

- [0. 目录](#)
- [1. 斜体和粗体](#)
- [2. 分级标题](#)
- [3. 超链接](#)
  - [3.1. 行内式](#)
  - [3.2. 参考式](#)
  - [3.3. 自动链接](#)
- [4. 锚点](#)
- [5. 列表](#)
  - [5.1. 无序列表](#)
  - [5.2. 有序列表](#)
  - [5.3. 定义型列表](#)
  - [5.3. 列表缩进](#)
  - [5.4. 包含段落的列表](#)
  - [5.5. 包含引用的列表](#)
  - [5.6. 包含代码区块的引用](#)
  - [5.7. 一个特殊情况](#)
- [6. 引用](#)
  - [6.1. 引用的多层嵌套](#)
  - [6.2. 引用其它要素](#)

- [7. 插入图像](#)
  - [7.1. 行内式](#)
  - [7.2. 参考式](#)
- [8. 内容目录](#)
- [9. 注脚](#)
- [10. LaTeX 公式](#)
  - [10.1. \\$ 表示行内公式:](#)
  - [10.2 \\$\\$ 表示整行公式:](#)
- [11. 流程图](#)
- [12. 表格](#)
- [13. 分隔线](#)
- [14. 代码](#)
  - [8.1. 行内式](#)
  - [8.2. 缩进式多行代码](#)
  - [8.3. 用六个`包裹多行代码](#)
  - [8.4. HTML 原始码](#)

---

## 1. 斜体和粗体

代码:

1. `*斜体*或_斜体_`
2. `**粗体**`
3. `***加粗斜体***`

#### 4. ~~删除线~~

显示效果：

- 这是一段斜体
- **这是一段粗体**
- 这是一段加粗斜体
- ~~这是一段删除线~~

---

## 2. 分级标题

第一种写法：

1. 这是一个一级标题

2. =====

3. 这是一个二级标题

4. -----  
-----

第二种写法：

1. # 一级标题

2. ## 二级标题

3. ### 三级标题

4. #### 四级标题

5. ##### 五级标题

6. ##### 六级标题

由于用了[TOC]标记编辑器会把所有标题写到目录大纲中，在这里写的演示标题也会列进去，所以就不演示了。同学们自己在编辑器中观察，很简单，一级标题字号最大，依级递减。

---

## 3. 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。

### 3.1. 行内式

语法说明：

- `[]`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。

代码：

```
1. 欢迎来到[梵居闹市]
   (http://blog.leanote.com/freewalk)
2. 欢迎来到[梵居闹市]
   (http://blog.leanote.com/freewalk "梵居闹市")
```

显示效果：

欢迎来到[梵居闹市](http://blog.leanote.com/freewalk)

欢迎来到[梵居闹市](http://blog.leanote.com/freewalk)

### 3.2. 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。

语法说明：

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。

如果链接文字本身可以做为链接标记，你也可以写成`[链接文字][链接文字]: 链接地址`的形式，见代码的最后一行。

代码：

1. 我经常去的几个网站[Google][1]、[Leanote][2]以及[自己的博客][3]
2. [Leanote 笔记][2]是一个不错的[网站][1]。
3. [1]:`http://www.google.com "Google"`
4. [2]:`http://www.leanote.com "Leanote"`
5. [3]:`http://http://blog.leanote.com/freewalk "梵居闹市"`
6. [网站]:`http://http://blog.leanote.com/freewalk`

显示效果：

我经常去的几个网站[Google](http://www.google.com)、[Leanote](http://www.leanote.com)以及[自己的博客](http://http://blog.leanote.com/freewalk)  
[Leanote 笔记](http://http://blog.leanote.com/freewalk)是一个不错的[网站](http://http://blog.leanote.com/freewalk)。

### 3.3. 自动链接

语法说明：

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：

代码：

1. `<http://example.com/>`
2. `<address@example.com>`

显示效果：

<http://example.com/>  
[address@example.com](mailto:address@example.com)

---

## 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。

注意：

1. Markdown Extra 只支持在标题后插入锚点，其它地方无效。
2. Leanote 编辑器右侧显示效果区域暂时不支持锚点跳转，所以点来点去发现没有跳转不必惊慌，但是你发布成笔记或博文后是支持跳转的。

语法描述：

在你准备跳转到的指定标题后插入锚点 {#标记}，然后在文档的其它地方写上连接到锚点的链接。

代码：

- ```
1. ## 0. 目录{#index}
2. 跳转到[目录](#index)
```

显示效果：

跳转到[目录](#)

