

Marp 幻灯片指南

AI + VS Code 轻松制作幻灯片

连玉君

2026-03-24

Table of contents

封面	1
关于本书	2
你能学到什么?	2
怎么用这本书?	3
这本书是怎么写的?	3
发布自己的在线讲义	3
I 正文	4
1 第一份 Marp 幻灯片	5
1.1 Marp 是什么?	5
1.2 Marp 是 AI 时代的幻灯片工具	5
1.3 安装 VS Code	6
1.4 安装 Marp 插件	6
1.5 新建第一个.md 文件	7
1.6 输入极简示例, 看到第一张幻灯片	8
1.6.1 点击预览图标	8
1.7 关于 <code>marp: true</code>	9
1.8 用 <code>---</code> 分页	9
1.9 导出 PDF 和 HTML	10
1.9.1 最简单的导出方式	10
2 基础设定	12
2.1 选择主题: <code>theme</code>	12
2.2 设置尺寸: <code>size</code>	12
2.3 开启页码: <code>paginate</code>	13
2.3.1 跳过某一页的页码	13
2.3.2 自定义页码格式	13
2.4 页眉与页脚: <code>header / footer</code>	13
2.4.1 某一页单独设置	14
2.5 分页方式: <code>---</code> 与标题分页	14
2.5.1 手动分页 (<code>---</code>)	14

2.5.2	自动分页: <code>headingDivider</code>	14
2.6	一份适合课程讲义的基础模板	15
3	字号、图片和公式	17
3.1	调整字号	17
3.1.1	只调整某一页	17
3.1.2	常用字号参考	17
3.2	数学公式	18
3.3	插入图片	19
3.3.1	基本写法	19
3.3.2	控制图片尺寸	19
3.3.3	图片居中	19
3.3.4	设置为背景图	19
3.4	代码块	20
3.4.1	调整代码字号	20
3.5	双栏排版	20
3.5.1	不等宽双栏	21
3.5.2	左代码右图片	21
3.6	小字号注释	22
3.7	提示框 (Callout) 的简化实现	22
3.7.1	蓝色提示框 (Note)	22
3.7.2	黄色警告框 (Warning)	22
3.7.3	绿色成功框 (Tip)	23
4	图片管理与文件输出	24
4.1	推荐方案: 本地 <code>Fig/</code> 文件夹	24
4.1.1	建立项目文件夹结构	24
4.1.2	在 Marp 里用相对路径插图	24
4.1.3	在 VS Code 里打开整个文件夹	25
4.2	为什么 PDF 更适合本地图片	25
4.2.1	如何导出 PDF	25
4.3	HTML 输出时怎么处理图片	25
4.3.1	本地图片 + HTML = 不能只发一个文件	25
4.3.2	正确的 HTML 分享方式	26
4.4	图床简介: 什么时候才需要它	26
4.4.1	图床是什么?	26
4.4.2	常用图床	26
4.4.3	为什么图床不是初学者的默认方案?	26
4.5	本章决策小结	26
5	用 AI 生成幻灯片	28
5.1	为什么 AI 特别适合帮你做 Marp?	28
5.2	用 AI 直接生成完整幻灯片	28
5.2.1	标准流程-文字版, 随后删除	28

5.2.2	为什么要提供模板给 AI?	29
5.2.3	方法一：提供模板，按模板风格生成	29
5.2.4	方法二：让 AI 自己设计风格并生成	29
5.2.5	方法三：先生成论文精要，再转为幻灯片	30
5.3	如何描述你想要的效果	30
5.3.1	描述三要素	30
5.3.2	示例对比	31
5.4	让 AI 保留你现有的模板风格	31
5.5	让 AI 只做局部修改	31
5.6	让 AI 只返回可直接复制的代码	32
5.6.1	例子	32
5.7	把低频需求交给 AI 处理	32
5.8	实用提示词模板合集	32
5.9	一个重要习惯：把 AI 的输出当“草稿”	34
6	八个模板	35
6.1	模板一览	35
6.2	快速选择指南	36
6.3	marp01_simple: 基本示例	36
6.4	marp02_basic: 基础讲义模板	37
6.5	marp03_styled: 带样式的讲义模板	37
6.6	marp04_twocol: 双栏与代码展示模板	38
6.7	marp05_paper: 学术报告模板	39
6.8	marp06_data: 数据分析汇报模板	39
6.9	marp07_teaching: 课程教学模板	40
6.10	marp08_report: 政策报告 / 工作汇报模板	41
II	附录：各类提示词	43
A	调整文字效果	44
A-1	调整全局正文字号	44
A-2	调整标题字号	45
A-3	小字号注释	45
A-4	修改页码样式	46
A-5	修改页眉页脚内容与样式	47
B	分页方式	48
B-1	手动分页：---	48
B-2	用 ## 自动分页	48
B-3	用 ### 自动分页	49
B-4	临时阻止某个标题触发分页	50
C	插入图片	51
C-1	插入本地图片	51

C-2 调整图片尺寸	51
C-3 设置背景图	52
C-4 图文混排	53
C-5 图片不显示怎么办	54
D 代码块	55
D-1 基本代码块（带语法高亮）	55
D-2 调整代码字号	55
D-3 左代码右输出	56
D-4 左代码右图片	57
D-5 长代码怎么处理	58
E 版式-双栏排版和提示框	60
E-1 双栏布局	60
E-2 提示框 / Callout	61
E-3 小字备注区域	63
E-4 制作封面页	64
F 导出 HTML 和 PDF	66
F-1 导出 PDF	66
F-2 导出 HTML	66
F-3 HTML 分享时图片不见了	67
F-4 图床基本思路	68
G 常用提示词	69
G-1 生成基础模板	69
G-2 按已有模板生成完整幻灯片	69
G-3 修改已有模板（保留风格）	70
G-4 生成双栏布局页	70
G-5 调整字号和配色	70
G-6 处理图片排版页	71
G-7 论文精要转幻灯片	71
G-8 遇到报错时如何提问	72
G-9 其他常用提示词	73
I 基本 Markdown 语法	74
标题	74
强调	74
列表	74
链接与图片	75
代码	75
引用	75
分隔线	76
表格	76

数学公式	76
HTML 标签	76
扩展阅读	77

封面

[在线版](#) | [PDF](#) | [GitHub](#)



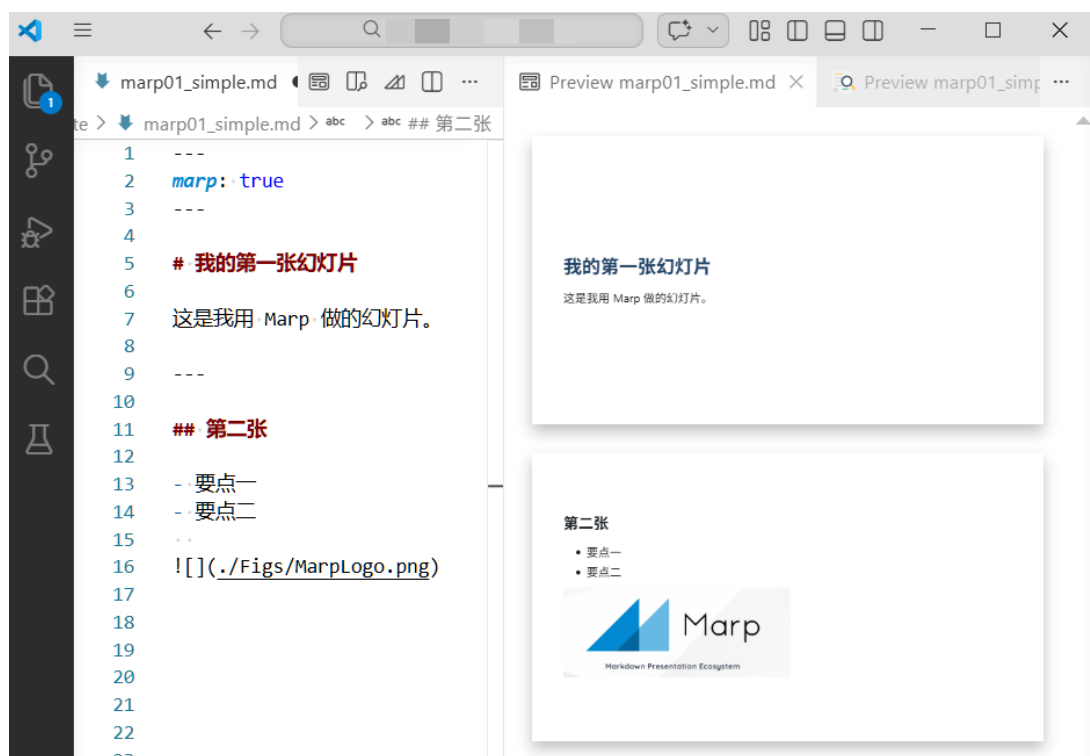
[主页](#) | [课程](#) | [视频](#) | [推文](#) | [资料](#)

关于本书

[在线版](#) | [PDF](#) | [GitHub](#)

大约 6 年前，我偶然发现了 Marp。从那以后，我几乎没有再碰过 PowerPoint。

用 Marp 是「写」幻灯片 (只需专注于内容即可)；
而用 PowerPoint 则是「做」幻灯片 (需要花大量的时间调整格式)。



随着 AI 的普及，Marp 的优势更加明显：你只需几句简单的提示词就可以把 PDF, Word, Markdown 等文档的内容转换成 Marp 幻灯片。由于 Marp 是采用 Markdown 标记语言来写的，所以无需任何格式方面的设定，直接将 AI 输出文本贴入 VS Code 的 Markdown 编辑器即可渲染为幻灯片。

你能学到什么？

这本书不是 Marp 技术手册。我不会把所有参数都列出来，也不会讲 CSS 主题开发的原理。相比于连享会此前发布的偏技术性的 [推文](#)，本书的目标在于：**让你尽快上手 Marp，并协助 AI 提示词实现你想要的各种效果。**

具体来说，读完这本书，你应该能够：

- 在 VS Code 里从零做出一份 Marp 幻灯片 [第 1 章](#)

- 掌握最常用的几个设定 [第 2 章](#)
- 知道怎么插图、排版代码、写数学公式 [第 3 章](#) 和 [第 4 章](#)
- 学会如何借助 AI 工具继续扩展和优化 [第 5 章](#)

最终，你会乐于使用 Marp 制作幻灯片。

怎么用这本书？

如果你完全没用过 Marp，请从 [第 1 章](#) 开始读，跟着操作一遍。大概 10 分钟，你就能看到自己做出的第一份幻灯片。

如果你已经会基本操作，想解决某个具体问题，可以直接跳到 [附录](#)，或者在左上角的搜索框中检索关键词。

这本书是怎么写的？

这本书是借助 [Claude code](#) 生成的，耗时约 10 小时。虽然整体上不错，但难免错漏。欢迎大家提出修改意见和建议。

你可以点击本书 [GitHub 仓库](#) 上方的 **Fork** 按钮，将本仓库复制到你的 GitHub 账号下，然后在你的仓库中进行修改和编辑，最后通过 **Pull Request** 向我提交修改建议。当然，你也可以发邮件给我：arlionn@163.com。

发布自己的在线讲义

你也可以修改本仓库的内容，形成符合你个人风格要求的书稿。具体步骤如下 (详情参见：[连玉君, 2025, Quarto Book](#)):

- **Fork** 在 [GitHub](#) 上 **Fork** 本仓库。
- **Clone** 在本地使用 [Github Desktop \(教程\)](#) 或者 `Git` 命令将 Fork 后的仓库 clone 到本地，然后在 VS Code 编辑器中修改内容。
- **Render** 在 VScode 终端执行 `quarto render` 命令生成新的网页内容。在此之前，你需要安装 [Quarto](#)，以及 VScode 的 [Quarto 扩展](#)。
- **Push** 使用 [Github Desktop](#) 或 `git` 命令将修改后的内容 push 到 GitHub 上。
- **Publish** 在 GitHub 上打开你的仓库，点击 `Settings` → `Pages`，在 `Source` 处选择 `main branch - docs`，然后点击 **Save**。
- 稍等几分钟后，你的在线讲义就会发布成功，网址为 <https://你的用户名.github.io/marp-book/>。

好了，我们开始吧。

[主页](#) | [课程](#) | [视频](#) | [推文](#) | [资料](#)

Part I

正文

Chapter 1

第一份 Marp 幻灯片

这一章只有一个目标：**让你看到自己做出的第一张幻灯片**。
我们不会讲太多原理，等你真的跑起来之后，再慢慢解释。

1.1 Marp 是什么？

Marp 是一个工具，让你用 [Markdown](#) 写幻灯片。

你只需要写普通的文字和简单的符号，它会自动帮你排成一张张幻灯片。和 PowerPoint 最大的不同是：你不需要用鼠标拖来拖去，只需要专注于内容本身。

一个最简单的 Marp 文件长这样：

```
---
marp: true
---

# 我的第一张幻灯片

这是第一张的内容。

---

# 第二张幻灯片

- 第一点
- 第二点
```

就这些。三行 YAML 头，加上几行 Markdown，就是一份幻灯片。

1.2 Marp 是 AI 时代的幻灯片工具

这是一个值得多说几句的问题。

PowerPoint 是“所见即所得”的工具——格式和内容是混在一起的：你需要一边想内容，一边操心字号、对齐、颜色、间距……每次改动都要动手调整，费时费力。

Marp 把这两件事彻底分开：**内容用 Markdown 写，样式用 YAML 和 CSS 控制**。这个特点，在 AI 时代变得格外有价值。

AI 可以直接帮你生成 Marp 幻灯片。因为 Marp 文件本质上是纯文本 (Markdown + YAML), AI 可以轻松读懂和生成。你可以:

- 把一篇论文或报告发给 AI, 让它**直接输出一份 Marp 幻灯片**
- 把一个模板发给 AI, 让它**按模板风格填充内容**
- 告诉 AI”这份幻灯片要讲给本科生听, 8 页以内”, AI 能直接给你适合的版本

相比之下, 用 AI 生成 PowerPoint 要困难得多——AI 不擅长控制 PPT 里的像素位置和图形布局, 而 Marp 的纯文本格式正是 AI 最擅长处理的。

一个典型的 AI 协作 workflow:

有一篇论文/报告/调研结果
↓
发给 AI, 附上一个 Marp 模板
↓
AI 输出完整的 .md 幻灯片文件
↓
在 VS Code 里预览、微调
↓
导出 PDF, 完成

这个 workflow 在第五章会详细展开。现在你只需要知道: **学会 Marp, 就是学会了和 AI 高效协作做幻灯片。**

i 不需要从零学所有内容

Marp 有很多参数和 CSS 细节, 但你不需要全部记住。只需要: 掌握本书前四章的基础内容, 有几个顺手的模板, 遇到新需求告诉 AI 来解决——这三件事加在一起, 就是这本书的全部目标。

1.3 安装 VS Code

VS Code 是我们用来编辑和预览 Marp 的编辑器。如果你还没有安装, 按下面的步骤来:

1. 打开浏览器, 访问 <https://code.visualstudio.com>
2. 点击页面上的蓝色下载按钮
3. 下载完成后双击安装包, 一路”下一步”即可

💡 已经有 VS Code 了?

