这篇文章提供了可以在 Hugo 的文章中使用的基本 Markdown 语法示例.

{{< admonition >}}
这篇文章借鉴了一篇很棒的来自 Grav 的文章.

如果你想了解 **Loveit** 主题的扩展 Markdown 语法, 请阅读扩展 Markdown 语法页面.
{{< /admonition >}}

事实上, 编写 Web 内容很麻烦. [WYSIWYG]^(所见即所得) 编辑器帮助减轻了这一任务. 但通常会导致代码太糟, 或更糟糕的是, 网页也会很丑.

没有通常伴随的所有复杂和丑陋的问题, **Markdown** 是一种更好的生成 **HTML** 内容的方式.

一些主要好处是:

1. Markdown 简单易学, 几乎没有多余的字符, 因此编写内容也更快.
2. 用 Markdown 书写时出错的机会更少.
3. 可以产生有效的 XHTML 输出.
4. 将内容和视觉显示保持分开, 这样就不会打乱网站的外观.
5. 可以在你喜欢的任何文本编辑器或 Markdown 应用程序中编写内容.
6. Markdown 使用起来很有趣!

John Gruber, Markdown 的作者如是说:

> Markdown 格式的首要设计目标是更具可读性.
> 最初的想法是 Markdown 格式的文档应当以纯文本形式发布,
> 而不会看起来像被标签或格式说明所标记.
> 虽然 Markdown 的语法受到几种现有的文本到 HTML 转换工具的影响,
> 但 Markdown 语法的最大灵感来源是纯文本电子邮件的格式.
>
> {{< style "text-align: right;" >}}-- *John Gruber*{{< /style >}}

话不多说, 我们来回顾一下 Markdown 的主要语法以及生成的 HTML 样式!

{{< admonition tip >}}
:(far fa-bookmark fa-fw): 将此页保存为书签，以备将来参考!
{{< /admonition >}}

1 标题

从 `h2` 到 `h6` 的标题在每个级别上都加上一个 `#` :

```
## h2 标题
### h3 标题
#### h4 标题
##### h5 标题
###### h6 标题
```

输出的 HTML 看起来像这样:

```
<h2>h2 标题</h2>
<h3>h3 标题</h3>
<h4>h4 标题</h4>
<h5>h5 标题</h5>
<h6>h6 标题</h6>
```

{{< admonition note "标题 ID" >}}
要添加自定义标题 ID, 请在与标题相同的行中将自定义 ID 放在花括号中:

```
### 一个很棒的标题 {#custom-id}
```

输出的 HTML 看起来像这样:

```
<h3 id="custom-id">一个很棒的标题</h3>
```

{{< /admonition >}}

2 注释

注释是和 HTML 兼容的:

```
<!--
这是一段注释
-->
```

不能看到以下的注释:

3 水平线

HTML 中的 `<hr>` 标签是用来在段落元素之间创建一个 "专题间隔" 的.
使用 Markdown, 你可以用以下方式创建一个 `<hr>` 标签:

- `___` : 三个连续的下划线
- `---` : 三个连续的破折号
- `***` : 三个连续的星号

呈现的输出效果如下:

4 段落

按照纯文本的方式书写段落, 纯文本在呈现的 HTML 中将用 `<p>` / `</p>` 标签包裹.

如下段落:

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

输出的 HTML 看起来像这样:

```
<p>Lorem ipsum dolor sit amet, graecis denique ei vel, at duo primis mandamus. Et legere ocurrer
```

可以使用一个空白行进行**换行**.

5 内联 HTML 元素

如果你需要某个 HTML 标签 (带有一个类), 则可以简单地像这样使用:

```
Markdown 格式的段落.

<div class="class">
    这是 <b>HTML</b>
</div>

Markdown 格式的段落.
```

6 强调

加粗

用于强调带有较粗字体的文本片段.

