



Vscode 中 Markdown 环境配置

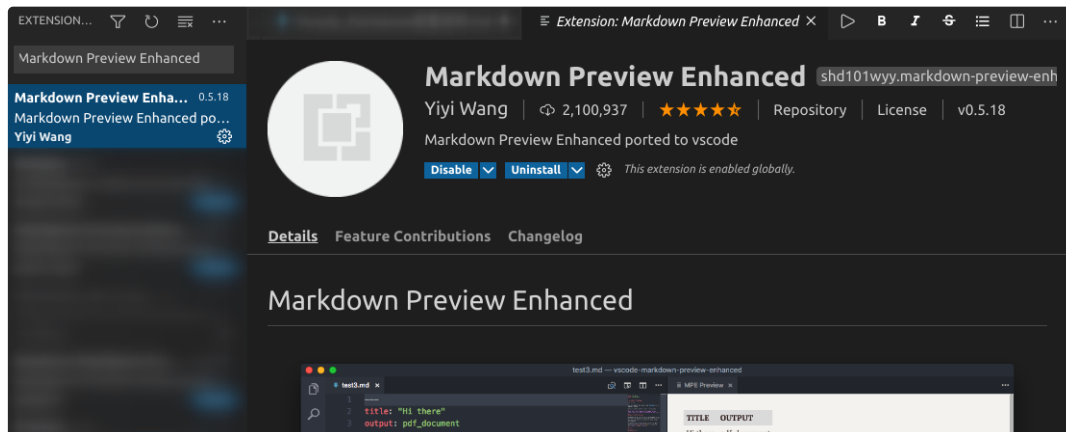
- Vscode 中 Markdown 环境配置
 - 1. 概述
 - 2. MPE(Markdown Preview Enhanced) 插件
 - 1. 在插件商店搜索 Markdown Preview Enhanced 并安装
 - 2. 预览
 - 3. 格式化导出
 - 4. 快速绘图
 - 4.1 ditaa快速绘图
 - 4.2 ditaa图片的导出配置
 - 4.3 palntuml快速绘图
 - 5. css渲染
 - 3. Markdown Shortcuts插件
 - 4. PicGo 插件 + github 图床
 - 5. Pandoc Latex格式文档打印
 - 6. Reveal.js 幻灯片制作
 - 7. 参考资料

1. 概述

本文旨在帮助开发者在vscode中配置用于书写markdown文本的环境，主要介绍书写markdown用到的常用插件，如MPE (Markdown Preview Enhanced) 插件的预览、导出功能等，以及PicGo+github图床的配置。

