

完整 MARKDOWN 参考手册

Markdown Cheat Sheet

全部语法与示例 • 2026

涵盖 CommonMark 和 GitHub Flavored Markdown 的每一项功能，从标题、表格到代码、数学公式和图表，每项都配有可复制粘贴的示例。

下载 Markdific

互联网上最完整的 Markdown 速查表。涵盖每一项 CommonMark 和 GitHub Flavored Markdown 功能，每项都配有可用的示例。

把这个页面加入书签。或者获取免费的 [Markdown 速查表 PDF 版本](#)，方便离线随时查阅。

Markdown 语法速览

元素	语法
标题	<code># H1 到 ##### H6</code>
粗体	<code>**加粗**</code>
斜体	<code>*斜体*</code>
删除线	<code>~~删除~~</code>
链接	<code>[文本](https://example.com)</code>
图片	<code>![替代文本](image.jpg)</code>
行内代码	<code>`code`</code>
代码块	三个反引号，你的代码，然后再三个反引号
引用	<code>> 引用文本</code>
无序列表	<code>- 项目</code>
有序列表	<code>1. 项目</code>
任务列表	<code>- [] 待办</code> 和 <code>- [x] 已完成</code>
表格	<code> A B </code> ，下面加一行 <code> --- --- </code>
分隔线	<code>---</code>

什么是 Markdown?

Markdown 是一种纯文本格式化语法，由 John Gruber 和 Aaron Swartz 于 2004 年创建。它使用简单的符号来标记文本，使其可以转换成 HTML 及其他格式。它的目标是可读性。一个 Markdown 文件在作为纯文本时应当看起来整洁，只有在渲染后才变得更美观。

如今 Markdown 无处不在。GitHub 的 README、Reddit 帖子、Discord 消息、Notion 页面、Obsidian 笔记、AI 聊天机器人的回复，以及大多数文档站点都在使用它。如果你在 2026 年用电脑写任何东西，你几乎肯定在生成 Markdown，无论你是否意识到。

要进一步了解这种格式本身，请阅读 [什么是 .md 文件](#)、[如何打开和阅读 .md 文件](#)，或 [Markdown 与 HTML 对比](#) 以理解这两种格式的差异。要获取完整的逐元素参考，请查看 [Markdown 文档](#)。

标题

在一行开头使用 `#` 符号来创建标题。井号的数量决定标题级别，从 H1（最大）到 H6（最小）。

```
# 标题 1
## 标题 2
### 标题 3
#### 标题 4
##### 标题 5
##### 标题 6
```

备用标题语法 (Setext)

仅限 H1 和 H2，你可以使用下划线样式的标题：

```
标题 1
=====

标题 2
-----
```

大多数作者坚持使用 `#` 语法，因为它在每个级别都有效，而且更易于浏览。

提示：

- 始终在 `#` 和标题文本之间加一个空格
- 每个文档只用一个 H1（它就是标题）
- 在标题前后各留一个空行，以获得最稳妥的渲染效果

完整指南：[Markdown 中的标题](#) →

段落与换行

段落之间用一个空行分隔。只需按两次回车。

```
这是第一段。

这是第二段。
```

段落内的换行

要在不开始新段落的情况下换行，可以在行末加**两个空格**再按回车，或者使用反斜杠 `\`：

```
第一行。
第二行在新的一行，同一段落。

第一行。 \
使用反斜杠换行的第二行。
```

两者在 HTML 中都会生成一个 `<br>` 标签。行末空格的方法更传统，但在编辑器中不可见，容易引起困惑。反斜杠方法更清晰，并且受 GitHub Flavored Markdown 支持。

完整指南：[段落](#) → · [换行](#) →

粗体、斜体及其他文本格式

这些是你经常会用到的格式基础。

```
**粗体文本** 或 __粗体文本__
*斜体文本* 或 _斜体文本_
***粗体加斜体*** 或 ___粗体加斜体___
~~删除线文本~~
```

