

# Markdown语法介绍

study

Markdown是一种用来写作的轻量级「标记语言」，它用简洁的语法代替排版，而不像一般我们用的字处理软件 Word 或 Pages 有大量的排版、字体设置。它使我们专心于码字，用「标记」语法，来代替常见的排版格式。

优点:

- 格式简单，即使在纯文本模式下也很易读。
- 让你专注于内容而不是排版。
- 格式转换方便，Markdown 的文本你可以轻松可以通过各种工具,如 [pandoc](#) 转换为 html、pdf、epub、mobi等格式。
- 应用广泛。在 GitHub, Reddit, StackOverflow 等一大批网站的影响下，在互联网上应用很广。
- 可以使用git、svn进行版本管理，协作也更方便。

缺点:

- 不支持复杂的格式（如复杂的表格, 艺术字等）。
- 格式难于自定义 (想要彩色的文本? no)

## Markdown基本语法

### 1. 斜体和粗体

使用 \* 和 \*\* 表示斜体和粗体, 也可以使用下划线'\_'。建议使用 '\*', 方便阅读。

示例:

- 这是 *\*斜体\**, 这是 **\*\*粗体\*\***, 这是\*\*\*粗斜体\*\*\*.
- 这是 \_斜体\_, 这是 \_\_粗体\_\_, 这是\_\_\_粗斜体\_\_\_.

显示效果如下:

a. 这是 *斜体*, 这是 **粗体**, 这是***粗斜体***.

b. 这是 *斜体*, 这是 **粗体**, 这是***粗斜体***.

## 2. 分级标题

行首加井号表示不同级别的标题 (H1-H6), 例如: # H1, ## H2, ### H3, #### H4.

示例:

1. # 这是一个一级标题
2. ## 这是一个二级标题
3. ### 这是一个三级标题
4. ##### 这是一个六级标题

其中, 一级标题和二级标题也可以使用 === 和 --- 表示.

示例:

1. 这是一个一级标题 (3+ 个 '=', 紧跟在下一行)
2. =====
- 3.
4. 这是一个二级标题 (3+ 个 '-', 紧跟在下一行)
5. -----

## 3. 无序列表

使用 \*, +, - 表示无序列表.

示例:

- 无序列表项 一
- \* 无序列表项 二
- + 无序列表项 三

- 无序列表项 一
- 无序列表项 二

- 无序列表项 三

## 4. 有序列表

使用数字和点表示有序列表.

示例:

1. 有序列表项 一
3. 有序列表项 二
3. 有序列表项 三
5. 有序列表项 四

1. 有序列表项 一
2. 有序列表项 二
3. 有序列表项 三
4. 有序列表项 四

注意: Markdown目前并不care你的序号数字是多少, 它只会按1, 2, 3, 4...的顺序排下去.  
如, 上面编辑有序列表时使用的序号是1,3,3,5, 而显示的是1, 2, 3, 4.

## 5. 文字引用

使用 > 表示文字引用.

示例:

这是一段引用文字.

## 6. 行内代码块

使用 `代码` 表示行内代码块.

示例:

一起学习 `Markdown`.

## 7. 代码块

使用 *四个缩进空格* 表示代码块.

示例:

```
这是一个代码块，此行左侧有四个不可见的空格。
```

## 8. 加强代码块

支持多种编程语言的语法高亮的显示, 行号显示.

非代码示例:

```
1. $ sudo apt-get install vim-gnome
```

Python 示例:

```
1. @requires_authorization
2. def somefunc(param1='', param2=0):
3.     '''A docstring'''
4.     if param1 > param2: # interesting
5.         print 'Greater'
6.     return (param2 - param1 + 1) or None
7.
8. class SomeClass:
9.     pass
10.
11. >>> message = '''interpreter
12. ... prompt'''
```

JavaScript 示例:

```
1. /**
2.  * nth element in the fibonacci series.
```

```

3.    * @param n >= 0
4.    * @return the nth element, >= 0.
5.    */
6.    function fib(n) {
7.        var a = 1, b = 1;
8.        var tmp;
9.        while (--n >= 0) {
10.            tmp = a;
11.            a += b;
12.            b = tmp;
13.        }
14.        return a;
15.    }
16.
17.    document.write(fib(10));

```

## 9. 外链接

写在前面:

下面方法中, "描述" 中的内容是可见的, "说明" 中的内容是隐藏的, 当鼠标悬停在链接文字上时, 会显示出"说明"中的文字.

- [描述](链接地址 "说明")

示例:

这是去往 [LeMaker](#) 的链接.

