

## Markdown语法说明（详解版）

推荐

17

收藏

**说明：**本文原载于[markdown.tw](http://markdown.tw)网站，本次转载时为方便阅读，将其改为简体中文，并修改个别用语。[@hi\\_jeanne](#)

**NOTE:** This is Simplified Chinese Edition Document of Markdown Syntax. If you are seeking for English Edition Document. Please refer to [Markdown: Syntax](#).

## Markdown: Syntax

- 概述
  - 哲学
  - 行内 HTML
  - 特殊字符自动转换
- 区块元素
  - 段落和换行
  - 标题
  - 区块引言
  - 列表
  - 程序代码区块
  - 分隔线
- 区段元素
  - 链接
  - 强调
  - 程序代码
  - 图片
- 其它
  - 转义字符
  - 自动链接
- 感谢

**注意：**这份文档是用 Markdown 写的，你可以看看附件中的原始文档。

## 概述

## 哲学

Markdown 的目标是实现「易读易写」。

不过最需要强调的便是它的可读性。一份使用 Markdown 格式撰写的文档应该可以直接以纯文本发布，并且看起来不会像是由许多标签或是格式指令所构成。Markdown 语法受到一些既有 text-to-HTML 格式的影响，包括 [Setext](#)、[atx](#)、[Textile](#)、[reStructuredText](#)、[Grutatext](#) 和 [EtText](#)，然而最大灵感来源其实是纯文本的电子邮件格式。

因此 Markdown 的语法全由标点符号所组成，并经过严谨慎选，是为了让它们看起来就像所要表达的意思。像是在文字两旁加上星号，看起来就像“强调”。Markdown 的列表看起来，嗯，就是列表。假如你用过电子邮件，区块引言看起来就真的像是引用一段文字。

## 行内 HTML

Markdown 的语法有个主要的目的：用来作为一种网络内容的写作语言。

搜索

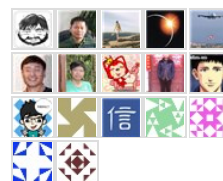
## 本文附件

- [markdown-syntax-zhs.txt](#)

## 本文标签

[markdown](#) × 37[文档](#) × 2

## 推荐会员



## 相关标签

Markdown 不是要来取代 HTML，甚至也没有要和它相似，它的语法种类不多，只和 HTML 的一部分有关系，重点不是要创造一种更容易写作 HTML 文档的语法，我认为 HTML 已经很容易写了，Markdown 的重点在于，它能让文档更容易阅读、编写。HTML 是一种发布的格式，Markdown 是一种编写的格式，因此，Markdown 的格式语法只涵盖纯文本可以涵盖的范围。

不在 Markdown 涵盖范围之内的标签，都可以直接在文档里面用 HTML 撰写。不需要额外标注这是 HTML 或是 Markdown；只要直接加标签就可以了。

只有区块元素——比如 `<div>`、`<table>`、`<pre>`、`<p>` 等标签，必需在前后加上空白，以利与内容区隔。而且这些（元素）的开始与结尾标签，不可以用 tab 或是空白来缩排。Markdown 的产生器有智能判断，可以避免在区块标签前后加上没有必要的 `<p>` 标签。

举例来说，在 Markdown 文档里加上一段 HTML 表格：

```
This is a regular paragraph.

<table>
  <tr>
    <td>Foo</td>
  </tr>
</table>

This is another regular paragraph.
```

请注意，Markdown 语法在 HTML 区块标签中将不会被进行处理。例如，无法在 HTML 区块内使用 Markdown 形式的 `*强调*`。

HTML 的区段标签如 `<span>`、`<cite>`、`<del>` 则不受限制，可以在 Markdown 的段落、列表或是标题里任意使用。依照个人习惯，甚至可以不用 Markdown 格式，而采用 HTML 标签来格式化。举例说明：如果比较喜欢 HTML 的 `<a>` 或 `<img>` 标签，可以直接使用这些标签，而不用 Markdown 提供的链接或是图片标示语法。

