

目 录

致谢

阅前必读

关于 Markdown

 第一个 Markdown 文档

 Hello.md

语法

 段落与换行

 标题

 引用

 列表

 代码

 分隔线

 超链接

 图片

 强调

 字符转义

扩展语法

 删除线

 代码块和语法高亮

 表格

 Task List

编辑器与扩展

 MarkdownPad

 Texts

 MarkPad

 MdCharm

 Markdown Edit

 CuteMarkEd

 Haroopad

 Mou

 MacDown

 Markdown Pro

 ReText

[sublime-markdown-extended](#)

[Atom Markdown Preview](#)

[IDEA Markdown](#)

[Cmd Markdown](#)

[StackEdit](#)

[Dillinger](#)

格式转换

[HTML](#)

[PDF](#)

[Word](#)

致谢

当前文档《Markdown 入门参考》由 进击的皇虫 使用 书栈(BookStack.CN) 进行构建，生成于 2018-02-25。

书栈(BookStack.CN) 仅提供文档编写、整理、归类等功能，以及对文档内容的生成和导出工具。

文档内容由网友们编写和整理，书栈(BookStack.CN) 难以确认文档内容知识点是否错漏。如果您在阅读文档获取知识的时候，发现文档内容有不恰当的地方，请向我们反馈，让我们共同携手，将知识准确、高效且有效地传递给每一个人。

同时，如果您在日常生活、工作和学习中遇到有价值有营养的知识文档，欢迎分享到 书栈(BookStack.CN)，为知识的传承献上您的一份力量！

如果当前文档生成时间太久，请到 书栈(BookStack.CN) 获取最新的文档，以跟上知识更新换代的步伐。

文档地址：<http://www.bookstack.cn/books/Learning-Markdown>

书栈官网：<http://www.bookstack.cn>

书栈开源：<https://github.com/TruthHun>

分享，让知识传承更久远！感谢知识的创造者，感谢知识的分享者，也感谢每一位阅读到此处的读者，因为我们都将成为知识的传承者。

阅前必读

- [Learning-Markdown \(Markdown 入门参考\)](#)
 - [文档目录](#)
 - [参考资料](#)
 - [共享协议](#)

Learning-Markdown (Markdown 入门参考)

[第一版备份在这里](#)

[Code on Github](#)

[在线阅读](#)

文档目录

- [关于 Markdown](#)
 - [第一个 Markdown 文档](#)
 - [Hello.md](#)
- [语法](#)
 - [段落与换行](#)
 - [标题](#)
 - [引用](#)
 - [列表](#)
 - [代码](#)
 - [分隔线](#)
 - [超链接](#)
 - [图片](#)
 - [强调](#)
 - [字符转义](#)
- [扩展语法](#)
 - [删除线](#)
 - [代码块和语法高亮](#)
 - [表格](#)
 - [Task List](#)
- [编辑器与扩展](#)
 - [MarkdownPad](#)
 - [Texts](#)
 - [MarkPad](#)

- [MdCharm](#)
- [Markdown Edit](#)
- [CuteMarkEd](#)
- [Haroopad](#)
- [Mou](#)
- [MacDown](#)
- [Markdown Pro](#)
- [ReText](#)
- [sublime-markdown-extended](#)
- [Atom Markdown Preview](#)
- [IDEA Markdown](#)
- [Cmd Markdown](#)
- [StackEdit](#)
- [Dillinger](#)
- [格式转换](#)
 - [HTML](#)
 - [PDF](#)
 - [Word](#)

参考资料

1. [Project Markdown](#)
2. [Markdown Syntax zh_TW](#)
3. [Markdown Syntax CN](#)
4. [Wiki Markdown](#)
5. [XBeta Wiki Markdown](#)

共享协议

本文由 [LearnShare](#) 整理并在 [CC BY-SA 3.0](#) 协议下发布，主要为了给自己和各位朋友作为学习 Markdown 的入门及参考资料。

请各位遵循 [Markdown: License](#) 及其它参考文献的共享协议来使用、修改和发布。

关于 Markdown

- [关于 Markdown](#)
 - [介绍](#)
 - [为什么选择 Markdown](#)
 - [兼容 HTML](#)

关于 Markdown

介绍

[Wiki: Markdown](#)

Markdown 是一种轻量级标记语言，创始人为约翰·格鲁伯（*John Gruber*）。它允许人们“使用易读易写的纯文本格式编写文档，然后转换成有效的 *XHTML*（或者 *HTML*）文档”。这种语言吸收了很多在电子邮件中已有的纯文本标记的特性。

为什么选择 Markdown

- 它基于纯文本，方便修改和共享；
- 几乎可以在所有的文本编辑器中编写；
- 有众多编程语言的实现，以及应用的相关扩展；
- 在 [GitHub](#) 等网站中有很好的应用；
- 很容易转换为 *HTML* 文档或其他格式；
- 适合用来编写文档、记录笔记、撰写文章。

兼容 HTML

Markdown 完全兼容 *HTML* 语法，可以直接在 Markdown 文档中插入 *HTML* 内容：

```
1. <table>
2.   <tr>
3.     <td>1</td>
4.     <td>2</td>
5.   </tr>
6.   <tr>
7.     <td>3</td>
8.     <td>4</td>
9.   </tr>
10. </table>
```

这段代码会变成下面的样子：

1	2
3	4

第一个 Markdown 文档

- [第一个 Markdown 文档](#)
 - [Hello.md](#)

第一个 Markdown 文档

Hello.md

打开你熟悉的文本编辑器（如 [Sublime Text](#)），新建一个 ‘hello.md’ 文件，写入下面的内容，并保存：

```
1. Hello
2. ====
3.
4. I like [Google](https://www.google.com/)
```

‘.md’ 和 ‘.markdown’ 都是被普遍支持的扩展名，不过 ‘.md’ 更加简单和方便。

它转换为 HTML 文档后，应该和 [Hello.md](#) 差不多：

Hello

I like [Google](#)

要了解将 *Markdown* 转换为 *HTML* 或其他格式，可以参考 [格式转换](#)。

Hello.md

- [Hello](#)

Hello

I like [Google](#)

语法

- [语法](#)

语法

这部分内容列出了 John Gruber 定义的 [Markdown 语法](#)，包括：

1. [段落与换行](#)
2. [标题](#)
3. [引用](#)
4. [列表](#)
5. [代码](#)
6. [分隔线](#)
7. [超链接](#)
8. [图像](#)
9. [强调](#)
0. [字符转义](#)

段落与换行

- [段落与换行](#)

段落与换行

1. 段落的前后必须是空行：

空行指的是行内什么都没有，或者只有空白符（空格或制表符）

相邻两行文本，如果中间没有空行
会显示在一行中（换行符被转换为空格）

2. 如果需要在段落内加入换行（`
`）：

可以在前一行的末尾加入至少两个空格
然后换行写其它的文字

3. Markdown 中的多数区块都需要在两个空行之间。

标题

- 标题
 - 1. Setext 形式
- H1
 - H2
 - 2. atx 形式
 - H4
 - H5
 - H4
 - H5
 - 左侧使用了空格
 - 内侧使用了空格

标题

1. Setext 形式

1. H1
2. =====
- 3.
4. H2
5. -----

H1

H2

`=` 和 `-` 的数量是没有限制的。通常的做法是使其和标题文本的长度相同，这样看起来比较舒服。或者可以像我一样，用四个 `-` 或 `=`。

Setext 形式只支持 `h1` 和 `h2` 两种标题。

2. atx 形式

- ① 可以用对称的 `#` 包括文本：

1. #####

- 2.
3. #####H5#####

H4

H5

② 也可以只在左边使用 # :

1. ####H4
- 2.
3. #####H5

H4

H5

③ 成对的 # 左侧和只在左边使用的 # 左侧都不可以有任何空白，但其内侧可以使用空白。

1. ###左侧使用了空格###
- 2.
3. ##### 内侧使用了空格

左侧使用了空格

内侧使用了空格

在这一点上，可能各种 *Markdown* 的实现会有不同的结果，不过仍然需要我们遵守语法规则。

引用

- 引用
 - 1. 引用内容
 - 2. 多行引用
 - 3. 嵌套引用
 - 4. 其他 Markdown

引用

1. 引用内容

在段落或其他内容前使用 `>` 符号，就可以将这段内容标记为‘引用’的内容（`<blockquote>`）：

1. `>`引用内容

引用内容

2. 多行引用

1. `>`多行引用
2. `>`可以在每行前加 ``>``

多行引用

可以在每行前加 `>`

1. `>`如果仅在第一行使用 ``>``，
2. 后面相邻的行即使省略 ``>``，也会变成引用内容

如果仅在第一行使用 `>`，
后面相邻的行即使省略 `>`，也会变成引用内容

1. `>`如果引用内容需要换行，
2. `>`可以在行尾添加两个空格
3. `>`
4. `>`或者在引用内容中加一个空行

如果引用内容需要换行，
可以在行尾添加两个空格

或者在引用内容中加一个空行

3. 嵌套引用

1. >也可以在引用中
2. >>使用嵌套的引用

也可以在引用中

使用嵌套的引用

4. 其他 Markdown

1. >在引用中可以使用使用其他任何 **Markdown** 语法

在引用中可以使用使用其他任何 *Markdown* 语法

列表

- [列表](#)
 - [无序列表](#)
 - [有序列表](#)
 - [嵌套的列表](#)
 - [语法和用法](#)

