

Windows 下博客编辑利器 工具 Typora

认识与入门 Markdown

[Markdown](#) 是一种轻量级的「标记语言」，它的优点很多，目前也被越来越多的写作爱好者，撰稿者广泛使用。看到这里请不要被「标记」、「语言」所迷惑，Markdown 的语法十分简单。常用的标记符号也不超过十个，这种相对于更为复杂的 HTML 标记语言来说，Markdown 可谓是十分轻量的，学习成本也不需要太多，且一旦熟悉这种语法规则，会有一劳永逸的效果。

认识 Markdown

在刚才的导语里提到，Markdown 是一种用来写作的轻量级「标记语言」，它用简洁的语法代替排版，而不像一般我们用的字处理软件 Word 或 Pages 有大量的排版、字体设置。它使我们专心于码字，用「标记」语法，来代替常见的排版格式。例如此文从内容到格式，甚至插图，键盘就可以通通搞定了。

目前来看，支持 Markdown 语法的编辑器有很多，包括很多网站（例如 [简书](#)）也支持了 Markdown 的文字录入。Markdown 从写作到完成，导出格式随心所欲，你可以导出 HTML 格式的文件用来网站发布，也可以十分方便的导出 PDF 格式，这种格式写出的简历更能得到 HR 的好感，甚至可以利用 [CloudApp](#) 这种云服务工具直接上传至网页用来分享你的文章，全球最大的轻博客平台 Tumblr，也支持使用 Mou 这类 Markdown 工具进行编辑并直接上传。

Markdown 官方文档

这里可以看到官方的 Markdown 语法规则文档，当然，后文我也会用自己的方式，阐述这些语法在实际使用中的用法。

- [创始人 John Gruber 的 Markdown 语法说明](#)
- [Markdown 中文版语法说明](#)

使用 Markdown 的优点

- 专注你的文字内容而不是排版样式，安心写作。
- 轻松的导出 HTML、PDF 和本身的 .md 文件。
- 纯文本内容，兼容所有的文本编辑器与字处理软件。
- 随时修改你的文章版本，不必像字处理软件生成若干文件版本导致混乱。
- 可读、直观、学习成本低。

使用 Markdown 的误区

Markdown 旨在简洁、高效，也由于 Markdown 的易读易写，人们用不同的编程语言实现了多个版本的解析器和生成器。这就导致了目前不同的 Markdown 工具集成了不同的功能（基础功能大致相同），例如流程图与时序图，复杂表格与复杂公式的呈现。

虽然功能的丰富并没有什么本质的缺点，但终归有些背离初衷，何况在编写的过程中很费神，不如使用专业的工具撰写来的更有效率，所以如果你需实现复杂功能，专业的图形界面工具会更加方便。

当然，如果你对折腾这些不同客户端对 Markdown 的定制所带来高阶功能感到愉悦的话，那也是无可厚非的。

Markdown 语法的简要规则

标题

用 Markdown 书写时，只需要在文本前面加上『# 』即可创建一级标题。同理，创建二级标题、三级标题等只需要增加『# 』个数即可，Markdown 共支持六级标题。如下所示：

```
# 一级标题
## 二级标题
### 三级标题
#### 四级标题
##### 五级标题
##### 六级标题
```

点击预览可以看到效果：

点击预览可以看到效果：



锚点

Coding 会针对每个标题，在解析时都会添加锚点 id，如

```
# 锚点
```

会被解析成：

```
<h1 id="user-content-锚点">锚点</h1>
```

注意我们添加了一个 user-content- 的前缀，所以如果要自己添加跳转链接要使用 Markdown 的形式，且链接要加一个 user-content- 前缀，如：

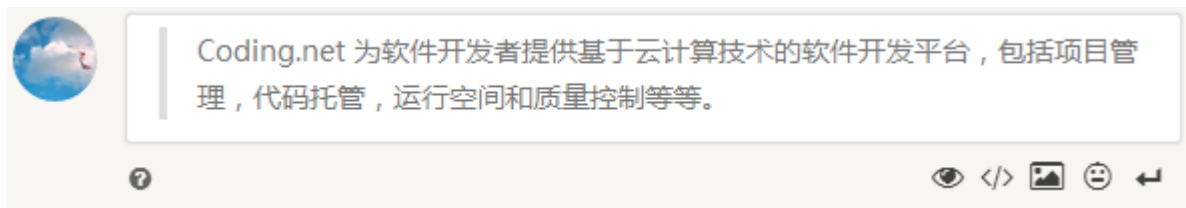
[访问链接](#user-content-锚点);

