

# 目录

前言	1.1
Gitbook简介	1.2
Gitbook命令行工具	1.3
电子书制作过程	1.4
Gitook插件	1.5
代码高亮	1.5.1
按钮	1.5.2
分享	1.5.3
脚注	1.5.4
提示和标注	1.5.5
章节折叠	1.5.6
Google Analysis统计	1.5.7
打赏	1.5.8
其他插件	1.5.9
配置Gitbook	1.6
多种输出格式	1.7
一键发布	1.8
Gitbook Editor	1.9
注意事项和使用心得	1.10
附录	1.11
参考资料	1.11.1

# 电子书制作利器：GitBook

- 最新版本： v1.0
- 更新时间： 20180702

## 简介

对电子书制作利器 GitBook 做个总结，包括电子书制作过程、常用插件、book.json的常用配置、利用 Makefile 实现一键发布、如何输出多种格式(html/pdf epub/mobi)，客户端Gitbook Editor的使用、一些常见的坑等等。

## 源码+浏览+下载

本书的各种源码、在线浏览地址、多种格式文件下载如下：

### Gitook源码

- [crifan/ebook\\_system\\_gitbook](#)

### 如何使用此Gitbook源码去生成发布为电子书

详见：[crifan/gitbook\\_template: demo how to use crifan gitbook template and demo](#)

### 在线浏览

- [电子书制作利器：GitBook book.crifan.com](#)
- [电子书制作利器：GitBook crifan.github.io](#)

### 离线下载阅读

- [电子书制作利器：GitBook PDF](#)
- [电子书制作利器：GitBook ePUB](#)
- [电子书制作利器：GitBook Mobi](#)

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件  
修订时间： 2018-07-02 18:15:22

# Gitbook简介

当谈论 gitbook 这个词时，有两方面的含义：

- gitbook.com 网站
  - gitbook.com 提供服务，可以在网站上面写 markdown ，系统帮你生成 电子书
  - 生成的电子书系统中，支持在线网页版和下载 PDF 、 ePub 、 Mobi 等其他格式
  - 现在已有很多人在gitbook.com上写了书，感兴趣的可以自己去找找看：
    - <https://legacy.gitbook.com/explore>
- gitbook系统：gitbook.com提供的免费的工具
  - 可以自己拿来使用去生成电子书的成套系统
  - 此外还提供了相关的客户端Gitbook Editor，以所见即所得的方式用Markdown去编写电子书

此处主要介绍第二种的Gitbook： Gitbook系统和相关的GitBook Editor客户端编辑器

可以总结为：

- gitbook是gitbook.com出的，可以用markdown的编写自己的 一套电子书制作的系统和工具
- GitBook Editor是gitbook.com出的，可以用markdown的编写自己的电子书的 桌面客户端

## 其他相关内容

其他还有常被拿来和Gitbook去对比的其他一些markdown的工具：

【整理】文档制作工具：[MkDocs](#) [Gitbook](#)等

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件  
修订时间： 2018-07-02 18:13:06

# Gitbook命令行工具

gitbook本身是套系统

除了有GUI客户端：[Gitbook Editor](#)以图形化方式让你写 book 外，还有相关的命令行工具 gitbook

对应各种命令：

TODO:

抽空把：

命令 · gitbook-guide

<https://zq99299.gitbooks.io/gitbook-guide/content/chapter/command.html>

整理过来。

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件

修订时间： 2018-07-02 16:49:20

# 电子书制作过程

新建一个 Gitbook 电子书，主要有两种方式：

- `Gitbook Editor`：用Gitbook Editor创建book+同步到远端git仓库
  - 关于 `Gitbook Editor` 的使用，详见：[Gitbook Editor 章节](#)
- 纯命令行：`gitbook-cli` 及相关的成套工具
  - 直接用 `git` 创建本地仓库
  - 用 `gitbook init` 创建本地的 `book`
  - 用 `git` 同步到远端仓库

