

Table of Contents

简介	1
什么是 Markdown?	1
为何使用 Markdown?	1
Markdown 是如何工作的?	2
Markdown 有什么用?	2
Markdown 的风格	3
其他资源	3
Markdown Cheat Sheet	3
基本语法 Basic Syntax	4
Heading	4
Bold	4
Italic	4
Blockquote	4
Ordered List	4
Unordered List	5
Code	5
Horizontal Rule	5
Link	5
Image	5
扩展语法 Extended Syntax	5
Table	6
Fenced Code Block	6
Footnote	6
Heading ID	6
Definition List	7
Strikethrough	7
Task List	7
Emoji	7
Highlight	7
Subscript	7

Superscript	8
Basic Syntax 基本语法	8
概述 Overview	8
标题 Headings	8
备用语法	8
标题最佳实践	8
段落 Paragraphs	9
段落最佳实践	9
换行 Line Breaks	9
换行 (Line Break) 用法的最佳实践	9
强调 Emphasis	10
粗体 Bold	10
粗体最佳实践	10
斜体 Italic	11
斜体最佳实践	11
粗体和斜体 Bold and Italic	11
粗体和斜体最佳实践	12
块引用 Blockquotes	12
多个段落的块引用	12
嵌套块引用 (Nested Blockquotes)	12
带有其它元素的块引用 (Blockquotes with Other Elements)	13
The quarterly results look great!	13
Blockquotes 最佳实践	13
列表 Lists	13
有序列表 (Ordered Lists)	14
无序列表 (Unordered Lists)	14
在列表中添加元素	16
代码块 Code Blocks	19
转义反引号	19
代码块	19
分隔线 Horizontal Rules	20
分隔线最佳实践	20
链接 Links	20
添加标题	21
URL 和电子邮件地址	21
格式化链接	21
引用式链接	21

图片 Images	22
带链接的图片	23
转义字符 Escaping Characters	24
可做转义的（英文）字符	24
HTML 标签	25
HTML 最佳实践	25
Extended Syntax 扩展语法	26
概述 Overview	26
可用性	26
轻量级标记语言	26
Markdown 处理器	26
表格 Tables	27
对齐	27
格式化表格中的文本	27
围栏代码块 Fenced Code Blocks	28
语法高亮	28
脚注 Footnotes	29
锚链接 Anchor Links	30
定义列表 Definition Lists	30
任务列表 Task Lists	30
删除线 Strikethrough	31
表情符号 Emoji	31
复制并粘贴表情符号	31
使用表情符号的简码（Shortcodes）	31
Highlight	32
下标 Subscript	32
上标 Superscript	32
自动将 URL 转换为链接	33
禁止自动将 URL 转换为链接	33
Pandoc' s Markdown	33
概述 Overview	33
段落	33
Extension: escaped_line_breaks	34
标题	34
Setext 样式	34
ATX 格式	35
Extension: blank_before_header	35
Extension: space_in_atx_header	35

标题标识符	36
Extension: header_attributes	36
Extension: implicit_header_references	36
引用块 Block quotations	37
代码块 Verbatim (code) blocks	38
缩进的代码块 Indented code blocks	38
围栏代码块 Fenced code blocks	39
Extension: fenced_code_blocks	39
Extension: backtick_code_blocks	39
Extension: fenced_code_attributes	39
行块 Line Blocks	40
内联格式 Inline formatting	41
Extension: intraword_underscores	41
Extension: strikethrough	41
Extension: superscript 和 subscript	41
链接 Links	41
自动链接 Automatic Links	41
内联链接 Inline Links	42
引用链接 Reference Links	42
Extension: shortcut_reference_links	42
内部链接 Internal Links	42
使用元数据 Metadata	43
Hacks	44
下划线 Underline	44
文字居中 Center	44
数学公式 Mathematical Expressions	44
Inline math	44
Display Math	45
交叉引用 Cross Referencing	45
使用 pandoc-crossref	45
安装 pandoc 和 pandoc-crossref	45
运行 pandoc-crossref	46
直接转换成 PDF 时的错误处理	46
Cross Referencing	46
参考资料 References	47

简介

什么是 **Markdown**?

Markdown 是一种轻量级标记语言，你可以使用它向纯文本文档添加格式元素。Markdown 由 John Gruber 于 2004 年创建，现在是世界上最流行的标记语言之一。

使用 Markdown 与使用 WYSIWYG 编辑器不同。在 Microsoft Word 等应用程序中，你可以单击按钮来设置单词和短语的格式，并且更改会立即显示。Markdown 并非如此。当你创建一个 Markdown 格式的文件时，你向文本添加 Markdown 语法来指示哪些单词和短语应显示为不同格式。

例如，要表示标题，你可以在标题前添加一个井号（例如，# 标题一）。或者要使短语变为粗体，你可以在短语前后添加两个星号（例如，**此文本为粗体**）。在文本中看到 Markdown 语法可能需要一段时间才能适应，特别是如果你习惯于 WYSIWYG 应用程序。

你可以使用文本编辑器应用程序向纯文本文件添加 Markdown 格式元素。或者，你可以使用 macOS、Windows、Linux、iOS 和 Android 操作系统的众多 Markdown 应用程序之一。还有专门用于编写 Markdown 的几个基于 Web 的应用程序。

根据你使用的应用程序，你可能无法实时预览格式化的文档。但这没关系。据 John Gruber 所说，Markdown 语法旨在具有可读性和不显眼性，因此即使没有呈现 Markdown 文件中的文本，也可以读取该文本。

Markdown 格式语法的主要设计目标是使其尽可能具有可读性。其理念是 Markdown 格式的文档应可按原样发布为纯文本，而不会看起来像是用标记或格式化指令标记的。A Markdown-formatted document should be publishable as-is, as plain text, without looking like it's been marked up with tags or formatting instructions. - John Gruber

为何使用 **Markdown**?

你可能想知道，人们为何使用 Markdown 而不是所见即所得编辑器。为什么在界面中按按钮格式化文本时还要使用 Markdown 来编写？事实证明，人们使用 Markdown 而不是所见即所得编辑器的原因有很多。

- Markdown 可用于一切。人们使用它来创建网站、文档、笔记、书籍、演示文稿、电子邮件和技术文档。
- Markdown 是可移植的。包含 Markdown 格式文本的文件几乎可以使用任何应用程序打开。如果你决定不喜欢当前使用的 Markdown 应用程序，你可以将 Markdown 文件导入另一个 Markdown 应用程序。这与 Microsoft Word 等将内容锁定为专有文件格式的文字处理应用程序形成了鲜明的对比。
- Markdown 与平台无关。你可以在运行任何操作系统的任何设备上创建 Markdown 格式的文本。

- Markdown 具有未来性。即使你使用的应用程序在未来某个时间点停止工作，你仍然可以使用文本编辑应用程序阅读 Markdown 格式的文本。对于需要无限期保存的书籍、大学论文和其他里程碑式文档，这是一个重要的考虑因素。
- Markdown 无处不在。像 Reddit 和 GitHub 这样的网站支持 Markdown，并且许多桌面和基于 Web 的应用程序也支持它。

Markdown 是如何工作的？

当您使用 Markdown 书写时，文本会存储在具有 .md 或 .markdown 扩展名的纯文本文件中。但接下来呢？Markdown 格式的文件如何转换为 HTML 或可打印的文档？

简而言之，您需要一个能够处理 Markdown 文件的 Markdown 应用程序。

Markdown 应用程序使用称为 Markdown 处理器（通常也称为“解析器”或“实现”）的东西，将 Markdown 格式的文本提取出来并将其输出为 HTML 格式。在这一点上，您的文档可以在网络浏览器中查看，或与样式表结合使用并打印出来。您可以在下面看到此过程的可视化表示。注意

