什么是 Markdown ？

> Markdown 是一种可以使用普通文本编辑器编写的标记语言，通过简单的标记语法，它可以使普通文本内容具有一定的格式。

目录

分级标题

第一种写法:

```
这是一个一级标题
==========

这是一个二级标题
----------
```

> # 这是一个一级标题
>
> ## 这是一个二级标题

第二种写法:

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

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

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:

```
* * *

***

*****

- - -

---------------------------------------
```

>

超链接

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。

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。

```
[Markdown Syntax](https://github.com/cdoco/markdown-syntax)
[Markdown Syntax](https://github.com/cdoco/markdown-syntax "Markdown Syntax")
```

> Markdown Syntax
>
> Markdown Syntax

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。

```
全球最大的搜索引擎网站是[Google][1]。

[1]:http://www.google.com "Google"
```

> 全球最大的搜索引擎网站是 Google。

自动链接

Markdown 支持以比较简短的自动链接形式来处理网址和电子邮件信箱，只要是用 `<>` 包起来，Markdown 就会自动把它转成链接。一般网址的链接文字就和链接地址一样，例如：

```
<https://google.com/>

<ocdoco@gmail.com>
```

> https://google.com/
>
> ocdoco@gmail.com

区块引用

区块引用需要在被引用的文本前加上 > 符号。

```
> 这是一个区块引用实例，

> 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 > ：

```
> 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。
```

> 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。

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 > ：

```
>>> 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。 - 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

引用其它要素

引用的区块内也可以使用其他的 Markdown 语法，包括标题、列表、代码区块等:

```
> - 入我相思门，知我相思苦。
>
> - 长相思兮长相忆，短相思兮无穷极。
>
> - 给出一些例子代码:
> ```markdown
>   return debug_backtrace();
> ```
```

> - 入我相思门, 知我相思苦。
> - 长相思兮长相忆, 短相思兮无穷极。
> - 给出一些例子代码:
>
>
>   ```
>   return debug_backtrace();
>   ```

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, 点击回到目录, 就能跳转到目录。 在目录中点击这一节, 就能跳过来。还有下一节的注脚。这些根本上都是用锚点来实现的。

```
**[⬆ top](#什么是-markdown-)**
```

⬆ top

强调

Markdown 使用星号 * 和底线 _ 作为标记强调字词的符号。

斜体

花自飘零水自流

> *花自飘零水自流*

粗体

花自飘零水自流

> **花自飘零水自流**

删除线

~~花自飘零水自流~~

> ~~花自飘零水自流~~

列表

使用 *，+，- 表示无序列表。

无序列表

- 白头吟
- 击鼓
- 断句

> - 白头吟
> - 击鼓
> - 断句

有序列表

有序列表则使用数字接着一个英文句点。

```
1. 白头吟
2. 击鼓
3. 断句
```

> 1. 白头吟
> 2. 击鼓
> 3. 断句

包含引用的列表

如果要在列表项目内放进引用，那 > 就需要缩进:

```
* 菩提偈:

  > 菩提本无树，明镜亦非台。
  > 本来无一物，何处惹尘埃！
```

- 菩提偈:
  > 菩提本无树，明镜亦非台。
  > 本来无一物，何处惹尘埃！

插入图片

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。

图片行内式

```
![图片Alt](图片地址 "图片Title")

  ![哆啦A梦](https://cdoco.com/images/duolaameng.jpeg "哆啦A梦")
```

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"` 。

```
![哆啦A梦][duolaameng]

[duolaameng]:https://cdoco.com/images/duolaameng.jpeg "哆啦A梦"
```

 哆啦A梦

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:

```
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
```

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:

```
|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|
```

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:

```
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。
```

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。

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。

代码行内式

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

> PHP打印堆栈信息 `debug_backtrace()` 。

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)。

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

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

用六个 ` 包裹多行代码

```
```php
$closure = function () use($name) {
  return $name;
}
` ``
```

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

HTML 原始码

在代码区块里面，`&` 、 `<` 和 `>` 会自动转成 HTML 实体, 这样的方式让你非常容易使用 Markdown 插入范例用的 HTML 原始码, 只需要复制贴上, 剩下的 Markdown 都会帮你处理，例如:
```

```
<table>
    <tr>
        <th rowspan="2">值班人员</th>
        <th>星期一</th>
        <th>星期二</th>
        <th>星期三</th>
    </tr>
    <tr>
        <td>李强</td>
        <td>张明</td>
        <td>王平</td>
    </tr>
</table>
```

值班人员	星期一	星期二	星期三
	李强	张明	王平

内容目录

在段落中填写 `[TOC]` 以显示全文内容的目录结构。

注脚

在需要添加注脚的文字后加上脚注名字 `[^注脚名字]`，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。

```
使用 Markdown[^1]可以效率的书写文档，直接转换成 HTML[^2]。

[^1]: Markdown 是一种纯文本标记语言

[^2]: HyperText Markup Language 超文本标记语言
```

PS: github 不支持注脚 😂

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。
```

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

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

过去 github 不支持 LaTeX 公式, 但是有个折中的解决方案, 使用 codecogs, 例如:

```
![](https://latex.codecogs.com/gif.latex?\sum_{i=1}^n%20a_i=0)
![](https://latex.codecogs.com/gif.latex?f\(x_1,x_x,\ldots,x_n\)%20=%20x_1^2%20+%20x_2^2%20+%20\
```

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

$$f(x_1,x_x,\ldots,x_n) = x_1^2 + x_2^2 + \cdots + x_n^2$$

现在 github 已经支持 LaTeX 公式:

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

$$f(x_1,x_x,\ldots,x_n) = x_1^2 + x_2^2 + \cdots + x_n^2$$

流程图

流程图大致分为两段, 第一段是定义元素, 第二段是定义元素之间的走向。

定义元素的语法 `tag=>type: content:>url` 。

- tag就是元素名字。
- type是这个元素的类型, 有6中类型,分别为:

type	含义
start	开始
end	结束
operation	操作
subroutine	子程序
condition	条件
inputoutput	输入或产出

content 就是在框框中要写的内容, 注意type后的冒号与文本之间一定要有个空格。

用 `->` 来连接两个元素, 需要注意的是condition类型, 因为他有yes和no两个分支, 所以要写成:

```
c2(yes)->io->e
c2(no)->op2->e
```

```flow
st=>start: Start:>https://www.markdown-syntax.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
```

PS: github 不支持流程图 😂

时序图

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

PS: github 不支持时序图 😂

待办事宜列表

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, 例如:

```
* <input type="checkbox" class="task-list-item-checkbox" > 早起跑步
* <input type="checkbox" class="task-list-item-checkbox"  checked> 看书
```

- [ ] 早起跑步
- [x] 看书

↑ top