

File Edit Selection View Go Run Terminal Help 5.4.1.md - Notes - Visual Studio Code

EXPLORER

unit 5.md unit 7.md README.md 5.4.1.md x

Preview 5.4.1.md x

数学分析 > 5.4.1.md > # 多元函数的 Taylor 展开 > ## 一元函数的 Taylor 展开

1 # 多元函数的 Taylor 展开

2

3 ## 一元函数的 Taylor 展开

4

5
$$f(x) = \sum_{k=0}^n \frac{f^{(k)}(x_0)}{k!} (x - x_0)^k + R_n(x)$$

6

7 其中
$$R_n(x) = o((x - x_0)^n) = \frac{f^{(n+1)}(\xi)}{(n+1)!} (x - x_0)^{n+1}$$

8

9 五大展开公式:

10

11
$$e^x = 1 + x + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!} + o(x^n)$$

12

13
$$\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \cdots + \frac{(-1)^{n+1}}{n} x^n + o(x^n)$$

14

15
$$(1+x)^a = 1 + ax + \frac{a(a-1)}{2!} x^2 + \cdots$$

16

17
$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \cdots$$

18

19
$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \cdots$$

20

21 ## 多元函数的 Taylor 展开

22

23
$$z = f(x, y)$$

24

25 令
$$\varphi(t) = f(x_0 + t\Delta x, y_0 + t\Delta y), \varphi(0) = f(x_0, y_0) = f(z_0)$$

26

27
$$f(x_0 + \Delta x, y_0 + \Delta y) = \varphi(1)$$

28

29
$$\varphi(t) = \sum_{k=0}^n \frac{\varphi^{(k)}(0)}{k!} t^k + R_n(t)$$

30

31
$$f(x_0 + \Delta x, y_0 + \Delta y) = \sum_{k=0}^n \frac{\varphi^{(k)}(1)}{k!} + R_n(1)$$

Ln 9, Col 8 Spaces: 4 UTF-8 CRLF

教程向: 在 VS Code 中用 Markdown 做「数字化」学习笔记

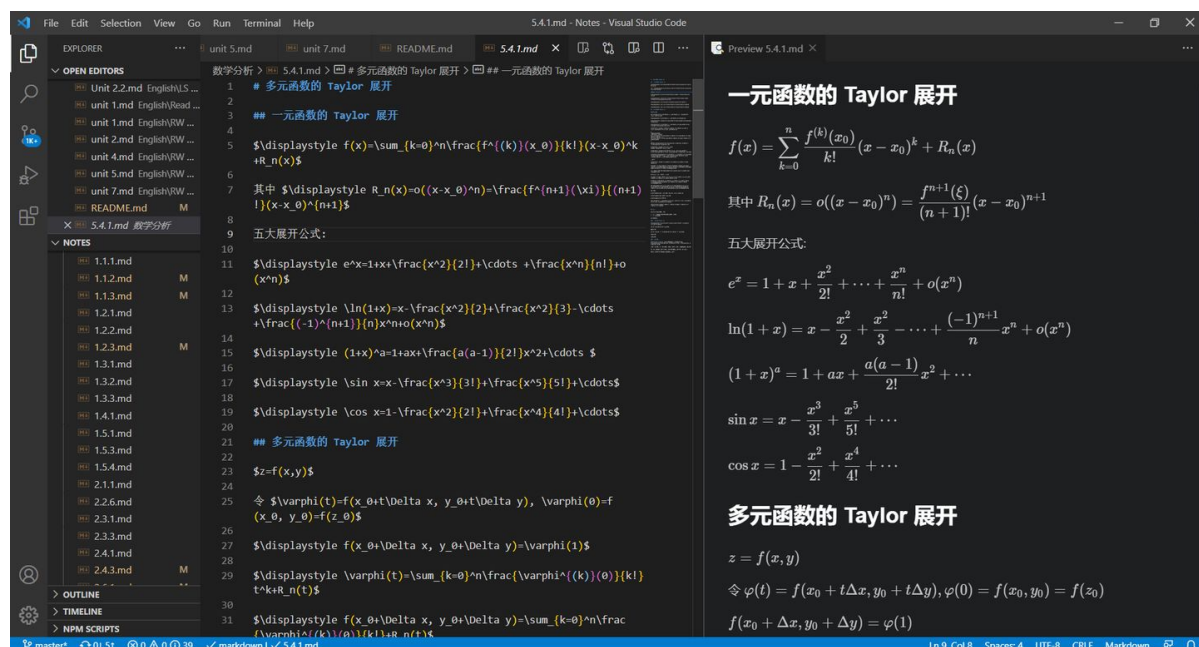


一只方橙

喜欢一切有趣的东西。GitHub: OrangeX4

743 人赞同了该文章

VS Code 配上 Markdown 语言, 就能写出酷炫, 便捷, 且适应互联网的「数字化」学习笔记.



File Edit Selection View Go Run Terminal Help 5.4.1.md - Notes - Visual Studio Code

EXPLORER

unit 5.md unit 7.md README.md 5.4.1.md x

Preview 5.4.1.md x

数学分析 > 5.4.1.md > # 多元函数的 Taylor 展开 > ## 一元函数的 Taylor 展开

1 # 多元函数的 Taylor 展开

2

3 ## 一元函数的 Taylor 展开

4

5
$$f(x) = \sum_{k=0}^n \frac{f^{(k)}(x_0)}{k!} (x - x_0)^k + R_n(x)$$

6

7 其中
$$R_n(x) = o((x - x_0)^n) = \frac{f^{(n+1)}(\xi)}{(n+1)!} (x - x_0)^{n+1}$$

8

9 五大展开公式:

10

11
$$e^x = 1 + x + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!} + o(x^n)$$

12

13
$$\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \cdots + \frac{(-1)^{n+1}}{n} x^n + o(x^n)$$

14

15
$$(1+x)^a = 1 + ax + \frac{a(a-1)}{2!} x^2 + \cdots$$

16

17
$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \cdots$$

18

19
$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \cdots$$

20

21 ## 多元函数的 Taylor 展开

22

23
$$z = f(x, y)$$

24

25 令
$$\varphi(t) = f(x_0 + t\Delta x, y_0 + t\Delta y), \varphi(0) = f(x_0, y_0) = f(z_0)$$

26

27
$$f(x_0 + \Delta x, y_0 + \Delta y) = \varphi(1)$$

28

29
$$\varphi(t) = \sum_{k=0}^n \frac{\varphi^{(k)}(0)}{k!} t^k + R_n(t)$$

30

31
$$f(x_0 + \Delta x, y_0 + \Delta y) = \sum_{k=0}^n \frac{\varphi^{(k)}(1)}{k!} + R_n(1)$$

Ln 9, Col 8 Spaces: 4 UTF-8 CRLF Markdown

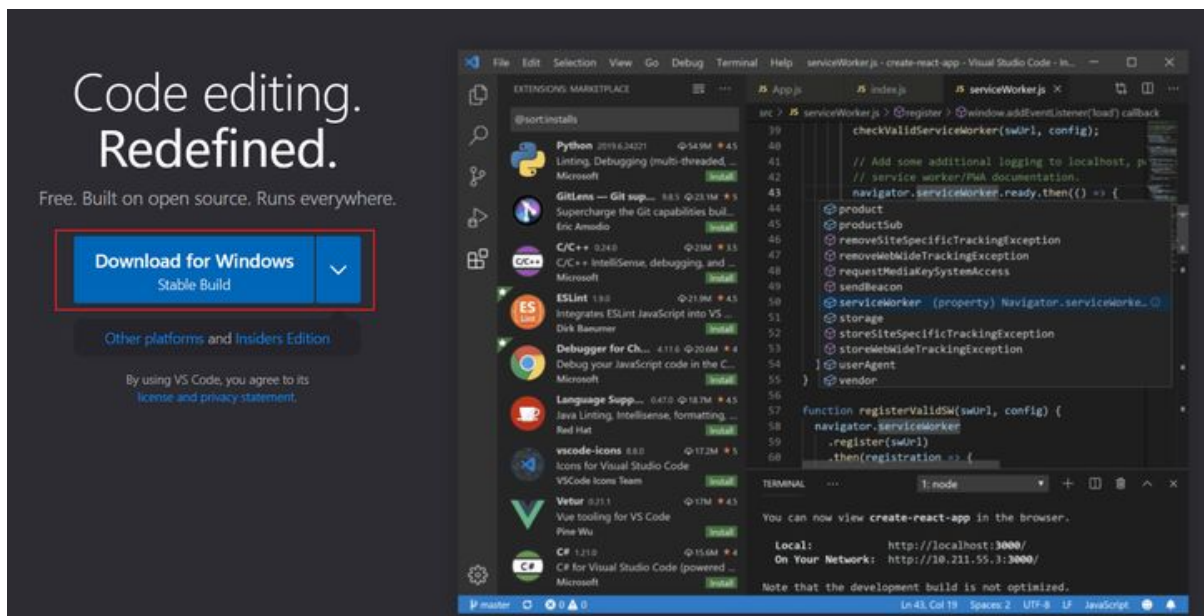
这个教程是我这段时间踩过各种各样的坑之后, 总结 VS Code 做学习笔记的最佳实践.



环境搭配

安装 VS Code

前往 [官网](#) 下载 VS Code, 点击 Download , 就会自动开始下载:



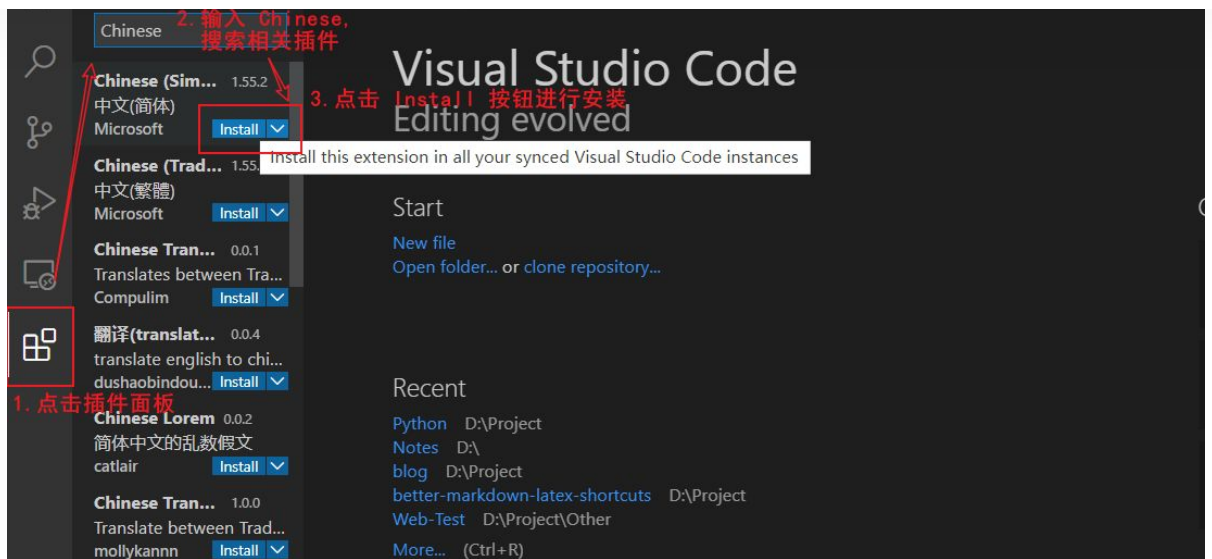
如果网速较慢, 你也可以在 [腾讯软件中心](#) 下载.

进入安装界面, 将选项全部勾选, 点 **下一步** 和 **安装**, 直到安装完成即可.

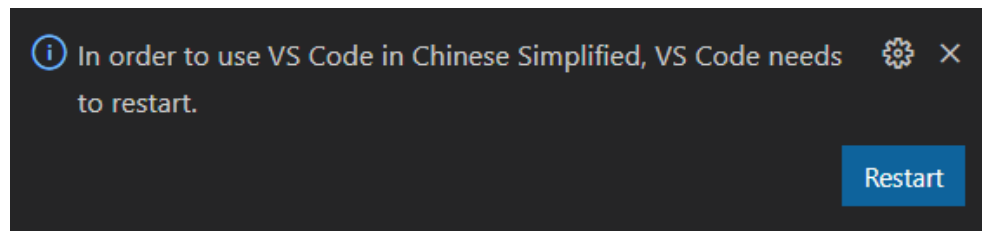
中文支持 (选做)

如果你不想看英文, 想要中文版本的 VS Code, 你可以安装一个中文支持插件. 虽然我个人不太喜欢, 但是每个人都有自己的喜好嘛.

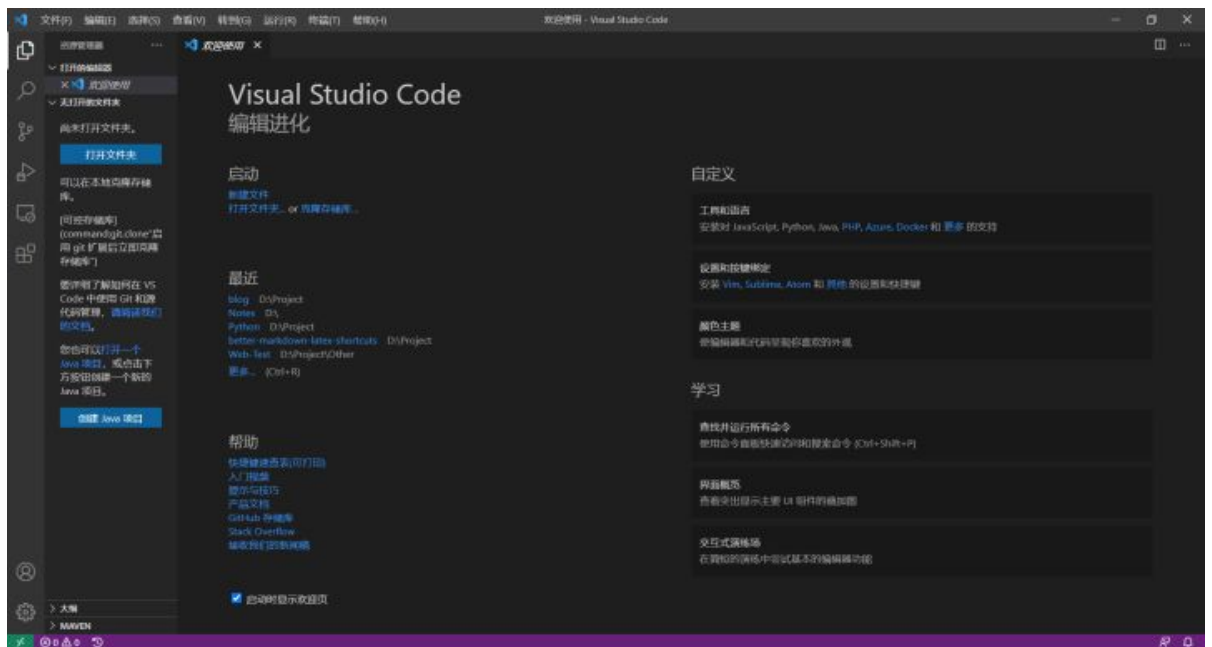
打开 VS Code, 点击插件面板, 输入 Chinese , 选择出现的第一个插件进行 Install 安装.