Markdown 应用程序和处理器是两个独立的组件。为了简洁起见，我在下图中将它们合并为一个元素（“Markdown 应用程序”）。

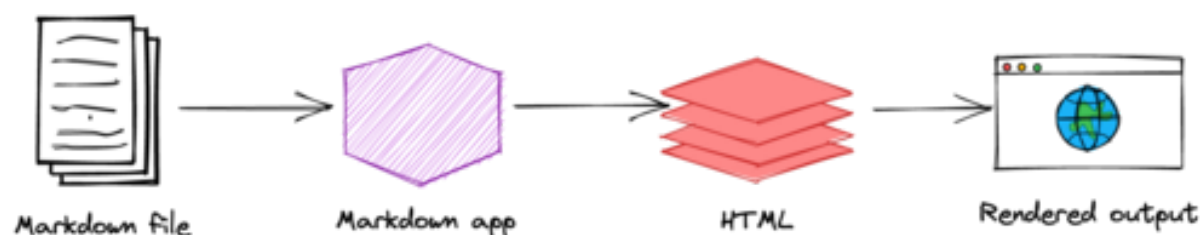


Figure 1: Markdown Flowchart

总而言之，这是一个由四部分组成的过程

1. 使用文本编辑器或专门的 Markdown 应用程序创建 Markdown 文件。该文件应具有 .md 或 .markdown 扩展名。
2. 在 Markdown 应用程序中打开 Markdown 文件。
3. 使用 Markdown 应用程序将 Markdown 文件转换为 HTML 文档。
4. 在网络浏览器中查看 HTML 文件，或使用 Markdown 应用程序将其转换为其他文件格式，例如 PDF。

Markdown 有什么用？

Markdown 是一种快速简便的方法，可用于做笔记、为网站创建内容以及生成可打印的文档。

学习 Markdown 语法并不需要很长时间，一旦您知道如何使用它，您就可以在几乎任何地方使用 Markdown 来书写。大多数人使用 Markdown 为网络创建内容，但 Markdown 适用于从电子邮件到购物清单的所有内容的格式化。

Markdown 的风格

使用 Markdown 最令人困惑的方面之一是几乎每个 Markdown 应用程序都实现了略有不同的 Markdown 版本。这些 Markdown 变体通常称为风格。您的工作是掌握应用程序已实现的任何风格的 Markdown。

为了理解 Markdown 语言的理念，可以将它们视为语言方言。纽约市的人们和伦敦的人们都说英语，但这两个城市使用的方言之间存在很大差异。使用不同 Markdown 应用程序的人们也是如此。

实际上，这意味着你永远不知道当一家公司说他们支持“Markdown”时，他们具体是什么意思。他们所说的仅仅是基本语法元素吗，还是所有基本和扩展语法元素的组合，或者是一些任意的语法元素组合？只有在阅读文档或开始使用应用程序后，你才能知道。

如果你刚开始，我能给你的最佳建议是选择一个 Markdown 支持良好的 Markdown 应用程序。这将极大地提高 Markdown 文件的可移植性。你可能希望在其他应用程序中存储和使用 Markdown 文件，而要做到这一点，你需要从提供良好支持的应用程序入手。你可以使用工具目录找到一个合适的应用程序。

其他资源

有许多资源可用于学习 Markdown。以下是一些其他入门资源：

- **John Gruber 的 Markdown 文档** 由 Markdown 创建者编写的原始指南。
- **Markdown 教程** 一个开源网站，允许你在网络浏览器中尝试 Markdown。
- **Awesome Markdown** Markdown 工具和学习资源列表。
- **Markdown 排版** 一个多部分系列，描述了使用 pandoc 和 ConTeXt 排版 Markdown 文档的生态系统。

Markdown Cheat Sheet

This Markdown cheat sheet provides a quick overview of all the Markdown syntax elements. It can't cover every edge case, so if you need more information about any of these elements, refer to the reference guides for basic syntax and extended syntax.

基本语法 **Basic Syntax**

These are the elements outlined in John Gruber's original design document. All Markdown applications support these elements.

Heading

Syntax:

```
1 # H1
2 ## H2
3 ### H3
```

Bold

Syntax:

```
1 **bold text**
```

Italic

Syntax:

```
1 *italicized text*
```

Blockquote

Syntax:

```
1 > blockquote
```

Ordered List

Syntax:

```
1 1. First item
2 2. Second item
3 3. Third item
```


Unordered List

Syntax:

```
1 - First item
2 - Second item
3 - Third item
```

Code

Syntax:

```
1 `code`
```

Horizontal Rule

Syntax:

```
1 ---
```

Link

Syntax:

```
1 [Markdown Guide](https://www.markdownguide.org)
```

Image

Syntax:

```
1 ![alt text](https://www.markdownguide.org/assets/images/tux.png)
```

扩展语法 **Extended Syntax**

Syntax:

These elements extend the basic syntax by adding additional features. Not all Markdown applications support these elements.

Table

Syntax:

```

1 | Syntax | Description |
2 | ----- | ----- |
3 | Header | Title |
4 | Paragraph | Text |

```

Fenced Code Block

Syntax:

```

1 ```
2 {
3   "firstName": "John",
4   "lastName": "Smith",
5   "age": 25
6 }
7 ```

```

OR

```

1 ~~~
2 {
3   "firstName": "John",
4   "lastName": "Smith",
5   "age": 25
6 }
7 ~~~

```

Footnote

Syntax:

```

1 Here's a sentence with a footnote. [1]
2
3 [1]: This is the footnote.

```

Heading ID

Syntax:

```

1 ### My Great Heading {#custom-id}

```

Definition List

Syntax:

```
1 term
2 : definition
```

Strikethrough

Syntax:

```
1 ~~The world is flat.~~
```

Task List

Syntax:

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

Emoji

Syntax:

```
1 That is so funny! :joy:
2
3 (See also [Copying and Pasting Emoji](https://www.markdownguide.org/extended-syntax/#copying-and-pasting-emoji))
```

Highlight

Syntax:

```
1 I need to highlight these ==very important words==.
```

Subscript

Syntax:

```
1 H~2~O
```

Superscript

Syntax:

```
1 X^2^
```

Basic Syntax 基本语法

概述 Overview

几乎所有 Markdown 应用程序都支持原始 Markdown 设计文档中概述的基本语法。Markdown 处理器之间存在细微的差异和出入，这些差异和出入在可能的情况下都会以内联方式注明。

标题 Headings

要创建标题，请在单词或短语前面添加井号 (#)。井号的数量代表了标题的级别。例如，添加三个井号即创建一个三级标题 (<h3>) (例如：### My Header)。

```
1 # Heading level 1
2 ## Heading level 2
3 ### Heading level 3
4 #### Heading level 4
5 ##### Heading level 5
6 ##### Heading level 6
```

备用语法

```
1 一级标题
2 =====
3
4 二级标题
5 -----
```

标题最佳实践

当井号 (#) 和标题文本之间没有空格时，各 Markdown 应用程序的处理方式是不一样的。为了兼容考虑，请在井号和标题文本之间添加一个空格。

段落 Paragraphs

要创建段落，请使用空白行将一行或多行文本进行分隔。

```
1 I really like using Markdown.  
2  
3 I think I'll use it to format all of my documents from now on.
```

段落最佳实践

除非段落在列表中，否则不要用空格（spaces）或制表符（tabs）缩进段落。

换行 Line Breaks

在一行的末尾添加两个或多个空格，然后按回车键（return），即可创建一个换行（line break）或新行（
）。

```
1 This is the first line.  
2 And this is the second line.
```

渲染效果如下：

This is the first line.
And this is the second line.

换行（Line Break）用法的最佳实践

几乎每个 Markdown 应用程序都支持两个或多个空格进行换行（称为“结尾空格（trailing whitespace）”）的方式，但这是有争议的，因为很难在编辑器中直接看到空格，并且很多人在每个句子后面都会有意或无意地添加两个空格。由于这个原因，你可能需要使用除结尾空格以外的其它方式来进行换行。如果你所使用的 Markdown 应用程序支持 HTML 的话¹，你可以使用 HTML 的
 标签来实现换行。

为了兼容性，请在行尾添加“结尾空格”或 HTML 的
 标签来实现换行。

还有两种其他方式我并不推荐使用。CommonMark 和其它几种轻量级标记语言支持在行尾添加反斜杠（\）的方式实现换行，但是并非所有 Markdown 应用程序都支持此种方式，因此从兼容性的角度来看，不推荐使用。并且至少有两种轻量级标记语言支持无须在行尾添加任何内容，只须键入回车键（return）即可实现换行。

¹Pandoc 转换程序不支持
 换行。

```
1 First line with two spaces after.
2 And the next line.
3
4 First line with the HTML tag after.<br>
5 And the next line.
```

渲染效果如下：

First line with two spaces after.