以下文本片段会被 **渲染为粗体**.

```
**渲染为粗体**
__渲染为粗体__
```

输出的 HTML 看起来像这样:

```
<strong>渲染为粗体</strong>
```

斜体

用于强调带有斜体的文本片段.

以下文本片段被 *渲染为斜体* .

```
*渲染为斜体*
_渲染为斜体_
```

输出的 HTML 看起来像这样:

```
<em>渲染为斜体</em>
```

删除线

按照 [GFM]^(GitHub flavored Markdown) 你可以使用删除线.

```
~~这段文本带有删除线.~~
```

呈现的输出效果如下:

~~这段文本带有删除线.~~

输出的 HTML 看起来像这样:

```
<del>这段文本带有删除线.</del>
```

组合

加粗, 斜体, 和删除线可以 组合使用.

```
***加粗和斜体***
~~**删除线和加粗**~~
~~*删除线和斜体*~~
~~***加粗, 斜体和删除线***~~
```

呈现的输出效果如下:

加粗和斜体

删除线和加粗

~~*删除线和斜体*~~

~~***加粗, 斜体和删除线***~~

输出的 HTML 看起来像这样:

```
<em><strong>加粗和斜体</strong></em>
<del><strong>删除线和加粗</strong></del>
<del><em>删除线和斜体</em></del>
<del><em><strong>加粗, 斜体和删除线</strong></em></del>
```

7 引用

用于在文档中引用其他来源的内容块.

在要引用的任何文本之前添加 `>` :

```
> **Fusion Drive** combines a hard drive with a flash storage (solid-state drive) and presents i
```

呈现的输出效果如下:

> **Fusion Drive** combines a hard drive with a flash storage (solid-state drive) and presents it as a single logical volume with the space of both drives combined.

输出的 HTML 看起来像这样:

```
<blockquote>
  <p>
    <strong>Fusion Drive</strong> combines a hard drive with a flash storage (solid-state drive)
  </p>
</blockquote>
```

引用也可以嵌套:

```
> 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.
```

呈现的输出效果如下:

> 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.

8 列表

无序列表

一系列项的列表, 其中项的顺序没有明显关系.

你可以使用以下任何符号来表示无序列表中的项:

```
* 一项内容
- 一项内容
+ 一项内容
```

例如:

```
* 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
```

呈现的输出效果如下:

- 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

输出的 HTML 看起来像这样:

```
<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>
```

有序列表

一系列项的列表, 其中项的顺序确实很重要.

```
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
```

呈现的输出效果如下:

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

输出的 HTML 看起来像这样:

```
<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>
```

{{< admonition tip >}}
如果你对每一项使用 `1.`, Markdown 将自动为每一项编号. 例如:

```
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
```

呈现的输出效果如下:

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

   {{< /admonition >}}

任务列表

任务列表使你可以创建带有复选框的列表.
要创建任务列表, 请在任务列表项之前添加破折号 (`-`) 和带有空格的方括号 (`[ ]`). 要选择一个复选框, 请在方括号之间添加 x (`[x]`).

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

呈现的输出效果如下:

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

9 代码

行内代码

用 ` 包装行内代码段.

```
在这个例子中，`<section></section>` 会被包裹成 **代码**.
```

呈现的输出效果如下:

在这个例子中，`<section></section>` 会被包裹成 **代码**.

输出的 HTML 看起来像这样:

```
<p>
  在这个例子中，<code>&lt;section&gt;&lt;/section&gt;</code> 会被包裹成 <strong>代码</strong>.
</p>
```

缩进代码

将几行代码缩进至少四个空格，例如:

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

呈现的输出效果如下:

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

输出的 HTML 看起来像这样:

```
<pre>
  <code>
    // Some comments
    line 1 of code
    line 2 of code
    line 3 of code
  </code>
</pre>
```

围栏代码块

使用 "围栏" ``` 来生成一段带有语言属性的代码块.

{{< highlight markdown >}}

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

{{< / highlight >}}

输出的 HTML 看起来像这样:

```
<pre language-html>
  <code>Sample text here...</code>
</pre>
```

语法高亮

[GFM]^(GitHub Flavored Markdown) 也支持语法高亮.

