



L^AT_EX 排版简要介绍

从入门到放弃

纸上得来终觉浅

数理学院数学系



2024 年 1 月 5 日, 中国上海



目录

介绍

TeX 排版系统历史

L^ATeX 利弊

L^ATeX 安装

Hello world 案例

填写创作

段落及排版样式

目录

图片

列表

数学公式

数学模式

公式

字体

常用环境

表格

代码

交叉引用



先声夺人

/ˈleɪtɛk/



LaTeX 是什么? ——你为什么学

Word 替代品?

「我受够了, 我以后什么都要用 LaTeX 写」

写论文神器?

「我就是为大 paper 而生的, 当然必须学 LaTeX 啦」

排版神器?

「别人都用 LaTeX 排版论文, 当然我也得学 LaTeX 啦」



L^AT_EX 是什么? ——你为什么学

- Word 替代品?
 - ▶ 「我受够了, 我以后什么都要用 L^AT_EX 写」
- 写论文神器?
 - ▶ 「我就是为大 paper 而生的, 当然必须学 L^AT_EX 啦」
- 打公式方便?
 - ▶ 「复杂公式输入哪家强, 当然首选 L^AT_EX 帮忙」



L^AT_EX 是什么? ——你为什么学

- Word 替代品?
 - ▶ 「我受够了, 我以后什么都要用 L^AT_EX 写」
- 写论文神器?
 - ▶ 「我就是为大 paper 而生的, 当然必须学 L^AT_EX 啦」
- 打公式方便?
 - ▶ 「复杂公式输入哪家强, 当然首选 L^AT_EX 帮忙」



L^AT_EX 是什么? ——你为什么学

- Word 替代品?
 - ▶ 「我受够了, 我以后什么都要用 L^AT_EX 写」
- 写论文神器?
 - ▶ 「我就是为大 paper 而生的, 当然必须学 L^AT_EX 啦」
- 打公式方便?
 - ▶ 「复杂公式输入哪家强, 当然首选 L^AT_EX 帮忙」

LaTeX 是什么? ——What you *think* is what you get!

```

1 \documentclass[12pt,a4paper]{article}
2 \usepackage{amsmath,graphicx}
3 \title{Normal distribution}
4 \author{Wikipedia, the free encyclopedia}
5 \date{\today}
6
7 \begin{document}
8 \maketitle
9 \section{Introduction}
10 In probability theory, the normal distribution is a very common
11 distribution. Normal distributions are important in statistics
12 and social sciences to represent real-valued random variables.
13 A random variable with a Gaussian distribution is said to be
14 normal deviate.
15
16 \begin{figure}[h]
17 \centering
18 \includegraphics[width=8cm]{normal-distribution-PDF.pdf}
19 \end{figure}
20
21 The probability density of the normal distribution is
22 \begin{equation}
23 f(x|\mu,\sigma)
24 = \frac{1}{\sqrt{2\pi\sigma^2}}
25 e^{-\frac{(x-\mu)^2}{2\sigma^2}}
26 \end{equation}
27 where
28 \begin{itemize}
29 \item  $\mu$  is the mean of the distribution
30 \item  $\sigma$  is the standard deviation
31 \end{itemize}
32 \end{document}
33
```

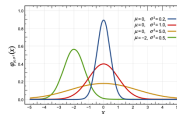
Normal distribution

Wikipedia, the free encyclopedia

February 13, 2023

1 Introduction

In probability theory, the normal distribution is a very common continuous probability distribution. Normal distributions are important in statistics and are often used in the natural and social sciences to represent real-valued random variables whose distributions are not known. A random variable with a Gaussian distribution is said to be normally distributed and is called a normal deviate.



The probability density of the normal distribution is

$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (1)$$

where

- μ is the mean of the distribution
- σ is the standard deviation



基本原则

- 排版 vs 文字处理
 - ▶ 《别把 L^AT_EX 当 Word 用》
 - ▶ 在固定版面内，摆置各种不同类型的资料，以最合适的方法呈现 W
- 遵循业界规范
- 追求良好的阅读体验 (readability)
- 内容与格式分离
- 内容永远比格式重要!



基本原则

- 排版 vs 文字处理
 - ▶ 《别把 L^AT_EX 当 Word 用》
 - ▶ 在固定版面内，摆置各种不同类型的资料，以最合适的方法呈现 W
- 遵循业界规范
 - 追求良好的阅读体验 (readability)
 - 内容与格式分离
 - 内容永远比格式重要!



基本原则

- 排版 vs 文字处理
 - ▶ 《别把 L^AT_EX 当 Word 用》
 - ▶ 在固定版面内，摆置各种不同类型的资料，以最合适的方法呈现 W
- 遵循业界规范
- 追求良好的阅读体验 (readability)
 - 内容与格式分离
 - 内容永远比格式重要!



基本原则

- 排版 vs 文字处理
 - ▶ 《别把 L^AT_EX 当 Word 用》
 - ▶ 在固定版面内，摆置各种不同类型的资料，以最合适的方法呈现 W
- 遵循业界规范
- 追求良好的阅读体验 (readability)
- 内容与格式分离
- 内容永远比格式重要!



基本原则

- 排版 vs 文字处理
 - ▶ 《别把 L^AT_EX 当 Word 用》
 - ▶ 在固定版面内，摆置各种不同类型的资料，以最合适的方法呈现 W
- 遵循业界规范
- 追求良好的阅读体验 (readability)
- 内容与格式分离
- 内容永远比格式重要!



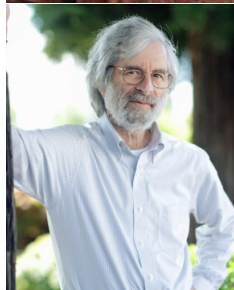
T_EX 与 L^AT_EX 的起源

- T_EX: $\tau\epsilon\chi$ (/ˈtɛx/, /ˈtɛk/)
 - ▶ 生成精美图书的排版系统
 - ▶ 最初由高德纳^a (Donald E. Knuth) 于 1978 年开发
 - ▶ 最新版本为 T_EX 3.14159265
 - ▶ 漂亮、美观、稳定、通用
 - ▶ 尤其擅长数学公式排版

- L^AT_EX (/ˈlɑːtɛx/, /ˈleɪtɛk/)
 - ▶ Leslie Lamport^b 开发的一种 T_EX 格式
 - ▶ 在 T_EX 的基础上提供宏包，降低使用门槛
 - ▶ 极其丰富的宏包，提供扩展功能
 - ▶ 广泛用于学术界，期刊会议论文模板

^a1974 年图灵奖得主，《计算机程序设计艺术》(The Art of Computer Programming) 作者。

^b2013 年图灵奖得主，对于分布式及并行系统的理论与实践具有基础性贡献。





L^AT_EX 的好处与坏处

好处

- 数学公式排版优雅 $\mathcal{F}(\xi) = \int_{-\infty}^{\infty} f(x) e^{-j2\pi\xi x} dx$
- 内容与格式分离
- 随心所欲的宏定义与自定义命令 `\newcommand`, `\def`

坏处

- 得到易读的版本，需要编译
- 输入相对 Word 繁琐
- 非开箱即用。有时自行解决编辑器、宏包，甚至是编译错误。



L^AT_EX 的好处与坏处

好处

- 数学公式排版优雅 $\mathcal{F}(\xi) = \int_{-\infty}^{\infty} f(x) e^{-j2\pi\xi x} dx$
- 内容与格式分离
- 随心所欲的宏定义与自定义命令 `\newcommand`, `\def`

坏处

- 得到易读的版本, 需要编译
- 输入相对 Word 繁琐
- 非开箱即用。有时自行解决编辑器、宏包, 甚至是编译错误。



选择发行版

■ T_EX 发行版 distribution

- ▶ 引擎、宏包、字体、文档的综合体
- ▶ 类比 Visual Studio
- ▶ T_EX Live、MacT_EX、W32T_EX、MiK_T_EX 等

■ T_EX Live

- ▶ 官方维护，首选，跨平台
- ▶ MacT_EX macOS 下的 T_EX Live
- ▶ 缺点：完整版体积大 7GB+、每年需重装

■ MiK_T_EX



- ▶ 由 Christian Schenk 维护（是个狠人）
- ▶ 宏包随用随装
- ▶ 缺点：部分细节与 T_EX Live 不兼容、网络问题

■ 不要安装 C_T_EX 套装！

- ▶ 存在严重 bug，并且完全过时（2012 年已经停止维护）。



选择发行版

- T_EX 发行版 distribution
 - ▶ 引擎、宏包、字体、文档的综合体
 - ▶ 类比 Visual Studio
 - ▶ T_EX Live、MacT_EX、W32T_EX、MiK_T_EX 等
- T_EX Live 
 - ▶ 官方维护，首选，跨平台
 - ▶ MacT_EX macOS 下的 T_EX Live
 - ▶ 缺点：完整版体积大 7GB+、每年需重装
- MiK_T_EX 
 - ▶ 由 Christian Schenk 维护（是个狠人）
 - ▶ 宏包随用随装
 - ▶ 缺点：部分细节与 T_EX Live 不兼容、网络问题
- 不要安装 C_T_EX 套装！
 - ▶ 存在严重 bug，并且完全过时（2012 年已经停止维护）。



选择发行版









- T_EX 发行版 distribution
 - ▶ 引擎、宏包、字体、文档的综合体
 - ▶ 类比 Visual Studio
 - ▶ T_EX Live、MacT_EX、W32T_EX、MiK_T_EX 等
- T_EX Live 
 - ▶ 官方维护，首选，跨平台
 - ▶ MacT_EX macOS 下的 T_EX Live
 - ▶ 缺点：完整版体积大 7GB+、每年需重装
- MiK_T_EX 
 - ▶ 由 Christian Schenk 维护（是个狠人）
 - ▶ 宏包随用随装
 - ▶ 缺点：部分细节与 T_EX Live 不兼容、网络问题
- 不要安装 C_T_EX 套装！
 - ▶ 存在严重 bug，并且完全过时（2012 年已经停止维护）。



选择本地编辑器

LaTeX 也是纯文本文件，后缀名为.tex，可以用任何文本编辑器编写

■ 专用型

- ▶ TeXworks: T_EX Live 自带   
- ▶ TeXStudio: 功能丰富，对新手友好   
- ▶ TeXShop: MacT_EX 自带 
- ▶ WinEdt: 功能丰富，收费 

■ 通用型









- ▶ Visual Studio Code: 借助插件 LaTeX Workshop + LaTeX Utilities
- ▶ Sublime Text: 收费
- ▶ Vim: q、q!、wq、wq!、...???
- ▶ Emacs: ctrl-s、ctrl-c ctrl-x、...???



