

[Markdown+Typora/VSCode 超全教程] 给大一新生便利的文本神器

😊 简要介绍

Markdown 是一种轻量型标记语言, 是一种语法. 以 `.md` 结尾的文本文件就是 Markdown 文件. 相较于 **Word**, 它更加像是 **HTML** 语言或是 `\LaTeX`, 并不是最淳朴的那种“所见即所得”. 它处处透露着一种极简主义. 高效简洁清晰的同时, 又很简单. 看起来舒服, 语法简单, 尤其在处理纯文本上有很大的优势.

它相较于 **Word**, 兼容性非常高, 可以跨平台使用, 不用担心奇奇怪怪的版本兼容问题. 同时, 有许多网站都支持或正在使用 **Markdown** 语法. 如 **Github** (等一系列代码托管平台), **StackOverflow**(等答疑平台), **简书**, **语雀** (等一系列笔记平台).

📐 实际应用

所有要写文本的时候都可以用上 **Markdown!** 它可以让你不再纠结什么字体, 什么样式, 什么排版. 而且逻辑清晰, 层次分明.

像我大一的时候就用 Markdown 来写各种笔记, 演讲稿, 课程论文, 实验报告, 代码的 `README.md` ... 包括本教程文档.

🍴 工具

Markdown 只是一种语法. 那么用来写 Markdown 的文本编辑器呢? 我推荐的是 **Typora** 或者 **VS Code**

Typora

Typora 应该是被广泛用于写 Markdown 的文本软件, 就和 Markdown 语法一样高效. 而且它还有很实用的扩展语法与自定义样式功能. 其能将 `.md` 导出成多种文件, 如 `.pdf`, `.html`, `.docx` (没想到吧, 能导出到 Word)

但是现在 Typora 已经发布正式版并且变为收费软件. 中文官网在此 [Typora 官方中文站](#).

我测试下的网址[百度](#)

我当然是推荐大家都用正版啦. 不过价格是永久版 ¥ 89, 好在可以用在 3 台设备上. 如果和你的两位同学/舍友均摊一下, 每人就只要 ¥ 30, 和一张游戏月卡差不多.

至于盗版以及破解方法[在此随便找一种改注册表时间方法的不介绍\(还挺多的其实\)](#).

~~还有一种免费白嫖的方法就是安装测试版/Beta 版. 官网有历史版本的下载链接[Typora 历史版本下载页](#)~~

白嫖 Beta 版已经寄了, 要么支持正版要么去学习一下破解方法吧 (还是忍不住啦, 看上面的链接 ✌)

VS Code

这是微软家的开源文本编辑器, 理论上来说所有代码, 语言, 都可以用 VS Code 来写, 同样是非常的简洁好用. 在下载插件 **Markdown All in One** 后对 Markdown 的基础支持也是非常的好. 若在 VS Code 下载 Markdown 各种附加扩展, 就能获得比 Typora 更加丰富的扩展语法与操作.

其他

大部分 IDE, 像 Jetbrain 的全家桶里应该每一款, 都支持 Markdown 语法, 在此不多赘述.

本文档主要使用 Typora 进行演示, 同时会介绍许多 Typora 所包含的扩展语法.

🔍 基础教程

当有多种标记方法时我会倾向其中一种.

标题有 * 表示该为扩展语法, 仅在 Typora 或 添加了扩展的 VS Code **本地生效**, 在大多数平台上**并不认可**.

0. 写 Markdown 的第零步

我们写文本的时候大多写的是中文, 可是输入法在输中文时使用的标点为全角标点, 如 , 。 ? ! () 【】：；“” . 这些标点是不被 Markdown 所认可的, 也是无法转义的.

我建议大家写 Markdown 的时候都用半角标点, 即英文标点, 如 , . ? ! () [] ; " " . 且每个半角标点在文本使用时加上后置空格, 符合英文标点的书写规范, 也更加美观.

以微软自带输入法举例, 在使用中文输入法时按下 **Ctrl + .**(这是个句号), 切换标点的全角与半角. 这样即可中文输入+半角标点.

1. 标题 [数个 "#" + 空格前置]

```
# 一级标题  
## 二级标题  
### 三级标题  
#### 四级标题  
##### 五级标题  
##### 六级标题
```

标题会在目录与大纲分级显示, 可以跳转.

在 Typora 中建议开启 **严格模式**, 即不应为 #**标题**, 应为 # 标题.

应该要手动补上空格, 使得 Markdown 语法在其他文本编辑器上兼容.

