

Table of Contents

1. [Introduction](#)
2. [簡介](#)
3. [安裝](#)
4. [使用](#)
5. [GitBook.com](#)
 - i. [新建書籍](#)
 - ii. [編輯書籍](#)
 - iii. [配置書籍](#)
 - i. [書籍主題](#)
 - ii. [GitHub 集成](#)
 - iii. [綁定域名](#)
6. [個性化配置](#)
 - i. [book.json](#)
7. [插件](#)
 - i. [主題插件](#)
 - i. [comscore](#)
 - ii. [實用插件](#)
 - i. [disqus](#)
 - ii. [multipart](#)
 - iii. [toggle-chapters](#)
 - iv. [codeblock-filename](#)
8. [發佈到 GitHub Pages](#)

GitBook 簡明教程

本教程主要圍繞 GitBook 的安裝，使用，集成，書籍發佈，個性化以及實用插件幾個方面。

其中借鑑了一些官方的文檔以及在互聯網上找到的一些內容，主要結合個人經驗寫成。由於經驗有限，難免存在諸多疏漏或者理解不到位的地方，還請斧正！

本書發佈在 <http://www.chengweiyang.cn/gitbook>。

簡介

Modern book format and toolchain using Git and Markdown

這是 [gitbook](#) 項目主頁上對 gitbook 的定義。

gitbook 首先是一個軟件，正如上面定義的那樣，它使用 Git 和 Markdown 來編排書本，如果用戶沒有聽過 Git 和 Markdown，那麼 gitbook 可能不適合你！

本書也是使用 gitbook 生成，所以在看到這裡的時候，你應該對 gitbook 的魔力有了初步印象！

安裝

[gitbook](#) 的安裝非常簡單，詳細指南可以參考 [gitbook 文檔](#)。

這裡的安裝只需要一步就能完成！

```
$ npm install gitbook -g
```

需要注意的是：用戶首先需要安裝 `nodejs`，以便能夠使用 `npm` 來安裝 `gitbook`。

基本使用

gitbook 的基本用法非常簡單，基本上就只有兩步：

1. 使用 `gitbook init` 初始化書籍目錄
2. 使用 `gitbook serve` 編譯書籍

下面將結合一個非常簡單的實例，來介紹 gitbook 的基本用法。

gitbook init

首先，創建如下目錄結構：

```
$ tree book/
book/
├── README.md
└── SUMMARY.md

0 directories, 2 files
```

README.md 和 SUMMARY.md 是兩個必須文件，README.md 是對書籍的簡單介紹：

```
$ cat book/README.md
# README

This is a book powered by [GitBook](https://github.com/GitbookIO/gitbook).
```

SUMMARY.md 是書籍的目錄結構。內容如下：

```
$ cat book/SUMMARY.md
# SUMMARY

* [Chapter1](chapter1/README.md)
  * [Section1.1](chapter1/section1.1.md)
  * [Section1.2](chapter1/section1.2.md)
* [Chapter2](chapter2/README.md)
```

創建了這兩個文件後，使用 `gitbook init`，它會為我們創建 SUMMARY.md 中的目錄結構。

```
$ cd book
$ gitbook init
$ tree
.
├── README.md
├── SUMMARY.md
├── chapter1
│   ├── README.md
│   └── section1.1.md
```

```
|   └─ section1.2.md
|   └─ chapter2
|       └─ README.md
```

2 directories, 6 files

注意：在我的實驗中，`gitbook init` 只支持兩級目錄！

gitbook serve

書籍目錄結構創建完成以後，就可以使用 `gitbook serve` 來編譯和預覽書籍了：

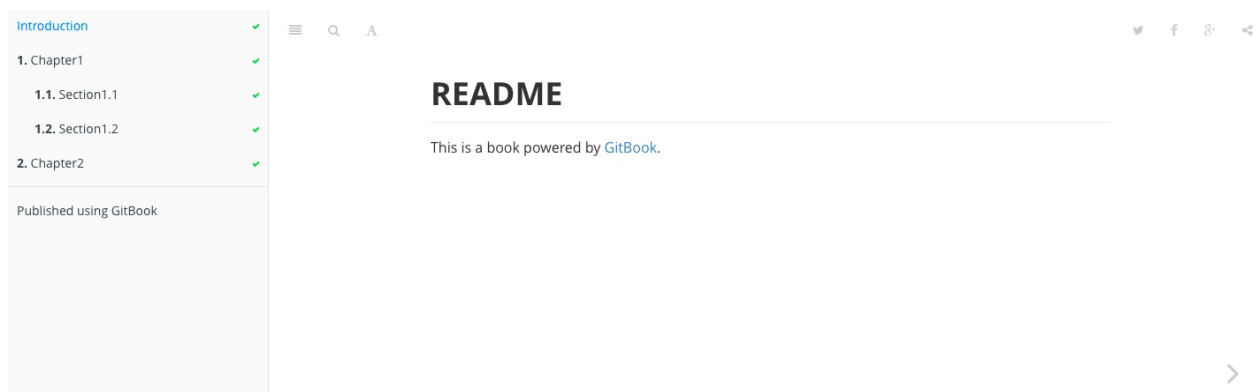
```
$ gitbook serve
Press CTRL+C to quit ...

Live reload server started on port: 35729
Starting build ...
Successfully built!

Starting server ...
Serving book on http://localhost:4000
```

