

# 第八讲 细节排版 (续)

潘建瑜

华东师范大学数学系

[jypan@math.ecnu.edu.cn](mailto: jypan@math.ecnu.edu.cn)

2014

## 本讲内容

---

- ① 页面设置与 geometry 宏包
- ② 页眉页脚与 francyhdr 宏包
- ③ 颜色与 xcolor 宏包
- ④ 链接与 hyperref 宏包
- ⑤ 多栏排版与 multicol 宏包
- ⑥ 横向排版与 lscapc 宏包
- ⑦ 算术运算与 calc 宏包
- ⑧ 文档的分割处理: input 与 include 命令

# 页面布局

---

## ☞ 页面上摆放文本的四个区域

- 页眉 (header)
- 页芯 (body)
- 页脚 (footer)
- 边注 (margin notes)

## ☞ 页面布局及常用参数

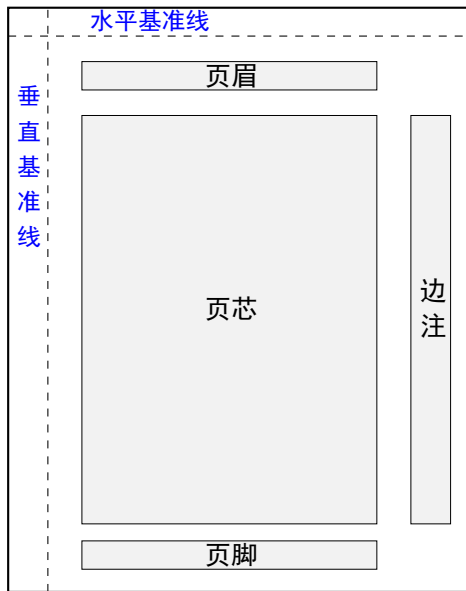
- `\textwidth`, `\textheight`
- `\hoffset`, `\voffset`
- `\oddsidemargin`, `\evensidemargin`

## ☞ `layout` 宏包

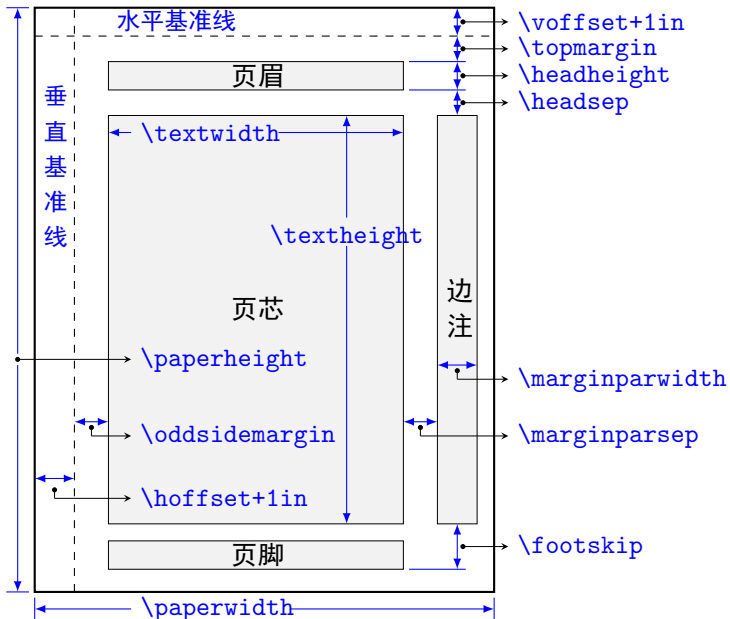
- 提供 `\layout` 命令, 输出当前页面的版面示意图
- 双页排版时 (如 `book`) 有两幅示意图, 分别显示右页和左页的版面布局
- 例: `91layout_art.tex`, `91layout_book.tex`

