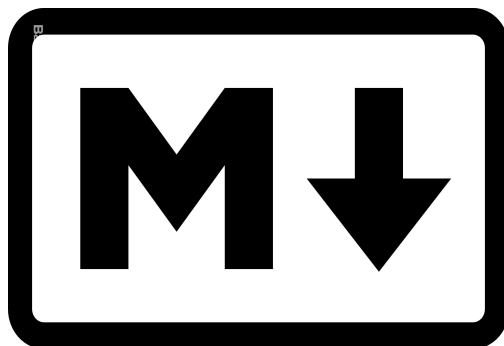


Table of Contents

简介	1.1
markdown标题	1.2
markdown段落格式	1.3
markdown列表	1.4
markdown区块	1.5
markdown代码	1.6
markdown链接	1.7
markdown图片	1.8
markdown表格	1.9
markdown高级用法	1.10

Markdown简介



Markdown 是一种轻量级标记语言，它允许人们使用易读易写的纯文本格式编写文档。

Markdown 语言在 2004 由约翰·格鲁伯（英语：John Gruber）创建。

Markdown 编写的文档可以导出 HTML、Word、图像、PDF、

Epub 等多种格式的文档。

Markdown 编写的文档后缀为 `.md`、`.markdown`。

gitbook git markdown三者的关系和运用

实际上，GitBook 是一个基于 Node.js 的命令行工具，支持 Markdown 和 AsciiDoc 两种语法格式，可以输出 HTML、PDF、eBook 等格式的电子书。所以可以把 GitBook 定义为文档格式转换工具。

因此，通常会选择合适的 Markdown 编辑工具以获得飞一般的写作体验；使用 GitBook 管理文档，预览、制作电子书；同时通过 Git 管理书籍内容的变更，并将其托管到云端（比如 GitHub、GitLab、码云，或者是自己搭建的 Git 服务器），实现多人协作。

为什么选择gitbook+git+typora

通常，我们最开始学习和使用的办公软件就是 Word、Excel 和 PowerPoint。这里不是说它们已经过时了，不是说 GitBook 能够替代它们。

相反，Microsoft 的办公软件很优秀并且经受了时间的考验，但是正因为它功能丰富，导致稍显臃肿（二八定律：80%的时间里我们只会只用20%的功能），同时因为它存在以二进制格式保存、软件不兼容、格式不兼容、难以进行版本控制、难以实时分享预览、难以多人协作等短板。而这恰恰是 GitBook + Markdown + Git 的长处。

简单来说，GitBook + Markdown + Git 带来的好处有：

- 语法简单
- 兼容性强
- 导出方便
- 专注内容
- 团队协作

Markdown 应用

Markdown 能被使用来撰写电子书、编辑文档，记笔记。如：Gitbook。

当前许多网站都广泛使用 Markdown 来撰写帮助文档或是用于论坛上发表消息。例如：GitHub、简书、reddit、Diaspora、Stack Exchange、OpenStreetMap、SourceForge等。

编辑器

我将使用 Typora 编辑器来讲解 Markdown 的语法，Typora 支持 MacOS、Windows、Linux 平台，且包含多种主题，编辑后直接渲染出效果。

支持导出HTML、PDF、Word、图片等多种类型文件。

Markdown 标题

1、使用 = 和 - 标记一级和二级标题

= 和 - 标记语法格式如下：

1 **我显示一级标题**
=====

我显示二级标题

2、使用 # 号标记

使用 # 号可表示 1-6 级标题，一级标题对应一个 # 号，二级标题对应两个 # 号，以此类推。

演示：

Markdown 段落

1、段落的换行是使用两个以上空格加上回车



2、也可以在段落后面使用一个空行来表示重新开始一个段落



字体

Markdown 可以使用以下几种字体：

```
*斜体文本*  
_斜体文本_  
**粗体文本**  
__粗体文本__  
***粗斜体文本***  
___粗斜体文本___
```

分隔线

你可以在一行中用三个以上的星号、减号、底线来建立一个分隔线，行内不能有其他东西。你也可以在星号或是减号中间插入空格。下面每种写法都可以建立分隔线：

```
***  
  
* * *  
  
*****  
  
- - -  
  
-----
```