粗体和斜体是 CommonMark 核心的一部分，各处都有效。删除线是 GFM 扩展，下文有更详细的介绍。

何时使用星号还是下划线

`**` 和 `__` 都能生成粗体。`*` 和 `_` 都能生成斜体。它们可以互换，所以选一种并坚持使用以保持一致。

实际差异出现在单词内部。CommonMark 和 GFM 会忽略字母之间的下划线，因此下划线保持原样，而星号总是触发强调：

```
保持原样：  this_word_has_underscores
变成斜体：  this*word*has*asterisks
```

在包含下划线的正文（如 `file_name`）中使用下划线，以避免意外斜体；当你真的想在单词内部强调时，使用星号：

```
un**believ**ably
```

完整指南：[强调：粗体和斜体](#) →

列表

列表有两种类型：无序（项目符号）和有序（编号）。

无序列表

使用 `-`、`*` 或 `+` 后跟一个空格。这三者作用完全相同。大多数样式指南偏好 `-`。

- 第一项
- 第二项
- 第三项
 - 嵌套项（缩进 2 或 4 个空格）
 - 另一个嵌套项
 - 深层嵌套项
- 第四项

有序列表

使用数字后跟一个句点。实际的数字无所谓，Markdown 会自动重新编号：

1. 第一项
2. 第二项
3. 第三项
 1. 嵌套项
 2. 另一个嵌套项
4. 第四项

你也可以这样写，得到相同的输出：

1. 第一项
1. 第二项
1. 第三项

这是特性，不是缺陷。它意味着你可以重新排序列表项，而无需手动重新编号。

混合列表

你可以在有序列表内嵌套无序列表，反之亦然：

1. 步骤一
2. 步骤二
 - 子要点 A
 - 子要点 B
3. 步骤三

完整指南：[Markdown 中的列表](#) →

链接

链接用方括号表示可见文本，用圆括号表示 URL。

基本行内链接

```
[Markdific](https://markdific.com)
[访问我们的博客](https://markdific.com/blog "可选的悬停标题")
```

自动链接

用尖括号包裹一个 URL 来自动链接它：

```
<https://markdific.com>
<hello@example.com>
```

引用式链接

当同一个链接出现多次，或者你想保持正文整洁时很有用：

阅读我们的 [速查表][1] 或 [博客][2]。

在文档后面或底部：

[1]: <https://markdific.com/resources/markdown-cheat-sheet>

[2]: <https://markdific.com/blog/>

你也可以使用命名引用：

看看 [Markdific][md-home]。

[md-home]: <https://markdific.com>

链接到章节（锚点链接）

GitHub 和大多数渲染器会根据标题文本自动生成锚点 ID（转为小写，空格替换为连字符）：

[跳转到表格](#tables)

[跳转到 GitHub Alerts](#github-alerts)

完整指南：[链接](#) → [自动链接](#) → [标题 ID](#) →

Wikilinks 和嵌入

Obsidian 及其他一些笔记工具在标准 Markdown 之上添加了双括号 wikilinks 和嵌入：

[[笔记标题]] 链接到另一篇笔记

[[笔记标题|别名]] 带自定义文本的链接

![[笔记标题]] 嵌入另一篇笔记的内容

![[image.png]] 嵌入一张图片

这些是 Obsidian 扩展，不是标准 Markdown，因此在 GitHub 或大多数静态站点上无法解析。完整用法请参见 [Obsidian 中的 Markdown](#)。

图片

图片语法几乎与链接相同，只是前面加一个感叹号。

![描述图片的替代文本](path/to/image.jpg)

![替代文本](path/to/image.jpg "可选的悬停标题")

图片尺寸

纯 Markdown 不支持图片尺寸设置。你有两个选择：

选项 1：直接使用 HTML

```

```

选项 2：使用特定平台的扩展

例如，Obsidian 支持：

```
![[image.jpg|500]]
```

引用式图片

```
![Markdific 徽标][logo]
```

```
[logo]: /images/logo.png "Markdific"
```

