

---

## 幻灯片演示

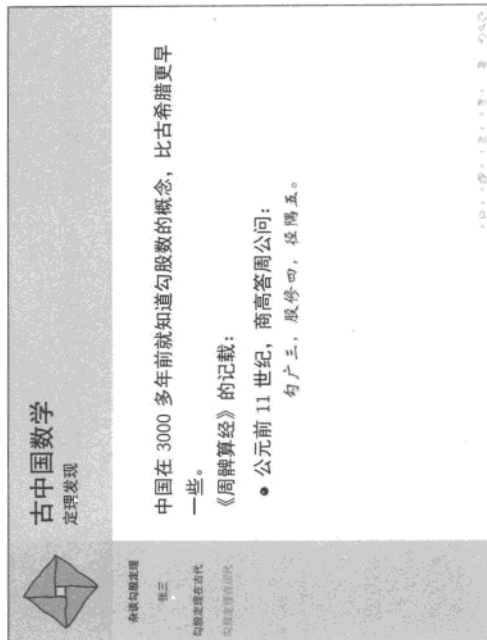
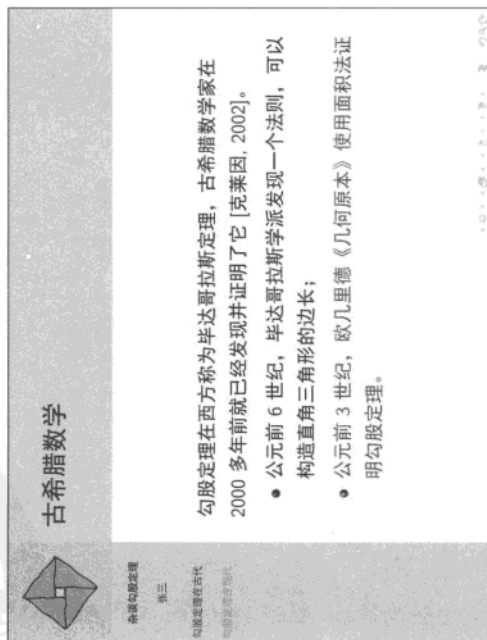
---

作为排版软件， $\text{\LaTeX}$  主要被用来制作书籍和文章。不过，由于现代的  $\text{\LaTeX}$  系统主要以 PDF 文件为输出方式，它被越来越多地用来生成专门的电子文档。授课、演讲用的计算机幻灯片就是  $\text{\LaTeX}$  的一个重要应用。

演示文档与纸面印刷品的要求很不一样，演示文档是输出到计算机屏幕或投影仪上的，通常要求使用大而清晰的字体、鲜艳的色彩、图形化的页面以及交互式的按钮链接。幻灯片的特殊用途同时也使页面成为内容组织的基本单位，文档不仅是按章节，同时也是直接逐页设计安排内容的。这些要求使得原本用来输出论文、书稿的标准文档类不再适用于生产幻灯片，演示幻灯片在  $\text{\LaTeX}$  中也催生了一类专门的工具宏包。

$\text{\LaTeX}$  中专门用来制作幻灯片的工具有很多种，如 `powerdot` 文档类、`prosper` 文档类、`pdfslide` 宏包、`ppower4` 宏包、`pdfscreen` 宏包等。但若说现在最为流行的，大概当属 `beamer` 文档类了。`beamer` 文档类<sup>[253]</sup> 是由 Lübeck 大学理论计算机研究所的 Till Tantau 教授发起的一个专用于幻灯演示的文档类，它以页面（被称为“帧”）为基本组织单位，提供丰富的功能选项和许多预定义的风格主题，支持各种编译程序，使用也相对方便。本章就以 `beamer` 为例，介绍在  $\text{\LaTeX}$  中编写幻灯片的方法。

这一章我们将继续 1.2 节的例子，把有关勾股定理的小短文改编成一个演讲幻灯片（413–415 页）。这里粗略的介绍并不打算代替 `beamer` 原有的详细文档和示例，Tantau et al. [253] 永远是使用 `beamer` 最重要的参考。



## 古中国数学

定理发现



余弦勾股定理

第三

勾股定理在近代

勾股定理在近代

中国在 3000 多年前就知道勾股数的概念，比古希腊更早一些。

《周髀算经》的记载：

- 公元前 11 世纪，商高答周公问：  
勾广三，股修四，弦隅五。
- 又载公元前 7-6 世纪陈子答荣方问，表述了勾股定理的一般形式：  
若求邪至日者，以日下为勾，日高为股，勾股各自乘，并而开方除之，得邪至日。

勾股定理在近代

## 现代叙述



余弦勾股定理

第三

勾股定理在近代

勾股定理在近代

定理 (勾股定理)

直角三角形斜边的平方等于两直角边的平方和。

勾股定理在近代

## 古中国数学

定理证明



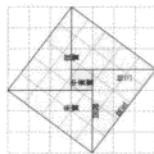
余弦勾股定理

第三

勾股定理在近代

勾股定理在近代

有论者认为早在公元前 11 世纪商高即已证明勾股定理 [曲安京, 1998]。完整的证明见于三国时 (公元 3 世纪) 赵爽对《周髀算经》的注释。



图：赵爽的弦图可给出勾股定理的一个富于对称美的证明

勾股定理在近代

## 现代叙述



余弦勾股定理

第三

勾股定理在近代

勾股定理在近代

定理 (勾股定理)

直角三角形斜边的平方等于两直角边的平方和。

可以用符号语言表述为：设直角三角形  $ABC$ ，其中  $\angle C = 90^\circ$ ，则有

$$AB^2 = BC^2 + AC^2. \quad (1)$$



勾股定理在近代



## 勾股数

角道勾股定理

第三

勾股定理的推广

勾股定理的现代

满足式 (1) 的整数称为勾股数。第 1 节所说毕达哥拉斯学派得到的三元数组就是勾股数。

直角边 $a$	直角边 $b$	斜边 $c$
3	4	5
5	12	13
7	24	25
8	15	17

表: 较小的几组勾股数

角道勾股定理 第三 勾股定理的推广 勾股定理的现代



## 参考文献

角道勾股定理

第三

勾股定理的推广

勾股定理的现代

- 克莱因 (2002).  
古今数学思想.  
上海科学技术出版社.
- 曲安京 (1998).  
商高、赵爽与刘徽关于勾股定理的证明.  
数学传播, 20(3).
- 矢野健太郎 (1986).  
几何的有名定理.  
上海科学技术出版社.