安装好后, 按照右下角提示重启 VS Code.



可以看出, 界面以及变为中文了.



安装 Markdown 相关插件

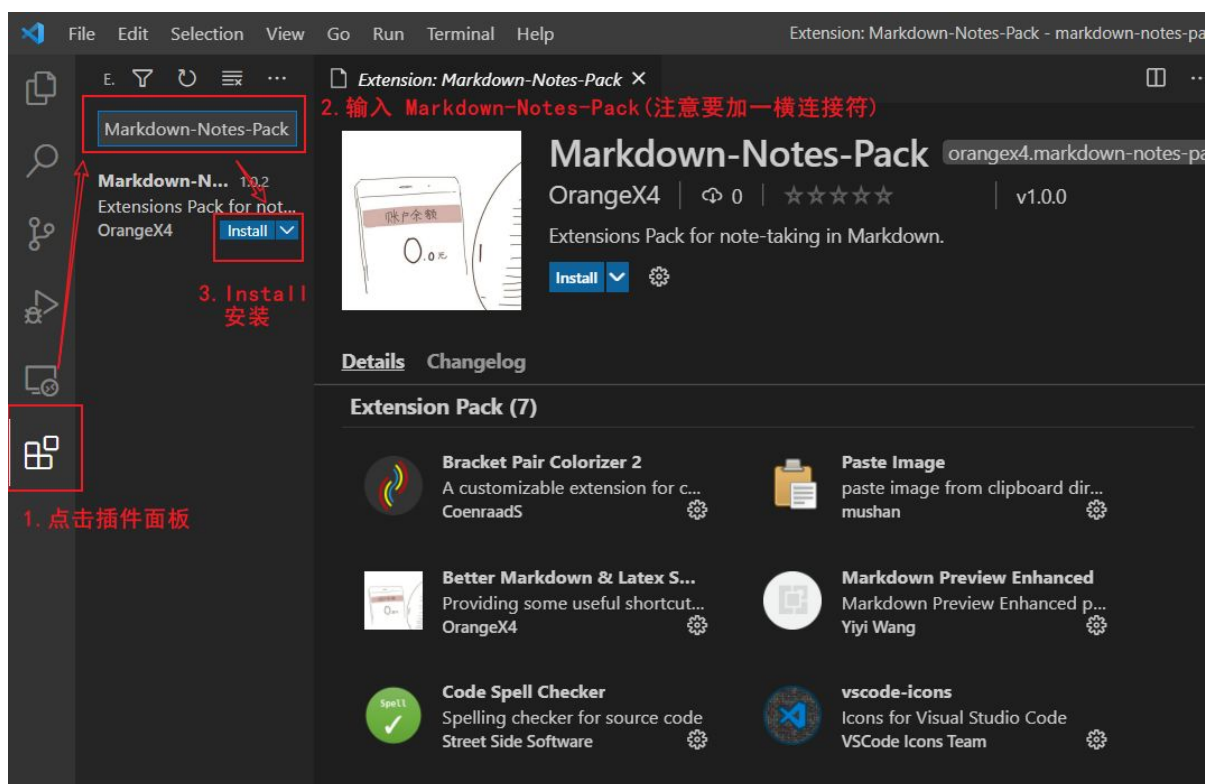
为了良好的记笔记体验, 你需要安装和 Markdown 相关的 VS Code 插件:

你可以下载我整理的这个 Extension Pack, 它包括了下面这些插件:



- Better Markdown & Latex Shortcuts
- Latex Sympy Calculator
- Python Brackets
- Orangex4 Snippets
- Paste Image
- Vscode Icons

就像安装中文支持那样操作, 点击插件面板, 搜索 `Markdown-Notes-Pack`, 点击 `Install` 进行安装, 可能会花一点时间.



第一个 Markdown 文件: Hello World!

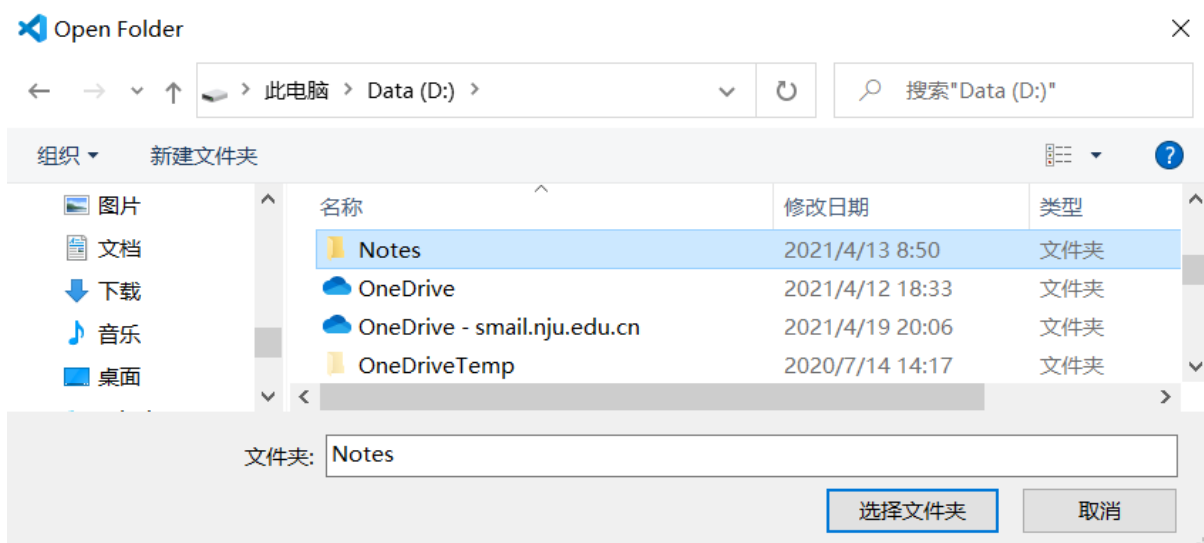
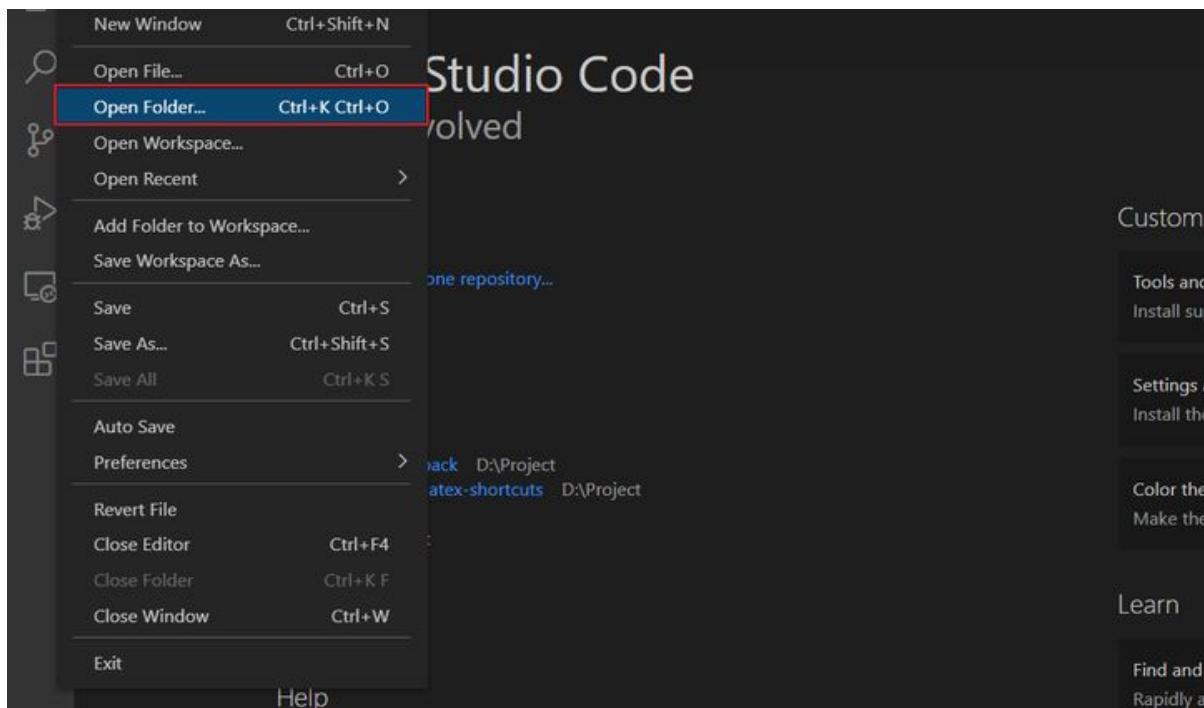
新建笔记工作区

工作区是什么?

工作区就是一个文件夹, 在你做笔记的时候, 工作区应该是你的笔记的根文件夹.

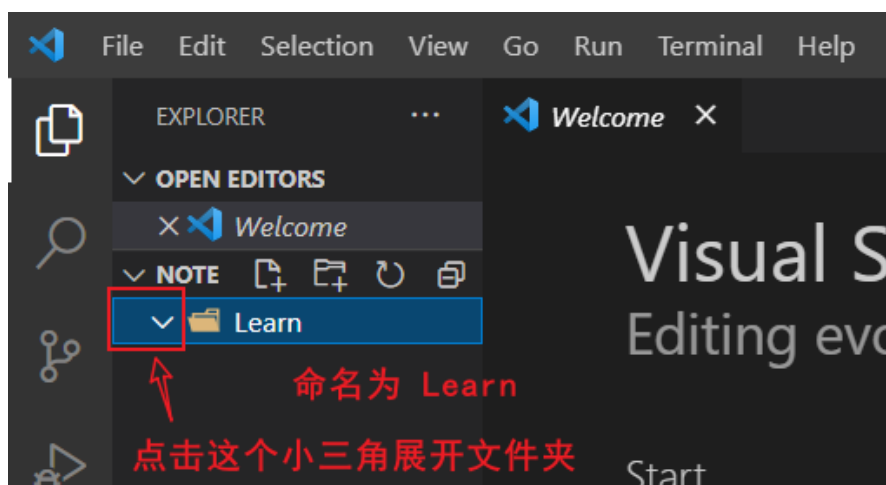
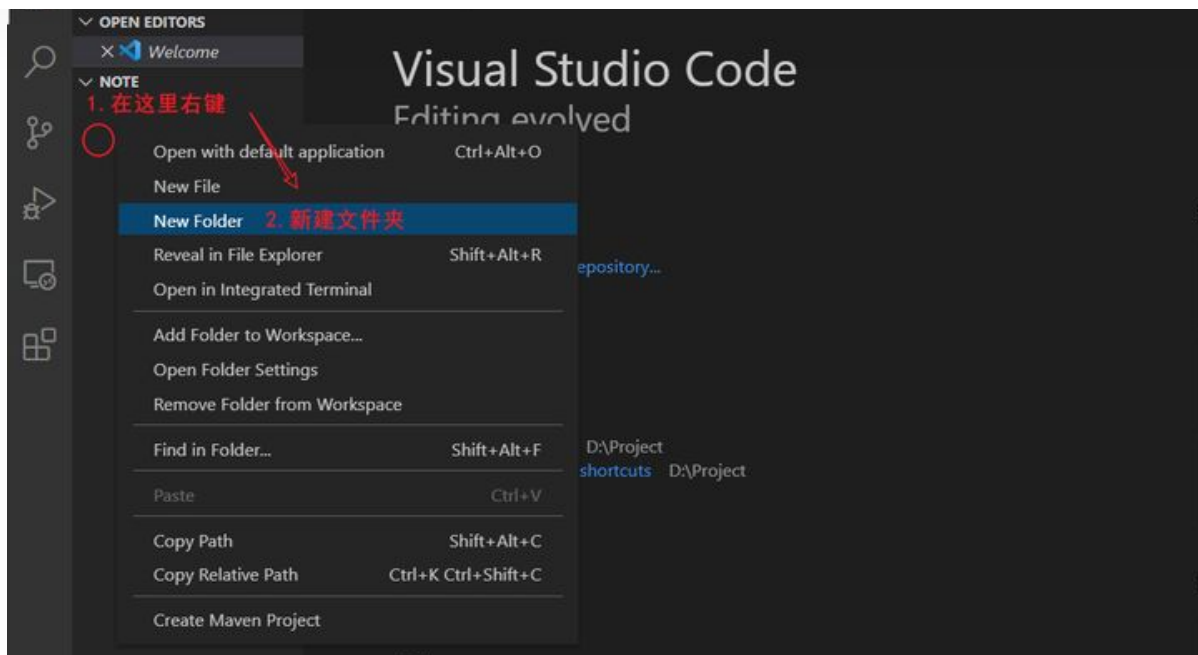
例如, 我在 D 盘创建了一个文件夹 `D:\Notes`, 那么我的工作区就应该是这个 `D:\Notes` 文件夹.

你可以像我一样, 在 D 盘新建一个 `Notes` 文件夹, 并在 VS Code 中打开这个文件夹.