为图片添加链接

要使图片可点击，把图片语法包裹在链接中：

```
[![替代文本](image.jpg)](https://destination-url.com)
```

对于纯 Markdown 无法覆盖的情况，比如本地文件、精确尺寸或 Base64 嵌入的图片，请使用上面展示的原生 HTML `<img>` 标签。

完整指南：[Markdown 中的图片](#) →

代码与代码块

代码格式化是 Markdown 最有用的功能之一。

行内代码

用单个反引号包裹代码：

```
在 Python 中使用 `print()` 函数。
```

如果你的代码包含反引号，用双反引号包裹它：

```
字符 `` `` 叫做反引号。
```

代码块（围栏式）

使用三个反引号来创建多行代码块。在起始反引号后添加语言标识符以实现语法高亮：

```
```python
def hello_world():
 print("Hello, world!")
```
```

```
```javascript
function helloWorld() {
 console.log("Hello, world!");
}
```
```

```
```bash
npm install markdown-it
```
```

常见语言标识符

- `python` 、 `py`
- `javascript` 、 `js`
- `typescript` 、 `ts`
- `bash` 、 `sh` 、 `shell`
- `html` 、 `xml`
- `css` 、 `scss`
- `json` 、 `yaml`
- `sql` 、 `graphql`
- `markdown` 、 `md`
- `diff` (用于 git diff)
- `plaintext` 、 `text` (无高亮)

缩进代码块（旧式）

将任意一行缩进四个空格或一个制表符，使其成为代码块。这种方式有效，但可读性不如围栏式代码块。请避免使用。

完整指南：[代码](#) → [围栏式代码块](#) → [语法高亮](#) →

引用

在一行开头使用 `>` 来创建引用。

- ```
> 这是一段引用。
> 它可以跨越多行。

> 你也可以有单行引用。
```

## 嵌套引用

堆叠 `>` 字符：

```
> 外层引用。
>
> > 嵌套引用。
> >
> > > 深层嵌套引用。
```

## 引用中包含其他 Markdown

引用可以包含标题、列表、代码及其他格式：

```
> ### 引用内的标题
>
> - 一个列表项
> - 另一个项目
>
> 一些 `行内代码` 和 粗体文本。
```

完整指南：[引用](#) →

## 分隔线

在单独的一行上使用三个或更多连字符、星号或下划线：

```



```

三者都渲染为 `<hr>`。请谨慎使用。大多数现代文档改用标题层级来分隔。

完整指南：[分隔线](#) →

## 表格

表格是 GitHub Flavored Markdown 扩展，不属于标准 CommonMark，但如今几乎各处都受支持。

### 基本表格

```
| 表头 1 | 表头 2 | 表头 3 |
|-----|-----|-----|
| 单元格 1 | 单元格 2 | 单元格 3 |
| 单元格 4 | 单元格 5 | 单元格 6 |
```

### 列对齐

在分隔行中添加冒号：

```
| 左对齐 | 居中对齐 | 右对齐 |
|:-----|:-----|:-----|
| 文本 | 文本 | 文本 |
| 较长的文本 | 较长的文本 | 较长的文本 |
```

- `:---` 左对齐（默认）
- `:---:` 居中对齐
- `---:` 右对齐

## 让表格更整洁的技巧

- 外侧的竖线可选，但能提升可读性
- 源代码中的列宽不影响输出（渲染器会忽略多余的空格）
- 单元格内的竖线字符必须转义为 `\|`
- 标准 GFM 表格不能有多行单元格或合并单元格；这些情况请使用 HTML