列表

无序列表

1. * 可以使用 ``*`` 作为标记
2. + 也可以使用 ``+``
3. - 或者 ``-``

- 可以使用 `*` 作为标记
- 也可以使用 `+`
- 或者 `-`

有序列表

1. 1. 有序列表以数字和 ``1.`` 开始；
2. 3. 数字的序列并不会影响生成的列表序列；
3. 4. 但仍然推荐按照自然顺序（1.2.3...）编写。

1. 有序列表以数字和 `1.` 开始；
2. 数字的序列并不会影响生成的列表序列；
3. 但仍然推荐按照自然顺序（1.2.3...）编写。

嵌套的列表

1. 1. 第一层
2. + 1-1
3. + 1-2
4. 2. 无序列表和有序列表可以随意相互嵌套

- ```
5. 1. 2-1
6. 2. 2-2
```

1. 第一层
  - 1-1
  - 1-2
2. 无序列表和有序列表可以随意相互嵌套
  - i. 2-1
  - ii. 2-2

## 语法和用法

1. 无序列表项的开始是：符号 空格；
2. 有序列表项的开始是：数字 `.` 空格；
3. 空格至少为一个，多个空格将被解析为一个；
4. 如果仅需要在行前显示数字和 `.`：

- ```
1. 05\ . 可以使用：数字\ . 来取消显示为列表
```

05. 可以使用：数字\ . 来取消显示为列表

`*`

的语法专门用来显示 *Markdown* 语法中使用的特殊字符，参考 [字符转义](#)

代码

- [代码](#)
 - [代码块](#)
 - [行内代码](#)
 - [转换规则](#)

代码

代码块

可以使用缩进来插入代码块：

1. `<html>` // Tab开头
2. `<title>Markdown</title>`
3. `</html>` // 四个空格开头

代码块前后需要有至少一个空行，且每行代码前需要有至少一个 Tab 或四个空格；

行内代码

也可以通过 ``，插入行内代码（` 是 `Tab` 键上边、数字 `1` 键左侧的那个按键）：

例如 `<title>Markdown</title>`

转换规则

代码块中的文本（包括 Markdown 语法）都会显示为原始内容，而特殊字符会被转换为 HTML 字符实体。

分隔线

- [分隔线](#)
 - *

分隔线

1. 可以在一行中使用三个或更多的 `*`、`-` 或 `_` 来添加分隔线 (`<hr>`)：

1. `***`
2. `-----`
3. `_____`

2. 多个字符之间可以有空格（空白符），但不能有其他字符：

1. `* * *`
2. `- - -`

超链接

- [超链接](#)
 - [行内式](#)
 - [参考式](#)
 - [自动链接](#)

超链接

行内式

格式为 `[link text](URL 'title text')` 。

① 普通链接：

```
1. \[Google\](http://www.google.com/)
```

[Google](#)

② 指向本地文件的链接：

```
1. \[icon.png\]($article-images-icon.png)
```

[icon.png](#)

③ 包含 ‘title’ 的链接：

```
1. \[Google\](http://www.google.com/ "Google")
```

[Google](#)

title 使用 ‘ 或 “ 都是可以的。

参考式

参考式链接的写法相当于行内式拆分成两部分，并通过一个 识别符 来连接两部分。参考式能尽量保持文章结构的简单，也方便统一管理 URL。

① 首先，定义链接：

```
1. [Google][link]
```

Google

第二个方括号内为链接独有的 识别符，可以是字母、数字、空白或标点符号。识别符是 不区分大小写的；

② 然后定义链接内容：

```
1. [link]: http://www.google.com/ "Google"
```

其格式为： `[识别符]: URL 'title'` 。

其中，*URL*可以使用 `<>` 包括起来，*title* 可以使用 `""`、`' '`、`()` 包括（考虑到兼容性，建议使用引号），*title* 部分也可以换行来写；

链接内容的定义可以放在同一个文件的 任意位置；

③ 也可以省略 识别符，使用链接文本作为 识别符：

```
1. [Google][]  
2. [Google]: http://www.google.com/ "Google"
```

Google

参考式相对于行内式有一个明显的优点，就是可以在多个不同的位置引用同一个 *URL*。

自动链接

使用 `<>` 包括的 *URL* 或邮箱地址会被自动转换为超链接：

```
1. <http://www.google.com/>  
2.  
3. <123@email.com>
```

<http://www.google.com/>

123@email.com

该方式适合行内较短的链接，会使用 *URL* 作为链接文字。邮箱地址会自动编码，以逃避抓取机器人。

超链接

图片

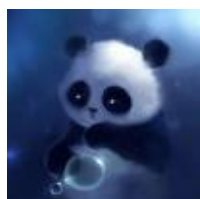
- 图像
 - 行内式
 - 参考式
 - 指定图片的显示大小

图像

插入图片的语法和插入超链接的语法基本一致，只是在最前面多一个 `!`。也分为行内式和参考式两种。

行内式

```
1. ![GitHub](https://avatars2.githubusercontent.com/u/3265208?v=3&s=100 "GitHub, Social Coding")
```



方括号中的部分是图片的替代文本，括号中的 ‘title’ 部分和链接一样，是可选的。

参考式

```
1. ![GitHub][github]
2.
3. [github]: https://avatars2.githubusercontent.com/u/3265208?v=3&s=100 "GitHub, Social Coding"
```



指定图片的显示大小

Markdown 不支持指定图片的显示大小，不过可以通过直接插入 `` 标签来指定相关属性：

```
1. 
```



强调

- 强调

强调

1. 使用 `* *` 或 `_ _` 包括的文本会被转换为 ``，通常表现为斜体：

1. 这是用来 **演示** 的 *_文本_*

这是用来 演示 的 文本

2. 使用 `** **` 或 `__ __` 包括的文本会被转换为 ``，通常表现为加粗：

1. 这是用来 ****演示**** 的 **__文本__**

这是用来 演示 的 文本

3. 用来包括文本的 `*` 或 `_` 内侧不能有空白，否则 `*` 和 `_` 将不会被转换（不同的实现会有不同的表现）：

1. 这是用来 ** 演示** 的 *_文本 _*

这是用来 演示 的 文本

4. 如果需要在文本中显示成对的 `*` 或 `_`，可以在符号前加入 `\` 即可：

1. 这是用来 *演示* 的 _文本_

这是用来 **演示** 的 *_文本_*

5. `*`、`**`、`_` 和 `__` 都必须 成对使用 。

字符转义

- 字符转义

字符转义

反斜线 (`\`) 用于插入在 Markdown 语法中有特殊作用的字符。

1. 这是用来 `*演示*` 的 `_文本_`
- 2.
3. 这是用来 `*演示*` 的 `_文本_`

这是用来 演示 的 文本

这是用来 `*演示*` 的 `_文本_`

这些字符包括：

1. `\`
2. ```
3. `*`
4. `_`
5. `{ }`
6. `[]`
7. `()`
8. `#`
9. `+`
10. `-`
11. `.`
12. `!`

扩展语法

- [扩展语法](#)
 - [CommonMark](#)

扩展语法

[Markdown 标准](#) 本身所包含的功能有限，所以产生了许多第三方的扩展语法，如 [GitHub Flavored Markdown](#)。

这里只介绍众多扩展语法中的一部分内容，它们在不同平台或工具的支持程度不同，请参考具体平台或工具的文档和说明来使用。

1. [删除线](#)
2. [代码块和语法高亮](#)
3. [表格](#)
4. [Task List](#)

CommonMark

[CommonMark](#) 试图将碎片化的 [Markdown](#) 实现和扩展进行标准化，提供统一的 [规范](#) 及不同语言的实现。

删除线

- [删除线](#)

删除线

1. 这就是 ~~删除线~~

这就是 删除线

代码块和语法高亮

- 代码块和语法高亮
 - 代码块
 - 代码高亮

代码块和语法高亮

代码块

与原来使用缩进来添加代码块的语法不同，这里使用 ```` ```` 来包含多行代码：

```
```  
<p>code here</p>
```
```

```
1. <p>code here</p>
```

三个 ````` 要独占一行。

代码高亮

在上面的代码块语法基础上，在第一组 ````` 之后添加代码的语言，如 `'javascript'` 或 `'js'`，即可将代码标记为 `JavaScript`：

```
```js  
window.addEventListener('load', function() {
 console.log('window loaded');
});
```
```

```
1. window.addEventListener('load', function() {  
2.   console.log('window loaded');  
3. });
```


