

# Markdown For Typora

## Overview概述

**Markdown** is created by [Daring Fireball](#), the original guideline is [here](#). Its syntax, however, varies between different parsers or editors. **Typora** is using [GitHub Flavored Markdown\(GFM\)](#).

**Markdown**是由[DaringFireball](#)创建，其最初的指南在[这里](#)。它的语法其实在不同的解析器和编辑器中有出入。**Typora**的语法使用的是[Github](#)格式。

Please note that HTML fragments in markdown source will be recognized but not parsed or rendered. Also, there may be small reformatting on the original markdown source code after saving.

请注意，在Markdown文件中，HTML页面将被直接展示而不会分析或渲染。同时，HTML也可能在未保存的Markdown源文件上被轻微地重新格式化（按HTML进行渲染）。

Outline

目录

以下<sup>1</sup>是键入[TOC]插入的。

### Markdown For Typora

Overview概述

Block Elements语法单元

Paragraph and line breaks段落和换行

Headers标题

Blockquotes块引用

Lists清单/列表

Task List任务清单

(Fenced) Code Blocks代码块

Math Blocks数学公式块

Tables表格

Footnotes脚注

Horizontal Rules水平线

YAML Front Matter（不太了解...）

Table of Contents (TOC)目录

Diagrams (Sequence, Flowchart and Mermaid)图表（包括时序图、流程图和绘图插件Mermaid）

Span Elements 行内元素

Links链接

Internal Links文内链接

Reference Links引用链接

URL统一资源定位地址

Images图像

Emphasis斜体/加重

Strong强调

Code代码

Strikethrough删除线

Underline下划线

Emoji 😊表情

HTML超文本链接标识语言

Inline Math文章内的数学公式  
Subscript下标  
Superscript上标  
Highlight高亮

## Block Elements语法单元

### Paragraph and line breaks段落和换行

A paragraph is simply one or more consecutive lines of text. In markdown source code, paragraphs are separated by more than one blank lines. In Typora, you only need to press `Return` to create a new paragraph.

段落仅仅是一行或多行连续的文本。在Markdown源文件中，段落由多个空白行分隔。在Typora中，你只需要按回车键创建一个新的段落。

Press `Shift` + `Return` to create a single line break. However, most markdown parser will ignore single line break, to make other markdown parsers recognize your line break, you can leave two whitespace at the end of the line, or insert `<br/>`.

按Shift +回车创建单行换行符。然而，大多数的Markdown解析器将忽略换行，为了让其他Markdown解析器识别出你的换行，你可以把两个空格排在最后，或者插入这个`<br/>`。

### Headers标题

Headers use 1-6 hash characters at the start of the line, corresponding to header levels 1-6. For example:

在一行的开头使用1至6个“#”符号，标明这是个1至6级的标题，例如：

```
1 # This is an H1
2
3 ## This is an H2
4
5 ##### This is an H6
```

In typora, input '#'s followed by title content, and press `Return` key will create a header.

在Typora中，输入“#”后跟标题内容，再按下回车即创建一个标题。

### Blockquotes块引用

Markdown uses email-style > characters for block quoting. They are presented as:

Markdown使用电子邮件样式的“>”字符块引用。它们被呈现为以下样式：

This is a blockquote with two paragraphs. This is first paragraph.

This is second paragraph. Vestibulum enim wisi, viverra nec, fringilla in, laoreet vitae, risus.

This is another blockquote with one paragraph. There is three empty line to seperate two blockquote.

\*注:

1. Markdown Reference中给的例子似乎有些问题——给的例子中，键入“>”展示的是如下面代码块的样式，而实际上却是上面的引用样式；如果用Typora中右键-插入-代码块，建立的是下面的样式；而在右键-插入中并没有上面的引用样式；
2. 在上面的引用样式中，输入完内容后单击回车仍在引用块中，连续按下两个回车就可以退出引用块。

```
1 > This is a blockquote with two paragraphs. This is first paragraph.
2 >
3 > This is second pragraph.Vestibulum enim wisi, viverra nec, fringilla in, laoreet vitae,
  risus.
4
5
6
7 > This is another blockquote with one paragraph. There is three empty line to seperate two
  blockquote.
```

In typora, just input '>' followed by quote contents a block quote is generated. Typora will insert proper '>' or line break for you. Block quote inside anther block quote is allowed by adding additional levels of '>'.