2. 强调 [用 "*" 或 "_" 包围]

```
**欢迎报考南京大学!** (我喜欢用这种)  
_欢迎报考南京大学!_
```

或者选中想要强调的文字按下 **Ctrl + B**.

E.G.

欢迎报考南京大学!

3. 斜体 [用 "*" 或 "_" 包围]

```
*欢迎大佬来浇浇我各种知识* (我喜欢用这种)
_欢迎大佬来浇浇我各种知识_
```

或者选中想要强调的文字按下 **Ctrl + I**.

E.G.

欢迎大佬来浇浇我各种知识

(P.S. **斜体并强调** [用 "****" 或 "__" 包围])

4. 删除线 [用 "~~" 包围]

```
~~我宣布个事儿，我是Sabiyary!~~
```

E.G.

我宣布个事儿，我是 Sabiyary!

5. *高亮 [用 "==" 包围]

(注意: 此为扩展语法)

```
==我喜欢黄色，也喜欢绿色==
```

E.G.

==我喜欢黄色，也喜欢绿色==

6. 代码 [用 "``" 包围]

```
`sudo rm -rf /*`
```

E.G.

sudo rm -rf /* (没事别乱敲这个!)

"请输入管理员密码: (闪烁的光标)"

99 我测试下代码块呀

E.G **javascript 的代码块呀**

```
console.log(1);
```

999 我测试下本地图片

1. 调整尺寸大小

(1) 调整尺寸大小

直接在连接地址后添加形如 `=600x600` 的数字即可调整图片大小。 (推荐)

```
1 | ! [Alt] (https://img-blog.csdnimg.cn/b937aa6a992d47d9b205f519bcbbc111.png =600x600)
```

说明: 等号 " = " 前面需要有一个空格

- 带尺寸的图片: `=600x600`
- 宽度确定高度等比例的图片: `=600x`
- 高度确定宽度等比例的图片: `=x600`



7. 代码块 [按三个 `` 并敲回车]

```
```
// 这里就可以开始输入你要的代码了
#include <stdio.h>
int main() {
```

```
print ("Hello, world!\n") ;
return 0;
}
``` // (这三个``文本编辑器会帮你自动补全 一般不用手动输)
```

(我之前都是用这个来展示各种语法的, 应该不用举例了吧)

要想显示行数的话, 一般要在 Typora 的设置里勾上这个显示行数的选项.

代码块里可以选择语言, 其会根据语言来自动高亮各个语句. 在选择语言后, ``` 会变为 ``+ 对应语言.

8. 引用 [> + 空格 前置]

```
> 24岁, 是学生.
> > 学生特有的无处不在(恼)
```

引用是可以嵌套的!

E.G.

```
24岁, 是学生.
    学生特有的无处不在(恼)
```

9. 无序列表 [-" 或 "+" + 空格 前置]

```
- 一颗是枣树 (我喜欢用这种)
+ 另一颗还是枣树
* (其实这种也可以, 不过由于在 Typora 中很难单个输入, 故不常用)
```

三种前置符都可以, 敲回车会自动补全, 可在 Typora 设置中调整补全的符号, 敲回车后按下 Tab 会缩进一级.

E.G.

- 一颗是枣树
- 另一颗还是枣树

10. 有序列表 [数字 + "." + 空格 前置]

```
我来这里就为了三件事:
1. 公平
2. 公平
3. 还是tm的公平!
```

敲回车会自动补全, 敲回车后按下 Tab 会缩进一级.

E.G.

我来这里就为了三件事:

1. 公平
2. 公平
3. 还是 tm 的公平!
4. 我自己加的呢

11. *上标 [用 "^" 包围]

(注意: 此为扩展语法)

C语言中int的上限是 $2^{31} - 1 = 2147483647$

E.G.

C 语言中 int 的上限是 $2^{31} - 1 = 2147483647$

12. *下标 [用 "~" 包围]

(注意: 此为扩展语法)

H~2~O 是剧毒的!

E.G.

H~2~O 是剧毒的!

13. *注释 ["[^]" 后置]

(注意: 此为扩展语法)

> 今日我们相聚于此, 是为了学习 Markdown 的使用, 它的教程对于全体「观众」而言, 值得足足两个硬币的支持鼓励![^1]

[^1]: 沃兹·基·硕德 改编自「公鸡」普契涅拉.

需要在文末写上注释对应的内容

E.G.