选择本地编辑器

LaTeX 也是纯文本文件，后缀名为.tex，可以用任何文本编辑器编写

■ 专用型


- ▶ TeXworks: T_EX Live 自带   
- ▶ TeXStudio: 功能丰富，对新手友好   
- ▶ TeXShop: MacT_EX 自带 
- ▶ WinEdt: 功能丰富，收费 

■ 通用型

- ▶ Visual Studio Code: 借助插件 LaTeX Workshop + LaTeX Utilities
- ▶ Sublime Text: 收费
- ▶ Vim: q、q!、wq、wq!、...???
- ▶ Emacs: ctrl-s、ctrl-c ctrl-x、...???




不想安装?

- 云端服务更好用...吗?
- 免去安装、升级等一系列烦恼, 可以多人协作
- 版本控制、模板市场
- Overleaf: [Overleaf](#) 
 - ▶ 模板丰富
 - ▶ 用户支持很好
 - ▶ 注意网络环境 (咳咳)




不想安装?

- 云端服务更好用...吗?
- 免去安装、升级等一系列烦恼, 可以多人协作
- 版本控制、模板市场
- Overleaf: [Overleaf](#) 
 - ▶ 模板丰富
 - ▶ 用户支持很好
 - ▶ 注意网络环境 (咳咳)



不想安装？

- 云端服务更好用...吗？
- 免去安装、升级等一系列烦恼，可以多人协作
- 版本控制、模板市场
- Overleaf: [Overleaf](#) 
 - ▶ 模板丰富
 - ▶ 用户支持很好
 - ▶ 注意网络环境（咳咳）



引擎

- 引擎即编译器，是编译源代码生成文档的排版引擎
 - ▶ 有 TeX、pdfTeX、XeTeX、LuaTeX 等
- 有两种语言的格式，分别为 plainTeX（最初版本）和 LaTeX
- 编译指令即编译源代码生成文档的命令，根据引擎和格式来选择

引擎	目标格式	使用 plainTeX	使用 LaTeX
TeX	DVI	tex	
pdfTeX	DVI	etex	latex
pdfTeX	PDF	pdftex	pdflatex
XeTeX	PDF	xetex	xelatex
LuaTeX	PDF	luatex	lualatex



编译指令

- 纯文本文件，后缀名为 `.tex`（类比 Markdown 的 `.md`）
- 选择正确的编译指令
 - ▶ 例如 `xelatex main.tex`
 - ▶ 文中含有交叉引用、参考文献、目录等情况，需要多次编译
 - ▶ `latexmk` 秒了
- VSCode: LaTeX Workshop 插件秒了
- Overleaf 秒了



编译指令

- 纯文本文件，后缀名为 `.tex`（类比 Markdown 的 `.md`）
- 选择正确的编译指令
 - ▶ 例如 `xelatex main.tex`
 - ▶ 文中含有交叉引用、参考文献、目录等情况，需要多次编译
 - ▶ `latexmk` 秒了
- VSCode: LaTeX Workshop 插件秒了
- Overleaf 秒了



编译指令

- 纯文本文件，后缀名为 `.tex`（类比 Markdown 的 `.md`）
- 选择正确的编译指令
 - ▶ 例如 `xelatex main.tex`
 - ▶ 文中含有交叉引用、参考文献、目录等情况，需要多次编译
 - ▶ `latexmk` 秒了
- VSCode: LaTeX Workshop 插件秒了
- Overleaf 秒了





推荐安装

- 本地版最佳实践
 - ▶ + MiK_TE_X
 - ▶ + Visual Studio Code (LaTeX Workshop 等插件)
 - ▶ + git (代码管理工具)
 - ▶ + Github (全世界最大的程序员交友网站)
- 网络版最佳实践
 - ▶ 在线编辑器 [Overleaf](#) 编写
- 保姆级手把手的教程 [🔗](#)



推荐安装

- 本地版最佳实践
 - ▶ + MiK_TE_X
 - ▶ + Visual Studio Code (LaTeX Workshop 等插件)
 - ▶ + git (代码管理工具)
 - ▶ + Github (全世界最大的程序员交友网站)
- 网络版最佳实践
 - ▶ 在线编辑器 [Overleaf](#)  编写
- 保姆级手把手的教程 



推荐安装

- 本地版最佳实践
 - ▶ + MiK_TE_X
 - ▶ + Visual Studio Code (LaTeX Workshop 等插件)
 - ▶ + git (代码管理工具)
 - ▶ + Github (全世界最大的程序员交友网站)
- 网络版最佳实践
 - ▶ 在线编辑器 [Overleaf](#) 编写
- 保姆级手把手的教程 [🔗](#)



Hello World!

```
% 用 pdfLaTeX、XeLaTeX 或 LuaLaTeX 编译
\documentclass{article}
\begin{document}
Hello world!
\end{document}
```

```
% 用 XeLaTeX 或 LuaLaTeX 编译
\documentclass{ctexart}
\begin{document}
你好中国!
\end{document}
```




Hello World!

% 用 *pdfLaTeX*、*XeLaTeX* 或 *LuaLaTeX* 编译

```
\documentclass{article}
```

```
\begin{document}
```

Hello world!

```
\end{document}
```

% 用 *XeLaTeX* 或 *LuaLaTeX* 编译

```
\documentclass{ctexart}
```

```
\begin{document}
```

你好中国!

```
\end{document}
```



语法

- 命令（控制序列）：以 \ 开头，大小写敏感
 - ▶ `\LaTeX{}` → \LaTeX
 - ▶ 有作用域的概念，可以用 {} 限定作用域
 - ▶ 可以接收参数，如 `\documentclass{book}`
 - ▶ 必需参数用 {}，可选参数用 []
- 注释以 % 开头
- 连续多个空格 = 单个空格 = 单个换行
- 可以定义自己的命令



语法

- 命令（控制序列）：以 \ 开头，大小写敏感
 - ▶ `\LaTeX{}` → \LaTeX
 - ▶ 有作用域的概念，可以用 {} 限定作用域
 - ▶ 可以接收参数，如 `\documentclass{book}`
 - ▶ 必需参数用 {}，可选参数用 []
- 注释以 % 开头
- 连续多个空格 = 单个空格 = 单个换行
- 可以定义自己的命令



文件的结构

```
\documentclass{article} % 文档类
% 导言区 (preamble): 加载宏包、定义命令、设置文档格式等

\begin{document}          % 文档开始
% 正文区 (body): 文档的内容
Hello world!

\input{content.tex}        % 也可以导入其他文件内容
\end{document}             % 文档结束，后续内容不会被编译
```



文档类与宏包

- L^AT_EX 文档开头必须包含 `\documentclass` 指定文档类
- 基础文档类: article、report、book、beamer 等
- 中文变体: ctexart、ctexrep、ctexbook、ctexbeamer 等
- 可以通过可选参数指定字号、纸张大小、双面打印等选项

```
\documentclass[12pt,a4paper,twoside]{article}
```

- 宏包相当于第三方库, 可以引入更丰富的扩展功能

```
\usepackage{amsmath}
```

```
\usepackage[final]{minted}
```

- T_EX Live full 自带大量宏包
- 通过 `texdoc package` 查看文档



文档类与宏包

- L^AT_EX 文档开头必须包含 `\documentclass` 指定文档类
- 基础文档类: article、report、book、beamer 等
- 中文变体: ctexart、ctexrep、ctexbook、ctexbeamer 等
- 可以通过可选参数指定字号、纸张大小、双面打印等选项

```
\documentclass[12pt,a4paper,twoside]{article}
```

- 宏包相当于第三方库, 可以引入更丰富的扩展功能

```
\usepackage{amsmath}
```

```
\usepackage[final]{minted}
```

- T_EX Live full 自带大量宏包
- 通过 texdoc *package* 查看文档



段落 (paragraph)

- 正常写作即可，以空行或 `\par` 分段
- 行首空格会被忽略，多个空格会被合并为一个
- 换行在非中文语境下视作一个空格

```
Consecutive spaces are  
squashed in \LaTeX{}. 中文语境下  
不会添加空格。Leading spaces are  
ignored. Trailing comment doesn't  
break a word.
```

```
A new paragraph.  
或者用 \verb|\par| 也可以。
```

Consecutive spaces are squashed in \LaTeX . 中文语境下不会添加空格。Leading spaces are ignored. Trailing comment doesn't break a word.

A new paragraph.

或者用 `\par` 也可以。



段落 (paragraph)

- 正常写作即可，以空行或 `\par` 分段
- 行首空格会被忽略，多个空格会被合并为一个
- 换行在非中文语境下视作一个空格

Consecutive spaces are
squashed in `\LaTeX{}`. 中文语境下
不会添加空格。Leading spaces are
ignored. Trailing comment doesn't
break a word.

A new paragraph.
或者用 `\verb|\par|` 也可以。

Consecutive spaces are squashed in \LaTeX . 中文语境下不会添加空格。Leading spaces are ignored. Trailing comment doesn't break a word.

A new paragraph.

或者用 `\par` 也可以。



断行与断页

- 可以用 `\\` 或 `\newline` 强制断行
 - ▶ 类似 Word 中的“软回车”，不会产生新的段落
- `\newpage` 强制断页
- `\clearpage` 强制断页并清空浮动体
 - ▶ We'll cover this soon ;)
 - ▶ 一般用 `\newpage` 即可

```
Starting a new paragraph. \\  
Oops,a new line within it.
```

```
Finally another paragraph. \newline  
Another new line,oh my!
```

Starting a new paragraph.
Oops, a new line within it.

