

# Markdown For Typora

---

## Overview

---

**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](#).

### Markdown For Typora

- Overview

- Block Elements

  - Paragraph and line breaks

  - Headers

  - Blockquotes

  - Lists

  - Task List

  - (Fenced) Code Blocks

  - Math Blocks

  - Tables

  - Footnotes

  - Horizontal Rules

  - YAML Front Matter

  - Table of Contents (TOC)

- Span Elements

  - Links

    - Internal Links

    - Reference Links

  - URLs

  - Images

Emphasis

Strong

Code

Strikethrough

Underlines

Emoji 🤪

Inline Math

Subscript

Superscript

Highlight

HTML

Embed Contents

Video

Other HTML Support

# Block Elements

---

## Paragraph and line breaks

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

Press `Shift` + `Return` to create a single line break. Most other markdown parsers will ignore single line breaks, so in order to make other markdown parsers recognize your line break, you can leave two spaces at the end of the line, or insert `<br />`.

# Headers

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



```
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.

# Blockquotes

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



```
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, inputting ‘>’ followed by your quote contents will generate a quote block. Typora will insert a proper ‘>’ or line break for you. Nested block quotes (a block quote inside another block quote) by adding additional levels of ‘>’ .

## Lists

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

Input `1. list item 1` will create an ordered list - their markdown source code is as follows:



```
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

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



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

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

## (Fenced) Code Blocks

Typora only supports fences in GitHub Flavored Markdown. Original code blocks in markdown are not supported.

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



```
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
```

# Math Blocks

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

To add a mathematical expression, input `$$` and press the 'Return' key. This will trigger an input field which accepts *Tex/LaTeX* source. For example:

$$\mathbf{V}_1 \times \mathbf{V}_2 = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \frac{\partial X}{\partial u} & \frac{\partial Y}{\partial u} & 0 \\ \frac{\partial X}{\partial v} & \frac{\partial Y}{\partial v} & 0 \end{vmatrix} \quad (1)$$

In the markdown source file, the math block is a *LaTeX* expression wrapped by a pair of ‘`$$`’ marks:

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

You can find more details [here](#).

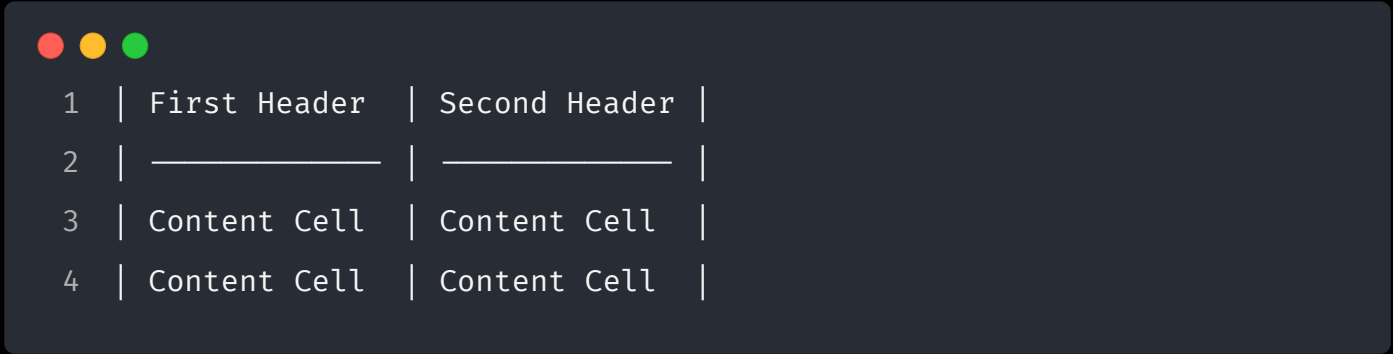
# Tables

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

After a table is created, putting focus on that table will open up a toolbar for the table where you can resize, align, or delete the table. You can also use the context menu to copy and add/delete individual columns/rows.

The full syntax for tables is described below, but it is not necessary to know the full syntax in detail as the markdown source code for tables is generated automatically by Typora.

In markdown source code, they look like:



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

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

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