角道勾股定理 第三 勾股定理的推广 勾股定理的现代

## 6.1 组织幻灯片内容

使用 `beamer` 类与编写一般的  $\text{\LaTeX}$  文档的代码结构是一样的, 我们同样可以写出这样的代码框架:


```
\documentclass{beamer}
\begin{document}
% ...
\end{document}
```

要在 `beamer` 类中使用中文, 则可以使用 `ctex` 或 `ctexcap` 宏包, 前者只引入必要的中文, 后者还会翻译图表等环境名称<sup>①</sup>。

```
\documentclass{beamer}
\usepackage[UTF8,noindent]{ctexcap}
\begin{document}
% ...
\end{document}
```

6-1-1

这里 `noindent` 选项用来阻止 `ctex` 宏包引入的段前缩进, 这也是 `beamer` 文档类本身的默认设置, 在幻灯片中的段落通常不使用首行缩进。

 因为旧版本 (3.17 以前) `beamer` 的一个实现问题, 使用  $\text{\XeLaTeX}$  编译 `beamer` 文档时页面底部的导航按钮会失效。因此使用中文或者系统字体的  $\text{\XeLaTeX}$  用户最好及时更新到 `beamer` 的最新版本。如果不方便更新, 也可以在文档中使用如下代码临时处理:

```
\makeatletter
\def\beamer@linkspace#1{%
  \begin{pgfpicture}{0pt}{-1.5pt}{#1}{5.5pt}
    \pgfsetfillopacity{0}
    \pgftext[x=0pt,y=-1.5pt]{.}
    \pgftext[x=#1,y=5.5pt]{.}
  \end{pgfpicture}}
\makeatother
```

6-1-2

<sup>①</sup> 如果使用  $\text{\pdfTeX}$  等传统编译方式及 GBK 编码, 使用时有一些限制, 如在导言区不能使用汉字, 全局的中文字体设置也不能放在导言区。

### 6.1.1 帧

如果你现在就在例 6-1-1 中填写内容, 编译后就可以得到一些很小的页面, 不过如果你把 `beamer` 类当成普通的文档类填写, 会发现效果很不好, 而且章节标题等内容并不会显示出来。事实上, 与普通的文档不同, 幻灯片的内容是一页一页在大屏幕上放出的 (偶尔也在计算机屏幕上观看), 在演讲中, 一页幻灯片通常就表示一个语义单元, 因此在 `beamer` 中组织内容也是一页一页完成的, 这样的演示页称为帧 (frame)。

在 `beamer` 中, 帧用 `frame` 环境得到。一帧里面的内容可以使用各种常用的  $\text{\LaTeX}$  命令和环境。在帧里面, 内容有一定的水平边距, 并且整体垂直居中显示, 例如:

```
\begin{frame}
这是简单的一帧。

帧里的内容是垂直居中的。
\end{frame}
```

6-1-3

幻灯版的每帧通常都有一个小标题, 用来说明这一帧的主要内容, 这可以使用 `\frametitle` 命令得到。`beamer` 甚至还提供了小标题的命令, 可以使用 `\framesubtitle` 得到。例如:

```
\begin{frame}
  \frametitle{标题}
  \framesubtitle{小标题}
  这是简单的一帧。
\end{frame}
```

6-1-4

不过, `beamer` 还提供了更为方便的方式, 即直接在 `frame` 环境后用花括号 `{}` 括起来的参数就表示帧的标题, 后面的第二个参数就是帧的小标题。`frame` 环境的这种参数是可选的, 这种语法与大多数  $\text{\LaTeX}$  环境不同, 但非常有效, 例如:

```
\begin{frame}{标题}{小标题}
  这是简单的一帧。
\end{frame}
```

6-1-5

`beamer` 重定义了大量  $\text{\LaTeX}$  环境, 使它们的格式更适合在幻灯版中显示, 其中最明显的就是列表环境和定理类环境, 尽管它们使用起来并没有多少区别。

于是, 我们幻灯片中关于古中国数学的一帧就可以这样得到, 这与在 `ctexart` 中排版普通的文章完全一样 (见图 6.1):

## 古中国数学

### 定理发现

中国在 3000 多年前就知道勾股数的概念，比古希腊更早一些。

《周髀算经》的记载：

- ▶ 公元前 11 世纪，商高答周公问：

勾广三，股修四，径隅五。

- ▶ 又载公元前 7-6 世纪陈子答荣方问，表述了勾股定理的一般形式：

若求邪至日者，以日下为勾，日高为股，勾股各自乘，并而开方除之，得邪至日。

◀ ◁ ▢ ▣ ▤ ▥ ▦ ▧ ▨ ▩ ▪ ▫ ▬ ▭ ▮ ▯ ▰ ▱ ▲ △ ▴ ▵ ▶ ▷ ▸ ▹ ► ▻ ▼ ▽ ▾ ▿ ▸ ▹ ► ▻ ▼ ▽ ▾ ▿ ▸ ▹ ► ▻ ▼ ▽ ▾ ▿

图 6.1 基本的 frame 环境

```

1 \begin{frame}{古中国数学}{定理发现}
2 中国在 3000 多年前就知道勾股数的概念，比古希腊更早一些。
3
4 《周髀算经》的记载：
5 \begin{itemize}
6 \item 公元前 11 世纪，商高答周公问：
7 \begin{quote}
8 勾广三，股修四，径隅五。
9 \end{quote}
10 \item 又载公元前 7--6 世纪陈子答荣方问，表述了勾股定理的一般形式：
11 \begin{quote}
12 若求邪至日者，以日下为勾，日高为股，勾股各自乘，并而开方除之，得邪至日。
13 \end{quote}
14 \end{itemize}

```

```
15 \end{frame}
```

6-1-6



## 练习

**6.1** 试写出其他幻灯片中的其他几帧，看看在 beamer 中的原来的文档格式会发生哪些变化？

**6.2** beamer 与 TikZ 的原作者同为 Till Tantau 教授，实际上早期版本的 pgf 正是 beamer 的一部分，试使用 TikZ 画出幻灯片定理中的直角三角形图。

### 6.1.2 标题与文档信息

beamer 提供了比标准文档类更为丰富的标题命令。设置标题信息的命令有：

<code>\title</code>	设置标题。命令可以带有一个可选参数，用来设置标题的短形式，短形式可能会出现在帧的顶部或底部，例如：  <code>\title[勾股定理]{勾股定理的历史、现状和对现代数学的影响}</code>
<code>\subtitle</code>	设置小标题。小标题一般会在标题下方以较小的字号显示。可以带一个可选参数设置短形式，这一项往往不设置。
<code>\author</code>	设置作者。可以带一个可选参数设置短形式。
<code>\institute</code>	设置作者所在的学院等机构。可以带一个可选参数设置短形式，也可以不设置。
<code>\date</code>	设置日期。可以带一个可选参数设置短形式，如果不设置，默认使用编译时的日期。
<code>\titlegraphic</code>	设置标题图形。可以使用 <code>\includegraphics</code> 插入较小幅的图案，通常不设置。

