Markdown 语法说明

概述

宗旨

Markdown 的目标是实现「易读易写」。

可读性，无论如何，都是最重要的。一份使用 Markdown 格式撰写的文件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，而最大灵感来源其实是纯文本电子邮件的格式。

总之， Markdown 的语法全由一些符号所组成，这些符号经过精挑细选，其作用一目了然。比如：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。Markdown 的区块引用看起来就真的像是引用一段文字，就像你曾在电子邮件中见过的那样。

兼容 HTML

Markdown 语法的目标是：成为一种适用于网络的书写语言。

Markdown 不是想要取代 HTML，甚至也没有要和它相近，它的语法种类很少，只对应 HTML 标记的一小部分。Markdown 的构想不是要使得 HTML 文档更容易书写。在我看来， HTML 已经很容易写了。Markdown 的理念是，能让文档更容易读、写和随意改。HTML 是一种发布的格式，Markdown 是一种书写的格式。就这样，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。

要制约的只有一些 HTML 区块元素——比如 `<div>`、`<table>`、`<pre>`、`<p>` 等标签，必须在前后加上空行与其它内容区隔开，还要求它们的开始标签与结尾标签不能用制表符或空格来缩进。Markdown 的生成器有足够智能，不会在 HTML 区块标签外加上不必要的 `<p>` 标签。

例子如下，在 Markdown 文件里加上一段 HTML 表格：

```
这是一个普通段落。

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

这是另一个普通段落。
```

请注意，在 HTML 区块标签间的 Markdown 格式语法将不会被处理。比如，你在 HTML 区块内使用 Markdown 样式的`*强调*`会没有效果。

HTML 的区段（行内）标签如 `<span>`、`<cite>`、`<del>` 可以在 Markdown 的段落、列表或是标题里随意使用。依照个人习惯，甚至可以不用 Markdown 格式，而直接采用 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 提供的链接或是图像标签语法。

和处在 HTML 区块标签间不同，Markdown 语法在 HTML 区段标签间是有效的。

特殊字符自动转换

在 HTML 文件中，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，如果你只是想要显示这些字符的原型，你必须要使用实体的形式，像是 `<` 和 `&`。

`&` 字符尤其让网络文档编写者受折磨，如果你要打「`AT&T`」 ，你必须要写成「`AT&T`」。而网址中的 `&` 字符也要转换。比如你要链接到：

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

你必须要把网址转换写为：

```
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忽略，这也可能是 HTML 标准检验所检查到的错误中，数量最多的。

Markdown 让你可以自然地书写字符，需要转换的由它来处理好了。如果你使用的 `&` 字符是 HTML 字符实体的一部分，它会保留原状，否则它会被转换成 `&`。

所以你如果要在文档中插入一个版权符号 `©`，你可以这样写：

```
&copy;
```

Markdown 会保留它不动。而若你写：

```
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 允许 兼容 HTML ，如果你是把 `<` 符号作为 HTML 标签的定界符使用，那 Markdown 也不会对它做任何转换，但是如果你写：

```
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，你要把所有的 `<` 和 `&` 都转换为 HTML 实体，才能在 HTML 文件里面写出 HTML code。）

区块元素

段落和换行

一个 Markdown 段落是由一个或多个连续的文本行组成，它的前后要有一个以上的空行（空行的定义是显示上看起来像是空的，便会被视为空行。比方说，若某一行只包含空格和制表符，则该行也会被视为空行）。普通段落不该用空格或制表符来缩进。

「由一个或多个连续的文本行组成」这句话其实暗示了 Markdown 允许段落内的强迫换行（插入换行符），这个特性和其他大部分的 text-to-HTML 格式不一样（包括 Movable Type 的「Convert Line Breaks」选项），其它的格式会把每个换行符都转成 `<br />` 标签。

如果你确实想要依赖 Markdown 来插入 `<br />` 标签的话，在插入处先按入两个以上的空格然后回车。

的确，需要多费点事（多加空格）来产生 `<br />` ，但是简单地「每个换行都转换为 `<br />`」的方法在 Markdown 中并不适合，Markdown 中 email 式的 区块引用 和多段落的 列表 在使用换行来排版的时候，不但更好用，还更方便阅读。

标题

Markdown 支持两种标题的语法，类 Setext 和类 atx 形式。

类 Setext 形式是用底线的形式，利用 `=` （最高阶标题）和 `-` （第二阶标题），例如：

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

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

任何数量的 `=` 和 `-` 都可以有效果。

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

```
# 这是 H1

## 这是 H2

###### 这是 H6
```

你可以选择性地「闭合」类 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，你就可以在行尾加上 `#`，而行尾的 `#` 数量也不用和开头一样（行首的井字符数量决定标题的阶数）：

```
# 这是 H1 #

## 这是 H2 ##

### 这是 H3 ######
```

区块引用 Blockquotes

Markdown 标记区块引用是使用类似 email 中用 `>` 的引用方式。如果你还熟悉在 email 信件中的引言部分，你就知道怎么在 Markdown 文件中建立一个区块引用，那会看起来像是你自己先断好行，然后在每行的最前面加上 `>` ：