删除线

如果段落上的文字要添加删除线，只需要在文字的两端加上两个波浪线 ~~ 即可

```
gitbook  
typora  
~~typora~~
```

下划线

下划线可以通过 HTML 的 标签来实现：

```
<u>带下划线文本</u>
```

脚注

脚注是对文本的补充说明。

Markdown 脚注的格式如下：

```
[^要注明的文本]
```

创建脚注格式类似这样 [typora](#)。

```
typora . well done! ↩
```

Markdown 列表

Markdown 支持有序列表和无序列表。

1、无序列表使用星号(*)、加号(+)或是减号(-)作为列表标记:

- 第一项
 - 第二项

2、有序列表使用数字并加上 . 号来表示, 如:

1. 第一项
2. 第二项
3. 第三项

3、嵌套列表

列表嵌套只需在子列表中的选项添加四个空格即可

1. 第一项:
 - 第一项嵌套的第一个元素
 - 第一项嵌套的第二个元素
2. 第二项:
 - 第二项嵌套的第一个元素
 - 第二项嵌套的第二个元素

Markdown 区块

Markdown 区块引用是在段落开头使用 > 符号，然后后面紧跟一个空格符号

```
>## typora
> gitbook
> github
```

typora

gitbook

github

另外区块是可以嵌套的，一个 > 符号是最外层，两个 > 符号是第一层嵌套

```
> 最外层
> > 第一层嵌套
> > > 第二层嵌套
```

最外层

第一层嵌套

第二层嵌套

区块中使用列表

```
> 区块中使用列表
> 1. 第一项
> 2. 第二项
>   + 第一项
>   + 第二项
>   + 第三项
```

区块中使用列表

1. 第一项

2. 第二项

- 第一项
- 第二项
- 第三项

列表中使用区块

如果要在列表项目内放进区块，那么就需要在 > 前添加四个空格的缩进

- * 第一项
 - > 菜鸟教程
 - > 学的不仅是技术更是梦想
- * 第二项

- 第一项
 - typora gitbook
- 第二项

Markdown 代码

如果是段落上的一个函数或片段的代码可以用反引号把它包起来（`）

```
`printf()`
```

printf() 函数

代码区块

代码区块使用 **4** 个空格或者一个制表符（**Tab** 键）

```
<?php
echo 'typora';
function test() {
    echo 'test'; }
```

```
<?php
echo 'typora';
function test() {
    echo 'test'; }
```

用 ``` 包裹一段代码，并指定一种语言（也可以不指定）

```
```javascript
$(document).ready(function () {
 alert('RUNOOB');
});
```

```
```javascript
$(document).ready(function () {
    alert('RUNOOB');
});
```

Markdown 链接

链接使用方法

```
[链接名称](链接地址)
```

或者

```
<链接地址>
```

```
这是一个链接 [百度大法好](https://www.baidu.com)
```

这是一个链接 [百度大法好](https://www.baidu.com)

直接使用链接地址

```
<https://www.runoob.com>
```

<https://www.baidu.com>

高级链接

链接也可以用变量来代替，文档末尾附带变量地址：

这个链接用 1 作为网址变量 [Google][1]

这个链接用 runoob 作为网址变量 [Runoob][runoob]

然后在文档的结尾为变量赋值（网址）

```
[1]: http://www.google.com/
```

```
[runoob]: http://www.runoob.com/
```

链接也可以用变量来代替，文档末尾附带变量地址： 这个链接用 1 作为网址变量 [豆瓣](#) 这个链接用 2 作为网址变量 [百度](#) 然后在文档的结尾为变量赋值（网址）

链接也可以用变量来代替，文档末尾附带变量地址：

这个链接用 1 作为网址变量 [豆瓣](#)[1]

这个链接用 1 作为网址变量 [百度](#)[2]

然后在文档的结尾为变量赋值（网址）

```
[1]: http://www.douban.com
```

```
[2]:http://www.baidu.com
```

markdown图片

Markdown 图片语法格式:

开头一个感叹号！接着一个方括号，里面放上图片的替代文字 接着一个普通括号，里面放上图片的网址，最后还可以用引号包住并加上选择性的 'title' 属性的文字。

```
![alt 属性文本](图片地址)
```

```
![alt 属性文本](图片地址 "可选标题")
```



或像网址那样对图片网址使用变量

这个链接用 1 作为网址变量 [火影][1].
然后在文档的结尾位变量赋值（网址）

```
[1]:picture.assets/hy.jpg
```

这个链接用 1 作为网址变量 火影. 然后在文档的结尾位变量赋值（网址）

现在Markdown 还没有办法指定图片的高度与宽度，如果你需要的话，你可以使用普通的img 标签

```

```

Markdown 表格

Markdown 制作表格使用 | 来分隔不同的单元格，使用 - 来分隔表头和其他行

```
| 表头 | 表头 |  
| ---- | ---- |  
| 单元格 | 单元格 |  
| 单元格 | 单元格 |
```