使用上述命令设置的标题、作者等信息同时也会记录到输出的 PDF 文件说明信息中，可以使用 Adobe Reader 等软件查看 PDF 信息（见图 6.2）。除了标题和作者，beamer 还提供了 `\subject` 和 `\keywords` 两个命令，用来设置 PDF 说明信息中的主题和关键字。

例如，我们关于勾股定理的幻灯片的标题信息，就可以使用下面的命令在导言区设置：

```
1 % beamer 导言区
```



```

2 \title{杂谈勾股定理}
3 \subtitle{数学史讲座之一}
4 \institute{九章学堂}
5 \author{张三}
6 \date{\today}
7 \subject{勾股定理}
8 \keywords{勾股定理, 历史}

```

6-1-7

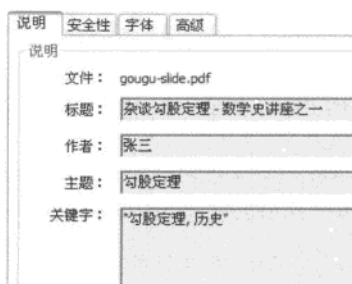


图 6.2 在 Adobe Reader 中查看 beamer 生成的 PDF 文件说明信息

而要输出标题, 则可以在 (空白的) 一帧里面使用 `\titlepage` 命令。`\maketitle` 也可以用来输出标题, 它在帧里面和 `\titlepage` 的功能相同, 如果不在帧里面, 则相当于在一空白的帧里面使用:

```

% 等价于 \maketitle
\begin{frame}
\titlepage
\end{frame}

```

例如对例 6-1-7 中的例子, 使用 `\maketitle` 就得到图 6.3。当然, 使用 `\titlepage` 时可以在整个文档的标题帧中增加一些其他的补充内容。

### 6.1.3 分节与目录

在 beamer 中可以使用 `\section`、`\subsection`、`\subsubsection` 以及 `\part` 命令对文档分节, 或使用 `\tableofcontents` 命令产生目录, 这与标准文档类一样。因此, 我们只有两节的幻灯片就可以这样简单地划分:

```
\section{勾股定理在古代}
```



图 6.3 beamer 的标题帧

```
...  
\section{勾股定理在现代}  
...
```

与普通的文档不同，在 beamer 中，分节命令并不会输出一行标题。单纯地使用分节命令，甚至在默认的格式下什么也看不到。分节的标题会在目录中显示，如果改变 beamer 的输出风格（参见 6.2 节），顶部或侧面的导航条也可能显示当前或全部章节的标题。我们目标中勾股定理的幻灯片就在每帧的左侧显示有文档的分节。

`\tableofcontents` 命令也必须放在帧里显示，因此我们的幻灯片就要这样写：

```
\begin{frame}{目录}  
  \tableofcontents  
\end{frame}  
  
\section{勾股定理在古代}  
...
```

`\part` 命令本身也不产生标题，但 beamer 提供了一个 `\partpage` 命令，它与 `\titlepage` 类似，可以在一帧中产生文档某部分的标题，例如：

```
\part{引言}  
\begin{frame}
```

```
\partpage
\end{frame}
```

6-1-8

幻灯片文档一般都不会很长，以一次完整的演讲报告的长度为宜，因此一个 beamer 文档通常控制在几十帧的篇幅，使用 `\part` 和 `\section` 至 `\subsubsection` 的命令分节，就足够划分一个演讲或报告了。如果是分为多次讲解的长篇报告，或是教师的课程讲义，则应该为每一讲单独制作一个幻灯片。但如果希望把所有内容放进一个单独的文件（例如为了方便分发或支持交叉引用），则可以使用 `\lecture` 命令进行作为更高层次的内容划分：

```
\lecture{杂谈勾股定理}{gougu} % 讲座标题和讲座标签
```

6-1-9

`\lecture` 命令本身不产生任何标题和效果，beamer 提供 `\insertlecture` 命令向文档中插入 `\lecture` 的标题。可以通过 `\AtBeginLecture` 在每一讲前面都添加一帧，代替 `\maketitle`：

```
% 导言区
\AtBeginLecture{
  \begin{frame}
    \Large
    本周论题：\insertlecture
  \end{frame}
}
```

6-1-10

`\lecture` 的另一个用途是使用 `\includeonlylecture{(标签)}` 来使文档编译时只输出选定标签的那一讲。这样，通常修改标签，就可以只使用一个文件来分别生成每一讲单独的幻灯片了。

beamer 中的 `\tableofcontents` 可以在可选参数中使用许多参数控制其格式，例如 `currentsection` 选项就可以只显示当前一节的目录结构，`currentsubsection` 选项则控制只显示当前一小节的目录结构。这对较长的幻灯片是非常有用的，演讲时可能需要在每一节的开头都显示一下即将讲到的内容结构，因而每一节前面都应该有一个小目录，特别是那些缺少导航条显示分节标题的格式更是如此。

为了方便在每一节前面增加一个目录，beamer 还提供了 `\AtBeginSection`、`\AtBeginSubsection` 命令，它们的用法与 L<sup>A</sup>T<sub>E</sub>X 原有的 `\AtBeginDocument` 命令功能类似，用来给每一节或每一小节前面增加一段代码（而 `\AtBeginDocument` 则在整个文档前面添加一段内容，它一般用在宏包制作），例如：

```
% 引言区
\AtBeginSection[] { % 空的可选项表示 \section* 前不加目录
  \begin{frame}{本节提要}
    \tableofcontents[currentsection]
  \end{frame}
}
```

6-1-11



## 练习

**6.3** 本章关于勾股定理的幻灯片太过短小了，所以不能显示出复杂的分节层次。试编写一个复杂讲座的提纲，在整个文档和每一节前面添加目录。

### 6.1.4 文献

在 **beamer** 中添加文献列表与普通文档的语法没有多少差别，不过与书面的文稿不同，在幻灯片中通常并不适合使用特别冗长的文献列表，列出的条目不宜过多，也不需要文献编号这类听众难以记忆的内容。因此，**beamer** 中的文献列表，文字排列比较宽松，列表前没有标题，默认格式也是没有编号的。

在我们的例子中，参考文献同样是通过 **BiBTeX** 生成的，插入文献的代码是（此文献数据库参见 1.2.7 节）：

```
\begin{frame}{参考文献}
\nocite{Shiye}
\bibliography{math}
\end{frame}
```

