

## Learn Git 2 - Markdown 学习笔记

现在，我们已经初步掌握了 GitHub 的基本操作。这里需要明确一下，Git 其实只能跟踪文本文件的改动，比如TXT、C语言代码等，而图片、视频这些二进制文件，虽然也能由 Git 管理，但没法跟踪文件的变化。因此，文本成为了 Git 的主角，而 Markdown 能够使人们使用易读易写的纯文本格式编写文档，几乎每一个 GitHub 项目中几乎都存在 Markdown 的身影。

还记得上一章 Tutorial 中出现的那个 README.md 吗？.md 就是 Markdown 文档的标准扩展名，下面我们来学习如何使用 Markdown，欢迎大家一起交流 Markdown 的使用经验和技巧！

本学习笔记主要参考了以下两个Markdown教程，感谢互联网对知识传播的巨大贡献！

- [Markdown Reference](#)（强烈推荐这个互动教程！）
- [Markdown 教程](#) | [菜鸟教程](#)

---

### Markdown 简介

编辑器

测试实例

### Markdown 标题

使用 = 和 - 标记一级和二级标题

使用 # 号标记多级标题

### Markdown 格式

段落

字体

分隔线

删除线

下划线

脚注

### Markdown 列表

无序列表

有序列表

列表嵌套

### Markdown 区块

区块中使用列表

列表中使用区块

### Markdown 代码

单行代码

代码区块

### Markdown 链接

普通链接

高级链接

### Markdown 图片

插入图片

设定图片的高度与宽度

### Markdown 表格

插入表格

对齐方式

### Markdown 高级技巧

支持的 HTML 元素

转义

公式

---

# Markdown 简介

---



- Markdown 是一种轻量级标记语言，它允许人们使用易读易写的纯文本格式编写文档。
- Markdown 语言在 2004 由约翰·格鲁伯（英语：John Gruber）创建。
- Markdown 编写的文档可以导出 HTML、Word、图像、PDF、Epub 等多种格式的文档。
- Markdown 编写的文档后缀为 **.md** 或 **.markdown**。
- Markdown 能被使用来撰写电子书，如：Gitbook。
- 当前许多网站都广泛使用 Markdown 来撰写帮助文档或是用于论坛上发表消息，如GitHub。

## 编辑器

---

本教程使用 Typora 编辑器来讲解 Markdown 的语法，Typora 支持 MacOS、Windows、Linux 平台，且包含多种主题，编辑后直接渲染出效果。Typora 支持导出HTML、PDF、Word、图片等多种类型文件。

Typora 官网：<https://typora.io/>

## 测试实例

---

使用 Typora 编写 Hello World! :

```
1 | # Hello world!
```

显示效果如下：

# Hello World!

---

## Markdown 标题

---

Markdown 标题有两种格式。

### 使用 = 和 - 标记一级和二级标题

---

= 和 - 标记语法格式如下：

```
1  我展示的是一级标题
2  =====
3
4  我展示的是二级标题
5  -----
```

显示效果如下：

# 我展示的是一级标题

## 我展示的是二级标题

## 使用 # 号标记多级标题

使用 # 号可表示 1-6 级标题，一级标题对应一个 # 号，二级标题对应两个 # 号，以此类推。

```
1  # 一级标题
2  ## 二级标题
3  ### 三级标题
4  #### 四级标题
5  ##### 五级标题
6  ##### 六级标题
```

显示效果如下所示：

# 一级标题

## 二级标题

### 三级标题

#### 四级标题

##### 五级标题

###### 六级标题

## Markdown 格式

本节介绍Markdown中的段落和字体格式。

## 段落

Markdown 段落没有特殊的格式，直接编写文字就好，**段落的换行是使用两个以上空格加上回车。**

1	这是一个段落 ↓
2	这是第二个段落

也可以在段落后面使用一个空行来表示重新开始一个段落。

1	这是一个段落
2	
3	这是第二个段落

# 字体

Markdown 可以使用以下几种字体：

1	<i>*斜体文本*</i>
2	<u>斜体文本</u>
3	<b>**粗体文本**</b>
4	<u>粗体文本</u>
5	<b><i>**粗斜体文本**</i></b>
6	<u><b><i>粗斜体文本</i></b></u>

显示效果如下所示：

<i>斜体文本</i>
<u>斜体文本</u>
<b>粗体文本</b>
<u>粗体文本</u>
<b><i>粗斜体文本</i></b>
<u><b><i>粗斜体文本</i></b></u>

# 分隔线

你可以在一行中用三个以上的星号、减号、底线来建立一个分隔线，行内不能有其他东西。你也可以在星号或是减号中间插入空格。下面每种写法都可以建立分隔线：

1	***
2	
3	* * *
4	
5	*****
6	
7	- - -
8	
9	-----

显示效果如下所示：


## 删除线

如果段落上的文字要添加删除线，只需要在文字的两端加上两个波浪线 ~~ 即可，实例如下：

```
1 | Markdown
2 | Markdown
```

显示效果如下所示：

```
Markdown
Markdown
```

## 下划线

下划线可以通过 HTML 的 标签来实现：

```
1 | <u>带下划线文本</u>
```

显示效果如下所示：

```
带下划线文本
```

## 脚注

脚注是对文本的补充说明。

Markdown 脚注的格式如下：

```
1 | 这是一条脚注[^FOOTNOTE].
2 |
3 | \[^FOOTNOTE\]: Let's try markdown!
```

