

Markdown-

learn to use Markdown

markdown 编写的文档可以导出为HTML、Word、图像、PDF、Epub等格式

目前很多网站使用Markdown来撰写帮助文档或用于论坛上发消息。例如：Github、简书、reddit、Diaspora、Stack Exchange、OpenStreetMap、SourceForge等。

使用Typora编辑器

标题

使用=和-标记一级和二级标题

使用#标记1-6级标题 #后面要加空格

段落

换行是两个以上空格加上回车 也可以在段落后加一个空行来开始一个新的段落

字体

斜体 **粗体**

前后加或

一个 斜体

两个* 粗体

三个* 粗斜体

分隔线 一行中用三个以上的星号、减号、底线来建立分隔线，行内不能有其他东西。

删除线 在文字两端各加上两个波浪线

下划线 前加尖括号里u 后加尖括号里/u

脚注 方括号 ^ 要注明的文本 方括号

列表

无序列表 用星号 加好 减号 后加一个空格 来标记

有序列表 使用数字加上 点 来表示

列表嵌套 在子列表前添加四个空格即可

区块

在段落开头使用 大于 号，后加一个 空格

区块可以嵌套 一个大于号是最外层 两个大于号是第一层嵌套 以此类推

区块中使用列表没有限制，列表中使用区块，需要在 大于号 前添加四个空格的缩进。

代码

用反引号把代码包起来（不是引号，是像顿号的那个）

代码区块使用4个空格或者一个制表符（Tab键）（好像没用）

可以用三个反引号包括一段代母，并指定一种语言

链接

左方括号 链接名称 右方括号 左圆括号 链接地址 右圆括号 （链接地址www啥的不显示）
或者 直接 左尖括号 链接地址 右尖括号 （直接显示www啥的）
用变量 设置链接 便于 统一修改 格式：
左方括号 Google 右方括号 左方括号 1 右方括号
文末： 左方括号 1 右方括号： <http://www.google.com/>

图片

叹号 左方括号 文字 右方括号 左圆括号 图片地址 “可选标题” 右圆括号
也可对图片网址使用变量 并在文末赋值
Markdown无法指定图片的高度与宽度，需要的话可以使用普通的 左尖括号 img 右尖括号 标签
左尖括号 img src="http://..." width="50%" 右尖括号



表格

用|来分隔不同单元格 用-来分隔表头和其他行
-: 右对齐 : -左对齐 -: 居中对齐

表头	表头
----	----

高级技巧

不在Markdown涵盖范围之内的标签，都可以直接在文档里用HTML撰写。
比如 （以下省略尖括号） kbd b i em sup sub br等
转义
用反斜杠转义特殊字符
公式
使用两个美元符号包裹TeX或LaTeX格式的数学公式，提交后，问答和文章页会根据需要加载Mathjax对数学公式进行渲染。

typora 画流程图、时序图（顺序图）、甘特图
用flow（或mermaid）语言画流程图

哲学

可读性高于一切，Markdown文档应该能直接以纯文本方式发布，而不致一眼看过去满眼都是标签和格式化指令

Markdown语法受几种text转HTML过滤器影响：[Setext][3],[atx][4],[Textile][5],[reStructuredText][6],[Grutatext][7],[EtText][8]，其中对Markdown语法影响最大的单一来源是纯文本的Email格式。

为实现这一哲学，Markdown语法几乎全部由标点符号构成且尽量做到望文生义，如*本来看上去就像“强调”

内嵌HTML

Markdown语法为“方便地在网上写作”这一目标而生。

Markdown语法只相当于HTML标签的一个非常非常小的子集。对Markdown语法无法支持的格式，可以直接用HTML，无需定界符来告诉Markdown要写HTML了，你直接写就是了。唯一的限制是那些块级HTML元素，如<div>，<table>，<pre>，<p>等等，必须使用空行与相邻内容分开，并且块元素的开始和结束标签之前不要留有空格和TAB。Markdown足够聪明不会添加额外的（也是不必要的）<p>标签包住这些块元素标签。