---

## 5. 列表

### 5.1. 无序列表

使用 \*, +, - 表示无序列表。

代码：

- ```
1. - 无序列表项 一
2. - 无序列表项 二
3. - 无序列表项 三
```

显示效果：

- 无序列表项 一
- 无序列表项 二
- 无序列表项 三

## 5.2. 有序列表

有序列表则使用数字接着一个英文句点。

代码：

1. 有序列表项 一
2. 有序列表项 二
3. 有序列表项 三

显示效果：

1. 有序列表项 一
2. 有序列表项 二
3. 有序列表项 三

## 5.3. 定义型列表

语法说明：

定义型列表由名词和解释组成。一行写上定义，紧跟一行写上解释。解释的写法：紧跟一个缩进(Tab)

代码：

1. Markdown
2. : 轻量级文本标记语言，可以转换成html，pdf等格式  
(左侧有一个可见的冒号和四个不可见的空格)
3. 代码块 2
4. : 这是代码块的定义 (左侧有一个可见的冒号和四个不可见的空格)
5. 代码块 (左侧有八个不可见的空格)

显示效果：

Markdown

轻量级文本标记语言，可以转换成html，pdf等格式

代码块 2

这是代码块的定义

### 1. 代码块（左侧有八个不可见的空格）

## 5.3. 列表缩进

语法说明：

列表项目标记通常是放在最左边，但是其实也可以缩进，最多 3 个空格，项目标记后面则一定要接着至少一个空格或制表符。

要让列表看起来更漂亮，你可以把内容用固定的缩进整理好（显示效果与代码一致）：

- 轻轻的我走了，正如我轻轻的来；我轻轻的招手，作别西天的云彩。

那河畔的金柳，是夕阳中的新娘；波光里的艳影，在我的心头荡漾。

软泥上的青荇，油油的在水底招摇；在康河的柔波里，我甘心做一条水草！

- 那榆荫下的一潭，不是清泉，是天上虹；揉碎在浮藻间，沉淀着彩虹似的梦。

寻梦？撑一支长篙，向青草更青处漫溯；满载一船星辉，在星辉斑斓里放歌。

但我不能放歌，悄悄是别离的笙箫；夏虫也为我沉默，沉默是今晚的康桥！



悄悄的我走了， 正如我悄悄的来； 我挥一挥衣袖， 不带走一片云彩。

但是如果你懒，那也行：

代码：

1. \* 轻轻的我走了， 正如我轻轻的来； 我轻轻的招手，  
作别西天的云彩。

2. 那河畔的金柳， 是夕阳中的新娘； 波光里的艳影， 在我的  
心头荡漾。

3. 软泥上的青荇， 油油的在水底招摇； 在康河的柔波里， 我  
甘心做一条水草！

4. \* 那榆荫下的一潭， 不是清泉， 是天上虹； 揉碎在浮  
藻间， 沉淀着彩虹似的梦。

5. 寻梦？撑一支长篙， 向青草更青处漫溯； 满载一船星辉，  
在星辉斑斓里放歌。

6. 但我不能放歌， 悄悄是别离的笙箫； 夏虫也为我沉默， 沉  
默是今晚的康桥！

7. 悄悄的我走了， 正如我悄悄的来； 我挥一挥衣袖， 不带走  
一片云彩。

显示效果：

- 轻轻的我走了， 正如我轻轻的来； 我轻轻的招手， 作别西  
天的云彩。

那河畔的金柳， 是夕阳中的新娘； 波光里的艳影， 在我的心头荡  
漾。

软泥上的青荇， 油油的在水底招摇； 在康河的柔波里， 我甘心做一  
条水草！

- 那榆荫下的一潭，不是清泉，是天上虹；揉碎在浮藻间，沉淀着彩虹似的梦。

寻梦？撑一支长篙，向青草更青处漫溯；满载一船星辉，在星辉斑斓里放歌。

但我不能放歌，悄悄是别离的笙箫；夏虫也为我沉默，沉默是今晚的康桥！

悄悄的我走了，正如我悄悄的来；我挥一挥衣袖，不带走一片云彩。

#### 5.4. 包含段落的列表

语法说明：

列表项目可以包含多个段落，每个项目下的段落都必须缩进 4 个空格或是 1 个制表符（显示效果与代码一致）：

- 轻轻的我走了，正如我轻轻的来；我轻轻的招手，作别西天的云彩。

那河畔的金柳，是夕阳中的新娘；波光里的艳影，在我的心头荡漾。

软泥上的青荇，油油的在水底招摇；在康河的柔波里，我甘心做一条水草！

那榆荫下的一潭，不是清泉，是天上虹；揉碎在浮藻间，沉淀着彩虹似的梦。

寻梦？撑一支长篙，向青草更青处漫溯；满载一船星辉，在星辉斑斓里放歌。

但我不能放歌，悄悄是别离的笙箫；夏虫也为我沉默，沉默是今晚的康桥！

- 悄悄的我走了，正如我悄悄的来；我挥一挥衣袖，不带走一片云彩。

如果你每行都有缩进，看起来会看好很多，当然，再次地，如果你很懒惰，Markdown 也允许：

代码：

1. \*      轻轻的我走了， 正如我轻轻的来； 我轻轻的招手，  
作别西天的云彩。
2.    那河畔的金柳， 是夕阳中的新娘； 波光里的艳影， 在我的  
心头荡漾。
3.    软泥上的青荇， 油油的在水底招摇； 在康河的柔波里， 我  
甘心做一条水草！
4.            那榆荫下的一潭， 不是清泉， 是天上虹； 揉碎在浮  
藻间， 沉淀着彩虹似的梦。
5.    寻梦？撑一支长篙， 向青草更青处漫溯； 满载一船星辉，  
在星辉斑斓里放歌。