表格

- 表格
 - 单元格和表头
 - 对齐
 - 插入其他内容

表格

单元格和表头

使用 `|` 来分隔不同的单元格，使用 `-` 来分隔表头和其他行：

```
1. name | age
2. ---- | ---
3. LearnShare | 12
4. Mike | 32
```

| name | age |
|------------|-----|
| LearnShare | 12 |
| Mike | 32 |

为了美观，可以使用空格对齐不同行的单元格，并在左右两侧都使用 `|` 来标记单元格边界：

```
1. | name | age |
2. | ---- | --- |
3. | LearnShare | 12 |
4. | Mike | 32 |
```

| name | age |
|------------|-----|
| LearnShare | 12 |
| Mike | 32 |

为了使 *Markdown* 更清晰，`|` 和 `-` 两侧需要至少有一个空格（最左侧和最右侧的 `|` 外就不需要了）。

对齐

在表头下方的分隔线标记中加入 `:`，即可标记下方单元格内容的对齐方式：

- `:---` 代表左对齐
- `---` 代表居中对齐
- `---` 代表右对齐

```
1. | left | center | right |
2. | :--- | :---: | ---: |
3. | aaaa | bbbbbb | ccccc |
4. | a    | b      | c      |
```

| left | center | right |
|------|--------|-------|
| aaaa | bbbbbb | ccccc |
| a | b | c |

如果不使用对齐标记，单元格中的内容默认左对齐；表头单元格中的内容会一直居中对齐（不同的实现可能会有不同表现）。

插入其他内容

表格中可以插入其他 Markdown 中的行内标记：

```
1. | name | age | blog |
2. | ----- | --- | ----- |
3. | _LearnShare_ | 12 | [LearnShare](http://xianbai.me) |
4. | __Mike__ | 32 | [Mike](http://mike.me) |
```

| name | age | blog |
|-------------------|-----|--|
| <i>LearnShare</i> | 12 | LearnShare |
| Mike | 32 | Mike |

Task List

- [Task List](#)

Task List

```
1. - [ ] Eat
2. - [x] Code
3.   - [x] HTML
4.   - [x] CSS
5.   - [x] JavaScript
6. - [ ] Sleep
```

- ☐ Eat
- ☒ Code
 - ☒ HTML
 - ☒ CSS
 - ☒ JavaScript
- ☐ Sleep

编辑器与扩展

- [编辑器与扩展](#)

编辑器与扩展

这部分给大家介绍一些可以用来书写、编辑 Markdown 的工具，包括独立客户端、编辑器/IDE 插件，以及基于 Web 的工具等，排名不分先后。

也欢迎你来推荐自己喜欢的工具和插件。

- [MarkdownPad](#) Windows
- [Texts](#) Windows OSX
- [MarkPad](#) Windows
- [MdCharm](#) Windows Linux
- [Markdown Edit](#) Windows
- [CuteMarkEd](#) Windows Linux
- [Haroopad](#) Windows OSX Linux
- [Mou](#) OSX
- [MacDown](#) OSX
- [Markdown Pro](#) OSX
- [ReText](#) Linux
- [sublime-markdown-extended](#) Sublime Text
- [Atom Markdown Preview](#) Atom
- [IDEA Markdown](#) IDEA
- [Cmd Markdown](#) Windows OSX Linux Web
- [StackEdit](#) Web
- [Dillinger](#) Web

MarkdownPad

- [MarkdownPad](#)
 - [介绍](#)
 - [截图](#)

MarkdownPad

主页: [MarkdownPad](#)

平台: Windows (XP+)

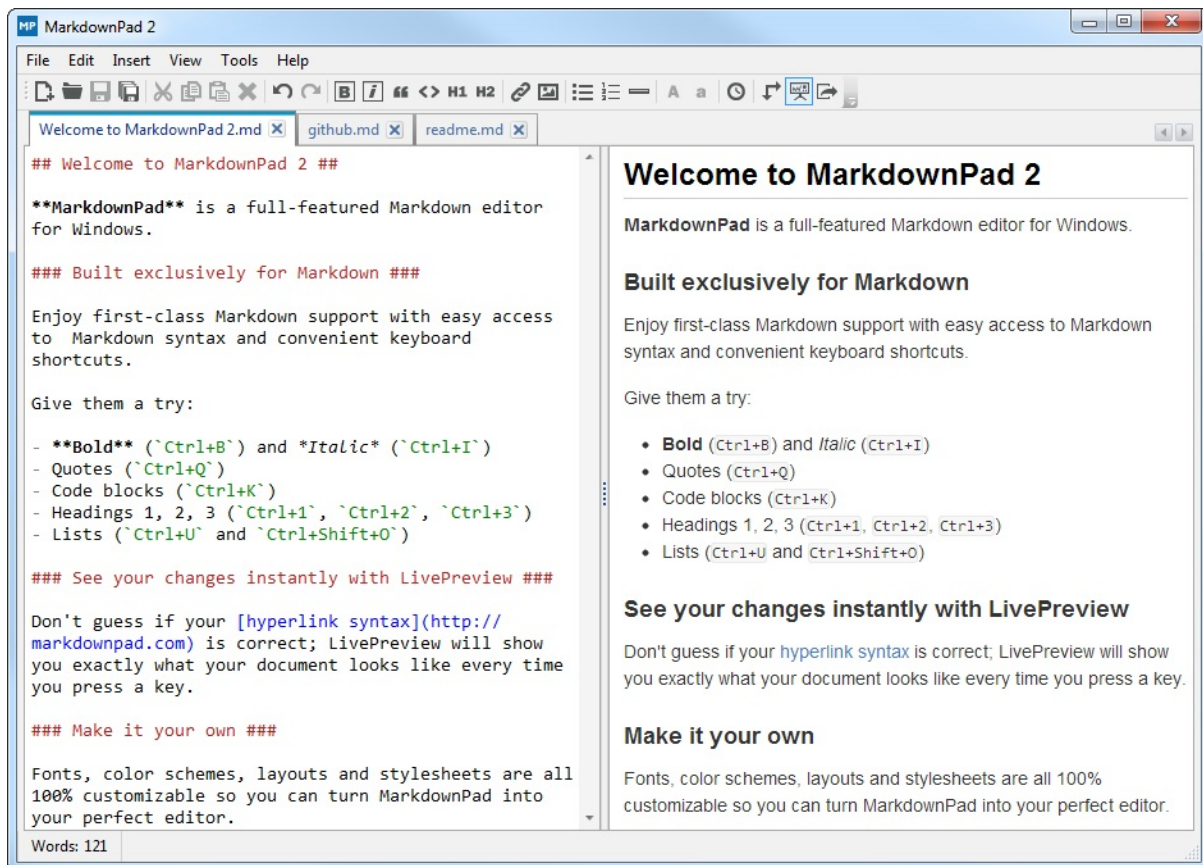
费用: 免费版、Pro 版 (\$14.95)

技术: Microsoft .NET

介绍

MarkdownPad 是一款 Windows 平台上的 Markdown 编辑器，提供了多文件编辑、实时预览、自定义样式、文件导出等功能。提供了免费版本和包含高级功能的 Pro 付费版 ([版本对比](#))，支持多国语言 (早期的中文翻译由我提供 [MarkdownPad 2 now speaks Chinese and Portuguese!](#))。

截图



Texts

- [Texts](#)
 - [介绍](#)
 - [截图](#)

Texts

主页: [Texts](#)

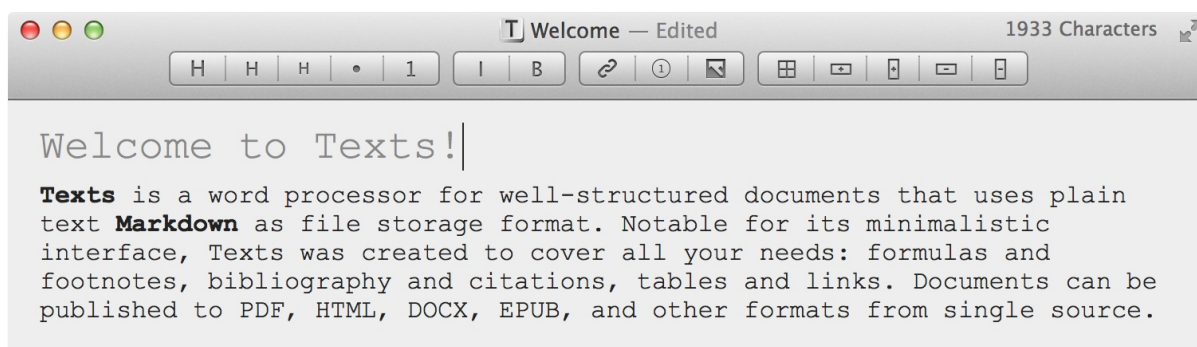
平台: Windows (XP+) OSX (10.8+)

费用: 免费试用 (\$29.00 \$14.50)

介绍

Texts 是一款所见即所得的 Markdown 编辑器，它只提供了在预览中进行编辑，无法直接编辑原始的 Markdown 文本。

截图



MarkPad

- [MarkPad](#)
 - [介绍](#)
 - [截图](#)

MarkPad

主页: [MarkPad](#)

平台: Windows (10)