And the next line.

First line with the HTML tag after. And the next line.

强调 **Emphasis**

通过将文本设置为粗体或斜体来强调其重要性。

粗体 **Bold**

要加粗文本，请在单词或短语的前后各添加两个星号（asterisks）或下划线（underscores）。如需加粗一个单词或短语的中间部分用以表示强调的话，请在要加粗部分的两侧各添加两个星号（asterisks）。

```
1 I just love bold text.
2
3 I just love bold text.
4
5 Loveisbold
```

渲染结果如下：

I just love **bold text**.

I just love **bold text**.

Love**is**bold

粗体最佳实践

Markdown 应用程序在如何处理单词或短语中间的下划线上并不一致。为兼容考虑，在单词或短语中间部分加粗的话，请使用星号（asterisks）。

斜体 **Italic**

要用斜体显示文本，请在单词或短语前后添加一个星号（**asterisk**）或下划线（**underscore**）。要斜体突出单词的中间部分，请在字母前后各添加一个星号，中间不要带空格。

```
1 Italicized text is the *cat's meow*.
2
3 Italicized text is the _cat's meow_.
4
5 A*cat*meow
```

渲染结果如下：

Italicized text is the *cat's meow*.

Italicized text is the *cat's meow*.

A*cat*meow

斜体最佳实践

Markdown 的众多应用程序在如何处理单词中间的下划线上意见不一致。为了兼容起见，请用星号标注单词中间的斜体来表示着重。

粗体和斜体 **Bold and Italic**

要同时用粗体和斜体突出显示文本，请在单词或短语的前后各添加三个星号或下划线。要加粗并用斜体显示单词或短语的中间部分，请在要突出显示的部分前后各添加三个星号，中间不要带空格。

```
1 This text is ***really important***.
2
3 This text is ___really important___.
4
5 This text is __*really important*__.
6
7 This text is **_really important_**.
8
9 This is really***very***important text.
```

渲染结果如下：

This text is ***really important***.

This text is ***really important***.

This text is ***really important***.

This text is ***really important***.

This is really***very***important text.

粗体和斜体最佳实践

Markdown 应用程序在处理单词或短语中间添加的下划线上并不一致。为了实现兼容性，请使用星号将单词或短语的中间部分加粗并以斜体显示，以示重要。

块引用 **Blockquotes**

要创建块引用，请在段落前添加一个 > 符号。

```
1 > Dorothy followed her through many of the beautiful rooms in her
   castle.
```

渲染效果如下：

```
Dorothy followed her through many of the beautiful rooms in her castle.
```

多个段落的块引用

块引用可以包含多个段落。为段落之间的空白行各添加一个> 符号。

```
1 > Dorothy followed her through many of the beautiful rooms in her
   castle.
2 >
3 > The Witch bade her clean the pots and kettles and sweep the floor and
   keep the fire fed with wood.
```

渲染效果如下：

```
Dorothy followed her through many of the beautiful rooms in her castle.

The Witch bade her clean the pots and kettles and sweep the floor and keep the fire fed with
wood.
```

嵌套块引用 (**Nested Blockquotes**)

块引用可以嵌套。在要嵌套的段落前添加一个 >> 符号。

```
1 > Dorothy followed her through many of the beautiful rooms in her
   castle.
2 >>
```



```
3 >> The Witch bade her clean the pots and kettles and sweep the floor
    and keep the fire fed with wood.
```

渲染效果如下：

Dorothy followed her through many of the beautiful rooms in her castle.

The Witch bade her clean the pots and kettles and sweep the floor and keep the fire fed with wood.

带有其它元素的块引用（**Blockquotes with Other Elements**）

块引用可以包含其他 Markdown 格式的元素。并非所有元素都可以使用，你需要进行实验以查看哪些元素有效。

```
1 > #### The quarterly results look great!
2 >
3 > - Revenue was off the chart.
4 > - Profits were higher than ever.
5 >
6 > *Everything* is going according to **plan**.
```

渲染效果如下：

The quarterly results look great!

- Revenue was off the chart.
- Profits were higher than ever.

Everything is going according to **plan**.

Blockquotes 最佳实践

在 blockquotes 的前后各加上一空行。

列表 Lists

你可以将多个条目组织成有序或无序列表。

有序列表 (Ordered Lists)

要创建有序列表，请在每个列表项前添加数字并紧跟一个英文句点。数字不必按数学顺序排列，但是列表应当以数字 1 起始。

```
1 1. First item
2 2. Second item
3 3. Third item
4 4. Fourth item
5   1. Indented item
6   2. Indented item
7 5. Fourth item
```

渲染效果:

1. First item
2. Second item
3. Third item
4. Fourth item
 1. Indented item
 2. Indented item
5. Fourth item

有序列表 (**Ordered List**) 用法的最佳实践 CommonMark 和其它几种轻量级标记语言可以让你使用括号 () 作为分隔符 (例如 1) First item)，但并非所有的 Markdown 应用程序都支持此种用法，因此，从兼容的角度来看，此用法不推荐。为了兼容起见，请只使用英文句点作为分隔符。

无序列表 (Unordered Lists)

要创建无序列表，请在每个列表项前面添加破折号 (-)、星号 (*) 或加号 (+)。缩进一个或多个列表项可创建嵌套列表。

```
1 - First item
2 - Second item
3 - Third item
4 - Fourth item
5
6 * First item
7 * Second item
8 * Third item
9 * Fourth item
10
11 + First item
```

```
12 + Second item
13 + Third item
14 + Fourth item
15
16 - First item
17 - Second item
18 - Third item
19     - Indented item
20     - Indented item
21 - Fourth item
```

渲染效果:

- First item
- Second item
- Third item
- Fourth item
- First item
- Second item
- Third item
- Fourth item
- First item
- Second item
- Third item
- Fourth item
- First item
- Second item
- Third item
 - Indented item
 - Indented item
- Fourth item

以数字开头的无序列表项 如果你需要以数字开头并且紧跟一个英文句号（也就是 . ）的无序列表项，则可以使用反斜线（\）来转义这个英文句号。

```
1 - 1968\. A great year!
2 - I think 1969 was second best.
```

渲染效果如下：

- 1968. A great year!
- I think 1969 was second best.

无序列表（**Unordered List**）用法的最佳实践 Markdown 应用程序在如何处理同一列表中混用不同分隔符上并不一致。为了兼容起见，请不要在同一个列表中混用不同的分隔符，最好选定一种分隔符并一直用下去。

在列表中添加元素

要在保留列表连续性的同时在列表中添加另一种元素，请将该元素缩进四个空格或一个制表符，如下例所示：

段落 列表中加入段落：

```
1 * This is the first list item.
2 * Here's the second list item.
3
4     I need to add another paragraph below the second list item.
5
6 * And here's the third list item.
```

渲染效果如下：

- This is the first list item.
- Here' s the second list item.
- I need to add another paragraph below the second list item.
- And here' s the third list item.

引用 列表中加入引用：

```
1 * This is the first list item.
2 * Here's the second list item.
3
4     > A blockquote would look great below the second list item.
5
6 * And here's the third list item.
```

渲染效果如下：

- This is the first list item.
- Here' s the second list item.

A blockquote would look great below the second list item.
- And here' s the third list item.