- [描述][标记]

[标记]: 链接地址 "说明,可省略"

上面这行可在全文任何地方出现,一般统一写在文档最下面:

示例:

这是去往 [LeMaker](#) 的链接.

- 在方法2的基础上省略标记:

[链接][]

同样的, 上面这行可在全文任何地方出现, 一般统一写在文档最下面.

示例:

[lemaker.org](http://lemaker.org)

## 10. 插入图像

Markdown文档中不能直接添加图片, 当需要使用图片时, 应获取图片的外链接. 网上很多人推荐使用七牛.

当然, 有的编辑器也可以作用它自己的服务器, 并做到了支持'一键上传'的功能, 即可以将剪切中的图片直接 `Ctrl+V` 粘贴到文档中. 这些是编辑器扩展的功能.

插入图像与增加文字链接方法相似, 只是在最前面多个感叹号 "!":

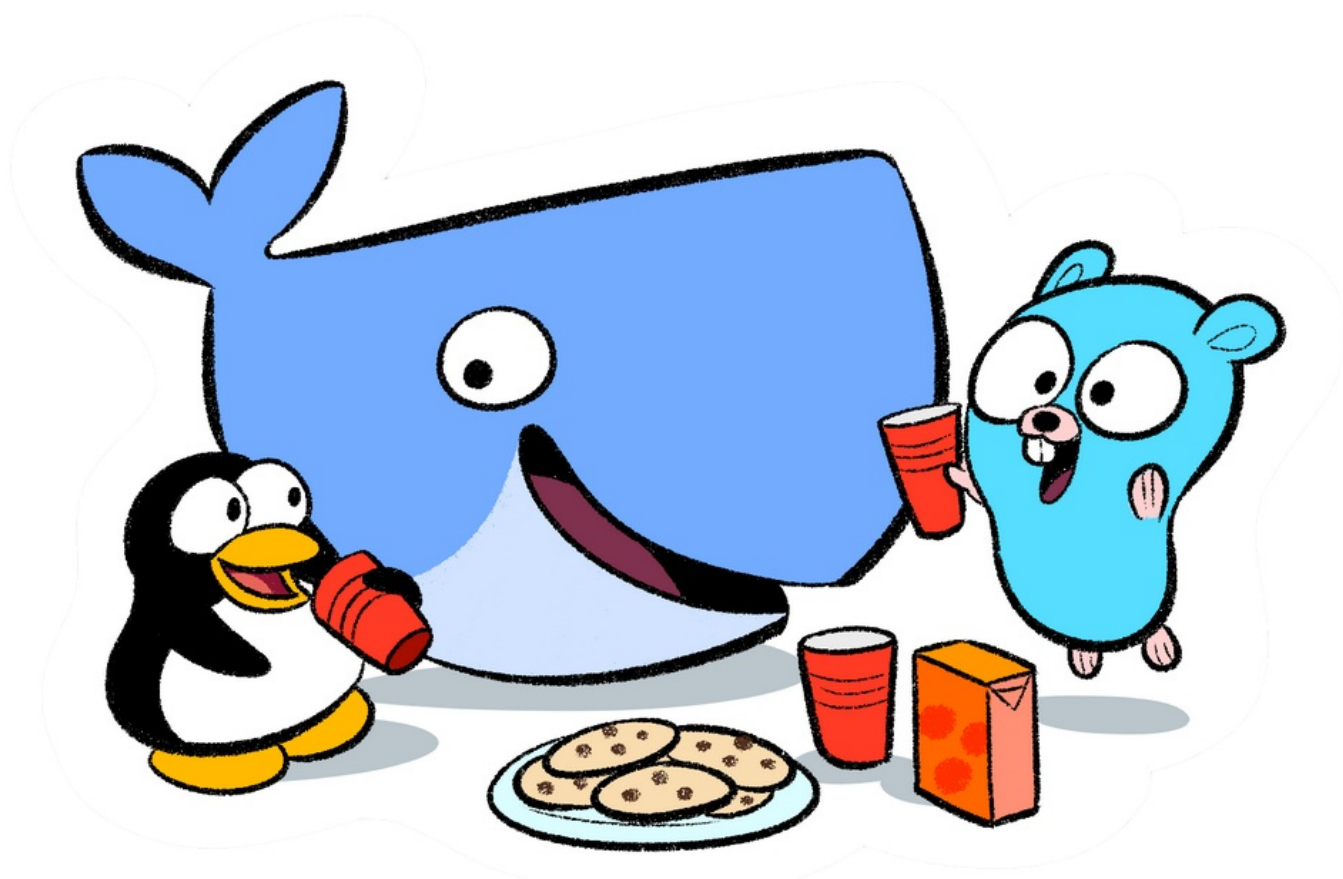
- ![描述](图片链接地址 "说明")