Finally another paragraph.
Another new line, oh my!



断行与断页

- 可以用 `\\` 或 `\newline` 强制断行
 - ▶ 类似 Word 中的“软回车”，不会产生新的段落
- `\newpage` 强制断页
- `\clearpage` 强制断页并清空浮动体
 - ▶ We'll cover this soon ;)
 - ▶ 一般用 `\newpage` 即可

Starting a new paragraph. `\\`
Oops,a new line within it.

Finally another paragraph. `\newline`
Another new line,oh my!

Starting a new paragraph.
Oops, a new line within it.

Finally another paragraph.
Another new line, oh my!



字形与字号

- 有两类修改字形的命令：
`{\bfseries bold}` 和 `\textbf{bold}`
 - ▶ 前者对作用域内的所有内容生效，后者只对参数内容生效
 - ▶ 都是“内置”字体的不同样式，自定义字体？留作习题
- 修改字号：`{\large Hi}`
- 汉字一般不使用斜体
- 不要滥用字体样式（本页除外）

```
\textbf{bold} \textsf{sans serif}
\texttt{typewriter} \textsc{Small Caps}
\textit{italic} \textsl{slanted}

{\tiny tiny} {\scriptsize scriptsize}
{\footnotesize footnotesize} {\small small}
{\normalsize normalsize} {\large large}
{\Large Large} {\LARGE LARGE}
{\huge huge} {\Huge Huge}
```

bold sans serif typewriter SMALL
CAPS *italic slanted*
tiny scriptsize footnotesize small normalsize
large Large LARGE huge
Huge



字形与字号

- 有两类修改字形的命令：
`{\bfseries bold}` 和 `\textbf{bold}`
 - ▶ 前者对作用域内的所有内容生效，后者只对参数内容生效
 - ▶ 都是“内置”字体的不同样式，自定义字体？留作习题
- 修改字号：`{\large Hi}`
- 汉字一般不使用斜体
- 不要滥用字体样式（本页除外）

```
\textbf{bold} \textsf{sans serif}
\texttt{typewriter} \textsc{Small Caps}
\textit{italic} \textsl{slanted}

{\tiny tiny} {\scriptsize scriptsize}
{\footnotesize footnotesize} {\small small}
{\normalsize normalsize} {\large large}
{\Large Large} {\LARGE LARGE}
{\huge huge} {\Huge Huge}
```

bold sans serif typewriter SMALL
CAPS *italic slanted*
tiny scriptsize footnotesize small normalsize
large Large LARGE huge
Huge



字形与字号

- 有两类修改字形的命令：
`{\bfseries bold}` 和 `\textbf{bold}`
 - ▶ 前者对作用域内的所有内容生效，后者只对参数内容生效
 - ▶ 都是“内置”字体的不同样式，自定义字体？留作习题
- 修改字号：`{\large Hi}`
- 汉字一般不使用斜体
- 不要滥用字体样式（本页除外）

```
\textbf{bold} \textsf{sans serif}
\texttt{typewriter} \textsc{Small Caps}
\textit{italic} \textsl{slanted}

{\tiny tiny} {\scriptsize scriptsize}
{\footnotesize footnotesize} {\small small}
{\normalsize normalsize} {\large large}
{\Large Large} {\LARGE LARGE}
{\huge huge} {\Huge Huge}
```

bold sans serif typewriter SMALL
CAPS *italic slanted*

tiny scriptsize footnotesize small normalsize

large Large LARGE huge
Huge



关于字符和标点

- 特殊字符需要转义，如 `_、\%、\$、\&、\{、\}` 等等
 - ▶ `\~{}`、`\^{}` 带大括号防止将后面的字符理解为参数
 - ▶ `\textbackslash` 反斜杠
- 西文语境下的引号：``double'`` “double”
 - ▶ 中文语境下正常输入即可
- 连字符：-、短破折号：--、长破折号：---
 - ▶ Newton-Leibniz Formula
 - ▶ Henry Kissinger (1923–2023)
 - ▶ Don't use \$\$ — it breaks spacing!
- 防止西文连字：`dif{ }f{ }icult` difficult
 - ▶ 对比：difficult



强制行距/间距

- 长度单位: pt, in (英寸), cm, mm, em (小写 'm' 的宽度), ex (小写 'x' 的高度)
- 行距: `\linespread{factor}` (默认行距 1.2 倍字号大小)
 - ▶ 局部修改要在后面加 `\selectfont`, 且要在范围内手动分段
- 插入水平间距 (多空格无效): `\hspace{length}`
 - ▶ `\quad` 相当于 1em, `\qquad` 相当于 2em
 - ▶ `\hspace*` 防止因为断行而消失
- 插入垂直间距: `\vspace{length}`
 - ▶ `\\[1em]` 可以在换行的同时插入垂直间距

```
{\linespread{1.6}\selectfont This is a paragraph with 1.6 linespread.\\
This is the next line and \quad space and \hspace{3em} more and
\hspace{\fill}lol\par

\vspace{-1em}Next paragraph.\\[1em] and next line.
```

This is a paragraph with 1.6 linespread.

This is the next line and space and more and lol
Next paragraph.

and next line.



强制行距/间距

- 长度单位: pt, in (英寸), cm, mm, em (小写 'm' 的宽度), ex (小写 'x' 的高度)
- 行距: `\linespread{factor}` (默认行距 1.2 倍字号大小)
 - ▶ 局部修改要在后面加 `\selectfont`, 且要在范围内手动分段
- 插入水平间距 (多空格无效): `\hspace{length}`
 - ▶ `\quad` 相当于 1em, `\qquad` 相当于 2em
 - ▶ `\hspace*` 防止因为断行而消失
- 插入垂直间距: `\vspace{length}`
 - ▶ `\\[1em]` 可以在换行的同时插入垂直间距

```
{\linespread{1.6}\selectfont This is a paragraph with 1.6 linespread.\\
This is the next line and \quad space and \hspace{3em} more and
\hspace{\fill}lol\par}

\vspace{-1em}Next paragraph.\\[1em] and next line.
```

This is a paragraph with 1.6 linespread.

This is the next line and space and more and lol
Next paragraph.

and next line.



标题页

在导言区：

- `\title{title}`：标题
- `\author{author}`：作者
- `\date{date}`：日期，不写则默认为当前日期 `\today`

在正文区使用 `\maketitle` 生成标题页

```
\title{\textbf{我是标题}}  
\author{箱子不知道哦}  
\date{1969 年 12 月 31 日}  
\begin{document}  
\maketitle  
\end{document}
```

我是标题

箱子不知道哦

1969 年 12 月 31 日



标题页

在导言区：

- `\title{title}`：标题
- `\author{author}`：作者
- `\date{date}`：日期，不写则默认为当前日期 `\today`

在正文区使用 `\maketitle` 生成标题页

```
\title{\textbf{我是标题}}  
\author{箱子不知道哦}  
\date{1969 年 12 月 31 日}  
\begin{document}  
\maketitle  
\end{document}
```

我是标题

箱子不知道哦

1969 年 12 月 31 日



环境

- 成对出现的命令，用于控制一段内容的格式
- 以 `\begin{environment}` 开始，以 `\end{environment}` 结束
- 常见环境：itemize、figure、table 等

```
\begin{itemize}
  \item macOS
  \item Windows
  \item Linux
\end{itemize}
```

- macOS
- Windows
- Linux



环境

- 成对出现的命令，用于控制一段内容的格式
- 以 `\begin{environment}` 开始，以 `\end{environment}` 结束
- 常见环境：itemize、figure、table 等

```
\begin{itemize}  
  \item macOS  
  \item Windows  
  \item Linux  
\end{itemize}
```

- macOS
- Windows
- Linux



章节和目录

- 应将文档合理地分割为章、节、小节等
 - ▶ `\part`、`\chapter` 仅适用于 book 和 report 文档类
 - ▶ `\section`、`\subsection`、`\subsubsection`
 - ▶ 不常用: `\paragraph`、`\subparagraph`
- 自动编号, 默认编号深度为 3, 可以修改
- 带星号的版本不会自动编号, 也不会出现在目录中 (`\section*`)
- `\tableofcontents` 生成目录
 - ▶ 如果不使用 latexmk, 需要多次编译
- `\appendix` 起为附录, 后续章节编号为大写字母



章节和目录

- 应将文档合理地分割为章、节、小节等
 - ▶ `\part`、`\chapter` 仅适用于 book 和 report 文档类
 - ▶ `\section`、`\subsection`、`\subsubsection`
 - ▶ 不常用: `\paragraph`、`\subparagraph`
- 自动编号, 默认编号深度为 3, 可以修改
- 带星号的版本不会自动编号, 也不会出现在目录中 (`\section*`)
- `\tableofcontents` 生成目录
 - ▶ 如果不使用 latexmk, 需要多次编译
- `\appendix` 起为附录, 后续章节编号为大写字母



层次与目录生成

```
\tableofcontents % 这里是目录
\part{有监督学习}
\chapter{支持向量机}
\section{支持向量机简介}
\subsection{支持向量机的历史}
\subsubsection{支持向量机的诞生}
\paragraph{一些趣闻}
\subparagraph{第一个趣闻}
```

第一部分 有监督学习

第一章 支持向量机

1. 支持向量机简介

1.1 支持向量机的历史

1.1.1 支持向量机的诞生

一些趣闻

第一个趣闻



图片

- 需要 graphicx 宏包 (注意不是 graphics)
- `\includegraphics[options]{filename}`
 - ▶ 常用选项: width、height、scale 宽高缩放
 - ▶ `filename` 可以是相对路径或绝对路径

```
\usepackage{graphicx}
\graphicspath{{lec4/}}
% ...
\begin{figure}
  \centering
  \includegraphics[...]{logo/logo.png}
  \caption{上海电力大学校徽}
  \label{fig:suep:LOGO}
\end{figure}
```



