{y} + \sum_{i=1}^n A_i \mathbf{x}_i$$

```
\documentclass[12pt]{article}
\usepackage{mathtools}
\begin{document}

$$\mathbf{y} := \mathbf{y} +$$


$$\mathbf{left} \left[ \begin{array}{c|c|c} A_{\{1\}} & \mathbf{cdots} & A_{\{n\}} \end{array} \right] \mathbf{right}$$


$$\mathbf{left} \left[ \begin{array}{c} \mathbf{boldsymbol{x}_{\{1\}} \\ \mathbf{vdots} \\ \mathbf{boldsymbol{x}_{\{n\}}} \end{array} \right] \mathbf{right}$$


$$= \mathbf{boldsymbol{y}} + \sum_{i=1}^n A_{\{i\}} \mathbf{boldsymbol{x}_{\{i\}} \}$$

```

$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} a & b & c \\ b & d & e \\ c & e & f \end{bmatrix} = \begin{bmatrix} c & e & f \\ b & d & e \\ a & b & c \end{bmatrix}$$

```
\documentclass[12pt]{article}
\usepackage{mathtools}
\begin{document}

$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} a & b & c \\ b & d & e \\ c & e & f \end{bmatrix} = \begin{bmatrix} c & e & f \\ b & d & e \\ a & b & c \end{bmatrix}$$

```

5.2 符号

作用于矩阵的符号可分为标记符号和运算符号，就标记符号而言，

- 矩阵的逆，如 \mathbf{A}^{-1} ，代码为`\mathbf{A}^{-1}`；
- 矩阵的伪逆，写作 $\mathbf{A}^+ \mathbf{A}^\dagger$ ，代码分别为`\mathbf{A}^{+}` 和`\mathbf{A}^{\dagger}`；
- 矩阵的转置，写作 $\mathbf{A}^T \mathbf{A}^\top$ ，代码分别为`\mathbf{A}^T` 和`\mathbf{A}^{\top}`；
- 酉矩阵的转置，写作 \mathbf{A}^H ，代码分别为`\mathbf{A}^H`；
- 矩阵的秩，写作 $\text{rank}(\mathbf{A})$ ，代码为`\operatorname{rank}(\mathbf{A})`；
- 矩阵的迹，写作 $\text{Tr}(\mathbf{A})$ ，代码为`\operatorname{Tr}(\mathbf{A})`；
- 矩阵的行列式，写作 $\det(\mathbf{A})$ ，代码为`\det(\mathbf{A})`；

$$(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$$

$$(\mathbf{A} + \mathbf{B})^{\top} = \mathbf{A}^{\top} + \mathbf{B}^{\top}$$

$$(\mathbf{A} + \mathbf{B})^H = \mathbf{A}^H + \mathbf{B}^H$$

$$\text{Tr}(\mathbf{A} + \mathbf{B}) = \text{Tr}(\mathbf{A}) + \text{Tr}(\mathbf{B})$$

$$\det(\mathbf{AB}) = \det(\mathbf{A})\det(\mathbf{B})$$

```
\documentclass[12pt]{article}
\usepackage{mathtools}
\begin{document}

$$\left(\mathbf{AB}\right)^{-1}=\mathbf{B}^{-1}\mathbf{A}^{-1}$$


$$\left(\mathbf{A}+\mathbf{B}\right)^{\top}=\mathbf{A}^{\top}+\mathbf{B}^{\top}$$


$$\left(\mathbf{A}+\mathbf{B}\right)^H=\mathbf{A}^H+\mathbf{B}^H$$


$$\text{Tr}(\mathbf{A}+\mathbf{B})=\text{Tr}(\mathbf{A})+\text{Tr}(\mathbf{B})$$


$$\det(\mathbf{AB})=\det(\mathbf{A})\det(\mathbf{B})$$

\end{document}
```

$$\frac{\partial}{\partial \mathbf{X}} \text{Tr}(\mathbf{AXB}) = \mathbf{A}^{\top} \mathbf{B}^{\top}$$

```
\documentclass[12pt]{article}
\usepackage{mathtools}
\begin{document}

$$\frac{\partial}{\partial \mathbf{X}} \text{Tr}(\mathbf{AXB}) = \mathbf{A}^{\top} \mathbf{B}^{\top}$$

\end{document}
```

$$p(\mathbf{x}) = \frac{1}{\sqrt{\det(2\pi)}} \exp \left[-\frac{1}{2} (\mathbf{x} - \mu)^{\top} \Sigma^{-1} (\mathbf{x} - \mu) \right]$$

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

$$p\left(\mathbf{x}\right)=
\frac{1}{\sqrt{\det\left(2\pi\mathbf{\Sigma}\right)}}
\exp\left[-\frac{1}{2}\left(\mathbf{x}-\mathbf{\mu}\right)^{\top}\mathbf{\Sigma}^{-1}\left(\mathbf{x}-\mathbf{\mu}\right)\right]$$

\end{document}
```

$$p(\mathbf{x}) = \sum_{k=1}^K \rho_k \frac{1}{\sqrt{\det(2\pi \mathbf{\Sigma}_k)}} \exp \left[-\frac{1}{2} (\mathbf{x} - \mu_k)^{\top} \mathbf{\Sigma}_k^{-1} (\mathbf{x} - \mu_k) \right]$$

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

$$p\left(\mathbf{x}\right)=\sum_{k=1}^K\rho_k\frac{1}{\sqrt{\det\left(2\pi\mathbf{\Sigma}_k\right)}}
\exp\left[-\frac{1}{2}\left(\mathbf{x}-\mathbf{\mu}_k\right)^{\top}\mathbf{\Sigma}_k^{-1}\left(\mathbf{x}-\mathbf{\mu}_k\right)\right]$$

\end{document}
```

$$\mathbf{X} \otimes \mathbf{Y} = \begin{bmatrix} x_{11}\mathbf{Y} & x_{12}\mathbf{Y} & \cdots & x_{1n}\mathbf{Y} \\ x_{21}\mathbf{Y} & x_{22}\mathbf{Y} & \cdots & x_{2n}\mathbf{Y} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1}\mathbf{Y} & x_{m2}\mathbf{Y} & \cdots & x_{mn}\mathbf{Y} \end{bmatrix} \quad (9)$$

```
\documentclass[12pt]{article}
\begin{document}
\begin{equation}
\mathbf{X}\otimes\mathbf{Y}=\left[\begin{array}{cccc}
x_{11}\mathbf{Y} & x_{12}\mathbf{Y} & \cdots & x_{1n}\mathbf{Y} \\
x_{21}\mathbf{Y} & x_{22}\mathbf{Y} & \cdots & x_{2n}\mathbf{Y} \\
\vdots & \vdots & \ddots & \vdots \\
x_{m1}\mathbf{Y} & x_{m2}\mathbf{Y} & \cdots & x_{mn}\mathbf{Y}
\end{array}\right]
\end{equation}
\end{document}
```

$$\begin{aligned} \mathbf{A}\mathbf{X} + \mathbf{X}\mathbf{B} &= \mathbf{C} \\ \Rightarrow \operatorname{vec}(\mathbf{X}) &= (\mathbf{I} \otimes \mathbf{A} + \mathbf{B}^\top \otimes \mathbf{I})^{-1} \operatorname{vec}(\mathbf{C}) \end{aligned} \quad (10)$$

```
\documentclass[12pt]{article}
\begin{document}
\begin{equation}
\begin{aligned}
&\&\mathbf{A}\mathbf{X}+\mathbf{X}\mathbf{B}=\mathbf{C} \quad \& \\
&\&\mathbf{Rrightarrow}\quad\&\operatorname{vec}(\mathbf{X})\mathbf{right} = \\
&\&\mathbf{left}(\mathbf{I}\mathbf{otimes}\mathbf{A}+\mathbf{B}^{\top}\mathbf{otimes}\mathbf{I})\mathbf{right} \\
&\quad\mathbf{left}^{-1} \\
&\operatorname{vec}(\mathbf{C})\mathbf{right} \\
&\end{aligned}
\end{equation}
\end{document}
```

5.3 范数

$$\frac{1}{2} \mathbf{x}^\top \mathbf{A}^\top \mathbf{A} \mathbf{x} - \mathbf{x}^\top \mathbf{A}^\top \mathbf{b} = \frac{1}{2} \|\mathbf{A} \mathbf{x} - \mathbf{b}\|_2^2 - \frac{1}{2} \|\mathbf{b}\|_2^2$$

```
\documentclass[12pt]{article}
\begin{document}
\begin{equation}
\begin{aligned}
&\&\frac{1}{2}\mathbf{x}^{\top}\mathbf{A}^{\top}\mathbf{A}\mathbf{x}-\mathbf{x}^{\top}\mathbf{A}^{\top}\mathbf{b} \\
&\&= \frac{1}{2}\mathbf{left}(\|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2-\|\mathbf{b}\|_2^2) \\
&\&\mathbf{right} \\
&\end{aligned}
\end{equation}
\end{document}
```

$$= \left\| \left\| \left[\begin{array}{cc|cc} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ \hline a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \\ \hline a_{51} & a_{52} & a_{53} & a_{54} \\ a_{61} & a_{62} & a_{63} & a_{64} \end{array} \right] - \left[\begin{array}{cc} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{array} \right] \otimes \left[\begin{array}{cc} c_{11} & c_{12} \\ c_{21} & c_{22} \end{array} \right] \right\|_F \right\|$$

```
\documentclass[12pt]{article}
\begin{document}
\begin{equation}
\begin{aligned}
&\&\left[\left[\left[\begin{array}{cccc} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \end{array}\right] \right. \\
&\quad \left.\hline a_{31} & a_{32} & a_{33} & a_{34} \right. \\
&\quad \left.\hline a_{41} & a_{42} & a_{43} & a_{44} \right. \\
&\quad \left.\hline a_{51} & a_{52} & a_{53} & a_{54} \right. \\
&\quad \left.\hline a_{61} & a_{62} & a_{63} & a_{64} \right. \\
&\quad \left.\vphantom{\begin{matrix} a_{11} \\ a_{21} \end{matrix}}\right]\right]\otimes \\
&- \left[\begin{array}{cc} b_{11} & b_{12} \\ b_{21} & b_{22} \end{array}\right] \left[\begin{array}{cc} b_{31} & b_{32} \end{array}\right] \\
&\quad \left.\vphantom{\begin{matrix} b_{11} \\ b_{21} \end{matrix}}\right]\otimes \\
&\left[\begin{array}{cc} c_{11} & c_{12} \\ c_{21} & c_{22} \end{array}\right] \left[\begin{array}{c} F \end{array}\right] \\
&= \&\left[\left[\left[\begin{array}{cccc} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \end{array}\right] \right. \\
&\quad \left.\hline a_{31} & a_{32} & a_{33} & a_{34} \right. \\
&\quad \left.\hline a_{41} & a_{42} & a_{43} & a_{44} \right. \\
&\quad \left.\hline a_{51} & a_{52} & a_{53} & a_{54} \right. \\
&\quad \left.\hline a_{61} & a_{62} & a_{63} & a_{64} \right. \\
&\quad \left.\vphantom{\begin{matrix} a_{11} \\ a_{21} \end{matrix}}\right]\right]\otimes \\
&- \left[\begin{array}{ccc} b_{11} & b_{21} & b_{31} \\ b_{12} & b_{22} & b_{32} \end{array}\right] \\
&\quad \left.\vphantom{\begin{matrix} b_{11} \\ b_{12} \end{matrix}}\right]\otimes \left[\begin{array}{cccc} c_{11} & c_{21} & c_{12} & c_{22} \end{array}\right]
\end{aligned}
\end{equation}
```

```
\end{array}\right]\right\|_{F}\}
\end{aligned}
\end{equation}
\end{document}
```

5.4 练习题

$$\mathbf{a} = \vec{a} = \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix} \quad (11)$$

```
\documentclass[12pt]{article}
\usepackage{amsmath}
\begin{document}
%% 提示：公式中的字符加粗使用\boldsymbol{}命令，带箭头的向量使用\vec{}或者\
    overrightarrow{}命令
\begin{equation}
    \begin{equation}
        \mathbf{a}=\vec{a}=\begin{bmatrix}
            a_{1} \\
            \mathbf{\vdots} \\
            a_{n}
        \end{bmatrix}
    \end{equation}
\end{equation}
\end{document}
```

$$\nabla f(\mathbf{x}) = \begin{pmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1} \\ \vdots \\ \frac{\partial f(\mathbf{x})}{\partial x_n} \end{pmatrix}$$

```
\documentclass[12pt]{article}
\usepackage{amssymb, amsfonts}
\begin{document}
%% 提示：梯度对应的命令为\nabla
\begin{equation}
\begin{equation}
\nabla f(\mathbf{x})=\begin{pmatrix}
\frac{\partial f(\mathbf{x})}{\partial x_1} \\
\vdots \\
\frac{\partial f(\mathbf{x})}{\partial x_n}
\end{pmatrix}
\end{equation}
\end{equation}
\end{document}
```

6 概率论与数理统计

概率论与数理统计是许多方向开展科学研究的基础，不管是描述客观存在的数据，还是刻画变量之间的关联规则，凭借概率论知识都能得心应手。概率论的数学公式也有其自身特点，本节主要介绍概率论与数理统计范畴内常用的数学公式在 LaTeX 中的写法。

6.1 概率论基础

概率论中有一个重要的准则叫做贝叶斯准则，贝叶斯准则的基础为贝叶斯公式，用来描述两个条件概率之间的关系。

$$p(\theta | y) = \frac{p(\theta, y)}{p(y)} = \frac{p(\theta) p(y | \theta)}{p(y)}$$

$$p(\theta | y) \propto p(\theta) p(y | \theta)$$

贝叶斯公式

```
\documentclass[12pt]{article}
\begin{document}
$$p\left(\theta\mid y\right)
=\frac{p\left(\theta,y\right)}{p\left(y\right)}
=\frac{p\left(\theta\right)p\left(y\mid\theta\right)}{p\left(y\right)}$$
$$p\left(\theta\mid y\right)\propto p\left(\theta\right)p\left(y\mid\theta\right)
\right)$$
\end{document}
```

$$\mathbb{E}(x) = \int xp(x) dx$$

$$\mathbb{V}(x) = \int (x - \mathbb{E}(x))^2 p(x) dx$$

期望方差

```
\documentclass[12pt]{article}
\begin{document}
$$\mathbb{E}\left(x\right)=\int xp\left(x\right)dx$$
$$\mathbb{V}\left(x\right)=
\int \left(x-\mathbb{E}\left(x\right)\right)^2p\left(x\right)dx$$
\end{document}
```

$$p(y) = \int p(y, \theta) d\theta = \int p(\theta) p(y | \theta) d\theta$$

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

$$p\left(y\right)=\int p\left(y,\theta\right)d\theta
=\int p\left(\theta\right)p\left(y\mid\theta\right)d\theta$$

\end{document}
```

6.2 概率分布

概率分布，是指用于表述随机变量取值的概率规律，是概率论中最常见的表达式之一

$$x \sim \mathcal{N}(\mu, \sigma^2)$$

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right)$$

正态分布

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

$$x \sim \mathcal{N}(\mu, \sigma^2)$$


$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right)$$

\end{document}
```

$$p(y) = \frac{\text{Poisson}(y \mid \theta) \text{Gamma}(\theta \mid \alpha, \beta)}{\text{Gamma}(\theta \mid \alpha + y, 1 + \beta)} = \frac{\Gamma(\alpha + y)\beta^\alpha}{\Gamma(\alpha)y!(1 + \beta)^{\alpha+y}} \quad (12)$$

```
\documentclass[12pt]{article}
\begin{document}
\begin{equation}
p(y) = \frac{\text{Poisson}(y \mid \theta)}{\text{Gamma}(\theta \mid \alpha + y, 1 + \beta)} = \frac{\Gamma(\alpha + y)\beta^\alpha}{\Gamma(\alpha)y!(1 + \beta)^{\alpha+y}}
\end{equation}
\end{document}
```

$$\theta \mid y \sim \text{Gamma}\left(\alpha + \sum_{i=1}^n y_i, \beta + \sum_{i=1}^n x_i\right)$$

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

$$\theta \mid y \sim \text{Gamma}\left(\alpha + \sum_{i=1}^n y_i, \beta + \sum_{i=1}^n x_i\right)$$

\end{document}
```

$$\sigma^2 \mid y \sim \text{Inv-}\chi^2(n, s^2)$$

```
\documentclass[12pt]{article}
\begin{document}
$$\boldsymbol{\sigma}^2 \mid y \sim \operatorname{Inv-}\chi^2(n, s^2)$$
\end{document}
```

$$p(\beta \mid \mu_1, \mu_2, \tau_1, \tau_2, \rho) = \prod_{j=1}^J \mathcal{N} \left(\begin{pmatrix} \beta_{1j} \\ \beta_{2j} \end{pmatrix} \mid \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \tau_1^2 & \rho\tau_1\tau_2 \\ \rho\tau_1\tau_2 & \tau_2^2 \end{pmatrix} \right)$$

```
\documentclass[12pt]{article}
\begin{document}
$$p(\boldsymbol{\beta} \mid \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\tau}_1, \boldsymbol{\tau}_2, \boldsymbol{\rho})
= \prod_{j=1}^J \mathcal{N} \left( \begin{pmatrix} \beta_{1j} \\ \beta_{2j} \end{pmatrix} \mid \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \tau_1^2 & \rho\tau_1\tau_2 \\ \rho\tau_1\tau_2 & \tau_2^2 \end{pmatrix} \right)$$
\end{document}
```

$$\begin{aligned} y_{ij} &\sim \mathcal{N}(\alpha_j + x_{ij}\beta_j, \sigma_y^2) \\ \begin{pmatrix} \alpha \\ \beta \end{pmatrix} &\sim \mathcal{N} \left(\begin{pmatrix} \mu_\alpha \\ \mu_\beta \end{pmatrix}, \begin{pmatrix} \sigma_\alpha^2 & \rho\sigma_\alpha\sigma_\beta \\ \rho\sigma_\alpha\sigma_\beta & \sigma_\beta^2 \end{pmatrix} \right) \end{aligned} \quad (13)$$

```
\documentclass[12pt]{article}
\begin{document}
\begin{equation}
\begin{aligned}
&y_{ij} \sim \mathcal{N}(\boldsymbol{\alpha}_j + \mathbf{x}_{ij} \boldsymbol{\beta}_j, \boldsymbol{\sigma}_y^2) \\
&\begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{pmatrix} \sim \mathcal{N} \left( \begin{pmatrix} \boldsymbol{\mu}_\alpha \\ \boldsymbol{\mu}_\beta \end{pmatrix}, \begin{pmatrix} \boldsymbol{\sigma}_\alpha^2 & \rho\boldsymbol{\sigma}_\alpha\boldsymbol{\sigma}_\beta \\ \rho\boldsymbol{\sigma}_\alpha\boldsymbol{\sigma}_\beta & \boldsymbol{\sigma}_\beta^2 \end{pmatrix} \right)
\end{aligned}
\end{equation}
\end{document}
```

```
\end{equation}
\end{document}
```

6.3 练习题

$$p(\alpha, \beta | y) \propto (\alpha, \beta) = \prod_{j=1}^J \frac{\Gamma(\alpha + \beta) \Gamma(\alpha + y_i) + \Gamma(\beta + n_j - y_j)}{\Gamma(\alpha) \Gamma(\beta) \Gamma(\alpha + \beta + n_j)}$$

```
\documentclass[12pt]{article}
\begin{document}
\begin{equation}
p\left(\alpha, \beta \mid y\right) \propto \left(\alpha, \beta\right)=\prod_{j=1}^J \frac{\Gamma(\alpha+\beta) \Gamma(\alpha+y_i)+\Gamma(\beta+n_j-y_j)}{\Gamma(\alpha) \Gamma(\beta) \Gamma(\alpha+\beta+n_j)}
\end{equation}
\end{document}
```

6.4 优化理论

优化理论是科学研究的另一个重要方向，它的关键在于使用数学模型对现实问题建模，并在若干约束的条件下，求问题的最优解，主要分为单目标优化、多目标优化两种类型。不管是单目标优化还是多目标优化，其数学表达式的主要内容均包括目标函数及约束条件两个主要部分。

$$\begin{aligned} \min_x f(x) \\ \text{s.t. } x \in \mathcal{X}, \quad g_j(x) \leq 0, \quad j = 1, \dots, r \end{aligned} \tag{14}$$

```
\documentclass[12pt]{article}
\begin{document}
\begin{equation}
\begin{aligned}
&\&\min_x f(x) \\
&\&\text{s.t. } x \in \mathcal{X}, \quad g_j(x) \leq 0, \quad j = 1, \dots, r
\end{aligned}
\end{equation}
```

```
\end{document}
```

$$x^* \in \arg \min_{x \in \mathcal{X}} L(x, \lambda^*)$$

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

$$x^* \in \operatorname{arg} \min_{x \in \mathcal{X}} L(x, \lambda^*)$$

\end{document}
```

$$\operatorname{conv}(\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_m\}) = \left\{ \sum_{i=1}^m \alpha_i \boldsymbol{x}_i \mid \alpha_i \geq 0, i = 1, \dots, m, \sum_{i=1}^m \alpha_i = 1 \right\}$$

```
\documentclass[12pt]{article}
\usepackage{mathtools}
\begin{document}

$$\operatorname{conv}(\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_m\}) = \left\{ \sum_{i=1}^m \alpha_i \boldsymbol{x}_i \mid \alpha_i \geq 0, i = 1, \dots, m, \sum_{i=1}^m \alpha_i = 1 \right\}$$

\end{document}
```

$$\begin{aligned} f^*(y) &= \sup_{x \in \mathbb{R}^n} \{xy - f(x)\}, \quad y \in \mathbb{R}^n \\ f^{**}(x) &= \sup_{y \in \mathbb{R}^n} \{xy - f^*(y)\}, \quad x \in \mathbb{R}^n \end{aligned} \tag{15}$$

```
\documentclass[12pt]{article}
\usepackage{amssymb, amsfonts}
\begin{document}

\begin{equation}
\begin{aligned}
& f^*(y) = \sup_{x \in \mathbb{R}^n} \{xy - f(x)\}, \quad y \in \mathbb{R}^n \\
& f^{**}(x) = \sup_{y \in \mathbb{R}^n} \{xy - f^*(y)\}, \quad x \in \mathbb{R}^n
\end{aligned}
\end{equation}
```

```
\end{aligned}
\end{equation}
\end{document}
```

$$f(x) = \inf_{z \in \mathbb{R}^m} F(x, z), \quad x \in \mathbb{R}^n \quad (16)$$

```
documentclass[12pt]{article}
\usepackage{amssymb, amsfonts}
\begin{document}
\begin{equation}
f(x)=\inf_{z \in \mathbb{R}^m} F(x, z), \quad \textbf{quad } x \in \mathbb{R}^n
\end{equation}
\end{document}
```

$$p(u) = \inf_{x \in \mathcal{X}} \sup_{z \in \mathcal{Z}} \{ \phi(x, z) - u'z \}, \quad u \in \mathbb{R}^m \quad (17)$$

```
\documentclass[12pt]{article}
\usepackage{amssymb, amsfonts}
\begin{document}
\begin{equation}
p(u)=\inf_{x \in \mathcal{X}} \sup_{z \in \mathcal{Z}} \left\{ \phi(x, z) - u'z \right. \\
\left. \textbf{right } \right\}, \quad \textbf{quad } u \in \mathbb{R}^m
\end{equation}
\end{document}
```

6.5 练习题

$$\begin{aligned}
 q(\mu) &= \inf_{u \in \mathbb{R}^m} \{p(u) + u'\mu\} \\
 &= \inf_{u \in \mathbb{R}^m} \inf_{x \in \mathcal{X}} \{p_x(u) + u'\mu\} \\
 &= \inf_{x \in \mathcal{X}} \inf_{u \in \mathbb{R}^m} \{p_x(u) + u'\mu\} \\
 &= \inf_{x \in \mathcal{X}} \{-p_x^*(-\mu)\}
 \end{aligned}$$

```

\begin{equation}
\begin{aligned}
q(\bm{\mu}) &= \inf_{u \in \mathbb{R}^m} \{p(u) + u' \bm{\mu}\} \\
&= \inf_{u \in \mathbb{R}^m} \inf_{x \in \mathcal{X}} \{p_x(u) + u' \bm{\mu}\} \\
&= \inf_{x \in \mathcal{X}} \inf_{u \in \mathbb{R}^m} \{p_x(u) + u' \bm{\mu}\} \\
&= \inf_{x \in \mathcal{X}} \{-p_x^*(-\bm{\mu})\}
\end{aligned}
\end{equation}

```

Part IV

表格制作

1 基本介绍

表格是展现数据的一种常用方式。L^AT_EX 提供了多种表格环境用于制作各类表格，例如，`tabular`、`tabular*`、`tabularx`、`tabulary`、`table`、`longtable` 等。其中比较常用的方法是将 `tabular` 环境嵌入到 `table` 环境中，可以创建包含表格内容、表格标题、引用标签等属性的完整表格。

1.1 `tabular` 环境：创建表格内容

通过创建 `tabular` 环境可以定义表格内容、对齐方式、外观样式等，使用方式与前面章节中介绍的使用 `array` 环境制作数表（即矩阵）的方式类似。例如：

| | | |
|-----|-----|-----|
| a | b | c |
| d | e | f |
| g | h | i |

代码

```


$$\left[ \begin{array}{ccc} a & b & c \\ d & e & f \\ g & h & i \end{array} \right]$$


```

将上述代码中的 `array` 环境改写为 `tabular` 环境，得到如下所示的代码语句：

| | | |
|-----|-----|-----|
| a | b | c |
| d | e | f |
| g | h | i |

这里制作出来的表格是文本模式下的，跟 `array` 环境制作的数学模式略有不同，不同之处在于，`array` 环境制作的数表是属于数学公式，而使用 `tabular` 环境制作得到的表格则属于文本内容，但两者的用法及命令格式极其相似。在 `tabular` 环境下：

- 在 `\begin{tabular}` 命令后的 `{}` 内设置表格的列类型参数，包括：
 - 设置每列的单元格对齐方式。对齐方式选项包括 `l`、`c` 和 `r`，即 `left`、`center` 和 `right` 的首字母，分别对应左对齐、居中对齐和右对齐，每个字母对应一列；
 - 创建表格列分隔线。表格列分隔线以 `|` 符号表示，`|` 符号的个数表示列分隔线中线的个数，如 `|` 表示使用单线分隔列，`||` 表示使用双线分隔列，以此类推。

分割线符号可以设置在列对齐方式选项的左侧或右侧，分别表示创建列的左分隔线和右分隔线。

- 使用 `\` 符号表示一行内容的结束；
- 使用 `&` 符号划分行内的单元格；
- 使用 `\hline` 命令创建行分隔线。

```
\documentclass[12pt]{article}
\begin{document}
\begin{tabular}{c|c|c}
a & b & c \\
\hline
d & e & f \\
\hline
g & h & i \\
\end{tabular}
\end{document}
```

在 `tabular` 环境中，行分割线亦也可以通过 `\usepackage{booktabs}` 调用 `booktabs` 宏包，并分别使用 `\toprule`、`\midrule` 和 `\bottomrule` 命令来添加不同粗细的横线。其中，在调用 `booktabs` 的情况下，可以通过 `\cmidrule[thickness]{a-b}` 来实现自定义横线，`[thickness]` 控制横线的粗细，`{a-b}` 指定横线需要横跨的列序号。

| | $x = 1$ | $x = 2$ | $x = 3$ | $x = 4$ |
|-----------|---------|---------|---------|---------|
| $y = x$ | 1 | 2 | 3 | 4 |
| $y = x^2$ | 1 | 4 | 9 | 16 |
| $y = x^3$ | 1 | 8 | 27 | 64 |

简单表格

```
\begin{center}
\begin{tabular}{l|cccc}
\hline
&  $x=1$  &  $x=2$  &  $x=3$  &  $x=4$  \\
\hline
 $y=x$  & 1 & 2 & 3 & 4 \\
 $y=x^2$  & 1 & 4 & 9 & 16 \\
 $y=x^3$  & 1 & 8 & 27 & 64 \\
\hline
\end{tabular}
\end{center}
```

1.2 table 环境：自动编号与浮动表格

使用 table 环境嵌套 tabular 环境，能够为创建的表格进行自动递增编号。此外，可以使用 `\caption{}` 命令设置表格标题、使用 `\label{}` 命令为表格建立索引标签、使用 `\centering` 命令将表格置于文档中间，如下所示：

```
\begin{table}
  \centering
    \caption{Title of a table.}
    \label{Label of the table}
    \begin{tabular}
      % 表格内容
    \end{tabular}
\end{table}
```

事实上，在 `\begin{table}` `\end{table}` 环境中创建的表格属于浮动元素：浮动元素 (floating) 是指不能跨页分割的元素，比如图片和表格。一般而言，浮动元素的显示位置未必是代码的位置，比如，当页面空间不足时，LaTeX 会根据内置的算法尝试将浮动元素放置到后面的页面中，避免出现内容跨页分割或者页面大量留白的情况，从而创建更协调也更专业的文档。

通过在 `\begin{table}` 的 `[]` 中设置位置控制参数，可以为浮动表格指定期望放置位置，各参数值及其含义如下：

- h: 英文单词 here 的首写字母，表示代码当前位置；
- t: 英文单词 top 的首写字母，表示页面顶部位置；
- b: 英文单词 bottom 的首写字母，表示页面底部位置；
- p: 英文单词 page 的首写字母，表示后面的页面；
- !: ! 参数一般与其它位置参数配合使用，表示当空间足够时，强制将表格放在指定位置。如 lh 表示将表格强制放到当前页面，但当页面空间不足时，也可能被放置到后续页面中；
- H: 表示将表格强制放在代码当前位置，具有比 lh 更严格的效果，使用时需要先在导言区使用 `\usepackage{float}` 声明语句调用 float 宏包。

根据需要，浮动元素的位置控制参数一般可以设置为 h、b、t、p、! 和 H 的任意无序组合。该参数的缺省值为 tbp，此时 LaTeX 会尝试将表格放在页面的顶端或者底端，否则会将表格放在下一页。

表 1: Title of a table.

| | $x = 1$ | $x = 2$ | $x = 3$ | $x = 4$ |
|-----------|---------|---------|---------|---------|
| $y = x$ | 1 | 2 | 3 | 4 |
| $y = x^2$ | 1 | 4 | 9 | 16 |
| $y = x^3$ | 1 | 8 | 27 | 64 |

表格嵌入到table环境中，创建了一个位置居中、有标题、索引、自动编号的表格。

```

\documentclass[12pt]{article}
\begin{document}
\begin{table}[h]
\centering
  \caption{Title of a table.}
  \label{first label}
  \begin{tabular}{l|cccc}
    \hline
    &  $x=1$  &  $x=2$  &  $x=3$  &  $x=4$  \\
    \hline
     $y=x$  & 1 & 2 & 3 & 4 \\
     $y=x^2$  & 1 & 4 & 9 & 16 \\
     $y=x^3$  & 1 & 8 & 27 & 64 \\
    \hline
  \end{tabular}
\end{table}
\end{document}

```

Table 2 shows the values of some basic functions.