直接跳到下一节安装插件就好。

1.4 安装 Marp 插件

VS Code 装好之后, 我们需要安装一个叫 **Marp for VS Code** 的插件。

步骤如下:

1. 打开 VS Code
2. 点击左侧竖排图标栏里的**扩展图标** (四个小方块组成的图标, 或按 `Ctrl+Shift+X` / `Cmd+Shift+X`)
3. 在搜索框里输入 Marp
4. 找到 **“Marp for VS Code”** (作者是 marp-team), 点击 **Install**

5. 等几秒钟，装好了
就这么简单。

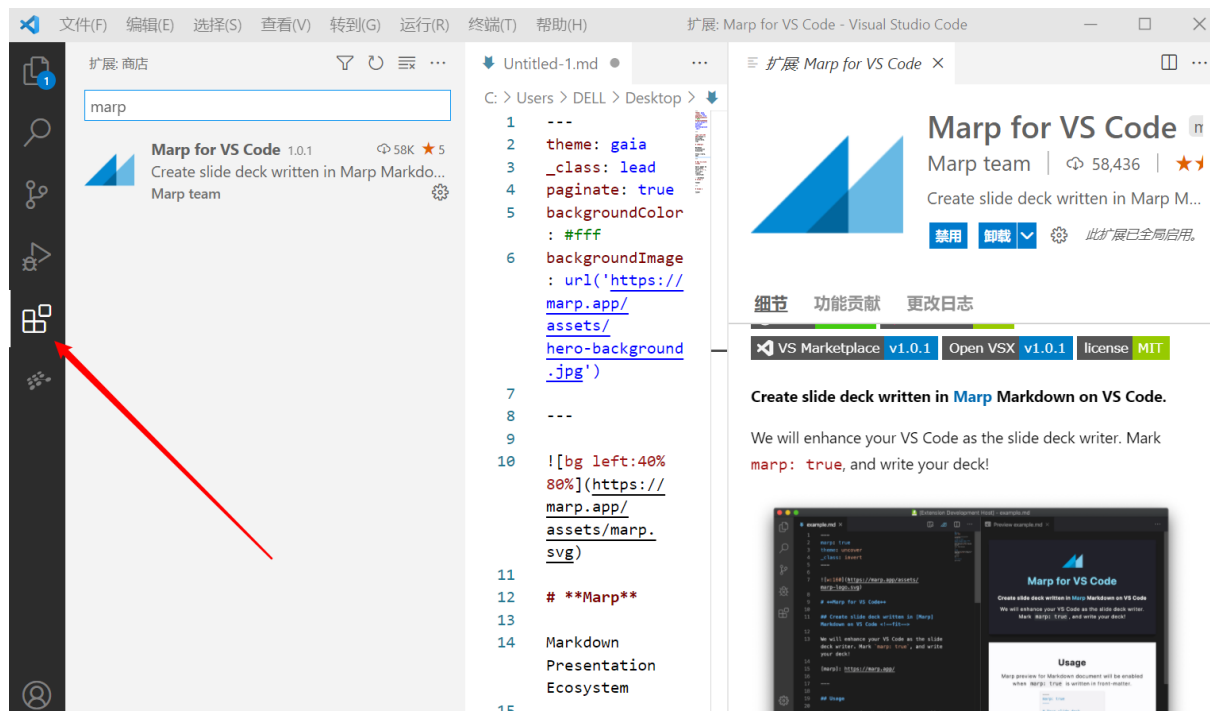


Figure 1.1: 在 VS Code 中安装 Marp 插件

i 插件安装好的标志

安装完成后，VS Code 顶部菜单栏会出现一个「两个三角形叠放」的图标。单击这个图标可以实现两个功能：

- 对于普通的 .md 文档，单击 Marp 图标 → Toggle Marp Feature……后，会自动在该文档顶部插入 YAML 信息：`marp: true`，使该文档变成一个 Marp Slides 文档。
- 对于已经包含 `marp: true` 的文档，点击 Export Slide Deck……可以将 Marp 文档导出为 PDF，HTML 等格式。

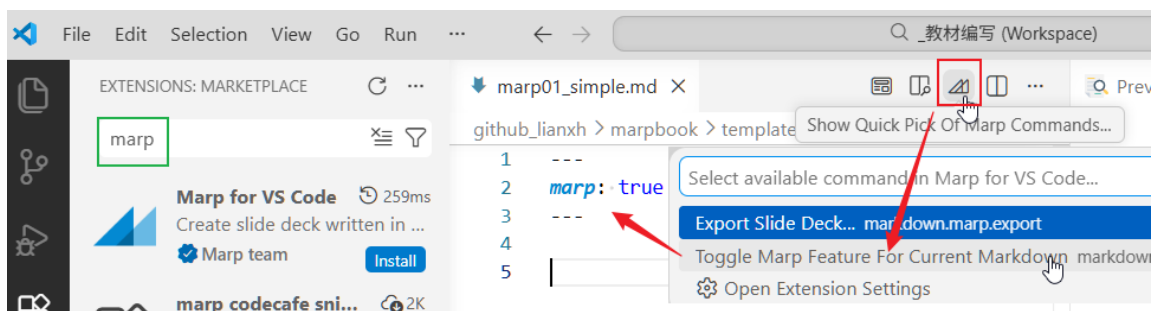


Figure 1.2: 开启 Marp 模式

1.5 新建第一个.md 文件

现在我们来新建一个文件：

1. 在 VS Code 里，点击菜单 **File** → **New File**（或按 **Ctrl+N** / **Cmd+N**）

2. 在弹出的对话框里，输入文件名 `slides.md`，选择一个你想保存的位置
3. 点击 Create

或者更简单：

1. 在桌面或某个文件夹里新建一个文本文件
2. 把文件扩展名改为 `.md`
3. 用 VS Code 打开它（右键 → 用 VS Code 打开）

1.6 输入极简示例，看到第一张幻灯片

文件打开后，把下面这段内容完整复制进去：

```
---
marp: true
---

# 我的第一张幻灯片

这是我用 Marp 做的幻灯片。

---

## 第二张

- 要点一
- 要点二


```

复制完，按 `Ctrl+S`（Mac 是 `Cmd+S`）保存文件。

1.6.1 点击预览图标

保存之后，注意 VS Code 编辑器右上角——应该出现了一个小图标，鼠标悬停上去会显示 `Open Preview to the Side` 或 `Marp: Open Preview` 之类的提示。

如果你的 VS Code 中还安装了 `Markdown Preview Enhanced` 或 `Markdown All in One` 插件，则点击预览图标后，VS Code 的右下角会显示如下信息，点击左侧按钮确认即可看到幻灯片的样子了。

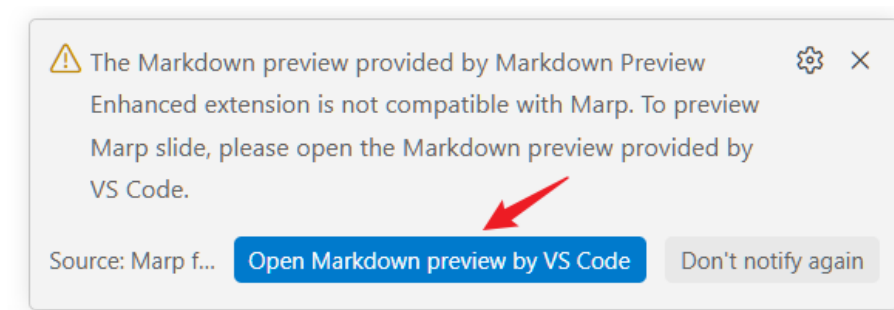


Figure 1.3: 切换至 Marp Slides 模式

💡 图标在哪里？

如果右上角没有看到 Marp 图标，可以试试：

- 确认文件是以 `.md` 为扩展名保存的
- 确认文件里第一行确实是 `---`，第二行是 `marp: true`
- 重新打开 VS Code，有时候插件需要重启才生效

恭喜！如果你看到了右边出现了幻灯片预览，你已经成功了。

1.7 关于 `marp: true`

文件开头的这三行是“YAML 头”，用来告诉 Marp 插件“这是一份 Marp 幻灯片”：

```
---  
marp: true  
---
```

这三行**必须放在文件的最开头**，一个字符都不能差。之后所有的 Marp 设定都会写在这里（在两个 `---` 之间）。

1.8 用 `---` 分页

在 Marp 里，用三个连字符 `---` 来分隔幻灯片页面：

```
---  
marp: true  
---  
  
# 第一页  
  
第一页的内容  
  
---  
  
# 第二页  
  
第二页的内容
```

注意：YAML 头里的 `---` 是用来“包住”设定的，不算分页。从 YAML 头结束之后，每次出现 `---` 就是新的一页。

最终的渲染效果如下：

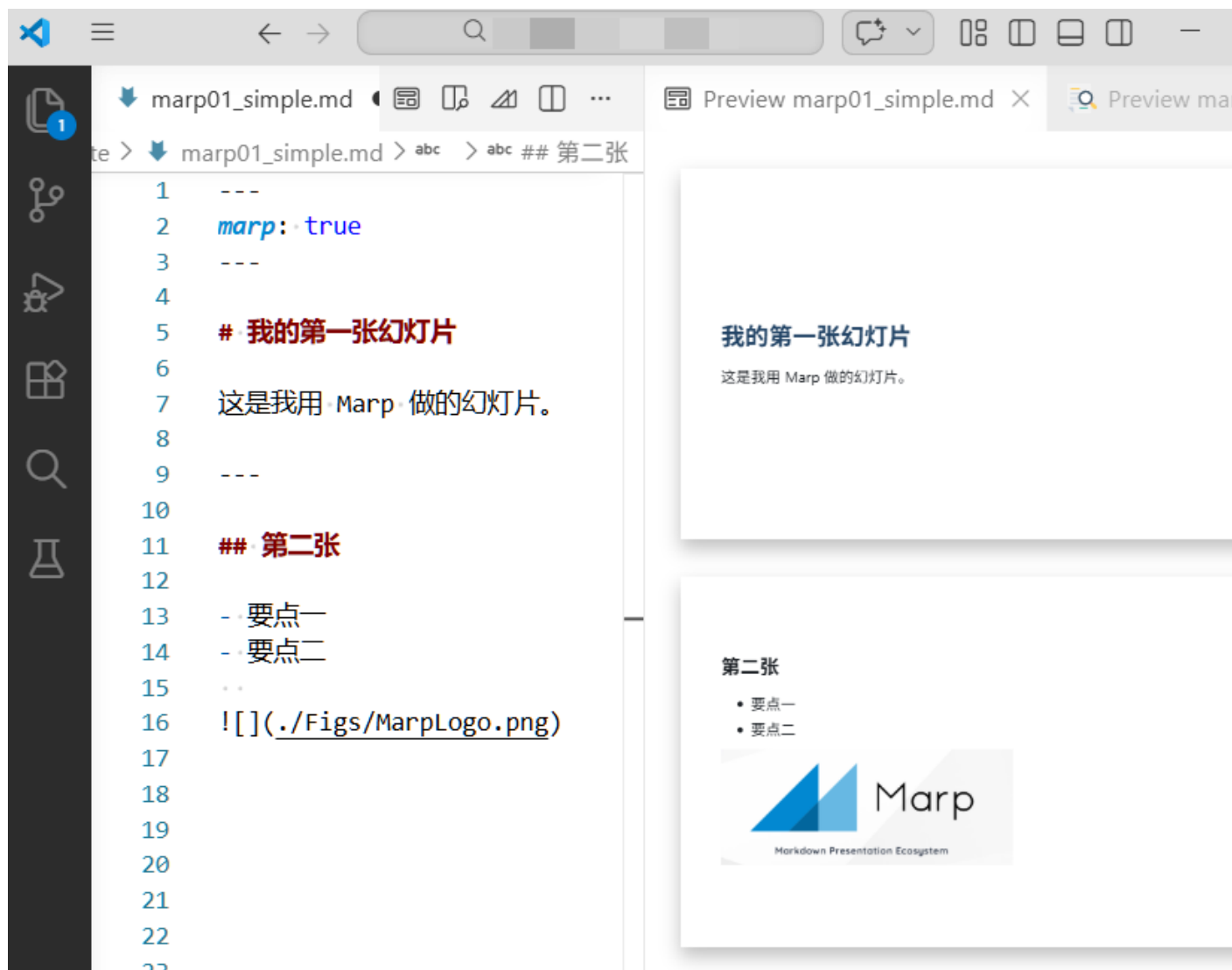


Figure 1.4: Marp 幻灯片预览

1.9 导出 PDF 和 HTML

预览满意了，就可以导出了。

1.9.1 最简单的导出方式

1. 在 VS Code 里打开你的 .md 文件
2. 按下 `Ctrl+Shift+P` (Mac 是 `Cmd+Shift+P`)，打开命令面板
3. 输入 `Marp`，你会看到几个选项：
 - `Marp: Export Slide Deck...` 一点这个
4. 在弹出的保存对话框里，选择保存格式：
 - 选 `.pdf` 保存为 PDF
 - 选 `.html` 保存为 HTML

PDF 是最推荐的格式，任何设备都能打开，图片也不会丢。

i 第一次导出 PDF 可能需要等一会儿

Marp 导出 PDF 时会在后台调用一个叫 Chromium 的浏览器内核，第一次使用时可能会自动下载，耐心等待一下就好。

本章小结

这一章你完成了：

- 安装 VS Code 和 Marp 插件
- 创建了第一个 .md 文件
- 用 `marp: true` 启用 Marp
- 用 `---` 分页
- 预览并导出了幻灯片