在Typora中，只要输入“>”，后面引用的内容块自动生成。Typora会为你适当地插入“>”或换行。在块引用中允许添加额外的“>”。

## Lists清单/列表

Input `* list item 1` will create an un-ordered list, the `*` symbol can be replace with `+` or `-`.

输入“\*”+“内容1”将创建一个无序的列表，除了“\*”，“+”和“-”也可以创建无序列表。

Input `1. list item 1` will create an ordered list, their markdown source code is like:

输入“1.（空格）”+“内容1”将创建一个有序的列表，两种方式的源代码如下：

```
1 ## un-ordered list
2 *   Red
3 *   Green
4 *   Blue
5
6 ## ordered list
7 1.   Red
8 2.   Green
9 3.   Blue
```

## Task List任务清单

Task lists are lists with items marked as either `[]` or `[x]` (incomplete or complete). For example:

任务清单是指有“[]”或“[X]”（未完成或完成）项的列表。举个例子，键入代码如下：

```
1 - [ ] a task list item
2 - [ ] list syntax required
3 - [ ] normal **formatting**, @mentions, #1234 refs
4 - [ ] incomplete
5 - [x] completed
```

样式如下：

- ☐ a task list item
- ☐ list syntax required
- ☐ normal **formatting**, @mentions, #1234 refs
- ☐ incomplete
- ☒ completed

You can change the complete/incomplete state by click the checkbox before the item.

单击项目前的复选框，可以改变完成状态。

## (Fenced) Code Blocks代码块

Typora only support fences in Github Flavored Markdown. Original code blocks in markdown is not supported.

Typora仅支持Github Flavored Markdown语法标准中的“围栏”，不支持Markdown中原始的代码块样式。

Using fences is easy: Input ``` and press `return`. Add an optional language identifier after ``` and we'll run it through syntax highlighting:

使用围栏很容易：输入```再回车。添加任意的字符在“”之后，我们就能将它渲染为语法高亮状态。

\*注：

1. 输入的```为Esc键下方的键，而不是单引号；
2. 另一种方法，使用“右键-插入-代码块”，可以建立；

```
1 Here's an example:
2
3 ```
4 function test() {
5     console.log("notice the blank line before this function?");
6 }
7 ```
8
9 syntax highlighting:
10 ```ruby
11 require 'redcarpet'
12 markdown = Redcarpet.new("Hello World!")
13 puts markdown.to_html
14 ```
```

## Math Blocks数学公式块

You can render *LaTeX* mathematical expressions using **MathJax**.

你可以使用LaTeX格式的数学表达式

Input `$$`, then press 'Return' key will trigger an input field which accept *Tex/LaTex* source. Following is an example:

输入“\$\$”，然后回车就能触发出一个能接收LaTeX代码的输入框，如下所示：

$$\mathbf{V}_1 \times \mathbf{V}_2 = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \frac{\partial X}{\partial u} & \frac{\partial Y}{\partial u} & 0 \\ \frac{\partial X}{\partial v} & \frac{\partial Y}{\partial v} & 0 \end{vmatrix}$$

In markdown source file, math block is *LaTeX* expression wrapped by '\$\$' mark:

在Markdown文件中，数学公式块是由“\$\$”标记的LaTeX表达式：

```
1  $$
2  \mathbf{V}_1 \times \mathbf{V}_2 = \begin{vmatrix}
3  \mathbf{i} & \mathbf{j} & \mathbf{k} \\
4  \frac{\partial X}{\partial u} & \frac{\partial Y}{\partial u} & 0 \\
5  \frac{\partial X}{\partial v} & \frac{\partial Y}{\partial v} & 0 \\
6  \end{vmatrix}
7  $$
```

## Tables表格

Input `| First Header | Second Header |` and press `return` key will create a table with two column.

输入“| 第一个项目 | 第二个项目 |”再回车就能创建一个两列的表格。

After table is created, focus on that table will pop up a toolbar for table, where you can resize, align, or delete table. You can also use context menu to copy and add/delete column/row.

表格创建之后，鼠标放在表格上会弹出一个表格工具栏，通过工具栏你可以调整表格的尺寸、对齐，也可以删除表格。你还可以使用上下文菜单来复制和添加/删除列/行。

Following descriptions can be skipped, as markdown source code for tables are generated by typora automatically.

