

QuickDocs | Markdown

January 2020

- 1. Overview / Cheat Sheet
 - 1.1. Basic Syntax
 - 1.2. Extended Syntax
- 2. Block Elements
 - 2.1. Headers
 - 2.1.1. atx Headers
 - 2.1.2. Setext Headers
 - 2.1.3. Defining Heading IDs
 - 2.1.4. Linking to Heading IDs
 - 2.2. Paragraphs
 - 2.3. Blockquotes
 - 2.4. Lists
 - 2.4.1. Unordered Lists
 - 2.4.2. Ordered Lists
 - 2.4.3. Definition Lists
 - 2.4.4. Task Lists
 - 2.5. Horizontal Rules
 - 2.6. Code Spans & Blocks
 - 2.6.1. Code Blocks
 - 2.6.2. Code Spans
 - 2.7. Inline/Embed HTML
- 3. Span Elements
 - 3.1. Links & References
 - 3.1.1. Inline Links
 - 3.1.2. Reference Links
 - 3.1.3. Automatic Links
 - 3.2. Emphasis
 - 3.2.1. Bold
 - 3.2.2. Italic
 - 3.2.3. Bold & Italic
 - 3.2.4. Strikethrough (Extended Syntax)
 - 3.3. Images
 - 3.4. Tables
 - 3.4.1. Alignment
 - 3.5. Footnotes
 - 3.6. Latex
 - 3.7. Escape Characters
 - 3.7.1. Automated Escaping
 - 3.7.2. Backslash Escapes
 - 3.8. Great Resources

1. Overview / Cheat Sheet

[Search Tags: >overview >cheat >sheet >cheatsheet >quick >revise >revision]

1.1. Basic Syntax

[Search Tags: >basicsyntax >syntax.basic>]

All Markdown applications support these elements.

Element	Markdown Syntax
Heading	# H1
	## H2
	### H3
Bold	**bold text**
Italic	<i>*italicized text*</i>
Blockquote	> blockquote
Ordered List	1. First item
	2. Second item
	3. Third item
Unordered List	- First item
	* Second item
	+ Third item
Code	`code`
Horizontal Rule	----
Link	[title](https://www.example.com)
Image	![alt text](image.jpg)

1.2. Extended Syntax

[Search Tags: >extendedsyntax >syntax.extended>]

These elements extend the basic syntax by adding additional features. Not all Markdown applications support these elements.

Element	Markdown Syntax
Table	Syntax Description
	----- -----
	Header Title
	Paragraph Text

Element	Markdown Syntax
Fenced Code Block	<pre>``` { "firstName": "John", "lastName": "Smith", "age": 25 } ```</pre>
Footnote	<p>Here's a sentence with a footnote. ^[1]</p> <p>^[1]: This is the footnote.</p>
Heading ID	<h3>My Great Heading {#custom-id}</h3>
Definition List	<p>term : definition</p>
Strikethrough	<p>~~The world is flat.~~</p>
Task List	<ul style="list-style-type: none"><input checked="" type="checkbox"/> Write the press release<input type="checkbox"/> Update the website<input type="checkbox"/> Contact the media

2. Block Elements

[Search Tags: >part01 >part1 >chapter1 >chap1 >blockelements >elementsblock >elementblock >part.01 >part.1 >chapter.1 >chap.1 >block.elements >elements.block >element.block]

2.1. Headers

[Search Tags: >headers >hders >hdrs >titles]

Markdown supports two styles of headers, [Setext](#) and [atx](#).

2.1.1. atx Headers

[Search Tags: >headers.atx >header.atx >atxheaders >atx.headers >atxhders >atxhdrs]

- Atx-style headers use 1-6 hash characters at the start of the line, corresponding to header levels 1-6.

Example:

```
# H1
## H2
### H3
#### H4
##### H5
##### H6
```

[Note: Optionally, you may “close” atx-style headers --- this is purely cosmetic. For example:

```
# This is an H1 #
## This is an H2 ##
### This is an H3 ##### <!-- They don't need to match. -->
```

- \space end \space note]

2.1.2. Setext Headers

[Search Tags: >headers.setex >header.setex >setextheaders >setexthdrs >settexthdrs]

- Setext-style headers are “underlined” using any number of underlining equal signs =’s or dashes –’s.