6.    但我不能放歌， 悄悄是别离的笙箫； 夏虫也为我沉默， 沉  
默是今晚的康桥！
7. \*      悄悄的我走了， 正如我悄悄的来； 我挥一挥衣袖，  
不带走一片云彩。

显示效果：

- 轻轻的我走了， 正如我轻轻的来； 我轻轻的招手， 作别西  
天的云彩。

那河畔的金柳， 是夕阳中的新娘； 波光里的艳影， 在我的心头荡  
漾。

软泥上的青荇， 油油的在水底招摇； 在康河的柔波里， 我甘心做一  
条水草！

那榆荫下的一潭， 不是清泉， 是天上虹； 揉碎在浮藻间， 沉淀着彩  
虹似的梦。

寻梦？撑一支长篙， 向青草更青处漫溯； 满载一船星辉， 在星辉斑斓里放歌。

但我不能放歌， 悄悄是别离的笙箫； 夏虫也为我沉默， 沉默是今晚的康桥！

- 悄悄的我走了， 正如我悄悄的来； 我挥一挥衣袖， 不带走一片云彩。

## 5.5. 包含引用的列表

语法说明：

如果要在列表项目内放进引用，那 > 就需要缩进：

代码：

1. \* 阅读的方法：
2. > 打开书本。
3. > 打开电灯。

显示效果：

- 阅读的方法：

打开书本。

打开电灯。

## 5.6. 包含代码区块的引用

语法说明：

如果要放代码区块的话，该区块就需要缩进两次，也就是 8 个空格或是 2 个制表符：

- 一列表项包含一个列表区块：

a. <代码写在这>

## 5.7. 一个特殊情况

在特殊情况下，项目列表很可能会不小心产生，像是下面这样的写法：

```
1. 1986. What a great season.
```

会显示成：

1. What a great season.

换句话说，也就是在行首出现数字-句点-空白，要避免这样的状况，你可以在句点前面加上反斜杠：

```
1. 1986\. What a great season.
```

会显示成：

1986. What a great season.

---

## 6. 引用

语法说明：

引用需要在被引用的文本前加上>符号。

代码：

- ```
1. > 这是一个有两段文字的引用，
2. > 无意义的占行文字1.
3. > 无意义的占行文字2.
4. >
5. > 无意义的占行文字3.
6. > 无意义的占行文字4.
```

显示效果：

这是一个有两段文字的引用，  
无意义的占行文字1.  
无意义的占行文字2.

无意义的占行文字3.

无意义的占行文字4.

Markdown 也允许你偷懒只在整个段落的第一行最前面加上 > :

代码:

1. > 这是一个有两段文字的引用,
2. 无意义的占行文字1.
3. 无意义的占行文字2.
4. > 无意义的占行文字3.
5. 无意义的占行文字4.

显示效果:

这是一个有两段文字的引用,

无意义的占行文字1.

无意义的占行文字2.

无意义的占行文字3.

无意义的占行文字4.

### 6.1. 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 > :

代码:

1. >>> 请问 `Markdwon` 怎么用? - 小白
2. >> 自己看教程! - 愤青
3. > 教程在哪? - 小白

显示效果:

    请问 `Markdwon` 怎么用? - 小白

    自己看教程! - 愤青

  教程在哪? - 小白

## 6.2. 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：

代码：

```
1. > 1.    这是第一行列表项。
2. > 2.    这是第二行列表项。
3. >
4. > 给出一些例子代码：
5. >
6. >      return shell_exec("echo $input |
    $markdown_script");
```

显示效果：

1. 这是第一行列表项。
2. 这是第二行列表项。

给出一些例子代码：

```
1. return shell_exec("echo $input |
    $markdown_script");
```

---

## 7. 插入图像

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。

### 7.1. 行内式

语法说明：![图片Alt](图片地址 “图片Title”)

代码：

1. 美丽花儿：

2. ![美丽花儿]

(`http://ww2.sinaimg.cn/large/56d258bdjw1eugeubg8  
ujj21kw16odn6.jpg "美丽花儿"`)

显示效果：

美丽花儿：



美丽花儿

## 7.2. 参考式

语法说明：

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”

代码：



1. 美丽花儿:

2. ![美丽花儿][flower]

3.