示例:



- ![描述][标记]  
[标记]: 图片外链 "说明"

示例:



- ![描述] []

[描述]: 图片外链 "说明"

示例:

```
13  activate: (state) ->
14    @subscriptions = new CompositeDisposable
15    @subscriptions.add atom.commands.add "atom-workspace
16      "activate-power-mode:toggle": => @toggle()
17
18
19
20    @activeItemSubscription = atom.workspace.onDidChange
21      @subscribeToActiveTextEditor()
22
23    @subscribeToActiveTextEditor()
24    @setupCanvas()      I
25
```

在后面两个示例中, 我把后面两个链接统一放到一文档最后, 编辑页显得更加干净整洁, 且链接都放在一起, 方便维护. 如链接失效或需替换时, 方便替换等.



## 11. 表格支持

Name	Age	Other
A	18	...
B	24	...
C	32	...

表格默认是左对齐的, 想要表格内容右对齐或居中对齐, 则在第二行添加冒号标记, 如上表格中, Age是右对齐, 冒号添加到右边, Other是居中对齐, 则两边都添加了冒号.

## Markdown扩展语法

**注意:** 基本的语法掌握后足以应对日常的文档编写工作了, 下面介绍几个扩展的语法, 这些不是所有的Markdown编辑器都支持的.

### 1. 内容目录

在段落中填写 `[TOC]` 以显示全文内容的目录结构.

#### Markdown语法介绍

##### Markdown基本语法

1. 斜体和粗体
2. 分级标题
3. 无序列表
4. 有序列表
5. 文字引用
6. 行内代码块
7. 代码块
8. 加强代码块
9. 外链接
10. 插入图像
11. 表格支持

## Markdown扩展语法

### 1. 内容目录

### 2. 删除线

### 3. 注脚

### 4. 流程图

示例

更多语法参考: [流程图语法参考](#)

### 5. 序列图

示例 1

示例 2

更多语法参考: [序列图语法参考](#)

### 6. Html 标签

### 7. 待办事宜 Todo 列表

### 8. LaTeX 公式

## 其它

### 1. 编辑器

### 2. 注意事项

换行

空行

一级标题

分隔符

链接

删除线

小尾巴

## 2. 删除线

github支持.

使用 ~~ 表示删除线. (需要加删除线的文本与 ~~ 之间不要有空格).

这是一段错误的文本.

