

Basic Markdown Syntax

Dillon included in [Markdown](#)
 01.12.2019 2783 words 14 minutes



CONTENTS

- 1 Headings
- 2 Comments
- 3 Horizontal Rules
- 4 Body Copy
- 5 Inline HTML
- 6 Emphasis
- 7 Blockquotes
- 8 Lists
- 9 Code
- 10 Tables
- 11 Links
- 12 Footnotes
- 13 Images

This article offers a sample of basic Markdown syntax that can be used in Hugo content files.

Note

This article is a shameful copy of the great [Grav original page](#).
 If you want to know about the extended Markdown syntax of [LoveIt theme](#), please read [extended Markdown syntax page](#).

Let's face it: Writing content for the Web is tiresome. WYSIWYG editors help alleviate this task, but they generally result in horrible code, or worse yet, ugly web pages.

Markdown is a better way to write [HTML](#), without all the complexities and ugliness that usually accompanies it.

Some of the key benefits are:

1. Markdown is simple to learn, with minimal extra characters, so it's also quicker to write content.
2. Less chance of errors when writing in Markdown.
3. Produces valid XHTML output.
4. Keeps the content and the visual display separate, so you cannot mess up the look of your site.
5. Write in any text editor or Markdown application you like.
6. Markdown is a joy to use!

John Gruber, the author of Markdown, puts it like this:

The overriding design goal for Markdown's formatting syntax is to make it as readable as possible. The idea is that a Markdown-formatted document should be publishable as-is, as plain text, without looking like it's been marked up with tags or formatting instructions. While Markdown's syntax has been influenced by several existing text-to-HTML filters, the single biggest source of inspiration for Markdown's syntax is the format of plain text email.

— John Gruber

Without further delay, let us go over the main elements of Markdown and what the resulting HTML looks like!

Tip

Bookmark this page for easy future reference!

1 Headings

Headings from `h2` through `h6` are constructed with a `#` for each level:

Markdown

- ```

1 ## h2 Heading
2 ### h3 Heading
3 #### h4 Heading
4 ##### h5 Heading
5 ##### h6 Heading

```

The HTML looks like this:

#### HTML

- ```

1 <h2>h2 Heading</h2>
2 <h3>h3 Heading</h3>
3 <h4>h4 Heading</h4>
4 <h5>h5 Heading</h5>
5 <h6>h6 Heading</h6>

```

Heading IDs

To add a custom heading ID, enclose the custom ID in curly braces on the same line as the heading:

Markdown

- ```
1 ## A Great Heading {#custom-id}
```

The HTML looks like this:

#### HTML

- ```
1 <h3 id="custom-id">A Great Heading</h3>
```

2 Comments

Comments should be HTML compatible.

HTML

- ```

1 <!--
2 This is a comment
3 -->

```

Comment below should NOT be seen:

## # 3 Horizontal Rules

The HTML `<hr>` element is for creating a "thematic break" between paragraph-level elements. In Markdown, you can create a `<hr>` with any of the following:

- \_\_\_ : three consecutive underscores
- --- : three consecutive dashes
- \*\*\* : three consecutive asterisks

The rendered output looks like this:

## # 4 Body Copy

Body copy written as normal, plain text will be wrapped with `<p></p>` tags in the rendered HTML.

So this body copy:

### Markdown

```
1 Lorem ipsum dolor sit amet, graecis denique ei vel, at duo primis mandamus. Et legere occur
2 animal tacimates complectitur ad cum. Cu eum inermis inimicus efficiendi. Labore officiis h
3 soluta officiis concludaturque ei qui, vide sensibus vim ad.
```

The HTML looks like this:

### HTML

```
1 <p>Lorem ipsum dolor sit amet, graecis denique ei vel, at duo primis mandamus. Et legere oc
```

A **line break** can be done with one blank line.

## # 5 Inline HTML

If you need a certain HTML tag (with a class) you can simply use HTML:

### HTML

```
1 Paragraph in Markdown.
2
3 <div class="class">
4 This is HTML
5 </div>
6
7 Paragraph in Markdown.
```

## # 6 Emphasis

### Bold

For emphasizing a snippet of text with a heavier font-weight.