引用

Markdown 标记区块引用和 email 中用『>』的引用方式类似，只需要在整个段落的第一行最前面加上『>』：

> Coding.net 为软件开发者提供基于云计算技术的软件开发平台，包括项目管理，代码托管，运行空间和质量控制等等。

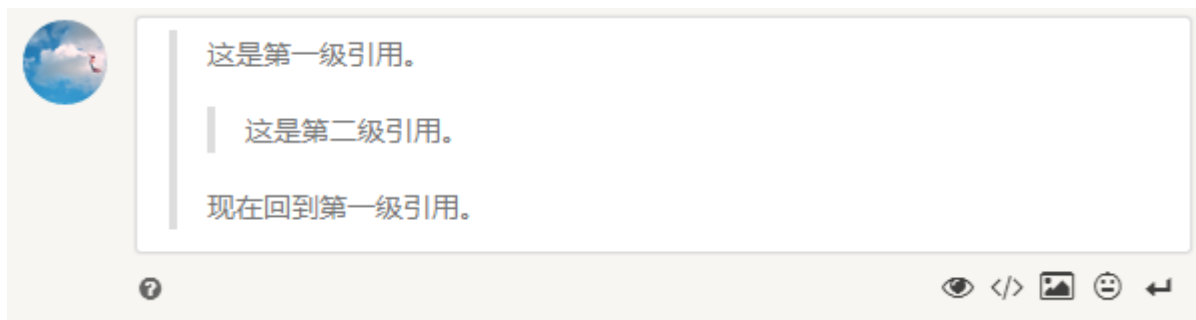
效果图如下：



区块引用可以嵌套，只要根据层次加上不同数量的『>』：

> 这是第一级引用。
>
> > 这是第二级引用。
>
> 现在回到第一级引用。

效果图如下：



引用的区块内也可以使用其他的 Markdown 语法，包括标题、列表、代码区块等：

> ## 这是一个标题。
> 1. 这是第一行列表项。
> 2. 这是第二行列表项。
>
> 给出一些例子代码：
>
> return shell_exec(`echo \$input | \$markdown_script`);