Symbol	Level
Equal Sign (=)	1st Level
Dashes (–)	2nd Level

Example:

```
Title (Level 1)
=====

Title (Level 2)
-----
```

2.1.3. Defining Heading IDs

[Search Tags: >defineheaderids >defheadids >definitionheaderids >headers.ids >header.ids >idheaders >idsheaders >ids.headers >idhdrs >idhdrs >hdrids >headerids >headeridentity >hdrsids >ids >headingids >headingidentity >headids >headrids >headidentity]

- Many Markdown processors support custom IDs for headings — some Markdown processors automatically add them.
- Adding custom IDs **allows you to link directly to headings** and modify them with CSS.
- To add a custom heading ID, enclose the custom ID in curly braces on the same line as the heading, where a custom ID can be any combination of letters and digits.

Form:

```
### <Heading> {#<custom-id>}
```

HTML:

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

2.1.4. Linking to Heading IDs

[Search Tags: >linkhdrs >linkheaders >linkheads >linkheadrs >headlinks >headlinking]

- You can link to headings with the custom IDs that you've defined, in your Markdown file, for your headers.
- The syntax is the exact same as that of normal URL links, except that in the `<link>` portion, write the heading ID instead, prepended by a `#` (*hash*):

Markdown	HTML	Rendered Output
[Heading ID] (#heading-id)	<code>Heading ID</code>	Heading ID

- Other websites can link to the heading by adding the custom heading ID to the full URL of the webpage, e.g.:

```
[<linked-text>] (https://www.website.com/section#<heading-ID>)
```

2.2. Paragraphs

[Search Tags: >paragraphs >paras >text]

- A paragraph is simply one or more consecutive lines of text, separated by one or more blank lines.
- Normal paragraphs should not be indented with spaces or tabs.

Example:

```
This is a paragraph [...] that
has reached its end.
```

```
This is another paragraph that is
seperated by a line and has thatreached
its end as well.
```

2.3. Blockquotes

[Search Tags: >quoteblocks >quotes >blockquotes >quotations >nestedquotes >nestquotes]

- Blockquotes are nice looking quote formatting, it looks like this:

```
"quote"
```

... and it is made by prepending (source) text with a right-arrow (`>`):

```
> "quote"
```

- Blockquotes can be nested (i.e. a blockquote-in-a-blockquote):

```
note level 1
```

```
note level 2
```

```
note level 3
```

... by adding additional levels of right-arrows (>):

```
> note level 1
>> note level 2
>>> note level 3
```

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

This is a 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");
```

... made like this:

```
> ## This is a 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");
```

2.4. Lists

[Search Tags: >lists >enums >bulletpoints >checklists >dropdownlists]

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

2.4.1. Unordered Lists

[Search Tags: >list.unorder >list.unordered >lists.unorder >lists.unordered >unorder.lists >unordered.lists >unorderedlists >unorderlists >listunordered >ulists]

- They use asterisks (*), pluses (+), and hyphens (−), interchangeably, as list markers:

```
* Red
+ Green
− Blue
```

All these make bulleted (unordered) lists:

- Red
- Green
- Blue

2.4.2. Ordered Lists

[Search Tags: >list.order >list.ordered >lists.order >lists.ordered >order.lists >ordered.lists >orderedlists >orderlists >listordered >olists]

- They use numbers (1. 2. 3. ...) followed by periods:

```
1. Bird
2. McHale
3. Parish
```

Result:

1. Bird
2. McHale
3. Parish

[Note: The actual numbers you use to mark the list have no effect on the HTML output Markdown produces. - \space end \space note]

- Adding a separation line between any of the elements of a list will cause it to be separated by a wider vertical space.

```
1. Bird

2. McHale
3. Parish
```

Result:

1. Bird

2. McHale

3. Parish

- You can have *words, sentences, paragraphs, quotes, code spans/blocks, images, etc.*, as list items.

2.4.3. Definition Lists

[Search Tags: >list.defs >list.definitions >definitionlists >listdefinitions >deflists >listsdefs >deflists >listdefs]

[Note: - Not all Markdown processors allow you to create *definition lists*. - \space end \space note]

- To create a definition list, type the term on the first line. On the next line, type a colon followed by a space and the definition.

```
First Term
: This is the definition of the first term.