The following snippet of text is **rendered as bold text**.

### Markdown

```
1 **rendered as bold text**
2 __rendered as bold text__
```

The HTML looks like this:

### HTML

```
1 rendered as bold text
```

### Italics

For emphasizing a snippet of text with italics.

The following snippet of text is *rendered as italicized text*.

### Markdown

```
1 *rendered as italicized text*
2 _rendered as italicized text_
```

The HTML looks like this:

### HTML

```
1 rendered as italicized text
```

### Strikethrough

In **GFM** you can do strikethroughs.

### Markdown

```
1 ~~Strike through this text.~~
```

The rendered output looks like this:

Strike through this text.

The HTML looks like this:

### HTML

```
1 Strike through this text.
```

### Combination

Bold, italics, and strikethrough can be used in combination.

### Markdown

```
1 ***bold and italics***
2 ~~*strikethrough and bold**~~
3 ~~*strikethrough and italics*~~
4 ~~~*bold, italics and strikethrough***~~
```

The rendered output looks like this:

**bold and italics**

~~strikethrough and bold~~

~~strikethrough and italics~~

~~bold, italics and strikethrough~~

The HTML looks like this:

### HTML

```
1 bold and italics
2 strikethrough and bold
3 strikethrough and italics
4 bold, italics and strikethrough
```

## # 7 Blockquotes

For quoting blocks of content from another source within your document.

Add `>` before any text you want to quote:

### Markdown

```
1 > **Fusion Drive** combines a hard drive with a flash storage (solid-state drive) and prese
```

The rendered output looks like this:

Fusion Drive combines a hard drive with a flash storage (solid-state drive) and presents it as a single logical volume with the space of both drives combined.

The HTML looks like this:

▼ HTML

```
1 <blockquote>
2 <p>
3 Fusion Drive combines a hard drive with a flash storage (solid-state d
4 </p>
5 </blockquote>
```

Blockquotes can also be nested:

▼ Markdown

```
1 > Donec massa lacus, ultricies a ullamcorper in, fermentum sed augue.
2 Nunc augue augue, aliquam non hendrerit ac, commodo vel nisi.
3 >> Sed adipiscing elit vitae augue consectetur a gravida nunc vehicula. Donec auctor
4 odio non est accumsan facilisis. Aliquam id turpis in dolor tincidunt mollis ac eu diam.
```

The rendered output looks like this:

```
Donec massa lacus, ultricies a ullamcorper in, fermentum sed augue. Nunc augue augue, aliquam non
hendrerit ac, commodo vel nisi.

Sed adipiscing elit vitae augue consectetur a gravida nunc vehicula. Donec auctor odio non est
accumsan facilisis. Aliquam id turpis in dolor tincidunt mollis ac eu diam.
```

## # 8 Lists

### Unordered

A list of items in which the order of the items does not explicitly matter.

You may use any of the following symbols to denote bullets for each list item:

▼ Markdown

```
1 * valid bullet
2 - valid bullet
3 + valid bullet
```

For example:

► Markdown

The rendered output looks like this:

- Lorem ipsum dolor sit amet
- Consectetur adipiscing elit
- Integer molestie lorem at massa
- Facilisis in pretium nisl aliquet
- Nulla volutpat aliquam velit
  - Phasellus aculis neque
  - Purus sodales ultricies
  - Vestibulum laoreet porttitor sem
  - Ac tristique libero volutpat at
- Faucibus porta lacus fringilla vel
- Aenean sit amet erat nunc
- Eget porttitor lorem

The HTML looks like this:

► HTML

### Ordered

A list of items in which the order of items does explicitly matter.

