

# L<sup>A</sup>T<sub>E</sub>X 的基础操作汇总

*wyl*

2020 年 11 月 18 日



# 目录

<b>第一章 基本结构</b>	<b>7</b>
一、 导言区 . . . . .	7
二、 文稿区 . . . . .	7
三、 总结代码 . . . . .	7
<b>第二章 设置中文环境</b>	<b>9</b>
一、 ctex 宏包的运用 . . . . .	9
二、 总结代码 . . . . .	9
<b>第三章 字体设置</b>	<b>11</b>
一、 字体属性 . . . . .	11
(一) 字体族 . . . . .	11
(二) 字体系列 . . . . .	11
(三) 字体形状 . . . . .	11
(四) 中文字体 . . . . .	11
(五) 字体大小 . . . . .	12
二、 进阶运用 . . . . .	13
<b>第四章 篇章结构-基础</b>	<b>15</b>
一、 板块分割 . . . . .	15
二、 目录格式设置 . . . . .	15
<b>第五章 篇章结构-进阶</b>	<b>17</b>
一、 文本设置 . . . . .	17
(一) 纸张大小设置 . . . . .	17
(二) 文本区域大小设置 . . . . .	17
(三) 文本区域文字分布 . . . . .	17
二、 边框设置 . . . . .	17
(一) 页眉 & 页脚的设置 . . . . .	17
(二) 页码的使用 . . . . .	18
三、 总结代码 . . . . .	18

<b>第六章 特殊符号</b>	<b>19</b>
一、空白符号 . . . . .	19
二、L <sup>A</sup> T <sub>E</sub> X 控制符 . . . . .	19
三、排版符号 . . . . .	20
四、T <sub>E</sub> X 标志符号 . . . . .	20
五、其他符号 . . . . .	20
<b>第七章 插图</b>	<b>21</b>
一、插入图片 . . . . .	21
二、编辑图片 . . . . .	21
<b>第八章 表格</b>	<b>23</b>
一、表格内部参数 . . . . .	23
二、总结代码 . . . . .	23
<b>第九章 浮动体</b>	<b>25</b>
一、表格 . . . . .	25
二、插图 . . . . .	26
<b>第十章 数学公式初步应用</b>	<b>27</b>
一、行内公式 . . . . .	27
(一) 美元符号 . . . . .	27
(二) 小括号 . . . . .	27
(三) math 环境 . . . . .	27
二、上下标 . . . . .	27
(一) 上标 . . . . .	27
(二) 下标 . . . . .	28
(三) 综合运用 . . . . .	28
三、希腊字母 . . . . .	28
四、数学函数 . . . . .	28
五、分式 . . . . .	29
六、行间公式 . . . . .	30
(一) 双美元符号 . . . . .	30
(二) 中括号 . . . . .	30
(三) displaymath 环境 . . . . .	31
(四) 自动编号公式 equation 环境 . . . . .	31
<b>第十一章 数学矩阵的应用</b>	<b>33</b>
一、通用矩阵 . . . . .	33
(一) 无边框 . . . . .	33
(二) 小括号 . . . . .	33
(三) 中括号 . . . . .	33
(四) 大括号 . . . . .	34

(五) 行列式 . . . . .	34
(六) 双竖线 . . . . .	34
二、 矩阵与数学符号结合 . . . . .	34
三、 省略号 . . . . .	35
(一) 常用省略号 . . . . .	35
(二) 跨行省略号 . . . . .	35
四、 分块矩阵 . . . . .	36
五、 行内小矩阵 . . . . .	36
<b>第十二章 数学多行公式</b>	<b>37</b>
一、 gather 和 gather* 环境 . . . . .	37
(一) gather 环境 . . . . .	37
(二) gather* 环境 . . . . .	37
(三) \notag 的使用 . . . . .	38
二、 align 和 align* 环境 . . . . .	38
(一) align 环境 . . . . .	38
(二) align* 环境 . . . . .	38
三、 split 环境 . . . . .	39
四、 cases 环境 . . . . .	39
<b>第十三章 参考文献 Bibtex</b>	<b>41</b>



# 第一章 基本结构

这里会讲解到 L<sup>A</sup>T<sub>E</sub>X 中最基础的两个区域，分别是导言区和文稿区

## 一、导言区

首先是导言区，导言区是 L<sup>A</sup>T<sub>E</sub>X 中用来引入宏包（更多其他辅助工具）以及设定变量的地方，用来控制 L<sup>A</sup>T<sub>E</sub>X 的排版，参数数值等；

其中，导言区的编辑范围为：从文章的第一行到文稿区之前，即在文稿区之前的全部范围都是导言区  
导言区的第一行代码通常是：

`\documentclass{xxxx}` 这里 xxxx 可以是：article, book, report 等，不同的内容代表不同的格式

`\title{xxxx}` 这里的 xxxx 就是标题的名称

`\author {xxxx}` 这里的 xxxx 就是作者名称；

一般最简单的文稿就只需要这三行代码即可；

## 二、文稿区

其次是文稿区，文稿区是 L<sup>A</sup>T<sub>E</sub>X 里面用来编辑文件内容的地方，也就写东西的地方；

在开始写文章之前，我们要先构建一个编辑环境来为编辑做准备（注：一个 L<sup>A</sup>T<sub>E</sub>X 文件中有且只能有一个编辑环境），具体操作如下：

`\begin{document}` 这里是开始编辑环境

`\maketitle` 这里不是必须的，但是通过这一行的代码我们可以把之前在导言区编辑的标题和作者加入到文稿中（会自动补上时间）

xxxx 这里 xxxx 就是指我们文章的内容了

`\end{document}` 这里是结束编辑环境来结束我们的写作；最后我们通过 `command+t` 进行排版；

补充：1. `\\` 代码来进行换行，不缩进（也可以在文稿区空一行）

2. `\par` 代码来进行换行，缩进

3. `\vspace{Xpt}` 代码来设置竖直间距（pt 为单位）

## 三、总结代码

所有的操作和在一起如下：

`\documentclass{xxxx}`

`\begin{document}`

xxxx

`\end{document}`





## 第二章 设置中文环境

上一个板块中我们只能用英语来编辑文稿内容，其中不能出现中文，但我们可以引用宏包来进行中文编辑；

### 一、ctex 宏包的运用

首先我们要下载 ctex 的宏包（不包含在基本 L<sup>A</sup>T<sub>E</sub>X 中）；

最基本也是最好用的方法就是在导言区直接引用宏包；这里我们不使用 `\usepackage` 的命令，我们选择直接在 `\documentclass` 直接进行编辑，具体如下：