## 单页排版页面布局/双页排版奇数页的页面布局

---



# 单页排版页面布局/双页排版奇数页的页面布局



# 页面设置

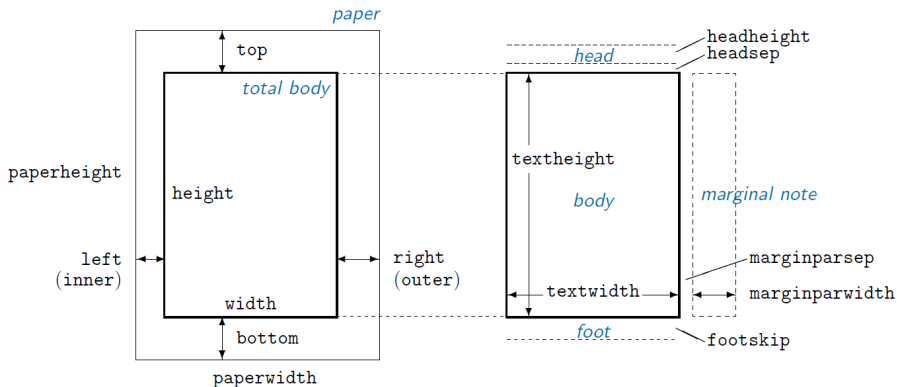
## geometry 宏包: 精确设置页面布局

```
\usepackage [选项]{geometry}
```

```
\usepackage{geometry}  
\geometry{选项}
```

### ● 常用选项

- `paperwidth`: 纸张宽度, 如 `paperwidth=21cm`
- `paperheight`: 纸张高度, 如 `paperheight=30cm`
- `papersize`: 同时指定纸张宽度和高度, 如 `papersize={21cm,30cm}`
- `textwidth`: 页芯宽度, 如 `textwidth=15cm`
- `textheight`: 页芯高度, 如 `textheight=20cm`
- `body`: 同时指定页芯宽度和高度, 如 `body={15cm,20cm}`
- `left`, `right`: 左右边界, 如 `left=2cm`, `right=2cm`
- `top`, `bottom`: 上下边界, 如 `top=3.5cm`, `bottom=2.5cm`



```
\usepackage[papersize={120mm,160mm}, %
margin={10mm,20mm}]{geometry}
```

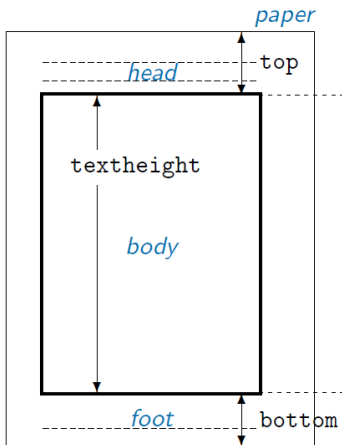
## ● 常用选项

- `hmargin`: 左右边界, 如 `hmargin={2cm,2cm}` 或 `hmargin={2cm}`
- `vmargin`: 上下边界, 用法同 `\hmargin`
- `margin`: `margin={A,B}` 等价于 `hmargin={A,B}, vmargin={A,B}`

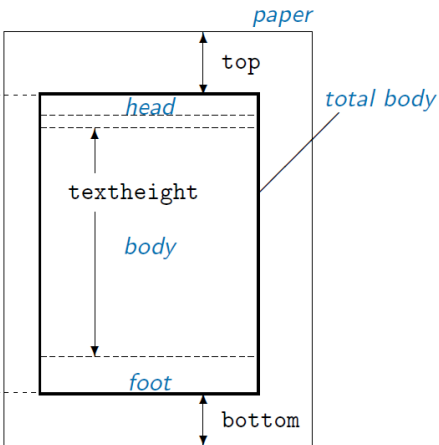
## ● 常用选项

- `includehead`: 将页眉放置在页芯中
- `includefoot`: 将页脚放置在页芯中
- `includeheadfoot`: 将页眉和页脚都放置在页芯中

(a) *default*



(b) `includehead` and `includefoot`





## 页面设置

---

- 常用选项

- `headheight`: 页眉高度, 如 `headheight=0.6cm`
- `headsep`: 页眉与正文之间的距离, 如 `headsep=0.3cm`
- `footskip`: 正文最后一行基线与页脚基线之间的距离
- `footnotesep`: 正文与页脚之间的距离, 如 `footnotesep=0.5cm`
- `nohead`: 取消页眉
- `nofoot`: 取消页脚
- `marginparwidth`: 边注的宽度
- `marginparsep`: 正文与边注之间的距离
- `nomarginpar`: 取消边注
- `columnsep`: 双栏排版时的栏间距
- `hoffset`, `voffset`: 修改 `\hoffset` 和 `\voffset` 的值

