Typography

Headings

Headings from `h1` through `h6` are constructed with a `#` for each level:

```
# h1 Heading
## h2 Heading
### h3 Heading
#### h4 Heading
##### h5 Heading
###### h6 Heading
```

Renders to:

h1 Heading

h2 Heading

h3 Heading

h4 Heading

h5 Heading

h6 Heading

HTML:

```
<h1>h1 Heading</h1>
<h2>h2 Heading</h2>
<h3>h3 Heading</h3>
<h4>h4 Heading</h4>
<h5>h5 Heading</h5>
<h6>h6 Heading</h6>
```

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

renders to:

Body Copy

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

So this body copy:

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

renders to this HTML:

```
<p>Lorem ipsum dolor sit amet, graecis denique ei vel, at duo primis mandamus. Et legere
ocurreret pri, animal tacimates complectitur ad cum. Cu eum inermis inimicus efficiendi.
Labore officiis his ex, soluta officiis concludaturque ei qui, vide sensibus vim ad.</p>
```

Emphasis

Bold

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

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

```
**rendered as bold text**
```

renders to:

rendered as bold text

and this HTML

```
<strong>rendered as bold text</strong>
```

Italics

For emphasizing a snippet of text with italics.

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

```
_rendered as italicized text_
```

renders to:

rendered as italicized text

and this HTML:

```
<em>rendered as italicized text</em>
```

strikethrough

In GFM you can do strickthroughs.

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

Which renders to:

~~Strike through this text.~~

Blockquotes

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

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

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

Renders to:

> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer posuere erat a ante.

and this HTML:

```
<blockquote>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer posuere erat a
ante.</p>
</blockquote>
```

Blockquotes can also be nested:

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

Renders to:

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

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:

```
* valid bullet
- valid bullet
+ valid bullet
```

For example

```
+ Lorem ipsum dolor sit amet
+ Consectetur adipiscing elit
+ Integer molestie lorem at massa
+ Facilisis in pretium nisl aliquet
+ Nulla volutpat aliquam velit
  - Phasellus iaculis 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
```

Renders to:

- Lorem ipsum dolor sit amet
- Consectetur adipiscing elit
- Integer molestie lorem at massa
- Facilisis in pretium nisl aliquet
- Nulla volutpat aliquam velit
  - Phasellus iaculis 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

And this HTML

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

Ordered

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

```
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
```

Renders to:

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

And this HTML:

```
<ol>
  <li>Lorem ipsum dolor sit amet</li>
  <li>Consectetur adipiscing elit</li>
  <li>Integer molestie lorem at massa</li>
  <li>Facilisis in pretium nisl aliquet</li>
  <li>Nulla volutpat aliquam velit</li>
  <li>Faucibus porta lacus fringilla vel</li>
  <li>Aenean sit amet erat nunc</li>
  <li>Eget porttitor lorem</li>
</ol>
```

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

```
1. Lorem ipsum dolor sit amet
1. Consectetur adipiscing elit
1. Integer molestie lorem at massa
1. Facilisis in pretium nisl aliquet
1. Nulla volutpat aliquam velit
1. Faucibus porta lacus fringilla vel
```

```
1. Aenean sit amet erat nunc
1. Eget porttitor lorem
```

Renders to:

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

Code

Inline code

Wrap inline snippets of code with `` ` ``.

For example, `<section></section>` should be wrapped as "inline".

```
For example, `<section></section>` should be wrapped as "inline".
```

Indented code

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

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

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

Block code "fences"

Use "fences" ```` ``` ```` to block in multiple lines of code.

~~~
``` html
Sample text here...
```
~~~

```
Sample text here...
```

HTML:

```
<pre>
  <p>Sample text here...</p>
</pre>
```

Syntax highlighting

GFM, or "GitHub Flavored Markdown" 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:

~~~
``` javascript
grunt.initConfig({
  assemble: {
    options: {
      assets: 'docs/assets',
      data: 'src/data/*.{json,yml}',
      helpers: 'src/custom-helpers.js',
      partials: ['src/partials/**/*.{hbs,md}']
    },
    pages: {
      options: {
        layout: 'default.hbs'
      },
      files: {
        './': ['src/templates/pages/index.hbs']
      }
    }
  }
};
```
~~~