6-1-12

即在单独的一帧中加入 `\bibliography` 命令。幻灯片在演讲中很难前后跳跃翻页，对文献进行编号引用用途不大，所以往往需要使用 `\nocite` 命令，指明需要列入文献列表的条目。

如果确实需要对文献进行引用，那么最好使用作者年代的引用方式，避免完全不直观的数字编号可能造成的问题。由于 **beamer** 的文献格式比较特殊，它并不支持 **natbib** 宏包及其对应的 `.bst` 文献格式，为此，可以使用相对简单的 **apalike** 文献格式，它按照美国心理协会（**APA**）的格式，提供了基本的作者年代引用方式：

```
% 提供简单的作者年代引用格式
\bibliographystyle{apalike}
```

6-1-13



## 练习



**6.4** biblatex 与 beamer 没有明显的冲突，可以在 beamer 中使用。试参考 biblatex 自带的示例，以 authoryear 格式选项处理 beamer 的文献，并给出一个例子。

### 6.1.5 定理与区块

在 beamer 中，已经预定义了许多定理类环境：theorem, corollary, definition, definitions, fact, example 以及 examples，它们都以英文名称给出，例如 theorem 环境的名称就是 “Theorem”。由于 beamer 调用了 amsthm 宏包<sup>[10]</sup> 定制定理类环境格式，因此也有用于证明的 proof 环境（参见 2.2.4 节）。不过我们需要的是中文定理环境，则可以使用 \newtheorem 另行定义，如：

6-1-14

```
\newtheorem{thm}{定理}
```

证明环境则需要进行汉化：

6-1-15

```
\renewcommand\proofname{证明}
```

在 beamer 中，定理环境的结果是一个彩色块，在不同的主题（参见 6.2.1 节）下可能会有丰富的效果。例如，在本章例子相似的主题下<sup>①</sup>，下面定理环境的效果见图 6.4：

6-1-16

```
\begin{frame}{现代叙述}
\begin{thm}[勾股定理]
直角三角形斜边的平方等于两直角边的平方和。
\end{thm}
\end{frame}
```

类似定理环境的这种彩色框效果，在 beamer 中还有其他的区块环境，可以用于强调一部分内容。

block, alertblock 和 exampleblock 环境就是 beamer 定义的三种区块环境，它们除了使用的配色不同外，用法和结果都大致相同，例如（见图 6.5）：

```
\begin{frame}
\begin{block}{块标题}
这是一个区块
\end{block}
\end{frame}
```

<sup>①</sup> PaloAlto 主题改用 seagull 的灰度色彩。

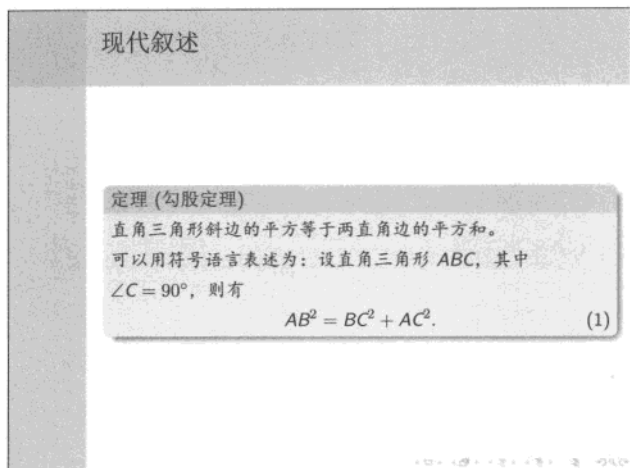


图 6.4 beamer 中的定理环境

```
\begin{block}{}% 无标题
这是另一个区块
\end{block}
\end{frame}
```

6-1-17

## 6.1.6 图表

在 beamer 中使用图表与在普通文档中的语法并无区别。不过，beamer 是按帧组织内容的，位置固定，因此 figure 和 table 环境不再是浮动的环境，而只用来区别标题。例如在我们勾股定理例子中的一页就写成：

```
\begin{frame}{古中国数学}{定理证明}
有论者认为早在公元前 11 世纪商高即已证明勾股定理\cite{quanjing}。
完整的证明见于三国时（公元 3 世纪）赵爽对《周髀算经》的注释。
\begin{figure}
\centering
\includegraphics[height=0.4\textheight]{xiantu.pdf}
\caption{赵爽的弦图可给出勾股定理的一个富于对称美的证明}
\end{figure}
```

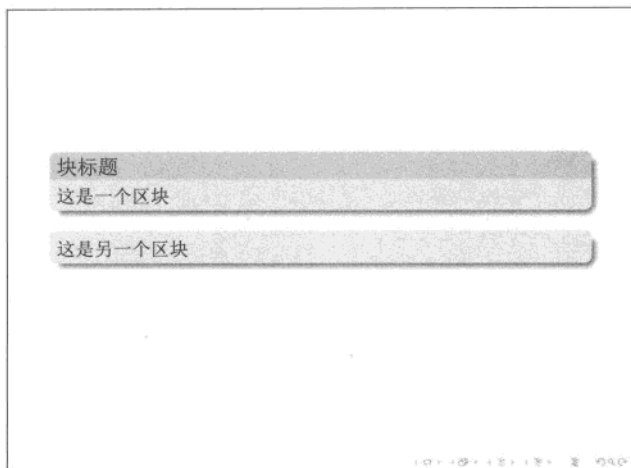


图 6.5 beamer 中的 block 环境

6-1-18

`\end{frame}`

除了标准的图表环境，beamer 还提供了一个 `\logo` 命令，把一个较小的图标放在幻灯片的角落里面，可以用它来放置校徽、公司商标等内容。`\logo` 命令一般放在导言区，如在勾股定理的例子中有：

6-1-19

`\logo{\includegraphics{logo.pdf}}`

这里 `logo.pdf` 是一个长宽为 1.5 cm 的小图形，它实际是用 Asymptote 语言画出来的。

beamer 内部使用 `pgf` 宏包绘制定理边框、幻灯片按钮等图形。因此，如果需要一些简单的数学图形，使用基于 `pgf` 的 `tikz` 宏包直接画图是最方便的。我们在勾股定理的例子中，直角三角形就是用 `tikz` 画的：

6-1-20

```
% 在 thm 环境最后：
\begin{center}
\begin{tikzpicture}[scale=0.5,font=\small]
\draw[thick] (0,0) node[left] {$A$}
-- (4,0) node[right] {$C$}
-- (4,3) node[above right] {$B$} -- cycle;
\draw (3.5,0) |- (4,0.5);
\end{tikzpicture}
\end{center}
```

