

标准 Markdown 教程

零基础入门系列教程

作者：魏明择

2025 年版

<https://weimingze.com>

目录

序

前言

第一章、标准 Markdown

块元素

1. 标题
2. 列表
3. 引用块
4. 段落和换行符
5. 代码块 (Code Blocks)
6. 水平线 (Horizontal Rules)

行内元素

1. 超链接
2. 强调
3. 代码
4. 图片

第二章、扩展 Markdown

1. 表格
2. 目录

序

前言

什么是 Markdown?

Markdown 是一种轻量级的标记语言，与超文本标记语言（HyperText Markup Language，简称 HTML）不同，Markdown 的目标是让人们能够使用易读易写的纯文本格式编写文档，然后可以轻松地转换成 HTML 文档或其他格式的文档，如：PDF 等。

Markdown 文件名后缀为 `.md` 或 `.markdown`。

为什么学 Markdown?

学习 Markdown 是一个提升编写效率、简化写作难度和提升文档质量的绝佳选择，大部分开源软件及运维的文档都是使用 Markdown 格式编写的。应用范围十分广泛。

Markdown 的语法非常简单，只包含少量易于记忆的符号。以下是标准 Markdown 的全部的语法标记符号。

标准 Markdown 语法标记

<code>\</code>	反斜杠
<code>`</code>	反引号
<code>*</code>	星号
<code>_</code>	下划线
<code>{}</code>	大括号（又名花括号）
<code>[]</code>	中括号（又名方括号）
<code>()</code>	小括号
<code>#</code>	井号
<code>+</code>	加号
<code>-</code>	减号
<code>.</code>	点号
<code>!</code>	叹号

在内容中用到上述符号，需要前面加上一个反斜杠 `\` 进行转义，如：`*` 表示一个星号 `*`。

Markdown 按换行方式分为两类：

1. 块元素：每一个新元素总是重新开始一个新行，并独自占用一行，前后不可有其他内容。类型如下：
 - 标题 (Headers)
 - 段落和换行符 (Paragraphs and Line Breaks)
 - 引用块 (Blockquotes)
 - 列表 (Lists)
 - 代码块 (Code Blocks)
 - 水平线 (Horizontal Rules)
2. 行内元素：不会从新行开始，只占据必要的宽度。类型如下：
 - 超链接 (Links)
 - 强调 (Emphasis)
 - 代码 (Code)
 - 图片 (Images)

Markdown 按语法规则分为两类：

1. 标准 Markdown。
 - 所有的 Markdown 软件都支持且兼容。如：列表、代码块。
2. 扩展 Markdown。
 - 扩展语法，部分软件支持。如：表格、目录、LeTeX 公式等。

常用的Markdown编辑软件：

- Visual Studio Code，下载地址：<https://code.visualstudio.com/download>
- Typora，下载地址：<http://typora.io/>
 - 可以直接保存成 HTML 文档和 PDF 文档
 - 旧版本不需注册，有需要可在 **明择工作室** 视频号或抖音 或 **明择编程** 公众号和我联系。
- PyCharm、Sublime Text、XCode 等。

第一章、标准 Markdown

块元素

块元素是指每一个新元素总是重新开始一个新行，并独自占用一行，前后不可有其他内容。

1. 标题

标题的语法通常用于文章的主标题、副标题和一些小标题。

标题的语法是在行首以英文的井号 # 开头，连着 1~6 个井号后加至少一个空格，然后写文章本标题。井号越多标题字体越小。这些标题在生成 HTML 时依次对应 `<h1>` ~ `<H6>` 标签。

示例

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

效果如下：

这是H1标题

这是H2标题

这是H3标题

这是H4标题

这是H5标题

这是H6标题

对应 HTML 文档如下:

```
<h1>这是H1标题</h1>  
<h2>这是H2标题</h2>  
<h3>这是H3标题</h3>  
<h4>这是H4标题</h4>  
<h5>这是H5标题</h5>  
<h6>这是H6标题</h6>
```

段落的一级和二级标题还可以写成标题内容 然后第二行的行首开始 三个等号 (===) 代表一级标题, 三个减号 (---) 代表二级标题.