HTML是一种适合发表的格式，而Markdown是一种书写格式。

注意：不要再块级HTML元素内使用Markdown格式化命令，Markdown不会处理它们。而在HTML行内元素内的Markdown语法标记会被正确处理

行内HTML标签，如，<cite>，，在一个Markdown段落里、列表中，或标题中，随使用。

可以用HTML标签代替Markdown格式化命令。如直接用HTML标签<a>或而不使用Markdown的链接和图片语法，随你的便。

自动转换特殊字符

在HTML中，有两个字符需要特殊对待：<和&。<用于标识HTML实体。如果打算把他们当成普通字符，必须使用反引号转义它们。在Markdown中，只需要关心那些必要的转义。如果用在HTML实体中使用&符号，它会保持不变，而其他场合，它会转换成&。

在Markdown代码行内标记和块级标记中，<和&始终会被自动编码。这使得在Markdown文件中书写HTML代码更容易。（相对于纯HTML，如果想在纯HTML里贴一段HTML代码，会非常复杂，代码中每一个<和&都必须转义。

块级元素

段落和换行

段落之间用一个或多个空行分隔（如果一行什么也没有或者只有空格和TAB都会被视为空行）正常的段落不要以空白或TAB字符开始。

Markdown支持“硬折行”。这一点与其他text转HTML程序完全不同（包括Moveable Type的“Convert Line Breaks”选项），他们会将段落中的每一个换行符转换成
标签。

如果你确实需要使用Markdown插入一个
换行符，只需要在每一行的末尾以两个或更多个空格符号结束，然后再打回车键。

在Markdown里生成一个
有点麻烦，但那种简单的“把每一个换行符都转换成
规则”并不适用于Markdown。Markdown Email风格的blockquoting和multi-paragraph [list items][l]更好用，更美观。

标题

Markdown支持两种风格的标题，[Setext][9]和[atx][10]

Setext风格的一级标题下面一行使用等号，二级标题下面使用连字符，至少有一个=和-就能正常工作。

Atx风格的标题在每行的开头使用1-6个井号字符，分别对应标题级别1-6。如果愿意，也可以“结束”atx-风格的标题。这纯粹是美观考虑，结束用的井号个数随便，不必与起始井号数量相同。

引用块

> 后加空格 来时实现引用块。

引用块中可包含其他Markdown元素，如标题、列表和代码块。

列表

Markdown支持有序列表和无序列表

无序列表可使用星号、加号和连字符作为列表标记：

有序列表使用数字加英文句点。

你在列表中输入的标记数字并不会反映到 Markdown输出的HTML之中。

列表标记通常从左边界开始，至多可以有三个空格的缩进。列表标记之后至少跟一个空格或TAB。

如果列表项之间用空行分隔，Markdown就会在HTML输出中使用<p>标签包裹列表项。

比如：

```
* Bird
* Magic
```

生成的HTML如下：

```
<ul>
<li>Bird</li>
<li>Magic</li>
</ul>
```

而这个：

```
* Brid

* Magic
```

生成的HTML是这样：

```
<ul>
<li><p>Bird</p></li>
<li><p>Magic</p></li>
</ul>
```

列表项有可能由多个段落组成，列表项的每个后续段落必须缩进至少4个空格或者一个TAB

要在列表中使用引用，引用定界符>需要缩进

要在列表中使用代码块，代码块需要缩进两次 -- 8个空格或者两个TAB

有时候会触发一个有序列表，如：

```
1986. what a great season.
```

以数字+句点+空格 序列起始的行会触发有序列表。为避免此情况，要对句点符号进行转义：

```
1986\. what a great season.
```

代码块

在写有关编程或标记语言源代码时用到预格式化的代码块。不像格式化普通段落，代码块中的行会按字面进行解释。Markdown对代码块同事使用

```
和<code>标签包裹。
```

在Markdown中要生成一个代码块，只需要在代码块内容的每一行缩进至少四个空格或一个TAB。Markdown会从生成的代码块中删除一级缩进--4个空格或者1个TAB。

代码块在遇到没有缩进的一行，或者文件末尾时自动结束。

在代码块中，&符号和<、>会自动转换成HTML实体。因此在Markdown中包含HTML源代码很简单：粘贴进去，缩进一下。

Markdown不会解析代码块中的Markdown标记。

水平线

在一行里只放三个或更多个连字符、星号或下划线，就会得到一个水平线标记（<hr />）。

行内元素

链接

Markdown支持两种风格的链接：行内链接和引用链接。两种风格的链接，链接文本都放在中括号之内[square brackets]。

要生成一个行内链接，在链接文本之后紧跟用一对小括号。小括号里放链接地址和可选的链接title。title要用引号包起来。例如：

```
只是一个 [an example](http://example.com/ "Title") 行内链接。
```

```
[这个链接](http://example.net/) 没有title属性。
```

markdown会生成：

```
<p>This is <a href="http://example.com/" title="Title">
an example</a> inline link.</p>

<p><a href="http://example.net/">This link</a> has no title attribute.</p>
```