要获取包括如何处理复杂表格在内的完整指南，以及我们免费的 [Markdown 表格生成器](#)，请阅读我们的 [Markdown 表格完整指南](#)。

完整指南：[Markdown 中的表格](#) →

## 任务列表

任务列表（也叫待办列表）是一项 GFM 扩展，受 GitHub、GitLab 及大多数现代编辑器支持。

- [x] 已完成的任务
- [ ] 未完成任务
- [ ] 另一个未完成任务
  - [x] 嵌套的已完成子任务
  - [ ] 嵌套的未完成子任务

`[x]` 在某些渲染器中区分大小写。为安全起见，请使用小写 `x`。

完整指南：[任务列表](#) →

## 脚注

脚注是一项 GitHub 扩展，也受 MultiMarkdown 和 Pandoc 支持。它们非常适合用于引用和补充说明。

这是一个带脚注的句子。<sup>[^1]</sup>

你可以有多个脚注。<sup>[^note]</sup>

<sup>[^1]</sup>：这是第一个脚注的内容。

<sup>[^note]</sup>：这是一个命名脚注，内容更长，  
如果缩进，可以跨越多行。

脚注会自动编号，并带着回链出现在渲染文档的底部。

完整指南：[脚注](#) →

---

## 定义列表

定义列表受 Pandoc、MultiMarkdown 及其他一些变体支持，但不属于核心 GFM。

术语

: 术语的定义。

Markdown

: 一种用于创建格式化文本的轻量级标记语言。

: 由 John Gruber 于 2004 年创建。

依赖这些之前请检查你的目标渲染器。它们在各平台之间不一致。

完整指南：[定义列表](#) →

---

## 缩写

MultiMarkdown 和 PHP Markdown Extra 让你定义一次缩写，然后在术语出现的任何地方展开它：

The HTML spec is maintained by the W3C.

\*[HTML]: HyperText Markup Language

\*[W3C]: World Wide Web Consortium

这不属于 CommonMark 或 GFM，因此大多数渲染器（包括 GitHub）会把定义行显示为字面文本。只在你确定目标支持时才使用它。

---

## 删除线

用双波浪号包裹文本：

~~这段文本被划掉了~~

**渲染效果：** 这段文本被划掉了

GFM 扩展。在现代渲染器中普遍受支持。

完整指南：[删除线](#) →

---

## 下标、上标和高亮

这三种行内标记是扩展功能。它们都不属于核心 CommonMark 或 GitHub Flavored Markdown，因此支持情况各异，但每一种都有可靠的 HTML 后备方案，在任何地方都能用。

### 下标和上标

Pandoc、MultiMarkdown 及少数其他变体支持波浪号和插入符语法：

```
Water is H~2~0.
The area is 10 m^2^.
```

在不支持该语法的地方（GitHub 和大多数 CommonMark 渲染器），改用 HTML 标签，它们在各处都能渲染：

```
Water is H₂O.
The area is 10 m².
```

### 高亮

Obsidian 及其他一些工具会高亮用双等号包裹的文本：

```
Markdown makes ==这部分== stand out.
```

对于不支持 `==` 的渲染器，请使用 HTML `<mark>` 标签： `<mark>highlighted</mark>`。

完整指南：[下标与上标](#) → · [高亮](#) →

---

## Emoji

在 Markdown 中使用 emoji 的三种方式：

### 1. 直接粘贴

直接输入或粘贴实际的 emoji 字符：🎉✅🚀

这在各处都有效，因为 emoji 只是 Unicode 字符。

### 2. 短代码（GFM）

GitHub 和 GitLab 支持短代码语法，将文本代码转换为 emoji：

```
:tada: :white_check_mark: :rocket:
```

在支持的平台上，这些会渲染为 🎉✅🚀。Discord 和 Slack 使用类似的 `:code:` 语法，但拥有各自的 emoji 库，因此可用的代码有所不同。参见 [Discord 中的 Markdown](#) 和 [Slack 中的 Markdown](#) 了解每个聊天应用如何处理格式。

### 3. HTML 实体

为了获得最大兼容性，你也可以使用 HTML 实体代码，不过短代码的可读性要高得多。

完整指南：[Markdown 中的 Emoji](#) →

---

# Markdown 中的 HTML

对于语法无法覆盖的情况，你可以在 Markdown 中嵌入原生 HTML：

```
<details>
<summary>点击展开</summary>

这段内容默认隐藏，用户点击时才会显示。

</details>
```

原生 HTML 的常见用途：

- `<details>` 和 `<summary>` 用于可折叠章节
- `<sub>` 和 `<sup>` 用于下标和上标
- `<kbd>` 用于键盘按键（渲染为 Ctrl + C 样式）
- `<mark>` 用于高亮
- 带 width 属性的 `<img>` 用于设定尺寸的图片
- 带合并单元格、多行单元格或复杂布局的表格