```
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   |
```

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.

## Footnotes



```
1 You can create footnotes like this[^footnote].
2
3 [^footnote]: Here is the *text* of the **footnote**.
```

will produce:

You can create footnotes like this <sup>1</sup>.

Hover over the ‘footnote’ superscript to see content of the footnote.

## Horizontal Rules

Inputting `***` or `---` on a blank line and pressing `return` will draw a horizontal line.

---



# YAML Front Matter

Typora now supports **YAML Front Matter**. Input `---` at the top of the article and then press `Return` to introduce a metadata block. Alternatively, you can insert a metadata block from the top menu of Typora.

## Table of Contents (TOC)

Input `[toc]` and press the `Return` key. This will create a “Table of Contents” section. The TOC extracts all headers from the document, and its contents are updated automatically as you add to the document.

## Span Elements

Span elements will be parsed and rendered right after typing. Moving the cursor in middle of those span elements will expand those elements into markdown source. Below is an explanation of the syntax for each span element.

## Links

Markdown supports two styles of links: inline and reference.

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

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:



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

will produce:

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

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

## Internal Links

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

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

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

## Reference Links

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



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

In Typora, they will be rendered like so:

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 — for example, to link the word “Google” to the google.com web site, you could simply write:



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

In Typora, clicking the link will expand it for editing, and command+click will open the hyperlink in your web browser.

## URLs

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

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

Typora will also automatically link standard URLs. e.g: [www.google.com](#).

# Images

Images have similar syntax as links, but they require an additional `!` char before the start of the link. The syntax for inserting an image looks like this:

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

You are able to use drag & drop to insert an image from an image file or your web browser. You can modify the markdown source code by clicking on the image. A relative path will be used if the image that is added using drag & drop is in same directory or sub-directory as the document you're currently editing.

If you're using markdown for building websites, you may specify a URL prefix for the image preview on your local computer with property `typora-root-url` in YAML Front Matters. For example, input `typora-root-url:/User/Abner/Website/typora.io/` in YAML Front Matters, and then `![alt](/blog/img/test.png)` will be treated as `!(alt)(file:///User/Abner/Website/typora.io/blog/img/test.png)` in Typora.

You can find more details [here](#).

# Emphasis

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



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

output:

*single asterisks*

*single underscores*

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

```
wow_great_stuff  
  
do_this_and_do_that_and_another_thing.
```

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:



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

Typora recommends using the `*` symbol.

## Strong

A double `*` or `_` will cause its enclosed contents to be wrapped with an HTML `<strong>` tag, e.g:



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

output:

**double asterisks**

**double underscores**

Typora recommends using the `**` symbol.

## Code

To indicate an inline 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:



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

will produce:

Use the `printf()` function.

## Strikethrough

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

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

# Underlines

Underline is powered by raw HTML.

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

# Emoji 😊

Input emoji with syntax `:smile:`.

User can trigger auto-complete suggestions for emoji by pressing `ESC` key, or trigger it automatically after enabling it on preference panel. Also, inputting UTF-8 emoji characters directly is also supported by going to `Edit` -> `Emoji & Symbols` in the menu bar (macOS).

# Inline Math

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

To trigger inline preview for inline math: input “\$” , then press the `ESC` key, then input a TeX command.

You can find more details [here](#).

# Subscript

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

## Superscript

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

## Highlight

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

## HTML

You can use HTML to style content what pure Markdown does not support. For example, use `<span style="color:red">this text is red</span>` to add text with red color.

## Embed Contents

Some websites provide iframe-based embed code which you can also paste into Typora. For example:





```
1 <iframe height='265' scrolling='no' title='Fancy Animated SVG Menu'  
  src='http://codepen.io/jeangontijo/embed/0xVywj/?height=265&theme-  
  id=0&default-tab=css,result&embed-version=2' frameborder='no'  
  allowtransparency='true' allowfullscreen='true' style='width:  
  100%;'></iframe>
```

## Video

You can use the `<video>` HTML tag to embed videos. For example:



```
1 <video src="xxx.mp4" />
```

## Other HTML Support

You can find more details [here](#).

---

1. Here is the *text* of the **footnote**. ↩