以下实例演示了脚注的用法：

这是一条脚注 <sup>1</sup>

## Markdown 列表

Markdown 支持有序列表和无序列表。

### 无序列表

无序列表使用星号(\*)、加号(+)或是减号(-)作为列表标记：

```
1  *  第一项
2  *  第二项
3  *  第三项
4
5  +  第一项
6  +  第二项
7  +  第三项
8
9
10 -  第一项
11 -  第二项
12 -  第三项
```

显示结果如下：

- 第一项
- 第二项
- 第三项
- 第一项
- 第二项
- 第三项
- 第一项
- 第二项
- 第三项

-

## 有序列表

有序列表使用数字并加上 . 号来表示，如：

```
1  1. 第一项
2  2. 第二项
3  3. 第三项
```

显示结果如下：

1. 第一项
2. 第二项
3. 第三项

-

## 列表嵌套

列表嵌套只需在子列表中的选项添加四个空格（一个Tab）即可：

```
1  1. 第一项：
2      - 第一项嵌套的第一个元素
3      - 第一项嵌套的第二个元素
4  2. 第二项：
5      - 第二项嵌套的第一个元素
6      - 第二项嵌套的第二个元素
```

显示结果如下：

1. 第一项：
  - 第一项嵌套的第一个元素
  - 第一项嵌套的第二个元素
2. 第二项：
  - 第二项嵌套的第一个元素
  - 第二项嵌套的第二个元素

## Markdown 区块

Markdown 区块引用是在段落开头使用 > 符号，然后后面紧跟一个空格符号：

```
1 | > 区块引用
```

显示结果如下：

区块引用

另外区块是可以嵌套的，一个 > 符号是最外层，两个 > 符号是第一层嵌套，以此类推退：

```
1 | > 最外层
2 | > > 第一层嵌套
3 | > > > 第二层嵌套
```

显示结果如下：

最外层

第一层嵌套

第二层嵌套

## 区块中使用列表

区块中使用列表实例如下：

```
1 | > 区块中使用列表
2 | > 1. 第一项
3 | > 2. 第二项
4 | > + 第一项
5 | > + 第二项
6 | > + 第三项
```

显示结果如下：

区块中使用列表

1. 第一项
2. 第二项
  - 第一项

- 第二项
- 第三项

## 列表中使用区块

如果要在列表项目内放进区块，那么就需要在 > 前添加四个空格的缩进。

区块中使用列表实例如下：

```
1 | * 第一项
2 |   > GitHub
3 |   > Markdown
4 | * 第二项
```

显示结果如下：

- 第一项
  - GitHub
  - Markdown
- 第二项

## Markdown 代码

本节介绍Markdown中插入代码块的方法。

### 单行代码

如果是段落上的一个函数或片段的代码可以用反引号把它包起来（`），例如：

```
1 | `printf()` //函数
```

显示结果如下：

```
printf() //函数
```

### 代码区块

代码区块使用 **4 个空格**或者一个制表符（Tab 键）。

例如：



```

1  ....def hanoi(n, a, b, c):
        if n == 1:
            print(a, '-->', c)
        else:
            hanoi(n-1, a, c, b)
            print(a, '-->', c)
            hanoi(n-1, b, a, c)
8      print(hanoi(5, 'a', 'b', 'c'))

```

渲染后结果如下:

```

1  def hanoi(n, a, b, c):
2      if n == 1:
3          print(a, '-->', c)
4      else:
5          hanoi(n-1, a, c, b)
6          print(a, '-->', c)
7          hanoi(n-1, b, a, c)
8  print(hanoi(5, 'a', 'b', 'c'))

```

(推荐) 也可以使用一对 `` 包裹一段代码, 并指定一种语言 (也可以不指定), 例如:

```

1  $(document).ready(function(){
2      alert('Bingo! ');
3  });

```

## Markdown 链接

本节介绍Markdown中的普通链接和高级链接。

-

### 普通链接

链接使用方法如下:

```

1  [链接名称](链接地址)
2
3  或者
4
5  <链接地址>

```

例如:

```

1  这是一个链接 [百度](https://www.baidu.com/)

```

显示结果如下:

这是一个链接 [百度](https://www.baidu.com/)

直接使用链接地址：

```
1 | <https://www.baidu.com/>
```

显示结果如下：

<https://www.baidu.com/>

## 高级链接

```
1 | 链接也可以用变量来代替，文档末尾附带变量地址：
2 | 这个链接用 1 作为网址变量 [Google].[1].
3 | 然后在文档的结尾为变量赋值（网址）
4 |
5 | [1]: http://www.google.com/
```

