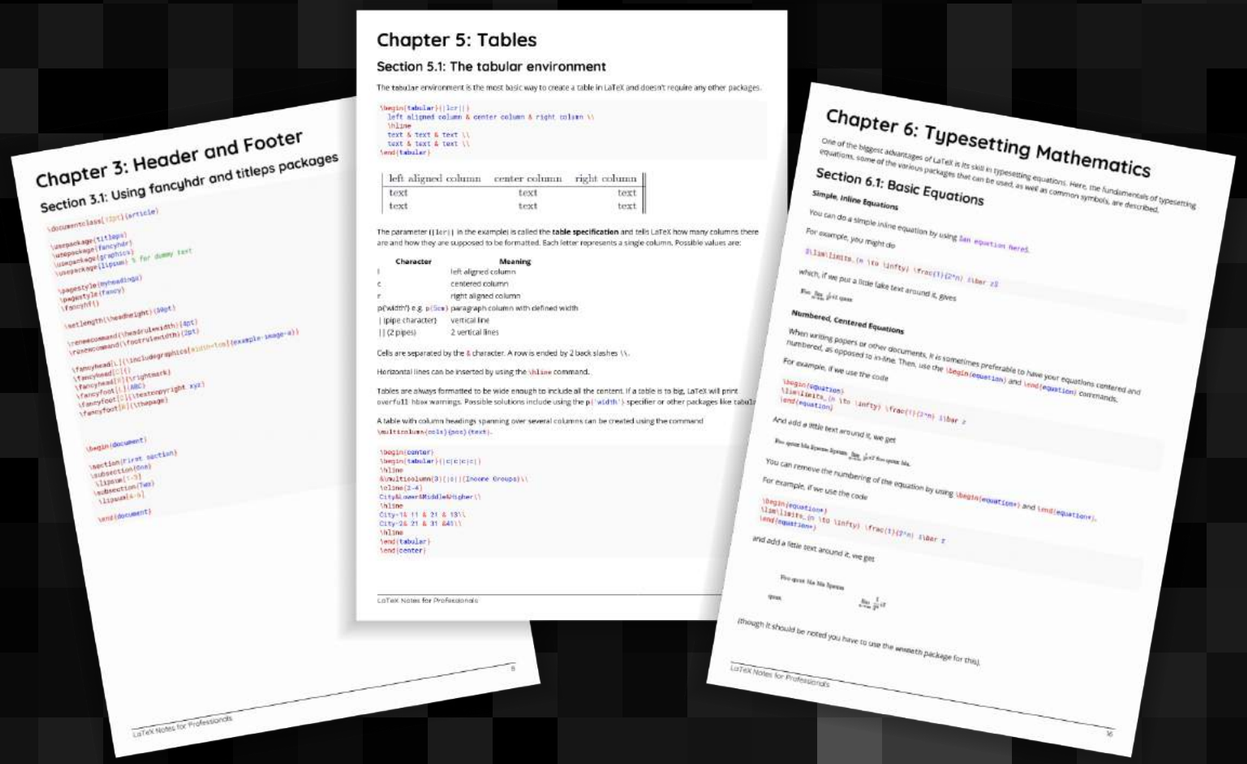
LaTeX
记

专业人士笔记



50+ 页

专业提示和技巧

GoalKicker.com

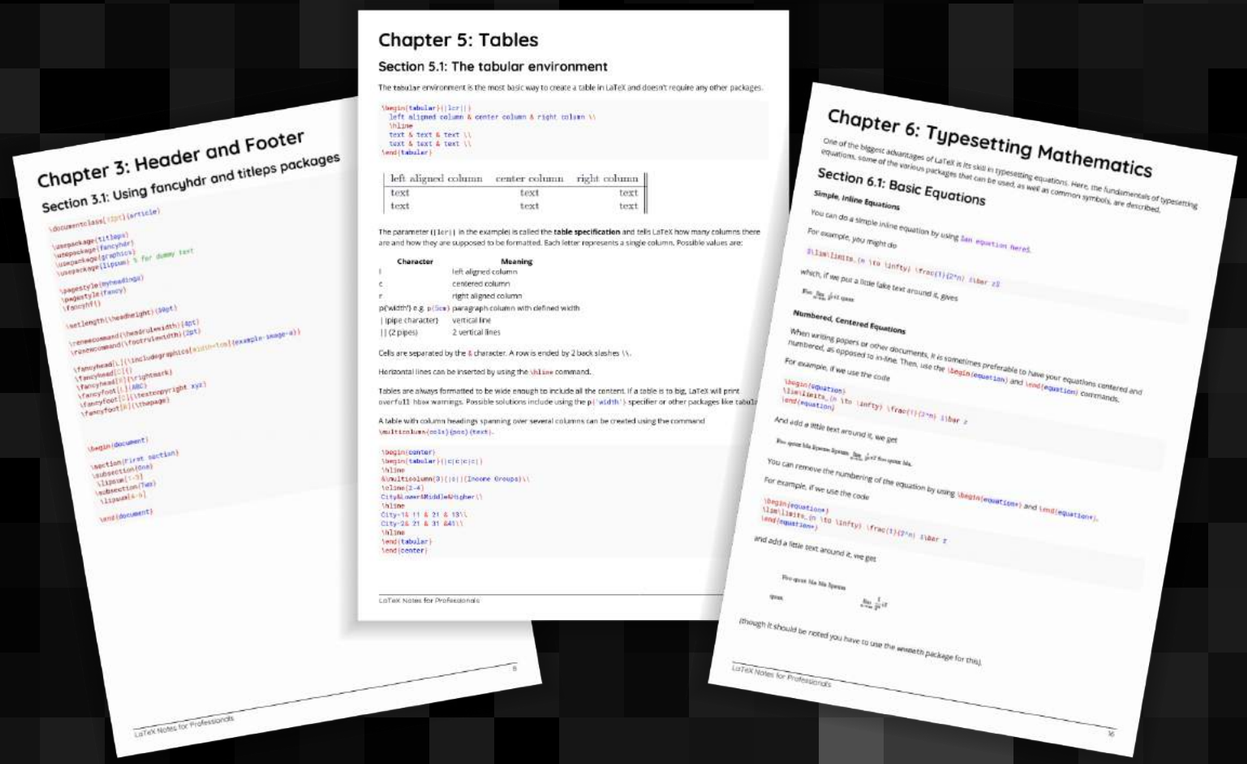
免费编程书籍

免责声明

这是一本非官方的免费书籍，旨在教育用途，
与官方LaTeX组织或公司无关。
所有商标和注册商标均为其各自所有者的财产

LaTeX

Notes for Professionals



50+ pages

of professional hints and tricks

GoalKicker.com

Free Programming Books

Disclaimer

This is an unofficial free book created for educational purposes and is not affiliated with official LaTeX group(s) or company(s). All trademarks and registered trademarks are the property of their respective owners

目录

Contents

关于

请随意免费分享此 PDF，
本书的最新版本可从以下网址下载：
https://goalkicker.com/LaTeXBook

这本 *专业人士的 LaTeX 笔记*是从 Stack Overflow 文档编译而成，内容由 Stack Overflow 的优秀人士撰写。

请将反馈和更正发送至 web@petercv.com

About

Please feel free to share this PDF with anyone for free,
latest version of this book can be downloaded from:
https://goalkicker.com/LaTeXBook

This *LaTeX Notes for Professionals* book is compiled from Stack Overflow Documentation, the content is written by the beautiful people at Stack Overflow.

Please send feedback and corrections to web@petercv.com

第1章：LaTeX入门

版本	发布日期
LaTeX 2.09	1985-09-01
LaTeX 2e	1994-06-01

第1.1节：LaTeX编辑器

虽然您可以使用任何编辑器创建LaTeX文档并通过控制台编译，但存在多个广泛使用编辑器的插件，以简化LaTeX文档的创建，并且还有专门的LaTeX编辑器。
TeX.SE（专注于TeX、LaTeX及相关内容的StackExchange网站）上有一份详尽的LaTeX编辑器列表。

根据该列表，最广泛使用的编辑器有：

- 带有AUCTeX扩展的Emacs编辑器。
- 带有 LaTeX 套件插件的 Vim 编辑器。
- Texmaker – 一个专门的 LaTeX 集成开发环境（IDE）。
- TeXstudio – 另一个 LaTeX 集成开发环境（IDE）。
- TeXworks – 又一个 LaTeX 集成开发环境（IDE）。

虽然有经验的 Emacs 或 Vim 用户可能更愿意坚持使用他们的编辑器（其插件提供了许多其他地方没有的功能），但对于初学者来说，专门的集成开发环境可能更容易安装和使用。列表中的后三个都有预览功能，可以看到文档编译的结果。

此外，还有一些在线 LaTeX 工具，适合初学者或需要协作的人使用，例如
ShareLaTeX 和 Overleaf。

第 1.2 节：安装与设置

你可以选择以下主要的 LaTeX 发行版：

- TeX Live（适用于 Windows、Linux 和 OS X），这是标准的跨平台发行版。
- MacTeX（Mac）是为OS X制作的TeX Live打包版本，包含一些Mac特定的工具
- MiKTeX（Windows）是一个完全独立的发行版

在理想情况下，所有发行版或多或少都是等效的。TeX Live的优势在于它可用于所有平台，因此拥有更好的社区支持。MiKTeX可以利用Windows特定的功能。
出于许可原因，MiKTeX还会分发一些TeX Live不会提供的软件包。

在所有情况下，建议进行完整安装。特别是，使用MiKTeX的按需下载功能会导致许多编辑器挂起或崩溃。

安装
Windows（TeXLive）

1. 从其官网下载安装最新的TeXLive install-tl-windows.exe。
2. 运行install-tl-windows.exe并按照说明操作。

Windows（MiKTeX）

1. 从他们的网站下载最新的MiKTeX安装程序。
2.运行安装程序并按照说明操作。

Mac OS X（TeXLive）

Chapter 1: Getting started with LaTeX

Version	Release Date
LaTeX 2.09	1985-09-01
LaTeX 2e	1994-06-01

Section 1.1: LaTeX Editors

While you can create LaTeX documents using any editor and compiling using the console, there exist several plugins for widely used editors to simplify creating LaTeX documents, and there are specialized LaTeX editors. An exhaustive list of LaTeX editors is available on TeX.SE (the StackExchange site, dedicated to TeX, LaTeX & Friends).

The most widely used editors, according to this list, are:

- The Emacs editor with the AUCTeX extension.
- The Vim editor with the LaTeX-suite plugin.
- Texmaker – a specialized LaTeX IDE.
- TeXstudio – another LaTeX IDE.
- TeXworks – one more LaTeX IDE.

While experienced users of Emacs or Vim may want to stick to their editor (whose plugins provide a host of functionality unavailable elsewhere), a specialized IDE might be easier to install/use for beginners. The last three on the list have a preview function where one can see the results of the compilation of the document.

Additionally, there are online LaTeX tools that can be of use to beginners or people that must collaborate, e.g. ShareLaTeX and Overleaf.

Section 1.2: Installation and Setup

You can choose between major distributions of LaTeX:

- TeX Live (Windows, Linux, and OS X), the standard, cross-platform distribution.
- MacTeX (Mac) A packaged version of TeX Live made for OS X with some Mac-specific tools
- MiKTeX (Windows) A separate distribution entirely that

All distributions are more or less equivalent in an ideal world. TeX Live has the advantage of being available on all platforms and thus has much better community support. MiKTeX can take advantage of Windows-specific features. For licensing reasons, MiKTeX will also distribute a few packages that TeX Live will not.

In all cases, the full install is recommended. Specifically, using MiKTeX's download-on-command feature will hang/crash many editors.

Installation
Windows (TeXLive)

1. Download the most recent TeXLive `install-tl-windows.exe` from their website.
2. Run `install-tl-windows.exe` and follow the instructions.

Windows (MiKTeX)

1. Download the most recent MiKTeX installer from their website.
2. Run the installer and follow the instructions.

Mac OS X (TeXLive)

1. 从其官网下载安装最新的MacTeX。
2. 运行MacTeX.pkg并按照说明操作。

Linux（TeXLive）

Linux用户有两种选择：

1.通过发行版的包管理器安装（通常版本较旧）
2.从上游安装（每年发布，频繁更新）

使用包管理器

- Arch Linux：`pacman -S texlive-most`
- Debian/Ubuntu/Mint：`apt-get install texlive-full`
- Fedora：`yum install texlive`

请注意，使用此方法意味着您将依赖该软件包的维护者来分发更新。这些软件包通常会比最新的发行版本落后几个版本，往往意味着关键更新会缺失。几乎总是最好从上游安装。另外请注意，发行版的软件包管理器可能不会识别直接安装的版本，并且在安装其他相关支持软件包时可能会尝试重新安装它。

从上游安装

1. 从他们的网站下载最新的 TeXLive install-tl-unx.tar.gz。
2. 使用 tar -zxvf install-tl-unx.tar.gz 解压归档文件。
3. 进入下载的文件夹，命令为 cd install-tl-unx。
4. 运行 ./install-tl 并按照指示操作。

   TeXLive 现在应该安装在 /usr/local/texlive/YEAR/ 目录下，其中 YEAR 是四位数的年份（例如2016）。通过这种方式，可以同时拥有多个 TeXLive 版本，并通过更改 PATH 变量在它们之间切换。

   打开此文件夹并检查bin文件夹。它应包含一个子文件夹，具体名称（取决于您的平台）可能是类似于i386-linux或x86_64-linux的名称。

5. 将 TeX Live 的二进制文件夹添加到你的路径中，命令为

```
EXPORT PATH=/usr/local/texlive/YEAR/bin/PLATFORM:$PATH
```

   其中YEAR是四位数的年份（例如2016），PLATFORM是你的平台（例如x86_64-linux）。

测试安装

LaTeX 安装现在已完成。要测试它，请使用你喜欢的文本编辑器创建一个新文件，命名为test.tex，并添加以下内容：

```
\documentclass{article}
\begin{document}
  Hello World!
\end{document}
```


1. Download the most recent MacTeX from their website.
2. Run `MacTeX.pkg` and follow the instructions.

Linux (TeXLive)

Linux users have two options:

1. Install via your distribution's package manager (usually several releases behind)
2. Install from upstream (released yearly, updated often)

Using Package Managers

- Arch Linux: `pacman -S texlive-most`
- Debian/Ubuntu/Mint: `apt-get install texlive-full`
- Fedora: `yum install texlive`

Note that using this method means that you will be dependent on that package's maintainer for the distribution for updates. These packages will often be several releases behind the most recent distribution, often meaning critical updates will be missing. It's almost always best to install from upstream. Also note that the distribution's package manager will probably not recognize the direct installation and could try to install it when one installs other related support packages.

Installing from Upstream

1. Download the most recent TeXLive `install-tl-unx.tar.gz` from their website.
2. Extract the files from the archive with `tar -zxvf install-tl-unx.tar.gz`.
3. Change into the downloaded folder with `cd install-tl-unx`.
4. Run `./install-tl` and follow the instructions.

   TeXLive should now be installed under `/usr/local/texlive/YEAR/`, where `YEAR` is the four digit year (e.g. `2016`). In this way, it is possible to have multiple TeXLive versions alongside each other and switch between them by changing your PATH variable.

   Open this folder and check the `bin` folder. It should contain a subfolder, which (depending on your platform) will be something like `i386-linux` or `x86_64-linux`.

5. Add the TeX Live binary folder to your path with

```
EXPORT PATH=/usr/local/texlive/YEAR/bin/PLATFORM:$PATH
```

   where `YEAR` is the four digit year (e.g. `2016`), and `PLATFORM` is your platform (e.g. `x86_64-linux`).

Test Installation

The LaTeX installation is now complete. To test it, create a new file with your favorite text editor, name it `test.tex` and add the following content:

```
\documentclass{article}
\begin{document}
  Hello World!
\end{document}
```

现在，打开控制台或终端，进入你保存test.tex的文件夹，然后运行

```
pdflatex test
```

（请注意，您的编辑器可能具备为您运行此命令的功能。）

这将创建多个新文件，包括 test.pdf。 这是输出文档，内容如下：

Now, open the console or terminal, navigate to the folder where you saved `test.tex` and run

```
pdflatex test
```

(Note that your editor may have facilities to run this for you.)

This creates several new files, including `test.pdf`. This is the output document, and looks like this:

Hello World!

1

Hello World!

1

恭喜，您已成功安装LaTeX，并创建了您的第一个LaTeX文档！

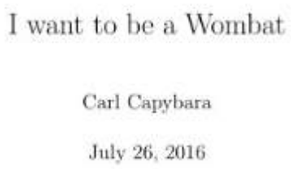
Congratulations, you have successfully installed LaTeX, and created your first LaTeX document!

第2章：标题页

第2.1节：标准报告标题页

```
\documentclass{report}

\begin{document}

\title{我想成为一只袋熊}
\author{卡尔·水豚}
\maketitle

\end{document}
```

这将创建一个仅有标题页且无其他内容的页面：

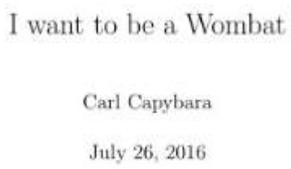


Chapter 2: Title Pages

Section 2.1: Standard report titlepage

```
\documentclass{report}

\begin{document}

\title{I want to be a Wombat}
\author{Carl Capybara}
\maketitle

\end{document}
```

This will create a title page with no other content:



第3章：页眉和页脚

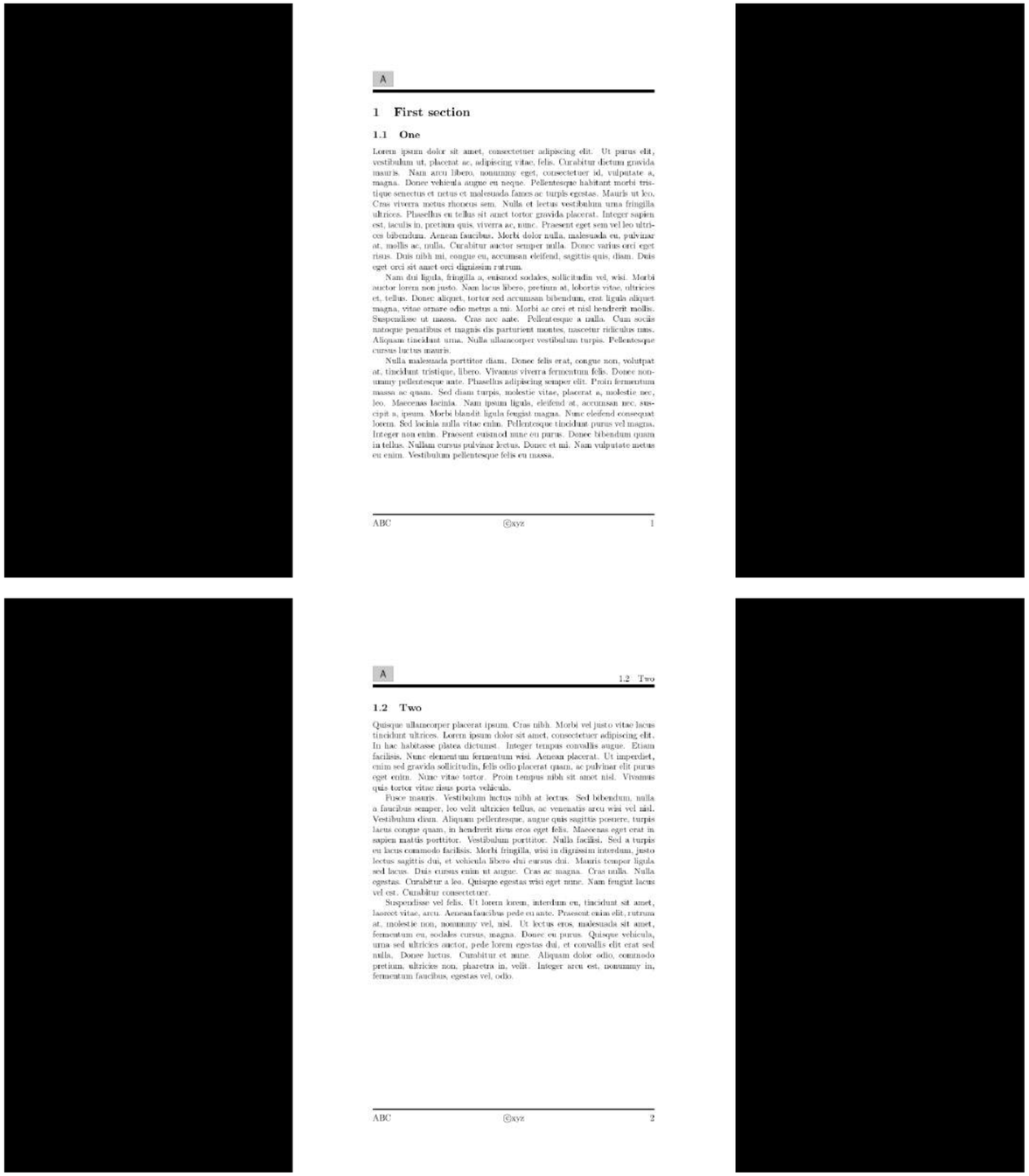
第3.1节：使用fancyhdr和titleps宏包

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

\usepackage{titleps}
\usepackage{fancyhdr}
\usepackage{graphicx}
\usepackage{lipsum} % 用于示例文本

\pagestyle{myheadings}
\pagestyle{fancy}
\fancyhf{}

\setlength{\headheight}{30pt}

\renewcommand{\headrulewidth}{4pt}
\renewcommand{\footrulewidth}{2pt}

\fancyhead[L]{\includegraphics[width=1cm]{example-image-a}}
\fancyhead[C]{}
\fancyhead[R]{\rightmark}
\fancyfoot[L]{ABC}
\fancyfoot[C]{extcopyright xyz}
\fancyfoot[R]{hepage}



\begin{document}

\section{第一节}
\subsection{一}
 \lipsum[1-3]
\subsection{Two}
 \lipsum[4-6]