代码块 代码块通常缩进四个空格或一个制表符。当它们在列表中时，将它们缩进八个空格或两个制表符。

```
1 1. Open the file.
2 2. Find the following code block on line 21:
3
4     <html>
5     <head>
6     <title>Test</title>
7     </head>
8
9 3. Update the title to match the name of your website.
```

渲染效果如下：

1. Open the file.
2. Find the following code block on line 21:

```
1 <html>
2 <head>
3 <title>Test</title>
4 </head>
```

3. Update the title to match the name of your website.

图片 列表中加入图片：

```
1 1. Open the file containing the Linux mascot.
2 2. Marvel at its beauty.
3
4     ![Tux, the Linux mascot](snippets/../images/tux.png)
5
6 3. Close the file.
```

渲染效果如下：

1. Open the file containing the Linux mascot.

2. Marvel at its beauty.



Figure 2: Tux, the Linux mascot

3. Close the file.

列表 你可以在有序列表中嵌套无序列表，反之亦然。

```
1 1. First item
2 2. Second item
3 3. Third item
4   - Indented item
5   - Indented item
6 4. Fourth item
```

渲染效果如下：

1. First item
2. Second item
3. Third item
 - Indented item
 - Indented item
4. Fourth item

代码块 **Code Blocks**

代码块（Code blocks）通常采用四个空格或一个制表符缩进。当它们被放在列表中时，请将它们缩进八个空格或两个制表符。要创建不缩进行的代码块，请使用 围栏代码块 Fenced Code Blocks。

```
1 The famous helloworld program for C language is :
2
3     #include <stdio.h>
4
5     int main(int argc, const char * argv[]) {
6         printf("Hello, World!\n");
7         return 0;
8     }
```

渲染效果如下：

The famous helloworld program for C language is :

```
1 #include <stdio.h>
2
3 int main(int argc, const char * argv[]) {
4     printf("Hello, World!\n");
5     return 0;
6 }
```

转义反引号

如果你要表示为代码的单词或短语包含一个或多个反引号，你可以通过用双反引号 (") 将单词或短语括起来来转义它。

```
1 ``在你的 Markdown 文件中使用 `code`。``
```

渲染效果如下：

在你的Markdown 文件中使用 `code`。

代码块

要创建代码块，请将块的每一行至少缩进四个空格或一个制表符。

```
1     <html>
2     <head>
3     </head>
4     </html>
```

渲染效果如下：

```
1 <html>
2   <head>
3   </head>
4 </html>
```

分隔线 **Horizontal Rules**

要创建分隔线，请在单独一行上使用三个或多个星号 (***)、破折号 (---) 或下划线 (___)，并且不能包含其他内容。

```
1 ***
2
3 ---
4
5 _ _ _ _ _
```

以上三个分隔线的渲染效果看起来都一样：



分隔线最佳实践

为了兼容性，请在分隔线的前后均添加空白行。

链接 **Links**

要创建链接，请将链接文本括在方括号（例如 [Duck Duck Go]）中，后面紧跟着括在圆括号中的 URL（例如 (https://duckduckgo.com)）。

```
1 My favorite search engine is [Duck Duck Go](https://duckduckgo.com).
```

渲染效果如下：

My favorite search engine is Duck Duck Go.

添加标题

您可以选择为链接添加标题。当用户将鼠标悬停在链接上时，它将显示为工具提示。要添加标题，请在 URL 后用引号将其括起来。

```
1 My favorite search engine is [Duck Duck Go](https://duckduckgo.com "The best search engine for privacy").
```

渲染效果如下：

My favorite search engine is Duck Duck Go.

URL 和电子邮件地址

要快速将 URL 或电子邮件地址变成链接，请用尖括号将其括起来。

```
1 <https://www.markdownguide.org>
2 <fake@example.com>
```

渲染效果如下：

<https://www.markdownguide.org> fake@example.com

格式化链接

要强调链接，请在方括号和圆括号前后添加星号。要将链接表示为代码，请在方括号中添加反引号。

```
1 I love supporting the **[EFF](https://eff.org)**.
2 This is the *[Markdown Guide](http://markdownguide.cn)*.
3 See the section on [code](#code).
```

渲染效果如下：

I love supporting the **EFF**. This is the *Markdown Guide*. See the section on `code`.

引用式链接

假设您将一个 URL 作为标准 URL 链接添加到一个段落，并且在 Markdown 中看起来像这样：

```
1 In a hole in the ground there lived a hobbit. Not a nasty, dirty, wet
  hole, filled with the ends
2 of worms and an oozy smell, nor yet a dry, bare, sandy hole with
  nothing in it to sit down on or to
3 eat: it was a [hobbit-hole](https://en.wikipedia.org/wiki/Hobbit#
  Lifestyle "Hobbit lifestyles"), and that means comfort.
```

虽然它可能指向有趣的附加信息，但显示的 URL 除了使其更难阅读之外，实际上并没有给现有原始文本增加太多内容。要解决这个问题，您可以像这样格式化 URL：

```
1 In a hole in the ground there lived a hobbit. Not a nasty, dirty, wet
  hole, filled with the ends
2 of worms and an oozy smell, nor yet a dry, bare, sandy hole with
  nothing in it to sit down on or to
3 eat: it was a [hobbit-hole][1], and that means comfort.
4
5 [1]: <https://en.wikipedia.org/wiki/Hobbit#Lifestyle> "Hobbit
  lifestyles"
```

在上面的两个实例中，呈现的输出将是相同的：

In a hole in the ground there lived a hobbit. Not a nasty, dirty, wet hole, filled with the ends of worms and an oozy smell, nor yet a dry, bare, sandy hole with nothing in it to sit down on or to eat: it was a hobbit-hole, and that means comfort.

链接最佳实践 Markdown 应用程序对于如何处理 URL 中间的空格没有达成一致。为了兼容性，请尝试使用 %20 对 URL 中的任何空格进行 URL 编码。或者，如果您的 Markdown 应用程序支持 HTML，则可以使用 a HTML 标签。

URL 中间的括号也可能带来问题。为了兼容性，尝试使用 %28 对左括号 (() 进行 URL 编码，使用 %29 对右括号)) 进行 URL 编码。或者，如果 Markdown 应用程序支持 HTML，可以使用 a HTML 标签。

图片 Images

要添加图片的话，首先请添加感叹号 (!)，然后紧跟着是方括号，方括号中可添加备用文本 (alt text，即图片显示失败后显示此文本)，最后跟着圆括号，圆括号中添加图片资源的路径或 URL。你可以选择在圆括号中的路径或 URL 之后添加标题 (即 title 属性)。

```
1 ![The San Juan Mountains are beautiful!](chapters/basic-syntax/images/
  san-juan-mountains.jpg "San Juan Mountains")
```

渲染效果如下：



Figure 3: The San Juan Mountains are beautiful!

带链接的图片

要为图片添加链接，请先为图片的 Markdown 标记添加一个方括号，然后紧跟着一个圆括号，并在圆括号中添加链接地址。

```
1 [![An old rock in the desert](chapters/basic-syntax/../../images/
  shiprock.jpg "Shiprock, New Mexico by Beau Rogers")](https://www.
  flickr.com/photos/beaurogers/31833779864/in/photolist-Qv3rFw-34mt9F-
  a9Cmfy-5Ha3Zi-9msKdv-o3hgjr-hWpUte-4WMsJ1-KUQ8N-deshUb-vssBD-6CQci6
  -8AFCiD-zsJWT-nNfsgB-dPDwZJ-bn9JGn-5HtSXY-6CUhAL-a4UTXB-ugPum-KUPSo-
  fBLNm-6CUmpy-4WMsc9-8a7D3T-83KJev-6CQ2bK-nNushJ-a78rQH-nw3NvT-7aq2qf
  -8wwBso-3nNceh-ugSKP-4mh4kh-bbeeQH-a7biME-q3PtTf-brFpgb-cg38zw-bXMZc
  -nJPELD-f58Lmo-bXMYG-bz8AAi-bxNtNT-bXMYi-bXMY6-bXMYv)
```

渲染效果如下：



转义字符 **Escaping Characters**

要显示原本用于格式化 Markdown 文档的字符，请在字符前面添加反斜杠字符 ()。

```
1 \* 如果没有开头的反斜杠字符的话，这一行将显示为无序列表。
```

渲染效果如下：

* 如果没有开头的反斜杠字符的话，这一行将显示为无序列表。

可做转义的（英文）字符

以下列出的字符都可以通过使用反斜杠字符从而达到转义目的。

字符	名称
\	反斜杠 (backslash)
`	backtick (另请参见 在代码中转义反引号)
*	星号 (asterisk)

字符	名称
<code>_</code>	下划线 (underscore)
<code>{}</code>	花括号 (curly braces)
<code>[]</code>	方括号 (brackets)
<code><></code>	angle brackets
<code>()</code>	圆括号或括号 (parentheses)
<code>#</code>	井号 (pound sign)
<code>+</code>	加号 (plus sign)
<code>-</code>	减号 (minus sign) (也叫连字符 hyphen)
<code>.</code>	句点 (dot)
<code>!</code>	感叹号 (exclamation mark)
<code> </code>	管道符 (pipe) (另请参见在表格中转义管道符)

HTML 标签

许多 Markdown 应用程序允许你在 Markdown 格式文本中使用 HTML 标签。如果你更喜欢某些 HTML 标签而不是 Markdown 语法，这会很有帮助。例如，有些人觉得使用 HTML 标签来处理图片更简单。当你需要更改元素的属性（例如指定文本颜色或更改图片的宽度）时，使用 HTML 也会很有帮助。

要使用 HTML，请将标签放置在 Markdown 格式文件文本中。

```
1 This word is bold. This word is italic.
```

渲染效果如下：

This **word** is bold. This word is italic.