HTML 区段标签和区块标签不同，在区段标签的范围内，Markdown 的语法是有效的。

## 特殊字符自动转换

在 HTML 文档中，有两个字符需要特殊处理：`<` 和 `&`。`<` 符号用于起始标签，`&` 符号则用于标记 HTML 实体，如果只是想要使用这些符号，必须要使用实体的形式，像是 `&lt;` 和 `&amp;`。

`&` 符号其实很让写作网络文档的人很困扰，如果要输入「AT&T」，必须要写成「`AT&amp;T`」，还得转换网址内的 `&` 符号，如果要链接到：

```
http://images.google.com/images?num=30&q=larry+bird
```

必须要把网址转成：

```
http://images.google.com/images?num=30&amp;q=larry+bird
```

才能放到链接标签的 `href` 属性里。不用说也知道这很容易忘记，这也可能是 HTML 标准检查所检查到的错误中，数量最多的。

Markdown 允许直接使用这些符号，但是要小心转义字符的使用，如果是在 HTML 实体中使用 `&` 符号的话，它不会被转换，而在其它情形下，它则会被转换成 `&amp;`。所以如果要在文档中插入一个著作权的符号，可以这样写：

```
&copy;
```

Markdown 不会对这段文字做修改，但是如果这样写：

```
AT&T
```

Markdown 就会将它转为：

```
AT&amp;T
```

类似的状况也会发生在 `<` 符号上，因为 Markdown 支持 行内 HTML，如果是使用 `<` 符号作为 HTML 标签使用，那 Markdown 也不会对它做任何转换，但是如果是写：

```
4 < 5
```

Markdown 将会把它转换为：

```
4 &lt; 5
```

不过需要注意的是，code 范围内，不论是行内还是区块，`<` 和 `&` 两个符号都一定会被转换成 HTML 实体，这项特性让你可以很容易地用 Markdown 写 HTML code（和 HTML 相对而言，HTML 语法中，要把所有的 `<` 和 `&` 都转换为 HTML 实体，才能在 HTML 文档里面写出 HTML code。）

## 区块元素

### 段落和换行

一个段落是由一个以上相连接的行句组成，而一个以上的空行则会切分出不同的段落（空行的定义是显示上看起来像是空行，便会被视为空行。比方说，若某一行只包含空白和 tab，则该行也会被视为空行），一般的段落不需要用空白或断行缩排。

「一个以上相连接的行句组成」这句话其实暗示了 Markdown 允许段落内的强迫断行，这个特性和其他大部分的 text-to-HTML 格式不一样（包括 MovableType 的「Convert Line Breaks」选项），其它的格式会把每个断行都转成 `<br />` 标签。

如果真的想要插入 `<br />` 标签，在行尾加上两个以上的空白，然后按 enter。

是的，这确实需要花比较多功夫来插入 `<br />`，但是「每个换行都转换为 `<br />`」的方法在 Markdown 中并不适合，Markdown 中 email 式的 区块引言 和多段落的 列表 在使用换行来排版的时候，不但更好用，还更好阅读。

### 标题

Markdown 支持两种标题的语法，Setext 和 atx 形式。

Setext 形式是用底线的形式，利用 `=`（一级标题）和 `-`（二级标题），例如：

```
This is an H1
=====

This is an H2
-----
```

任何数量的 `=` 和 `-` 都可以有效果。

Atx 形式则是在行首插入 1 到 6 个 `#`，对应到标题 1 到 6 级，例如：

```
# This is an H1

## This is an H2

##### This is an H6
```

可以选择性地「关闭」atx 样式的标题，这纯粹只是美观用的，若是觉得这样看起来比较舒适，就可以在行尾加上 `#`，而行尾的 `#` 数量也不用和开头一样（行首的井字数量决定标题的级别）：

```
# This is an H1 #

## This is an H2 ##

### This is an H3 #####
```

