

目录

前言	1.1
第一章	1.2
第二章	1.3
第二章第一节	1.3.1
第二章第二节	1.3.2
演示各种功能和效果	1.4
当前Gitbook源码目录结构	1.4.1
提示hint和插图编号callout	1.4.2
代码高亮显示	1.4.3
附录	1.5
参考资料	1.5.1

Gitbook演示

简介

crifan 的 Gitbook 模版，用于演示如何创建一个自己的 Gitbook

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/gitbook_demo: Gitbook演示](#)

在线浏览

- [Gitbook演示 book.crifan.com](#)
- [Gitbook演示 crifan.github.io](#)

离线下载阅读

- [Gitbook演示 PDF](#)
- [Gitbook演示 ePub](#)
- [Gitbook演示 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： 2018-06-15 21:59:18

第一章标题

普通的无序列表：

- 无序列表元素a
- 无序列表元素b
- 无序列表元素c

有序列表：

1. 有序列表编号1
2. 有序列表编号2
3. 有序列表编号3

第一章第一节

第一章第一节的内容

第一章第二节

第一章第二节的内容

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： 2018-06-15 21:23:42

第二章标题

下面用于演示效果，会把子章节放到单独的 `md` 文件中。

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： 2018-06-15 20:53:16

第二章第一节

第二章第一节的内容

演示引用 `img` 文件夹中的图片的效果：

The screenshot shows a GitHub user profile for 'crifan'. The profile picture is a stylized orange 'U' shape composed of several squares. Below the picture, the user's name 'Crifan Li' and handle 'crifan' are displayed, along with a link to their website: <http://www.crifan.com/about/me/>. There is also an 'Edit bio' button. On the right side of the profile, there are sections for 'Popular repositories' and 'Customize your pinned repositories'. Under 'Popular repositories', there are four listed: 'crifanLib', 'scientific_network_summary', 'http_restful_api', and 'BlogsToWordpress'. Under 'Customize your pinned repositories', there are two listed: 'program_common_logic' and 'program_code_style'. At the bottom of the profile, a timeline shows '604 contributions in the last year' with a chart from June to May.

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： 2018-06-15 20:56:00

第二章第二节

第二章第二节的内容

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: 2018-06-15 20:56:36

演示各种功能和效果

下面各个章节用来演示 gitbook 中的各种功能和效果。

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: 2018-06-15 21:12:22

当前 Gitbook 源码目录结构

```
├── Makefile
├── README.md
├── book.json
├── book_current.json
├── debug/ # 用于保存debug时的文件
├── node_modules/ # 用于保存Gitbook安装后的各种plugin
└── src/
    ├── README.md
    ├── SUMMARY.md
    ├── appendix/
    │   ├── README.md
    │   └── reference.md
    ├── assets/
    │   ├── favicon.ico
    │   └── img/
    │       └── crifan_github_home.png
    ├── chapter_1/
    │   └── README.md
    ├── chapter_2/
    │   ├── README.md
    │   ├── ch2_section_1.md
    │   └── ch2_section_2.md
    └── chapter_3/
        ├── README.md
        ├── ch3_code_highlight.md
        ├── ch3_hint_callout.md
        └── ch3_structure_list.md
```

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: 2018-06-15 21:57:38

提示 hint 和插图编号 callout

hint == callout == 提示/警告/错误

演示用插件 callouts 实现的 callout == hint , 即各种类型的提示/提醒的效果

语法:

```
> ##### {type}:: Your Title  
>  
> Content
```

其中 {type} 是下面中的任意一种:

- primary
- success
- danger
- warning
- info

| 如果你用过 Bootstrap 就能看出来, 其实这几种标题类型就是 Bootstrap 中的标题的类型

效果如下的显示:

提示= tip = note = info

提醒类信息的标题

提醒类信息中的 内容

成功= success =

成功的标题

成功中的 内容

警告= warning =⚠

警告的标题

警告中的 内容

错误= error = danger =

错误的标题

错误中的 内容

其他几种(不常见的)类型

Tag的标题

Tag中的 内容

Note的标题

Note中的 内容

Comment的标题

Comment中的 内容

Hint的标题

Hint中的 内容

Caution的标题

Caution中的 内容

Quote的标题

Caution中的 内容

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: 2018-06-15 23:12:18

代码高亮显示

此处 gitbook 中代码高亮的插件用的是 prism

官网是：<http://prismjs.com>

下面列出常见的几种代码的代码高亮效果。

markdown

HTTP简介

`HTTP` = `Hyper Text Transfer Protocol` = `超文本传输协议`

HTTP的总体思路是：

- * 客户端：向服务器发送一个‘请求’ = `Request`，请求头包含请求的方法、URI、协议版本、以及包含请求修饰符、客户信息和内容的类似于MIME的消息结构。
- * ‘服务器’：以一个状态行作出‘响应’ = `Response`，相应的内容包括消息协议的版本，成功或者错误编码加上包含服务器信息、实体元信息以及可能的实体内容。