示例

```
这是一级标题  
=====  
  
这是二级标题  
-----
```

效果如下:

这是一级标题

这是二级标题

对应 HTML 文档如下：

```
<h1>这是一级标题</h1>  
<h2>这是二级标题</h2>
```

2. 列表

列表是用于组织项目的一种结构，他可以向清单一样自上而下排列。

列表分为 **有序列表** 和 **无序列表** 两种，分别对应 HTML 里的 `` 标签和 `` 标签。

无序列表

无序列表 的语法是以 星号 (*)、加号 (+) 或 减号 (-) 开始，后跟至少一个空格后写文字内容。

示例

- 北京
- 上海
- 重庆

效果如下：

- 北京
- 上海
- 重庆

对应 HTML 文档如下：

```
<ul>
<li>北京</li>
<li>上海</li>
<li>重庆</li>
</ul>
```

使用列表时可以用空格缩进代表包含关系，标准的缩进规则是 **四个空格** 代表一级缩进，如下所示：

示例

```
+ 北京
  - 东城区
  - 朝阳区
+ 上海
  + 徐汇区
  + 闵行区
+ 重庆
```

效果如下：

- 北京
 - 东城区
 - 朝阳区
- 上海
 - 徐汇区
 - 闵行区
- 重庆

还可以有三级、四级缩进...

对应 HTML 文档如下：

```
<ul>
<li>北京<ul>
<li>东城区</li>
<li>朝阳区</li>
</ul>
</li>
<li>上海<ul>
```

```
<li>徐汇区</li>
<li>闵行区</li>
</ul>
</li>
<li>重庆</li>
</ul>
```

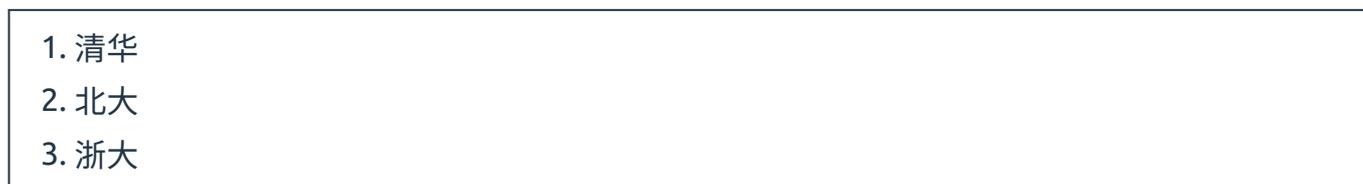
有序列表

有序列表的语法是以数字开头后加一个英文的点 (.) 后跟至少一个空格后写文字内容。数字是多少没有实际意义。Markdown 会自上而下从 1 开始排列。

示例

```
1. 清华
1. 北大
1. 浙大
```

效果如下：



对应 HTML 文档如下：

```
<ol>
<li>清华</li>
<li>北大</li>
<li>浙大</li>
</ol>
```

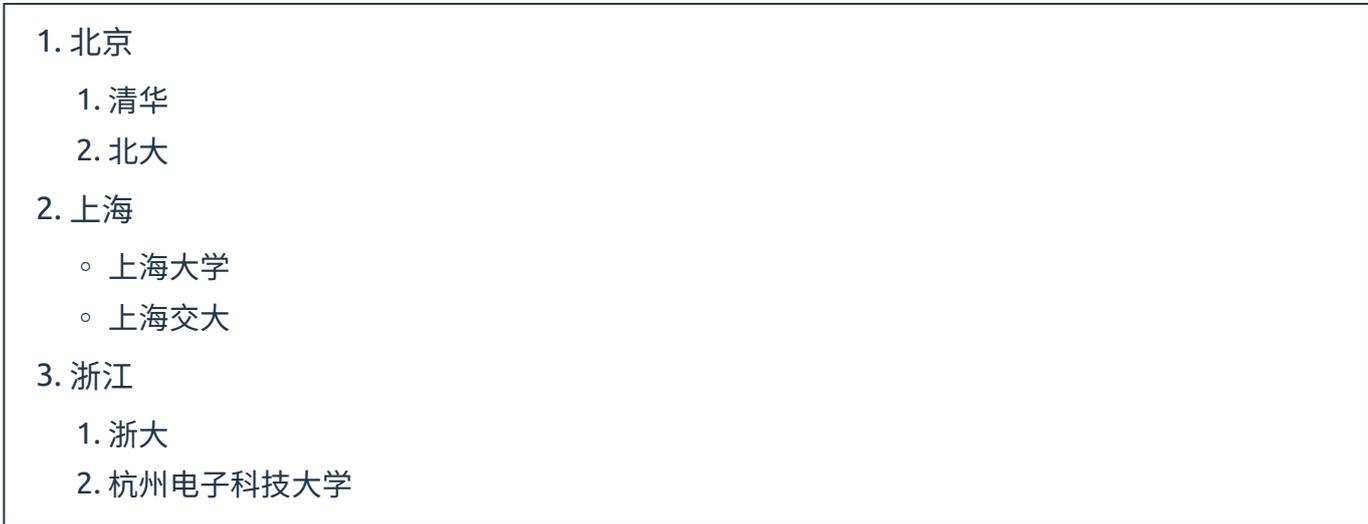
同样使用无序列表时也可以用空格缩进代表包含关系，标准的缩进规则是四个空格代表一级缩进。