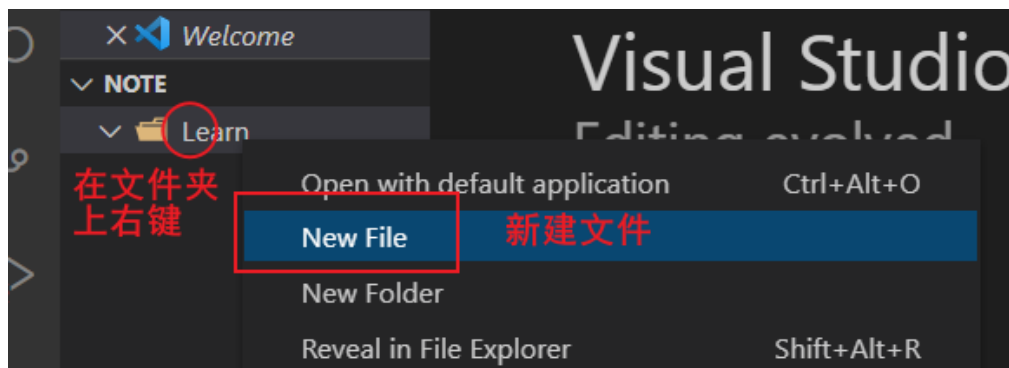


在工作区子笔记文件夹

我个人比较喜欢将笔记按照不同的课程放在不同的文件夹下, 比如在我们的 Markdown 学习之旅中, 我们可以先新建一个文件夹叫 Learn .



新建一个文件, 命名为 `test.md`, 注意后缀一定要是 `.md` !!! 这样表明它是一个 **Markdown** 文件. 你的所有笔记文件都应该以 `.md` 作为后缀.



写下第一行 Markdown 笔记

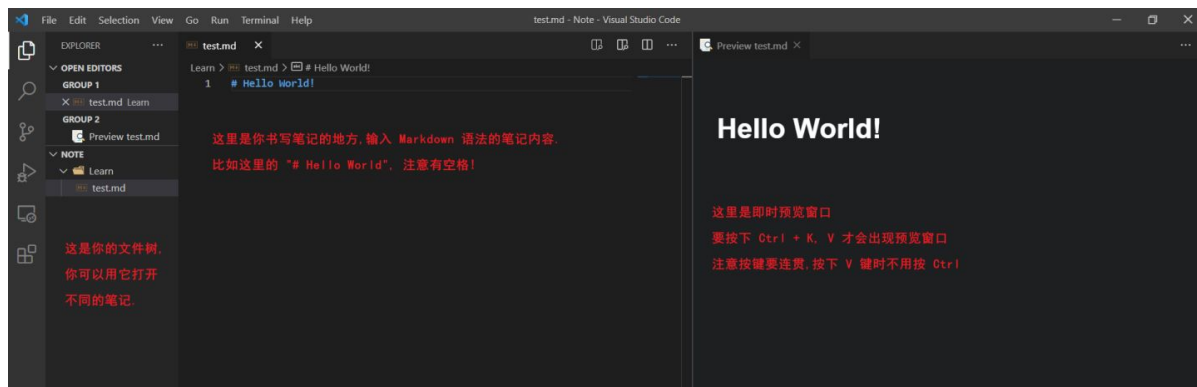
输入笔记内容:



(注意, # 和 文本之间要**加入空格**!)

然后**按照先后顺序**按下快捷键 `Ctrl + K, V`, 然后在右边就会出现预览窗口.

(注意, 快捷键按下时要**连贯**, 并且按下 `V` 的同时**不用按** `Ctrl` 键!)



如果正常地显示出了标题 "Hello, World!", 那么恭喜你, 第一个 Markdown 笔记写成功了!

修改预览窗口背景 (选做)

相信聪明的你已经发现了一个问题:

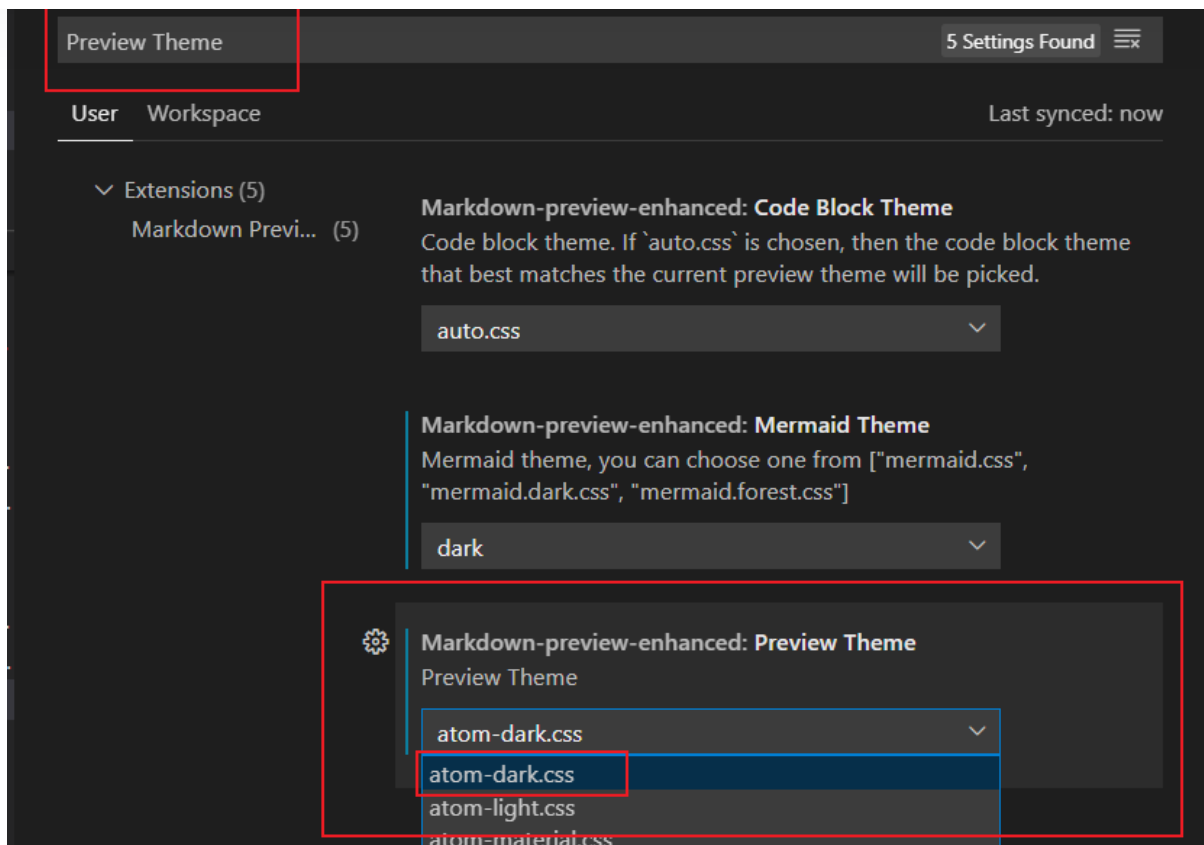
为什么我的预览窗口是白色的, 太刺眼了! 为什么你的预览窗口和代码窗口颜色一样?

诶, 别急, 这就告诉你. 这里的预览窗口的 **Markdown Preview Enhanced** 插件生成的. 你只需要修改一下这个插件的配置就可以了.

按下快捷键 `Ctrl + ,`, 就会打开设置窗口.

在搜索框输入 `Preview Theme`, 看见标题为 `Preview Theme` 的选项, 在下拉框中选择 `atom-dark.css`, 就能让预览窗口的颜色改变了.

(当然, 颜色还是和 VS Code 有一点差别, 就不要在意这些细节啦)



Markdown 语法

纯粹的 Markdown 语法其实非常简单, 70 行的语法介绍就可以写完.

一级标题

二级标题

三级标题

每写完一个段落要隔一行空行.

就像这样, 隔了一行空行.

分割线

****重点加粗****

斜体

~~删除线~~

列表:

* 无序列表



* 无序列表
* 无序列表

- 1. 有序列表 1
 - 1. 嵌套有序列表 1
 - 2. 嵌套有序列表 2
- 2. 有序列表 2
- 3. 有序列表 3

引用文本：

- > 引用别人说的话
- > 就这样写
- > By. OrangeX4

这是 `行内代码` 语法。

代码块语法：

```
''' python
print("Hello, World!")
'''
```

请将 ` ` 替换成 ` `。

[超链接名称](链接地址)

![图片提示语](图片地址)

表格：

表头 表头
---- ----
内容 内容
内容 内容

注释：

<!-- 你看不见我 -->

标题

你可以用



二级标题

三级标题

段落.

或

一级标题

=====

二级标题

的方式表示标题, 个人更推荐前一种方式. (不用对齐, 强迫症福音!)

段落

段落和段落之间要隔一行.

就像这样.

分割线

三条横线 (或更多的横线) 表示分割线.

就像这样.

三条横线 (或更多的横线) 表示分割线.

就像这样.

加粗, 斜体和删除线

做笔记的时候, 我们常常会有这样的要求, "加粗, 斜体和删除线", 作为笔记的

****重点加粗****

斜体



Markdown Preview Enhanced 拓展功能:

==高亮==

重点加粗

斜体

~~删除线~~

高亮

快捷键:

选中文本之后, 按下 `Ctrl + B` 可以给选中文本加粗.

同理 `Ctrl + I` 可以让选中文本变为斜体.

列表

列表是记笔记时非常基本的元素, 在 Markdown 中, 可以用很方便的格式书写列表:

- * 无序列表
 - * 嵌套无序列表
 - * 嵌套无序列表
- * 无序列表
- * 无序列表
- 1. 有序列表 1
 - 1. 嵌套有序列表 1
 - 2. 嵌套有序列表 2
- 2. 有序列表 2
- 3. 有序列表 3

Markdown Preview Enhanced 拓展功能:

任务列表:

- [x] 已经完成的事 1
- [x] 已经完成的事 2
- [x] 已经完成的事 3
- [] 仍未完成的事 4
- [] 仍未完成的事 5



- 嵌套无序列表
- 嵌套无序列表
- 无序列表
- 无序列表

- 1. 有序列表 1
 - 1. 嵌套有序列表 1
 - 2. 嵌套有序列表 2
- 2. 有序列表 2
- 3. 有序列表 3

- ☒ 已经完成的事 1
- ☒ 已经完成的事 2
- ☒ 已经完成的事 3
- ☐ 仍未完成的事 4
- ☐ 仍未完成的事 5

快捷键:

要进行**缩进** (书写嵌套列表), 你可以使用 VS Code 的快捷键 `Ctrl + [` 和 `Ctrl +]`.

这个快捷键可以将代码向左或向右进行缩进!

引用和代码

引用文本:

- > 引用别人说的话
- > 就这样写
- > By. OrangeX4

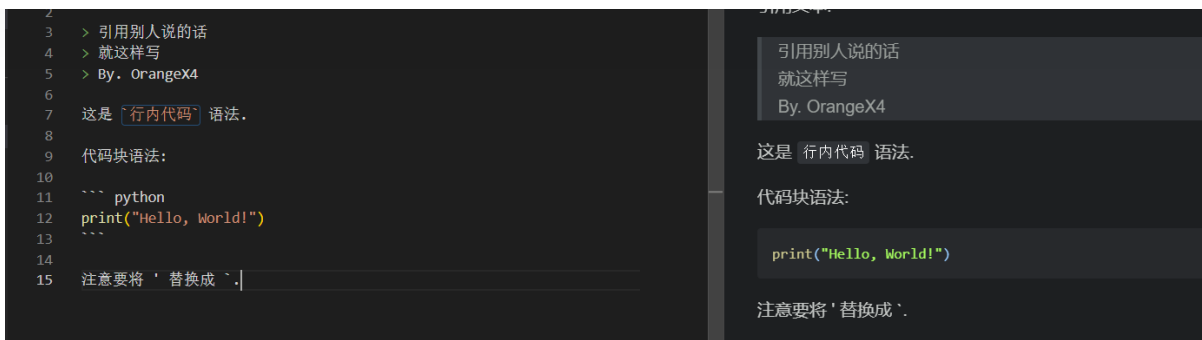
这是 ``行内代码`` 语法.

代码块语法:

```
''' python
print("Hello, World!")
'''
```

注意要将 `'` 替换成 ```.



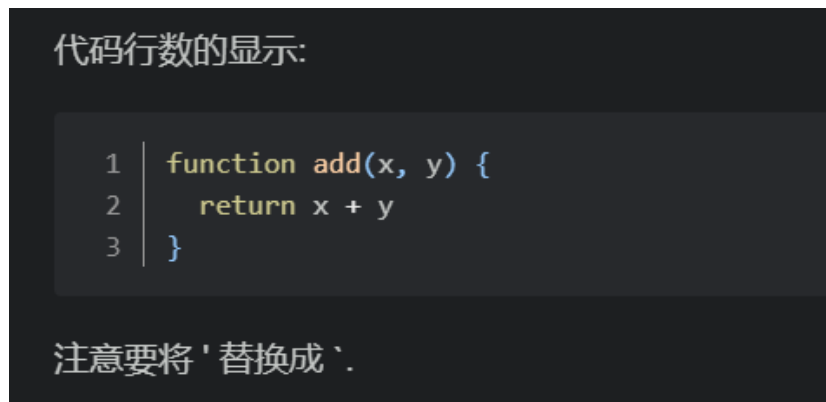


Markdown Preview Enhanced 拓展功能:

代码行数的显示:

```
''' javascript {.line-numbers}
function add(x, y) {
    return x + y
}
'''
```

注意要将 ' 替换成 `.