`\documentclass{ctexXXX}` 其中 XXX 为格式，如 art, book 等；

之后的操作与第一章基本操作一样，并且可以用中文编辑；

### 二、总结代码

所有的操作和在一起如下：

```
\documentclass{ctexXXX}
\begin{document}
xxxx
\end{document}
```

这里请务必注意一点（非常重要！）：如果是选择用 mac 中的 texpad 进行编辑，texpad 的自动编辑器为 LIVE 的，无法添加更多的宏包，故如果要用到其他宏包，则都需要如下操作：

1. 点击左上角的‘单个齿轮’的按钮
2. 在上方有两个板块，一个是 LIVE 的板块，另外一个 External 板块，我们选择 External 板块
3. 在两个板块下面有一个自动鉴别板式，选择关闭它（不然它会自动检测，仍然无法达到运用宏包的目的）
4. 在下面的板块中有六个板块可以选择，我们选择 general 那一栏的，先看到第三个栏（document format）然后选择其中的 LaTeX；接着看上面一栏（typeset engineer）然后选择其中的 XeLaTeX 即可；（注：这里要先看 format 再看 engineer，不然会找不到我们要用的 engineer）



## 第三章 字体设置

### 一、字体属性

#### (一) 字体族

字体族中有三种不同的字体：罗马字体 (Roman Family)，无衬线字体 (Sana Serif Family)，打印机字体 (Typewriter Family) 如果要单独使用此三种字体，有三种不同的方式，也即：

罗马字体: 1. `\textrm{xxxx}`    2. `{\rmfamily xxxx}`    3. `\rmfamily xxxx`

无衬线字体: 1. `\textsf{xxxx}`    2. `{\sffamily xxxx}`    3. `\sffamily xxxx`

打印机字体: 1. `\texttt{xxxx}`    2. `{\ttfamily xxxx}`    3. `\ttfamily xxxx`

注意：以上的三种字体都是针对英文的，对于中文没有变化；如果在用完最后一种字体后没有转换的话，会一直延续；

#### (二) 字体系列

字体系列中有两种不同的系列：中体字 (Medium Series)，黑体字 (**Boldface Series**) 如果要单独使用这两种系列，有两种不同的方式，也即：

中体字: 1. `\textmd{xxxx}`    2. `{\mdseries xxxx}`

黑体字: 1. `\textbf{xxxx}`    2. `{\bfseries xxxx}`

注意：与字体族一样，只适用于英文字符，并且会延续；

#### (三) 字体形状

字体形状中有四种不同的形状：直立 (Upright Shape)，斜体 (*Italic Shape*)，伪斜体 (*Slanted Shape*)，小型大写 (SMALL CAPS SHAPE) 如果要单独使用此四种形状，有两种不同的方式，也即：

直立: 1. `\textup{xxxx}`    2. `{\upshape xxxx}`

斜体: 1. `\textit{xxxx}`    2. `{\itshape xxxx}`

伪斜体: 1. `\textsl{xxxx}`    2. `{\slshape xxxx}`

小型大写: 1. `\textsc{xxxx}`    2. `{\scshape xxxx}`

注意：如上面两种属性一样，只适用于英文字符，并且会延续；

#### (四) 中文字体

中文字体有多种样式，在这里介绍六种：宋体，黑体，仿宋，楷书，粗体，斜体；

如果要单独使用此六种字体，即如下表达：

宋体: `{\songti xxxx}`

黑体: `{\heiti xxxx}`

仿宋: `{\fangsong xxxx}`

楷书: { \kaishu xxxx }

粗体: \textbf{xxxx}

斜体: \textit{xxxx}

注意: 这里的设置只能在中文字符中使用, 因此如果没有中文字符宏, 也无法运用;

## (五) 字体大小

字体大小有多种设置方法, 如下:

### 总设置

直接在导言区进行编辑, 来设置正篇文章的字体大小, 操作如下:

\documentclass[xpt]{ctexXXXX} 这里的 [xpt] 中的 x 即设置的字体大小;

### 单独设置

单独设置中有多种字体大小可以使用, 在这里介绍十种 (后附带代码):

7 号, { \tiny xxxx }

小六, { \scriptsize xxxx }

六号, { \footnotesize xxxx }

小五, { \small xxxx }

五号, { \normalsize xxxx }

小四, { \large xxxx }

小四, { \Large xxxx }

小二, { \LARGE xxxx }

二号, { \huge xxxx }

一号, { \Huge xxxx }

除此之外, 还有另外一种编辑方式, 如下:

\zihao{x}XXXX 这里的 {x} 就是后面 XXXX 的字体大小;

下表是字体大小的综合 (L<sup>A</sup>T<sub>E</sub>X 符号 – word 字号 – pt 数值):

L <sup>A</sup> T <sub>E</sub> X	word	pt
\tiny	七号	5.5
\scriptsize	小六	6.5
\footnotesize	六号	7.5
\small	小五	9
\normalsize	五号	10.5
\large	小四	12
\Large	小三	15
\LARGE	小二	18
\huge	二号	22
\Huge	一号	26

除了在文本编辑区域之外, 还可以在其他地方使用字体设置, 如:

在导言区编辑标题和作者的时候, 可以使用字体设置;

`\title {\zihao {X} \字体 XXXX }` 其中 XXXX 就是标题名称;

`\author {\zihao{x} \字体 XXX}` 其中 XXX 就是作者姓名;

## 二、进阶运用

上面讲述的都是单独运用的情况,那如果我们需要对文字进行多种属性的编辑(如:无衬线字体 + 黑体字 + 斜体),则有更加快捷与简便的形式,操作方式如下:

1. 首先我们在导言区进行编辑(即设置我们需要的字体样式);代码如下:

```
\newcommand{\xxxx}{ \格式 1{格式 2{内容}}}
```

2. 在导言区编辑好样式和内容后接着去文稿区,直接输入:

```
\xxxx
```

就会出现我们按格式 1 与格式 2 出来的内容了;

例:

导言区: `\newcommand{\myfonta}{ \textit{\textbf{\textsf{Fancy Text}}}}` 即为:斜体 + 粗体 + 无衬线字体的‘Fancy Text’

文稿区: `\myfonta`

PDF: ***Fancy Text***



## 第四章 篇章结构-基础

对于一篇文章来讲，目录和板块的分类是非常重要的，在此板块会针对目录格式进行详细讲解：

### 一、板块分割

在 L<sup>A</sup>T<sub>E</sub>X 中，可以将标题按照重要性进行划分；具体如下：（重要性由上往下依次递减，chapter 最大，一般不进行格式更改）