HTML 最佳实践

出于安全原因，并非所有 Markdown 应用程序都支持 Markdown 文档中的 HTML。如有疑问，请查看 Markdown 应用程序的文档。一些应用程序仅支持 HTML 标签的子集。

使用空行来分隔块级 HTML 元素，如 `<div>`、`<table>`、`<pre>` 和 `<p>`，使其与周围内容分开。尽量不要使用制表符或空格缩进标签，因为这会影响格式。

您不能在块级 HTML 标签中使用 Markdown 语法。例如，`<p>italic and bold</p>` 将不起作用。

Extended Syntax 扩展语法

概述 Overview

原始 Markdown 设计文档中概述的基本语法添加了许多日常所需元素，但对某些人来说还不够。这就是扩展语法发挥作用的地方。

一些个人和组织通过添加表格、代码块、语法高亮、URL 自动链接和脚注等其他元素，自行扩展了基本语法。这些元素可以通过使用基于基本 Markdown 语法的轻量级标记语言或向兼容的 Markdown 处理器添加扩展来启用。

可用性

并非所有 Markdown 应用程序都支持扩展语法元素。您需要检查应用程序使用的轻量级标记语言是否支持您想要使用的扩展语法元素。如果不支持，您仍然可以在 Markdown 处理器中启用扩展。

轻量级标记语言

有几种轻量级标记语言是 Markdown 的超集。它们包含基本语法，并通过添加表格、代码块、语法高亮、URL 自动链接和脚注等其他元素对其进行扩展。许多最流行的 Markdown 应用程序都使用以下轻量级标记语言之一

- CommonMark
- GitHub Flavored Markdown (GFM)
- Markdown Extra
- MultiMarkdown
- R Markdown

Markdown 处理器

有数十个 Markdown 处理器可用。其中许多允许您添加启用扩展语法元素的扩展。有关更多信息，请查看处理器的文档。

表格 Tables

如需添加表格，请使用三个或更多个连字符（—）来为每个列创建表头，并使用管道符（|）来分隔每个列。为兼容考虑，你还应该在行的两侧添加管道符。

1	Syntax	Description	
2	-----	-----	
3	Header	Title	
4	Paragraph	Text	

渲染效果:

Syntax	Description
Header	Title
Paragraph	Text

对齐

通过在标题行中的连字符（hyphens）的左侧或右侧或两侧添加冒号（:），可以将对应列中的文本向左或向右或居中对齐。

1	Right	Left	Center	Default	
2	-----:	:-----	:-----:	-----	
3	12	12	12	12	
4	123	123	123	123	
5	1	1	1	1	

渲染效果:

Right	Left	Center	Default
12	12	12	12
123	123	123	123
1	1	1	1

格式化表格中的文本

你可以为表格中的文本设置格式。例如，可以添加链接（links）、代码（code）（注意，只能为单词或短语添加反引号（```），不能添加代码块（code blocks））以及强调（emphasis）。

不支持的格式包括标题（headings）、块引用（blockquotes）、列表（lists）、水平分割线（horizontal rules）、图片（images）或大部分 HTML 标签。

1	Syntax	Description
2	-----	-----
3	Bold	Title
4	Paragraph	<code>Code</code>

渲染效果:

Syntax	Description
Bold	Title
Paragraph	<code>Code</code>

围栏代码块 Fenced Code Blocks

Markdown 的基本语法允许你通过缩进四个空格或一个制表符来创建代码块。如果你觉得不方便，可以试试围栏代码块（fenced code blocks）。根据 Markdown 解析器或编辑器的不同，代码块的前后各行可以使用三个反引号（`````）或三个波浪号（`~~~`）来标记围栏代码块。这有什么优势吗？你不必费力缩进任何行了！

```

1 ```
2 #include <stdio.h>
3
4 int main(int argc, const char * argv[]) {
5     printf("Hello, World!\n");
6     return 0;
7 }
8 ```

```

渲染效果如下:

```

1 #include <stdio.h>
2
3 int main(int argc, const char * argv[]) {
4     printf("Hello, World!\n");
5     return 0;
6 }

```

语法高亮

许多 Markdown 处理器支持对带围栏的代码块进行语法高亮。此功能允许您为代码所用语言添加颜色高亮。要添加语法高亮，请在带围栏的代码块前在反引号旁边指定一种语言。


```
1 ``json
2 {
3   "firstName": "John",
4   "lastName": "Smith",
5   "age": 25
6 }
7 ``
```

渲染效果如下：

```
1 {
2   "firstName": "John",
3   "lastName": "Smith",
4   "age": 25
5 }
```

脚注 **Footnotes**

脚注允许您添加注释和参考，而不会使文档正文杂乱。创建脚注时，您添加脚注参考的位置将出现带链接的上标数字。读者可以单击链接跳转到页面底部的脚注内容。

要创建脚注参考，请在方括号内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。标识符仅将脚注参考与脚注本身关联——在输出中，脚注按顺序编号。

使用方括号内的另一个插入符号和数字以及冒号和文本添加脚注（`[^1]: My footnote.`）。您不必将脚注放在文档末尾。您可以在列表、块引用和表格等其他元素之外的任何位置放置它们。

```
1 Here's a simple footnote,[1] and here's a longer one.[^bignote]
2
3 [1]: This is the first footnote.
4
5 [^bignote]: Here's one with multiple paragraphs and code.
6
7     Indent paragraphs to include them in the footnote.
8
9     `{ my code }`
10
11     Add as many paragraphs as you like.
```

渲染效果如下：

Here's a simple footnote,² and here's a longer one.³

²This is the first footnote.

³Here's one with multiple paragraphs and code.

Indent paragraphs to include them in the footnote.

{ my code }

Add as many paragraphs as you like.

锚链接 **Anchor Links**

许多 Markdown 处理器支持 标题 的自定义 ID——一些 Markdown 处理器会自动添加它们。添加自定义 ID 允许您直接链接到标题并使用 CSS 修改它们。要添加自定义标题 ID，请在与标题同行的花括号中括起自定义 ID。

```
1 ### My Great Heading {#custom-id}
```

输出的 HTML 如下所示：

```
1 <h3 id="custom-id">My Great Heading</h3>
```

你可以为文档中的标题创建一个 标准链接，链接的格式是井号（#）开头，其后面跟着自定义的标题 ID，从而能够链接到这个标题。这类链接通常被称为 锚链接（anchor links）。

```
1 [Heading IDs](#heading-ids)
```

如：锚链接

定义列表 **Definition Lists**

某些 Markdown 解析器允许你创建术语（terms）及其相应的定义（definitions）的列表，即 定义列表（definition lists）。要创建定义列表，请在第一行键入术语，然后在下一行中键入冒号，冒号后跟着空格和此术语的具体定义。

```
1 First Term
2 : This is the definition of the first term.
3
4 Second Term
5 : This is one definition of the second term.
6 : This is another definition of the second term.
```

渲染效果如下：

First Term This is the definition of the first term.

Second Term This is one definition of the second term.

This is another definition of the second term.