Second Term
: This is one definition of the second term.
: This is another definition of the second term.
```

The HTML looks like this:

```
<dl>
  <dt>First Term</dt>
  <dd>This is the definition of the first term.</dd>
  <dt>Second Term</dt>
  <dd>This is one definition of the second term. </dd>
  <dd>This is another definition of the second term.</dd>
</dl>
```

The rendered output looks like this:

First Term : This is the definition of the first term.

Second Term : This is one definition of the second term. : This is another definition of the second term.

2.4.4. Task Lists

[Search Tags: >list.tasks >tasklists >listtasks >tasklists >liststasks]

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

- [x] Write the press release
- [] Update the website
- [] Contact the media

The rendered output looks like this:

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

[Note: Not all Markdown processors support **task lists**. - \space end \space note]

2.5. Horizontal Rules

[Search Tags: >horizontalrules >rules >lines >straightlines >hlines >horilines >horizlines >linehorizontal >lineshorizontal >horizontalis >horzis]

- You can produce a line --- horizontal rule tag (`<hr />`) --- by placing three or more *hyphens* (`-`), *asterisks* (`*`), or *underscores* (`_`) on a line by themselves.

Example:

```
* * *
***
*****
- - -
-----
_____
```

Output:

```
=====
=====
=====
=====
=====
```

2.6. Code Spans & Blocks

[Search Tags: >code >ttfamily >typewriter >raw]

There are 2 methods to inserting code in your markdown. One is inlined, **code spans**, the other is not, **code blocks**.

2.6.1. Code Blocks

[Search Tags: >codeblocks >blockscodes >blockcode]

- Code blocks are (non-inlined) code environments.
- Code blocks are opened and closed with three backtick characters (`` ` ``) on separate lines.

- You can specify the particular language of the environment --- after the three opening backtick characters --- to enable syntax highlighting, specific to that language.

General Form:

```
```<language>
<code>
<code>
```
```

Example:

```
```python
print('Hello, world !')
print('and now ... Good-Bye, world !')
```
```

[Note:] It is also possible to create a code block by prepending each line of code with one level of indentation --- 4 spaces or a tab. Syntax highlighting is not supported with this; it is better to stick to the former way.

2.6.2. Code Spans

[Search Tags: >codespans >spanscodes >spancode >codesspans]

- Code spans are inlined code environments:
- You can make one by enclosing (source) text with backticks (`)

Some markdown `code-span`.

- If your code itself contains backticks, you may have to use multiple backticks as delimiters:

```
```The name ``Tuple`2`` is a valid .NET type name.```
```

--- his code span needed three backticks, on either sides, to be enclosed properly.

## 2.7. Inline/Embed HTML

[Search Tags: >inlinehtml >embed.html >embedhtml >htmlmdded >htmlembedding >htmlinline]

- You can write **HTML** inline in your markdown.
- 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.

- You can't use **Markdown-style** (e.g. *\*emphasis\**) inside an **HTML** block.

#### Example:

This is a regular paragraph.

```
<table>
 <tr>
 <td>Foo</td>
 </tr>
</table>
```

This is another regular paragraph.

## 3. Span Elements

---

[Search Tags: >part02 >part2 >chapter2 >chap2 >spanelements >elementsspans >elementspans  
>part.02 >part.2 >chapter.2 >chap.2 >span.elements >spans.elements >elements.spans  
>element.spans]

### 3.1. Links & References

[Search Tags: >links >linking >linkage >references >refs >urls >websites]

- Markdown supports two style of links: **inline** and **reference**
- In both styles, the **link text** is delimited by square brackets `[]`.

#### 3.1.1. Inline Links

[Search Tags: >inlinelinks >inlinelinking >inlinelinkage >inlinereferences >inlinerefs  
>link.inline >links.inline >linking.inline >linkage.inline >references.inline >refs.inline]

- 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.
- If you're referring to a local resource on the same server, you can use relative paths.
- To quickly turn a URL or email address into a link, enclose it in angle brackets. See {#Automatic Links}

#### Example:

1. This is [an example](http://example.com/ "Title") inline link.
2. [This link](http://example.net/) has no title attribute.
3. See my [About](/about/) page for details.