\end{document}
```


Chapter 3: Header and Footer

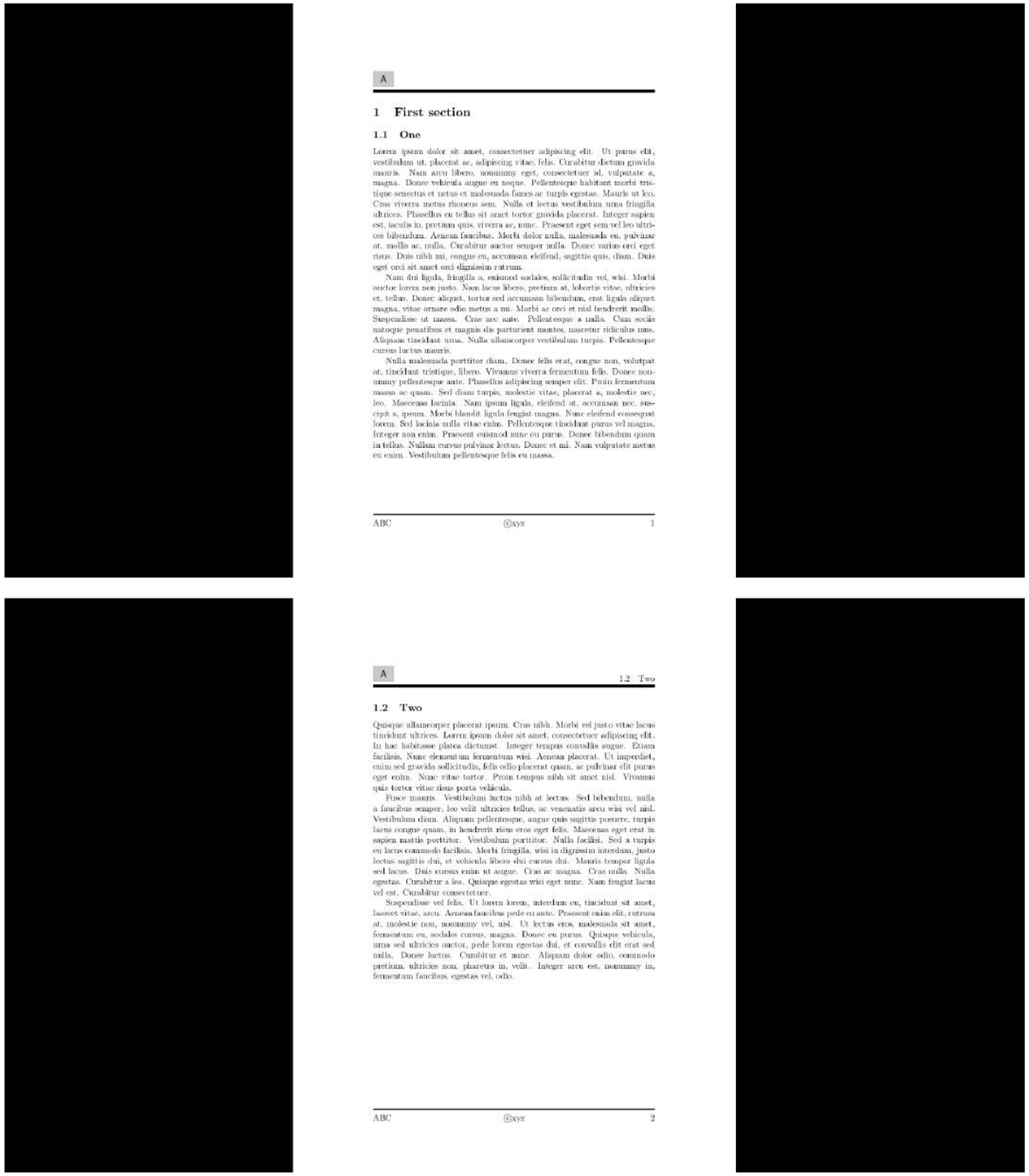
Section 3.1: Using fancyhdr and titleps packages

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

\usepackage{titleps}
\usepackage{fancyhdr}
\usepackage{graphicx}
\usepackage{lipsum} % for dummy text

\pagestyle{myheadings}
\pagestyle{fancy}
\fancyhf{}

\setlength{\headheight}{30pt}

\renewcommand{\headrulewidth}{4pt}
\renewcommand{\footrulewidth}{2pt}

\fancyhead[L]{\includegraphics[width=1cm]{example-image-a}}
\fancyhead[C]{}
\fancyhead[R]{\rightmark}
\fancyfoot[L]{ABC}
\fancyfoot[C]{\textcopyright xyz}
\fancyfoot[R]{\thepage}



\begin{document}

\section{First section}
\subsection{One}
 \lipsum[1-3]
\subsection{Two}
 \lipsum[4-6]

\end{document}
```



第3.2节：页脚中的页码格式为 当前页/总页数

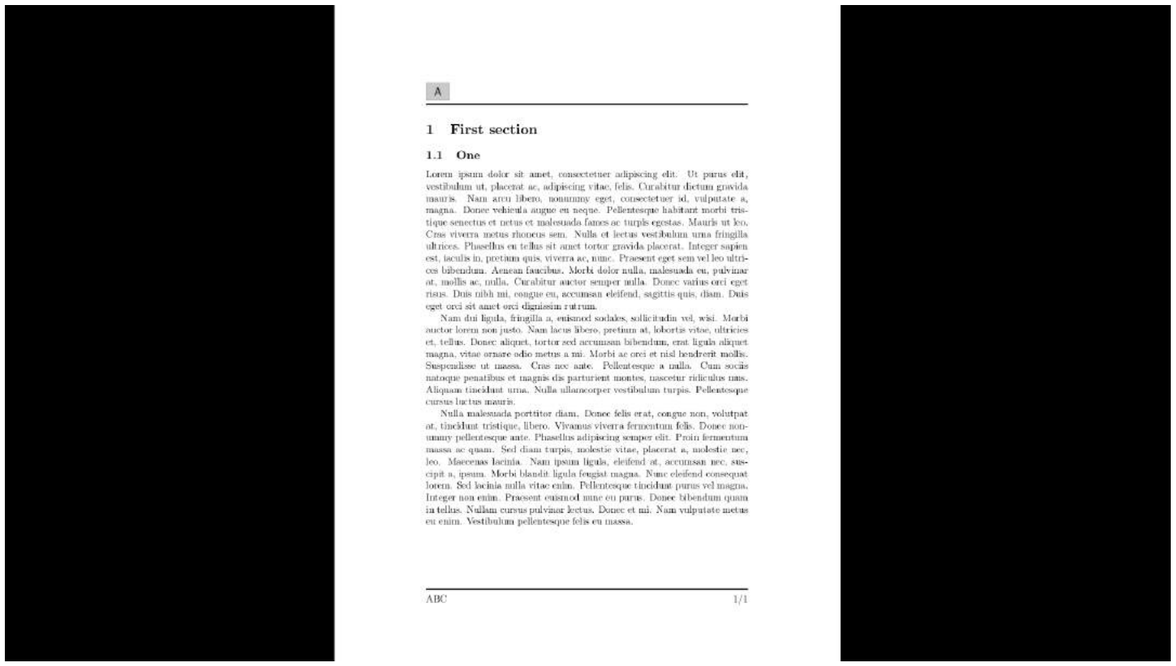
```
\documentclass[12pt]{article}

\usepackage{lastpage}
\usepackage{fancyhdr}
\usepackage{graphicx}
\usepackage{lipsum} % 用于示例文本

\pagestyle{myheadings}
```

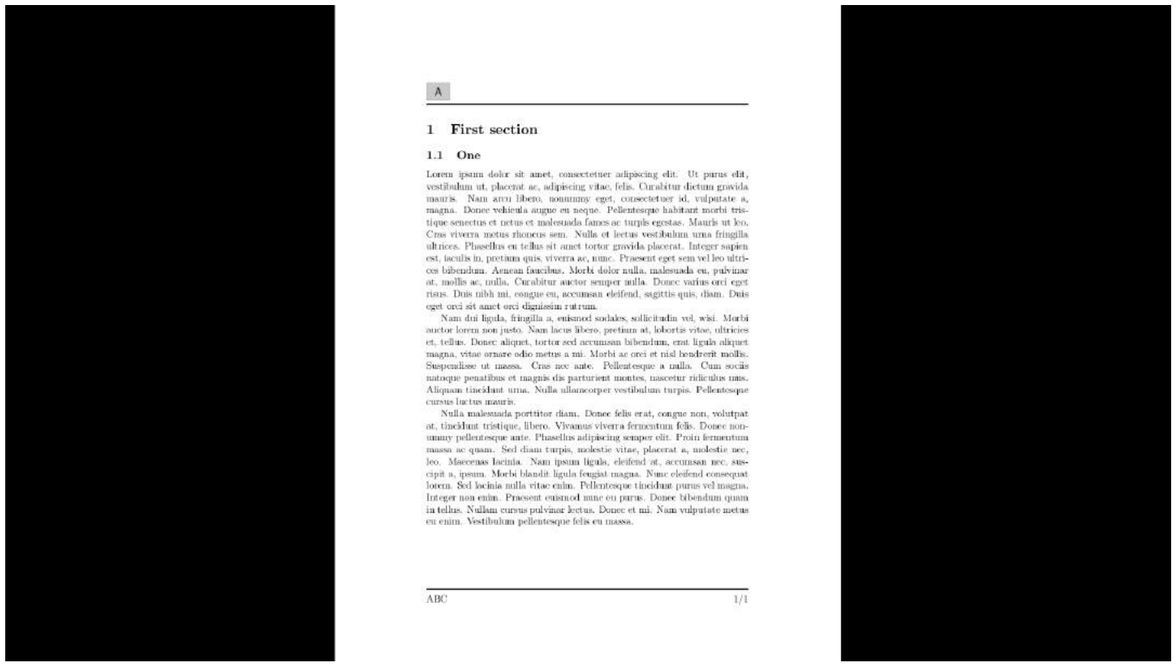
Section 3.2: Page number as CurrPage/TotalPages in footer

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

\usepackage{lastpage}
\usepackage{fancyhdr}
\usepackage{graphicx}
\usepackage{lipsum} % for dummy text

\pagestyle{myheadings}
```

```
\pagestyle{fancy}
\fancyhf{}

\setlength{\headheight}{30pt}

\renewcommand{\headrulewidth}{1pt}
\renewcommand{\footrulewidth}{2pt}

\lhead{\includegraphics[width=1cm]{example-image-a}}
\rhead{}

\lfoot{ABC}
\rfoot{hepage/\pageref{LastPage}}

\begin{document}

\section{第一节}
\subsection{一}
 \lipsum[1-3]

\end{document}
```




```
\pagestyle{fancy}
\fancyhf{}

\setlength{\headheight}{30pt}

\renewcommand{\headrulewidth}{1pt}
\renewcommand{\footrulewidth}{2pt}

\lhead{\includegraphics[width=1cm]{example-image-a}}
\rhead{}

\lfoot{ABC}
\rfoot{\thepage/\pageref{LastPage}}

\begin{document}

\section{First section}
\subsection{One}
 \lipsum[1-3]

\end{document}
```



第4章：文本格式

第4.1节：加粗文本

为了将文本设置为粗体，使用 \textbf：

```
\textbf{这段文本设置为粗体。}
```

第4.2节：强调文本

为了强调文本，可以使用命令 \emph，通常会以斜体字体显示文本：

```
这是一段包含 \emph{强调词语} 的文本。
```

第4.3节：删除线文本

包 ulem 中的命令 \sout 用于给文本添加删除线：

```
\sout{这段文本带有删除线}
```

包 ulem 重新定义了命令 \emph。如果不希望出现这种行为，可以使用带有选项 normalem 的包 ulem：

```
\usepackage[normalem]{ulem}
```


Chapter 4: Text Formatting

Section 4.1: Bold text

In order to typeset text in bold, use `\textbf`:

```
\textbf{This text is typeset in bold.}
```

Section 4.2: Emphazise Text

In order to emphasize text the command `\emph` can be used which usually displays the text in an italics font:

```
This is some text with \emph{emphasized words}.
```

Section 4.3: Strike through text

The command `\sout` of the package `ulem` strikes through a text:

```
\sout{This text is striked through}
```

The package `ulem` redefines the command `\emph`. When you do not want to have this behavior you can use the package `ulem` with the option `normalem`:

```
\usepackage[normalem]{ulem}
```

第5章：表格

第5.1节：表格环境

tabular 环境是LaTeX中创建表格的最基本方式，不需要其他宏包。

```
\begin{tabular}{|lcr||}
左对齐列 & 居中列 & 右对齐列 \\
  \hline
文本 & 文本 & 文本 \\
  文本 & 文本 & 文本 \\
\end{tabular}
```

left aligned column	center column	right column
text	text	text
text	text	text

参数（示例中的|lcr||）称为table specification，告诉LaTeX有多少列以及它们应如何格式化。每个字母代表一列。可能的取值有：

字符	含义
l	左对齐列
c	居中列
r	右对齐列
p{'width'} 例如 p{5cm}	定义宽度的段落列
| （竖线字符）	竖线
|| （2个竖线）	2条竖线

单元格由&字符分隔。行以两个反斜杠\\结束。

可以使用\hline命令插入横线。

表格总是格式化为足够宽以包含所有内容。如果表格太大，LaTeX将打印 overfull hbox 警告。可能的解决方案包括使用 p{'width'} 指定符或其他如 tabularx 的宏包。

可以使用以下命令创建带有跨多列的列标题的表格 \multicolumn{cols}{pos}{text}。

```
\begin{center}
\begin{tabular}{|c|c|c|c|}
\hline
&\multicolumn{3}{|c|}{收入组}\\
\cline{2-4}
城市&低&中&高\\
\hline
城市-1& 11 & 21 & 13\\
城市-2& 21 & 31 &41\\
\hline
\end{tabular}
\end{center}
```


Chapter 5: Tables

Section 5.1: The tabular environment

The `tabular` environment is the most basic way to create a table in LaTeX and doesn't require any other packages.

```
\begin{tabular}{|lcr||}
  left aligned column & center column & right column \\
  \hline
  text & text & text \\
  text & text & text \\
\end{tabular}
```

left aligned column	center column	right column
text	text	text
text	text	text

The parameter (`|lcr||` in the example) is called the **table specification** and tells LaTeX how many columns there are and how they are supposed to be formatted. Each letter represents a single column. Possible values are:

Character	Meaning
l	left aligned column
c	centered column
r	right aligned column
p{'width'} e.g. p{5cm}	paragraph column with defined width
| (pipe character)	vertical line
|| (2 pipes)	2 vertical lines

Cells are separated by the & character. A row is ended by 2 back slashes \\.

Horizontal lines can be inserted by using the \hline command.

Tables are always formatted to be wide enough to include all the content. If a table is to big, LaTeX will print `overfull hbox` warnings. Possible solutions include using the p{'width'} specifier or other packages like `tabularx`.

A table with column headings spanning over several columns can be created using the command \multicolumn{cols}{pos}{text}.

```
\begin{center}
\begin{tabular}{|c|c|c|c|}
\hline
&\multicolumn{3}{|c|}{Income Groups}\\
\cline{2-4}
City&Lower&Middle&Higher\\
\hline
City-1& 11 & 21 & 13\\
City-2& 21 & 31 &41\\
\hline
\end{tabular}
\end{center}
```

City	Income Groups		
	Lower	Middle	Higher
City-1	11	21	13
City-2	21	31	41

注意命令 \multicolumn 有三个必需参数：第一个参数指定标题跨越的列数；第二个参数指定标题的位置(l,c,r)；第三个参数是标题文本。命令 \cline{2-4} 指定了要绘制线条的起始列（这里是2）和结束列（这里是4）。

第5.2节：表格着色

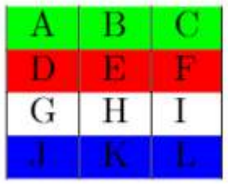
为了使表格更易读，以下是为其着色的方法：

1.行
2.列
3.线
4.单元格

为行着色

使用\rowcolor（由colortbl提供；在[table]包选项下由xcolor也会加载）。示例：

```
\documentclass{article}
\usepackage[table]{xcolor}

\begin{document}

\begin{tabular}{ | l | l | l | }
  \rowcolor{green}
A & B & C \\
  \rowcolor{red}
  D & E & F \\
  G & H & I \\
  \rowcolor{blue}
  J & K & L
\end{tabular}

\end{document}
```



为列着色

列可以通过以下方式着色：

City	Income Groups		
	Lower	Middle	Higher
City-1	11	21	13
City-2	21	31	41

Note that the command `\multicolumn` has three mandatory arguments: the first argument specifies the number of columns over which the heading spans; the second argument specifies the position of the heading`(1,c,r)`; and the third argument is the text for heading. The command `\cline{2-4}` specifies the the starting column(here, 2) and ending column(here, 4) over which a line is to be drawn.

Section 5.2: Coloring Table

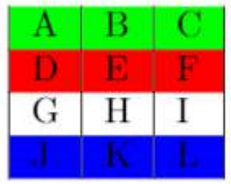
To make the table more readable, following are the ways to color it:

1. Rows
2. Columns
3. Lines
4. Cells

Coloring Rows

Use `\rowcolor` (provided by colortbl; also loaded by xcolor under the `[table]` package option). Example:

```
\documentclass{article}
\usepackage[table]{xcolor}

\begin{document}

\begin{tabular}{ | l | l | l | }
  \rowcolor{green}
  A & B & C \\
  \rowcolor{red}
  D & E & F \\
  G & H & I \\
  \rowcolor{blue}
  J & K & L
\end{tabular}

\end{document}
```



Coloring Columns

Columns can be colored using following ways:

- 在表格标签外定义列颜色属性，使用 ewcolumntype：

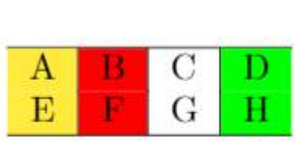
```
ewcolumntype{a}{ >{\columncolor{yellow}} c }
```

- 在表格参数内定义列颜色属性

```
\begin{tabular}{ | >{\columncolor{red}} c | l | l }
```

示例：

```
\documentclass{article}
\usepackage[table]{xcolor}

ewcolumntype{a}{>{\columncolor{yellow}}c}
ewcolumntype{b}{>{\columncolor{green}}c}

\begin{document}

\begin{tabular}{ a | >{\columncolor{red}}c | l | b }
  \hline
A & B & C & D \\
  E & F & G & H \\
  \hline
\end{tabular}

\end{document}
```



为行着色

使用 \arrayrulecolor。示例：

```
\documentclass{article}
\usepackage[table]{xcolor}

\arrayrulecolor{blue}

\begin{document}

\begin{tabular}{ | l | l | l | }
  \hline
A & B & C \\
  \hline
D & E & F\\
  \hline
G & H & I \\
  \hline
\end{tabular}

\end{document}
```


- Defining column color property outside the table tag using \newcolumntype:

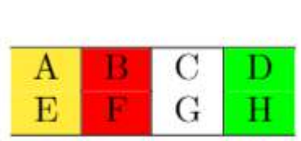
```
\newcolumntype{a}{ >{\columncolor{yellow}} c }
```

- Defining column color property inside the table parameters

```
\begin{tabular}{ | >{\columncolor{red}} c | l | l }
```

Example:

```
\documentclass{article}
\usepackage[table]{xcolor}

\newcolumntype{a}{>{\columncolor{yellow}}c}
\newcolumntype{b}{>{\columncolor{green}}c}

\begin{document}

\begin{tabular}{ a | >{\columncolor{red}}c | l | b }
  \hline
  A & B & C & D \\
  E & F & G & H \\
  \hline
\end{tabular}

\end{document}
```



Coloring Lines

Use \arrayrulecolor. Example:

```
\documentclass{article}
\usepackage[table]{xcolor}

\arrayrulecolor{blue}

\begin{document}

\begin{tabular}{ | l | l | l | }
  \hline
  A & B & C \\
  \hline
  D & E & F\\
  \hline
  G & H & I \\
  \hline
\end{tabular}

\end{document}
```

A	B	C
D	E	F
G	H	I

单元格着色

使用 \cellcolor。例如：

```
\documentclass{article}
\usepackage[table]{xcolor}

\begin{document}

\begin{tabular}{ | l | l | l | }
  \hline
A & B & C \\
  \hline
D & E & \cellcolor{green}F \\
  \hline
G & H & I \\
  \hline
\end{tabular}

\end{document}
```

A	B	C
D	E	F
G	H	I

我们也可以使用包 colortbl 自定义颜色。以下是标签示例：

```
\definecolor{Gray}{gray}{0.85}
\columncolor[RGB]{230, 242, 255}}
\columncolor[HTML]{AAACED}
```


A	B	C
D	E	F
G	H	I

Coloring Cells

Use \cellcolor. Example:

```
\documentclass{article}
\usepackage[table]{xcolor}

\begin{document}

\begin{tabular}{ | l | l | l | }
  \hline
  A & B & C \\
  \hline
  D & E & \cellcolor{green}F \\
  \hline
  G & H & I \\
  \hline
\end{tabular}

\end{document}
```

A	B	C
D	E	F
G	H	I

We can define our own colors too using package colortbl. Following are the tags examples:

```
\definecolor{Gray}{gray}{0.85}
\columncolor[RGB]{230, 242, 255}}
\columncolor[HTML]{AAACED}
```

第6章：数学排版

LaTeX最大的优势之一是其排版方程式的能力。这里介绍了排版方程式的基础知识、可用的一些不同宏包以及常用符号。

第6.1节：基本方程式

简单的行内方程式

你可以使用$在这里写方程式$来做一个简单的行内方程式。

例如，你可以写