任务列表 **Task Lists**

任务列表（Task lists，也称为 checklists 或者 todo lists）允许你创建带有复选框的项目列表。在支持任务列表的 Markdown 应用程序中，复选框将显示在内容旁边。要创建任务列表，请在任务列表项前面添加破折号（-）和中间带空格的方括号（[]）。要选中复选框，请在方括号中间添加一个 x，即（[x]）。

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

渲染效果如下：

- ☒ Write the press release
- ☐ Update the website
- ☐ Contact the media

删除线 **Strikethrough**

你可以贯穿单词的中心放一条横线从而删除这些单词。其效果看起来是这样的：like this。此功能允许你标记某些单词是错误的，不应该出现在文档中。在单词前面和后面分别放置两个波浪号（~~）来表示删除这些单词。

```
1 ~~The world is flat.~~ We now know that the world is round.
```

渲染效果如下所示：

~~The world is flat.~~ We now know that the world is round.

表情符号 **Emoji**

有两种方式可以将表情符号添加到 Markdown 文档中：将表情符号复制并粘贴到 Markdown 格式的文本中，或者键入表情符号的简码（emoji shortcuts）⁴。

复制并粘贴表情符号

在大多数情况下，你可以简单地从 Emojipedia 等来源复制表情符号，然后将其粘贴到文档中。许多 Markdown 应用程序就会自动以 Markdown 格式的文本来显示表情符号。从 Markdown 应用程序导出的 HTML 和 PDF 文件也是可以显示表情符号的。

使用表情符号的简码（**Shortcodes**）

某些 Markdown 应用程序允许你通过键入表情符号的简码（shortcodes）来插入表情符号。简码以冒号开头和结尾，两个冒号中间是表情符号的名称。

⁴调用 pandoc 时需要加选项 `--from markdown+emoji`。如果要转换 tex/pdf，模板中需要支持包，系统中需要支持表情符号的字体，如 Noto Color Emoji，并使用 LuaTeX 编译。

```
1 Gone camping! :tent: Be back soon.
2
3 That is so funny! :joy:
```

渲染效果如下：

Gone camping! :tent: Be back soon.

That is so funny! :joy:

Highlight

This isn't common, but some Markdown processors allow you to highlight text. The result looks like this. To highlight words, use two equal signs (==) before and after the words.

```
1 I need to highlight these ==very important words==.
```

The rendered output looks like this:

I need to highlight these very important words.

Alternatively, if your Markdown application supports HTML, you can use the mark HTML tag.

I need to highlight these very important words.

下标 Subscript

这种情况并不常见，但一些 Markdown 处理器允许你使用下标将一个或多个字符置于正常文本行的下方。要创建下标，请在字符前后使用一个波浪号 (~)。

```
1 H~2~O
```

渲染效果如下：

H₂O

上标 Superscript

这并不常见，但一些 Markdown 处理器允许你使用上标将一个或多个字符放置在正常文本行上方。要创建上标，请在字符前后使用一个插入符号 (^)。

```
1 X^2^
```

渲染效果如下：

X²

自动将 **URL** 转换为链接

许多 Markdown 解析器会自动将 URL 转换为链接。这意味着，即使你没有使用中括号，如果你输入 `http://www.example.com`，你的 Markdown 解析器也会自动将其转换为链接。

`http://www.example.com`

渲染效果如下所示：

<http://www.example.com>

禁止自动将 **URL** 转换为链接

如果你不希望自动将 URL 转换为链接，则可以通过反引号将 URL 表示为代码。

`http://www.example.com`

渲染效果如下所示：

`http://www.example.com`

Pandoc's Markdown

概述 Overview

Pandoc 中支持扩展修订版本的 Markdown 语法。

- 使用 pandoc 中支持的 Markdown 语法用 `-f markdown`
- 使用标准 Markdown 语法用 `-f markdown_strict`

Pandoc 所支持的语法各种对标准 Markdown 语法的扩展可以通过在格式后以 `+EXTENSION` 添加或 `-EXTENSION` 去除，如：

- `-f markdown-footnotes` 表示识别除了 `footnotes` 扩展之外的所有 pandoc Markdown 语法
- `-f markdown_strict+footnotes+pipe_tables` 表示识别标准 Markdown 语法加上 `footnotes` 和 `pipe_tables` 扩展语法。

段落

段落由一行或多行文本组成，后跟一行或多行空行。换行符被视为空格，因此您可以根据需要重新排列段落。如果需要强制换行，请在行尾添加两个或更多空格。

Extension: escaped_line_breaks

反斜杠后跟换行符也是一种硬换行符。注意：在多行和网格表格单元格中，这是创建硬换行符的唯一方法，因为单元格中的尾随空格会被忽略。

```

1  反斜杠后跟换行符 (`\`)\
2  也是一种硬换行符
3
4  +-----+-----+-----+
5  | Fruit      | Price      | Advantages  |
6  +-----+-----+-----+
7  | Bananas    | first line\ | first line\ |
8  |            | next line  | next line   |
9  +-----+-----+-----+
10 | Bananas    | first line\ | first line\ |
11 |            | next line  | next line   |
12 +-----+-----+-----+

```

渲染效果如下：

反斜杠后跟换行符 (\)
也是一种硬换行符

Fruit	Price	Advantages
Bananas	first line next line	first line next line
Bananas	first line next line	first line next line

标题

Pandoc 可以使用两种标题格式：Setext 和 ATX。

Setext 样式

Setext 样式的标题是一行带有“下划线”的文本，其中带有一行 = 符号（对于一级标题）或 - 符号（对于二级标题）。标题文本可能包括内联格式，例如强调和斜体。

```

1  A level-one heading
2  =====
3
4  A level-two heading

```

```
5 -----
6
7 A *formatted* heading
8 =====
9
10 Introduction to [Makedown Guide](https://www.markdownguide.com)
11 -----
```

ATX 格式

ATX 样式标题由 1 到 6 个连续的#符号和一行文本组成，在行尾可能有任意数量的符号。行首的符号#数量即为标题的级别。与 Setext 标题一样，ATX 标题允许内联格式。

```
1 # A level-one heading
2
3 ## A level-two heading
4
5 ### A *formatted* heading
6
7 #### Introduction to [Makedown Guide](https://www.markdownguide.com)
```

Extension: blank_before_header

原始 Markdown 语法不需要标题前有空行。Pandoc 确实需要这个（当然，文档的开头除外）。提出这一要求的原因是，# 很容易意外地出现在一行的开头（可能由于换行）。

```
1 I like several of their flavors of ice cream:
2 # 22, for example, and # 5.
```

Extension: space_in_atx_header

许多 Markdown 实现并不要求 ATX 标题开头的 # 与标题文本之间有空格，因此 # heading 1 和 #heading 1 都算作标题。Pandoc 默认要求# 与标题文本之间有空格，可以用 -f markdown-space_in_atx_header 来取消这个要求。

```
1 #head
2
3 # head
```

编译时运行：`pandoc -f markdown-space_in_atx_header input.md -o output.md`。

标题标识符

Extension: header_attributes

可以在包含标题文本的行末尾使用以下语法为标题分配属性：

```
1 {#identifier .class .class key=value key=value}
```

请注意，虽然此语法允许分配类和键/值属性，但编写者通常不会使用所有这些信息。标识符、类和键/值属性用于 HTML 和基于 HTML 的格式（例如 EPUB 和 slidy）。标识符用于 LaTeX、ConTeXt、Textile、Jira 标记和 AsciiDoc 编写器中的标签和链接锚点。

即使指定了，带有类的标题 `unnumbered` 也不会被编号 `-number-sections`。属性上下文中的单个连字符 (-) 相当于 `unnumbered`，并且在非英语文档中更可取。因此，

```
1 # My heading {-}
```

和

```
1 # My heading {.unnumbered}
```

如果存在 `unnumbered` 和 `unlisted`，则标题将不会包含在目录中。（目前此功能仅适用于某些格式：基于 LaTeX 和 HTML、PowerPoint 和 RTF 的格式。）

```
1 # Heading identifiers in HTML {#foo}
2
3 你可以这样使用：
4
5 [标题 1](#foo)
6
7
8 # 中文标题 identifiers in HTML {#bar}
9
10 你可以这样使用：
11
12 [标题 2](#bar)
```