幻灯片中的表格务求简明清晰，因此使用 5.4.1 节介绍的表行色彩相间的彩色表格就十分合适。`xcolor` 宏包的 `\rowcolors` 命令需要给宏包加 `table` 选项，但由于 `beamer` 本身就会载入 `xcolor`，因此宏包选项使用如下方式传递：

```
\documentclass[xcolor=table]{beamer}
```

6-1-21

于是勾股定理例子中的表格就可以使用下面的代码得到：

```
% 颜色 craneorange 是在 crane 色彩主题中定义的
\rowcolors{2}{craneorange!25}{craneorange!50}
\begin{tabular}{rrr}
\rowcolor{craneorange}直角边 $a$ & 直角边 $b$ & 斜边 $c$\\
3 & 4 & 5 \\
5 & 12 & 13 \\
7 & 24 & 25 \\
8 & 15 & 17 \\
\end{tabular}
```

6-1-22

## 6.2 风格的要素

### 6.2.1 使用主题

修改 `beamer` 幻灯片格式的基本方式就是使用主题 (theme)。`beamer` 提供了二十多种不同风格的幻灯片主题，可以使用 `\usetheme` 命令选择。例如，要得到本章例子中的整体效果，就只要在导言区使用

```
\usetheme{PaloAlto}
```

6-2-1

预定义的主题有：`default`、`AnnArbor`、`Antibes`、`Bergen`、`Berkeley`、`Berlin`、`Boadilla`、`boxes`、`CambridgeUS`、`Copenhagen`、`Darmstadt`、`Dresden`、`Frankfurt`、`Goettingen`、`Hannover`、`Ilmenau`、`JuanLesPins`、`Luebeck`、`Madrid`、`Malmoe`、`Marburg`、`Montpellier`、`PaloAlto`、`Pittsburgh`、`Rochester`、`Singapore`、`Szeged`、`Warsaw` 等。这些主题大多是以作者游历的一些地点命名的，其中 `default` 是默认的主题。

实际上，`beamer` 的主题是由不同的内部主题 (inner theme)、外部主题 (outer theme)、色彩主题 (color theme)、字体主题 (font theme) 等组合而成的，可以分别使用 `\useinnertheme`、`\useoutertheme`、`\usecolortheme`、`\usefonttheme` 选择。本章的例子在使用了 `PaloAlto` 主题后，又把冷色调换成了暖色调，用了如下命令：



6-2-2

```
\usecolortheme{crane}
```

在这里，内部主题主要控制的是标题页、列表项目、定理环境、图表环境、脚注等在一帧之内的内容格式。预定义的内部主题有 default、circles、rectangles、rounded、inmargin 等。

外部主题主要控制的是幻灯片顶部尾部的信息栏、边栏、图标、帧标题等一帧之外的格式。预定义的外部主题有 default、infolines、miniframes、smoothbars、sidebar、split、shadow、tree、smoothtree 等。

色彩主题控制各个部分的色彩。预定义的色彩主题包括 default、albatross、beaver、beetle、crane、dolphin、dove、fly、lily、orchid、rose、seagull、seahorse、sidebartab、structure、whale、wolverine 等。

字体主题则控制幻灯片的整体字体风格。预定义的 beamer 字体主题包括 default、professionalfonts、serif、structurebold、structureitalicserif、structuresmallcapsserif 等。其中默认字体主题 default 的效果是整个幻灯片使用无衬线字体，这是多数幻灯片的选择；serif 主题则改用衬线字体，不过此时最好使用较大的字号和较粗的字体；professionalfonts 不对字体有特别的设置，需要使用另外专门的宏包进行设置；structure 开头的几个主题则对 beamer 中的几个结构有特别设置。这里举一个调用 arev 宏包的例子，这个宏包是专门为制作幻灯片设计的无衬线字体包，对正文字体和数学字体都有详细的调整，因此不需要对 beamer 做额外的设置：

6-2-3

```
\usebeamerfonttheme{professionalfonts}
\usepackage{arev}
```

不同主题的效果可以参见 beamer 手册 Tantau et al. [253]，里面对所有主题以图例的方式列出了效果。有的主题还带有一些选项，也可以在文档中得到详细说明。



## 练习

6.5 查看手册 Tantau et al. [253]，beamer 提供了哪些选项设置文档整体的字号大小？默认值是哪个？

### 6.2.2 自定义格式

尽管 beamer 已经提供了许多预定义的主题，也还是有一些格式需要单独进行设置。对于我们的例子来说，最明显的是关于中文的设置，配合无衬线的西文字

体, 中文也需要使用黑体等在投影仪上清晰可辨的字体。在较新的 **xeCJK** 中可以重定义 `\CJKfamilydefault` 为 `\CJKsfdefault`, 设置中文默认字体为无衬线族 (参见 2.1.3.2 节):

```
% XeLaTeX 编译
\usepackage[noindent,UTF8]{ctexcap}
\setCJKsansfont[ItalicFont={华文新魏}]{黑体}
\renewcommand\CJKfamilydefault{\CJKsfdefault}
```

6-2-4

也可以直接用 `\setCJKmainfont` 定义中文字体, 这对没有 `\CJKfamilydefault` 的旧版本 **xeCJK** 也适用:

```
% XeLaTeX 编译, 可适用于较旧版本的 xeCJK
\usepackage[noindent,UTF8]{ctexcap}
\setCJKmainfont[ItalicFont={华文新魏}]{黑体}
```



**X<sub>ε</sub>LaTeX** 下面 `fontspec` 宏包被调用, 可能会影响数学字体, 使 **beamer** 设置的部分无衬线体数学字体失效。此时需要单独用 `\setmathrm` 等命令进行设置数学字母的字体, 或给 **beamer** 加上 `[no-math]` 的全局选项传递给 `fontspec` (参见 2.1.3.2 节), 例如:

```
\documentclass[no-math]{beamer}
\usepackage[noindent,UTF8]{ctexcap}
\setCJKsansfont[ItalicFont={华文新魏}]{黑体}
\renewcommand\CJKfamilydefault{\CJKsfdefault}
```

6-2-5

除了中文设置, **beamer** 自己也提供了自己独有的设置命令。

**beamer** 使用一种模板 (template) 机制, 将幻灯片的不同内容组件格式抽象为模板代码、模板字体、模板色彩。模板代码是实现组件的具体代码, 例如 `itemize` 列表项的模板代码就包含大量 `pgf` 的绘图命令, 用来画出具有很炫效果的项目来。

在 **beamer** 中, 我们使用 `\setbeamercolor`、`\setbeamerfont` 和 `\setbeamertheme` 来分别设置不同部分组件的色彩、字体和模板的具体实现代码。修改模板具体实现代码往往会很复杂, 不过 `\setbeamertheme` 命令可以从多个预定义的模板中选择一个出来。设置组件单独的色彩与字体也相当实用, 例如对 `itemize` 列表, 可以设置:

\*本节后面内容首次阅读可略过。

```
\setbeamertemplate{itemize items}[circle]
\setbeamercolor{itemize item}[fg=black]
\setbeamercolor{itemize/enumerate body}{fg=gray}
\setbeamerfont{itemize/enumerate body}{family=\rmfamily}
```

6-2-6

这样就设置了列表项的符号是一个黑色的圆形，同时列表内的内容是 `\rmfamily` 的灰色文字。

实际上，`beamer` 主题实现中的多数代码就是由这几个设置命令组成的，你也可以使用类似的方式定义自己的主题格式。



## 练习

**6.6** 简单的样式修改可以通过组合现有的主题和背景插图来完成，但对 `beamer` 进行更详细的样式定制时，字体、色彩与模板名目繁多，可能需要参考在 `beamer` 安装目录下 `theme/` 目录中的模板源代码，结合文档说明，才能顺利完成。

`beamer` 的每个主题对应一个后缀为 `.sty` 的主题文件，使用命令 `\usetheme` 或 `\useinnertheme` 等切换主题，其实就是载入主题文件。查看文档 Tantau et al. [253] 或 Kim [122]，每类主题文件的文件名规则是什么？模仿已有的主题，试编写一个自己的主题文件。

**6.7** 使用 `\setbeamertheme` 命令也可以直接设置模板代码。试修改帧背景 `background` 的模板实现代码，给幻灯片添加背景图片。



## 内容与格式分离

$\text{\LaTeX}$  的一个重要的想法是将内容与格式分离。对于一个焦急的投稿人，他可以只写 `article` 文档类，给出 `\title`，标出 `\section`、`\subsection`，或者插入一个 `table` 环境，而不必关心标题（`title`）应该居中还是左对齐，小节（`subsection`）应该用粗体还是斜体。而在  $\text{\TeX}$  的历史中，情况并非如此。高德纳最早给出的 Plain  $\text{\TeX}$  格式中，作者不仅仅需要考虑文档的内容，还需要处理对齐、位置、字体、字号等设计师的工作，甚至了解断行、分页等更深入的排版知识。Plain  $\text{\TeX}$  并没有预先定义标题、章节或是浮动图表的命令，逐一为不同的内容定义不同的格式命令相当烦琐无味，而且不同的文档之间也无法统一，因此很容易吸引人写出充斥着各种低级格式控制命令的文档来。相比之下，“傻瓜式”的  $\text{\LaTeX}$  自一问世，就立即吸引了来自学术界的许多用户，成为最为流行的  $\text{\TeX}$  格式。

这种内容与格式分离的想法，鼓励人们在正文中只写与文档的逻辑结构和语义相关的控制命令，而不写任何直接的具体格式设置。这样，同样的一篇数学论文，作者使用 `article` 文档类写草稿，再换用 `amsart` 向美国数学会提交正式稿件，只需要修改 `\documentclass` 的一个参数，而不必在正文中作任何修改。而如果希望对文档格式有进一步的变更，也可以在导言区进行全局的格式设置。

例如， $\text{\LaTeX}$  的第一个广为使用的版本的  $\text{\LaTeX}$  2.09，它的导言区如：

```
\documentstyle[11pt,psfig]{article}
```

其中，`article`（文件是 `article.sty`）指出总的文档格式，选项 `11pt` 是 `article` 的选项，标明整体字号，而 `psfig` 则调用宏包文件 `psfig.sty` 来支持 PostScript 图形。

$\text{\LaTeX}$  2.09 的这种格式只能同时使用很少的格式宏包，而随着第三方宏包的不断增多，在  $\text{\LaTeX}$  2<sub>ε</sub> 中就区分了 `\documentclass` 与 `\usepackage`，为选用更为丰富的宏包提供支持，同时也增加了许多用户命令以更好地支持文档结构划分、宏定义、字体选择等等 [143]。这就是我们今天所见到的  $\text{\LaTeX}$ 。

在实际使用中，内容与格式分离的做法受限于  $\text{\LaTeX}$  提供的语义结构数量和格式控制能力。如果  $\text{\LaTeX}$  没有提供足够多的语义结构，例如“题注”，那么用户就必须自己定义一个生成题注的命令，或者寻找提供题注功能的宏包，但这就难以保持一致性和分离特征；另外，如果  $\text{\LaTeX}$  对已有语义结构的格式控制能力不足，如没有设置脚注字体的接口，那么用户也必须自己重定义 `\footnote` 的底层实现，或者寻找相关宏包。更为严重的问题是，不同的宏包以及用户自己的定义都可能不兼容，导致无法同时使用。

$\text{\LaTeX}$  2<sub>ε</sub> 内核太小，功能不够，特别是内容编写与格式设置的关系仍不够明晰，这正是已经有近 20 年历史的  $\text{\LaTeX}$  2<sub>ε</sub> 格式的缺憾，也逐渐成为制约  $\text{\LaTeX}$  进一步发展的重要因素。在这里，`beamer` 文档类无疑提供了一种更为细致的三层抽象界面：

- 对幻灯片作者，只需要了解帧、定理、列表、图表、文献、强调这些与演示内容相关的概念和命令，就可以完成演示稿件，在格式方面也只需要选择现成的主题（`theme`）。
- 对幻灯片模板设计者或者主题的编写人员，则无须考虑具体的演示内容，只要使用 `\setbeamertheme`、`\setbeamerfont`、`\setbeamercolor` 等命令，选择每个表意成分的基本模板，并调整字体、彩色等参数。

- 对主题的高级编写人员或者说  $\text{\LaTeX}$  编程人员，则可以具体定义每个表意成分的模板，定义可调整的参数，直接组织每个组件的细节结构。

这种三层式的界面，自然地把  $\text{\LaTeX}$  用户分成了“作者/设计师/程序员”三个层次，这三个部分的人各司其职，把内容编写、格式设计、具体实现彻底分离开来。

事实上，下一代的  $\text{\LaTeX}3$  格式也使用了与 beamer 类似的分层组织结构，其成果就是 xtemplate 宏包<sup>[260]</sup>。xtemplate 宏包细致地区分了模板的接口与实现，以及模板的实例使用，从而也将内容编写、格式设计、与代码实现分离开来，读者可通过 xfrac<sup>[111]</sup> 宏包的实现一窥其中奥妙。

在  $\text{\LaTeX}$  中，应该尽量遵循内容与格式分离的原则，对于作者来说这能让人把更多精力集中在文档内容上，同时也会使文档的修改和移植更加方便。特别是篇幅长、内容层次多的文档，更应该注意使用抽象的样式。