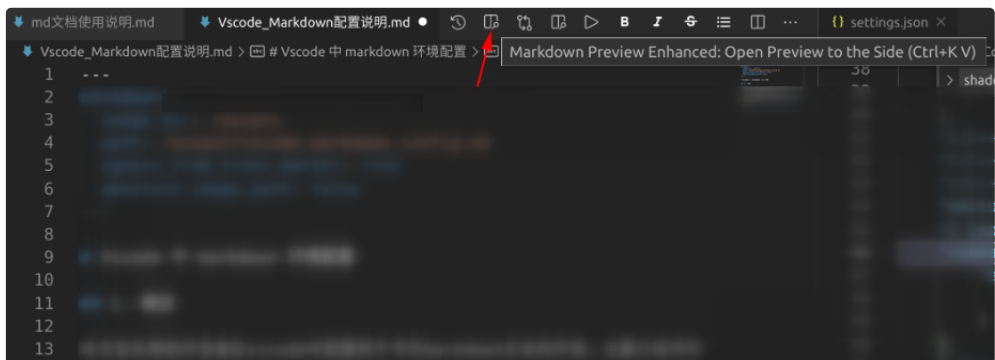
2. MPE(Markdown Preview Enhanced) 插件

1. 在插件商店搜索 Markdown Preview Enhanced 并安装

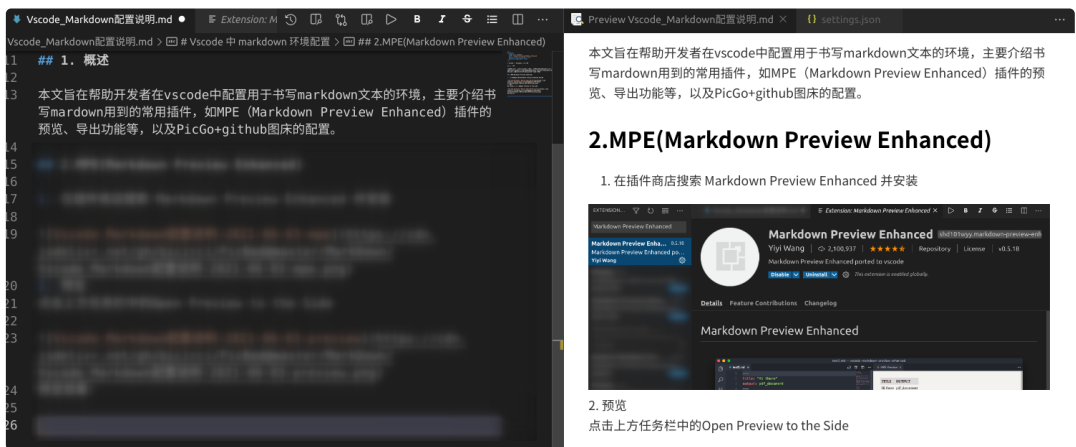


2. 预览

点击上方任务栏中的Open Preview to the Side



预览效果：



3. 格式化导出

在右侧预览界面点击右键，即可选择想要的格式导出



4. 快速绘图

- 环境配置

要在MPE中使用一些快速绘图功能，需要先进行环境的配置：

1. 安装Java
2. 勾选vscode设置中MPE的Enable Scirpt Execution设置选项
3. (Windows)安装Imagemagick软件:<https://download.imagemagick.org/ImageMagick/download/binaries/ImageMagick-7.1.0-1-Q16-HDRI-x64-dll.exe>
注意勾选旧版接口兼容convert
4. (Ubuntu)Ubuntu默认应该已经包含Imagemagick，若没有可直接apt-get install graphicsmagick-imagemagick-compat
需要注意Ubuntu本身缺少中文字库，导致MPE插件导出的图片无法显示中文，暂时没有很好的解决办法，只能通过第三方程序指定字库参数，将图片代码生成可用的图片。

4.1 ditaa快速绘图

MPE支持ditaa快速绘制Ascii Art的图片。ditaa语法参见<https://github.com/stathissideris/ditaa>。

MPE中渲染ditaa图片和通常的代码块类似

```
“
``` ditaa (cmd, run_on_save=true, align="center", args=["-E"], hide=true)

+-----+
| Text cYEL |--> //生成一个YELLOW的矩形
+-----+
```
”
```



cmd或cmd=ture 开启ditaa代码渲染，注意：

- 需要预先安装JAVA
- 可能需要在插件设置中开启Enable Scirpt Execution

run_on_save 在保存文档时运行代码，重新渲染ditaa图形

align="center"居中显示

args=[""] 配置ditaa启动参数，参考ditaa语法说明

4.2 ditaa图片的导出配置

当前附带了ditaa代码的markdown文档，离开MPE就无法显示为图形了，因此还需要经过MPE的导出操作，生成可以粘贴到别处的markdown。

在预览界面右键导出时选择 **Save as Markdown** 格式，此时MPE会将预览中渲染生成的ditaa图生成一张本地的png图片，并将原ditaa代码块替换为一个本地图片的链接。

具体的导出选项，可以在文档开头单独设置

```
“
---
markdown:
image_dir: /assets 图片的输出文件夹
absolute_image_path: false 图片文件夹是否采用绝对位置
path: vscode_markdown_config.md 导出的markdown文件名
ignore_from_front_matter: true 导出时是否将这几行配置信息隐藏
---
”
```

```

1  ---
2  markdown:
3    image_dir: /assets
4    path: vscode_markdown_config.md
5    ignore_from_front_matter: true
6    absolute_image_path: false
7  ---
8
9  # Vscode 中 Markdown 环境配置
10
11 [toc]
12
13 ## 1. 概述

```

Vscode 中 Markdown 环境配置

- Vscode 中 Markdown 环境配置
 - 1. 概述
 - 2. MPE(Markdown Preview Enhanced)
 - 1. 在
 - 2. 预