下面就来介绍，如何用 `gitbook` 的命令行工具，去制作自己的电子书。

## 准备工作

我们需要一个 `git` 代码仓库，用于保存自己的 `gitbook` 的源码。

此处选用 `github`。所以先去github中，新建一个空的git仓库，比如：

[https://github.com/crifan/improve\\_work\\_efficiency.git](https://github.com/crifan/improve_work_efficiency.git)

## 初始化 gitbook

用：

```
gitbook init
```

去创建一个空的 `book`：

举例：

```
→ Import gitbook init improve_work_efficiency
warn: no summary file in this book
info: create README.md
info: create SUMMARY.md
info: initialization is finished
```

## 编辑 gitbook 的 markdown 源码

去编辑相关的 `gitbook` 的 `markdown` 的源码，以及相关的 `book.json` 等内容

比如我去拷贝别的book的内容过来后，文件如下：

```
→ improve_work_efficiency git:(master) ✘ git status
On branch master
```

```
No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)

  .gitignore
  Makefile
  README.md
  book.json
  src/

nothing added to commit but untracked files present (use "git add" to track)
```

然后再继续去编辑内容

## gitbook install 安装插件

在 book.json 配置好之后，要去安装对应的 gitbook 的插件。

```
improve_work_efficiency git:(master) ✘ gitbook install
info: installing 16 plugins using npm@3.9.2
info:
info: installing plugin "theme-comscore"
info: install plugin "theme-comscore" (*) from NPM with version 0.0.3
/Users/crifan/GitBook/Library/Import/improve_work_efficiency
└── gitbook-plugin-theme-comscore@0.0.3

info: >> plugin "theme-comscore" installed with success
info:
info: installing plugin "search-plus"
runTopLevelLifecycle → 3 | ██████████| ...
/Users/crifan/GitBook/Library/Import/improve_work_efficiency
├── gitbook-plugin-search-plus@1.0.3
│   └── html-entities@1.2.0
└── gitbook-plugin-theme-comscore@0.0.3

...
info: >> plugin "alerts" installed with success
info:
info: installing plugin "toolbar-button"
postinstall:gitbook-plugi | ██████████| ...
/Users/crifan/GitBook/Library/Import/improve_work_efficiency
├── gitbook-plugin-alerts@0.2.0
├── gitbook-plugin-copy-code-button@0.0.2
├── gitbook-plugin-disqus@0.1.0
├── gitbook-plugin-donate@1.0.2
├── gitbook-plugin-expandable-chapters-small@0.1.7
├── gitbook-plugin-ga@1.0.1
└── gitbook-plugin-github-buttons@3.0.0
```

```

├── gitbook-plugin-prism@2.3.0
├── gitbook-plugin-prism-themes@0.0.2
├── gitbook-plugin-search-plus@1.0.3
├── gitbook-plugin-sharing-plus@0.0.2
├── gitbook-plugin-sitemap-general@0.1.1
├── gitbook-plugin-splitter@0.0.8
├── gitbook-plugin-tbfed-pagefooter@0.0.1
├── gitbook-plugin-theme-comscore@0.0.3
└── gitbook-plugin-toolbar-button@0.0.1

info: >> plugin "toolbar-button" installed with success

```

## 本地调试

然后用我自己写的一键发布中的 `Makefile` 去调试：

```

→ improve_work_efficiency git:(master) ✘ make debug
-----
Author : crifan.com
Version : 20171231
Function: Auto use gitbook to generated files: website/pdf epub/mobi; upload to remote server; commit to github io repo
Run 'make help' to see usage
-----
rm -rf /Users/crifan/GitBook/Library/Import/improve_work_efficiency/debug
mkdir -p /Users/crifan/GitBook/Library/Import/improve_work_efficiency/debug
gitbook serve /Users/crifan/GitBook/Library/Import/improve_work_efficiency /Users/crifan/GitBook/Library/Import/improve_work_efficiency/debug
Live reload server started on port: 35729
Press CTRL+C to quit ...

info: 23 plugins are installed
info: 19 explicitly listed
info: loading plugin "search-plus"... OK
info: loading plugin "disqus"... OK
info: loading plugin "prism"... OK
info: loading plugin "prism-themes"... OK
info: loading plugin "github-buttons"... OK
info: loading plugin "splitter"... OK
info: loading plugin "sharing-plus"... OK
info: loading plugin "tbfed-pagefooter"... OK
info: loading plugin "expandable-chapters-small"... OK
info: loading plugin "ga"... OK
info: loading plugin "donate"... OK
info: loading plugin "sitemap-general"... OK
info: loading plugin "copy-code-button"... OK
info: loading plugin "alerts"... OK
info: loading plugin "toolbar-button"... OK
info: loading plugin "livereload"... OK
info: loading plugin "fontsettings"... OK
info: loading plugin "theme-comscore"... OK