图: 上海电力大学校徽



图片

- 需要 graphicx 宏包 (注意不是 graphics)
- `\includegraphics[options]{filename}`
 - ▶ 常用选项: width、height、scale 宽高缩放
 - ▶ `filename` 可以是相对路径或绝对路径

```
\usepackage{graphicx}
\graphicspath{{lec4/}}
% ...
\begin{figure}
  \centering
  \includegraphics[...]{logo/logo.png}
  \caption{上海电力大学校徽}
  \label{fig:suep:LOGO}
\end{figure}
```



图: 上海电力大学校徽



浮动体

- 再次强调：别把 \LaTeX 当 Word 用

```
\begin{figure}[htbp]
% ...
\end{figure}
```

- 摆脱“见下图”、“如上图所示”、“如表所示”.....
- [htbp] 是浮动体的位置选项：here / top / bottom / page
- 使用交叉引用：`\label` 和 `\ref`、`\pageref`、`\nameref`、`\eqref`
- 进阶：hyperref 与 cleveref 等宏包

图 `\ref{fig:ckc}` 是上海电力大学校徽
(在 PDF 中是 `\pageref{fig:ckc}` 页)。

图 ?? 是上海电力大学校徽 (在 PDF 中是 ?? 页)。

推荐阅读：漫谈 \LaTeX 排版常见概念误区



浮动体

- 再次强调：别把 \LaTeX 当 Word 用

```
\begin{figure}[htbp]
% ...
\end{figure}
```

- 摆脱“见下图”、“如上图所示”、“如表所示”.....
- [htbp] 是浮动体的位置选项：here / top / bottom / page
- 使用交叉引用： \label 和 \ref 、 \pageref 、 \nameref 、 \eqref
- 进阶：hyperref 与 cleveref 等宏包

图 $\text{\ref{fig:ckc}}$ 是上海电力大学校徽
(在 PDF 中是 $\text{\pageref{fig:ckc}}$ 页)。

图 ?? 是上海电力大学校徽 (在 PDF 中是 ?? 页)。

推荐阅读：漫谈 \LaTeX 排版常见概念误区



设置页大小及页边距

- geometry 宏包
- 全局分栏: `\twocolumn` 和 `\onecolumn`
- 局部分栏: multicol 宏包的 multicol 环境

```
\usepackage{geometry}

\geometry{paper=b5paper}           % 标准纸张大小
\geometry{paperwidth=260mm,
          paperheight=185mm}      % 自定义纸张大小
\geometry{margin=1in}             % 全相等
\geometry{left=1in,right=1in,
          top=1in,bottom=1in}    % 分别设置
\geometry{hmargin=1.5in,vmargin=1in} % 水平和垂直
```



列表

- itemize: 无序列表
- enumerate: 有序列表
- description: 描述列表
- 列表中的每一项都是一个 `\item`
 - ▶ 可选参数可以修改编号样式
- 推荐使用 `enumitem` 宏包

```
\begin{description}
  \item[Slackware] 历史最悠久
  \item[Debian] 历史第二悠久
  \item[Ubuntu] 基于 Debian, 最流行
  \item[Kali Linux] 面向安全工作者
\end{description}
```

Slackware 历史最悠久
Debian 历史第二悠久
Ubuntu 基于 Debian, 最流行
Kali Linux 面向安全工作者



列表

- itemize: 无序列表
- enumerate: 有序列表
- description: 描述列表
- 列表中的每一项都是一个 `\item`
 - ▶ 可选参数可以修改编号样式
- 推荐使用 `enumitem` 宏包

```
\begin{description}
  \item[Slackware] 历史最悠久
  \item[Debian] 历史第二悠久
  \item[Ubuntu] 基于 Debian, 最流行
  \item[Kali Linux] 面向安全工作者
\end{description}
```

Slackware 历史最悠久
Debian 历史第二悠久
Ubuntu 基于 Debian, 最流行
Kali Linux 面向安全工作者



对齐环境

- center 居中, flushleft 左对齐, flushright 右对齐
 - ▶ 这里指的是对齐环境
 - ▶ 会在环境上下额外生成间距
- `\centering`、`\raggedright`、`\raggedleft`
 - ▶ 这里指的是对齐命令
 - ▶ 不会生成额外间距, 直接改变对齐方式
 - ▶ 注意左对齐是 `\raggedright`

```
\begin{center}
```

```
some text
```

```
\end{center}
```

```
\begin{flushright}
```

```
some text
```

```
\end{flushright}
```

```
\centering some text \par
```

```
\raggedleft some text
```

some text

some text

some text

some text



对齐环境

- center 居中, flushleft 左对齐, flushright 右对齐
 - ▶ 这里指的是对齐环境
 - ▶ 会在环境上下额外生成间距
- \centering、\raggedright、\raggedleft
 - ▶ 这里指的是对齐命令
 - ▶ 不会生成额外间距, 直接改变对齐方式
 - ▶ 注意左对齐是 \raggedright

```
\begin{center}  
  some text  
\end{center}  
\begin{flushright}  
  some text  
\end{flushright}  
  
\centering some text \par  
\raggedleft some text
```

some text

some text

some text

some text



数学模式

- 数学公式排版是 \LaTeX 的绝对强项
- 一切数学公式都要在数学模式下输入
 - ▶ 建议始终调用 `amsmath` 宏包，由美国数学学会 (AMS) 提供
 - ▶ 还有更加现代的 `unicode-math` 宏包，兼容了 Unicode 字符和 OpenType 字体
- 数学模式与文本模式的区别
 - ▶ 一些符号的输出不同
 - ▶ 有自己的字体、间距等规则
 - ▶ 无视空格，不可有空行
- 行内 (inline) 公式，用 `$...$` 或 `\(...\)` 包裹
- 行间 (display) 公式
 - ▶ 单行公式用 `\[...\]` 包裹
 - ▶ 多行公式用 `equation` / `align` / `gather` 等环境
 - ▶ 不要用 `$$...$$`: \TeX 原始语法，会产生很多问题
 - ▶ 想输入正常的文本? `\text{...}`



数学模式 — cont.

Does $\sum_{n=1}^{+\infty} n$ equal to

$(-\frac{1}{12})$?

$[\sum_{n=1}^{+\infty} n$

$= \lim_{n \rightarrow +\infty} \frac{n(n+1)}{2}$

$= +\infty \neq -\frac{1}{12}]$

$\begin{equation} \label{eq:cauchy}$

$\frac{f'(\xi)}{g'(\xi)} = \frac{f(b)-f(a)}{g(b)-g(a)}$

$\end{equation}$

式 $\eqref{eq:cauchy}$ 称为 Cauchy 中值定理。

\begin{align}

$\cos^2 \theta$

$= \cos^2 \theta - \sin^2 \theta$

$= 2 \cos^2 \theta - 1$

\end{align}

Does $\sum_{n=1}^{+\infty} n$ equal to $-\frac{1}{12}$?

$$\sum_{n=1}^{+\infty} n = \lim_{n \rightarrow +\infty} \frac{n(n+1)}{2} = +\infty \neq -\frac{1}{12}$$

$$\frac{f'(\xi)}{g'(\xi)} = \frac{f(b)-f(a)}{g(b)-g(a)} \quad (1)$$

式 (1) 称为 Cauchy 中值定理。

$$\cos 2\theta = \cos^2 \theta - \sin^2 \theta \quad (2)$$

$$= 2 \cos^2 \theta - 1 \quad (3)$$



公式排版

- 所有的字母都作为变量处理，注意命令后面的空格
- 上下标：`^` 和 `_`
- 函数与常用运算符：`\sin`、`\log`、`\lim`、`\max` 等
- 巨算符：`\sum`、`\prod`、`\int` 等
 - ▶ 在行内公式中，上下标会被压缩（见 34 页）
 - ▶ 可以使用 `\limits` 强制显示上下标
 - ▶ 建议阅读 `lshort` 的相关章节
- 手动调节间距：`\,`、`\:`、`_`、`\!`、`\quad`、`\qquad` 等
- 分式：`\frac{num}{denom}`
 - ▶ 行内分式不好看？考虑写成 a/b 或改用行间公式
 - ▶ 不推荐 `\dfrac` 一把梭



常用数学符号

- 希腊字母: `\alpha` α 、`\beta` β 、`\Gamma` Γ 、`\Delta` Δ 等
- 无穷大: `\infty` ∞
- 根式: `\sqrt{2}` $\sqrt{2}$ 、`\sqrt[n]{x}` $\sqrt[n]{x}$
- 省略号: `\dots` \dots 、`\ldots` \ldots 、`\cdots` \cdots 、`\vdots` \vdots 、`\ddots` \ddots
- 关系: `\leq` \leq vs. `\leqslant` \leqslant 、`\neq` \neq 、`\in` \in 、`\subset` \subset 等
- 矩阵与行列式: `matrix`、`pmatrix`、`vmatrix` 等环境

```
\[ \begin{vmatrix}
1 & 2 & 3 \\
2 & 3 & 1 \\
3 & 1 & 2
\end{vmatrix} \]
```

注意上下标: `a_{ij}^{xy}` vs. `a_{ij}^{xy}`.

$$\begin{vmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 3 & 1 & 2 \end{vmatrix}$$

注意上下标: a_{ij}^{xy} vs. a_{ij}^{xy} 。



常用数学符号

- 希腊字母: `\alpha` α 、`\beta` β 、`\Gamma` Γ 、`\Delta` Δ 等
- 无穷大: `\infty` ∞
- 根式: `\sqrt{2}` $\sqrt{2}$ 、`\sqrt[n]{x}` $\sqrt[n]{x}$
- 省略号: `\dots` \dots 、`\ldots` \ldots 、`\cdots` \cdots 、`\vdots` \vdots 、`\ddots` \ddots
- 关系: `\leq` \leq vs. `\leqslant` \leqslant 、`\neq` \neq 、`\in` \in 、`\subset` \subset 等
- 矩阵与行列式: `matrix`、`pmatrix`、`vmatrix` 等环境

```
\[ \begin{vmatrix}
1 & 2 & 3 \\
2 & 3 & 1 \\
3 & 1 & 2
\end{vmatrix}
```

注意上下标: `a_{ij}^{xy}` vs. `a_{ij}^{xy}`.