`gitbook serve` 命令實際上會首先調用 `gitbook build` 編譯書籍，完成以後會打開一個 web 服務器，監聽在本地的 4000 端口。

現在，可以用瀏覽器打開 <http://127.0.0.1:4000> 查看書籍的效果，如下圖：



現在，gitbook 為我們創建了書籍目錄結構後，就可以向其中添加真正的內容了，文件的編寫使用 markdown 語法，在文件修改過程中，每一次保存文件，`gitbook serve` 都會自動重新編譯，所以可以持續通過瀏覽器來查看最新的書籍效果！

另外，用戶還可以下載 [gitbook 編輯器](#)，做到所見即所得的編輯，如下圖所示：



gitbook editor 的使用非常簡單，這裡不再介紹！

GitBook.com

GitBook.com 是一個圍繞 gitbook 發行書籍的社區，於 2014 年初創，GitBook.com 提供免費和付費的服務，而且免費賬戶就可以享受諸多服務，包括：

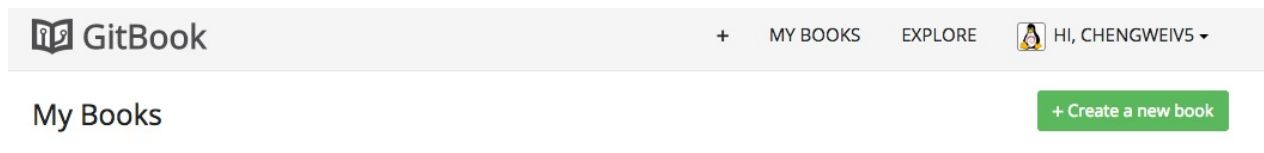
- 1 本私有書籍
- 託管不限數量的公開書籍
- 售賣不限數量的書籍，並分享 80% 的書籍收入
- 不限數量的協作者
- 免費的在線書籍編輯器

對於普通用戶來說，免費賬號就已經夠用，因為雖然限制為 1 本私有書籍，但是並沒有限制書籍的大小，所以對於個人的學習筆記來說非常合適，用戶甚至可以將所有知識歸類放入一本私有書籍中！當然，GitBook.com 限制私有書籍數量但不限公開書籍數量的政策，顯然是鼓勵用戶能夠共享知識！

新建書籍

要使用 GitBook.com 來託管你的書籍，首先需要註冊一個賬號，免費註冊後，用戶也可以選擇升級為付費用戶，享受更多的服務！

登陸 [GitBook.com](https://gitbook.com) 後，在用戶頁面，可以管理現有書籍以及創建新的書籍，如下圖：



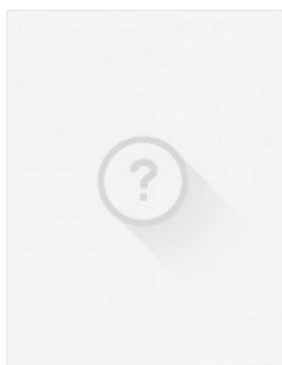
點擊 "+ Create a new book" 後，跳轉到新建書籍頁面，如下圖：

A screenshot of the 'Create book' form in GitBook. At the top, there are four theme options: 'Basic' (selected, with a book icon), 'Science' (with a microscope icon), 'Exercises' (with a code icon), and 'Quizzes' (with a list icon). Below these is a dropdown menu showing 'chengweiv5' and a text input field labeled 'Enter a Title'. Underneath is a URL field showing 'http://chengweiv5.gitbooks.io/' and a label 'Name'. A link 'Show more options' is below the URL field. At the bottom is a large green button labeled 'Create book'.

目前有 4 中書籍主題可以選擇，這裡以默認的 `Basic` 主題為例，輸入書籍名字後，點擊 "Create book"，完成書籍的創建。

創建完成後，就會進入書籍屬性頁面，如下圖所示：

💡 **Quick setup** — Open this book in the web editor and click 'save' to generate a first version of your book. Or [learn more](#) about using the git command line.