超链接和图片

你可以用下面的语法插入超链接和图片:

[超链接名称](链接地址)

![图片提示语](图片地址)

例如, 可以使用网址和图床:

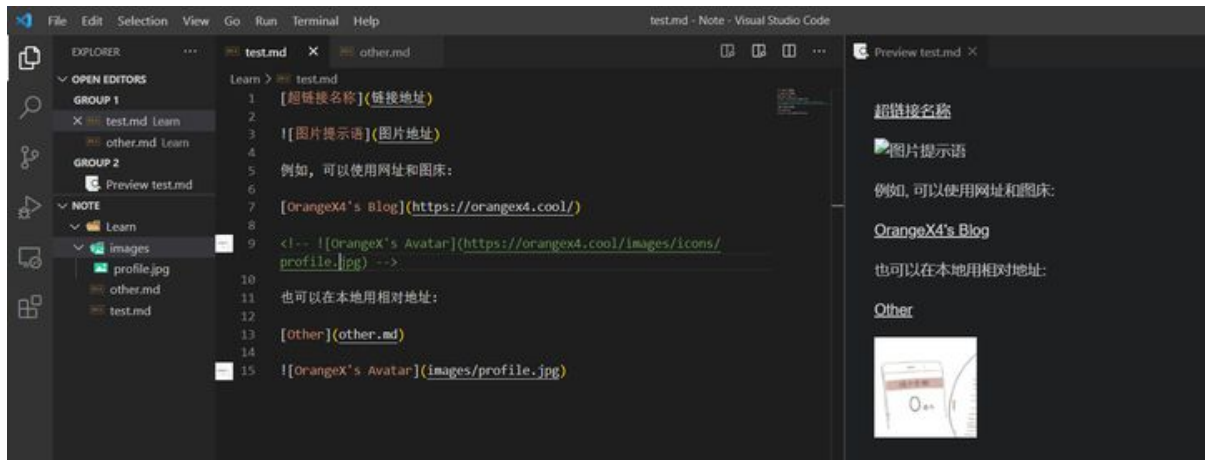
[OrangeX4's Blog](https://orangex4.cool/)

![OrangeX's Avatar](https://orangex4.cool/images/icons/profile.jpg)

也可以在本地图床用相对地址:



![OrangeX's Avatar](images/profile.jpg)



剪贴板图片插入

看了上面的插入图片语法, 也许你会觉得, **好麻烦啊**.

我就插入一张图片, 居然还要把图片上传到网上. 或者把图片保存到本地, 还要移动图片, 命名什么的.

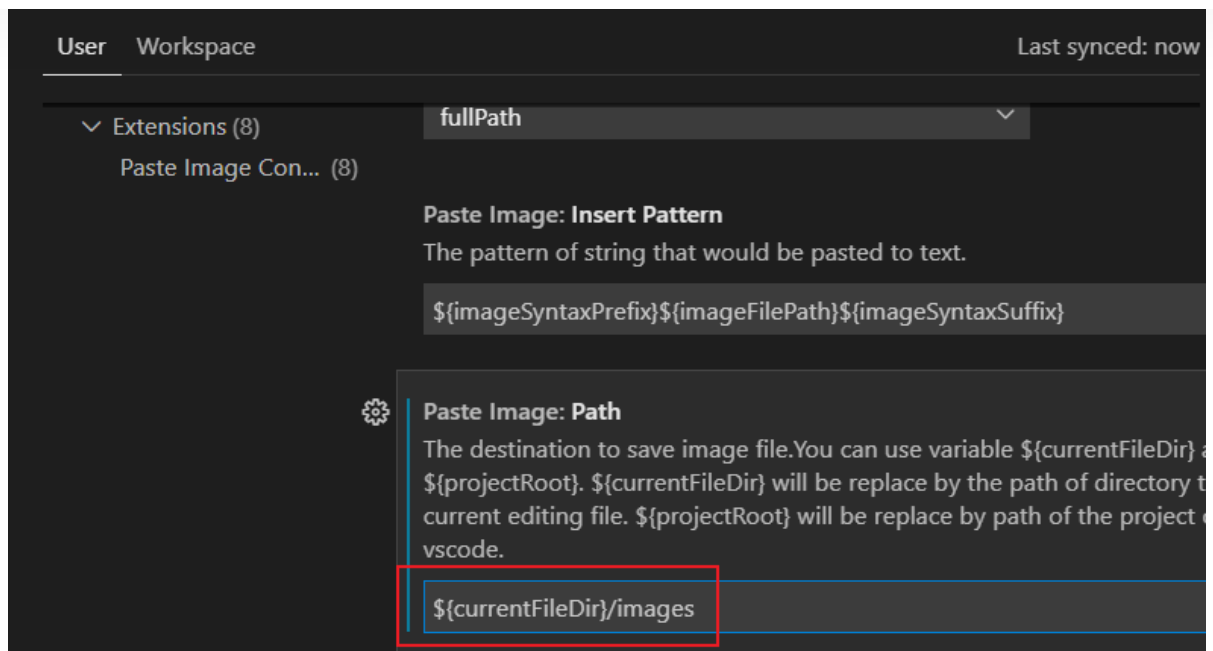
就不能像 Word 那样, 直接**剪贴板粘贴图片**吗?

万能的 VS Code 当然可以做到类似的事!

其实, 使用你已经安装了的 **Paste Image** 插件就行.

不过在使用之前, 你要做一点小调整:

按下 `Ctrl + ,`, 打开设置窗口, 输入 `Paste Image Path` 并搜索, 将框内的文本改成 `${currentFileDir}/images`.



注意标题是 **Paste Image: Path**, 不要弄错了!

设置好之后, 你就可以使用剪贴板粘贴功能了!

按下快捷键 `Ctrl + Alt + V`,

就能把图片自动保存到当前目录下, 并以正确的格式粘贴到当前的 Markdown 文件中.

表格

你可以很方便地在做笔记的时候加入表格:

表格:

	表头		表头	
	----		----	
	内容		内容	
	内容		内容	

Markdown Preview Enhanced 拓展功能:

拓展语法:

	表头		表头	
	----		----	
	内容		内容	
	>		内容	

	表头		表头	
	----		----	
	内容		内容	
	^		内容	

表格:

表头	表头
内容	内容
内容	内容

拓展语法:

表头	表头
内容	内容
内容	

表头	表头
内容	内容
	内容

快捷键:

自动表格对齐: Shift + Alt + F

注释

注释不会被渲染出来.

你可以随手做点草稿, 如果还想保留着, 但是不显示, 就可以按下**快捷键** `Ctrl + \` **将当前行注释 / 反注释**.

注释:

```
<!-- 你看不见我 -->
```

```
<!-- 多行注释  
就像这样 -->
```

有一点很重要的就是, **VS Code** 会在你每次修改代码之后, 重新渲染一遍.

如果有很多很多的数学公式, 渲染会很慢, 这时候有两个建议:

- 分成多个文件, 避免单文件过大!
- 将你暂时不看的部分注释掉, 加快渲染速度!



Markdown 的数学公式吸纳了大部分的 Latex 语法, 你可以以一种简单的方式在 VS Code 中书写数学公式.

行内公式:

单位圆 $x^2+y^2=1$

公式块:

```
$$
\begin{cases}
x=\rho\cos\theta \\
y=\rho\sin\theta
\end{cases}
$$
```

行内公式:

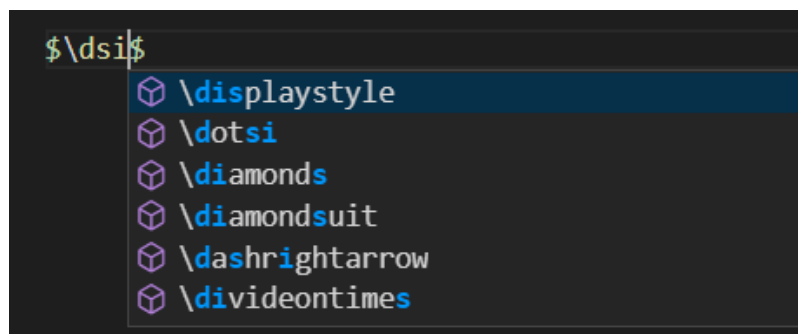
单位圆 $x^2 + y^2 = 1$

公式块:

$$\begin{cases} x = \rho \cos \theta \\ y = \rho \sin \theta \end{cases}$$

VS Code 有着非常便捷好用的自动补全功能, 只需要简单地打出你想打的内容的几个字母 (乱序也行), 再使用 $\uparrow \downarrow$ 进行选择, 最后按下回车就可以打出你想要的内容.

再使用 **HyperSnips** 插件, 就能够写得飞快.



还有, 不要在公式内使用中文, 除非是 `\text{中文}` (但是也不推荐)

1. 上标和下标

上标 $x_1 + y_{12} = 1$

上标 $x^2 + y^{12} = 1$

上标 $x_1 + y_{12} = 1$

HyperSnips 自动扩展：

xsr => $x^{\{2\}}$

xtp => $x^{\{\dots\}}$

x1 => x_1

xii => x_i

xsb => $x_{\{\dots\}}$

2. 分式

较小的行内行分数 $\frac{1}{2}$

展示型的分式 $\displaystyle\frac{x+1}{x-1}$

其中 `\displaystyle` 用于将行内展示转为块状展示.

较小的行内行分数 $\frac{1}{2}$

展示型的分式 $\frac{x+1}{x-1}$

HyperSnips 自动扩展：

1/ => $\frac{1}{\{\dots\}}$

(1 + 2)/ => $\frac{(1+2)}{\{\dots\}}$

// => $\frac{\{\dots\}}{\{\dots\}}$

3. 根式



开 n 次方 $\sqrt[n]{2}$

开平方 $\sqrt{2}$

开 n 次方 $\sqrt[n]{2}$

HyperSnips 自动扩展:

hsq => $\sqrt{\dots}$

4. 空格

数学公式中的 **空格和换行** 都会在编译时 **被忽略**，想要实现「空格」的效果，需要用特别的命令。

紧贴 $a!b$

没有空格 ab

小空格 $a\,b$

中等空格 $a\;b$

大空格 $a\; b$

quad 空格 $a\quad b$

两个 quad 空格 $a\quad\quad b$

紧贴 ab

没有空格 ab

小空格 $a\,b$

中等空格 $a\;b$

大空格 $a\; b$

quad 空格 $a\quad b$

两个 quad 空格 $a\quad\quad b$

5. 累加, 累乘和积分

累加 $\sum_{k=1}^n \frac{1}{k}$ $\quad \quad \quad \displaystyle \sum_{k=1}^n \frac{1}{k}$



累加 $\sum_{k=1}^n \frac{1}{k}$

累乘 $\prod_{k=1}^n \frac{1}{k}$

积分 $\int_0^1 x \mathrm{d}x \quad \iint_{D_{xy}} \quad \iiint_{\Omega_{xyz}}$

HyperSnips 自动扩展：

sum => `\sum_{...}`

prod => `\prod_{<n=1>}^{<\infty>} ...`

int => `\int`

dint => `\int_{<-\infty>}^{<\infty>} ... \mathrm{d}x`

Tips: 按下 Tab 键可以跳转光标。

6. 括号修饰

用 `\left` 和 `\right` 可以让括号适配内部大小

圆括号 $\displaystyle \left(\sum_{k=1}^n \frac{1}{k} \right)^2$

方括号 $\displaystyle \left[\sum_{k=1}^n \frac{1}{k} \right]^2$

花括号 $\displaystyle \left\{\sum_{k=1}^n \frac{1}{k} \right\}^2$

尖括号 $\displaystyle \left\langle \sum_{k=1}^n \frac{1}{k} \right\rangle^2$

圆括号 $\left(\sum_{k=1}^n \frac{1}{k}\right)^2$ 方括号 $\left[\sum_{k=1}^n \frac{1}{k}\right]^2$ 花括号 $\left\{\sum_{k=1}^n \frac{1}{k}\right\}^2$ 尖括号 $\left\langle\sum_{k=1}^n \frac{1}{k}\right\rangle^2$

HyperSnips 自动扩展:

```
@( => \left( ... \right)
```

```
@[ => \left[ ... \right]
```

```
@{ => \left\{ ... \right\}
```

```
@< => \left< ... \right>
```

```
set => \{ ... \}
```

7. 多行算式对齐

居中:

```


$$\begin{aligned}
y &= (x+5)^2 - (x+1)^2 \\
&= (x^2 + 10x + 25) - (x^2 + 2x + 1) \\
&= 8x + 24
\end{aligned}$$


```

左对齐:

```


$$\begin{aligned}
y &= (x+5)^2 - (x+1)^2 \\
&= (x^2 + 10x + 25) - (x^2 + 2x + 1) \\
&= 8x + 24
\end{aligned}$$


```



居中:

$$\begin{aligned}
 y &= (x+5)^2 - (x+1)^2 \\
 &= (x^2 + 10x + 25) - (x^2 + 2x + 1) \\
 &= 8x + 24
 \end{aligned}$$

左对齐:

$$\begin{aligned}
 y &= (x+5)^2 - (x+1)^2 \\
 &= (x^2 + 10x + 25) - (x^2 + 2x + 1) \\
 &= 8x + 24
 \end{aligned}$$

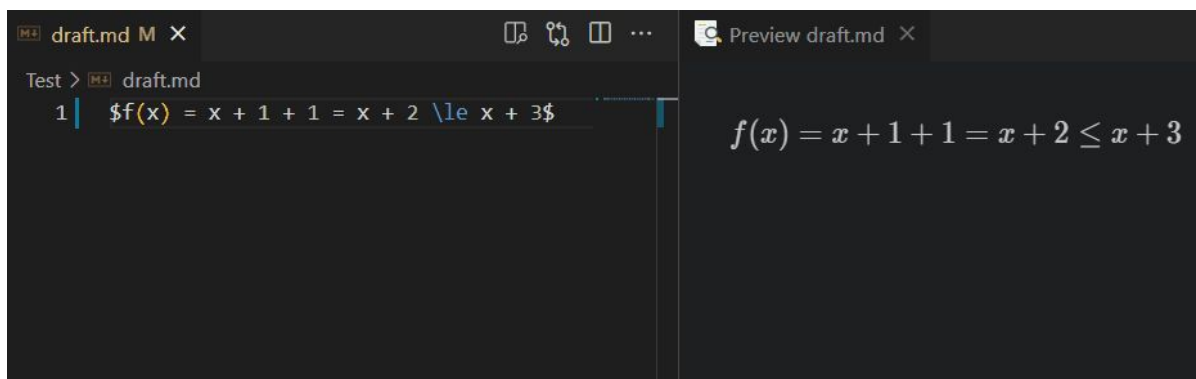
HyperSnips 自动扩展:

```

ali =>
\begin{aligned}
... \\
\end{aligned}

```

如果你安装了 `Better Markdown & Latex Shortcuts` 插件, 你还可以按下 `Shift + Ctrl + Alt + c` 可以将行公式转为 aligned 环境。



8. 方程组

```

$$
\begin{cases}
k_{11}x_1+k_{12}x_2+\cdots+k_{1n}x_n=b_1 \\
k_{21}x_1+k_{22}x_2+\cdots+k_{2n}x_n=b_2 \\
\cdots \\
k_{n1}x_1+k_{n2}x_2+\cdots+k_{nn}x_n=b_n
\end{cases}
$$