- 更多选项参见宏包手册

- ① 页面设置与 geometry 宏包
- ② 页眉页脚与 francyhdr 宏包
- ③ 颜色与 xcolor 宏包
- ④ 链接与 hyperref 宏包
- ⑤ 多栏排版与 multicol 宏包
- ⑥ 横向排版与 lscapc 宏包
- ⑦ 算术运算与 calc 宏包
- ⑧ 文档的分割处理: input 与 include 命令

# 页眉页脚

## ☞ 选择页面版式的命令

```
\pagestyle{版式}
```

### ● 常用版式

- `empty` → 页眉页脚都为空, 不显示页码
- `plain` → `article` 文档类的缺省选项, 页眉为空, 页码位于页脚中间
- `headings` → `book` 类的缺省选项, 由系统指定页眉, 没有页脚
- `myheadings` → 与 `headings` 类似, 但可以定制页眉标题

### ● `headings` 版式中页眉的内容

- `article`: 左端是**节标题**, 右端是**页码**
- `book`: 左页页眉的左端是**页码**, 右端是**章标题**, 右页页眉的左端是**节标题**, 右端是**页码**; 若是单面格式, 则左端是**章标题**, 右端是**页码**
- `\leftmark`, `\rightmark` → 分别存放左页与右页的页眉内容
- `article` 的首页和 `book` 的每一章首页都不显示页眉, 但在底部显示页码

# 页眉页脚

---

## 🔗 myheadings 版式: 定制页眉与页脚

```
\markright{右页页眉标题}  
\markboth{左页页眉标题}{右页页眉标题}
```

- 单面格式中使用 `\markright`, 双面格式中使用 `\markboth`
- book 类文档缺省将页眉内容全部转为大写 (自动调用命令 `\MakeUppercase`)
- 例: 92headart.tex, 92headbook.tex

## 🔗 改变当前页的页面版式: `\thispagestyle{版式}`

## 🔗 空白页的页眉

在 book 类文档中, 默认新章总是从右页开始, 这时可能会造成空白页, 但页眉仍然存在. 若需清除这个页眉, 可在前一章结尾处插入命令

```
\clearpage{\pagestyle{empty}\cleardoublepage}
```

# 页码显式样式

## 指定页码样式

```
\pagenumbering{数字样式}
```

- 可选样式有: `arabic`, `roman`, `Roman`, `alph`, `Alph`
- 页码计数器: `page`
- 每调用一次 `\pagenumbering`, 计数器 `page` 自动置零

## book 类中三个控制版式命令

- `\frontmatter` → 页码采用罗马数字, 章不自动编号, 通常放在前言和目录前面
- `\mainmatter` → 页码切换为阿拉伯数字, 并重设页码计时器, 即从 1 开始计数, 章节自动编号, 通常放在正文主体前面
- `\backmatter` → 不改变页码样式, 也不重设页码计数器, 章不自动编号, 通常放在参考文献和索引前面

## 定制页眉与页脚

### 定制页眉与页脚 → fancyhdr 宏包

```
\usepackage{fancyhdr}
\pagestyle{fancy}
```

- 该宏包将页面分为奇数页(右页)和偶数页(左页)
- 页眉和页脚都被分为三个区域: 左区, 中区, 右区
- 左区居左, 中区居中, 右区居右, 各区内容不能太长, 否则会重叠在一起

### 定制页眉与页脚的命令

```
\fancyhead[选项]{内容} % 设置页眉各部分的内容
\fancyfoot[选项]{内容} % 设置页脚各部分的内容
```

- 选项为字母 L, C, R (代表区域) 和 O, E (代表奇偶页) 的组合  
若选项中不含区域字母, 则表示所有区域  
若选项中不含奇偶页字母, 则表示所有页面
- 用在导言区, 可以设置全文的页眉页脚; 也可以用在正文中, 修改部分页面的页眉页脚