$$\begin{vmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 3 & 1 & 2 \end{vmatrix}$$

注意上下标: $a_i j^x y$ vs. a_{ij}^{xy} 。



括号与定界符

- 基本括号 () [] { }
- 绝对值、范数: $|x|$ $|x|$ 、 $\|x\|$ $\|x\|$
 - 或使用 `\vert`、`\Vert`
- 注意区别: `\langle` x `\rangle` $\langle x \rangle$ vs. $\langle x \rangle$ $\langle x \rangle$
- 自动调节大小: 使用 `\left` 和 `\right`
- 手动调节大小: `\big`、`\Big`、`\bigg`、`\Bigg`

```
\[ \sec(\theta^2) \]
```

```
\[ \sec\big(\theta^2\big) \]
```

```
\[ \lfloor \frac{xy}{x+y} \rfloor \]
```

```
\[ \left\lfloor \frac{xy}{x+y} \right\rfloor \]
```

$\sec(\theta^2)$

$\sec(\theta^2)$

$\left\lfloor \frac{xy}{x+y} \right\rfloor$

$\left\lfloor \frac{xy}{x+y} \right\rfloor$



括号与定界符

- 基本括号 () [] { }
- 绝对值、范数: $|x|$ $|x|$ 、 $\|x\|$ $\|x\|$
 ▶ 或使用 `\vert`、`\Vert`
- 注意区别: `\langle x \rangle` $\langle x \rangle$ vs. $\langle x \rangle < x >$
- 自动调节大小: 使用 `\left` 和 `\right`
- 手动调节大小: `\big`、`\Big`、`\bigg`、`\Bigg`

```
\[ \sec(\theta^2) \]
```

```
\[ \sec\big(\theta^2\big) \]
```

```
\[ \lfloor \frac{xy}{x + y} \rfloor \]
```

```
\[ \left\lfloor \frac{xy}{x + y} \right\rfloor \]
```

$$\sec(\theta^2)$$



$$\sec(\theta^2)$$

$$\left\lfloor \frac{xy}{x + y} \right\rfloor$$

$$\left\lfloor \frac{xy}{x + y} \right\rfloor$$



符号与字体

- 符号不是按钮点出来的，也不是天上掉下来的
 - ▶ (几乎) 所有的符号都由字体提供
 - ▶ 分清「它是什么」和「它长什么样」(术语: character 和 glyph)
- 寻找符号
 - ▶ 最常用的额外字体包: `amssymb`
 - ▶ \LaTeX 公式大全 
 - ▶ 在线 \LaTeX 公式编辑器 (支持图片识别) 
- 数学字体
 - ▶ 你们要的「Times New Roman»: `newtxmath` 宏包
 - ▶ 不要用 `times` 和 `mathptmx` 宏包
 - ▶ 加粗: 使用 `bm` 宏包的 `\bm` 命令 (`\mathbf` 只有直立的字母)





符号与字体

- 符号不是按钮点出来的，也不是天上掉下来的
 - ▶ (几乎) 所有的符号都由字体提供
 - ▶ 分清「它是什么」和「它长什么样」(术语: character 和 glyph)
- 寻找符号
 - ▶ 最常用的额外字体包: `amssymb`
 - ▶ \LaTeX 公式大全
 - ▶ 在线 \LaTeX 公式编辑器 (支持图片识别)
- 数学字体
 - ▶ 你们要的「Times New Roman»: `newtxmath` 宏包
 - ▶ 不要用 `times` 和 `mathptmx` 宏包
 - ▶ 加粗: 使用 `bm` 宏包的 `\bm` 命令 (`\mathbf` 只有直立的字母)





符号与字体

- 符号不是按钮点出来的，也不是天上掉下来的
 - ▶ (几乎) 所有的符号都由字体提供
 - ▶ 分清「它是什么」和「它长什么样」(术语: character 和 glyph)
- 寻找符号
 - ▶ 最常用的额外字体包: `amssymb`
 - ▶ \LaTeX 公式大全 
 - ▶ 在线 \LaTeX 公式编辑器 (支持图片识别) 
- 数学字体
 - ▶ 你们要的「Times New Roman»: `newtxmath` 宏包
 - ▶ 不要用 `times` 和 `mathptmx` 宏包
 - ▶ 加粗: 使用 `bm` 宏包的 `\bm` 命令 (`\mathbf` 只有直立的字母)




符号与字体

- 符号不是按钮点出来的，也不是天上掉下来的
 - ▶ (几乎) 所有的符号都由字体提供
 - ▶ 分清「它是什么」和「它长什么样」(术语: character 和 glyph)
- 寻找符号
 - ▶ 最常用的额外字体包: `amssymb`
 - ▶ \LaTeX 公式大全 
 - ▶ 在线 \LaTeX 公式编辑器 (支持图片识别) 
- 数学字体
 - ▶ 你们要的「Times New Roman»: `newtxmath` 宏包
 - ▶ 不要用 `times` 和 `mathptmx` 宏包
 - ▶ 加粗: 使用 `bm` 宏包的 `\bm` 命令 (`\mathbf` 只有直立的字母)



符号与字体

- 符号不是按钮点出来的，也不是天上掉下来的
 - ▶ (几乎) 所有的符号都由字体提供
 - ▶ 分清「它是什么」和「它长什么样」(术语: character 和 glyph)
- 寻找符号
 - ▶ 最常用的额外字体包: `amssymb`
 - ▶ \LaTeX 公式大全 
 - ▶ 在线 \LaTeX 公式编辑器 (支持图片识别) 
- 数学字体
 - ▶ 你们要的「Times New Roman»: `newtxmath` 宏包
 - ▶ 不要用 `times` 和 `mathptmx` 宏包
 - ▶ 加粗: 使用 `bm` 宏包的 `\bm` 命令 (`\mathbf` 只有直立的字母)



特殊数学字体

- 数学模式中不要使用文本模式的字体命令（除非你知道自己在做什么）
- 针对数学环境中的字符有特定的命令

命令	样式	备注
<code>\mathrm{...}</code>	ABCDEabcde1234	粗斜体使用 <code>\boldsymbol</code>
<code>\mathit{...}</code>	<i>ABCDEabcde1234</i>	
<code>\mathbf{...}</code>	ABCDEabcde1234	
<code>\mathsf{...}</code>	ABCDEabcde1234	
<code>\mathtt{...}</code>	ABCDEabcde1234	
<code>\mathcal{...}</code>	<i>ABCDE</i>	只有大写
<code>\mathbb{...}</code>	ABCDE	只有大写，依赖 <code>amssymb</code>
<code>\mathfrak{...}</code>	<i>ABCDEabcde1234</i>	依赖 <code>amssymb</code>
<code>\mathscr{...}</code>	<i>ABCDE</i>	只有大写，依赖 <code>mathrsfs</code>



现代的数学输入方式

- L^AT_EX 的公式确实很强大，但是..... 符号有点难记？
- 新方案：unicode-math 提供了几乎所见即所得的公式输入
 - ▶ 符号、字体、样式精调的一揽子解决方案
 - ▶ 可直接输入各类符号对应的 Unicode 字符（需要使用 UTF-8 编码）
 - ▶ 彻底修改底层，不可与传统方案混用
 - ▶ 自动加载 amsmath，不需要再使用 `\usepackage{amsmath}`

```
\begin{equation*}
\Gamma(x) \, dx = \pm\infty
\end{equation*}

\begin{align*}
\boldsymbol{\beta} &= \beta \, \boldsymbol{I} \, \\
\boldsymbol{a} &= a \, \boldsymbol{I}
\end{align*}
```

$$\Gamma(x)dx = \pm\infty$$

$$\boldsymbol{\beta} = \beta \boldsymbol{I}$$

$$\boldsymbol{a} = a \boldsymbol{I}$$



现代的数学输入方式

- L^AT_EX 的公式确实很强大，但是..... 符号有点难记？
- 新方案：unicode-math 提供了几乎所见即所得的公式输入
 - ▶ 符号、字体、样式精调的一揽子解决方案
 - ▶ 可直接输入各类符号对应的 Unicode 字符（需要使用 UTF-8 编码）
 - ▶ 彻底修改底层，不可与传统方案混用
 - ▶ 自动加载 amsmath，不需要再使用 `\usepackage{amsmath}`

```
\begin{equation*}
\Gamma(x) \, dx = \pm\infty
\end{equation*}

\begin{align*}
\mathbf{\beta} &= \beta \mathbf{I} \\
\mathbf{a} &= a \mathbf{I}
\end{align*}
```

$$\Gamma(x)dx = \pm\infty$$

$$\boldsymbol{\beta} = \beta \mathbf{I}$$


$$\mathbf{a} = a \mathbf{I}$$



一些需要注意的规范写法


- 特定函数一定要用专门命令，或写为正体

▶  `\mathrm{lim}_{x\to 0}` $\log_2 x$ $\lim_{x\rightarrow 0} \log_2 x$


▶  `\lim_{x\to 0} \log_2 x` $\lim_{x\rightarrow 0} \log_2 x$


- 除了变量以外都要用正体，特别是微分算子

▶  `\frac{d}{dx}` $\frac{d}{dx}$

▶  `\frac{\mathrm{d}}{\mathrm{d}x}` $\frac{\mathrm{d}}{\mathrm{d}x}$


- 建议在微分算子之前加上 `\`，调整间距


▶  `\int x\mathrm{d}x` $\int x dx$

▶  `\int x\,,\mathrm{d}x` $\int x \, dx$

- 多字符变量使用 `\mathit` 或其他字体，不要裸写

```
\[ \begin{matrix}
XYZ & Duration \\
\mathit{XYZ} & \mathit{Duration}
\end{matrix} \]
```

