

Markdown



Markdown is a plain text formatting syntax created by John Gruber, aiming to provide a easy-to-read and feasible markup. The original Markdown syntax specification can be found [here](#).

Scrivo can be used as editor for Markdown documents. It can render your Markdown contents into HTML (or PDF, ePub, etc.), and display them in a preview window.

Scrivo supports all the original Markdown syntaxes. But it can do so much more!

The Basics

Before we tell you about all the extra syntaxes and capabilities Scrivo have, we'll introduce you to the basics of standard Markdown. Lets jump right in.

Line Breaks

To force a line break, put two spaces and a newline (return) at the end of the line.

```
These lines  
won't break  
  
These lines  
will break
```

Strong and Emphasize

Strong: ****Strong**** or **__Strong__** (Command-B)

Emphasize: **Emphasize** or *_Emphasize_* 1 (Command-I)

Headers (like this one!)

```
Header 1  
=====
```

```
Header 2
```

```
-----
```

OR

```
# Header 1
## Header 2
### Header 3
#### Header 4
##### Header 5
##### Header 6
```

Links and Email

Inline

Just put angle brackets around an email and it becomes clickable: contact@studio5apps.com
<contact@studio5apps.com>

Same thing with urls: <http://www.studio5apps.com>
<http://www.studio5apps.com>

Perhaps you want to some link text like this: [Studio 5 Website](http://www.studio5apps.com)
[Scrivo Website](http://www.studio5apps.com "Title") (The title is optional)

Reference style

Sometimes it looks too messy to include big long urls inline, or you want to keep all your urls together.

Make [a link](#) [a link][arbitrary_id] then on it's own line anywhere else in the file:
[arbitrary_id]: http://www.studio5apps.com "Title"

If the link text itself would make a good id, you can link [like this](#) [like this][], then on it's own line anywhere else in the file:
[like this]: http://www.studio5apps.com

Images

Inline

```
![Alt Image Text](path/or/url/to.jpg "Optional Title")
```

Reference style

`![Alt Image Text][image-id]`
on it's own line elsewhere:
`[image-id]: path/or/url/to.jpg "Optional Title"`

Lists

- Lists must be preceded by a blank line (or block element)
- Unordered lists start each item with a *
- - works too
 - Indent a level to make a nested list
 1. Ordered lists are supported.
 2. Start each item (number-period-space) like 1.
 3. It doesn't matter what number you use, Scrivo will render them sequentially
 4. So you might want to start each line with 1. and let Scrivo sort it out

Here is the code:

```
* Lists must be preceded by a blank line (or block element)
* Unordered lists start each item with a `*`
- `-` works too
  * Indent a level to make a nested list
    1. Ordered lists are supported.
    2. Start each item (number-period-space) like `1.`
    42. It doesn't matter what number you use, Scrivo will render them sequentially
    1. So you might want to start each line with `1.` and let Scrivo sort it out
```

Block Quote

Angle brackets > are used for block quotes.
Technically not every line needs to start with a > as long as there are no empty lines between paragraphs.
Looks kinda ugly though.

Block quotes can be nested.

Multiple Levels

Most markdown syntaxes work inside block quotes.

- Lists
- [Links](#)
- Etc.

Here is the code:

```
> Angle brackets `>` are used for block quotes.  
Technically not every line needs to start with a `>` as long as  
there are no empty lines between paragraphs.  
> Looks kinda ugly though.  
> > Block quotes can be nested.  
> > > Multiple Levels  
>  
> Most markdown syntaxes work inside block quotes.  
>  
> * Lists  
> * [Links][arbitrary_id]  
> * Etc.
```

Inline Code

Inline code is indicated by surrounding it with backticks:
``Inline code``

If your code has ``backticks`` that need to be displayed, you can use double backticks:
```Code with `backticks` ``` (mind the spaces preceding the final set of backticks)

## Block Code

If you indent at least four spaces or one tab, Scrivo'll display a code block.

```
print('This is a code block')
print('The block must be preceded by a blank line')
print('Then indent at least 4 spaces or 1 tab')
 print('Nesting does nothing. Your code is displayed Literally')
```

Scrivo also know how to do something called [Fenced Code Blocks](#) which we will tell you about later.

## Horizontal Rules

If you type three asterisks `***` or three dashes `---` on a line, I'll display a horizontal rule:

---

## Document Formatting

The *Smartyants* extension automatically transforms straight quotes ( " and ' ) in your text into typographer's quotes ( “, ”, ‘, and ’ ) according to the context. Very useful if you're a typography freak like we are. Quote and Smartyants are syntactically incompatible. If both are enabled, Quote takes

precedence.

## Block Formatting

### Table

This is a table:

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

You can align cell contents with syntax like this:

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

The left- and right-most pipes (|) are only aesthetic, and can be omitted. The spaces don't matter, either. Alignment depends solely on : marks.

### Fenced Code Block

This is a fenced code block:

```
print('Hello world!')
```

You can also use waves (~) instead of back ticks (`):

```
print('Hello world!')
```

## Inline Formatting

The following is a list of special inline markups supported:

Option name	Markup	Result if enabled
Intra-word emphasis	So A*maz*ing	So <i>Amazing</i>
Strikethrough	~~Much wow~~	<del>Much wow</del>
Underline <sup>2</sup>	_So doge_	<u>So doge</u>
Quote <sup>3</sup>	“Such editor”	"Such editor"
Highlight	==So good==	<b>So good</b>
Superscript	hoge^(fuga)	hoge <sup>fuga</sup>
Autolink	<a href="http://t.co">http://t.co</a>	<a href="http://t.co">http://t.co</a>
Footnotes	[^4] and [^4]:	<sup>4</sup> and footnote 4

## The Rendering Preference Pane

### MathML Math Syntax

Scrivo can also render MathML math syntaxes.<sup>5</sup> For example, using the block math:

$$A_S^T = B$$
$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}.$$

### Task List Syntax

1. [x] Scrivo can render checkbox list syntax
  - [x] Scrivo support nesting
  - [x] Scrivo support ordered *and* unordered lists
2. [ ] Scrivo don't support clicking checkboxes directly in the html window

### Jekyll front-matter

If you like, Scrivo can display Jekyll front-matter in a nice table. Just make sure you put the front-matter at the very beginning of the file, and fence it with `---`. For example:

```

title: "Scrivo is my friend"
date: 2016-06-06 20:00:00

```

## Happy Markdown writing!

---

1. `_this notation_` will render as underlined instead of emphasized 
2. If **Underline** is disabled `_this_` will be rendered as *emphasized* instead of being underlined. 
3. **Quote** replaces literal " characters with html `<q>` tags. **Quote** and **Smarty pants** are syntactically incompatible. If both are enabled, **Quote** takes precedence. Note that **Quote** is different from *blockquote*, which is part of standard Markdown. 
4. You don't have to use a number. Arbitrary things like `[^footy note4]` and `[^footy note4]:` will also work. But they will *render* as numbered footnotes. Also, no need to keep your footnotes in order, Scrivo will sort out the order for you so they appear in the same order they were referenced in the text body. You can even keep some footnotes near where you referenced them, and collect others at the bottom of the file in the traditional place for footnotes. 
5. Internet connection required. 