4. `<https://www.fakewebsite.org>` or `<fake@example.com>`

Output:

1. This is [an example](#) inline link.
2. [This link](#) has no title attribute.
3. See my [About](#) page for details.
4. <https://www.fakewebsite.org> or [fake@example.com](mailto:fake@example.com)

### 3.1.2. Reference Links

[Search Tags: `>referencelinks` `>referencelinking` `>referencelinkage` `>references` `>link.references` `>links.references` `>linking.references` `>linkage.references`]

- 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**, i.e assign to it a link.

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

- Link definition names (`id`) may consist of *letters*, *numbers*, *spaces*, and *punctuation* — but they are **not case sensitive**, i.e `a` `<==>` `A`.
- Explicit linking:**

```
[<Linked-Text>] [<Link-Name>]
...
[<Link-Name>]: <Link> "<Link-Info-Title>"
```

- Implicit linking:**

```
[<Linked-Text>] []
...
[<Linked-Text>]: <Link>
```

**Example:**

- Reference links in action:*

```
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"
```

- Implicit link name shortcut:

```
I get 10 times more traffic from [Google][] than from
[Yahoo][] or [MSN][].
```

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

### 3.1.3. Automatic Links

```
[Search Tags: >autolinks >automaticlinks >linksautomatic >linkautomatic >emails >linkemails
>emaillinks >link.emails >links.emails >emaillinkage >emailslinkage >emailreferences
>referenceemails]
```

- Markdown supports a shortcut style for creating “automatic” links for URLs and email addresses.
- To quickly turn a URL or email address into a link, enclose it in angle brackets:

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

Becomes:

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

## 3.2. Emphasis

```
[Search Tags: >fonts >fontstyles >emphasis >styles]
```

### 3.2.1. Bold

```
[Search Tags: >bold >strong]
```

Markdown	HTML	Rendered Output
I just love <b>bold text</b> .	I just love <b>bold text</b> .	I just love bold text.
I just love <u><b>bold text</b></u> .	I just love <b>bold text</b> .	I just love bold text.

Markdown	HTML	Rendered Output
Love**is**bold	Love <b>is</b> bold	Love <b>is</b> bold

3.2.2. Italic

[Search Tags: >italics]

Markdown	HTML	Rendered Output
Italicized text is the <i>*cat's meow*</i> .	Italicized text is the <i>cat's meow</i> .	Italicized text is the cat's meow.
Italicized text is the <i>_cat's meow_</i> .	Italicized text is the <i>cat's meow</i> .	Italicized text is the cat's meow.
A <i>*cat*meow</i>	A <i>cat</i> meow	Acatmeow

3.2.3. Bold & Italic

Markdown	HTML	Rendered Output
This text is <i>***really important***</i> .	This text is <b><i>really important</i></b> .	This text is really important.
This text is <i>__really important__</i> .	This text is <b><i>really important</i></b> .	This text is really important.
This text is <i>__*really important*__</i> .	This text is <b><i>really important</i></b> .	This text is really important.
This text is <i>**_really important_**</i> .	This text is <b><i>really important</i></b> .	This text is really important.

3.2.4. Strikethrough (Extended Syntax)

[Search Tags: >strikethrough >strikes]

**Example:** *~~The world is flat.~~*

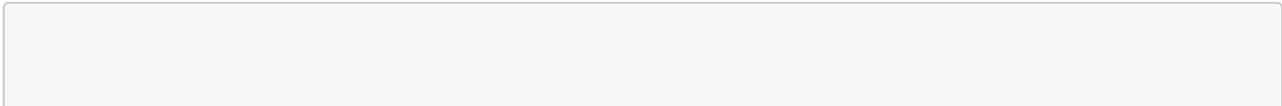
**Output:** ~~The world is flat.~~

3.3. Images

[Search Tags: >images >graphics >pics >pictures >photos >pasteimage >imagepaste >imagepasting]

- Inserting images into markdown is done in the exact same manner and syntax as links but with a preceding exclamation (!) mark:

**Inline:**



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

**Reference:**

```
! [Alt text] [id]
...
[id]: url/to/image "Optional title attribute"
```

- As of this writing, Markdown has no syntax for specifying the dimensions of an image; if this is important to you, you can simply use regular HTML `<img>` tags.

**Example:**

```
! [Lion | King of the Animal Kingdom] (Assets/lion.jpg)
```

**Rendered Output:**



### 3.4. Tables

[Search Tags: >tabls >tebles >tables >tabs >dictionaries >charts >grids]

- To add a table, use three or more hyphens (---) to create each column’s header, and use pipes (|) to separate each column. You can optionally add pipes on either end of the table.
- Cell widths can vary, as shown below. The rendered output will look the same.

