

文档编号: COMPANY-DEPARTMENT-00000000

---

---

# Markdown 语法简明教程

## V0.1

---

---

DOMINIC



COMPANYNAME

2019-08

## 修订记录

序号	修订人	修订时间	修订记录
V0.1	Dominic	2022-08-20T00:00:00.000Z	创建示例文档

## 目录

<b>1</b>	<b>Markdown 简介</b>	<b>1</b>
1.1	Markdown 是什么?	1
1.2	谁创造了它?	1
1.3	为什么要使用它?	1
1.4	怎么使用?	2
1.5	谁在用?	2
<b>2</b>	<b>语法介绍</b>	<b>2</b>
2.1	标题	2
2.2	段落	2
2.3	区块引用	3
2.3.1	引用的多层嵌套	3
2.4	锚点	4
2.5	代码区块	4
2.6	强调	4
2.6.1	斜体	4
2.6.2	粗体	5
2.6.3	删除线	5
2.7	列表	5
2.8	分割线	6
2.9	链接	6
2.9.1	行内式	6
2.9.2	参考式	6
2.10	图片	7
2.10.1	图片行内式	7
2.10.2	图片参考式	8
2.11	反斜杠\	8
2.12	代码	9
2.12.1	代码行内式	9
2.12.2	缩进式多行代码	9
2.12.3	用六个 ' 包裹多行代码	9
2.12.4	内容目录	10
2.13	表格	10
2.14	注脚	11

2.15 LaTeX 公式 . . . . .	11
2.15.1 \$ 表示行内公式 . . . . .	11
2.15.2 \$\$ 表示整行公式 . . . . .	12
<b>3 尝试一下 . . . . .</b>	<b>12</b>
<b>4 参考资料 . . . . .</b>	<b>12</b>

## List of Figures

1	MarkdownLogo . . . . .	8
---	------------------------	---

## List of Tables

# 1 Markdown 简介

## 1.1 Markdown 是什么？

**Markdown** 是一种轻量级**标记语言**，它以纯文本形式 (易读、易写、易更改) 编写文档，并最终  
以 HTML 格式发布。

**Markdown** 也可以理解为将以 MARKDOWN 语法编写的语言转换成 HTML 内容的工具。

## 1.2 谁创造了它？

它由[Aaron Swartz](#)和 **John** 共同设计，**Aaron Swartz** 就是那位于去年 (2013 年 1 月 11 日)  
自杀, 有着**开挂**一般人生经历的程序员。维基百科对他的[介绍](#)是：**软件工程师、作家、政治组织  
者、互联网活动家、维基百科人**。

- **14 岁**参与 RSS 1.0 规格标准的制订。
- **2004 年**入读**斯坦福**，之后退学。
- **2005 年**创建[Infogami](#)，之后与[Reddit](#)合并成为其合伙人。
- **2010 年**创立求进会 (Demand Progress)，积极参与禁止网络盗版法案 (SOPA) 活动，  
最终该提案被撤回。
- **2011 年 7 月 19 日**，因被控从 MIT 和 JSTOR 下载 480 万篇学术论文并以免费形式上  
传于网络被捕。
- **2013 年 1 月**自杀身亡。

## 1.3 为什么要使用它？

- 它是易读 (看起来舒服)、易写 (语法简单)、易更改**纯文本**。处处体现着**极简主义**的影子。
- 兼容 HTML，可以转换为 HTML 格式发布。
- 跨平台使用。
- 越来越多的网站支持 Markdown。
- 更方便清晰地组织你的电子邮件。(Markdown-here, Airmail)
- 摆脱 Word

## 1.4 怎么使用？

如果不算**扩展**，Markdown 的语法绝对**简单**到让你爱不释手。

Markdown 语法主要分为如下几大部分：

**标题，段落，区块引用，代码区块，强调，列表，分割线，链接，图片，反斜杠 \，符号’ ”。**

## 1.5 谁在用？

Markdown 的使用者：+ GitHub + 简书 + Stack Overflow + Apollo + Moodle + Reddit + 等等

# 2 语法介绍

## 2.1 标题

两种形式：

1. 使用=和-标记一级和二级标题。

```
1 > 一级标题
2 > `=====`
3 > 二级标题
4 > `-----`
```

1. 使用#，可表示 1-6 级标题。

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

## 2.2 段落

段落的前后要有空行，所谓的空行是指没有文字内容。若想在段内强制换行的方式是使用**两个以上**空格加上回车（引用中换行省略回车）。



## 2.3 区块引用

区块引用需要在被引用的文本前加上 > 符号。

```
1 > 这是一个区块引用实例，  
2  
3 > Markdown.
```

这是一个区块引用实例，

Markdown.