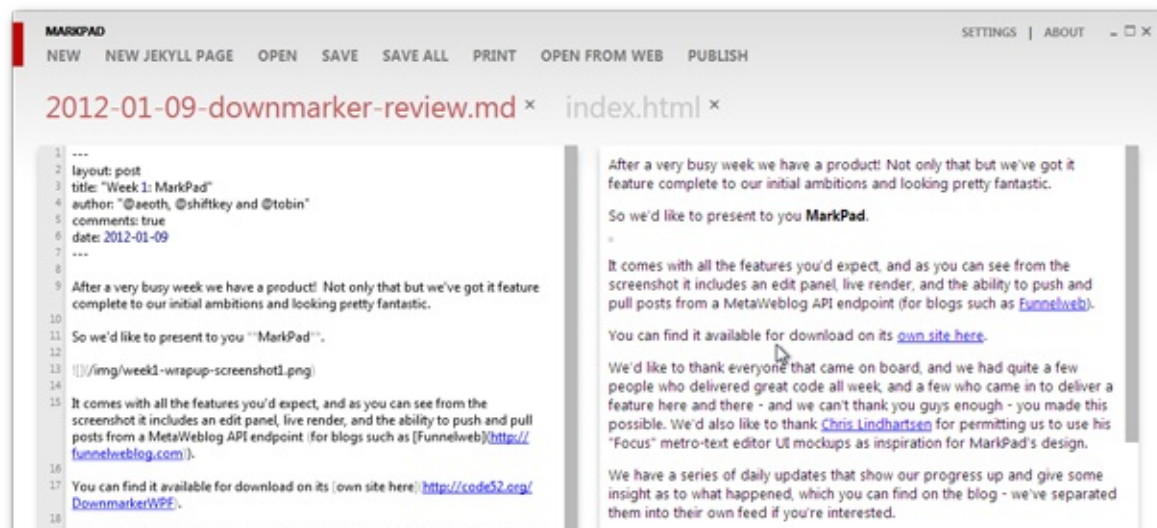
费用: 免费、开源

技术: C#

介绍

MarkPad 是一款开源的、Windows 风格的 Markdown 编辑器（目前只支持 Windows 10，曾经可以支持低版本），你可以直接将正在编辑的文档发布到 Github 等网站。

截图



MdCharm

- [MdCharm](#)
 - [介绍](#)
 - [截图](#)

MdCharm

主页：[MdCharm](#)

平台：Windows (XP+) Linux

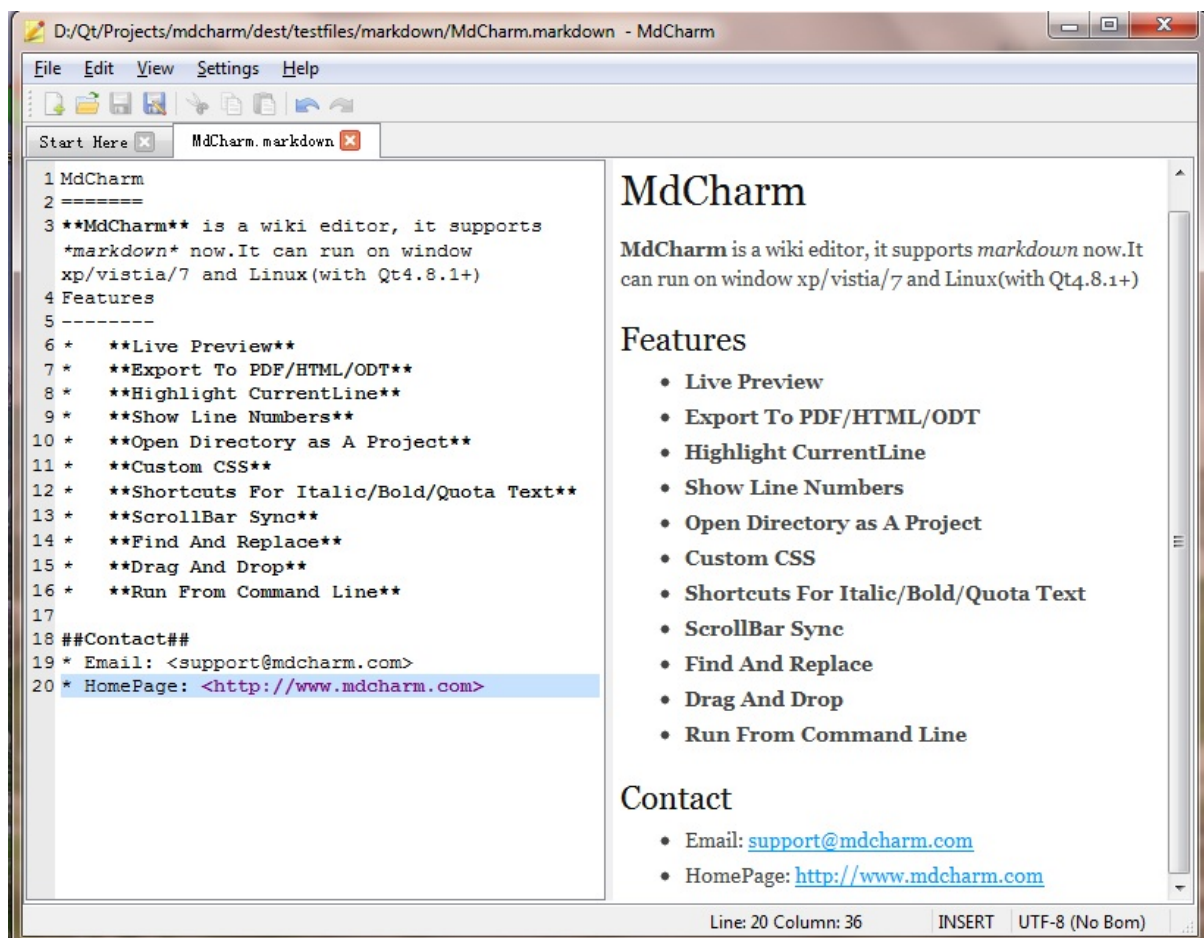
费用：免费、开源

技术：Qt

介绍

MdCharm 是一款 Qt 开发的 Markdown 编辑器，支持 Window 和 Linux 平台。是我使用最久的一款 Markdown 编辑器，不过目前已经停止维护。

截图



Markdown Edit

- [Markdown Edit](#)
 - [介绍](#)
 - [截图](#)

Markdown Edit

主页：[Markdown Edit](#)

平台：Windows

费用：免费、开源

技术：C#

介绍

Markdown Edit 是一款支持 CommonMark 的开源的编辑器。

截图

MARKDOWN EDIT - markdown-spec.md
15,847

Introduction

What is Markdown?

Markdown is a plain text format for writing structured documents, based on conventions used for indicating formatting in email and usenet posts. It was developed in 2004 by John Gruber, who wrote the first Markdown-to-HTML converter in Perl, and it soon became widely used in websites. By 2014 there were dozens of implementations in many languages. Some of them extended basic Markdown syntax with conventions for footnotes, definition lists, tables, and other constructs, and some allowed output not just in HTML but in LaTeX and many other formats.

Why is a spec needed?

John Gruber's [canonical description of Markdown's syntax](http://daringfireball.net/projects/markdown/syntax) does not specify the syntax unambiguously. Here are some examples of questions it does not answer:

1. How much indentation is needed for a sublist? The spec says that continuation paragraphs need to be indented four spaces, but is not fully explicit about sublists. It is natural to think that they, too, must be indented four spaces, but `Markdown.pl` does not require that. This is hardly a "corner case," and divergences between implementations on this issue often lead to surprises for users in real documents. (See [this comment by John Gruber](http://article.gmane.org/gmane.text.markdown.general/1997).)
2. Is a blank line needed before a block quote or header? Most implementations do not require the blank line. However, this can lead to unexpected results in hard-wrapped text, and also to ambiguities in parsing (note that some implementations put the header inside the blockquote, while others do not). (John Gruber has also spoken [in favor of requiring the blank lines](http://article.gmane.org/gmane.text.markdown.general/2146).)
3. Is a blank line needed before an indented code block? (`Markdown.pl` requires it, but this is not mentioned in the documentation, and some implementations do not require it.)

```

''' markdown
paragraph
  code
'''

```

Introduction

What is Markdown?

Markdown is a plain text format for writing structured documents, based on conventions used for indicating formatting in email and usenet posts. It was developed in 2004 by John Gruber, who wrote the first Markdown-to-HTML converter in Perl, and it soon became widely used in websites. By 2014 there were dozens of implementations in many languages. Some of them extended basic Markdown syntax with conventions for footnotes, definition lists, tables, and other constructs, and some allowed output not just in HTML but in LaTeX and many other formats.

Why is a spec needed?

John Gruber's [canonical description of Markdown's syntax](http://daringfireball.net/projects/markdown/syntax) does not specify the syntax unambiguously. Here are some examples of questions it does not answer:

1. How much indentation is needed for a sublist? The spec says that continuation paragraphs need to be indented four spaces, but is not fully explicit about sublists. It is natural to think that they, too, must be indented four spaces, but `Markdown.pl` does not require that. This is hardly a "corner case," and divergences between implementations on this issue often lead to surprises for users in real documents. (See [this comment by John Gruber](http://article.gmane.org/gmane.text.markdown.general/1997).)
2. Is a blank line needed before a block quote or header? Most implementations do not require the blank line. However, this can lead to unexpected results in hard-wrapped text, and also to ambiguities in parsing (note that some implementations put the header inside the blockquote, while others do not). (John Gruber has also spoken [in favor of requiring the blank lines](http://article.gmane.org/gmane.text.markdown.general/2146).)
3. Is a blank line needed before an indented code block? (`Markdown.pl` requires it, but this is not mentioned in the documentation, and some implementations do not require it.)

paragraph
code?
4. What is the exact rule for determining when list items get wrapped in `<p>` tags? Can a list be partially "loose" and partially "tight"? What should we do with a list like this?

