

markdown语言帮助

▼ markdown语言帮助

- 概述
- 基本语法
- 标题
- 目录
- 引用
- 粗体与斜体
- 高亮显示文字
- 删除线
- 下划线
- 列表
- 任务列表
- 代码块
- 代码块背景颜色
- 表格
- 数学公式
- 图片
- 超链接
- 分隔线
- 表情
- 脚注
- 下标与上标

概述

开头

Markdown是一种轻量级的标记语言，设计初衷是让人们能够使用简单的纯文本格式来编写结构化的内容。它旨在易于阅读、易于编写，并且能够转换成HTML等其他格式。

以下是关于Markdown的一些概述信息：

1. 简洁易读：Markdown使用简单的标记语法来表示不同的文本元素，如标题、段落、列表、链接和代码块等。它的语法非常简洁和易于阅读，没有复杂的标记和样式设置。
2. 平台无关性：Markdown是一种纯文本格式，它可以在几乎所有的文本编辑器和平台上进行编写和阅读。它不依赖于特定的文本编辑器或操作系统，因此可以在各种环境中使用。
3. 转换性：尽管Markdown本身是一种纯文本格式，但可以通过使用Markdown解析器将其转换为其他格式，如HTML、PDF和Word等。这种转换使得Markdown的内容可以被呈现和共享在不同的平

台和媒体上。

4. 常见应用：Markdown广泛应用于各种场景中，如撰写技术文档、博客文章、论坛帖子、README文件、文档注释、笔记和文本编辑等。许多网站和工具（如GitHub、Stack Overflow和Jekyll）都支持Markdown格式。

基本语法

开头

在Markdown中，每条语句的结尾通常不需要使用分号。Markdown是一种轻量级的标记语言，用于格式化文本的编写，通常用于编写文档和网页。

在Markdown中，通过使用空行或换行符来分隔段落和语句。不同的行被视为独立的语句或段落。因此，Markdown不依赖于分号来划分语句。您可以直接在新行上编写下一个语句或段落。

例如，下面是一些常见的Markdown示例，它们没有末尾使用分号：

这是第一段话。

这是第二段话。

- 列表项1
- 列表项2
- 列表项3

尽管如此，在某些情况下，您可以选择在Markdown语句的末尾使用分号。这可能允许您在同一行上编写多个语句，或是为了遵循特定的编写风格和规范。但是请注意，Markdown解析器在处理时会忽略分号，因此不会影响呈现的结果。

总而言之，在Markdown中，每条语句的结尾通常不需要使用分号，而是通过空行或换行符来分隔语句和段落。

标题

开头

在Markdown中，你可以使用不同数量的井号（#）来表示不同级别的标题。Markdown支持六级标题，对应着分别使用一个到六个井号。

以下是示例及对应的Markdown标题表示方法：

一级标题

二级标题

三级标题

四级标题

五级标题

六级标题

这些标题将会被以不同的样式呈现，通常是不同大小的文本、不同的字体加粗程度或不同的排列方式等，具体样式根据你使用的Markdown解析器和主题而有所区别。

你也可以使用下划线 (=) 和减号 (-) 来表示一级和二级标题：

一级标题
=====

二级标题

这种方式在一些Markdown解析器中也会被解析成相应级别的标题。

需要注意的是，一级和二级标题在文档中通常只使用一次，表示整个文档的标题和副标题。从三级标题开始才是正文中使用的标题。

目录

开头

在Markdown中，TOC（Table of Contents，目录）是一种用于自动生成文档章节的列表的技术。虽然Markdown本身没有原生支持TOC的功能，但可以通过使用一些工具和扩展来生成TOC。

以下是两种生成Markdown TOC的常用方法：

1. 使用插件和工具：许多Markdown编辑器和工具提供了生成TOC的插件或内置功能。例如，Typora、VS Code（使用插件如Markdown All in One或Markdown TOC）、Pandoc和Markdown-TOC等工具都可以自动生成TOC。它们会扫描文档中的标题，并生成相应的目录。
2. 使用预处理器：另一种方法是使用Markdown预处理器，如Markdown-PP、Pandoc和markdown-toc等。这些工具可以在构建或转换Markdown文档之前，将特殊的TOC标记替换成实际的目录列表。你可以将 [toc] 或类似的标记放在文档中的任何位置，预处理器会根据实际的标题生成TOC。