Markdown 也允许你偷懒只在整个段落的第一行最前面加上 >：

```
1 > 平生不会相思，  
2  才会相思，  
3  便害相思。  
4  
5 > 空一缕余香在此，  
6  盼千金游子何之。
```

平生不会相思, 才会相思, 便害相思。

空一缕余香在此, 盼千金游子何之。

### 2.3.1 引用的多层嵌套

区块引用可以嵌套（例如：引用内的引用），只要根据层次加上不同数量的 >：

```
1 >>> 锄禾日当午，汗滴禾下土。 - 李绅  
2  
3 >> 山有木兮木有枝，心悦君兮君不知。 - 越人歌  
4  
5 > 去年今日此门中，人面桃花相映红。 - 崔护
```

锄禾日当午, 汗滴禾下土。 - 李绅

山有木兮木有枝, 心悦君兮君不知。 - 越人歌

去年今日此门中, 人面桃花相映红。 - 题都城南庄

## 2.4 锚点

网页中, 锚点其实就是页内超链接, 也就是链接本文档内部的某些元素, 实现当前页面中的跳转。比如我这里写下一个锚点, 点击跳转到指定章节。

1 [\[点击跳转至区块引用\]\(#区块引用\)](#)

点击跳转至区块引用

## 2.5 代码区块

代码区块的建立是在每行加上 4 个空格或者一个制表符 (如同写代码一样)。如普通段落:

```
void main()
{
printf( "Hello, Markdown." );
}
```

代码区块:

```
1 void main()
2 {
3     printf("Hello, Markdown.");
4 }
```

**注意:** 需要和普通段落之间存在空行。

## 2.6 强调

Markdown 使用星号\*和底线\_作为标记强调字词的符号。

### 2.6.1 斜体

```
1 *花自飘零水自流*
```

花自飘零水自流

## 2.6.2 粗体

```
1 **花自飘零水自流**
```

花自飘零水自流

## 2.6.3 删除线

```
1 ~~花自飘零水自流~~
```

花自飘零水自流

## 2.7 列表

使用`.`、`+`、或`-`标记无序列表，如：

```
- (+*) 第一项 - (+*) 第二项 - (+*) 第三项
```

**注意：**标记后面最少有一个`_ 空格 _`或`_ 制表符 _`。若不在引用区块中，必须和前方段落之间存在空行。

效果：

- 第一项
- 第二项
- 第三项

有序列表的标记方式是将上述的符号换成数字，并辅以`.`，如：

1. 第一项
2. 第二项
3. 第三项

效果:

1. 第一项
2. 第二项
3. 第三项

## 2.8 分割线

分割线最常使用就是三个或以上\*, 还可以使用-和\_。

## 2.9 链接

Markdown 支持两种形式的链接语法: 行内式和参考式两种形式, 行内式一般使用较多。

### 2.9.1 行内式

[]里写链接文字, ()里写链接地址, ()中的 "" 中可以为链接指定 title 属性, title 属性可加可不加。title 属性的效果是鼠标悬停在链接上会出现指定的 title 文字。[链接文字](链接地址 "链接标题") 这样的形式。链接地址与链接标题前有一个空格。

```
1 [MPPL: Markdown to PDF with Pandoc via Latex](https://github.com/Dunky-Z/MPPL)
2 [MPPL: Markdown to PDF with Pandoc via Latex](https://github.com/Dunky-Z/MPPL "MPPL")
```

MPPL: Markdown to PDF with Pandoc via Latex

MPPL: Markdown to PDF with Pandoc via Latex

### 2.9.2 参考式

参考式超链接一般用在学术论文上面, 或者另一种情况, 如果某一个链接在文章中多处使用, 那么使用引用的方式创建链接将非常好, 它可以让你对链接进行统一的管理。

参考式链接分为两部分, 文中的写法 [链接文字][链接标记], 在文本的任意位置添加 [链接标记]: 链接地址 "链接标题", 链接地址与链接标题前有一个空格。

```
1 全球最大的搜索引擎网站是[Google][1]。  
2  
3 [1]:http://www.google.com "Google"
```