```
$\lim\limits_{n o \infty} \frac{1}{2^n} i\bar z$
```

如果我们在它周围加上一些假文本，会得到

Foo $\lim\limits_{n \to \infty} \frac{1}{2^n} i\bar z$ quux

带编号且居中的方程式

在撰写论文或其他文档时，有时更希望方程式是居中且带编号的，而不是行内的。此时，可以使用\begin{equation}和\end{equation}命令。

例如，如果我们使用代码

```
\begin{equation}
\lim\limits_{n o \infty} frac{1}{2^n} i\bar z
\end{equation}
```

并在其周围添加一些文字，我们得到

Foo quux bla lipsum lipsum $\lim\limits_{n \to \infty} \frac{1}{2^n} i\bar z$ foo quux bla.

您可以使用 \begin{equation*} 和 \end{equation*} 来去除公式编号。

例如，如果我们使用代码

```
\begin{equation*}
\lim\limits_{n o \infty} frac{1}{2^n} i\bar z
\end{equation*}
```

并在其周围添加一些文字，我们得到

Foo quux bla bla lipsum

$$\lim_{n \to \infty} \frac{1}{2^n} i\bar z$$

quux.

（不过需要注意的是，您必须使用 amsmath 宏包才能实现此功能）。

Chapter 6: Typesetting Mathematics

One of the biggest advantages of LaTeX is its skill in typesetting equations. Here, the fundamentals of typesetting equations, some of the various packages that can be used, as well as common symbols, are described.

Section 6.1: Basic Equations

Simple, Inline Equations

You can do a simple inline equation by using `$an equation here$`.

For example, you might do

```
$\lim\limits_{n \to \infty} \frac{1}{2^n} i\bar z$
```

which, if we put a little fake text around it, gives

Foo $\lim\limits_{n \to \infty} \frac{1}{2^n} i\bar z$ quux

Numbered, Centered Equations

When writing papers or other documents, it is sometimes preferable to have your equations centered and numbered, as opposed to in-line. Then, use the `\begin{equation}` and `\end{equation}` commands.

For example, if we use the code

```
\begin{equation}
\lim\limits_{n \to \infty} \frac{1}{2^n} i\bar z
\end{equation}
```

And add a little text around it, we get

Foo quux bla lipsum lipsum $\lim\limits_{n \to \infty} \frac{1}{2^n} i\bar z$ foo quux bla.

You can remove the numbering of the equation by using `\begin{equation*}` and `\end{equation*}`.

For example, if we use the code

```
\begin{equation*}
\lim\limits_{n \to \infty} \frac{1}{2^n} i\bar z
\end{equation*}
```

and add a little text around it, we get

Foo quux bla bla lipsum

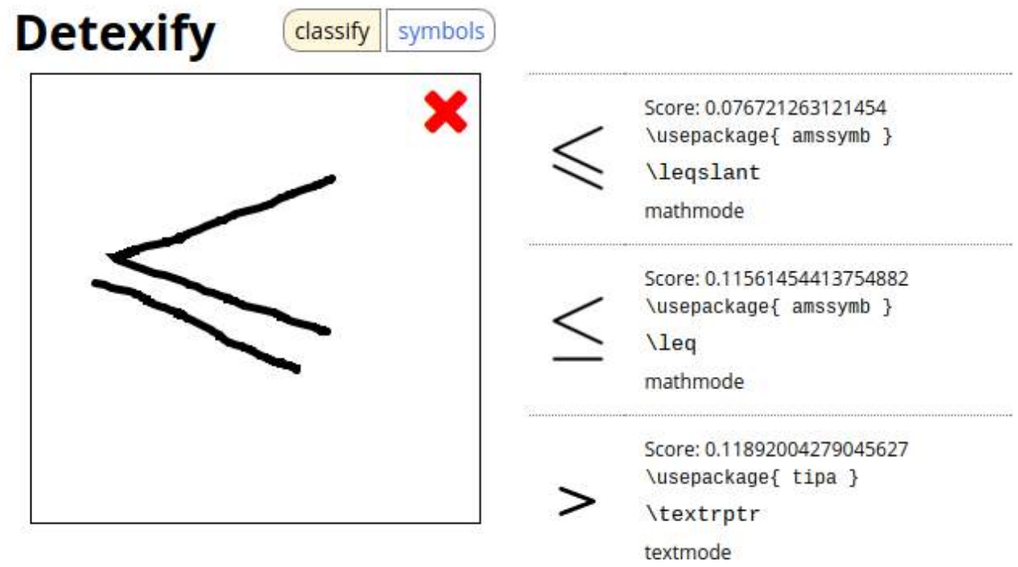
$$\lim_{n \to \infty} \frac{1}{2^n} i\bar z$$

quux.

(though it should be noted you have to use the `amsmath` package for this).

第6.2节：查找符号

有时，找到所需的数学符号可能比较困难。这里有几种选择。第一种（也是最快的）是使用 Detexify，您可以画出想要的符号，它会尝试帮您找到对应的符号，如下所示：



另一种选择是使用综合的 LaTeX 符号列表，可以在这里找到。如果您正在使用unicode-math包，这个所有支持符号的列表会很有帮助。另一种选择是这个网站，里面有常用的数学符号。

第6.3节：可用的包

虽然大多数简单的数学公式和方程只需要标准的 LaTeX，但有时需要更多的符号和工具。有多个包可以增强您的方程并提供更多功能。下面介绍三个主要的包。记住，要加载包，请在文档导言区输入

\usepackage{package}。

amsmath

amsmath包是一个非常有用的包。它用于使您的方程居中但不编号，如\begin{equation*}中所示，它用于创建矩阵（如下所述），并引入了许多其他有用的命令，如\overset和\underset，详见下文。amsmath包的文档可以在这里找到。

mathtools

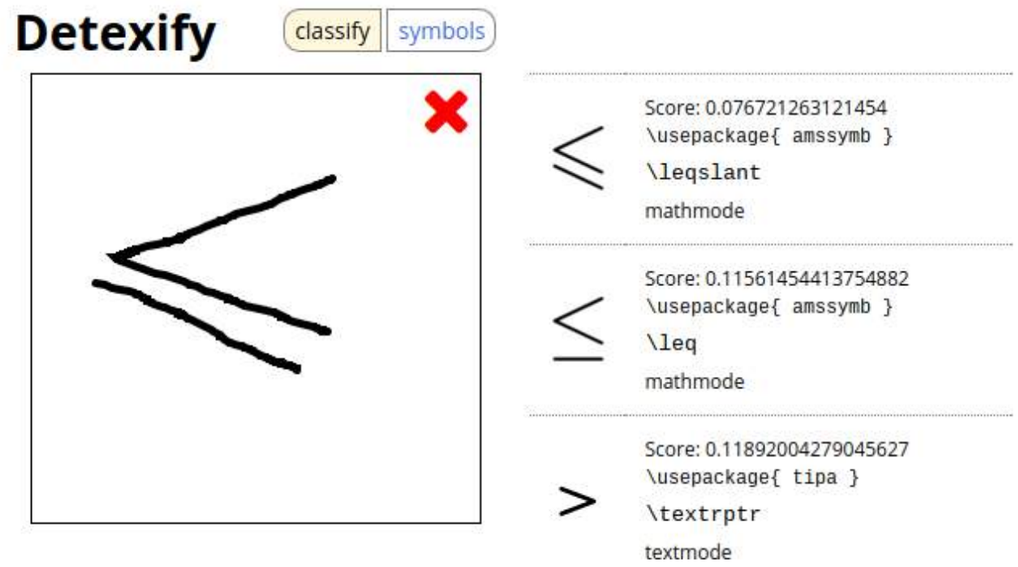
mathtools包是在amsmath包基础上扩展的，增加了更多有用的符号和工具。它会自动加载amsmath包，因此您不需要在文档导言区同时加载这两个包。mathtools文档可以在这里找到。

amssymb

amssymb包提供了许多额外的符号，对于更复杂的方程非常有用。amssymb文档可以在这里找到。

Section 6.2: Finding Symbols

Sometimes, it can be difficult to find the mathematical symbol you need. There are several options here. The first (and quickest) is to use Detexify, where you draw the symbol you'd like, and it tries to find what you want, like as shown below:



Another option is to use the comprehensive LaTeX symbols list, which can be found here. If you are using the package `unicode-math` this list of all supported symbols can be helpful. Another option is this website, which has common math symbols.

Section 6.3: Packages available for use

While standard LaTeX is all that is needed for most simple mathematical formulae and equations, sometimes more symbols and tools are needed. There are multiple packages available that will enhance your equations and provide you with more to work with. Three of the main packages are described below. Remember, to load a package, type `\usepackage{package}` in your document preamble.

amsmath

The `amsmath` package is an incredibly useful package. It is used to allow your equations to be centered but not numbered, as in `\begin{equation*}`, it is used to create matrices (as described below) and it introduces many other useful commands, such as `\overset` and `\underset`, described below. The `amsmath` package documentation can be found here.

mathtools

The `mathtools` package builds off of the `amsmath` package, adding further useful symbols and tools. It automatically loads the `amsmath` package, so you do not need to load both in your document preamble. The `mathtools` documentation can be found here.

amssymb

The `amssymb` package provides many extra symbols that can be very handy for more complex equations. The `amssymb` documentation can be found here.

字体包

你还可以使用各种字体来排版你的公式，具体请参见TeX Stack Exchange上关于TeX、LaTeX及其相关内容的这个问题。

本文简明扼要地解释了一些宏包以及标准LaTeX所提供的不同功能；非常有帮助。

第6.4节：值得了解的好命令

一些最常用的命令包括：

- **分数和平方根：**分数使用`\frac{分子}{分母}`。平方根使用`\sqrt[根指数]{数字}`。
- 希腊字母：使用下表中给出的命令：

αA	`\alpha A`	νN	`\nu N`
βB	`\beta B`	$\xi \Xi$	`\xi\Xi`
$\gamma \Gamma$	`\gamma \Gamma`	oO	`o O`
$\delta \Delta$	`\delta \Delta`	$\pi \Pi$	`\pi \Pi`
$\epsilon \varepsilon E$	`\epsilon \varepsilon E`	$\rho \varrho P$	`\rho\varrho P`
ζZ	`\zeta Z`	$\sigma \Sigma$	`\sigma \Sigma`
ηH	`\eta H`	τT	`\tau T`
$\theta \vartheta \Theta$	`\theta \vartheta \Theta`	$\upsilon \Upsilon$	`\upsilon \Upsilon`
ιI	`\iota I`	$\phi \varphi \Phi$	`\phi \varphi \Phi`
κK	`\kappa K`	χX	`\chi X`
$\lambda \Lambda$	`\lambda \Lambda`	$\psi \Psi$	`\psi \Psi`
μM	`\mu M`	$\omega \Omega$	`\omega \Omega`

- 运算符：\leq 表示小于等于符号，\geq 表示大于等于符号，eq 表示不等于符号，\sum 表示求和符号，\partial 表示偏导符号，abla 表示拉普拉斯算子，imes 表示叉乘或乘法符号，\cdot 表示点乘或乘法符号，\int 表示积分符号。

- 箭头：\rightarrow 和 \leftarrow 分别表示右箭头和左箭头。
- 百分号：在LaTeX中输入%时，必须加反斜杠，即\%，因为百分号通常用于注释。

- 上标和下标：上标可以输入 x^2，或者对于较长的上标，输入 x^{2x}。下标可以输入 x_a，或者对于较长的下标，输入 x_{ab}。
- 加粗： 使用 \boldmath{...} 来使你的数学符号加粗。其他选项请参见该 TeX.SX 问题。

Font packages

There are also various fonts you can use for your equations, as described on this question on the TeX stack exchange, for TeX, LaTeX, and friends.

This paper is a concise explanation of the different features provided by some packages as well as standard LaTeX; it is very helpful.

Section 6.4: Good Commands to Know

Some of the most common commands include:

- **Fractions and Square Roots:** For fractions, use `\frac {numerator}{denominator}`. For square roots, use `\sqrt[root]{number}`.
- **Greek letters:** use the commands given in the table below:

αA	`\alpha A`	νN	`\nu N`
βB	`\beta B`	$\xi \Xi$	`\xi\Xi`
$\gamma \Gamma$	`\gamma \Gamma`	oO	`o O`
$\delta \Delta$	`\delta \Delta`	$\pi \Pi$	`\pi \Pi`
$\epsilon \varepsilon E$	`\epsilon \varepsilon E`	$\rho \varrho P$	`\rho\varrho P`
ζZ	`\zeta Z`	$\sigma \Sigma$	`\sigma \Sigma`
ηH	`\eta H`	τT	`\tau T`
$\theta \vartheta \Theta$	`\theta \vartheta \Theta`	$\upsilon \Upsilon$	`\upsilon \Upsilon`
ιI	`\iota I`	$\phi \varphi \Phi$	`\phi \varphi \Phi`
κK	`\kappa K`	χX	`\chi X`
$\lambda \Lambda$	`\lambda \Lambda`	$\psi \Psi$	`\psi \Psi`
μM	`\mu M`	$\omega \Omega$	`\omega \Omega`

- **Operators:** `\leq` gives the less than or equal to symbol, `\geq` gives the greater than or equal to symbol, `\neq` gives the not equal symbol, `\sum` gives the summation symbol, `\partial` gives the partial derivative symbol, `\nabla` gives the Laplacian operator, `\times` gives the cross product or multiplication symbol, `\cdot` gives the dot product or multiplication symbol, and `\int` gives the integral symbol.
- **Arrows:** `\rightarrow` and `\leftarrow` give right and left arrows, respectively.
- **Percents:** If typing % in LaTeX, it is important to include a backslash, `\%` as the percent symbol is normally used for comments.
- **Superscripts and Subscripts:** To do a superscript, you can type `x^2`, or, for longer superscripts, `x^{2x}`. To do a subscript, you can type `x_a`, or, for longer subscripts, `x_{ab}`.
- **Bold:** Use `\boldmath{...}` to make your math symbols bold. Other options are given at this TeX.SX question.

数学符号默认自动斜体；如果你不希望这样，请按照下面描述的方法设置你的公式文本。

- 无穷大： 写无穷大时，使用命令 \infty。
- 在另一个符号上方或下方放置项目： 首先，仅针对数学运算符，有一种替代方法。你可以输入数学运算符，比如 \int，然后使用 \limits 命令。示例为 \int\limits_{\infty}或 \int\limits^{\infty}。然后，对于一般情况，你可以使用 \overset{top}{normal} 或 \underset{bottom}{normal}。这对于表示向量非常有用。例如，你可以写 \vec{x}。使用 amsmath 宏包可以支持 overset 和 underset。
- 花括号： 由于花括号在命令中有特殊用途，必须输入 \{ 或 \} 来表示花括号。

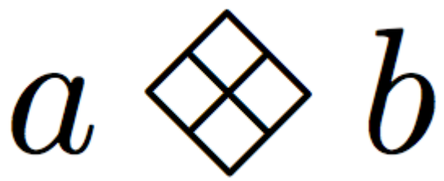
- 文本： 若要在公式中包含文本，请在导言区输入 \usepackage{amsmath}，然后使用 ext{...}。
- 空格： 若要在公式中添加空格，请在两个项目之间输入 \quad（例如，你可以写 $2x \quad cos）。

第6.5节：创建新符号

假设你找不到所需的符号。你可以创建自定义符号。例如，代码

```
\documentclass{article}
usepackage{graphicx,amsmath,amssymb}
\DeclareRobustCommand{\diamondtimes}{%
  \mathbin{ext{\rotatebox[origin=c]{45}{$\boxplus$}}}%
}

\begin{document}
$a\diamondtimes b$
\end{document}
```

创建并调用一个符号，表示



这是一个更简单的例子；它仅仅需要旋转一个已存在的符号。然而，你可以创建更复杂的符号。

本节内容正在扩展中。

第6.6节：矩阵

矩阵

如果你要使用以下命令，必须始终使用amsmath宏包。主要有四种矩阵类型，如下面的代码所示：

```
\begin{matrix}
    a & b \\
    c & d
\end{matrix}
\quad
```


Math symbols are automatically italicized; if you don't want this to be true, make your equation text as described below.

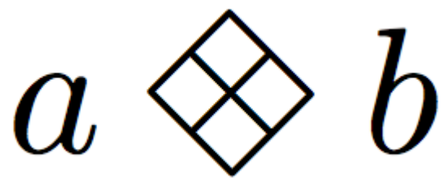
- **Infinity:** To write infinity, use the command \infty.
- **Moving items over or under another:** First, for math operators only, there is an alternate method. You can type the math operator, say \int, and then use the \limits command. An example is \int\limits_{\infty} or \int\limits^{\infty}. Then, for normal cases, you can do \overset{top}{normal} or \underset{bottom}{normal}. This can be very useful for doing vectors. For example, you might do \vec{x} The amsmath package is need for overset and underset.
- **Curly Braces:** Because curly braces are used in commands, it is necessary to type \{ or \} to get curly braces.
- **Text:** To include text in equations, type \usepackage{amsmath} in the preamble, and then type \text{...}.
- **Space:** To add space in your equations, type \quad between the two items you want to separate (for example, you might have $2x \quad cos).

Section 6.5: Creating New Symbols

Let's say you cannot find the symbol you need anywhere. You can create a custom symbol. For example, the code

```
\documentclass{article}
\usepackage{graphicx,amsmath,amssymb}
\DeclareRobustCommand{\diamondtimes}{%
  \mathbin{\text{\rotatebox[origin=c]{45}{$\boxplus$}}}%
}

\begin{document}
$a\diamondtimes b$
\end{document}
```

creates and calls a symbol, giving



This is a simpler example; it merely has to rotate an already existent symbol. However, you can create more complex symbols.

This section is in the process of being expanded.

Section 6.6: Matrices

Matrices

You must always use the amsmath package if you are going to use the following commands. There are four main types of matrix, as shown in the code below:

```
\begin{matrix}
    a & b \\
    c & d
\end{matrix}
\quad
```

```
\begin{pmatrix}
   a & b \\
   c & d
\end{pmatrix}
\quad
\begin{bmatrix}
    a & b \\
    c & d
\end{bmatrix}
\quad
\begin{vmatrix}
    a & b \\
    c & d
\end{vmatrix}
\quad
\begin{Vmatrix}
    a & b \\
    c & d
\end{Vmatrix}
```

这段代码生成

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad \begin{vmatrix} a & b \\ c & d \end{vmatrix} \quad \begin{Vmatrix} a & b \\ c & d \end{Vmatrix}$$

关于这一点，有几个重要事项需要注意：

1. 重要的是你要将矩阵放在 equation 、 equation* 或 $...$ 环境中—— bmatrix command 本身不是数学环境。
2. 矩阵的构造实际上相当简单。对于每一行，你创建每个元素（例如 x_{11}），然后放一个&，然后写下一个元素。对于多行，在每行末尾放\\（最后一行不必这样做）。这与表格非常相似。


```
\begin{pmatrix}
   a & b \\
   c & d
\end{pmatrix}
\quad
\begin{bmatrix}
    a & b \\
    c & d
\end{bmatrix}
\quad
\begin{vmatrix}
    a & b \\
    c & d
\end{vmatrix}
\quad
\begin{Vmatrix}
    a & b \\
    c & d
\end{Vmatrix}
```

This code produces

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad \begin{vmatrix} a & b \\ c & d \end{vmatrix} \quad \begin{Vmatrix} a & b \\ c & d \end{Vmatrix}$$

There are a couple important things to note about this:

1. It is important you put your matrix within the `equation`, `equation*`, or `$...$` environment - the `bmatrix` command is not a math environment on its own.
2. The construction of the matrix is actually fairly simple. For each row, you create each element (say `x_{11}`), then put a `&`, and then write the next element. For multiple rows, at the end of each row put `\\` (you do not have to do this for the last row). It is fairly similar to a table in this.

第七章：创建参考书目

参数	详细信息
参考文献环境	该环境设置实际参考文献的范围。它定义了一个类似列表的环境，在其中可以使用`\bibitem`来设置参考文献条目。
{x}	thebibliography环境接受一个参数，该参数表示在\bibitem枚举中预期的最宽元素。对于少于10个条目，使用单个字符/数字；对于少于100个条目，使用两个字符/数字，依此类推。
\bibitem{<a>} <b>	设置参考文献条目<b>，并使其可通过文档中的\cite使用标签<a>。

第7.1节：使用biber的基本参考文献

要开始编写参考文献，您需要定义您的来源。创建一个数据库文件（如`sources.bib`）并包含一些内容：

```
@book{Doe1993,
     Author = {John Doe},
Publisher = {Earth University},
    Title = {Creating a bibliography with biber},
     Year = {1993}}
```

然后你可以在主文档中包含你的数据库文件，并引用新的来源（Doe1993）。

```
\documentclass{article}

% 引入 biblatex 宏包并指定使用 biber 作为后端。
% 如果不指定后端，默认使用 biber。
\usepackage[backend=biber]{biblatex}

% 定义 biber 查找你的资源的位置
\addbibresource{sources.bib}

\begin{document}
“Biber 并不难。” \cite{Doe1993}
% 使用 |cite{source-ID} 生成引用

% 打印参考文献
\printbibliography

\end{document}
```

要编译文档，您需要依次运行3个命令：

1. pdflatex 用于创建一个辅助文件，告诉biber需要哪些文献来源
2. biber 用于创建一个包含所有文献来源的辅助文件，供pdflatex使用
3. pdflatex 用于包含辅助文件并生成PDF

Chapter 7: Creating a Bibliography

Parameter	Detail
thebibliography	This environment sets the scope for the actual bibliography. It defines a list-like environment within which you can use `\bibitem` to set a bibliography item.
{x}	The `thebibliography` environment takes a single argument that represents the widest element to be expected in the enumeration of the `\bibitem`s. For less than 10 entries, use a single character/digit; for less than 100 entries, use two characters/digits, ...
`\bibitem{<a>}` <b>	Set the bibliography item **<b>** and make it available to `\cite` within the document using the label **<a>**.

Section 7.1: Basic bibliography with biber

To start a bibliography you need to define your sources. Create a database file (like `sources.bib`) and include some content:

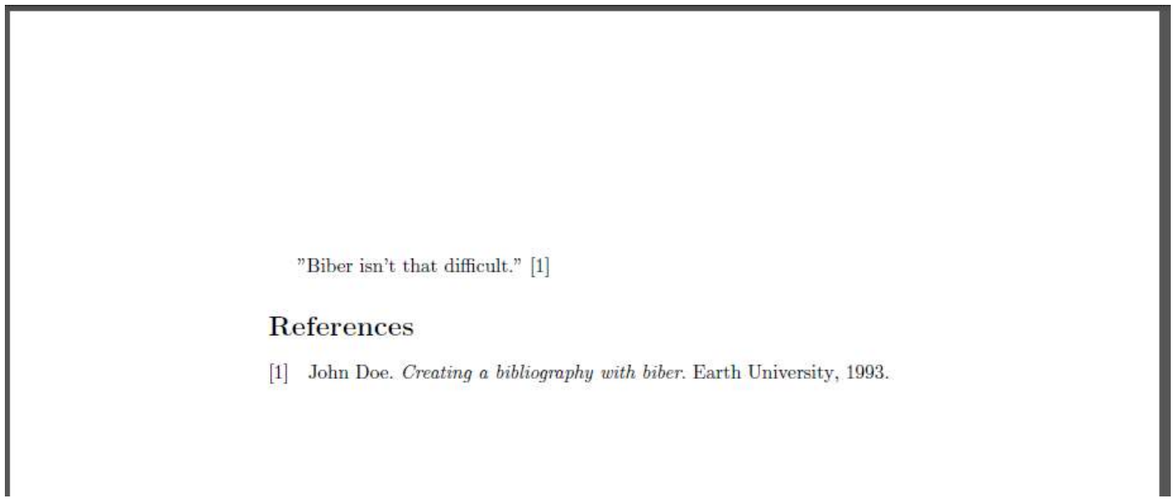
```
@book{Doe1993,
    Author = {John Doe},
    Publisher = {Earth University},
    Title = {Creating a bibliography with biber},
    Year = {1993}}
```

You can then include your database file in your main document and cite the new source (`Doe1993`).