1. one
2. two

CuteMarkEd

- [CuteMarkEd](#)
 - [介绍](#)
 - [截图](#)

CuteMarkEd

主页: [CuteMarkEd](#)

平台: Windows Linux

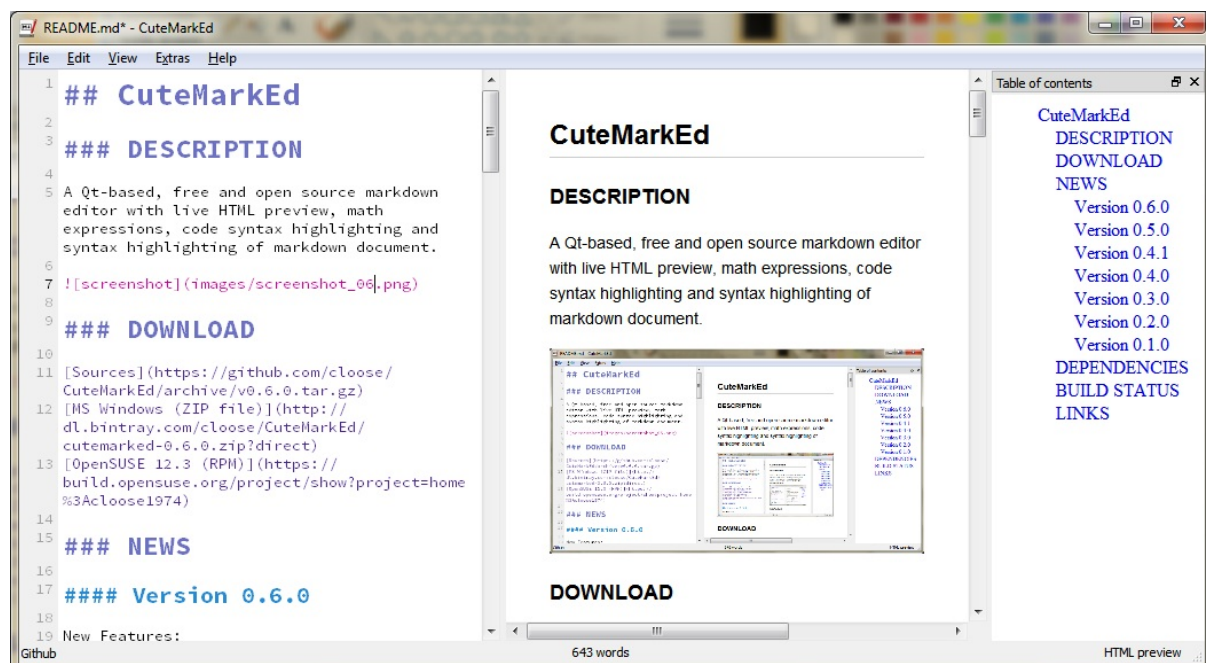
费用: 免费、开源

技术: Qt

介绍

CuteMarkEd 是一款使用 Qt 开发的 Markdown 编辑器, 支持 Windows 和 Linux 平台, 提供了 Markdown 编辑、预览, [mermaid](#) 流程图, [reveal.js](#) 演示等功能。它也提供了多国语言的支持 (简体中文部分主要由我提供)。

截图



Haroopad

- [Haroopad](#)
 - [介绍](#)
 - [截图](#)

Haroopad

主页: [Haroopad](#)

平台: Windows (7+) OSX Linux

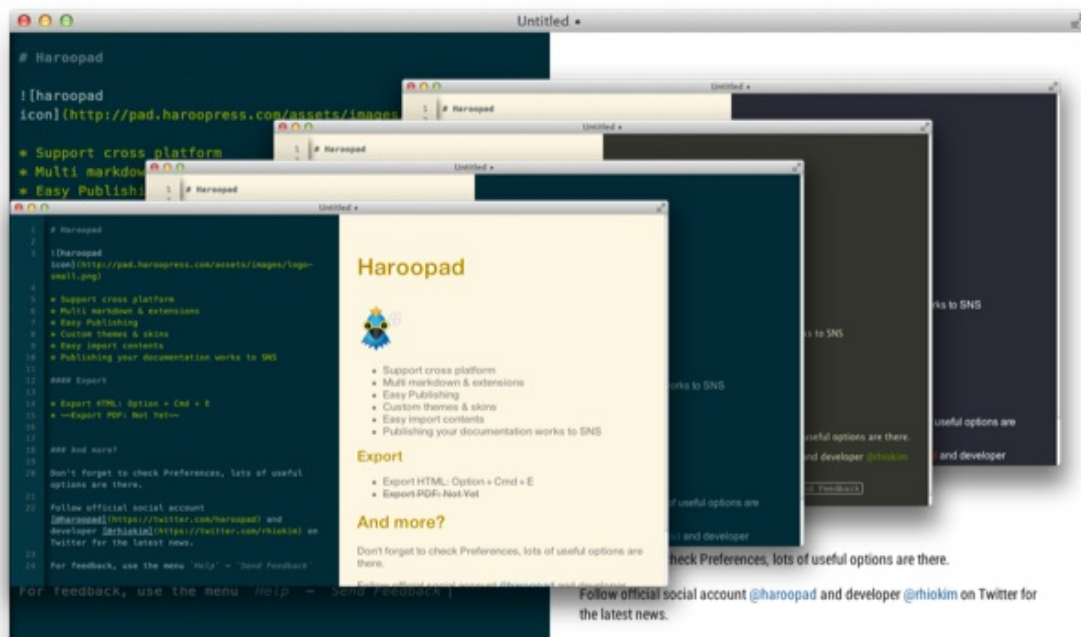
费用: 免费、开源

技术: Web

介绍

Haroopad 是一款使用 Web 技术开发的跨平台的 Markdown 编辑器, 提供了丰富的自定义功能以及众多 Markdown 扩展的支持。

截图



Mou

- [Mou](#)
 - [介绍](#)
 - [截图](#)

Mou

主页：[Mou](#)

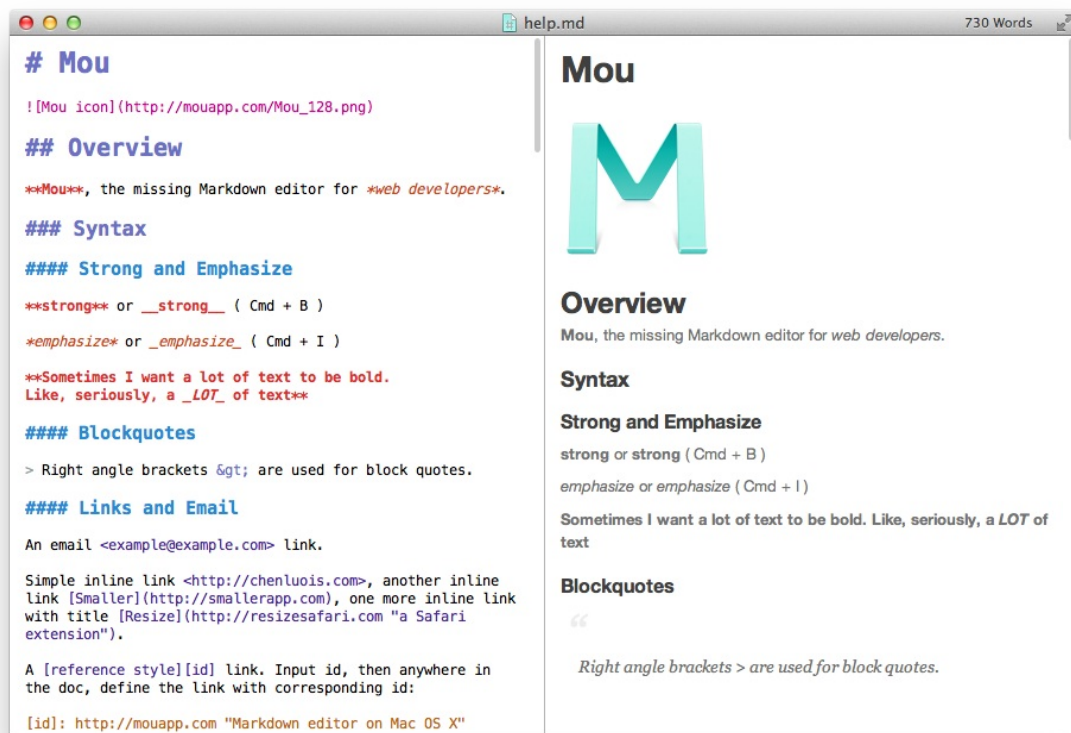
平台：OSX

费用：免费、收费（~~\$30~~ \$15）

介绍

Mou 是 OSX 上最受欢迎的 Markdown 编辑器，没有之一。目前正在 [众筹](#) 中的 1.0 版本，将在 2015-08-31 发布，不再继续跳票了。