## 键盘按键示例

```
Press <kbd>Ctrl</kbd> + <kbd>C</kbd> to copy.
```

## 注意事项

某些平台出于安全考虑会剥离或净化 HTML（例如 Reddit）。GitHub 允许大多数 HTML，但会屏蔽 `<script>` 及少数其他标签。请始终在你的目标平台上测试。

完整指南：[Markdown 中的行内 HTML](#) →

# 注释

Markdown 没有官方的注释语法，但有两种可靠的方法可以在文件中留下备注，而它们永远不会出现在渲染输出中。

## HTML 注释

任何允许 HTML 的渲染器也会遵循 HTML 注释：

```
<!-- 这条备注在渲染页面中不可见。 -->

可见内容从这里继续。
```

文件渲染时该注释会被跳过。不过它仍然存在于 HTML 源代码中，所以应把它当作对读者隐藏，而非真正私密。

## 引用链接技巧

对于会剥离 HTML 的渲染器（某些聊天应用和沙箱查看器），使用一个空的引用链接定义。它不产生任何输出，因为从来没有东西链接到它：

```
[//]: #（这是一条什么都不渲染的注释。）
```

```
[comment]: #（写同一件事的另一种方式。）
```

这在几乎所有解析器中都有效，因为一个从未被引用的链接定义会被直接丢弃。这是隐藏备注最具可移植性的方法。

## 转义字符

要按字面显示一个 Markdown 字符，在它前面加一个反斜杠：

```
* 这不是斜体 *
\# 这不是标题
\[这不是链接\](not-a-url)
```

你可以转义的字符：

`\ ` * _ {} [] () # + - . ! |`

完整指南：[转义字符](#) →

## 数学公式（LaTeX）

数学公式作为渲染器功能，受 GitHub、GitLab、Obsidian、Notion 及大多数现代文档工具支持。在美元符号内使用 LaTeX 语法。

### 行内数学

用单个美元符号表示行内数学：

```
The Pythagorean theorem is $a^2 + b^2 = c^2$.
```

GitHub 还支持一种备用的行内定界符，在数学两侧各用一个美元符号和一个反引号包裹，用于你的数学中含有与 Markdown 冲突的字符的情况：

```
Here $`a + b = c`$ uses the alternate delimiters.
```

### 块级数学

用双美元符号表示块级公式：

```
$$
\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi}
$$
```

## 常见 LaTeX 模式

分数：	<code>\$\$\frac{a}{b}\$\$</code>
平方根：	<code>\$\$\sqrt{x}\$\$</code>
幂：	<code>\$\$x^{2}\$\$</code>
下标：	<code>\$\$x_{i}\$\$</code>
希腊字母：	<code>\$\$\alpha, \beta, \gamma, \pi, \theta\$\$</code>
求和：	<code>\$\$\sum_{i=1}^n i\$\$</code>
积分：	<code>\$\$\int_0^1 x \, dx\$\$</code>
矩阵：	<code>\$\$\begin{pmatrix} a &amp; b \\ c &amp; d \end{pmatrix}\$\$</code>

各渲染器的语法有所不同。GitHub 使用 MathJax。有些平台需要 KaTeX，它的约定略有不同。

完整指南：[LaTeX 数学公式](#) →

## Mermaid 图表

Mermaid 是一个把文本转换为图表的 JavaScript 库。GitHub、GitLab、Notion、Obsidian 及大多数现代 Markdown 渲染器原生支持它。它是现代 Markdown 中最有用的功能之一。

### 流程图

```
```mermaid
graph TD
    A[开始] --> B{是否正常工作?}
    B -->|是| C[发布]
    B -->|否| D[调试]
    D --> B
```
```

### 时序图

```
```mermaid
sequenceDiagram
    participant User
    participant Browser
    participant Markdific
    User->>Browser: 打开 .md 文件
    Browser->>Markdific: 渲染 markdown
    Markdific-->>Browser: 返回 HTML
    Browser-->>User: 显示格式化页面
```
```