4.3 plantuml快速绘图

plantuml是专门用于标准UML图形的绘图软件，其语法可以参考<https://plantuml.com/zh/>

“

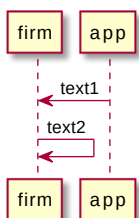
```

```puml (align="center")
title This is title
firm<-app: text1
firm->firm:text2
```

```

”

This is title



5. css渲染

MPE支持用户自定义渲染CSS脚本，并且以HTML格式导出时可以保持原样。

适合Markdown显示用的CSS脚本可以自行搜索并下载，导入MPE：

- 在vscode的插件文件夹\$HOME/.vscode/extension中搜索preview_theme文件夹，以定位MPE插件的CSS保存位置。
- 将下载的css脚本foo.css放入preview_theme文件夹
- ctrl+shift+p 输入 Open Settings(Json) 打开vscode的配置文件settings.json, 添加 "markdown-preview-enhanced.previewTheme": "foo.css", "markdown-preview-enhanced.printBackground": true 两句。

```

{} settings.json • # markdown-1.css Preview Vscode_Markdown配置说明.md
home > eric > .config > Code > User > {} settings.json > ...
50 },
51 "editor.fontSize": 16,
52 "picgo.picBed.current": "github",
53 "picgo.picBed.github.branch": "master",
54 "picgo.picBed.github.token": "ghp_TZohPVMP1Mzn4MUkl3DYBfqnvLi7x2",
55 "picgo.picBed.github.customUrl": "https://cdn.jsdelivr.net/gh/bicirci/PicBed",
56 "picgo.picBed.github.path": "MarkDown/",
57 "picgo.picBed.github.repo": "bicirci/PicBed",
58 "picgo.customUploadName": "${mdFileName}-${date}-${fileName}${ext}",
59 "markdown-preview-enhanced.previewTheme": "foo.css",
60 "markdown-preview-enhanced.printBackground": true,
61 // "markdown-preview-enhanced.codeBlockTheme": "MdCode-2.css"
62

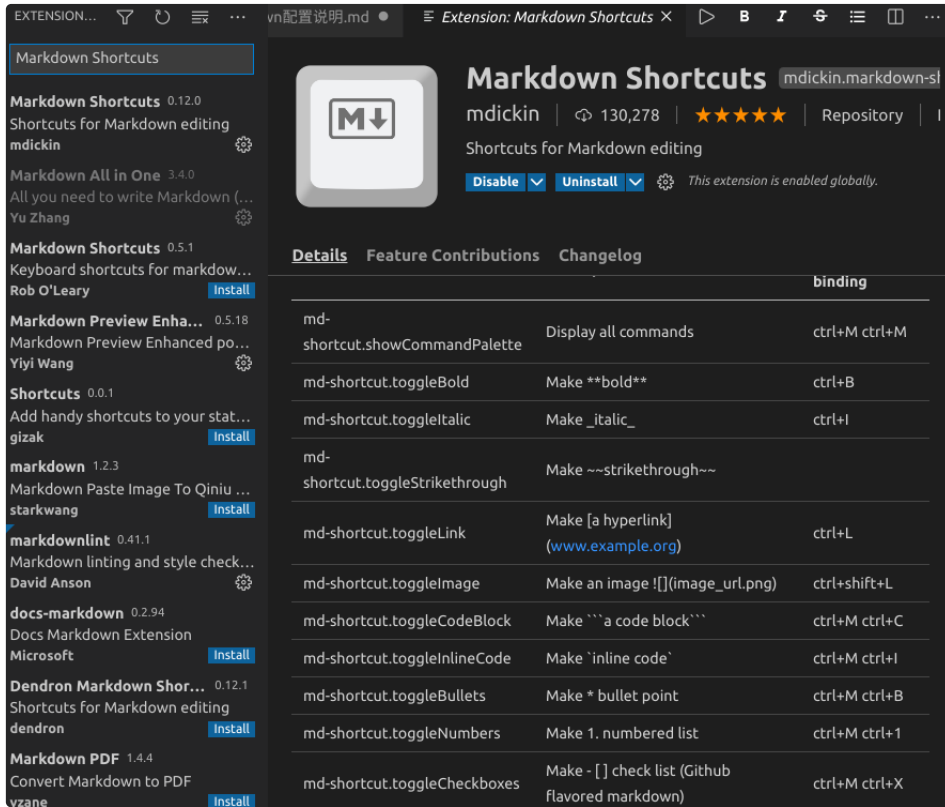
```