表 2: The values of some basic functions.

| | $x = 1$ | $x = 2$ | $x = 3$ | $x = 4$ |
|-----------|---------|---------|---------|---------|
| $y = x$ | 1 | 2 | 3 | 4 |
| $y = x^2$ | 1 | 4 | 9 | 16 |
| $y = x^3$ | 1 | 8 | 27 | 64 |

在table环境中将表格的位置控制参数设置为htbp。

```

\documentclass[12pt]{article}
\begin{document}
Table~\ref{table1} shows the values of some basic functions.
\begin{table}[htbp] % 设置位置参数
\centering
  \caption{The values of some basic functions.}
  \begin{tabular}{l|cccc}
    \hline

```

```

& $x=1$ & $x=2$ & $x=3$ & $x=4$ \\
\hline
$y=x$ & 1 & 2 & 3 & 4 \\
$y=x^{\{2\}}$ & 1 & 4 & 9 & 16 \\
$y=x^{\{3\}}$ & 1 & 8 & 27 & 64 \\
\hline
\end{tabular}
\label{table1}
\end{table}
\end{document}

```

2 合并单元格

如果需要合并单元格, 首先应在导言区声明`\usepackage{multirow}` 以导入 `multirow` 宏包, 并使用`\multicolumn` 命令合并同行不同列的单元格、使用`\multirow` 命令合并同行不同列的单元格。

2.1 合并不同列的单元格

合并不同列的单元格时, 应在 `tabular` 环境中使用`\multicolumn{合并列数}{合并后的列类型参数}{单元格内容}` 语句定义合并单元格。此时, 合并后的单元格的列类型将由`\multicolumn` 给出, 而非`\begin{tabular}` 中预设的列类型参数。

| Column1 | Column2 | Column3 | Column4 |
|-----------|---------|---------|---------|
| A1 and A2 | | A3 | A4 |
| B1 | B2 | B3 | B4 |
| C1 | C2 | C3 | C4 |

在 `tabular` 环境中使用`\multicolumn` 命令合并不同列的单元格。

```

\documentclass[12pt]{article}
\usepackage{multirow}
\begin{document}
\begin{tabular}{|l|l|l|l|}
\hline
Column1 & Column2 & Column3 & Column4 \\
\hline
\multicolumn{2}{|c|}{A1 and A2} & A3 & A4 \\
\hline
B1 & B2 & B3 & B4 \\
\hline
\end{tabular}
\end{document}

```

```

\hline
C1 & C2 & C3 & C4 \\
\hline
\end{tabular}
\end{document}

```

2.2 合并不同行的单元格

合并不同行的单元格时使用的语句为`\multirow{合并行数}{合并后的宽度}{单元格内容}`。如果把`{合并后的宽度}`参数设置为`{*}`，那么 LaTeX 会根据文本内容自动设置单元格宽度。在绘制行分隔线时，使用`\hline`命令会创建一条横跨表格左右两端的横线，显然不适用于合并单元格后的行。此时应用`\cline{起始列号-终止列号}`命令，通过指定行分隔线的起始列和终止列，从而定制跨越了部分列的行分隔线。

合并多行的单元格时，除了第一个单元格处使用`\multirow`命令定义单元格，其余被合并的单元格处均留空。

| Column1 | Column2 | Column3 | Column4 |
|-----------|---------|---------|---------|
| A1 and B1 | A2 | A3 | A4 |
| | B2 | B3 | B4 |
| C1 | C2 | C3 | C4 |

在 tabular 环境中使用`\multirow`命令合并不同列的单元格，并使用`\cline`命令定制行分隔线的起始点。

```

\documentclass[12pt]{article}
\usepackage{multirow}
\begin{document}
\begin{tabular}{|l|l|l|l|}
\hline
Column1 & Column2 & Column3 & Column4 \\
\hline
\multirow{2}{*}{A1 and B1} & A2 & A3 & A4 \\
\cline{2-4} % 创建一条从第2列到第4列的行分隔线
& B2 & B3 & B4 \\
\hline
C1 & C2 & C3 & C4 \\
\hline
\end{tabular}
\end{document}

```

2.3 合并多行多列的单元格

通过嵌套使用`\multicolumn` 和`\multirow` 命令可以实现对多行多列单元格的合并操作，具体语句为`\multicolumn{合并列数}{合并后的列类型参数}{\multirow{合并行数}{合并后的宽度}{单元格内容}}`。在同时合并涉及多行多列的单元格时，除了第一行使用`\multicolumn` 和`\multirow` 嵌套命令定义单元格，其余被合并的行处均使用内容为空的`\multicolumn` 命令。

| Column1 | Column2 | Column3 | Column4 |
|-------------------|---------|---------|---------|
| A1, A2, B1 and B2 | | A3 | A4 |
| | | B3 | B4 |
| C1 | C2 | C3 | C4 |

在`tabular`环境中嵌套使用`\multicolumn`和`\multirow`命令合并多行多列的单元格。

```
\documentclass[12pt]{article}
\usepackage{multirow}
\begin{document}
\begin{tabular}{|1|1|1|1|}
  \hline
  Column1 & Column2 & Column3 & Column4 \\
  \hline
  \multicolumn{2}{|c|}{\multirow{2}{*}{A1, A2, B1 and B2}} & A3 & A4 \\
  \cline{3-4} % 创建一条从第3列到第4列的行分隔线
  \multicolumn{2}{|c|}{} & B3 & B4 \\
  \hline
  C1 & C2 & C3 & C4 \\
  \hline
\end{tabular}
\end{document}
```

3 插入斜线与标注

3.1 插入斜线

在制作斜线表头或填充空白单元格时，经常需要用到斜线。在 L^AT_EX 中，我们可以通过调用 `diagbox` 宏包及其提供的`\diagbox[参数]{单元格内容 1}...{单元格内容 n}` 命令将一个单元格划分为 `n` 个部分（即插入 `(n-1)` 条斜线），并且可以在 `[]` 中设置不同参数，从而对斜线宽度、高度、方向等属性进行调整，主要包括：

- `width`：设置斜线宽度；

- height: 设置斜线高度;
- font: 设置单元格字体大小和字体类型;
- linewidth: 设置线宽;
- linecolor: 设置线的颜色 (需结合 xcolor 或其他宏包使用);
- dir: 设置斜线方向, 包括 NW (默认)、NE、SW 和 SE, 分别表示西北方向、东北方向、西南方向、东南方向。当仅插入一个斜线时, dir=NW 与 dir=SE、dir=NE 与 dir=SW 效果相同, 分别表示插入反斜线和斜线; 当插入两个斜线时, 如 \diagbox[设置 dir 参数]{A}{B}{C}, 使用 NW、NE、SW 和 SE 的效果如下图所示

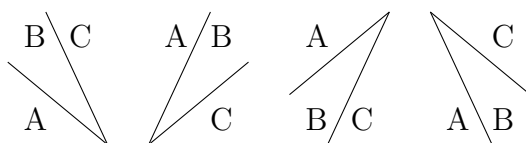


表 3: The value of some basic functions.

| <i>value</i> \ <i>x</i> | <i>x</i> = 1 | <i>x</i> = 2 | <i>x</i> = 3 | <i>x</i> = 4 |
|-------------------------|--------------|--------------|--------------|--------------|
| <i>y</i> | | | | |
| $y = x$ | 1 | 2 | 3 | 4 |
| $y = x^2$ | 1 | 4 | 9 | 16 |
| $y = x^3$ | 1 | 8 | 27 | 64 |

使用 \usepackage{diagbox} 宏包中的 \diagbox 命令制作双斜线表头。

```
\documentclass[12pt]{article}
\usepackage{diagbox}
\begin{document}
\begin{table}[htbp] % 设置位置参数
  \centering
  \caption{The values of some basic functions.}
  \begin{tabular}{l|cccc}
    \hline
    \diagbox[width=5em]{y}{value}{x} & x=1 & x=2 & x=3 & x=4 \\
    \hline
    y=x & 1 & 2 & 3 & 4 \\
    y=x^2 & 1 & 4 & 9 & 16 \\
    y=x^3 & 1 & 8 & 27 & 64 \\
    \hline
  \end{tabular}
  \label{table1}
\end{table}
\end{document}
```

| | $x = 1$ | $x = 2$ | $x = 3$ | $x = 4$ |
|-----------|----------------|---------|---------|---------|
| $y = x$ | 1 [*] | 2 | 3 | 4 |
| $y = x^2$ | 1 | 4 | 9 | 16 |
| $y = x^3$ | 1 | 8 | 27 | 64 |

¹ This is a remark example.

² This is another remark example and with a very long content, but the contents will be wrapped.

* This is 1.

3.2 插入表注

表格中的文本应当尽可能地保持简洁明了。因此，在保持简明的基础上，可以采用注释的方式以添加必要的细节对文本内容进行说明补充。通常，在以表格为载体的内容中，为了保持表格内容的完整性和独立性，我们往往不采用脚注`\footnote{}`的形式，而是将注释添加在表格底部（称之为表注）。在 L^AT_EX 中添加表注的方式有多种，其中比较常用的一种是使用 `threeparttable` 宏包及其相关命令，可以在表格底部生成与表格内容同宽的表注，并且当注释内容过长时可以实现自动换行，相比于其它方式更协调一致。

具体是在 `tabular` 环境外嵌套一层 `threeparttable` 环境，并在 `tabular` 环境之后将表注内容添加在 `tablenotes` 环境中，由此得到的表注将会显示在表格底部。如果需要将表格内容与表注建立关联关系，可以在表格内容的相应位置使用 `\tnote{索引标记}` 添加表注的索引标记，并且在 `tablenotes` 环境中使用 `item[索引标记]` 命令创建这项表注。

```
使用 threeparttable 宏包添加表注。
\documentclass[12pt]{article}
\usepackage{booktabs}
\usepackage{threeparttable}
\begin{document}
\begin{table}
  \centering
  \begin{threeparttable}
    \begin{tabular}{l|cccc}
      \toprule
      &  $x=1$  &  $x=2$  &  $x=3$  &  $x=4$  \\
      \midrule
       $y=x$  & 1* & 2 & 3 & 4 \\
       $y=x^2$  & 1 & 4 & 9 & 16 \\
       $y=x^3$  & 1 & 8 & 27 & 64 \\
      \bottomrule
    \end{tabular}
  \end{threeparttable}
\end{table}
```



```

\begin{tablenotes}
  \footnotesize
  \item[1] This is a remark example.
  \item[2] This is another remark example and with a very long
            content, but the contents will be wrapped.
  \item[*] This is 1.
\end{tablenotes}
\end{threeparttable}
\end{table}
\end{document}

```

4 调整表格样式

通过调用一些宏包及命令可以定制表格样式，从而创建更符合要求的表格。对表格样式的调整可以分为以下 7 个方面：表格尺寸、单元格自动对齐与换行、小数点对齐、行高、列宽、线宽、以及表格字体大小。

4.1 表格尺寸

如果想要修改表格尺寸，首先使用 `\usepackage{graphicx}` 语句调用 `graphicx` 宏包，并使用 `\resizebox{宽度}{高度}{内容}` 命令，该命令以 `tabular` 环境构建的表格作为内容。为了避免产生不协调的尺寸，在设置参数时只需要设置 {宽度} 和 {高度} 中的其中一个即可，另一个以 `!` 作为参数，表示根据宽高比进行自动调整。

This is the description for the following table. This is the description for the following table. This is the description for the following table. This is the description for the following table.

表 4: Title of a table.

| Column1 | Column2 | Column3 | Column4 |
|---------|---------|---------|---------|
| A1 | A2 | A3 | A4 |
| B1 | B2 | B3 | B4 |
| C1 | C2 | C3 | C4 |

```

\documentclass[12pt]{article}
\usepackage{graphicx}
\begin{document}
This is the description for the following table.

```

```

\begin{table}[h]
\centering
\caption{Title of a table.}
\label{first label}
\resizebox{0.8\textwidth}{!}{
\begin{tabular}{|l|l|l|l|}
\hline
Column1 & Column2 & Column3 & Column4 \\
\hline
A1 & A2 & A3 & A4 \\
\hline
B1 & B2 & B3 & B4 \\
\hline
C1 & C2 & C3 & C4 \\
\hline
\end{tabular}}
\end{table}
\end{document}

```

4.2 单元格自动对齐与换行

使用列类型参数 l、c 或 r 可以对每列的单元格设置左对齐、横向居中对齐和右对齐，但由此创建的单元格不仅无法设置顶部对齐、纵向居中对齐、以及底部对齐方式，而且单元格内容不论长短都被拉长为一行，显得不够灵活。下面介绍几种方式用于实现单元格自动对齐与换行。

4.2.1 使用 array 宏包实现单元格自动对齐与换行

首先在导言区使用 `\usepackage{array}` 语句声明调用 array 宏包，该宏包提供了以下 6 个列类型参数分别对应不同的对齐方式：

- 首先在导言区使用 `\usepackage{array}` 语句声明调用 array 宏包，该宏包提供了以下 6 个列类型参数分别对应不同的对齐方式：
- `m{列宽}`：单元格内容将根据设置的列宽自动换行，并且对齐方式为纵向居中对齐
- `b{列宽}`：单元格内容将根据设置的列宽自动换行，并且对齐方式为底部对齐
- `>\raggedright\arraybackslash`：将一列的单元格内容设置为左对齐
- `>\centering\arraybackslash`：将一列的单元格内容设置为横向居中对齐
- `>\raggedleft\arraybackslash`：将一列的单元格内容设置为右对齐

默认情况下，如果单独使用 p、m 或 b 参数，默认为左对齐。我们可以对上述参数进行组合使用，从而获得不同的对齐效果。需要注意的是，此时应使用 \tabularnewline 取代 \\ 符号作为表格一行的结束。通过调用 array 宏包的方式虽然可以实现自动换行，但常常需

表 5: Title of a table.

| | | | |
|---------|--------------------|-------------------------------|------------------------------------------|
| Column1 | Column2 Column2 | Column3 Column3 Column3 | Column4 Column4 Column4 Column4 |
| Value1 | Value2 Value2 | Value3 Value3 Value3 | Value4 Value4 Value4 Value4 |
| Value1 | Value2 Value2 | Value3 Value3 Value3 | Value4 Value4 Value4 Value4 |

要经过反复试验才能获得想要的宽度，更方便的方式是使用 tabularx 宏包或 tabulary 宏包及其相关命令自动计算列宽。对于涉及文本的表格，更推荐使用 tabulary 宏包。下面分别介绍通过这两个宏包及其命令如何实现自动换行

```
\documentclass[12pt]{article}
\usepackage{array}
\begin{document}
\begin{table}[h]
\centering
  \caption{Title of a table.}
  \label{first label}
  \begin{tabular}{|>{\raggedright\arraybackslash}m{2.3cm}|>{\centering\arraybackslash}m{2.3cm}|>{\centering\arraybackslash}m{2.3cm}|>{\raggedleft\arraybackslash}m{2.3cm}|}
    \hline
    Column1 & Column2 & Column2 & Column3 & Column3 & Column3 & Column4
      Column4 & Column4 & Column4 \tabularnewline
    \hline
    Value1 & Value2 & Value2 & Value3 & Value3 & Value3 & Value4 & Value
      4 & Value4 \tabularnewline
    \hline
    Value1 & Value2 & Value2 & Value3 & Value3 & Value3 & Value4 & Value
      4 & Value4 \tabularnewline
    \hline
  \end{tabular}
\end{table}
\end{document}
```

4.2.2 使用 tabularx 宏包实现自动换行

首先在导言区声明调用 tabularx 宏包，然后使用 `\begin{tabularx} \end{tabularx}` 环境取代 `\begin{tabular} \end{tabular}` 环境创建表格内容，tabularx 环境的使用方式与 tabular 类似，不同之处主要在于：`\begin{tabularx}{表格宽度}{列类型}` 中应设置表格宽度；在 tabularx 环境中，对于需要自动换行的列，其列类型应设置为大写的 X。X 参数可以与 `>\raggedright\arraybackslash`、`>\centering\arraybackslash` 或 `>\raggedleft\arraybackslash` 进行组合使用，从而修改单元格的对齐方式。

表 6: Title of a table.

| Column1 | Column2 | Column3 | Column4 |
|-------------------------------------------------------|-------------------------------------------------------|-------------------------------------------------------|-------------------------------------------------------|
| This is Value1. This is Value1. | This is Value2. This is Value2. | This is Value3. This is Value3. | This is Value4. This is Value4. |
| This is Value1. This is Value1. This is Value1. | This is Value2. This is Value2. This is Value2. | This is Value3. This is Value3. This is Value3. | This is Value4. This is Value4. This is Value4. |

调用 tabularx 宏包并设置列类型参数 X 从而实现单元格内容自动换行。

```
\documentclass[12pt]{article}
\usepackage{array}
\usepackage{tabularx} % 调用 tabularx 宏包
\begin{document}
\begin{table}[h]
\centering
\caption{Title of a table.}
\label{first label}
\begin{tabularx}{\linewidth}{|X|X|X|>\centering\arraybackslashX|} %
  将需要自动换行的列的列类型参数设为X
\hline
Column1 & Column2 & Column3 & Column4 \\
\hline
This is Value1. This is Value1. & This is Value2. This is Value2. &
  This is Value3. This is Value3. & This is Value4. This is Value
  4. \\
\hline
This is Value1. This is Value1. This is Value1. & This is Value2.
  This is Value2. This is Value2. & This is Value3. This is Value
  3. This is Value3. & This is Value4. This is Value4. This is
  Value4. \\
\hline
```

```
\end{tabularx}
\end{table}
\end{document}
```

4.2.3 使用 tabulary 宏包实现自动换行

类似地,调用 tabulary 宏包并使用`\begin{tabulary}{表格宽度}{列类型}` `\end{tabulary}` 环境创建表格。对于需要自动换行的列,只需将列类型改为大写字母即可,即,大写 L 表示左对齐并自动换行、大写 C 表示居中对齐并自动换行、大写 R 表示右对齐并自动换行。

表 7: Title of a table.

| Column1 | Column2 | Column3 | Column4 |
|-----------------|-----------------|-----------------|-----------------|
| This is Value1. | This is Value2. | This is Value3. | This is Value4. |
| This is Value1. | This is Value2. | This is Value3. | This is Value4. |
| This is Value1. | This is Value2. | This is Value3. | This is Value4. |
| This is Value1. | This is Value2. | This is Value3. | This is Value4. |
| This is Value1. | This is Value2. | This is Value3. | This is Value4. |

调用 tabulary 宏包并设置大写列类型参数 (L、C 和 R) 从而实现单元格内容自动换行。

```
\documentclass[12pt]{article}
\usepackage{array}
\usepackage{tabulary} % 调用 tabulary 宏包
\begin{document}
\begin{table}[h]
\centering
\caption{Title of a table.}
\label{first label}
\begin{tabulary}{\linewidth}{|L|C|C|R|} % 将需要自动换行的列的列类型参数改为大写
\hline
Column1 & Column2 & Column3 & Column4 \\
\hline
This is Value1. This is Value1. & This is Value2. This is Value2. & This is Value3. This is Value3. & This is Value4. This is Value4. \\
\hline
This is Value1. This is Value1. This is Value1. & This is Value2. This is Value2. This is Value2. & This is Value3. This is Value3. & This is Value4. This is Value4. \\
\hline
\end{tabulary}
\end{table}
```

```

\end{tabulary}
\end{table}
\end{document}

```

4.2.4 使用\parbox 命令实现人工换行

我们也可以通过使用\parbox 命令对表格内容进行强制换行:

| | | | |
|---|---|---|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| a | b | c | d |
| a | b | c | In probability theory and statistics, the continuous uniform distribution or rectangular distribution is a family of symmetric probability distributions. |

用\parbox命令来实现单元格中文本强制换行。

```

\documentclass{article}
\begin{document}
\begin{center}
\begin{tabular}{|c|c|c|c|}
\hline
a & b & c & d \\
\hline
a & b & c & \parbox[t]{5cm}{In probability theory and statistics ,
the continuous uniform distribution\\ or rectangular
distribution is a family of symmetric probability distributions
.} \\
\hline
\end{tabular}
\end{center}
\end{document}

```

4.3 小数点对齐

为了更好地描述数据,在表格中常常将数据在小数点处进行对齐,在 L^AT_EX 中我们可以通过使用 dcolumn 包实现这一目的。这个包提供了一个名为 D 的列类型,可以方便实现基于小数点的数字对齐以及基于其它符号的对齐,使用方式为 D{输入符号}{输出符号}{符号后的数字位数}。对于基于小数点的数字对齐,输入符号一般为“.”;有

时需要根据特定符号进行数字对齐，比如千分位逗号，这时输入符号即为“,”。例如，`D{.}{\cdot}{2}` 表示将某列的数据根据“.”符号对齐，输出时将该符号显示为点乘符号，并且显示 2 个小数位数字。

列类型 D 可以像其它列类型一样在表格环境的开始命令处直接进行设置，但会导致语句过长，所以一般使用 `array` 宏包的 `\newcolumntype` 命令定义一个新的列类型，并为这个列类型赋予一个比较短的名称以方便调用。定义新的列类型的语句为 `\newcolumntype{新列类型名称}[新列类型的参数个数]{定义新列类型}`，例如：`\newcolumntype{d}[1]{D{.}{\cdot}{#1}}` 表示创建一个名为 d 的新列类型，该列类型的内容为 `D{.}{\cdot}` 符号后的数字位数，其中数字位数是传给 d 的参数。

| Left | Center | Right | Decimal |
|-------|--------|-------|---------|
| 1.1 | 1.1 | 1.1 | 1.1 |
| 33.3 | 33.3 | 33.3 | 33.3 |
| 3.333 | 3.333 | 3.333 | 3.333 |

调用 `dcolumn` 宏包和列类型 D 来实现表格数据的小数点对齐。

```
\documentclass[12pt]{article}
\usepackage{dcolumn}
\newcolumntype{d}[1]{D{.}{\cdot}{#1}}
\begin{document}
  \begin{tabular}{|l|c|r|d{3}|}
    \hline
    Left & Center & Right & \mathrm{Decimal}\\
    \hline
    1.1 & 1.1 & 1.1 & 1.1\\
    \hline
    33.3 & 33.3 & 33.3 & 33.3\\
    \hline
    3.333 & 3.333 & 3.333 & 3.333\\
    \hline
  \end{tabular}
\end{document}
```

4.4 行高

如果需要调整表格整体行高，可以在导言区使用 `\renewcommand{\arraystretch}{行高倍数}` 命令，从而根据设置的行高倍数在默认值的基础上对行高进行扩大或缩小。

另一种调整行高的方式是通过在每行的结束标志 `\\` 后加上行高增减量选项，即 `\\[行高增减量]`，从而在默认值的基础上对各行行高进行增减。

| Column1 | Column2 | Column3 | Column4 |
|---------|---------|---------|---------|
| A1 | A2 | A3 | A4 |
| B1 | B2 | B3 | B4 |
| C1 | C2 | C3 | C4 |

使用 `\renewcommand{\arraystretch}{2}` 命令将表格整体行高设为两倍行距。

```
\documentclass[12pt]{article}
\renewcommand{\arraystretch}{2}
\begin{document}
\begin{tabular}{|c|c|c|c|}
\hline
Column1 & Column2 & Column3 & Column4\\
\hline
A1 & A2 & A3 & A4\\
\hline
B1 & B2 & B3 & B4\\
\hline
C1 & C2 & C3 & C4\\
\hline
\end{tabular}
\end{document}
```

使用 `\\[行高增减量]` 命令为表格各行设置不同的行高。

```
\documentclass[12pt]{article}
\begin{document}
\begin{tabular}{|c|c|c|c|}
\hline
Column1 & Column2 & Column3 & Column4\\
\hline
A1 & A2 & A3 & A4\\
\hline
B1 & B2 & B3 & B4\\
\hline
C1 & C2 & C3 & C4\\
\hline
\end{tabular}
\end{document}
```


4.5 列宽

也可以在导言区使用`\setlength{\tabcolsep}{文本和列分隔线的间距}`命令修改表格列宽，默认情况下，单元格内容与列分隔线的间距为 6pt。

```
使用\setlength{\tabcolsep}{12pt}命令将表格单元格文本和列分隔线的间距设为12
pt。
\documentclass[12pt]{article}
\setlength{\tabcolsep}{12pt}
\begin{document}
\begin{tabular}{|c|c|c|c|}
\hline
Column1 & Column2 & Column3 & Column4\\
\hline
A1 & A2 & A3 & A4\\
\hline
B1 & B2 & B3 & B4\\
\hline
C1 & C2 & C3 & C4\\
\hline
\end{tabular}
\end{document}
```

4.6 线宽

通过在导言区使用`\setlength{\arrayrulewidth}{线宽}`命令，可以修改表格线宽，默认为 0.4pt。然而当线宽设置过大时，可能导致表格线交叉处不连续的情况。对此，在导言区调用 `xcolor` 宏包、并设置 `table` 选项可以解决。

```
在导言区使用\usepackage[table]{xcolor}命令调用设置了table选项的xcolor宏包，
并使用\setlength{\arrayrulewidth}{线宽}命令设置表格线宽。
\documentclass[12pt]{article}
\usepackage[table]{xcolor} % 调用设置了table选项的xcolor宏包
\setlength{\arrayrulewidth}{2pt} % 修改表格线宽
\begin{document}
\begin{tabular}{|l|l|l|l|}
\hline
Column1 & Column2 & Column3 & Column4\\
\hline
A1 & A2 & A3 & A4\\
\hline
B1 & B2 & B3 & B4\\
\hline
\end{tabular}
```

```

\hline
C1 & C2 & C3 & C4\
\hline
\end{tabular}
\end{document}

```

4.7 表格字体大小

在文本编辑中我们知道，调整字体大小的方式既有全局方式也有局部方式，其中，全局方式是通过在文档类型中指定字体大小，例如`\documentclass[12pt]{article}`，而局部方式则是通过一系列设置字体大小的命令，例如`\large`、`Large`、`huge`、`\fontsize`等，在全局字体大小的基础上作进一步的调整。类似地，在使用 LaTeX 创建表格时，我们也可以对表格字体大小做全局或局部调整。

表 8: 正常字体

| | $x = 1$ | $x = 2$ | $x = 3$ | $x = 4$ |
|-----------|---------|---------|---------|---------|
| $y = x$ | 1 | 2 | 3 | 4 |
| $y = x^2$ | 1 | 4 | 9 | 16 |
| $y = x^3$ | 1 | 8 | 27 | 64 |

表 9: Large 字体

| | $x = 1$ | $x = 2$ | $x = 3$ | $x = 4$ |
|-----------|---------|---------|---------|---------|
| $y = x$ | 1 | 2 | 3 | 4 |
| $y = x^2$ | 1 | 4 | 9 | 16 |
| $y = x^3$ | 1 | 8 | 27 | 64 |

使用\Large命令调整表格局部字体大小。

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

% 正常字体大小
\begin{table}[htp]
  \centering
  \begin{tabular}{l|cccr}
    \hline
    & $x=1$ & $x=2$ & $x=3$ & $x=4$ \\
    \hline
    $y=x$ & 1 & 2 & 3 & 4 \\
    $y=x^2$ & 1 & 4 & 9 & 16 \\
    $y=x^3$ & 1 & 8 & 27 & 64 \\
    \hline
  \end{tabular}
\end{table}

% Large 字体大小
\begin{table}[htp]
  \Large
  \centering
  \begin{tabular}{l|cccr}
    \hline
    & $x=1$ & $x=2$ & $x=3$ & $x=4$ \\
    \hline
    $y=x$ & 1 & 2 & 3 & 4 \\
    $y=x^2$ & 1 & 4 & 9 & 16 \\
    $y=x^3$ & 1 & 8 & 27 & 64 \\
    \hline
  \end{tabular}
\end{table}
\end{document}
```

使用 `\fontsize` 命令通过具体设置来调整表格局部字体大小。

```
\documentclass[12pt]{article}
\begin{document}
% 正常字体大小
\begin{table}[htp]
  \centering
  \begin{tabular}{l|ccc}
    \hline
    &  $x=1$  &  $x=2$  &  $x=3$  &  $x=4$  \\
    \hline
     $y=x$  & 1 & 2 & 3 & 4 \\
     $y=x^2$  & 1 & 4 & 9 & 16 \\
     $y=x^3$  & 1 & 8 & 27 & 64 \\
    \hline
  \end{tabular}
\end{table}
% 将字体大小设为18pt、行距设为24pt
\begin{table}[htp]
  \fontsize{18pt}{24pt}\selectfont
  \centering
  \begin{tabular}{l|ccc}
    \hline
    &  $x=1$  &  $x=2$  &  $x=3$  &  $x=4$  \\
    \hline
     $y=x$  & 1 & 2 & 3 & 4 \\
     $y=x^2$  & 1 & 4 & 9 & 16 \\
     $y=x^3$  & 1 & 8 & 27 & 64 \\
    \hline
  \end{tabular}
\end{table}
\end{document}
```

4.8 文字环绕表格

如果想要实现文字环绕表格效果，可以使用 `wrapfig` 宏包，并使用其提供的 `wraptable` 环境嵌套 `tabular` 环境创建表格，从而达到文字环绕表格的效果。

In descriptive statistics, a box plot or boxplot is a method for graphically depicting groups of numerical data through their quartiles. Box plots may also have lines extending from the boxes (whiskers) indicating variability outside the upper and lower quartiles, hence the terms box-and-whisker plot and box-and-whisker diagram. Outliers may be plotted as individual points. Box plots are non-parametric: they display variation in

samples of a statistical population without making any assumptions of the underlying statistical distribution (though Tukey's boxplot assumes symmetry for the whiskers and normality for their length).

The spacings between the different parts of the box indicate the degree of dispersion (spread) and skewness in the data, and show outliers. In addition to the points themselves, they allow one to visually estimate various L-estimators, notably the interquartile range, midhinge, range, mid-range, and trimean. Box plots can be drawn either horizontally or vertically. Box plots received their name from the box in the middle, and from the plot that they are.

| | $x = 1$ | $x = 2$ | $x = 3$ | $x = 4$ |
|-----------|---------|---------|---------|---------|
| $y = x$ | 1 | 2 | 3 | 4 |
| $y = x^2$ | 1 | 4 | 9 | 16 |
| $y = x^3$ | 1 | 8 | 27 | 64 |