如果引用本地资源或同一站点的资源，可以使用相对路径：

如果想进一步了解我，请参阅我的[关于我] (/about/) 页。

引用风格的链接，在链接文本后紧跟又一对中括号。这对中括号里放的是该链接的标识符（可以理解为别名）：

这是一个饮用型链接 [示例] [id]。

如果你嫌弃两对中括号过于亲密，Markdown允许你在两对中括号之间放一个空格：

这是一个饮用型链接 [示例] [id]。

然后，可以在文档的任意位置，像下面这样定义链接标识与链接的对应关系（一行一个链接）：

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

即：

- 中括号内放链接标识符（行前可选缩进，至多不超过三个空格）；
- 之后紧跟一个冒号；
- 再后面是一个或多个空格（TAB也行）；
- 接下来是链接URL；
- 最后是可选的用双引号或单引号或小括号括起来的链接title。

下面三种链接定义方式是等价的：

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

链接URL，还支持使用一对可选的尖括号包裹起来：

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

也可以将链接的title属性放在下一行并使用额外的空格或TAB填充，这样较长的URL会比较美观：

```
[id]:http://example.com/longish/path/to/resource/here    "Optional Title
Here"
```

链接定义仅供Markdown解析器使用。最终输出的HTML当中不会包含链接定义。

链接标识符可以由字母、数字、空格和标点符号组成 不区分大小写。下面这两个链接：

```
[link text][a]
[link text][A]
```

是等价的。

隐式链接标识允许我们省略链接标识，这时链接文本本身就是链接标识。在链接文本之后加一对空的中括号--例如，使用“Google”文本链接到google.com站点，可以这样写：

```
[Google][]
```

然后这样定义它的链接：

```
[Google]:http://google.com
```

链接名字可以包含空格。

链接定义可放于Markdown文档的任意位置。建议把它们就近放到最先使用它的段落之后。也可放到文档末尾，当成某种形式的尾注。

引用型链接的亮点并不在于它更容易书写，而在于让你的文档可读性更好。这样源文档更接近于最终的浏览器输出效果。Markdown允许将标记有关的元数据移到段落之外，你尽管添加链接，而不必担心打断文件的故事情节。

强调

Markdown使用星号（*）和下划线（_）标识强调。一个*或_包裹的文本会使用HTML标签包裹；使用两个*或_包裹的文本会使用HTML标签包裹。

使用空格包裹单独的*或_，它们就失去了强调的含义，而成为字面上的星号或下划线。

代码

要在行内标识代码，用反引号（`）包住它。

Use the `printf()` function

要在一个行内代码中使用反引号（`）本身，用多个反引号作为定界符包住它：

```
There is a literal backtick (`) here.
```

包住行内代码的反引号定界符可以包括空格，即在起始反引号之后，结束反引号之前可以有一个空格。这使得我们能够在行内代码的开始或结束处使用反引号：

A single backtick in a code span: `

A backtick-delimited string in a code span: `foo`

在行内代码中，&和<和>会自动编码为HTML实体，以方便包含HTML标签。

Please don't use any `<blink>` tags.

`—` is the decimal-encoded equivalent of `—`.

图片

Markdown使用了类似链接语法来表示图片，同样有两种风格：行内图片和引用图片。

```
![Alt text](/path/to/img.jpg)
![Alt text](/path/to/img.jpg "Optional title")
```

即：

- 一个感叹号！开头；
- 其后紧跟一对中括号，中括号内存放图片alt属性；
- 其后紧跟一对小括号，内存放图片的URL或路径，及可选的双引号或单引号或小括号括起来的图片title

引用图片语法如下：

```
![Alt text][id]
```

```
[id]: url/to/image "Optional title attribute"
```

Markdown尚无语法指定图片大小，若想达到此功能，直接使用标签。

杂七杂八

自动链接

“自动”定义链接和Email地址：直接用一对尖括号把URL或Email地址包住。这表示链接文本就是URL本身，Email文本就是Email本身。这样就得到一个可点击的链接，如：