## 3. 注脚

使用 [^keyword] 表示注脚。

这是一个注脚<sup>[1]</sup>的样例。

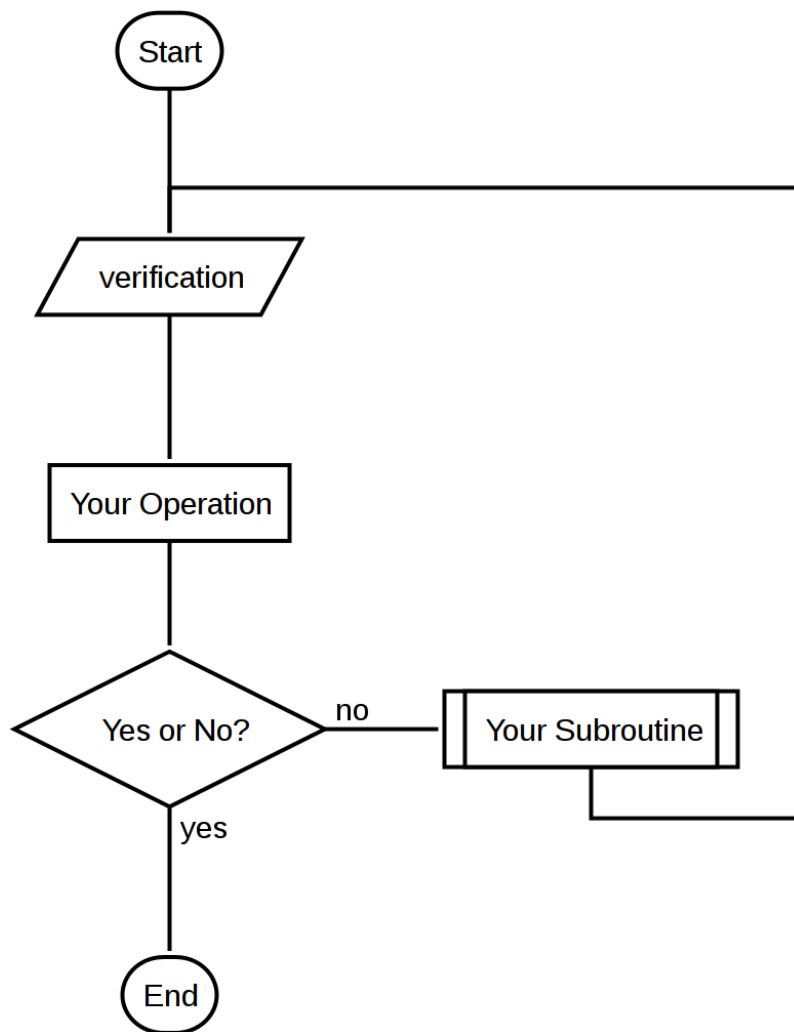
这是第二个注脚<sup>[2]</sup>的样例。

## 4. 流程图

### 示例

```
```flow
st=>start: Start:>https://www.zybuluo.com
io=>inputoutput: verification
op=>operation: Your Operation
cond=>condition: Yes or No?
sub=>subroutine: Your Subroutine
e=>end

st->io->op->cond
cond(yes)->e
cond(no)->sub->io
```
```

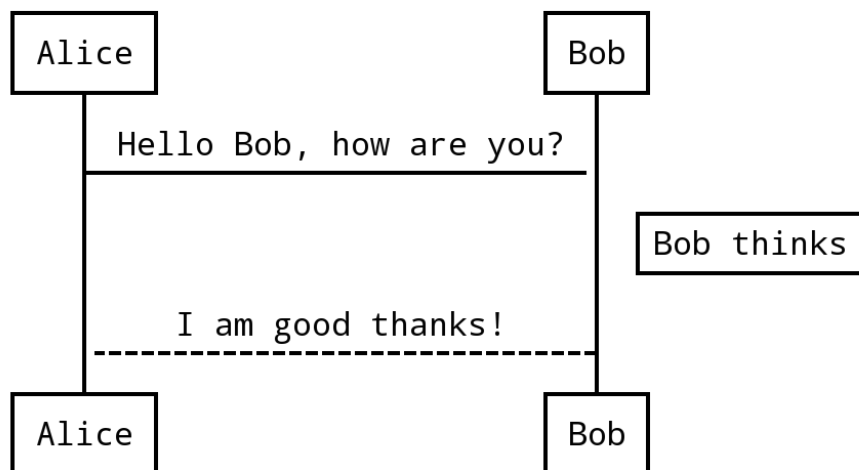


更多语法参考: [流程图语法参考](#)

## 5. 序列图

### 示例 1

```
```seq
Alice->Bob: Hello Bob, how are you?
Note right of Bob: Bob thinks
Bob-->Alice: I am good thanks!
```
```

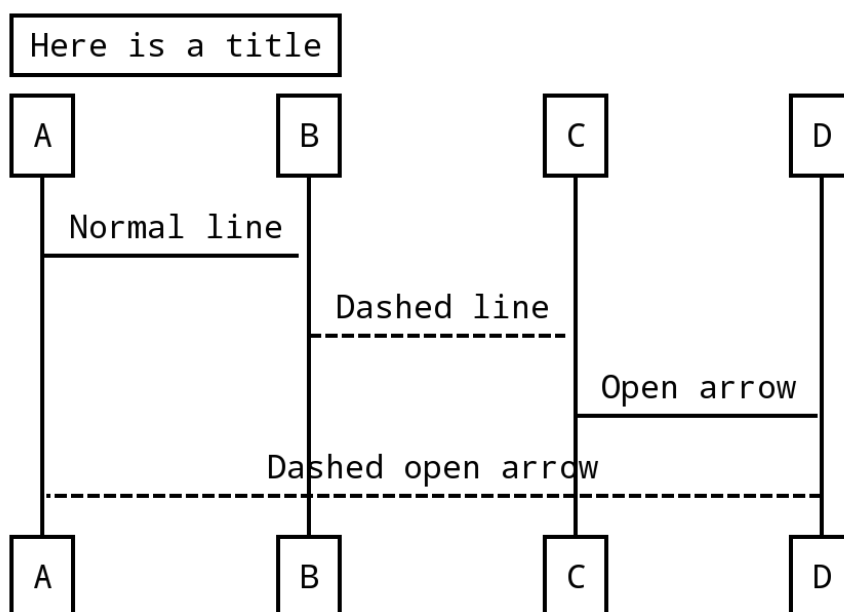


## 示例 2

```

```seq
Title: Here is a title
A->>B: Normal line
B-->>C: Dashed line
C->>D: Open arrow
D-->>A: Dashed open arrow
```