```

```

info: loading plugin "theme-default"... OK
info: found 6 pages
info: found 1 asset files
warn: "options" property is deprecated, use config.get(key) instead
warn: "options.generator" property is deprecated, use "output.name" instead
init
finish
info: >> generation finished with success in 1.3s |

Starting server ...
Serving book on http://localhost:4000
Restart after change in file src/good_tool_soft/README.md

```

然后即可在浏览器打开：

<http://localhost:4000>

去看自己的 book 的内容了：

The screenshot shows a web browser window displaying a page from a GitBook. The title of the page is '参考资料' (References). The content lists several tips and tricks, such as using Notepad++ for regex replacement, synchronizing Evernote notes with WordPress, and automating tasks with Makefiles. The sidebar on the left contains a table of contents for the book, including sections like '前言', '好用的软件和工具', '将手动变自动', '自己写代码和工具', and '附录' (with a '参考资料' section under it). At the bottom of the page, there is a footer with copyright information and a '赞赏' (Tip) button.

## 把 gitbook 源码上传到远程仓库中

接着就是把 gitbook 的源码去上传到远程的 github 的 git 仓库中。

如果你对 git 操作很熟悉，可以忽略下面内容而自行操作。

## git创建本地仓库

```

→ Import cd improve_work_efficiency
→ improve_work_efficiency ll

```

```
total 16
-rw-r--r-- 1 crifan staff  16B 1 16 14:03 README.md
-rw-r--r-- 1 crifan staff  40B 1 16 14:03 SUMMARY.md
→ improve_work_efficiency git init
Initialized empty Git repository in /Users/crifan/GitBook/Library/Import/improve_work_efficiency/.git/
→ improve_work_efficiency git:(master) ✘ git status
On branch master

No commits yet

Untracked files:
(use "git add <file>..." to include in what will be committed)

 README.md
 SUMMARY.md

nothing added to commit but untracked files present (use "git add" to track)
```

## 用 git add 把文件加入到本地

```
→ improve_work_efficiency git:(master) ✘ git status
On branch master

No commits yet

Untracked files:
(use "git add <file>..." to include in what will be committed)

 .gitignore
 Makefile
 README.md
 book.json
 src/

nothing added to commit but untracked files present (use "git add" to track)
→ improve_work_efficiency git:(master) ✘ git add *
The following paths are ignored by one of your .gitignore files:
debug
node_modules
Use -f if you really want to add them.
→ improve_work_efficiency git:(master) ✘ git status
On branch master

No commits yet

Changes to be committed:
(use "git rm --cached <file>..." to unstage)

 new file: Makefile
 new file: README.md
 new file: book.json
```

```

new file: src/README.md
new file: src/SUMMARY.md
new file: src/appendix/README.md
new file: src/appendix/reference.md
new file: src/assets/favicon.ico
new file: src/good_tool_soft/README.md
new file: src/manual_to_automatic/README.md
new file: src/self_write_soft/README.md

Untracked files:
  (use "git add <file>..." to include in what will be committed)

  .gitignore

→ improve_work_efficiency git:(master) ✘ git add .gitignore
→ improve_work_efficiency git:(master) ✘ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

  new file: .gitignore
  new file: Makefile
  new file: README.md
  new file: book.json
  new file: src/README.md
  new file: src/SUMMARY.md
  new file: src/appendix/README.md
  new file: src/appendix/reference.md
  new file: src/assets/favicon.ico
  new file: src/good_tool_soft/README.md
  new file: src/manual_to_automatic/README.md
  new file: src/self_write_soft/README.md