Renders to:

```
grunt.initConfig({
  assemble: {
    options: {
```

```
      assets: 'docs/assets',
      data: 'src/data/*.{json,yml}',
      helpers: 'src/custom-helpers.js',
      partials: ['src/partials/**/*.{hbs,md}']
    },
    pages: {
      options: {
        layout: 'default.hbs'
      },
      files: {
        './': ['src/templates/pages/index.hbs']
      }
    }
  }
};
```

And this complicated HTML:

```
<div class="highlight"><pre><span class="nx">grunt</span><span class="p">.</span><span class="nx">initConfig</span><span class="p">({</span>
  <span class="nx">assemble</span><span class="o">:</span> <span class="p">{</span>
    <span class="nx">options</span><span class="o">:</span> <span class="p">{</span>
      <span class="nx">assets</span><span class="o">:</span> <span class="s1">'docs/assets'</span><span class="p">,</span>
      <span class="nx">data</span><span class="o">:</span> <span class="s1">'src/data/*.{json,yml}'</span><span class="p">,</span>
      <span class="nx">helpers</span><span class="o">:</span> <span class="s1">'src/custom-helpers.js'</span><span class="p">,</span>
      <span class="nx">partials</span><span class="o">:</span> <span class="p">[</span><span class="s1">'src/partials/**/*.{hbs,md}'</span><span class="p">]</span>
    <span class="p">},</span>
    <span class="nx">pages</span><span class="o">:</span> <span class="p">{</span>
      <span class="nx">options</span><span class="o">:</span> <span class="p">{</span>
        <span class="nx">layout</span><span class="o">:</span> <span class="s1">'default.hbs'</span>
      <span class="p">},</span>
      <span class="nx">files</span><span class="o">:</span> <span class="p">{</span>
        <span class="s1">'./'</span><span class="o">:</span> <span class="p">[</span><span class="s1">'src/templates/pages/index.hbs'</span><span class="p">]</span>
      <span class="p">}</span>
    <span class="p">}</span>
  <span class="p">}</span>
<span class="p">};</span>
</pre></div>
```

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.

```
| 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. |
```

Renders to:

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.

And this HTML:

```
<table>
  <tr>
    <th>Option</th>
    <th>Description</th>
  </tr>
  <tr>
    <td>data</td>
    <td>path to data files to supply the data that will be passed into templates.</td>
  </tr>
  <tr>
    <td>engine</td>
    <td>engine to be used for processing templates. Handlebars is the default.</td>
  </tr>
  <tr>
    <td>ext</td>
    <td>extension to be used for dest files.</td>
  </tr>
</table>
```

Right aligned text

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

```
| 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. |
```

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.

Links

Basic link

```
[Assemble](http://assemble.io)
```

Renders to (hover over the link, there is no tooltip):

Assemble

HTML:

```
<a href="http://assemble.io">Assemble</a>
```

Add a title

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

Renders to (hover over the link, there should be a tooltip):

Upstage

HTML:

```
<a href="https://github.com/upstage/" title="Visit Upstage!">Upstage</a>
```

Named Anchors

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

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

will jump to these sections:

```
## Chapter 1 <a id="chapter-1"></a>
Content for chapter one.

## Chapter 2 <a id="chapter-2"></a>
Content for chapter one.

## Chapter 3 <a id="chapter-3"></a>
Content for chapter one.
```