效果图如下：

这是一个标题

1. 这是第一行列表项。
2. 这是第二行列表项。

给出一些例子代码：

```
return shell_exec(echo $input | $markdown_script);
```

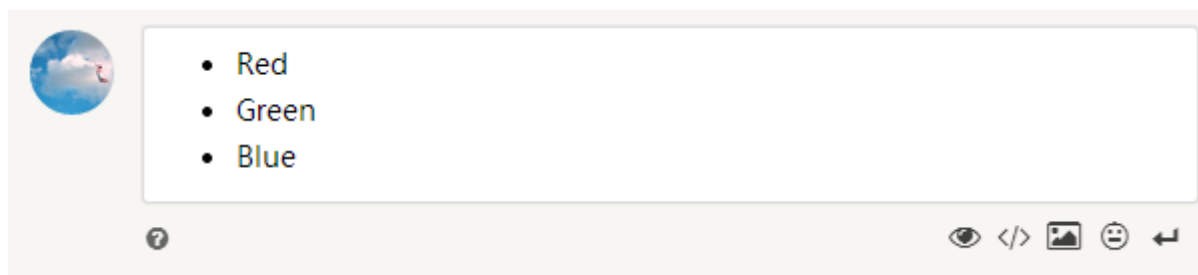
列表

列表项目标记通常放在最左边，项目标记后面要接一个字符的空格。

无序列表：使用星号、加号或是减号作为列表标记

- Red
- Green
- Blue

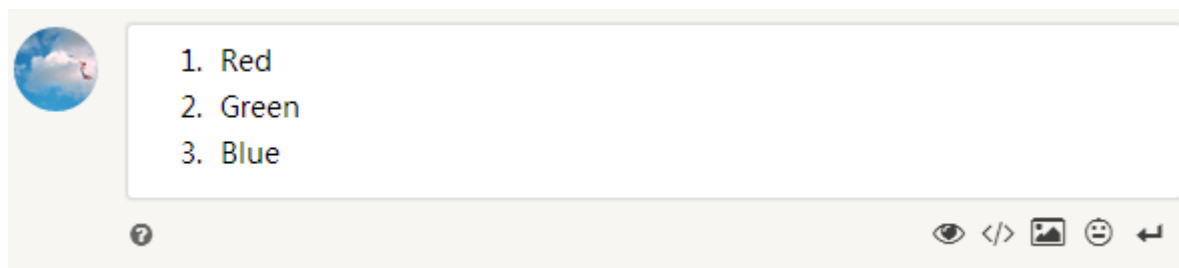
效果图如下：



有序列表：使用数字接着一个英文句点

1. Red
2. Green
3. Blue

效果图如下：



如果要在列表项目内放进引用，那『>』就需要缩进：

- * Coding.net有以下主要功能:
 - > 代码托管平台
 - > 在线运行环境
 - > 代码质量监控
 - > 项目管理平台

效果图如下:



代办列表: 表示列表是否勾选状态 (注意: [] 前后都要有空格)

- [] 不勾选
- [x] 勾选

效果图如下:



代码

只要把你的代码块包裹在“`”之间，你就不需要通过无休止的缩进来标记代码块了。在围栏式代码块中，你可以指定一个可选的语言标识符，然后我们就可以为它启用语法着色了。举个例子，这样可以为一段 Ruby 代码着色：

```
```ruby
require 'redcarpet'
markdown = Redcarpet.new("Hello World!")
puts markdown.to_html
```
```

效果图如下:



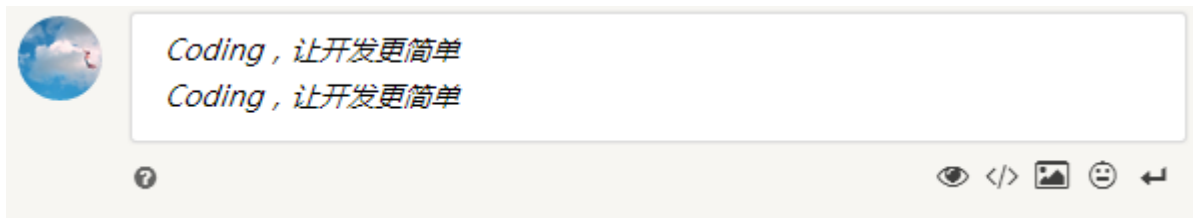
强调

在Markdown中，可以使用 * 和 _ 来表示斜体和加粗。

斜体：

```
*Coding, 让开发更简单*
_Coding, 让开发更简单_
```

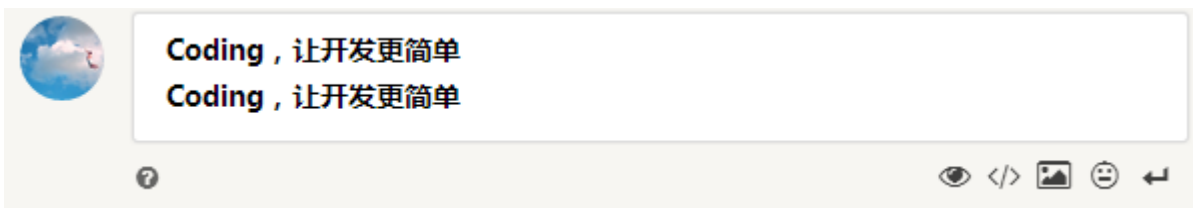
效果图如下：



加粗：

```
**Coding, 让开发更简单**
__Coding, 让开发更简单__
```

效果图如下：

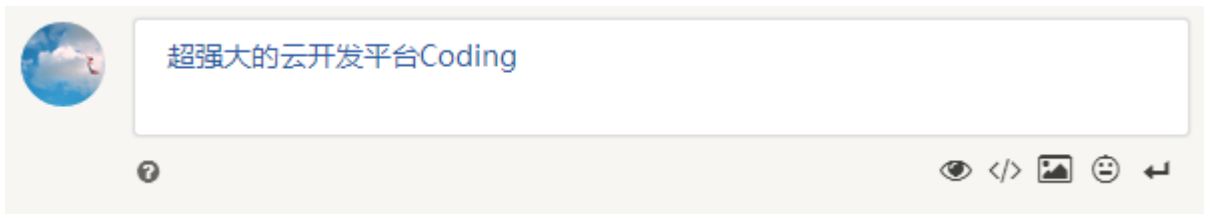


自动链接

方括号显示说明，圆括号内显示网址，Markdown 会自动把它转成链接，例如：

```
[超强大的云开发平台Coding](http://coding.net)
```