显示结果如下：

链接也可以用变量来代替，文档末尾附带变量地址：  
这个链接用 1 作为网址变量 [Google](#)  
然后在文档的结尾为变量赋值（网址）

## Markdown 图片

Markdown中插入图片的方法和插入链接的方法类似。

-

## 插入图片

Markdown 图片语法格式如下：

```
1 | ![alt 属性文本](图片地址)
2 |
3 | ![alt 属性文本](图片地址 "可选标题").
```

- 开头一个感叹号！
- 接着一个方括号，里面放上图片的替代文字
- 接着一个普通括号，里面放上图片的网址，最后还可以用引号包住并加上选择性的 'title' 属性的文字。

使用实例：

```
1 | ![GitHub Desktop](https://desktop.github.com/images/desktop-icon.svg)
2 |
3 | ![GitHub Desktop](https://desktop.github.com/images/desktop-icon.svg "GitHub Desktop Icon").
```

显示结果如下：



(将鼠标移至下面的图片时会显示 "GitHub Desktop Icon")



你还可以像网址那样对图片网址使用变量:

```
1 | 这个链接用 1 作为网址变量 \[GitHub Desktop\].\[1\].  
2 | 然后在文档的结尾位变量赋值 (网址)  
3 |  
4 | \[1\]: https://desktop.github.com/images/desktop-icon.svg
```

显示结果如下:

这个链接用 1 作为网址变量 [GitHub Desktop](#).  
然后在文档的结尾位变量赋值 (网址)

## 设定图片的高度与宽度

目前 Markdown 还没有办法指定图片的高度与宽度, 如果你需要的话, 你可以使用 html 中的 标签。

```
1 | 
```

显示结果如下:



# Markdown 表格

本节介绍Markdown的表格制作方法。

## 插入表格

Markdown 制作表格使用 | 来分隔不同的单元格，使用 - 来分隔表头和其他行。

语法格式如下：

```
1 | | 表头 | 表头 |
2 | | ---- | ---- |
3 | | 单元格 | 单元格 |
4 | | 单元格 | 单元格 |
```

以上代码显示结果如下：

表头	表头
单元格	单元格
单元格	单元格

## 对齐方式

我们可以设置表格的对齐方式：（单元格中的内容默认左对齐）

- -: 设置内容和标题栏居右对齐。
- -: 设置内容和标题栏居左对齐。
- :: 设置内容和标题栏居中对齐。
- 注意：和 - 之间不能有空格，- 的数量无限制。

实例如下：

```
1 | | 左对齐 | 右对齐 | 居中对齐 |
2 | | :----- | -----: | :-----: |
3 | | 单元格 | 单元格 | 单元格 |
4 | | 单元格 | 单元格 | 单元格 |
```

以上代码显示结果如下：

左对齐	右对齐	居中对齐
单元格	单元格	单元格
单元格	单元格	单元格

# Markdown 高级技巧

本节介绍一些常用的 Markdown 高级技巧，更多内容请参考 < <https://spec.commonmark.org/> >

## 支持的 HTML 元素

不在 Markdown 涵盖范围之内的标签，都可以直接在文档里面用 HTML 撰写。

目前支持的 HTML 元素有：<code><b><i><em><sup><sub><br> 等，如：

```
1 | 使用 <code>Ctrl</code>+<code>Alt</code>+<code>Del</code> 重启电脑
```

输出结果为：

使用 `Ctrl` + `Alt` + `Del` 重启电脑

## 转义

Markdown 使用了很多特殊符号来表示特定的意义，如果需要显示特定的符号则需要使用转义字符，Markdown 使用反斜杠转义特殊字符：

```
1 | **文本加粗**
2 | \*\* 正常显示星号 \*\*
```

输出结果为：

**文本加粗**  
**\*\* 正常显示星号 \*\***

Markdown 支持以下这些符号前面加上反斜杠来帮助插入普通的符号：

```
1 | \_ 反斜线
2 | ` 反引号
3 | * 星号
4 | _ 下划线
5 | {} 花括号
6 | [] 方括号
7 | () 小括号
8 | # 井字号
9 | + 加号
10 | - 减号
11 | . 英文句点
12 | ! 感叹号
```

## 公式

当你需要在编辑器中插入数学公式时，可以使用两个美元符 \$\$ 包裹 TeX 或 LaTeX 格式的数学公式来实现。提交后，会根据需要加载 Mathjax 对数学公式进行渲染。如：

```

1  $$
2  \mathbf{v}_1 \times \mathbf{v}_2 = \begin{vmatrix}
3  \mathbf{i} & \mathbf{j} & \mathbf{k} \\
4  \frac{\partial X}{\partial u} & \frac{\partial Y}{\partial u} & 0 \\
5  \frac{\partial X}{\partial v} & \frac{\partial Y}{\partial v} & 0
6  \end{vmatrix}.
7  $$

```

输出结果为：

$$\mathbf{V}_1 \times \mathbf{V}_2 = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \frac{\partial X}{\partial u} & \frac{\partial Y}{\partial u} & 0 \\ \frac{\partial X}{\partial v} & \frac{\partial Y}{\partial v} & 0 \end{vmatrix}$$

1. [Let's try markdown!](#)