

Markdown quick reference

Escaping

'&' and '<' are automatically escaped. So :

```
&copy;
```

will be left as is, but :

```
S&P 500
```

will be translated to :

```
S&amp;P 500
```

Similarly, inline HTML will not be escaped, while the following will :

```
1 < 2.
```

Generally speaking, Markdown is not meant to be a layer above HTML. So HTML is needed, use inline HTML.

Headers

The easiest possibility for header tags is as follows :

```
# My title
## My subtitle
### My sub-subtitle
#### You get the idea
```

Another, more readable possibility for <h1> and <h2> tags :

```
My title
=====

My subtitle
-----
```

Quoting

You like in text emails, quoting is achieved using "greater-than" signs :

```
> This is a quote
```

Quotes can be nested, as in :

```
> A quote
> > A subquote
> Back to the first level
```

Quotes can contain most other Markdown elements, including headers, lists and code blocks (see below).

Lists

Unordered lists are created by putting a plus, a minus or a star at the beginning of each line :

```
* One
* Two
* Three
```

or

```
- One
- Two
- Three
```

or

```
+ One
+ Two
+ Three
```

Ordered lists are created by putting a number followed by a dot at the beginning of each line :

```
1. One
2. Two
```

Note that the actual numbers that you use do not matter, so the example above would produce the exact same HTML code than the following :

```
29. One
2. Two
```

Which is :

```
<ol>
  <li>One</li>
  <li>Two</li>
</ol>
```

The numbers actually don't even need to be different from each other.

If you want each item to be wrapped in a <p> tag, just add a blank line between each line :

```
* One
* Two
```

This will produce :

```
<ul>
  <li><p>One</p></li>
  <li><p>Two</p></li>
</ul>
```

Each item can contain multiple paragraphs.

Ordered lists can inadvertently be triggered, if for some reason a line begins with a number followed by a dot. In such a case, one has to escape the dot using a backslash, as in :

```
1\. This is not a list.
```

Lists can contain a variety of other elements, including quotes and code blocks.

Code blocks

Code blocks are triggered by indenting text by four (or more spaces), or one tab.

```
Not code
Code
```

Means :

```
<p>Not code</p>
<pre><code>Code</code></pre>
```

In a code block, "&", "<" and ">" characters are automatically escaped. Also note that Markdown syntax is not processed in a code block.

Alternatively, a span of code is marked by surrounding some text with backticks, as in :

```
The main function of a C program is defined by
`int main(int argc, char **argv)`
```

Link

Here is the basic syntax for links :

```
Check out [thingylab](http://www.thingylab.net
"Thingylab") !
```

The text between angle brackets is the clickable link, then come between parenthesis the address and the title attribute. Note that the latter is optional. Relative addresses are allowed too.

One can also use another, more readable syntax :

```
Check out [thingylab][link], you will find a
Markdown reference there.
```

```
[link]: http://www.thingylab.net "Thingylab"
```

You can place the second part anywhere in the document, but the best place is probably just after each paragraph.

The link text can also be implicitly used as the reference :

```
Check out [thingylab][], you will find a Markdown
referende there.
```

```
[thingylab]: http://www.thingylab.net
```

Note that references are not case-sensitive, so the two following lines refer to the same address :

```
[link][a]
[link][A]
```

If you want to use an URL as the text for a link, you can just use the following syntax :

```
<http://www.google.com>
```

The same goes for email addresses.

Images

The syntax for images is similar to links :

```
![Alt text](http://www.site.com/img.jpg)
```

References are also admitted :

```
![Alt text][img]
Blah blah blah
```

```
[img]: http://www.site.com/img.jpg
```

 and tags

Emphasis is marked by surrounding text with "*", "_", "***" or "___" :

```
*emphasis*
_emphasis_
**more emphasis**
__more emphasis__
```

yields :

```
<em>emphasis</em>
<em>emphasis</em>
<strong>more emphasis</strong>
<strong>more emphasis</strong>
```

Those can also be escaped using a backslash.

Horizontal rules

If you really need those, it's easy, place three or more "*" or "-" on a line (without anything else).

Escaping

The following tags can be escaped using a backslash : \ ` * _ { } [] () # + - . !

Credits

This reference was compiled by Pierre DOUCY (pierre<at>thingylab.net). Don't hesitate to send me your remarks.