[flower]:<http://ww2.sinaimg.cn/large/56d258bdjw1eugeubg8ujj21kw16odn6.jpg> "美丽花儿"

显示效果:

美丽花儿:



美丽花儿

---

## 8. 内容目录

在段落中填写 `[TOC]` 以显示全文内容的目录结构。

效果参见最上方的目录

---

## 9. 注脚

语法说明:

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。

注意:经测试注脚与注脚之间必须空一行,不然会失效。成功后会发现,即使你没有把注脚写在文末,经Markdown转换后,也会自动归类到文章的最后。

代码:

1. 使用 `Markdown[^1]` 可以效率的书写文档,直接转换成 `HTML[^2]`, 你可以使用 `Leanote[^Le]` 编辑器进行书写。
2. `[^1]:Markdown` 是一种纯文本标记语言
3. `[^2]:HyperText Markup Language` 超文本标记语言
4. `[^Le]:` 开源笔记平台,支持Markdown和笔记直接发为博文

显示效果:

使用 `Markdown1` 可以效率的书写文档,直接转换成 `HTML2`, 你可以使用 `Leanote3` 编辑器进行书写。

注:脚注自动被搬运到最后面,请到文章末尾查看,并且脚注后方的链接可以直接跳转回到加注的地方。

---

## 10. LaTeX 公式

### 10.1. \$ 表示行内公式:

代码:

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。

显示效果:

质能守恒方程可以用一个很简洁的方程式 `E=mc2` 来表达。

### 10.2 \$\$ 表示整行公式:

代码:

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

显示效果:

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

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

$$\sum_{k=0}^{j-1} \gamma_{kj} z_k$$

访问 [MathJax](#) 参考更多使用方法。

## 11. 流程图

代码:

1. flow
2. st=>start: Start:><https://www.zybuluo.com>
3. io=>input/output: verification
4. op=>operation: Your Operation
5. cond=>condition: Yes or No?
6. sub=>subroutine: Your Subroutine
7. e=>end
8. st->io->op->cond
9. cond(yes)->e
10. cond(no)->sub->io

显示效果:

更多语法参考: [流程图语法参考](#)

## 12 .表格

语法说明：

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。

代码：

简单方式写表格：

1. 学号|姓名|分数
2. -|-|-
3. 小明|男|75
4. 小红|女|79
5. 小陆|男|92

原生方式写表格：

1. |学号|姓名|分数|
2. |-|-|-|
3. |小明|男|75|
4. |小红|女|79|
5. |小陆|男|92|

为表格第二列指定方向：

1. 产品|价格
2. -|-:
3. Leanote 高级账号|60元/年

## 4. Leanote 超级账号 | 120元/年

显示效果：

简单方式写表格：

| 学号 | 姓名 | 分数 |
|----|----|----|
| 小明 | 男  | 75 |
| 小红 | 女  | 79 |
| 小陆 | 男  | 92 |

原生方式写表格：

| 学号 | 姓名 | 分数 |
|----|----|----|
| 小明 | 男  | 75 |
| 小红 | 女  | 79 |
| 小陆 | 男  | 92 |

为表格第二列指定方向：

| 产品              | 价格     |
|-----------------|--------|
| Leanote<br>高级账号 | 60元/年  |
| Leanote<br>超级账号 | 120元/年 |

---

## 13. 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：

代码：

1. \* \* \*

2. \*\*\*

3. \*\*\*\*\*

4. - - -

显示效果都一样：

---

## 14. 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。

语法说明：

1. 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。
2. 插入多行代码，可以使用缩进或者“`code`”，具体看示例。

注意： 缩进式插入前方必须有空行

### 8.1. 行内式

代码：

1. C语言里的函数 `scanf()` 怎么使用？

显示效果：

C语言里的函数 `scanf()` 怎么使用？

### 8.2. 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）。

代码：

```
1.      #include <stdio.h>
2.      int main(void)
3.      {
4.          printf("Hello world\n");
```

```
5.      }
```

显示效果:

```
1. #include <stdio.h>
2. int main(void)
3. {
4.     printf("Hello world\n");
5. }
```

### 8.3. 用六个`包裹多行代码

代码:

```
1. ```
2. #include <stdio.h>
3. int main(void)
4. {
5.     printf("Hello world\n");
6. }
7. ```
```

显示效果:

```
1. #include <stdio.h>
2. int main(void)
3. {
4.     printf("Hello world\n");
5. }
```