此外也可以通过 `cmd-shift-p` 然后运行 Markdown Preview Enhanced: Customize CSS
此时style.less 文件将会被打开, 在该文件中编写样式同样可以实现渲染后的预览。

- 代码块的格式定义在同级的prism_theme文件夹下

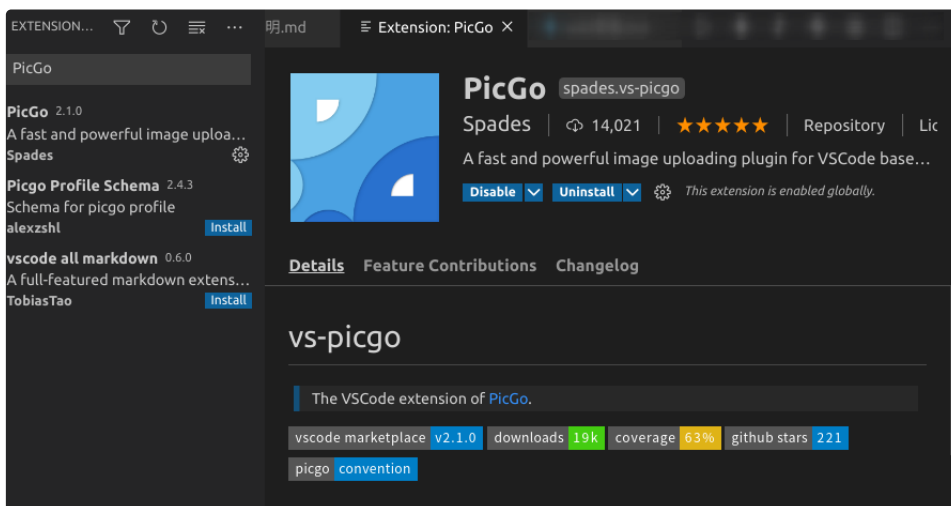
3. Markdown Shortcuts插件

提供markdown语法的快捷键



4. PicGo 插件 + github 图床

PicGo插件可以将本地图片直接上传至所需的图床, 并自动生成图片链接

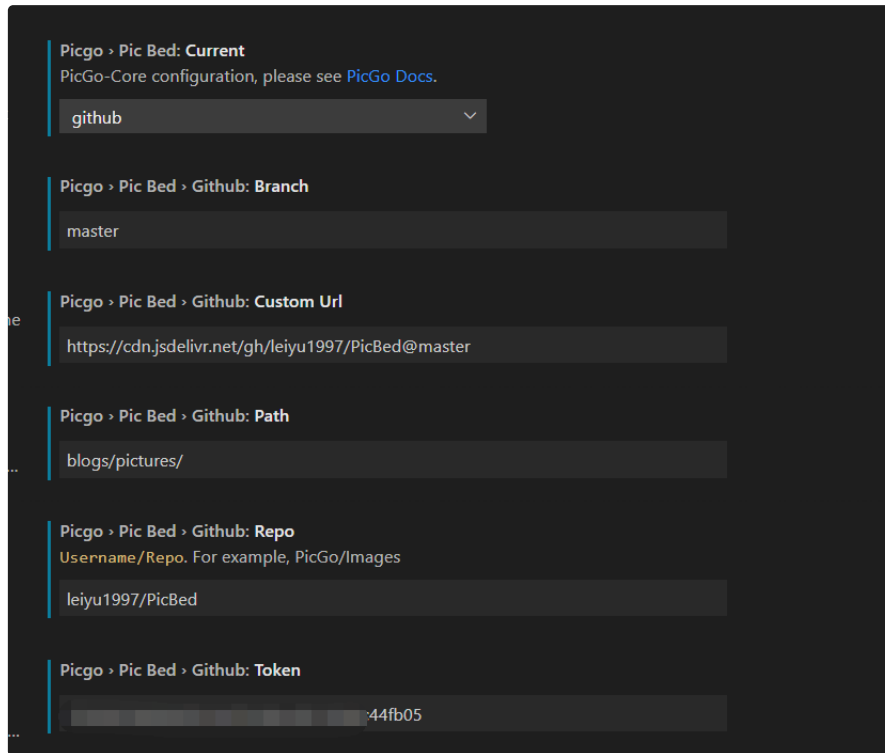


配置方法参考https://blog.csdn.net/qq_35621494/article/details/106432399, 需要注意:

- 新建的github的repo一定要为public

- 生成的访问token，要及时保存。此外该token可以访问帐号的所有repo，安全起见，建议新建专门帐号
- 建立repo后确认当前默认branch是master还是main

之后打开PicGo的配置选项



“

- 1.current设置为GitHub
- 2.Branch是我们仓库的分支，默认为master
- 3.custom url 是上传后图片在实际访问时使用的url，有两种方式可以使用
原生方式
-使用GitHub原生连接，格式为https://raw.githubusercontent.com/[用户名]/[仓库名]/[分支名]
cdn加速方式
-格式为https://cdn.jsdelivr.net/gh/[用户名]/[仓库名]@[分支名]
jsdelivr是免费的cdn中转商，cdn加速的优点是国内访问速度比较快
- 4.path是我们的图片存储在仓库中的路径，比如我的是blogs/pictures/
- 5.Repo是我们的仓库，比如我的是leiyu1997/PicBed
- 6.Token即为github用户设置-开发者选项内生成的token

”

ctrl+shift+E 为上传本地图片

5. Pandoc Latex格式文档打印

对于简单的markdown文档，可以通过chrome的pdf导出选项直接打印。但如果希望格式更加细致的话，可能就需要用到latex的文档打印。

latex是一种排版格式语言，和markdown类似，实现了内容和格式的分离，从而便于统一格式的传播。

为了让书写的markdown具有更适合打印阅读的格式，Markdown preview enhanced引入了Pandoc的支持。Pandoc是文档格式转换工具，可以完美的将markdown文件通过latex语法进行更加详细的格式化设计后，打印成pdf，从而包含目录，页眉，页尾，封面等额外信息，并且对数学公式的支持十分完备。

和前面的ditaa图形绘制模块已经预装在MPE中不同，Pandoc模块和latex的引擎都需要自行安装。

- Pandoc:
Pandoc官网有各个系统的安装包可以直接使用.
[Pandoc下载](#)
Pandoc下载好后, 还需要安装latex转换引擎和字库
- 中文字库:
如果是Ubuntu系统,安装好后, 此时Pandoc还是不支持中文的, 因此需要下载字库文件, 并根据后文的说明进行配置.字库的下载可以在ubuntu下直接执行 `sudo apt-get install texlive-latex-base texlive-fonts-recommended texlive-fonts-extra texlive-latex-extra` 安装texlive字库集.
安装好字库文件后, 可以通过 `fc-list :lang=zh-cn` 指令查看所有中文字库的标准名称.
如果想自行下载安装字库, 可以参考[字库安装](#)
- xelatex引擎:
pandoc支持多种转换引擎, 默认使用pdflatex, 但是使用中发现xelatex的中文支持要更好一些.
`sudo apt-get install texmaker` 安装引擎(其实之前的texlive-latex-base包可能已经包含了xelatex引擎).

之后就可以在markdown文档的头部配置信息中添加Pandoc的配置内容.

主要有两部分内容可以配置, Pandoc有一些选项, 可以自行生成对应的Latex模板. 当然更细致的自定义格式, 还是需要额外的latex模板实现.如下文配置中最后一句 `template: ./book.tex`.

需要特别注意的是, Pandoc的默认选项, 和模板的配置是有可能发生冲突的, 冲突时推荐把Pandoc的默认选项注释掉就行.

具体的选项说明可以参考: [Pandoc选项说明](#)

而模板的来源就比较多样了, 可以自行搜索 latex模板进行挑选.