 XYZ $Duration$

 XYZ $Duration$



小露身手

$$\oint \mathcal{D}[x(t)] \sqrt{\frac{3\pi^2 - \sum_{q=0}^{\infty} (z + \hat{L})^q \exp(i^2 \hbar x)}{(\text{Tr } \mathcal{A}) \left(\Lambda_{j_1 j_2}^{i_1 i_2} \Gamma_{i_1 i_2}^{j_1 j_2} \hookrightarrow \vec{D} \cdot \mathbf{P} \right)}} = \underbrace{\left\langle \frac{\notin \emptyset}{\varpi \alpha_{k \uparrow}} \left| \frac{\partial_{\mu} T_{\mu\nu}}{2} \right. \right\rangle}_{\text{K}_3\text{Fe}(\text{CN})_6}, \forall z \in \mathbb{R}$$

```
\begin{equation*} % \usepackage{unicode-math}
\oint \mathscr{D}[x(t)] \sqrt{\frac{3 \pi^2 - \sum_{q=0}^{\infty} (z + \hat{L})^q
\exp \left( \mathrm{i}^2 \ \hbar x \right) }
{(\operatorname{Tr} \ \mathscr{A}) \left( \Lambda_{j_{\scriptscriptstyle 1} \ j_{\scriptscriptstyle 2}}^{i_{\scriptscriptstyle 1} \ i_{\scriptscriptstyle 2}} \ \Gamma_{i_{\scriptscriptstyle 1} \ i_{\scriptscriptstyle 2}}^{j_{\scriptscriptstyle 1} \ j_{\scriptscriptstyle 2}}
\hookrightarrow \vec{D} \cdot \mathrm{P} \right) }} =
\underbrace{\widetilde{\left\langle \frac{\notin \emptyset}{\varpi \alpha_{k \uparrow}} \left| \frac{\partial_{\mu} T_{\mu\nu}}{2} \right. \right\rangle}}_{\mathrm{K}_3\mathrm{Fe}(\mathrm{CN})_6}, \forall z \in \mathbb{R}
\end{equation*}
```



小露身手

$$\oint \mathcal{D}[x(t)] \sqrt{\frac{3\pi^2 - \sum_{q=0}^{\infty} (z + \hat{L})^q \exp(i^2 \hbar x)}{(\text{Tr } \mathcal{A}) \left(\Lambda_{j_1 j_2}^{i_1 i_2} \Gamma_{i_1 i_2}^{j_1 j_2} \hookrightarrow \vec{D} \cdot \text{P} \right)}} = \underbrace{\left\langle \frac{\notin \emptyset}{\varpi \alpha_{k \uparrow}} \left| \frac{\partial_{\mu} T_{\mu\nu}}{2} \right. \right\rangle}_{\text{K}_3\text{Fe}(\text{CN})_6}, \forall z \in \mathbb{R}$$

```
\begin{equation*} % \usepackage{unicode-math}
\oint \mathscr{D}[x(t)] \sqrt{\frac{3 \pi^2 - \sum_{q=0}^{\infty} (z + \hat{L})^q}{\exp \left( \mathrm{i}^2 \ \hbar x \right)}}
\{ (\operatorname{Tr} \ \mathscr{A}) \left( \Lambda_{j_1 j_2}^{i_1 i_2} \Gamma_{i_1 i_2}^{j_1 j_2} \hookrightarrow \vec{D} \cdot \mathrm{P} \right) \} =
\underbrace{\widetilde{\left\langle \langle \frac{\notin \emptyset}{\varpi \alpha_{k \uparrow}} \middle| \frac{\partial_{\mu} T_{\mu\nu}}{2} \right\rangle}}_{\mathrm{K}_3\mathrm{Fe}(\mathrm{CN})_6}, \forall z \in \mathbb{R}
\end{equation*}
```



表格

- tabular 环境，一般包裹在 table 环境中变成浮动体 `\begin{tabular}{column spec}`
 - ▶ l / c / r: 左/中/右对齐
 - ▶ | : 竖线分隔; @{} : 去除列间距; @{\dots} : 自定义列间内容
 - ▶ {num}{col} : 重复 num 次 col 列格式
- 在表格内容中 & 分隔列, \\ 换行, \hline 画横线
- booktabs 宏包提供三线表式样
- 也可以合并单元格、拆分单元格等更复杂的操作
- 推荐使用
 - ▶ Tables Generator 网站
 - ▶ Tables Editor 网站
 - ▶ 表格真的太难写了



代码环境

- 行内代码使用 `\verb<delim>...<delim>`
 - ▶ 区分于正常的 `{...}`
 - ▶ `delim` 可以是除了星号 `*` 的任意字符
 - ▶ `\verb*` 命令表示显示空格
- 代码环境: `verbatim`
 - ▶ 默认使用等宽字体, 不解析 `LaTeX` 命令
 - ▶ 内容中的特殊字符都不需要转义
- `listings` 宏包提供代码高亮
- `minted` 功能更强大, 需要安装 Python 依赖, 本教程的代码环境都是 `minted`

```
\verb|\LaTeX ^_^|  
and  
\verb*`printf("Hello,world!\n");`
```

```
\LaTeX ^_^ and  
printf("Hello,world!\n");
```



代码环境

- 行内代码使用 `\verb<delim>...<delim>`
 - ▶ 区分于正常的 `{...}`
 - ▶ `delim` 可以是除了星号 `*` 的任意字符
 - ▶ `\verb*` 命令表示显示空格
- 代码环境: `verbatim`
 - ▶ 默认使用等宽字体, 不解析 `LaTeX` 命令
 - ▶ 内容中的特殊字符都不需要转义
- `listings` 宏包提供代码高亮
- `minted` 功能更强大, 需要安装 Python 依赖, 本教程的代码环境都是 `minted`

```
\verb|\LaTeX ^_^|  
and  
\verb*`printf("Hello,world!\n");`
```

```
\LaTeX ^_^ and  
printf("Hello,world!\n");
```



交叉引用

- \LaTeX 中使用 `\label` 标记，然后可以使用 `\ref` 来引用这个标记。`\label` 可以放在使用计数器的对象之后。
- 为了使得对公式编号的引用带有括号，推荐使用 AMSmath 宏包中的 `\eqref` 命令。对于多行公式环境，每一个换行符前都可以添加一个 `\label` 用于引用该行公式。

```
\begin{equation}
  a = b + c
\label{eq:example}
\end{equation}
\begin{figure}
  \includegraphics[width=3cm]{example-image-a}
  \caption{示例}\label{fig:example}
\end{figure}
如公式 \eqref{eq:example} 所示，
如图 \ref{fig:example} 所示
```

$$a = b + c \quad (4)$$



图: 示例

如公式 (4) 所示，如图II所示



交叉引用

- \LaTeX 中使用 `\label` 标记，然后可以使用 `\ref` 来引用这个标记。`\label` 可以放在使用计数器的对象之后。
- 为了使得对公式编号的引用带有括号，推荐使用 AMSmath 宏包中的 `\eqref` 命令。对于多行公式环境，每一个换行符前都可以添加一个 `\label` 用于引用该行公式。

$$a = b + c \quad (4)$$

```
\begin{equation}
  a = b + c
\label{eq:example}
\end{equation}
\begin{figure}
  \includegraphics[width=3cm]{example-image-a}
  \caption{示例}\label{fig:example}
\end{figure}
如公式 \eqref{eq:example} 所示，
如图\ref{fig:example}所示
```



图: 示例

如公式 (4) 所示，如图II所示



交叉引用

```
\begin{table} % \usepackage{booktabs}
\caption{人员名单}
\label{tab:example}
\centering
\begin{tabular}{cccccc}
\toprule
序号 & 姓名 & 性别 & 年龄 & 身高/cm & 体重/kg \\
\midrule
1 & 张三 & M & 16 & 163 & 50 \\
2 & 王红 & F & 15 & 159 & 47 \\
3 & 李二 & M & 17 & 165 & 52 \\
\bottomrule
\end{tabular}
\end{table}
```




表: 人员名单

序号	姓名	性别	年龄	身高/cm	体重/kg
1	张三	M	16	163	50
2	王红	F	15	159	47
3	李二	M	17	165	52

如表1 所示






文献管理

- 建议自动生成（你只有三篇参考文献？）
- `.bib` 数据库
 - ▶ Google Scholar 可直接复制：点击  -> BibTeX
 - ▶ 用 EndNote、Jabref 等生成
- 传统方法（大部分会议、期刊模板）：BibTeX 后端
 - ▶ 控制文献、引用样式：natbib 宏包
 - ▶ 国家标准 GB/T 7714-2015  : gbt7714 宏包
- 现代方法：biber 后端 + biblatex 宏包
 - ▶ 国家标准：biblatex-gb7714-2015 宏包
- 需多次编译
 - ▶ pdfL^AT_EX -> BibT_EX -> pdfL^AT_EX -> pdfL^AT_EX
 - ▶ XeL^AT_EX -> BibT_EX -> XeL^AT_EX -> XeL^AT_EX
 - ▶ 一键使用：VS Code plugin, MakeFile, Batch script, latexmk





文献管理

- 建议自动生成（你只有三篇参考文献？）
- `.bib` 数据库
 - ▶ Google Scholar 可直接复制：点击  -> BibTeX
 - ▶ 用 EndNote、Jabref 等生成
- 传统方法（大部分会议、期刊模板）：BibTeX 后端
 - ▶ 控制文献、引用样式：natbib 宏包
 - ▶ 国家标准 GB/T 7714-2015  : gbt7714 宏包
- 现代方法：biber 后端 + biblatex 宏包
 - ▶ 国家标准：biblatex-gb7714-2015 宏包
- 需多次编译
 - ▶ pdfL^AT_EX -> BibT_EX -> pdfL^AT_EX -> pdfL^AT_EX
 - ▶ XeL^AT_EX -> BibT_EX -> XeL^AT_EX -> XeL^AT_EX
 - ▶ 一键使用：VS Code plugin, MakeFile, Batch script, latexmk





文献管理

- 建议自动生成（你只有三篇参考文献？）
- |.bib| 数据库
 - ▶ Google Scholar 可直接复制：点击 **”** -> BibTeX
 - ▶ 用 EndNote、Jabref 等生成
- 传统方法（大部分会议、期刊模板）：BibTeX 后端
 - ▶ 控制文献、引用样式：natbib 宏包
 - ▶ 国家标准 GB/T 7714-2015  : gbt7714 宏包
- 现代方法：biber 后端 + biblatex 宏包
 - ▶ 国家标准：biblatex-gb7714-2015 宏包
- 需多次编译
 - ▶ pdfL^AT_EX -> BibT_EX -> pdfL^AT_EX -> pdfL^AT_EX
 - ▶ XeL^AT_EX -> BibT_EX -> XeL^AT_EX -> XeL^AT_EX
 - ▶ 一键使用：VS Code plugin, MakeFile, Batch script, latexmk





