

Writing MkDocs. Markdown

On line links

Writing your docs

Markdown: Syntax

Paragraphs and Line breaks

A paragraph is simply one or more consecutive lines of text, separated by one or more blank lines. (A blank line is any line that looks like a blank line — a line containing nothing but spaces or tabs is considered blank.) Normal paragraphs should not be indented with spaces or tabs.

When you do want to insert a `
` break tag using Markdown, you end a line with two or more spaces, then type return.

Headers

Setext-style headers are “underlined” using equal signs (for first-level headers) and dashes (for second-level headers). For example:

```
This is an H1
=====

This is an H2
-----
```

Any number of underlining =’s or -’s will work.

Atx-style 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
```

New line

New line with two or more spaces in end of line or tag `
`.

```
One line' '  
Two line <br/>  
Three line
```

One line
Two line
Three line

Emphasis

Markdown treats asterisks (`*`) and underscores (`_`) as indicators of emphasis. Text wrapped with one `*` or `_` will be wrapped with an HTML `` tag; double `'`s or `_`'s will be wrapped with an HTML `` tag. E.g., this input:

```
*single asterisks*  
**double asterisks**
```

single asterisks

single underscores

double asterisks

double underscores

Emphasis can be used in the middle of a word:

```
un**frigging**believable
```

un**frigging**believable

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\*
```

this text is surrounded by literal asterisks

Code (text as is)

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.
```

Use the `printf()` function.

To include a literal backtick character within a code span, you can use multiple backticks as the opening and closing delimiters:

```
``There is a literal backtick (') here.``
```

```
There is a literal backtick (') here.
```

A backtick-delimited string in a code span: ``foo``

```
`` `foo` ``
```

Code blocks

One level of indentation — 4 spaces or 1 tab — is removed from each line of the code block. For example, this:

Here is an example of AppleScript:

```
tell application "Foo"
  beep
end tell
```

Fenced code blocks

The fenced code blocks extension adds an alternate method of defining code blocks without indentation.

The first line should contain 3 or more backtick (‘) characters, and the last line should contain the same number of backtick characters (‘):

```
Fenced code blocks are like Standard
Markdown’s regular code blocks, except that
they’re not indented and instead rely on
start and end fence lines to delimit the
code block.
```

Horizontal rules

Each of the following lines will produce a horizontal rule tag `<hr/>` :

```
* * *  
***  
*****  
- - -  
-----
```

Comments in md-files

```
[comment]: <> (This is a comment, it will not be included)  
[comment]: <> (in the output file unless you use it in)  
[comment]: <> (a reference style link.)
```

Or you could go further:

```
[//]: <> (This is also a comment.)
```

To improve platform compatibility (and to save one keystroke) it is also possible to use # (which is a legitimate hyperlink target) instead of <>:

```
[//]: # (This may be the most platform independent comment)
```

Span element Links

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

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

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

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

See my [About](../about/) page for details.

See my [About](#) page for details.

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"
```

Example: I get 10 times more traffic from [Google] [1] than from [Yahoo] [2] or [MSN] [3].

```
[1]: http://google.com/      "Google"
[2]: http://search.yahoo.com/ "Yahoo Search"
[3]: http://search.msn.com/  "MSN Search"
```

I get 10 times more traffic from [Google](#) than from [Yahoo](#) or [MSN](#).

Automatic links

Markdown supports a shortcut style for creating “automatic” links for URLs and email addresses: simply surround the URL or email address with angle brackets. What this means is that if you want to show the actual text of a URL or email address, and also have it be a clickable link, you can do this:

```
<http://example.com/>
<address@example.com>
```

In text link: <http://example.com/>

In text e-mail: address@example.com

Internal links

MkDocs allows you to interlink your documentation by using regular Markdown links. However, there are a few additional benefits to formatting those links specifically for MkDocs as outlined below.

Linking to pages

```
Please see the [about](../about.md) for further details.
```

Please see the [about](#) for further details.

Link to a section within a target document by using an anchor link.

```
[Commands](../10_start/install.md#Commands)
```

Please see the for use MkDocs [Commands](#).

Inline HTML

Markdown’s syntax is intended for one purpose: to be used as a format for writing for the web.