No content published yet

0

views yesterday



View Book



Edit Book



這裡可以進行對書籍的各個屬性進行配置，例如：

- 編輯書籍（Edit Book）
- 書籍主題（Theme）
- 綁定 GitHub（GitHub）
- 綁定域名（Domain Names）

等。

以上幾個配置將在後面介紹，其它的配置比較簡單，請讀者自行探索！

編輯書籍

在創建了書籍後，可以使用免費的在線編輯器進行編輯，也可以使用 [gitbook editor](#) 編輯，甚至使用任何喜歡的文本編輯器來編輯，例如：Vim。

在線編輯

進入到書籍的屬性頁面後，點擊 "Edit Book" 按鈕即可打開在線編輯器。

GitBook 的在線編輯器對於國內用戶來說，很可能不能訪問，所以最好還是下載 [gitbook editor](#) 到本地，安裝後使用，或者使用自己喜歡的文本編輯器直接編輯。

gitbook editor

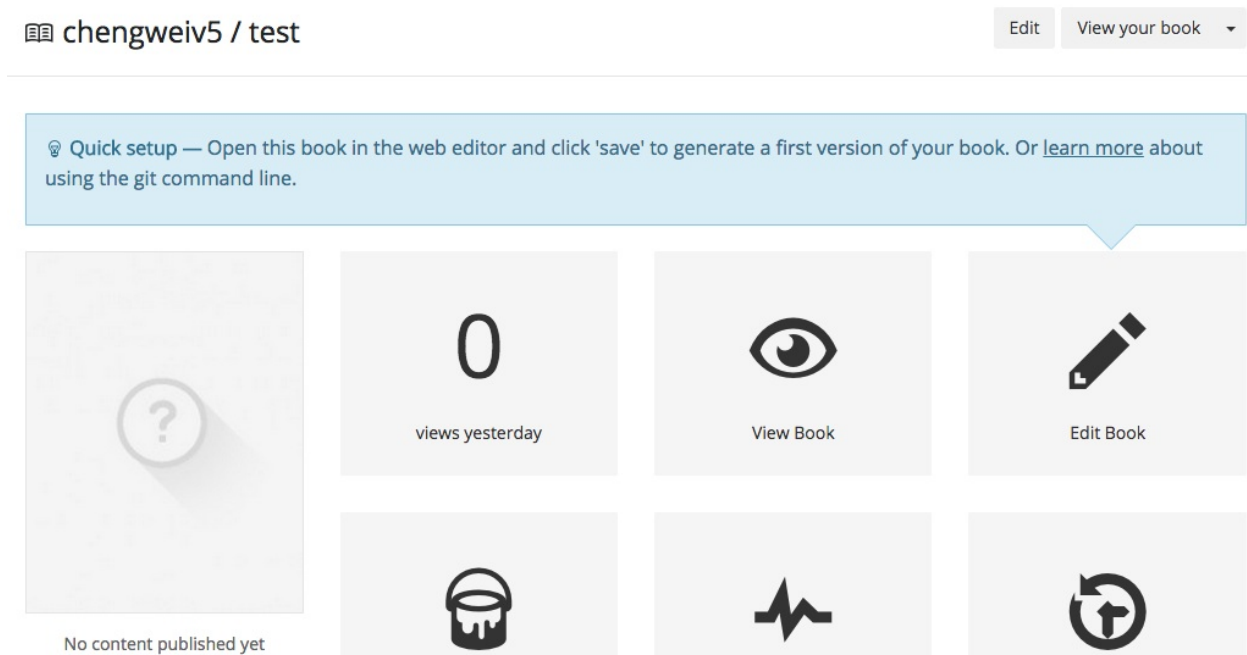
gitbook editor 實際上就是一個本地應用版的在線編輯器，使用方式和在線編輯器類似，所見即所得，這裡不再介紹，讀者可以參考 [gitbook 使用](#) 中的內容。

Git & Markdown

另一種方式，是直接使用文本編輯器，編寫 Markdown 文檔，然後，使用 Git 提交到書籍的遠程項目，當然，提交前，最好在本地使用 `gitbook` 預覽效果；提交後，GitBook.com 會自動生成更新書籍的內容。

克隆書籍源代碼

GitBook.com 上的每本書都使用 Git 項目來管理，所以，這裡首先需要克隆需要編輯書籍的 Git 項目，登陸 GitBook.com 後，跳轉到書籍的屬性頁面，如下圖所示：



點擊 "Edit Book" 上方的 "learn more"，將會展現此書籍的 Git 項目地址，以及簡單的使用方法，如下

圖所示：

chengweiv5 / test

Edit

View your book

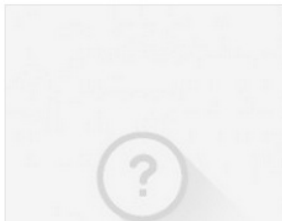
💡 Quick setup — Open this book in the web editor and click 'save' to generate a first version of your book. Or [learn more](#) about using the git command line.