示例选项:

```
---
#latex 选项
fontsize: 12pt
linkcolor: blue
urlcolor: green
citecolor: cyan
filecolor: magenta
toccolor: red
# geometry: margin=0.3in //后面的book.tex模板内也定义了geometry插件的格式, 会造成冲突, 因此这里注释掉
papersize: A4
documentclass: article
CJKmainfont: WenQuanYi Micro Hei Mono //字体也会造成冲突, 模板内对于中文字符, 英文字符等都可以分别设置字体.这里没有那么细致的要求,

title: "API Design"
author: Eric Luo
date: Dec 3, 2021
output:
  pdf_document:
    path: ./output/API设计指南.pdf
    toc: true
    toc_depth: 3
    # number_sections: true
    highlight: tango //代码块高亮
    latex_engine: xelatex
    template: ./book.tex //使用自定义的latex模板
---
```

通常latex模板内最需要定制的是页眉和页尾的内容, 这里通过latex的fancyhdr插件进行设置. 其内部预制了几种常用的格式, 在book.tex内如下代码:

```
\usepackage{fancyhdr}
\usepackage{lastpage}
\pagestyle{fancyplain} %使用fancyplain页眉页脚模板, 自带页眉内的段落标题
\renewcommand{\headrulewidth}{0.5pt} % 页眉分割线
\renewcommand{\footrulewidth}{0.5pt} % 页脚分割线
```

更加细致的fancyhdr插件设置可以参考:

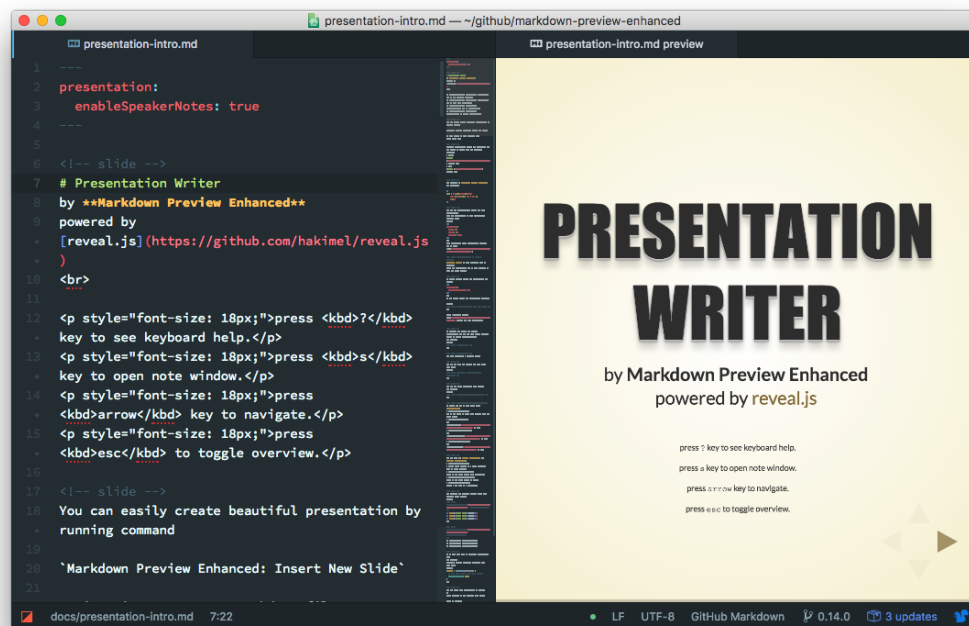
[fancyhdr官方文档](#)

[latex设置](#)

注意其中的单双页设置是跟documentclass的设置相关的, 如果是article模式, 那么是不分单双页的, 也就是只有单页和全局的设置会生效.

6. Reveal.js 幻灯片制作

MPE可以使用reveal.js进行文本的幻灯片渲染. reveal提供了一套幻灯片制作框架, 通过MPE封装的命令, 或者简单的html语句, 实现常用的幻灯片操作. 比如前后左右的翻页, 淡入淡出, 图片堆叠等.