```

使用 wraptable 环境嵌套 tabular 环境创建表格，实现文字环绕表格；并使用 \begin{
  wraptable}{r}{8cm} 将表格置于文字右侧，同时将表格和文字的距离设为 8cm。
\documentclass[12pt]{article}
\usepackage{wrapfig}
\begin{document}
In descriptive statistics, a box plot or boxplot is a method for
graphically depicting groups of numerical data through their quartiles.
Box plots may also have lines extending from the boxes (whiskers)
indicating variability outside the upper and lower quartiles, hence the
terms box-and-whisker plot and box-and-whisker diagram. Outliers may be
plotted as individual points. Box plots are non-parametric: they display
variation in samples of a statistical population without making any
assumptions of the underlying statistical distribution (though Tukey's
boxplot assumes symmetry for the whiskers and normality for their length
).
\begin{wraptable}{r}{8cm}
\centering
\begin{tabular}{lcccc}
\hline
&  $x=1$  &  $x=2$  &  $x=3$  &  $x=4$  \\
\hline
 $y=x$  & 1 & 2 & 3 & 4 \\
 $y=x^2$  & 1 & 4 & 9 & 16 \\
 $y=x^3$  & 1 & 8 & 27 & 64 \\
\hline
\end{tabular}
\end{wraptable}
\end{document}

```

The spacings between the different parts of the box indicate the degree of dispersion (spread) and skewness in the data, and show outliers. In addition to the points themselves, they allow one to visually estimate various L-estimators, notably the interquartile range, midhinge, range, mid-range, and trimean. Box plots can be drawn either horizontally or vertically. Box plots received their name from the box in the middle, and from the plot that they are.

`\end{document}`

5 创建彩色表格

有时，根据表达需要，表格中的内容需要突出显示，彩色表格即为突出显示的一种重要方式。通过对表格的单元格、行或列填充颜色，可以创建不同的彩色表格。为此，首先应在导言区使用`\usepackage[table]{xcolor}` 声明语句，通过调用 xcolor 宏包提供的相关命令可以实现颜色填充。

填充单元格时，使用`\cellcolor{单元格填充颜色}` 单元格内容命令定义单元格内容即可

| Column1 | Column2 | Column3 | Column4 |
|---------|---------|---------|---------|
| A1 | A2 | A3 | A4 |
| B1 | B2 | B3 | B4 |
| C1 | C2 | C3 | C4 |

在导言区使用`\usepackage[table]{xcolor}`命令调用设置了table选项的xcolor宏包，并使用`\cellcolor`命令定义具有颜色填充效果的单元格。

`\documentclass[12pt]{article}`

`\usepackage[table]{xcolor}` % 调用设置了table选项的xcolor宏包

`\begin{document}`

`\begin{tabular}{|l|l|l|l|}`

`\hline`

`Column1 & Column2 & Column3 & Column4\\`

`\hline`

`\cellcolor{red!80}A1 & A2 & A3 & A4\\` % 使用`\cellcolor`命令设置单元格填充颜色

`\hline`

`\cellcolor{red!50}B1 & B2 & B3 & B4\\`

```

\hline
\cellcolor{red!20}C1 & C2 & C3 & C4\\
\hline
\end{tabular}
\end{document}

```

为了达到更好的可视化效果，有时候需要为表格的奇数行和偶数行交替设置不同的填充颜色，那么只需要在 tabular 环境前使用 `\rowcolors{开始填充的行编号}{第一个行填充颜色}{第二个行填充颜色}` 命令即可

| Column1 | Column2 | Column3 | Column4 |
|---------|---------|---------|---------|
| A1 | A2 | A3 | A4 |
| B1 | B2 | B3 | B4 |
| C1 | C2 | C3 | C4 |

```

\documentclass[12pt]{article}
\usepackage[table]{xcolor} % 调用设置了table选项的xcolor宏包
\begin{document}

\rowcolors{2}{red!50}{red!20} % 设置表格交替填充行颜色
\begin{tabular}{|l|l|l|l|}
\hline
Column1 & Column2 & Column3 & Column4\\
\hline
A1 & A2 & A3 & A4\\
\hline
B1 & B2 & B3 & B4\\
\hline
C1 & C2 & C3 & C4\\
\hline
\end{tabular}

\end{document}

```

当然，我们也可以设置列填充颜色，只需要在列类型参数中加上`>\columncolor{列填充颜色}`即可：

```
\documentclass[12pt]{article}
\usepackage[table]{xcolor} % 调用设置了table选项的xcolor宏包
\begin{document}
\begin{tabular}{|>\columncolor{red!50}}1|>\columncolor{red!20}}1|>\columncolor{red!50}}1|>\columncolor{red!20}}1|} % 设置列填充颜色
\hline
Column1 & Column2 & Column3 & Column4\\
\hline
A1 & A2 & A3 & A4\\
\hline
B1 & B2 & B3 & B4\\
\hline
C1 & C2 & C3 & C4\\
\hline
\end{tabular}
\end{document}
```

6 创建三线表格

`booktabs` 宏包提供了更美观的行分隔线创建命令，常用于创建三线表格。其中，`\toprule` 命令常用于创建表格顶线、`\bottomrule` 命令常用于创建表格底线、`\midrule` 命令常用于创建表格标题栏和表格内容的分隔线、以及`\cmidrule{起始列号-终止列号}` 命令用于创建标题栏内部的分隔线并设置分隔线的跨越范围。

| Type1 | | | |
|---------|---------|---------|---------|
| Column1 | Column2 | Column3 | Column4 |
| A1 | A2 | A3 | A4 |
| B1 | B2 | B3 | B4 |
| C1 | C2 | C3 | C4 |

调用 `booktabs` 宏包及其相关命令创建三线表。

```
\documentclass[12pt]{article}
\usepackage{booktabs}
\usepackage{multirow}
\begin{document}
```



```

\begin{tabular}{cccc}
\toprule
\multicolumn{2}{c}{\textbf{Type1}} & & \\
\cmidrule{1-2}
Column1 & Column2 & Column3 & Column4\\
\midrule
A1 & A2 & A3 & A4\\
B1 & B2 & B3 & B4\\
C1 & C2 & C3 & C4\\
\bottomrule
\end{tabular}
\end{document}

```

7 创建跨页表格

当表格太长时，使用 `tabular` 环境创建的表格会被裁剪掉页面放不下的部分。如果想让表格在行数太多时实现自动分页，可以通过调用 `longtable` 宏包并使用 `longtable` 环境创建表格。

在 `longtable` 环境中创建表格的方式与使用 `table` 和 `tabular` 嵌套环境类似，我们也能使用 `\caption`、`\label` 命令分别创建表格标题和索引标签。不同之处主要在于，`longtable` 环境中可以设置跨页表格在每一页的重复表头和表尾，按照使用顺序，包括以下四个命令：

- `\endfirsthead`：`\begin{longtable}` 和 `\endfirsthead` 之间的内容只会出现在表格第一页的表头部分；
- `\endhead`：`\endfirsthead` 和 `\endhead` 之间的内容将会出现在表格除第一页之外的表头部分；
- `\endfoot`：`\endhead` 和 `\endfoot` 之间的内容将会出现在除表格最后一页之外的表尾部分；
- `\endlastfoot`：`\endfoot` 和 `\endlastfoot` 之间的内容只会出现在表格最后一页的表尾部分。

以上四个命令需要放置在 `longtable` 环境的开始处。

表 10: Title of a table

| Column1 | Column2 | Column3 | Column4 |
|---------|---------|---------|---------|
| A1 | A2 | A3 | A4 |
| B1 | B2 | B3 | B4 |
| C1 | C2 | C3 | C4 |
| A1 | A2 | A3 | A4 |
| B1 | B2 | B3 | B4 |
| C1 | C2 | C3 | C4 |
| C1 | C2 | C3 | C4 |
| A1 | A2 | A3 | A4 |
| B1 | B2 | B3 | B4 |
| C1 | C2 | C3 | C4 |
| C1 | C2 | C3 | C4 |
| A1 | A2 | A3 | A4 |
| B1 | B2 | B3 | B4 |
| C1 | C2 | C3 | C4 |
| C1 | C2 | C3 | C4 |
| A1 | A2 | A3 | A4 |
| B1 | B2 | B3 | B4 |
| C1 | C2 | C3 | C4 |
| B1 | B2 | B3 | B4 |
| C1 | C2 | C3 | C4 |
| C1 | C2 | C3 | C4 |
| A1 | A2 | A3 | A4 |
| B1 | B2 | B3 | B4 |
| C1 | C2 | C3 | C4 |
| C1 | C2 | C3 | C4 |
| A1 | A2 | A3 | A4 |
| B1 | B2 | B3 | B4 |
| C1 | C2 | C3 | C4 |
| C1 | C2 | C3 | C4 |
| A1 | A2 | A3 | A4 |
| B1 | B2 | B3 | B4 |
| C1 | C2 | C3 | C4 |
| A1 | A2 | A3 | A4 |
| B1 | B2 | B3 | B4 |

表 10: Title of a table - Continued

| Column1 | Column2 | Column3 | Column4 |
|----------------------|---------|---------|---------|
| C1 | C2 | C3 | C4 |
| End of table. | | | |

调用 longtable 宏包及环境创建跨页表格。

```

\documentclass[12pt]{article}
\usepackage{longtable}
\usepackage{multirow}
\begin{document}
\begin{longtable}[c]{cccc}
    % 创建表格第一页的表头部分
    \caption{Title of a table}\\
    \hline
    Column1 & Column2 & Column3 & Column4\\
    \hline
    \endfirsthead
    % 创建表格除第一页之外的表头部分
    \caption{Title of a table – Continued}\\
    \hline
    Column1 & Column2 & Column3 & Column4\\
    \hline
    \endhead
    % 创建表格除最后一页之外的表尾部分
    \hline
    \endfoot
    % 创建表格最后一页的表尾部分
    \multicolumn{4}{c}{\textbf{End of table.}}\\
    \hline
    \endlastfoot
    % 表格内容
    A1 & A2 & A3 & A4\\
    B1 & B2 & B3 & B4\\
    C1 & C2 & C3 & C4\\
    A1 & A2 & A3 & A4\\
    ... % 省略中间部分
    B1 & B2 & B3 & B4\\
    C1 & C2 & C3 & C4\\
    \hline
\end{longtable}
\end{document}

```

8 旋转表格

当表格列数太多时，横向表格的展现效果较差，这时需要将表格旋转 90 度，以纵向表格的形式展现。在 LaTeX 中可以通过调用 rotating 宏包，并使用 sidewaysstable 环境取代 table 环境、嵌套 tabular 环境创建纵向表格（表格逆时针旋转 90 度）。

调用 rotating 宏包及 sidewaysstable 环境创建纵向表格。

```
\documentclass[12pt]{article}
\usepackage{rotating}
\usepackage{booktabs}
\begin{document}
\begin{sidewaysstable}[h]
\centering
  \caption{Title of a table.}
  \label{first label}
  \begin{tabular}{cccc}
    \toprule
    Column1 & Column2 & Column3 & Column4\\
    \midrule
    A1 & A2 & A3 & A4\\
    B1 & B2 & B3 & B4\\
    C1 & C2 & C3 & C4\\
    \bottomrule
  \end{tabular}
\end{sidewaysstable}
\end{document}
```

表 11: Title of a table.

| Column1 | Column2 | Column3 | Column4 |
|---------|---------|---------|---------|
| A1 | A2 | A3 | A4 |
| B1 | B2 | B3 | B4 |
| C1 | C2 | C3 | C4 |

9 导入表格

一个表格有时候可能涉及大量数据，这时在 L^AT_EX 中手动输入数据并创建表格显然不够灵活，对此，可以通过导入 csv 文件数据的方式创建表格。L^AT_EX 提供了 csvsimple、pgfplotstable、csvtools 等宏包可以帮助用户实现基于 csv 文件快速导入表格，其中，使用 csvsimple 宏包及其命令是一种比较常用的方式，我们下面将对其展开介绍。

9.1 快速创建表格

首先在导言区使用 `\usepackage{csvsimple}` 语句声明对 csvsimple 宏包的调用，然后在文档主体内容中使用 `\csvautotabular{csv 文件名或文件路径}` 命令即可导入 csv 文件从而快速创建表格。作为导入数据的 csv 文件，既可以预先放在指定目录下，也可以在 filecontents 环境中创建。

通过上述语句，创建了一个名为“dataimport.csv”的 csv 文件。

```
\begin{filecontents}{dataimport.csv}
  COLUMNa,COLUMNb,COLUMNc,COLUMNd
  1.1,2.2,3.3,4.4
  11.1,22.2,33.3,44.4
  1.111,2.222,3.333,4.444
\end{filecontents}
```

使用 filecontents 环境创建一个名为“dataimport.csv”的 csv 文件，并调用 csvsimple 宏包及 `\csvautotabular` 命令快速根据 csv 文件创建表格。

```
\begin{filecontents}{dataimport.csv}
COLUMNa,COLUMNb,COLUMNc,COLUMNd
1.1,2.2,3.3,4.4
11.1,22.2,33.3,44.4
1.111,2.222,3.333,4.444
\end{filecontents}

\documentclass{article}
\usepackage{csvsimple}
\begin{document}
\begin{table}
\centering
\caption{A table imported from csv file}
\label{labeloftable1}
\csvautotabular{dataimport.csv}
\end{table}
\end{document}
```

9.2 创建三线表

也可以结合 booktabs 宏包，使用 `\csvautobooktabular` 命令自动读取 csv 文件并创建更专业的三线表格。

```

使用 filecontents 环境创建一个名为 “dataimport.csv” 的 csv 文件，并调用
    csvsimple 和 booktabs 宏包，使用 \csvautobooktabular 命令快速根据 csv 文件创建
    三线表格。
\begin{filecontents}{dataimport.csv}
COLUMNa,COLUMNb,COLUMNc,COLUMNd
1.1,2.2,3.3,4.4
11.1,22.2,33.3,44.4
1.111,2.222,3.333,4.444
\end{filecontents}

\documentclass{article}
\usepackage{csvsimple}
\usepackage{booktabs}
\begin{document}
\begin{table}
\centering
    \caption{A table imported from csv file}
    \label{labeloftable1}
    \csvautobooktabular{dataimport.csv}
\end{table}
\end{document}

```

9.3 设置表格属性

如果想要调整导入的表格样式、表头、指定导入列等属性，可以使用 `\csvreader[属性设置]{csv 文件名或文件路径}{定义数据列名}{需要导入的数据列名}` 命令读取 csv 文件创建表格，并通过设置属性选项、指定需要导入的数据列名从而调整表格属性。在属性设置选项中，主要包括以下属性：

- tabular: 定义列类型。列类型个数应与需要导入的列数一致；
- table head: 定义表头，包括标题行的顶线、列名、以及底线。由此可以对各列名进行重定义或省略；
- late after line: 定义行分隔线。如，单行分隔线设置表示为 `late after line=\\hline`。

使用 `csvreader` 命令读取 `csv` 文件创建表格，并将表格列名从 “`COLUMNa,COLUMNb,COLUMNc,COLUMNd`” 重命名为 “`column1,column2,column3,column4`”，并分别设置数据列名 “`ca,cb,cc,cd`” 用于指定导入哪些列。

```
\begin{filecontents*}{dataimport.csv}
COLUMNa,COLUMNb,COLUMNc,COLUMNd
1.1,2.2,3.3,4.4
11.1,22.2,33.3,44.4
1.111,2.222,3.333,4.444
\end{filecontents*}

\documentclass{article}
\usepackage{csvsimple}
\begin{document}
\begin{table}
\centering
\caption{A table imported from csv file}
\label{labeloftable1}
\csvreader[tabular=|1|1|1|1|,
table head=\hline column1 & column2 & column3 & column4\\ \hline,
late after line=\\ \hline] % 表格属性设置
{dataimport.csv} % csv文件名
{COLUMNa =\ca, COLUMNb =\cb, COLUMNc =\cc, COLUMNd =\cd} % 定义数据列名
{\ca & \cb & \cc & \cd} % 需要导入的数据列名
\end{table}
\end{document}
```


使用 csvreader 命令读取 csv 文件创建表格，指定导入的数据列为前两列，并使用 \thecsvrow 命令增加行标签列。

```
\begin{filecontents}{dataimport.csv}
COLUMNa,COLUMNb,COLUMNc,COLUMNd
1.1,2.2,3.3,4.4
11.1,22.2,33.3,44.4
1.111,2.222,3.333,4.444
\end{filecontents}

\documentclass{article}
\usepackage{csvsimple}
\begin{document}
\begin{table}
\centering
\caption{A table imported from csv file}
\label{labeloftable1}
\csvreader[tabular=|1|1|1|,
table head=\hline & column1 & column2\\\hline,
late after line=\\\hline] % 表格属性设置
{dataimport.csv} % csv 文件名
{COLUMNa =\ca, COLUMNb =\cb, COLUMNc =\cc, COLUMNd =\cd} % 定义数据列名
{\thecsvrow & \ca & \cb} % 需要导入的数据列名
\end{table}
\end{document}
```

Part V

图形插入

1 插入浮动图片

LaTeX 中可以支持插入.pdf、.jpg、.jpeg、.png、*.eps 等常见格式的图片，而对于 LaTeX 不支持的图片文件格式，如 SVG 格式的矢量图，则需要先转换再插入。一般而言，读者可以通过截图、MS Visio 等绘图工具、或者 Matlab 等编程工具制作并导出目标图片。

在 LaTeX 中插入图片可以使用 graphicx 宏包，该宏包提供的 `\includegraphics[参数]{文件名或文件路径}` 命令可以用于插入图片，以及设置参数以调整图片样式，常用参数包括：

- width：设置图片宽度；
- height：设置图片高；
- scale：设置图片的缩放倍数；
- angle：设置图片的顺时针旋转角度（负值表示逆时针旋转）等。

一般而言，对于参数 height 和 width，只需要调整其中一个即可，另一个参数将根据图片比例进行自动缩放。而如果同时调整了参数 height 和 width（不推荐），可能会改变图片比例，导致图片变形。



在导言区使用`\usepackage{graphicx}`声明语句，在主体代码中使用`\includegraphics`命令插入图片，并调整图片样式参数。

```
\documentclass[12pt]{article}
\usepackage{graphicx}
\begin{document}
The following figure shows a beautiful butterfly.
\includegraphics[width = 0.5\textwidth]{butterfly.JPG} % 插入第一张图片
\vspace{12pt}
Rotate the figure by 90 degrees.
\includegraphics[width = 0.5\textwidth,angle = 90]{butterfly.JPG}
\end{document}
```

此外，`graphicx` 宏包提供了 `figure` 环境语句，通过嵌套 `\includegraphics` 命令可以以浮动体的形式插入图片，从而能够实现自动递增编号、设置位置控制参数、利用 `\caption` 命令创建标题名称等。

Figure 1 shows a beautiful butterfly.



图 1: A beautiful butterfly.

使用 `figure` 环境嵌套 `\includegraphics` 命令插入浮动图片，并使用 `\label` 命令为图片创建索引标签，然后在文本内容中使用 `\ref` 命令引用该图片。

```
\documentclass[12pt]{article}
\usepackage{graphicx}
\begin{document}
Figure \ref{fig:1} shows a beautiful butterfly.
\begin{figure}[htbp]
\centering
\label{fig:1}
\includegraphics[width = 0.8\textwidth]{butterfly.JPG}
\caption{A beautiful butterfly.}
\end{figure}
\end{document}
```

2 插入非浮动图片

通过 figure 环境插入图片虽然能够实现自动编号和创建图片标题，但创建结果为浮动图片，图片的显示位置与在代码中的位置未必一致。然而有时我们想要以非浮动体的形式插入图片，使得图片显示位置与代码中的位置一致，同时能够实现自动编号和创建标题，要实现这一效果，我们可以使用 minipage 环境或 center 环境替代 figure 环境插入图片，同时使用 caption 宏包提供的 `\captionof{figure}{图片标题名称}` 命令创建图片标题。

使用 minipage 环境插入图片的方式与 figure 环境类似，不同之处主要在于使用 minipage 环境插入的图片与上下文中的文本内容紧挨着，为了避免这种情况，minipage 环境前后可以使用 `\vspace` 纵向距离调整图片与文本的纵向空间距离。

Figure 2 shows a beautiful butterfly.



图 2: A beautiful butterfly.

```
Figure \ref{fig:2} shows a beautiful butterfly.
\vspace{12pt}
\begin{minipage}{\linewidth}
\centering
  \label{fig:2}
  \includegraphics[width = 0.6\linewidth]{LaTeX.png}
  \captionof{figure}{A beautiful butterfly.}
\end{minipage}
\vspace{12pt}
```

在论文写作中，有时需要将多个图片放在同一行进行排列以便于比较。在 figure 环境中，使用 minipage 环境即可实现图片并排显示，并连续编号。



图 3: LaTeX-1



图 4: LaTeX-2

```

\begin{figure}[htbp]
  \centering
  \begin{minipage}[t]{0.48\textwidth}
    \centering
    \includegraphics[width=6cm]{LaTeX.png}
    \caption{LaTeX-1}
  \end{minipage}%
  \begin{minipage}[t]{0.48\textwidth}
    \centering
    \includegraphics[width=6cm]{LaTeX.png}
    \caption{LaTeX-2}
  \end{minipage}
\end{figure}

```

3 插入图表目录

插入图目录或表目录的命令语句分别为`\listoffigures`和`\listoftables`,可以罗列`\caption`命令创建的图表标题名称,但对于使用`\caption*`命令创建的无编号的标题名称而言,则不会出现在目录中。

在一份专业文档中,目录总是和正文内容在不同页显示,为此可以使用`\newpage`命令进行分页;此外目录页中一般无页码编号,为此可以使用`\thispagestyle{empty}`取消当页页码设置,并在正文页之前使用`\pagenumbering{页码样式}`命令表示重新从1开始设置页码、同时设置页码样式。

默认的图目录名和表目录名分别是“List of Figures”、“List of Tables”,读者可以在导言区通过使用`\renewcommand{\listfigurename}{新图目录名}`命令修改图目录名、使用`\renewcommand{\listtablename}{新表目录名}`命令修改表目录名。

```

使用\listoffigures命令和\listoftables命令创建图目录和表目录,使用\
renewcommand命令修改图目录名和表目录名,并使用\newpage、\thispagestyle和
\pagenumbering命令对页码进行相应调整。
\documentclass{article}
\usepackage{graphicx}
\renewcommand{\listfigurename}{Figures}
\renewcommand{\listtablename}{Tables}
\begin{document}
\thispagestyle{empty} % 取消页码编号
\listoffigures
\listoftables
\newpage % 插入新页
\pagenumbering{arabic} % 设置页码样式为小写的阿拉伯数字

```

```

Here are three created tables ...
\begin{table}[h!]
% ...
\caption{The first table.}
\caption{The second table.}
\caption{The third table.}
\end{table}
Here are three inserted figures ...
\begin{figure}[h!]
% ...
\caption{The first figure.}
\caption*{The second figure.}
\caption{The third figure.}
% ...
\end{figure}
\end{document}

```

4 定制图表标题样式

使用`\caption`命令为浮动图片或者浮动表格创建的标题主要包含四部分：1 标题头部、2 自动编号、3 编号分隔符、以及 4 标题名称，如下图所示。其中，对标题名称的设置比较简单，下面分别就标题前三部分的调整方式展开介绍。

4.1 调整标题头部

默认情况下，图片和表格标题头部分别为“Figure”和“Table”，我们可以分别使用`\renewcommand{\figurename}`
`{新的图片标题头部}`和`\renewcommand{\tablename}{新的表格标题头部}`命令对其进行修改。



Fig 5: LaTeX

```

使用\renewcommand命令将图表题头部改为Fig。
\documentclass[12pt]{article}

```

```

\usepackage{graphicx}
\renewcommand{\figurename}{Fig}
\begin{document}
\begin{figure}
\centering
\includegraphics[width = 0.5\textwidth]{graphics/butterfly.JPG}
\caption{There is a beautiful butterfly.}
\label{butterfly}
\end{figure}
\end{document}

```

4.2 调整编号

创建浮动表格或者浮动图片时，LaTeX 会根据内置的计数器对其进行自动递增计数，计数值即为图表标题编号和引用编号，默认为小写的阿拉伯数字。

如果需要取消图表的自动编号，可以使用 caption 宏包提供的 `\captionsetup[浮动体类型]{labelformat=empty}` 命令，其中浮动体类型可以为 figure、subfigure、table、或 subtable，分别表示图片、子图、表格、子表。使用该命令后，指定浮动体类型下的所有浮动体将取消自动编号，但其标题和编号仍会显示在图表目录中。

使用 `\caption*` 命令取消部分表格的自动编号，同时使用 `\captionsetup` 命令取消所有图片的自动编号。

```

\documentclass{article}
\usepackage{graphicx}
\usepackage{caption}
\begin{document}
\captionsetup[figure]{labelformat=empty} % 取消所有图片的自动编号
Here are three created tables...
\begin{table}[h!]
% ...
\caption{The first table.}
\caption*{The second table.} % 取消该标题的自动编号
\caption{The third table.}
\end{table}
Here are three inserted figures...
\begin{figure}[h!]
% ...
\caption{The first figure.}
\caption{The second figure.}
\caption{The third figure.}
% ...
\end{figure}

```

```
\end{document}
```

如果想要修改图表编号样式，可以在导言区使用`\renewcommand{浮动体的自动计数器}{计数器样式}`命令。其中，浮动体的自动计数器名称可以为`\thefigure`、`\thesubfigure`、`\thetable`、或`\thesubtable`；定义计数器样式可以使用`\alph{浮动体类型}`、`\Alph{浮动体类型}`、`\Roman{浮动体类型}`、`\arabic{浮动体类型}`等命令。

```
使用\renewcommand命令调整图表标题头部和编号样式。
\documentclass{article}
\usepackage{graphicx}
\renewcommand{\figurename}{Fig} % 调整图片头部
\renewcommand{\tablename}{Tab} % 调整新表格头部
\renewcommand{\thefigure}{\Alph{figure}} % 调整图片编号样式为大写字母
\renewcommand{\thetable}{\alph{table}} % 调整表格编号样式为小写字母
\begin{document}
Here are three created tables...
\begin{table}[h!]
% ...
\caption{The first table.}
\caption{The second table.}
\caption{The third table.}
\end{table}
Here are three inserted figures...
\begin{figure}[h!]
% ...
\caption{The first figure.}
\caption{The second figure.}
\caption{The third figure.}
% ...
\end{figure}
\end{document}
```

4.3 调整编号分隔符

图表中的编号分隔符默认为英文冒号“:”，如果需要对其进行调整，可以使用`\captionsetup[浮动体类型]{设置 labelsep 选项}`命令，通过设置不同的`labelsep`选项实现，各选项及其含义如下：

- `colon`: 默认值，即英文冒号“:”
- `none`: 无编号分隔符

- period: 英文句号 “.”;
- space: 一个空格
- quad: 一个字符 “M” 大小的空格
- newline: 换行

使用 caption 宏包调整图标题编号分隔符为换行。

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

\usepackage{graphicx}
\usepackage{caption}
\renewcommand{\figurename}{Fig}
\captionsetup[figure]{labelsep=newline}

\begin{document}

\begin{figure}
\centering
\includegraphics[width = 0.5\textwidth]{graphics/butterfly.JPG}
\caption{There is a beautiful butterfly.}
\label{butterfly}
\end{figure}

\end{document}
```

5 插入子图

有时候需要将一组图片以子图的方式呈现，达到对比或者互补的效果。在 LaTeX 中，插入子图比较常用的方式是使用 subfigure 环境。

5.1 基本介绍

子图一般在 subfigure 环境中创建，多个子图环境嵌套在 figure 环境中从而形成同一组子图。subfigure 环境与 figure 环境的使用方式基本类似，可以为每个子图分别创建标题和索引标签，方便说明和引用。

Figure 6 contains sub-figure 6a, sub-figure 6b and sub-figure 6c.

在上例中，每个子图都用到了两次宽度设置选项，分别具有不同含义：

1. `\begin{subfigure}{.3\linewidth}` 表示将该子图环境的宽度设置为页面宽度的 0.3 倍；



Fig 6: Three flowers with different colors.

