

1、标题的几种写法：

第一种：

代码：

```
# 一级标题
## 二级标题
### 三级标题
#### 四级标题
##### 五级标题
##### 六级标题
##### 七级标题
```

一级标题

二级标题

三级标题

四级标题

五级标题

六级标题

七级标题

前面带#号，后面带文字，分别表示h1-h6,上面可以看出，只到h6，而且h1下面会有一条横线，注意，#号后面有空格

第二种：

代码：

```
这是一级标题
=
这是二级标题
-
```

这是一级标题

=

这是二级标题

-

这种方式只能表示一级和二级标题这种方式好像只能表示一级和二级标题，而且=和-的数量没有限制，只要大于一个就行

第三种：

代码：

```
# 一级标题 #
## 二级标题 ##
```

一级标题

二级标题

这里的标题支持h1-h6，这个比较好理解，相当于标签闭合，注意，标题与#号要有空格

那既然3种都可以使用，可不可以混合使用呢？我试了一下，是可以的，但是为了让页面标签的统一性，不建议混合使用，推荐使用第一种，比较简洁，全面。

2、列表

列表分为有序列表和无序列表

无序列表

代码:

```
* 1
* 2
* 3
+ 1
+ 2
+ 3
- 1
- 2
- 3
```

- 1
- 2
- 3
- 1
- 2
- 3
- 1
- 2
- 3

可以看到，无序列表可以用*， +， - 来创建，用在线编辑器看，实际上是转换成了ul>li，所以使用哪个都可以，推荐使用*吧

有序列表

代码:

```
1. 列表1
2. 列表2
3. 列表3
```

1. 列表1
2. 列表2
3. 列表3

有序列表就相对简单一点，只有这种方式，注意，数字后面的点只能是英文的点，特别注意，有序列表的序号是根据第一行列表的数字顺序来的，比如说：

代码:

```
3. 列表1
2. 列表2
1. 列表3
```

1. 列表1

- 2. 列表2
- 3. 列表3

乱序

代码:

- ```
3. 列表1
9. 列表2
1. 列表3
```

- 1. 列表1
- 2. 列表2
- 3. 列表3

### 3、区块引用

比如说，你想对某个部分做的内容做一些说明或者引用某某的话等，可以用这个语句

代码:

- ```
* 列表1
> 这是毛主席说的
* 列表2
> 注意:
```

- 列表1
 - 这是毛主席说的
- 列表2
 - 注意:

无序列表下方的便是引用，可以有多种用途，看你的需求了，用法就是在语句前面加一个>，注意是英文的那个右尖括号，注意空格

引用因为是一个区块，理论上应该是可以放什么内容都可以放，比如说：标题，列表，引用等等

代码:

- ```
>## 标题
>* 列表1
> > 这是毛主席说的
>* 列表2
> > 注意:
```

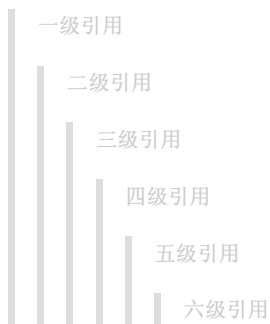
#### 标题

- 列表1
  - 这是毛主席说的
- 列表2
  - 注意:

将上面的代码稍微改一下，全部加上引用标签，就变成了一个大的引用，还有引用里面还有引用，那引用嵌套引用还没有别的写法呢？

代码:

- ```
> 一级引用
>> 二级引用
>>> 三级引用
>>>> 四级引用
>>>>> 五级引用
>>>>>> 六级引用
```



想要在上一次引用中嵌套一层引用，只需多加一个>，理论上可以无限嵌套，我就不整那么多了，注意：多层嵌套的>是不需要连续在一起的，只要在一行就可以了，中间允许有空格，但是为了好看，还是把排版搞好吧

4、华丽的分割线

分割线可以由* - _（星号，减号，底线）这3个符号的至少3个符号表示，注意至少要3个，且不需要连续，有空格也可以

代码:

```
***
*****
---
- - -
* * *
-----
- - -
```



应该看得懂吧，但是为了代码的排版好看，你们自己定规则吧，前面有用到星号，建议用减号

5、链接

支持2种链接方式：行内式和参数式，不管是哪一种，链接文字都是用 [方括号] 来标记。

行内式

代码:

```
[妙语连珠](http://www.baidu.com "百度")是什么意思
```

妙语连珠是什么意思

行内式的链接格式是：链接的文字放在[]中，链接地址放在随后的（）中，举一反三，经常出现的列表链接就应该这样写：

代码:

```
* [列表1](http://www.baidu.com)
* [列表2](http://www.baidu.com)
```

- [列表1](#)
- [列表2](#)