### 8.4. 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：  
代码：

第一个例子：

```
1. <div class="footer">
2.     © 2004 Foo Corporation
3. </div>
```

第二个例子：

```
1. <table>
2.     <tr>
3.         <th rowspan="2">值班人员</th>
4.         <th>星期一</th>
```

## 5. 0. 目录

- 1. 斜体和粗体
- 2. 分级标题
- 3. 超链接
  - 3.1. 行内式
  - 3.2. 参考式
  - 3.3. 自动链接
- 4. 锚点
- 5. 列表
  - 5.1. 无序列表
  - 5.2. 有序列表



- [5.3. 定义型列表](#)
- [5.3. 列表缩进](#)
- [5.4. 包含段落的列表](#)
- [5.5. 包含引用的列表](#)
- [5.6. 包含代码区块的引用](#)
- [5.7. 一个特殊情况](#)
- [6. 引用](#)
  - [6.1. 引用的多层嵌套](#)
  - [6.2. 引用其它要素](#)
- [7. 插入图像](#)
  - [7.1. 行内式](#)
  - [7.2. 参考式](#)
- [8. 内容目录](#)
- [9. 注脚](#)
- [10. LaTeX 公式](#)
  - [10.1. \\$ 表示行内公式:](#)
  - [10.2 \\$\\$ 表示整行公式:](#)
- [11. 流程图](#)
- [12. 表格](#)
- [13. 分隔线](#)

- [14. 代码](#)

- [8.1. 行内式](#)
- [8.2. 缩进式多行代码](#)
- [8.3. 用六个`包裹多行代码](#)
- [8.4. HTML 原始码](#)

---

## 1. 斜体和粗体

代码：

1. \*斜体\*或\_斜体\_
2. \*\*粗体\*\*
3. \*\*\*加粗斜体\*\*\*
4. ~~删除线~~

显示效果：

- 这是一段斜体
- **这是一段粗体**
- 这是一段加粗斜体
- 这是一段删除线

---

## 2. 分级标题

第一种写法：

1. 这是一个一级标题
2. =====
3. 这是一个二级标题
4. -----  
-----

第二种写法:

1. # 一级标题
2. ## 二级标题
3. ### 三级标题
4. #### 四级标题
5. ##### 五级标题
6. ##### 六级标题

由于用了[TOC]标记编辑器会把所有标题写到目录大纲中，在这里写的演示标题也会列进去，所以就不演示了。同学们自己在编辑器中观察，很简单，一级标题字号最大，依级递减。

---

### 3. 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。

#### 3.1. 行内式

语法说明:

- []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。

代码:

1. 欢迎来到[梵居闹市]  
(<http://blog.leanote.com/freewalk>)
2. 欢迎来到[梵居闹市]  
(<http://blog.leanote.com/freewalk> "梵居闹市")

显示效果:

欢迎来到[梵居闹市](#)

欢迎来到[梵居闹市](#)

### 3.2. 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。

语法说明：

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。*

如果链接文字本身可以做为链接标记，你也可以写成[链接文字][  
[链接文字]: 链接地址的形式，见代码的最后一行。

代码：

1. 我经常去的几个网站[Google][1]、[Leanote][2]以及[自己的博客][3]
2. [Leanote 笔记][2]是一个不错的[网站][ ]。
3. [1]:<http://www.google.com> "Google"
4. [2]:<http://www.leanote.com> "Leanote"
5. [3]:<http://http://blog.leanote.com/freewalk>  
"梵居闹市"
6. [网  
站]:<http://http://blog.leanote.com/freewalk>

显示效果：

我经常去的几个网站[Google](#)、[Leanote](#)以及[自己的博客](#)  
[Leanote 笔记](#)是一个不错的[网站](#)。

### 3.3. 自动链接

语法说明：

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：

代码：

1. `<http://example.com/>`
2. `<address@example.com>`

显示效果：

<http://example.com/>  
[address@example.com](mailto:address@example.com)

---

## 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。

注意：

1. Markdown Extra 只支持在标题后插入锚点，其它地方无效。
2. Leanote 编辑器右侧显示效果区域暂时不支持锚点跳转，所以点来点去发现没有跳转不必惊慌，但是你发布成笔记或博文后是支持跳转的。

语法描述：

在你准备跳转到的指定标题后插入锚点{#标记}，然后在文档的其它地方写上连接到锚点的链接。

代码：

1. `## 0. 目录{#index}`
2. `跳转到[目录](#index)`

显示效果：

## 5. 列表

### 5.1. 无序列表

使用 \*, +, - 表示无序列表。

代码：

- 1. - 无序列表项 一
- 2. - 无序列表项 二
- 3. - 无序列表项 三

显示效果：

- 无序列表项 一
- 无序列表项 二
- 无序列表项 三

### 5.2. 有序列表

有序列表则使用数字接着一个英文句点。

代码：

- 1. 1. 有序列表项 一
- 2. 2. 有序列表项 二
- 3. 3. 有序列表项 三

显示效果：

- 1. 有序列表项 一
- 2. 有序列表项 二
- 3. 有序列表项 三

### 5.3. 定义型列表

语法说明：

定义型列表由名词和解释组成。一行写上定义，紧跟一行写上解释。解释的写法：紧跟一个缩进(Tab)