2. `\includegraphics[width=.5\linewidth]` 表示将该图片的宽度设置为当前子图环境宽度的 0.5 倍。

在导言区使用 `\usepackage{subcaption}` 语句，在代码主体区域使用 `figure` 环境嵌套三个 `subfigure` 环境从而创建三个子图，并为各子图分别创建索引和标题。

```
\documentclass[12pt]{article}
\usepackage{graphicx}
\usepackage{subcaption}
\begin{document}
```

Figure `\ref{fig:fig1}` contains sub-figure `\ref{subfig:subfig1}`, sub-figure `\ref{subfig:subfig2}` and sub-figure `\ref{subfig:subfig3}`.

```
\begin{figure}[h!]
\centering
% 插入第一张子图
\begin{subfigure}{.3\linewidth}
\centering
\includegraphics[width=.5\linewidth]{redflower.png}
\caption{A red flower.}
\label{subfig:subfig1}
\end{subfigure}
% 插入第二张子图
\begin{subfigure}{.3\linewidth}
\centering
\includegraphics[width=.5\linewidth]{yellowFlower.png}
\caption{A yellow flower.}
\label{subfig:subfig2}
\end{subfigure}
% 插入第三张子图
\begin{subfigure}{.3\linewidth}
\centering
\includegraphics[width=.5\linewidth]{blueFlower.png}
\caption{A blue flower.}
\label{subfig:subfig3}
\end{subfigure}
\caption{Three flowers with different colors.}
\label{fig:fig1}
\end{figure}
```

```
\end{document}
```

5.2 调整子图间距

通过调整子图的横向和纵向间距，可以创建更协调美观的图片。

具体而言，存在以下三类命令可用于调整图片的横向间距：

- `\hfill`：对于位于相同行的子图，通过在相邻的 `subfigure` 环境间使用该命令，可以实现多个子图横向等距分布的效果；
- `\hspace` 横向距离：定制任意长度的横向图片距离。当该值设为负值时，可以产生图片重叠的效果；
- `\quad`、`\qquad` 等：设置不同预设长度的横向图片距离。

使用 `\hfill` 命令实现子图横向等距分布，以及使用 `\hspace{}` 命令实现子图重叠效果。

```
\documentclass[12pt]{article}
\usepackage{graphicx}
\usepackage{subcaption}
\usepackage{amssymb}
\usepackage{amsmath}
\begin{document}
```

% 使用 `\hfill` 命令调整子图横向间距

The horizontal space among Sub-figures in figure `\ref{fig:fig1}` is controlled by `\backslash\textit{hfill}`.

```
\begin{figure}[h!]
```

```
\centering
```

% 插入第一张子图

```
\begin{subfigure}{.3\linewidth}
```

```
\centering
```

```
\includegraphics[width=.5\linewidth]{redflower.png}
```

```
\caption{A red flower.}
```

```
\end{subfigure}
```

```
\hfill
```

% 插入第二张子图

```
\begin{subfigure}{.3\linewidth}
```

```
\centering
```

```
\includegraphics[width=.5\linewidth]{yellowFlower.png}
```

```

        \caption{A yellow flower.}
    \end{subfigure}
    \hfill
    % 插入第三张子图
    \begin{subfigure}{.3\linewidth}
        \centering
        \includegraphics[width=.5\linewidth]{blueFlower.png}
        \caption{A blue flower.}
    \end{subfigure}
    \caption{Three flowers with different colors.}
    \label{fig:fig1}
    \end{figure}

% 使用\space{}命令调整子图横向间距

The horizontal space among Sub-figures in figure \ref{fig:fig2} is
    controlled by $\backslash\textit{space}$.

\begin{figure}[h!]
    \centering
    % 插入第一张子图
    \begin{subfigure}{.3\linewidth}
        \centering
        \includegraphics[width=.5\linewidth]{redflower.png}
    \end{subfigure}
    \hspace{-5cm}
    % 插入第二张子图
    \begin{subfigure}{.3\linewidth}
        \centering
        \includegraphics[width=.5\linewidth]{yellowFlower.png}
    \end{subfigure}
    \hspace{-5cm}
    % 插入第三张子图
    \begin{subfigure}{.3\linewidth}
        \centering
        \includegraphics[width=.5\linewidth]{blueFlower.png}
    \end{subfigure}
    \caption{Three flowers with different colors.}
    \label{fig:fig2}
    \end{figure}

\end{document}

```

使用 `\vfill` 命令实现子图纵向等距分布。

```
\documentclass[12pt]{article}
\usepackage{graphicx}
\usepackage{subcaption}
\begin{document}

\begin{figure}[h!]
\centering
% 插入第一张子图
\begin{subfigure}{.3\linewidth}
\centering
\includegraphics[width=.5\linewidth]{redflower.png}
\caption{A red flower.}
\end{subfigure}
\vfill
% 插入第二张子图
\begin{subfigure}{.3\linewidth}
\centering
\includegraphics[width=.5\linewidth]{yellowFlower.png}
\caption{A yellow flower.}
\end{subfigure}
\vfill
% 插入第三张子图
\begin{subfigure}{.3\linewidth}
\centering
\includegraphics[width=.5\linewidth]{blueFlower.png}
\caption{A blue flower.}
\end{subfigure}
\caption{Three flowers with different colors.}
\label{fig:fig1}
\end{figure}

\end{document}
```

位置调整

```
\documentclass[12pt]{article}
\usepackage{graphicx}
\begin{document}

There is a blue and black butterfly dancing among the colorful flowers.

\begin{figure}[t!]
\centering
\includegraphics[width = 0.5\textwidth]{graphics/butterfly.JPG}
```

```
\caption{There is a beautiful butterfly.}
\label{butterfly}
\end{figure}

\end{document}
```

文字环绕

```
\documentclass[12pt]{article}
\usepackage{graphicx}
\usepackage{wrapfig}
\begin{document}
```

There is a blue and black butterfly dancing among the colorful flowers.

```
\begin{wrapfigure}{r}{8cm}
\centering
\includegraphics[width =0.35\textwidth]{graphics/butterfly.JPG}
\caption{There is a beautiful butterfly.}
\label{butterfly}
\end{wrapfigure}
```

In descriptive statistics, a box plot or boxplot is a method for graphically depicting groups of numerical data through their quartiles. Box plots may also have lines extending from the boxes (whiskers) indicating variability outside the upper and lower quartiles, hence the terms box-and-whisker plot and box-and-whisker diagram. Outliers may be plotted as individual points. Box plots are non-parametric: they display variation in samples of a statistical population without making any assumptions of the underlying statistical distribution (though Tukey's boxplot assumes symmetry for the whiskers and normality for their length).

```
\end{document}
```

Part VI

图形绘制

1 基本介绍

TikZ 宏包是在 LaTeX 中创建图形元素的最复杂和最强大的工具。在本节中，我们将通过一些简单的示例来介绍如何在 tikzpicture 环境中创建基本的图形元素，如：线、点、曲线、圆、矩形等。

1.1 使用 tikzpicture 环境创建图形元素

首先，我们需要通过 `\usepackage{tikz}` 命令调用 TikZ 宏包。在绘制图形之前，需要声明 tikzpicture 环境。在此我们先给出两个用 TikZ 绘图例子，其后再进一步详细介绍具体的绘图命令。



使用 tikzpicture 环境制作一个简单的图形。

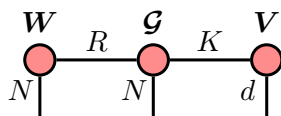
```
\documentclass[12pt]{article}
\usepackage{tikz}
\begin{document}

\begin{tikzpicture}

  \draw[red,fill=red] (0,0) .. controls (0,0.75) and (-1.5,1.00) ..
    (-1.5,2) arc (180:0:0.75) — cycle;
  \draw[red,fill=red] (0,0) .. controls (0,0.75) and ( 1.5,1.00) .. (
    1.5,2) arc (0:180:0.75) — cycle;

\end{tikzpicture}

\end{document}
```



使用tikz宏包中的tikzpicture环境创建一个张量网络图。

```
\begin{tikzpicture}

  \node[circle, line width = 0.4mm, draw = black, fill = red!45, inner
    sep = 0pt, minimum size = 0.4cm] (w) at (0, 0) {};
  \node at (0, 0.5) {\small{$\boldsymbol{W}$}};

  \node[circle, line width = 0.4mm, draw = black, fill = red!45, inner
    sep = 0pt, minimum size = 0.4cm] (g) at (1.5, 0) {};
  \node at (1.5, 0.5) {\small{$\boldsymbol{\mathcal{G}}$}};

  \node[circle, line width = 0.4mm, draw = black, fill = red!45, inner
    sep = 0pt, minimum size = 0.4cm] (v) at (3, 0) {};
  \node at (3, 0.5) {\small{$\boldsymbol{V}$}};

  \path [draw, line width = 0.4mm, -] (w) edge (g);
  \node at (0.75, 0.25) {\small{$R$}};
  \path [draw, line width = 0.4mm, -] (g) edge (v);
  \node at (2.25, 0.25) {\small{$K$}};

  \draw [line width = 0.4mm] (w) — (0, -0.8);
  \node at (-0.25, -0.4) {\small{$N$}};
  \draw [line width = 0.4mm] (g) — (1.5, -0.8);
  \node at (1.5-0.25, -0.4) {\small{$N$}};
  \draw [line width = 0.4mm] (v) — (3, -0.8);
  \node at (3-0.25, -0.4) {\small{$d$}};

\end{tikzpicture}
```

1.2 绘制直线

从绘制一条直线开始，入门这个强大的 LaTeX 绘图工具。首先，画一条直线需要给出起始点坐标和终止点坐标，我们可以简单地通过如下代码



Fig 7: 直线

绘制直线显示标题并居中显示

```
\begin{figure}[h]
  \begin{center}
    \begin{tikzpicture}
      \draw (1,1) — (1,3);
    \end{tikzpicture}
    \caption{直线}
  \end{center}
\end{figure}
```

我们可以通过设定一系列的坐标点，来实现多条线段的连续绘制。



Fig 8: 多条线段直线

```
\begin{figure}[h]
  \begin{center}
    \begin{tikzpicture}
      \draw (-2,0) — (2,0) — (2,2) — (-2,2) — (-2,0);
    \end{tikzpicture}
    \caption{多条线段直线}
  \end{center}
\end{figure}
```

多行命令，实现多段线条分开绘制

值得注意的是，在 tikzpicture 环境中，像 Matlab 语言一样，我们需要采用 ; 符号来标记一个指令的结束。这样的指令结束标记让我们不但可以在多行完成一条指令，同时也可以在一行内实现多条指令。

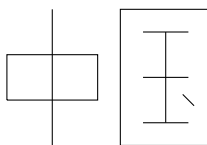


Fig 9: 多条线段分开绘制

多段线条分开绘制

```
\begin{figure}[h]
  \begin{center}
    \begin{tikzpicture}
      \draw (-2,0) — (2,0) — (2,2) — (-2,2) — (-2,0);
      \draw (0,4) — (0,-2);
      \draw (3,-2) — (3,4) — (7,4) — (7,-2) — (3,-2);
      \draw (4,3) — (6,3); \draw (4,1) — (6,1); \draw (4,-1) —
        (6,-1);
      \draw (5,3) — (5,-1); \draw (5.75,0.25) — (6.25,-0.25);
    \end{tikzpicture}
    \caption{多条线段分开绘制}
  \end{center}
\end{figure}
```

1.3 图形缩放

在上小节中，我们绘制图形需要给出精确的坐标点。但是在绘制好之后，如果需要调整图形大小，我们可以采用 `scale` 的方式对图形进行缩放。

横向缩放使用命令 `[xscale=]`，纵向缩放使用命令 `[yscale=]`，横纵缩放使用命令 `[xscale=,yscale=]`

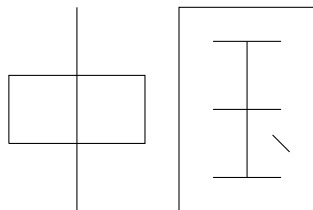


Fig 10: 多条线段分开绘制

整体缩放。

```
\documentclass[12pt]{article}
\usepackage{tikz}
\begin{document}
\begin{tikzpicture}[scale=0.5]
  \draw (-2,0) — (2,0) — (2,2) — (-2,2) — (-2,0);
  \draw (0,4) — (0,-2);
  \draw (3,-2) — (3,4) — (7,4) — (7,-2) — (3,-2);
  \draw (4,3) — (6,3); \draw (4,1) — (6,1); \draw (4,-1) — (6,-1);
  \draw (5,3) — (5,-1); \draw (5.75,0.25) — (6.25,-0.25);
\end{tikzpicture}
\end{document}
```

横向缩放。

```
\documentclass[12pt]{article}
\usepackage{tikz}
\begin{document}
\begin{tikzpicture}[xscale=1.5]
  \draw (-2,0) — (2,0) — (2,2) — (-2,2) — (-2,0);
  \draw (0,4) — (0,-2);
  \draw (3,-2) — (3,4) — (7,4) — (7,-2) — (3,-2);
  \draw (4,3) — (6,3); \draw (4,1) — (6,1); \draw (4,-1) — (6,-1);
  \draw (5,3) — (5,-1); \draw (5.75,0.25) — (6.25,-0.25);
\end{tikzpicture}
\end{document}
```

横纵缩放。

```
\documentclass[12pt]{article}
\usepackage{tikz}
\begin{document}
\begin{tikzpicture}[xscale=1.5, yscale = 2]
  \draw (-2,0) — (2,0) — (2,2) — (-2,2) — (-2,0);
  \draw (0,4) — (0,-2);
  \draw (3,-2) — (3,4) — (7,4) — (7,-2) — (3,-2);
  \draw (4,3) — (6,3); \draw (4,1) — (6,1); \draw (4,-1) — (6,-1);
  \draw (5,3) — (5,-1); \draw (5.75,0.25) — (6.25,-0.25);
\end{tikzpicture}
\end{document}
```

1.4 绘制箭头

在绘制直线的基础上，我们往往需要通过绘制箭头来指向性地表达意图。箭头的绘制只需要在直线绘制的基础上，增加 [option] 进行声明即可。

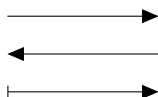


Fig 11: 箭头

```
\begin{figure}[h!]
  \begin{center}
    \begin{tikzpicture}
      \draw [->] (0,0) — (2,0);
      \draw [<-] (0,-0.5) — (2,-0.5);
    \end{tikzpicture}
  \end{center}
\end{figure}
```

```

\draw [|->] (0,-1) — (2,-1);
\end{tikzpicture}
\caption{箭头}
\end{center}
\end{figure}

```



Fig 12: 坐标轴

利用绘制箭头的例子以及多条线段连续绘制的例子，用一行命令绘制一个直角坐标系。

```

\begin{figure}[h!]
\begin{center}
\begin{tikzpicture}
\draw [<->] (0,2) — (0,0) — (3,0);
\end{tikzpicture}
\caption{坐标轴}
\end{center}
\end{figure}

```

1.5 调整线条粗细

采用\draw 命令时，增加的 [option] 声明也可以用来调整线条的粗细

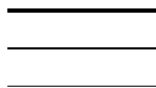


Fig 13: 线条粗细

绘制不同粗细的线条。

```

\documentclass[12pt]{article}
\usepackage{tikz}
\begin{document}
\begin{figure}[h!]
\begin{center}
\begin{tikzpicture}

```

```

\draw [ultra thick] (0,1) — (2,1);
\draw [thick] (0,0.5) — (2,0.5);
\draw [thin] (0,0) — (2,0);
\end{tikzpicture}
\caption{线条粗细}
\end{center}
\end{figure}
\end{document}

```

线条的粗细可以通过不同的指令来控制，从细到粗分别可调用：ultra thin, very thin, thin, semithick, thick, very thick, ultra thick。

除此之外，我们也可以自行定义线条的粗细，如 [line width=5]、[line width=0.2cm]。值得注意的是，当我们直接声明数值而不声明单位时，其默认单位均为 pt。

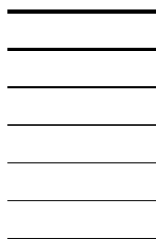


Fig 14: 不同线条粗细

```

\begin{figure}[h!]
\begin{center}
\begin{tikzpicture}
\draw [ultra thin] (0,0) — (2,0);
\draw [very thin] (0,0.5) — (2,0.5);
\draw [thin] (0,1) — (2,1);
\draw [semithick] (0,1.5) — (2,1.5);
\draw [thick] (0,2) — (2,2);
\draw [very thick] (0,2.5) — (2,2.5);
\draw [ultra thick] (0,3) — (2,3);
\end{tikzpicture}
\caption{不同线条粗细}
\end{center}
\end{figure}

```

1.6 虚线

我们也可以在 [option] 声明中增加对于线条形状的定义。如虚线 dashed 和点线 dotted。

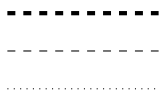


Fig 15: 虚线

绘制虚线

```
\begin{figure}[h!]
  \begin{center}
    \begin{tikzpicture}
      \draw [dashed, ultra thick] (0,1) — (2,1); %我们可以通过组合多种option来声明线条的多种特征。
      \draw [dashed] (0, 0.5) — (2,0.5);
      \draw [dotted] (0,0) — (2,0);
    \end{tikzpicture}
    \caption{虚线}
  \end{center}
\end{figure}
```

1.7 颜色

我们也可以在 [option] 声明中增加对于线条颜色的定义。如红色 red、绿色 green、蓝色 blue 等等。



Fig 16: 不同颜色虚线

绘制不同颜色的直线

```
\begin{figure}[h!]
  \begin{center}
    \begin{tikzpicture}
      \draw [red, dashed, ultra thick] (0,1) — (2,1); %我们可以通过组合多种option来声明线条的多种特征。
      \draw [green, dashed] (0, 0.5) — (2,0.5);
      \draw [blue, dotted] (0,0) — (2,0);
    \end{tikzpicture}
    \caption{不同颜色虚线}
  \end{center}
\end{figure}
```

2 节点介绍

节点是 TikZ 中的一个常用功能。在绘制节点时，通常需要声明其位置和形状，部分节点可以在其中添加文字，同时也可以为节点赋予一个名称，用于后续参考。在本节中，我们将详细介绍节点的相关功能及其应用。

2.1 节点基本介绍

这里需要注意的是，在 `[shape=circle,draw=blue!50,fill=blue!20]` 中，`shape` 指令声明节点形状，`draw` 指令声明是否显现该形状的边框，并用 `draw=` 来声明边框颜色，`fill` 命令指明该节点是否要填充，并用 `fill=` 来声明填充颜色。若要在节点中显示文字，需在后面 `{}` 中填写对应的文字即可。

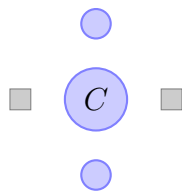


Fig 17: 绘制节点

```
\begin{figure}[h!]
  \begin{center}
    \begin{tikzpicture}
      \path (0,2) node [shape=circle ,draw=blue!50 ,fill=blue!20 ,thick]
        {}
        (0,1) node [shape=circle ,draw=blue!50 ,fill=blue!20 ,
          thick] {$C$}
        (0,0) node [shape=circle ,draw=blue!50 ,fill=blue!20 ,
          thick] {}
        (1,1) node [shape=rectangle ,draw=black!50 ,fill=black
          !20] {}
        (-1,1) node [shape=rectangle ,draw=black!50 ,fill=black
          !20] {}
    \end{tikzpicture}
    \caption{绘制节点}
  \end{center}
\end{figure}
```

等价于

```
\begin{figure}[h!]
  \begin{center}
    \begin{tikzpicture}
      \node [shape=circle ,draw=blue!50 ,fill=blue!20 ,thick] at (0,2)
        {}
    \end{tikzpicture}
  \end{center}
\end{figure}
```

```

\begin{figure}
\begin{center}
\begin{tikzpicture}
\node [shape=circle ,draw=blue!50 ,fill=blue!20 ,thick] at (0,1)
{$C$};
\node [shape=circle ,draw=blue!50 ,fill=blue!20 ,thick] at (0,0)
{};
\node [shape=rectangle ,draw=black!50 ,fill=black!20] at (1,1)
{};
\node [shape=rectangle ,draw=black!50 ,fill=black!20] at (-1,1)
{};
\end{tikzpicture}
\caption{绘制节点}
\end{center}
\end{figure}

```

2.2 节点样式

当某一种形状及颜色的节点需要在不同位置多次出现时，上述代码显得不够优美。我们可以通过一段代码提前声明该节点的样式，并反复调用这段代码即可。

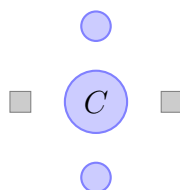


Fig 18: 节点样式

```

\begin{figure}[h!]
\begin{center}
\begin{tikzpicture}
\tikzstyle{aaa}=[circle ,draw=blue!50 ,fill=blue!20 ,thick]
\tikzstyle{bbb}=[rectangle ,draw=black!50 ,fill=black!20]
\path (0,2) node [aaa] {}
(0,1) node [aaa] {$C$}
(0,0) node [aaa] {}
(1,1) node [bbb] {}
(-1,1) node [bbb] {};
\end{tikzpicture}
\caption{节点样式}
\end{center}
\end{figure}

```


2.3 节点命名

为了将节点相互连接起来，我们需要指明连接哪两个节点。因此，每个节点需要我们先声明一个名称。声明名称有两种方式，一种是采用 `name=` 的方式，另一种是在 `\node` 后用括号声明，如 `\node (name) ...`。

绘制节点，并声明节点名称，方法一。

```
\documentclass[12pt]{article}
\usepackage{tikz}
\begin{document}
\tikzstyle{aaa}=[circle,draw=blue!50,fill=blue!20,thick]
\tikzstyle{bbb}=[rectangle,draw=black!50,fill=black!20]
\begin{tikzpicture}
  \path (0,2) node [aaa,name=a1] {}
        (0,1) node [aaa,name=a2] {$C$}
        (0,0) node [aaa,name=a3] {}
        (1,1) node [bbb,name=b1] {}
        (-1,1) node [bbb,name=b2] {}
\end{tikzpicture}
\end{document}
```

绘制节点，并声明节点名称，方法二。

```
\documentclass[12pt]{article}
\usepackage{tikz}
\begin{document}
\tikzstyle{aaa}=[circle,draw=blue!50,fill=blue!20,thick]
\tikzstyle{bbb}=[rectangle,draw=black!50,fill=black!20]
\begin{tikzpicture}
  \node (a1) [aaa] at (0,2) {};
  \node (a2) [aaa] at (0,1) {$C$};
  \node (a3) [aaa] at (0,0) {};
  \node (b1) [bbb] at (1,1) {};
  \node (b2) [bbb] at (-1,1) {};
\end{tikzpicture}
\end{document}
```

2.4 基于相对位置绘制节点

给每个节点命名后，我们便可以通过 `above of` (上)、`below of` (下)、`left of` (左)、`right of` (右) 等命令来声明新节点与某个节点的相对位置来绘制图形。

利用相对位置绘制节点。

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

```

\usepackage{tikz}
\begin{document}
\tikzstyle{aaa}=[circle ,draw=blue!50 ,fill=blue!20 ,thick]
\tikzstyle{bbb}=[rectangle ,draw=black!50 ,fill=black!20]
\begin{tikzpicture}
  \node (a1) [aaa]          {};
  \node (a2) [aaa] [below of=a1] {$C$};
  \node (a3) [aaa] [below of=a2] {};
  \node (b1) [bbb] [right of=a2] {};
  \node (b2) [bbb] [left of=a2]  {};
\end{tikzpicture}
\end{document}

```

2.5 连接节点

有了节点名称了，我们就可以对节点进行连接。我们拿连接 A 与 B 节点为例，在连接时，我们通常需要声明 A 节点的哪个位置与 B 节点的哪个位置连接。位置声明通常采用 east (右)、west (左)、north (上)、south (下)、center (中心) 等命令。

利用相对位置连接节点。

```

\documentclass[12pt]{article}
\usepackage{tikz}
\begin{document}
\tikzstyle{aaa}=[circle ,draw=blue!50 ,fill=blue!20 ,thick]
\tikzstyle{bbb}=[rectangle ,draw=black!50 ,fill=black!20]
\begin{tikzpicture}
  \node (a1) [aaa]          {$a_1$};
  \node (a2) [aaa] [below of=a1] {$C$};
  \node (a3) [aaa] [below of=a2] {$a_3$};
  \node (b1) [bbb] [right of=a2] {$b_1$};
  \node (b2) [bbb] [left of=a2]  {$b_2$};
  \draw [->] (a2.west) — (b2.east);
  \draw [->] (a2.east) — (b1.west);
  \draw [->] (a2.north) — (a1.south);
  \draw [->] (a2.south) — (a3.north);
\end{tikzpicture}
\end{document}

```

该代码等价于

```

\documentclass[12pt]{article}
\usepackage{tikz}
\begin{document}
\tikzstyle{aaa}=[circle ,draw=blue!50 ,fill=blue!20 ,thick]
\tikzstyle{bbb}=[rectangle ,draw=black!50 ,fill=black!20]

```

```
\begin{tikzpicture}
  \node (a1) [aaa]                {$a_1$};
  \node (a2) [aaa] [below of=a1] {$C$};
  \node (a3) [aaa] [below of=a2] {$a_3$};
  \node (b1) [bbb] [right of=a2]  {$b_1$};
  \node (b2) [bbb] [left of=a2]   {$b_2$};
  \draw [->] (a2) — (b2);
  \draw [->] (a2) — (b1);
  \draw [->] (a2) — (a1);
  \draw [->] (a2) — (a3);
\end{tikzpicture}
\end{document}
```

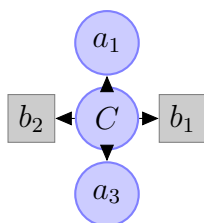


Fig 19: 连接节点

利用edge命令连接节点，方法一。

```
\documentclass[12pt]{article}
\usepackage{tikz}
\begin{document}
\tikzstyle{aaa}=[circle ,draw=blue!50 ,fill=blue!20 ,thick]
\tikzstyle{bbb}=[rectangle ,draw=black!50 ,fill=black!20]
\begin{tikzpicture}
  \node (a1) [aaa]                {$a_1$};
  \node (a2) [aaa] [below of=a1] {$C$}   edge [->] (a1);
  \node (a3) [aaa] [below of=a2] {$a_3$} edge [<-] (a2);
  \node (b1) [bbb] [right of=a2]  {$b_1$} edge [<-] (a2);
  \node (b2) [bbb] [left of=a2]   {$b_2$} edge [<-] (a2);
\end{tikzpicture}
\end{document}
```

等价于

```
\documentclass[12pt]{article}
\usepackage{tikz}
\begin{document}
\tikzstyle{aaa}=[circle ,draw=blue!50 ,fill=blue!20 ,thick]
\tikzstyle{bbb}=[rectangle ,draw=black!50 ,fill=black!20]
\begin{tikzpicture}
```

```

\path (0,2) node [aaa,name=a1] {$a_1$}
      (0,1) node [aaa,name=a2] {$C$} edge [->] (a1)
      (0,0) node [aaa,name=a3] {$a_3$} edge [<-] (a2)
      (1,1) node [bbb,name=b1] {$b_1$} edge [<-] (a2)
      (-1,1) node [bbb,name=b2] {$b_2$} edge [<-] (a2);
\end{tikzpicture}
\end{document}

```

声明每个节点的连接位置，将周围节点边缘连接至中间节点的中心。

```

\documentclass[12pt]{article}
\usepackage{tikz}
\begin{document}
\tikzstyle{aaa}=[circle,draw=blue!50,fill=blue!20,thick]
\tikzstyle{bbb}=[rectangle,draw=black!50,fill=black!20]
\begin{tikzpicture}
  \node (a1) [aaa] {$a_1$};
  \node (a2) [aaa] [below of=a1] {};
  \node (a3) [aaa] [below of=a2] {$a_3$};
  \node (b1) [bbb] [right of=a2] {$b_1$};
  \node (b2) [bbb] [left of=a2] {$b_2$};
  \draw [->] (a2.center) — (b2.east);
  \draw [->] (a2.center) — (b1.west);
  \draw [->] (a2.center) — (a1.south);
  \draw [->] (a2.center) — (a3.north);
\end{tikzpicture}
\end{document}

```

3 高级功能

3.1 矩形、圆形、曲线

我们可以通过 `\draw (x,y) rectangle (w,h);` 的方式绘制一个矩形，其左下角坐标位于点 (x,y) 处，长度为 w ，高度为 h 。类似地，我们也可以通过 `\draw (x,y) circle [radius=r];` 的方式绘制一个圆形，其圆心落在点 (x,y) 处，半径为 r 。除此之外，我们可以通过 `\draw (x,y) arc [radius=r, start angle=a1, end angle=a2]` 的方式绘制一条弧线，它从点 (x,y) 处开始绘制，该弧线曲率半径为 r ，其起始角度为所对应曲率圆的 a_1 处，终止角度为所对应曲率圆的 a_2 处。

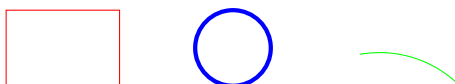


Fig 20: 矩形圆形曲线

```
\begin{figure} [!h]
  \begin{center}
    \begin{tikzpicture}
      \draw [red] (0,0) rectangle (1.5,1);
      \draw [blue, ultra thick] (3,0.5) circle [radius=0.5];
      \draw [green] (6,0) arc [radius=1.5, start angle=45, end angle
        =100];
    \end{tikzpicture}
    \caption{矩形圆形曲线}
  \end{center}
\end{figure}
```

3.2 平滑过渡曲线

在绘图时，一种不突兀地连接两条直线的方式是采用平滑过渡圆角/曲线。在本节中，我们将介绍两种平滑过渡曲线的绘制方法：绘制带圆角的曲线和绘制过渡曲线。



Fig 21: 圆角坐标系

```
\begin{figure} [!h]
  \begin{center}
    \begin{tikzpicture}
      \draw [<->, rounded corners, thick, purple] (0,2) — (0,0) —
        (3,0);
    \end{tikzpicture}
    \caption{圆角坐标系}
  \end{center}
\end{figure}
```

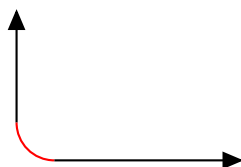


Fig 22: 过渡曲线

```
\begin{figure}
  \begin{center}
    \begin{tikzpicture}
      \draw[<-, thick] (0,2) — (0,0.5);
      \draw[thick, red] (0,0.5) to [out=270,in=180] (0.5,0);
      \draw[->, thick] (0.5,0) — (3,0);
    \end{tikzpicture}
    \caption{过渡曲线}
  \end{center}
\end{figure}
```

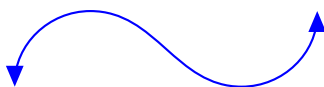


Fig 23: S 曲线

```
\begin{figure}
  \begin{center}
    \begin{tikzpicture}
      \draw [<->,thick, blue] (0,0) to [out=90,in=180] (1,1) to [out=
        =0,in=180] (3,0) to [out=0,in=-90] (4,1) ;
    \end{tikzpicture}
    \caption{S曲线}
  \end{center}
\end{figure}
```

3.3 根据函数绘制曲线

TikZ 宏包的强大之处在于，它还提供了可供绘制函数的数学引擎。在此我们先给出一个示例，再详细讲解如何利用该宏包绘制函数所对应的曲线。

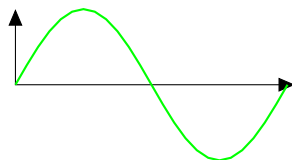


Fig 24: S 曲线

利用函数绘制正弦曲线。

```
\documentclass[12pt]{article}
\usepackage{tikz}
\begin{document}
\begin{tikzpicture}[xscale=0.01,yscale=1]
  \draw[<->] (0,1) — (0,0) — (370,0);
  \draw[green, thick, domain=0:360] plot (\x, {\sin(\x)}); % 这里需要注意
  带上{}
\end{tikzpicture}
\end{document}
```

