



Marp for VS Code

[Marp](#)(Markdown Presentation Ecosystem)是一款可以轻松将Markdown转换为HTML/PDF/PPTX的工具，能让我们专注于PPT的内容而不是排版。如果你是精通Markdown和css的程序员，那么用起来会更加顺手。

Marp的优点显而易见，对于喜欢Markdown、讨厌做PPT的人来说是一大利器，写好一套属于自己的主题样式就是一劳永逸。缺点就是排版没有那么灵活，也不支持动画效果，导出的PPT文件无法二次编辑。但对于程序员来说，不需要花里胡哨的PPT，Marp已经非常够用了。

Marp有两种使用方式，[Marp for VS Code](#)和[Marp CLI](#)。Marp for VS Code已经能够满足大多数使用场景，也更加便捷。如果插件无法满足你的需求，那么就可以选择Marp CLI，它支持更丰富的功能。本文就简单介绍下如何使用Marp for VS Code。

开始使用

- 如果还不会熟练使用Markdown，可以查阅[Markdown中文教程](#)进行学习。
- 在VS Code中安装[Marp for VS Code](#)插件。
- 新建Markdown文件（后缀名为 `.md` ），在头部插入以下代码，点击右上角预览按钮即可实时编辑。

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

- 点击VS code右上角marp标志，选择 `Export Slide Deck` 可以导出PDF、HTML、PPT等格式文件（注意：导出的PPT不能编辑）。
- 点击VS code右上角marp标志，选择 `Toggle Marp Feature For Current Markdown` 可以切换预览模式。

语法 Syntaxes

Marp的Markdown语法基于CommonMark，有一些值得注意的语法：

- 使用 `---` 或 `___` 或 `***` 或 `- - -` 来拆分幻灯片。
- 段落中的换行符将自动转换为 `
`，也可以直接使用 `
` 换行。
- 支持一些（不常见的）列表标记，如 `*` 和 `1)`。
- 支持GitHub Flavored Markdown(GFM)的一些扩展语法，如自动链接、表情符号简码 (例如 😊)、删除线 (`~~strike~~`)、代码块语法高亮、表格。
- 出于安全原因，Marp默认情况下只允许使用 `<style>` 和 `
` 这两种HTML标签。如需启用所有HTML标签，请在当前工具中更改配置。

指令 Directives

文档头部必须要有`front-matter`来定义指令。指令分为全局指令（Global directives，控制所有幻灯片）和局部指令（Local directives，控制特定幻灯片）。定义方式也有两种：

front-matter头部定义全局指令：

```
---  
theme: default  
paginate: true  
---
```

HTML注释定义局部指令：

注释也用于PPT演示者备注，但当它被解析为指令时则不会被收集展示。

```
<!--  
theme: default  
paginate: true  
-->
```

全局指令

全局指令有**title**、**author**、**description**、**keywords**（逗号分隔）这类描述信息，也有**url**、**image**这类HTML专用属性，以下是常用的几种全局指令。

名称	描述
marp	是否在VSCode中启用Marp。
theme	指定样式主题， 内置主题 有 <code>Default</code> 、 <code>Gaia</code> 、 <code>Uncover</code> 。
style	用于快捷调整整体主题的CSS。
size	指定幻灯片尺寸比例，内置尺寸有 <code>4:3</code> 、 <code>16:9</code> 、 <code>4K</code> 。
headingDivider	在指定级别标题前自动添加幻灯片分隔符，支持1~6数字或数组。
math	指定数学渲染引擎，支持 <code>mathjax</code> 、 <code>katex</code> ，建议显式声明。

局部指令

- 局部指令会将设置应用于当前页和后续页，若想仅适用于当前页，可添加下划线前缀，如 `_paginate`。
- 除下面的常用属性，还有**color**、**backgroundColor**、**backgroundImage**、**backgroundPosition**、**backgroundRepeat**、**backgroundSize**这类样式属性。

名称	描述
paginate	设置为 <code>true</code> ，将在右下角显示页码。
header	指定幻灯片页眉内容，支持内联Markdown。
footer	指定幻灯片页脚内容，支持内联Markdown。
class	为单页幻灯片外层元素 <code><section></code> 设置class属性。

列表

Markdown一般是使用 `-` 和 `1.` 来展示列表，Marp支持使用 `*` 和 `1)` 实现逐条显示的动画效果（仅支持HTML，在PDF和PPT只是普通列表）。

1. One
 2. Two
 3. Three
- One
 - Two
 - Three

自适应标题

当在标题中插入 `<!--fit-->`，标题将被缩放以适合单行，就像此页的标题。

小标题也可以哦

数学公式

Marp支持MathJax、KaTeX这两种主流数学渲染引擎，属性值分别为 `mathjax`、`katex`，不同版本Marp默认引擎不一致，建议显式声明，更多信息请查阅[官方文档-Math typesetting](#)。