当然，内容与格式分离的写法也不是没有缺点。由于抽象层次增多，beamer 的实现代码要比通常的  $\text{\LaTeX}$  文档类复杂得多，如果需要修改已有组件的基本格式，就要修改模板实现，这需要了解 beamer 的组件模板和参数机制，特别是不同组件之间的关系，初次使用会比较困难。如果编写的只是一些格式很简单甚至缺乏逻辑关系的内容，那么内容与格式分离的做法就会显得过于烦琐了。

## 6.3 动态展示

尽管滥用动画效果可能会使观众反感（特别是学术演讲），动态效果仍然是幻灯片中不可或缺的部分。适当地使用动态功能可以强化沟通的效果，这里我们将看到 beamer 中几种简单的动态效果。

### 6.3.1 覆盖浅说

覆盖（overlay）是最为基本的一种幻灯片效果。严格地说，它甚至不是“动态”效果，而只是把同一帧幻灯片的不同内容按一定的次序拆分成几页显示出来。

利用覆盖可以让内容逐步显示，例如让一个列表项分成几步显示出来，也可以让不同的内容依次代替，产生类似动画的效果。

逐步显示是最为常用的覆盖效果。其基本的命令是 `\pause`，表示幻灯片在此处会停顿一下，在 `\pause` 后面的所有内容会在 PDF 文件的下一页显示。例如，可以在一帧

的每段话后面使用 `\pause`，让文字一段一段地显示。

可以给目录命令 `\tableofcontents` 加上 `pausesections` 选项，这样目录会在每一项后面暂停，例如：

```
\begin{frame}{目录}
\tableofcontents[pausesections]
\end{frame}
```

更为一般的是 `\onslide`，它可以指定内容在一帧中的第几步显示，使用 `\onslide` 时不显示的内容还占用原来的位置，例如：

```
\begin{frame}
\onslide<1>{只有第 1 步}

\onslide<2->{第 2 步之后}

\onslide<1,3>{第 1, 3 两步}
\end{frame}
```

6-3-1

在 `\onslide` 后面尖括号里面的内容就是覆盖步骤的设置。覆盖语法支持单个的步骤，也支持多个步骤和区间。

`\only` 命令与 `\onslide` 命令类似，不过 `\only` 命令在不显示的步骤没有额外的占位，可以得到内容代替的效果，例如：

```
\begin{frame}
计数：\only<1>{1}\only<2>{2}\only<3>{3}\only<4->{4}

\onslide<5> 数完了。
\end{frame}
```

6-3-2

`\onslide` 和 `\only` 命令还有许多变种，这些命令的效果都大同小异，`\uncover` 和 `\visible` 与 `\onslide` 大体相同，只是用不同的方式隐藏文字；`\invisible` 与 `\visible` 具有相反的效果；`\alt` 可以分别设置在指定步骤和步骤外的内容，而 `\temporal` 则可以分别指定指定步骤与此步骤前后的内容，这里不再详细举例。

这种使用尖括号表示步骤的覆盖语法，实际上在 `beamer` 的很多命令和环境后面都可以使用，例如：

```
\begin{frame}
```

```
\textbf<3>{只在第 3 步加粗}
\end{frame}
```

6-3-3

或者

```
\begin{frame}
\begin{theorem}<2->
  第 2 步以后显示的定理
\end{theorem}
\end{frame}
```

6-3-4

最为常用的则是列表环境，可以给 `\item` 命令加上使用覆盖的步骤号，例如：

```
\begin{frame}
\begin{itemize}
  \item<1-> 开始显示
  \item<3-> 最后显示
  \item<2-> 然后显示
\end{itemize}
\end{frame}
```

6-3-5

在覆盖的语法中，使用加号 + 就类似使用了 `\pause`，这可以避免手工计数。连续使用多个 `\item<+->` 就可以表示 `\item<1->`、`\item<2->`……的效果。可以在整个 `enumerate` 或 `itemize` 环境后面加上 `[<+->]` 的可选项，相当于对每个 `\item` 后面都使用了 `<+->`，非常方便：

```
\begin{frame}
\begin{itemize}[<+->]
  \item 开始显示
  \item 其次显示
  \item 最后显示
\end{itemize}
\end{frame}
```

6-3-6

`\structure` 和 `\alert` 命令则用于在指定的步骤设置高亮，前者使用幻灯片中结构的色彩，后者使用更鲜明的警告色彩（一般是红色）。它们也可以带上覆盖的语法，例如：

```
\alert<2>{在第 2 步强调重要的内容}
```

6-3-7

高亮命令可以作为覆盖语法的一部分，用在 `\item` 等命令后面，这样既可以控制条目何时显示，也可以控制条目高亮。例如，在我们的例子中，就使用了下面的列表：

```
\begin{itemize}
\item<+| alert@+>
    公元前 6 世纪，毕达哥拉斯学派发现一个法则，可以构造直角三角形的边长；
\item<+| alert@+>
    公元前 3 世纪，欧几里德《几何原本》使用面积法证明勾股定理。
\end{itemize}
```

6-3-8

当然，把它改成只在 `itemize` 环境后面使用一次 [`<+| alert@+>`] 也是可以的。



## 练习

6.8 完成勾股定理幻灯片的编写，在里面添加合适的覆盖和高亮命令。

6.9 覆盖对于幻灯片的页码有什么影响？在使用覆盖时，`beamer` 每帧底部的导航按钮有什么好处？

### 6.3.2 活动对象与多媒体



在 `beamer` 中也可以使用一些真正的动态演示功能，包括 PDF 动画、JavaScript 表单、多媒体对象等内容。注意并非所有 PDF 阅读器都支持这些动态功能，使用 Adobe Reader 可以完整地显示包含动态功能的 PDF 幻灯片。

PDF 动画是把动画内容分别画在许多页中，再通过在 PDF 文件中自动快速翻页产生的效果。`beamer` 提供了 `\animate` 和 `\animatevalue` 命令生成动画。`\animate` 定义自动步进的步数，指定的几步会迅速翻页；`\animatevalue` 则设置动画变量，变量是整数或长度寄存器<sup>[69、126]</sup>，可以用于控制画面，其语法格式如下：

```
\animate<{自动步进的步数}>
\animatevalue<{起步},{止步}>{ {寄存器} }{ {起值} }{ {止值} }
```


例如：

<sup>\*</sup>本节内容初次阅读可略过。



```
\newdimen\xoffset
\begin{frame}
% 第一步是静止的，之后自动运动
\animate<2-10>
\animatevalue<1-10>{\xoffset}{0cm}{5cm}
\hspace{\xoffset}从左到右
\end{frame}
```