名称	L <sup>A</sup> T <sub>E</sub> X 中代码
chapter	<code>\chapter{}</code>
section	<code>\section{}</code>
subsection	<code>\subsection{}</code>
subsubsection	<code>\subsubsection{}</code>
⋮	⋮

也就是当需要用到标题时，按照重要性输入代码，在 { } 中输入标题名称即可；（注：用 TexPad 进行编辑则生成的标题会在左边大纲栏出现，并且点击大纲栏的标题就会跳转到相对应的文稿区部分）

除此之外，如果在已经有标题的情况下生成文档的目录，则需要以下的代码：

`\tableofcontents`

注意：此代码需要添加在文稿区部分，具体为

`\begin{document}` 后面进行添加（如果有 maketitle，则要在 maketitle 后面，不然文章第一页就是目录，后面才是封面）；

### 二、目录格式设置

在编写完文章中的大小标题以及创建目录后，我们可以对目录的格式进行修改（其中包括序号的格式 [中文/数字][黑体/宋体/等] 以及标题之间的间隔（注：这里指改变目录的，文章中的标题间隔会在下个板块进行讲解））

因为这部分大致相同，故只举出两个例子，剩下的可以举一反三；

在 L<sup>A</sup>T<sub>E</sub>X 中设置格式的代码：（注：是在导言区进行编辑）

1. `format+ = \zihao{x} \xxxx \raggedright(centering/raggedleft),`

编辑的就是 section 标题的形式编辑（由字号，字体样式，对齐 [分别是居左，居中，居右] 组成）

2. `name={x},`

这里 x 就是 section 序号格式；如 ( ) 出来的就是一、二、三、四、五、六、七、八、九、十、

3. `number = \chinese(arabic),`

选择序号是中文还是阿拉伯数字

4.beforeskip = X.Xex plus X.Xex minus .Xex,

这里是指目录中标题前的空间，第一个 X.X 表示原始空间的大小，后面的 plus 表示可以延伸到 X.X，minus 表示可以收缩到 X.X（这里省略了第一个 X）[注：此处的 ex 是指正常情况下一个‘x’字符的大小]

5.afterskip = X.Xex plus X.Xex minus .Xex,

同理，afterskip 就是目录中标题后的空间，剩下的与 beforeskip 一致

6.aftername = \hspace{Xpt}

此处编辑的是目录中序号和标题的水平间隔，用 pt 为单位来确定

因此，将上述六个部分和在一起即可完成一种标题的编辑，合并起来后效果如下：（用 section 来举例）

```
section={
  format+ = \zihao{x} \xxxx \raggedright(centering/raggedleft),
  name = {x,},
  number = \chinese(arabic){section},
  beforeskip = X.Xex plus X.Xex minus .Xex,
  afterskip = X.Xex plus X.Xex minus .Xex,
  aftername = \hspace{Xpt}
}
```

在已经会设置单个标题后，我们将所有的标题和在一起，并且在最开始用\ctexset 来在导言区进行设置；此篇文章的部分设置如下（以下程序是可以直接复制粘贴使用的）：

```
\ctexset{
  section={
    format+ = \zihao{-3} \fangsong \raggedright,
    name = {,、 },
    number = \chinese{section},
    beforeskip = 1.0ex plus 0.2ex minus .2ex,
    afterskip = 1.0ex plus 0.2ex minus .2ex,
    aftername = \hspace{0pt}
  },
  subsection={
    format+ = \zihao{4} \kaishu \raggedright,
    name = {(,)},
    number = \chinese{subsection},
    beforeskip = 1.0ex plus 0.2ex minus .2ex,
    afterskip = 1.0ex plus 0.2ex minus .2ex,
    aftername = \hspace{0pt}
  }
}
```



## 第五章 篇章结构-进阶

在前面介绍完目录设置后，在此介绍对文本内的布局格式说明，分为两个板块：文本设置和边框设置；

### 一、文本设置

在文本设置中，可以更改纸张大小与文字排版大小（即文字在整张纸中占的位置）；

在更改文本格式之前，需要引入宏包（geometry），也即在导言区要输入：`\usepackage{geometry}`

#### (一) 纸张大小设置

`\geometry{xxxx}` 其中 xxxx 就是纸张大小，例如 `a4paper`，`a3paper` 等；

`\geometry{paperheight=XXcm,paperwidth=XXcm}` 这种方式就可以直接设置长和宽

#### (二) 文本区域大小设置

`\geometry{left=Xcm}` 这里是设置文本距离纸张左侧（页面左棱）的距离

`\geometry{right=Xcm}` 这里是设置文本距离纸张右侧（页面右棱）的距离

`\geometry{top=Xcm}` 这里是设置文本距离纸张上侧（页面上棱）的距离

`\geometry{bottom=Xcm}` 这里是设置文本距离纸张下侧（页面下棱）的距离

`\geometry{textheight=Xcm,textwidth=Xcm}` 这种方式可以直接设置文本区域的长和宽

#### (三) 文本区域文字分布

`\geometry{vcentering}` 若输入此代码，则文本成竖直居中

`\geometry{hcentering}` 若输入此代码，则文本成水平居中

`\geometry{centering}` 若输入此代码，则文本成水平，竖直居中

### 二、边框设置

在边框设置中，可以设置页眉页脚的注释，以及页码的编辑；

在更改边框格式之前，需要引入宏包（fancyhdr），也即在导言区要输入：

`\usepackage{fancyhdr}` （注：fancyhdr 要写在 geometry 宏包之后，不然会出错）

`\fancyhf{}` （此行代码也需要输入）

#### (一) 页眉 & 页脚的设置

页眉，即文章最上面的一行；页脚，即文章最下面的一行

`\lhead{xxxx}`    xxxx 是左页眉的内容  
`\chead{xxxx}`    xxxx 是中页眉的内容  
`\rhead{xxxx}`    xxxx 是右页眉的内容（此文章的右页眉就是：L<sup>A</sup>T<sub>E</sub>X 基础操作汇总）  
`\lfoot{xxxx}`    xxxx 是左页脚的内容  
`\cfoot{xxxx}`    xxxx 是中页脚的内容  
`\rfoot{xxxx}`    xxxx 是右页脚的内容

注意：此上六个代码都是在导言区进行编辑，如果设置页眉，会自动出现页眉线

## (二) 页码的使用

如果需要在页底出现代码，则输入指令：

`\rfoot{thepage}`

注意：如果文章前面有了 `\maketitle`，`\maketitle` 指令会自动屏蔽我们设置的页码指令，因此我们需要手动加上：

`\thispagestyle{fancy}`

如果有 `\chapter`，则需要在每一个 `\chapter` 底下加上此行代码

## 三、总结代码

导言区：（以本片文章的设置来举例）

```

\usepackage{geometry}
\geometry{a4paper}
\geometry{left=2cm}
\geometry{right=2cm}
\geometry{top=3cm}
\geometry{bottom=3cm}
\usepackage{fancyhdr}
\pagestyle{fancy}
\fancyfh{}
\rhead{\LaTeX{} 基础操作汇总}
\rfoot{\thepage}