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

自动Email地址工作方式相似，只有一点不同，Markdown自动用一些十进制和十六进制数字表示你的Email，以防止遭遇垃圾邮件袭击。例如：

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

会被转换为：

```
<a href="&#x6D;&#x61;i&#x6C;&#x74;&#x6F;:&#x61;&#x64;&#x72;&#x65;&#x115;&#x115;&#x64;&#x101;&#x120;&#x61;&#x109;&#x70;&#x6C;e&#x2E;&#x99;&#x111;&#x109;">&#x61;&#x64;&#x64;&#x72;&#x65;&#x115;&#x115;&#x64;&#x101;&#x120;&#x61;&#x109;&#x70;&#x6C;e&#x2E;&#x99;&#x111;&#x109;</a>
```

这种实体编码的小方法可以骗过一些收集邮件地址的机器人，不过它无法骗过所有的机器人。有总比没有强，能阻止一点就阻止一点好了。

反斜线转义

使用反斜线转义那些Markdown元字符，让它们失去原有的“魔力”。

以下字符支持使用反斜线转义：

\ 反斜线
` 反引号
* 星号
_ 下划线
{ } 大括号
[] 中括号
() 小括号
井号
+ 加号
- 减号 (连字符)
. 句点
! 感叹号

以下非markdown语法，而是typora里能用的语法

O^3

`O^3^` 或 `O3`

O_3

`O~3~` 或 `O$_3$`

Typora数学公式汇总

公式块与行内公式的添加

- 公式块
 - 创建独立的一块公式区域

\$\$

`h_\theta(x) = \theta_0 + \theta_1 x`

\$\$

公式 ✓

$$h_{\theta}(x) = \theta_0 + \theta_1 x$$

方法一：左上角点击“段落”，再点击“公式块”

方法二：在文中输入\$\$，再按下回车

- 行内公式

方法一：在\$\$中间加入需要的公式

简便方法：先按\$，再按“Esc”



行内公式需要先设置下

常用公式代码

上下标，正负无穷

数学表达式	LaTeX代码
x^2	<code>x^2</code>
y_1	<code>y_1</code>
∞	<code>\infty</code>
$-\infty$	<code>-\infty</code>

加减乘，分式，根号，省略号

数学表达式	LaTeX代码
$a + b - c * d$	<code>a+b-c*d</code>
$a \div b$	<code>a\div{b}</code>
$a \pm b$	<code>a\pm{b}</code>
$\frac{a}{b}$	<code>\frac{a}{b}</code>
\sqrt{b}	<code>\sqrt{b}</code>
\dots	<code>\cdots</code>

三角函数

数学表达式	LaTeX代码
$\sin \theta$	<code>\sin{\theta}</code>
$\cos \theta$	<code>\cos{\theta}</code>
$\tan \theta$	<code>\tan{\theta}</code>
$\cot \theta$	<code>\cot{\theta}</code>

矢量，累加累乘，极限

数学表达式	LaTeX代码
\vec{F}	<code>\vec{F}</code>
$\sum_{i=1}^n a_i$	<code>\sum_{i=1}^n{a_i}</code>
$\prod_{i=1}^n a_i$	<code>\prod_{i=1}^n{a_i}</code>
$\lim_{a \rightarrow +\infty} a + b$	<code>\lim_{a\rightarrow+\infty}{a+b}</code>

希腊字母

数学表达式	LaTeX代码
α	<code>\alpha</code>
β	<code>\beta</code>
γ	<code>\gamma</code>
δ	<code>\delta</code>
ϵ	<code>\epsilon</code>
ε	<code>\varepsilon</code>
η	<code>\eta</code>
θ	<code>\theta</code>
κ	<code>\kappa</code>
ι	<code>\iota</code>
ζ	<code>\zeta</code>
λ	<code>\lambda</code>
μ	<code>\mu</code>
ϕ	<code>\phi</code>
π	<code>\pi</code>
ρ	<code>\rho</code>
ξ	<code>\xi</code>
ν	<code>\nu</code>
υ	<code>\upsilon</code>
φ	<code>\varphi</code>
χ	<code>\chi</code>
ψ	<code>\psi</code>
ω	<code>\omega</code>

关系运算符

数学表达式	LaTeX代码
\leq	<code>\leq</code>
\geq	<code>\geq</code>

矩阵

简单矩阵

使用 `\begin{matrix}...\end{matrix}` 生成，每行以`\\`结尾表示换行，元素间以`&`间隔，式子的表示序号 `\tag{1}`（右边的序号）。