代码：

1. Markdown

2. : 轻量级文本标记语言，可以转换成html，pdf等格式  
(左侧有一个可见的冒号和四个不可见的空格)

3. 代码块 2

4. : 这是代码块的定义 (左侧有一个可见的冒号和四个不可见的空格)

5. 代码块 (左侧有八个不可见的空格)

显示效果：

Markdown

轻量级文本标记语言，可以转换成html，pdf等格式

代码块 2

这是代码块的定义

1. 代码块 (左侧有八个不可见的空格)

### 5.3. 列表缩进

语法说明：

列表项目标记通常是放在最左边，但是其实也可以缩进，最多 3 个空格，项目标记后面则一定要接着至少一个空格或制表符。

要让列表看起来更漂亮，你可以把内容用固定的缩进整理好（显示效果与代码一致）：

- 轻轻的我走了，正如我轻轻的来；我轻轻的招手，作别西天的云彩。

那河畔的金柳， 是夕阳中的新娘； 波光里的艳影， 在我的心头荡漾。

软泥上的青荇， 油油的在水底招摇； 在康河的柔波里， 我甘心做一条水草！

- 那榆荫下的一潭， 不是清泉， 是天上虹； 揉碎在浮藻间， 沉淀着彩虹似的梦。

寻梦？撑一支长篙， 向青草更青处漫溯； 满载一船星辉， 在星辉斑斓里放歌。

但我不能放歌， 悄悄是别离的笙箫； 夏虫也为我沉默， 沉默是今晚的康桥！

悄悄的我走了， 正如我悄悄的来； 我挥一挥衣袖， 不带走一片云彩。

但是如果你懒，那也行：

代码：

1. \* 轻轻的我走了， 正如我轻轻的来； 我轻轻的招手， 作别西天的云彩。

2. 那河畔的金柳， 是夕阳中的新娘； 波光里的艳影， 在我的心头荡漾。

3. 软泥上的青荇， 油油的在水底招摇； 在康河的柔波里， 我甘心做一条水草！

4. \* 那榆荫下的一潭， 不是清泉， 是天上虹； 揉碎在浮藻间， 沉淀着彩虹似的梦。

5. 寻梦？撑一支长篙， 向青草更青处漫溯； 满载一船星辉， 在星辉斑斓里放歌。

6. 但我不能放歌， 悄悄是别离的笙箫； 夏虫也为我沉默， 沉默是今晚的康桥！



7. 悄悄的我走了， 正如我悄悄的来； 我挥一挥衣袖， 不带走一片云彩。

显示效果：

- 轻轻的我走了， 正如我轻轻的来； 我轻轻的招手， 作别西天的云彩。

那河畔的金柳， 是夕阳中的新娘； 波光里的艳影， 在我的心头荡漾。

软泥上的青荇， 油油的在水底招摇； 在康河的柔波里， 我甘心做一条水草！

- 那榆荫下的一潭， 不是清泉， 是天上虹； 揉碎在浮藻间， 沉淀着彩虹似的梦。

寻梦？撑一支长篙， 向青草更青处漫溯； 满载一船星辉， 在星辉斑斓里放歌。

但我不能放歌， 悄悄是别离的笙箫； 夏虫也为我沉默， 沉默是今晚的康桥！

悄悄的我走了， 正如我悄悄的来； 我挥一挥衣袖， 不带走一片云彩。

## 5.4. 包含段落的列表

语法说明：

列表项目可以包含多个段落，每个项目下的段落都必须缩进 4 个空格或是 1 个制表符（显示效果与代码一致）：

- 轻轻的我走了， 正如我轻轻的来； 我轻轻的招手， 作别西天的云彩。

那河畔的金柳， 是夕阳中的新娘； 波光里的艳影， 在我的心头荡漾。

软泥上的青荇，油油的在水底招摇；在康河的柔波里，我甘心做一条水草！

那榆荫下的一潭，不是清泉，是天上虹；揉碎在浮藻间，沉淀着彩虹似的梦。

寻梦？撑一支长篙，向青草更青处漫溯；满载一船星辉，在星辉斑斓里放歌。

但我不能放歌，悄悄是别离的笙箫；夏虫也为我沉默，沉默是今晚的康桥！

- 悄悄的我走了，正如我悄悄的来；我挥一挥衣袖，不带走一片云彩。

如果你每行都有缩进，看起来会看好很多，当然，再次地，如果你很懒惰，Markdown 也允许：

代码：

1. \* 轻轻的我走了，正如我轻轻的来；我轻轻的招手，作别西天的云彩。
2. 那河畔的金柳，是夕阳中的新娘；波光里的艳影，在我的心头荡漾。
3. 软泥上的青荇，油油的在水底招摇；在康河的柔波里，我甘心做一条水草！
4. 那榆荫下的一潭，不是清泉，是天上虹；揉碎在浮藻间，沉淀着彩虹似的梦。
5. 寻梦？撑一支长篙，向青草更青处漫溯；满载一船星辉，在星辉斑斓里放歌。
6. 但我不能放歌，悄悄是别离的笙箫；夏虫也为我沉默，沉默是今晚的康桥！
7. \* 悄悄的我走了，正如我悄悄的来；我挥一挥衣袖，不带走一片云彩。

显示效果：

- 轻轻的我走了，正如我轻轻的来；我轻轻的招手，作别西天的云彩。

那河畔的金柳，是夕阳中的新娘；波光里的艳影，在我的心头荡漾。

软泥上的青荇，油油的在水底招摇；在康河的柔波里，我甘心做一条水草！