下一章，我们来让幻灯片”好看一点”——选主题、加页码、设页眉页脚。

Chapter 2

基础设定

上一章你已经能做出一份幻灯片了。这一章我们来让它“看起来更像一份正式的 Slides”。所有的基础设定，都写在 YAML 头里——也就是文件最开头两个 `---` 之间的区域。

2.1 选择主题：theme

Marp 内置了三个主题，你可以直接用：

主题名	风格
<code>default</code>	白底黑字，简洁干净
<code>gaia</code>	深色背景，现代感
<code>uncover</code>	浅色系，学术感

最小写法：

```
---  
marp: true  
theme: default  
---
```

把 `default` 换成 `gaia` 或 `uncover` 就能切换主题。建议先用 `default`，简单稳妥。

💡 推荐从 `default` 开始

初学阶段不用纠结主题，`default` 已经很好看了。等熟悉之后，再试试其他两个。

2.2 设置尺寸：size

默认尺寸是 **16:9**（宽屏）。如果你需要别的比例，可以这样设置：

```
---  
marp: true  
size: 16:9  
---
```

常用的选项：

值	用途
16:9	默认, 适合投影和屏幕展示
4:3	传统比例, 部分旧投影仪适用
A4	适合打印成讲义

大多数情况下不用改, 用默认的 16:9 就好。

2.3 开启页码: paginate

加一行就能显示页码:

```
---
marp: true
paginate: true
---
```

页码默认显示在右下角。

2.3.1 跳过某一页的页码

如果想让某一张 (比如封面) 不显示页码, 在那一页的最开头加上:

```
<!-- _paginate: false -->
```

下划线 `_` 表示”只对这一页生效”。

2.3.2 自定义页码格式

想显示”当前页 / 总页数”格式? 在 YAML 头里加:

```
---
marp: true
paginate: true
---
```

然后在某张幻灯片里用:

```
<!-- _paginate: "### %page / %total" -->
```

这会在那一页显示如”3 / 12”的格式。如果想全局生效, 可以写一个简单的 CSS (第三章会讲)。

2.4 页眉与页脚: header / footer

在 YAML 头里直接写就好:

```
---
marp: true
paginate: true
header: " 计量经济学导论"
footer: " 作者: 张三 | 2024 年秋"
---
```

- `header` 显示在每张幻灯片顶部
- `footer` 显示在每张幻灯片底部

2.4.1 某一页单独设置

如果某页需要不同的页眉（比如封面不要页眉），在那页开头加：

```
<!-- _header: "" -->
<!-- _footer: "" -->
```

空字符串表示“这一页不显示页眉/页脚”。

i 记住这个规律

（下划线）开头的设定 = 只对当前这一页生效。没有下划线 = 全局生效。
这个规律对很多 Marp 设定都适用。

2.5 分页方式：--- 与标题分页

目前你已经知道用 --- 手动分页。除此之外，Marp 还可以让你用标题自动分页。

2.5.1 手动分页（---）

最直接的方式：

```
# 第一页

内容

---

# 第二页

内容
```

2.5.2 自动分页：headingDivider

如果你的幻灯片每页都以一个标题开头，可以用 `headingDivider` 省掉手动写 ---：

```
---
marp: true
headingDivider: 2
---

## 第一页

内容

## 第二页

内容
```

设置 `headingDivider: 2` 后，每遇到 `##` 就会自动开始新的一页，不用手动加 ---。

值	效果
1	每个 # 都自动分页
2	每个 ## 自动分页
3	每个 ### 自动分页

值	效果
[1,2]	# 和 ## 都自动分页

💡 哪种更好?

- 内容不多、结构灵活 → 手动 ---
- 内容多、每页都有标题 → 用 `headingDivider: 2`

两种方式可以混用：设置了 `headingDivider` 之后，仍然可以在任意位置手动插入 ---。

2.6 一份适合课程讲义的基础模板

把上面学到的东西组合起来，这是一份可以直接复制使用的基础模板：

```
---
marp: true
theme: default
size: 16:9
paginate: true
header: " 课程名称"
footer: " 姓名 | 日期"
headingDivider: 2
---

<!-- _paginate: false -->
<!-- _header: "" -->
<!-- _footer: "" -->

# 课程标题

## 副标题或教师姓名

---

## 第一节：主题一

- 要点 1
- 要点 2
- 要点 3

## 第二节：主题二

正文内容写在这里。

可以是段落，也可以是列表。

## 小结

- 总结要点 1
- 总结要点 2
```

💡 直接复制使用

这个模板你可以直接复制，然后：

1. 把 课程名称、姓名、日期替换成你的信息
2. 把 课程标题和正文内容替换掉
3. 保存，预览，搞定

之后所有的幻灯片都可以基于这个模板修改。

本章小结

这一章你学会了：

- 用 `theme` 选择主题
- 用 `size` 设置幻灯片比例
- 用 `paginate` 开启页码
- 用 `header / footer` 设置页眉页脚
- 用 `---` 手动分页，或用 `headingDivider` 自动分页
- 获得了一份可以直接用的基础模板

下一章，我们来解决”如何在幻灯片里展示各种内容”——图片、公式、代码、双栏排版。

Chapter 3

字号、图片和公式

这一章解决”我知道能用 Marp，但不知道怎么放图/公式/代码”的问题。

我们不追求面面俱到，只讲最常用的几类。每种都有可以直接复制的最小示例。

3.1 调整字号

默认的字号有时候太大或太小，我们可以用 CSS 来调整。不用担心，你不需要懂 CSS——直接复制下面的代码就行。

在 Marp 文件里，可以在 YAML 头之后、第一张幻灯片之前加一个 `<style>` 块：

```
---
marp: true
---

<style>
section {
  font-size: 24px;
}
</style>

# 第一页

正文字号变成 24px 了
```

3.1.1 只调整某一页

如果只想改某一张幻灯片的字号，用 `scoped`：

```
<style scoped>
section {
  font-size: 20px;
}
</style>

# 这一页字号更小

其他页不受影响。
```

3.1.2 常用字号参考

用途	推荐字号
标准正文	24px 或 26px
内容较多时	20px 或 22px
标题 (h1)	40px 左右 (默认较大)
标题 (h2)	34px 左右

调整标题字号的写法:

```
<style>
h1 {
  font-size: 38px;
}
h2 {
  font-size: 30px;
}
section {
  font-size: 22px;
}
</style>
```

💡 字号不满意? 让 AI 帮你调

直接告诉 AI: “我的 Marp 幻灯片正文字号太大, 帮我把全局正文改成 22px, 标题 h2 改成 30px, 返回我需要加的 <style> 块。”

3.2 数学公式

Marp 支持 LaTeX 数学公式, 写法和 Markdown 里一样。

行内公式 (在文字中间):

质能方程为 $E = mc^2$, 这是物理学的基础。

独立公式块:

```
$$
\hat{\beta} = (X'X)^{-1}X'y
$$
```

多行公式:

```
$$
\begin{aligned}
y_i &= \alpha + \beta x_i + \varepsilon_i \\
\varepsilon_i &\sim N(0, \sigma^2)
\end{aligned}
$$
```

📄 公式不显示?

在 YAML 头里加上 `math: mathjax` 或 `math: katex`:

```
---  
marp: true  
math: katex  
---
```

KaTeX 渲染速度快，推荐用这个。

3.3 插入图片

3.3.1 基本写法

```
![图片描述](./Figs/MarpLogo.png)
```

这里 `./Figs/MarpLogo.png` 是图片的相对路径——意思是“在当前文件夹下的 `Figs` 文件夹里找 `MarpLogo.png`”。图片管理的具体方法在第四章会详细讲。

3.3.2 控制图片尺寸

```
![width:400px](./Figs/MarpLogo.png)
```

```
![width:60%](./Figs/MarpLogo.png)
```

```
![height:300px](./Figs/MarpLogo.png)
```

在方括号里加 `width:` 或 `height:` 就能控制大小。

i 注意

请不要把 `400px` 简写为 `400`，否则可能无法生效。

3.3.3 图片居中

Marp 的图片默认是左对齐的。要居中，可以用 HTML：

```
<div style="text-align: center;">  
![width:500px](./Figs/MarpLogo.png)  
</div>
```

或者更简单，用 CSS 全局设置：

```
<style>  
img {  
  display: block;  
  margin: 0 auto;  
}  
</style>
```

3.3.4 设置为背景图

```
![bg](./Figs/MarpLogo.png)
```

加上 `bg` 关键字，图片就会铺满整张幻灯片作为背景。

在普通页面中，你可以使用如下代码指定图片位于幻灯片右侧，占页面的 40% 篇幅；图片本身的宽度为 400 px。如此以来，左边 60% 的篇幅就可以插入说明文字或代码块。这是实现「图文双栏排版」的一个最简洁的方式。

```
![bg right:40% w:400px](./Figs/MarpLogo.png)

- text 1
- text 2
```

调整背景图透明度：

```
![bg opacity:0.3](./Figs/MarpLogo.png)
```

3.4 代码块

用三个反引号包住代码，并在开头注明语言，就能得到带语法高亮的代码块：

```
```python
import pandas as pd

df = pd.read_csv("data.csv")
print(df.head())
```
```

常用语言标识：`python`、`r`、`stata`、`bash`、`sql`、`javascript`、`markdown`

3.4.1 调整代码字号

代码默认字号可能偏小，可以这样调：

```
<style>
pre {
  font-size: 18px;
}
</style>
```

或者只调某一页：

```
<style scoped>
pre {
  font-size: 16px;
}
</style>
```

3.5 双栏排版

双栏是讲义里非常常用的布局，比如“左边代码，右边结果”，或者“左边文字，右边图”。

Marp 用 CSS Grid 实现双栏。直接复制下面的代码：

```
<style scoped>
.cols {
  display: grid;
  grid-template-columns: 1fr 1fr;
  gap: 1em;
}
```

```
}
</style>

<div class="cols">
<div>

** 左栏内容 **

- 要点一
- 要点二

</div>
<div>

** 右栏内容 **

- 要点三
- 要点四

</div>
</div>
```

! 注意空行

<div> 标签和 Markdown 内容之间必须有空行，否则 Markdown 不会被正确解析。

3.5.1 不等宽双栏

比如左边占 40%，右边占 60%：

```
<style scoped>
.cols {
  display: grid;
  grid-template-columns: 2fr 3fr;
  gap: 1em;
}
</style>
```

改 grid-template-columns 里的数字比例就行。

3.5.2 左代码右图片

```
<style scoped>
.cols { display: grid; grid-template-columns: 1fr 1fr; gap: 1em; }
</style>

<div class="cols">
<div>

```python
import matplotlib.pyplot as plt
x = [1, 2, 3, 4]
y = [2, 4, 6, 8]
plt.plot(x, y)
plt.show()
```
```

```

</div>
<div>

![width:100%](../Figs/plot.png)

</div>
</div>

```

3.6 小字号注释

幻灯片底部常常需要放一些小字备注，比如数据来源、参考文献。

```

<div style="font-size: 14px; color: #888; margin-top: auto; padding-top: 1em; border-top: 1px
solid #ddd;">

注：数据来源于《中国统计年鉴》(2023)，单位为亿元。

</div>

```

或者更简单，用一个小 CSS class：

```

<style>
.note {
  font-size: 14px;
  color: #888;
  margin-top: 1em;
  border-top: 1px solid #eee;
  padding-top: 0.5em;
}
</style>

正文内容……

<div class="note">注：数据来源……</div>

```

3.7 提示框 (Callout) 的简化实现

Marp 没有内置 callout，但用一点 CSS 就能模仿出来。下面给你几个可以直接复制的样式：

3.7.1 蓝色提示框 (Note)

```

<div style="background:#e8f4fd; border-left:4px solid #2196F3; padding:0.8em 1em;
border-radius:4px; margin:0.5em 0;">

** 提示： ** 这里是提示内容，告诉读者一个有用的信息。

</div>

```

3.7.2 黄色警告框 (Warning)

```
<div style="background:#fff8e1; border-left:4px solid #FFC107; padding:0.8em 1em; border-radius:4px; margin:0.5em 0;">
```

**** 注意: **** 这里是需要注意的内容。

```
</div>
```

3.7.3 绿色成功框 (Tip)

```
<div style="background:#e8f5e9; border-left:4px solid #4CAF50; padding:0.8em 1em; border-radius:4px; margin:0.5em 0;">
```

**** 做法: **** 这里是推荐做法。

```
</div>
```

💡 嫌麻烦? 让 AI 生成

告诉 AI: “给我一个 Marp 用的蓝色提示框 HTML 块, 内容是'.....', 保持简洁, 直接返回可复制的代码。”

AI 会帮你生成, 你只需要改文字内容就好。

本章小结

这一章你学会了:

- 用 `<style>` 调整字号
- 写行内公式和独立公式块
- 插入图片并控制尺寸
- 写带语法高亮的代码块
- 用 CSS Grid 实现双栏排版
- 加小字注释和提示框

这些内容覆盖了日常讲义的绝大多数需求。记不住没关系, 用的时候来这里复制就行。

下一章, 我们重点解决” 图片管理” 和” 导出输出” 这两个新手最容易卡住的问题。

Chapter 4

图片管理与文件输出

这一章解决新手最容易卡住的两个问题：

1. 图片怎么管理，才不会到处找不到、输出后又消失？
2. PDF 和 HTML 有什么区别，我该输出哪种？

答案其实很简单。先给你结论：

初学者默认推荐：本地图片 + Fig/ 文件夹 + 输出 PDF。

下面来解释为什么，以及怎么做。

4.1 推荐方案：本地 Fig/ 文件夹

4.1.1 建立项目文件夹结构

每次开始一个新项目（比如一门课的幻灯片），建议建这样一个文件夹结构：

```
my_slides/  
  slides.md      ← 你的 Marp 文件  
  Fig/          ← 所有图片放这里  
    chart1.png  
    diagram.png  
    photo.jpg
```

步骤：

1. 在你喜欢的位置新建一个文件夹，比如叫 `my_slides`
2. 在这个文件夹里新建 `slides.md`
3. 在这个文件夹里新建一个子文件夹叫 `Fig`
4. 把所有要用的图片都放进 `Fig/`

就这些。非常简单。

4.1.2 在 Marp 里用相对路径插图

在 `slides.md` 里，插图时用**相对路径**：

```
![图片说明](./Fig/chart1.png)  
  
![width:60%](./Fig/diagram.png)
```