今日我们相聚于此, 是为了学习 Markdown 的使用, 它的教程对于全体「观众」而言, 值得足足两个硬币的支持鼓励![^1]

[^1]: 沃兹·基·硕德 改编自「公鸡」普契涅拉.

14. 链接 [常用 "[" + "(" 分别包围文本与链接]

(注意: 文内跳转为扩展用法)

[来看看我贫瘠的仓库罢](<https://github.com/Sakiyary>)

[基础教程: 12. 下标](#12. 下标 [用 "~" 包围])

支持网页链接与文内跳转, 按住 **Ctrl** 并 **单击鼠标左键** 即可跳转.

E.G.

[来看看我贫瘠的仓库罢](#)

[基础教程: 12. 下标](#12. 下标 [用 "~" 包围])

15. 任务列表 ["- []" + 空格 前置]

TodoList:

- [] 刷B站
- [] 写代码
- [x] 起床

用 **x** 代替 **[]** 中的空格来勾选任务列表. 在 Typora 中可以直接用鼠标左键单击勾选框.

E.G. TodoList:

- 刷 B 站
- 写代码
- 起床

16. 表格 [用 "|" 绘制表格边框]

| 学号 | 姓名 | 年龄 |
|---------|----|----|
| 114514 | 田所 | 24 |
| 1919810 | 浩三 | 25 |

| 学号 | 姓名 | 年龄 |
| :--- | :---: | ---: | (引号的位置代表着 左对齐, 居中, 右对齐)
|114514|田所|24|
|1919810|浩三|25|

第一行为表头, 并由第二行分割线决定对齐方式与长度, 第三行及之后即表格数据

E.G.

| 学号 | 姓名 | 年龄 |
|---------|----|----|
| 114514 | 田所 | 24 |
| 1919810 | 浩三 | 25 |

17. 图片 [直接拖进来或者复制粘贴]

```
![图片](图片的位置)
```

我还是会选择拖进来或者复制粘贴啦~ 在 Typora 的设置里也可以改图片的储存方式.

18. 分割线 [按三个 "*" 或 "-" 或 "_" 并敲回车]

```
***
```

```
--- (我喜欢用这种)
```

```
// (其实按三个及以上都可以)
```

由于 * 与 _ 均会自动补全, 所以我觉得 - 最为方便.

E.G.



19. Emoji 表情 [":" 前置]

(注意: 英文输入为扩展语法)

```
:sweat_smile:  
:drooling_face:  
:clown_face:  
// (敲回车或者鼠标点击, 后置的":":一般不需要手动输)
```

这个功能唯一的要求就是英语水平要高, 或者大概记得各个 Emoji 的英文名.

E.G.

😅 :drooling_face: :clown_face:

对于其余普通的 Markdown 文本编辑器, 可以直接将 Emoji 表情复制进来, 这是直接硬编码的 (刻进 DNA 里)

E.G.

😅 😂 😅

用好这个功能可以让你的文本非常的可爱! 太抽象了

这里分享一个可以复制[全 Emoji 的网站](#), 非常好用! 我之前的 C 语言大作业也是从这里下载的资源!

💡 进阶教程

1. 目录 [自动生成]

[TOC] (此为 Typora 特有的, 如本文档开头)

若使用 VS Code 搭配 Markdown All in One 扩展, 可在 VS Code 的命令面板 (即 VS Code Command Palette) 输入 Create Table of Contents 自动生成目录, 且可在扩展设置中细调目录参数.

2. 内联 HTML 代码 [用 "<> </>" 包围]

```
<div style="text-align:center">
    <font style="color:red">我不会 HTML 呜呜呜... 浇浇我</font>
</div>

<center>简单的文字居中也可以这样</center>

<u>我差点忘了还有下划线这东西...</u>
```

只要你会写, 你完全可以把 Markdown 当作 **HTML** 来写.

同时, .md 文件可以直接导出成一个网页.

下划线可以选中想要下划的文字按下 **Ctrl + U**.

E.G.

我不会 HTML 呜呜呜... 浇浇我
简单的文字居中也可以这样

我差点忘了还有下划线这东西...