## 定制页眉与页脚

---

### ☞ 简化的页眉页脚命令

```
\thead, \chead, \rhead, \tfoot, \cfoot, \tfoot
```

### ☞ 页眉页脚与正文之间分隔线的粗细

```
\headrulewidth  
\footrulewidth
```

- 需用 `\renewcommand` 命令修改, 缺省值分别为 0.4pt 和 0pt
- 例: 93headfancyart.tex

### ☞ 更一般的页眉页脚命令: `\fancyhf`

- `\fancyhead` → 带 H 选项的 `\fancyhf`
- `\fancyfoot` → 带 F 选项的 `\fancyhf`

### ☞ 在首页显示页眉 (缺省不显示)

在 `\maketitle` 后使用 `\thispagestyle{fancy}`

## 定制页眉与页脚

### ☞ 使用章节标题作页眉

- `\leftmark` 和 `\rightmark` → 缺省的左页与右页页眉标题
- 为了符合中文习惯, 需要修改 `\chaptermark` 和 `\sectionmark` (若是双面格式的 article, 则修改 `\sectionmark` 和 `\subsectionmark`)

```

\fancyhf{}      % 清除原来的页眉与页脚
\renewcommand{\chaptermark}[1]%
  {\markboth{第\CJKnumber{\value{chapter}}章\ #1}{}}
\renewcommand{\sectionmark}[1]%
  {\markright{\thesection\ #1}{}}
\fancyhead[RE]{\leftmark}
\fancyhead[LO]{\rightmark}
\fancyhead[RO,LE]{$\cdots$ \thepage\ $\cdots$}
\renewcommand{\headrulewidth}{0.8pt}
  
```

- 例: 93heahfancybook.tex



- ① 页面设置与 geometry 宏包
- ② 页眉页脚与 francyhdr 宏包
- ③ 颜色与 xcolor 宏包
- ④ 链接与 hyperref 宏包
- ⑤ 多栏排版与 multicol 宏包
- ⑥ 横向排版与 lscapc 宏包
- ⑦ 算术运算与 calc 宏包
- ⑧ 文档的分割处理: input 与 include 命令

# 颜色

## 👉 常见的颜色模式 (color model)

- `gray` → 灰色模式
  - 用  $[0, 1]$  之间的一个数字来表示灰度, 0 表示黑, 1 表示白
- `rgb` → 三基色模式, 自然界的颜色大都是基于三基色模式
  - 用三个  $[0, 1]$  之间的数字, 通过线性组合来表示一个颜色
- `cmymk` → 四分色模式/印刷色彩模式
  - cyan(青), magenta(品红/洋红), yellow(黄), black(黑)
  - 用四个  $[0, 1]$  之间的数字, 通过线性组合来表示一个颜色


















## 👉 在 L<sup>A</sup>T<sub>E</sub>X 中使用彩色 → `xcolor` 宏包

```
\usepackage[选项]{xcolor}
```

- 常用选项有
  - `dvipsnames` → 预定义了 68 种 `cmymk` 色彩
  - `svgnames` → 预定义了 151 种 `rgb` 色彩
  - `x11names` → 预定义了 317 种 `rgb` 色彩

# 颜色

## 可以直接使用的颜色

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

## 更多颜色 (需加 `dvipsnames` 选项)

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

# 颜色

## 使用彩色

`\color{颜色名}` → 声明, 对后面的所有文本起作用  
`\textcolor{颜色名}{对象}` → 命令

- 颜色名 必须是已经有定义的, 如

`{ \color{blue} 这里用蓝色显示 }`  
`\textcolor{red}{这里用红色显示}`

- 使用颜色表达式表示的混合色

`\textcolor{green!75!yellow!50!red}{这里使用颜色表达式}`

颜色表达式

$$\begin{aligned}
 & C_0!P_1!C_1!P_2!\dots!P_n!C_n \\
 &= p_n \cdots p_1 \cdot C_0 + p_n \cdots p_2(1 - p_1) \cdot C_1 + p_n \cdots p_3(1 - p_2) \cdot C_2 \\
 &\quad + \cdots + p_n(1 - p_{n-1}) \cdot C_{n-1} + (1 - p_n) \cdot C_n
 \end{aligned}$$