```

文稿区： `\begin{document}`

```

    \maketitle
    \tableofcontents
    \thispagestyle{fancy}
    \chapter{xxxx}
    \thispagestyle{fancy}
    \vdots
\end{document}

```

## 第六章 特殊符号

### 一、空白符号

空白符号也称之为空格符号，因为 L<sup>A</sup>T<sub>E</sub>X 的文档区空格（中文编辑情况下）并不会应用到 PDF 的文档中，所以如果我们需要空格，则要用特殊命令来表示空格；空白符号中的空格形式有十一种，表现形式如下：

1. 1em（即‘M’）空格：a    b
2. 2em 空格：a        b
3.  $\frac{1}{6}$ em 空格：a b
4.  $\frac{1}{2}$ em 空格：a   b
5. 空格：a b
6. 硬空格：a b
7. 1pc 空格：a   b（注：1pc=12pt=4.218mm）
8. 反书写：ba
9. 自控制空格：a            b
10. 占位空格：a        b（注：这里 a 和 b 中间的空格就是 xxxx 的长度）
11. 弹性空格：a

b

在看过上面空格的种类的长度后，选取需要的进行运用，代码如下：

1. 1em: a \quad b
2. 2em: a \quad\quad b
3.  $\frac{1}{6}$ em: a \thinspace b        或者        a\, b
4.  $\frac{1}{2}$ em: a \enspace b
5. 空格: a \b
6. 硬空格: a~b
7. 1pc 空格: a\kern 1pc b
8. 反空格: a\kern -1em b
9. 自控制空格: a\hspace{xxpt}
10. 占位空格: a \hphantom{xxxx}b        (注: 这里的 xxxx 就是在 ab 之间正常排列的符号, 省略了)
11. 弹性空格: a\hfill b

### 二、L<sup>A</sup>T<sub>E</sub>X 控制符

对于一些特别的符号，我们在文档区编辑的话无法在 PDF 现实出来（如%，在文档区打出来是编辑符号，而不是出来% 符号）；因此，需要用到特定的表示方法来将此类符号表示出来，具体符号如下：

符号	L <sup>A</sup> T <sub>E</sub> X 中的表示方法
#	\#
\$	\\$
%	\%
{	\{
}	\}
~	\~{} { }
_	\_{} { }
\	\textbackslash
&	\&

### 三、排版符号

除了  $\text{\LaTeX}$  控制符以外还有其他的一些符号可以进行运用, 例如:  $\text{\textcircled{R}}$ ,  $\text{\textcircled{TM}}$ ,  $\text{\textcircled{SM}}$  等, 具体操作如下:

符号	L <sup>A</sup> T <sub>E</sub> X 中的表示方法
§	\S
¶	\P
†	\dag
‡	\ddag
©	\copyright
£	\pounds

#### 四、T<sub>E</sub>X 标志符号

LaTeX 和 TeX 都有自己的专用标志符号, 通过引入宏包, 我们可以来加入更多的符号, 具体如下: (注: 这里我们需要引入宏包: xltextra, 也即需要在导言区加上)

$$\mathrm{T_{\mathrm{F}}X} \quad \backslash \mathrm{Tex}\{\}$$
$$\text{\LaTeX} \quad \backslash \text{\LaTeX}\{ \}$$
$$\text{\LaTeX 2}_{\epsilon} \quad \backslash \text{\LaTeXe{}}$$

## 五、其他符号

除了以上四种外，还有其他一些常用/可能用到的符号，如引号类：

‘ ’ “ ”, 左引号用键盘 1 键左侧撇来表示 (双引号用两次), 右引号用单引号表示 (双引号用两次)

还有连词符类:

- - -，用不同长度的横杠（键盘 0 键右侧）来表示不同长度连词符；

以及非英文字符: œ, Œ, æ, Æ, å, Å, ø, Ø, ł, Ł, ß, , ¡, ¿.

分别对应: \oe, \OE, \ae, \AE, \aa, \AA, \o, \O, \l, \L, \ss, \SS, !', ?'

还有重音符号: ò ó ô 'o õ ö ø ö ö ö ö ö ö ö (注: 这里只是用 o 做示范)

分别对应: \`o \^o \~o \=o \.o \u{o} \v{o} \H{o} \r{o} \t{o} \b{o} \c{o} \d{o}

## 第七章 插图

插图一种非常实用的表达方式，在此介绍图片的插入与部分排版信息；

为了正常的使用插图，需要在导言区引入宏包（graphicx），也即需要编写代码：

```
\usepackage{graphicx}
```

### 一、插入图片

1. 地址：在引入图像之前，需要先将所需的图像放入文件夹中，方便引入（可以分为多个文件夹，文件夹最好不要重名，并且文件夹与 L<sup>A</sup>T<sub>E</sub>X 的文件在同一层中）

代码：\graphicspath{{< 地址 1/>},{< 地址 2/>}...}（注：此部分在导言区编辑）

2. 文件格式：L<sup>A</sup>T<sub>E</sub>X 中适用的格式有：EPS，PDF，PNG，JPEG，BMP

3. 引入图像：\includegraphics{< 文件夹名称 >/< 图像名称 >}（注：若用 texpad，则可以自动选择，不需要手打）

例 1：（现在在 figures 文件夹中有 Fig1，在 pics 文件夹中有 Fig2）

导言区：

```
\usepackage{graphicx}
\graphicspath{{figures/},{pics/}}
```

文稿区：

```
\includegraphics{figures/IMG_0400}
\includegraphics{pics/IMG_0401}
```

PDF：这里原图片尺寸太大了，不放上来了，等到编辑完大小再放上来；

### 二、编辑图片

编辑图片可以分为几个方面：大小，延伸，转动；

1. 大小编辑：在文稿区代码中加入：[scale=X] 即可；

```
\includegraphics[scale=X]{figures/IMG_0400}
```

2. 延伸编辑：在文稿区代码中加入：[height=Xcm] 或者 [width=Xcm] 即可；

```
\includegraphics[height(width)=Xcm]{figures/IMG_0400}
```

3. 转动编辑：在文稿区代码中加入：[angle=X] 即可；

```
\includegraphics[angle=Xcm]{figures/IMG_0400}
```

4. 综合编辑：可以将上述的代码共同使用；

```
\includegraphics[angle=X,scale=X]{figures/IMG_0400}
```