截图



MacDown

- [MacDown](#)
 - [介绍](#)
 - [截图](#)

MacDown

主页：[MacDown](#)

平台：OSX

费用：免费、开源

技术：Object-C

介绍

MacDown 是一款深受 [Mou](#) 影响的免费、开源的 Markdown 编辑器，你可能要二选一了。

截图



Markdown Pro

- [Markdown Pro](#)
 - [介绍](#)
 - [截图](#)

Markdown Pro

主页: [Markdown Pro](#)

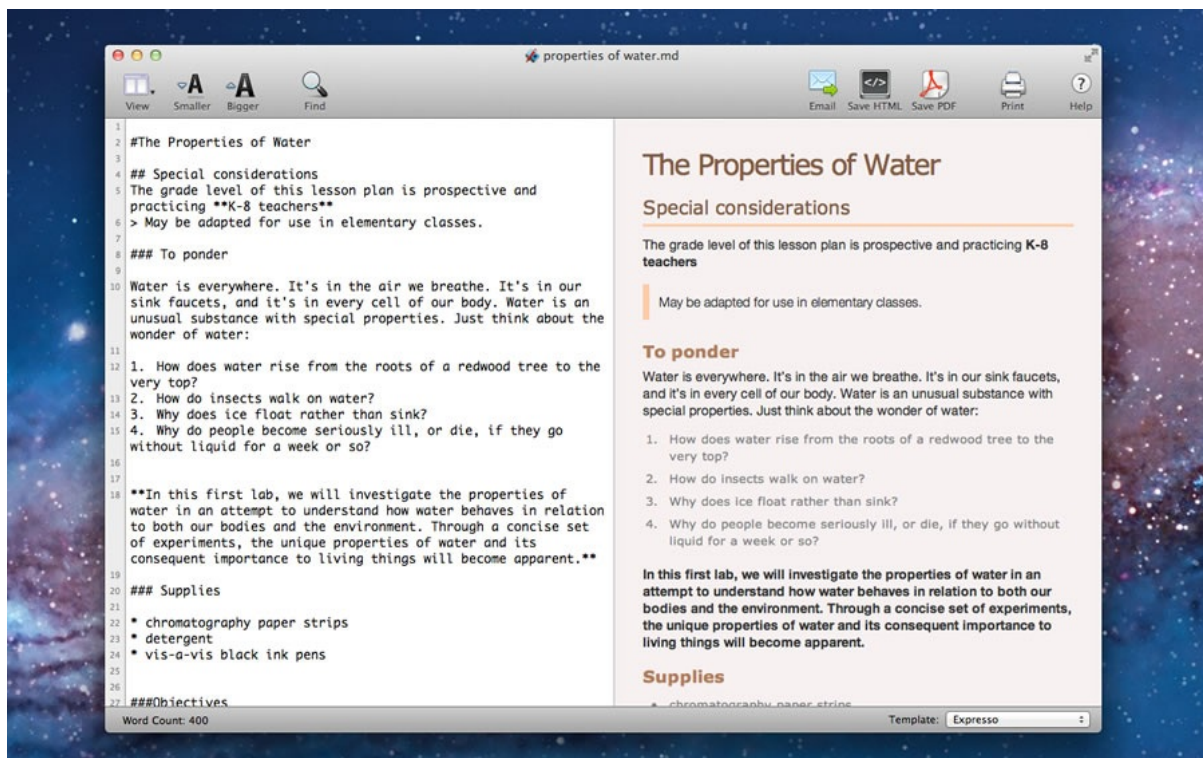
平台: OSX

费用: \$9.99

介绍

Markdown Pro 提供了 iCloud 支持、文件导出等功能。

截图



ReText

- [ReText](#)
 - [介绍](#)
 - [截图](#)

ReText

主页: [ReText](#)

平台: Linux

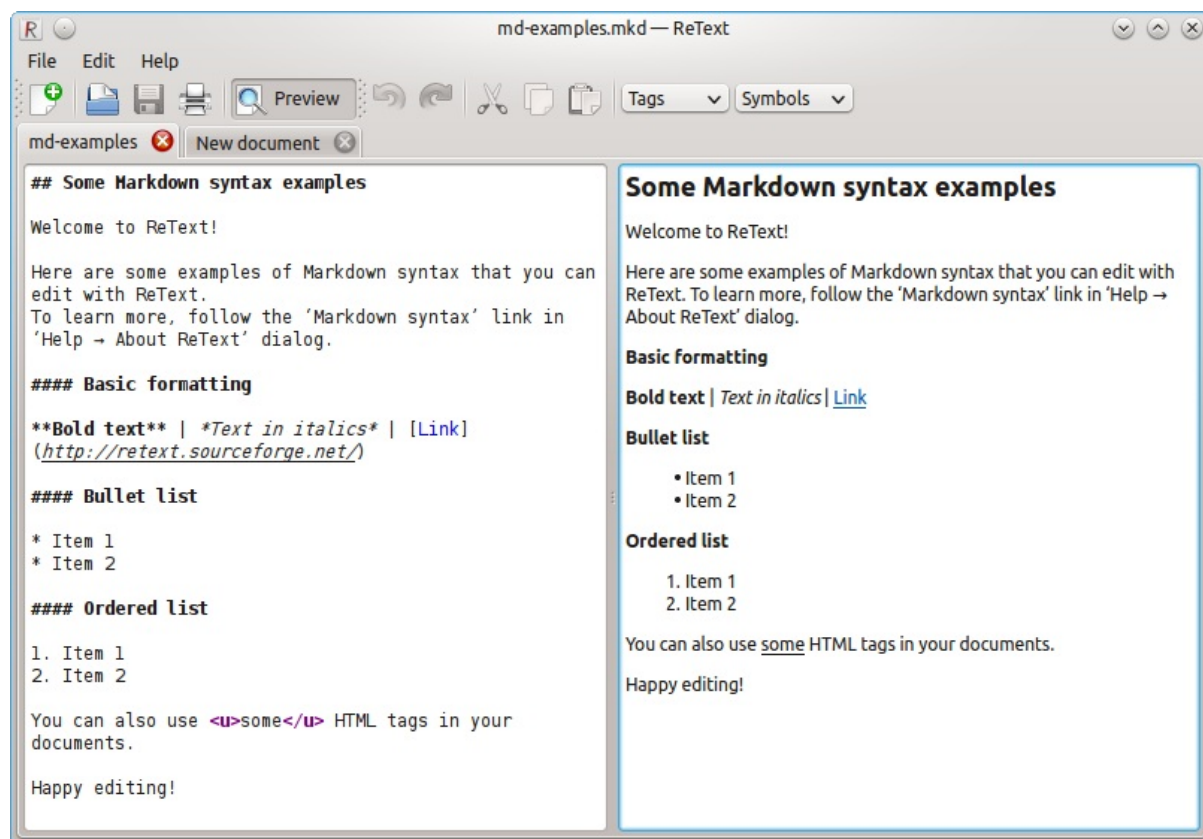
费用: 免费、开源

技术: Python、Qt

介绍

ReText 是一款 Linux 平台上的 Markdown 编辑器，提供了多种文件的导出功能。

截图



sublime-markdown-extended

- [sublime-markdown-extended](#)
 - [介绍](#)
 - [截图](#)

sublime-markdown-extended

主页: [sublime-markdown-extended](#)

应用: [Sublime Text](#) (2/3)

费用: 免费、开源

介绍

sublime-markdown-extended 是为 Sublime Text 开发的一款 Markdown 插件，提供了 Markdown 高亮、代码高亮等功能，大大提升了编辑 Markdown 文档的体验。