下面的描述可以略过，在Typora中会自动渲染Markdown文件的表格。

In markdown source code, they look like:

源代码如下所示：

```
1  | First Header | Second Header |
2  | ----- | ----- |
3  | Content Cell | Content Cell |
4  | Content Cell | Content Cell |
```

渲染结果如下：

First Header	Second Header
Content Cell	Content Cell
Content Cell	Content Cell

You can also include inline Markdown such as links, bold, italics, or strikethrough.

你也可以在表格内插入链接、设置字体格式或对齐格式。

Finally, by including colons `:` within the header row, you can define text to be left-aligned, right-aligned, or center-aligned:

最后，通过在标题行的下一行添加“`:`”可以定义文本为左对齐、右对齐或者居中显示。

1		Left-Aligned		Center Aligned		Right Aligned	
2		:-----		:-----:		-----:	
3		col 3 is		some wordy text		\$1600	
4		col 2 is		centered		\$12	
5		zebra stripes		are neat		\$1	

渲染格式如下：

Left-Aligned	Center Aligned	Right Aligned
col 3 is	some wordy text	\$1600
col 2 is	centered	\$12
zebra stripes	are neat	\$1

A colon on the left-most side indicates a left-aligned column; a colon on the right-most side indicates a right-aligned column; a colon on both sides indicates a center-aligned column.

左边有冒号表示所在的列左对齐；后略。

## Footnotes脚注

1	You can create footnotes like this[^footnote].
2	你可以像这样创建脚注[^脚注]。
3	
4	[^footnote]: Here is the <i>*text*</i> of the <b>**footnote**</b> .

will produce:

将会产生：

You can create footnotes like this[^footnote].

[^]: Here is the *text* of the **footnote**.

Mouse on the ‘footnote’ superscript to see content of the footnote.

鼠标悬停在“footnote”上标上就能看见脚注内容。

## Horizontal Rules水平线

Input `***` or `---` on a blank line and press `return` will draw a horizontal line.

在空白的一行输入\*\*\*或---回车即可画一条水平线。如下所示：

---

---

## YAML Front Matter（不太了解...）

Typora support [YAML Front Matter](#) now. Input `---` at the top of the article and then press `Enter` will introduce one. Or insert one metadata block from the menu.

Typora现在支持YAML Front Matter。在Markdown文件的首页第一行输入“---”回车即可创建。或者从菜单中插入元数据块。

## Table of Contents (TOC)目录

Input `[toc]` then press `Return` key will create a section for “Table of Contents” extracting all headers from one’s writing, its contents will be updated automatically.

输入[toc]后回车即可创建目录，目录会自动提取文中所有的标题，目录内容会随文章的修改自动更新。

## Diagrams (Sequence, Flowchart and Mermaid)图表（包括时序图、流程图和绘图插件Mermaid）

Typora supports, [sequence](#), [flowchart](#) and [mermaid](#), after this feature is enabled from preference panel.

Typora可从选项面板中启用对时序图、流程图和绘图插件Mermaid的支持。

See this [document](#) for detail.

详细信息可查阅[文件](#)。

## Span Elements 行内元素

Span elements will be parsed and rendered right after your typing. Moving cursor in middle of those span elements will expand those elements into markdown source. Following will explain the syntax of those span element.

行内元素将在键入后立即解析和呈现。在这些行内元素中，移动光标将使这些元素融入Markdown源。下面将解释这些跨度元素的语法。

## Links链接

Markdown supports two style of links: inline and reference.

Markdown支持两种样式的链接：内联链接和引用。

In both styles, the link text is delimited by [square brackets].

这两种样式中，链接的文本都是在方括号内的。

To create an inline link, use a set of regular parentheses immediately after the link text's closing square bracket. Inside the parentheses, put the URL where you want the link to point, along with an optional title for the link, surrounded in quotes. For example:

要创建内联链接，在方括号后立即添加一组圆括号，圆括号内填入想指向的URL，该URL的标题（可以是任意文本）放入之前的方括号中，例如：

```
1 This is [an example](http://example.com/ "Title") inline link.
2
3 [This link](http://example.net/) has no title attribute.
```

will produce:

将会产生：

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

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


## Internal Links文内链接

**You can set the href to headers**, which will create a bookmark that allow you to jump to that section after clicking. For example:

你可以设置链接的标题，这将创建一个书签，点击后让你跳到该章节。例如：

Command(on Windows: Ctrl) + Click [This link](#) will jump to header `Block Elements`.