```
\documentclass{article}

% Include the biblatex package and tell it to use biber as a backend.
% Without specifying the backend, it assumes biber.
\usepackage[backend=biber]{biblatex}

% Define where biber can find your sources
\addbibresource{sources.bib}

\begin{document}
"Biber isn't that difficult." \cite{Doe1993}
% Use \cite{source-ID} to generate a citation

% Print the bibliography
\printbibliography

\end{document}
```

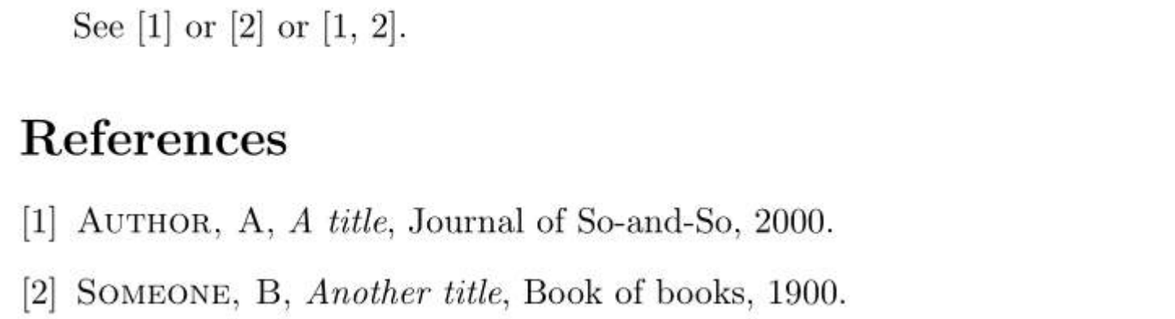
To compile the document, you will need to run 3 commands in sequence:

1. `pdflatex` to create an auxiliary file which tells biber what sources are needed
2. `biber` to create an auxiliary file with all the sources which can be used by `pdflatex`
3. `pdflatex` to include the auxiliary file and create the PDF



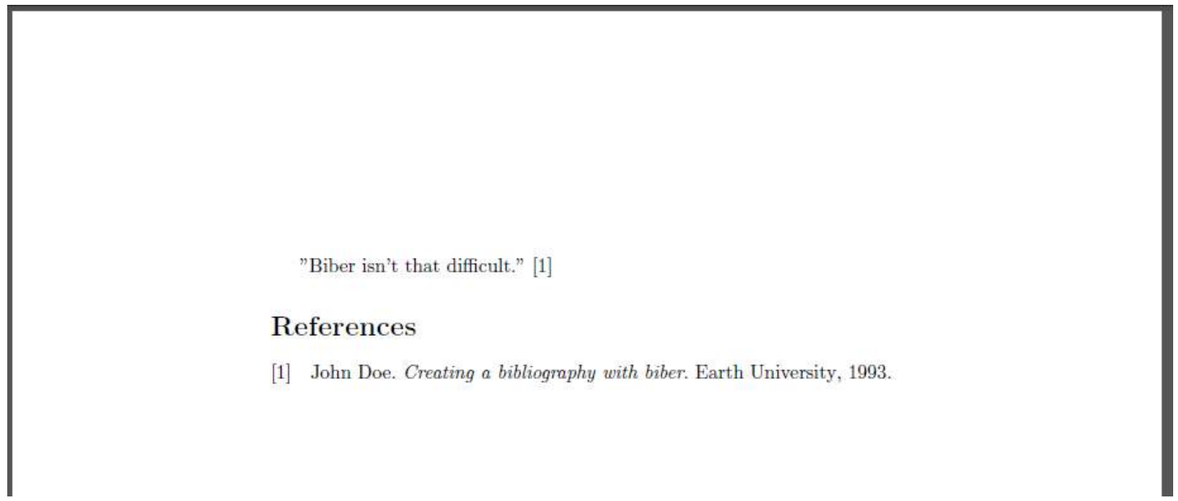
在CTAN上的包文档中可以找到更多选项和bib文件的附加字段。

第7.2节：无包的基本参考文献（手动格式化）



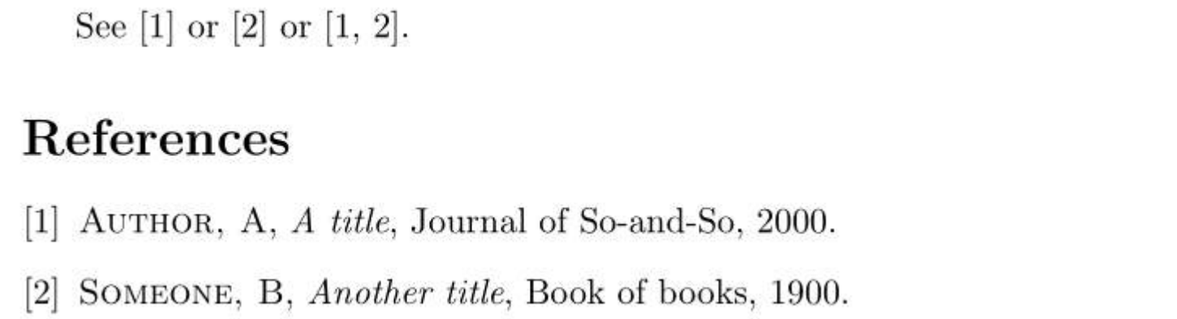

```
\documentclass{article}% 或 book, report, ...

\begin{document}

参见\cite{citeA}或\cite{citeB}或\cite{citeA, citeB}。

\begin{thebibliography}{x}
  % \bibitem{<biblabel>} <citation>
  \bibitem{citeA}
    {\scshape 作者, A}, {\itshape 一个标题}, 某某期刊, 2000年。
  \bibitem{citeB}
    {\scshape 某人, B}, {\itshape 另一个标题}, 书籍之书, 1900年。
\end{thebibliography}

\end{document}
```

请注意，除非你真的知道*为什么*，否则最好不要这样做。使用指定的宏包（参见其他示例）更为合适。



Find many more options and additional fields for bib files in the package documentation on CTAN.

Section 7.2: Basic bibliography without packages (manual formatting)




```
\documentclass{article}% or book, report, ...

\begin{document}

See \cite{citeA} or \cite{citeB} or \cite{citeA, citeB}.

\begin{thebibliography}{x}
  % \bibitem{<biblabel>} <citation>
  \bibitem{citeA}
    {\scshape Author, A}, {\itshape A title}, Journal of So-and-So, 2000.
  \bibitem{citeB}
    {\scshape Someone, B}, {\itshape Another title}, Book of books, 1900.
\end{thebibliography}

\end{document}
```

Note that unless you really know *why*, you should probably not do this. Using designated packages (see other examples) is preferable.

第8章：添加引用

第8.1节：向已有的LaTeX文档添加引用

在文档末尾添加以下内容：

\bibliographystyle{style}

\bibliography{file location}

创建一个扩展名为.bib的文件，并按如下方式保存引用：

```
@inproceedings{citation_name,
   title={论文标题},
author={作者列表},
   pages={45--48},
   year={2013},
organization={组织名称}
}
```

引用时使用以下格式：\citet{citation_name}

Chapter 8: Add Citation

Section 8.1: Add citation to already existing LaTeX document

At the end of the document add the following:

\bibliographystyle{*style*}

\bibliography{*file location*}

Create a file with extension *.bib* and save the citation as follows:

```
@inproceedings{citation_name,
  title={Paper Title},
  author={List Authors},
  pages={45--48},
  year={2013},
  organization={organization name}
}
```

To cite use the following: **\citet{*citation_name*}**

第9章：计数器、条件语句和LaTeX中的循环

第9.1节：计数器的操作

此示例展示了如何使用计数器进行数学运算。它可能对 LaTeX 中的循环有用。

加法：\addtocounter{num}{n}

此命令将 n 加到 num 上，其中 num 是计数器，n 是正整数。

减法：\addtocounter{num}{-n}

此命令从 num 中减去 n，其中 num 是计数器，n 是正整数。

乘法：\multiply\value{num} 乘以 n

此命令将 num 乘以 n，其中 num 是计数器，n 是整数。

除法 \divide\value{num} 除以 n

此命令将 num 除以 n 并取商的整数部分（num 是计数器，n 是整数）

```
\documentclass{article}
    \begin{document}
    ewcounter{num}
    \setcounter{num}{3}
    \addtocounter{num}{10}
    henum\\%打印 13
    \addtocounter{num}{-3}
    henum\\%打印 10
    \stepcounter{num}
    henum\\%打印 11
    \multiply\value{num} 乘以 \value{num}
    henum\\%打印 121
    \multiply\value{num} 乘以 2
    henum\\%打印 242
    \divide\value{num} 除以 60
    henum%打印 4
\end{document}
```

\newcommand{num} 声明计数器。 \setcounter{num}{3} 将 num 值设为 3。

\addtocounter{num}{10} 给 num 加 10。

\addtocounter{num}{-3} 从 num 中减去 3。

\stepcounter{num} 给 num 加 1

\multiply\value{num} 乘以 \value{num} 计算 num 的平方。

\multiply\value{num} 乘以 2 将 num 翻倍。

\divide\value{num} 除以 60 将 num 除以 60 并取整数部分。

代码的结果：13\\10\\11\\121\\242\\4

Chapter 9: Counters, if statements and loops with LaTeX

Section 9.1: Operations with counters

This example shows how to use mathematical operations with counters. It may be useful for loops in latex.

Addition: \addtocounter{num}{n}

this command adds n to num, where num is a counter and n is a positive integer.

Subtraction: \addtocounter{num}{-n}

this command subtracts n from num, where num is a counter and n is a positive integer.

Multiplication: \multiply\value{num} by n

this command multiply num by n, where num is a counter and n is an integer.

Division \divide\value{num} by n

this command divides num by n and gets the integer part of the quotient (num is a counter and n is an integer)

```
\documentclass{article}
    \begin{document}
    \newcounter{num}
    \setcounter{num}{3}
    \addtocounter{num}{10}
    \thenum\\%prints 13
    \addtocounter{num}{-3}
    \thenum\\%prints 10
    \stepcounter{num}
    \thenum\\%prints 11
    \multiply\value{num} by \value{num}
    \thenum\\%prints 121
    \multiply\value{num} by 2
    \thenum\\%prints 242
    \divide\value{num} by 60
    \thenum%prints 4
\end{document}
```

\newcommand{num} declares counter. \setcounter{num}{3} sets num value to 3.

\addtocounter{num}{10} adds 10 to num.

\addtocounter{num}{-3} subtract 3 from num.

\stepcounter{num} adds 1 to num

\multiply\value{num} by \value{num} squares num.

\multiply\value{num} by 2 doubles num.

\divide\value{num} by 60 divides num by 60 and gets the integer part.

The result of the code: 13\\10\\11\\121\\242\\4

(\\ 表示换行)

intcalc 宏包添加了一些其他整数运算，例如 mod、pow、sng、abs、inv ...

intcalc_package.pdf

第9.2节：计数器声明、初始化及打印到pdf

在 LaTeX 中可以使用整数变量。要创建一个新变量，我们需要使用 ewcounter{name}命令，其中 name 是新计数器的名称。 name 必须只包含字母。该命令创建一个名为 hename 的新计数器。通过该命令我们可以将 name 变量打印到文档中。 name 的初始值为 0。要给 "name" 赋值，可以使用 \setcounter{name}{n} ，其中 n 是一个整数。 \value{name} 是一个返回 name 值的函数。

```
\documentclass{article}
\begin{document}
ewcounter{num}                    % 新计数器，初始值为 0
henum                           % 打印 0
\setcounter{num}{3}                 % 将 num 设为 3
henum                           % 打印 3
ewcounter{number}
\setcounter{number}{\value{num}}    % 将 number 设为 num 的值
henumber                        % 打印 3

Latex 提供了一些其他格式来打印数字。

其他类型的打印：

\arabic{num}\\
\Roman{num}\\ %→ I, II, III, IV, . . . (num = 1, 2, 3, . . . )
\roman{num}\\ %→ i, ii, iii, iv, . . . (num = 1, 2, 3, . . . )
\Alph{num}\\  %→ A, B, C, D, . . . (num = 1, 2, 3, . . . , 26)
\alph{num}\\  %→ a, b, c, d, . . . (num = 1, 2, 3, . . . , 26)
\fnsymbol{num}\\ %→ *, †, ‡, §, ¶, k, **, ††, ‡‡ (num = 1, 2, 3, . . . , 9)
\end{document}
```

0
3
3
Latex provides some other formats to print a number.
Other types of printing:
3
III
iii
C
c
‡

第9.3节：条件语句

在LaTeX中，我们可以使用内置命令来根据条件是否成立执行代码。

比较两个整数：`\ifnum\value{num}>n {A} \else {B}\fi`

这段代码在num>n时执行A，否则执行B。我们可以用<和=替换>。

(\\ symbolizes new line)

intcalc package adds some other integer operations e.g. mod, pow, sng, abs, inv ...

intcalc_package.pdf

Section 9.2: Counter declaration, initialization and printing to pdf

It is possible to use integer variables with latex. To create a new variable we need the `\newcounter{name}` command, where `name` is the name of the new counter. The `name` must contain only letters. This command creates a new one with name `\thename`. With this command we can print `name` variable onto the paper. The initial value of `name` is 0. To give value to "name" we can use `\setcounter{name}{n}` where n is an integer. `\value{name}` is a function which returns with the value of `name`.

```
\documentclass{article}
\begin{document}
\newcounter{num}                    %new counter, initial value is 0
\thenum                             %print 0
\setcounter{num}{3}                 %set num to 3
\thenum                             %print 3
\newcounter{number}
\setcounter{number}{\value{num}}    %set number to value of num
\thenumber                          %print 3

Latex provides some other formats to print a number.

Other types of printing:

\arabic{num}\\
\Roman{num}\\ %→ I, II, III, IV, . . . (num = 1, 2, 3, . . . )
\roman{num}\\ %→ i, ii, iii, iv, . . . (num = 1, 2, 3, . . . )
\Alph{num}\\  %→ A, B, C, D, . . . (num = 1, 2, 3, . . . , 26)
\alph{num}\\  %→ a, b, c, d, . . . (num = 1, 2, 3, . . . , 26)
\fnsymbol{num}\\ %→ *, †, ‡, §, ¶, k, **, ††, ‡‡ (num = 1, 2, 3, . . . , 9)
\end{document}
```

0
3
3
Latex provides some other formats to print a number.
Other types of printing:
3
III
iii
C
c
‡

Section 9.3: If statements

In latex we can use built-in commands to execute code whether the conditions are true or not.

Comparing two integers: `\ifnum\value{num}>n {A} \else {B}\fi`

This code executes A if num>n else B. We can substitute > with < and =.

判断一个数是否为奇数：`\ifodd\value{num} {A}\else {B}\fi`

如果num是奇数，则执行A，否则执行B。

带条件的if语句：`\ifthenelse{condition}{A}{B}`

使用此命令需要加载ifthen宏包。如果条件为真，则执行A，否则执行B。

可以使用\(\)、\AND、\OR、\NOT来创建复杂条件。

例如：\ifthenelse{\(\NOT 4<2 \OR 4>11\)\AND\isodd{4}}{A}{B}这段代码会在页面

上写下"B"。\NOT 4<2为真，4>11为假。如果用"OR"连接一个假和一个真，结果为真。所以\(\NOT 4<2 \OR 4>11\)为真。\isodd{4}为假，因为4是偶数。
一个假命题和一个真命题通过"与"连接时结果为假，因此输出为B。

示例代码：

```
\documentclass{article}
\usepackage{ifthen}
\begin{document}
      ewcounter{num}
      \setcounter{num}{10}

如果 num$>$100，则下一句为"Num is large."，否则为"Num is small."

    Num 是 \ifnum \value{num}>100 {large} \else {small}.

如果 num 是奇数，则下一句以"Odd"开头，否则以"Even"开头

     \ifodd \value{num} {Odd} \else {Even} 数字很酷。

如果 (num$>$3 且 (1$<$0 或 num$=$10)) 为真，则下一句为"True."，否则为
"False."

     \ifthenelse{\value{num}>3\AND\(1<0 \OR \value{num}=10\)}{True.}{False.}


\end{document}
```

```
    If num>100 then the next sentence will be "Num is large." else "Num is
small."
    Num is small.
    If num is odd then the next sentence will begin with "Odd", if not then with
"Even"
    Even numbers are cool.
    If (num>3 and (1<0 or num=10)) is true then the next sentence will be
"True." else "False."
    True.
```

第9.4节：循环——重复操作

我们可以在 LaTeX 中创建循环。它们类似于其他编程语言中的循环，但没有那么多自定义选项。
使用循环的一个替代方案是 @loops。如果我们使用名称中包含"@"的命令，必须将其放在\makeatletter和\makeatother之间。不允许在描述新定义的宏中使用它们。

错误示范：

If a number is odd: `\ifodd\value{num} {A}\else {B}\fi`

If num is odd then it executes A else B.

If with condition: `\ifthenelse{condition}{A}{B}`

We have to load ifthen package to use this command. If condition are true then it executes A else B.

It is possible to create complex condition with `\( \)`, `\AND`, `\OR`, `\NOT`.

For example: `\ifthenelse{\(\NOT 4<2 \OR 4>11\)\AND\isodd{4}}{A}{B}`

This piece of code writes down "B" on the page. `\NOT` 4<2 is true and 4>11 is false. If we connect a false and a true statement with "OR" then the result is true. So `\(\NOT 4<2 \OR 4>11\)` is true. `\isodd{4}` is false because 4 is even. A false and a true statement connected with "AND" is false, so the output is B.

An example code:

```
\documentclass{article}
\usepackage{ifthen}
\begin{document}
    \newcounter{num}
    \setcounter{num}{10}

    If num$>$100 then the next sentence will be "Num is large." else "Num is small."

    Num is \ifnum \value{num}>100 {large} \else {small}.

    If num is odd then the next sentence will begin with "Odd" if not then with "Even"

    \ifodd \value{num} {Odd} \else {Even} numbers are cool.

    If (num$>$3 and (1$<$0 or num$=$10)) is true then the next sentence will be "True." else
"False."