下面是一个示例Markdown文档，展示了如何使用Typora生成TOC：

目录

<!-- TOC -->

- [第一章](#第一章)
- [第二章](#第二章)
 - [第二章的子节1](#第二章的子节1)
 - [第二章的子节2](#第二章的子节2)
- [第三章](#第三章)

<!-- /TOC -->

第一章

这是第一章的内容。

第二章

这是第二章的内容。

第二章的子节1

这是第二章的第一个子节的内容。

第二章的子节2

这是第二章的第二个子节的内容。

第三章

这是第三章的内容。

Typora会自动解析以上代码中的TOC标记，并生成一个目录列表，也可以随着文档的变化而更新。

需要注意的是，每个标题要有唯一的ID，以便在TOC中正确链接到相应的章节。某些Markdown解析器和工具会自动为标题生成ID，但某些情况下你可能需要手动指定ID。

引用

开头

在Markdown中，引用（Quote）用于引用他人的话语、文本片段、代码等。引用可以帮助强调他人观点、给出例子或引用文献资料。

在Markdown中，可以使用大于号（>）表示引用。以下是一些示例：

- > 这是一个引用的文本片段。
- > 引用可以包含多行文本。
- > 这是第二行引用。
- > 这是第三行引用。

你也可以在引用中嵌套其他引用：

- > 这是一个引用。
- >
- > > 这是一个嵌套的引用。

引用还可以与其他Markdown元素一起使用，例如链接、列表、代码块等。

- > 这是一个引用包含链接的例子：[点击这里](http://www.baidu.com)。
- > 引用中的列表：
 - > - 第一项
 - > - 第二项
 - > - 第三项
- > 引用中的代码块：
 - > ```
 - > print("Hello, World!")
 - > ```

有些Markdown编辑器还会自动为多行文本块添加引用。你可以使用缩进或多个大于号来表示这种引用。

- > 这是一个
- > 多行文本的引用。
- >> 这是一个缩进的多行文本引用。

引用可以与其他Markdown元素一起使用，以提供更丰富的文档结构和格式。

粗体与斜体

开头

在Markdown中，你可以使用星号（*）或下划线（_）将文本标记为粗体（bold）。将文本包裹在两个星号或下划线之间，即可实现粗体效果。

以下是一些示例：

- 这是*粗体*文本。
- 这是_粗体_文本。

上述示例中的"粗体"文本会以粗体样式显示。

Markdown也支持使用双星号或双下划线对文本进行粗体强调：

这是**粗体**文本。

这是__粗体__文本。

另外，Markdown还支持使用三个星号或三个下划线表示粗体和斜体的组合效果。将文本包裹在三个星号或三个下划线之间，即可同时应用粗体和斜体样式。

这是***粗体斜体***文本。

这是___粗体斜体___文本。

需要注意的是，在使用星号或下划线标记粗体时，开启标记的星号或下划线和结束标记的星号或下划线之间不应该有空格，否则Markdown将不会将其识别为粗体标记。

高亮显示文字

开头

Markdown 并没有原生支持 == 语法来实现文字的高亮或标记。Markdown 主要关注文本的结构和格式，而不是提供强调或高亮文字的功能。

实际上，在一些扩展版本的 Markdown 中，如 GitHub Flavored Markdown (GFM) 和 GitLab Markdown，确实支持使用 == 语法来实现文本的高亮效果。

在这些扩展中，你可以使用 == 将文本包围起来，以使其突出显示。这种语法通常用于在文档中注释或标记重点内容。

以下是一个示例：

这是一段普通的文本。

==这段文本将被高亮显示。==

这是另一段普通的文本。

在上述示例中，==这段文本将被高亮显示。== 这段文字将以高亮的样式显示。

然而，需要注意的是，这种使用 == 语法来实现高亮效果的方式是特定于某些 Markdown 解析器和工具的，而非标准的 Markdown 语法。因此，在使用这种语法时，你需要确保你使用的 Markdown 工具或渲染器支持该扩展。