在上述例子中，domain 指令声明了横坐标 x 的范围。在本示例中，我们利用 \sin 函数绘制了一段正弦曲线。

除了本示例中的正弦曲线 \sin 函数，我们还可以调用大量其他函数，在此列举一部分作为示例：阶乘函数： $\text{factorial}(\backslash x)$ 、平方根函数： $\text{sqrt}(\backslash x)$ 、幂函数： $\text{pow}(\backslash x, y)$ （该函数为 xy ）、指数函数： $\text{exp}(\backslash x)$ 、对数函数： $\text{ln}(\backslash x)$ 、 $\text{log10}(\backslash x)$ 、 $\text{log2}(\backslash x)$ 、绝对值函数： $\text{abs}(\backslash x)$ 、取余函数： $\text{mod}(\backslash x, y)$ （即求 x 被 y 除后的余数）、圆整函数： $\text{round}(\backslash x)$ 、 $\text{floor}(\backslash x)$ 、 $\text{ceil}(\backslash x)$ 、三角函数： $\text{sin}(\backslash x)$ 、 $\text{cos}(\backslash x)$ 、 $\text{tan}(\backslash x)$ 等等。 $\text{sin}(\backslash x)$ 、 $\text{cos}(\backslash x)$ 、 $\text{tan}(\backslash x)$ 值得注意的是，在三角函数中，通常默认自变量 x 以度 ($^\circ$) 为单位。若要采用弧度制，则需要将函数分别改写为 $\text{sin}(\backslash x \text{ r})$ 、 $\text{cos}(\backslash x \text{ r})$ 、 $\text{tan}(\backslash x \text{ r})$ 。除了这部分常用函数之外，我们通常还会使用两个常数： e ($e=2.718281828$) 和 π ($\pi=3.141592654$)。

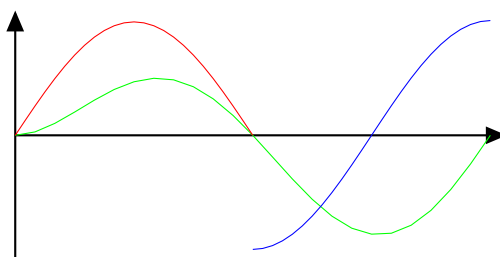


Fig 25: S 曲线

```
\begin{figure} [!h]
  \begin{center}
    \begin{tikzpicture} [yscale=1.5]
      \draw [thick, ->] (0,0) — (6.5,0);
      \draw [thick, ->] (0,-1.1) — (0,1.1);
      \draw [green, domain=0:2*pi] plot (\x, {(sin(\x r)* ln(\x+1))
        /2});
      \draw [red, domain=0:pi] plot (\x, {sin(\x r)});
      \draw [blue, domain=pi:2*pi] plot (\x, {cos(\x r)*exp(\x/exp(2*
        pi))});
    \end{tikzpicture}
    \caption{S 曲线}
  \end{center}
\end{figure}
```

3.4 简单图形的区域填充

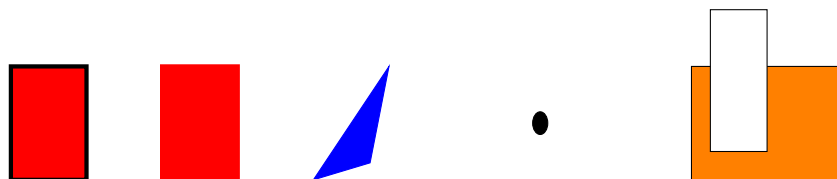


Fig 26: 简单形状区域填充

简单形状的区域填充。

```
\documentclass[12pt]{article}
\usepackage{tikz}
\begin{document}
\begin{figure} [!h]
  \begin{center}
    \begin{tikzpicture} [yscale=1.5]
      \draw [fill=red, ultra thick] (0,0) rectangle (1,1);
      \draw [fill=red, ultra thick, red] (2,0) rectangle (3,1); % 这里
        的第二个red声明了区域周围边框线的颜色
      \draw [blue, fill=blue] (4,0) — (5,1) — (4.75,0.15) — (4,0);
      \draw [fill] (7,0.5) circle [radius=0.1];
      \draw [fill=orange] (9,0) rectangle (11,1);
      \draw [fill=white] (9.25,0.25) rectangle (10,1.5);
    \end{tikzpicture}
    \caption{简单形状区域填充}
  \end{center}
\end{figure}
```



```

\end{center}
\end{figure}
\end{document}

```

如上例中的注释所提，我们可以通过声明图形边框线的颜色来对边框线进行个性化更改。若并不希望出现边框线，我们可以采用 `path` 命令替换 `\draw` 命令。



Fig 27: path 区域填充

简单形状的区域填充。

```

\documentclass[12pt]{article}
\usepackage{tikz}
\begin{document}
\begin{figure}[!h]
  \begin{center}
    \begin{tikzpicture}[yscale=1.5]
      \path [fill=red,thick] (0,0) rectangle (1.5,1);
      \draw [fill=red,thick] (2,0) rectangle (3.5,1);
    \end{tikzpicture}
    \caption{path 区域填充}
  \end{center}
\end{figure}
\end{document}

```

3.5 在图形中填写标签

在绘图时，在合适的位置加入适当的文字进行说明，对内容的表达具有很重要的作用。在本节中，我们将通过 `\node` 来实现这一功能。



Fig 28: 直角坐标系插入文字

```
\begin{figure} [!h]
  \begin{center}
    \begin{tikzpicture} [scale=1.5]
      \draw [thick,<->] (0,1)--(0,0)--(1,0);
      \node at (0.5,0.5) {\LaTeX};
    \end{tikzpicture}
    \caption{直角坐标系插入文字}
  \end{center}
\end{figure}
```

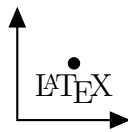


Fig 29: 直角坐标系指定位置下方插入文字

```
\begin{figure} [!h]
  \begin{center}
    \begin{tikzpicture} [scale=1.5]
      \draw [thick,<->] (0,1)--(0,0)--(1,0);
      \draw [fill] (0.5,0.5) circle [radius=0.05];
      \node [below] at (0.5,0.5) {\LaTeX};
    \end{tikzpicture}
    \caption{直角坐标系指定位置下方插入文字}
  \end{center}
\end{figure}
```



Fig 30: 直角坐标系指定位置插入文字

```
\begin{figure} [!h]
  \begin{center}
    \begin{tikzpicture} [scale=5]
      \draw [thick,<->] (0,1) -- (0,0) -- (1,0);
      \draw [fill] (0.5,0.5) circle [radius=0.025];
      \node [below] at (0.5,0.5) {\LaTeX};
      \node [above] at (0.5,0.5) {\LaTeX};
      \node [left] at (0.5,0.5) {\LaTeX};
      \node [right] at (0.5,0.5) {\LaTeX};
    \end{tikzpicture}
    \caption{直角坐标系指定位置插入文字}
  \end{center}
\end{figure}
```

```
\draw [thick, <->] (0,2) — (0,0) — (2,0);
\draw [fill] (1,1) circle [radius=0.025];
\node [below right, red] at (1,1) {below right};
\node [above left, green] at (1,1) {above left};
\node [below left, purple] at (1,1) {below left};
\node [above right, magenta] at (1,1) {above right};
\draw [thick, <->] (0,2) — (0,0) — (2,0);
\node [below right] at (2,0) {$x$};
\node [left] at (0,2) {$y$};
\draw[fill] (1,1) circle [radius=.5pt];
\node[above right] at (1,1) {$\theta$};
```

3.6 复杂模型实战解析

科研论文中的复杂模型供读者们进一步解析学习。

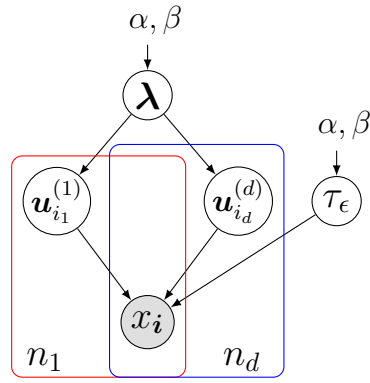


Fig 31: BCPF

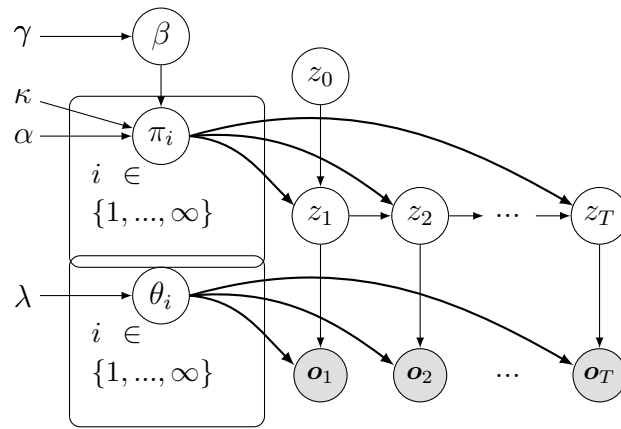


Fig 32: Sticky HDP-HMM

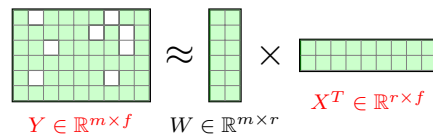


Fig 33: 矩阵分解

BCPF

```

\tikzset{>=latex}
\tikzstyle{plate caption} = [caption, node distance=0, inner sep=0pt,
below left=5pt and 0pt of #1.south]
\begin{figure} [!h]
    \begin{center}
        \begin{tikzpicture}
            \node [obs] (x) at (0,0) {\large  $x_{\boldsymbol{i}}$ };
            \node [circle,draw=black,fill=white,inner sep=0pt,minimum size
                =0.6cm] (u1) at (-1.2,1.6) {  $\boldsymbol{u}_{i_1}^{(1)}$ };
            \node [circle,draw=black,fill=white,inner sep=0pt,minimum size
                =0.6cm] (u3) at (1.2,1.6) {  $\boldsymbol{u}_{i_d}^{(d)}$ };
            \node [circle,draw=black,fill=white,inner sep=0pt,minimum size
                =0.65cm] (lambda) at (0,3.0) {\large  $\boldsymbol{\lambda}$ 
                };
            \node[mark size=1pt,color=black] at (0,1.6) {\pgfuseplotmark
                {*}};
            \node[mark size=1pt,color=black] at (-0.2,1.6) {\pgfuseplotmark
                {*}};
            \node[mark size=1pt,color=black] at (0.2,1.6) {\pgfuseplotmark
                {*}};
            \node [text width=0.5cm] (c0) at (0,4) { $\boldsymbol{\alpha}, \boldsymbol{\beta}$ };
            \node [text width=0.5cm] (a0) at (2.5,2.6) { $\boldsymbol{\alpha}, \boldsymbol{\beta}$ };
            \node [circle,draw=black,fill=white,inner sep=0pt,minimum size
                =0.65cm] (tau_epsilon) at (2.5,1.6) {\large  $\boldsymbol{\tau}_{\boldsymbol{\epsilon}}$ 
                };
            \path [draw,->] (u1) edge (x);
            \path [draw,->] (u3) edge (x);
            \path [draw,->] (lambda) edge (u1);
            \path [draw,->] (lambda) edge (u3);
            \path [draw,->] (c0) edge (lambda);
            \path [draw,->] (tau_epsilon) edge (x);
            \path [draw,->] (a0) edge (tau_epsilon);
            \plate [color=red] {part1} {(x)(u1)} { };
            \plate [color=blue] {part3} {(x)(u3)(part1.north east)} { };
            \node [text width=2cm] at (-0.6,-0.5) {\large  $n_1$ };
            \node [text width=2cm] at (2,-0.5) {\large  $n_d$ };
        \end{tikzpicture}
        \caption{BCPF}
    \end{center}
\end{figure}

```

Sticky HDP-HMM

```
\begin{figure}[!h]
  \begin{center}
    \begin{tikzpicture}[x=0.75pt,y=0.75pt,yscale=-1,xscale=1]
      \node[obs](o1)at(20,20){$\boldsymbol{o}_1$};
      \node[circle,draw=black,fill=white,inner sep=0pt,minimum size=0.75cm](p1)at(20,-60){$z_1$};
      \node[circle,draw=black,fill=white,inner sep=0pt,minimum size=0.75cm](p0)at(20,-130){$z_0$};
      \node[obs](o2)at(70,20){$\boldsymbol{o}_2$};
      \node[circle,draw=black,fill=white,inner sep=0pt,minimum size=0.75cm](p2)at(70,-60){$z_2$};
      \node[obs](o3)at(160,20){$\boldsymbol{o}_T$};
      \node[circle,draw=black,fill=white,inner sep=0pt,minimum size=0.75cm](p3)at(160,-60){$z_T$};
      \node[circle,draw=black,fill=white,inner sep=0pt,minimum size=0.75cm](theta)at(-60,-20){$\boldsymbol{\theta}_i$};
      \node[circle,draw=black,fill=white,inner sep=0pt,minimum size=0.75cm](pi)at(-60,-100){$\boldsymbol{\pi}_i$};
      \node[circle,draw=black,fill=white,inner sep=0pt,minimum size=0.75cm](beta)at(-60,-150){$\boldsymbol{\beta}$};
      \node[text width=2cm](inf1)at(-57,10){$\small\{i\in\left\{1,\dots,\infty\right\}\}$};
      \node[text width=2cm](inf2)at(-57,-70){$\small\{i\in\left\{1,\dots,\infty\right\}\}$};
      \node[text width=.2cm](lambda)at(-130,-20){$\boldsymbol{\lambda}$};
      \node[text width=.2cm](alpha)at(-130,-100){$\boldsymbol{\alpha}$};
      \node[text width=.2cm](kappa)at(-130,-120){$\boldsymbol{\kappa}$};
      \node[text width=.2cm](gamma)at(-130,-150){$\boldsymbol{\gamma}$};
      \node[text width=.2cm](gamma)at(-130,-150){$\boldsymbol{\gamma}$};
      \node[text width=.4cm](1dot)at(115,-60){$...$};
      \node[text width=.4cm](2dot)at(115,20){$...$};
      \path[draw,->](p2)edge(1dot);
      \path[draw,->](1dot)edge(p3);
      \path[draw,->](p1)edge(o1);
      \path[draw,->](p2)edge(o2);
      \path[draw,->](p3)edge(o3);
      \path[draw,->](p1)edge(p2);
      \path[draw,->](p0)edge(p1);
      \path[draw,->](alpha)edge(pi);
      \path[draw,->](kappa)edge(pi);
      \path[draw,->](beta)edge(pi);
      \path[draw,->](gamma)edge(beta);
      \path[draw,->](lambda)edge(theta);
    \end{tikzpicture}
  \end{center}
\end{figure}
```

```

\plate [color=black] {part2} {(theta)(inf1)} { };
\plate [color=black] {part3} {(pi)(inf2)} { };
\draw[black,->,thick] (pi.east) to [in=-150,out=-16] (147,-65);
\draw[black,->,thick] (pi.east) to [in=-143,out=-5] (57,-65);
\draw[black,->,thick] (pi.east) to [in=-135,out=5] (7,-65);
\draw[black,->,thick] (theta.east) to [in=-150,out=-16]
(147,15);
\draw[black,->,thick] (theta.east) to [in=-143,out=-5] (57,15);
\draw[black,->,thick] (theta.east) to [in=-135,out=5] (7,15);
\end{tikzpicture}
\caption{Sticky HDP-HMM}
\end{center}
\end{figure}

```

矩阵分解

```
\begin{figure}[!h]
  \begin{center}
    \begin{tikzpicture}
      \draw [very thick] (0,0) rectangle (3.6/2,2.4/2);
      \filldraw [fill=green!20!white,draw=green!40!black] (0,0)
        rectangle (3.6/2,2.4/2);
      \filldraw [fill=white] (0.4/2,0.4/2) rectangle (0.8/2,0.8/2);
      \filldraw [fill=white] (2.4/2,0.4/2) rectangle (2.8/2,0.8/2);
      \filldraw [fill=white] (0.8/2,1.2/2) rectangle (1.2/2,1.6/2);
      \filldraw [fill=white] (2.0/2,1.6/2) rectangle (2.4/2,2.0/2);
      \filldraw [fill=white] (0.4/2,2.0/2) rectangle (0.8/2,2.4/2);
      \filldraw [fill=white] (2.4/2,2.0/2) rectangle (2.8/2,2.4/2);
      \filldraw [fill=white] (2.8/2,1.2/2) rectangle (3.2/2,2.0/2);
      \draw [step=0.4/2, very thin, color=gray] (0,0) grid
        (3.6/2,2.4/2);
      \draw (1.8/2,-0.3) node {{\color{red}\scriptsize{$Y\mathbf{in}\mathbb{R}^m\times f$}}};
      \draw (4.4/2,1.2/2) node {{\color{black}\large{$\approx$}}};
      \draw [very thick] (5.2/2,0) rectangle (6.0/2,2.4/2);
      \filldraw [fill=green!20!white,draw=green!40!black] (5.2/2,0)
        rectangle (6.0/2,2.4/2);
      \draw [step=0.4/2, very thin, color=gray] (5.2/2,0) grid
        (6.0/2,2.4/2);
      \draw (5.6/2,-0.3) node {{\color{black}\scriptsize{$W\mathbf{in}\mathbb{R}^m\times r$}}};
      \draw (6.8/2,1.2/2) node {{\color{black}\large{$\times$}}};
      \draw [very thick] (7.6/2,0.8/2) rectangle (11.2/2,1.6/2);
      \filldraw [fill=green!20!white,draw=green!40!black]
        (7.6/2,0.8/2) rectangle (11.2/2,1.6/2);
      \draw [step=0.4/2, very thin, color=gray] (7.6/2,0.8/2) grid
        (11.2/2,1.6/2);
      \draw (9.4/2,0) node {{\color{red}\scriptsize{$X^T\mathbf{in}\mathbb{R}^r\times f$}}};
    \end{tikzpicture}
    \caption{矩阵分解}
  \end{center}
\end{figure}
```


Part VII

建立索引及文献引用

科技论文或技术报告中的图片、表格、公式和参考文献往往会被编号，方便读者进行查看。在实际写作过程中，有时图片、表格、公式会不在文本引用位置附近，而参考文献更是一般都总结放在文末结尾处。因此，读者在阅读过程中，为了查看该处引用内容的详细信息，需要翻看全文查找被引用的内容。这个过程一定是非常繁琐低效、并且会影响读者的阅读流畅性。为此，建立索引和文献引用就非常有必要了。建立索引一般指对文档中的图片、表格、公式等进行索引的设置，这个建立是自动编号完成；而文献引用同样是对文中存在引用参考文献的地方自动编号建立索引。因此，建立索引和文献引用的过程较为简单，并且建立索引和引用后的效果非常好，可以实现读者根据索引和文献引用直接跳转至想要查看的内容，能够有效提高读者的阅读效率和流畅性。

1 图表和公式的索引

1.1 公式的索引

LaTeX 中，公式的索引分为主要分为两个部分，一部分是给公式添加标签，可以使用 `\label{标签名}` 命令。可以使用 `equation` 环境插入带标签的公式；另一部分是在文档中进行引用，可以使用 `\ref{标签名}` 命令。

(1.1) is a binary equation

(1.1) is a binary quadratic equation.

$$x + y = 2 \tag{1}$$

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

使用 `\label{标签名}` 及 `\ref{标签名}` 在文中对公式进行索引。

```
\documentclass[12pt]{article}
\begin{document}
(\ref{eq1}) is a binary equation
(\ref{eq2}) is a binary quadratic equation.
\begin{equation}
x+y=2\label{eq1}
\end{equation}

\begin{equation}
x^2+y^2=2\label{eq2}
```

```
\end{equation}
\end{document}
```

1.2 图形的索引

插入图片需要使用 `graphicx` 宏包，图形的索引与公式的索引类似。同样分为两部分，一部分是使用 `\label{标签名}` 命令给添加图形标签，另一部分是使用 `\ref{标签名}` 在文档中进行引用。

Figure 34 is a photo of Latex.



Fig 34: LaTeX

1.3 表格的索引

表格的索引与公式及图形的索引类似，同样分为两部分，一部分是使用 `\label{标签名}` 命令给添加表格标签。根据第 5 章，可以使用 `tabular` 和 `table` 两种环境制作带标签的表格；另一部分是使用 `\ref{标签名}` 在文档中进行引用。

Table 12 shows the values of some basic functions.

表 12: The values of some basic functions.

| | $x = 1$ | $x = 2$ | $x = 3$ | $x = 4$ |
|-----------|---------|---------|---------|---------|
| $y = x$ | 1 | 2 | 3 | 4 |
| $y = x^2$ | 1 | 4 | 9 | 16 |
| $y = x^3$ | 1 | 8 | 27 | 64 |

Table~\ref{table1} shows the values of some basic functions.

```
\begin{table}
  \centering
  \caption{The values of some basic functions.}
  \begin{tabular}{l|cccc}
    \hline
    & $x=1$ & $x=2$ & $x=3$ & $x=4$ \\
    \hline
    $y=x$ & 1 & 2 & 3 & 4 \\
    $y=x^2$ & 1 & 4 & 9 & 16 \\
    $y=x^3$ & 1 & 8 & 27 & 64 \\
    \hline
  \end{tabular}
  \label{table1}% 索引标签
\end{table}
```

2 创建超链接

超链接指按内容链接，可以从一个文本内容指向文本其他内容或其他文件、网址等。超链接可以分为文本内链接、网页链接以及本地文件链接。L^AT_EX 提供了 hyperref 宏包，可用于生成超链接。在使用时，只需在前导代码中申明宏包即可，即 `\usepackage{hyperref}`。

2.1 超链接类型

2.1.1 文本内链接

文本内链接在篇幅较大的文档中，查阅内容会比较繁琐，因此，往往会在目录中使用超链接来进行文本内容的快速高效浏览。可以使用 hyperref 宏包创建文本内超链接。在导入 hyperref 时必须非常小心，一般而言，它必须是最后一个要导入的包。

使用 `\usepackage{hyperref}` 创建一个简单的目录链接文本内容的例子。

```
\documentclass{book}
\usepackage{blindtext}
\usepackage{hyperref} %超链接包

\begin{document}

\frontmatter
\tableofcontents
```

```

\clearpage

\addcontentsline{toc}{chapter}{Foreword}
{\huge {\bf Foreword}}

This is foreword.
\clearpage

\mainmatter

\chapter{First Chapter}

This is chapter 1.


\clearpage

\section{First section} \label{second}

This is section 1.1.
\end{document}

```

2.1.2 网址链接

众所周知，在文档中插入网址之类的文本同样需要用到超链接，同样的，使用 `hyperref` 宏包可以创建网页超链接。有时我们需要将超链接命名并隐藏网址，这时我们可以使用 `href` 命令进行插入；有时，我们插入的网址链接太长，但 LaTeX 不会自动换行，往往会造成格式混乱的问题，这时，我们可以使用 `url` 工具包，并在该工具包中声明一个参数即可解决这个问题，相关命令为 `\usepackage[hyphens]{url}`。

在 LaTeX 中使用 `hyperref` 及 `url` 工具包插入网页链接并设置自动换行。

```

\documentclass[12pt]{article}
\usepackage[hyphens]{url}
\usepackage{hyperref}

\begin{document}

This is the website of open-source latex-cookbook repository: \href{https
  ://github.com/xinychen/latex-cookbook}{LaTeX-cookbook} or go to the next
  url: \url{https://github.com/xinychen/latex-cookbook}.

\end{document}

```

2.1.3 本地文件链接

在LaTeX中使用href命令打开本地文件。

```
\documentclass[12pt]{article}
\usepackage[hyphens]{url}
\usepackage{hyperref}

\begin{document}

This is the text of open-source latex-cookbook repository: \href{run:./
  LaTeX-cookbook.dox}{LaTeX-cookbook}.

\end{document}
```

2.2 超链接格式

当然,有时候为了突出超链接,也可以在工具包hyperref中设置特定的颜色,设置的命令为\hypersetup,一般放在前导代码中,例如colorlinks=true, linkcolor=blue, urlcolor=blue, filecolor=magenta。默认设置以单色样式的空间字体打印链接, \urlstyle{same}命令将改变这个样式,并以与文本其余部分相同的样式显示链接。

在LaTeX中使用hyperref工具包插入超链接并设置超链接颜色为蓝色。

```
\documentclass{book}
\usepackage{blindtext}
\usepackage{hyperref} %超链接包
\hypersetup{colorlinks=true, %链接将被着色, 默认颜色是红色
  linkcolor=blue, % 内部链接显示为蓝色
  urlcolor=cyan, % 网址链接为青色
  filecolor=magenta} % 本地文件链接为洋红色
\urlstyle{same}

\begin{document}

\frontmatter
\tableofcontents
\clearpage

\addcontentsline{toc}{chapter}{Foreword}
{\huge {\bf Foreword}}

This is foreword.
\clearpage
```

```

\mainmatter

\chapter{First Chapter}

This is chapter 1.
\clearpage

\section{First section} \label{second}

This is section 1.1.

This is the website of open-source latex-cookbook repository: \href{https
://github.com/xinychen/latex-cookbook}{LaTeX-cookbook} or go to the next
url: \url{https://github.com/xinychen/latex-cookbook}.

This is the text of open-source latex-cookbook repository: \href{run:./
LaTeX-cookbook.dox}{LaTeX-cookbook}

\end{document}

```

3 Bibtex 用法

LaTeX 主要有两种管理参考文献的方法，第一种方法是在.tex 文档中嵌入参考文献，参考文献格式需符合特定的文献引用格式；另一种方法则是使用 BibTeX 进行文献管理，文件的拓展名为.bib。其中，使用外部文件 BibTeX 管理文献更加便捷高效。

3.1 创建参考文献

在 LaTeX 中，插入参考文献的一种直接方式是使用 thebibliography 环境，以列表的形式将参考文献进行整理起来，配以标签，以供正文引用，文档中引用的命令为\cite{}

Some examples for showing how to use thebibliography environment:

- Book reference sample: The L^AT_EX companion book [1].
- Paper reference sample: On the electrodynamics of moving bodies [2].
- Open-source reference sample: Knuth: Computers and Typesetting [3].

参考文献

- [1] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The L^AT_EX Companion*. Addison-Wesley, Reading, Massachusetts, 1993.
- [2] Albert Einstein. *Zur Elektrodynamik bewegter Körper*. (German) [*On the electrodynamics of moving bodies*]. Annalen der Physik, 322(10):891–921, 1905.
- [3] Knuth: Computers and Typesetting,
<http://www-cs-faculty.stanford.edu/~uno/abcde.html>

使用 thebibliography 环境在文档中插入参考文献并进行编译。

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

Some examples for showing how to use \texttt{thebibliography} environment:

```
\begin{itemize}
  \item Book reference sample: The \LaTeX\ companion book \cite{
    latexcompanion}.
  \item Paper reference sample: On the electrodynamics of moving bodies \
    cite{einstein}.
  \item Open-source reference sample: Knuth: Computers and Typesetting \
    cite{knuthwebsite}.
\end{itemize}
```

```
\begin{thebibliography}{9}
```

```
\bibitem{latexcompanion}
```

Michel Goossens, Frank Mittelbach, and Alexander Samarin.

\textit{The \LaTeX\ Companion}.

Addison–Wesley, Reading, Massachusetts, 1993.

```
\bibitem{einstein}
```

Albert Einstein.

\textit{Zur Elektrodynamik bewegter K{"o}rper}. (German)

[\textit{On the electrodynamics of moving bodies}].