```

$$\begin{cases} k_{21}x_1 + k_{22}x_2 + \cdots + k_{2n}x_n = b_2 \\ \cdots \\ k_{n1}x_1 + k_{n2}x_2 + \cdots + k_{nn}x_n = b_n \end{cases}$$

HyperSnips 自动扩展：

```
case =>
\begin{cases}
... \\
\end{cases}
```

9. 矩阵

矩阵：

```
$$
\begin{pmatrix}
1 & 1 & \cdots & 1 \\
1 & 1 & \cdots & 1 \\
\vdots & \vdots & \ddots & \vdots \\
1 & 1 & \cdots & 1
\end{pmatrix}
```

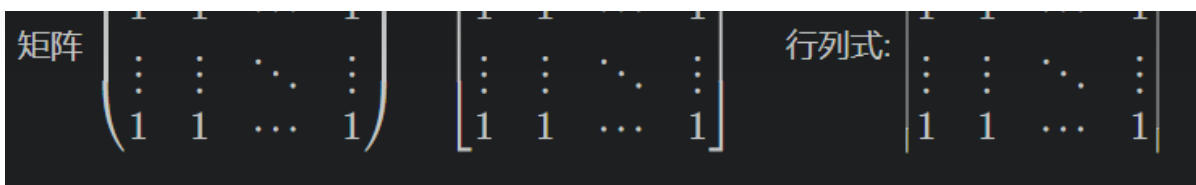
\quad

```
\begin{bmatrix}
1 & 1 & \cdots & 1 \\
1 & 1 & \cdots & 1 \\
\vdots & \vdots & \ddots & \vdots \\
1 & 1 & \cdots & 1
\end{bmatrix}
$$
```

行列式：

```
$$
\begin{vmatrix}
1 & 1 & \cdots & 1 \\
1 & 1 & \cdots & 1 \\
\vdots & \vdots & \ddots & \vdots \\
1 & 1 & \cdots & 1
\end{vmatrix}
$$
```





HyperSnips 自动扩展:

```
bmat2 => \begin{bmatrix} ... & ... \\ ... & ... \end{bmatrix}
```

```
vec2 => \begin{pmatrix} ... \\ ... \end{pmatrix}
```

tips: 按下 Tab 键可以切换到下一个位置。

10. 特殊字符

α	<code>\alpha</code>	κ	<code>\kappa</code>	ψ	<code>\psi</code>	F	<code>\digamma</code>	Δ	<code>\Delta</code>	Θ	<code>\Theta</code>
β	<code>\beta</code>	λ	<code>\lambda</code>	ρ	<code>\rho</code>	ε	<code>\varepsilon</code>	Γ	<code>\Gamma</code>	Υ	<code>\Upsilon</code>
χ	<code>\chi</code>	μ	<code>\mu</code>	σ	<code>\sigma</code>	\varkappa	<code>\varkappa</code>	Λ	<code>\Lambda</code>	Ξ	<code>\Xi</code>
δ	<code>\delta</code>	ν	<code>\nu</code>	τ	<code>\tau</code>	φ	<code>\varphi</code>	Ω	<code>\Omega</code>	\aleph	<code>\aleph</code>
ϵ	<code>\epsilon</code>	ω	<code>\omega</code>	θ	<code>\theta</code>	ϖ	<code>\varpi</code>	Φ	<code>\Phi</code>	\beth	<code>\beth</code>
η	<code>\eta</code>	ϕ	<code>\phi</code>	v	<code>\upsilon</code>	ϱ	<code>\varrho</code>	Π	<code>\Pi</code>	\daleth	<code>\daleth</code>
γ	<code>\gamma</code>	π	<code>\pi</code>	ξ	<code>\xi</code>	ς	<code>\varsigma</code>	Ψ	<code>\Psi</code>	\gimel	<code>\gimel</code>
ι	<code>\iota</code>			ζ	<code>\zeta</code>	ϑ	<code>\vartheta</code>	Σ	<code>\Sigma</code>		

更多特殊符号可以 [上网查询](#)

可以搜索 "Latex 符号表"

HyperSnips 自动扩展:

```
alpha => \alpha
```

```
Sigma => \Sigma
```

11. 公式编号与引用

```
$$
x+2 \tag{1.2}
$$
```

```
$$
\begin{equation}
x^n+y^n=z^n
\end{equation}
$$
```

由公式 (1.2) 可得到结论



2	\$\$	$x + 2$	(1.2)
3	x+2 \tag{1.2}		
4	\$\$	$x^n + y^n = z^n$	(1)
5			
6	\$\$		
7	\begin{equation}	由公式 (1.2) 可得到结论	
8	x^n+y^n=z^n		
9	\end{equation}		
10	\$\$		
11			
12	由公式 \$(1.2)\$ 可得到结论		
13			
14			知乎 @一只方橙

12. 零碎的重要语法

- 点乘 \cdot , 叉乘 \times , 异或 \otimes , 直和 \oplus , 加减 \pm , 复合 \circ .
- 小于等于 \leq , 大于等于 \geq , 不等 \neq , 恒等 \equiv , 约等 \approx , 等价 \cong , 相似 \sim , 相似等于 \simeq , 点等 \doteq .
- 逻辑与 \wedge , 逻辑或 \vee , 逻辑非 \neg , 蕴涵 \rightarrow , 等价 \leftrightarrow .
- 因为 \because , 所以 \therefore , 存在 \exists , 任意 \forall .
- 左小箭头 \leftarrow , 右小箭头 \rightarrow , 左大箭头 \Leftarrow , 右大箭头 \Rightarrow , 右长箭头 $\xrightarrow{fgh}\{abcde\}$.
- 属于 \in , 包含于 \subset , 真包含于 \subseteq , 交 \cap , 并 \cup , 空集 \emptyset .
- 短向量 \vec{x} , 长向量 \overrightarrow{AB} , 上横线 \overline{p} .
- 无限 ∞ , 极限 \lim , 微分 d , 偏导 ∂ , 点求导 \dot{y} , 点二阶导 \ddot{y} , 变化量 Δ , 梯度 ∇ .
- 横省略 \cdots , 竖省略 \vdots , 斜省略 \ddots .
- 常见函数 \sin , \cos , \tan , \arcsin , \arccos , \arctan , \ln , \log , \exp .

- 点乘 \cdot , 叉乘 \times , 异或 \otimes , 直和 \oplus , 加减 \pm , 复合 \circ .
- 小于等于 \leq , 大于等于 \geq , 不等 \neq , 恒等 \equiv , 约等 \approx , 等价 \cong , 相似 \sim , 相似等于 \simeq , 点等 \doteq .
- 逻辑与 \wedge , 逻辑或 \vee , 逻辑非 \neg , 蕴涵 \rightarrow , 等价 \leftrightarrow .
- 因为 \because , 所以 \therefore , 存在 \exists , 任意 \forall .
- 左小箭头 \leftarrow , 右小箭头 \rightarrow , 左大箭头 \Leftarrow , 右大箭头 \Rightarrow , 右长箭头 $\xrightarrow{abcde}_{fgh}$.
- 属于 \in , 包含于 \subset , 真包含于 \subseteq , 交 \cap , 并 \cup , 空集 \emptyset .
- 短向量 \vec{x} , 长向量 \overrightarrow{AB} , 上横线 \overline{p} .
- 无限 ∞ , 极限 \lim , 微分 d , 偏导 ∂ , 点求导 \dot{y} , 点二阶导 \ddot{y} , 变化量 Δ , 梯度 ∇ .
- 横省略 \cdots , 竖省略 \vdots , 斜省略 \ddots .
- 常见函数 \sin , \cos , \tan , \arcsin , \arccos , \arctan , \ln , \log , \exp .

```

** => \cdot
xx  => \times
otimes => \otimes
<=  => \le
!=   => \neq
==   => \equiv
~~   => \thickapprox
sim  => \sim
land => \land
lor  => \lor
bec  => \because
thr  => \therefore
EE   => \exists
AA   => \forall
inn  => \in
sse  => \subseteq
sqsubseteq => \sqsubseteq
cap  => \cap
cup  => \cup
empty => \emptyset
oo   => \infty
lim  => \lim_{<n> \to <\infty>}
dd   => \mathrm{d}
part => \frac{\partial <V>}{\partial <x>}
Delta => \Delta
nabla => \nabla
...   => \cdots
sin   => \sin

```

还有特别重要的数集、向量、矩阵符号：

```

RR  => \mathbb{R}
NN  => \mathbb{N}
txt  => \text{...}
xbar => \bar{x}
xhat => \hat{x}
xhvec => \vec{x}
xhdot => \dot{x}
Xbb  => \mathbb{X}
Xbs  => \boldsymbol{X}
Xbm  => \bm{X}
Xbf  => \mathbf{X}
Xsf  => \mathsf{X}
Xcal => \mathcal{X}
Xfrak => \mathfrak{X}
Xrm  => \mathrm{X}

```

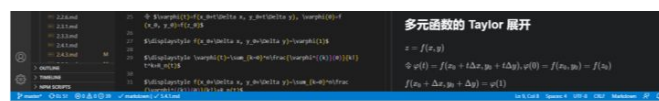
输出为 PDF

我们在右侧预览栏邮件菜单, 选择 **在浏览器中打开**.





在浏览器中右键选择 **打印**。



这个教程是我这段时间踩过各种各样的坑之后，总结 VSCode 做学习笔记的最佳实践。

如果你对在 VSCode 中做笔记有什么好的主意，欢迎留下你的意见，我会第一时间在教程中加入！

环境搭配

安装 VSCode

前往 [官网](#) 下载 VSCode，点击 [Download](#)，就会自动开始下载：

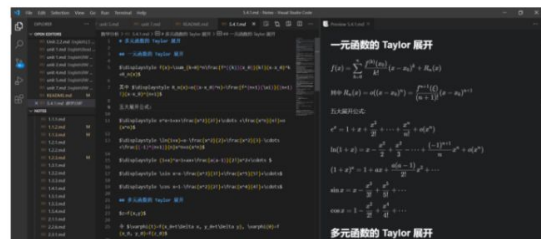


就能够 **输出为 PDF** 了。



教程向：在 VSCode 中用 Markdown 做「数字化」学习笔记

VSCode 配上 **Markdown** 语言，就能写出酷炫，便捷，且适应互联网的「数字化」学习笔记。

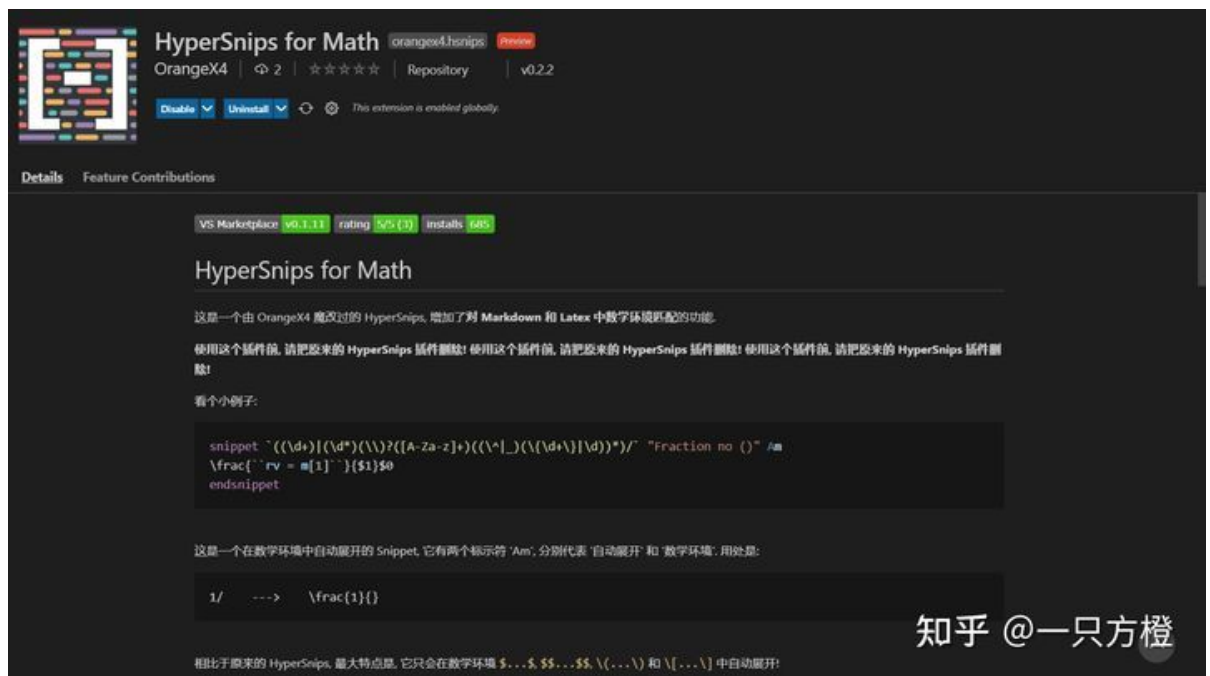


类似 UltiSnips 的自动补全

也许你曾经看过这篇文章：

[机器之心：世界上最好的编辑器Vim：1700多页数学笔记是如何实时完成的](#)
于是你很想试一下，然后就被 Vim 和 Latex 给劝退了。



插件安装: [HyperSnips for Math - Visual Studio Marketplace](#)

安装完成后, 按下快捷键 `Ctrl + Shift + P`, 输入命令 `Open Snippets Directory`, 就可以打开一个文件夹. 在该文件夹新建一个文件 `markdown.hsnips`, 并将 [这个网址](#) 里面的内容输入进去, 保存, 就可以使用了.

例如, 你可以在数学环境中输入: $(1+2)/$, 它会自动变为 `\frac{1+2}{}`.

你也可以创建自己的 Snippets, 按照格式, 发挥想象来写就好!

先看个 **普通例子**:

```
snippet RR "R" iAm
\mathbb{R}
endsnippet
```

这是一个在数学环境中自动展开的 Snippet, 它有三个标示符 `iAm`, 分别代表 "在词语内部也会触发", "自动展开" 和 "数学环境".

这个例子会在数学环境内, 自动将 `RR` 展开成为 `\mathbb{R}`, 代表 "实数".

