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]().

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.

Outline

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 more than one blank lines. In Typora, you only need to press `Return` to create a new paragraph.

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/>`.

Headers

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

```
# This is an H1

## This is an H2

###### 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:

```
> This is a blockquote with two paragraphs. This is first paragraph.
>
> This is second pragraph.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.
```

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 '>'.

Lists

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

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

```
## un-ordered list
*    Red
*    Green
*    Blue

## ordered list
1.   Red
2.   Green
3.   Blue
```

Task List

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

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

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

(Fenced) Code Blocks

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

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

```
Here's an example:

```
function test() {
  console.log("notice the blank line before this function?");
}
```

syntax highlighting:
```ruby
require 'redcarpet'
markdown = Redcarpet.new("Hello World!")
puts markdown.to_html
```
```

Math Blocks

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

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

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

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

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

Tables

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

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.

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

In markdown source code, they look like:

```
| 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.

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

```
| 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.

Footnotes

```
You can create footnotes like this[^footnote].

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

will produce:

You can create footnotes like this[1].

Mouse on the 'footnote' superscript to see content of the footnote.

Horizontal Rules

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

YAML Front Matter

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.

Table of Contents (TOC)

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.

Diagrams (Sequence, Flowchart and Mermaid)

Typora supports, sequence, flowchart and mermaid, after this feature is enabled from preference panel.

See this document for detail.

Span Elements

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.

Links

Markdown supports two style 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:

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

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

will produce:

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

This link 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:

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

Then, anywhere in the document, you define your link label like this, on a line by itself:

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

In typora, they will be rendered like:

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:

```
[Google][]
And then define the link:

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

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

URLs

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

`<i@typora.io>` becomes i@typora.io.

Typora will aslo auto link standard URLs. e.g: www.google.com.

Images

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

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

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

You are able to use drag & drop to insert image from image file or we 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.

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

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:

```
*single asterisks*

_single underscores_
```

output:

single asterisks

single underscores

GFM will ignores 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:

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

Typora recommends to use `*` symbol.

Strong

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

```
**double asterisks**

__double underscores__
```

output:

double asterisks

double underscores

Typora recommends to use `**` symbol.

Code

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:

```
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.~~

Underline

Underline is powered by raw HTML.

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

Emoji 🗆

Input emoji with syntax `: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.

HTML

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

- Underline: `<u>underline</u>`
- Image: `<img src="http://www.w3.org/html/logo/img/mark-word-icon.png" width="200px" />` (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.

But those HTML will be exported on print or export.

Inline Math

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.

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

Writing $\LaTeX$: $

Subscript

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

Superscript

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

Highlight

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

---

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