Extension: implicit_header_references

Pandoc 的行为就像每个标题都定义了引用链接（reference links）一样。因此，要链接到标题，只需要直接加上 []，如：

```
1 # Heading identifiers in HTML
2
3 你可以简单地写
4
```



```
5 [Heading identifiers in HTML]
6
7 或者
8
9 [Heading identifiers in HTML][]
10
11 或者
12
13 [the section on heading identifiers][heading identifiers in
14 HTML]
15
16 而不是明确给出标识符：
17
18 [Heading identifiers in HTML](#heading-identifiers-in-html)
19
20
21 # 中文标题 identifiers in HTML
22
23 你可以简单地写
24
25 [中文标题 identifiers in HTML]
26
27 或者
28
29 [中文标题 identifiers in HTML][]
30
31 或者
32
33 [the section on heading identifiers][中文标题 identifiers in
34 HTML]
35
36 而不是明确给出标识符：
37
38 [中文标题 identifiers in HTML](#中文标题-identifiers-in-html)
```

如果有多个标题具有相同的文本，则相应的引用将仅链接到第一个标题，并且您需要使用明确的链接来链接到其他标题，如上所述。

与常规引用链接一样，这些引用不区分大小写。

显式链接引用定义始终优先于隐式标题引用。

引用块 **Block quotations**

Markdown 使用电子邮件约定来引用文本块。块引用是指一个或多个段落或其他块元素（例如列表或标题），每行前面都有一个 > 字符和一个可选的空格。（引用 > 不必从左边距开始，但缩进不应超过三个空格。）

块引用中可以包含其他块引用的块元素。也就是说，块引用可以嵌套：

```
1 > This is a block quote.
2 >
3 > > A block quote within a block quote.
```

渲染效果如下：

```
This is a block quote.
  A block quote within a block quote.
```

注意：上面的嵌套块引用中需要加入空行。不加入空行则不会变成块引用：

```
1 This is NOT a block quote.
2 Lorem Ipsum 始于西塞罗 (Cicero) 在公元前45年作的 “de Finibus Bonorum
  et Malorum” (善恶之尽) 里 1.10.32 和 1.10.33 章节。这本书是一本关于
  道德理论的论述，曾在文艺复兴时期非常流行。Lorem Ipsum 的第一行是：
3 > Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do
  eiusmod tempor incididunt ut labore et dolore magna aliqua.
```

原始 Markdown 语法并不要求块引用前有空行。Pandoc 则要求这样做（当然，文档开头除外）。这样做的原因是，a 很容易意外地 > 出现在行首（例如由于换行）。因此，除非 `markdown_strict` 使用这种格式，否则以下代码在 Pandoc 中不会生成嵌套的块引用：

```
1 > This is a block quote.
2 >> Not nested, since `blank_before_blockquote` is enabled by default
```

代码块 **Verbatim (code) blocks**

缩进的代码块 **Indented code blocks**

缩进四个空格（或一个制表符）的文本块将被视为逐字文本：也就是说，特殊字符不会触发特殊格式，所有空格和换行符都会保留。例如：

```
1     if (a > 3) {
2         moveShip(5 * gravity, DOWN);
3     }
```

初始（四个空格或一个制表符）缩进不被视为逐字文本的一部分，并将在输出中删除。

注意：逐字文本中的空白行不必以四个空格开头。

围栏代码块 **Fenced code blocks**

Extension: fenced_code_blocks

除了标准缩进的代码块外，Pandoc 还支持带围栏的代码块。这些代码块以一行三个或更多波浪号(~)开始，以一行波浪号结束，波浪号的长度必须至少与起始行一样长。这些行之间的所有内容都被视为代码。无需缩进：

```
1 if (a > 3) {  
2   moveShip(5 * gravity, DOWN);  
3 }
```

与常规代码块一样，围栏代码块必须用空行与周围文本分隔开。

如果代码本身包含一行波浪号或反引号，则只需在开始和结束处使用一排较长的波浪号或反引号：

```
1 ~~~~~  
2 code including tildes  
3 ~~~~~
```

Extension: backtick_code_blocks

与 `fenced_code_blocks` 相同，但使用反引号(`)而不是波浪号(~)。

Extension: fenced_code_attributes

或者，您可以使用以下语法将属性附加到隔离或反引号代码块：

```
1 HelloWorld in Python:  
2  
3 ```{#helloworld .python .numberLines startFrom="100"}  
4 print('hello,world')  
5 ```  
6  
7 The `numberLines` (or `number-lines`) class will cause the lines of the  
   code block to be numbered, starting with 1 or the value of the `  
   startFrom` attribute. The `lineAnchors` (or `line-anchors`) class  
   will cause the lines to be clickable anchors in HTML output.  
8  
9 ```{#hello .c .numberLines startFrom="101" .lineAnchors}  
10 #include <stdio.h>  
11  
12 int main(int argc, const char * argv[]) {  
13   printf("Hello, World!\n");  
14   return 0;  
15 }
```

这里 `#helloworld` 是一个标识符，`python` 和 `numberLines` 是类，并且 `startFrom` 是一个值为 100 的属性。某些输出格式可以使用此信息进行语法高亮显示。目前，唯一使用此信息的输出格式是 HTML、LaTeX、Docx、Ms 和 PowerPoint。如果您的输出格式和语言支持高亮显示，则上面的代码块将高亮显示，并带有编号行。（要查看支持哪些语言，请输入 `pandoc --list-highlight-languages`。）

`numberLines`（或 `number-lines`）类将使代码块的行以 1 或 `startFrom` 的属性值开始进行编号。`lineAnchors`（或 `line-anchors`）类将使这些行在 HTML 输出中成为可点击的锚点。

要阻止所有突出显示，请使用 `--no-highlight` 标志。要设置突出显示样式，请使用 `--highlight-style`。有关突出显示的更多信息，请参阅下文的“语法突出显示”。

行块 **Line Blocks**

行块是由一系列以竖线（|）开头、后跟空格的行组成的序列。行的划分以及前导空格将在输出中保留；否则，这些行将被格式化为 Markdown 格式。这对于诗句和地址很有用：

```
1 | The limerick packs laughs anatomical
2 | In space that is quite economical.
3 |     But the good ones I've seen
4 |     So seldom are clean
5 | And the clean ones so seldom are comical
6
7 | 200 Main St.
8 | Berkeley, CA 94718
```

如果需要，可以将行硬换行，但续行必须以空格开头。

```
1 | The Right Honorable Most Venerable and Righteous Samuel L.
2 | Constable, Jr.
3 | 200 Main St.
4 | Berkeley, CA 94718
```

内容中允许使用行内格式（例如强调）（但不能跨越行边界）。块级格式（例如块引用或列表）无法识别。

此语法借用自 `reStructuredText`。

内联格式 **Inline formatting**

Extension: intraword_underscores

由于 `_` 有时会在单词和标识符内部使用，因此 `pandoc` 不会将 `_` 字母数字字符包围的 解释为强调标记。如果只想强调单词的一部分，请使用 `*`：

```
1 feas*ible*, not feas*able*.
```

`Pandoc` 默认是启用该扩展的，如果要想将单词内部的 `_` 解析强调语法，需要取消该扩展：`-f markdown-intraword_underscores`。

Extension: strikeout

要用水水平线划掉一段文本，请以 开始和结束该部分 `~~`。例如，

```
1 This ~~is deleted text.~~
```

Extension: superscript 和 subscript

上标可以用字符包围上标文本来书写[^]；下标也可以用字符包围下标文本来书写_~。例如，

```
1 H~2~0 is a liquid. 2^10^ is 1024.
```

^{^...^}或之间的文本_{~...~}不能包含空格或换行符。如果上标或下标文本包含空格，则必须使用反斜杠转义这些空格。（这是为了防止在日常使用_~和[^]时意外地将文本变为上标或下标[^]，以及与脚注产生不良交互。）因此，如果您希望下标中包含字母 P 和 “a cat”，请使用 `P~a\ cat~`，而不是 `P~a cat~`。

链接 **Links**

自动链接 **Automatic Links**

如果将 URL 或电子邮件地址放在尖括号中，它将变成链接：

```
1 <https://google.com>
2 <sam@green.eggs.ham>
```