例 2: (现在再次导入图像, 并且进行尺寸编辑)

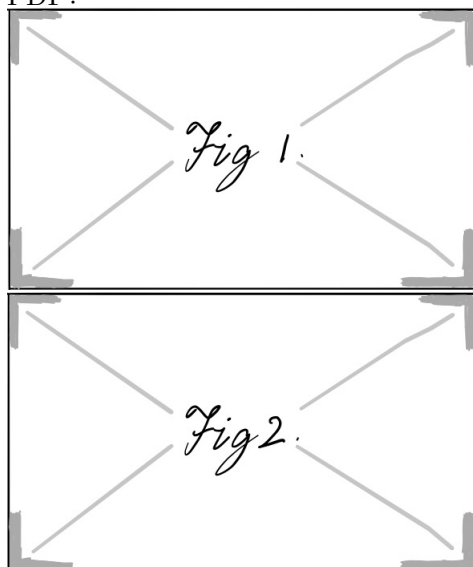
导言区:

```
\usepackage{graphicx}
\graphicspath{{figures/},{pics/}}
```

文稿区:

```
\includegraphics[scale=0.3]{figures/IMG_0400}
\includegraphics[scale=0.3]{pics/IMG_0401}
```

PDF:



## 第八章 表格

### 一、表格内部参数

表格时编辑 L<sup>A</sup>T<sub>E</sub>X 中比较重要的一个功能，在此只介绍最基本的表格；具体操作如下：

1. `\begin{tabular}` 这里是构建一个专门建立表格的环境（类似构建文稿区）

2. `{|l|c|r|}` 这一步是建立表格的竖列格式，其中一个 `|` (shift+ 中文顿号) 就是在表格中建立一个列分割线（如果连续两条 `|` 则意味着在两列之间有两条列分割线），`l` 表示左对齐，`c` 表示居中对齐，`r` 表示右对齐

3. `\hline` 此代码的意义为构建一个行分割线（如果有连续两条 `\hline \hline` 则意味着在两行之间有两条行分割线）

4. `a&b&c\\` 这行代码表示：此行中三列表格中的数分别为：a, b, c

注意两点！：1. 注意在最开始编辑了列数，即 `l, c, r` 的个数，一定不能超过，可以不填，用空格代替；

2. 注意在编写内容的时候最后一定要 `\\` 一定要记得换行

### 二、总结代码

文稿区：

```
\begin{tabular} {|l|c|r|}
\hline
(1,1) & (1,2) & (1,3) \\
\hline
(2,1) & (2,2) & (2,3) \\
\hline
(3,1) & (3,2) & (3,3) \\
\hline
\end{tabular}
```

PDF:

(1,1)	(1,2)	(1,3)
(2,1)	(2,2)	(2,3)
(3,1)	(3,2)	(3,3)





## 第九章 浮动体

浮动体是指 L<sup>A</sup>T<sub>E</sub>X 中的插图和表格等，可以通过一些指令来更改他们的位置；  
在这里介绍六个代码：

1. `\label` 此代码一般写在浮动题后面，为浮动题设置标签
2. `\ref` 此代码一般写在需要引用浮动题标签的地方，会自动标号
3. `\caption{xxxx}` 此代码一般写在浮动题后面，为浮动题命名
4. `\begin{table}(figure)` 此代码构建编辑浮动题环境
5. `\centering` 此代码一般用于环境第一行，使浮动题居中
6. `[h/t/b/p]` 此处为四种设置格式，具体见下表 9.1

### 一、表格

根据上面的代码，进行举例说明：

例 1：

文稿区：

```
\begin{table}[h]
  \centering
  \caption{h/t/b/p 使用说明} \label{table-function}
  \begin{tabular}
    此处为表格设置，不再赘述
  \end{tabular}
end{table}
```

PDF：

表 9.1: h/t/b/p 使用说明

字母	名称	作用
h	此处 (here)	代码所在的上下文位置
t	页顶 (top)	代码所在的页面或之后页面的顶部
b	页底 (bottom)	代码所在的页面或之后页面的底部
p	独立一页 (page)	浮动页面

## 二、插图

插图的使用与上述的表格使用差不多，根据上面的代码，进行举例说明：

例 2：

文稿区：

下图是引用并编辑插图的使用说明，插图见图\ref{fig-1}

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

```
\centering
```

```
\includegraphics[scale=0.3]{figures/IMG_0400
```

```
\caption{Fig.1} \label{fig-1}
```

```
\end{figure}
```

PDF：

下图是引用并编辑插图的使用说明，插图见图 9.1



图 9.1: Fig.1

## 第十章 数学公式初步应用

接下来讲解重要的一部分，数学板块的运用；（此部分会讲解的比较详细）

L<sup>A</sup>T<sub>E</sub>X 的文本区中分为两种排版，一个是普通文本排版，另一个是数学公式排版；因此当我们需要运用到数学公式排版的时候，就需要创建一个数学排版区域进行编排；

### 一、行内公式

#### (一)美元符号

通常情况下，都运用单美元符号 \$ 来进行数学公式的排版，这是最基础也是最好用的方法，具体操作方式如下：

文稿区：\$ f(x)=3x+5\$

PDF:  $f(x) = 3x + 5$

#### (二)小括号

除了使用美元符号以外，还可以使用小括号来构建数学公式排版环境（二者作用一样），具体操作方式如下：

文稿区：\ (a+b=b+a\)

PDF:  $a + b = b + a$

#### (三)math 环境

再其次，也可以直接构建一个环境来进行数学公式的排版（类似构建编辑环境来进行编辑），具体操作方式如下：

文稿区：\begin{math} x+y=y+x \end{math}

PDF:  $x + y = y + x$

### 二、上下标

上下标是我们在数学中经常使用的符号，如高次方程，数列等都会用到，在此介绍 L<sup>A</sup>T<sub>E</sub>X 中上下标的使用：

#### (一)上标

我们运用 ^ 符号（即 shift+6 键）来运用上标；例：

文稿区：x<sup>{2}</sup>+y<sup>{3}</sup>

PDF:  $x^2 + y^3$

这里我们文稿区运用了 {}, 对于 1 位数也可以不加, 但 2 位数及以上需要 {}

除此之外, 也可以在上标上继续标上标, 也即:

文稿区:  $e^{\{x^{\{2\}}+x\}}$  (一层一层)

PDF:  $e^{x^2+x}$

## (二) 下标

我们运用 \_ 符号 (即 shift+0 键右边的横杠) 来运用下标; 例:

文稿区:  $\ln_{\{2\}}$

PDF:  $\ln_2$

这里同样用到了 {}, 与上标一样, 2 位数及以上需要 {}

除此之外, 也可以在下标下继续标下标, 也即:

文稿区:  $\ln_{\{x_{\{2\}}+x_{\{3\}}\}}$  (一层一层)

PDF:  $\ln_{x_2+x_3}$

## (三) 综合运用

在会运用上标和下标之后, 可以上下标同时使用并且不会有影响;

例 1:

文稿区:  $x^{\{a_{\{1\}}-a_{\{2\}}\}}$

PDF:  $x^{a_1-a_2}$

例 2:

文稿区:  $x_{\{a^{\{2\}}+b^{\{2\}}\}}$

PDF:  $x_{a^2+b^2}$

## 三、希腊字母

在数学算式中, 通常会遇到不少希腊字母, 在此介绍一部分希腊字母的大小写即代码:

小写	小写代码	大写	大写代码
$\alpha$	<code>\alpha</code>	null	null
$\beta$	<code>\beta</code>	null	null
$\gamma$	<code>\gamma</code>	$\Gamma$	<code>\Gamma</code>
$\delta$	<code>\delta</code>	$\Delta$	<code>\Delta</code>
$\theta$	<code>\theta</code>	$\Theta$	<code>\Theta</code>
$\omega$	<code>\omega</code>	$\Omega$	<code>\Omega</code>

除此之外, 希腊字母也可以与上标和下标结合: 例:

文稿区:  $\alpha_1^{\{3\}} + \beta_2^{\{2\}} = \gamma_3^{\{5\}}$

PDF:  $\alpha_1^3 + \beta_2^2 = \gamma_3^5$

## 四、数学函数

数学函数是在撰写数学论文时非常重要的一环, 在这里先介绍基本函数的表示方法, 具体如下:

数学函数表达式	数学函数代码
log	<code>\$ \log \$</code>
sin	<code>\$ \sin \$</code>
cos	<code>\$ \cos \$</code>
arcsin	<code>\$ \arcsin \$</code>
ln	<code>\$ \ln \$</code>

同理，数学函数也可以与上标和下标结合使用：例：

文稿区：`$ \sin^{2} + \cos^{2} = 1 $`

PDF： $\sin^2 + \cos^2 = 1$

以此类推，也可以构建其他的函数：

例 1：

文稿区：`$ y=\sin^{-1} x $`

PDF： $y = \sin^{-1} x$

例 2：

文稿区：`$ y= \log_{2} x $`

PDF： $y = \log_2 x$

例 3：

文稿区：`$ y= \ln x $`

PDF： $y = \ln x$

此外，有一个特殊函数表达符号，即根号函数，应用如下：

文稿区：`$ \sqrt{x}{xxxx} $`（注：其中 xxxx 的位置即是底数位置）；

例 1：

文稿区：`$ \sqrt{2} $`

PDF： $\sqrt{2}$

例 2：

文稿区：`$ \sqrt{x^{2} + y^{2}} $`

PDF： $\sqrt{x^2 + y^2}$

例 3：

文稿区：`$ \sqrt{2+\sqrt{2}} $`

PDF： $\sqrt{2 + \sqrt{2}}$

例 4：

文稿区：`$ \sqrt[4]{x} $`（注：这里的 [4] 是指开四次方）

PDF： $\sqrt[4]{x}$

## 五、分式

分式也是数学表示中非常重要的一部分，在此，给出两种表达分式的方法，具体如下：

1. 直接用斜杠来表示（比较容易编辑，但观感度很差）：

例：

文稿区：`$ x/y $`

PDF:  $x/y$

2. 运用函数 `frac` 来表示分数（编辑起来较为麻烦，但即为美观）：

例：

文稿区：\$ `frac{x}{y}` \$（注：前者为分子，后者为分母）

PDF:  $\frac{x}{y}$

同理，可以与上述的上下标和函数结合起来；如下：

例 1：

文稿区：\$ `\frac{x}{x^2+x+1}` \$

PDF:  $\frac{x}{x^2+x+1}$

例 2：

文稿区：\$ `\frac{\sqrt{1+x}}{\sqrt{x-1}}` \$

PDF:  $\frac{\sqrt{1+x}}{\sqrt{x-1}}$

：例 3：

文稿区：\$ `\frac{1}{1+\frac{1}{x}}` \$

PDF:  $\frac{1}{1+\frac{1}{x}}$

例 4：

文稿区：\$ `\sqrt{\frac{x}{x^2+x+1}}` \$

PDF:  $\sqrt{\frac{x}{x^2+x+1}}$

## 六、行间公式

行间公式，顾名思义，就是公式不再跟在文字的后面，而是另起一行，并且居中；接下来给出几种表达方式：

### (一) 双美元符号

双美元符号同单美元符号，是最好用且最简单的，操作如下：

文稿区：\$\$`a+b=b+a`\$\$

PDF:

$$a + b = b + a$$

或者：

文稿区：\$\$`\sqrt{\alpha}+\frac{\beta}{\gamma^3}=\frac{\beta}{\gamma^3}+\sqrt{\alpha}`

\$\$

PDF:

$$\sqrt{\alpha} + \frac{\beta}{\gamma^3} = \frac{\beta}{\gamma^3} + \sqrt{\alpha}$$

### (二) 中括号

除了双美元符号以外，也可以用中括号来进行编排行间公式，具体操作如下：

文稿区：\[`a+b=b+a`\]

PDF:

$$a + b = b + a$$

或者：

文稿区：`\[\sqrt{\alpha}+\frac{\beta}{\gamma^3}=\frac{\beta}{\gamma^3}+\sqrt{\alpha}\]`

PDF:

$$\sqrt{\alpha} + \frac{\beta}{\gamma^3} = \frac{\beta}{\gamma^3} + \sqrt{\alpha}$$

### (三) `displaymath` 环境

此外，还可以运用直接构建行间公式环境来进行编辑，（类似 `math` 环境和文稿排版环境），具体操作如下：

文稿区：`\begin{displaymath} a+b=b+a \end{displaymath}`

PDF:

$$a + b = b + a$$

或者：

文稿区：`\begin{displaymath}`

`\sqrt{\alpha}+\frac{\beta}{\gamma^3}=\frac{\beta}{\gamma^3}+\sqrt{\alpha}`  
`\end{displaymath}`

PDF:

$$\sqrt{\alpha} + \frac{\beta}{\gamma^3} = \frac{\beta}{\gamma^3} + \sqrt{\alpha}$$

### (四) 自动编号公式 `equation` 环境

使用自动编号环境可以再每一个公式后面进行自动编号，（注：是自动的，因此每一个都会标上序号，这里会用到`\ref`和`\label`，在浮动题中有详细讲解），具体操作如下：

