

目 录

Markdown语法

Markdown语法

- [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“`”](#)

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。

标题

两种形式：

1) 使用 `=` 和 `-` 标记一级和二级标题。

一级标题

```
=====
```

二级标题

```
-----
```

效果：

一级标题

二级标题

2) 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

效果：

一级标题

二级标题

三级标题

四级标题

五级标题

六级标题

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）。

区块引用

在段落的每行或者只在第一行使用符号 `>` ,还可使用多个嵌套引用，如：

```
> 区块引用
>> 嵌套引用
```

效果：

```
    区块引用
    |
    |  嵌套引用
```

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。

强调

在强调内容两侧分别加上 `*` 或者 `_` ，如：

```
*斜体*, _斜体_  
**粗体**, __粗体__
```

效果：

```
斜体, 斜体  
粗体, 粗体
```

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, 并辅以一个 `.`，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
```

分割线

分割线最常使用就是三个或以上 `*`，还可以使用 `-` 和 `_`。

链接

链接可以由两种形式生成：行内式和参考式。
行内式：

```
[掘金量化](https://www.myquant.cn “掘金量化”)。
```

效果：

```
掘金量化。
```

参考式：

```
[掘金量化][1]
[掘金社区][2]
[1]:https://www.myquant.cn “掘金量化”
[2]:https://www.myquant.cn/community “掘金社区”
```

效果：

[掘金量化](#)
[掘金社区](#)

注意：上述的 `[1]:https://www.myquant.cn "Markdown"` 不出现在区块中。

4.9 图片

添加图片的形式和链接相似，只需在链接的基础上前方加一个 `!`。

示例：

```
1. ![掘金量化](/static/images/logo-share.png)
```

效果：



反斜杠 \

相当于反转义作用。使符号成为普通符号。

符号 ` ``

起到标记作用。如：

```
`ctrl+a`
```

效果：

`ctrl+a`