
Markdown Guide

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。

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”）。

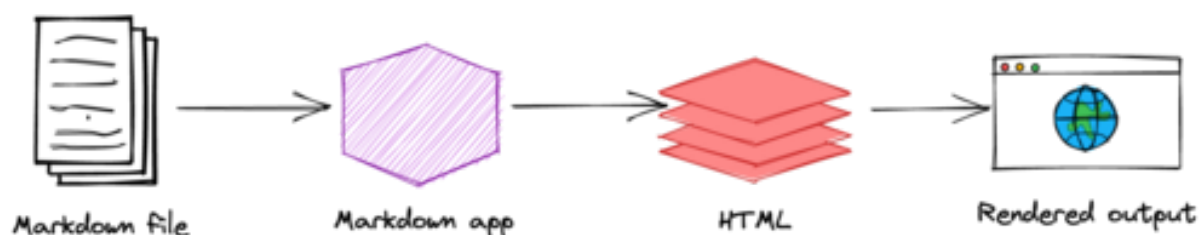


Figure 1: Markdown Flowchart

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.

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)
```

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 }
```

```
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

总览 Overview

几乎所有 Markdown 应用程序都支持原始 Markdown 设计文档中列出的 Markdown 基本语法。但是，Markdown 处理程序之间存在着细微的变化和差异，我们都会尽可能给出提示。

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
```

标题 (**Heading**) 用法的最佳实践 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 (**Paragraph**) 用法的最佳实践 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。

```
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。

¹Pandoc 转换程序不支持
 换行。

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側各添加两个星号（asterisks）。

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

粗体（**Bold**）用法最佳实践 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 Italicized text is the cat's meow.
```

斜体（**Italic**）用法的最佳实践 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 **Bold** 和斜体 **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 This text is really important.
```

粗体（**Bold**）和斜体（**Italic**）用法的最佳实践 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。

在列表中添加列表项 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：

```
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.

```
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 **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。另外见扩展语法 **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 }
```

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（**Horizontal Rule**）用法最佳实践 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：

My favorite search engine is Duck Duck Go.

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 2: 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](/assets/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)
_	下划线 (underscore)
{ }	花括号 (curly braces)

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

HTML 标签

大多 Markdown 应用程序允许你在 Markdown 格式文本中添加 HTML 标签。如果你喜欢使用某些 HTML 标签胜于 Markdown 语法的话, 这将很有帮助。例如, 某些人发现通过 HTML 标签添加图片更加容易。当你需要更改某个 HTML 元素的属性时 (例如为文本指定颜色或更改图片的宽度), 使用 HTML 标签更方便些。

如需使用 HTML, 请将 HTML 标签添加到 Markdown 格式文本中即可。

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

渲染效果如下:

This word is bold. This word is italic.

HTML 用法最佳实践 出于安全原因, 并非所有 Markdown 应用程序都支持在 Markdown 文档中添加 HTML。如有疑问, 请查看 Markdown 应用程序的文档。某些应用程序只支持 HTML 标签的子集。

对于 HTML 的块级元素 `<div>`、`<table>`、`<pre>` 和 `<p>`, 请在其前后使用空行 (blank lines) 与其它内容进行分隔。尽量不要使用制表符 (tabs) 或空格 (spaces) 对 HTML 标签做缩进, 否则将影响格式。

在 HTML 块级标签内不能使用 Markdown 语法。例如

italic and **bold**

将不起作用。

Extended Syntax 扩展语法

概述

即便 Markdown 的原始设计文档中所列出的 Markdown 基本语法 已经囊括了许多满足日常所需的元素，但对于某些人来说仍然不够。这就是 Markdown 扩展语法出现的缘由。

一些个人和组织通过添加表格 (tables)、代码块 (code blocks)、语法高亮、将 URL 自动转换为链接和脚注 (footnotes) 等额外的元素来扩展 Markdown 的基本语法。这些额外添加的元素可以通过使用构建于 Markdown 之上的轻量级标记语言或通过向兼容的 Markdown 解析器添加扩展来启用这些新元素。

可用性

并非所有的 Markdown 应用程序都支持扩展语法。你需要确认你所使用的应用程序是否支持你所需要使用的扩展语法。如果不支持，则有可能是需要开启相应的扩展模块。

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	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` |
```

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 }
```

脚注 Footnotes

脚注（Footnotes）允许你添加注释（notes）和引用（references），而不会使文档正文混乱。当你创建脚注时，带有链接的上标数字会出现在你引用脚注的位置。读者可以单击链接以跳转至页面底部的脚注内容处。

要创建一个脚注的引用，请在方括号中添加一个插入符（caret）以及一个标识符，标识符可以是数字或单词，但不能包含空格或制表符。标识符的作用仅是将脚注的引用和脚注本身进行关联，在输出中，脚注按顺序编号。

另一种创建脚注的方式是在方括号内添加一个插入符（caret）以及一个数字，后面跟着冒号和文本，即（²: My footnote.）。这种方式让你不必在文档末尾添加脚注。你可以将脚注放到除列表（lists）、块引用（block quotes）和表格（tables）之外的任何位置上。

```
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：

²This is the first footnote.

Here' s a simple footnote,³ and here' s a longer one.⁴

锚链接 **Anchor Links**

许多 Markdown 解析器都支持为 标题 (headings) 自定义 ID, 某些 Markdown 解析器会自动为标题添加 ID。通过添加自定义 ID, 能够让你直接链接到这个标题, 并且还能使用 CSS 修改其样式。如需为标题添加自定义 ID, 请将自定义 ID 用花括号括起来并与标题一起放在同一行。

```
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.

³This is the first footnote.

⁴Here' s one with multiple paragraphs and code.

Indent paragraphs to include them in the footnote.

```
{ my code }
```

Add as many paragraphs as you like.

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 shortcodes)⁵。

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。

⁵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字体, 如 Noto Color Emoji, 并使用 LuaTeX 编译。

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。

```
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：

x^2

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`

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⁶。

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

⁶pandoc-crossref と Eisvogel を併用する

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (1)$$

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

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](#)