效果图如下：



表格

在 Markdown 中，可以制作表格，例如：

```
First Header	Second Header	Third Header
Content Cell | Content Cell | Content Cell
Content Cell | Content Cell | Content Cell
```

效果图如下：

| First Header | Second Header | Third Header |
|--------------|---------------|--------------|
| Content Cell | Content Cell | Content Cell |
| Content Cell | Content Cell | Content Cell |

或者也可以让表格两边内容对齐，中间内容居中，例如：

```
First Header	Second Header	Third Header
Left        | Center       | Right
Left        | Center       | Right
```

效果图如下：

| First Header | Second Header | Third Header |
|--------------|---------------|--------------|
| Left | Center | Right |
| Left | Center | Right |

分割线

在 Markdown 中，可以使用 3 个以上『-』符号制作分割线，例如：

```
这是分隔线上部分内容
---
这是分隔线上部分内容
```

效果图如下：



图片

Markdown 使用了类似链接的语法来插入图片, 包含两种形式: **内联** 和 **引用**.

内联图片语法如下:

```
![Alt text](/path/to/img.jpg)
或
![Alt text](/path/to/img.jpg "Optional title")
```

也就是:

- 一个惊叹号『!』
- 接着一个方括号，里面是图片的替代文字
- 接着一个普通括号，里面是图片的网址，最后还可以用引号包住并加上 选择性的『title』文字。

引用图片语法如下:

```
![Alt text][id]
```

『id』 是图片引用的名称. 图片引用使用链接定义的同语法:

```
[id]: url/to/image "Optional title attribute"
```

流程图

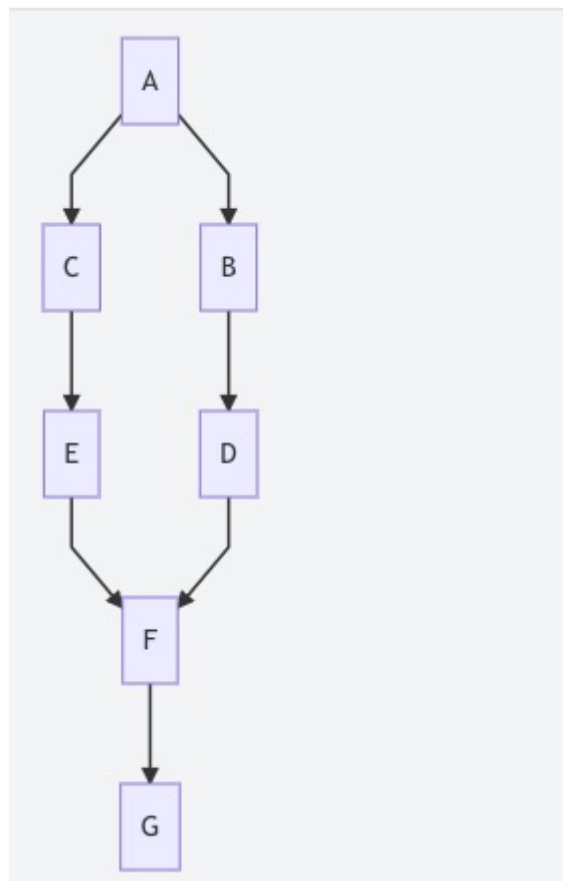
Markdown 编辑器已支持绘制流程图、时序图和甘特图。通过 mermaid 实现图形的插入，点击查看 [更多语法详情](#)。


```

graph TD
    A-->B;
    A-->C;
    B-->D;
    C-->E;
    E-->F;
    D-->F;
    F-->G;