请注意，不同的 Markdown 工具和渲染器可能对扩展语法的支持程度有所不同。因此，如果你计划在特定平台或环境中使用扩展语法，最好查阅相关文档以了解其具体用法和支持情况。

删除线

开头

在Markdown中，你可以使用两个波浪线（~）将文本标记为删除线（`strikethrough`）。将文本包裹在两个波浪线之间，即可实现删除线效果。

以下是一些示例：

这是一个带有删除线的文本：~~~~被删除的文本~~~~。

上述示例中的"被删除的文本"将以带有删除线的样式显示。

请注意，在使用波浪线进行删除线标记时，开启标记的波浪线和结束标记的波浪线之间不应该有空格，否则Markdown将不会将其识别为删除线标记。

下划线

开头

在标准的Markdown语法中，并没有直接支持在文本中添加下划线的方法。Markdown主要关注于文本的结构和格式，而不是提供给用户直接控制下划线样式的选项。

然而，如果你需要在Markdown中添加下划线，可以使用HTML标签来实现。你可以使用 `<u>` 标签将文本包裹起来，从而在渲染时显示为下划线格式。例如：

这是带有下划线的文本。

上述示例中的"带有下划线"将在渲染时显示为带有下划线样式的文本。

请注意，在使用HTML标签时，不同的Markdown解析器和渲染器的效果可能会有所不同。一些解析器可能会完全支持HTML标签，而其他解析器可能对HTML标签有限的支持或完全禁用它们。

为了确保你所使用的Markdown解析器或渲染器支持HTML标签，请参考其文档或进行相应的测试。

列表

在Markdown中，你可以使用几种方式来创建列表，包括有序列表和无序列表。

无序列表使用短横线（-）、加号（+）或星号（*）作为列表标记：

- 无序列表项1
- 无序列表项2
- 无序列表项3

或

- + 无序列表项1
- + 无序列表项2
- + 无序列表项3

或

- * 无序列表项1
- * 无序列表项2
- * 无序列表项3

上述示例中的每个列表项都以短横线、加号或星号开始，并通过换行来表示不同的列表项。

有序列表使用数字后跟英文句点 (.) 作为列表标记：

1. 有序列表项1
2. 有序列表项2
3. 有序列表项3

上述示例中的每个列表项以数字和句点开始，并通过换行来表示不同的列表项。

你还可以在列表中嵌套其他列表，只需在嵌套列表前插入四个空格或一个制表符缩进即可。

- 无序列表项1
 - 嵌套的无序列表项1
 - 嵌套的无序列表项2
- 无序列表项2
 - 1. 嵌套的有序列表项1
 - 2. 嵌套的有序列表项2

请注意，在使用有序列表时，Markdown会自动根据你输入的第一个项目编号来确定后续项目的顺序，所以你可以按照你喜欢的顺序编写列表项。

任务列表

开头

在Markdown中，你可以使用任务列表来创建复选框列表，用于表示待办事项或任务清单。任务列表使用短横线 (-) 后跟一个空格，再加上用方括号 ([]) 包围的小写字母x或空格来表示是否完成该任务。