    \ifthenelse{\value{num}>3\AND\(1<0 \OR \value{num}=10\)}{True.}{False.}


\end{document}
```

```
    If num>100 then the next sentence will be "Num is large." else "Num is
small."
    Num is small.
    If num is odd then the next sentence will begin with "Odd", if not then with
"Even"
    Even numbers are cool.
    If (num>3 and (1<0 or num=10)) is true then the next sentence will be
"True." else "False."
    True.
```

Section 9.4: Loops - repeating things

We can create loops in latex. They are similar but not as customizable as loops in other programming languages. One alternative to use loops are @loops. If we use a command which includes "@" in its name, we must be put it between `\makeatletter` and `\makeatother`. It is not allowed to use them in a macro which describes a new definition.

Wrong:

```
\def\is#1#2{\makeatletter\@ifstar{#1}{#2}\makeatother
```

正确示范：

```
\makeatletter\def\is#1#2{\@ifstar{#1}{#2}}\makeatother
```

@for 循环：`\@for\command:={list}\do{commands}`

示例：

```
\makeatletter
\@for\sun:={rising,setting}\do{太阳是 \sun。}
\makeatother
```

它生成以下文本：太阳正在升起。太阳正在落下。

@whilenum 循环：`\@whilenum 条件\do{命令}`

示例：

```
\makeatletter
ewcounter{int}
\@whilenum\value{int}<10\do
{\stepcounter{int}\ifthenelse{\isodd{\value{int}}}{heint}{}}
\makeatother
```

此代码输出从1到9的奇数。

“loop repeat” 循环：`\loop {命令} \ifnum 条件 \repeat`

执行命令直到条件为真。

示例

```
\setcounter{int}{1}
\loop
heint
\addtocounter{int}{2}
\ifnum \value{int}<10
\repeat
```

这段代码的功能与 @whilenum 循环相同。

示例代码：

```
\documentclass{article}
\usepackage{ifthen}
usepackage{amsmath} %ext{} 命令需要此宏包
\begin{document}
@for 循环演示：

    \makeatletter
    \@for\sun:={rising,setting}\do{太阳是 \sun。 }
    \makeatother

    ewcounter{int}

@whilenum 循环：
```



```
\def\is#1#2{\makeatletter\@ifstar{#1}{#2}\makeatother
```

Right:

```
\makeatletter\def\is#1#2{\@ifstar{#1}{#2}}\makeatother
```

@for loop: `\@for\command:={list}\do{commands}`

Example:

```
\makeatletter
\@for\sun:={rising,setting}\do{The sun is \sun.}
\makeatother
```

It creates the following text: The sun is rising. The sun is setting.

@whilenum loop: `\@whilenum condition\do{commands}`

Example:

```
\makeatletter
\newcounter{int}
\@whilenum\value{int}<10\do
{\stepcounter{int}\ifthenelse{\isodd{\value{int}}}{\theint}{}}
\makeatother
```

This code writes odd numbers from 1 to 9.

"loop repeat" loop: `\loop {commands} \ifnum condition \repeat`

Executes commands till condition is true.

Example

```
\setcounter{int}{1}
\loop
\theint
\addtocounter{int}{2}
\ifnum \value{int}<10
\repeat
```

This code does the same as @whilenum loop.

An example code:

```
\documentclass{article}
\usepackage{ifthen}
\usepackage{amsmath} %\text{} command needs this package
\begin{document}
    Demonstration of @for loop:

    \makeatletter
    \@for\sun:={rising,setting}\do{The sun is \sun. }
    \makeatother

    \newcounter{int}

    @whilenum loop:
```

```
    \setcounter{int}{0}
    \makeatletter
    \@whilenum\value{int}<20\do
    {\stepcounter{int}\ifthenelse{\isodd{\value{int}}}{heintext{ }}{}}
    \makeatother

“循环重复”循环：

    \setcounter{int}{1}
    \loop
    heint
    ext{ }\addtocounter{int}{2}\ifnum\value{int}<20
    \repeat
\end{document}
```

```
Demonstration of @for loop:
The sun is rising. The sun is setting.
@whilenum loop:
1 3 5 7 9 11 13 15 17 19
"loop repeat" loop:
1 3 5 7 9 11 13 15 17 19
```

第9.5节：在Tikz中使用循环

循环在Tikz中很有用。

下面的代码绘制了一个没有数字的时钟：

```
\documentclass{article}
\usepackage{ifthen}
\usepackage{intcalc}
\usepackage{tikz}
ewcounter{num}

\begin{document}
\begin{tikzpicture}
    \makeatletter
    \setcounter{num}{1}
    ewcounter{angle}
    \draw (0,0) circle (3cm);
    \@whilenum\value{num}<13\do{
    \setcounter{angle}{360}
    \multiply\value{angle} by \value{num}
    \divide\value{angle} by 12
    \ifnum \intcalcMod{\value{num}}{3}=0{
    \draw[line width=4pt] (heangle:2cm) -- (heangle:3cm);    }\else
    {
    \draw[line width=1pt] (heangle:2.3cm) -- (heangle:3cm);
    }\fi
    \addtocounter{num}{1}
    }
    \makeatother
\end{tikzpicture}
\end{document}
```

结果：


```
    \setcounter{int}{0}
    \makeatletter
    \@whilenum\value{int}<20\do
    {\stepcounter{int}\ifthenelse{\isodd{\value{int}}}{\theint\text{ }}{}}
    \makeatother

    "loop repeat" loop:

    \setcounter{int}{1}
    \loop
    \theint
    \text{ }\addtocounter{int}{2}\ifnum\value{int}<20
    \repeat
\end{document}
```

```
Demonstration of @for loop:
The sun is rising. The sun is setting.
@whilenum loop:
1 3 5 7 9 11 13 15 17 19
"loop repeat" loop:
1 3 5 7 9 11 13 15 17 19
```

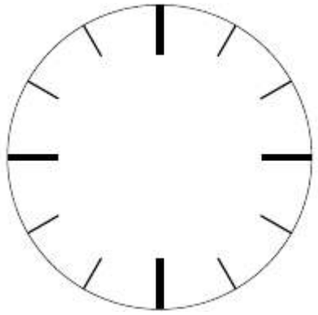
Section 9.5: Using loops in Tikz

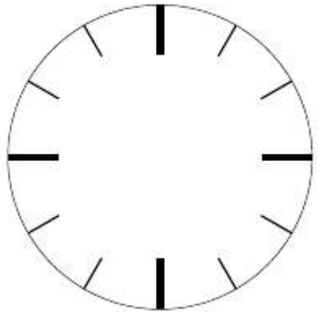
Loops are useful in Tikz.

The following code draws a clock without numbers:

```
\documentclass{article}
\usepackage{ifthen}
\usepackage{intcalc}
\usepackage{tikz}
\newcounter{num}

\begin{document}
\begin{tikzpicture}
    \makeatletter
    \setcounter{num}{1}
    \newcounter{angle}
    \draw (0,0) circle (3cm);
    \@whilenum\value{num}<13\do{
    \setcounter{angle}{360}
    \multiply\value{angle} by \value{num}
    \divide\value{angle} by 12
    \ifnum \intcalcMod{\value{num}}{3}=0{
    \draw[line width=4pt] (\theangle:2cm) -- (\theangle:3cm);    }\else
    {
    \draw[line width=1pt] (\theangle:2.3cm) -- (\theangle:3cm);
    }\fi
    \addtocounter{num}{1}
    }
    \makeatother
\end{tikzpicture}
\end{document}
```

The result:





第10章：文档类

第10.1节：文章类

```
\documentclass{article}
```

何时使用文章类？

用于科学期刊文章、演示文稿、简短报告、程序文档、邀请函等..... 1

该类的特点是什么？

文章类不包含章节或部分。它可以分为节、小节和段落等。

默认情况下，标题显示在首页顶部，而不是单独的标题页上。

简单示例

```
\documentclass{article}

\title{Hello world}
\author{Me    }
\date{\today}

\begin{document}

\maketitle

你好，世界！
\end{document}
```

第10.2节：Beamer

```
\documentclass{beamer}
```

何时使用beamer类？

用于演示幻灯片。

该类的特点是什么？

输出为横向。文档被分成“帧”（幻灯片）。

简单示例

以下示例改编自：sharelatex.com/learn/Beamer

```
\documentclass{beamer}

\usepackage[utf8]{inputenc}

\title{Sample title}
author{我}
\date{\today}

\begin{document}

\frame{itlepage}
```


Chapter 10: Document Classes

Section 10.1: Article

```
\documentclass{article}
```

When to use the article class ?

For articles in scientific journals, presentations, short reports, program documentation, invitations, ... 1

What are the specificities of this class ?

An article doesn't contain chapters or parts. It can be divided in sections, subsections and paragraphs etc.

By default, the title is shown at the top of the first page, and not on a separate title page.

Simple example

```
\documentclass{article}

\title{Hello world}
\author{Me    }
\date{\today}

\begin{document}

\maketitle

Hello, World!
\end{document}
```

Section 10.2: Beamer

```
\documentclass{beamer}
```

When to use the beamer class ?

For presentation slides.

What are the specificities of this class ?

The output is landscape-oriented. The document is separated in "frames" (slides).

Simple example

Following example was adapted from : sharelatex.com/learn/Beamer

```
\documentclass{beamer}

\usepackage[utf8]{inputenc}

\title{Sample title}
\author{Me}
\date{\today}

\begin{document}

\frame{\titlepage}
```

```
\begin{frame}
\frametitle{示例幻灯片标题}
这是第一张幻灯片中的文本。这是第一张幻灯片中的文本。这是第一张幻灯片中的文本。
\end{frame}

\end{document}
```

第10.3节：定义文档类

每个LaTeX程序的第一行都应该这样写。它应遵循形式\documentclass{...}。
你放在大括号内的内容非常重要。有些文档类会提供额外的命令供你使用，
其他文档类使用不同的格式，且所有文档类都有你可以输入的特定参数（在参数部分有描述）。


```
\begin{frame}
\frametitle{Sample frame title}
This is a text in first frame. This is a text in first frame. This is a text in first frame.
\end{frame}

\end{document}
```

Section 10.3: Defining the document class

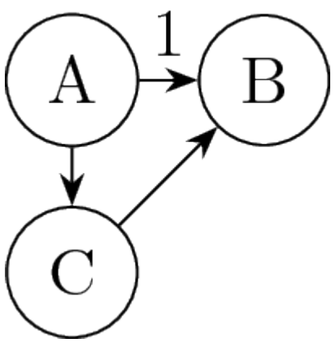
The very first line in each of your LaTeX programs should do this. It should follow the form `\documentclass{...}`. What you put within the curly braces is very important. Some document classes give you extra commands to use, others use a different format, and all have specific parameters you can input (described in the parameters section).

第11章：绘制图形

第11.1节：TikZ——图形规格说明

TikZ 提供了类似于 DOT 的语法，可以大大简化你的图形绘制代码。

```
\documentclass{standalone}

\usepackage{tikz}
\usetikzlibrary{graphs,quotes,arrows.meta}

\begin{document}
  \begin{tikzpicture}
    \graph[nodes={draw,circle},edges={-{Stealth[]}}] {
      A -> ["1"] B,
      A -> C,
C -> B
    };
  \end{tikzpicture}
\end{document}
```



如你所见，你用更简单的语法换取了细粒度的控制。 当你指定更复杂的图时，graphs 库真正发挥了作用：

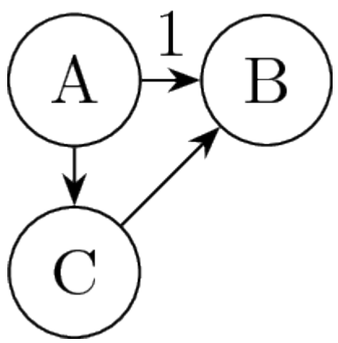
```
\documentclass{standalone}

\usepackage{tikz}
\usetikzlibrary{graphs,graphs.standard}

\begin{document}
  \begin{tikzpicture}
    \graph {
A -> { 子图 I_n [V= {B,C,D}] } -> E
    };
  \end{tikzpicture}
\end{document}
```


Chapter 11: Drawing Graphs

Section 11.1: TikZ -- Graph specifications

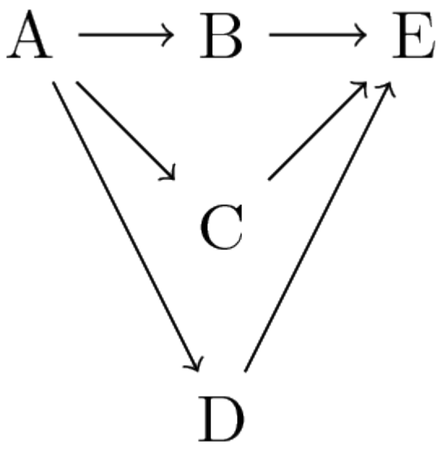
TikZ provides syntax similar to DOT which you can use to tighten up your graph drawing code considerably.

```
\documentclass{standalone}

\usepackage{tikz}
\usetikzlibrary{graphs,quotes,arrows.meta}

\begin{document}
  \begin{tikzpicture}
    \graph[nodes={draw,circle},edges={-{Stealth[]}}] {
      A -> ["1"] B,
      A -> C,
      C -> B
    };
  \end{tikzpicture}
\end{document}
```



As you can see, you trade fine-grained control for easier syntax. The `graphs` library really shines when you specify more complicated graphs:

```
\documentclass{standalone}

\usepackage{tikz}
\usetikzlibrary{graphs,graphs.standard}

\begin{document}
  \begin{tikzpicture}
    \graph {
      A -> { subgraph I_n [V= {B,C,D}] } -> E
    };
  \end{tikzpicture}
\end{document}
```

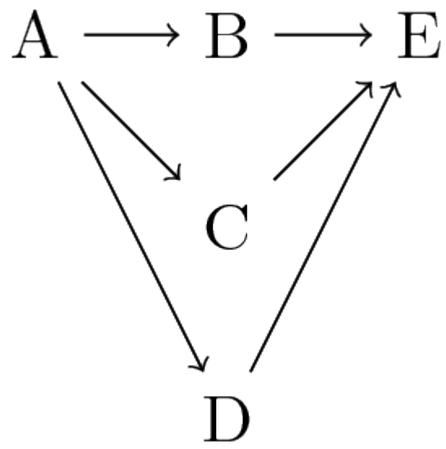


还有更多选项和预定义图；详见 TikZ 手册第19节。

第11.2节：TikZ —— 算法图绘制

TikZ 实现了多种自动图布局算法（需要 LuaLaTeX）。

```
\documentclass{article}

\usepackage{tikz}
\usetikzlibrary{graphs,graphdrawing,quotes}
\usegdlibrary{force}

\begin{document}
  \begin{tikzpicture}
    \graph[spring layout] {
      A -> ["1"] B,
      A -> {C, D},
      C -> {B, D},
    };
  \end{tikzpicture}
\end{document}
```

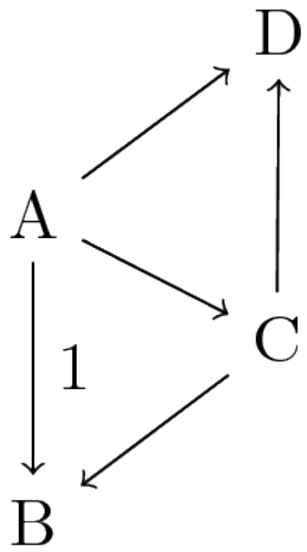



There are many more options and pre-defined graphs; see section 19 of the TikZ manual.

Section 11.2: TikZ -- Algorithmic graph drawing

TikZ implements several algorithms for *automatic* graph layouts (requires LuaLaTeX).

```
\documentclass{article}

\usepackage{tikz}
\usetikzlibrary{graphs,graphdrawing,quotes}
\usegdlibrary{force}

\begin{document}
  \begin{tikzpicture}
    \graph[spring layout] {
      A -> ["1"] B,
      A -> {C, D},
      C -> {B, D},
    };
  \end{tikzpicture}
\end{document}
```



有多种算法和许多选项可供调整。详情请参见 TikZ 手册第四部分。

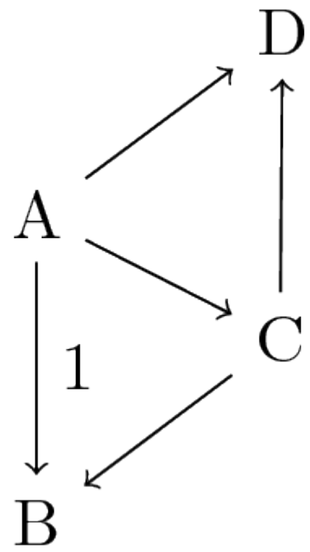
第11.3节：马尔可夫链的状态转移图

假设以下矩阵是与马尔可夫链相关的转移概率矩阵。

```
0.5   0.2  0.3
P=  0.0   0.1  0.9
    0.0   0.0  1.0
```

为了研究马尔可夫链状态的性质，绘制了马尔可夫链的状态转移图。

```
\documentclass[12pt,a4paper]{article}
\usepackage{tikz}
\usetikzlibrary{shapes,arrows,positioning}
\begin{tikzpicture}[->,>=stealth',shorten >=2pt, line width=3pt,
                                node distance=2cm, style ={minimum size=20mm}]
ikzstyle{every node}=[font=\huge]
ode [circle, draw] (a) {1};
\path  (a) edge [loop above] (a);
ode [circle, draw] (b) [right=of a] {2};
\path  (b) edge [loop above] (b);
\draw[->] (a) -- (b);
ode [circle, draw] (c) [below=of a] {3};
\path  (c) edge [loop below] (c);
\draw[->] (a) -- (c);
\draw[->] (b) -- (c);
\end{tikzpicture}
```

There are several algorithms and many options to influence them. See part IV of the TikZ manual for details.

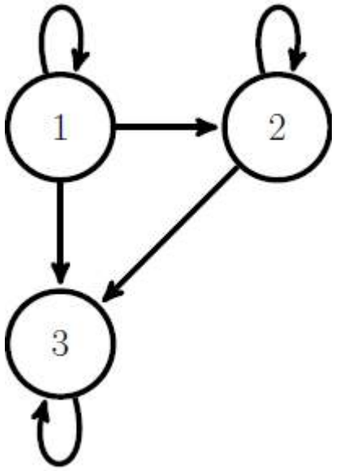
Section 11.3: State Transition Diagram of a Markov Chain

Suppose the following matrix is the transition probability matrix associated with a Markov chain.

```
    0.5   0.2  0.3
P=  0.0   0.1  0.9
    0.0   0.0  1.0
```

In order to study the nature of the states of a Markov chain, a state transition diagram of the Markov chain is drawn.

```
\documentclass[12pt,a4paper]{article}
\usepackage{tikz}
\usetikzlibrary{shapes,arrows,positioning}
\begin{tikzpicture}[->,>=stealth',shorten >=2pt, line width=3pt,
                                node distance=2cm, style ={minimum size=20mm}]
\tikzstyle{every node}=[font=\huge]
\node [circle, draw] (a) {1};
\path  (a) edge [loop above] (a);
\node [circle, draw] (b) [right=of a] {2};
\path  (b) edge [loop above] (b);
\draw[->] (a) -- (b);
\node [circle, draw] (c) [below=of a] {3};
\path  (c) edge [loop below] (c);
\draw[->] (a) -- (c);
\draw[->] (b) -- (c);
\end{tikzpicture}
```

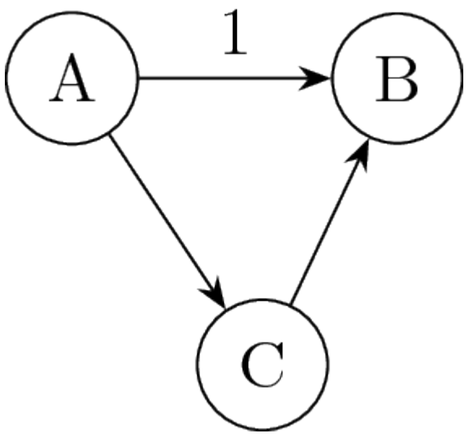


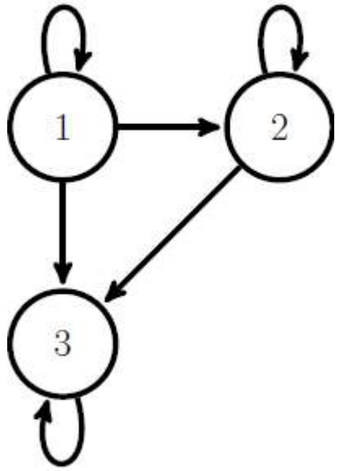
第11.4节：TikZ —— 手动布局

TikZ包非常适合绘制图形。

这是一个小示例（需要 TikZ 3.0 及以上版本）：

```
\documentclass{standalone}

\usepackage{tikz}
\usetikzlibrary{positioning,arrows.meta}

\begin{document}
  \begin{tikzpicture}[auto,vertex/.style={draw,circle}]
    ode[vertex] (a) {A};
    ode[vertex,right=1cm of a] (b) {B};
    ode[vertex,below right=1cm and 0.5cm of a] (c) {C};

    \path[-{Stealth[]}]
(a) edge node {1} (b)
     (a) edge (c)
(c) edge (b);
  \end{tikzpicture}
\end{document}
```



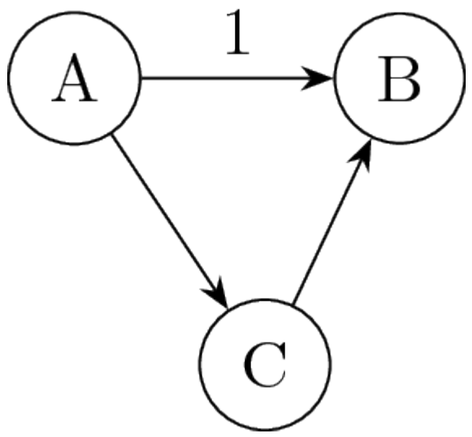


Section 11.4: TikZ -- Manual layout

Package TikZ lends itself very well to drawing graphs.

This is a small example (requires TikZ 3.0+):

```
\documentclass{standalone}

\usepackage{tikz}
\usetikzlibrary{positioning,arrows.meta}

\begin{document}
  \begin{tikzpicture}[auto,vertex/.style={draw,circle}]
    \node[vertex] (a) {A};
    \node[vertex,right=1cm of a] (b) {B};
    \node[vertex,below right=1cm and 0.5cm of a] (c) {C};

    \path[-{Stealth[]}]
      (a) edge node {1} (b)
      (a) edge (c)
      (c) edge (b);
  \end{tikzpicture}
\end{document}
```



你可以创建任意复杂的图形；但要注意代码长度。请记住有 \foreach，并注意所有定位和样式选项（参见 TikZ 手册，第13至17节）。

You can create arbitrarily complex graphs; beware lengthy code, though. Recall that there is `\foreach` and take note of all the positioning and styling options (cf. TikZ manual, section 13 to 17).

第12章：使用 beamer 宏包进行演示

主题 **AnnArbor**
颜色主题 seahoarse

第12.1节：简单的单作者标题幻灯片

```
\documentclass{beamer}

\mode<presentation>

\usetheme{AnnArbor}

\usecolortheme{seahorse}

\title[简短主题]{精彩的长主题}

\author[姓名]{全名}

\institute[机构简称]{机构全称}

\date{\today}

\begin{document}

\maketitle

\end{document}
```


Chapter 12: Presentation with beamer package

theme **AnnArbor**
color theme seahorse

Section 12.1: Simple one author title slide