6-3-9

 beamer 本身的动画功能比较简单，如果要求更高，可以使用 animate 宏包<sup>[91]</sup> 也可以在幻灯片中画出 PDF 动画来。animate 宏包的动画是嵌入在一个 PDF 页面内的，不需要自动翻页，并且通过 JavaScript 代码提供了更丰富的控制功能。

beamer 还支持 PDF 页面的动画切换效果（见表 6.1），这些效果只在 PDF 文件全屏观看时有效。例如，可以设置一帧中的第 2 步，页面从左边飞入：

```
\begin{frame}{动画切换}
\only<1>{旧内容}
\only<2>{新内容}
\transcover<2>
\end{frame}
```

beamer 的一个附属包 multimedia 可以用来在 PDF 幻灯片中嵌入视频、音频等多媒体信息。multimedia 提供的基本命令是 \movie：

**\movie**[(可选项)]{(文字)}{(多媒体文件名)}

例如，下面的代码可以用来播放 4:3 的 AVI 视频 foo.avi：

```
% \usepackage{multimedia}
\begin{frame}{AVI movie}
\movie[width=4cm,height=3cm]{Click to play}{foo.avi}
\end{frame}
```

6-3-10

类似地，\sound 命令可以用来在 PDF 幻灯片中插入音频，例如：

```
% \usepackage{multimedia}
\begin{frame}{Music}
% 自动播放，无显示内容
\sound[autostart]{}{foo.au}
\end{frame}
```

6-3-11

表 6.1 beamer 支持的 PDF 页面切换效果

命令	效果
<code>\transblindshorizontal</code>	水平百叶窗
<code>\transblindvertical</code>	垂直百叶窗
<code>\transboxin</code>	盒状收缩
<code>\transboxout</code>	盒状展开
<code>\transcover</code>	新页面飞入, 覆盖旧页面
<code>\transdissolve</code>	溶解
<code>\transfade</code>	渐显
<code>\transglitter</code>	闪烁 (与溶解类似)
<code>\transpush</code>	新页面推进, 推走旧页面
<code>\transsplitverticalin</code>	垂直收缩
<code>\transsplitverticalout</code>	垂直展开
<code>\transsplithorizontalin</code>	水平收缩
<code>\transsplithorizontalout</code>	水平展开
<code>\transuncover</code>	旧页面飞走, 揭开新页面
<code>\transwipe</code>	沿直线消除旧页面

不过要注意 `multimedia` 的多媒体功能必须使用 `pdfLaTeX` 进行编译<sup>①</sup>, 无法使用 `XLaTeX` 处理中文。

 `media9` 宏包<sup>[89]</sup> 提供了比 `multimedia` 更为强大的多媒体功能, 可以嵌入 Adobe Reader 9 所支持各类媒体和 3D 对象, 并支持各种编译引擎和输出驱动, 使用 `XLaTeX` 可以改用它来代替 `multimedia`。`media9` 的前身是 `movie15` 宏包<sup>[88]</sup>, 它只支持 `pdfLaTeX`。

除了动画和多媒体信息, 使用第三方宏包, 还可以在 `beamer` 中添加更多的动态内容。

`tdclock` 宏包<sup>[221]</sup> 使用 JavaScript 代码和 PDF 表单, 可以在幻灯片中插入日期和时间, 可以用来在演讲中计时, 例如:


```
% \usepackage{tdclock}
\begin{frame}
  当前时间:\tdtime; 已经过去时间:\crono
```

① `\movie` 命令也支持 `Dvips` 方式生成 PDF 文件。

6-3-12

```
\end{frame}
```

不过 `tdclock` 宏包对  $\text{pdf}\text{\LaTeX}$  的支持最好, 如果使用  $\text{Xe}\text{\LaTeX}$  可能需要仔细调整其边框和字号。另外在 Adobe Reader 阅读器中查看表单时钟时, 应该调整阅读器选项, 不突出显示域边框。

 `ocgtools` 宏包<sup>[154]</sup> 提供了所谓可选内容块 (Optional Content Group) 的功能, 可以在幻灯片中添加一个可弹出的可选内容块, 例如:

6-3-13

```
% \usepackage{ocgtools}
\begin{frame}
% 点击公式会显示定理名称
\ocgminitext{ $a^2+b^2=c^2$ }{Pythagoras Theorem}
\end{frame}
```

`ocgtools` 宏包要求使用  $\text{pdf}\text{\LaTeX}$ , 同时要求安装 `acrotex` 包<sup>[243]</sup> 组件<sup>①</sup>。

## 本章注记

除了 `beamer` 官方文档 Tantau et al. [253], `beamer` 的一个优秀的入门介绍是 Kim [122] (`beamer guide`), [122] 有中文黄旭华的中文翻译《`beamer v3.0` 指南》<sup>②</sup>。

除了随 `beamer` 文档类自带的主题, 还有部分第三方的 `beamer` 主题可以使用, 例如 `JLTree` 主题、`nirma` 主题 (这两个主题没有文档, 可直接试用), `beamer2thesis` 包<sup>[77]</sup> 中的 `TorinoTh` 主题等, 都是可以在  $\text{\LaTeX}$  发行版中直接使用的第三方主题。在互联网上还可以找到其他一些用户发布的主题。

除了第三方主题, 在 CTAN 的

`macros/latex/contrib/beamer-contrib/`

目录下, 还可以找到其他一些关于 `beamer` 的有用的宏包。

`beamer` 等幻灯片文档中经常会使用到各种绘图包 (特别是 5.5.2 节的 `PSTricks` 与 `TikZ`) 来完成特殊的效果, 特别是 `beamer` 本身的许多功能也是用 `TikZ` 的底层 `pgf` 完成的, 因此要深入使用  $\text{\LaTeX}$  做幻灯片, 也应该了解相关的绘图工具。不仅是作图, 基于 PDF 超链接、表单功能与 JavaScript 代码的宏包, 可以为幻灯片带来诸如时钟、动画、交互式内容等多方面的功能, 在较高级的幻灯片中也时常用到。除了 6.3.2 节中

① 因授权许可问题,  $\text{\LaTeX}$  Live 不预装 `ocgtools` 与 `acrotex` 包, 因此需要手工安装。

② [http://www.math.ecnu.edu.cn/~Elatex/beamer/beamer\\_guide\\_cn.pdf](http://www.math.ecnu.edu.cn/~Elatex/beamer/beamer_guide_cn.pdf)

介绍的工具体外，还有类似于 ocgtools 的 fancytooltips 宏包<sup>[155]</sup>（也基于 acrotex）、ocgx 宏包<sup>[113]</sup>（不依赖 acrotex 与 JavaScript，但功能较弱）等许多工具，需要读者在实践中积累与选用。