```



更多语法参考: [序列图语法参考](#)

## 6. Html 标签

Markdown是支持HTML语法的,如: 你可以用 Html 写一个纵跨两行的表格:

```
1. <table>
2.   <tr>
3.     <th rowspan="2">值班人员</th>
4.     <th>星期一</th>
5.     <th>星期二</th>
6.     <th>星期三</th>
7.   </tr>
8.   <tr>
9.     <td>李强</td>
10.    <td>张明</td>
11.    <td>王平</td>
12.  </tr>
13. </table>
```

| 值班人员 | 星期一 | 星期二 | 星期三 |
|------|-----|-----|-----|
|      | 李强  | 张明  | 王平  |

## 7. 待办事宜 Todo 列表

github支持

使用带有 [ ] 或 [x] ( 未完成或已完成 ) 项的列表语法撰写一个待办事宜列表, 并且支持子列表嵌套以及混用Markdown语法, github也支持To Do List, 例如:

- [ ] \*\*七月旅行准备\*\*
  - [ ] 准备邮轮上需要携带的物品
  - [ ] 浏览日本免税店的物品
  - [x] 购买蓝宝石公主号七月一日的船票

对应显示如下待办事宜 Todo 列表:

### ☐ 七月旅行准备

- ☒ 准备邮轮上需要携带的物品
- ☐ 浏览日本免税店的物品
- ☒ 购买蓝宝石公主号七月一日的船票

## 8. LaTeX 公式

\$ 表示行内公式：

质能守恒方程可以用一个很简洁的方程式  $E = mc^2$  来表达。

\$\$ 表示整行公式：

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

$$f(x_1, x_2, \dots, x_n) = x_1^2 + x_2^2 + \dots + x_n^2$$

$$\sum_{k=0}^{j-1} \hat{\gamma}_{kj} z_k$$

$$\textit{latex} f(x; \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

访问 [MathJax](#) 参考更多使用方法。

## 其它

### 1. 编辑器

由于Markdown编辑器不同, 各自对所扩展的语法的支持程度也有所不同, 有的编辑器支持的扩展语法各类比较少, 但是基本语法都是支持的.

下面推荐几个网上大多数人比较认可的:

mac: Mou

windows: marddownpad2 markpad2

linux: 没有呼声特别高的. 也有用vim + chrome插件

因为语法是相同的, 所以用什么编辑器都是可以的.

特别的有:

- atom

它是github写的一个号称'21世纪的编辑器', 开源, 并支持markdown, 有多个markdown相关的插件.

- Cmd Markdown(作业部落)

国内一个比较好的markdown编辑器, 在线编辑, 自动同步. 保存到本地要先'导出markdown'到本地.

另外, 它的上传本地图片和导出pdf文件是会员功能.

## 2. 注意事项

因为是写教程到github上, 并统一内部编写风格, 同时也为了避免不同编辑器之间的小的渲染差异, 建议在编写markdown文档时注意以下几点:

### 换行

换行时, 前一行的末尾要加上 **两个** 不可见的空格. 这个主要是因为在github上目前一定要这样, 虽然大多编辑器不用这么麻烦了.

### 空行

在 标题/列表/代码块/引用文字 的上/下各有一个空行.

这样可以防止不同编辑器之间的渲染差异, 保证在不同编辑器之间有相同的渲染结果.

### 一级标题

一级标题用于文档名, 每一个文档要有且只有一个一级标题. 并在下方加分隔符做显示区分.

由于markdown可以转换为不同的其它格式, 有的会把第一个标题转换为文档名. 且写一个文档名符合大多文档格式的习惯.

### 分隔符

当段落内容间差异较大或其它情况要使用分隔符时(单独一行3个或以上的'-'), 其上/下各空一行.

### 链接

不建议直接复制链接地址, 当链接地址过长时, 建议使用上面介绍的方法2或方法3. 地址统一写在文档最下方. 保持原文件清爽, 方便阅读.



另: [标记]: (英文冒号+一个空格)链接地址 "可省略的说明"

## 删除线

删除线与被包围内容间不要有空格.

## 小尾巴

以上仅做为建议,以本文做为参考, 有如疑问或不合理的地方, 请找Tab

---

## Markdown官方文档-语法

---

[1] 这是一个 *注脚* 的 **文本**。↩

[2] 这是另一个 *注脚* 的 **文本**。↩