```
\documentclass{beamer}

\mode<presentation>

\usetheme{AnnArbor}

\usecolortheme{seahorse}

\title[Short topic]{Awesome long topic}

\author[Name]{Full name}

\institute[Institute short form]{Full name of institute}

\date{\today}

\begin{document}

\maketitle

\end{document}
```

第12.2节：多作者及隶属关系标题幻灯片

```
\documentclass[compress]{beamer}
\mode\<presentation>
\title[]{PQRS系统分析的ABCDE}
\author[] {
AA AAAA \inst{1}
      和 BB BBBB \inst{1}
      和 CC CCCC \inst{1}
      和 DD DDDD \inst{1}
      和 EE EEEE\inst{2}
      和 FF FFFF\inst{3}
      和 GG GGGG \inst{3}}
\institute[]
  {
```


Section 12.2: Multiple author and affiliation title slide

```
\documentclass[compress]{beamer}
\mode\<presentation>
\title[]{ABCDE for analysis of PQRS systems}
\author[] {
      AA AAAA \inst{1}
      \and BB BBBB \inst{1}
      \and CC CCCC \inst{1}
      \and DD DDDD \inst{1}
      \and EE EEEE\inst{2}
      \and FF FFFF\inst{3}
      \and GG GGGG \inst{3}}
\institute[]
  {
```

```
    \inst{1}%
XYZ大学紫外线系

    \和

    \inst{2}%
XYZ大学MN系

    \和

    \inst{3}
PQR高级中心

  }

\date[]{\today}

\begin{document}

\begin{frame}

itlepage

\end{frame}

\end{document}
```

ABCDE for analysis of PQRS systems

AA AAAA [1] BB BBBB [1] CC CCCC [1]
DD DDDD [1] EE EEEE[2] FF FFFF[3] GG GGGG [3]

[1]Department of UV, Univ. of XYZ

[2]Department of MN, Univ. of XYZ

[3]Advanced Centre for PQR

July 1, 2017


```
    \inst{1}%
    Department of UV, Univ. of XYZ

    \and

    \inst{2}%
    Department of MN, Univ. of XYZ

    \and

    \inst{3}
    Advanced Centre for PQR

  }

\date[]{\today}

\begin{document}

\begin{frame}

\titlepage

\end{frame}

\end{document}
```

ABCDE for analysis of PQRS systems

AA AAAA [1] BB BBBB [1] CC CCCC [1]
DD DDDD [1] EE EEEE[2] FF FFFF[3] GG GGGG [3]

[1]Department of UV, Univ. of XYZ

[2]Department of MN, Univ. of XYZ

[3]Advanced Centre for PQR

July 1, 2017

第13章：定义宏

参数	详情
`\macro`	要定义的宏
`argcount`	宏期望的参数数量（可选）
替换文本	宏的替换文本。在该文本中#1、#2等将被宏的参数替换。

第13.1节：宏的基本定义

定义一个新的基本命令

宏可以使用\newcommand来定义。例如：

```
\newcommand{\foo}{Just foo, you see?}
```

定义了一个宏\foo，展开为Just foo, you see?。然后它可以像任何内置命令一样使用，例如在该定义之后：

```
他说："\foo"
```

展开为

```
他说："Just foo, you see?"
```

定义一个带参数的新命令

宏也可以带参数。参数的数量作为可选参数给出，位于命令名称和替换文本之间。在替换文本中，参数通过#1、#2等访问。例如：

```
\newcommand{\better}[2]{A #1 is better than a #2.}
\better{solution}{problem} % 结果：A solution is better than a problem
```

重新定义已存在的命令

如果宏已经被定义，\newcommand 会报错。要为已存在的命令给出新的定义，应使用 \renewcommand。除了名称不同，语法完全相同。例如，在上面定义了 \foo 之后，可以使用：

```
\renewcommand{\foo}{请给我另一个 foo。}
```

重新定义后，宏 \foo 不再展开为 你看，只是 foo 吗？，而是展开为 请给我另一个 foo。

Chapter 13: Defining macros

Parameter	Details
`\macro`	The macro to define
`argcount`	The number of arguments the macro expects (optional)
`replacement text`	The replacement text for the macro. Inside that text #1, #2 etc. are replaced with the macro arguments.

Section 13.1: Basic definition of macros

Define a new basic command

A macro can be defined using `\newcommand`. For example:

```
\newcommand{\foo}{Just foo, you see?}
```

defines a macro `\foo` that expands to `Just foo, you see?`. It can then be used like any built-in command, for example after that definition:

```
He said: ``\foo''
```

expands to

```
He said: ``Just foo, you see?''
```

Define a new command with arguments

Macros can also have arguments. The number of arguments is given as optional argument between the command name and the replacement text. In the replacement text, the arguments are accessed with #1, #2 etc. For example:

```
\newcommand{\better}[2]{A #1 is better than a #2.}
\better{solution}{problem} % gives: A solution is better than a problem
```

Redefining an existing command

If a macro has already been defined, `\newcommand` gives an error. To give a new definition for an existing command, `\renewcommand` is used instead. Other than the different name, the syntax is exactly the same. For example, after the definition of `\foo` above, one could use:

```
\renewcommand{\foo}{Another foo, please.}
```

After that redefinition, the macro `\foo` no longer expands to `Just foo, you see?` but to `Another foo, please.`

第14章：构建工具

第14.1节：Arara

Arara 是一个跨平台的自动化工具，专为 TeX 设计。它包含在标准发行版中，因此无需额外安装。它最有效的理解方式是将编译指令记录在 TeX 文件本身中：

```
% arara: pdflatex
\documentclass{article}
\begin{document}
你好，世界
\end{document}
```

不过，这些可能会复杂得多：

```
% arara: pdflatex
% arara: biber
% arara: pdflatex

% 为支持自包含示例，动态生成一个 BibTeX 文件
\begin{filecontents}{references.bib}
@article{dijkstra,
author = {迪克斯特拉，埃兹格·W.},
title = {尽管存在分布式控制的自稳定系统},
 journal = {Commun. ACM},
issue_date = {1974年11月},
 volume = {17},
number = {11},
 month = nov,
year = {1974},
issn = {0001-0782},
pages = {643--644},
numpages = {2},
url = {http://doi.acm.org/10.1145/361179.361202},
 doi = {10.1145/361179.361202},
acmid = {361202},
publisher = {ACM},
address = {纽约州纽约市，美国},
keywords = {分布式控制，错误恢复，和谐合作，多处理，互斥，网络，鲁棒性，自我修复，自我稳定，共享，同步},

}
\end{filecontents}


\documentclass{article}
\usepackage[backend=biber]{biblatex}
\addbibresource{references.bib}

\begin{document}
你好，世界！ \cite{dijkstra}.
\printbibliography
\end{document}
```


Chapter 14: Build Tools

Section 14.1: Arara

Arara is a cross-platform automation tool that's specially designed for TeX. It's included in a standard distribution, so there's no need to install anything additional. It's most effectively understood as a means to record the compilation instructions in the TeX file itself:

```
% arara: pdflatex
\documentclass{article}
\begin{document}
  Hello, world
\end{document}
```

These can be much more complicated, though:

```
% arara: pdflatex
% arara: biber
% arara: pdflatex

% To support a self-contained example, this builds a BibTeX file on-the-fly
\begin{filecontents}{references.bib}
@article{dijkstra,
 author = {Dijkstra, Edsger W.},
 title = {Self-stabilizing Systems in Spite of Distributed Control},
 journal = {Commun. ACM},
 issue_date = {Nov. 1974},
 volume = {17},
 number = {11},
 month = nov,
 year = {1974},
 issn = {0001-0782},
 pages = {643--644},
 numpages = {2},
 url = {http://doi.acm.org/10.1145/361179.361202},
 doi = {10.1145/361179.361202},
 acmid = {361202},
 publisher = {ACM},
 address = {New York, NY, USA},
 keywords = {distributed control, error recovery, harmonious cooperation, multiprocessing, mutual
exclusion, networks, robustness, self-repair, self-stabilization, sharing, synchronization},
}
\end{filecontents}


\documentclass{article}
\usepackage[backend=biber]{biblatex}
\addbibresource{references.bib}

\begin{document}
Hello, World! \cite{dijkstra}.
\printbibliography
\end{document}
```

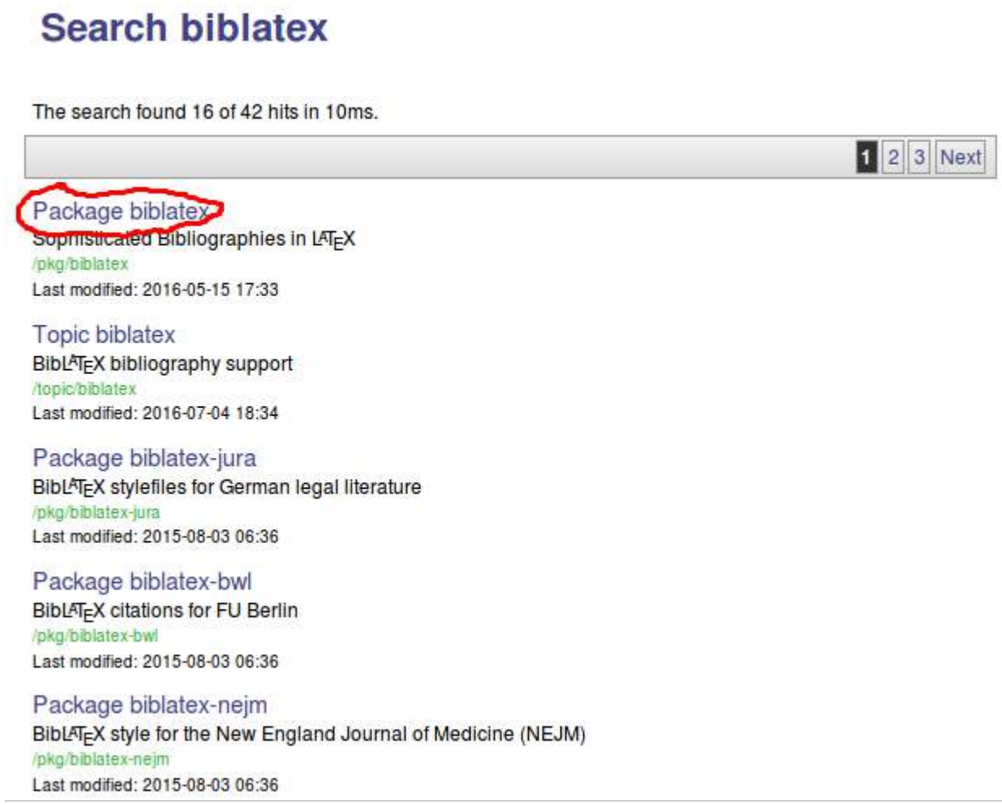
第15章：访问LaTeX宏包的文档

第15.1节：CTAN

综合TeX档案网络（CTAN）确实是LaTeX宏包的综合仓库。
几乎所有优质宏包（甚至更多）都收录于此，且所有优秀的宏包都附带文档。

1. 在搜索栏输入宏包名称。

2. 从列表中选择宏包。



3. 访问文档文件。

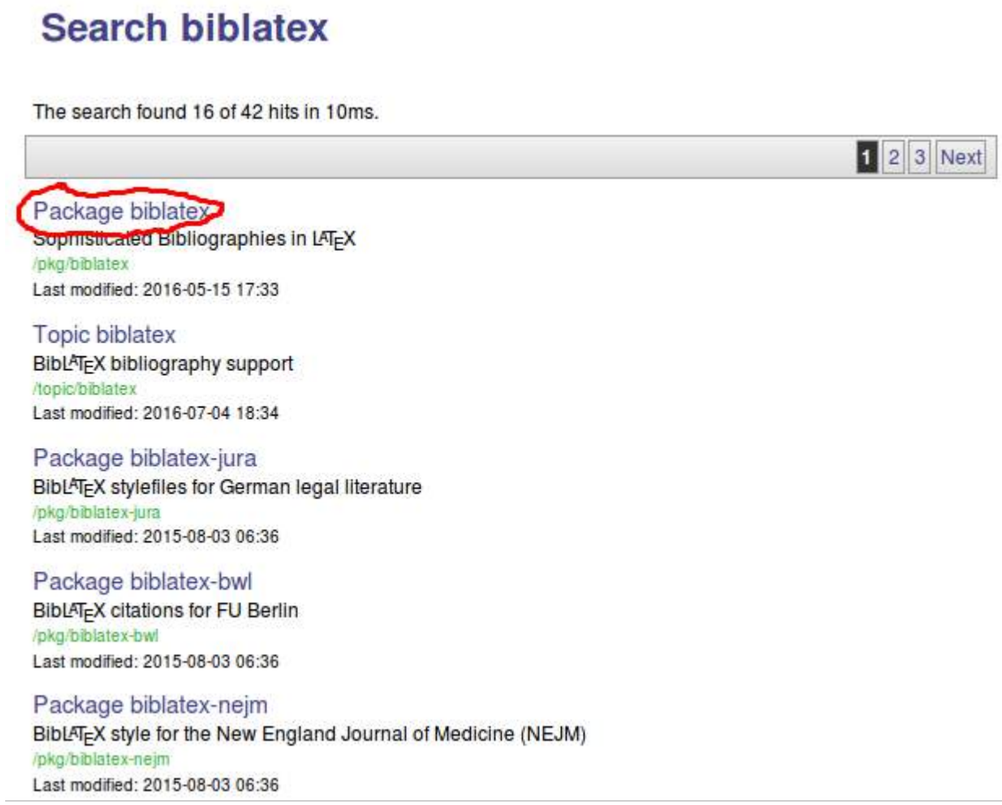
Chapter 15: Accessing documentation of LaTeX packages

Section 15.1: CTAN

The Comprehensive TeX Archive Network (CTAN) is indeed that, *the* comprehensive repository of LaTeX packages. Most if not all quality packages (and more) are on there, and all the good ones include documentation.

1. Enter the package name into the search bar.

2. Select the package from the list.



3. Access the documentation documents.

BibLaTeX – Sophisticated Bibliographies in LaTeX

BibLaTeX is a complete reimplementation of the bibliographic facilities provided by LaTeX. Formatting of the bibliography is entirely controlled by LaTeX macros, and a working knowledge of LaTeX should be sufficient to design new bibliography and citation styles. BibLaTeX uses its own data backend program called "biber" to read and process the bibliographic data. With biber, BibLaTeX has many features rivalling or surpassing other bibliography systems. To mention a few:

- Full Unicode support
- Highly customisable sorting using the Unicode Collation Algorithm + CLDR tailoring
- Highly customisable bibliography labels
- Complex macro-based on-the-fly data modification without changing your data sources
- A tool mode for transforming bibliographic data sources
- Multiple bibliographies and lists of bibliographic information in the same document with different sorting
- Highly customisable data source inheritance rules
- Polyglossia and babel suppport for automatic language switching for bibliographic entries and citations
- Automatic bibliography data recoding (UTF-8 -> latin1, LaTeX macros -> UTF-8 etc)
- Remote data sources
- Highly sophisticated automatic name and name list disambiguation system
- Highly customisable data model so users can define their own bibliographic data types
- Validation of bibliographic data against a data model
- Subdivided and/or filtered bibligraphies, bibliographies per chapter, section etc.

Apart from the features unique to BibLaTeX, the package also incorporates core features of the following packages: babelbib, bibtopic, bibunits, chapterbib, cite, inlinebib, mcite and mciteplus, mlbib, multibib, splitbib.

Sources	/macros/latex/contrib/biblatex
Documentation	Readme Package documentation (English) Release notes for current version
Version	3.4
License	The LaTeX Project Public License 1.3
Copyright	2012–2016 Philipp Lehman, Joseph Wright, Audrey Boruvka, Philip Kime 2006–2012 Philipp Lehman
Maintainer	Philipp Lehman (inactive) Philip Kime
Contained in	TeX Live as biblatex MiKTeX as biblatex
Topics	BibLaTeX bibliography support bibliography processor

Download the contents of this package in one zip archive (17.2M).

重要提示： CTAN保存的是最新版本。如果你的安装版本过旧，文档将不匹配！
在这种情况下，请参考随你的LaTeX发行版附带的文档文件。

BibLaTeX – Sophisticated Bibliographies in LaTeX

BibLaTeX is a complete reimplementation of the bibliographic facilities provided by LaTeX. Formatting of the bibliography is entirely controlled by LaTeX macros, and a working knowledge of LaTeX should be sufficient to design new bibliography and citation styles. BibLaTeX uses its own data backend program called "biber" to read and process the bibliographic data. With biber, BibLaTeX has many features rivalling or surpassing other bibliography systems. To mention a few:

- Full Unicode support
- Highly customisable sorting using the Unicode Collation Algorithm + CLDR tailoring
- Highly customisable bibliography labels
- Complex macro-based on-the-fly data modification without changing your data sources
- A tool mode for transforming bibliographic data sources
- Multiple bibliographies and lists of bibliographic information in the same document with different sorting
- Highly customisable data source inheritance rules
- Polyglossia and babel suppport for automatic language switching for bibliographic entries and citations
- Automatic bibliography data recoding (UTF-8 -> latin1, LaTeX macros -> UTF-8 etc)
- Remote data sources
- Highly sophisticated automatic name and name list disambiguation system
- Highly customisable data model so users can define their own bibliographic data types
- Validation of bibliographic data against a data model
- Subdivided and/or filtered bibligraphies, bibliographies per chapter, section etc.

Apart from the features unique to BibLaTeX, the package also incorporates core features of the following packages: babelbib, bibtopic, bibunits, chapterbib, cite, inlinebib, mcite and mciteplus, mlbib, multibib, splitbib.

Sources	/macros/latex/contrib/biblatex
Documentation	Readme Package documentation (English) Release notes for current version
Version	3.4
License	The LaTeX Project Public License 1.3
Copyright	2012–2016 Philipp Lehman, Joseph Wright, Audrey Boruvka, Philip Kime 2006–2012 Philipp Lehman
Maintainer	Philipp Lehman (inactive) Philip Kime
Contained in	TeX Live as biblatex MiKTeX as biblatex
Topics	BibLaTeX bibliography support bibliography processor

Download the contents of this package in one zip archive (17.2M).

Important: CTAN holds the most recent versions. If your installation is outdated the documentation won't match! In that case, refer to the documentation documents shipped with your LaTeX distribution.

第15.2节：TeX Live -- texdoc

如果您使用 TeX Live 发行版，可以使用命令行程序texdoc。例如，

```
texdoc biblatex
```

将打开包biblatex的文档。

或者如果您不熟悉命令行，也可以在 http://www.texdoc.net/ 在线找到相同内容。

Section 15.2: TeX Live -- texdoc

If you use the TeX Live distribution you can use the command-line program `texdoc`. For instance,

```
texdoc biblatex
```

will open the documentation of package `biblatex`.

Or if you are not command-line-savvy, the same can be found online at http://www.texdoc.net/

第16章：使用 beamer 创建海报

使用 beamerposter 包创建海报与创建单个幻灯片非常相似。将内容放入列中。在每列中，使用区块分隔内容。

第16.1节：方向和尺寸

添加 beamerposter 包时，提供所需参数。

```
\usepackage[orientation=landscape,size=a1]{beamerposter}
```

您也可以自定义海报的尺寸。

```
usepackage[orientation=portrait,size=custom,height=110,width=80,scale=1.4]{beamerposter}
```

这里的高度和宽度尺寸单位为厘米。 scale 用于字体大小。

第16.2节：beamer海报的基本大纲

横向布局

```
documentclass[final,t]{beamer}
\mode<presentation>
  {
  \usetheme{Berlin}
  }

usepackage[orientation=landscape,size=a1,scale=1,debug]{beamerposter}
usepackage{lipsum} % 用于示例文本


title[]{huge 绝佳标题}
author[]{\large textbf{作者姓名1} 和 作者姓名2 和 作者姓名3}
\institute[]{\Large XYZ系，ABC研究所}
date{}

\begin{document}

\begin{frame}
\maketitle
\begin{columns}[t]
  \begin{column}{.32\linewidth}

  \begin{block}{Some heading}
  \lipsum[1]
  \end{block}

  \begin{block}{Some heading}
  \lipsum[1]
  \end{block}

  \begin{block}{Some heading}
  \lipsum[1]
  \end{block}


  \end{column}

 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```


Chapter 16: Creating posters using beamer

Creating a poster using beamerposter package is very similar to creating a single frame. Put the content in columns. Within each column, separate the content using blocks.

Section 16.1: Orientation and size

While adding the beamerposter package, provide the required parameters.

```
\usepackage[orientation=landscape,size=a1]{beamerposter}
```

You can also customize the size of the poster.

```
\usepackage[orientation=portrait,size=custom,height=110,width=80,scale=1.4]{beamerposter}
```

The height and width dimensions here, are in cms. The `scale` is used for the font size.