Command键（Ctrl键）+鼠标单击[这个链接](#)将会跳转到语法单元。

To see how to write that, please move cursor or click that link with  key pressed to expand the element into markdown source.

如果要查看上面的例子是如何建立的文内链接，请将光标移至文内链接上并单击。

## Reference Links引用链接

Reference-style links use a second set of square brackets, inside which you place a label of your choosing to identify the link:

引用样式的链接使用了两组方括号，第二个方括号内部填入你选择的标签名，这个标签的内容是你想指向的超链接：

```
1 This is [an example][id] reference-style link.
2
3 Then, anywhere in the document, you define your link label like this, on a line by itself:
4
5 [id]: http://example.com/ "Optional Title Here"
```



In typora, they will be rendered like:

在Typora中，它们会被渲染为：

This is [an example](#) reference-style link.

The implicit link name shortcut allows you to omit the name of the link, in which case the link text itself is used as the name. Just use an empty set of square brackets — e.g., to link the word “Google” to the google.com web site, you could simply write:

隐式链接的标签名允许省略，在这种情况下将使用超链接文本（第一个方括号的内容）作为标签名。第二个方括号内部为空，不填写任何内容。举个例子，要链接到“谷歌”到Google.com网站，可以简单地写为：

```
1  [Google][]  
2  And then define the link:  
3  
4  [Google]: http://google.com/
```

In typora click link will expand it for editing, command+click will open the hyperlink in web browser.

在Typora中点击链接将会跳转到编辑模式，Command键+单击将会在浏览器中打开超链接。

## URL统一资源定位地址

Typora allows you to insert urls as links, wrapped by `<` brackets `>`.

Typora允许你使用尖括号“<>”插入URL。

`<i@typora.io>` becomes [i@typora.io](mailto:i@typora.io).

Typora will also auto link standard URLs. e.g: [www.google.com](http://www.google.com).

Typora也会自动转换标准的URL为超链接。举例：[www.google.com](http://www.google.com)。

## Images图像

Image looks similar with links, but it requires an additional `!` char before the start of link. Image syntax looks like this:

图像看起来和链接很相似，但是它要在链接前使用“!”字符。插入图像的语法如下：

```
1  ![Alt text](G:/Desktop/Batman.jpg)  
2  
3  ![Alt text](/path/to/img.jpg "Optional title")
```

### 示例

You are able to use drag & drop to insert image from image file or web browser. And modify the markdown source code by clicking on the image. Relative path will be used if image is in same directory or sub-directory with current editing document when drag & drop.

你可以使用拖放来从图片文件或浏览器中插入图像。可以通过点击图片来改变Markdown文件代码。当拖放的图像与当前的编辑文档在同一目录或子目录中时，将使用相对路径。

For more tips on images, please read <http://support.typora.io/Images/>

获取更多关于图像的提示可访问 <http://support.typora.io/Images/>

## Emphasis斜体/加重

Markdown treats asterisks (\*) and underscores (\_) as indicators of emphasis. Text wrapped with one \* or \_ will be wrapped with an HTML <em> tag. E.g:

Markdown视“\*”和“\_”作为强调的指示符。被“\*”或“\_”包括的文本会被渲染为HTML的<em>标签。如下面的例子：

```
1  *single asterisks*
2
3  _single underscores_
```

output:

输出：

*single asterisks*

*single underscores*

GFM will ignores underscores in words, which is commonly used in code and names, like this:

GFM语法会忽略文字间的下划线，视其为正常的文本，像下面这样：

```
1  wow_great_stuff
2
3  do_this_and_do_that_and_another_thing.
4
```

To produce a literal asterisk or underscore at a position where it would otherwise be used as an emphasis delimiter, you can backslash escape it:

可以使用反斜杠转义正常显示“\*”和“\_”，避免加重。

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

Typora recommends to use \* symbol.

Typora推荐使用“\*”符号。

## Strong强调

double \*'s or \_'s will be wrapped with an HTML <strong> tag, e.g:

使用“\*\*”或“\_\_”将会渲染为HTML中的<strong>标签，举个例子：

```
1  **double asterisks**
2
3  __double underscores__
```

output:

### double asterisks

### double underscores

Typora recommends to use `**` symbol.