示例

```
1. 北京
    1. 清华
    1. 北大
2. 上海
    * 上海大学
```

```
* 上海交大
3. 浙江
  1. 浙大
  2. 杭州电子科技大学
```

效果如下：



对应 HTML 文档如下：

```
<ol>
<li>北京<ol>
<li>清华</li>
<li>北大</li>
</ol>
</li>
<li>上海<ul>
<li>上海大学</li>
<li>上海交大</li>
</ul>
</li>
<li>浙江<ol>
<li>浙大</li>
<li>杭州电子科技大学</li>
</ol>
</li>
</ol>
```

3. 引用块

引用块通常用于表示一段引用的文字，比如引用文献、注解或突出某段文字。引用块语法是：在行首使用一个大于号 (>) ，然后跟至少一个空格，接着写引用的内容。

示例

```
> 我是一个引用块的内容。
```

效果如下：

我是一个引用块的内容。

对应 HTML 文档如下：

```
<blockquote>
<p>我是一个引用块的内容。</p>
</blockquote>
```

写入多行是需要空一个同级别的引用块，否则下一行内容视为上面同一行。也可以用如下的方式来进行应用嵌套。如下：

示例

```
> 我是引用块的第一段。
>
> 我是引用块的第二段。
>
> > 我是引用块的二级嵌套的内容。
>
> 1. 列表项一
>
> 1. 列表项二
```

效果如下：

我是引用块的第一段。

我是引用块的第二段。

我是引用块的二级嵌套的内容。

1. 列表项一

2. 列表项二

4. 段落和换行符

在 Markdown 中写入的没有语法的文字。Markdown 会将其视为一个段落。在生成 HTML 后是一个 `<p>` 标签。

在两个段落之间加一个空行，则下一行会认为是另外一个段落，并且不会同上一行合并成同一个段落。

示例1

以下内容是虽然写在两行，但会认为是同一个段落。

```
我是一段文字。  
我是另一段文字。
```

效果如下：

```
我是一段文字。 我是另一段文字。
```

对应 HTML 文档如下：

```
<p>我是一段文字。  
我是另一段文字。</p>
```

示例2

加入空行，变为两段文字。

```
我是一段文字。  
  
我是另一段文字。
```

效果如下：

```
我是一段文字。  
  
我是另一段文字。
```

对应 HTML 文档如下:

```
<p>我是一段文字。</p>
<p>我是另一段文字。</p>
```

5. 代码块 (Code Blocks)

代码块的主要用于以固定宽度的字体显示一段代码，并且保留代码中的空格、换行和其他格式。这样可以让代码在文档中清晰地展示，与普通文本区分开来，提高可读性。

在 Markdown 中，有两种方式创建代码块：

1. 缩进创建代码块：
 - 使用四个空格或一个制表符 (Tab) 来缩进每一行代码。
2. 三个反引号 (```) 创建代码

缩进创建代码块

示例

以下是一个代码块。

```
# include <stdio.h>
int main(int argc, char *argv[]) {
    return 0;
}
```

