Markdown: Basics

Bernard Schmidt

Wednesday, September 17, 2014

Markdown: Basics

```
<li><a href="/projects/markdown/" title="Markdown Project Page">Main</a></li>
<li><a class="selected" title="Markdown Basics">Basics</a></li>
<li><a href="/projects/markdown/syntax" title="Markdown Syntax Documentation">Syntax</a></li>
<li><a href="/projects/markdown/license" title="Pricing and License Information">License</a></li>
<li><a href="/projects/markdown/dingus" title="Online Markdown Web Form">Dingus</a></li>
```

Getting the Gist of Markdown's Formatting Syntax

This page offers a brief overview of what it's like to use Markdown. The syntax page provides complete, detailed documentation for every feature, but Markdown should be very easy to pick up simply by looking at a few examples of it in action. The examples on this page are written in a before/after style, showing example syntax and the HTML output produced by Markdown.

It's also helpful to simply try Markdown out; the Dingus is a web application that allows you type your own Markdown-formatted text and translate it to XHTML.

Note: This document is itself written using Markdown; you can see the source for it by adding '.text' to the URL.

Paragraphs, Headers, Blockquotes

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.

Markdown offers two styles of headers: *Setext* and *atx*. Setext-style headers for `<h1>` and `<h2>` are created by "underlining" with equal signs (`=`) and hyphens (`-`), respectively. To create an atx-style header, you put 1-6 hash marks (`#`) at the beginning of the line – the number of hashes equals the resulting HTML header level.

Blockquotes are indicated using email-style '`>`' angle brackets.

Markdown:

```
A First Level Header
====================

A Second Level Header
---------------------

Now is the time for all good men to come to
the aid of their country. This is just a
regular paragraph.

The quick brown fox jumped over the lazy
dog's back.
```

```
### Header 3
```

```
> This is a blockquote.
>
> This is the second paragraph in the blockquote.
>
> ## This is an H2 in a blockquote
```

Output:

```
<h1>A First Level Header</h1>

<h2>A Second Level Header</h2>

<p>Now is the time for all good men to come to
the aid of their country. This is just a
regular paragraph.</p>

<p>The quick brown fox jumped over the lazy
dog's back.</p>

<h3>Header 3</h3>

<blockquote>
    <p>This is a blockquote.</p>

    <p>This is the second paragraph in the blockquote.</p>

    <h2>This is an H2 in a blockquote</h2>
</blockquote>
```

Phrase Emphasis

Markdown uses asterisks and underscores to indicate spans of emphasis.

Markdown:

```
Some of these words *are emphasized*.
Some of these words _are emphasized also_.

Use two asterisks for **strong emphasis**.
Or, if you prefer, __use two underscores instead__.
```

Output:

```
<p>Some of these words <em>are emphasized</em>.
Some of these words <em>are emphasized also</em>.</p>

<p>Use two asterisks for <strong>strong emphasis</strong>.
Or, if you prefer, <strong>use two underscores instead</strong>.</p>
```

Lists

Unordered (bulleted) lists use asterisks, pluses, and hyphens (`*`, `+`, and `-`) as list markers. These three markers are interchangable; this:

```
*    Candy.
*    Gum.
*    Booze.
```

this:

```
+    Candy.
+    Gum.
+    Booze.
```

and this:

```
-    Candy.
-    Gum.
-    Booze.
```

all produce the same output:

```
<ul>
<li>Candy.</li>
<li>Gum.</li>
<li>Booze.</li>
</ul>
```

Ordered (numbered) lists use regular numbers, followed by periods, as list markers:

```
1.   Red
2.   Green
3.   Blue
```

Output:

```
<ol>
<li>Red</li>
<li>Green</li>
<li>Blue</li>
</ol>
```

If you put blank lines between items, you'll get `<p>` tags for the list item text. You can create multi-paragraph list items by indenting the paragraphs by 4 spaces or 1 tab:

```
*    A list item.

     With multiple paragraphs.

*    Another item in the list.
```

Output:

```
<ul>
<li><p>A list item.</p>
<p>With multiple paragraphs.</p></li>
<li><p>Another item in the list.</p></li>
</ul>
```

Links

Markdown supports two styles for creating links: *inline* and *reference*. With both styles, you use square brackets to delimit the text you want to turn into a link.

Inline-style links use parentheses immediately after the link text. For example:

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

Output:

```
<p>This is an <a href="http://example.com/">
example link</a>.</p>
```

Optionally, you may include a title attribute in the parentheses:

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

Output:

```
<p>This is an <a href="http://example.com/" title="With a Title">
example link</a>.</p>
```

Reference-style links allow you to refer to your links by names, which you define elsewhere in your document:

```
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"
```

Output:

```
<p>I get 10 times more traffic from <a href="http://google.com/"
title="Google">Google</a> than from <a href="http://search.yahoo.com/"
title="Yahoo Search">Yahoo</a> or <a href="http://search.msn.com/"
title="MSN Search">MSN</a>.</p>
```

The title attribute is optional. Link names may contain letters, numbers and spaces, but are *not* case sensitive:

```
I start my morning with a cup of coffee and
[The New York Times][NY Times].
```

```
[ny times]: http://www.nytimes.com/
```

Output:

```
<p>I start my morning with a cup of coffee and
<a href="http://www.nytimes.com/">The New York Times</a>.</p>
```

Images

Image syntax is very much like link syntax.

Inline (titles are optional):

```
![alt text](/path/to/img.jpg "Title")
```

Reference-style:

```
![alt text][id]
```

```
[id]: /path/to/img.jpg "Title"
```

Both of the above examples produce the same output:

```
<img src="/path/to/img.jpg" alt="alt text" title="Title" />
```

Code

In a regular paragraph, you can create code span by wrapping text in backtick quotes. Any ampersands (&)
and angle brackets (< or >) will automatically be translated into HTML entities. This makes it easy to use
Markdown to write about HTML example code:

```
I strongly recommend against using any `<blink>` tags.
```

```
I wish SmartyPants used named entities like `&mdash;`
instead of decimal-encoded entites like `&#8212;`.
```

Output:

```
<p>I strongly recommend against using any
<code>&lt;blink&gt;</code> tags.</p>
```

```
<p>I wish SmartyPants used named entities like
<code>&amp;mdash;</code> instead of decimal-encoded
entites like <code>&amp;#8212;</code>.</p>
```

To specify an entire block of pre-formatted code, indent every line of the block by 4 spaces or 1 tab. Just like
with code spans, &, <, and > characters will be escaped automatically.

Markdown:

```
If you want your page to validate under XHTML 1.0 Strict,
you've got to put paragraph tags in your blockquotes:

    <blockquote>
        <p>For example.</p>
    </blockquote>
```

Output:

```
<p>If you want your page to validate under XHTML 1.0 Strict,
you've got to put paragraph tags in your blockquotes:</p>

<pre><code>&lt;blockquote&gt;
    &lt;p&gt;For example.&lt;/p&gt;
&lt;/blockquote&gt;
</code></pre>
```