文献管理

- 建议自动生成（你只有三篇参考文献？）
- |.bib| 数据库
 - ▶ Google Scholar 可直接复制：点击 **”** -> BibTeX
 - ▶ 用 EndNote、Jabref 等生成
- 传统方法（大部分会议、期刊模板）：BibTeX 后端
 - ▶ 控制文献、引用样式：natbib 宏包
 - ▶ 国家标准 GB/T 7714-2015  : gbt7714 宏包
- 现代方法：biber 后端 + biblatex 宏包
 - ▶ 国家标准：biblatex-gb7714-2015 宏包
- 需多次编译
 - ▶ pdfL^AT_EX -> BibT_EX -> pdfL^AT_EX -> pdfL^AT_EX
 - ▶ XeL^AT_EX -> BibT_EX -> XeL^AT_EX -> XeL^AT_EX
 - ▶ 一键使用：VS Code plugin, MakeFile, Batch script, latexmk





文献管理

- 建议自动生成（你只有三篇参考文献？）
- |.bib| 数据库
 - ▶ Google Scholar 可直接复制：点击 **”** -> BibTeX
 - ▶ 用 EndNote、Jabref 等生成
- 传统方法（大部分会议、期刊模板）：BibTeX 后端
 - ▶ 控制文献、引用样式：natbib 宏包
 - ▶ 国家标准 GB/T 7714-2015  : gbt7714 宏包
- 现代方法：biber 后端 + biblatex 宏包
 - ▶ 国家标准：biblatex-gb7714-2015 宏包
- 需多次编译
 - ▶ pdfL^AT_EX -> BibT_EX -> pdfL^AT_EX -> pdfL^AT_EX
 - ▶ XeL^AT_EX -> BibT_EX -> XeL^AT_EX -> XeL^AT_EX
 - ▶ 一键使用：VS Code plugin, MakeFile, Batch script, latexmk



文献管理

- 建议自动生成（你只有三篇参考文献？）
- |.bib| 数据库
 - ▶ Google Scholar 可直接复制：点击 **”** -> BibTeX
 - ▶ 用 EndNote、Jabref 等生成
- 传统方法（大部分会议、期刊模板）：BibTeX 后端
 - ▶ 控制文献、引用样式：natbib 宏包
 - ▶ 国家标准 GB/T 7714-2015  : gbt7714 宏包
- 现代方法：biber 后端 + biblatex 宏包
 - ▶ 国家标准：biblatex-gb7714-2015 宏包
- 需多次编译
 - ▶ pdfL^AT_EX -> BibT_EX -> pdfL^AT_EX -> pdfL^AT_EX
 - ▶ XeL^AT_EX -> BibT_EX -> XeL^AT_EX -> XeL^AT_EX
 - ▶ 一键使用：VS Code plugin, MakeFile, Batch script, latexmk



文献引用样例

% In body.tex

“真理只有一个，而究竟谁发现了真理，不依靠主观的夸张，而依靠客观的实践。” -- 毛泽东\cite{毛泽东 1949 新民主主义论}。

% In references.bib

```
@book{毛泽东 1949 新民主主义论,  
  title={新民主主义论},  
  author={毛泽东},  
  year={1949},  
  publisher={长江出版社}  
}
```

“真理只有一个，而究竟谁发现了真理，不依靠主观的夸张，而依靠客观的实践。”— 毛泽东^[1]。

[1] 毛泽东. 新民主主义论[M]. 长江出版社, 1949.



中文支持

■ 中文有什么特殊？

- ▶ 汉字太多 (92,856+)
- ▶ 横排 + 直排、标点禁则、行间注 [🔗](#)

■ 已淘汰：

- ▶ CCT 系统、CJK 宏包（裸用）
- ▶ CT_EX 套装

■ 目前推荐手段：

- ▶ **ctex 宏集**（此 ctex 非彼 CT_EX）
- ▶ Xe_{La}T_EX 编译

■ 可以用，不推荐：

- ▶ xeCJK 宏包（裸用）
- ▶ ctex 宏集 + 其他引擎编译



中文支持

- 中文有什么特殊？
 - ▶ 汉字太多 (92,856+)
 - ▶ 横排 + 直排、标点禁则、行间注 [🔗](#)
- 已淘汰：
 - ▶ CCT 系统、CJK 宏包（裸用）
 - ▶ CT_EX 套装
- 目前推荐手段：
 - ▶ **ctex 宏集**（此 ctex 非彼 CT_EX）
 - ▶ Xe_ΛT_EX 编译
- 可以用，不推荐：
 - ▶ xeCJK 宏包（裸用）
 - ▶ ctex 宏集 + 其他引擎编译



中文支持

- 中文有什么特殊？
 - ▶ 汉字太多 (92,856+)
 - ▶ 横排 + 直排、标点禁则、行间注 [🔗](#)
- 已淘汰：
 - ▶ CCT 系统、CJK 宏包（裸用）
 - ▶ CT_EX 套装
- 目前推荐手段：
 - ▶ **ctex 宏集**（此 ctex 非彼 CT_EX）
 - ▶ Xe_ΛT_EX 编译
- 可以用，不推荐：
 - ▶ xeCJK 宏包（裸用）
 - ▶ ctex 宏集 + 其他引擎编译



中文支持

- 中文有什么特殊？
 - ▶ 汉字太多 (92,856+)
 - ▶ 横排 + 直排、标点禁则、行间注 [🔗](#)
- 已淘汰：
 - ▶ CCT 系统、CJK 宏包（裸用）
 - ▶ CT_EX 套装
- 目前推荐手段：
 - ▶ **ctex 宏集**（此 ctex 非彼 CT_EX）
 - ▶ Xe_ΛT_EX 编译
- 可以用，不推荐：
 - ▶ xeCJK 宏包（裸用）
 - ▶ ctex 宏集 + 其他引擎编译



中文支持

- 中文有什么特殊？
 - ▶ 汉字太多 (92,856+)
 - ▶ 横排 + 直排、标点禁则、行间注 [🔗](#)
- 已淘汰：
 - ▶ CCT 系统、CJK 宏包（裸用）
 - ▶ CT_EX 套装
- 目前推荐手段：
 - ▶ **ctex 宏集**（此 ctex 非彼 CT_EX）
 - ▶ Xe_ΛT_EX 编译
- 可以用，不推荐：
 - ▶ xeCJK 宏包（裸用）
 - ▶ ctex 宏集 + 其他引擎编译



中文支持

- 中文有什么特殊？
 - ▶ 汉字太多 (92,856+)
 - ▶ 横排 + 直排、标点禁则、行间注
- 已淘汰：
 - ▶ CCT 系统、CJK 宏包（裸用）
 - ▶ CT_EX 套装
- 目前推荐手段：
 - ▶ **ctex 宏集**（此 ctex 非彼 CT_EX）
 - ▶ Xe_{La}TeX 编译
- 可以用，不推荐：
 - ▶ xeCJK 宏包（裸用）
 - ▶ ctex 宏集 + 其他引擎编译



中文示例

- 编辑 `hello.tex` (Windows 下不要用中文文件名, 注意 \LaTeX 对文件名大小写敏感)

```
\documentclass{ctexart} % 使用中文适配的 article 文档类
\usepackage{xeCJK}% 如果要在一般的文档内使用中文, 一般只需引入此包
\begin{document}
\TeX{}你好!
\end{document}
```

- ▶ Windows 下缺省使用中易字体
- ▶ Linux、macOS 下需要注意字体 (参见 `ctex` 文档)

- 使用 $\text{Xe}\text{\LaTeX}$ 引擎编译, 得到 PDF 文档

\TeX 你好!



中文示例

- 编辑 `hello.tex` (Windows 下不要用中文文件名, 注意 \LaTeX 对文件名大小写敏感)

```
\documentclass{ctexart} % 使用中文适配的 article 文档类
\usepackage{xeCJK}% 如果要在一般的文档内使用中文, 一般只需引入此包
\begin{document}
\TeX{}你好!
\end{document}
```

- ▶ Windows 下缺省使用中易字体
- ▶ Linux、macOS 下需要注意字体 (参见 `ctex` 文档)
- 使用 $\text{Xe}\text{\LaTeX}$ 引擎编译, 得到 PDF 文档

\TeX 你好!



加载宏包

■ 「宏」包

- ▶ 提供扩展功能的组件
- ▶ 也就是别人造好的轮子
- ▶ 形式上为.sty 扩展名的纯文本文件

■ 怎么用

- ▶ `\usepackage{ctex}`
- ▶ 小心载入顺序

■ 哪里找？

- ▶ The Comprehensive T_EX Archive Network 
- ▶ GitHub



加载宏包

■ 「宏」包

- ▶ 提供扩展功能的组件
- ▶ 也就是别人造好的轮子
- ▶ 形式上为.sty 扩展名的纯文本文件

■ 怎么用

- ▶ `\usepackage{ctex}`
- ▶ 小心载入顺序

■ 哪里找？

- ▶ The Comprehensive T_EX Archive Network 
- ▶ GitHub
- ▶ 教程、博客、帖子（[维基问答](#)）



加载宏包


■ 「宏」包

- ▶ 提供扩展功能的组件
- ▶ 也就是别人造好的轮子
- ▶ 形式上为.sty 扩展名的纯文本文件

■ 怎么用


- ▶ `\usepackage{ctex}`
- ▶ 小心载入顺序

■ 哪里找？

- ▶ The Comprehensive T_EX Archive Network 
- ▶ GitHub
- ▶ 教程、博客、帖子（留意时效性）



加载宏包

- 「宏」包
 - ▶ 提供扩展功能的组件
 - ▶ 也就是别人造好的轮子
 - ▶ 形式上为.sty 扩展名的纯文本文件
- 怎么用
 - ▶ `\usepackage{ctex}`
 - ▶ 小心载入顺序
- 哪里找？
 - ▶ The Comprehensive T_EX Archive Network 
 - ▶ GitHub
 - ▶ 教程、博客、帖子（留意时效性）



宏包推荐（先读文档后使用）

- 国际单位制: `siunitx`
- 物理符号: `physics`
- 插图与绘图: `tikz`、`pgfplots`、`asymptote`
- 算法描述: `algorithm2e`、`algorithmicx`
- 更炫的框: `mdframed`、`tcolorbox`
- Unicode 数学支持: `unicode-math`
- 自定义章节标题格式: `titlesec`
- 更高级的表格: `tabularx`、`longtable`
- 字体: `fontspec`
- 化学式与化学方程式: `chemfig`、`mhchem`
- 乐谱: `musixtex`
- ...