The only restrictions are that block-level HTML elements — e.g. `<div>`, `<table>`, `<pre>`, `<p>`, etc. — must be separated from surrounding content by blank lines, and the start and end tags of the block should not be indented with tabs or spaces. Markdown is smart enough not to add extra (unwanted) `<p>` tags around HTML block-level tags.

For example, to add an HTML table to a Markdown article:

```
<h2>Header h2</h2>
```

This is a table:

```
<table>
<thead>
  <tr>
    <th colspan="2">The table header</th>
  </tr>
</thead>
<tbody>
  <tr>
    <td>The table body</td>
    <td>with two columns</td>
  </tr>
</tbody>
</table>
```

The table header	
The table body	with two columns

This is another regular paragraph.

Blockquotes

```
> This is a blockquote with two paragraphs. Lorem ipsum dolor sit amet,
> consectetur adipiscing elit. Aliquam hendrerit mi posuere lectus.
> Vestibulum enim wisi, viverra nec, fringilla in, laoreet vitae, risus.
>
> Donec sit amet nisl. Aliquam semper ipsum sit amet velit. Suspendisse
> id sem consectetur libero luctus adipiscing.
```

or

```
> This is a blockquote with two paragraphs. Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Aliquam hendrerit mi posuere lectus.
Vestibulum enim wisi, viverra nec, fringilla in, laoreet vitae, risus.

> Donec sit amet nisl. Aliquam semper ipsum sit amet velit. Suspendisse
id sem consectetur libero luctus adipiscing.
```

This is a blockquote with two paragraphs. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam hendrerit mi posuere lectus. Vestibulum enim wisi, viverra nec, fringilla in, laoreet vitae, risus.

Donec sit amet nisl. Aliquam semper ipsum sit amet velit. Suspendisse id sem consectetur libero luctus adipiscing.

Nested:

```
> This is the first level of quoting.  
>  
> > This is nested blockquote.  
>  
> Back to the first level.
```

This is the first level of quoting.

This is nested blockquote.

Back to the first level.

Blockquotes can contain other Markdown elements, including headers, lists, and code blocks:

```
> ### Blockquotes header.  
>  
> 1. This is the first list item.  
> 2. This is the second list item.  
>  
> Here's some example code:  
>  
>   return shell_exec("echo $input | $markdown_script");
```

Blockquotes header

1. This is the first list item.
2. This is the second list item.

Here's some example code:

```
return shell_exec("echo $input | $markdown_script");
```

Lists

Markdown supports ordered (numbered) and unordered (bulleted) lists.

Unordered lists use asterisks, pluses, and hyphens — interchangeably — as list markers:

- * Red
- * Green
- * Blue

is equivalent to:

- + Red
- + Green
- + Blue

and:

- Red
- Green
- Blue

- Red
- Green
- Blue

Ordered lists use numbers followed by periods:

1. Bird
2. McHale
3. Parish

1. Bird
2. McHale
3. Parish

It's important to note that the actual numbers you use to mark the list have no effect on the HTML output Markdown produces. The HTML Markdown produces from the above list is:

```
<ol> <li>Bird</li>
<li>McHale</li>
<li>Parish</li>
</ol>
```

If you instead wrote the list in Markdown like this:

1. Bird
1. McHale
1. Parish

or even:

3. Bird
1. McHale
8. Parish

you'd get the exact same HTML output.

Images

Inline image syntax looks like this:

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



Reference-style image syntax looks like this:

```
![Alt text][id]
```

Where “id” is the name of a defined image reference. Image references are defined using syntax identical to link references:

```
[id]: url/to/image "Optional title attribute"
```



BackSlash escapes

Markdown allows you to use backslash escapes to generate literal characters which would otherwise have special meaning in Markdown's formatting syntax. For example, if you wanted to surround a word with literal asterisks (instead of an HTML `` tag), you can use backslashes before the asterisks, like this:

```
\*literal asterisks\*
```

Markdown provides backslash escapes for the following characters:

```
\ backslash  
` backtick  
* asterisk  
_ underscore  
{ curly braces
```

[] square brackets
() parentheses
hash mark
+ plus sign
- minus sign (hyphen)
. dot
! exclamation mark