3. 内联 \$\\LaTeX\$ 公式 [用 "\$" 包围]

(注意: 部分编译器会不识别部分符号)

```
$\\LaTeX$ 是最好用的论文排版语言! 不信你看!

$a^n+b^n=c^n$


$$
\%usepackage{unicode-math}
\displaystyle \ointclockwise \mathcal{D}[x(t)]
\sqrt{\frac{\displaystyle 3\mu^2-\sum_{q=0}^{\infty}(z+\hat{L})^q}{\exp(\sum_i q^2 \hbar x)}} \displaystyle (\sum_{Tr} \mathbf{A}) 
\left( \sum_{\Lambda_{j_1 j_2}}^{i_1 i_2} \Gamma_{i_1 i_2}^{j_1 j_2} \right)
\hookrightarrow \vec{D} \cdot \sum_{P} \mathbf{A}_P
=\underbrace{\widetilde{\left( \frac{\partial \mu}{\partial \nu} \right)}_{K_3}}_{Fe(CN)_6}, \forall z \in \mathbb{R}
$$
```

用 \$ 包围为单条公式, 按下两个 \$ 并敲回车即生成公式块.

E.G.

\$\LaTeX\$ 是最好用的论文排版语言! 不信你看!

$a^n + b^n = c^n$

```
$$ \%usepackage{unicode-math} \displaystyle \ointclockwise \mathcal{D}[x(t)] \sqrt{\frac{\displaystyle 3\uppi^2 - \sum_{q=0}^{\infty} (z+\hat{L})^q \exp(\text{symrm}{i}q^2 \hbar x)}{\text{symsup}{Tr}\text{symbfcal}{A}}} \left( \text{symbf}{\Lambda}_{j_1 j_2}^{i_1 i_2} \text{Gamma}_{i_1 i_2}^{j_1 j_2} \right) \rightarrow \underbrace{\widetilde{\left( \langle \frac{\not{in} \emptyset}{\varphi_i \alpha_k \uparrow} \middle| \text{vert} \frac{\partial_\mu T_{\mu\nu}}{\mu\nu} \rangle \right)}_{\mathcal{K}_3}}_{\mathcal{F}_e(\mathcal{C}_N)} , \forall z \in \mathbb{R} $$
```

4. *网络图床

(这是学长给我的网络图床教程, 我并未实践过...)

分享一个 Typora 搭配腾讯云 COS/阿里云 OSS 图床的[例子](#). 新用户免费试用 6 个月, 另外还可选择七牛云或者路过大图床.

5. *Typora 的常用快捷键

| 按键 | 效果 | 按键 | 效果 |
|-------------------------|-----------|-------------------------|-------------|
| Ctrl + D | 选中当前词 | Ctrl + L | 选中当前句/行 |
| Ctrl + E | 选中当前区块 | Ctrl + F | 搜索当前选中 |
| Ctrl + B | 加粗当前选中 | Ctrl + H | 替换当前选中 |
| Ctrl + I | 倾斜当前选中 | Ctrl + U | 下划当前选中 |
| Ctrl + K | 将当前选中生成链接 | Ctrl + J | 滚动屏幕将选中滚至顶部 |
| Ctrl + W | 关闭当前窗口 | Ctrl + N | 打开新窗口 |
| Ctrl + O | 打开文件 | Ctrl + P | 搜索文件并打开 |
| Ctrl + 回车 | 表格下方插入行 | Ctrl + , | 打开偏好设置 |
| Ctrl + . | 切换全角/半角标点 | Ctrl + / | 切换正常/源代码视图 |
| Ctrl + Shift + - | 缩小视图缩放 | Ctrl + Shift + + | 放大视图缩放 |

还有一些不常用的/三键的快捷键不在此列出.

6. *Typora 的主题样式与检查元素

Markdown 在编译后约等于 HTML. 而 Typora 的正常视图就是编译后的 Markdown, 故 Typora 的主题样式本质就是 CSS 文件.

可以下载各种好看的主题给 Typora 换上, 同时也可以自己调整对应的 CSS 文件, 或者自己手搓.

在 Typora 设置中开启 [调试模式](#) 后即可在正常视图右击打开 [检查元素](#), 在其中就可以完全将 Markdown 文件当成 HTML 来编辑.

100 总结

至此, **Markdown + Typora / VSCode** 的手册教程也告一段落.

不知你看完这么长的教程/手册, 是否能体会到 Markdown 的精妙简洁之处呢?

其实 Markdown 只是标记语言的最开始, 我的感受是会了 Markdown 之后对于理解 HTML 也有帮助, 对于用 \LaTeX 来写论文也有帮助. 标记语言正是为了摆脱 Word 那种虽然"所见即所得", 但又过于花哨浮华, 很差的兼容性与闭源的编码的缺陷. 当你能掌握这样的"所写即所得"时, 你肯定会感受到用 Markdown 这类语言来处理文本的妙处!