## Blockquotes

Markdown 使用 email 形式的区块引用，如果你很熟悉如何在 email 信件中引用，就知道怎么在 Markdown 文档中建立一个区块引用，那会看起来像是强迫断行，然后在每行的最前面加上 `>`：

```
> This is a blockquote with two paragraphs. Lorem ipsum dolor sit amet,  
> consectetur adipiscing elit. Aliquam hendrerit mi posuere lectus.  
> Vestibulum enim wisi, viverra nec, fringilla in, laoreet vitae, risus.  
>  
> Donec sit amet nisl. Aliquam semper ipsum sit amet velit. Suspendisse  
> id sem consectetur libero luctus adipiscing.
```

Markdown 也允许只在整个段落的第一行最前面加上 `>`：

```
> This is a blockquote with two paragraphs. Lorem ipsum dolor sit amet,  
consectetur adipiscing elit. Aliquam hendrerit mi posuere lectus.  
Vestibulum enim wisi, viverra nec, fringilla in, laoreet vitae, risus.  
  
> Donec sit amet nisl. Aliquam semper ipsum sit amet velit. Suspendisse  
id sem consectetur libero luctus adipiscing.
```

区块引用可以有级别（例如：引言内的引言），只要根据级别加上不同数量的 `>`：

```
> This is the first level of quoting.  
>  
> > This is nested blockquote.  
>  
> Back to the first level.
```

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

```
> ## This is a header.  
>  
> 1. This is the first list item.  
> 2. This is the second list item.  
>  
> Here's some example code:  
>  
>     return shell_exec("echo $input | $markdown_script");
```

任何标准的文本编辑器都能简单地建立 email 样式的引用，例如 BBEdit，可以选择文字后，从菜单中选择 *增加引言级别*。

## 列表

Markdown 支持有序列表和无序列表。

无序列表使用星号、加号或是减号作为列表标记：

```
* Red  
* Green  
* Blue
```

等同于：

```
+ Red  
+ Green  
+ Blue
```

也等同于：

```
- Red  
- Green  
- Blue
```

有序列表则使用数字接着一个英文句点：

1. Bird
2. McHale
3. Parish

很重要的一点是，在列表标记上使用的数字并不会影响输出的 HTML 结果，上面的列表所产生的 HTML 标记为：

```
<ol>
<li>Bird</li>
<li>McHale</li>
<li>Parish</li>
</ol>
```

如果你的列表标记写成：

1. Bird
1. McHale
1. Parish

或甚至是：

3. Bird
1. McHale
8. Parish

都会得到完全相同的 HTML 输出。重点在于，你可以让 Markdown 文档的列表数字和输出的结果相同，或是懒一点，可以完全不用在意数字的正确性。

如果使用懒惰的写法，建议第一个项目最好还是从 1. 开始，因为 Markdown 未来可能会支持有序列表的 `start` 属性。

列表项目标记通常是放在最左边，但是其实也可以缩排，最多三个空白，项目标记后面则一定要接着至少一个空白或 `tab`。

要让列表看起来更漂亮，可以把内容用固定的缩排整理好：

- \* Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam hendrerit mi posuere lectus. Vestibulum enim wisi, viverra nec, fringilla in, laoreet vitae, risus.
- \* Donec sit amet nisl. Aliquam semper ipsum sit amet velit. Suspendisse id sem consectetur libero luctus adipiscing.

但是如果你很懒，那也不一定需要：

- \* Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam hendrerit mi posuere lectus. Vestibulum enim wisi, viverra nec, fringilla in, laoreet vitae, risus.
- \* Donec sit amet nisl. Aliquam semper ipsum sit amet velit. Suspendisse id sem consectetur libero luctus adipiscing.