## 其他 Mermaid 图表类型

- `flowchart` (流程图)
- `sequenceDiagram` (随时间的交互)
- `classDiagram` (UML 类图)
- `stateDiagram-v2` (状态机)
- `erDiagram` (数据库模式)
- `gantt` (项目时间线)
- `pie` (饼图)
- `mindmap` (思维导图)
- `journey` (用户旅程)
- `gitGraph` (git 分支可视化)

Mermaid 支持的图表类型远比这里展示的多。哪些能渲染取决于你的工具的 Mermaid 版本，GitHub 和 GitLab 保持其版本最新，因此大多数图表类型开箱即用。

完整指南：[Mermaid 图表](#) →

## GitHub Alerts

GitHub Alerts（也叫“callouts”或“admonitions”）是 GitHub 于 2023 年末推出的功能。它们渲染为带图标和颜色的特殊引用，如今在其他 Markdown 渲染器中也广泛受支持。

- > `[!NOTE]`  
> 用户应当知道的有用信息。
- > `[!TIP]`  
> 把事情做得更好的有用建议。
- > `[!IMPORTANT]`  
> 用户需要知道的关键信息。
- > `[!WARNING]`  
> 需要立即关注的紧急信息。
- > `[!CAUTION]`  
> 提示风险或负面后果。

支持的五种类型是 `NOTE`、`TIP`、`IMPORTANT`、`WARNING` 和 `CAUTION`。其他 Markdown 工具（Obsidian、MkDocs、Docusaurus）为 callouts 使用类似但略有不同的语法。参见 [GitHub 上的 Markdown](#) 了解 alerts 及其他 GitHub 功能如何渲染。

---

# YAML Frontmatter

Frontmatter 是位于 Markdown 文件顶部的一块 YAML 元数据，被静态站点生成器（Hugo、Jekyll、Astro、Next.js）、CMS 平台和笔记应用使用。

```

title: "我的文章标题"
date: 2026-01-15
author: Jane Doe
tags:
 - markdown
 - tutorial
 - reference
draft: false
description: 用于 SEO 的简短摘要。

文章内容从这里开始
```

三个连字符打开和关闭该块。在内部使用标准 YAML 语法。常见字段：

- `title`
- `date`
- `author` 或 `authors`
- `tags` 或 `categories`
- `description`（用于 meta 标签）
- `draft`（布尔值）
- `slug` 或 `permalink`
- `image`（特色图片路径）

Frontmatter 在渲染输出中不可见。它纯粹是供处理该文件的系统使用的元数据。

如果你的 frontmatter 在输出中显示为字面文本，说明解析器可能不支持它，请检查文件是否在第 1 行以 `---` 开头，并且上方没有空行。

完整指南：[Front matter](#) →

---

---

## Markdown 变体（flavors）对比

Markdown 有几种“变体（flavors）”，各有不同的功能。了解你的目标平台使用哪一种可以防止渲染出错。

| 变体(flavors)                    | 使用场景                  | 关键功能                                                  |
|--------------------------------|-----------------------|-------------------------------------------------------|
| CommonMark                     | Reddit、Stack Overflow | 标准化的核心。可预测但功能有限。                                      |
| GitHub Flavored Markdown (GFM) | GitHub、GitLab、大多数现代工具 | CommonMark + 表格、任务列表、删除线、自动链接、脚注、alerts               |
| MultiMarkdown                  | 学术和技术写作               | 增加表格、脚注、数学、引用、元数据                                     |
| Pandoc Markdown                | Pandoc 转换器            | 功能最强大。增加定义列表、数学、表格、引用、原生 HTML                         |
| Obsidian Markdown              | Obsidian 笔记应用         | GFM + wikilinks <code>[[note]]</code> 、callouts、嵌入、标签 |
| R Markdown                     | R、RStudio、Quarto      | 增加用于数据分析的可执行代码块                                       |
| Discord / Slack                | 聊天应用                  | 带有平台特定语法的自定义子集（剧透、提及、emoji 代码）                        |