全球最大的搜索引擎网站是 [Google](#)。

## 2.10 图片

图片的创建方式与超链接相似, 而且和超链接一样也有两种写法, 行内式和参考式写法。

语法中图片 Alt 的意思是如果图片因为某些原因不能显示, 就用定义的图片 Alt 文字来代替图片。图片 Title 则和链接中的 Title 一样, 表示鼠标悬停与图片上时出现的文字。Alt 和 Title 都不是必须的, 可以省略, 但建议写上。

### 2.10.1 图片行内式

![图片Alt](图片地址 "图片Title")

```
1 ![MarkdownLogo](../img/Markdown-mark.png "MarkdownLogo")
```

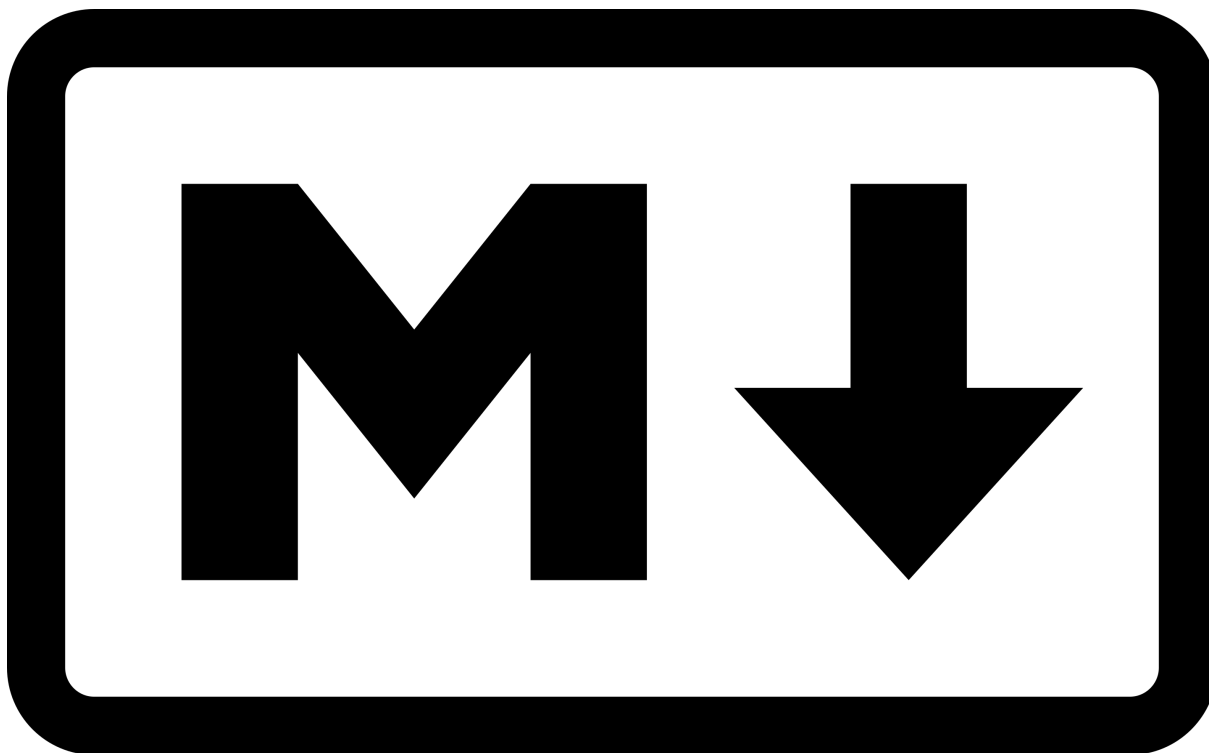


图 1. MarkdownLogo

### 2.10.2 图片参考式

在文档要插入图片的地方写 `![图片Alt] [标记]`。

在文档的最后写上 `[标记]:图片地址 "Title"`。

```
1  ![MarkdownLogo][MarkdownLogo]
2
3  [MarkdownLogo]:../img/Markdown-mark.png "MarkdownLogo"
```

### 2.11 反斜杠\

相当于**反转义**作用。使符号成为普通符号。

## 2.12 代码

对于程序员来说这个功能是必不可少的, 插入程序代码的方式有两种, 一种是利用缩进 (Tab), 另一种是利用 “ ” 符号 (一般在 ESC 键下方) 包裹代码。

- 插入行内代码, 即插入一个单词或者一句代码的情况, 使用 ‘code’ 这样的形式插入。
- 插入多行代码, 可以使用缩进或者 “ ‘code’ ”, 具体看示例。