链接还可以带title属性，好像也只能带title，带不了其他属性，注意，是链接地址后面空一格，然后用引号引起来

参数式

代码:

```
这里是[百度][link1], 这里是[谷歌][link2]
[link1]: http://www.baidu.com "百度"
[link2]: http://www.google.com "谷歌"
```

这里是[百度](#), 这里是[谷歌](#)

这就好理解了, 就是把链接当成参数, 适合多出使用相同链接的场景, 注意参数的对应关系, 参数定义时, 这3种写法都可以:

```
[foo]: http://example.com/ "Optional Title Here"

[foo]: http://example.com/ 'Optional Title Here'

[foo]: http://example.com/ (Optional Title Here)
```

还支持这种写法, 如果你不想混淆的话:

```
[foo]: <http://example.com/> "Optional Title Here"
```

隐式链接

其实还有一种隐式链接的写法:

代码: `[google][] [google]: http://google.com/`
[google](#)

链接的定义可以放在文件中的任何一个地方, 我比较偏好直接放在链接出现段落的后面, 你也可以把它放在文件最后面, 就像是注解一样。


6、图片

图片也有2种方式: 行内式和参数式。
用法跟链接的基本一样, 唯一的不同就是, 图片前面要写一个! (这是必须的)

行内式

代码:


```
![东财](http://www.eastmoney.com/favicon.ico "东财")东财标识
```

东财标识

参数式

代码:

```
[东财]: http://www.eastmoney.com/favicon.ico "东财"
参数式图片, 这里是![东财]
```

参数式图片, 这里是

7、代码框

如果代码量比较少, 只有单行的话, 可以用单反引号包起来, 如下:

代码:

```
`<p><a href="http://www.baidu.com" title="百度" ></a></p>`
```

```
<p><a href="www.baidu.com" title="百度" ></a></p>
```

多行用三个反引号(MarkdownPad不识别，推荐使用制表符。如果想让MarkdownPad支持，需设置Tools>Options>Markdown>Markdown Process为GitHub Flavored Markdown(Offline))，如果要写注释，可以在反引号后面写，如下：

代码:

```
` `` 注释
<p><a href="www.baidu.com" title="百度" ></a></p>
<p><a href="www.baidu.com" title="百度" ></a></p>
<p><a href="www.baidu.com" title="百度" ></a></p>
` ``
```

```
<p><a href="www.baidu.com" title="百度" ></a></p>
<p><a href="www.baidu.com" title="百度" ></a></p>
<p><a href="www.baidu.com" title="百度" ></a></p>
```

8、表格

如果想让MarkdownPad支持，需设置Tools>Options>Markdown>Markdown Process为GitHub Flavored Markdown(Offline)或者Markdown(Extra).

代码:

```
| name | age | sex |
|:----:|:----:|:----:|
| tony | 20 | 男 |
| lucy | 18 | 女 |
```

| name | age | sex |
|------|-----|-----|
| tony | 20 | 男 |
| lucy | 18 | 女 |

代码:

```
表头1 | 表头2
---|---
Content Cell | Content Cell
Content Cell | Content Cell
```

| 表头1 | 表头2 |
|--------------|--------------|
| Content Cell | Content Cell |
| Content Cell | Content Cell |

代码:

```
学号|姓名|分数
-|-|-
小明|男|75
小红|女|79
```

| 学号 | 姓名 | 分数 |
|----|----|----|
| 小明 | 男 | 75 |
| 小红 | 女 | 79 |

从这3种不同写法看，表格的格式不一定要对的非常起，但是为了好看，对齐肯定是最好的，第一种的分割线后面的冒号表示对齐方式，写在左边表示左对齐，右边为右对齐，两边都写表示居中，还是有点意思的，不过现实出来的结果是，表格外面并没有线框包起来

9、强调

代码:

```
*字体倾斜*
_字体倾斜_
**字体加粗**
__字体加粗__
```

字体倾斜
字体倾斜
字体加粗
字体加粗

一个星号或者是一个下划线包起来，会转换为 倾斜，如果是2个，会转换为加粗

10、转义

代码:

```
* \\
* \`
* \~
* \*
* \_
* \-
* \+
* \.
* \!
```

- \
- `
- ~
- *
- _
- -
- +
- .
- !

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：

```
\  反斜线
`  反引号
*  星号
_  底线
{} 花括号
[] 方括号
() 括弧
# 井字号
+ 加号
- 减号
. 英文句点
! 惊叹号
```

11、删除线

代码:

```
~~请删掉我吧~~
```

~~请删掉我吧~~

12、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，只要是用方括号包起来，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：

```
<http://example.com/>
```

Markdown 会转为：

<http://example.com/>

邮址的自动链接也很类似，只是 Markdown 会先做一个编码转换的过程，把文字字符转成 16 进位码的 HTML 实体，这样的格式可以糊弄一些不好的邮址收集机器人，例如：

```
<address@example.com>
```

Markdown 会转为：

address@example.com