效果如下:

以下是一个代码块。

```
# include <stdio.h>
int main(int argc, char *argv[]) {
    return 0;
}
```

对应 HTML 文档如下:

```
<p>以下是一个代码块。</p>
<pre><code># include &lt;stdio.h&gt;
int main(int argc, char *argv[]) {
    return 0;
}
```

```
}  
</code></pre>
```

三个反引号 (```) 创建代码块

示例

```
```c  
include <stdio.h>
int main(int argc, char *argv[]) {
 return 0;
}
```
```

在开始的三个反引号 (```) 后面的 `c` 表示当前代码的语言，有些编辑器会根据此表示给出语法高亮。

这个语言标识可以是：python、c、cpp、mysql、shell等。

效果如下：

```
# include <stdio.h>  
int main(int argc, char *argv[]) {  
    return 0;  
}
```

对应 HTML 文档如下：

```
<p><strong>效果如下: </strong></p>  
<pre><code class="language-c"># include &lt;stdio.h&gt;  
int main(int argc, char *argv[]) {  
    return 0;  
}  
</code></pre>
```

6. 水平线 (Horizontal Rules)

水平线的作用主要是 **分隔文档不同章节或内容区块**。

水平线的语法如下所示，下面的每一行都可以产生一个水平线。

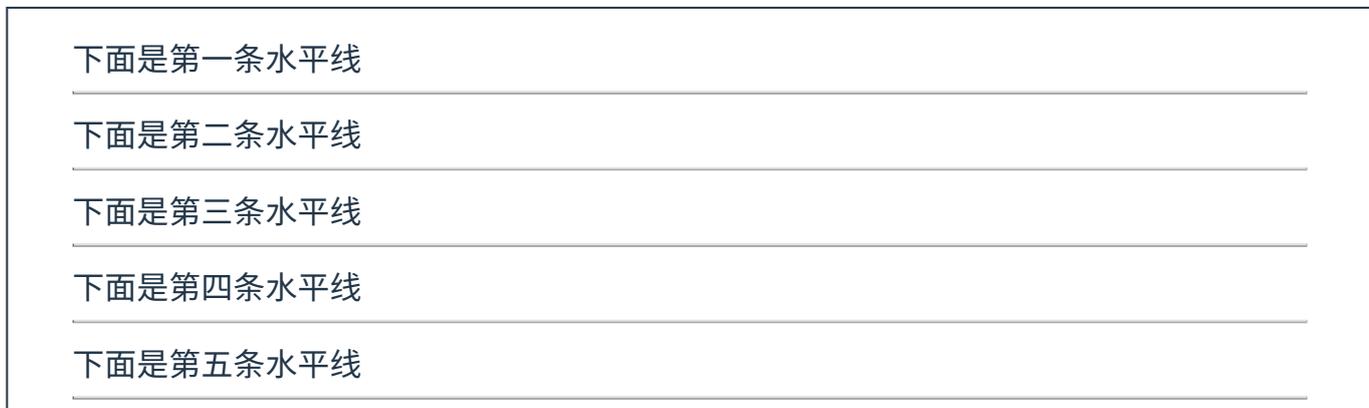
一条水平线对应 HTML 的一个 `<hr />` 标签。

```
* * *  
  
***  
  
*****  
  
- - -  
  
-----
```

示例

```
下面是第一条水平线  
* * *  
  
下面是第二条水平线  
***  
  
下面是第三条水平线  
*****  
  
下面是第四条水平线  
- - -  
  
下面是第五条水平线  
  
-----
```

效果如下：



对应 HTML 文档如下：

```
<p>下面是第一条水平线</p>  
<hr />  
<p>下面是第二条水平线</p>  
<hr />  
<p>下面是第三条水平线</p>  
<hr />  
<p>下面是第四条水平线</p>  
<hr />
```

```
<p>下面是第五条水平线</p>  
<hr />
```