`./Fig/chart1.png` 的意思是：“从 `slides.md` 所在的文件夹出发，进入 `Fig` 子文件夹，找 `chart1.png`。”

💡 为什么叫“相对路径”？

因为路径是相对于你的 `.md` 文件位置来写的，不是写死的完整路径（如 `C:\Users\zhang\Desktop\...`）。

相对路径的好处是：整个项目文件夹移动到哪里都没事，图片都能找到。

4.1.3 在 VS Code 里打开整个文件夹

建议用 VS Code 的”打开文件夹”而不是”打开文件”：

- 菜单 **File** → **Open Folder**，选择 `my_slides` 这个文件夹
- 左侧的文件资源管理器里可以看到所有文件
- 这样也方便你管理图片和文件

4.2 为什么 PDF 更适合本地图片

输出 PDF 时，Marp 会把图片直接嵌入到 PDF 文件里。这意味着：

- 你只需要发送一个 `.pdf` 文件
- 对方打开时，图片都在里面，不会丢
- 任何设备都能打开，不依赖插件或网络

PDF 是本地图片方案的最佳搭档。

4.2.1 如何导出 PDF

1. 在 VS Code 里打开 `slides.md`
2. 按 `Ctrl+Shift+P` (Mac: `Cmd+Shift+P`)
3. 输入 `Marp: Export`，选择 **Marp: Export Slide Deck...**
4. 在保存对话框里，文件类型选 `.pdf`，点保存
5. 等几秒，PDF 就好了

i PDF 导出很慢？

正常的。Marp 导出 PDF 用的是 Chromium (Chrome 浏览器内核)，第一次可能需要下载，之后就快了。

如果等了超过 2 分钟还没反应，可以试试重新打开 VS Code 再导出。

4.3 HTML 输出时怎么处理图片

HTML 输出比 PDF 稍微复杂一点，需要特别注意。

4.3.1 本地图片 + HTML = 不能只发一个文件

当你用本地图片、输出 HTML 时，图片不会被嵌入到 HTML 文件里。HTML 文件只是记录了”图片在 `./Fig/chart.png` 这个位置”，但图片本身还是单独存在的。

这意味着：如果你只发出 `slides.html` 这个文件，对方打开时图片会消失。

4.3.2 正确的 HTML 分享方式

如果一定要分享 HTML，需要把整个项目文件夹一起发出去：

```
my_slides/      ← 把这整个文件夹压缩发出去
  slides.html   ← 导出的 HTML 文件
  Fig/
    chart1.png
    diagram.png
```

对方解压后，在自己电脑上打开 slides.html，图片就能正常显示。

! 小结

- 分享给别人看 → 用 PDF，发单个文件，最省心
- 需要在网页上展示（比如放到网站上）→ 再考虑 HTML，或者考虑图床

4.4 图床简介：什么时候才需要它

如果你经常需要把 HTML 版幻灯片分享到网上，或者你的图片很多、每次都要带着文件夹很麻烦，这时候可以考虑图床。

4.4.1 图床是什么？

图床是一种在线图片存储服务。你把图片上传上去，它给你一个链接，你在 Marp 里直接用这个链接：

```
![图片描述](https://example.com/img/chart1.png)
```

这样，图片就存在网上了。不管你发出去的是 PDF 还是 HTML，图片都能正常显示（只要有网络）。

4.4.2 常用图床

服务	特点
GitHub + jsDelivr	免费、稳定，适合技术用户
阿里云 OSS	付费但稳定，适合长期使用
微博/微信图床	不推荐，链接可能失效
SM.MS	免费，小图可用

4.4.3 为什么图床不是初学者的默认方案？

- 需要额外注册账号、学习上传流程
- 图片链接可能失效（特别是免费服务）
- 本地图片 + PDF 已经能解决 90% 的需求

等你熟悉了 Marp 基本操作，有在线分享的持续需求，再考虑图床。

4.5 本章决策小结

我要分享幻灯片

直接发文件给别人看

用 PDF ← 推荐，最简单

放到网页上 / 在线展示
不怕麻烦，带着图片文件夹一起
用 HTML + Fig/ 文件夹
希望只有一个链接
考虑图床 + HTML

本章小结

这一章你学会了：

- 用 slides.md + Fig/ 文件夹的结构管理本地图片
- 用相对路径 ./Fig/xxx.png 插图
- 明白了为什么 PDF 是本地图片的最佳搭档
- 明白了 HTML 输出时图片的注意事项
- 了解了图床的基本概念和适用场景

记住：初学阶段，本地图片 + PDF 输出，最省心。

下一章，我们来聊这本书最重要的话题：如何借助 AI 让自己持续进步。

Chapter 5

用 AI 生成幻灯片

这是全书最重要的一章。

前四章介绍了 Marp 的基本操作。本章的目标是：教大家写提示词，以便借助 AI（如豆包、DeepSeek、ChatGPT、Claude 等），轻松实现 Marp 的一些高级功能：自定义主题、复杂动画、响应式布局等。

5.1 为什么 AI 特别适合帮你做 Marp？

Marp 幻灯片由三类内容组成：

1. **Markdown 文字内容**（AI 最擅长）
2. **YAML 头设定**（格式固定，AI 很少出错）
3. **CSS 样式**（AI 对 CSS 熟悉度很高）

这三样加在一起，正好是 AI 最擅长的领域。你只需要描述你想要的效果，AI 基本上都能给出可用的代码。

5.2 用 AI 直接生成完整幻灯片

这是 Marp + AI 组合最强大的用法：把一篇论文、报告或调研结果交给 AI，让它直接输出一份完整的 Marp 幻灯片。

5.2.1 标准流程-文字版，随后删除

从论文到幻灯片，四步走，通常 15 分钟以内可以完成。

完整流程

第一步：把材料发给 AI

将论文全文、报告或需要制作成幻灯片的文档，粘贴给 AI 工具（豆包、DeepSeek、ChatGPT、Claude 等均可）。

第二步：先生成大纲

我要基于上述内容生成 Marp 幻灯片。

听众：{博士研究生 / 本科生 / 政府工作人员}

篇幅：30 页左右；正文字号：24pt

请先生成一个幻灯片大纲，不要直接输出正文。

第三步：确认大纲后生成正文

检查大纲，按需调整章节结构和详略，然后输入：

大纲确认，请输出完整的 Marp 幻灯片 Markdown 文本。

格式要求参照：<https://lianxhcn.github.io/marp-book/>

模板风格参照 marp05_paper，模板内容如下：

[粘贴模板的 YAML 头和 <style> 块]

第四步：导入 VS Code，导出幻灯片

将 AI 生成的 Markdown 内容粘贴进 VS Code，保存为 .md 文件，点击 Marp 图标预览，满意后导出为 PDF 或 HTML。

i 为什么要先出大纲再出正文？

直接让 AI 一步生成完整幻灯片，它往往会在每页塞太多内容，或者结构与你的预期不符，反复修改反而费时。

先定结构，再填内容，是和 AI 合作效率最高的方式。大纲阶段修改成本极低，正文阶段修改成本高。

5.2.2 为什么要提供模板给 AI？

AI 在生成幻灯片时，最难把握的是**每一页放多少内容**。如果不提供字号信息，AI 可能给每页塞太多文字，实际渲染时溢出幻灯片边界。

把 **template** 文件夹里某个模板的 CSS 块发给 AI，它就能知道”这个字号下，一页大概能放多少行”，从而合理分页。这是生成质量最高的一个技巧。

5.2.3 方法一：提供模板，按模板风格生成

💡 提示词

我有一篇论文/报告，内容如下：

[粘贴论文摘要、引言或完整正文]

请根据以上内容，生成一份完整的 Marp 幻灯片 (.md 格式)。

风格参考：请参照以下模板的 YAML 头和 CSS 样式，保持相同的字号、配色和版式：

[粘贴模板文件的 YAML 头和 <style> 块]

其他要求：

- 面向受众：[例如：经济学专业研究生]
- 幻灯片总页数：[例如：12-15 页]
- 包含封面页、目录页、各节内容页、结论页
- 每页正文要点控制在 4-5 条以内，不要太满
- 只返回完整的 .md 文件内容，用 markdown 代码块包住

5.2.4 方法二：让 AI 自己设计风格并生成

如果你还没有现成模板，可以直接描述你要的样式：

💡 提示词

我有一份报告，内容如下：

[粘贴内容]

请帮我生成一份完整的 Marp 幻灯片，要求：

样式设定：

- 主题：default，尺寸：16:9

- 正文字号：22px，h2 标题：28px，颜色：深蓝 #1565c0
- 封面页：深蓝渐变背景，白色文字

内容要求：

- 面向受众：[听众描述]
- 总页数：[X] 页左右
- 包含封面、目录、内容页、结论页
- 每页文字量适中，重要数据用加粗突出

只返回完整的 .md 文件，用 markdown 代码块包住，不加任何解释。

5.2.5 方法三：先生成论文精要，再转为幻灯片

如果原始材料是很长的论文，可以分两步走，质量更稳定。

第一步：生成论文精要

💡 提示词（第一步：生成精要）

请阅读以下论文，用中文写一份 800–1000 字的精要，要求：

- 包含研究背景、研究问题、数据与方法、核心发现、政策含义
- 突出关键数据和因果结论
- 如需插入文中的图片或表格截图，请预留位置，并写好说明文字，我随后手动截图后插入图片。
- 语言简洁，适合转化为学术汇报幻灯片

[粘贴论文全文或摘要 + 引言 + 结论]

第二步：精要转幻灯片

💡 提示词（第二步：精要转幻灯片）

以下是一篇论文的精要：

[粘贴第一步生成的精要]

请将其转化为一份 Marp 幻灯片（10–12 页），要求：

- 参考以下模板风格：[粘贴模板的 YAML 头和 `<style>` 块]
- 面向受众：经济学领域同行
- 结构：封面、引言、数据/方法、核心结果（2–3 页）、结论
- 结果页保留关键系数和显著性标注（*** ** *）
- 只返回 .md 文件，用代码块包住，不加解释

5.3 如何描述你想要的效果

AI 不是魔法，它需要你**说清楚你想要什么**。

5.3.1 描述三要素

一个好的 Marp 相关提示词，通常包含三个部分：

1. **目标**：我想要什么效果？
2. **约束**：不要改哪些东西？要保留什么？
3. **输出格式**：只返回代码，还是带解释？

5.3.2 示例对比

不好的提示词：

```
> 帮我美化一下幻灯片
```

这太模糊了，AI 不知道从哪里下手。

好的提示词：

我有一份 Marp 幻灯片，主题是 default，YAML 头如下：

```
```yaml

marp: true
theme: default
paginate: true

```
```

请帮我在 YAML 头后面加一个全局 `<style>` 块，把正文字号改为 22px，h2 标题改为 28px。不要修改其他设定，只返回需要插入的 `<style>` 块代码。

这个提示词的好处：

- 说明了当前状态（default 主题，有 paginate）
- 说清楚了目标（22px 正文，28px h2）
- 说清楚了约束（不改其他设定）
- 说清楚了输出格式（只返回 `<style>` 块）

5.4 让 AI 保留你现有的模板风格

你已经有一套用得顺手的模板，想在上面加新功能，但不想让 AI 把整个模板改掉——怎么办？

关键词：把现有代码贴给 AI，明确说“只做 X，不改其他”。

💡 提示词

我有一份 Marp 幻灯片模板，代码如下：

[把你整个 .md 文件的 YAML 头和 `<style>` 块贴在这里]

请在这个模板基础上，帮我添加 [具体需求，例如：双栏布局的 CSS class]。

要求：

1. 保留现有的所有 YAML 设定不变
2. 保留现有的 `<style>` 内容不变
3. 只添加新的 CSS class，放在 `<style>` 块的末尾
4. 只返回修改后的完整 YAML 头和 `<style>` 块，不需要解释

5.5 让 AI 只做局部修改

有时候你只想改一张幻灯片的效果，不想影响其他页——可以这样要求 AI：

💡 提示词

我有一张 Marp 幻灯片，内容如下：

[把那一张幻灯片的内容贴在这里]

请帮我把这张幻灯片改成双栏布局：左边放文字要点，右边放图片（图片路径：`./Figs/chart.png`）。

要求：

1. 只改这一张幻灯片，不要改 YAML 头或全局样式
2. 用 `<style scoped>` 而不是全局 `<style>`
3. 直接返回可以替换这张幻灯片的完整代码

5.6 让 AI 只返回可直接复制的代码

AI 有时候会给你一大段解释，但你只需要代码。你只需在提示词末尾加上下面这类表述（酌情选择一个即可）：

- “只返回代码，不需要任何解释”
- “用代码块包住输出，不要在代码外面加任何文字”
- “直接给我可以复制粘贴进 VS Code 的内容”

5.6.1 例子

💡 提示词

帮我生成一个 Marp 幻灯片页面，内容是一个三栏对比表，比较 OLS、IV、DID 三种方法的适用场景。

要求：用 CSS Grid 实现三栏，每栏有标题和 3-4 个要点。

只返回这一张幻灯片的完整代码（从 `---` 分页符开始到下一个 `---`），用 markdown 代码块包住，不加任何解释。

5.7 把低频需求交给 AI 处理

有些功能你可能只用一两次，不值得花时间系统学。遇到这类需求，直接扔给 AI 就好。

典型的”低频需求”：

- 做一个花哨的封面页
- 给某张幻灯片加渐变背景色
- 做一个进度条（显示当前是第几章）
- 把一张密密麻麻的表格变得好看一点
- 给幻灯片加上自定义字体

这些东西你可能半年用一次，记住它们没什么意义。用的时候描述给 AI，AI 给你代码，复制进去，搞定。

5.8 实用提示词模板合集

下面是一组可以直接使用的提示词模板，覆盖了最常见的场景。把方括号里的内容替换成你的实际情况就可以用。

💡 提示词：生成基础 Marp 模板

请帮我生成一份 Marp 幻灯片的基础模板。

要求：

- 主题：default，尺寸：16:9
- 开启页码，格式为”当前页 / 总页”
- 页眉显示”[课程名称]“，页脚显示”[姓名] | [年份]”
- 用 headingDivider: 2 自动分页
- 全局字号 24px，h2 标题 30px
- 包含封面页（不显示页码和页眉页脚）
- 包含 2-3 张示例内容页

只返回完整的 .md 文件内容，用 markdown 代码块包住。

💡 提示词：修改已有模板（保留风格）

我有一份 Marp 模板，YAML 头和样式如下：

[把你的 YAML 头和 <style> 块粘贴在这里]