Annalen der Physik, 322(10):891 – 921, 1905.

```
\bibitem{knuthwebsite}
```

Knuth: Computers and Typesetting,

\\ \texttt{http://www-cs-faculty.stanford.edu/~{}uno/abcde.html}

```
\end{thebibliography}
```

```
\end{document}
```

3.2 使用 BibTeX 文件

BibTeX 是 L^AT_EX 最为常用的一个文献管理工具，它通常以一个独立的文件出现，其拓展名为.bib。它是伴随着 L^AT_EX 文档排版系统出现的，1985 年兰伯特博士与其合作者开发了这一工具。作为一种特殊的且独立于 L^AT_EX 文件.tex 之外的数据库，它能大大简化 L^AT_EX 文档中的文献引用。实际上，BibTeX 文件中的文献都是用列表的形式罗列的，且不分先后顺序。通过使用引用命令如\cite{} 即可在文档中自动生成特定格式的参考文献，其中，文档中的参考文献格式一般是提前设定好的。

使用 BibTeX 命令一个文献管理文件为 sample.bib，将文献按照指定格式进行整理，插入参考文献并进行编译。

% 创建 BibTeX 文件，并将其命名为 sample.bib

```
@article{einstein,
  author = "Albert Einstein",
  title = "{Zur Elektrodynamik bewegter K{\o}rper}. ({German})
    [{On} the electrodynamics of moving bodies]",
  journal = "Annalen der Physik",
  volume = "322",
  number = "10",
  pages = "891--921",
  year = "1905",
  DOI = "http://dx.doi.org/10.1002/andp.19053221004"
}

@book{latexcompanion,
  author = "Michel Goossens and Frank Mittelbach and Alexander Samarin",
  title = "The \LaTeX\ Companion",
  year = "1993",
  publisher = "Addison-Wesley",
  address = "Reading, Massachusetts"
}

@misc{knuthwebsite,
  author = "Donald Knuth",
  title = "Knuth: Computers and Typesetting",
  url = "http://www-cs-faculty.stanford.edu/~{ }uno/abcde.html"
}
```

在这三条文献中，einstein、latexcompanion、knuthwebsite 是文献的标签，在文档中，只需要在适当位置用引用命令如\cite{} 便可以引用这些文献，例如，\cite{einstein}。

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



```

\begin{document}

Some examples for showing how to use \texttt{thebibliography} environment:
\begin{itemize}
  \item Book reference sample: The \LaTeX\ companion book \cite{
    latexcompanion}.
  \item Paper reference sample: On the electrodynamics of moving bodies \
    cite{einstein}.
  \item Open-source reference sample: Knuth: Computers and Typesetting \
    cite{knuthwebsite}.
\end{itemize}

\bibliographystyle{unsrt}
\bibliography{sample}

\end{document}

```

在 sample.bib 文件中，根据文献类型可定义文献列表，对于每篇文献，需要整理 author（作者信息）、title（文献标题）等基本信息。在 L^AT_EX 文档中，我们需要使用 \bibliographystyle 命令申明参考文献的格式，如本例中的 unsrt，同时，使用 \bibliography 命令申明参考文献的源文件，即 sample.bib。

当然，BibTeX 文献管理也有很多优点：

- 无需重复输入每条参考文献。文献放在 BibTeX 之后，引用文献的标签即可在文档中显示参考文献。
- 文档中的参考文献格式是根据文档样式自动设置的，且所有文献的引用风格是一致的。
- 引用同一作者同年的文献过多时，引用格式会自动调整。
- BibTeX 文件中的文献只有在文档中明确引用才会显示在文档的参考文献中。

在 BibTeX 文件中，不同类型的文献是需要进行分类的：

- article：对应着期刊或杂志上发表的论文，必须添加的信息有 author（作者）、title（标题）、journal（期刊）、year（年份）、volume（卷），可供选择添加的信息包括 number（期）、pages（页码）、month（月份）、doi（数字对象识别码）等。
- book：对应着书籍，必须添加的信息有 author/editor（作者或主编）、title（书名）、publisher（出版社）、year（年份），可供选择添加的信息包括 volume/number（卷/期）、series（系列）、address（出版地址）、edition（版号）、month（月份）、url（网址）等。

- `inbook`: 书籍中的一部分或者某一章节, 必须添加的信息有 `author/editor`、`title` (标题)、`chapter/pages` (章节/页码)、`publisher` (出版社)、`year` (年份), 其他可供选择添加的信息与 `book` 一致。
- `inproceedings`: 对应着会议论文, 必须添加的信息有 `author` (作者)、`title` (论文标题)、`booktitle` (论文集标题)、`year` (年份), 可供选择添加的信息包括 `editor` (版号)、`volume/number` (卷或期)、`series` (系列)、`pages` (页码)、`address` (地址)、`month` (月份)、`organization` (组织方)、`publisher` (出版社) 等。
- `conference`: 对应着会议论文, 与 `inproceedings` 用法一致。
- `mastersthesis` 和 `proceedings`: 分别对应着硕士学位论文和博士学位论文, 必须添加的信息有 `author` (作者)、`title` (标题)、`school` (学校或研究机构)、`year` (年份)

4 文献引用格式

Bibtex 的最大特点是采用了标准化的数据库, 对于论文、著作以及其他类型的文献, 我们可以自定义文献的引用格式。Bibtex 的样式会改变所引用文献的引用格式。

一般而言, LaTeX 中有一系列标准样式 (standard styles) 可供选择和使用。具体而言, 这些标准样式对应的文件包括:

- `plain.bst`
- `acm.bst`: 对应于 Association for Computing Machinery 期刊
- `ieeetr.bst`: 对应于 IEEE Transactions 期刊
- `alpha.bst`
- `abbrv.bst`
- `siam.bst`: 对应于 SIAM

当然, 实际上还有很多 `.bst` 文件, 这里给出的几个文件只是最为常用的。不得不提的是 `natbib` 工具包, 这一工具包对一系列引用命令进行了标准化, 而这种标准化不受不同文献样式的影响。

Part VIII

幻灯片制作

* 该部分代码未经全部验证，故不能保证正确运行 *

2003 年，柏林工业大学的博士生 Till Tantau 发布了一款用于制作演示文稿的工具 Beamer。Beamer 是 Till Tantau 利用业余时间开发的，他的初衷是方便自己使用 LaTeX 制作幻灯片，不过出乎意料的是，后来 Beamer 的受欢迎程度完全超出了他的想象。在 Till Tantau 开发 Beamer 期间，他收到了很多人的建议和反馈，这些都直接推动了开发工作。2010 年，Till Tantau 将 Beamer 交由他人维护和管理。

Beamer 作为 LaTeX 的一种特殊文档类型，它的出现无疑丰富了 LaTeX 制作演示文稿的功能。虽然 Beamer 并非 LaTeX 中第一款用于制作演示文稿的工具，但直到今日，Beamer 却是最受大家欢迎的一款。Beamer 的使用方式简单灵活，只需设定 LaTeX 文档类型为 beamer，便可开始创作。同时，Beamer 提供了大量的幻灯片模板，这些模板包含了丰富多彩的视觉效果，创作者可以直接使用。毫不夸张地说，Beamer 的出现极大地提高了人们使用 LaTeX 制作幻灯片的热情。值得一提的是，在 2005 年，Till Tantau 又发布了一款非常便捷的 LaTeX 绘图工具 TikZ。TikZ 不仅可以辅助 Beamer 幻灯片的制作，也可以应用于科技文档中的各类绘图任务。

Beamer 是随着 LaTeX 的发展而衍生出来的一种特殊文档类型，也常常被看作是一个功能强大的宏包，可以支撑科研工作者制作幻灯片的需求。使用 Beamer 制作幻灯片与 Office 办公软件（如 PowerPoint）完全不同，从本质上来说，使用 Beamer 制作幻灯片其实和 LaTeX 中的其他文档类型没有太大区别：代码结构都是由前导代码和主体代码组成，完全沿用了 LaTeX 的文档环境与基本命令。因此，使用 Beamer 制作幻灯片也有一个缺点，那就是必须掌握 LaTeX 制作文档的基础。

在呈现方式上，Beamer 制作的幻灯片最终会在 LaTeX 中被编译成 PDF 文档，在不同的系统上打开幻灯片时不存在不兼容等问题。在功能上，使用 Beamer 制作幻灯片时，我们可以对常规文本、公式、列表、图表甚至动画效果、视觉效果和主题样式等进行调整，最终达到想要的视觉效果。

事实上，LaTeX 中用于制作演示文稿的工具并非只有 Beamer，但相比其他工具，Beamer 具有如下优点：

- 拥有海量的模板和丰富的主题样式，且使用方便；
- 能满足制作幻灯片的功能性需求，从创建标题、文本和段落到插入图表、参考文献等操作，且与常规文档中的使用规则几乎一致；
- 使用方式简单，在主体代码中使用 `\begin{frame}` `\end{frame}` 环境就能创建一页幻灯片。

1 基本介绍

1.1 Beamer 简介

在上述章节中，我们主要介绍了 LaTeX 中比较常用的文档类型 `article`，可用于创建期刊论文、技术报告等。本章中我们将介绍另一种文档类型：`beamer`。Beamer 的开发者 Till Tantau 说，“BEAMER is a LATEX class for creating presentations”，显然，Beamer 是一种用于制作演示文稿或者幻灯片的文档类型。

从使用角度来说，`beamer` 文档类型和 `book`、`article` 等文档类型一样，都是在以 `.tex` 为拓展名的文件上编写程序和文档内容，然后再通过编译生成 PDF 文档。当然，Beamer 也兼具常用演示文稿如 PowerPoint 的主要功能，可以自行设置动态效果、甚至使用主题样式修改幻灯片的外观。

与其他文档类型相似的是，`beamer` 文档类型中拥有很多视觉效果极好的模板，这些模版已经设置好了特定的主题样式，有时候甚至只需要加入创作内容即可得到心仪的幻灯片。使用 Beamer 制作幻灯片时，我们可以体验 LaTeX 排版论文的几乎所有优点，公式排版、图表排版、参考文献设置等也非常便捷，有时候甚至可以将常规文档中的内容直接复制到 Beamer 文档类型中，稍加调整便能得到样式合适的幻灯片。另外，我们也可以根据需要，在前导代码中使用全局设置调整幻灯片的主题样式、颜色主题、字体主题等。

使用 `beamer` 制作幻灯片仍然遵循着 LaTeX 的一般使用方法，代码结构分为前导代码和主体代码，前导代码除了申明文档类型为 `beamer` 外，即 `\documentclass{beamer}`，调用宏包等与常规文档的制作基本是一致的。

使用 `beamer` 文档类型创建一个简单的幻灯片。

```
\documentclass{beamer}

\title{A Simple Beamer Example}
\author{Author's Name}
\institute{Author's Institute}
\date{\today}

\begin{document}

\frame{\titlepage}

\end{document}
```

在例子中，`\title{}`、`\author{}` 和 `\date{}` 这几个命令分别对应着标题、作者以及日期，一般放在标题页，如果想在幻灯片首页显示这些信息，可以在使用 `\frame{\titlepage}` 命令新建标题页。

总结来说，标题及作者信息对应的特定命令包括：

- 标题：对应的命令为`\title[A]{B}`，其中，位置 A 一般填写的是简化标题，而位置 B 则填写的是完整标题，这里的完整标题有时候可能会很长。
- 副标题：对应的命令为`\subtitle[A]{B}`，其中，位置 A 一般填写的是简化副标题，而位置 B 则填写的是完整副标题，这里的完整副标题有时候也可能会很长。
- 作者：对应的命令为`\author[A]{B}`，用法类似。
- 日期：对应的命令为`\date[A]{B}`，用法类似。
- 单位：对应的命令为`\institution[A]{B}`，用法类似。

我们知道，在常规文档 `article` 中，申明文档类型时可以指定正文字体大小，在文档类型的申明语句`\documentclass{beamer}`中，我们也可以通过特定选项调整幻灯片内容的字体大小，一般默认为 11pt，我们也可以根据需要使用 8pt、9pt、10pt、12pt、14pt、17pt、20pt 字体大小，例如使用`\documentclass[12pt]{beamer}`可以将字体大小设置为 12pt。

制作幻灯片时，有时候为了达到特定的投影效果，会设置幻灯片的长宽比例，比较常用的两种长宽比例分别为 4:3 和 16:9。一般来说，Beamer 制作出来的幻灯片默认大小为长 128 毫米、宽 96 毫米，对应着默认的长宽比例 4:3，有时候，我们也可以根据需将幻灯片的长宽比例调整为 16:9、14:9、5:4 甚至 3:2。

使用 beamer 文档类型创建一个简单的幻灯片，将幻灯片的长宽比例调整为 16:9。在例子中，选项 `aspectratio` 对应着长宽比例，数字 169 对应着长宽比例 16:9，类似地，149、54、32 分别对应着长宽比例 14:9、5:4、3:2。

```
\documentclass[aspectratio = 169]{beamer}

\title{A Simple Beamer Example}
\author{Author's Name}
\institute{Author's Institute}
\date{\today}

\begin{document}

\frame{\titlepage}

\end{document}
```

1.2 创建幻灯片

frame 这个词在计算机编程中非常常见，这一英文单词的字面意思可以翻译为“帧”，假如我们将幻灯片视作“连环画”，是由一页一页单独的幻灯片组成，那么每一页幻灯片则对应着连环画中的帧。使用 Beamer 制作幻灯片时，幻灯片就是用 frame 环境创建出来的，然而，有时候为了让幻灯片产生动画视觉效果，Beamer 中的帧（即 frame）与每页幻灯片并非严格意义上的一一对应。

在 beamer 文档类型中，制作幻灯片的环境一般为 `\begin{frame} \end{frame}`。在 `\begin{document} \end{document}` 构成的主体代码中，一个 frame 环境一般对应着一页幻灯片。

每张幻灯片一般都有一个标题，有时也会有一个副标题。若要创建标题和副标题，用户可以通过使用 `\begin{frame}{}{} \end{frame}` 的命令格式，其中第一、二个 `{}` 中分别为幻灯片的标题和副标题；此外，用户也可以通过在 frame 环境中，使用 `\frametitle{}` 和 `\framesubtitle{}` 命令分别创建标题和副标题。由此创建的标题和副标题一般位于幻灯片的顶部，标题相对于副标题字体稍大一点。

实际上，Beamer 与其他文档类型并没有特别大的差异，常规文档中的基本列表环境都可以在 Beamer 中使用，包括：有序列表环境 `\begin{enumerate} \end{enumerate}`、无序列表环境 `\begin{itemize} \end{itemize}` 以及解释性列表环境 `\begin{description} \end{description}`。

```

使用beamer文档类型中的\begin{frame} \end{frame}环境、\frametitle{}和\framesubtitle{}命令创建一个简单的幻灯片。
有时为了简化代码，也可以直接用\frame{}命令取代\begin{frame} \end{frame}环境
    囊括幻灯片内容
\documentclass{beamer}
\usefonttheme{professionalfonts}
\begin{document}
\begin{frame}
\frametitle{Parent function}
\framesubtitle{A short list}
Please check out the following parent function list.
\begin{enumerate}
\item  $y=x$ 
\item  $y=|x|$ 
\item  $y=x^2$ 
\item  $y=x^3$ 
\item  $y=x^b$ 
\end{enumerate}
\end{frame}
\end{document}

```

使用 Beamer 制作幻灯片时，幻灯片内容会在标题下方自动居中对齐，如果想调整对其方式，可以在 `\begin{frame} \end{frame}` 环境中设置参数，具体而言，有以下几种：

1. `\begin{frame}[c] \end{frame}` 是居中对齐，字母 c 对应着英文单词 center 的首字母，一般而言，[c] 作为默认参数，无需专门设置；
2. `\begin{frame}[t] \end{frame}` 中的 [t] 可以让幻灯片内容进行顶部对齐，其中，字母 t 对应着英文单词 top 的首字母；
3. `\begin{frame}[b] \end{frame}` 中的 [b] 可以让幻灯片内容进行底部对齐，其中，字母 b 对应着英文单词 bottom 的首字母。

使用 beamer 文档类型中的 `\begin{frame} \end{frame}` 环境创建一个简单的幻灯片，并让幻灯片内容进行顶部对齐。

```
\documentclass{beamer}
\usefonttheme{professionalfonts}
\begin{document}
\begin{frame}[t]
\frametitle{Parent function}
\framesubtitle{A short list}
Please check out the following parent function list.
\begin{enumerate}
\item  $y=x$ 
\item  $y=|x|$ 
\item  $y=x^2$ 
\item  $y=x^3$ 
\item  $y=x^b$ 
\end{enumerate}
\end{frame}
\end{document}
```

1.3 创建章节与生成目录

类似 article 文档类，beamer 中可以利用 `\part{}`、`\section{}`、`\subsection{}`、以及 `\subsubsection{}` 等命令构建演示稿中的章节层次，但此时 `\chapter{}` 命令无效。其中，章节标题写在 {} 中，但编译后不会出现在创建章节的位置，仅在目录和导航条中显示。类似地，可以通过加 * 号使得章节标题不出现在目录中，但仍然会在导航条中显示。

在 beamer 中，可以使用 `\tableofcontents` 命令自动生成演示稿目录，通过在 frame 幻灯片页中添加该命令即可。由此生成的目录实际上是超链接，点击之后会自动跳转到相应章节

在beamer文档类型中使用\tableofcontents命令为幻灯片生成目录，并使用\section{}和\subsection{}创建章节。

```
\documentclass{beamer}
```

```
\begin{document}
```

```
\begin{frame}{Table of contents}
```

```
\tableofcontents
```

```
\end{frame}
```

```
\section{Section A}
```

```
\begin{frame}{frame1}
```

```
\subsection{a1}
```

```
This is subsection a1. This is subsection a1.
```

```
\subsection{a2}
```

```
This is subsection a2. This is subsection a2.
```

```
\subsection{a3}
```

```
This is subsection a3. This is subsection a3.
```

```
\end{frame}
```

```
\section{Section B}
```

```
\begin{frame}{frame2}
```

```
\subsection{b1}
```

```
This is subsection b1. This is subsection b1. % 在下方插入空行，使得内容分行显示.
```

```
\subsection{b2}
```

```
This is subsection b2. % 在下方插入空行，使得内容分行显示.
```

```
This is subsection b2.
```

```
\end{frame}
```

```
\section*{Section C}
```

```
\begin{frame}{frame3}
```

```
\subsection*{c1}
```

```
This is subsection c1. This is subsection c1. % 在下方插入空行，使得内容分行显示.
```

```
\subsection*{c2}
```

```
This is subsection c2. This is subsection c2.
```

```
\end{frame}
```

```
\end{document}
```


如果想让相邻章节或者同章节的内容分行显示，只需要在相应位置插入空行即可。

默认情况下，目录页中包含所有不含 * 号的章节标题，甚至是三级节标题。但有时目录只需要显示到一级节标题即可，而二级节标题及其次级标题则不需要显示，为此，只需要在 `\tableofcontents` 命令后设置选项 `[hideallsubsections]` 即可。

在beamer文档类型中使用 `\tableofcontents[hideallsubsections]` 命令为幻灯片生成一级节目录。

```
\documentclass{beamer}

\begin{document}

\begin{frame}{table of contents}
\tableofcontents[hideallsubsections]
\end{frame}

\section{Section A}
\begin{frame}{frame1}
\subsection{a1}
This is subsection a1. This is subsection a1.

\subsection{a2}
This is subsection a2. This is subsection a2.

\subsection{a3}
This is subsection a3. This is subsection a3.

\end{frame}

\section{Section B}
\begin{frame}{frame2}
\subsection{b1}
This is subsection b1. This is subsection b1.

\subsection{b2}
This is subsection b2. This is subsection b2.
\end{frame}

\end{document}
```

一般而言，使用 `\tableofcontents` 命令生成的目录只会显示在相应的幻灯片页。有时候为了更好地梳理演示稿脉络，需要在各章节前均插入目录页，为此，一种更简便的方式是使用 `\AtBeginSection{}`、`\AtBeginSubsection{}`、或 `\AtBeginSubsubsection{}` 命令分别在一级节、二级节、三级节前均插入目录页。此外，使用 `\tableofcontents[currentsection]`

命令或`\tableofcontents[currentsubsection]` 命令可以在各章节前的目录页中突出显示当前一级节标题或二级节标题。

在beamer文档类型中使用`\AtBeginSection{}`以及`\tableofcontents[currentsection]` 命令在幻灯片的各一级节前均插入目录页，并突出显示当前一级节标题。

```
\documentclass{beamer}
\begin{document}
\AtBeginSection
{
\begin{frame}{table of contents}
\tableofcontents[currentsection]
\end{frame}
}

\section{Section A}
\begin{frame}{frame1}
\subsection{a1}
This is subsection a1. This is subsection a1.

\subsection{a2}
This is subsection a2. This is subsection a2.

\subsection{a3}
This is subsection a3. This is subsection a3.
\end{frame}

\section{Section B}
\begin{frame}{frame2}
\subsection{b1}
This is subsection b1. This is subsection b1.

\subsection{b2}
This is subsection b2. This is subsection b2.
\end{frame}

\section{Section C}
\begin{frame}{frame3}
\subsection{c1}
This is subsection c1. This is subsection c1.

\subsection{c2}
This is subsection c2. This is subsection c2.
\end{frame}
\end{document}
```

生成目录时，自定义目录显示的动画格式，通过使用`\tableofcontents[pausesections]`命令，同时在前导代码中申明`\setbeamercovereddynamic` 语句即可。

在beamer文档类型中使用`\tableofcontents`命令生成幻灯片的目录，同时使用`\tableofcontents[pausesections]`对目录设置动画格式。

```
\documentclass{beamer}
\setbeamercovered{dynamic}

\begin{document}

\begin{frame}
\frametitle{Table of Contents}

\tableofcontents[pausesections]

\end{frame}

\section{Intro to Beamer}
\subsection{About Beamer}
\subsection[Basic Structure]{Basic Structure}
\subsection{How to Compile}
\section{Overlaying Concepts}
\subsection{Specifications}
\subsection[Examples]{Examples: Lists , Graphics , Tables}
\section[Sparkle]{Adding that Sparkle}
\subsection{Sections}
\subsection{Themes}
\section*{References}

\begin{frame}

\end{frame}

\end{document}
```

1.4 幻灯片分栏

对幻灯片内容进行分栏有两种常用方式，第一种是使用 `multicol` 宏包中的 `\begin{multicols}{A}` `\end{multicols}` 环境，其中位置 A 可用于设定分栏列数；第二种是使用 `\begin{columns}` `\end{columns}` 环境。

在beamer文档类型中使用multicol宏包对列表内容进行分栏处理。

```
\documentclass{beamer}
\usefonttheme{professionalfonts}
\usepackage{multicol}

\begin{document}

\begin{frame}
\frametitle{Parent function}
\framesubtitle{A short list}

Please check out the following parent function list.
\begin{enumerate}
\begin{multicols}{3}
\item  $y=x$ 
\item  $y=|x|$ 
\item  $y=x^2$ 
\item  $y=x^3$ 
\item  $y=x^b$ 
\end{multicols}
\end{enumerate}

\end{frame}

\end{document}
```

在beamer文档类型中使用\begin{columns} \end{columns}环境对幻灯片内容进行分栏处理。

```
\documentclass{beamer}
\usefonttheme{professionalfonts}

\begin{document}

\begin{frame}
\frametitle{Parent function}
\framesubtitle{A short list}

\begin{columns}
\begin{column}{0.5\textwidth}

Please check out the following parent function list.
\begin{enumerate}
```

```

\item $y=x$
\item $y=|x|$
\item $y=x^{\{2\}}$
\item $y=x^{\{3\}}$
\item $y=x^{\{b\}}$
\end{enumerate}

\end{column}

\begin{column}{0.5\textwidth}

Please check out the following parent function list.
\begin{enumerate}
\item $y=x$
\item $y=|x|$
\item $y=x^{\{2\}}$
\item $y=x^{\{3\}}$
\item $y=x^{\{b\}}$
\end{enumerate}

\end{column}
\end{columns}

\end{frame}

\end{document}

```

2 添加动画效果

在制作幻灯片时有时需要添加动画效果。由于 LaTeX 制作幻灯片会被编译成 PDF 文档，因此，在 Beamer 中，实现动画效果的方式是将具有动画内容的幻灯片按照次序拆分成若干页内容，在播放时通过翻页达到“动态”视觉效果。为了便于说明，以下将一个 frame 环境创建的内容称为一页幻灯片或幻灯片页、将动画效果拆分后得到的每一页内容称为该幻灯片的某一帧。

2.1 \pause 命令

。 \pause 是 Beamer 中最常用的一种动画效果命令，它的使用方式极其简单，通过在文本或段落中添加 \pause 命令，便可将一页幻灯片拆分成若干帧。一般来说， \pause 命令后的内容将会在下一帧中显示，从而使幻灯片在内容显示上呈现出动画效果。比

如，一般情况下，使用列表环境创建的每项内容（使用`\item` 创建）都会在同一帧幻灯片中显示，为了达到各项内容逐个显示的动画效果，可以在两个相邻的`\item` 语句之间插入`\pause` 命令。

在beamer文档类型中使用`\pause`命令实现一个简单的动画效果。

```
\documentclass{beamer}
\usefonttheme{professionalfonts}

\begin{document}

\begin{frame}
\frametitle{Parent function}
\framesubtitle{A short list}

Please check out the following parent function list.
\begin{enumerate}
\item  $y=x$ 
\pause
\item  $y=|x|$ 
\pause
\item  $y=x^2$ 
\pause
\item  $y=x^3$ 
\pause
\item  $y=x^b$ 
\end{enumerate}
\end{frame}

\end{document}
```

2.2 `\item<>` 命令

对于列表环境中的各项内容，也可以通过设置选项指定在该幻灯片的哪些步骤中显示该项内容，从而更加灵活地定制动画效果。具体是使用`\item<>` 命令，其中 `<>` 用于指定显示步骤，对于没有指定 `<>` 显示范围的内容项默认会在所有幻灯片页面中显示。具体而言，`<>` 中的内容存在以下四种格式：

- `<A,B,C>`：表示内容项将在第 A、B、C 步中显示。如，`\item<2, 3, 4> $y = x^2$` 表示该项内容将出现在该页幻灯片放映的第 2、3、4 步；
- `<A-B>`：表示内容项将在第 A 至 B 步中显示。如，`\item<1-4> $y = x$` 表示该项内容将出现在该页幻灯片放映的第 1 4 步；

- `<A->`: 表示内容项将在第 A 步及以后显示。如, `\item<2-> y = x` 表示该内容将出现在该页幻灯片放映的第 2 步及之后的步骤中;
- `<-A>`: 表示内容项将在第 A 步及之前显示。如, `\item<-3> y = x` 表示该内容将出现在该页幻灯片放映的第 3 步及之前的步骤中。

如果要在某一帧中突出某项内容:

使用 `\alert` 命令为该内容指定需要高亮显示的步骤。具体用法如:

`\item<2-| alert@3-4> The second item.`, 此时内容项 “The second item.” 将出现在第 2 步之后的步骤中, 并通过命令 `\alert` 及前缀 `@` 使其在第 3 4 步中高亮显示。

使用 `\color< 范围 >{显示颜色}` 命令改变内容项的颜色。如

`\item<1-> \color<1>{red} The first item.` 语句, 内容 The first item. 将出现在第一步及之后的步骤中, 通过 `\color<1>{red}` 令该内容在第一步显示颜色为红色, 而在其它步骤中仍为默认颜色。

在 beamer 文档类型中使用 `\item<>` 定制内容显示范围并使用 `\alert` 对内容项进行高亮显示, 从而实现一个简单的动画效果。

```
\documentclass{beamer}
\usefonttheme{professionalfonts}
```

```
\begin{document}
```

```
\begin{frame}
```

```
\frametitle{Parent function}
```

```
\framesubtitle{A short list}
```

Please check out the following parent function list.

```
\begin{enumerate}
```

```
\item<1-| alert@1> $y=x$
```

```
\item<2-| alert@2> $y=|x|$
```

```
\item<3-| alert@3> $y=x^{\{2\}}$
```

```
\item<4-| alert@4> $y=x^{\{3\}}$
```

```
\end{enumerate}
```

```
\end{frame}
```

```
\end{document}
```

在 beamer 文档类型中使用 `\item<>` 定制内容显示范围并使用 `\color<>{}` 对内容项进行高亮显示。

```
\documentclass{beamer}
```

```
\usefonttheme{professionalfonts}
```

```

\begin{document}

\begin{frame}
\frametitle{Parent function}
\framesubtitle{A short list}

Please check out the following parent function list.
\begin{enumerate}
\item<1-> \color{red} $y=x$
\item<2-> \color{red} $y=|x|$
\item<3-> \color{red} $y=x^2$
\item<4-> \color{red} $y=x^3$
\end{enumerate}

\end{frame}

\end{document}

```

2.3 其他命令

LaTeX 还提供了其它命令可以实现类似的动画效果, 同样可以在可选参数 `<>` 中指定内容项或内容块的显示范围, 主要包括 `\onslide`、`\uncover`、`\only`、`\visible`、`\invisible` 等命令:

`\onslide<>{}`: 该命令可以指定内容在当前幻灯片页放映的第几步显示。使用该命令时不显示的内容将被“遮挡”, 仍将占用其指定的位置 (`\uncover<>{}` 或 `\visible<>{}` 也能实现类似效果);

`\only<>{}`: 该命令可以指定内容在当前幻灯片页放映的第几步插入。使用该命令时, 不显示的内容对应的位置将腾空出来, 可以用于显示其它内容

`\invisible<>{}`: 该命令的作用效果与 `\visible<>{}` 相反, 用于指定内容在当前幻灯片页放映的第几步不可见。但与 `\visible<>{}` 相同的是, 使用 `\invisible<>{}` 命令时, 不可见的内容仍占据着对应的位置, 不可用于显示其它内容。如下例中的代码所示, 编译后得到的效果与图 9.1.8 一致

在 beamer 文档类型中使用 `\onslide<>{}` 命令实现一个简单的动画效果。

```

\documentclass{beamer}
\usefonttheme{professionalfonts}

\begin{document}

\begin{frame}

```



```

\frametitle{Parent function}
\framesubtitle{A short list}

\onslide<1->{Please check out the following parent function list.}

\onslide<2,4>{1.  $y=x$ }

\onslide<1-4>{2.  $y=|x|$ }

\onslide<2->{3.  $y=x^2$ }

\onslide<3->{4.  $y=x^3$ }

\onslide<4>{5.  $y=x^b$ }

\end{frame}

\end{document}

```

在beamer文档类型中使用\only<>{}命令实现一个简单的动画效果。

```

\documentclass{beamer}
\usefonttheme{professionalfonts}
\begin{document}
\begin{frame}
\frametitle{Parent function}
\framesubtitle{A short list}

\only<1->{Please check out the following parent function list.}

\only<2,4>{1.  $y=x$ }

\only<1-4>{2.  $y=|x|$ }

\only<2->{3.  $y=x^2$ }

\only<3->{4.  $y=x^3$ }

\only<4>{5.  $y=x^b$ }

\end{frame}
\end{document}

```

在beamer文档类型中使用`\invisible<>\}`命令实现一个简单的动画效果。

```
\documentclass{beamer}
\usefonttheme{professionalfonts}

\begin{document}

\begin{frame}
\frametitle{Parent function}
\framesubtitle{A short list}

\visible<1-4>{Please check out the following parent function list.}

\invisible<1,3>{1.  $y=x$ }

\invisible<>{2.  $y=|x|$ }

\invisible<1>{3.  $y=x^2$ }

\invisible<1-2>{4.  $y=x^3$ }

\invisible<1-3>{5.  $y=x^b$ }

\end{frame}

\end{document}
```

2.4 自动计数

上述介绍的动画效果定制命令均通过在`<>`中指定具体的数字指定内容显示范围。此时，如果想要在开始处或中间插入新的内容项，则其余所有内容项的`<>`显示范围都必须作出相应调整，显然不够灵活。L^AT_EX提供了一种更巧妙的方式可以解决这类问题：使用“+”号代替具体数字，从1开始进行自动递增计数。就例9-13而言，可以用“+”符号代替各`<>`中的具体数字，可以得到完全相同的编译效果。

在beamer文档类型中使用+符号灵活定制幻灯片效果。

```
\documentclass{beamer}
\usefonttheme{professionalfonts}

\begin{document}

\begin{frame}
\frametitle{Parent function}
```

```

\framesubtitle{A short list}

Please check out the following parent function list.
\begin{enumerate}
\item<+| alert@+>  $y=x$  % 此时 “+” 号对应数字1
\item<+| alert@+>  $y=|x|$  % 此时 “+” 号对应数字2
\item<+| alert@+>  $y=x^2$  % 此时 “+” 号对应数字3
\item<+| alert@+>  $y=x^3$  % 此时 “+” 号对应数字4
\end{enumerate}

\end{frame}

\end{document}