内联链接 **Inline Links**

内联链接由方括号内的链接文本和括号内的 URL 组成。(URL 后面也可以跟着链接标题, 并用引号引起来。)

```
1 This is an [inline link](/url), and here's [one with
2 a title](https://fsf.org "click here for a good time!").
```

方括号和圆括号之间不能有空格。链接文本可以包含格式 (例如强调), 但标题不能。

内联链接中的电子邮件地址无法自动检测, 因此必须加上前缀 `mailto:`:

```
1 [Write me!](mailto:sam@green.eggs.ham)
```

引用链接 **Reference Links**

Extension: shortcut_reference_links

在快捷引用链接中, 第二对括号可以完全省略:

```
1 See [my website].
2
3 [my website]: http://foo.bar.baz
```

内部链接 **Internal Links**

要链接到同一文档的其他部分, 请使用自动生成的标识符 (请参阅标题标识符)。例如:

```
1 See the [Introduction].
```

或者

```
1 See the [Introduction](#introduction).
```

或者

```
1 See the [Introduction].
2
3 [Introduction]: #introduction
```

内部链接目前支持 HTML 格式 (包括 HTML 幻灯片和 EPUB)、LaTeX 和 ConTeXt。

使用元数据 **Metadata**

YAML 最初是 “Yet Another Markup Language” 的缩写，但后来为了强调其数据导向（而非标记语言）的特性，官方将其重新定义为 “YAML Ain’ t Markup Language”（一种递归缩写，即 “YAML 不是标记语言”）。最初设计时，YAML 被视作类似 XML 的标记语言，但后来转向更简洁的数据描述格式，专注于存储和传输结构化数据（如配置、API 请求等）。与 XML（标记语言）不同，YAML 避免使用标签（< >），而是依赖缩进和符号表示结构。

在 Pandoc 中，YAML 元数据是指文档开头的一段 YAML 格式的内容，用于描述文档的属性和元信息。通过读取 Pandoc 标记文件中的 YAML 元数据，我们可以获取文档的各种属性，如标题、作者、日期等，以及自定义的元信息。

读取 Pandoc 标记文件中的 YAML 元数据可以通过使用 Pandoc 提供的命令行参数或 API 来实现。以下是一个示例命令行使用方式：

```
1 pandoc --metadata-file=metadata.yaml input.md -o output.html
```

上述命令中，metadata.yaml 是包含 YAML 元数据的文件，input.md 是待转换的标记文件，output.html 是转换后的输出文件。

在实际应用中，读取 Pandoc 标记文件中的 YAML 元数据可以用于自动化文档处理、生成静态网页、构建电子书等场景。通过提取元数据，我们可以根据文档属性进行个性化处理，如根据作者生成不同样式的页面，根据日期进行归档等。

Pandoc 命令中的几个相关选项：

--metadata-file=FILE Read metadata from the supplied YAML (or JSON) file. This option can be used with every input format, but string scalars in the metadata file will always be parsed as Markdown. (If the input format is Markdown or a Markdown variant, then the same variant will be used to parse the metadata file; if it is a non-Markdown format, pandoc’s default Markdown extensions will be used.) This option can be used repeatedly to include multiple metadata files; values in files specified later on the command line will be preferred over those specified in earlier files. Metadata values specified inside the document, or by using -M, overwrite values specified with this option. The file will be searched for first in the working directory, and then in the metadata subdirectory of the user data directory (see -data-dir).

-M KEY[=VAL], --metadata=KEY[:VAL] Set the metadata field KEY to the value VAL. A value specified on the command line overrides a value specified in the document using YAML metadata blocks. Values will be parsed as YAML boolean or string values. If no value is specified, the value will be treated as Boolean true. Like -variable, -metadata causes template variables to be set. But unlike -variable, -metadata affects the metadata of the underlying document (which is accessible from filters and may be printed in some output formats) and metadata values will be escaped when inserted into the template.

-V KEY[=VAL], --variable=KEY[:VAL] Set the template variable KEY to the string value VAL when rendering the document in standalone mode. If no VAL is specified, the key will be given the value true. Structured values (lists, maps) cannot be assigned using this option, but they can be assigned in the variables section of a defaults file.

注意: `CJKmainfont="方正楷体_GBK"` 写入 `metadata.yaml` 时, 会被解析成 “方正楷体_GBK”, 导致编译出错。

Hacks

下划线 **Underline**

Markdown 原生不支持下划线, 可能是因为网页文本中的下划线通常表示超链接。然而, 在写论文或报告时, 可能需要使用下划线。一些应用 (如 Bear 和 Simplenote) 支持文本下划线, 但如果你的 Markdown 处理器支持 HTML, 可以使用标签:

```
1 一些文字 <ins>将被加下划线</ins>。
```

渲染效果如下:

一些文字 将被加下划线。

文字居中 **Center**

Markdown 没有文本对齐的语法, 但可以使用 HTML 标签

(已废弃) 或 CSS 解决:

```
1 <p style="text-align:center">这段文字居中显示。</p>
```

渲染效果如下:

这段文字居中显示。

数学公式 **Mathematical Expressions**

Inline math

```
1 Euler's formula is remarkable:  $e^{i\pi} + 1 = 0$ .
```


渲染效果如下：

Euler’ s formula is remarkable: $e^{i\pi} + 1 = 0$.

Display Math

```
1 When $a \neq 0$, there are two solutions to  $(ax^2 + bx + c = 0)$  and
   they are
2
3  $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ 
```

渲染效果如下：

When $a \neq 0$, there are two solutions to $(ax^2 + bx + c = 0)$ and they are

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

交叉引用 Cross Referencing

使用 **pandoc-crossref**

安装 **pandoc** 和 **pandoc-crossref**

要想使用 pandoc-crossref，需要安装相互兼容的 pandoc 和 pandoc-crossref；否则 Pandoc 会提示兼容性问题。如：pandoc-3.6.4 和 pandoc-crossref-v0.3.18.2。

- pandoc 下载地址
- pandoc-crossref 下载地址

解压缩 pandoc 可执行文件到一个目录中，如 `~/.local/bin/pandoc`。解压缩 pandoc-crossref 可执行文件到 `~/.local/bin/pandoc-crossref` 中。帮助文档拷贝到 `~/.local/share/man/man1` 中，目录结构如下：

```
1 $ tree ~/.local/share/man
2 /home/USERNAME/.local/share/man
3 └─ man1
4     ├── pandoc.1.gz
5     ├── pandoc-crossref.1
6     ├── pandoc-lua.1.gz
7     └── pandoc-server.1.gz
```

运行 **pandoc-crossref**

首先，需要设置环境变量 `PATH`：

```
export PATH=~/.local/bin:$PATH
```

在 macOS 中，首次运行 `pandoc --filter pandoc-crossref` 时会提示：“无法打开“pandoc-crossref”，因为无法验证开发者。”点击“取消”，然后在“系统偏好设置-安全性与隐私-通用”中“仍然允许”通过验证。

直接转换成 **PDF** 时的错误处理

直接转换时出现以下错误：

```
1 Error producing PDF.
2 ! Undefined control sequence.
3 l.340 (\passthrough
```

可以先转换成 `.tex` 文件，然后在文件的导言区中加入：

```
1 \usepackage{listings}
2 \newcommand{\passthrough}[1]{#1}
```

或者：添加 `-M listings` 选项⁵。

Cross Referencing

```
1 When $a \ne 0$, there are two solutions to $(ax^2 + bx + c = 0)$ and
   they are
2
3 $$ x = \{-b \pm \sqrt{b^2-4ac} \over 2a\} $$ {#eq:quad}
4
5 We can solve the equation $x^2 - x - 2 = 0$ based on [eq:quad]:
6 $x_1=-1, x_2=2$.
```

渲染效果如下：

When $a \neq 0$, there are two solutions to $(ax^2 + bx + c = 0)$ and they are

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (0.1)$$

We can solve the equation $x^2 - x - 2 = 0$ based on eq. 0.1: $x_1 = -1, x_2 = 2$.

⁵pandoc-crossref と Eisvogel を併用する

参考资料 **References**

- [Markdown Guide](#)
- [Markdown 指南中文版](#)
- [A writer's guide to Pandoc's Markdown](#)
- [Introducing Markdown and Pandoc](#)