行内元素

行内元素：不会从新行开始，只占据必要的宽度。也就是说行内元素，多个元素可以写在一行内。

1. 超链接

超链接（Hyperlink）是用于创建可点击的文本或图像，使用户能够快速跳转到指定资源的元素。其作用是建立内容间的连接关系，提升信息的关联性和可访问性。

常用的超链接有三种语法：

1. 内联链接。
2. 自动链接。
3. 引用链接。

1. 内联链接

主要用于显示的内容和链接地址不一样时的超链接。

语法

```
[显示文本](链接地址 "超链接的文本注释")`
```

"超链接的文本注释" 可以不写。

示例

主页地址：[魏明择的官方网站](https://weimingze.com "点我跳转到魏明择的网站")

教程位置：[Markdown教程](https://weimingze.com/markdown/)

效果如下：

主页地址：[魏明择的官方网站](https://weimingze.com)

教程位置: [Markdown教程](#)

对应 HTML 文档如下:

```
<p>主页地址: <a href="https://weimingze.com" title="點我跳转到魏明择的网站">魏明择的官方网站</a></p>
<p>教程位置: <a href="https://weimingze.com/markdown/">Markdown教程</a></p>
```

2. 自动链接

用于显示的内容和链接地址一致的超链接。主要用于 **URL 链接** 和 **E-Mail 链接**。语法简单。

语法

```
<链接地址>
```

示例

```
Python教程的链接地址: <https://weimingze.com/pythonbase/>
```

效果如下:

Python教程的链接地址: <https://weimingze.com/pythonbase/>

对应 HTML 文档如下:

```
<p>Python教程的链接地址: <a href="https://weimingze.com/pythonbase/">https://weimingze.com/pythonbase/</a></p>
```

3. 引用链接

引用链接 (Reference Link) 可以让你定义一个链接,并在文档中多次引用它,使文档更整洁易读。

引用链接分为两部分:

1. **定义引用**: 定义一个链接标签,也就是给链接取个名字或标号,便于关联此链接。
 - 定义引用可以在文档的任意位置,一般在文档的最下端。

◦ 定义引用的部分不会在文档中显示。

2. **使用链接引用**：创建超链接，链接地址给出标签（而非直接给出链接地址）。

定义引用的语法

```
[链接标签名]:  链接地址  "提示文本"  
# 或  
[链接标签名]:  链接地址
```

语法说明

1. **链接标签名**可以是字符，数字和下划线(不区分大小写)
2. **提示文本**是鼠标停靠提示语，**提示文本**和**链接地址**之间要有一个或多个空格分隔。
3. 提示文本是可选择，文本两段可以用单引号，也可以用双引号括起来。
4. 停靠光标提示语可以放到一下行，可以缩进。

使用链接引用的语法

```
[文字描述][链接标签名]
```

语法说明

1. **文字描述**是用户可见部分。
2. **链接标签名**必须是在文档中定义的标签。

示例

在网络上有一个 [标准 Markdown 教程][1]，他是 [魏明择][weimz]老师写的，这个教程可以在 [魏明择的官方网站][weimz] 找到。

关于 Markdown 的语法非常简单，建议大家学习。

```
[1]: https://weimingze.com/markdown/  "这是中文标准 Markdown 教程!"  
[weimz]: https://weimingze.com/
```

效果如下：

在网络上有一个 [标准 Markdown 教程](#)，他是 [魏明择](#)老师写的，这个教程可以在 [魏明择的官方网站](#) 找到。

关于 Markdown 的语法非常简单，建议大家学习。

对应 HTML 文档如下:

```
<p>在网上有一个 <a href="https://weimingze.com/markdown/" title="这是中文标准 Markdown 教程!">标准 Markdown 教程</a>, 他是 <a href="https://weimingze.com/">魏明择</a>老师写的, 这个教程可以在 <a href="https://weimingze.com/">魏明择的官方网站</a> 找到。</p>
<p>关于 Markdown 的语法非常简单, 建议大家学习。</p>
```

2. 强调

强调 是在一个行内的文字增加加粗、斜体、粗斜体等特殊字体风格, 让读者引起注意。

标准Markdown 的强调类型有:

1. 加粗
2. 斜体
3. 粗斜体

加粗的语法: 以两个星号 (**) 或两个下划线 (__) 开始和结束。中间包裹起来的文字会加粗显示。

斜体的语法: 以一个星号 (*) 或一个下划线 (_) 开始和结束。中间包裹起来的文字会斜体显示。

粗斜体的语法: 以三个星号 (***) 或三个下划线 (___) 开始和结束。中间包裹起来的文字会用粗斜体显示。

语法说明: 强调语法的开始符号前面最好加空格, 后面不能加空格; 结束符号的前面不能加空格。并且开始和结束符号要成对出现。

如: ** 这不是加粗字体 **, 因为文字前后有空格。

如: _这不是斜体字_, 因为开始符号和结束符号不一致。

示例

我是**粗体字**, 我也是 __粗体字__。

我是*斜体字*, 我也是 _斜体字_。

我是 ***加粗的斜体字***。

效果如下：

我是**粗体字**，我也是 **粗体字**。
我是*斜体字*，我也是 *斜体字*。
我是 ***加粗的斜体字***。

对应 HTML 文档如下：

```
<p>我是<strong>粗体字</strong>，我也是 <strong>粗体字</strong>。</p>  
<p>我是<em>斜体字</em>，我也是 <em>斜体字</em>。</p>  
<p>我是 <strong><em>加粗的斜体字</em></strong>。</p>
```

3. 代码

代码同代码块一样，内部可以显示空格和 Markdown 的语法标记符号。不同的地方在于代码是行内元素，不能独占一行，也不能换行。

代码的语法：以一个反引号（```）开始和结束。中间包裹起来的文字会用原始内容显示。

如果代码的内部有不连续的反引号，可以使用两个反引号（````）开始和结束。这样就可以显示内部的反引号了。

示例

```
GUN C 语言的 main 函数的格式是 `int main(int argc, char * argv[]) {}`，你能看懂吗？  
Linux 下命令替换的语法是 `` `命令 参数` 或 $(命令 参数)`
```

效果如下：

GUN C 语言的 main 函数的格式是 `int main(int argc, char * argv[]) {}`，你能看懂吗？
Linux 下命令替换的语法是 ``命令 参数` 或 $(命令 参数)`

对应 HTML 文档如下：

```
<p>GUN C 语言的 main 函数的格式是  
<code>int main(int argc, char * argv[]) {}</code>，你能看懂吗? </p>  
<p>Linux 下命令替换的语法是 <code>`命令 参数` 或 $(命令 参数)</code></p>
```

4. 图片

图片是用于在行内创建插入图片显示的语法。他允许在 Markdown 内插入 图片。

图片的语法

```
![图片不存在时的显示文本](图片链接地址 "图片提示文本")
```

或

```
![图片不存在时的显示文本](图片链接地址)
```

语法说明

1. "图片提示文本" 用于鼠标放在图片上给出的提示文字，可以不写。
2. 图片接连地址可以是 URL，也可以是本地相对路径或绝对路径,如: `./images/tkinter/logo.png`

示例

```
这个一个小鱼图片: ![小鱼图片](https://weimingze.com/static/images/favicon.ico)
```

```
这个一个本地图片: ![本地图片](./images/markdown_small.png "魏明择制作的Markdown图标")
```

```
这个一个不存在的图片: ![图片走丢了](https://weimingze.com/xxx.jpg)
```

效果如下:

这个一个小鱼图片:



这个一个本地图片:



这个一个不存在的图片:
图片走丢了

对应 HTML 文档如下:

```
<p>这个一个小鱼图片:  
    
</p>  
<p>这个一个本地图片: </p>  
<p>这个一个不存在的图片:   
</p>
```