```

上述语句每一条\item格式相同，因此也可以简写为如下语句：

```

\documentclass{beamer}
\usefonttheme{professionalfonts}

\begin{document}

\begin{frame}
\frametitle{Parent function}
\framesubtitle{A short list}

Please check out the following parent function list.
\begin{enumerate}[<+| alert@+>]
\item  $y=x$ 
\item  $y=|x|$ 
\item  $y=x^2$ 
\item  $y=x^3$ 
\end{enumerate}

\end{frame}

\end{document}

```

有时在 <> 中使用的数字不总是从 1 开始递增，那么就需要使用 “+(偏移量)” 的命令格式。比如，如果当前 “+” 号对应的计数器值为 3，那么 <+(2)-> 意味着在当前计数器值的基础上加 2，<+(-2)-> 则意味着在当前计数器值的基础上减 2。

在beamer文档类型中使用+(偏移量)符号灵活定制任意显示步骤的幻灯片效果。

```

\documentclass{beamer}
\usefonttheme{professionalfonts}

\begin{document}

```

```

\begin{frame}
\frametitle{Parent function}
\framesubtitle{A short list}

Please check out the following parent function list.
\begin{enumerate}
\item<+(1)->  $y=x$  % 相当于 '\item<2->  $y=x$ '
\item<-+(2)>  $y=|x|$  % 相当于 '\item<-4>  $y=|x|$ '
\item<+(-1)-+(1)>  $y=x^2$  % 相当于 '\item<2-4>  $y=x^2$ '
\item<+(-1)>  $y=x^3$  % 相当于 '\item<3>  $y=x^3$ '
\end{enumerate}

\end{frame}

\end{document}

```

3 创建文本框

在幻灯片中框选文本或图片等元素是常见的操作，可以对幻灯片内容进行划分或者突出重点内容。在 Beamer 中，可以通过添加区块环境（block environments）或创建盒子（box）结构的方式将文本等元素放在各式各样的框中。

3.1 区块环境

Beamer 提供了区块环境（block）可用于编辑文本内容，通过 block 环境创建的文本内容将放置在一个框中，使其与普通文本区分开。根据内容样式和使用目的的不同，包括三种区块环境：

- block：一般性区块环境。使用语法为 `\begin{block}< 指定显示步骤 >{设置标题}`
`\end{block}`；
- alertblock：警示性区块环境，主要用于创建警示信息。使用语法为
`\begin{alertblock}< 指定显示步骤 >{设置标题}` `\end{alertblock}`；
- exampleblock：示例性区块环境，主要用于创建示例文本。使用语法为
`\begin{exampleblock}< 指定显示步骤 >{设置标题}` `\end{exampleblock}`。

在三种区块环境的开始命令中（如：`\begin{block}< 指定显示步骤 >{设置标题}`），“<>”可用于指定当前区块内容显示的步骤，实现动画效果；第二个“{”可用于设置

该区块内容的标题，标题将显示在区块内容的上面。此外，区块内容的样式由使用的 Beamer 主题样式决定。

```

在 beamer 文档类型中使用 block 环境插入一个一般文本框、使用 alertblock 环境插入一个
警示性文本框、以及使用 exampleblock 环境插入一个示例性文本框。

\documentclass{beamer}
\usefonttheme{professionalfonts}
\usetheme{Copenhagen}

\begin{document}

\begin{frame}
\begin{block}<1>{Block1}
This is a generic block.
\end{block}

\begin{alertblock}<1>{Block2}
This is an alert block.
\end{alertblock}

\begin{exampleblock}<1>{Block3}
This is an example block.
\end{exampleblock}
\end{frame}

\end{document}

```

3.2 定理类环境

对于定理、引理、推论、示例等定理类文本，除了可以考虑使用区块环境创建之外，Beamer 也预定义了相应的命令环境可供使用，包括：

- definition: 定义环境。使用语法为 `\begin{definition}<指定显示步骤>\end{definition}`;
- fact: 事实环境。使用语法为 `\begin{fact}<指定显示步骤>\end{fact}`;
- theorem: 定理环境。使用语法为 `\begin{theorem}<指定显示步骤>\end{theorem}`;
- lemma: 引理环境。使用语法为 `\begin{lemma}<指定显示步骤>\end{lemma}`;

- `proof`: 证明环境。使用语法为`\begin{proof}< 指定显示步骤 >{设置名称}`
`\end{proof}`;
- `corollary`: 推论环境。使用语法为`\begin{corollary}< 指定显示步骤 >{设置名称}`
`\end{corollary}`;
- `example`: 示例环境等。使用语法为`\begin{example}< 指定显示步骤 >{设置名称}`
`\end{example}`。

定理类环境的使用与区块环境类似：使用定理类环境可以创建文本框；开始命令中的“<>”可用于指定当前内容显示的步骤，实现动画效果；第二个“”可用于设置该定理类内容的名称。不同于区块环境的是：定理类内容的标题默认为对应的定理类型，如在 `definition` 环境下，标题即为“Definition”，显示在定理类内容的上方；而定理类内容的名称允许用户自行定制，通常位于定理类内容的左侧，以较大的斜体字标示。

在beamer文档类型中使用 `definition` 环境插入一个定义文本框、使用 `theorem` 环境插入一个定理文本框、以及使用 `example` 环境插入一个示例文本框。

```
\documentclass{beamer}
\usefonttheme{professionalfonts}
\usetheme{Copenhagen}

\begin{document}

\begin{frame}{Definition, theorem and example}
\begin{definition}<1>{Definition Demo}
This is a definition.
\end{definition}

\begin{theorem}<1>{Theorem Demo}
This is a theorem.
\end{theorem}

\begin{example}<1>{Example Demo}
This is an example.
\end{example}
\end{frame}

\end{document}
```

3.3 Beamer 中的盒子

Beamer 也支持通过绘制外框的方式为幻灯片的元素（如，文本、图片等）加上外框，或者说创建盒子（box）。常用的语法包括调用 `\fbox{}` 命令绘制简单矩形框、或调

用 fancybox 宏包提供的命令 (`\shadowbox`, `\doublebox`, `\ovalbox` 和 `\Ovalbox`) 创建不同类型的外框。

使用 `\fbox{}` 命令可以创建简单的矩形盒子, 调用以下命令可以对盒子参数进行修改:

- `\setlength{\fboxsep}{}`: 设置盒子内的元素与其边框之间的距离, 默认值为 3pt;
- `\setlength{\fboxrule}{}`: 设置盒子边框线的粗细, 默认值为 0.4pt。

此外, 盒子之间的行间距可以使用 `\vskip` 命令进行修改。

在 beamer 文档类型中使用 `\fbox{}` 命令三个文本盒子、使用 `\setlength` 命令设置不同参数、并使用 `\vskip` 命令设置行间距。

```
\documentclass{beamer}

\begin{document}

\begin{frame}
\setlength{\fboxsep}{3pt}
\setlength{\fboxrule}{0.4pt}
\fbox{This is our 1st text box.}
\vskip 5mm
\setlength{\fboxsep}{6pt}
\setlength{\fboxrule}{0.8pt}
\fbox{This is our 2nd text box.}
\vskip 5mm
\setlength{\fboxsep}{9pt}
\setlength{\fboxrule}{1.2pt}
\fbox{This is our 3rd text box.}
\end{frame}

\end{document}
```

对于较短的文本内容, 使用 `\fbox{}` 命令可以实现较好的效果。但由于在 `\fbox{}` 命令中换行符 `\\` 不起作用, 因此如果要对段落文本或长文本创建盒子, 需要先将文本内容放置到段落环境中, 然后再调用 `\fbox{}` 命令。其中, `\begin{minipage}[外部对齐方式][高度][内部对齐方式]{宽度}{内容}` `\end{minipage}` 环境和 `\parbox[外部对齐][高度][内部对齐]{宽度}{内容}` 命令是比较常用的处理段落的语法。

在 beamer 文档类型中使用 `\fbox{}` 命令和 `minipage` 环境创建段落文本盒子。

```
\documentclass{beamer}

\begin{document}
```

```

\begin{frame}

\fbbox{
\begin{minipage}[c][1.8cm][t]{5cm}
{This is our paragraph text box. This is our paragraph text box. This is
  our paragraph text box. This is our paragraph text box.}
\end{minipage}}

\end{frame}

\end{document}

```

在beamer文档类型中使用\fbbox{}命令和\parbox命令创建段落文本盒子。

```

\documentclass{beamer}

\begin{document}

\begin{frame}

\fbbox{
\parbox[c][1.8cm][t]{5cm}
{This is our paragraph text box. This is our paragraph text box. This is
  our paragraph text box. This is our paragraph text box.}}

\end{frame}

\end{document}

```

当然，除了为文本内容创建盒子之外，\fbbox命令也能为图片等非文本内容创建盒子。在beamer文档类型中使用figure环境插入三张图片，并使用\fbbox{}命令将三种图片装入一个盒子中。

```

\documentclass{beamer}

\begin{document}

\begin{frame}

\begin{figure}
\centering
\fbbox{
\includegraphics[width=0.2\linewidth]{redflower.png}
\includegraphics[width=0.2\linewidth]{yellowflower.png}

```



```

\includegraphics[width=0.2\linewidth]{blueflower.png}
}
\caption{Here is a figure box.}
\end{figure}

\end{frame}

\end{document}

```

在 fancybox 宏包中，提供了以下四个命令用来创建不同样式的盒子：

- `\shadowbox{}`：创建阴影盒子；
- `\doublebox{}`：创建两重线盒子；
- `\ovalbox{}`：创建细边线椭圆盒子；
- `\Ovalbox{}`：创建粗边线椭圆盒子。

在 beamer 文档类型中使用 `\shadowbox`，`\doublebox`，`\ovalbox` 和 `Ovalbox` 命令创建不同样式的盒子。

```

\documentclass{beamer}
\usepackage{fancybox}
\begin{document}

\begin{frame}

\setlength{\fboxsep}{5pt}
\setlength{\fboxrule}{2pt}

\shadowbox{This is a shadowbox.}

\vskip 5mm

\doublebox{This is a doublebox.}

\vskip 5mm

\ovalbox{This is an ovalbox.}

\vskip 5mm

\Ovalbox{This is an Ovalbox.}

```

```
\end{frame}

\end{document}
```

在beamer文档类型中使用figure环境插入四张图片，使用\shadowbox，\doublebox，\ovalbox和Ovalbox命令分别为每张图片创建盒子，并使用\parbox命令把图片和标题均包含在盒子中。

```
\documentclass{beamer}
\usepackage{fancybox}
\begin{document}

\setlength{\fboxsep}{5pt}
\setlength{\fboxrule}{2pt}

\begin{frame}
\begin{figure}
\centering
\shadowbox{
\parbox[c][6cm][t]{5cm}{
\includegraphics[width=1\linewidth]{redflower.png}
\caption{A red flower.}}}
\end{figure}
\end{frame}

\begin{frame}
\begin{figure}
\centering
\doublebox{
\parbox[c][6cm][t]{5cm}{
\includegraphics[width=1\linewidth]{yellowflower.png}
\caption{A yellow flower.}}}
\end{figure}
\end{frame}

\begin{frame}
\begin{figure}
\centering
\ovalbox{
\parbox[c][6cm][t]{5cm}{
\includegraphics[width=1\linewidth]{blueflower.png}
\caption{A blue flower.}}}
\end{figure}
\end{frame}
```

```

\begin{frame}
\begin{figure}
\centering
\Ovalbox{
\parbox[c][6cm][t]{5cm}{
\includegraphics[width=1\linewidth]{magentaflower.png}
\caption{A magenta flower.}}}
\end{figure}
\end{frame}

\end{document}

```

4 设置主题样式

使用 Beamer 制作幻灯片的一道特色就是有现成的主题样式可供选择和直接使用，其中，主题样式对于幻灯片的演示效果而言十分重要，简言之，主题样式就是幻灯片的“外观”，改变幻灯片最简单的方式就是变换不同的主题样式。Beamer 中提供的每种主题样式都具有良好的可用性和可读性，这也使得 Beamer 制作出来的幻灯片看起来十分专业，同时，反复使用的难度也不大。

在英文中，主题对应的英文单词为 theme。狭义来看，幻灯片主题是指幻灯片的主题样式；但从广义来看，其实幻灯片主题包括了包括主题样式、颜色主题、字体主题、内部主题、外部主题。

4.1 基本介绍

使用 Beamer 制作幻灯片时，我们可以选择很多已经封装好的幻灯片主题样式，不同样式可以达到不同的视觉效果。其实，使用这些主题样式的方法非常简单。通常来说，在前导代码中插入 `\usetheme{}` 命令即可，例如使用 Copenhagen（哥本哈根主题样式）只需要在前导代码中申明 `\usetheme{Copenhagen}`，这种方式调用主题样式是非常省事。

在 Beamer 文档类型中，有几十种主题样式可供选择和使用，比较常用的主题样式包括以下这些：

- Berlin：柏林主题样式，默认样式为蓝色调。
- Copenhagen：哥本哈根主题样式，默认样式为蓝色调。
- CambridgeUS：美国剑桥主题样式，默认样式为红色调。
- Berkeley：伯克利主题样式，默认样式为蓝色调。

- Singapore: 新加坡主题样式。
- Warsaw: 默认样式为蓝色调。

在beamer文档类型中使用CambridgeUS主题样式制作一个简单的幻灯片。

```
\documentclass{beamer}
\usetheme{CambridgeUS}
```

```
\begin{document}
```

```
\begin{frame}{Example}
```

This is a simple example for the CambridgeUS theme.

```
\end{frame}
```

```
\end{document}
```

当然，在这些主题样式基础上，我们也能够使用一些特定的主题样式如颜色主题、字体主题、内部主题、外部主题对幻灯片的显示效果进行调整。

4.2 颜色主题

使用 Beamer 制作幻灯片时，我们能够自行设置幻灯片主题样式的色调，使用 `\usecolortheme{}` 命令即可，这些色调包括 beetle、beaver、orchid、whale、dolphin 等。这里的色调又被称为颜色主题，它定义了幻灯片各部分的颜色搭配，设置特定的颜色主题后，我们能够得到不同的组合样式，具体可参考[网站提供的组合样式矩阵](#)

在beamer文档类型中使用CambridgeUS主题样式和dolphin色调制作一个简单的幻灯片。

```
\documentclass{beamer}
\usetheme{CambridgeUS}
\usecolortheme{dolphin}
```

```
\begin{document}
```

```
\begin{frame}{Example}
```

This is a simple example for the CambridgeUS theme with dolphin (color theme).

```
\end{frame}
```

```
\end{document}
```

4.3 字体主题

实际上，对于幻灯片的文本字体，我们可以调用字体样式对其进行调整。在 Beamer 中，字体样式被称为字体主题，它定义了幻灯片的字体搭配。具体使用方法是：在前导代码中要用到的命令为 `\usefonttheme{A}`，位置 A 填写的一般是字体类型，例如 serif。

我们知道：在常规文档中，可以使用各种字体对应的宏包达到调用字体的作用，使用规则为 `\usepackage{A}`，位置 A 填写的一般是字体类型，包括 serif、avant、bookman、chancery、charter、euler、helvet、mathtime、mathptm、mathptmx、newcent、palatino、pifont、utopia 等。

使用 beamer 文档类型创建一个简单的幻灯片，并在前导代码中申明使用 serif 对应的字体样式。

```
\documentclass{beamer}
\usefonttheme{serif}

\begin{document}

\begin{frame}

This is a simple example for using \alert{serif} font theme.

\end{frame}

\end{document}
```

使用 beamer 文档类型创建一个简单的幻灯片，并在前导代码中申明使用字体 palatino 对应的宏包。

```
\documentclass{beamer}
\usepackage{palatino}

\begin{document}

\begin{frame}

This is a simple example for using \alert{palatino} font.

\end{frame}

\end{document}
```

4.4 内部主题

内部主题定义了幻灯片展示区域的样式，如列表、定理等，内部主题不包括页眉、页脚、导航栏等部分。每一种主题样式都有默认的内部主题，更换内部主题需使用`\useinnertheme{A}`命令，位置A可供选择的内部主题包括circles、rectangles、rounded和inmargin。

在beamer文档类型中分别使用circles和inmargin两种内部主题制作幻灯片。

使用circles内部主题：

```
\documentclass{beamer}
\usetheme{CambridgeUS}
\usefonttheme{professionalfonts}
\useinnertheme{circles}

\begin{document}

\begin{frame}
\frametitle{Parent function}
\framesubtitle{A short list}

Please check out the following parent function list.
\begin{enumerate}
\item  $y=x$ 
\item  $y=|x|$ 
\item  $y=x^2$ 
\item  $y=x^3$ 
\item  $y=x^b$ 
\end{enumerate}

\end{frame}

\end{document}
```

使用inmargin内部主题：

```
\documentclass{beamer}
\usetheme{CambridgeUS}
\usefonttheme{professionalfonts}
\useinnertheme{inmargin}

\begin{document}

\begin{frame}
\frametitle{Parent function}
\framesubtitle{A short list}
```

Please check out the following parent function list.

```
\begin{enumerate}
\item  $y=x$ 
\item  $y=|x|$ 
\item  $y=x^2$ 
\item  $y=x^3$ 
\item  $y=x^b$ 
\end{enumerate}

\end{frame}

\end{document}
```

4.5 外部主题

外部主题定义了幻灯片的边框、页眉、页脚、导航栏等部分的样式。更换外部主题需使用`\useoutertheme{A}`，位置 A 可供选择的外部主题包括 `infolines`、`smoothbars`、`sidebar`、`split` 和 `tree`。

4.6 表格字体大小

在 Beamer 中制作表格，当我们想对表头或者表格内容文字大小进行调整时，可以使用在前导代码中申明使用 `caption` 宏包，即`\usepackage{caption}`，然后设置具体的字体大小即可，如`\captionsetup{font = scriptsize, labelfont = scriptsize}` 可以将表头和表格内容字体大小调整为 `scriptsize`。

使用`\begin{table}` `\end{table}`环境创建一个简单表格，并使用`caption`宏包将表头字体大小设置为`Large`、将表格内容字体大小设置为`large`。

其中，单就设置表头字体大小而言，除了使用`caption`宏包之外，还可以通过对幻灯片设置全局参数达到调整字体大小的效果，例如`\setbeamerfont{caption}{size = Large}`。

```
\documentclass{beamer}
\usepackage{booktabs}
\usepackage{caption}
\captionsetup{font = large, labelfont = Large}

\begin{document}

\begin{frame}

\begin{table}
```

```

\caption{A simple table.}
\begin{tabular}{l|ccc}
\toprule
& \textbf{header3} & \textbf{header4} & \textbf{header5} & \\
\midrule
\textbf{header1} & cell1 & cell2 & cell3 & \\
\midrule
\textbf{header2} & cell4 & cell5 & cell6 & \\
\bottomrule
\end{tabular}
\end{table}

\end{frame}

\end{document}

```

4.7 样式调整

在 Beamer 文档类型中，除了可以使用各种主题样式，另外也可以根据幻灯片组成部分，分别对侧边栏、导航栏以及 Logo 等进行调整。其中，侧边栏是由所选幻灯片主题样式自动生成的，主要用于显示幻灯片目录。有时为了显示幻灯片的层次，使用侧边栏进行目录索引。

```

使用 Berkeley 主题样式，并将侧边栏显示在右侧。
\documentclass{beamer}
\PassOptionsToPackage{right}{beamerouterthemesidebar}
\usetheme{Berkeley}
\usefonttheme{professionalfonts}
\begin{document}
\begin{frame}
\frametitle{Parent function}
\framesubtitle{A short list}
Please check out the following parent function list.
\begin{enumerate}
\item  $y=x$ 
\item  $y=|x|$ 
\item  $y=x^2$ 
\item  $y=x^3$ 
\item  $y=x^b$ 
\end{enumerate}
\end{frame}
\end{document}

```


很多时候我们会发现，在各类学术汇报中，幻灯片的首页通常会有主讲人所在的研究机构Logo。在Beamer文档类型中，有`\logo`和`\titlegraphic`两个命令可供使用，使用`\logo`命令添加的Logo会在每一页幻灯片中都显示，而使用`\titlegraphic`命令添加的Logo只出现在标题页。

使用`\logo`命令在幻灯片中添加Logo。

```
\documentclass{beamer}
\usefonttheme{professionalfonts}

\title{A Simple Beamer Example}
\author{Author's Name}
\institute{Author's Institute}

\logo{\includegraphics[width=2cm]{logopolito}}

\begin{document}
\begin{frame}
\titlepage
\end{frame}
\begin{frame}{Parent function}{A short list}
Please check out the following parent function list.
\begin{enumerate}
\item  $y=x$ 
\item  $y=|x|$ 
\item  $y=x^2$ 
\item  $y=x^3$ 
\item  $y=x^b$ 
\end{enumerate}
\end{frame}
\end{document}
```

使用`\titlegraphic`命令在幻灯片的标题页添加Logo。

```
\documentclass{beamer}
\usefonttheme{professionalfonts}

\title{A Simple Beamer Example}
\author{Author's Name}
\institute{Author's Institute}

\titlegraphic{\includegraphics[width=2cm]{logopolito}\hspace*{4.75cm}~
\includegraphics[width=2cm]{logopolito}}

\begin{document}
```

```

\begin{frame}
\titlepage
\end{frame}

\begin{frame}{Parent function}{A short list}
Please check out the following parent function list.
\begin{enumerate}
\item  $y=x$ 
\item  $y=|x|$ 
\item  $y=x^2$ 
\item  $y=x^3$ 
\item  $y=x^b$ 
\end{enumerate}
\end{frame}

\end{document}

```

5 插入程序源代码

使用 Beamer 制作幻灯片时，可以使用 verbatim 宏包中的 `\begin{verbatim}` `\end{verbatim}` 环境插入程序源代码，相应地，`\begin{frame}[fragile]` `\end{frame}` 环境中需要添加 `fragile` 选项，否则会导致编译报错。

```

在beamer文档类型中使用\begin{frame}[fragile] \end{frame}和\begin{verbatim}
\end{verbatim}环境插入几行简单的Python程序。
\documentclass{beamer}
\usefonttheme{professionalfonts}
\usepackage{verbatim}

\begin{document}

\begin{frame}[fragile]
\frametitle{Parent function}
\framesubtitle{A short list}

Please check out the following parent function list.
\begin{enumerate}
\item  $y=x$ 
\item  $y=|x|$ 
\item  $y=x^2$ 
\item  $y=x^3$ 

```

```

\item $y=x^{\{b\}}$
\end{enumerate}

\textbf{Python code:}

\begin{verbatim}
import numpy as np

b = 5
y = np.zeros(100)
for x in range(1, 101):
    y[x] = x ** b
\end{verbatim}

\end{frame}

\end{document}

```

除了verbatim宏包，还可以使用listings宏包中的`\begin{lstlistings}` `\end{lstlistings}`插入程序源代码。

在beamer文档类型中使用`\begin{frame}[fragile]` `\end{frame}`和`\begin{lstlistings}` `\end{lstlistings}`环境插入几行简单的Python程序。

```

\documentclass{beamer}
\usefonttheme{professionalfonts}
\usepackage{listings}

```

```

\begin{document}

```

```

\begin{frame}[fragile]
\frametitle{Parent function}
\framesubtitle{A short list}

```

Please check out the following parent function list.

```

\begin{enumerate}
\item $y=x$
\item $y=|x|$
\item $y=x^{\{2\}}$
\item $y=x^{\{3\}}$
\item $y=x^{\{b\}}$
\end{enumerate}

```

```

\textbf{Python code:}

```

```

\begin{lstlisting}

```

```
import numpy as np

b = 5
y = np.zeros(100)
for x in range(1, 101):
    y[x] = x ** b
\end{lstlisting}
\end{frame}
\end{document}
```

6 添加参考文献

通常来说，学术汇报的幻灯片时常需要添加与本研究相关的参考文献作为支撑材料。使用 LaTeX 制作常规文档时，最常用的文献管理工具是 Bibtex，但事实上，Beamer 并不支持编译 Bibtex。因此，拓展名为.bib 的文献管理文件在 Beamer 中是无法使用的，不过我们可以使用 `\begin{thebibliography} \end{thebibliography}` 环境添加参考文献。

在常规文档中，使用 L^AT_EX 创建参考文献的简单方式是使用 `\begin{thebibliography} \end{thebibliography}` 环境添加参考文献。有了参考文献的条目和标签，在正文中使用 `\cite{}` 命令进行引用便可让参考文献显示出来，这种做法在 Beamer 中也是适用的。只不过在添加参考文献时，我们需要对文献的类型进行指定，著作对应着 `\beamertemplatebookbibitems` 命令，而论文则对应着 `\beamertemplatearticlebibitems`。需要注意的是，为了避免文献数量过多而导致的参考文献页面排版问题，可以在 `\begin{frame}[allowframebreaks] \end{frame}` 环境中申明自动跨页。

```
使用beamer文档类型创建幻灯片并在\begin{thebibliography} \end{
    thebibliography}环境中创建参考文献。
\documentclass{beamer}
\usetheme{CambridgeUS}

\begin{document}

\begin{frame}
\frametitle{Reference Example}

If you have any interest in matrix computations, please referring to \cite{
    golub2013matrix, petersen2008the}.

\end{frame}

\begin{frame}
```

```

\frametitle<presentation>\{Further Reading\}

\begin{thebibliography}{10}

\beamertemplatebookbibitems
\bibitem{golub2013matrix}
  Gene H. Golub and Charles F. Van Loan.
  \newblock {\em Matrix computations}.
  \newblock JHU press, 2013.

\beamertemplatearticlebibitems
\bibitem{petersen2008the}
  Kaare Brandt Petersen, Michael Syskind Pedersen.
  \newblock The matrix cookbook.
  \newblock {\em Technical University of Denmark}, 510, 2008.

\end{thebibliography}
\end{frame}
\end{document}

```

使用beamer文档类型创建幻灯片并在`\begin{thebibliography}` `\end{thebibliography}`环境中创建参考文献，其中，需要对幻灯片申明自动跨页。

```

\documentclass{beamer}
\usetheme{CambridgeUS}

\begin{document}

\begin{frame}
\frametitle{Reference Example}

If you have any interest in matrix computations, please referring to \cite{
  golub2013matrix, petersen2008the}.

\end{frame}

\begin{frame}[allowframebreaks]
\frametitle<presentation>\{Further Reading\}

\begin{thebibliography}{10}

\beamertemplatebookbibitems
\bibitem{golub2013matrix}
  Gene H. Golub and Charles F. Van Loan.