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

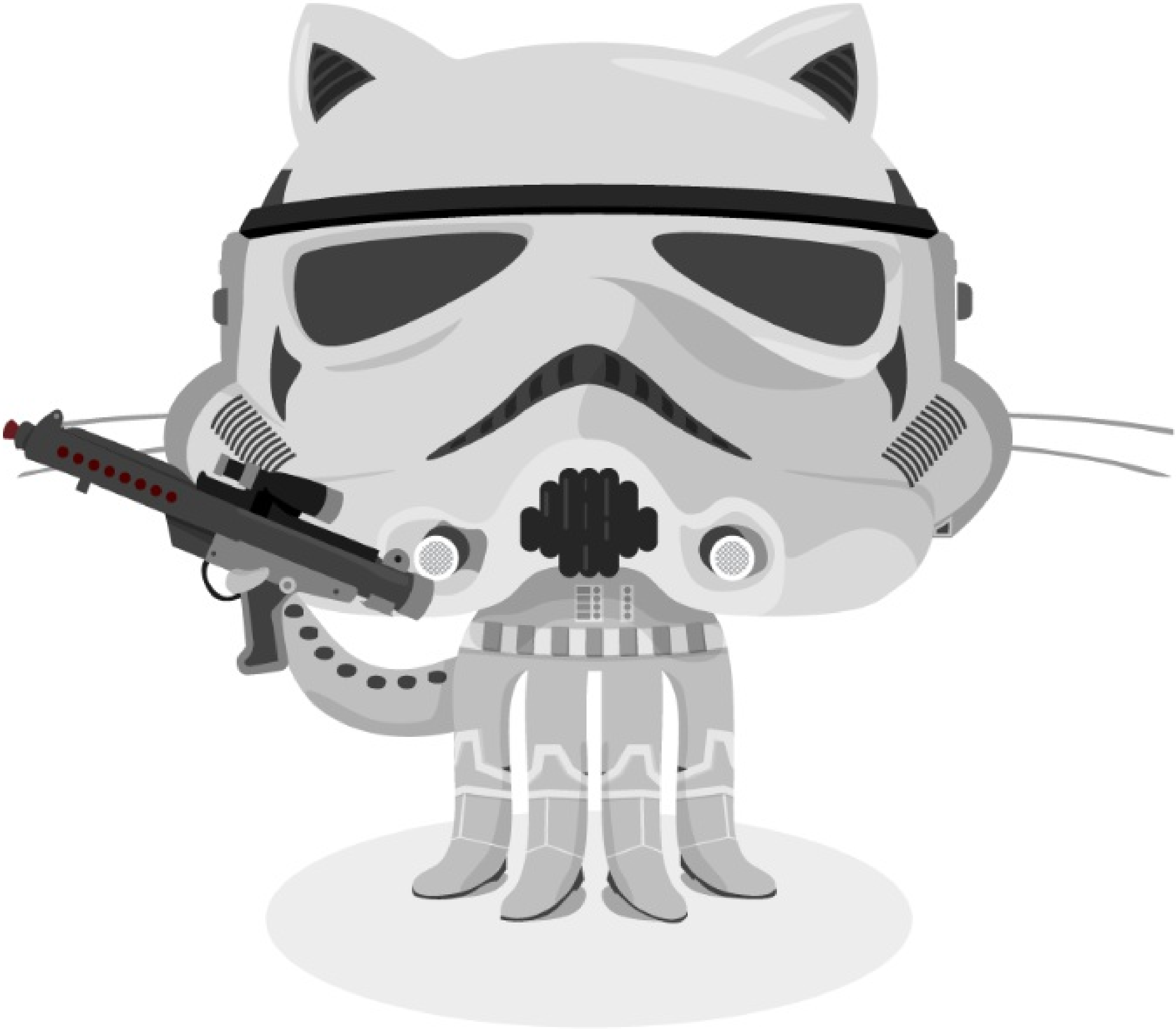
Images

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

```
![Minion](http://octodex.github.com/images/minion.png)
```

or

```
![Alt text](http://octodex.github.com/images/stormtroopocat.jpg "The Stormtroopocat")
```



Like links, Images also have a footnote style syntax

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

With a reference later in the document defining the URL location:

```
[id]: http://octodex.github.com/images/dojocat.jpg  "The Dojocat"
```