第二章、扩展 Markdown

扩展 Markdown 的语法大大丰富了 Markdown 的功能。

扩展 Markdown 的语法仅在部分 Markdown 编辑器上才能够正常显示。

1. 表格

表格是一种使用对齐的行和列来展示文档中结构化数据展示的方式。

Markdown 表格是一种使用纯文本语法在 Markdown 文档中创建结构化数据展示（行和列）的方式。

Markdown 表格一定带有表头部分。

Markdown 表格的语法标记是 竖线 `|` 和减号 `-`。

含有两列，只有表头，没有数据的 Markdown 表格的写法如下：

```
| 标题1 | 标题2 |  
| - | - |
```

效果如下：

标题1	标题2

语法说明

- 第一行是表格标题部分，用竖线 (`|`) 来分隔各个列。
- 第二行是表格标题结束行，用竖线 (`|`) 来分隔各个列，每个竖线间至少要有一个减号。这一行的作用有两个：
 1. 分隔表格标题和表格内容。
 2. 设置表格列的对齐方式。
- 第二行是表格标题结束行中的减号 (`-`) 至少要有一个，可以是多个，并且左右可以加多个空格来装饰，如 `| - - - - - |`。
- 第一行表格标题中的竖线 (`|`) 的个数一定要等于第二行竖线的个数。
- 如果表格文字内容中含有竖线，需要用反斜杠和竖线 (`\|`) 来转义，否则无法显示竖线。

含有三列，只有表头，没有数据的 Markdown 表格的写法如下：

```
| 标题1 | 标题2 | 标题3 |
| - | - | - |
```

效果如下：

标题1	标题2	标题3

含有两列，同时有两行数据的 Markdown 表格的写法如下：

```
| 标题1 | 标题2 |
| - | - |
| 1行1列 | 1行2列 |
| 2行1列 | 2行2列 |
```

效果如下：

标题1	标题2
1行1列	1行2列
2行1列	2行2列

表格列的对齐方式

扩展 Markdown 的表格对齐方式有三种：

1. 左对齐（默认），用一个或多个减号（-）表示或减号左侧加一个冒号（:-）表示。
2. 居中对齐，减号的左右各加一个冒号，如： :-- 或 :-----:。
3. 右对齐，减号的右侧加一个冒号，如： -- 或 -----:。

示例

显示三行三列数据，分别示意三种对齐方式。

```
| 左对齐 | 居中对齐 | 右对齐 |
| :---- | :-----: | -----: |
| 一 | 一 | 一 |
| 二 三 | 二 三 | 二 三 |
| 四 五 六 | 四 五 六 | 四 五 六 |
```

效果如下：

左对齐	居中对齐	右对齐
—	—	—
二三	二三	二三
四五六	四五六	四五六

表格单元格内多余的空格会被删除，最多只保留一个。

2. 目录

扩展 Markdown 语法可以插入 **目录**。

目录的语法是：[TOC]，只要在需要的地方插入 [TOC] 标签。则在标签处就会生成目录。

示例

```
## Markdown 教程

[TOC]

### 第一章、标准 Markdown
#### 1、标提
内容...
#### 2、列表
内容...

### 第二章、扩展 Markdown
#### 1. 表格
内容...
#### 2. xxxxx
内容...
```

效果如下：

Markdown 教程

Markdown 教程

第一章、标准 Markdown

- 1、标提
- 2、列表

第二章、扩展 Markdown

- 1. 表格
- 2. xxxxx

第一章、标准 Markdown

1、标提

内容...

2、列表

内容...

第二章、扩展 Markdown

1. 表格

内容...

2. xxxxx

内容...

对应 HTML 文档如下:

```
<h2 id="markdown">Markdown 教程</h1>
<div class="toc">
<ul>
<li><a href="#markdown">Markdown 教程</a><ul>
<li><a href="#markdown_1">第一章、标准 Markdown</a><ul>
```

```
<li><a href="#1">1、标提</a></li>
<li><a href="#2">2、列表</a></li>
</ul>
</li>
<li><a href="#markdown_2">第二章、扩展 Markdown</a><ul>
<li><a href="#1_1">1. 表格</a></li>
<li><a href="#2-xxxxx">2. xxxxx</a></li>
</ul>
</li>
</ul>
</li>
</ul>
</div>
<h3 id="markdown_1">第一章、标准 Markdown</h1>
<h4 id="1">1、标提</h1>
<p>内容...</p>
<h4 id="2">2、列表</h1>
<p>内容...</p>
<h3 id="markdown_2">第二章、扩展 Markdown</h1>
<h4 id="1_1">1. 表格</h1>
<p>内容...</p>
<h4 id="2-xxxxx">2. xxxxx</h1>
<p>内容...</p>
```