以下是一个示例：

- [x] 完成任务1
- [] 完成任务2
- [] 完成任务3

上述示例中，任务1被标记为已完成，而任务2和任务3未完成。渲染时，已完成的任務會顯示為被勾選的复选框，而未完成的任務會顯示為未勾選的复选框。

你可以根据需要修改勾选框的状态，将 ☒ 改为 ☐ 表示任务未完成，或者将 ☐ 改为 ☒ 表示任务已完成。

还可以在任务列表中嵌套其他Markdown元素，例如：

- ☐ 完成任务1
 - ☒ 完成子任务1
 - ☐ 完成子任务2
- ☐ 完成任务2
- ☐ 完成任务3

上述示例中的任务列表中包含一个嵌套的子任务列表。

代码块

开头

在Markdown中，你可以使用代码块来展示代码或其他可显示的文本。代码块可以以两种方式呈现：缩进式代码块和围栏式代码块。

1. 缩进式代码块：使用四个空格或一个制表符进行缩进。

示例：

```
这是一个缩进式代码块。
它可以包含多行代码或其他文本。
缩进的部分将被显示为等宽字体。
```

注意：在这种格式下，如果需要在代码块中插入换行符，请使用两个以上的空行来分隔行。

2. 围栏式代码块：使用三个反引号 (```)包围代码块，还可以指定代码块的语言，以便进行语法高亮。

示例：

```
# 这是一个围栏式代码块，使用Python语言
def hello():
    print("Hello, world!")
```

围栏式代码块还可以指定语言名称，以实现代码语法高亮。例如，在三个反引号后面添加编程语言名称，如：python、javascript、``java等。

代码块可以是单行或多行的，具体取决于你要展示的内容。

代码块背景颜色

开头

在标准的Markdown语法中，并没有直接支持在代码块中更改背景颜色的方法。Markdown主要关注于文本的结构和格式，而不是提供给用户直接控制代码块样式和外观的选项。

然而，有些Markdown扩展或特定的Markdown编辑器/渲染器可能允许你自定义代码块的样式，包括背景颜色。这些扩展或编辑器通常使用额外的标记或CSS样式来实现此功能。

例如，如果你使用的是GitHub Flavored Markdown（GFM），你可以使用HTML `<div>` 标签和内联CSS来自定义代码块的样式，包括背景颜色：

```
<div style="background-color: #f0f0f0; padding: 10px;">
这是一个有自定义背景颜色的代码块。
</div>
```

上述示例中，`style` 属性定义了背景颜色为 `#f0f0f0`，并设置了内边距为 `10px`。

请注意，使用此类自定义样式可能会限制可移植性，因为不是所有Markdown编辑器或渲染器都支持或解释这些自定义样式。在使用特定样式时，要确保你的目标平台能够支持。

如果你使用的是特定的Markdown编辑器或渲染器，可以参考其文档或进行相应的测试，以了解如何自定义代码块的样式，包括背景颜色。

表格

开头

在Markdown中，你可以使用竖线（`|`）和连字符（`-`）来创建简单的表格。下面是一个简单的示例：

列1标题	列2标题	列3标题
单元格1	单元格2	单元格3
单元格4	单元格5	单元格6

在这个示例中，第一行是表头，使用竖线和连字符表示表格的列和列之间的分隔符。第二行是表头和表格内容之间的分隔线。其余行是表格的内容。

你可以根据需要添加或删除单元格，并使其对齐：

左对齐	居中对齐	右对齐
:-----	:-----:	-----:
左对齐内容	居中对齐内容	右对齐内容

在这个示例中，使用冒号 (:) 在分隔线上指定了单元格的对齐方式。冒号在左边表示左对齐，冒号在右边表示右对齐，两边都有冒号表示居中对齐。

对于更复杂的表格、合并单元格或其他样式的需求，Markdown本身的功能有限。在这种情况下，您可能需要使用HTML表格标签或使用特定的Markdown扩展来实现所需的表格样式。

数学公式

开头

在Markdown中，您可以使用LaTeX语法来插入数学公式。只需将数学公式放在两个美元符号 (\$) 之间即可。

例如，要插入一个简单的公式： $y = mx + b$ ，您可以使用以下Markdown代码：

```
$y = mx + b$
```

如果您希望公式单独显示并居中，可以使用两对美元符号 (\$\$)：

```
$$y = mx + b$$
```

如果您需要插入更复杂的数学表达式或符号，请使用LaTeX语法编写公式。Markdown将自动将LaTeX代码转换为可视化的数学公式。

例如，要插入一个求和公式： $\sum_{i=1}^n i$ ，您可以使用以下Markdown代码：

```
$$\sum_{i=1}^n i$
```

图片

开头

在Markdown中插入图片很简单。您可以使用Markdown的图像语法来插入本地或远程图片。

- 1. 插入本地图片：

```
![Alt Text](path/to/image.jpg)
```

其中，Alt Text 是图片的替代文本，path/to/image.jpg 是本地图片的路径。请确保路径正确，并且在Markdown的当前工作目录下可访问到该图片。

- 2. 插入远程图片：

```
![Alt Text](URL/to/image.jpg)
```

其中，Alt Text 是图片的替代文本，URL/to/image.jpg 是远程图片的URL链接。请确保URL链接正确，并且可以在网络上访问到该图片。

您还可以添加可选的标题，将鼠标悬停在图片上时显示：

```
![Alt Text](path/to/image.jpg "Optional Title")
```