Typora推荐使用“\*\*”符号。

## Code代码

To indicate a span of code, wrap it with backtick quotes (```). Unlike a pre-formatted code block, a code span indicates code within a normal paragraph. For example:

要表示一行代码，用反引号引用(```)。与预先格式化的代码块不同，代码行标示正常段落中的代码。例如：

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

will produce:

将会产生：

Use the `printf()` function.

## Strikethrough删除线

GFM adds syntax to create strikethrough text, which is missing from standard Markdown.

GFM补充了创建删除线的语法，这是标准Markdown语法中没有的。

`~~Mistaken text.~~` becomes ~~Mistaken text.~~

## Underline下划线

Underline is powered by raw HTML.

下划线由原始的HTML支持。

`<u>Underline</u>` becomes Underline.

## Emoji 😊表情

Input emoji with syntax `:smile:`.

输入Emoji表情的语法为`\:smile\:`

User can trigger auto-complete suggestions for emoji by pressing `ESC` key, or trigger it automatically after enable it on preference panel. Also, input UTF8 emoji char directly from `Edit` -> `Emoji & Symbols` from menu bar is also supported.

用户可以按下Esc键触发Emoji的自动填充，或在选项面板启用后自动触发。另外，也可以从菜单栏输入UTF-8编码的Emoji字符直接生产Emoji。

## HTML超文本链接标识语言

Typora cannot render html fragments. But typora can parse and render very limited HTML fragments, as an extension of Markdown, including:

Typora不能渲染HTML页面。但Typora可以分析和渲染非常有限的、Markdown扩展的HTML片段，如：

- Underline: `<u>underline</u>`
- Image: `` (And `width`, `height` attribute in HTML tag, and `width`, `height`, `zoom` style in `style` attribute will be applied.)
- Comments: `<!-- This is some comments -->`
- Hyperlink: `<a href="http://typora.io" target="_blank">link</a>`.

Most of their attributes, styles, or classes will be ignored. For other tags, typora will render them as raw HTML snippets.

大多数属性、样式或类都会被忽视。除上述外的标签，Typora将渲染为原始的HTML片段。

But those HTML will be exported on print or export.

但这些HTML可以在打印或导出中渲染。

## Inline Math文章内的数学公式

To use this feature, first, please enable it in `Preference` Panel -> `Markdown` Tab. Then use `$` to wrap TeX command, for example: `$\lim_{x \to \infty} \exp(-x) = 0$` will be rendered as LaTeX command.

要使用这项特性，请在选项面板中开启Markdown Tab。然后使用“\$”转为TeX命令，例如：`$\lim_{x \to \infty} \exp(-x) = 0$`。

将会生成：

$$\lim_{x \rightarrow \infty} \exp(-x) = 0$$

To trigger inline preview for inline math: input “\$”, then press `ESC` key, then input TeX command, a preview tooltip will be visible like below:

要触发文内公式预览，输入“\$”然后按Esc键，就能转到TeX命令，产生像下面一样的预览工具：

## Subscript下标

To use this feature, first, please enable it in `Preference` Panel -> `Markdown` Tab. Then use `~` to wrap subscript content, for example: `H~2~0`, `X~long\ text~` /

要使用这个功能，首先请在选项面板中开启Markdown Tab。然后使用“~”来渲染下标内容，例如：`H~2~0`，`X~long\ text~` /。

将会产生：

H<sub>2</sub>O, X<sub>long text</sub>

## Superscript上标

To use this feature, first, please enable it in `Preference` Panel -> `Markdown` Tab. Then use `^` to wrap superscript content, for example: `X^2^`.

要使用这个功能，首先请在选项面板中开启Markdown Tab。然后使用"^"来渲染上标内容，举个例子：`x^2^`。

将会产生：

$x^2$

## Highlight高亮

To use this feature, first, please enable it in `Preference` Panel -> `Markdown` Tab. Then use `==` to wrap superscript content, for example: `==highlight==`.

要使用这个功能，首先请在选项面板中开启Markdown Tab。然后使用“==”来渲染指定内容，举个例子：

`==highlight==`。

将会产生：

highlight

---

1. 另起一行键入TOC，回车即可。 [↩](#)