1. 构建一个专门有序号排版的环境

`\begin (end) {equation}`

2. 在公式后面运用`\label`，就可以在文稿中直接引用标号

例 1:

文稿区:

加法交换律见式`\ref{eq:commutative1}`

`\begin{equation}`

`a + b = b + a \label{eq:commutative1}`

`\end{equation}`

PDF:

加法交换律见式 10.1

$$a + b = b + a \tag{10.1}$$

注意：此处的序号是自动标记在式子的最后，并且是按顺序，有（1）之后就一定是（2），（2）后面就一定是（3），（2）后面不可能再次出现（1）





## 第十一章 数学矩阵的应用

矩阵也数学公式中重要的一部分，多用于线性代数方面，在此介绍几种矩阵的表达方式；具体如下：

不同于一般的数学公式，在使用矩阵的时候，需要引入特定的宏包（amsmath），也即需要在导言区编写代码：

```
\usepackage{amsmath}
```

### 一、通用矩阵

在编辑任何矩阵之前都需要构建一个编辑矩阵的环紧，即在编辑矩阵（从一开始一直到所有矩阵都写完）用：\[ 和 \] 来构建一个编辑矩阵的整体环境；

在设置完整体环境后，再创建单个矩阵的编辑环境（与构建数学公式排版类似），也即用\begin{matrix} 和\end{matrix} 来构建单个环境；接下来来介绍六种基本通用矩阵，具体如下：（注：如果只有一个矩阵，会居中排版，多个矩阵则居左）

#### (一)无边框

文稿区：\[ \begin{matrix}

$$\begin{matrix} 0 & 1 \\ 1 & 0 \end{matrix}$$

\end{matrix} \]

PDF:

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

#### (二)小括号

文稿区：\[ \begin{pmatrix}

$$\begin{pmatrix} 0 & i \\ i & 0 \end{pmatrix}$$

\end{pmatrix} \]

PDF:

$$\begin{pmatrix} 0 & i \\ i & 0 \end{pmatrix}$$

#### (三)中括号

文稿区：\[ \begin{bmatrix}

$$\begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$$

`\end{bmatrix} \]`

PDF:

$$\begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$$

#### (四) 大括号

文稿区: `\[ \begin{Bmatrix}`

`1 & 0 \\\`

`0 & 1`

`\end{Bmatrix} \]`

PDF:

$$\begin{Bmatrix} 1 & 0 \\ 0 & 1 \end{Bmatrix}$$

#### (五) 行列式

文稿区: `\[ \begin{vmatrix}`

`a & b \\\`

`c & d`

`\end{vmatrix} \]`

PDF:

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix}$$

#### (六) 双竖线

文稿区: `\[ \begin{Vmatrix}`

`i & 0 \\\`

`0 & i`

`\end{Vmatrix} \]`

PDF:

$$\begin{Vmatrix} i & 0 \\ 0 & i \end{Vmatrix}$$

## 二、矩阵与数学符号结合

同理, 可以将矩阵与之前的数学符号 (如上下标, 函数) 结合在一起使用; (不需要使用构建数学公式的符号)

例 1:

文稿区:

`\[ A= \begin{pmatrix}`

`a_{11}^2 & a_{12}^2 & a_{13}^2 \\\`

`0 & a_{22}^2 & a_{23}^2 \\\`

`0 & 0 & a_{33}^2`

`\end{pmatrix}_{n \times n} \]`

PDF:

$$A = \begin{pmatrix} a_{11}^2 & a_{12}^2 & a_{13}^2 \\ 0 & a_{22}^2 & a_{23}^2 \\ 0 & 0 & a_{33}^2 \end{pmatrix}_{n \times n}$$

### 三、省略号

在运用矩阵的时候，会遇到很多情况需要用到省略号（例如行列式的表达），在此介绍两种省略号表达方式，具体操作如下：

#### (一)常用省略号

常用省略号分为三种：水平，垂直，右斜；具体操作如下：（三者使用方式一致，一次性讲解）

1. LaTeX 中的代码：(1) 水平 `\dots` (2) 垂直 `\vdots` (3) 右斜 `\ddots`

2. 应用方法：即将原本矩阵中的元素替换为省略号即可；

例 1:

文稿区:

```
\[ A= \begin{bmatrix}
a_{11} & \dots & a_{1n} \\
& \ddots & \vdots \\
0 & & a_{nn} \end{bmatrix}_{n \times n}
```

PDF:

$$A = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ & \ddots & \vdots \\ 0 & & a_{nn} \end{bmatrix}_{n \times n}$$

#### (二)跨行省略号

比起常用省略号，跨行省略号不是那么常见，但也可以当成一种额外补充，具体操作如下：

1. LaTeX 中的代码：跨行省略号 `\hdotsfor{< 列数 >}`（列数代表的是省略号占据列的数量）

2. 应用方法：即将原本矩阵中的元素（们）替换为省略号即可；

例 1:

文稿区:

```
\[ \begin{pmatrix}
1 & \frac{1}{2} & \dots & \frac{1}{n} \\
2 & \hdotsfor{2} & \frac{2}{n} \\
\vdots & & \vdots \\
m & \frac{m}{2} & \dots & \frac{m}{n} \end{pmatrix}
```

PDF:

$$\begin{pmatrix} 1 & \frac{1}{2} & \dots & \frac{1}{n} \\ 2 & \dots\dots\dots & \dots & \frac{2}{n} \\ \vdots & & & \vdots \\ m & \frac{m}{2} & \dots & \frac{m}{n} \end{pmatrix}$$

## 四、分块矩阵

分块矩阵也称为矩阵嵌套，即一个矩阵里面的一个元素可以是一个矩阵，具体操作如下：

应用原理：就是把原本矩阵中的元素用一个矩阵来表达，即构建一个环境编辑一个矩阵；

补充：如果能让一个元素占据  $2 \times 2$  的格子，则可以用代码：`\text{\Large x}` (其中 x 就是放大的字符, text 的意思为在数学排版模式中临时切换回文本排版模式)

例 1:

文稿区:

```
\[ \begin{bmatrix}
\begin{pmatrix} 1&0 \\ 0&1 \end{pmatrix} & \text{\Large 0} \\
\text{\Large 0} & \begin{pmatrix} 1&0 \\ 0&-1 \end{pmatrix}
\end{bmatrix} \]
```

PDF:

$$\begin{bmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} & 0 \\ 0 & \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \end{bmatrix}$$