其中  $P_k \in [0, 100]$  且  $p_0 = 0, p_k = P_k/100, k = 1, \dots, n$

# 颜色

---

## 👉 页面背景色

```
\pagecolor{颜色名} → 声明, 指定页面的背景色
```

## 👉 彩色盒子

```
\colorbox{颜色名}{对象}
```

### ● 带框的彩色盒子

```
\fcolorbox{边框颜色}{背景颜色}{对象}
```

- `\fboxrule` → 边框的宽度
- `\fboxsep` → 盒子边沿与对象之间的距离

## 👉 `\normalcolor`: 恢复正常色

# 颜色

## 使用彩色的另一种用法

```
\color[颜色模式]{颜色定义}  
\textcolor[颜色模式]{颜色定义}{对象}  
\pagecolor[颜色模式]{颜色定义}  
\colorbox[颜色模式]{颜色定义}{对象}  
\fcolorbox[颜色模式]{边框颜色定义}{背景颜色定义}{对象}
```

- 使用颜色模式的例子

```
\textcolor[gray]{0.6}{这里用灰色显示}  
\textcolor[rgb]{0.5,0.3,0.8}{这是 rgb 混合色}  
\pagecolor[cmymk]{0.35,0.15,0.65,0}
```

# 颜色

---

## 自定义颜色

```
\definecolor{颜色名}{颜色模式}{颜色定义}  
\providecolor{颜色名}{颜色模式}{颜色定义}
```

- 如果自定义的颜色名已有定义, `\providecolor` 不起作用, 而 `\definedcolor` 则覆盖原来的定义
- 自定义颜色举例

```
\definecolor{mygray}{gray}{0.75}  
\definecolor{mylightblue}{rgb}{0,0,0.6}
```

- ① 页面设置与 geometry 宏包
- ② 页眉页脚与 francyhdr 宏包
- ③ 颜色与 xcolor 宏包
- ④ 链接与 hyperref 宏包
- ⑤ 多栏排版与 multicol 宏包
- ⑥ 横向排版与 lscapc 宏包
- ⑦ 算术运算与 calc 宏包
- ⑧ 文档的分割处理: input 与 include 命令



# 链接

## ☞ 链接: `hyperref` 宏包

```
\usepackage [选项列表] {hyperref}
```

```
\usepackage [选项列表] {hyperref}  
\hypersetup{参数列表}
```

- 在有交叉引用的地方 (如目录, 书签, 参考文献, 公式引用等) 建立链接
- 提供对外部文件, 互联网网址, 邮件地址的链接命令

## ☞ 常用选项

- 驱动选项: `dvipdfmx`, `dvips`, `pdftex`, `dvips`, `xetex`, ...
- 书签选项
  - `bookmarks` → 创建书签, 缺省为 `true`
  - `CJKbookmarks` → 支持中日韩文字的书签 (中文书签必须带该选项)
  - `bookmarksopen` → 展开所有书签, 缺省只展开章标题
  - `bookmarksnumbered` → 在书签中添加章节编号, 缺省不带编号

# 链接

## 🔗 常用选项

### ● 链接属性

- `colorlinks` → 使用彩色显示链接, 缺省为红色方框
- `linkcolor` → 内部普通链接 (如页码) 的颜色, 缺省为 red
- `citecolor` → 文献引用链接的颜色, 缺省为 green
- `filecolor` → 本地文件链接的颜色, 缺省为 cyan
- `urlcolor` → URL 链接的颜色, 缺省为 magenta
- `linktocpage` → 在目录中用页码做链接, 缺省是用标题
- `breaklinks` → 允许在链接中断行, 缺省不允许

### ● 文件属性

- `pdftitle`, `pdfauthor`, `pdfsubject`, `pdfkeywords`, ...