如果列表项目间用空行分开，Markdown 会把项目的内容在输出时用 `<p>` 标签包起来，举例来说：

- \* Bird
- \* Magic

会被转换为：

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

但是这个：

- \* Bird
- \* Magic

会被转换为：

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

列表项目可以包含多个段落，每个项目下的段落都必须缩排 4 个空白或是一个 tab：

```
1. This is a list item with two paragraphs. Lorem ipsum dolor
   sit amet, consectetur adipiscing elit. Aliquam hendrerit
   mi posuere lectus.

   Vestibulum enim wisi, viverra nec, fringilla in, laoreet
   vitae, risus. Donec sit amet nisl. Aliquam semper ipsum
   sit amet velit.

2. Suspendisse id sem consectetur libero luctus adipiscing.
```

如果每行都有缩排，看起来会好看很多，当然，再次地，如果你很懒惰，Markdown 也允许：

```
* This is a list item with two paragraphs.

   This is the second paragraph in the list item. You're
   only required to indent the first line. Lorem ipsum dolor
   sit amet, consectetur adipiscing elit.

* Another item in the same list.
```

如果要在列表项目内放进引言，那 `>` 就需要缩排：

```
* A list item with a blockquote:

  > This is a blockquote
  > inside a list item.
```

如果要放程序代码区块的话，该区块就需要缩排两次，也就是 8 个空白或是两个 tab：

```
* A list item with a code block:

   <code goes here>
```

当然，项目列表很可能会不小心产生，像是下面这样的写法：

```
1986. 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，可以在句点前面加上反斜杠。

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

## 程序代码区块

和程序相关的写作或是标签语言原始代码通常会有已经排版好的程序代码区块，通常这些区块我们并不希望它以一般段落文档的方式去排版，而是照原来的样子显示，Markdown 会用 `<pre>` 和 `<code>` 标签来把程序代码区块包起来。

要在 Markdown 中建立程序代码区块很简单，只要简单地缩排 4 个空白或是 1 个 tab 就可以，例如，下面的输入：

```
This is a normal paragraph:

    This is a code block.
```

Markdown 会转换成：

```
<p>This is a normal paragraph:</p>

<pre><code>This is a code block.
</code></pre>
```

---

这个每行一级的缩排（4 个空白或是 1 个 tab），都会被移除，例如：

```
Here is an example of AppleScript:

    tell application "Foo"
        beep
    end tell
```

会被转换为：

```
<p>Here is an example of AppleScript:</p>

<pre><code>tell application "Foo"
    beep
end tell
</code></pre>
```

一个程序代码区块会一直持续到没有缩排的那一行（或是文档结尾）。

在程序代码区块里面，`&`、`<` 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 原始代码，只需要复制贴上，再加上缩排就可以了，剩下的 Markdown 都会帮你处理，例如：

```
<div class="footer">
    &copy; 2004 Foo Corporation
</div>
```

会被转换为：

```
<pre><code>&lt;div class="footer"&gt;
    &amp;copy; 2004 Foo Corporation
&lt;/div&gt;
</code></pre>
```

程序代码区块中，一般的 Markdown 语法不会被转换，像是星号便只是星号，这表示你可以很容易地以 Markdown 语法撰写 Markdown 语法相关的文档。

## 分隔线

你可以在一行中用三个或以上的星号、减号、底线来建立一个分隔线，行内不能有其他东西。你也可以在星号中间插入空白。下面每种写法都可以建立分隔线：

```
* * *

***

*****

- - -

-----
```

---

## 区段元素

### 链接

Markdown 支持两种形式的链接语法：*行内*和*参考*两种形式。

不管是哪一种，链接的文字都是用 [方括号] 来标记。

要建立一个行内形式的链接，只要在方括号后面马上接着括号并插入网址链接即可，如果你还想要加上链接的 title 文字，只要在网址后面，用双引号把 title 文字包起来即可，例如：

```
This is [an example](http://example.com/ "Title") inline link.