**Example:**

Syntax	Description	
-----	-----	
Header	Title	
Paragraph	Text	



```
| Syntax | Description |
| --- | ----- |
| Header | Title |
| Paragraph | Text |
```

**Rendered Output:** (for both)

Syntax	Description
Header	Title
Paragraph	Text

3.4.1. Alignment

[Search Tags: >tabalignment >aligntables >aligntabs]

- You can align text in the columns to the left, right, or center by adding a colon (:) to the left, right, or on both side of the hyphens within the header row.

```
| Syntax | Description | Test Text |
| :--- | :----: | ----: |
| Header | Title | Here's this |
| Paragraph | Text | And more |
```

**Rendered Output:**

yntax	Description	Test Text
uper	Title	Here
ad	Text	And more

3.5. Footnotes

[Search Tags: >footnotes >notesfoot]

- 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**: `[^<footnote-id>]`, e.g.:

```
Here's a simple footnote,[^1] and here's a longer one.[^bignote]
```

- To create the **footnote**: `[^<footnote-id>]: <footnote>`

```
[^1]: This is a footnote example.
[^bignote]: Here's Here's another foot note example, one with multiple
paragraphs
 and code.

 Indent paragraphs to include them in the footnote.
 `{ my code }`
 Add as many paragraphs as you like.
```

- 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.
- 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.
- The rendered output looks like this:

Here's a simple footnote,[^1] and here's a longer one.[^bignote]

[^1]: This is a footnote example. [^bignote]: Here's another foot note example, one with multiple paragraphs and code.

```
 Indent paragraphs to include them in the footnote.
 `{ my code }`
 Add as many paragraphs as you like.
```

## 3.6. Latex

[Search Tags: >LaTeX]

[Note:\$ Not all Markdown processors support  $\LaTeX$ .  - \space end \space note]

- You can insert inline  $\LaTeX$  by enclosing (source) text in dollar signs:  $\text{<text>}$ .
- You can create a  $\LaTeX$  environment with two enclosing double dollar signs ( $\$$ ), e.g.:

```
$$
<latex>
$$
```

### Example:

Here's Newton's universal law of gravitation:

```
$$
F = GM \cdot m / r^{2}
```

\$\$

\$\$ ``If \space I \space have \space ever \space made \space any \space valuable \space discoveries, \space it \space has \space been \space owing \space more \space to \space patient \space attention, \space than \space to \space any \space other \space talent." \\ - Isaac \space Newton \$\$

### Rendered Output:

\$\$  $F = GM \cdot m / r^{\{2\}}$  \$\$

\$\$ ``If \space I \space have \space ever \space made \space any \space valuable \space discoveries, \space it \space has \space been \space owing \space more \space to \space patient \space attention, \space than \space to \space any \space other \space talent." \ - Isaac \space Newton \$\$

## 3.7. Escape Characters

[Search Tags: >escapecharacers >escapechars >escchars >esccharacters >& >< >escapesequences >escpcharacters >escpchars >escpechars]

### 3.7.1. Automated Escaping

[Search Tags: >autoescapes >autoescapechars >autoescapecharacters]

- In **HTML**, there are two characters that demand special treatment: `<` and `&`. Left angle brackets are used to start tags; ampersands are used to denote HTML entities. If you want to use them as entities, e.g. `&lt;`, and `&amp;`.
- **Markdown** on the other hand allows you to use these character naturally. It takes care of all the necessary escaping for you. If you use an ampersand as part of an **HTML** entity, it remains unchanged; otherwise it will be translated into `&amp;`.

### 3.7.2. Backslash Escapes

[Search Tags: >backslashcharacters >backslashescps >backslashescapecharacters >bkslhchars >backslashchars]

- Markdown allows you to use backslash escapes to generate literal characters which would otherwise have special meaning in Markdown's formatting syntax.
- 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

## 3.8. Great Resources

#	Type	Author	Link
1	Documentation	Markdown Author(s)	<a href="#">Official Markdown Documentation</a>
2	Guide	Markdownguide	<a href="#">Markdown Documentation</a>
3	Cheat Sheet	Markdownguide	<a href="#">Markdown Cheat Sheet</a>