### ● 打开方式

- `pdfstartview`, `pdfpagemode`, ...

### ● 更多选项参见宏包手册

# 链接

- `hyperref` 宏包举例

```

\usepackage [pdftex] {hyperref}
\hypersetup {CJKbookmarks=true,           % 支持中文书签
              colorlinks=true,           % 使用彩色链接
              citecolor=blue,           % 引用标记颜色
              linkcolor=blue,           % 内部普通链接的颜色
              urlcolor=blue,           % url 链接的颜色
              bookmarksnumbered=true,    % 书签带章节编号
              bookmarksopen=true,       % 书签目录展开
              breaklinks=true,          % 允许在链接处换行
              pdfstartview=FitH        % 文件初始视图
            }
    
```

# 链接

---

 网页链接: `\url` 和 `\href` 命令

```
\url{网址}
```

```
\href{网址}{文本}
```

- `\url` → 生成网址的同时在页面上输出其内容
- `\href` → 生成网址的同时在页面上输出文本的内容
- 例: 94hyperref.tex

- ① 页面设置与 geometry 宏包
- ② 页眉页脚与 francyhdr 宏包
- ③ 颜色与 xcolor 宏包
- ④ 链接与 hyperref 宏包
- ⑤ 多栏排版与 multicol 宏包**
- ⑥ 横向排版与 lscapc 宏包
- ⑦ 算术运算与 calc 宏包
- ⑧ 文档的分割处理: input 与 include 命令

# 双栏排版

## ☞ 双栏排版

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


- 双栏 (多栏) 排版通常用于普通论文的排版
- 双栏排版时, 先排满左栏, 再排右栏
- 相关参数
  - `\columnsep` → 栏间距, 缺省为 10pt
  - `\columnseprule` → 栏间分隔线的粗细, 缺省为 0pt
  - `\columnwidth` → 栏宽, 等于  $(\text{\textwidth}-\text{\columnsep})/2$
- `\onecolumn` → 新起一页, 改为单栏排版

## ☞ 部分文档分栏 (例 95multicol.tex)

```
\twocolumn[通栏文本]
```

- 新起新页, 以两栏格式排版其后面的文本
- **通栏文本** → 在双栏之上, 按单栏格式排版

# 多栏排版

 多栏排版: `multicol` 宏包, 提供多栏环境 `multicols`

```
\begin{multicols}{栏数}[标题][高度]
... ..
\end{multicols}
```

- **栏数**: 可以是 2 到 10
- **标题**: 位于各栏之上, 按单栏排版, 标题与各栏之间不分页
- **高度**: 若当前页面剩余部分小于所设**高度**, 则从下页开始多栏排版
- 使用 `multicol` 宏包, 不用在 `\documentclass` 中加 `twocolumn` 选项

## 2 The User Interface

To use the environment one simply says

```
\begin{multicols}{<number>}
  <multicolumn text>
\end{multicols}
```

where `<number>` is the required number of columns and `<multicolumn text>` may contain arbitrary L<sup>A</sup>T<sub>E</sub>X commands, except

ple, the text you are now reading was started with

```
\begin{multicols}{3}
  [\section{The User
    Interface}] ...
```

If such text is unusually long (or short) the value of `\premulticols` might need adjusting to prevent a bad page

Since narrow columns tend to need adjustments in interline spacing we also provide a `<skip>` parameter called `\multicolbaselineskip` which is added to the `\baselineskip` parameter inside the `multicols` environment. Please use this parameter with care or leave it alone; it is intended only for

# 多栏排版

---

## ☞ 优点

- 自动平衡每栏的高度, 保持每栏文本底部对齐
- 可在同一页中进行单栏和多栏的转换 (不用另起新页)
- 可嵌套在其它环境中, 如 `minipage`
- 各栏中的脚注统一放在页面底部

## ☞ 缺点

- 破坏了  $\text{\LaTeX}$  的浮动体处理机制, 浮动图表不能在多栏内显示, 而是放在之后以单栏页面格式排印, 所以应尽量避免在 `multicols` 环境中使用浮动图表
- 解决方法 → `multicap` 宏包 (详见宏包手册)