## 五、行内小矩阵

故名思义，就是类似构建数学公式排版，在句子后面跟着矩阵（而不是独立成一行并居中），大小与行间距一样；

L<sup>A</sup>T<sub>E</sub>X 中的代码：`$ \left( \begin{smallmatrix} xxxx \end{smallmatrix} \right) $`

注意：这里需要构建数学环境，不再需要用到`\[和\]`，但是要有`( )`，这个括号是用来构建矩阵左右两边的`( )`用的；

例 1:

文稿区:

```
行内小矩阵就是：
$ \left(
\begin{smallmatrix}
x & -y \\ y & x
\end{smallmatrix}
\right) $
```

PDF: 行内小矩阵就是:  $\begin{pmatrix} x & -y \\ y & x \end{pmatrix}$

## 第十二章 数学多行公式

数学公式除了以单独的方式出现外，还有以成对成组出现的情况，因此在此给出成对成组出现时的四种编辑方式，具体操作如下：

与矩阵一样，在进行多行公式编辑的时候要引入宏包 (amsmath) 和 (amssymb)，也即需要在导言区编辑：`\usepackage{amsmath}` 以及 `\usepackage{amssymb}`

### 一、gather 和 gather\* 环境

#### (一)gather 环境

gather 环境中编辑的公式是自动分行的，一句一行，并且每一句都会在最后有编号；

例 1：

文稿区：

```
\begin{gather}
  a + b = b + a \\
  ab = ba
\end{gather}
```

PDF：

$$a + b = b + a \tag{12.1}$$

$$ab = ba \tag{12.2}$$

#### (二)gather\* 环境

gather\* 环境中编辑的公式格式与 gather 环境中的几乎一样，但是 gather\* 环境中每行公式最后都不带有标号；

例 1：

文稿区：

```
\begin{gather*}
  a + b = b + a \\
  ab = ba
\end{gather*}
```

PDF：

$$a + b = b + a$$

$$ab = ba$$

### (三)\notag 的使用

但事实上也可能遇到一种情况：一些公式需要标号，一些不需要，因此就有了\notag；在数学公式后面使用\notag，可以使得当行数学公式不进行标号；

例 1:

文稿区:

```
\begin{gather}
  a + b = b + a \\
  ab = ba \notag \\
  a^{\{2\}} + b^{\{2\}} = c^{\{2\}}
\end{gather}
```

PDF:

$$a + b = b + a \tag{12.3}$$

$$ab = ba$$

$$a^2 + b^2 = c^2 \tag{12.4}$$

## 二、align 和 align\* 环境

### (一)align 环境

align 环境是另一种编辑多行公式的方法，其排版结果与 gather 环境一致；不同之处在于，align 环境中每个字符之间要用 & 符号来进行对齐（& 后面的符号会自动对齐，但要对齐的字符要跟在与 & 后面，不能空格）

例 1:

```
\begin{align}
  x&=t+\cos t +1 \\
  y &=2\sin t
\end{align}
```

PDF:

$$x = t + \cos t + 1 \tag{12.5}$$

$$y = 2 \sin t \tag{12.6}$$

### (二)align\* 环境

同理，align\* 环境的作业与 gather\* 环境的作用一致，但是需要在字符之间加上 &

例 1:

```
\begin{align*}
  x &= t & x &= \cos t & x &= t \\
  y &= 2t & y &= \sin(t+1) & y &= \sin t
\end{align*}
```

PDF:

$$\begin{array}{lll} x = t & x = \cos t & x = t \\ y = 2t & y = \sin(t + 1) & y = \sin t \end{array}$$

### 三、split 环境

split 环境是针对多行公式但合起来只有一个编号使用的，具体操作方式与前者类似（但是嵌入前者里面进行操作）；

例 1:

文稿区:

```
\begin{equation}
  \begin{split}
    \cos 2x &= \cos^2 x - \sin^2 x \\
    &= 2\cos^2 x - 1
  \end{split}
\end{equation}
```

PDF:

$$\begin{aligned} \cos 2x &= \cos^2 x - \sin^2 x \\ &= 2\cos^2 x - 1 \end{aligned} \tag{12.7}$$

### 四、cases 环境

cases 环境的作用与 split 环境的作用类似（需要 & 来分隔同一行公式的两部分），也是将多个公式和在一起用一个序号，但是它还可以用大括号来进行联立；

例 1: (其中\in 为属于符号；\mathbb 为花体)

文稿区:

```
\begin{equation}
  D(x) = \begin{cases}
    1, & \text{如果 } x \in \mathbb{Q}; \\
    0, & \text{如果 } x \in \mathbb{R} \setminus \mathbb{Q}
  \end{cases}
\end{equation}
```

PDF:

$$D(x) = \begin{cases} 1, & \text{如果 } x \in \mathbb{Q}; \\ 0, & \text{如果 } x \in \mathbb{R} \setminus \mathbb{Q} \end{cases} \tag{12.8}$$





## 第十三章 参考文献 Bibtex

在写论文的时候经常会用到参考文献，对于从其他文献中引用的资料，常常需要在最后，或者该页面底部进行备注，因此参考文献注释的使用是非常重要的，在此介绍此方法，具体操作如下：

`\begin{thebibliography}{样本编号}` 创建环境及编号；

`\bibitem[文献格式]< 引用信息 >` 具体信息和内容

`\bibitem[文献格式]< 引用信息 >`

.....

`\end{thebibliography}`

注意：引用信息包括：作者，题目，出版社，年代，版本，页码等

例 1：

文稿区：此处仍然可以使用引用符号（不是`\ref`和`\label`，而是用`\cite{文献格式}`）

引用`\cite{article1}`和`\cite{book1}`

`\begin{thebibliography}{99}`

`\bibitem{article1}` 陈立辉，苏伟，菜川，陈晓云.\emph{基于 LaTeX 的 Web 数学公式提取方法研究}[J]. 计算机科学.2014(06)

`\bibitem{book1}`William H. Press,Saul A. Teukolsky,William T. Vetterling,Brain P. Flannery,  
`\emph{Numerical Recipes 3rd Edition: The Art of Scientific Computing}` Cambridge University Press,  
 New York, 2007.

`\end{thebibliography}`

PDF：引用 [1] 和 [2]

注意：这里的参考文献不在这一页，而是在（本 chapter）最后的一页



## 参考文献

- [1] 陈立辉, 苏伟, 菜川, 陈晓云. 基于 *LaTeX* 的 *Web* 数学公式提取方法研究 [J]. 计算机科学.2014(06)
- [2] William H. Press, Saul A. Teukolsky, William T. Vetterling, Brain P. Flannery, *Numerical Recipes 3rd Edition: The Art of Scientific Computing* Cambridge University Press, New York, 2007.