[This link](http://example.net/) has no title attribute.
```

会产生：

```
<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>
```

如果是要链接到同样主机的资源，可以使用相对路径：

```
See my [About](/about/) page for details.
```

参考形式的链接使用另外一个方括号接在链接文字的括号后面，而在第二个方括号里面要填入用以辨识链接的标签：

```
This is [an example][id] reference-style link.
```

也可以选择性地在两个方括号中间加上空白：

```
This is [an example] [id] reference-style link.
```

接着，在文档的任意处，可以把这个标签的链接内容定义出来：

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

链接定义的形式为：

- 方括号，里面输入链接的辨识用标签
- 接着一个冒号
- 接着一个以上的空白或 tab
- 接着链接的网址
- 选择性地接着 title 内容，可以用单引号、双引号或是括号包着

下面这三种链接的定义都是相同：

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

**请注意：**有一个已知的问题是 Markdown.pl 1.0.1 会忽略单引号包起来的链接 title。

链接网址也可以用方括号包起来：

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

你也可以把 title 属性放到下一行，也可以加一些缩排，网址太长的话，这样会比较好看：

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

网址定义只有在产生链接的时候用到，并不会直接出现在文档之中。

链接辨识标签可以有字母、数字、空白和标点符号，但是并不区分大小写，因此下面两个链接是一样的：

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

*默认的链接标签功能*让你可以省略指定链接标签，这种情形下，链接标签和链接文字会视为相同，要用默认链接标签只要在链接文字后面加上一个空的方括号，如果要让 "Google" 链接到 google.com，可以简化成：

```
[Google][]
```



然后定义链接内容：

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

由于链接文字可能包含空白，所以这种简化的标签内也可以包含多个文字：

```
Visit [Daring Fireball][] for more information.
```

然后接着定义链接：

```
[Daring Fireball]: http://daringfireball.net/
```

链接的定义可以放在文档中的任何一个地方，我比较偏好直接放在链接出现段落的后面，也可以把它放在文档最后面，就像是批注一样。

下面是一个参考式链接的范例：

```
I get 10 times more traffic from [Google] [1] than from  
[Yahoo] [2] or [MSN] [3].
```

```
[1]: http://google.com/      "Google"  
[2]: http://search.yahoo.com/ "Yahoo Search"  
[3]: http://search.msn.com/   "MSN Search"
```

如果改成用链接名称的方式写：

```
I get 10 times more traffic from [Google][] than from  
[Yahoo][] or [MSN][].
```

```
[google]: http://google.com/      "Google"  
[yahoo]:  http://search.yahoo.com/ "Yahoo Search"  
[msn]:    http://search.msn.com/   "MSN Search"
```

上面两种写法都会产生下面的 HTML。

```
<p>I get 10 times more traffic from <a href="http://google.com/"  
title="Google">Google</a> than from  
<a href="http://search.yahoo.com/" title="Yahoo Search">Yahoo</a>  
or <a href="http://search.msn.com/" title="MSN Search">MSN</a>.</p>
```

下面是用行内形式写的同样一段内容的 Markdown 文档，提供作为比较之用：

```
I get 10 times more traffic from [Google](http://google.com/ "Google")  
than from [Yahoo](http://search.yahoo.com/ "Yahoo Search") or  
[MSN](http://search.msn.com/ "MSN Search").
```

参考式的链接其实重点不在于它比较好写，而是它比较好读，比较一下上面的范例，使用参考式的文章本身只有 81 个字符，但是用行内形式的链接则会增加到 176 个字符，如果是用纯 HTML 格式来写，会有 234 个字符，在 HTML 格式中，标签比文字还要多。

使用 Markdown 的参考式链接，可以让文档更像是浏览器最后产生的结果，把一些标记相关的信息移到段落文字之外，这样增加链接，文章的阅读感也不会被打断。

## 强调

Markdown 使用星号（\*）和底线（\_）作为标记强调字词的符号，被 \* 或 \_ 包围的字词会被转成用 `<em>` 标签包围，用两个 \* 或 \_ 包起来的话，则会被转成 `<strong>`，例如：