### 2.12.1 代码行内式

```
1 PHP打印堆栈信息 `debug_backtrace()`。
```

```
PHP 打印堆栈信息 debug_backtrace()。
```

### 2.12.2 缩进式多行代码

缩进 4 个空格或是 1 个制表符。

一个代码区块会一直持续到没有缩进的那一行 (或是文件结尾)。

```
1     $closure = function () use($name) {
2         return $name;
3     }
```

```
1 $closure = function () use($name) {
2     return $name;
3 }
```

### 2.12.3 用六个 ‘ 包裹多行代码

```
1 > ```c # 为了能够在markdown里演示, 所以加了>符号
2 #include <stdio.h>
3 int main()
4 {
5     printf("Hello, World!\n");
6     return 0;
7 }
8 > ```
```

```
1 #include <stdio.h>
2 int main()
3 {
4     printf("Hello, World!\n");
5     return 0;
6 }
```

## 2.12.4 内容目录

在段落中填写 `[TOC]` 以显示全文内容的目录结构。

## 2.13 表格

1. 不管是哪种方式, 第一行为表头, 第二行分隔表头和主体部分, 第三行开始每一行为一个表格行。
2. 列于列之间用管道符`|`隔开。原生方式的表格每一行的两边也要有管道符。
3. 第二行还可以为不同的列指定对齐方向。默认为左对齐, 在-右边加上:就右对齐。

简单方式:

```
1 诗名|作者|朝代
2  -|-|-
3 白头吟|卓文君|两汉
4 锦瑟|李商隐|唐代
5 登科后|孟郊|唐代
```

诗名	作者	朝代
白头吟	卓文君	两汉
锦瑟	李商隐	唐代
登科后	孟郊	唐代

原生方式:

```
1 |诗名|作者|朝代|
2 |-|-|-|
3 |白头吟|卓文君|两汉|
4 |锦瑟|李商隐|唐代|
```



5 | 登科后 | 孟郊 | 唐代 |

诗名	作者	朝代
白头吟	卓文君	两汉
锦瑟	李商隐	唐代
登科后	孟郊	唐代

为表格第二列指定方向:

```
1 诗名 | 名句
2  -|-:
3  梦微之 | 君埋泉下泥销骨。
4  上邪 | 上邪，我欲与君相知，长命无绝衰。
```

诗名	名句
梦微之	君埋泉下泥销骨。
上邪	上邪，我欲与君相知，长命无绝衰。

## 2.14 注脚

在需要添加注脚的文字后加上脚注名字<sup>[[注脚名字](#)]</sup>, 称为加注。然后在文本的任意位置 (一般在最后) 添加脚注, 脚注前必须有对应的脚注名字。

```
1 使用 Markdown[^1] 可以效率的书写文档，直接转换成 HTML[^2]。
2
3 [^1]: Markdown 是一种纯文本标记语言
4
5 [^2]: HyperText Markup Language 超文本标记语言
```

## 2.15 LaTeX 公式

### 2.15.1 \$ 表示行内公式

```
1 质能守恒方程可以用一个很简洁的方程式  $E=mc^2$  来表达。
```

质能守恒方程可以用一个很简洁的方程式  $E = mc^2$  来表达。

### 2.15.2 \$\$ 表示整行公式

```
1  $$\sum_{i=1}^n a_i=0$$
2  $$f(x_1,x_x,\ldots,x_n) = x_1^2 + x_2^2 + \cdots + x_n^2 $$
3  $$\sum^{j-1}_{k=0}\{\widehat{\gamma}_{kj} z_k\}$$
```

$$\sum_{i=1}^n a_i = 0$$

## 3 尝试一下

- **Chrome** 下的插件诸如[stackedit](#)与[markdown-here](#)等非常方便，也不用担心平台受限。
- **在线的** [dillinger.io](#) 评价也不错
- **Windows** 下的 MarkdownPad 也用过，不过免费版的体验不是很好。
- **Mac** 下的 Mou 是国人贡献的，口碑很好。
- **Linux** 下的 ReText 不错。

当然，最终境界永远都是笔下是语法，心中格式化。

---

**注意：**不同的 Markdown 解释器或工具对相应语法（扩展语法）的解释效果不尽相同，具体可参见工具的使用说明。虽然有人想出面搞一个所谓的标准化的 Markdown，[没想到还惹怒了健在的创始人 John Gruber](#)。\*\*\*\* 以上基本是所有 traditonal markdown 的语法。

关于其它扩展语法可参见具体工具的使用说明。

## 4 参考资料

1. [Markdown 基本语法](#)。
2. [cdoco/markdown-syntax: Markdown 语法详解](#)。