幻灯片

■ 基本框架

- ▶ beamer 或 ctexbeamer 文档类
- ▶ 页面由 frame 环境组织
- ▶ 文本内容：建议使用 itemize 和 enumerate
- ▶ 图表：不再浮动，不建议使用交叉引用
- ▶ 定理及强调：theorem、proof、block 等
- ▶ 分栏：columns + columns 环境

■ 主题与样式

- ▶ `\usetheme、\lstinline[style=style@inline]+\use[font|color|inner|outer]theme+`
- ▶ 更现代的主题：metropolis
- ▶ 使用「默认」字体：`\usefonttheme{serif}`

■ 动画（覆盖）

- ▶ `\pause` 命令
- ▶ `\onslide<1>、\item<1->` 等



幻灯片

■ 基本框架

- ▶ beamer 或 ctexbeamer 文档类
- ▶ 页面由 frame 环境组织
- ▶ 文本内容：建议使用 itemize 和 enumerate
- ▶ 图表：不再浮动，不建议使用交叉引用
- ▶ 定理及强调：theorem、proof、block 等
- ▶ 分栏：columns + columns 环境

■ 主题与样式

- ▶ `\usetheme`、`\lstinline[style=style@inline]+\use[font|color|inner|outer]theme+`
- ▶ 更现代的主题：metropolis
- ▶ 使用「默认」字体：`\usefonttheme{serif}`

■ 动画（覆盖）

- ▶ `\pause`命令
- ▶ `\onslide<1>`、`\item<1->`等



幻灯片

■ 基本框架

- ▶ beamer 或 ctexbeamer 文档类
- ▶ 页面由 frame 环境组织
- ▶ 文本内容：建议使用 itemize 和 enumerate
- ▶ 图表：不再浮动，不建议使用交叉引用
- ▶ 定理及强调：theorem、proof、block 等
- ▶ 分栏：columns + columns 环境

■ 主题与样式

- ▶ `\usetheme`、`\lstinline[style=style@inline]+\use[font|color|inner|outer]theme+`
- ▶ 更现代的主题：metropolis
- ▶ 使用「默认」字体：`\usefonttheme{serif}`

■ 动画（覆盖）

- ▶ `\pause` 命令
- ▶ `\onslide<1>`、`\item<1->` 等



Git 版本管理

■ 版本管理的必要性

- ▶ 远离「初稿，第二稿……终稿，终稿（打死也不改了）」命名
- ▶ 有底气做大范围修改、重构
- ▶ 方便与他人协同合作

■ 基本用法

- ▶ 跟踪更改: `git init`、`git add` `git commit`
- ▶ 撤销与回滚: `git reset`、`git revert`
- ▶ 分支与高级用法: `git branch`、`git checkout` `git rebase`
- ▶ 远端仓库操作: `git pull`、`git push`、`git fetch`
- ▶ 推荐用 VS Code 等进行可视化操作
- ▶ 参考链接:

■ GitHub & more

- ▶ 远程 Git 仓库
- ▶ Clone & fork
- ▶ Issues & pull requests



Git 版本管理

■ 版本管理的必要性

- ▶ 远离「初稿，第二稿……终稿，终稿（打死也不改了）」命名
- ▶ 有底气做大范围修改、重构
- ▶ 方便与他人协同合作

■ 基本用法

- ▶ 跟踪更改: `git init`、`git add` `git commit`
- ▶ 撤销与回滚: `git reset`、`git revert`
- ▶ 分支与高级用法: `git branch`、`git checkout` `git rebase`
- ▶ 远端仓库操作: `git pull`、`git push`、`git fetch`
- ▶ 推荐用 VS Code 等进行可视化操作
- ▶ 参考链接: [🔗](#) [🔗](#)

■ GitHub & more

- ▶ 远程 Git 仓库
- ▶ Clone & fork
- ▶ Issues & pull requests



Git 版本管理

■ 版本管理的必要性

- ▶ 远离「初稿，第二稿……终稿，终稿（打死也不改了）」命名
- ▶ 有底气做大范围修改、重构
- ▶ 方便与他人协同合作

■ 基本用法

- ▶ 跟踪更改: `git init`、`git add` `git commit`
- ▶ 撤销与回滚: `git reset`、`git revert`
- ▶ 分支与高级用法: `git branch`、`git checkout` `git rebase`
- ▶ 远端仓库操作: `git pull`、`git push`、`git fetch`
- ▶ 推荐用 VS Code 等进行可视化操作
- ▶ 参考链接: [🔗](#) [🔗](#)

■ GitHub & more

- ▶ 远程 Git 仓库
- ▶ Clone & fork
- ▶ Issues & pull requests
- ▶ 提醒: 绑定 .edu 邮箱可以有更多优惠



Git 版本管理

■ 版本管理的必要性

- ▶ 远离「初稿，第二稿.....终稿，终稿（打死也不改了）」命名
- ▶ 有底气做大范围修改、重构
- ▶ 方便与他人协同合作

■ 基本用法

- ▶ 跟踪更改: `git init`、`git add` `git commit`
- ▶ 撤销与回滚: `git reset`、`git revert`
- ▶ 分支与高级用法: `git branch`、`git checkout` `git rebase`
- ▶ 远端仓库操作: `git pull`、`git push`、`git fetch`
- ▶ 推荐用 VS Code 等进行可视化操作
- ▶ 参考链接:

■ GitHub & more

- ▶ 远程 Git 仓库
- ▶ Clone & fork
- ▶ Issues & pull requests
- ▶ 提醒: 绑定 `.edu` 邮箱可以有更多优惠



模板



■ 是什么？

- ▶ 已经设计好的格式框架
- ▶ 好的模板：使用户专注于内容
- ▶ 不应将时间花费在调整框架上

■ 再提 Office 和 Word

- ▶ 很少有人会有意识地在 Word 中使用模板
- ▶ 定义自己的标题？定义自己的列表？定义自己的段落样式？
- ▶ 自动化，还是手工调？
- ▶ 经常被折腾的精疲力竭
- ▶ 学习 \LaTeX 能帮助自己更好科学地使用 Word

■ 有哪些？

- ▶ 期刊：revtex、elsarticle、IEEEtran、acmart.....
- ▶ 学位论文：thuthesis、fduthesis、SUEPThesis 
- ▶ Elegant \LaTeX  系列模板，比裸着用 \LaTeX 要好看很多



模板



■ 是什么？

- ▶ 已经设计好的格式框架
- ▶ 好的模板：使用户专注于内容
- ▶ 不应将时间花费在调整框架上

■ 再提 Office 和 Word

- ▶ 很少有人会有意识地在 Word 中使用模板
- ▶ 定义自己的标题？定义自己的列表？定义自己的段落样式？
- ▶ 自动化，还是手工调？
- ▶ 经常被折腾的精疲力竭
- ▶ 学习 \LaTeX 能帮助我们更好科学地使用 Word

■ 有哪些？

- ▶ 期刊：revtex、elsarticle、IEEEtran、acmart.....
- ▶ 学位论文：thuthesis、fduthesis、SUEPThesis 
- ▶ Elegant LaTeX  系列模板，比裸着用 \LaTeX 要好看很多



使用模板

- 模板就是预设好的文档类，一般直接放到文件夹，通过 `\documentclass` 使用
- 也是 \LaTeX 源代码，一般后缀名为 `.cls`
- GitHub 和 CTAN 上有大量的 \LaTeX 模板，可以搜索使用，还有期刊官网
- 例如 beamer 用于制作幻灯片
- 获取模板
 - ▶ 随发行版自带、手动官网下载
 - ▶ 模板文档类 `.cls` 文件
 - ▶ 示例 `.tex` 文件
- 通常模板都有怎么使用的文档，看文档，看文档，看文档



使用模板

- 模板就是预设好的文档类，一般直接放到文件夹，通过 `\documentclass` 使用
- 也是 \LaTeX 源代码，一般后缀名为 `.cls`
- GitHub 和 CTAN 上有大量的 \LaTeX 模板，可以搜索使用，还有期刊官网
- 例如 beamer 用于制作幻灯片
- 获取模板
 - ▶ 随发行版自带、手动官网下载
 - ▶ 模板文档类 `.cls` 文件
 - ▶ 示例 `.tex` 文件
- 通常模板都有怎么使用的文档，看文档，看文档，看文档



论文排版举例

IEEE 期刊论文

- 获取模板：已随发行版自带
 - ▶ 在安装目录 `<prefix>/texlive/2023/texmf-dist/doc/latex/IEEEtran` 下找到 `bare_jrn1.tex`
 - ▶ 复制到某个文件夹 (比如个人存论文的目录)
- 编辑 `bare_jrn1.tex` 文件 (英文模板：不支持中文)
- 编译
 - ▶ 英文文献：Xe \LaTeX 、pdf \LaTeX 编译均可



特别鸣谢

- 清华大学 `thu-latex-talk`
- 南方科技大学 `latex-talk`
- 南京大学 `latex-talk`
- 华东师范大学潘建瑜《 \LaTeX 科技排版入门》
- 浙江大学计算机学院朋辈辅学： \LaTeX 排版简要介绍



谢谢

本幻灯片下载地址 <https://github.com/SUEPaper/latex-talk>