```

效果图如下：



时序图

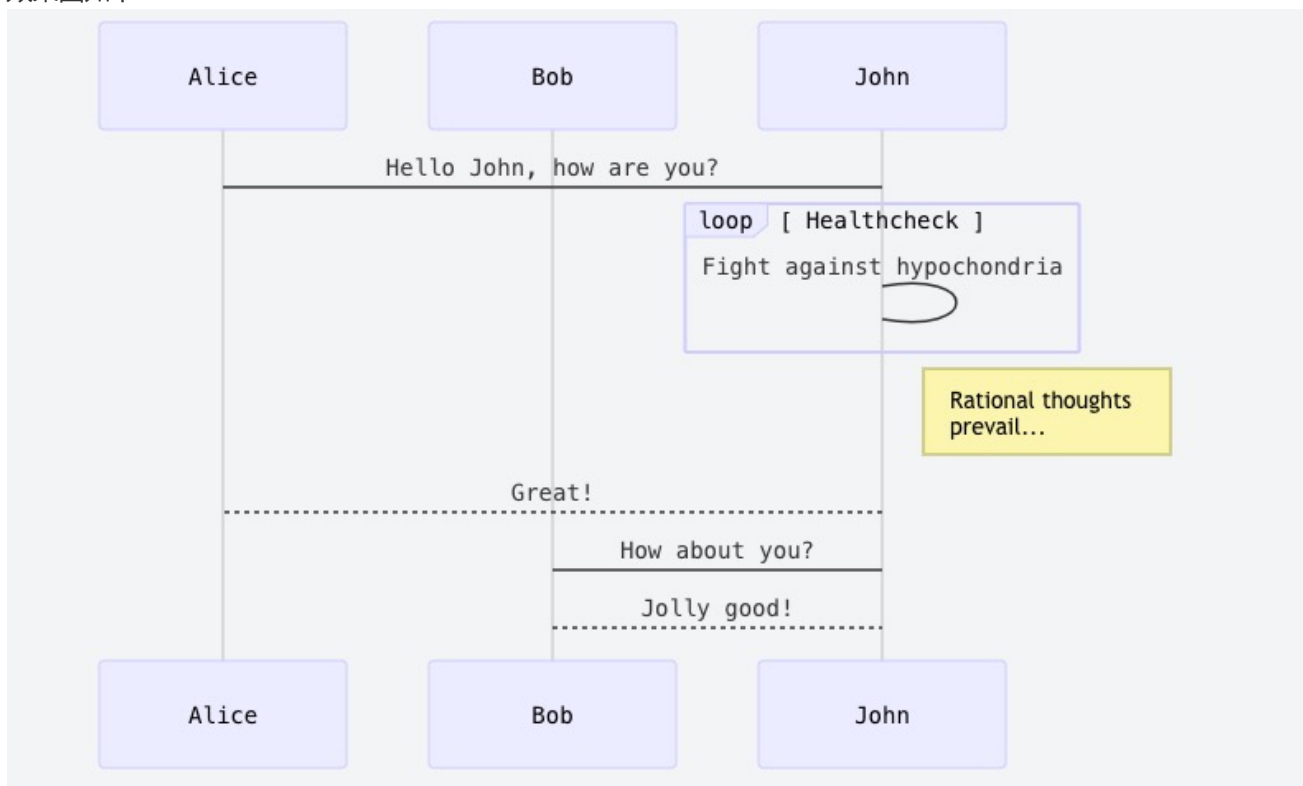
```

graph TD
    participant Alice
    participant Bob
    Alice->John: Hello John, how are you?
    loop Healthcheck
        John->John: Fight against hypochondria
    end
    Note right of John: Rational thoughts
    prevail...
    John-->Alice: Great!

```

John->Bob: How about you?
Bob-->John: Jolly good!
...

效果图如下:



甘特图

```
```graph
ganttt

dateFormat YYYY-MM-DD
title Adding GANTT diagram functionality to mermaid
section A section
Completed task :done, des1, 2014-01-06,2014-01-08
Active task :active, des2, 2014-01-09, 3d
Future task : des3, after des2, 5d
Future task2 : des4, after des3, 5d
section Critical tasks
Completed task in the critical line :crit, done, 2014-01-06,24h
Implement parser and jison :crit, done, after des1, 2d
Create tests for parser :crit, active, 3d
Future task in critical line :crit, 5d
Create tests for renderer :2d
Add to mermaid :1d
```
```

效果图如下:

Adding GANTT diagram functionality to mermaid