```

## 用 git commit 提交文件到本地仓库

```

→ improve_work_efficiency git:(master) ✘ git commit -m "1. complete basic content"
[master (root-commit) e09122d] 1. complete basic content
12 files changed, 1017 insertions(+)
create mode 100644 .gitignore
create mode 100644 Makefile
create mode 100644 README.md
create mode 100644 book.json
create mode 100644 src/README.md
create mode 100644 src/SUMMARY.md
create mode 100644 src/appendix/README.md
create mode 100644 src/appendix/reference.md
create mode 100644 src/assets/favicon.ico
create mode 100644 src/good_tool_soft/README.md
create mode 100644 src/manual_to_automatic/README.md

```

```
create mode 100644 src/self_write_soft/README.md
```

## 用 git remote add 后 git push 去推送内容到远端git仓库

```
→ improve_work_efficiency git:(master) git push
fatal: No configured push destination.
Either specify the URL from the command-line or configure a remote repository using

git remote add name <url>

and then push using the remote name

git push <name>

→ improve_work_efficiency git:(master) git remote add origin https://github.com/crifan/improve_work_efficiency.git
→ improve_work_efficiency git:(master) git push
fatal: The current branch master has no upstream branch.
To push the current branch and set the remote as upstream, use

git push --set-upstream origin master

→ improve_work_efficiency git:(master) git push --set-upstream origin master
Counting objects: 19, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (14/14), done.
Writing objects: 100% (19/19), 10.48 KiB | 2.10 MiB/s, done.
Total 19 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/crifan/improve_work_efficiency.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
```

然后远端git仓库github上就可以看到代码了：

crifan/improve\_work\_efficiency: 总结一下如何把手动工作换成自动以提高工作效率

[https://github.com/crifan/improve\\_work\\_efficiency](https://github.com/crifan/improve_work_efficiency)

## 之后每次更新代码重新上传最新代码

每次编辑更新 gitbook 的 markdown 源码后，再去：

```
git add *
git commit -m "describe your update"
git push
```

即可上传最新改动。

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件  
修订时间：2018-07-02 16:45:13

## Gitook插件

gitbook有很多插件

去官网

<https://plugins.gitbook.com>

中可以搜到自己想要的插件

但是有个缺点，很多插件页面中都没有截图介绍，不够直观

所以往往都是网上看看别人评价如何，最后决定采用哪个插件

我自己折腾期间，主要是参考了几个别人整理的gitbook的知识：

TODO:

把

【记录】给Gitbook添加更多配置和功能

[http://www.crifan.com/gitbook\\_add\\_more\\_config\\_and\\_function](http://www.crifan.com/gitbook_add_more_config_and_function)

中的：

参考链接

安装了哪些插件

等等，都整理过来

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件

修订时间： 2018-07-02 16:50:30

# 代码高亮

用的是prism

Prism

<http://prismjs.com>

对于支持的语言,

TODO:

Prism

<http://prismjs.com>

中的列表整理过来

插件配置:

```
"plugins": [  
    "-highlight",  
    "prism",  
    "prism-themes",  
],
```

注: 写markdown时, 可以拷贝:

prismjs 代码高亮 · gitbook-guide

<https://zq99299.gitbooks.io/gitbook-guide/content/chapter/plugin/prismjs.html>

中的内容, 然后用 Sublime / VSCode 中去正则替换, 变成自己要的 md 的格式

Markup - markup -»

```
`Markup` : `markup`
```

另外要注意, 要写后面的小写 (简写)

比如对于Makefile, 要写makefile

-» 如果写Makefile, 则无法识别

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件

修订时间: 2018-07-02 18:08:02

# 按钮

插件:

- `github-buttons`
- `toolbar-button`

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件  
修订时间: 2018-07-02 18:06:53

# 分享

插件:

```
"-sharing",  
"sharing-plus",
```

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件  
修订时间: 2018-07-02 18:03:59