请在这个基础上，帮我 [具体修改内容，例如：把页脚的颜色改为深灰色 #555，把正文字号从 24px 改为 22px]。

要求：

- 只修改我要求的部分，其他内容完全保留
- 返回修改后的完整 YAML 头和 <style> 块

💡 提示词：生成双栏内容页

请帮我生成一张 Marp 幻灯片，内容如下：

- 标题：[幻灯片标题]
- 左栏：[左栏内容描述，例如：DID 方法的三个假设，用要点列表]
- 右栏：[右栏内容描述，例如：一段配图说明，图片路径./Figs/did.png]

要求：

- 用 CSS Grid 双栏布局，左右各 50%
- 使用 <style scoped> 避免影响其他页
- 直接返回这一张幻灯片的完整代码（从 --- 开始到下一个 ---），不加解释

💡 提示词：调整字号和配色

我的 Marp 幻灯片目前使用 default 主题，现有 <style> 块如下：

[贴上你现有的 <style> 块，或写"暂无"]

请帮我生成一个 <style> 块，实现以下效果：

- 正文字号：22px
- h1 字号：38px，颜色：#1a1a2e
- h2 字号：30px，颜色：#16213e
- 链接颜色：#0066cc
- 代码块背景：#f5f5f5，字号：18px

只返回 <style> 块的完整代码。

💡 提示词：处理图片排版页

请帮我生成一张 Marp 幻灯片，要求：

- 标题：[页面标题]
- 主体：一张图片居中显示，宽度为幻灯片宽度的 70%，图片路径：./Figs/[图片名]
- 图片下方有一行小字注释（字号 14px，颜色 #888）：[注释内容]

直接返回这张幻灯片的完整代码，不加解释。

💡 提示词：遇到问题时求助 AI

我的 Marp 幻灯片出现了问题：[描述问题，例如：图片不显示 / 双栏布局某栏内容跑到外面去了 / 导出 PDF 时代码块被截断]

相关代码如下：

[把出问题的那张幻灯片的代码贴在这里]

我用的是 VS Code + Marp for VS Code 插件。

请帮我找出原因，并给出修正后的代码。只返回修正后的代码块，不需要详细解释。

5.9 一个重要习惯：把 AI 的输出当“草稿”

AI 生成的代码通常是对的，但不一定 100% 符合你的需求。

建议养成这个习惯：

1. 让 AI 生成代码
2. 复制进 VS Code，**立刻预览**
3. 发现问题 → 回到 AI，告诉它“这个效果不对，具体是……，请调整”
4. 重复 2-3 次，通常就能得到满意的结果

不要一次性让 AI 做太多。**小步迭代**，比一次要求完美更有效。

本章小结

这一章你掌握了：

- 用 AI 直接把论文/报告转化为完整 Marp 幻灯片的标准流程和三种方法
- 为什么先出大纲、再出正文，效率更高
- 为什么提供模板给 AI 能显著提升生成质量
- 如何写出有效的提示词（目标 + 约束 + 输出格式）
- 如何让 AI 保留现有模板风格、只做局部修改
- 6 个可直接使用的提示词模板

最后一句话：这本书教你入门，AI 陪你进阶。把模板给 AI，把需求说清楚，大部分工作 AI 都能替你完成。

Chapter 6

八个模板

与其从空白文件开始，不如直接复制一个模板、改改内容就能用。

Github 仓库的 **【template】** 文件夹里提供了 8 个由浅入深的模板，覆盖了从入门练习到学术报告的常见场景。找到最接近你需求的那个，复制过去，改文字，就完成了。

i 查看和下载模板文件

方法 1: 下载单个模板

以 `marp01_simple.md` 为例，你可以通过如下步骤查看或下载该模板文件：

- **查看**：点击 [raw](#) → 全选 → 复制 → 粘贴到新建的 `.md` 文档中。
- **下载**：访问 [template/marp01_simple.md](#) 页面 → 点击右上角的下载按钮



方法 2: 手动批量下载

- 访问 [Github 仓库: marp-book](#)，点击绿色 `<>Code` 按钮，选择 **Download ZIP** 下载本仓库。
- 解压缩后，在 VS Code 编辑器中，依次点击 **File** → **Open folder** (`Ctrl+K / Ctrl+O`)，打开 `[marp-book]` 文件夹，即可在 `[template]` 子文件夹中查看模板文件。

方法 2: 使用 github desktop 下载

如果你熟悉 git，可以在 VScode 终端使用 git 命令 Fork 或 clone 本仓库。对于大多数人来说，推荐使用 [GitHub Desktop](#)，详情参见：[GitHub Desktop 使用方法介绍: 可视化 Git 管理的效率工具](#)。

i 模板只是起点

所有模板都可以用 AI 继续修改和扩展。遇到想要的效果，把模板代码贴给 AI，描述需求，AI 会帮你调整。具体提示词见 [第五章](#)。

6.1 模板一览

| 文件名 | 难度 | 适合场景 |
|-------------------------------|----|---------------|
| <code>marp01_simple.md</code> | 入门 | 第一次练习，验证环境 |
| <code>marp02_basic.md</code> | 基础 | 日常课程讲义、组会汇报 |
| <code>marp03_styled.md</code> | 基础 | 需要公式、提示框、小字注释 |

| 文件名 | 难度 | 适合场景 |
|--------------------|----|-------------|
| marp04_twocol.md | 进阶 | 代码演示、双栏对比 |
| marp05_paper.md | 进阶 | 学术论文汇报、会议展示 |
| marp06_data.md | 进阶 | 数据分析报告、图表展示 |
| marp07_teaching.md | 进阶 | 课程讲授、逐步推导 |
| marp08_report.md | 进阶 | 政策报告、工作汇报 |

6.2 快速选择指南

我是第一次用 Marp

marp01_simple ← 先跑起来

我要做一份普通课程讲义

marp02_basic ← 够用了

我需要公式或彩色提示框

marp03_styled

我要展示代码和运行结果

marp04_twocol

我要做学术报告或论文汇报

marp05_paper

我要做数据分析报告（图表、KPI、回归结果）

marp06_data

我要做课程教学（板书、逐步推导、对比框）

marp07_teaching

我要做政策报告或工作汇报

marp08_report

6.3 marp01_simple: 基本示例

难度:

适合: 第一次使用 Marp, 验证环境是否安装正确

这是一份只有三张幻灯片的基本文件, 没有任何额外设定, 只用了 `marp: true` 启用 Marp, 以及 `---` 手动分页。

包含的内容:

- 一级标题页 (#)
- 带要点列表和本地图片的页面
- 一张空白页

💡 图片路径说明

marp01_simple.md 里引用了 `./Figs/MarpLogo.png`。你需要在同一文件夹下建 `Figs/` 子文件夹并放入图片，或者直接把图片行删掉。

6.4 marp02_basic: 基础讲义模板

难度:

适合: 日常课程讲义、组会汇报、部门分享

在 marp01 的基础上，加入了最常用的几个全局设定：主题、页码、页眉页脚，以及一个不显示页眉页脚的封面页。

包含的内容:

- YAML 头: `theme`、`size`、`paginate`、`header`、`footer`
- 封面页 (无页眉、页脚、页码)
- 3 张示例内容页

使用步骤:

1. 复制文件，修改 YAML 头里的 `header` 和 `footer` 内容
2. 把封面页的标题和姓名改掉
3. 按需增减内容页，用 `---` 分页

可以直接修改的地方:

```
header: " 课程名称"      ← 改成你的课程/项目名
footer: " 姓名 · 日期"   ← 改成你的信息
```

6.5 marp03_styled: 带样式的讲义模板

难度:

适合: 需要数学公式、彩色提示框、底部注释的讲义

在 marp02 的基础上，增加了一个 `<style>` 块，定义了字号、标题颜色、提示框 class 和小字注释 class。

新增功能:

- 自定义字号 (正文 22px, h1/h2 带颜色)
- 4 种提示框: `.box-note` (蓝)、`.box-warn` (黄)、`.box-tip` (绿)
- 底部注释 class: `.note`
- KaTeX 数学公式支持

使用提示框:

```
<div class="box-note">

** 提示: ** 文字内容写在这里。

</div>
```

使用底部注释:

```
<div class="note">注: 数据来源于……</div>
```

i 关于 <div> 和空行

<div> 标签和里面的 Markdown 内容之间**必须留空行**，否则 Markdown 不会被正确解析（会显示成原始文字）。

6.6 marp04_twocol: 双栏与代码展示模板

难度:

适合: 代码教学、方法对比、数据可视化展示

在 marp03 的基础上，重点增加了双栏布局相关的 CSS class，并提供了三种典型页面的示例：左文右图、左代码右结果、三栏方法对比。

新增 CSS class:

Class	用途
.cols	等宽双栏 (50:50)
.cols-6040	左宽右窄 (60:40)
.cols-4060	左窄右宽 (40:60)

左代码右结果的写法:

```
<div class="cols">
<div>

```python
左边写代码
```

</div>
<div>

```text
右边写输出
```

</div>
</div>
```

三栏对比: 模板里提供了一个带卡片样式的三栏页面 (OLS / IV / DID 对比)，可以直接修改内容复用。

💡 图片尺寸

双栏里的图片建议用 HTML 写法，确保宽度 100% 填满栏位:

```

```

6.7 marp05_paper: 学术报告模板

难度:

适合: 论文汇报、学术会议展示、课题答辩

这是功能最完整的模板，内置了深蓝渐变封面、节标题过渡页、双栏数据表格、回归结果展示等学术报告常用版式。

特色功能:

封面页（深蓝渐变背景）

```
<!-- _class: cover -->
<!-- _paginate: false -->

# 论文标题
## A Subtitle in English
作者 · 单位 · 年份
```

节标题过渡页（浅蓝背景，居中大字）

```
<!-- _class: section-title -->
<!-- _paginate: false -->

## 01 · 引言
```

回归结果页（左表右文，含发现框和注释）

模板里已经写好了一个完整的“基准回归结果”页，包括回归表格、核心发现提示框、底部注释。直接替换数字即可。

致谢结尾页

复用封面的深蓝样式，简洁大方。

使用建议:

- 先改封面（标题、作者、单位）
- 按需增减“节标题页 → 内容页”的组合
- 把数据表格里的数字替换成你的结果
- 不需要的功能直接删掉就行

💡 觉得太复杂？

可以先用 `marp03_styled` 或 `marp04_twocol`，等熟悉了再升级到 `marp05_paper`。或者，把 `marp05_paper.md` 直接给 AI 看，告诉它“保留这个风格，帮我把内容替换成关于.....的论文汇报”，AI 会帮你完成大部分工作。

6.8 marp06_data: 数据分析汇报模板

难度:

适合: 数据分析报告、实证结果展示、研究汇报

以图表和数据呈现为核心，内置 KPI 卡片、双图对比、回归结果 + 解读双栏等版式，适合有大量数据和图表要展示的场所。

特色功能：

KPI 数字卡片（三栏，带涨跌色）

```
<div class="kpi-row">
<div class="kpi-card">
<div class="num">3,842</div>
<div class="label">样本量</div>
<div class="delta">↑ 12.3% 较上期</div>
</div>
<!-- 再加两个 kpi-card ... -->
</div>
```

左图右文（图 60%，文字 40%）

使用 `.chart-cols` class，右侧配合 `.finding`（绿色发现框）和 `.insight`（黄色洞察框）排版解读文字。

双图并排（各 50%）

使用 `.chart-cols-half` class，两张图并排，各配独立图注。

左表右结论

回归结果表放左边，右边用 `.finding` 和 `.insight` 框写核心发现，逻辑清晰。

💡 图片占位

模板里的图片路径（如 `./Figs/trend.png`）是占位符，替换成你的实际图片路径即可。如果图片还没准备好，可以先用 `<div style="background:#e3f2fd;height:200px;border-radius:6px;"></div>` 做占位块。

6.9 marp07_teaching: 课程教学模板

难度：

适合：课程讲授、知识点讲解、逐步推导演示

以“教学流程”为核心，设计了板书风格的代码展示区、带编号的步骤列表、左错右对的对比框，以及高亮定义框——适合需要一步步引导学生思考的课堂讲授场景。

特色功能：

步骤列表（带状态：已完成 / 当前 / 待讲）

```
<div class="step-row">
  <div class="step-num done"></div> <!-- 绿色：已讲 -->
  <div>内容已讲</div>
</div>
<div class="step-row">
  <div class="step-num now">→</div> <!-- 橙色：当前 -->
  <div>本节重点内容</div>
</div>
<div class="step-row">
  <div class="step-num">4</div> <!-- 蓝色：待讲 -->
```

```
<div>后续内容</div>
</div>
```

板书区域（深色背景，等宽字体）

```
<div class="board">
<span class="comment"># 注释文字（灰色斜体）</span>
<br>
关键变量: <span class="hl">高亮内容（青色）</span>
<br>
结论: <span class="result">结果高亮（黄色）</span>
</div>
```

左错右对对比框

```
<div class="compare">
<div class="wrong"> <strong>错误做法: </strong>说明</div>
<div class="right"> <strong>正确做法: </strong>说明</div>
</div>
```

定义框（蓝色边框，带标签）

```
<div class="def">
<div class="label"> 定义: 工具变量</div>
正式定义写在这里……
</div>
```

i 关于“逐步揭示”

Marp 不支持 PowerPoint 那种点击动画逐步显示内容。这个模板用**步骤列表的状态样式**（已完成 / 当前 / 待讲）来模拟“当前讲到哪里”的视觉效果，配合手动 --- 分页来实现逐步推进的感觉。

6.10 marp08_report: 政策报告 / 工作汇报模板

难度:

适合: 政策研究报告、政府汇报、部门工作汇报、咨询报告

采用绿色系商务配色，封面设计为左右分栏（左侧深色信息栏 + 右侧标题区），整体风格简洁、正式、有说服力。

特色功能:

左右分栏封面（左侧深绿信息栏）

封面不使用全页背景图，而是左右分区，左侧放单位/日期/标签，右侧放标题和汇报人信息，层次清晰。

要点列表增强（绿点 + 分隔线）

```
<ul class="bullet-list">
<li>要点一: 说明文字</li>
<li>要点二: 说明文字</li>
</ul>
```