表头	表头
单元格	单元格
单元格	单元格

我们可以设置表格的对齐方式：

- -: 设置内容和标题栏居右对齐。
- :- 设置内容和标题栏居左对齐。
- :-: 设置内容和标题栏居中对齐。

```
| 左对齐 | 右对齐 | 居中对齐 |  
| :-----| -----: | :-----: |  
| 单元格 | 单元格 | 单元格 |  
| 单元格 | 单元格 | 单元格 |
```

左对齐	右对齐	居中对齐
单元格	单元格	单元格
单元格	单元格	单元格

Markdown 高级技巧

支持的 HTML 元素

不在 Markdown 涵盖范围之内的标签，都可以直接在文档里面用 HTML 撰写。

目前支持的 HTML 元素有：<kbd> <i> <sup> <sub>
 等

```
使用 <kbd>Ctrl</kbd>+<kbd>Alt</kbd>+<kbd>Del</kbd> 重启电脑
```

使用 Ctrl+Alt+Del 重启电脑

转义

Markdown 使用了很多特殊符号来表示特定的意义，如果需要显示特定的符号则需要使用转义字符，Markdown 使用反斜杠转义特殊字符

```
**文本加粗**  
\*\* 正常显示星号 \*\*
```

文本加粗 ** 正常显示星号 **

Markdown 支持以下这些符号前面加上反斜杠来帮助插入普通的符号

```
\  反斜线  
` 反引号  
*  星号  
_  下划线  
{ } 花括号  
[ ] 方括号  
( ) 小括号  
#  井字号  
+  加号  
-  减号  
.  英文句点  
!  感叹号
```

typora 画流程图、时序图(顺序图)、甘特图

```
graph LR  
A[方形] --> B(圆角)  
B --> C{条件a}  
C -->|a=1| D[结果1]  
C -->|a=2| E[结果2]  
F[横向流程图]
```

```

graph TD
A[方形] --> B(圆角)
    B --> C{条件a}
    C --> |a=1| D[结果1]
    C --> |a=2| E[结果2]
    F[竖向流程图]

```

```

st=>start: 开始框
op=>operation: 处理框
cond=>condition: 判断框(是或否?)
sub1=>subroutine: 子流程
io=>inputoutput: 输入输出框
e=>end: 结束框
st->op->cond
cond(yes)->io->e
cond(no)->sub1(right)->op

```

```

对象A->对象B: 对象B你好吗?(请求)
Note right of 对象B: 对象B的描述
Note left of 对象A: 对象A的描述(提示)
对象B-->对象A: 我很好(响应)
对象A->对象B: 你真的好吗?

```

```

Title: 标题: 复杂使用
Typora->gitbook: gitbook你好吗?(请求)
Note right of gitbook: gitbook的描述
Note left of Typora: Typora的描述(提示)
gitbook-->Typora: 我很好(响应)
gitbook->github: 你好吗
github-->>Typora: gitbook找我了
Typora->gitbook: 你真的好吗?
Note over github,gitbook: 我们是朋友

```

```

participant node.js
Note right of node.js: 没人陪我玩

```

```

%% 语法示例

gantt
dateFormat YYYY-MM-DD
title 软件开发甘特图
section 设计
    需求                :done,    des1, 2019-11-16,1d
    原型                :active,  des2, 3d
    UI设计              :des3, after des2, 5d
    未来任务            :des4, after des3, 5d
section 开发
    学习准备理解需求    :crit, done, 2019-11-16,24h
    设计框架            :crit, done, after des2, 2d
    开发                :crit, active, 3d
    未来任务            :crit, 5d
    要                  :2d
section 测试
    功能测试            :active, a1, after des3, 3d
    压力测试            :after a1 , 20h
    测试报告            : 24h

```

mermaid: 图中红框第一行内容与最后一行内容，表示当前使用的语言为mermaid。 任务行：

- **done** 已完成
- **active** 正在进行
- **crit** 关键任务 任务状态：

任务描述: 在**des1**、**des2**位置添加任务描述，其它任务引用时直接引用**des1**就可以。**after:** 描述任务时间关系。**des3, after des2**表示**des3**紧跟在**des2**之后。 任务时长有三种方式：

- 时间范围 如：2014-01-06,2014-01-08
- 指定天数 如：5d
- 指定开始日期+天数 如：2014-01-06, 5d