再看个 **正则表达式** 的例子:

```
snippet `((\\d+)|(\\d*)(\\|\\|)?([A-Za-z]+)((\\^|_|)(\\{\\d+\\}|\\d))*)/` "Fraction no ()" Am
\\frac{`rv = m[1]`}{`$1`}$0
endsnippet
```

其中 `rv = m[1]` 是 JavaScript 代码, 表示将正则表达式的第一个组 `m[1]` 输出给 "返回值" `rv`, 然后输出出去.

1/ ---> \frac{1}{{}}

相比于原来的 HyperSnips, 最大特点是, 它只会在数学环境 \dots , \dots , \dots 和 \dots 中自动展开!

还有 $\{\text{VISUAL}\}$ 语法:

```
snippet fr "frac" iAm
\\frac{${1:${VISUAL}}}{${2}}
endsnippet
```

这个语法会保存最近选中的内容, 然后替换掉 $\{\text{VISUAL}\}$ 部分.

要开启在 markdown 下的 **自动补全提示**, 请使用 Shift + Ctrl + P 然后输入 open settings json 打开配置文件, 然后加入以下部分:

```
"[markdown]": {
  "editor.quickSuggestions": true
},
```

VS Code 快捷键

快捷键是用 VS Code 记笔记的**精髓**之一, 每一个都**非常重要**!

请务必**认真记住**这些快捷键!!!

原生快捷键

1. 通用操作

1. Ctrl + C, 复制当前文本
2. Ctrl + V, 粘贴当前文本
3. Ctrl + Z, 撤销
4. Ctrl + Shift + Z, 反撤销
5. Shift + Alt + F, 整理代码
6. Ctrl + /, 将当前行注释 / 反注释, **当多行文本被选中时**, 将多行文本注释

1. 光标操作

1. Ctrl + ← 将光标向左移动一个**单词**
2. Ctrl + → 将光标向右移动一个**单词**
3. Ctrl + Alt + ↑, 向上加入一个光标
4. Ctrl + Alt + ↓, 向下加入一个光标
5. Ctrl + Alt + U, 撤销上次光标操作

1. 界面移动



1. 选中操作

1. Shift + ← 向左选中 / 反选中一个**字符**(重要)
2. Shift + → 向右选中 / 反选中一个**字符**(重要)
3. Ctrl + Shift + ← 向左选中 / 反选中一个**单词**(重要)
4. Ctrl + Shift + → 向右选中 / 反选中一个**单词**(重要)
5. Ctrl + D **当前有选中文本时**, 将下一个与其相同的文本加入选中 (重要)

1. 文本行操作

1. Ctrl + C **当前无选中文本时**, 复制当前行
2. Shift + Alt + ↑ 向上**复制**当前行, **当多行文本被选中时**, 向上复制多行 (重要)
3. Shift + Alt + ↓ 向下**复制**当前行, **当多行文本被选中时**, 向下复制多行 (重要)
4. Alt + ↑ 向上**移动**当前行, **当多行文本被选中时**, 将当前多行文本向上移动 (重要)
5. Alt + ↓ 向下**移动**当前行, **当多行文本被选中时**, 将当前多行文本向下移动 (重要)

插件增加的快捷键

1. Markdown 语法

1. Ctrl + B **当前有选中文本时**, 将文本加粗
2. Ctrl + I **当前有选中文本时**, 将文本变为斜体
3. Ctrl + M 进入数学公式模式 (加入美元符)

1. 图片粘贴

1. Ctrl + Alt + V 粘贴剪贴板图片 (本地)
2. Ctrl + Alt + V 粘贴剪贴板图片 (图床)

1. 光标操作

1. Ctrl + Alt + U 将多选光标变为单选

1. 选中操作

1. Shift + Alt + ← 向左**复制**当前选中文本 (重要)
2. Shift + Alt + → 向右**复制**当前选中文本 (重要)
3. Alt + ← 向左**移动**当前选中文本一个**字符**(重要)
4. Alt + → 向右**移动**当前选中文本一个**字符**(重要)
5. Ctrl + Alt + ← 向左**移动**当前选中文本一个**单词**(重要)
6. Ctrl + Alt + → 向右**移动**当前选中文本一个**单词**(重要)

1. 计算器功能

1. Ctrl + Shift + Alt + E 计算当前选中表达式, **用等号连接并输出**
2. Ctrl + Shift + Alt + R 计算当前选中表达式, **并替换当前选中表达式**
3. Ctrl + Shift + Alt + D 定义当前选中表达式, **无输出**



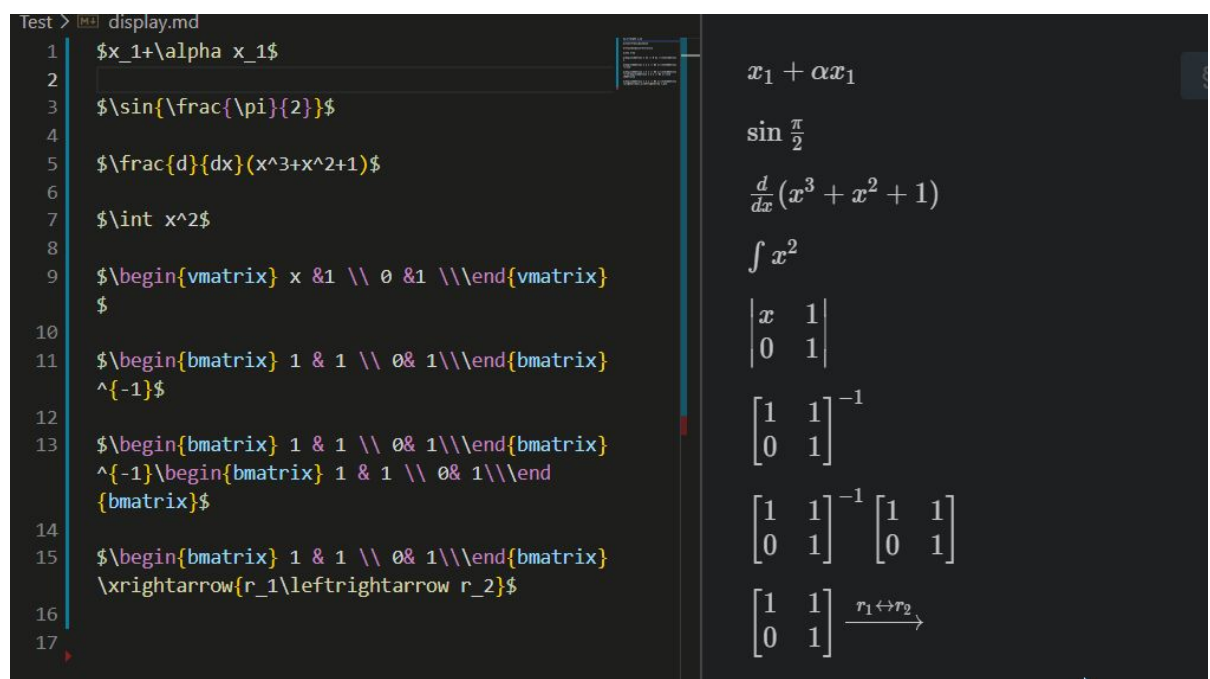
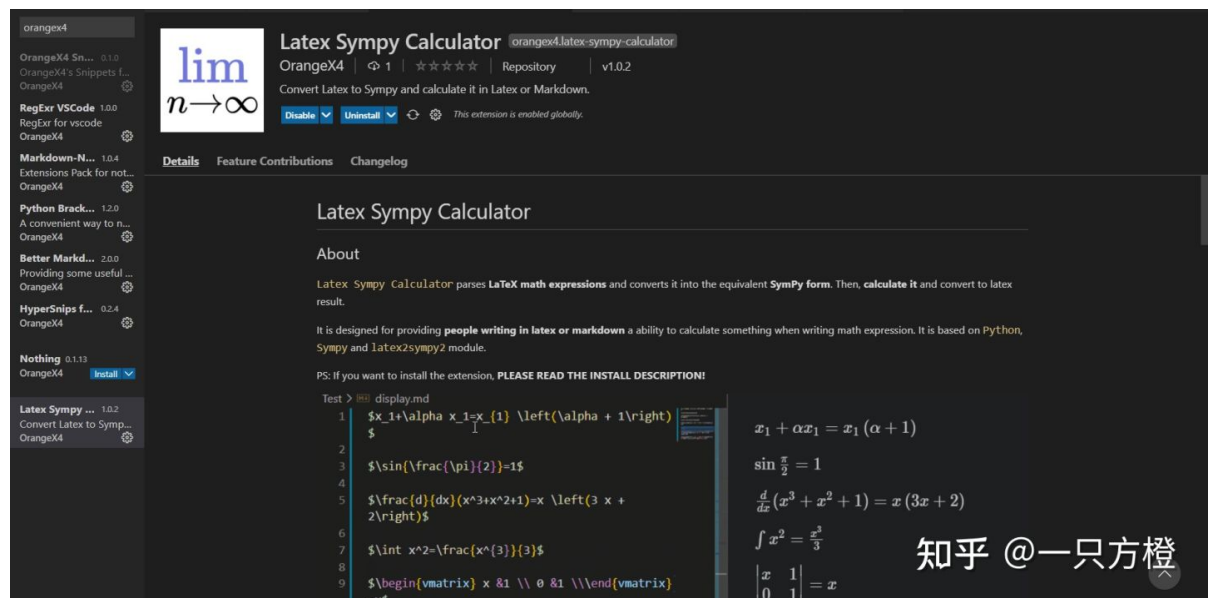
使用我开发的一个插件 Latex Markdown Calculator , 可以很方便地在 VS Code 中用 Latex 公式语法进行科学计算。

即选中一段数学表达式, 按下一个快捷键就能帮你计算出结果, 一些简单的计算再也不用打开电脑自带的计算器和 Matlab~

因为是基于 Python Sympy 的, 所以现在它已经有很多强大的功能了。

比如说: 符号运算, 求积分, 求微分, 基于矩阵的符号运算(如行列式, 点乘, 转置, 求逆等功能), 解方程...

快捷键为 Shift + Ctrl + Alt + E , 详细内容请查看插件的说明文档。



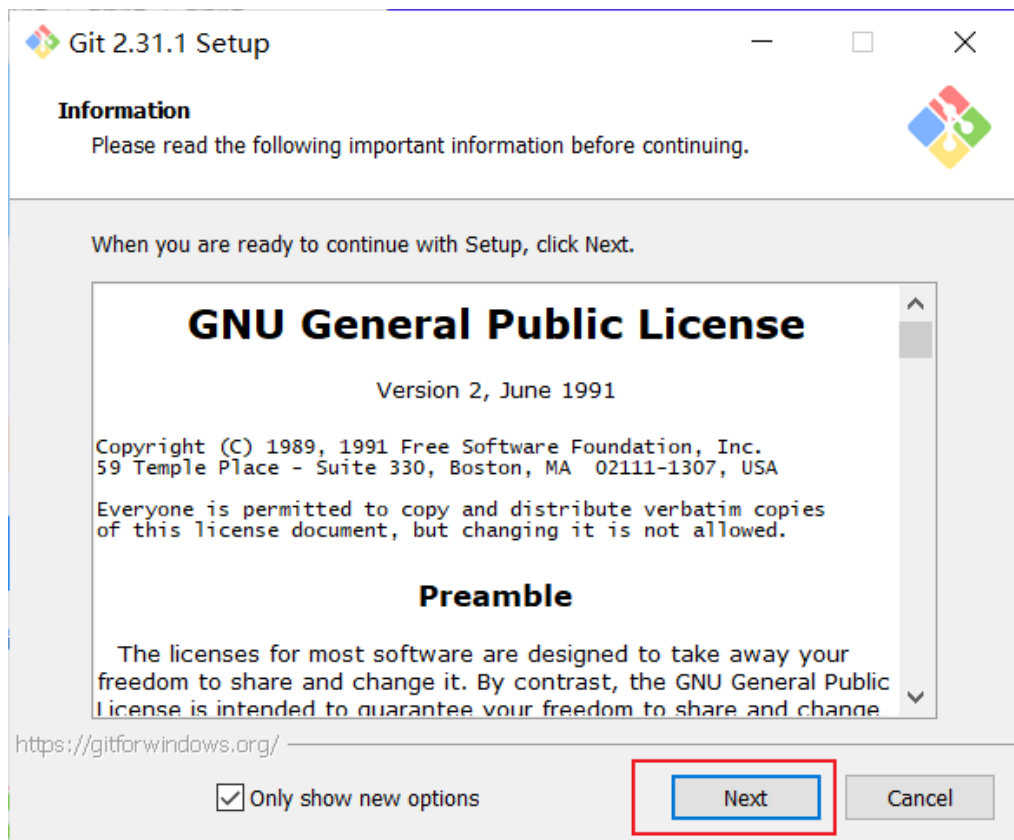
安装 Git

你可以前往 [官网](#) 下载 Git, 有 Windows 版本, 也有 Mac 版本和 Linux 版本.



如果你网速不佳, 也可以去 [腾讯软件中心](#) 下载.

安装过程中, 基本上一路点 Next 就好.



安装好了之后, 按下 Win 键, 输入 cmd , 回车, 便出现了命令行.