- ① 页面设置与 geometry 宏包
- ② 页眉页脚与 francyhdr 宏包
- ③ 颜色与 xcolor 宏包
- ④ 链接与 hyperref 宏包
- ⑤ 多栏排版与 multicol 宏包
- ⑥ 横向排版与 lscape 宏包**
- ⑦ 算术运算与 calc 宏包
- ⑧ 文档的分割处理: input 与 include 命令

## 横向排版

- 通常文档都是纵向排版, 若需要横向排版, 可使用 `landscape` 选项, 它使得全文都横向排版, 但无法实现仅改变某一页或某几页的排版方向
- `lscape` 宏包 → 提供 `landscape` 环境

```
\begin{landscape}  
    ...  
\end{landscape}
```

- $\text{\LaTeX}$  会自动将该环境中的内容按横向排版, 其它部分仍按纵向排版
- 可以实现超宽表格的排版
- 可以跨页, 可以结合 `longtable` 宏包实现超宽超长表格的排版
- 不改变页眉页脚方向

- ① 页面设置与 geometry 宏包
- ② 页眉页脚与 francyhdr 宏包
- ③ 颜色与 xcolor 宏包
- ④ 链接与 hyperref 宏包
- ⑤ 多栏排版与 multicols 宏包
- ⑥ 横向排版与 lscapc 宏包
- ⑦ 算术运算与 calc 宏包**
- ⑧ 文档的分割处理: input 与 include 命令

# 算术运算

## L<sup>A</sup>T<sub>E</sub>X 中的计数器和长度设置命令

```

\setcounter{计数器}{数值}
\addtocounter{计数器}{数值}
\setlength{长度命令}{长度}
\addtolength{长度命令}{长度}
    
```

## calc 宏包 → 提供 数值 和 长度 的加减乘除运算

```

\setcounter{equation}{\value{equation}+3}
\setlength{\textwidth}{\paperwidth-4cm}
    
```


```

\begin{minipage}{\textwidth-2cm}
... ..
\end{minipage}
    
```


- ① 页面设置与 geometry 宏包
- ② 页眉页脚与 francyhdr 宏包
- ③ 颜色与 xcolor 宏包
- ④ 链接与 hyperref 宏包
- ⑤ 多栏排版与 multicol 宏包
- ⑥ 横向排版与 lscapc 宏包
- ⑦ 算术运算与 calc 宏包
- ⑧ 文档的分割处理: input 与 include 命令

## 文档的分割处理

---

 文档的分割处理: 将大文件分成几个小文件


- 主文件 + 子文件

 `\input` 命令: `\input{文件名}`

- 将指定的文件读入到主文件的当前位置
- 若读入的文件是 tex 文件, 可以省略扩展名, 否则要指定全名
- 文件名中可以带路径
- `\input` 命令可以出现在文档的任何部分 (导言区或正文)
- `\input` 命令可以嵌套, 即 input 的文件中可以包含其它文件
- `\listfiles` → 放在导言区, 在屏幕和 log 文件中列出文件清单
- 编译时, 读入所有文件, 再进行编译

## 文档的分割处理

---

 `\include` 命令: `\include{文件名}`

- 相当于 `\clearpage\input{文件名}\clearpage`
- 只能读入 tex 文件
- 只能出现在正文部分
- `\include` 命令不可以嵌套
- 保持有关页面, 章节, 公式编号等相关信息, 保持交叉引用
- 优点: 与 `\includeonly{文件清单}` 配合使用
  - 编译时, 只读入 `\includeonly` 指定的文件清单
  - `\includeonly` 只能放在导言区
  - 若没有 `\includeonly` 命令, 则读入所有文件

```
\documentclass [11pt , a4paper , openany] {ctexbook}
\input{myformat.tex}

\begin{document}
\include{cover.tex}

\frontmatter
\tableofcontents % \listoffigures\listoftables
\include{preface.tex}

\mainmatter
\include{chap01.tex}
... ..

\backmatter
\include{reference.tex}
\addcontentsline{toc}{chapter}{Subject Index}
\printindex

\end{document}
```