```
> This is a blockquote with two paragraphs. Lorem ipsum dolor sit amet,
> consectetuer 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 consectetuer libero luctus adipiscing.
```

Markdown 也允许你偷懒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,
consectetuer 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 consectetuer libero luctus adipiscing.
```

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 `>` ：

```
> 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、代码区块等：

```
> ## 这是一个标题。
>
> 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：
>
>     return shell_exec("echo $input | $markdown_script");
```

任何像样的文本编辑器都能轻松地建立 email 型的引用。例如在 BBEdit 中，你可以选取文字后然后从选单中选择增加引用阶层。

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，你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
```

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

如果你使用懒惰的写法，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：

```
*   Lorem ipsum dolor sit amet, consectetuer 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 consectetuer libero luctus adipiscing.
```

但是如果你懒，那也行：

```
*   Lorem ipsum dolor sit amet, consectetuer 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 consectetuer libero luctus adipiscing.
```

如果列表项目间用空行分开，在输出 HTML 时 Markdown 就会将项目内容用 `<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进 4 个空格或是 1 个制表符：

```
1.  This is a list item with two paragraphs. Lorem ipsum dolor
    sit amet, consectetuer 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 consectetuer libero luctus adipiscing.
```

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：

```
*   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, consectetuer adipiscing elit.

*   Another item in the same list.
```

如果要在列表项目内放进引用，那 `>` 就需要缩进：

```
*   A list item with a blockquote:

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

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：

        <代码写在这>
```

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，你可以在句点前面加上反斜杠。

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

代码区块

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

要在 Markdown 中建立代码区块很简单，只要简单地缩进 4 个空格或是 1 个制表符就可以，例如，下面的输入：

```
这是一个普通段落：

    这是一个代码区块。
```

Markdown 会转换成：

```
<p>这是一个普通段落：</p>

<pre><code>这是一个代码区块。
</code></pre>
```

这个每行一阶的缩进（4 个空格或是 1 个制表符），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>
```

一个代码区块会一直持续到没有缩进的那一行（或是文件结尾）。

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，再加上缩进就可以了，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>
```

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 语法相关的文件。

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：

```
* * *

***

*****

- - -

---------------------------------
```

区段元素

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。

不管是哪一种，链接文字都是用 [方括号] 来标记。

要建立一个行内式的链接，只要在方块括号后面紧接着圆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>
```

如果你是要链接到同样主机的资源，你可以使用相对路径：

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

参考式的链接是在链接文字的括号后面再接上另一个方括号，而在第二个方括号里面要填入用以辨识链接的标记：

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

你也可以选择性地在两个方括号中间加上一个空格：

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

接着，在文件的任意处，你可以把这个标记的链接内容定义出来：

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

链接内容定义的形式为：

- 方括号（前面可以选择性地加上至多三个空格来缩进），里面输入链接文字
- 接着一个冒号
- 接着一个以上的空格或制表符
- 接着链接的网址
- 选择性地接着 title 内容，可以用单引号、双引号或是括弧包着

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，也可以加一些缩进，若网址太长的话，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件之中。

链接辨别标签可以有字母、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]
```

隐式链接标记功能让你可以省略指定链接标记，这种情形下，链接标记会视为等同于链接文字，要用隐式链接标记只要在链接文字后面加上一个空的方括号，如果你要让 "Google" 链接到 google.com，你可以简化成：

```
[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，所以这种简化型的标记内也许包含多个单词：

```
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。

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>
```

下面是用行内式写的同样一段内容的 Markdown 文件，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，但是用行内形式的却会增加到 176 个字元，如果是用纯 HTML 格式来写，会有 234 个字元，在 HTML 格式中，标签比文本还要多。

使用 Markdown 的参考式链接，可以让文件更像是浏览器最后产生的结果，让你可以把一些标记相关的元数据移到段落文字之外，你就可以增加链接而不让文章的阅读感觉被打断。

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，你用什么符号开启标签，就要用什么符号结束。

强调也可以直接插在文字中间：

```
un*frigging*believable
```

但是**如果你的 `*` 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，你可以用反斜线：

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

代码

如果要标记一小段行内代码，你可以用反引号把它包起来（`` ` ``），例如：

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

会产生：

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

如果要在代码区段内插入反引号，你可以用多个反引号来开启和结束代码区段：

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

这段语法会产生：

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

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，这样你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>`</code></p>

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

在代码区段内，`&` 和方括号**都**会被自动地转成 HTML 实体，这使得插入 HTML 原始码变得很容易，Markdown 会把下面这段：

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

转为：

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

你也可以这样写：

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

以产生：

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

图片

很明显地，要在纯文字应用中设计一个「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： 行内式和参考式。

行内式的图片语法看起来像是：

```
![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，里面放上图片的替代文字
- 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连结参考一样：

```
[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，如果你需要的话，你可以使用普通的 `<img>` 标签。

其它

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：

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

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，这段字串（其实是 `<a href="mailto:address@example.com">address@example.com</a>` ）会变成一个可以点击的「address@example.com」链接。

（这种作法虽然可以糊弄不少的机器人，但并不能全部挡下来，不过总比什么都不做好些。不管怎样，公开你的信箱终究会引来广告信件的。）

反斜杠

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：如果你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），你可以在星号的前面加上反斜杠：

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

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
```