输入

git

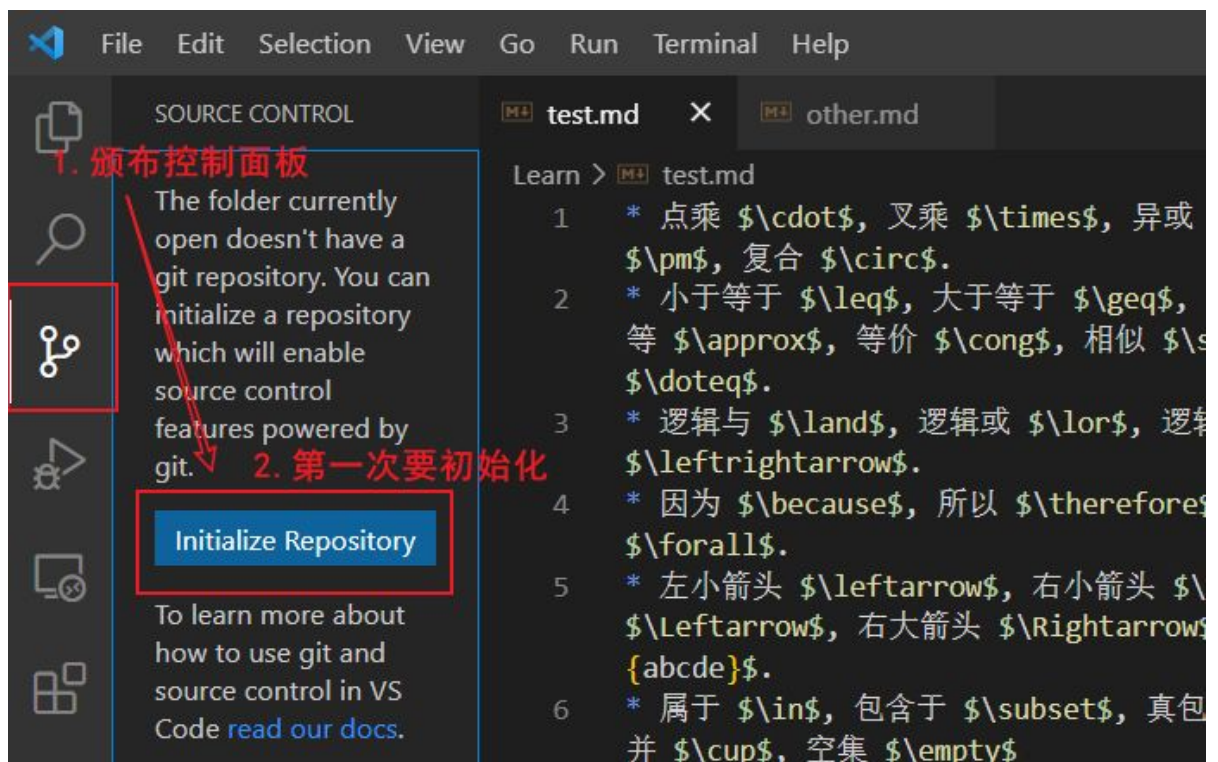


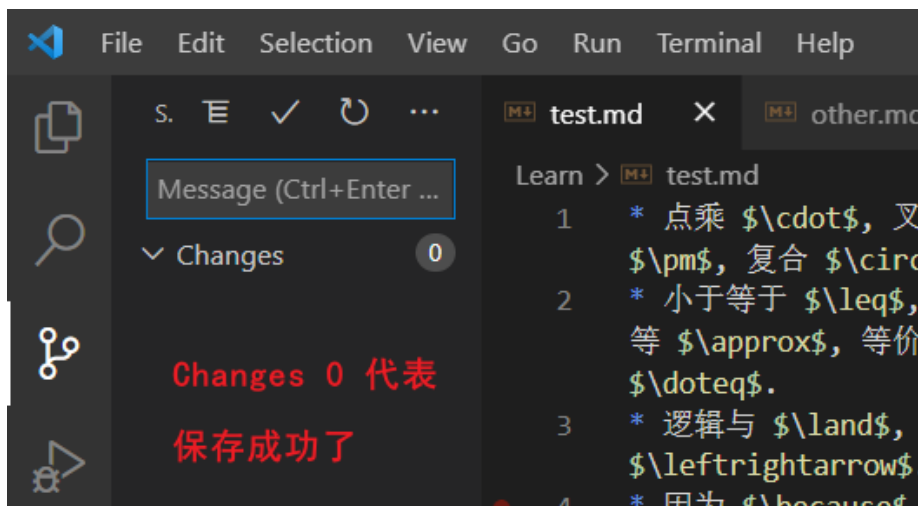
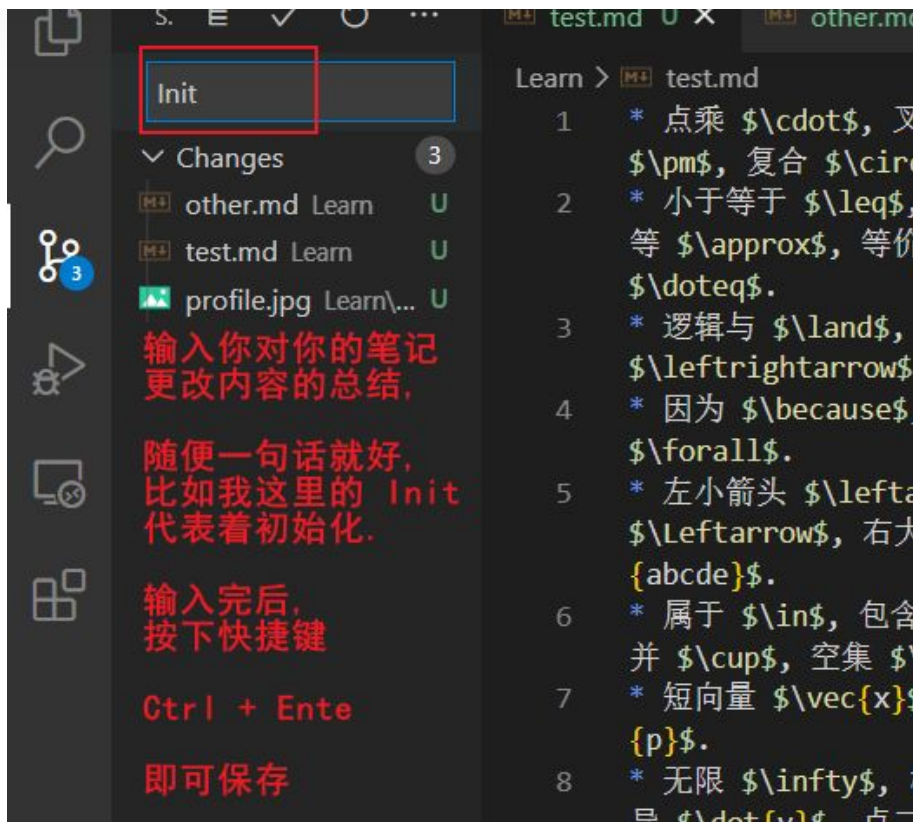

```
(base) PS C:\Users\Orangex4> git
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
      [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
      [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
      [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
      <command> [<args>]
```

版本控制

我们使用 Git 来实现版本控制, 或者是笔记保存效果.

每次你更改了笔记, 在传到云端之前, 都要进行一次笔记保存 Commit.





上传到 GitHub

GitHub 是世界上最大的代码托管平台，你当然也可以把你的笔记托管在 GitHub 上。

你可以看我的这篇 [GitHub 简易指南](#)，加深对 GitHub 的理解。

要上 GitHub，你可能需要一点小技巧，这我就不教你了。

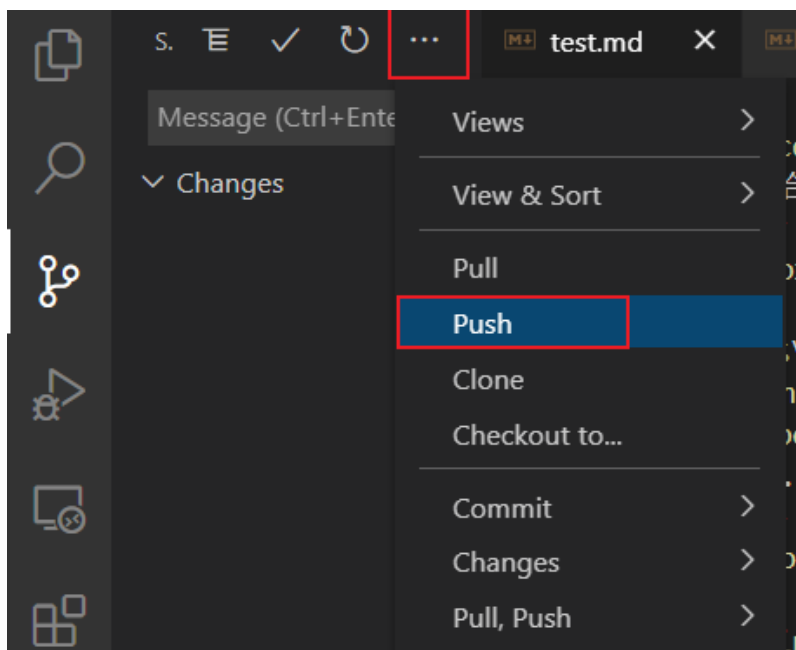
不然你也可以选择国内的 Gitee 或者国内能访问的 GitLab。

看完这篇指南之后，你大概就知道怎么把笔记托管到云端了，而且，完全免费噢。

(虽然可能麻烦了点)



如果你设置好了 GitHub, 需要上传到云端了, 你可以 Git Push 上去.



在线浏览

如果你把你的笔记上传到了云端, 你就可以随时随地地查看你的笔记, 并且可以很方便地分享给别人.

比如, 你要看我的笔记, 可以访问这个链接:

notes.orangex4.cool/?...

如果你把你的笔记上传到了 GitHub, 想要查看, 你可以用这个网址:

notes.orangex4.cool/

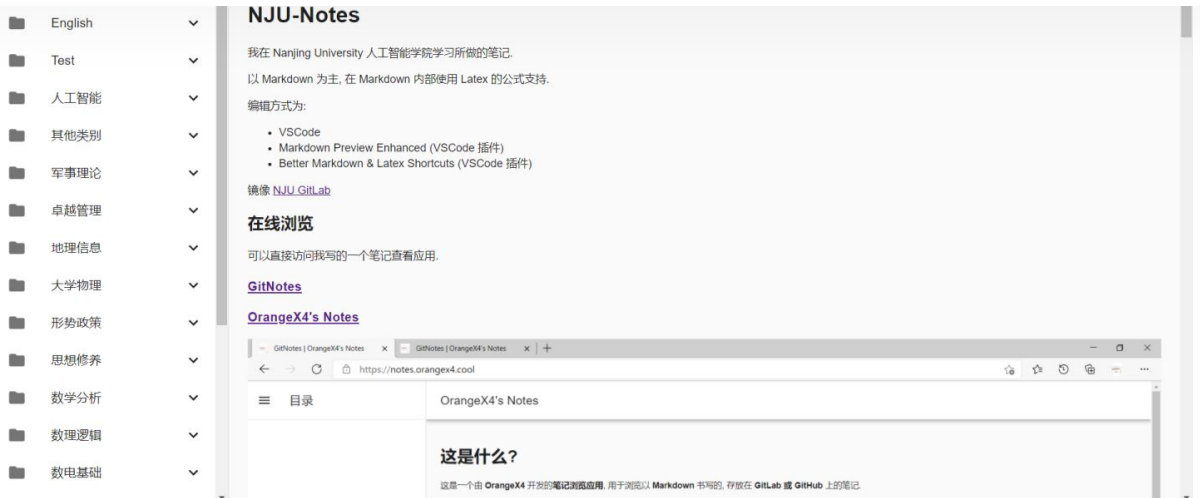
并且仿照这个格式:

notes.orangex4.cool/?...

大概格式是:

notes.orangex4.cool/?...用户名/笔记仓库

你就有了这种效果的笔记!



Markdown 进阶 (选看)

Markdown Preview Enhanced 提供了非常强大的功能, 你可以访问 [MPE 简介](#) 获取更多相关的信息.

例如说, 你可以:

自动目录

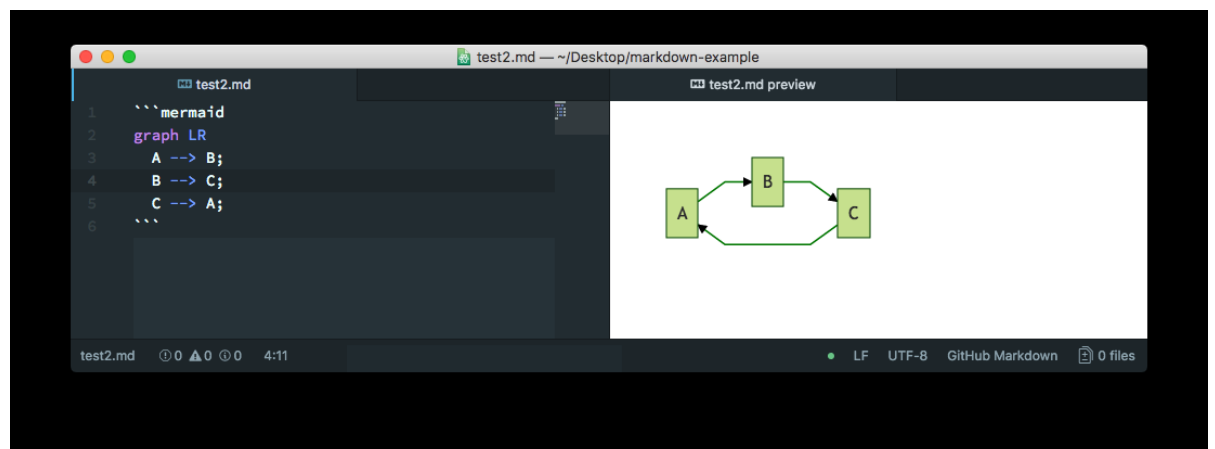
只需要使用语法

```
[TOC]
```

就能自动生成目录.