注意：在使用上述语法时，请替换 Alt Text 为适当的图片描述，并将 path/to/image.jpg 或 URL/to/image.jpg 替换为实际图片的路径或URL。

超链接

开头

在Markdown中插入超链接，您可以使用以下语法：

1. 行内链接：

```
[链接文本](链接URL)
```

示例：

```
[点击这里查看更多信息](https://www.baidu.com)
```

在上述示例中，"点击这里查看更多信息"是链接的文本，而"<https://www.baidu.com>"是链接的URL地址。

2. 引用链接：

```
[链接文本][链接标识]
```

在文档的其他位置定义链接标识：

```
[链接标识]: 链接URL
```

示例：

```
[点击这里查看更多信息][baidu]
...
[baidu]: https://www.baidu.com
```

在上述示例中，通过在文档其他位置定义链接标识，并在链接文本中使用链接标识，在使用时可以提供更简洁的链接语法。

3. 自动链接：

```
<链接URL>
```

示例：

<<https://www.baidu.com>>

自动链接适用于简单的URL链接，整个URL将作为可点击的链接。

分隔线

开头

在Markdown中插入分隔线，可以使用以下三种方式：

1. 使用三个或更多连续的 `***`、`---` 或 `___`：

`***`

`---`

`___`

2. 在一行中使用三个或更多的连续的 `-`（减号）：

`---`

3. 在一行中使用三个或更多的连续的 `*`（星号）：

`***`

以上三种方式都会在文档中创建一条水平分隔线。

表情

开头

在Markdown中，您可以使用Emoji表情来增添表达力和趣味性。以下是一些使用Emoji表情的示例：

1. 直接使用Emoji字符：

- 😊 表示笑脸
- 🌞 表示太阳
- 🎉 表示庆祝

2. 使用Emoji的名称：

您可以通过冒号 `:` 来引用Emoji的名称，例如 `:smile:` 将会被转换为 😊 的笑脸表情。

- 😊
- 🌞
- 🎉

另外，您还可以在前后加上冒号 `:` 来实现更多效果，例如 `:smile:` 会被转换为 😊 的大笑脸表情。

请注意，不同的平台和编辑器可能对Emoji的支持程度不同，请确保您使用的平台和编辑器支持在Markdown中使用Emoji。

脚注

开头

在Markdown中插入脚注，可以使用以下方法：

- 1. 使用脚注标记符号 (^) 加上脚注内容：

这是一个脚注的示例^[^1]。

^[^1]：这里是脚注的内容。

在上述示例中，文本 "这是一个脚注的示例" 后面的 "[^1]" 是脚注的标记符号，而脚注的内容 "[^1]: 这里是脚注的内容。" 需要放在文档的其他位置。

- 2. 使用连续的数字作为脚注标记符号：

这是一个脚注的示例.^[1]

^[1]：这里是脚注的内容。

在上述示例中， "[^1]" 是脚注的标记符号，而脚注的内容 "[^1]: 这里是脚注的内容。" 需要放在文档的其他位置。

使用脚注可以为文本提供额外的注释或解释，同时保持文档的整洁性。

下标与上标

开头

在Markdown中，可以使用HTML标记来实现上标和下标的效果。以下是两种方法：

上标 (Superscript)：

使用 `<sup>` 标签将要上标的文本包裹起来。

```
10<sup>th</sup>
```

输出结果为：10th

下标 (Subscript)：

使用 `<sub>` 标签将要下标的文本包裹起来。

H₂</sub>O

输出结果为：H2O

请注意，在一些Markdown编辑器和平台中，也许不支持或解析HTML标记。因此，在使用上述方法前，请先检查您的编辑器或平台是否支持HTML标记的使用。

在扩展Markdown中，可以使用^ 表示上标和 ~ 表示下标。

下标示例：

H~2~O 是水的化学式。

在上述示例中， ~2~ 对应的文本 "2" 将会被渲染为下标。

上标示例：

x^2^ + y^2^ = z^2^

在上述示例中， ^2^ 对应的文本 "2" 将会被渲染为上标。

请注意，Markdown的渲染效果依赖于具体的解析器和使用的工具，因此在不同的平台或工具中，渲染效果可能会略有不同。