## 脚注

插件: tbfed-pagefooter

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件

修订时间: 2018-07-02 18:04:25

## 提示和标注

【记录】gitbook换用更好看的hint callout提示

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件

修订时间: 2018-07-02 11:40:43

## 章节折叠

插件: expandable-chapters-small

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件

修订时间: 2018-07-02 18:04:55

# Google Analysis统计

插件: `ga`

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件

修订时间: 2018-07-02 18:05:15

## 打赏

插件: [donate](#)

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件

修订时间: 2018-07-02 18:05:30

## 其他插件

插件:

- sitemap-general
- copy-code-button
- anchors
- disqus

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件

修订时间: 2018-07-02 18:08:23

# 配置Gitbook

Gitbook中的配置内容，主要就是一个 `book.json`。

此外和配置算有关的，还有一个文件夹结构，即相关的目录、内容、图片等组织结构。

## book.json配置

最开始弄了个基本的book.json：

【整理】 Gitbook中的book.json参数配置

最后基本定型的book.json如下：

```
{
  "title": "编程习惯和代码风格",
  "description": "介绍为何要和如何形成良好的编程习惯和代码风格",
  "author": "Crifan Li <admin@crifan.com>",
  "language": "zh-hans",
  "gitbook": "3.2.3",
  "root": "./src",
  "links": {
    "sidebar": {
      "主页": "http://www.crifan.com"
    }
  },
  "plugins": [
    "theme-comscore",
    "-lunr",
    "-search",
    "search-plus",
    "disqus",
    "-highlight",
    "prism",
    "prism-themes",
    "github-buttons",
    "splitter",
    "-sharing",
    "sharing-plus",
    "tbfed-pagefooter",
    "expandable-chapters-small",
    "ga",
    "donate",
    "sitemap-general",
    "copy-code-button",
    "alerts",
    "toolbar-button"
  ],
  "pluginsConfig": {
    "theme-default": {
      "color": "#333399"
    }
  }
}
```

```
        "showLevel": true
    },
    "disqus": {
        "shortName": "crifan"
    },
    "prism": {
        "css": [
            "prism-themes/themes/prism-atom-dark.css"
        ]
    },
    "github-buttons": {
        "buttons": [
            {
                "user": "crifan",
                "repo": "program_code_style",
                "type": "star",
                "count": true,
                "size": "small"
            },
            {
                "user": "crifan",
                "type": "follow",
                "width": "120",
                "count": false,
                "size": "small"
            }
        ]
    },
    "sharing": {
        "douban": false,
        "facebook": true,
        "google": false,
        "hatenaBookmark": false,
        "instapaper": false,
        "line": false,
        "linkedin": false,
        "messenger": false,
        "pocket": false,
        "qq": true,
        "qzone": false,
        "stumbleupon": false,
        "twitter": true,
        "viber": false,
        "vk": false,
        "weibo": true,
        "whatsapp": false,
        "all": [
            "douban",
            "facebook",
            "google",
            "instapaper",
            "line",
            "linkedin",
            "messenger",
            "pocket",
            "qq",
            "qzone",
            "stumbleupon",
            "viber",
            "vk",
            "weibo",
            "whatsapp"
        ]
    }
}
```

```
        "pocket",
        "qq",
        "qzone",
        "stumbleupon",
        "twitter",
        "viber",
        "vk",
        "weibo",
        "whatsapp"
    ],
},
"tbfd-pagefooter": {
    "copyright": "crifan.com, 使用
```

## 文件结构

之前自己折腾的过程：

[【记录】gitbook优化book的文件组织结构](#)

现在分享出自己的，供参考：

TODO：把自己的gitbook的

各个文件夹和文件的结构整理出来

且贴出来：

SUMMARY.md

README.md

等

相关md的基本内容，供参考。

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件

修订时间： 2018-07-02 17:18:53

# 多种输出格式

Gitbook 支持将电子书导出为多种格式：

- html
- pdf
- ePub
- Mobi

下面就来详细介绍一下如何操作。

## 导出为 html

## 导出为 pdf

[【记录】](#) Gitbook的本地预览和导出为html和PDF等其他格式

## 导出为 ePub / Mobi

可以安装Calibre去看看导出后的效果：

[【记录】](#) 下载安装使用电子书阅读器转换器Calibre

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件  
修订时间： 2018-07-02 17:14:44