- 使用 `$` 符号包裹即可使用行内公式，例如 `$ax^2+bc+c$` 会渲染为 $ax^2 + bc + c$ 。
- 使用 `$$` 包裹就会渲染为公式块，如：

```
$$  
f(x) =  
  \int_{-\infty}^{\infty}  
  \hat{f}(\xi) \, e^{2 \pi i \xi x}  
  \, d\xi  
$$
```

$$f(x) = \int_{-\infty}^{\infty} \hat{f}(\xi) e^{2\pi i \xi x} d\xi$$

设置

点击右上角Marp标志，选择 `Open Extension Settings` 可快速打开Marp设置页面。

- **Breaks**: 设置如何渲染换行符。
- **Chrome Path**: 设置浏览器自定义路径以导出PDF、PPTX和图片。如果为空，则会自动寻找已安装的浏览器。
- **Enable HTML**: 是否启用所有HTML标签，默认只允许使用 `style` 和 `br` 标签。
- **Export Type**: 选择默认的导出文件类型。
- **Math Typesetting**: 选择默认的数学渲染引擎。
- **Outline Extension**: 是否启用编辑器大纲视图功能。
- **Pdf: Note Annotations**: 是否将演示者注释作为备注添加到导出的PDF中。
- **Pdf: Outlines**: 设置是否在导出的PDF中添加书签。
- **Strict Path Resolution During Export**: 是否在导出过程中启用严格的路径解析。
- **Themes**: 声明主题样式文件路径，可以是本地文件，也可以是远程文件链接。

工作区配置文件

也可以打开/新建工作区配置文件 `.vscode/settings.json` 来更好地管理设置：

```
{
  "markdown.marp.breaks": "on", // 渲染换行符
  "markdown.marp.chromePath": "", // 浏览器路径
  "markdown.marp.enableHtml": true, // 启用所有HTML标签
  "markdown.marp.exportType": "pdf", // 默认的导出文件类型
  "markdown.marp.mathTypesetting": "mathjax", // 默认的数学渲染引擎
  "markdown.marp.outlineExtension": true, // 启用大纲视图
  "markdown.marp.pdf.noteAnnotations": true, // 将注释作为备注添加到导出的PDF中
  "markdown.marp.pdf.outlines": "headings", // 在导出的PDF中添加书签
  "markdown.marp.strictPathResolutionDuringExport": false, // 导出禁用严格的路径解析
  "markdown.marp.themes": [ // 声明主题样式文件
    "./themes/companyLightBlue.css",
  ],
}
```

美化

美化PPT有多种方式

- 学会排版设计，灵活利用图片。
- 使用 `theme` 指令直接套用写好的主题样式。
- 指令设置 `style`、`background`、`class` 等相关样式属性。
- 使用 `<style>` 在MD文档中编写内联样式，`scoped` 可以只应用在当前页。
- 在设置中启用HTML功能后，可以直接插入html代码。

图片

- 通过 `w:` 和 `h:` 关键字定义宽高。

`inline-pic1`

- 支持 **CSS filters**, 属性有 `blur`、`brightness`、`contrast`、`drop-shadow`、`grayscale`、`hue-rotate`、`opacity` 等。

`inline-pic`

- 添加 `bg` 设置背景图片, 可设置背景大小(`cover`、`contain`、`fit`、`auto`、`x%`)、拆分背景(`left`、`right`) 和多重背景排列 (默认横向, `vertical` 可改为纵向) 。

A

B

C

自定义主题

创建样式文件 `./themes/your-theme.css`，第一行须为 `/* @theme theme-name */`。

```
/* @theme your-theme */

/* 可引入远程样式文件，如修改代码高亮主题、引入字体文件等 */
@import 'https://cdnjs.cloudflare.com/ajax/libs/highlight.js/11.7.0/styles/base16/github.min.css';

/* 可继承其他主题样式。当涉及Sass、Less这类文件时，建议使用@import-theme引入 */
@import-theme 'Gaia';

section { /* section是每张幻灯片的根元素 */
  width: 960px; /* 可通过设置width和height修改幻灯片尺寸 */
  height: 720px;
  /* 支持使用变量，详见https://github.com/marp-team/marp-core/tree/main/themes */
  --color-background: #ddd;
}

/* :root和section类似，但:root优先级比section高。*/
:root.contentPage { /* 可自定义类的样式，在MD文档中使用class指令即可 */
  background: #02669d;
}

/* 自定义页码样式 */
section::after {
  content: 'Page ' attr(data-marpit-pagination) ' / ' attr(data-marpit-pagination-total); /* 可添加页码前缀和总页码 */
}

/* 自定义页眉和页脚样式 */
header, footer {}
```

打开或新建工作区配置文件 `.vscode/settings.json`，引入主题文件，支持本地文件和远程文件：

```
{
  "markdown.marp.themes": [
    "https://example.com/foo/bar/custom-theme.css",
    "../themes/your-theme.css"
  ]
}
```

在Markdown文件中定义theme指令，主题名为样式文件头部第一行 `@theme` 后的名称。

```
---
marp: true
theme: your-theme
---

# Use your own theme
```

参考文章

- [Marp官方文档](#)
- [Marp-theme](#)
- [用 Marp 基于 markdown 制作幻灯片 | CAI](#)
- [marpstyle](#)