▼ Markdown

```
1 1. Lorem ipsum dolor sit amet
2 2. Consectetur adipiscing elit
3 3. Integer molestie lorem at massa
4 4. Facilisis in pretium nisl aliquet
5 5. Nulla volutpat aliquam velit
6 6. Faucibus porta lacus fringilla vel
7 7. Aenean sit amet erat nunc
8 8. Eget porttitor lorem
```

The rendered output looks like this:

1. Lorem ipsum dolor sit amet
2. Consectetur adipiscing elit
3. Integer molestie lorem at massa
4. Facilisis in pretium nisl aliquet
5. Nulla volutpat aliquam velit
6. Faucibus porta lacus fringilla vel
7. Aenean sit amet erat nunc
8. Eget porttitor lorem

The HTML looks like this:

▼ HTML

```
1
2 Lorem ipsum dolor sit amet
3 Consectetur adipiscing elit
4 Integer molestie lorem at massa
5 Facilisis in pretium nisl aliquet
6 Nulla volutpat aliquam velit
7 Faucibus porta lacus fringilla vel
8 Aenean sit amet erat nunc
9 Eget porttitor lorem
10
```

Tip

If you just use `1.` for each number, Markdown will automatically number each item. For example:

▼ Markdown

```
1 1. Lorem ipsum dolor sit amet
2 1. Consectetur adipiscing elit
3 1. Integer molestie lorem at massa
4 1. Facilisis in pretium nisl aliquet
5 1. Nulla volutpat aliquam velit
6 1. Faucibus porta lacus fringilla vel
7 1. Aenean sit amet erat nunc
8 1. Eget porttitor lorem
```

The rendered output looks like this:

1. Lorem ipsum dolor sit amet
2. Consectetur adipiscing elit
3. Integer molestie lorem at massa
4. Facilisis in pretium nisl aliquet
5. Nulla volutpat aliquam velit

```
5. Nulla volutpat aliquam vel
6. Faucibus porta lacus fringilla vel
7. Aenean sit amet erat nunc
8. Eget porttitor lorem
```

## Task Lists

Task lists allow you to create a list of items with checkboxes. To create a task list, add dashes ( - ) and brackets with a space ( [ ] ) before task list items. To select a checkbox, add an x in between the brackets ( [x] ).

### Markdown

```
1 - [x] Write the press release
2 - [] Update the website
3 - [] Contact the media
```

The rendered output looks like this:

- Write the press release
- Update the website
- Contact the media

## # 9 Code

### Inline Code

Wrap inline snippets of code with ` .

### Markdown

```
1 In this example, '<section>/<section>' should be wrapped as **code**.
```

The rendered output looks like this:

In this example, `<section>/<section>` should be wrapped as **code**.

The HTML looks like this:

### HTML

```
1 <p>
2 In this example, <code><section></section></code> should be wrapped with <str
3 </p>
```

### Indented Code

Or indent several lines of code by at least four spaces, as in:

### Markdown

```
1 // Some comments
2 Line 1 of code
3 Line 2 of code
4 Line 3 of code
```

The rendered output looks like this:

```
// Some comments
Line 1 of code
Line 2 of code
Line 3 of code
```

The HTML looks like this:

### HTML

```
1 <pre>
2 <code>
3 // Some comments
4 Line 1 of code
5 Line 2 of code
6 Line 3 of code
7 </code>
8 </pre>
```

## Block Fenced Code

Use “fences” ` ` ` to block in multiple lines of code with a language attribute.

### Markdown

```
1 ````markdown
2 Sample text here...
3 ````
```

The HTML looks like this:

### HTML

```
1 <pre language="html">
2 <code>Sample text here...</code>
3 </pre>
```

## Syntax Highlighting

GitHub Flavored Markdown

GFM also supports syntax highlighting.

To activate it, simply add the file extension of the language you want to use directly after the first code “fence”, ` ` `js , and syntax highlighting will automatically be applied in the rendered HTML.

For example, to apply syntax highlighting to JavaScript code:

### Markdown

The rendered output looks like this:

### JavaScript

### Note

Syntax highlighting page in Hugo Docs introduces more about syntax highlighting, including highlight shortcode.

## # 10 Tables

Tables are created by adding pipes as dividers between each cell, and by adding a line of dashes (also separated by bars) beneath the header. Note that the pipes do not need to be vertically aligned.