## 一键发布

后来为了便于管理手上如此多（几十个）的 gitbook 的电子书，所以去优化结构，利用 `Makefile` 去实现生成 `gitbook`、代码上传、上传/发布生成的文件等操作，实现了一键发布（到自己的网站和 github 的 pages 上）的效果。

详见：

[【已解决】搭建Gitbook环境使得可以发布book到crifan.com上](#)

[【记录】Mac本地搭建Gitbook环境](#)

[【记录】把http\\_summary的gitbook上传到二级域名book.crifan.com中](#)

TODO：贴出来对应的 `Makefile` 的内容，供参考

crifan.com，使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved，powered by Gitbook 该文件

修订时间： 2018-07-02 17:21:52

# Gitbook Editor

【Gitbook Editor：用Gitbook Editor创建book+同步到远端git仓库】

TODO：

把之前帖子中，如何用gitbook editor创建

并同步到远端git仓库的截图贴过来

[【整理】Gitbook和Gitbook Editor使用心得](#)

[【记录】用Gitbook Editor中的Publish去发布gitbook源码](#)

[【已解决】gitbook中用命令行去创建book](#)

[【记录】试用Gitbook Editor去编辑Markdown的文章](#)

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件

修订时间： 2018-07-02 17:23:23

## Gitbook的注意事项和使用心得

折腾gitbook期间，也算踩过一些坑，总结了些经验，整理如下：

### 最新版本 Gitbook 的bug很多，以及没有命令行工具了

详见：

[【记录】gitbook升级到新版本](#)

### 多章节的编号显示有问题

虽然开始好像是用插件支持了：

[【已解决】gitbook中安装multipart插件](#)

[【已解决】Gitbook中添加章节的编号](#)

但是最终还是没法正常的实现自己要的 章节按照1, 2, 3去编号：

[【部分解决】Gitbook中添加多章节且希望一级章节也显示编号](#)

后来又去试了试：

[【未解决】Gitbook中实现多章节中一级章节目录显示一级编号](#)

虽然找到了别人为了解决次bug而去改动的gitbook的源码的位置，但是由于无法找到

此处Mac中用 `npm install gitbook` 后，应该去哪里找到对应的位置，所以此处还是无法解决问题。

### 调试时添加 `--log` 以便于看到详细的错误信息

在用 `serve` 去调试本地的 `gitbook` 时，有些细节和错误，比如 `markdown` 的插件中语法写错了，是不知道，也看不到错误信息的。

而想要看到，则需要添加 `--log` 的参数，可以输出详细的log日志信息，从而可以看出原因和细节。

比如，[语法高亮插件](#)中用的是 `prism`，而 `markdown` 源码写了个 `md` 的语法：

```
```md
`Markup` : `markup`
```
```

时，结果log中即可看到错误信息：

```
gitbook serve /Users/crifan/GitBook/Library/Import/ebook_system_gitbook /Users/crifan/GitBook/Library/Import/ebook_system_gitbook/debug --log debug
...
```

```
debug: index page gitbook_plugins/code_highlight.md
Failed to load prism syntax: md
  Error: Cannot find module 'prismjs/components/prism-md.js'
    at Function.Module._resolveFilename (internal/modules/cjs/loader.js:594:15)
    at Function.Module._load (internal/modules/cjs/loader.js:520:25)
    at Module.require (internal/modules/cjs/loader.js:650:17)
    at require (internal/modules/cjs/helpers.js:20:18)
    at requireSyntax (/Users/crifan/GitBook/Library/Import/ebook_system_gitbook/node_modules/gitbook-plugin-prism/index.js:31:3)
    at Object.code (/Users/crifan/GitBook/Library/Import/ebook_system_gitbook/node_modules/gitbook-plugin-prism/index.js:103:11)
    at Record.TemplateBlock.applyBlock (/Users/crifan/.gitbook/versions/3.2.3/lib/models/templateBlock.js:205:23)
    at /Users/crifan/.gitbook/versions/3.2.3/lib/output/getModifiers.js:56:33
    at /Users/crifan/.gitbook/versions/3.2.3/lib/output/modifiers/highlightCode.js:47:24
    at /Users/crifan/.gitbook/versions/3.2.3/lib/output/modifiers/editHTMLElement.js:11:16
  code: 'MODULE_NOT_FOUND' }
```