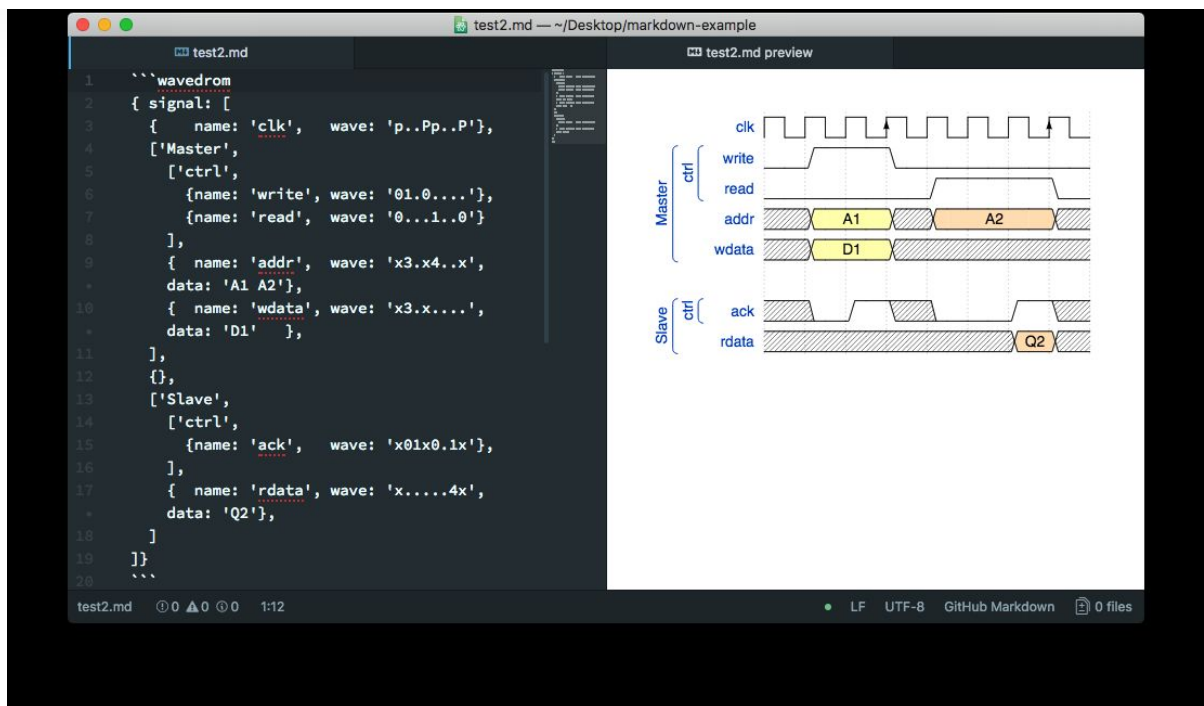
画图

流程图



时序图





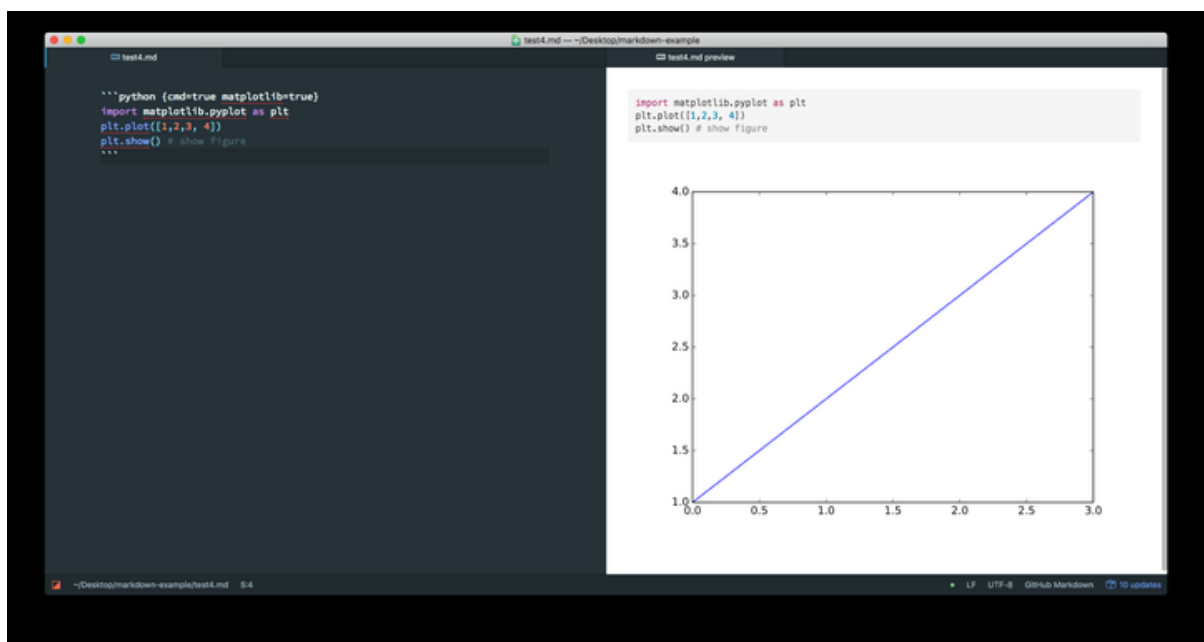
其他

你可以自己访问 [MPE 简介](#) 获取更多相关的信息.

代码块

你可以在 Markdown 中运行 **Python**, **Javascript** 之类的代码!

甚至可以运行 **Latex**!



详见 [Code Chunk](#).

软件推荐





GeoGebra 官网

你可以很方便地画函数图像!



超良心的免费画图软件, 你可以用它画流程图, 电路图之类的各种图.

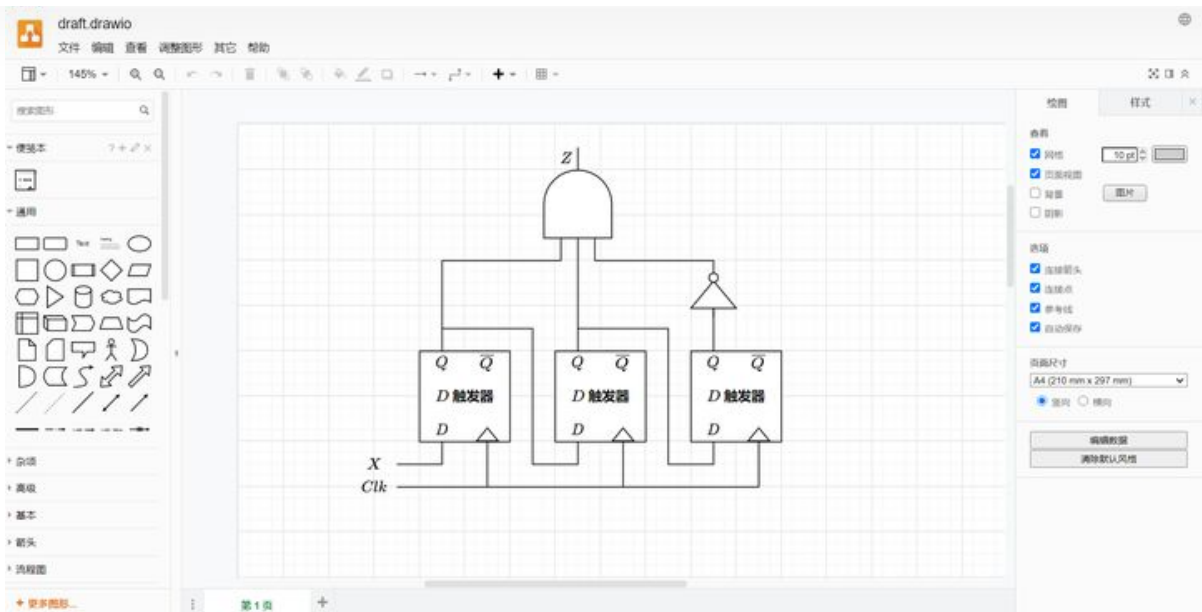
对齐功能超赞, 而且完全免费!

而且可以**不用下载, 在线使用!**

当然你要离线用也可以下离线版本, 甚至可以下载对应的 VS Code 插件!

Drawio 官网





总结

暂时想到的就这些啦, 更多的内容后续再补充, 也欢迎给我提交意见~

顺便推荐一下我写的「计算机学生的第零课」系列文章。您可以通过下方链接查看完整目录。

计算机学生的第零课
[orangex4.cool/post/lesson-zero-for-cs-st...](https://orangex4.cool/post/lesson-zero-for-cs-st/)
[orangex4.cool/post/lesson-zero-for-cs-st...](https://orangex4.cool/post/lesson-zero-for-cs-st/)



知乎专栏:

计算机学生的第零课
www.zhihu.com/column/c_147950670...
www.zhihu.com/column/c_147950670021...



本文使用 [Zhihu On VSCode](#) 创作并发布

编辑于 2022-04-10 15:19

[Visual Studio Code](#) [笔记](#) [学习](#)

▲ 赞同 743 ▼ ● 153 条评论 ↗ 分享 ♥ 喜欢 ★ 收藏 📄 申请转载 ...

写下你的评论...

153 条评论

默认 时间



一只方橙 作者

各位, 现在可以在安装 Python Brackets 插件之后, 按 Ctrl + Shift + Backspace 删除匹配的括号, Ctrl + Shift + / 选择匹配的括号内容, 这个对做数学作业有奇效哦





2021-09-05

👍 6



Spark

🤔🤔🤔我一直很想知道像你们这样的大佬是怎么get到这些知识的

2021-04-21

👍 5



一只方橙 作者

主要还是多关注一下相关信息吧，看见别人发布的東西，或者在网上看见有趣的东西，就马上动手试试，还有就是多逛 GitHub👉

2021-04-22

👍 2



斑马

一点点学的吧🤔或者说，多看、多交流

2021-04-28

👍 赞



一只方橙 作者

更新了哦~用 Sympy 重写了计算器功能，现在支持符号运算，求导，求积分，各种矩阵运算了🤔

2021-05-12

👍 3



宇宙之大

想问一下用paste image，要用什么软件截图才有用啊，我试了vx qq win10自带截图，都显示 “There is not a image in clipboard”

06-24

👍 1



一只方橙 作者

我一般用的 snipaste

06-24

👍 1



宇宙之大 ▸ **一只方橙**

谢谢大佬 用了这个确实可以了

06-24

👍 赞



别在知乎讲现实

太感谢了，非常有用，学到了很多。唯一的缺点就是，这个在线浏览只有GitHub和gitlab。目前国内大环境下，上GitHub属实不太方便，尤其是个人笔记这种，随时要查看的情况下。所以要是国内的gitee版本，就更好了。

2021-12-16

👍 赞



一只方橙 作者

各位，文章更新了哦，加入了 HyperSnips 相关的内容，基本可以达到那位用 vim 和 latex 做笔记的小哥的速度了🤔

2021-05-09

👍 2



让幻想飞

强啊

2021-06-21

👍 赞



君炎

请问，我按照您的方法实现不了单元格的合并是怎么回事呢

2021-07-08

👍 1



一只方橙 作者



选择开启

2021-07-08

👍 1



让幻想飞

终于知道为什么不建议公式中用中文了，一般的导出没问题，但是pandoc不行，可能是与LaTeX引擎有关

2021-06-21

👍 1



一只方橙 作者

markdown 允许公式带中文，会警告不会报错，但是 latex 公式里有中文会直接报错😓

2021-06-21

👍 赞



Spark

能修改字体的类型和格式吗？

2021-05-17

👍 1



Gothink

你直接当html写也没问题，不过这不是复杂化了吗

2021-12-09

👍 赞



让幻想飞

可以爆改css，能改出很多风格。

2021-06-21

👍 赞



废寝忘学

自动补全还不错😓不过我喜欢vim+ultisnips

2021-04-22

👍 1



一只方橙 作者

都挺好，vscode 配置一下 vim 应该也很不错😓

2021-04-22

👍 赞



伯牙牙牙牙

看到熟人嘻嘻

2021-04-22

👍 1



午门十九 🏆

太厉害了！

2021-04-22

👍 1



难自拔于世界之大

非常详细，感谢作者大大分享！👍

2021-12-10

👍 1



希子

真的感恩

2021-12-06

👍 1



axii

怎么这么优秀😓

2021-11-23

👍 赞



剑气萧心

按下快捷键`Ctrl + Shift + P`，输入命令`Open Snippets Directory`，就可以打开一个文件夹。在该文件夹新建一个文件`markdown.hsnips`，并将 这个网址 里面的内容输入进去，保存，



2021-11-11

赞

**一只方橙** 作者 ▶ **剑气萧心**

是的，放完了之后把 txt 后缀改成 hsnips

2021-11-12

1

**剑气萧心** ▶ **一只方橙**

markdown.hsnips是什么类型的文件啊，是建立一个txt文本放进去吗

2021-11-12

赞

[查看全部 7 条回复](#)**唐吉诃德**

您好感谢 您的插件! 想请教下这个"pmat" 为何不能触发呢?

```
snippet pmat "pmat" wm
\begin{pmatrix}
${1:}
\end{pmatrix} $0
endsnippet
```

我理解的wm 触发方式,是输入单词后空格, 再输入pmat, 然后按tab 键, 请问😭

2021-10-31

赞

**一只方橙** 作者

你要在 settings.json 开启一个选项才能正常使用

2021-10-31

赞

**miya**

请问一下，为什么我不能实现自动补全啊，在数学环境里输入内容后没有变化，而且安装了HyperSnips for Math按流程操作后操作后也用不了Snippets.....😭求解

话说Paste Image好像只能粘截图，我复制了本地图片后反馈剪贴板上没有图片——然后我就用了Markdown Image😭😭😭

2021-09-04

赞

**Rainsky** ▶ **miya**

这个可能是因为系统默认隐藏了已知文件的扩展名，可以自行上网搜索更改设置，显示文件扩展名后进行更改，否则你的文件永远是一个txt文件

07-21

赞

**miya** ▶ **一只方橙**

大大你好，我觉得是不是因为我保存的文件还是.txt的原因啊.....😭😭😭

2021-09-10

赞

[查看全部 10 条回复](#)**Ueily**

大佬请教一下，HyperSnips for Math能否将选中的文本作为变量。我看UltiSnip支持用{VISUAL}😭😭😭

2021-08-12

赞

**Ueily** ▶ **一只方橙**

非常感谢!

2021-08-26

赞



, 例子可以有

```
snippet fr "frac" iAm
\\frac{$1:${VISUAL}}{$2}
endsnippet
2021-08-26
```

👍 赞

展开其他 2 条回复 >



Charlie0103

作者作者! 我又来了😭!

1 paste image 怎么把照片弄到剪贴板上😭

2 为什么现在Github网站已经打不开了 那怎么办啊🙄

2021-08-11

👍 赞



一只方橙 作者

截图软件或图片浏览软件复制图片。 github 可能需要科学上网才能上, 这个你自己找, 我不能教。😭

2021-08-11

👍 赞



Charlie0103

感谢大佬不吝赐教!😭

2021-08-11

👍 赞

点击查看全部评论 >

推荐阅读

Markdown-LaTeX: 经管人的VSCode配置大全

全文阅读:

<https://www.lianxh.cn/news/7371>

目录1. VS Code 安装及初始配置

1.1 什么是 VS Code? 1.2 常用的扩展及安装方式1.3 settings 文件与云同步1.4 提高编辑体验的几...

连玉君

Markdown学习笔记

Markdown学习笔记1 (vs code) (2020-03-14)

fitzzz



免费好用的编辑器,

少数派