要激活它，只需在第一个代码 "围栏" 之后直接添加你要使用的语言的文件扩展名, `` ```js ``，语法高亮显示将自动应用于渲染的 HTML 中.

例如, 在以下 JavaScript 代码中应用语法高亮:

{{< highlight markdown >}}

```
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']
      }
    }
  }
};
```

{{< / highlight >}}

呈现的输出效果如下:

```
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']
      }
    }
  }
};
```

{{< admonition >}}
Hugo 文档中的 语法高亮页面 介绍了有关语法高亮的更多信息,
包括语法高亮的 shortcode.
{{< /admonition >}}

10 表格

通过在每个单元格之间添加竖线作为分隔线, 并在标题下添加一行破折号 (也由竖线分隔) 来创建表格. 注意, 竖线不需要垂直对齐.

```
| 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. |
```

呈现的输出效果如下:

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.

输出的 HTML 看起来像这样:

```
<table>
  <thead>
    <tr>
      <th>Option</th>
      <th>Description</th>
    </tr>
  </thead>
  <tbody>
    <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>
  </tbody>
</table>
```

{{< admonition note "文本右对齐或居中对齐" >}}
在任何标题下方的破折号右侧添加冒号将使该列的文本右对齐.

在任何标题下方的破折号两边添加冒号将使该列的对齐文本居中.

```
| 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. |
```

呈现的输出效果如下:

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.

{{< /admonition >}}

11 链接

基本链接

```
<https://assemble.io>
<contact@revolunet.com>
[Assemble](https://assemble.io)
```

呈现的输出效果如下 (将鼠标悬停在链接上，没有提示):

https://assemble.io

contact@revolunet.com

Assemble

输出的 HTML 看起来像这样:

```
<a href="https://assemble.io">https://assemble.io</a>
<a href="mailto:contact@revolunet.com">contact@revolunet.com</a>
<a href="https://assemble.io">Assemble</a>
```

添加一个标题

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

呈现的输出效果如下 (将鼠标悬停在链接上，会有一行提示):

Upstage

输出的 HTML 看起来像这样:

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

定位标记

定位标记使你可以跳至同一页面上的指定锚点. 例如, 每个章节:

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

将跳转到这些部分:

```
## 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.
```

{{< admonition >}}
定位标记的位置几乎是任意的. 因为它们并不引人注目, 所以它们通常被放在同一行了.
{{< /admonition >}}

12 脚注

脚注使你可以添加注释和参考, 而不会使文档正文混乱.
当你创建脚注时, 会在添加脚注引用的位置出现带有链接的上标编号.
读者可以单击链接以跳至页面底部的脚注内容.

要创建脚注引用, 请在方括号中添加插入符号和标识符 (`[^1]`).
标识符可以是数字或单词, 但不能包含空格或制表符.
标识符仅将脚注引用与脚注本身相关联 - 在脚注输出中, 脚注按顺序编号.

在中括号内使用插入符号和数字以及用冒号和文本来添加脚注内容 (`[^1]: 这是一段脚注`).
你不一定要在文档末尾添加脚注. 可以将它们放在除列表, 引用和表格等元素之外的任何位置.

```
这是一个数字脚注[^1].
这是一个带标签的脚注[^label]

[^1]: 这是一个数字脚注
[^label]: 这是一个带标签的脚注
```

这是一个数字脚注[1].