带标签的要点（彩色小标签）

```
<span class="label-tag">重点</span> 说明文字
```

引用框（左边绿色竖线，斜体文字）

```
<div class="quote">
" 政策引用或权威表述……"
<br><span style="font-size:13px;">——文件名称, 年份</span>
</div>
```

数字高亮（绿色加粗数字）

```
核心指标: <span class="highlight-num">12.3%</span>
```

建议卡片网格（2×2，带顶部绿色边框）

```
<div class="rec-grid">
<div class="rec-card">
<h3>建议一: 标题</h3>
<p>具体措施说明。</p>
</div>
<!-- 再加三个 rec-card -->
</div>
```

Part II

附录：各类提示词

A 调整文字效果

这里收录了所有与文字相关的常见需求。每个条目的结构是：我想实现什么 → 基本写法 → 推荐写法 → 常见问题 → 提示词。

A-1 调整全局正文字号

我想实现什么： 让整份幻灯片的正文文字更大或更小。

基本写法：

```
<style>
section {
  font-size: 22px;
}
</style>
```

推荐写法： 把 `<style>` 块放在 YAML 头之后、第一张幻灯片之前：

```
---
marp: true
theme: default
---

<style>
section {
  font-size: 22px;
}
</style>

# 第一张幻灯片

正文字号现在是 22px。
```

常见问题：

- 改了没有反应 → 检查 `<style>` 是否在 YAML 头外面（不是在 `---` 里面）
- 想只改某一张 → 用 `<style scoped>` 代替 `<style>`

💡 提示词

帮我在 Marp 幻灯片中把全局正文字号改为 22px，h1 改为 38px，h2 改为 30px。当前没有 `<style>` 块。只返回需要加入的 `<style>` 块代码。

A-2 调整标题字号

我想实现什么：单独调整 # (h1) 或 ## (h2) 的字号，而不影响正文。

基本写法：

```
<style>
h1 { font-size: 40px; }
h2 { font-size: 32px; }
</style>
```

推荐写法：通常和正文字号一起设定：

```
<style>
section { font-size: 24px; }
h1 { font-size: 40px; }
h2 { font-size: 32px; }
h3 { font-size: 26px; }
</style>
```

常见问题：

- h1 和 h2 颜色也想改 → 在同一条规则里加 color: #333;
- 默认主题的 h1 很大，如果不主动设置，gaia 和 uncover 主题会各有默认值

💡 提示词

我的 Marp 幻灯片用 default 主题，请帮我生成一个 <style> 块，把 h1 字号设为 38px 颜色 #1a1a2e，h2 字号设为 30px 颜色 #333，正文 24px。只返回 <style> 块。

A-3 小字号注释

我想实现什么：在幻灯片底部放一行小字（数据来源、参考文献等）。

基本写法：

```
<small>注：数据来源……</small>
```

推荐写法：加边框线和颜色，更清晰：

```
<div style="font-size:14px; color:#888; border-top:1px solid #ddd; margin-top:auto; padding-top:0.5em;">
```

注：数据来源于《中国统计年鉴》(2023 年)。

```
</div>
```

如果经常用，定义一个 class 更方便：

先在全局 <style> 里加：

```
<style>
.note {
  font-size: 14px;
  color: #888;
  border-top: 1px solid #ddd;
  margin-top: auto;
}
```

```
padding-top: 0.5em;
}
</style>
```

然后每次这样用：

```
<div class="note">注：数据来源……</div>
```

常见问题：

- 注释不在底部 → 给外层 section 加 `display: flex; flex-direction: column;`，然后用 `margin-top: auto` 把注释推到底部
- 字太小看不清 → 改为 16px

💡 提示词

帮我在 Marp 幻灯片里实现一个底部小字注释样式，要求：字号 14px，颜色灰色，上方有一条细分隔线，注释始终贴在幻灯片底部。请给我完整的 CSS class 定义和使用示例。

A-4 修改页码样式

我想实现什么：改变页码的位置、字号或格式（比如显示“3 / 12”）。

基本写法（开启页码）：

```
---
marp: true
paginate: true
---
```

显示“当前页 / 总页数”格式：

在某一张幻灯片开头加（对那一页生效）：

```
<!-- _paginate: "### %page / %total" -->
```

用 CSS 修改页码位置和字号：

```
<style>
section::after {
  font-size: 16px;
  color: #666;
  content: attr(data-marpit-pagination) ' / ' attr(data-marpit-pagination-total);
}
</style>
```

隐藏某一页的页码：

```
<!-- _paginate: false -->
```

💡 提示词

我的 Marp 幻灯片用 `paginate: true`，请帮我用 CSS 把页码改为“当前页 / 总页数”格式，字号 14px，颜色 #999，显示在右下角。只返回 `<style>` 块。

A-5 修改页眉页脚内容与样式

我想实现什么： 设置页眉页脚，或者改变它们的字体大小和颜色。

基本写法：

```
---
marp: true
header: " 标题"
footer: " 姓名 | 日期"
---
```

修改样式 (字号、颜色)：

```
<style>
header {
  font-size: 16px;
  color: #888;
}
footer {
  font-size: 14px;
  color: #aaa;
}
</style>
```

某一页单独关闭：

```
<!-- _header: "" -->
<!-- _footer: "" -->
```

某一页单独修改内容：

```
<!-- _header: " 第二章：方法论" -->
```

提示词

我的 Marp 幻灯片设置了 header 和 footer，请帮我用 CSS 把页眉字号改为 16px 颜色 #666，页脚字号 13px 颜色 #aaa，并在页脚左侧显示内容。只返回 <style> 块。

B 分页方式

B-1 手动分页：---

我想实现什么：在任意位置手动开始一张新幻灯片。

基本写法：

```
# 第一张幻灯片  
  
内容在这里  
  
---  
  
# 第二张幻灯片  
  
内容在这里
```

注意：YAML 头里的 --- 是用来包住设定的，不算分页。第一个分页符是 YAML 头结束后出现的 ---。

推荐写法：在 --- 前后各留一个空行，避免渲染错误：

```
正文内容  
  
---  
  
# 下一页标题
```

常见问题：

- --- 没有分页 → 检查 --- 前后是否有空行
- 不小心把 YAML 头里的 --- 当成分页用了 → YAML 头的 --- 要成对出现，包住设定，不是分页

B-2 用 ## 自动分页

我想实现什么：每遇到二级标题（##）就自动开始新的一页，不用手动写 ---。

基本写法：

```
---  
marp: true  
headingDivider: 2
```

```

---
## 第一节
内容
## 第二节
内容 (自动分页)

```

推荐写法：通常配合封面页一起用。封面页用 # (一级标题)，不会被 headingDivider: 2 分页：

```

---
marp: true
headingDivider: 2
---

# 课程标题

## 副标题

---

## 第一章：引论

内容……

## 第二章：方法

内容……

```

常见问题：

- 某一页不想被自动分页 → 在那个标题前面手动加 `<!-- headingDivider: 0 -->` (局部覆盖)
- 想让 # 和 ## 都自动分页 → 用 `headingDivider: [1, 2]`

提示词：

我用了 `headingDivider: 2`，但有一个 ## 开头的内容不想分页（想和上一页合并）。请告诉我怎么处理，并给出示例。

B-3 用 ### 自动分页

我想实现什么：让三级标题也自动分页。

基本写法：

```

---
marp: true
headingDivider: 3
---

```

设为 3 时，#、##、### 遇到都会自动分页。

如果只想让 ## 和 ### 分页，不想让 # 分页：

```

---
marp: true
headingDivider: [2, 3]
---

```

常见问题：

- 内容层级太多、分页太碎 → 降回 `headingDivider: 2`，或者改用手动 `---`
- 想在某个 `###` 前不分页 → 在那行前加 `<!-- headingDivider: 0 -->`

B-4 临时阻止某个标题触发分页

我想实现什么：我用了 `headingDivider: 2`，但某个 `##` 标题我不想让它分页（想和上面的内容在同一页）。

基本写法：

```

## 正常分页的标题

内容

<!-- headingDivider: 0 -->

## 这个标题不触发分页

这段内容会和上面在同一页。

```

在不想分页的 `##` 标题前，加一行 `<!-- headingDivider: 0 -->`，它会覆盖全局设定，让这个标题不触发分页。

推荐写法：如果只有个别页需要特殊处理，手动 `---` 和 `headingDivider` 混用是最灵活的：

```

---
marp: true
headingDivider: 2
---

## 正常分页标题

内容

---

这是手动分页，不用标题

继续写内容……

## 下一个自动分页标题

```

提示词：

我用了 `headingDivider: 2`，有一张幻灯片需要在同一页里显示两个 `##` 标题，请告诉我怎么实现并给出代码示例。

C 插入图片

C-1 插入本地图片

我想实现什么：把我电脑上的图片放进幻灯片。

简洁写法：

```
![图片说明](./Fig/chart.png)
```

推荐写法：先建好文件夹结构，再用相对路径：

```
my_slides/  
  slides.md  
  Fig/  
    chart.png
```

```
![回归系数图](./Fig/chart.png)
```

常见问题：

- 图片不显示 → 检查路径是否正确，文件名区分大小写 (Chart.png chart.png)
- 移动文件夹后图片消失 → 用相对路径就不会有这个问题，绝对路径会

更多图片问题见 [C-5](#)。

C-2 调整图片尺寸

我想实现什么：让图片变大或变小，不要占满整页。

简洁写法（按宽度）：

```
![width:400px](./Fig/chart.png)
```

按百分比：

```
![width:60%](./Fig/chart.png)
```

按高度：

```
![height:300px](./Fig/chart.png)
```

同时限制宽高（不拉伸）：

```
![width:400px height:300px](./Fig/chart.png)
```

推荐写法：一般用 width 的百分比，这样在不同尺寸幻灯片上都能保持比例：

```
![width:70%](./Fig/chart.png)
```

图片居中：

Marp 图片默认左对齐。居中有两种方法：

方法一，用 HTML 包住：

```
<div style="text-align:center;">
  ![width:60%](./Fig/chart.png)
</div>
```

方法二，全局 CSS（所有图片居中）：

```
<style>
img {
  display: block;
  margin: 0 auto;
}
</style>
```

💡 提示词

我的 Marp 幻灯片里有一张图片，我想让它宽度为幻灯片宽度的 65%，并且居中显示，下方有一行 14px 的图片说明。请给我完整的代码示例。

C-3 设置背景图

我想实现什么：把图片铺满作为幻灯片背景。

简洁写法：

```
![bg](./Fig/background.jpg)
```

调整透明度（让文字更易读）：

```
![bg opacity:0.3](./Fig/background.jpg)
```

背景图 + 文字内容：

```
![bg opacity:0.15](./Fig/background.jpg)
```

```
# 封面标题
```

```
副标题文字
```

背景图只占一侧（左图右文）：

```
![bg left:40%](./Fig/chart.png)
```

```
# 右边的标题
```

```
右边的正文内容
```

`left:40%` 表示背景图占左边 40%，右边 60% 用来放文字。同理，`right:40%` 是右边放图。

常见问题：

- 背景图太亮，文字看不清 → 用 `opacity:0.2` 降低透明度
- 背景图比例不对被裁剪了 → 这是正常的，Marp 会自动裁剪以填满

💡 提示词

帮我生成一张 Marp 封面幻灯片，背景图路径 `./Fig/cover.jpg`，透明度 0.2，上面显示标题“计量经济学”和副标题“2024 年秋季学期”，{深蓝色} 文字。给我完整的代码。

C-4 图文混排

我想实现什么：左边是文字，右边是图片（或反过来）。

推荐写法（用 CSS Grid）：

```
<style scoped>
.cols {
  display: grid;
  grid-template-columns: 1fr 1fr;
  gap: 1.5em;
  align-items: center;
}
</style>

<div class="cols">
<div>

** 要点一 **：这是解释文字

** 要点二 **：这是解释文字

** 要点三 **：这是解释文字

</div>
<div>

![width:100%](./Fig/chart.png)

</div>
</div>
```

用背景图方法（更简单）：

```
![bg right:45%](./Fig/chart.png)

## 左边的标题

- 要点一
```

- 要点二
- 要点三

bg right:45% 让图片作为右侧背景占 45%，文字自然在左侧 55%。

💡 提示词

帮我生成一张 Marp 幻灯片，左边是 3 个要点列表（文字内容我自己填），右边是一张图片（路径./Fig/result.png），要求左右各占 50%，图片垂直居中。给我完整代码。

C-5 图片不显示怎么办

我想实现什么：排查为什么图片在预览或导出后看不到。

逐步排查清单：

1. 检查路径拼写

```
<!-- 检查这里：路径和文件名是否完全一致 -->

```

- 文件名区分大小写：Chart.png chart.png
- 文件夹名拼写要对：Fig fig
- 后缀名要正确：.jpg .png

2. 检查文件是否真的存在

在 VS Code 左侧文件资源管理器里，确认 Fig/ 文件夹和图片文件确实在那里。

3. 确认路径是相对路径

相对路径从 .md 文件所在位置出发：

```
<!-- .md 文件在 my_slides/ 下 -->
<!-- 图片在 my_slides/Fig/ 下 -->
<!-- 正确写法： -->


<!-- 错误写法（绝对路径，换电脑就不行）： -->

```

4. 导出 PDF 后图片不见了

重新导出，确认 slides.md 和 Fig/ 文件夹在同一个文件夹里。

5. 分享 HTML 后对方看不到图片

HTML 不嵌入本地图片。需要把整个项目文件夹（含 Fig/）一起压缩发给对方，或者改用图床。详见 [附录 F-3](#)。

💡 提示词

我的 Marp 幻灯片里图片不显示，我的文件结构是：slides.md 在 my_slides/ 下，图片在 my_slides/Fig/chart.png，我在 Markdown 里写的是 ，但预览里看不到图片。请帮我排查可能的原因。

D 代码块

D-1 基本代码块（带语法高亮）

我想实现什么：在幻灯片里显示带颜色的代码。

简洁写法：

```
```python
x = [1, 2, 3]
print(sum(x))
```
```

推荐写法：始终在代码块开头注明语言：

| 语言 | 标识 |
|------------|------------|
| Python | python |
| R | r |
| Stata | stata |
| SQL | sql |
| Bash/Shell | bash |
| JavaScript | javascript |
| YAML | yaml |
| Markdown | markdown |
| 纯文本 | text 或不写 |

常见问题：

- 代码没有颜色 → 检查语言标识是否正确，Python 和 python 效果一样，但 py 不行
- 代码块太长溢出幻灯片 → 见 [D-5](#)