```
*single asterisks*  
  
_single underscores_  
  
**double asterisks**  
  
__double underscores__
```

会转成：

```
<em>single asterisks</em>

<em>single underscores</em>

<strong>double asterisks</strong>

<strong>double underscores</strong>
```

你可以随便用你喜欢的样式，唯一的限制是，用什么符号开启标签，就要用什么符号结束。

强调也可以直接差在文字中间：

```
un*frigging*believable
```

但是如果 `*` 和 `_` 两边都有空白的话，它们就只会被当成普通的符号。

如果要在文字前后直接插入普通的星号或底线，可以用反斜杠：

```
\*this text is surrounded by literal asterisks\*
```

## 程序代码

如果要标记一小段行内程序代码，可以用反引号把它包起来（```），例如：

```
Use the printf() function.
```

会产生：

```
<p>Use the printf() function.</p>
```

如果要在程序代码区段内插入反引号，可以用多个反引号来开启和结束程序代码区段：

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

这段语法会产生：

```
<p><code>There is a literal backtick (`) here.</code></p>
```

程序代码区段的起始和结束端都可以放入一个空白，起始端后面一个，结束端前面一个，这样就可以在区段的一开始就插入反引号：

```
A single backtick in a code span: `` ` ``

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

会产生：

```
<p>A single backtick in a code span: `</code></p>

<p>A backtick-delimited string in a code span: `foo`</code></p>
```

在程序代码区段内，`&` 和方括号都会被转成 HTML 实体，这样会比较容易插入 HTML 原始代码，Markdown 会把下面这段：

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

转为：

```
<p>Please don't use any &lt;blink&gt; tags.</p>
```

也可以这样写：

```
`&#8212;` is the decimal-encoded equivalent of `&mdash;`.
```

以产生：

```
<p><code>&#8212;</code> is the decimal-encoded  
equivalent of <code>&mdash;</code>.</p>
```

## 图片

很明显地，要在纯文本应用中设计一个「自然」的语法来插入图片是有一定难度的。

Markdown 使用一种和链接很相似的语法来标记图片，同样也允许两种样式：*行内*和*参考*。

行内图片的语法看起来像是：

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

详细叙述如下：

- 一个惊叹号 `!`
- 接着一个方括号，里面放上图片的替换文字
- 接着一个普通括号，里面放上图片的网址，最后还可以用引号包住并加上 选择性的 'title' 文字。

参考式的图片语法则长得像这样：

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

[id] 是图片参考的名称，图片参考的定义方式则和链接参考一样：

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

到目前为止，Markdown 还没有办法指定图片的宽高，如果需要的话，可以使用普通的 `<img>` 标签。

## 其它

### 自动链接

Markdown 支持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，链接的文字就和链接位置一样，例如：

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

Markdown 会转为：

```
<a href="http://example.com/">http://example.com</a>
```

自动的邮件链接也很类似，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，这样的格式可以混淆一些不好的信箱地址收集机器人，例如：

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

Markdown 会转成：

```
<a href="&#x6D;&#x61;i&#x6C;&#x74;&#x6F;:&#x61;&#x64;&#x64;&#x72;&#x65;  
&#115;&#115;&#64;&#101;&#120;&#x61;&#109;&#x70;&#x6C;e&#x2E;&#99;&#111;  
&#109;">&#x61;&#x64;&#x64;&#x72;&#x65;&#115;&#115;&#64;&#101;&#120;&#x61;  
&#109;&#x70;&#x6C;e&#x2E;&#99;&#111;&#109;</a>
```