### Markdown

```
1 | Option | Description |
2 | ----- | -----
3 | data | path to data files to supply the data that will be passed into templates. |
4 | engine | engine used for processing templates. Handlebars is the default. |
5 | ext | extension to be used for dest files. |
```

The rendered output looks like this:

| Option | Description                                                               |
|--------|---------------------------------------------------------------------------|
| data   | path to data files to supply the data that will be passed into templates. |

|        |                                                                        |
|--------|------------------------------------------------------------------------|
| engine | engine to be used for processing templates. Handlebars is the default. |
| ext    | extension to be used for dest files.                                   |

The HTML looks like this:

> **HTML**

**Right or center aligned text**

Adding a colon on the right side of the dashes below any heading will right align text for that column.  
Adding colons on both sides of the dashes below any heading will center align text for that column.

Markdown

```
1 | option | Description |
2 |-----| -----
3 | data | path to data files to supply the data that will be passed into templates. |
4 | engine | engine to be used for processing templates. Handlebars is the default. |
5 | ext | extension to be used for dest files. |
```

The rendered output looks like this:

| Option | Description                                                               |
|--------|---------------------------------------------------------------------------|
| data   | path to data files to supply the data that will be passed into templates. |
| engine | engine to be used for processing templates. Handlebars is the default.    |
| ext    | extension to be used for dest files.                                      |

## # 11 Links

### | Basic Link

Markdown

```
1 <https://assemble.iocontact@revolunet.comhttps://assemble.io)
```

The rendered output looks like this (hover over the link; there is no tooltip):

<https://assemble.io>  
[contact@revolunet.com](mailto:contact@revolunet.com)  
Assemble

The HTML looks like this:

Markdown

```
1 https://assemble.io
2 contact@revolunet.com
3 Assemble
```

### | Add a Title

Markdown

```
1 [Upstage] (https://github.com/upstage/ "Visit Upstage!")
```

The rendered output looks like this (hover over the link; there should be a tooltip):

Upstage

The HTML looks like this:

Markdown

```
1 Upstage
```

### | Named Anchors

Named anchors enable you to jump to the specified anchor point on the same page. For example, each of these chapters:

Markdown

```
1 ## Table of Contents
2 * [Chapter 1](#chapter-1)
3 * [Chapter 2](#chapter-2)
4 * [Chapter 3](#chapter-3)
```

will jump to these sections:

Markdown

```
1 ## Chapter 1
2 Content for chapter one.
3
4 ## Chapter 2
5 Content for chapter one.
6
7 ## Chapter 3
8 Content for chapter one.
```

**Note**

The specific placement of the anchor tag seems to be arbitrary. They are placed inline here since it seems to be unobtrusive, and it works.

## # 12 Footnotes

Footnotes allow you to add notes and references without cluttering the body of the document. When you create a footnote, a superscript number with a link appears where you added the footnote reference. Readers can click the link to jump to the content of the footnote at the bottom of the page.

To create a footnote reference, add a caret and an identifier inside brackets ( <sup>[^1]</sup> ). Identifiers can be numbers or words, but they can't contain spaces or tabs. Identifiers only correlate the footnote reference with the footnote itself — in the output, footnotes are numbered sequentially.

Add the footnote using another caret and number inside brackets with a colon and text ( <sup>[^1]: My footnote</sup> ). You don't have to put footnotes at the end of the document. You can put them anywhere except inside other elements like lists, block quotes, and tables.

Markdown

```
1 This is a digital footnote[^1].
2 This is a footnote with "label"[^label]
3
4 [^1]: This is a digital footnote
5 [^label]: This is a footnote with "label"
```

This is a digital footnote<sup>1</sup>.

This is a footnote with "label"<sup>2</sup>

## # 13 Images

Images have a similar syntax to links but include a preceding exclamation point.

▼ Markdown

1 ! [Minion] (<https://octodex.github.com/images/minion.png>)



or:

▼ Markdown

1 ! [Alt text] (<https://octodex.github.com/images/stormtroopocat.jpg> "The Stormtroopocat")