```

```

\newblock {\em Matrix computations}.
\newblock JHU press , 2013.

\beamertemplatearticlebibitems
\bibitem{petersen2008the}
Kaare Brandt Petersen , Michael Syskind Pedersen .
\newblock The matrix cookbook .
\newblock {\em Technical University of Denmark}, 510, 2008.

\beamertemplatebookbibitems
\bibitem{golub2013matrix}
Gene H. Golub and Charles F. Van Loan .
\newblock {\em Matrix computations}.
\newblock JHU press , 2013.

\beamertemplatearticlebibitems
\bibitem{petersen2008the}
Kaare Brandt Petersen , Michael Syskind Pedersen .
\newblock The matrix cookbook .
\newblock {\em Technical University of Denmark}, 510, 2008.

\beamertemplatebookbibitems
\bibitem{golub2013matrix}
Gene H. Golub and Charles F. Van Loan .
\newblock {\em Matrix computations}.
\newblock JHU press , 2013.

\beamertemplatearticlebibitems
\bibitem{petersen2008the}
Kaare Brandt Petersen , Michael Syskind Pedersen .
\newblock The matrix cookbook .
\newblock {\em Technical University of Denmark}, 510, 2008.

\end{thebibliography}

\end{frame}

\end{document}

```

7 插入表格

在 Beamer 幻灯片中插入表格的规则与常规文档一致，可以使用 `tabular` 环境创建表格内容，或者在 `tabular` 环境外额外嵌套一层 `table` 环境以创建标题、索引标签等元素，当然对于合并单元格、创建三线表、创建彩色表格等操作也是类似的。

在 beamer 文档类型中使用 `table` 环境和 `tabular` 环境制作一个简单表格。

```
\documentclass{beamer}

\begin{document}

\begin{frame}

The created table\ref{tab:table_a} is shown as follows.
\begin{table}[bt]
\caption{A table in Beamer}
\label{tab:table_a}
\begin{tabular}{|l|c|c|} \hline
\textbf{Code Structure} & \textbf{Component} & \textbf{Others} \\ \hline
preamble & figures & title \\
body & tables & footline \\
& equations & list \\
& normal texts & block \\ \hline
\end{tabular}
\end{table}
\end{frame}
\end{document}
```

在 Beamer 中，我们可以使用前面介绍的动画效果命令（如 `\uncover{}`）为表格添加动画，让表格中的各单元格内容分步呈现。

在 beamer 文档类型中使用 `\begin{table}` `\end{table}` 环境制作一个简单表格，同时使用 `\uncover{}` 设置动画格式。

```
\documentclass{beamer}
\begin{document}
\begin{frame}
\begin{table}[bt]
\begin{tabular}{|l|c|c|} \hline
\textbf{Code Structure} & \textbf{Component} & \textbf{Others} \\ \hline
\uncover<1->{preamble} & \uncover<2->{figures} & \uncover<3->{title} \\
\uncover<1->{body} & \uncover<2->{tables} & \uncover<3->{footline} \\ \hline
\end{tabular}
\end{table}
\end{frame}
\end{document}
```

```

& \uncover<2->{equations} & \uncover<3->{list} \\
& \uncover<2->{normal texts} & \uncover<3->{block} \\
\hline
\end{tabular}
\end{table}
\end{frame}
\end{document}

```

8 插入与调整图片

在制作幻灯片时经常需要插入图片以辅助演讲汇报，对此，Beamer 提供了相关宏包及命令可以支持在幻灯片中插入图片并对其进行一系列操作，包括编辑图片、插入子图、调整排列方式、调整布局方式、设置背景图片、在标题页中插入图片、添加动画效果等。

8.1 插入图片

类似于 article 文档中插入图片操作，在 Beamer 中可以基于 `\begin{figure}` `\end{figure}` 环境、并使用 `\includegraphics[图片参数]{图片名或图片路径}` 命令为幻灯片插入图片、以及设置 `height`、`width`、`angle` 等参数。不同之处主要在于：

- Beamer 文档类自带 `graphicx` 工具包，可以省略声明语句 `\usepackage{graphicx}`；
- 在 Beamer 中浮动图形位置参数 `h`、`t`、`b`、`p` 无效，因此需要使用其它方式调整图片位置；
- 在 Beamer 中使用 `\caption{}` 命令仅创建图片标题，而不再为图片进行自动编号。为此，需要在导言区额外添加声明语句 `\setbeamertemplate{caption}[numbered]`，表示对演示稿中的图片进行自动编号。

在 beamer 文档类型中使用 `\begin{figure}` `\end{figure}` 环境以及 `\includegraphics` 命令插入图片，并添加声明 `\setbeamertemplate{caption}[numbered]` 使得图片进行自动编号。

```

\documentclass{beamer}
% 使图片进行自动编号
\setbeamertemplate{caption}[numbered]

\begin{document}

\begin{frame}{Two pictures}

```



```

\begin{figure}
\centering
% 插入第一张图片并设置标题、索引标签
\includegraphics[width=0.2\linewidth]{blueflower.png}
\caption{First figure.}
\label{fig:figlabel1}
% 插入第二张图片并设置标题、索引标签
\includegraphics[width=0.2\linewidth]{yellowFlower.png}
\caption{Second figure.}
\label{fig:figlabel2}
\end{figure}
\end{frame}

\end{document}

```

8.2 编辑图片

读者可以通过修改 Beamer 主题选项从而为演示稿定制图片标题样式，包括：

- `\setbeamerfont{caption}{size= 定制的字体大小}`：设置图片标题的字体大小，如设置为 `\small`、`\large`、`\Large` 等；
- `\setbeamercolor{caption}{fg= 定制的颜色}`：设置图片标题编号的颜色，如设置为 `red`、`blue` 等；
- `\setbeamercolor{caption name}{fg= 定制的颜色}`：设置图片标题内容的颜色，如设置为 `red`、`blue` 等。

读者还可以改变图片的不透明度。为此，首先需要调用 `tikz` 宏包，使用语句 `\usepackage{tikz}` 即可；然后将插入的图片作为一个节点 `\node` 放置到 `tikzpicture` 环境内，并使用 `\node[opacity= 定制的不透明度]` 语句设置不透明度选项参数，如下所示：

```

\begin{tikzpicture}
  \node[opacity=定制的不透明度]{\includegraphics[width=0.3\textwidth]{图
    片路径}};
\end{tikzpicture}

```

```

在beamer文档类型中使用\setbeamerfont{caption}、\setbeamercolor{caption}和\
setbeamercolor{caption name}命令修改图片标题样式；并调用tikz宏包，使用
tikzpicture环境和\node命令设置图片不透明度。
\documentclass{beamer}

```

```

\usepackage{tikz}
\setbeamertemplate{caption}[numbered]
% 定制图片标题样式
\setbeamerfont{caption}{size=\large}
\setbeamercolor{caption name}{fg=red}
\setbeamercolor{caption}{fg=blue}

\begin{document}

\begin{frame}{Two pictures}

\begin{figure}
\centering
% 插入第一张图片并设置标题、创建索引标签
\begin{tikzpicture}
\node[opacity=0.3]
{\includegraphics[width=0.2\linewidth]{blueflower.png}};
\end{tikzpicture}
\caption{First figure.}
\label{fig:figlabel1}
% 插入第二张图片并设置图片不透明度、设置标题、创建索引标签
\begin{tikzpicture}
\node[opacity=0.3]
{\includegraphics[width=0.2\linewidth]{yellowflower.png}};
\end{tikzpicture}
\caption{Second figure.}
\label{fig:figlabel2}
\end{figure}

\end{frame}

\end{document}

```

8.3 插入子图

有时需要在 Beamer 中插入子图，用到的宏包与命令与 article 文档情况类似。读者可以只需要使用语句 `\usepackage{subcaption}` 调用 subcaption 宏包，并在 figure 环境中创建多个 subfigure 环境，每个 subfigure 环境内都可以进行插入子图、设置子图标题和标签等操作。下面来看一个具体的例子：

```

在beamer文档类型中，调用subcaption宏包，并使用figure环境中的subfigure环境创建子图。
\documentclass{beamer}

```

```

% 调用关键宏包 subcaption
\usepackage{subcaption}
\setbeamertemplate{caption}[numbered]

\begin{document}

\begin{frame}{Two sub-figures}
\begin{figure}
\centering
% 插入第一张子图
\begin{subfigure}{0.4\linewidth}
\centering
\includegraphics[width=0.5\linewidth]{blueflower.png}
\caption{First subfigure.}
\label{fig:subfiglabel1}
\end{subfigure}
% 插入第二张子图
\begin{subfigure}{0.4\linewidth}
\centering
\includegraphics[width=0.5\linewidth]{yellowFlower.png}
\caption{Second subfigure.}
\label{fig:subfiglabel2}
\end{subfigure}
\caption{A figure contains two sub-figures.}
\label{fig:figlabel}
\end{figure}
\end{frame}
\end{document}

```

8.4 调整排列方式

使用`\centering`命令可以将图片放置到整个幻灯片页面的中央，但有时需要对图片的排列位置进行更多样化的调整。

8.4.1 横向分布与纵向分布

使用`\hfill`、`\vfill`命令可以分别控制图片的水平间距和垂直间距。在同一行中，每个`\hfill`产生的间隔是相同的，从而达到将多个图片横向等间距分布的效果；类似地，`\vfill`命令可以实现纵向等间距分布的效果。

在beamer文档类型中，使用`\hfill`命令实现图片横向分布，以及使用`\vfill`命令实现图片纵向分布。

```

\documentclass{beamer}

```

```

\begin{document}
% 实现图片横向分布
\begin{frame}{Three pictures with the same lateral distance}
\begin{figure}
\includegraphics[width=0.2\linewidth]{redflower.png}
\hfill
\includegraphics[width=0.2\linewidth]{blueflower.png}
\hfill
\includegraphics[width=0.2\linewidth]{yellowflower.png}
\end{figure}
\end{frame}
\begin{frame}{Three pictures with the same longitudinal distance}
% 实现图片纵向分布
\begin{figure}
\includegraphics[width=0.2\linewidth]{redflower.png}
\vfill
\includegraphics[width=0.2\linewidth]{blueflower.png}
\vfill
\includegraphics[width=0.2\linewidth]{yellowflower.png}
\end{figure}
\end{frame}

\end{document}

```

8.4.2 并排显示

在幻灯片中，将不同内容（图片、文本、表格、公式等）并排分栏显示是一种常用的展示方式。前面我们介绍了使用`\hfill`命令实现多个图片并排显示的方式，这里我们接着介绍一种更通用的方式，不仅可以实现图片的并排显示，也可以实现文本等其它内容的并排显示。为此，需要使用`\begin{columns}` `\end{columns}`环境，并嵌套多个`\begin{column}` `\end{column}`环境对内容进行分栏，每个`column`环境中的内容都独立成栏，不同栏的内容并排显示。

在beamer文档类型中，使用`columns`环境创建多列内容实现并排显示效果。

```

\documentclass{beamer}

\begin{document}
\begin{frame}{Frame1}

\begin{columns}
% 创建第一列

```

```

\begin{column}{0.4\linewidth}
    Here is a description for the blue flower on the right. Here is a
    description for the blue flower on the right.
\end{column}
% 创建第二列
\begin{column}{0.4\linewidth}
    \begin{figure}
        \centering
        \includegraphics[width=0.7\linewidth]{blueflower.png}
        \caption{A blue flower.}
    \end{figure}
\end{column}
\end{columns}

\end{frame}

\end{document}

```

8.4.3 设置任意图片位置

通过调用 tikz 宏包及其命令可以为图片指定任意摆放位置。首先使用 `\usepackage{tikz}` 语句调用 tikz 宏包，然后调用如下所示的语句：

```

\begin{tikzpicture}[remember picture, overlay]
    \node[锚点位置偏移量] at (current page.锚点位置)
    {
        \includegraphics{图片路径}
    };
\end{tikzpicture}

```

上述示例语句中，包括：

- 在幻灯片中创建 tikzpicture 环境，使用 `\node` 命令创建一个内容为插入图片的节点；
- 设置 `\begin{tikzpicture}` 的选项为 `remember picture, overlay`，从而允许自由放置图片；
- 通过使用 `\node[] at ()` 语句指定图片摆放位置：
 - 1) `()` 中用于设置图片的锚点位置或摆放位置，设置格式为 `current page. 锚点位置`。在锚点位置设置上：一种方式是使用描述性语言，如 `current page.center`

表示将图片放在页面正中央、current page.east 表示将图片放在页面边缘的右方（正东方向）、current page.north east 表示将图片放在页面边缘的右上方（东北方向），以此类推；另一种方式是使用相对于页面正东方向的逆时针角度，如 current page.0 表示将图片放在页面边缘右方、current page.45 表示将图片放在页面边缘右上方、current page.-90 表示将图片放在页面边缘下方；

- 2) [] 中用于设置锚点位置的偏移量。一种方式是使用描述性语言，包括 left、right、above、below，分别表示将图片相对于锚点位置往左、右、上、下偏移，如 \node[left=1cm,above=2cm] at (current page.center) 表示将图片位置设置为页面正中央往左偏移 1cm、往上偏移 2cm 的位置；另一种方式是使用偏移坐标参数 xshift 和 yshift，如 \node[xshift=-1cm,yshift=2cm] at (current page.center) 的效果等价于 \node[left=1cm,above=2cm] at (current page.center)。

在 beamer 文档类型中，使用 tikzpicture 环境创建以插入的图片作为内容的节点，为 \begin{tikzpicture} 设置选项 remember picture, overlay, 并使用 \node[] at () 语句设置图片的摆放位置。

```
\documentclass{beamer}
% 调用 tikz 宏包
\usepackage{tikz}

\begin{document}
% 第一页幻灯片
\begin{frame}{Frame1}
% 创建 tikzpicture 环境，并使用描述性语言设置锚点位置
\begin{tikzpicture}[remember picture, overlay]
    % 将图片放置到幻灯片页中央
    \node at (current page.center)
    {
        \includegraphics[width=0.2\linewidth]{blueflower.png}
    };
    % 将图片放置到幻灯片页的右边（正东方向）
    \node at (current page.east)
    {
        \includegraphics[width=0.2\linewidth]{blueflower.png}
    };
    % 将图片放置到幻灯片页的下方（正南方向）
    \node at (current page.south)
    {
        \includegraphics[width=0.2\linewidth]{blueflower.png}
    };
    % 将图片放置到幻灯片页的左边（正西方向）
```

```

\node at (current page.west)
{
    \includegraphics [width=0.2\linewidth]{blueflower.png}
};
% 将图片放置到幻灯片页的上方（正北方向）
\node at (current page.north)
{
    \includegraphics [width=0.2\linewidth]{blueflower.png}
};
\end{tikzpicture}
\end{frame}
% 第二页幻灯片
\begin{frame}{Frame2}
% 创建tikzpicture环境，并使用角度设置锚点位置、使用描述性语言设置锚点偏移量
\begin{tikzpicture}[remember picture, overlay]
    % 将图片放置到幻灯片页的右边，并往左偏移0.5cm
    \node[left=0.5cm] at (current page.0)
    {
        \includegraphics [width=0.2\linewidth]{blueflower.png}
    };
    % 将图片放置到幻灯片页的下方，并往上偏移0.5cm
    \node[below=0.5cm] at (current page.-90)
    {
        \includegraphics [width=0.2\linewidth]{blueflower.png}
    };
    % 将图片放置到幻灯片页的左边，并往右偏移0.5cm
    \node[right=0.5cm] at (current page.180)
    {
        \includegraphics [width=0.2\linewidth]{blueflower.png}
    };
    % 将图片放置到幻灯片页的上方，并往下偏移0.5cm
    \node[above=0.5cm] at (current page.90)
    {
        \includegraphics [width=0.2\linewidth]{blueflower.png}
    };
\end{tikzpicture}
\end{frame}

\end{document}

```

8.5 调整布局方式

在幻灯片中，常见的图文布局方式有两种：文字环绕于图片、文字浮于图片上方。下面分别展开介绍。

8.5.1 文字环绕图片

为了实现文字环绕图片的显示效果，首先需要使用`\usepackage{wrapfig}` 语句调用 wrapfig 宏包，并创建`\begin{wrapfigure}` `\end{wrapfigure}` 环境插入图片，在 wrapfigure 环境后面输入环绕文本内容，由此创建的文本内容将环绕于插入的图片周围。其中，wrapfigure 环境与我们熟悉的 figure 环境类似，同样可以使用`\includegraphics`、`\caption` 等语句实现相关功能。在`\begin{wrapfigure}{}` 语句的 `{}` 中可以设置图片位置参数，参数选项为 `r` 或 `l`，分别表示将图片设置为页面右侧或左侧。

在beamer文档类型中，使用 wrapfigure 环境插入图片并设置图片位置为页面右侧，实现文字环绕图片的效果。

```
\documentclass{beamer}
% 调用 wrapfig 宏包
\usepackage{wrapfig}

\begin{document}
\begin{frame}{Frame1}
% 创建 wrapfigure 环境并插入图片
\begin{wrapfigure}{r}{0.5\linewidth}
  \centering
  \includegraphics[width=0.5\linewidth]{blueflower.png}
  \caption{A blue flower.}
\end{wrapfigure}
% 示意性环绕文本
Here is a description for the picture. Here is a description for the
picture. Here is a description for the picture. Here is a description
for the picture. Here is a description for the picture. Here is a
description for the picture. Here is a description for the picture. Here
is a description for the picture. Here is a description for the picture
. Here is a description for the picture. Here is a description for the
picture. Here is a description for the picture.
\end{frame}

\end{document}
```


8.5.2 文字浮于图片上方

图片浮于文字上方是另一种常用的图文布局方式。为此，读者需要使用 `\usepackage{tikz}` 语句调用 tikz 宏包，在 tikzpicture 环境中使用 `\node` 命令创建一个内容为插入图片的节点、一个内容为文本的节点，并将文本节点的位置设为图片节点的中央。

在 beamer 文档类型中，使用 tikzpicture 环境创建一个内容为插入图片的节点、一个内容为文本的节点，并设置第一个节点的名称和图片透明度、第二个节点的文本样式和位置。

```
\documentclass{beamer}
% 调用 tikz 宏包
\usepackage{tikz}

\begin{document}
\begin{frame}{Frame1}

\begin{tikzpicture}
% 创建一个内容为图片的节点，节点名为 myfirstfigure
\node(myfirstfigure)[opacity=0.4]
{
    \includegraphics[width=0.3\linewidth]{greyflower.png}
    \includegraphics[width=0.3\linewidth]{greyflower.png}
    \includegraphics[width=0.3\linewidth]{greyflower.png}
};
% 创建一个内容为文本的节点
\node
[
    text=teal,
    font={\huge\bfseries}
] at (myfirstfigure.center) {Text over the picture!};
\end{tikzpicture}

\end{frame}

\end{document}
```

8.6 设置背景图片

在 Beamer 中可以很方便地为演示稿设置全局或局部的背景图片，通过使用 `\setbeamertemplate{background canvas}{插入背景图片}` 语句设置背景画布选项即可。

在beamer文档类型中，`\setbeamertemplate{background canvas}`{插入背景图片}语句设置全局及局部背景画布选项。

```
\documentclass{beamer}
\usetheme{AnnArbor}

% 设置全局背景画布选项
\setbeamertemplate{background canvas}
{
  \includegraphics[width=\paperwidth,height=\paperheight]{lines.png}%
}

\begin{document}
% 第一页幻灯片使用全局背景画布
\begin{frame}{Frame1}
\end{frame}
{
% 设置局部特定背景画布选项
\setbeamertemplate{background canvas}
{
  \includegraphics[width=\paperwidth,height=\paperheight]{AI.png}%
}
% 第二页幻灯片使用局部特定背景画布
\begin{frame}{Frame2}
\end{frame}
}
% 第三页幻灯片使用全局背景画布
\begin{frame}{Frame3}
\end{frame}
% 第四页幻灯片使用全局背景画布
\begin{frame}{Frame4}
\end{frame}
\end{document}
```

如上例所示，为了让插入的图片充满整个页面，应使用 `width=\paperwidth` 和 `height=\paperheight` 选项；修改局部特定背景画布时，应用 `{}` 符号将 `\setbeamertemplate` 语句及其作用的局部 `frame` 环境放在一起。

8.7 在标题页中插入图片

在前面的章节中我们介绍了在 Beamer 中使用 `\titlepage` 命令创建标题页的语句。事实上，除了可以在标题页中添加标题名称、作者、机构单位等信息之外，在标题页中添加图片或图标也是一种常见的操作。为此，只需要在标题页信息中添加 `\titlegraphic{插`

入图片}命令即可，如下例所示：

在beamer文档类型中，使用\titlegraphic{插入图片}命令为标题页插入图片，并使用\titlepage命令创建标题页。

```
\documentclass{beamer}
\usetheme{AnnArbor}

\begin{document}

\title{A presentation about AI}
\author{The author's name}
\institute{The institute's name}
\date{2021/1/1}
% 在标题页中添加图片
\titlegraphic
{
    \includegraphics[width=2cm]{robot.png}
}
\begin{frame}
    \titlepage
\end{frame}

\end{document}
```

在beamer文档类型中，使用\titlegraphic{插入图片}命令为标题页插入图片，并在tikzpicture环境中使用\node命令调整图片位置为右下方。

```
\documentclass{beamer}
% 调用tikz宏包
\usepackage{tikz}
\usetheme{AnnArbor}

\begin{document}

\title{A presentation about AI}
\author{The author's name}
\institute{The institute's name}
\date{2021-8-30}
% 在标题页中添加图片并指定位置
\titlegraphic
{
    \begin{tikzpicture}[overlay, remember picture]
        \node[above=0.8cm] at (current page.-43){
            \includegraphics[width=1.5cm]{robot.png}
        };
    \end{tikzpicture}
}
```

```
    \end{tikzpicture}
}
\begin{frame}
    \titlepage
\end{frame}

\end{document}
```

8.8 添加动画效果

如果想要为图片添加动画效果，使得不同的图片分步显示，使用设置了显示范围选项的`\includegraphics< 显示范围 >{图片路径}`命令插入图片即可。

在beamer文档类型中使用`\includegraphics<>{}`命令插入多张图片并分步显示。

```
\documentclass{beamer}

\begin{document}

\begin{frame}

\includegraphics<1->[width=0.2\linewidth]{redflower.png}
\includegraphics<2->[width=0.2\linewidth]{yellowflower.png}
\includegraphics<3->[width=0.2\linewidth]{blueflower.png}

\end{frame}

\end{document}
```

Part IX

LaTeX 进阶

LaTeX 作为一款文档排版系统，拥有众多文档类型，可用于制作科技论文、技术报告以及幻灯片等，除此之外，LaTeX 还拥有包括编辑数学公式、制作图形与表格等功能，LaTeX 用户可根据自身需要解锁 LaTeX 的各种用途。

添加程序源代码和算法伪代码对于科研报告往往是必要且有效的，因为代码可以展现计算机编程的思路和算法，可以供读者学习和使用。所以，能够学会添加简洁优美、整齐大方的源代码和伪代码是科研工作者的一项重要技能。对于科研工作者，在有些学术交流中，有一种非常重要的展现成果方式就是海报。LaTeX 可以制作出优美简洁的海报，有很多模版可以方便制作者使用。另外，简历制作也是科研工作者经常需要用到的，同样 LaTeX 提供了很多好用的模版，可以方便其使用。

1 添加程序源代码

很多时候，在技术文档中添加程序源代码具有一定的必要性，这源于：

- 在很多文档（如实验报告）中，程序源代码往往作为重要组成部分，必须作为辅助材料放在文档末尾的附录中。
- 程序源代码既可以直接展现计算机编程的实现过程和细节，又可以评估实验的真实性，同时也能供读者学习和使用。

事实上，使用 LaTeX 制作文档时，添加程序源代码是一件看似简单、但又比较考验技巧的事，因为在文档中添加程序源代码并不能通过简单的“复制 + 粘贴”来实现。我们需要保持代码在原来程序语言中的格式，包括代码所采用的高亮颜色和等宽字体，目的都是为了让代码本来的面貌得以完美展现。

在 LaTeX 中，有很多宏包可供制作文档时添加程序源代码到正文或附录中，最常用的宏包包括 listings 和 minted 这两种，除此之外，还有一种插入程序源代码非常简便的一种方式，即使用 `\begin{verbatim}` `\end{verbatim}` 环境。

1.1 使用 verbatim 插入程序源代码

在 LaTeX 中插入 Python 代码可以使用 verbatim 环境，即在 `\begin{verbatim}` `\end{verbatim}` 之间插入代码，代码的文本是等宽字体。需要注意的是，这一环境不会对程序源代码进行高亮处理。

Python code example:

```
import numpy as np
x=np.random.rand(4)
print(x)
```

使用 verbatim 环境插入如下 Python 代码：

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

```
\begin{document}
```

Python code example:

```
\begin{verbatim}
```

```
import numpy as np
```

```
x = np.random.rand(4)
```

```
print(x)
```

```
\end{verbatim}
```

```
\end{document}
```

1.2 使用 listings 插入程序源代码

如果想要对程序源代码进行高亮处理，可以使用专门排版代码的工具包 listings，除了在前导代码中申明使用 listings 工具包，即`\usepackage{listings}`，有时候还可以根据需要自定义一些参数。

Python code example:

```
import numpy as np
```

```
x = np.random.rand(4)
```

```
print(x)
```

使用 listings 工具包插入 Python 代码，并自定义代码高亮。

```
\documentclass[12pt]{article}
\usepackage{listings}
\usepackage{color}
\definecolor{codegreen}{rgb}{0,0.6,0}
\definecolor{codegray}{rgb}{0.5,0.5,0.5}
\definecolor{codepurple}{rgb}{0.58,0,0.82}
\definecolor{backcolour}{rgb}{0.95,0.95,0.92}

\lstdefinestyle{mystyle}{
  backgroundcolor=\color{backcolour},
  commentstyle=\color{codegreen},
  keywordstyle=\color{magenta},
  numberstyle=\tiny\color{codegray},
  stringstyle=\color{codepurple},
  basicstyle=\ttfamily\footnotesize,
  breakatwhitespace=false,
  breaklines=true,
  captionpos=b,
  keepspaces=true,
  numbers=left,
  numbersep=5pt,
  showspace=false,
  showstringspaces=false,
  showtabs=false,
  tabsize=2
}

\lstset{style=mystyle}

\begin{document}

Python code example:

\begin{lstlisting}[language = python]
import numpy as np

x = np.random.rand(4)
print(x)

```

2 算法伪代码

算法这个词的英文是 algorithm，它几乎贯穿了整个计算机的各个领域。算法伪代码作为自然语言与类编程语言组成的混合结构，它在描述算法结构和思路方面要比纯编程语言更简洁且可读性更好、相比自然语言则更准确。同时，我们也能很容易地将算法伪代码转换成计算机程序。因此，在计算机相关的技术文档或文献中，适当使用算法伪代码解释技术架构会更方便读者理解。

在 LaTeX 中，为了便于创建算法伪代码，现有很多相关的宏包，例如 algorithm 和 algorithmic，在前导代码中申明使用这些宏包便可使用相应的算法伪代码环境。宏包 algorithm 提供的算法伪代码环境为 `\begin{algorithm} \end{algorithm}` 和 `\begin{algorithmic} \end{algorithmic}`。

This is a simple example:

Algorithm 1 Inner product of vectors

Input: $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$

Output: c

```
1:  $c = 0$ 
2: for  $i = 1$  to  $n$  do
3:    $c = c + x_i y_i$ 
4: end for
```

在例子中，语句 `\renewcommand{\algorithmicrequire}{\textbf{Input:}}` 表示将算法伪代码中的关键词 require 替换成 Input，同理，我们也能将关键词 ensure 替换成 Output。

除了 algorithm 和 algorithmic 这两个专门用于创建算法伪代码的宏包，还有一个非常常用的宏包，叫 algorithm2e，它与 algorithm 宏包创建出来的算法伪代码在样式上略有不同，algorithm2e 也提供了一种 `\begin{algorithm} \end{algorithm}` 环境。

在例子中，申明使用宏包 algorithm2e 时将参数设置为 linesnumbered 和 boxed，这两个参数分别表示对算法伪代码各行进行编号和对算法伪代码区域加边框，作为全局参数，会成为算法伪代码中的默认设置。

使用 algorithm 宏包中相应的环境创建一个简单的算法伪代码。

```
\documentclass[12pt]{article}
\usepackage[linesnumbered, boxed]{algorithm2e}
\usepackage{amsmath, amsfonts}

\begin{document}

This is a simple example:

\IncMargin{1em}
\begin{algorithm}
```



```

\SetKwInOut{Input}{Input}
\SetKwInOut{Output}{Output}
\caption{Inner product of vectors}
\Input{$\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$}
\Output{$c$}
$c=0$;
\For{$i=1$ \KwTo $n$}{
$c=c+x_i y_i$;
}
\end{algorithm}

\end{document}

```

3 简历制作

LaTeX 可制作各类文档，其中也不乏简历，与其他各类文档相比，简历在制作的过程中注重内容的简洁与清晰。使用 LaTeX 制作简历有诸多优势：第一，无需考虑字体、颜色、排版等问题；第二，LaTeX 拥有众多简历模板可供选择，易于切换简历的排版风格。

3.1 使用 article 文档类型制作简历

article 文档类型是 LaTeX 中极为常用的一种文档类型，使用 article 文档类型制作简历时，可将文档的结构命令如 `\section{}`、`\subsection{}`、`\subsubsection{}` 等格式稍作调整。

3.2 自定义简历格式

在 LaTeX 文件中编写 `documentclassarticle` 时，包括了类文件 `article.cls`。该类文件定义了组织文档的所有命令，比如片段和标题，它还配置这些命令如何影响页面的格式和布局。使用 LaTeX 制作简历时，我们需要自定义文档格式。其中最简洁的方法是将所有信息保存在个人类文件中，这样可以使文档的结构与格式完全分离，从而便于使用。因此，我们需要创建一个简历的类文件，例如 `CV.cls`，然后在类文件内定义文档格式。

所有类文件都应该以下面两行代码开头，应该将它们添加到 `CV.cls` 的顶部。

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesClass{CV}[2021/08/29 My custom CV class]

```

`\NeedsTeXFormat{LaTeX2e}` 命令告诉编译器该包适用于哪个版本的 L^AT_EX, L^AT_EX 的当前版本是 L^AT_EX2e。`\ProvidesClass{CV}[2021/08/29 My custom CV class]` 第一个参数应该匹配类文件的文件名, 并告诉 LaTeX 包的名称。第二个参数是可改变的, 它提供了类的描述, 这些描述将出现在日志和其他地方。

随后, 我们创建一个编译文件 `CV.tex`, 并将以下代码键入文件, 填写简历中的个人信息。

```
\documentclass{CV}

\begin{document}

\section{Research Interests}
\subsection{Machine Learning}

\section{Education}
\subsection{University of Nowhere}

\end{document}
```

标准的文章标题并不适合简历, 所以我们希望用更整洁的格式取代它们。为此, 我们可以在 `CV.cls` 文件中重新定义 `section` 命令以输出自定义格式。在这里我们需要使用 `titlesec` 宏包, 调用命令为 `\RequirePackage{titlesec}`, 随后, 我们便可以自定义标题格式。在 `CV.cls` 文件中键入以下代码:

```
\RequirePackage{titlesec}

\titleformat{\section}
  {\bfseries\Large\scshape\raggedright}
  {}{0em}
  {}
  [\titlerule]

\titleformat{\subsection}
  {\large\scshape\raggedright}
  {}{0em}
  {}
```

自定义标题格式可以使用以下命令:

- `\bfseries`, `\itshape`: 标题加粗或加斜体;
- `\scshape`: 小型资本;

- `\small`, `\normalsize`, `\large`, `\Large`, `\LARGE`, `\huge`, `\Huge`: 设定字型大小;
- `\rmfamily`, `\sffamily`, `\ttfamily`: 将字体类型分别设置为有衬线字体、无衬线字体或打字机字体。

简历的部分章节需要添加日期，我们在 CV.cls 文件中定义一个新命令 `\datedsubsection`，命令可以让我们在子章节标题中添加日期，新命令代码为：

```
\newcommand{\datedsubsection}[2]{%
\subsection[#1]{#1 \hfill #2}%
}
```

键入以上代码后，在 CV.tex 文件中更改部分代码，使用新定义命令：

```
\documentclass{CV}

\begin{document}

\section{Research Interests}
\subsection{Machine Learning}

\section{Education}
\datedsubsection{University of Nowhere}{2012---2016} %使用新定义命令

\end{document}
```

在简历中，名字通常在最上面，并且包含相关的联系方式，同样地我们在 CV.cls 文件中定义一个新命令 `\name` 来添加名字，定义另一个新命令 `\contact` 来添加联系方式。

```
\newcommand{\name}[1]{%
\centerline{\Huge{#1}}
}

\newcommand\contact[5]{%
\centerline{%
\makebox[0pt][c]{%
#1 {\large\textperiodcentered} #2 {\large\textperiodcentered}
#3
\ #4 \ \ #5%
}%
}%
}
```

键入以上代码后，在 CV.tex 文件中使用新定义命令\name 和\contact：

```
\documentclass{CV}

\begin{document}

\name{John Kim}

\contact{123 Broadway}{London}{UK 12345}{john@kim.com}{(000)-111-1111}

\section{Research Interests}
\subsection{Machine Learning}
My research interest is machine learning.

\section{Education}
\datedsubsection{University of Nowhere}{2012---2016}
I attended the University of Nowhere from 2012 to 2016.

\end{document}
```

当然也可以自定义一些列表：

```
\newcommand{\researchitems}[3]{%
  \begin{itemize}
    \item #1
    \item #2
    \item #3
  \end{itemize}%
}
```

键入以上代码后，在CV.tex文件中使用新定义命令\researchitems：

```
\documentclass{CV}

\begin{document}

\name{John Kim}

\contact{123 Broadway}{London}{UK 12345}{john@kim.com}{(000)-111-1111}

\section{Research Interests}
\subsection{Machine Learning}
My research interest is machine learning.
\researchitems
{Logistic regression}
```

```
{Neural Networks}
{SVM}

\section{Education}
\datedsubsection{University of Nowhere}{2012---2016}
I attended the University of Nowhere from 2012 to 2016.

\end{document}
```