拿不准时，就写 GFM。它是 2026 年最接近通用 Markdown 标准的东西。关于平台特定行为，参见 [Obsidian 中的 Markdown](#) 和 [Notion 中的 Markdown](#)。

完整指南：[Markdown 变体对比](#) →

## 常见错误

这些会不断绊倒写作者。请避免它们。

### 1. 忘记空行

Markdown 常常需要在元素之间留一个空行：

```
这是一段。
- 这个列表可能不会渲染
- 取决于渲染器。
```

用一个空行修复：

```
这是一段。

- 这个列表正确渲染。
- 每个渲染器都一致。
```

### 2. 混用缩进

当嵌套列表或向列表项添加段落时，使用一致的缩进（2 或 4 个空格）。混用制表符和空格会破坏渲染。

### 3. 用行末空格换行

用于软换行的“行末两个空格”技巧不可见且容易失效。请改用反斜杠，或者干脆接受段落通常才是你想要的。

## 4. 不转义特殊字符

在正文中写 `func_name_with_underscores` 在 CommonMark 和 GFM 中是安全的，它们会忽略字母之间的下划线，不过一些较旧的解析器仍会把它们读作强调。星号在这里风险更大，因为 `a*b*c` 会变斜体。为了在各处都安全，把技术术语用反引号包裹为行内代码。其他常见冲突：`<` 和 `>` 被读作 HTML 标签，以及表格单元格内的 `|` 破坏表格布局。

## 5. 依赖非标准功能

如果你使用了 Obsidian wikilinks `[[note]]`，然后发布到一个不支持它们的静态站点，你的链接就会失效。要了解你的目标渲染器。

## 6. 太宽的表格

GFM 表格不能水平滚动。如果一个表格对页面来说太宽，它就会溢出。把宽表格拆成多个较窄的表格，或者使用 HTML。

## 7. 空标题或只含格式的标题

```
粗体标题
```

这能用，但某些渲染器会剥离标题中的格式或生成奇怪的锚点 ID。让标题保持朴素。

# 下载 PDF 速查表

想离线使用这份速查表吗？[下载可打印的 PDF 版本](#)，放在桌上或分享给你的团队。

## 常见问题

### 什么是 Markdown 速查表？

Markdown 速查表是一份快速参考，列出每一个 Markdown 语法元素以及如何书写它的示例，包括标题、粗体和斜体文本、列表、链接、图片、代码块、表格等等。本页涵盖了所有 CommonMark 以及 GitHub Flavored Markdown 扩展。

### 如何在 Markdown 中把文本加粗？

用两个星号包裹文本：`**粗体文本**`。两个下划线（`__粗体文本__`）也有效。斜体用单个星号或下划线；粗体加斜体一起用三个：`***文本***`。

### 如何在 Markdown 中创建表格？

用竖线字符分隔各列，并在表头行下方加一行连字符，将其标记为表格：

```
名称	角色
Ada	负责人
```

或者完全跳过语法，用我们免费的 [Markdown 表格生成器](#) 可视化地构建一个。

## Markdown 和 GitHub Flavored Markdown 有什么区别？

CommonMark 是标准化的核心语法。GitHub Flavored Markdown (GFM) 用表格、任务列表、删除线、自动链接、脚注和 alerts 扩展了它。GFM 是受支持最广泛的变体，因此它是 2026 年最稳妥的默认选择。

## Markdown 和 HTML 是一回事吗？

不是。Markdown 是一种轻量级的纯文本语法，会转换成 HTML。Markdown 用于写作和编辑；HTML 用于显示。我们的 [Markdown 与 HTML 对比](#) 指南解释了何时使用哪一种。

## 如何在 Markdown 中添加链接？

把可见文本放在方括号中，后跟圆括号中的 URL：[\[Markdific\]\(https://markdific.com\)](https://markdific.com)。要把图片变成链接，用同样的方括号包裹图片语法。