Section 16.2: Basic outline of a beamer poster

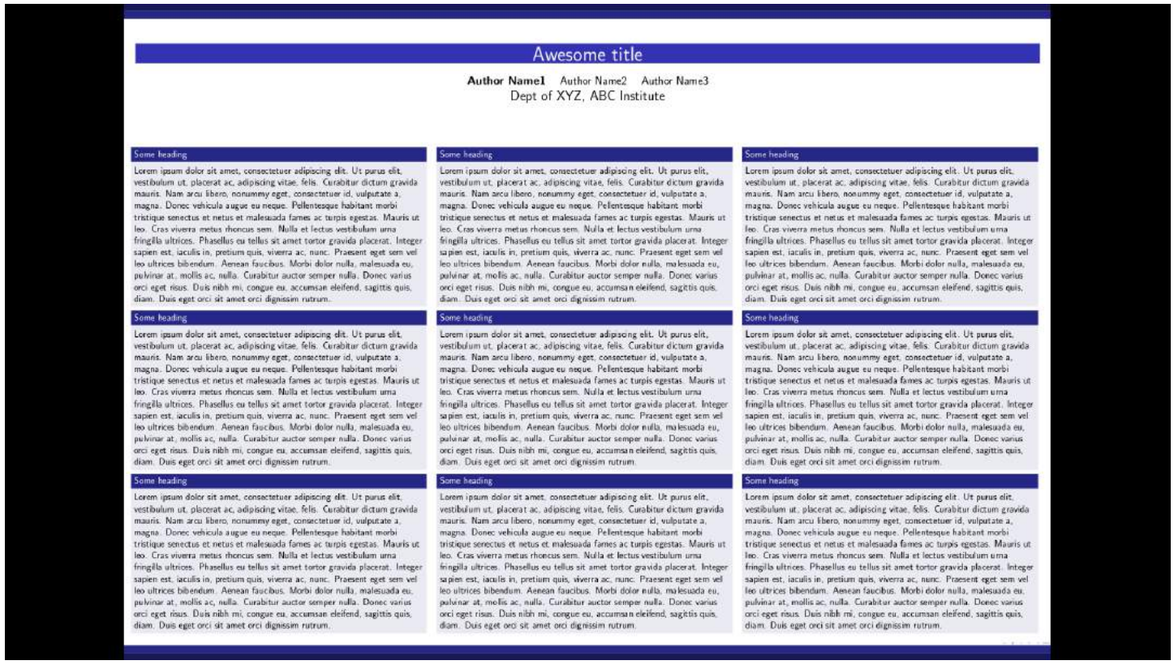
In landscape orientation

```
\documentclass[final,t]{beamer}
\mode<presentation>
  {
  \usetheme{Berlin}
  }

\usepackage[orientation=landscape,size=a1,scale=1,debug]{beamerposter}
\usepackage{lipsum} % for dummy text


\title[]{\huge Awesome title}
\author[]{\large \textbf{Author Name1} \and Author Name2 \and Author Name3}
\institute[]{\Large Dept of XYZ, ABC Institute}
\date{}

\begin{document}

\begin{frame}
\maketitle
\begin{columns}[t]
  \begin{column}{.32\linewidth}

  \begin{block}{Some heading}
  \lipsum[1]
  \end{block}

  \begin{block}{Some heading}
  \lipsum[1]
  \end{block}

  \begin{block}{Some heading}
  \lipsum[1]
  \end{block}


  \end{column}

 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
\begin{column}{.32\linewidth}

\begin{block}{Some heading}
\lipsum[1]
\end{block}

\begin{block}{Some heading}
\lipsum[1]
\end{block}

\begin{block}{Some heading}
\lipsum[1]
\end{block}


\end{column}


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

\begin{column}{.32\linewidth}

\begin{block}{Some heading}
\lipsum[1]
\end{block}

\begin{block}{Some heading}
\lipsum[1]
\end{block}

\begin{block}{Some heading}
\lipsum[1]
\end{block}


\end{column}
\end{columns}

\end{frame}

\end{document}
```

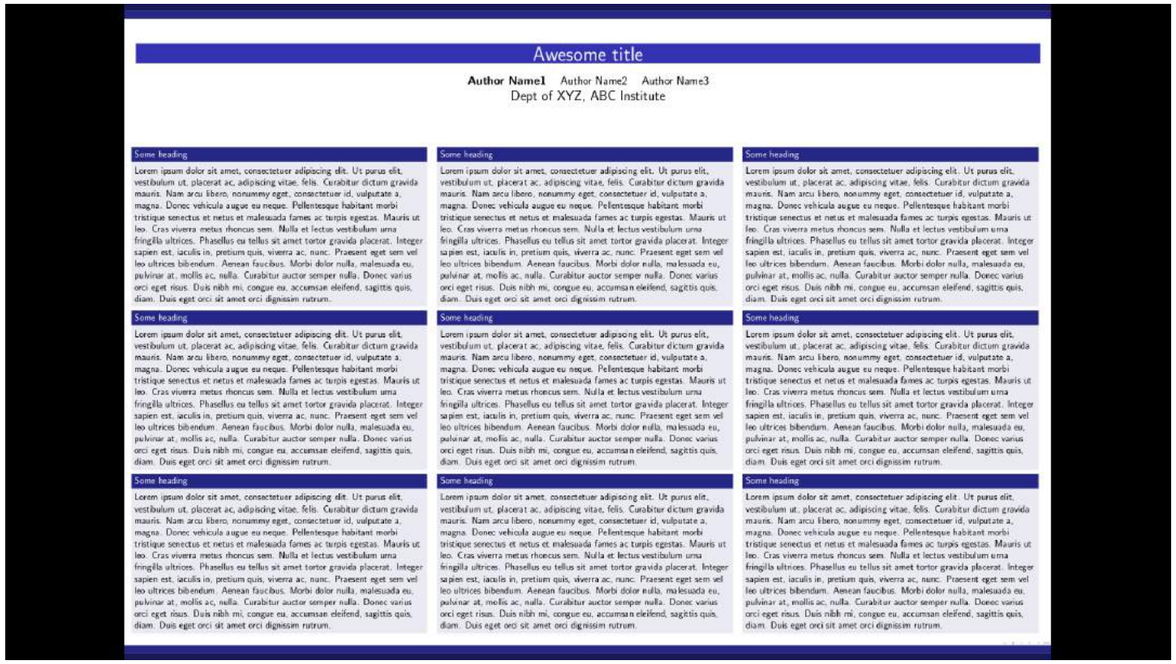


```
\begin{column}{.32\linewidth}

\begin{block}{Some heading}
\lipsum[1]
\end{block}

\begin{block}{Some heading}
\lipsum[1]
\end{block}

\begin{block}{Some heading}
\lipsum[1]
\end{block}


\end{column}


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

\begin{column}{.32\linewidth}

\begin{block}{Some heading}
\lipsum[1]
\end{block}

\begin{block}{Some heading}
\lipsum[1]
\end{block}

\begin{block}{Some heading}
\lipsum[1]
\end{block}


\end{column}
\end{columns}

\end{frame}

\end{document}
```



纵向

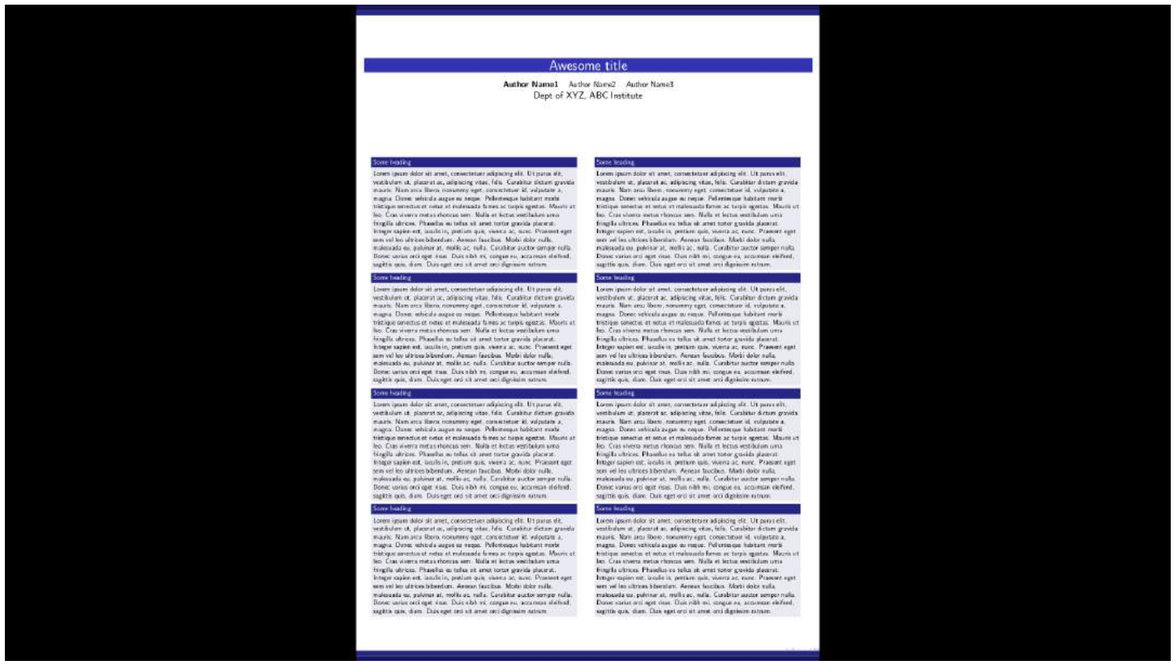
```
documentclass[final,t]{beamer}
\mode<presentation>
  {
  \usetheme{Berlin}
  }

\usepackage[orientation=portrait,size=a1,scale=1,debug]{beamerposter}
\usepackage{lipsum} % 用于虚拟文本


title[]{huge 绝佳标题}
author[]{\large textbf{作者姓名1} 和 作者姓名2 和 作者姓名3}
\institute[]{\Large XYZ系，ABC研究所}
date{}

\begin{document}

\begin{frame}
\maketitle
\begin{columns}[t]
  \begin{column}{.45\linewidth}

  \begin{block}{Some heading}
  \lipsum[1]
  \end{block}

  \begin{block}{Some heading}
  \lipsum[1]
  \end{block}

  \begin{block}{Some heading}
  \lipsum[1]
  \end{block}
```

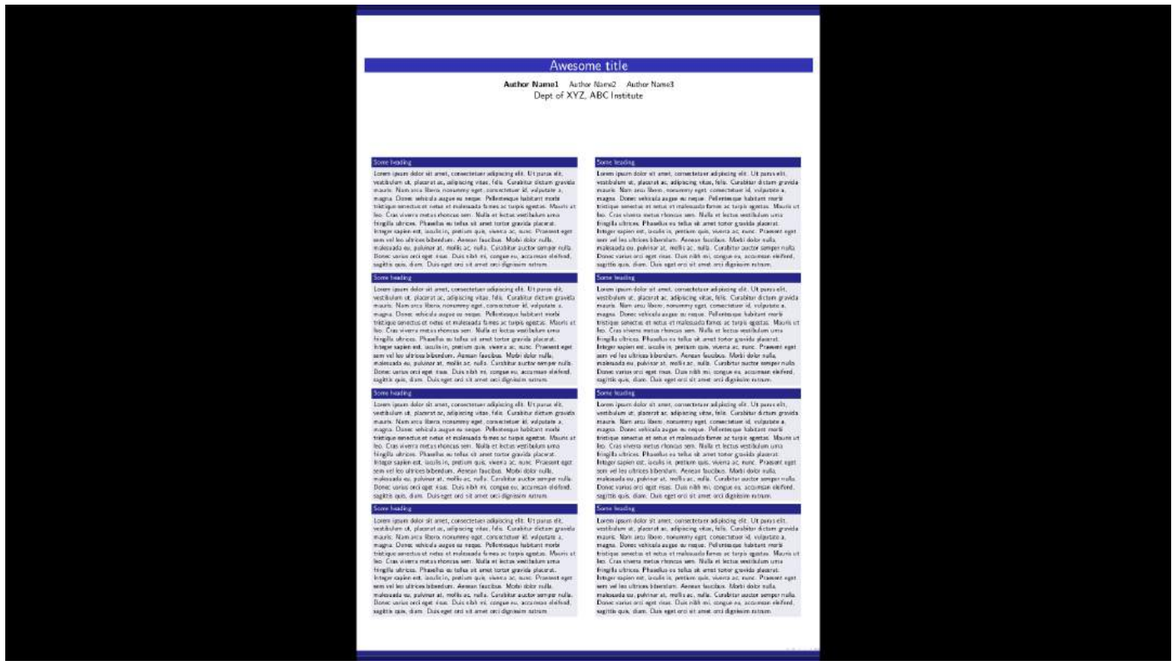
In portrait orientation

```
\documentclass[final,t]{beamer}
\mode<presentation>
  {
  \usetheme{Berlin}
  }

\usepackage[orientation=portrait,size=a1,scale=1,debug]{beamerposter}
\usepackage{lipsum} % for dummy text


\title[]{\huge Awesome title}
\author[]{\large \textbf{Author Name1} \and Author Name2 \and Author Name3}
\institute[]{\Large Dept of XYZ, ABC Institute}
\date{}

\begin{document}

\begin{frame}
\maketitle
\begin{columns}[t]
  \begin{column}{.45\linewidth}

  \begin{block}{Some heading}
  \lipsum[1]
  \end{block}

  \begin{block}{Some heading}
  \lipsum[1]
  \end{block}

  \begin{block}{Some heading}
  \lipsum[1]
  \end{block}
```

```
  \begin{block}{Some heading}
  \lipsum[1]
  \end{block}

  \end{column}

 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  \begin{column}{.45\linewidth}

  \begin{block}{某个标题}
  \lipsum[1]
  \end{block}

  \begin{block}{Some heading}
  \lipsum[1]
  \end{block}

  \begin{block}{Some heading}
  \lipsum[1]
  \end{block}

  \begin{block}{Some heading}
  \lipsum[1]
  \end{block}

  \end{column}
  \end{columns}

\end{frame}

\end{document}
```




```
  \begin{block}{Some heading}
  \lipsum[1]
  \end{block}

  \end{column}

 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  \begin{column}{.45\linewidth}

  \begin{block}{Some heading}
  \lipsum[1]
  \end{block}

  \begin{block}{Some heading}
  \lipsum[1]
  \end{block}

  \begin{block}{Some heading}
  \lipsum[1]
  \end{block}

  \begin{block}{Some heading}
  \lipsum[1]
  \end{block}

  \end{column}
  \end{columns}

\end{frame}

\end{document}
```



第16.3节：beamer海报的完整示例

```
documentclass[final,t]{beamer}
\mode<presentation>
  {
  \usetheme{Berlin}
  }

usepackage[orientation=landscape,size=a1,scale=1,debug]{beamerposter}
usepackage{lipsum} % 用于示例文本
\usepackage{graphicx} % 用于虚拟图片
\usepackage{tikz} % 用于tikzpicture
\usepackage{pgfplots} % 用于绘图
\usetikzlibrary{arrows,shapes,positioning}


\title[]{\huge 棒极了的标题}
\author[]{\large \textbf{作者姓名1} \and 作者姓名2 \and 作者姓名3}
\institute[]{\Large XYZ系，ABC研究所}
date{}

\begin{document}

\begin{frame}
\maketitle
\begin{columns}[t]

\begin{column}{.32\linewidth}


  \begin{block}{第一段落}
  \lipsum[1]
  \end{block}

  \begin{block}{第一张图}
  一些描述图的文字
  \begin{center}
   \begin{figure}
    \includegraphics[scale=0.7]{example-image-a}
    \caption{第一张图说明}
   \end{figure}

  \end{center}


  \end{block}


  \begin{block}{第一列表}
  \begin{itemize}
   \item Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi.
   项目 Morbi auctor lorem non justo.
   项目 名称 lacus 自由，价格在，生命的边缘，超越，和，地球。
   项目 完成的选择，托尔托尔 sed accumsan bibendum，erat ligula 选择大，生命的装饰讨厌
metus a mi。
   项目 Morbi ac orci et nisl hendrerit mollis。
   项目 保持质量。
   项目 Cras nec ante。
   项目 Pellentesque a nulla。
        项目 Cum sociis natoque penatibus et magnis dis parturient montes，nascetur ridiculus mus。
   项目 Aliquam tincidunt urna。
```


Section 16.3: Full example of beamer poster

```
\documentclass[final,t]{beamer}
\mode<presentation>
  {
  \usetheme{Berlin}
  }

\usepackage[orientation=landscape,size=a1,scale=1,debug]{beamerposter}
\usepackage{lipsum} % for dummy text
\usepackage{graphicx} % for dummy image
\usepackage{tikz} % for tikzpicture
\usepackage{pgfplots} % for plot
\usetikzlibrary{arrows,shapes,positioning}


\title[]{\huge Awesome title}
\author[]{\large \textbf{Author Name1} \and Author Name2 \and Author Name3}
\institute[]{\Large Dept of XYZ, ABC Institute}
\date{}

\begin{document}

\begin{frame}
\maketitle
\begin{columns}[t]

\begin{column}{.32\linewidth}


  \begin{block}{First paragraph}
  \lipsum[1]
  \end{block}

  \begin{block}{First figure}
  Some text describing figure
  \begin{center}
   \begin{figure}
    \includegraphics[scale=0.7]{example-image-a}
    \caption{First figure caption}
   \end{figure}

  \end{center}


  \end{block}


  \begin{block}{First list}
  \begin{itemize}
   \item Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi.
   \item Morbi auctor lorem non justo.
   \item Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus.
   \item Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio
metus a mi.
   \item Morbi ac orci et nisl hendrerit mollis.
   \item Suspendisse ut massa.
   \item Cras nec ante.
   \item Pellentesque a nulla.
   \item Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.
   \item Aliquam tincidunt urna.
```

```
  项目 Nulla ullamcorper vestibulum turpis。
  项目 Pellentesque cursus luctus mauris.
 \end{itemize}

 \end{block}


  \end{column}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

 \begin{column}{.32\linewidth}

 \begin{block}{Second list}
 \begin{enumerate}
  项目 Nulla malesuada porttitor diam.
  项目 Donec felis erat, congue non, volutpat at, tincidunt tristique, libero.
  项目 Vivamus viverra fermentum felis.
  项目 Donec nonummy pellentesque ante.
  项目 Phasellus adipiscing semper elit.
  项目 Proin fermentum massa ac quam.
  项目 Sed diam turpis, molestie vitae, placerat a, molestie nec, leo.
  项目 Maecenas lacinia.
  项目 Nam ipsum ligula，eleifend at，accumsan nec，suscipit a，ipsum。
 \end{enumerate}

 \end{block}

 \begin{block}{第一个分块}
   \begin{columns}
     \begin{column}{0.5\linewidth}

     \begin{center}
     \begin{figure}
      includegraphics[width=0.55\linewidth]{example-image-b}
      caption{第二张图说明}
     \end{figure}
    \end{center}
    \begin{center}
     \begin{figure}
      includegraphics[width=0.55\linewidth]{example-image-c}
      caption{第三张图说明}
     \end{figure}
    \end{center}

    \end{column}

    \begin{column}{0.5\linewidth}

Morbi blandit ligula feugiat magna。
    \begin{itemize}
    \item 现在的结果令人满意。
    \item 但无论如何，生活中没有什么是无用的。
    \item 纯净的触感非常重要。
    \item 整数不是无用的。
    \item 现在的样式非常纯净。
    \item 章节在地面上很重要。
    \item 课程的重点是轻松。
    \item 现在和我一起。
    \item 变化对我来说很重要。
    项目 盆栽植物在质量上表现出色。
    \end{itemize}
```



```
  \item Nulla ullamcorper vestibulum turpis.
  \item Pellentesque cursus luctus mauris.
 \end{itemize}

 \end{block}


  \end{column}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

 \begin{column}{.32\linewidth}

 \begin{block}{Second list}
 \begin{enumerate}
  \item Nulla malesuada porttitor diam.
  \item Donec felis erat, congue non, volutpat at, tincidunt tristique, libero.
  \item Vivamus viverra fermentum felis.
  \item Donec nonummy pellentesque ante.
  \item Phasellus adipiscing semper elit.
  \item Proin fermentum massa ac quam.
  \item Sed diam turpis, molestie vitae, placerat a, molestie nec, leo.
  \item Maecenas lacinia.
  \item Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum.
 \end{enumerate}

 \end{block}

 \begin{block}{First split block}
   \begin{columns}
     \begin{column}{0.5\linewidth}

     \begin{center}
     \begin{figure}
      \includegraphics[width=0.55\linewidth]{example-image-b}
      \caption{Second figure caption}
     \end{figure}
    \end{center}
    \begin{center}
     \begin{figure}
      \includegraphics[width=0.55\linewidth]{example-image-c}
      \caption{Third figure caption}
     \end{figure}
    \end{center}

    \end{column}

    \begin{column}{0.5\linewidth}

Morbi blandit ligula feugiat magna.
    \begin{itemize}
    \item Nunc eleifend consequat lorem.
    \item Sed lacinia nulla vitae enim.
    \item Pellentesque tincidunt purus vel magna.
    \item Integer non enim.
    \item Praesent euismod nunc eu purus.
    \item Donec bibendum quam in tellus.
    \item Nullam cursus pulvinar lectus.
    \item Donec et mi.
    \item Nam vulputate metus eu enim.
    \item Vestibulum pellentesque felis eu massa.
    \end{itemize}
```

```
  \end{column}
  \end{columns}

  \end{block}


  \begin{block}{第一个tikz图}
        Morbi blandit ligula feugiat magna。现在eleifend consequat lorem。Sed lacinia nulla vitae enim。
佩伦特斯克
tincidunt purus vel magna。

  \begin{center}
  \begin{figure}

  \begin{tikzpicture}

      % 定义
      \pgfmathsetmacro{\b}{75}
      \pgfmathsetmacro{\a}{15}
      \pgfmathsetmacro{\R}{2}
      \pgfmathsetmacro{\r}{1}
      \pgfmathsetmacro{\P}{\R*tan(\b)}
      \pgfmathsetmacro{\Q}{\R/cos(\b)}
      \pgfmathsetmacro{\p}{\r/tan(\a)}
      \pgfmathsetmacro{\q}{\r/sin(\a)}

      % 皮带轮

      % 大皮带轮
      \draw (0,0) circle (\R) ;
      \fill[left color=gray!80, right color=gray!60, middle
        color=white] (0,0) circle (\R) ;
      \draw[thick, white] (0,0) circle (.8*\R);
      \shade[ball color=white] (0,0) circle (.3) node[left,xshift=-5] {$P$};

      % 小皮带轮
      \draw (\Q+\q-.3, 0) circle (\r);
      \fill[left color=gray!80, right color=gray!60, middle
        color=white] (\Q+\q-.3, 0) circle (\r) ;
      \draw[thick, white] (\Q+\q-.3,0) circle (.8*\r);
      \shade[ball color=white] (\Q+\q-.3,0) circle (.15)
      node[right, xshift=2] {$Q$};

      % 皮带和点标签
      \begin{scope}[ultra thick]
        \draw (\b:\R) arc (\b:360-\b:\R) ;
        \draw (\b:\R) -- ( \P, 0 );
        \draw (-\b:\R) -- ( \P, 0 );
        \draw (\Q-.3,0) -- + (\a:\p)  arc (105:-105:\r) ;
        \draw (\Q-.3,0) -- + (-\a:\p);
        %\draw (\b:\R) arc (\b:360-\b:\r) ;
      \end{scope}

      \draw (0,0) -- (\b:\R) node[midway, above,sloped] {$R$} node[above] {$A$};
      \draw (-\b:\R)--(0,0) ;
      \draw (\Q+\q-.3,0) -- +(105:\r) node[midway,above, sloped] {$r$}
        node[above] {$E$};
      \draw (\Q+\q-.3,0) -- +(-105:\r) node[below] {$D$};
      ode[below] at (-\b:\R) {$B$};
      ode[below] at (\Q-.3,0) {$C$};

      % 中心线
```