如此，才真正知道自己写错了，然后才能想到，去参考[prism官网](#)，找到并修改为正确写法是：`markdown`：

```
```markdown
`Markup` : `markup`
``
```

即可解决错误，代码能真正的语法高亮：

中的内容，然后用 Sublime / VSCode 中去正则替换，变成自己要的 `md` 的格式

Markup - markup -»

`Markup` : `markup`

Copy

< 另外要注意，要写后面的小写（简写）

比如对于Makefile，要写makefile

-» 如果写Makefile，则无法识别

## 代码中包含花括号

【已解决】Gitbook中Markdown的代码中如何包含左右花括号{}

## 代码块显示代码行号

不过gitbook中的prism插件暂时还没有直接支持行号显示：

【未解决】Gitbook中Markdown的代码段中如何添加行号

## 跨页面引用和跳转

跨页面的话，貌似只能直接用链接地址了：

【已解决】gitbook如何实现book内部的跨页面的跳转或引用

后记：

已经实现了：

【已解决】gitbook中想要实现引用别的章节的内容

## 引用gitbook的root目录之外是不支持的

【未解决】gitbook中用markdown的include实现根目录外部的README.md文件内容同步

不过倒是可以通过：

```
{% include "../README.md" %}
```

去引用gitbook内部的其他的章节的。

## 左侧添加下载PDF按钮

没有完全实现，间接实现了类似效果：

【部分解决】给gitbook中左侧添加PDF及其他格式的下载按钮

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件  
修订时间： 2018-07-02 18:02:39

## 附录

下面列出相关参考资料。

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件  
修订时间: 2018-07-01 10:37:46

## 参考资料

- 【记录】给Gitbook添加更多配置和功能
- 【整理】文档制作工具：MkDocs Gitbook等
- 命令 · gitbook-guide
- 【记录】给Gitbook添加更多配置和功能
- prismjs 代码高亮 · gitbook-guide
- 【记录】gitbook换用更好看的hint callout提示
- 【已解决】gitbook中book.json中能否把公共部分提取出来
- 【记录】gitbook升级到新版本
- 【整理】Gitbook中的book.json参数配置
- 【记录】gitbook优化book的文件组织结构
- 【记录】Gitbook的本地预览和导出为html和PDF等其他格式
- 【记录】下载安装使用电子书阅读器转换器Calibre
- 【已解决】搭建Gitbook环境使得可以发布book到crifan.com上
- 【记录】Mac本地搭建Gitbook环境
- 【记录】把http\_summary的gitbook上传到二级域名book.crifan.com中
- 【整理】Gitbook和Gitbook Editor使用心得
- 【记录】用Gitbook Editor中的Publish去发布gitbook源码
- 【已解决】gitbook中用命令行去创建book
- 【记录】试用Gitbook Editor去编辑Markdown的文章
- 【记录】gitbook升级到新版本
- 【已解决】gitbook中安装multipart插件
- 【已解决】Gitbook中添加章节的编号
- 【部分解决】Gitbook中添加多章节且希望一级章节也显示编号
- 【未解决】Gitbook中实现多章节中一级章节目录显示一级编号
- 【未解决】Gitbook中Markdown的代码段中如何添加行号
- 【已解决】gitbook如何实现book内部的跨页面的跳转或引用
- 【已解决】gitbook中想要实现引用别的章节的内容
- 【未解决】gitbook中用markdown的include实现根目录外部的README.md文件内容同步
- 【部分解决】给gitbook中左侧添加PDF及其他格式的下载按钮

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件

修订时间： 2018-07-02 18:12:44