## 如何在 Markdown 中添加注释？

Markdown 没有官方的注释语法。最具可移植性的方法是一个空的引用链接，`[//]: # (你的备注)`，它什么都不渲染。在任何允许 HTML 的渲染器中，HTML 注释 `<!-- 你的备注 -->` 也有效。

## 如何在 Markdown 中创建清单？

使用任务列表：每一项以 `- [ ]` 开头表示未完成任务，或 `- [x]` 表示已完成任务。任务列表是 GitHub Flavored Markdown 功能，受 GitHub、GitLab 及大多数现代编辑器支持。

## 如何在 Markdown 中添加图片？

使用一个感叹号、方括号中的替代文本，以及圆括号中的路径：`![替代文本](path/to/image.jpg)`。要设定图片尺寸，请改用带 `width` 属性的 HTML `<img>` 标签。

## 如何在 Markdown 中写代码块？

把代码用三个反引号包裹在各自单独的行上。在起始反引号后添加语言名称以实现语法高亮。对于句子中的简短片段，用单个反引号包裹 `code`。

## 如何在 Markdown 中制作目录？

Markdown 没有内置的目录，但你可以用锚点链接（如 [\[章节标题\]\(#section-title\)](#)）构建一个，使用转为小写、空格替换为连字符的标题。GitHub、GitLab 和许多编辑器也会自动生成一个。

## 如何将 Markdown 转换为 Word 或 PDF？

使用转换器。Markdific 拥有免费的在线工具，可将 Markdown 转换为 [Word](#)、[PDF](#)、[HTML](#) 等，全部在你的浏览器中运行，无需注册。

---

## 相关指南和工具

按平台分类的 Markdown。看看 Markdown 在你日常使用的工具中究竟如何表现：

- [GitHub 上的 Markdown](#)
- [Obsidian 中的 Markdown](#)
- [Notion 中的 Markdown](#)
- [VS Code 中的 Markdown](#)
- [Discord 中的 Markdown](#)

- [Slack 中的 Markdown](#)
- [Jupyter 中的 Markdown](#)
- [Confluence 中的 Markdown](#)
- [如何写 README](#)
- [面向 AI 智能体的 Markdown 文件](#)

**转换你的 Markdown。** 在浏览器中运行的免费在线转换器，无需注册：

- [Markdown 转 Word](#)
- [Markdown 转 PDF](#)
- [Markdown 转 HTML](#)
- [Markdown 转文本](#)
- [Markdown 转图片](#)
- [Markdown 转 Google Docs](#)

---

## 结语

这就是你在 2026 年实际会遇到的每一种 Markdown 语法。核心要点：

- **CommonMark + GFM** 覆盖了 95% 的实际 Markdown 使用场景
- **表格、任务列表、脚注、数学、mermaid 和 alerts** 是值得记住的扩展
- **HTML 后备方案** 处理 Markdown 无法覆盖的边缘情况
- **Frontmatter** 对任何静态站点或内容系统都至关重要
- **Mermaid 图表** 是 Markdown 中最被低估的功能之一

随着 AI 生成内容用 .md 文件涌入网络，这种格式会不断演进，但其核心语法自 2004 年以来一直稳定，不会改变。学一次，受益数十年。

想看到你的 Markdown 被干净地渲染吗？[Markdific](#) 是一款为 Mac 和 Windows 打造的快速专用 Markdown 查看器和编辑器。要了解更多信息，请阅读 [Markdown 与 HTML 对比](#)，浏览我们的 [Markdown 资源](#)，或阅读完整的 [Markdown 文档](#)。

---

## Markdific

Markdific 把原始 Markdown 变成任何设备上干净、易读的文档。打开任意 `.md` 文件，即刻看到正确的标题、表格、代码、数学公式和图片，然后用自动保存进行编辑、切换主题，并导出为 PDF、Word 或 HTML。

---



Markdific 是一款轻量的 Markdown 查看器和编辑器，打开任意 .md 文件即刻渲染，把原始 Markdown 变成干净、格式化的标题、表格、代码、数学公式和图片。支持自动保存编辑、切换主题，并导出为 PDF、Word 或 HTML。

[下载 Markdific](#)