这是一个带标签的脚注[2]

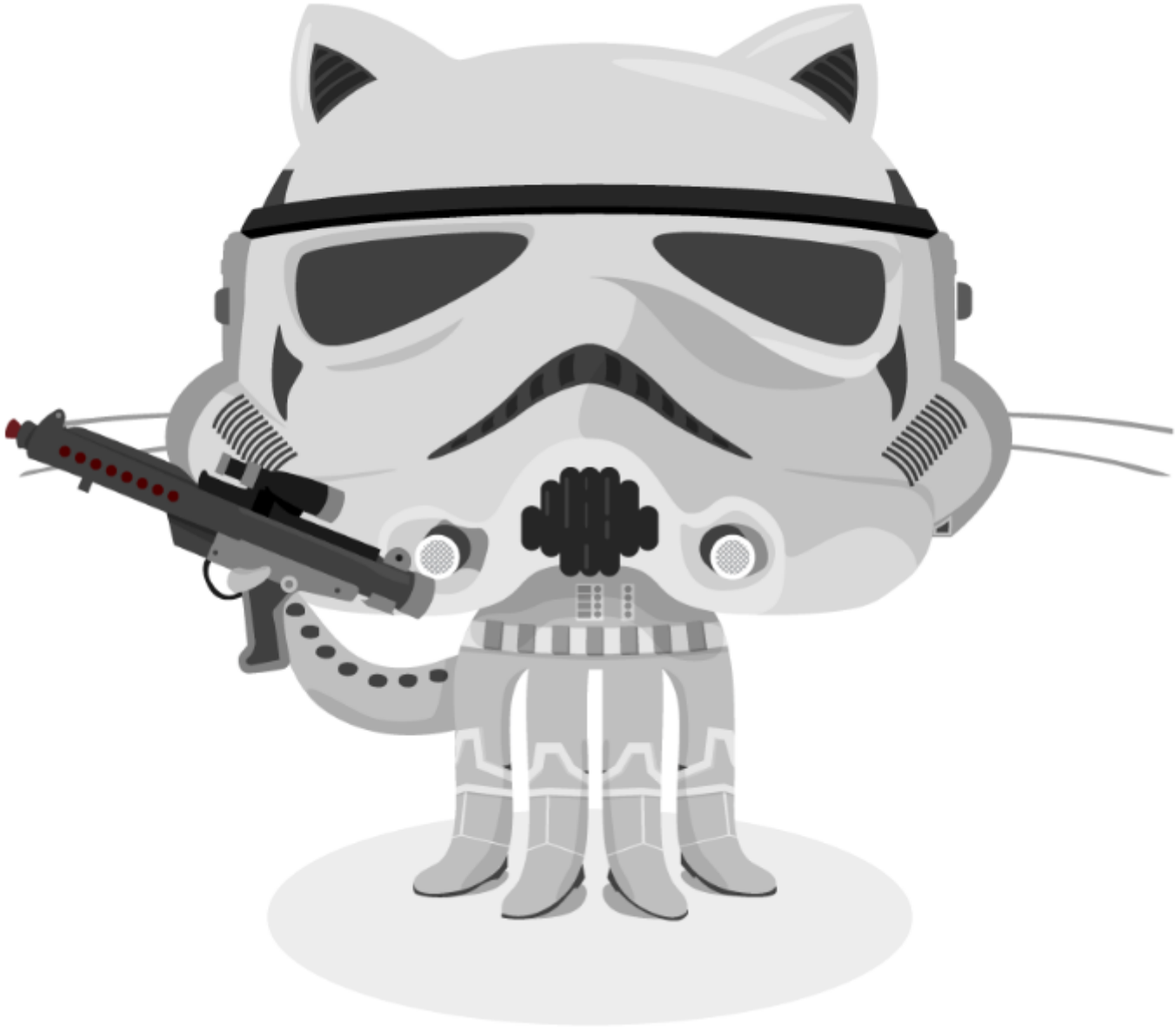
13 图片

图片的语法与链接相似, 但包含一个在前面的感叹号.

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

或者:

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



像链接一样, 图片也具有脚注样式的语法:

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

稍后在文档中提供参考内容, 用来定义 URL 的位置:

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

{{< admonition tip >}}
LoveIt 主题提供了一个包含更多功能的 图片的 shortcode.
{{< /admonition >}}

1. 这是一个数字脚注 ↩
2. 这是一个带标签的脚注 ↩