在浏览器里面，这段字符串会变成一个可以点击的「address@example.com」链接。

（这种作法虽然可以混淆不少的机器人，但无法全部挡下来，不过这样也比什么都不做好些。无论如何，公开你的信箱终究会引来广告信件的。）

转义字符

Markdown 可以利用反斜杠来插入一些在语法中有其它意义的符号，例如：如果想要用星号加在文字旁边的方式来做出强调效果（但不用 `<em>` 标签），可以在星号的前面加上反斜杠：

```
\*literal asterisks\*
```

Markdown 支持在下面这些符号前面加上反斜杠来帮助插入普通的符号：

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


感谢

感谢 [leafy7382](#) 协助翻译，[hlb](#)、[Randylien](#) 帮忙润稿，[ethantw](#) 的汉字标准格式 · CSS Reset，[WM](#) 回报文字错误。

本文仅用于学习和交流目的，不代表图灵社区观点。非商业转载请注明译者、出处，并保留本文的原始链接。

markdown 文档

分享长微博 10



杨帆

发表于 2011-11-03 11:40

6



开源书和开源技术 -Markdown篇



Markdown测试学习



怎样在 Markdown 中使程序代码带上行号



学习使用 Markdown 语法



怎样使用 Markdown



markdown 的bug 吗？

无关联推荐[?]

评论

时间 推荐

推荐

1

Markdown在保持版面干净方面确实很方便，不过对于用惯了所见即所得，靠写作软件来进行排版的人来说，确实不好用。

毕竟即使是用Html，一般人也不是直接写代码，而是用编辑器。Html的编辑器比较多，也比较成熟了。而Markdown的编辑器好像还没有覆盖全部的排版功能。

单独的编辑软件，windows下有MarkdownPad，Mac下有Mou，程序都比较小，大家可以试试，在宽屏上更好用一些。



杨帆

2011-11-04 10:55

推荐

0

问一下，目录是怎么产生的



辛再甫

2014-01-03 14:49

是指社区电子书的目录吗？那个是由合集自动生成的，您可以创建一个合集看看 - 杨帆 2014-01-03 14:56

推荐

0

MARK

<http://www.xiaoshujiang.com>



看客看

2013-07-22 14:55

推荐

0

Mark



gyzhao

2013-06-23 17:38

推荐

0

正文中说：强调也可以直接差在文字中间  
应该是 插在。



黄志斌

2013-04-08 20:04

推荐

0

正文中说：  
不在 Markdown 涵盖范围之外的标签，...  
这里的 之外 应改为 之内。



黄志斌

2013-03-13 08:59

好，不过我发现涵盖范围 后面多停顿一下 好像就没问题 ^^ - 杨帆 2013-03-13 09:10

推荐

0

看小冰微博转发的Sublime Text 2，最近我也在用，感觉用来写Markdown很好玩。  
<http://www.ituring.com.cn/article/6815>



格瑞图

2012-07-21 12:14

推荐

0

就如同JSON之于XML，Markdown是一个简化版的HTML，在很多领域有非常好的效果。在图灵社区中，它可以有效地使广大会员发表的文章格式更加干净、清晰、风格统一。使广大的阅读者有更好的阅读体验。

当然，使用Markdown可能会存在这几个问题：

第一是不少人没有用过Markdown，需要少量的学习时间。但是从图灵社区这两个月的实践来看，基本上没有人提出由于Markdown语法太难，而无法发文。基本上几分钟的时间就可以掌握Markdown，比掌握HTML容易得多。

第二是对于某些比较特殊的要求的格式，Markdown不支持，但是绝大多数可以使用变通的方法来解决。

第三是对于直接从其他网站转贴的文章，通常格式需要调整，而不能简单一贴了之。从图灵社区的角度，并不鼓励直接转贴文章，如果确实有好文章值得转贴，也请发表人做一些适应性的版式调整，符合图灵社区网站的整体风格，这样可以大大改善读者的阅读体验。





温谦

2011-11-03 12:17

---

## 我要评论

需要登录后才能发言

☐ 记住我

登录