截图

```
```javascript
var sortObject = function(o) {
 var sorted = {},
 key, a = [];
 for (key in o) {
 if (o.hasOwnProperty(key)) {
 a.push(key);
 }
 }
 a.sort();
 for (key = 0; key < a.length; key++) {
 sorted[a[key]] = o[a[key]];
 }
 return sorted;
}
```

# Atom Markdown Preview

- [Atom Markdown Preview](#)
  - [介绍](#)
  - [截图](#)

## Atom Markdown Preview

---

主页: [Markdown Preview package](#)

应用: [Atom](#)

费用: 免费、开源

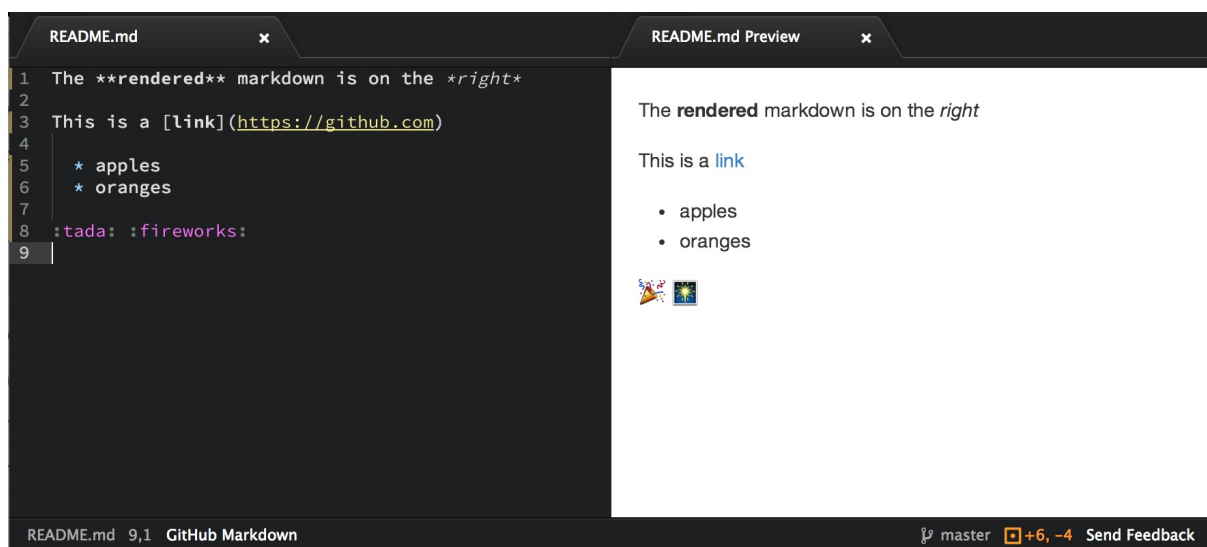
## 介绍

---

Markdown Preview 为 Atom 提供了将当前文档渲染成 HTML 并预览的功能。

## 截图

---





# IDEA Markdown

- [IDEA Markdown](#)
  - [介绍](#)
  - [截图](#)

## IDEA Markdown

---

主页：[IDEA Markdown](#)

应用：[IDEA](#) 等

费用：免费、开源

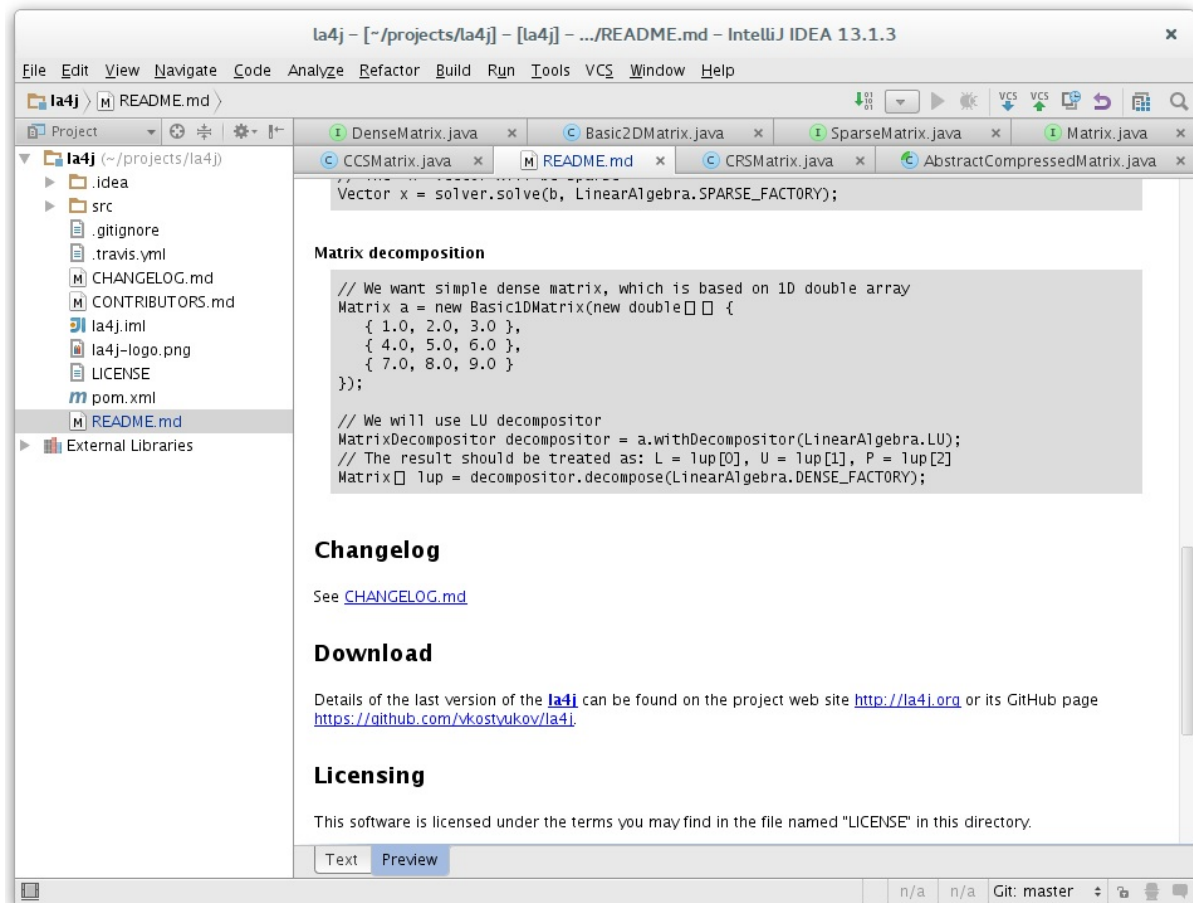
## 介绍

---

IDEA Markdown 为 IDEA 等 IDE 提供了 Markdown 支持。

## 截图

---



# Cmd Markdown

- [Cmd Markdown](#)
  - [介绍](#)
  - [截图](#)

## Cmd Markdown

---

主页: [Cmd Markdown](#)

平台: Windows OSX Linux Web

费用: 免费

技术: Web

## 介绍

---

Cmd Markdown 是一款使用 Web 技术开发的，全面且强大的 Markdown 编辑器。它拥有实时预览、离线编辑、智能同步、发布共享等功能，并提供了 Web 版和 Windows、OSX、Linux 平台的客户端，体验很棒。

## 截图

---

B I
🔗 “ <> 🖼️
☰ ☰ ☰ - ↶ ↷
注册 登录 发布 新文稿
☰ ☰ ⚙️ ⓘ

# 欢迎使用 Cmd Markdown 编辑阅读器
🔍 ⬆️ 💬 ?

-----

我们理解您需要更便捷更高效的工具记录思想，整理笔记、知识，并将其中承载的价值传播给他人，**\*\*Cmd Markdown\*\*** 是我们给出的答案 —— 我们为记录思想和分享知识提供更专业的工具。 您可以使用 Cmd Markdown：

- > \* 整理知识，学习笔记
- > \* 发布日记，杂文，所见所想
- > \* 撰写发布技术文稿（代码支持）
- > \* 撰写发布学术论文（LaTeX 公式支持）

![cmd-markdown-logo](https://www.zybuluo.com/static/img/logo.png)

除了您现在看到的这个 Cmd Markdown 在线版本，您还可以前往以下网址下载：

#### [Windows/Mac/Linux 全平台客户端](https://www.zybuluo.com/cmd/)

> 请保留此份 Cmd Markdown 的欢迎稿兼使用说明，如需撰写新稿件，点击顶部工具栏右侧的 *icon-file* **\*\*新文稿\*\*** 或者使用快捷键

☰ ⚙️ ⓘ 💬 🖼️

## 欢迎使用 Cmd Markdown 编辑阅读器

我们理解您需要更便捷更高效的工具记录思想，整理笔记、知识，并将其中承载的价值传播给他人，**Cmd Markdown** 是我们给出的答案 —— 我们为记录思想和分享知识提供更专业的工具。 您可以使用 Cmd Markdown：

- 整理知识，学习笔记
- 发布日记，杂文，所见所想
- 撰写发布技术文稿（代码支持）
- 撰写发布学术论文（LaTeX 公式支持）



# StackEdit

- [StackEdit](#)
  - [介绍](#)
  - [截图](#)

## StackEdit

---

主页：[StackEdit](#)

平台：Web

费用：免费、开源

技术：Web

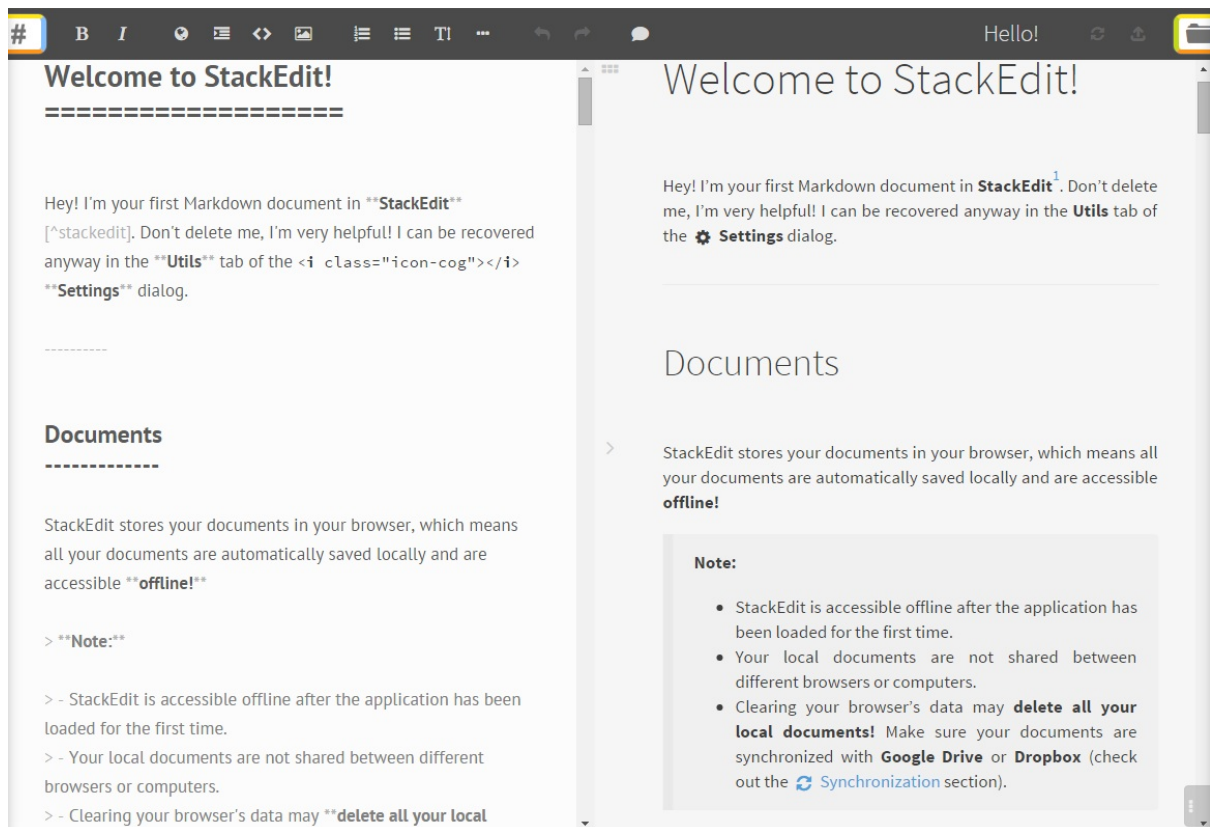
## 介绍

---

StackEdit 是一款基于 Web 技术开发的 Markdown 编辑器，它可以将编辑的文档发布到 Blogger、Wordpress，也可以与 Dropbox、Github、Google Drive 同步。

## 截图

---



# Dillinger

- [Dillinger](#)
  - [介绍](#)
  - [截图](#)

## Dillinger

主页: [Dillinger](#)

平台: Web

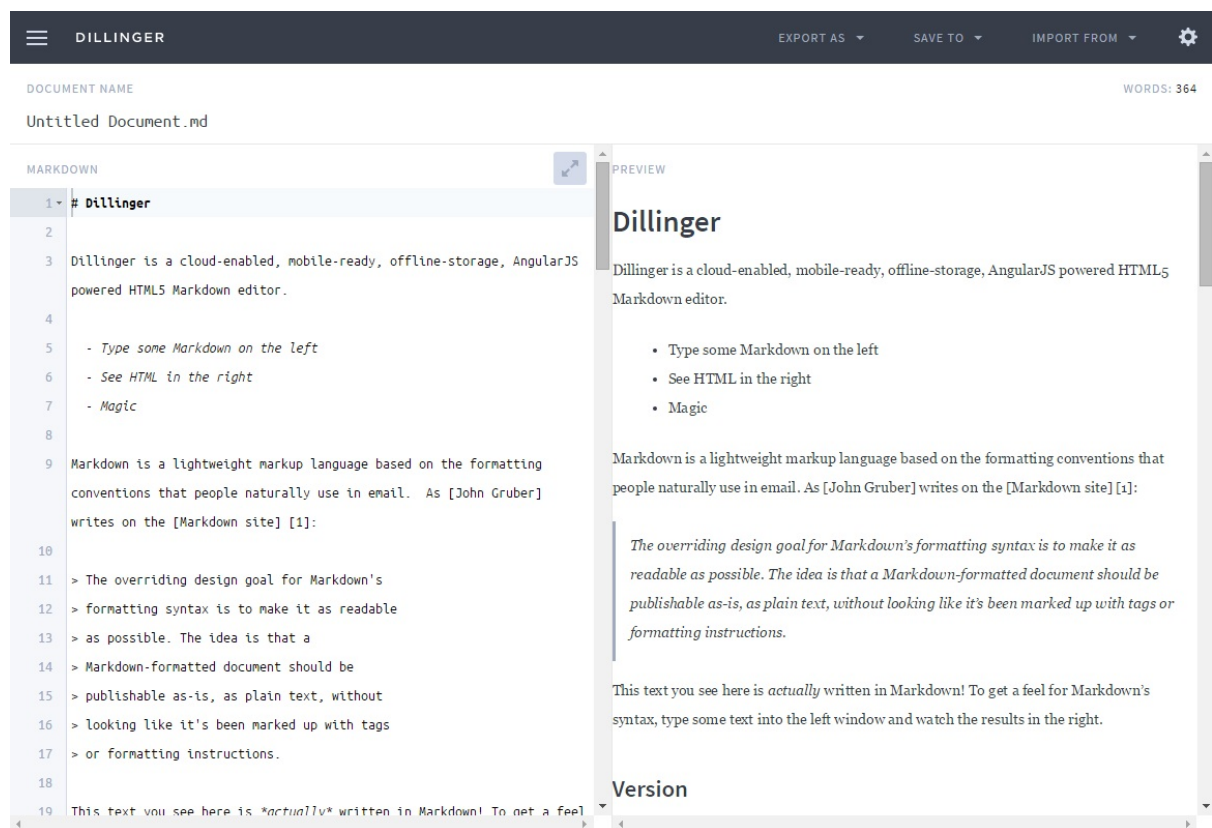
费用: 免费、开源

技术: Web

## 介绍

Dillinger 是一款类似于 [StackEdit](#) 的 Markdown 编辑器。

## 截图







# 格式转换

- [格式转换](#)

# 格式转换

---

Markdown 文档可以方便地转换为 HTML、Word、PDF 等格式的文档。这些转换既可以通过你正在使用的 Markdown 编辑器完成，也可以通过一些命令行工具（如 [Pandoc](#)、[Gitbook](#)）来完成，甚至可以用你熟悉的语言编程实现。

这个部分主要介绍通过编辑器或命令行工具来实现 Markdown 文档到下列格式的转换：

- [HTML](#)
- [PDF](#)
- [Word](#)

# HTML

- [转换为 HTML 文档](#)
  - [MdCharm](#)
  - [Pandoc](#)

## 转换为 HTML 文档

---

### MdCharm

---

选择 'File', 'Export to...', 勾选 'HTML', 点击 'Browser...' 选择导出目录并输入导出的文件名, 点击 'OK', 即可将当前的 Markdown 文档转换为 HTML 文档。

如果不满意 HTML 文档的样式, 可以在设置中自定义 CSS。

### Pandoc

---

参考 [Installing](#) 安装 Pandoc。

打开命令行, 进入文档所在目录:

```
1. cd /path/to/file/
```

执行下面的命令, 将 Markdown 转换为 HTML:

```
1. pandoc -o hello.html hello.md
```

默认的转换, 只是将 *Markdown* 内容转换为 *HTML* 标签, 所以只能看到浏览器的默认样式。

可以执行下面的命令, 为导出的 HTML 添加自定义样式:

```
1. pandoc -o hello.html -c style.css hello.md
```

`style.css` 仍然是以 `<link>` 的方式关联到 *HTML* 文档中的, 所以在发布的时候需要将 *CSS* 一同发布出去。

# PDF

- [转换为 PDF 文档](#)
  - [MdCharm](#)
  - [Pandoc](#)
  - [Chrome](#)

## 转换为 PDF 文档

---

### MdCharm

---

与导出 HTML 文档类似，选择 ‘File’，‘Export to...’，勾选 ‘PDF’，点击 ‘Browser...’ 选择导出目录并输入导出的文件名，点击 ‘OK’，即可将当前的 Markdown 文档转换为 PDF 文档。

如果不满意 PDF 文档的样式，可以在设置中自定义 CSS。

### Pandoc

---

使用 Pandoc 导出 PDF 文档，需要先安装某个 LaTeX 引擎（参考 [Creating a PDF](#)）。然后执行命令：

```
1. pandoc -o hello.pdf hello.md
```

当然，也可以通过 `-c style.css` 来指定样式文件。

### Chrome

---

在将 Markdown [转换为 HTML 文档](#) 之后，可以通过 [Chrome 浏览器](#) 打开它。选择 ‘打印’（Ctrl+P），然后更改 ‘目标打印机’ 为 ‘另存为 PDF’，再进行一些设置后，即可保存为 PDF 文档。

# Word

- [转换为 Word 文档](#)
  - [复制粘贴](#)
  - [Pandoc](#)

## 转换为 Word 文档

---

### 复制粘贴

---

在导出为 HTML 文档之后，可以（在浏览器中）手动复制 HTML 页面的内容，然后粘贴到 Word 文档中，保存即可。

### Pandoc

---

执行下面的命令，即可将 Markdown 文档转换为 Word 文档：

```
1. pandoc -o hello.docx hello.md
```