那榆荫下的一潭，不是清泉，是天上虹；揉碎在浮藻间，沉淀着彩虹似的梦。

寻梦？撑一支长篙，向青草更青处漫溯；满载一船星辉，在星辉斑斓里放歌。

但我不能放歌，悄悄是别离的笙箫；夏虫也为我沉默，沉默是今晚的康桥！

- 悄悄的我走了，正如我悄悄的来；我挥一挥衣袖，不带走一片云彩。

## 5.5. 包含引用的列表

语法说明：

如果要在列表项目内放进引用，那 > 就需要缩进：

代码：

1. \* 阅读的方法：
2. > 打开书本。
3. > 打开电灯。

显示效果：

- 阅读的方法：

打开书本。

打开电灯。

## 5.6. 包含代码区块的引用

语法说明：

如果要放代码区块的话，该区块就需要缩进两次，也就是 8 个空格或是 2 个制表符：

- 一列表项包含一个列表区块：

a. <代码写在这>

## 5.7. 一个特殊情况

在特殊情况下，项目列表很可能会不小心产生，像是下面这样的写法：

```
1. 1986. What a great season.
```

会显示成：

1. What a great season.

换句话说，也就是在行首出现数字-句点-空白，要避免这样的状况，你可以在句点前面加上反斜杠：

```
1. 1986\. What a great season.
```

会显示成：

1986. What a great season.

---

## 6. 引用

语法说明：

引用需要在被引用的文本前加上>符号。

代码：

1. > 这是一个有两段文字的引用,
2. > 无意义的占行文字1.
3. > 无意义的占行文字2.
4. >
5. > 无意义的占行文字3.
6. > 无意义的占行文字4.

显示效果:

这是一个有两段文字的引用,  
无意义的占行文字1.  
无意义的占行文字2.  
无意义的占行文字3.  
无意义的占行文字4.

Markdown 也允许你偷懒只在整个段落的第一行最前面加上 > :

代码:

1. > 这是一个有两段文字的引用,
2. 无意义的占行文字1.
3. 无意义的占行文字2.
4. > 无意义的占行文字3.
5. 无意义的占行文字4.

显示效果:

这是一个有两段文字的引用,  
无意义的占行文字1.  
无意义的占行文字2.  
无意义的占行文字3.  
无意义的占行文字4.

## 6.1. 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 >：

代码：

- ```
1. >>> 请问 Markdwon 怎么用? - 小白
2. >>  自己看教程! - 愤青
3. >   教程在哪? - 小白
```

显示效果：

```
    请问 Markdwon 怎么用? - 小白
  自己看教程! - 愤青
教程在哪? - 小白
```

## 6.2. 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：

代码：

- ```
1. > 1.    这是第一行列表项。
2. > 2.    这是第二行列表项。
3. >
4. > 给出一些例子代码：
5. >
6. >      return shell_exec("echo $input |
    $markdown_script");
```

显示效果：

- ```
1. 这是第一行列表项。
2. 这是第二行列表项。
```

```
给出一些例子代码：
```

```
1. return shell_exec("echo $input |  
$markdown_script");
```

---

## 7. 插入图像

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。

### 7.1. 行内式

语法说明：![图片Alt](图片地址 “图片Title”)

代码：

1. 美丽花儿：

2. ![美丽花儿]

(<http://ww2.sinaimg.cn/large/56d258bdjw1eugeubg8ujj21kw16odn6.jpg> "美丽花儿")

显示效果：

美丽花儿：



美丽花儿

## 7.2. 参考式

语法说明：

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”

代码：

1. 美丽花儿：

2. ！[美丽花儿][flower]

3.