简而言之就是一个你问我答的协议：

- * 客户端Client 问=‘请求’ = `Request`
- * 服务器Server 答=‘响应’ = `Response`

关于HTTP的相关知识，或许很多人不是很熟悉。

但是：

- 作为普通电脑用户的你，经常用‘浏览器’（`IE` / `Chrome` / `Firefox` 等）去浏览网页时
- 作为程序开发，用网络库去调用服务器后台提供的接口时

其实内部都用到了HTTP的技术。

python

```
def saveJsonToFile(jsonDict, fullFilename, indent=2, fileEncoding="utf-8"):
    """
        save dict json into file
        for non-ascii string, output encoded string, without \uxxxx
    """

    with codecs.open(fullFilename, 'w', encoding="utf-8") as outputFp:
        json.dump(jsonDict, outputFp, indent=indent, ensure_ascii=False)

#####
# Main Part
#####

templateJson = {}
currentJson = {}
```

```
# run python in : /Users/crifan/GitBook/Library/Import/youdao_note_summary
currPath = os.getcwd()
curBasename = os.path.basename(currPath)
CurrentGitbookName = curBasename
print("CurrentGitbookName=%s" % CurrentGitbookName) #youdao_note_summary

bookJsonTemplateFullPath = os.path.join(BookRootPath, BookJsonTemplateFilename)
# print("bookJsonTemplateFullPath=%s" % bookJsonTemplateFullPath)
# /Users/crifan/GitBook/Library/Import/book_common.json
with open(bookJsonTemplateFullPath) as templateJsonFp:
    templateJson = json.load(templateJsonFp, encoding="utf-8")
```

javascript

```
// extract single sub string from full string
// eg: extract '012345678912345' from 'www.ucows.cn/qr?id=012345678912345'
export function extractSingleStr(curStr, pattern, flags='i') {
    let extractedStr = null;

    let rePattern = new RegExp(pattern, flags);
    console.log(rePattern);

    let matches = rePattern.exec(curStr);
    console.log(matches);

    if (matches) {
        extractedStr = matches[0];
        console.log(extractedStr);
    }

    if (extractedStr === null) {
        extractedStr = "";
    }

    console.log(`curStr=${curStr}, pattern=${pattern}, flags=${flags} -> extractedStr=${extractedStr}`);
}

return extractedStr;
}
```

html

```
<!DOCTYPE HTML>
<html lang="zh-hans" >
  <head>
    <meta charset="UTF-8">
    <meta content="text/html; charset=utf-8" http-equiv="Content-Type">
    <title>HTTP相关 · HTTP知识总结</title>
```

```
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
<meta name="description" content="">
<meta name="generator" content="GitBook 3.2.3">
<meta name="author" content="Crifan Li <admin@crifan.com>">

<link rel="stylesheet" href="../gitbook/style.css">
<link rel="stylesheet" href="../gitbook/gitbook-plugin-search-plus/search.css">
```

CSS

```
a#weixin {
    position: relative;
}

#weixin img {
    visibility: hidden;
    opacity: 0;
    transform: translate(0, 10px);
    transition: all 0.3s ease-in-out;
    position: absolute;
    right: -30px;
    bottom: 40px;
    width: 150px;
    height: 150px;
}

#weixin:hover img {
    visibility: visible;
    transform: translate(0, 0px);
    opacity: 1;
}
```

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: 2018-06-15 21:21:42

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: 2018-06-15 20:45:43

参考资料

- [crifan \(Crifan Li\)](#)
- [【记录】给Gitbook添加更多配置和功能](#)
- [【已解决】提取Gitbook中Makefile公共部分](#)
- [【已解决】gitbook中book.json中能否把公共部分提取出来](#)

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: 2018-06-15 22:07:24