D-2 调整代码字号

我想实现什么：代码字号太大或太小，想调整。

简洁写法（全局）：

```
<style>
pre, code {
  font-size: 18px;
}
```

```
}
</style>
```

只改某一页 (scoped):

```
<style scoped>
pre, code {
  font-size: 16px;
}
</style>
```

常用字号参考:

| 代码量 | 推荐字号 |
|---------------|-------------|
| 代码少 (5 行以内) | 20px ~ 22px |
| 代码适中 (5-15 行) | 17px ~ 19px |
| 代码多 (15 行以上) | 14px ~ 16px |

提示词:

我的 Marp 幻灯片里代码块字号太大，请帮我把代码块全局字号改为 17px，行间距 1.5，代码块有浅灰色背景 #f5f5f5。只返回 `<style>` 块。

D-3 左代码右输出

我想实现什么：左边显示代码，右边显示运行结果（文字或表格）。

推荐写法:

```
<style scoped>
.cols {
  display: grid;
  grid-template-columns: 1fr 1fr;
  gap: 1.5em;
}
.cols > div { overflow: hidden; }
</style>

<div class="cols">
<div>

** 代码 **

```python
import pandas as pd
df = pd.DataFrame({
 'x': [1,2,3],
 'y': [4,5,6]
})
print(df)
```

</div>
<div>
```

```
** 输出 **
```

```
```text
 x y
0 1 4
1 2 5
2 3 6
```

</div>
</div>
```

注意： <div> 标签和 Markdown 内容之间必须留空行。

提示词：

帮我生成一张 Marp 幻灯片，左边是 Python 代码（内容是用 pandas 读取 CSV），右边是对应的输出结果（文字表格形式）。左右各 50%，标题”数据读取示例”。给我完整代码。

D-4 左代码右图片

我想实现什么： 左边显示绘图代码，右边显示生成的图片。

推荐写法：

```
<style scoped>
.cols {
  display: grid;
  grid-template-columns: 55% 45%;
  gap: 1em;
  align-items: center;
}
</style>

<div class="cols">
<div>

```python
import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(0, 2*np.pi, 100)
y = np.sin(x)

plt.plot(x, y, color='steelblue')
plt.xlabel('x')
plt.ylabel('sin(x)')
plt.tight_layout()
plt.savefig('Fig/sinx.png')
```

</div>
<div>

![width:100%](./Fig/sinx.png)
```

```
</div>
</div>
```

比例建议：

- 代码短图片重要 → `grid-template-columns: 1fr 1fr`
- 代码多图片小 → `grid-template-columns: 60% 40%`
- 代码少图片大 → `grid-template-columns: 40% 60%`

💡 提示词

帮我生成一张 Marp 幻灯片，左边（55% 宽）是 Python 绘图代码，右边（45% 宽）是图片（路径./Fig/plot.png）垂直居中。标题”可视化结果”，代码字号 16px。只返回这张幻灯片的完整代码。

D-5 长代码怎么处理

我想实现什么：代码太长，超出幻灯片范围了，怎么办？

方法一：缩小字号（最简单）

```
<style scoped>
pre { font-size: 14px; line-height: 1.4; }
</style>
```

方法二：让代码块可以滚动

```
<style scoped>
pre {
  font-size: 15px;
  max-height: 400px;
  overflow-y: auto;
}
</style>
```

这样代码块会出现滚动条（在 HTML 预览中有效，PDF 中不支持滚动）。

方法三：拆成两页（推荐）

如果代码真的很长，最好拆开，用两页幻灯片来呈现：

```
## 代码（第一部分）

```python
数据读取和清洗
import pandas as pd
df = pd.read_csv("data.csv")
df = df.dropna()
```

---

## 代码（第二部分）

```python
```

```
回归分析
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(df[['x']], df['y'])
...
```

#### 方法四：只展示关键部分，用注释省略

```
数据准备（省略）...

核心代码
result = model.fit(X, y)
print(result.summary())
```

#### 💡 提示词

我的 Marp 幻灯片里有一段 20 行的 Python 代码，显示不完整。我希望把这段代码分成两张幻灯片展示，每张最多 10 行，标题分别是“代码（上）”和“代码（下）”。请帮我拆分并生成两张幻灯片的代码。

# E 版式-双栏排版和提示框

---

## E-1 双栏布局

我想实现什么：把幻灯片内容分成左右两栏。

简洁写法：

```
<style scoped>
.cols { display: grid; grid-template-columns: 1fr 1fr; gap: 1em; }
</style>

<div class="cols">
<div>

左栏内容

</div>
<div>

右栏内容

</div>
</div>
```

推荐写法（更多配置选项）：

```
<style scoped>
.cols {
 display: grid;
 grid-template-columns: 1fr 1fr; /* 左右各占 50% */
 gap: 1.5em; /* 两栏间距 */
 align-items: start; /* 顶部对齐, 改成 center 可以垂直居中 */
}
</style>

<div class="cols">
<div>

** 左栏标题 **

- 要点一
- 要点二
- 要点三

</div>
```

```
<div>

** 右栏标题 **

- 要点四
- 要点五

![width:90%](./Fig/chart.png)

</div>
</div>
```

### 不等宽变体:

```
/* 左 40% 右 60% */
grid-template-columns: 2fr 3fr;

/* 左 60% 右 40% */
grid-template-columns: 3fr 2fr;

/* 左 1/3 右 2/3 */
grid-template-columns: 1fr 2fr;
```

### 三栏布局:

```
<style scoped>
.cols { display: grid; grid-template-columns: 1fr 1fr 1fr; gap: 1em; }
</style>

<div class="cols">
<div>第一栏</div>
<div>第二栏</div>
<div>第三栏</div>
</div>
```

### 常见问题:

- 内容溢出栏外 → 检查是否给 .cols > div 加了 overflow: hidden
- 两栏高度不一致 → 加 align-items: stretch 让两栏等高

#### 💡 提示词

帮我生成一张 Marp 双栏对比幻灯片，左栏介绍 OLS 方法（3-4 个要点），右栏介绍 IV 方法（3-4 个要点），两栏宽度相同，标题”方法比较”。给我完整代码，用 scoped 样式。

## E-2 提示框 / Callout

我想实现什么：在幻灯片里放一个颜色突出的提示框，吸引注意力。

### 简洁写法（蓝色提示）:

```
<div style="background:#e8f4fd;border-left:4px solid #2196F3;padding:0.8em 1em;border-radius:4px;">
```

这里是提示内容。

```
</div>
```

四种颜色方案，直接复制使用：

蓝色（提示/信息）：

```
<div style="background:#e8f4fd;border-left:4px solid #2196F3;padding:0.8em 1em;border-radius:4px;margin:0.5em 0;">
```

**\*\* 提示： \*\*** 内容写在这里。

```
</div>
```

黄色（警告/注意）：

```
<div style="background:#fff8e1;border-left:4px solid #FFC107;padding:0.8em 1em;border-radius:4px;margin:0.5em 0;">
```

**\*\* 注意： \*\*** 内容写在这里。

```
</div>
```

绿色（正确/推荐）：

```
<div style="background:#e8f5e9;border-left:4px solid #4CAF50;padding:0.8em 1em;border-radius:4px;margin:0.5em 0;">
```

**\*\* 做法： \*\*** 内容写在这里。

```
</div>
```

红色（错误/危险）：

```
<div style="background:#fce4ec;border-left:4px solid #f44336;padding:0.8em 1em;border-radius:4px;margin:0.5em 0;">
```

**\*\* 错误： \*\*** 内容写在这里。

```
</div>
```

推荐写法（定义成 CSS class，复用更方便）：

在全局 <style> 里加：

```
<style>
.box-note { background:#e8f4fd; border-left:4px solid #2196F3; padding:0.8em 1em;
border-radius:4px; margin:0.5em 0; }
.box-warn { background:#fff8e1; border-left:4px solid #FFC107; padding:0.8em 1em;
border-radius:4px; margin:0.5em 0; }
.box-tip { background:#e8f5e9; border-left:4px solid #4CAF50; padding:0.8em 1em;
border-radius:4px; margin:0.5em 0; }
.box-error { background:#fce4ec; border-left:4px solid #f44336; padding:0.8em 1em;
border-radius:4px; margin:0.5em 0; }
</style>
```

然后使用：

```
<div class="box-note">
```

```
 ** 提示: ** 这是一条有用的信息。
```

```
</div>
```

### 💡 提示词

我想在 Marp 幻灯片里使用提示框，需要四种颜色：蓝色（提示）、黄色（警告）、绿色（推荐）、红色（错误）。请帮我生成全局 CSS class 定义（放在 `<style>` 块里），以及每种框的使用示例。

## E-3 小字备注区域

**我想实现什么：** 在幻灯片底部放一块小字的数据来源或参考文献区域。

**简洁写法：**

```
<div style="font-size:14px;color:#888;margin-top:1em;border-top:1px solid
#ddd;padding-top:0.4em;">
注：数据来源……
</div>
```

**推荐写法（贴在底部）：**

先在全局 `<style>` 里加：

```
<style>
section {
 display: flex;
 flex-direction: column;
}
.note {
 font-size: 14px;
 color: #888;
 margin-top: auto;
 padding-top: 0.5em;
 border-top: 1px solid #e0e0e0;
}
</style>
```

**使用时：**

```
图表标题
```

```
主要内容……
```

```
![width:70%](./Fig/chart.png)
```

```
<div class="note">
```

```
注：数据来源于国家统计局（2023）。单位：亿元。
```

```
</div>
```

## 💡 提示词

帮我在 Marp 幻灯片里实现底部小字备注，要求备注始终贴在幻灯片最底部，字号 13px，颜色 #999，上方有一条浅灰色分割线。给我完整的 CSS 和使用示例。

## E-4 制作封面页

**我想实现什么：**做一张好看的封面页，和正文页风格有所区分。

**最简单写法（用 `_class` 区分）：**

先定义一个封面样式，在全局 `<style>` 里加：

```
<style>
section.cover {
 background: #1a237e;
 color: white;
 display: flex;
 flex-direction: column;
 justify-content: center;
 align-items: center;
 text-align: center;
}
section.cover h1 {
 font-size: 48px;
 color: white;
 border: none;
}
section.cover p {
 font-size: 22px;
 color: #c5cae9;
}
</style>
```

然后在封面页用 `<!-- _class: cover -->`：

```
<!-- _class: cover -->
<!-- _paginate: false -->
<!-- _header: "" -->
<!-- _footer: "" -->
```

# 计量经济学导论

2024 年秋季学期

张三 | 某某大学

**用背景图做封面：**

```
<!-- _paginate: false -->
<!-- _header: "" -->
<!-- _footer: "" -->
```

```
![bg opacity:0.15](./Fig/cover-bg.jpg)
```

# 课程标题

副标题 | 2024 年

**推荐写法：**结合背景色和简单排版，不需要复杂设计：

```

marp: true
theme: default

<style>
section.cover {
 background: linear-gradient(135deg, #1565c0, #283593);
 color: white;
}
section.cover h1 { color: white; font-size: 44px; margin-bottom: 0.3em; }
section.cover h2 { color: #90caf9; font-size: 24px; font-weight: normal; border: none; }
section.cover p { color: #bbdefb; font-size: 18px; }
</style>

<!-- _class: cover -->
<!-- _paginate: false -->

实证研究方法

Empirical Research Methods

2024 年秋季 | 某某大学经济学院
```

### 💡 提示词

帮我生成一张 Marp 封面页，要求：深蓝色渐变背景（从 #1565c0 到 #283593），标题白色大字，副标题浅蓝色，不显示页码和页眉页脚，用 `_class: cover` 实现。给我完整的 CSS 和封面页代码。

# F 导出 HTML 和 PDF

---

## F-1 导出 PDF

**我想实现什么：**把幻灯片保存为 PDF 文件，方便分享和打印。

**操作步骤：**

1. 在 VS Code 里打开 .md 文件
2. 按 `Ctrl+Shift+P` (Mac: `Cmd+Shift+P`) 打开命令面板
3. 输入 `Marp`，选择 **Marp: Export Slide Deck...**
4. 在保存对话框中选择 `.pdf` 格式
5. 选择保存位置，点击保存
6. 等待几秒（进度条在底部状态栏显示）

**PDF 的优点：**

- 本地图片会被嵌入，发一个文件就够
- 任何设备都能打开，不需要插件
- 字体和排版完全固定

**常见问题：**

- 导出后打开 PDF，图片不见了 → 图片路径有误，重新检查相对路径
- 导出很慢，一直等 → 正常，特别是第一次使用时 Marp 会下载 Chromium
- 导出报错 → 重启 VS Code 后再试

**批量导出（命令行方式）：**

如果你安装了 Node.js，可以在终端里用命令行导出：

```
npx @marp-team/marp-cli slides.md --pdf
```

---

## F-2 导出 HTML

**我想实现什么：**把幻灯片保存为 HTML 文件，可以在浏览器里展示。

**操作步骤：**

和导出 PDF 完全一样，只是在保存对话框里选 `.html` 格式。

**HTML 的优点：**

- 可以在浏览器里展示，支持鼠标点击翻页

- 文件通常比 PDF 小
- 可以放到网站上

#### HTML 的注意事项:

- 本地图片不会嵌入 HTML (这是最大的坑)
- 如果幻灯片有本地图片, 分享时必须连图片文件夹一起发
- 详见 F-3

#### 导出带嵌入图片的 HTML (实验性功能):

Marp CLI 可以用 `--html` 配合 `--allow-local-files` 导出嵌入图片的 HTML:

```
npx @marp-team/marp-cli slides.md --html --allow-local-files
```

注意: 这个功能在 VS Code 插件里不一定支持, 命令行方式更可靠。

---

## F-3 HTML 分享时图片不见了

**我想实现什么:** 搞清楚为什么发出去的 HTML 里图片消失了, 以及怎么解决。

#### 原因:

Marp 导出的 HTML 文件里, 图片是用路径引用的:

```

```

这个路径只在你的电脑上有效。对方拿到 HTML 文件后, 他们的电脑上没有 `./Fig/chart.png`, 所以图片消失。

#### 解决方案一: 发整个文件夹 (推荐)

```
my_slides/ ↳ 把这个文件夹压缩成 zip 发出去
 slides.html
 Fig/
 chart.png
 diagram.png
```

对方解压后, 在自己电脑上打开 `slides.html`, 图片就能显示。

#### 解决方案二: 用图床代替本地图片

把图片上传到图床, 用网络链接替代本地路径:

```
![] (https://your-image-host.com/chart.png)
```

这样 HTML 文件单独发出去也能显示图片 (需要网络)。

#### 解决方案三: 直接发 PDF

本地图片 + PDF 是最省心的组合。如果只是发给别人查看, 不需要 HTML。

#### 决策建议:

```
我要分享幻灯片
 给别人查看 → PDF (最省心)
 在网站上展示 + 图片不多 → 图床 + HTML
 在网站上展示 + 不想折腾图床 → 压缩文件夹 + HTML
```

## F-4 图床基本思路

**我想实现什么：** 了解图床是什么，以及什么时候值得用。

**图床是什么：**

图床是一种在线图片存储服务。你把图片上传上去，它给你一个公开可访问的链接。在 Marp 里用这个链接替代本地路径：

```
<!-- 本地路径 -->

<!-- 图床链接 -->

```

用了图床之后，任何设备、任何地方打开这份幻灯片（PDF 或 HTML），都能看到图片——只要有网络。

**常用图床对比：**

服务	费用	稳定性	适合场景
GitHub + jsDelivr	免费	高	长期使用，技术用户
Cloudflare Images	付费	很高	重要项目
SM.MS	免费（有限额）	中	临时测试
阿里云 OSS	付费	很高	商业使用
微博/公众号图片	免费	低	<b>不推荐</b> ，链接会失效

**什么时候值得用图床：**

- 你有持续在线分享幻灯片的需求
- 你的幻灯片需要发布到网站上
- 你需要多人协作，大家都能看到图片
- 你的图片数量少，上传不麻烦

**什么时候不需要：**

- 你只是偶尔发 PDF 给别人
- 你的幻灯片只在自己电脑上用
- 你是初学阶段，不想增加工作量

**推荐的 GitHub 图床 workflow（简版）：**

1. 在 GitHub 上建一个仓库，专门用来存图片
2. 把图片上传到这个仓库
3. 用 jsDelivr 生成 CDN 链接：<https://cdn.jsdelivr.net/gh/用户名/仓库名 @main/图片名.png>
4. 在 Marp 里用这个链接

### 💡 提示词

帮我介绍如何在 GitHub 上建立一个图床仓库，并用 jsDelivr 生成图片链接。要求步骤简洁，面向没有 GitHub 使用经验的初学者。

# G 常用提示词

这个附录收录了最常用的提示词模板。每个模板都可以直接复制使用，把**方括号里的内容**替换成你的实际情况。

适用的 AI: 豆包、DeepSeek、ChatGPT、Claude 等主流 AI 都可以。

---

## G-1 生成基础模板

**场景:** 我想从零开始，让 AI 帮我生成一份可以直接用的 Marp 模板。

### 💡 提示词

请帮我生成一份 Marp 幻灯片的完整基础模板。

要求:

- 主题: default, 尺寸: 16:9
- 开启页码 (格式: 当前页 / 总页数)
- 页眉: [填入课程名或项目名], 页脚: [填入姓名和日期]
- 用 headingDivider: 2 自动分页
- 全局字号 24px, h1 字号 40px, h2 字号 30px
- 包含一个蓝色深色背景的封面页 (不显示页码/页眉/页脚)
- 封面页后面包含 3 张示例内容页 (一张纯文字、一张带代码块、一张双栏)

只返回完整的 .md 文件内容, 用 markdown 代码块包住, 不加任何说明文字。

---

## G-2 按已有模板生成完整幻灯片

**场景:** 我有一篇论文或报告, 想让 AI 直接输出一份完整的幻灯片, 并保持我现有模板的风格。

### 💡 提示词

我有一篇论文/报告, 内容如下:

[粘贴论文摘要、引言或完整正文]

请根据以上内容, 生成一份完整的 Marp 幻灯片 (.md 格式)。

**风格参考:** 请参照以下模板的 YAML 头和 CSS 样式, 保持相同的字号、配色和版式:

[粘贴模板文件的 YAML 头和 <style> 块]

**其他要求:**

- 面向受众: [例如: 经济学专业研究生]
- 幻灯片总页数: [例如: 12-15 页]

- 包含封面页、目录页、各节内容页、结论页
- 每页正文要点控制在 4-5 条以内，不要太满
- 只返回完整的 .md 文件内容，用 markdown 代码块包住

## G-3 修改已有模板（保留风格）

场景：我有自己的模板，只想加一个新功能，不想改其他的。

### 💡 提示词

我有一份 Marp 幻灯片模板，YAML 头和 `<style>` 块如下：

[把你的 YAML 头和 `<style>` 块完整粘贴在这里]

请在这个基础上帮我添加：[具体描述你想加的功能，例如：

- “一个蓝色提示框的 CSS class (叫 .box-note)”
- “把 h2 标题颜色改为 #1565c0”
- “代码块背景改为浅灰色，字号改为 17px”]

要求：

1. 保留现有 YAML 和 CSS 完全不变
2. 只在现有 `<style>` 末尾追加新内容
3. 返回修改后的完整 YAML 头和 `<style>` 块
4. 不需要任何解释

## G-4 生成双栏布局页

场景：我需要一张双栏幻灯片，不知道怎么写 CSS Grid。

### 💡 提示词

请帮我生成一张 Marp 幻灯片，双栏布局，具体内容如下：

- 幻灯片标题：[填入标题]
- 左栏内容：[描述左栏内容，例如“OLS 方法的 3 个假设，用要点列表”]
- 右栏内容：[描述右栏内容，例如“一张图片，路径 ./Figs/ols.png”]
- 左右宽度比：[例如 50:50 或 60:40]

要求：

- 用 `<style scoped>` 避免影响其他页
- `<div>` 标签和 Markdown 内容之间留空行
- 返回从 `---` 分页符开始到下一个 `---` 的完整代码
- 不加任何解释

## G-5 调整字号和配色

场景：我觉得当前幻灯片的字号或颜色不好看，想调整。

### 💡 提示词

我的 Marp 幻灯片当前的 `<style>` 块如下：

[粘贴现有 `<style>` 块，或写“暂无 `<style>` 块”]

请帮我调整以下样式：

- 正文字号：改为 [目标字号，例如 22px]
- h1 字号：改为 [目标字号]，颜色：[目标颜色，例如 #1a237e]
- h2 字号：改为 [目标字号]，颜色：[目标颜色]
- [其他需求，例如“代码块字号 16px，背景色 #f5f5f5”]

要求：如果已有 `<style>` 块，只修改相关部分，其余保留。返回完整的修改后 `<style>` 块，不加解释。

## G-6 处理图片排版页

**场景：**我想让某张幻灯片以某种方式展示图片，不知道怎么排。

### 💡 提示词

请帮我生成一张 Marp 幻灯片，要求如下：

图片展示方式：[选一种描述，例如：

- “一张图片居中，宽度 70%，下方有图片说明”
- “左边是 3 个文字要点，右边是一张图片”
- “上半张是图片，下半张是文字描述”
- “背景是一张低透明度图片，前景是标题和要点”]

图片路径：./Figs/[图片文件名]

幻灯片标题：[标题文字]

文字内容：[要点或描述文字]

要求：

- 用 `<style scoped>` 隔离样式
- 给我从 --- 到下一个 --- 的完整代码
- 不加任何解释

## G-7 论文精要转幻灯片

**场景：**先让 AI 写论文精要，再把精要转成幻灯片——两步走，质量更稳。

### 第一步：生成精要

#### 💡 提示词（第一步）

请阅读以下论文，用中文写一份 800–1000 字的精要，要求：

- 包含研究背景、研究问题、数据与方法、核心发现、政策含义
- 突出关键数据和因果结论
- 语言简洁，适合转化为学术汇报幻灯片

[粘贴论文全文或摘要 + 引言 + 结论]

### 第二步：精要转幻灯片

**💡 提示词（第二步）**

以下是一篇论文的摘要：

[粘贴第一步生成的摘要]

请将其转化为一份 Marp 幻灯片（10–12 页），要求：

- 参考以下模板风格：[粘贴模板的 YAML 头和 `<style>` 块]
- 面向受众：经济学领域同行
- 结构：封面、引言、数据/方法、核心结果（2–3 页）、结论
- 结果页保留关键系数和显著性标注（\*\*\* \*\* \*）
- 只返回 .md 文件，用代码块包住，不加解释

## G-8 遇到报错时如何提问

**场景：**幻灯片排版出问题了（显示不对、图片消失、导出报错等），想让 AI 帮忙排查。

**💡 提示词**

我的 Marp 幻灯片出现了问题，描述如下：

**问题现象：** [具体描述，例如：

- “双栏布局中右栏内容跑到左栏下面去了”
- “代码块显示超出幻灯片边界”
- “导出 PDF 时某张幻灯片空白”
- “图片在预览里能看到，导出 PDF 后不见了”]

**出问题的代码：**

[把出问题的那张幻灯片的完整代码粘贴在这里]

**环境：** VS Code + Marp for VS Code 插件

请帮我找出原因，并给出修正后的代码。只返回修正后的代码块。

**常见问题快速定位：**

问题	最可能的原因	快速提示词
双栏布局乱了	<code>&lt;div&gt;</code> 和 Markdown 之间没有空行	描述给 AI：“div 和内容之间加了空行还是乱”
图片不显示（预览）	路径拼写错误或大小写不对	“图片路径是./Figs/xxx.png，这样写对吗”
图片不显示（PDF）	文件夹结构不对	“slides.md 和 Figs 文件夹的位置关系”
字号改了没变化	<code>&lt;style&gt;</code> 写在了 YAML 头里面	“ <code>&lt;style&gt;</code> 应该写在 YAML 头的哪里”
导出 PDF 超慢	Chromium 第一次下载	等待即可，或重启 VS Code
数学公式不渲染	忘记加 <code>math: katex</code>	YAML 里加 <code>math: katex</code>

## G-9 其他常用提示词

### 💡 提示词：生成数学公式页

帮我生成一张 Marp 幻灯片，展示以下公式推导：[描述公式内容，例如“OLS 估计量的推导，从最小化残差平方和到求解 $\hat{\beta}$ ”]。要求：用 LaTeX 公式块 ( $\dots$ )，每步推导一行，对齐符号使用 `aligned` 环境。给我完整的幻灯片代码。

### 💡 提示词：生成表格对比页

帮我生成一张 Marp 幻灯片，用 Markdown 表格对比以下内容：[描述要对比的内容，例如“OLS、IV、DID 三种方法，对比维度包括：适用条件、核心假设、典型应用”]。表格样式整洁，字号适中。给我完整的幻灯片代码。

### 💡 提示词：生成目录页

帮我生成一张 Marp 目录页，包含以下章节：

1. [章节一名称]
2. [章节二名称]
3. [章节三名称]
4. [章节四名称]

要求：编号清晰，当前章节用不同颜色高亮（当前是第 [N] 章）。给我完整的幻灯片代码，用 `scoped` 样式。

# I 基本 Markdown 语法

Marp 使用 Markdown 来写幻灯片内容。Markdown 是一种极简的文本标记语言——你只需要在普通文字里加一些符号，就能表达标题、加粗、列表、代码等格式。

这个附录列出了在 Marp 幻灯片里最常用的 Markdown 语法，方便随时查阅。

---

## 标题

用 # 号表示标题，# 越多层级越低：

```
一级标题（幻灯片主标题）
二级标题（常用于分节标题）
三级标题（子标题）
```

在 Marp 里，# 和 ## 是最常用的两级，### 偶尔用于小节。

---

## 强调

```
** 加粗文字 **
* 斜体文字 *
~~ 删除线 ~~
`行内代码`
```

渲染效果：**加粗文字**，*斜体文字*，~~删除线~~，`行内代码`

---

## 列表

无序列表（用 - 或 \*）：

```
- 要点一
- 要点二
 - 子要点（缩进两个空格）
 - 子要点
- 要点三
```

有序列表：

1. 第一步
2. 第二步
3. 第三步

### 💡 Marp 里的列表建议

幻灯片里的列表不要超过 5 条，超过就考虑拆成两页。子要点也尽量少用，层级太深会让幻灯片显得很拥挤。

---

## 链接与图片

### 链接：

```
[链接文字](https://example.com)
```

### 图片：

```
![图片描述](./Figs/chart.png)
```

### Marp 专用：图片尺寸控制

```
![width:400px](./Figs/chart.png)
![width:60%](./Figs/chart.png)
![height:300px](./Figs/chart.png)
```

---

## 代码

### 行内代码（用反引号包住）：

```
用 `marp: true` 启用 Marp。
```

### 代码块（用三个反引号，注明语言）：

```
```python
import pandas as pd
df = pd.read_csv("data.csv")
```
```

常用语言标识：python、r、stata、bash、sql、yaml、markdown

---

## 引用

- > 这是一段引用文字。
- > 可以跨多行。

渲染后会显示为带左侧竖线的缩进段落。

## 分隔线

```

```

在 Marp 里，`---` 用于**分页**（见**第一章**）。在普通 Markdown 里，它表示水平分隔线。

---

## 表格

```
列一	列二	列三
内容	内容	内容
内容	内容	内容
```

对齐控制：

```
左对齐	居中	右对齐
内容	内容	内容
```

## 数学公式

Marp 支持 LaTeX 数学公式（需在 YAML 头加 `math: katex`）：

行内公式：

```
质能方程为 $E = mc^2$ 。
```

独立公式块：

```
$$
\hat{\beta} = (X'X)^{-1}X'y
$$
```

## HTML 标签

Marp 支持在 Markdown 里直接写 HTML，常用于：

```
<!-- 图片尺寸（最稳妥的写法） -->

<!-- 文字颜色 -->
蓝色文字

<!-- 小字注释 -->
<small>这是小字</small>
```

## 扩展阅读

如果你想系统学习 Markdown，以下资源值得参考：

- **官方规范**：[CommonMark Spec](#)——Markdown 的标准规范，有交互式示例
- **中文速查表**：[Markdown Guide 中文版](#)——按功能分类，查找方便
- **Marp 官方文档**：[Marp Core Markdown](#)——专门介绍 Marp 扩展的 Markdown 语法（图片背景、指令等）

**i** 不需要全部记住

实际写 Marp 幻灯片时，你用到最多的只有：标题（# ##）、加粗（\*\*）、列表（-）、代码块（``）和图片（![]() 或 <img>）。其他语法用到的时候来这里查就好。