- MPE默认已经打开了幻灯片的渲染功能. 每页幻灯片以 `<!-- slide -->` 作为分隔符. 常见的一些命令可以参考MPE官方提供的案例: [官方demo](#)
- 在markdown的 Front-Matter 头部配置选项可以对幻灯片的渲染进行部分设置.

```
---
presentation:
  theme: beige.css
  width: 1024
  height: 768
  controls: true
  progress: true
  margin: 0.1
  slideNumber: true
```



```

    keyboard: true
    center: true
    ---

@import "../format.less"

# title

```

具体支持的选项可以查阅: [MPE Front-matter设置](#)

- 其中@import是导入自定义的css格式文件

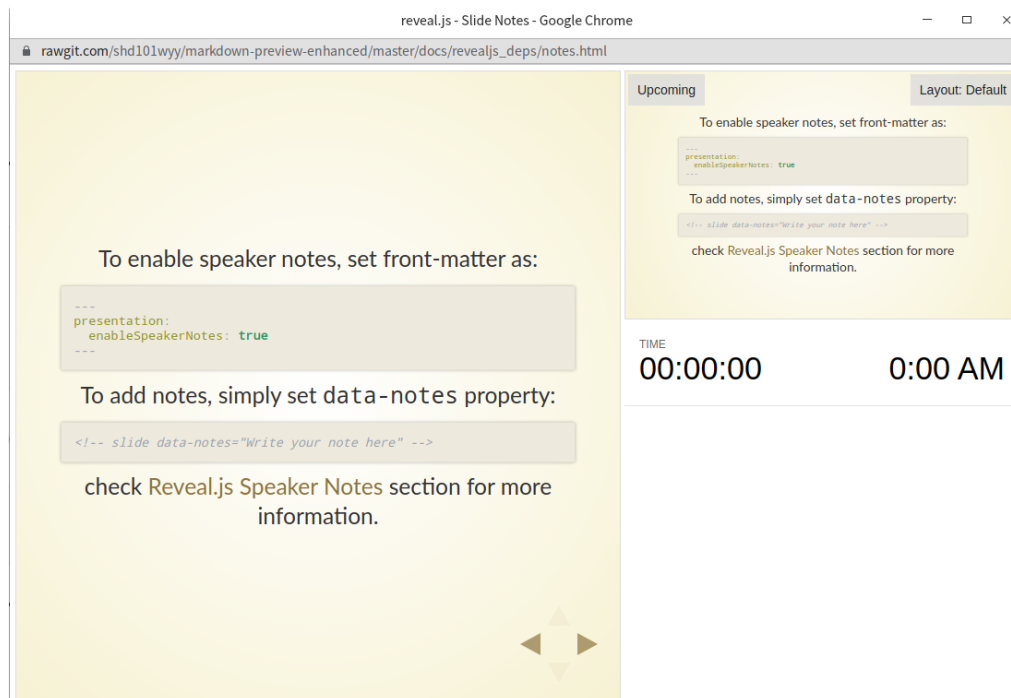
```

// custom presentation style
.reveal .slides {
  h4 { font-size: 120%; text-align: left; text-transform: none; }
  // modify all slides
}

// 自定义 presentation 样式
.slides > section:nth-child(1) {
  // 修改 `第 1 个幻灯片`
}

```

- MPE制作的幻灯片还具有演讲者模式, 通过 `<!-- slide data-notes="Write your note here" -->` 进行提示文本的编写.



- MEP针对文本支持简单的淡入淡出动画, 通过修饰语句实现.

```
- Item 1 <!-- .element: class="fragment" data-fragment-index="2" -->
```

- 更加多样化的动画配置, 比如针对图片的操作等, 可以通过直接在markdown内插入reveal.js 支持的html代码实现. 常用的有 fragments的淡入淡出. [fragments写法](#)
图片的堆叠:

基本规则: 1.首张图使用fade-out 2.前两张图使用相同的data-fragment-index 3.第二张开始全都使用current-visible, 直到最后一张改为普通的 fragment即可

```
<div class="r-stack">
  
  
  
</div>
```

- 同一级标题下的子页面可以通过 `<!-- slide vertical=true-->` 改为树状, 逻辑性更强.
- 导出时直接在preview页面右键选择html->cdn hosted即可



7. 参考资料

在 VSCode 下用 Markdown Preview Enhanced 愉快地写文档

vscode Markdown 预览样式美化

VSCode+PicGo+Github搭建Markdown图床

Latex模板参考