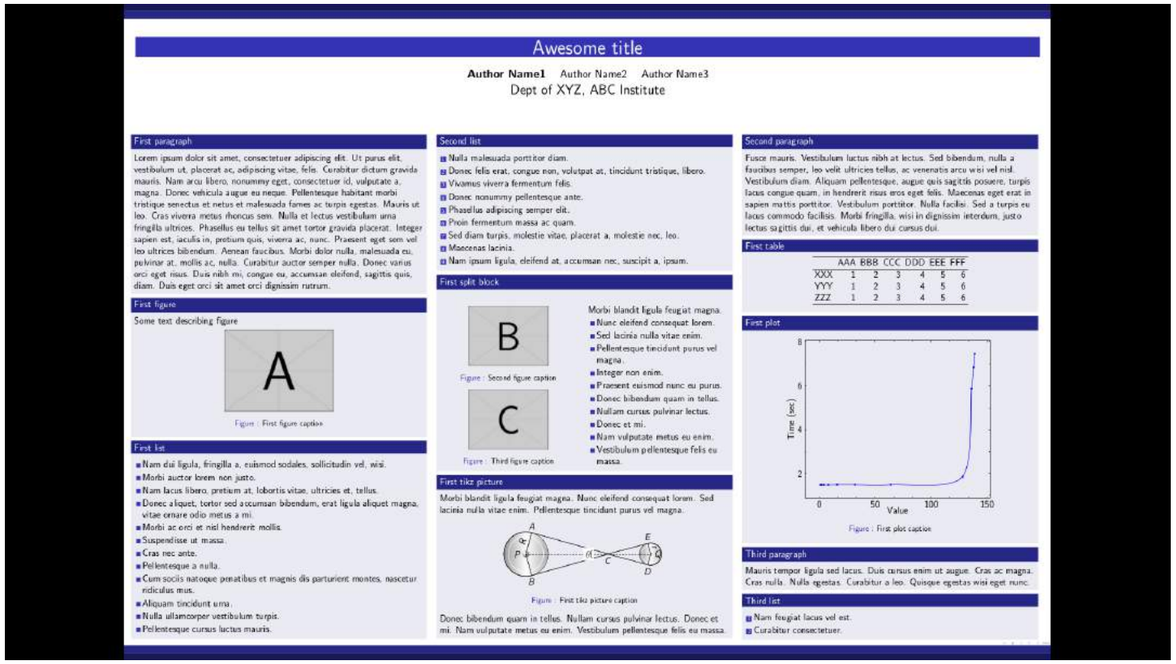
```
  \end{column}
  \end{columns}

  \end{block}


  \begin{block}{First tikz picture}
  Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim.
Pellentesque
tincidunt purus vel magna.

  \begin{center}
  \begin{figure}

  \begin{tikzpicture}

      % Definitions
      \pgfmathsetmacro{\b}{75}
      \pgfmathsetmacro{\a}{15}
      \pgfmathsetmacro{\R}{2}
      \pgfmathsetmacro{\r}{1}
      \pgfmathsetmacro{\P}{\R*tan(\b)}
      \pgfmathsetmacro{\Q}{\R/cos(\b)}
      \pgfmathsetmacro{\p}{\r/tan(\a)}
      \pgfmathsetmacro{\q}{\r/sin(\a)}

      % Pulleys

      % big pulley
      \draw (0,0) circle (\R) ;
      \fill[left color=gray!80, right color=gray!60, middle
        color=white] (0,0) circle (\R) ;
      \draw[thick, white] (0,0) circle (.8*\R);
      \shade[ball color=white] (0,0) circle (.3) node[left,xshift=-5] {$P$};

      % small pulley
      \draw (\Q+\q-.3, 0) circle (\r);
      \fill[left color=gray!80, right color=gray!60, middle
        color=white] (\Q+\q-.3, 0) circle (\r) ;
      \draw[thick, white] (\Q+\q-.3,0) circle (.8*\r);
      \shade[ball color=white] (\Q+\q-.3,0) circle (.15)
      node[right, xshift=2] {$Q$};

      % belt and point labels
      \begin{scope}[ultra thick]
        \draw (\b:\R) arc (\b:360-\b:\R) ;
        \draw (\b:\R) -- ( \P, 0 );
        \draw (-\b:\R) -- ( \P, 0 );
        \draw (\Q-.3,0) -- + (\a:\p)  arc (105:-105:\r) ;
        \draw (\Q-.3,0) -- + (-\a:\p);
        %\draw (\b:\R) arc (\b:360-\b:\r) ;
      \end{scope}

      \draw (0,0) -- (\b:\R) node[midway, above,sloped] {$R$} node[above] {$A$};
      \draw (-\b:\R)--(0,0) ;
      \draw (\Q+\q-.3,0) -- +(105:\r) node[midway,above, sloped] {$r$}
        node[above] {$E$};
      \draw (\Q+\q-.3,0) -- +(-105:\r) node[below] {$D$};
      \node[below] at (-\b:\R) {$B$};
      \node[below] at (\Q-.3,0) {$C$};

      % center line
```

```
      \draw[dash pattern=on5pt off3pt] (0,0) -- (\Q+\q-.3,0);

      % 角度标签
      ode[fill=white] at (0.73*\Q, 0) {$heta$} ;
      \draw (\Q-1.8,0) arc (180:195:1.5);
      \draw (\Q-1.8,0) arc (180:165:1.5);

  \end{tikzpicture}
  caption{第一个tikz图说明}
  \end{figure}
  \end{center}

Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu
enim. Vestibulum pellentesque felis eu massa.
  \end{block}

  \end{column}


  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  \begin{column}{.32\linewidth}

  \begin{block}{第二段落}
Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit
ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue
quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget
erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus
commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et
vehicula libero dui cursus dui.
  \end{block}

  \begin{block}{第一个表格}
  \begin{center}
  \begin{tabular}{lrrrrrr}
  \hline
  & AAA & BBB & CCC & DDD & EEE & FFF\\ \hline
  XXX & 1 & 2 & 3 & 4 & 5 & 6 \\
YYY & 1 & 2 & 3 & 4 & 5 & 6 \\
  ZZZ & 1 & 2 & 3 & 4 & 5 & 6 \\

  \hline
  \end{tabular}

  \end{center}
  \end{block}

  \begin{block}{第一张图}
  \begin{center}
  \begin{figure}

    \begin{tikzpicture}
    \begin{axis}[
width=0.7\linewidth,
    max space between ticks=50,
    minor x tick num=2,
     minor y tick num=1,
     tick style={semithick,color=black},
    xlabel=数值,
ylabel=时间（秒）,
     xtick={0, 50, 100, 150},
    ytick={0, 2, 4, 6, 8}]
```

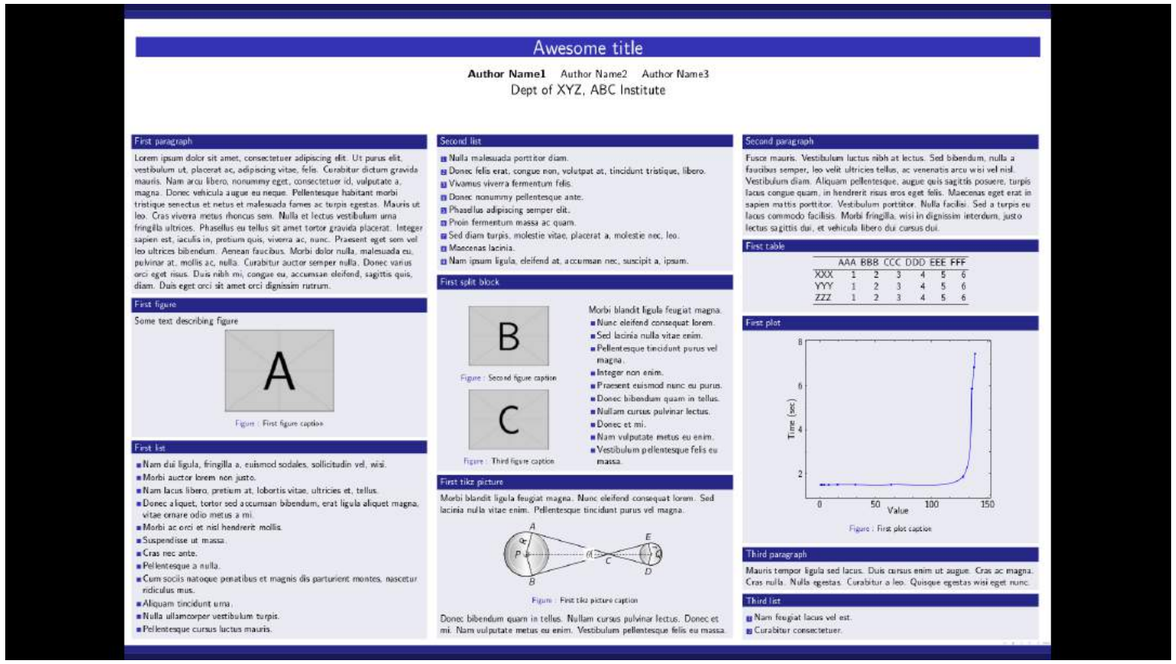


```
      \draw[dash pattern=on5pt off3pt] (0,0) -- (\Q+\q-.3,0);

      % angle label
      \node[fill=white] at (0.73*\Q, 0) {$\theta$} ;
      \draw (\Q-1.8,0) arc (180:195:1.5);
      \draw (\Q-1.8,0) arc (180:165:1.5);

  \end{tikzpicture}
  \caption{First tikz picture caption}
  \end{figure}
  \end{center}

 Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu
enim. Vestibulum pellentesque felis eu massa.
  \end{block}

  \end{column}


  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  \begin{column}{.32\linewidth}

  \begin{block}{Second paragraph}
  Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit
ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue
quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget
erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus
commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et
vehicula libero dui cursus dui.
  \end{block}

  \begin{block}{First table}
  \begin{center}
  \begin{tabular}{lrrrrrr}
  \hline
  & AAA & BBB & CCC & DDD & EEE & FFF\\ \hline
  XXX & 1 & 2 & 3 & 4 & 5 & 6 \\
  YYY & 1 & 2 & 3 & 4 & 5 & 6 \\
  ZZZ & 1 & 2 & 3 & 4 & 5 & 6 \\

  \hline
  \end{tabular}

  \end{center}
  \end{block}

  \begin{block}{First plot}
  \begin{center}
  \begin{figure}

    \begin{tikzpicture}
    \begin{axis}[
    width=0.7\linewidth,
    max space between ticks=50,
    minor x tick num=2,
    minor y tick num=1,
    tick style={semithick,color=black},
    xlabel=Value,
    ylabel=Time (sec),
    xtick={0, 50, 100, 150},
    ytick={0, 2, 4, 6, 8}]
```

```
    \addplot[smooth, blue, mark=*] coordinates { (1,1.48) (2,1.48) (4,1.48) (8,1.48) (16,1.49)
(32,1.49) (64,1.49) (128,1.85) (136,5.87) (138,6.84) (139,7.46)};
    \end{axis}
    \end{tikzpicture}


  \caption{第一张图说明}
  \end{figure}
  \end{center}
  \end{block}

  \begin{block}{第三段}
Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla
egestas. Curabitur a leo. Quisque egestas wisi eget nunc.
  \end{block}

  \begin{block}{第三列表}
   \begin{enumerate}
    \item 名为“Nam feugiat lacus vel est.”
    \item Curabitur consectetuer.
   \end{enumerate}

  \end{block}


  \end{column}
  \end{columns}

\end{frame}

\end{document}
```




```
    \addplot[smooth, blue, mark=*] coordinates { (1,1.48) (2,1.48) (4,1.48) (8,1.48) (16,1.49)
(32,1.49) (64,1.49) (128,1.85) (136,5.87) (138,6.84) (139,7.46)};
    \end{axis}
    \end{tikzpicture}


  \caption{First plot caption}
  \end{figure}
  \end{center}
  \end{block}

  \begin{block}{Third paragraph}
   Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla
egestas. Curabitur a leo. Quisque egestas wisi eget nunc.
  \end{block}

  \begin{block}{Third list}
   \begin{enumerate}
    \item Nam feugiat lacus vel est.
    \item Curabitur consectetuer.
   \end{enumerate}

  \end{block}


  \end{column}
  \end{columns}

\end{frame}

\end{document}
```



第17章：乐谱排版

第17.1节：LilyPond

LilyPond乐谱排版器可以通过lilypond-book命令与LaTeX一起使用。首先，让我们创建一个LaTeX文档（文件扩展名为.lytex）来嵌入我们的音乐：

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

\begin{document}

\begin{center}
    {\fontsize{24pt}{24pt}\textbf{两只乌鸦}}\\
\end{center}

\begin{flushright}
    extsc{你的名字}
\end{flushright}

% 我们不需要为此做任何要求，因为 lilypond-book 会处理它。
\lilypondfile{TwaCorbies.ly}
\end{document}
```

然后我们创建我们的 LilyPond 文件（.ly），包括 lilypond-book-preamble.ly 文件（LilyPond 会知道如何找到它）：

```
\version "2.16.2"

\include "lilypond-book-preamble.ly"

voice = <<
    \relative c' {
        empo "con affetto"
        \clef 低音谱号
        \key e 小调
        ime 3/4

a a b | c a a | g a2 |
       a4 a b | c2 ~ c8 a8 | a8 g a2 |
        \bar "|."
    }
    \addlyrics{
当我独自一人漫步时
       我听见两只乌鸦在哀鸣。
    }
>>

\score {
    <<
        ew Staff = "voice" {
            \voice
        }
    >>
    \layout { }
    \midi {
        \context {
            \Score
tempoWholesPerMinute = #(ly:make-moment 90 4)
       }
```


Chapter 17: Engraving Sheet Music

Section 17.1: LilyPond

The LilyPond notation engraver can be used with LaTeX via the `lilypond-book` command. First lets create a LaTeX document (with the file extension `.lytex`) to embed our music in:

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

\begin{document}

\begin{center}
    {\fontsize{24pt}{24pt}\textbf{Twa Corbies}}\\
\end{center}

\begin{flushright}
    \textsc{Your Name}
\end{flushright}

% We don't need to require anything for this because lilypond-book will process it.
\lilypondfile{TwaCorbies.ly}
\end{document}
```

Then we create our LilyPond file (`.ly`), including the `lilypond-book-preamble.ly` file (which LilyPond will know how to find):

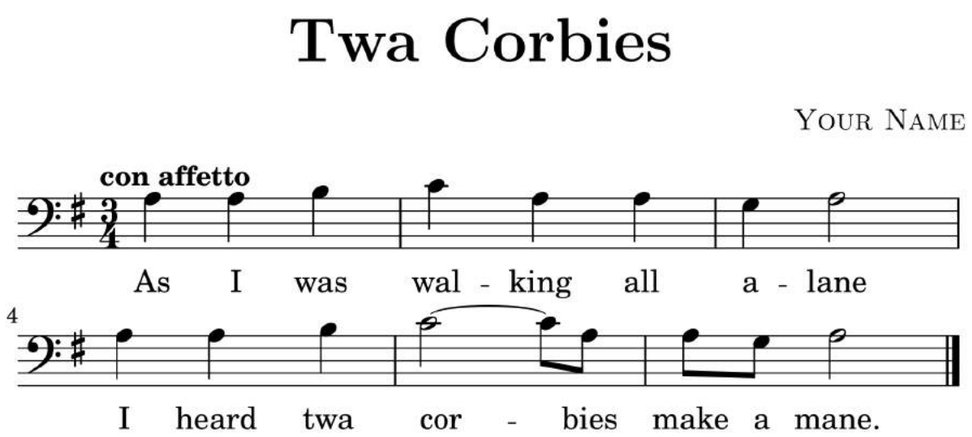
```
\version "2.16.2"

\include "lilypond-book-preamble.ly"

voice = <<
    \relative c' {
        \tempo "con affetto"
        \clef bass
        \key e \minor
        \time 3/4

        a a b | c a a | g a2 |
        a4 a b | c2 ~ c8 a8 | a8 g a2 |
        \bar "|."
    }
    \addlyrics{
        As I was wal -- king all a -- lane
        I heard twa cor -- bies make a mane.
    }
>>

\score {
    <<
        \new Staff = "voice" {
            \voice
        }
    >>
    \layout { }
    \midi {
        \context {
            \Score
            tempoWholesPerMinute = #(ly:make-moment 90 4)
        }
```

```
}
}
```

要构建，我们接着运行lilypond-book命令：

```
lilypond-book --include=mymusicsourcedirectory/ --pdf TwaCorbies.lytex
```

这将输出包含您用LilyPond排版的乐谱的PDF：

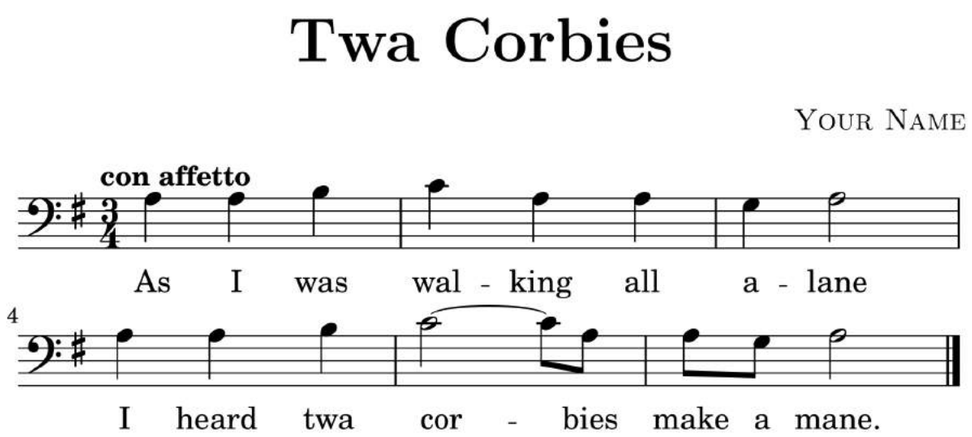



```
    }
}
```

to build, we then run the `lilypond-book` command:

```
lilypond-book --include=mymusicsourcedirectory/ --pdf TwaCorbies.lytex
```

which will output a PDF containing your LilyPond engraved music:



鸣谢

非常感谢所有来自Stack Overflow文档的人员帮助提供此内容，
更多更改可发送至web@petercv.com，以发布或更新新内容

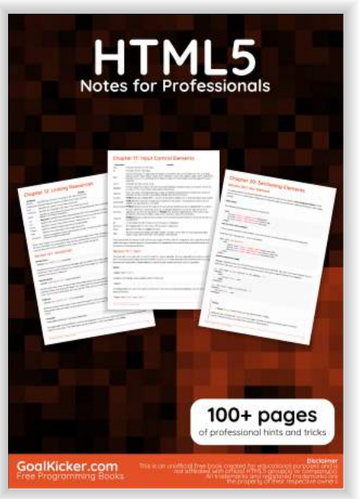
celtschk	第4章和第13章
csekri	第9章
哈姆扎维	第8章
哈利	第1章
hbaderts	第1章
希瑟	第6章和第10章
贾尼	第1章
约翰内斯_B	第2章
L.V.拉奥	第5章和第11章
路易斯	第1章
Nijin22	第1、2、5和7章
尼基塔·贾因	第5章
拉斐尔	第7、11和15章
萨姆·怀特德	第17章
samcarter	第15章
肖恩·奥尔雷德	第1章和第14章
Spacedman	第1章
斯特凡·库拉	第4章
strpeter	第13章
苏坎雅·B	第3章、第12章和第16章
萨姆纳·埃文斯	第1章
维尔纳	第七章

Credits

Thank you greatly to all the people from Stack Overflow Documentation who helped provide this content,
more changes can be sent to web@petercv.com for new content to be published or updated

celtschk	Chapters 4 and 13
csekri	Chapter 9
Hamzawey	Chapter 8
Harry	Chapter 1
hbaderts	Chapter 1
heather	Chapters 6 and 10
jani	Chapter 1
Johannes_B	Chapter 2
L.V.Rao	Chapters 5 and 11
Louis	Chapter 1
Nijin22	Chapters 1, 2, 5 and 7
Nikita Jain	Chapter 5
Raphael	Chapters 7, 11 and 15
Sam Whited	Chapter 17
samcarter	Chapter 15
Sean Allred	Chapters 1 and 14
Spacedman	Chapter 1
Stephan Kulla	Chapter 4
strpeter	Chapter 13
Sukanya B	Chapters 3, 12 and 16
Sumner Evans	Chapter 1
Werner	Chapter 7

你可能也喜欢



You may also like