[flower]:<http://ww2.sinaimg.cn/large/56d258bdjw1eugeubg8ujj21kw16odn6.jpg> "美丽花儿"

显示效果：

美丽花儿：





美丽花儿

---

## 8. 内容目录

在段落中填写 `[TOC]` 以显示全文内容的目录结构。

效果参见最上方的目录

---

## 9. 注脚

语法说明：

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。

注意：经测试注脚与注脚之间必须空一行，不然会失效。成功后会发现，即使你没有把注脚写在文末，经Markdown转换后，也会自动归类到文章的最后。

代码：

1. 使用 `Markdown1` 可以效率的书写文档，直接转换成 `HTML2`，你可以使用 `Leanote3` 编辑器进行书写。
2. `1`:`Markdown` 是一种纯文本标记语言
3. `2`:`HyperText Markup Language` 超文本标记语言
4. `3`:`开源笔记平台`，支持`Markdown`和笔记直接发为博文

显示效果：

使用 `Markdown1` 可以效率的书写文档，直接转换成 `HTML2`，你可以使用 `Leanote3` 编辑器进行书写。

注：脚注自动被搬运到最后面，请到文章末尾查看，并且脚注后方的链接可以直接跳转回到加注的地方。

---

## 10. LaTeX 公式

### 10.1. `$` 表示行内公式：

代码：

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。

显示效果：

质能守恒方程可以用一个很简洁的方程式 `E=mc2` 来表达。

### 10.2 `$$` 表示整行公式：

代码：

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

显示效果：

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

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

$$\sum_{k=0}^{j-1} \gamma^k z^k$$

访问 [MathJax](#) 参考更多使用方法。

---

## 11. 流程图

代码：

```

1. flow
2. st=>start: Start:>https://www.zybuluo.com
3. io=>inputoutput: verification
4. op=>operation: Your Operation
5. cond=>condition: Yes or No?
6. sub=>subroutine: Your Subroutine
7. e=>end
8. st->io->op->cond
9. cond(yes)->e
10. cond(no)->sub->io

```

显示效果：

---

更多语法参考：[流程图语法参考](#)

---

## 12. 表格

语法说明：

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。

代码：

简单方式写表格：

```
1. 学号|姓名|分数
2. -|-|-
3. 小明|男|75
4. 小红|女|79
5. 小陆|男|92
```

原生方式写表格：

```
1. |学号|姓名|分数|
2. |-|-|-|
3. |小明|男|75|
4. |小红|女|79|
5. |小陆|男|92|
```

为表格第二列指定方向：

```
1. 产品|价格
2. -|-:
3. Leanote 高级账号|60元/年
4. Leanote 超级账号|120元/年
```

显示效果：

简单方式写表格：

学号	姓名	分数
小明	男	75
小红	女	79

小陆	男	92
----	---	----

原生方式写表格：

学号	姓名	分数
小明	男	75
小红	女	79
小陆	男	92

为表格第二列指定方向：

产品	价格
Leanote 高级账号	60元/年
Leanote 超级账号	120元/年

## 13. 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：

代码：

1. \* \* \*
2. \*\*\*
3. \*\*\*\*\*
4. - - -
5. -----

显示效果都一样：

## 14. 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。

语法说明：

1. 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。
2. 插入多行代码，可以使用缩进或者 ````code````, 具体看示例。

注意： 缩进式插入前方必须有空行

## 8.1. 行内式

代码：

1. C语言里的函数 `scanf()` 怎么使用？

显示效果：

C语言里的函数 `scanf()` 怎么使用？

## 8.2. 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）。

代码：

```
1.      #include <stdio.h>
2.      int main(void)
3.      {
4.          printf("Hello world\n");
5.      }
```

显示效果：

```
1. #include <stdio.h>
2. int main(void)
3. {
4.     printf("Hello world\n");
```

```
5. }
```

### 8.3. 用六个`包裹多行代码

代码：

```
1. ```
2. #include <stdio.h>
3. int main(void)
4. {
5.     printf("Hello world\n");
6. }
7. ```
```

显示效果：

```
1. #include <stdio.h>
2. int main(void)
3. {
4.     printf("Hello world\n");
5. }
```

### 8.4. 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：

代码：

第一个例子：

```
1. <div class="footer">
2.     © 2004 Foo Corporation
3. </div>
```

第二个例子:

```
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>
```

显示效果:

第一个例子:

© 2004 Foo Corporation

第二个例子:

值班人员	星期一	星期二	星期三
	李强	张明	王平

```
<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>
```

显示效果：

第一个例子：

© 2004 Foo Corporation

第二个例子：

值班人员	星期一	星期二	星期三
	李强	张明	王平