例如：

$$\begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{matrix}$$

(1)

```
$$
\begin{matrix}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9
\end{matrix} \tag{1}
$$
```

带左右括号的矩阵

方法一：在 `\begin{}` 之前和 `\end{}` 之后添加左右括号的代码。

大括号：

$$\left\{ \begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{matrix} \right\}$$

(2)

```
$$
\left\{
\begin{matrix}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9
\end{matrix}
\right\} \tag{2}
$$
```

中括号：

$$\left[\begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{matrix} \right]$$

(3)

```

$$
\left[
\begin{matrix}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9
\end{matrix}
\right] \tag{3}
$$

```

小括号：

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \tag{4}$$

```

$$
\left(
\begin{matrix}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9
\end{matrix}
\right) \tag{4}
$$

```

注意：代码中大括号要加转义符`\left\{`，中括号不用`\left[`，小括号不用`\left(`

方法二：改变 `\begin{matrix}` 和 `\end{matrix}` 中的 `{matrix}`

大括号改为`Bmatrix`，中括号改为`bmatrix`

包含希腊字母与省略号

行省略号 `\cdots`，列省略号 `\vdots`，斜向省略号（左上至右下） `\ddots`。

$$\begin{pmatrix} 1 & 2 & \cdots & 5 \\ 6 & 7 & \cdots & 10 \\ \vdots & \vdots & \ddots & \vdots \\ \alpha & \alpha + 1 & \cdots & \alpha + 4 \end{pmatrix}$$

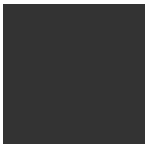
```

$$
\left\{
\begin{matrix}
1 & & 2 & & \cdots & & 5 & \\
6 & & 7 & & \cdots & & 10 & \\
\vdots & & \vdots & & \ddots & & \vdots & \\
\alpha & & \alpha+1 & & \cdots & & \alpha+4 & 
\end{matrix}
\right\}
$$

```

表格

简易表格



```
$$
\begin{array}{|c|c|c|}
    \hline 2&9&4\\
    \hline 7&5&3\\
    \hline 6&1&8\\
    \hline
\end{array}
$$
```

开头结尾: `\begin{array}`, `\end{array}`

定义式: 例: `{|c|c|c|}`, 其中 `c` `l` `r` 分别代表居中, 左对齐及右对齐。

分割线:

- 竖直分割线: 在定义式中插入 `|`, (`||` 表示两条竖直分割线)。
- 水平分割线: 在下一行输入前插入 `\hline`, 以下图真值表为例。
- 其他: 每行元素间均须要插入 `&`, 每行元素以 `\\` 结尾。

真值表:

<i>A</i>	<i>B</i>	<i>F</i>
0	0	0
0	1	1
1	0	1
1	1	1

```
$$
\begin{array}{cc|c}
    A&B&F\\
    \hline 0&0&0\\
    0&1&1\\
    1&0&1\\
    1&1&1
\end{array}
$$
```

行列式

多行等式对齐

$$\begin{aligned} a &= b + c \\ &= d + e + f \end{aligned}$$

```


$$\begin{aligned} a &= b+c \\ &= d+e+f \end{aligned}$$


```

$$\begin{cases} 3x + 5y + z \\ 7x - 2y + 4z \\ -6x + 3y + 2z \end{cases}$$

$$\begin{cases} 3x + 5y + z \\ 7x - 2y + 4z \\ -6x + 3y + 2z \end{cases}$$

$$f(n) = \begin{cases} n/2, & \text{if } n \text{ is even} \\ 3n + 1, & \text{if } n \text{ is odd} \end{cases}$$

```

$$
f(n)=
\begin{cases}
n/2,& \text{if } n \text{ is even} \\
3n+1,& \text{if } n \text{ is odd}
\end{cases}
$$

```

$$a \setminus !b + ab + a \setminus, b + a \setminus; b + a \setminus b + a \quad b + a \quad b + a \quad b$$

$$ab + ab + ab + ab + ab + a \quad b + a \quad b$$

- 紧贴 `\!`
- 无空格 小空格 `\,` 中空格 `\;` 大空格 `\` (`\`后加空格)
- 真空格 `\quad` 双真空格 `\qquad`