Create a new repository on the command line

```
touch README.md SUMMARY.md
git init
git add README.md SUMMARY.md
git commit -m "first commit"
git remote add gitbook https://git.gitbook.com/chengweiv5/test.git
git push -u gitbook master
```

...or push an existing repository from the command line

```
git remote add gitbook https://git.gitbook.com/chengweiv5/test.git
git push -u gitbook master
```



0

views yesterday



View Book



Edit Book

使用如下命令，克隆書籍的源代碼：

```
$ git clone https://git.gitbook.com/chengweiv5/test.git
Cloning into 'test'...
remote: Counting objects: 28, done.
remote: Compressing objects: 100% (17/17), done.
remote: Total 28 (delta 6), reused 28 (delta 6)
Unpacking objects: 100% (28/28), done.
Checking connectivity... done.

$ cd test/

$ ls
README.md  SUMMARY.md

$ git log --oneline
07bde6c Cleanup example
6d368db Add _book to gitignore
20779f5 Add explanation in README.md
1b5b1a6 Create chapter-1/ARTICLE1.md
77b1858 Add help message in SUMMARY.md
210e3fe Create chapter-1/README.md
5570112 Create SUMMARY.md
2a8a0c3 Initial commit
```

可以看到，創建好的書籍默認已經創建了一些內容，但是這些內容是還沒有發佈的，所以其它人不能閱讀！

編輯內容

現在，可以參考 [gitbook 使用](#) 中的內容來編輯書籍內容，使用 `gitbook init`，`gitbook serve` 來預覽，完成後，可以提交修改：

```
$ git commit -asm "init book"
```

發佈內容

最後，提交到遠程 Git 項目：

```
$ git push
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 362 bytes | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
To https://git.gitbook.com/chengweiv5/test.git
07bde6c..b6a8b3f master -> master
```

閱讀書籍

提交到 [GitBook.com](#) 後，書籍就自動發佈了，用戶就可以通過書籍的地址訪問了，例如：<http://chengweiv5.gitbooks.io/test/>



test

Chengwei Yang

Published with GitBook

test

Chengwei Yang

READ REVIEW

You can also download this book as: [ePUB](#) (for iPad, iPhone, Mac), [MOBI](#) (for Kindle) or [PDF](#) (for Windows and others).

Like 0 Tweet 0 Share 0

Email me when this book is updated:

Enter your email SUBSCRIBE

點擊 "READ" 按鈕，就可以閱讀書籍的內容了，如下圖所示：

Return to book	
Review this book	
About the author	
Introduction	✓
1. Chapter1	
1.1. Section1.1	
1.2. Section1.2	
2. Chapter2	
Published using GitBook	

GitBook Example

Welcome in your GitBook!



配置書籍

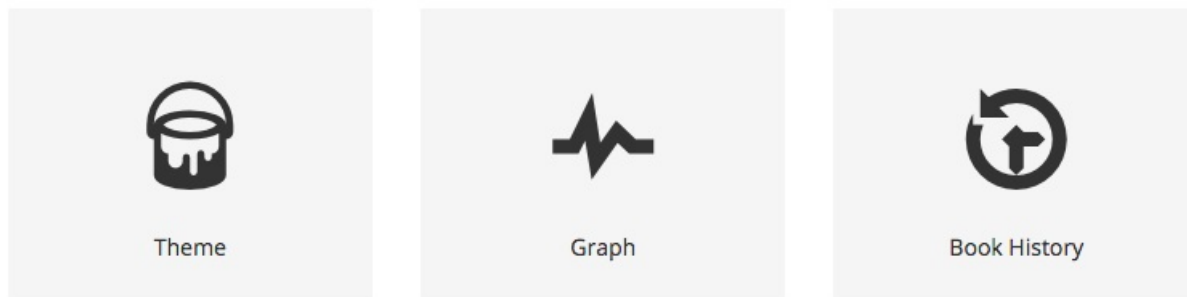
每本書籍都有一個屬性頁面，可以配置書籍的各方面屬性，接下來將介紹以下幾個方面的配置：

- 書籍主題
- GitHub 集成
- 綁定域名

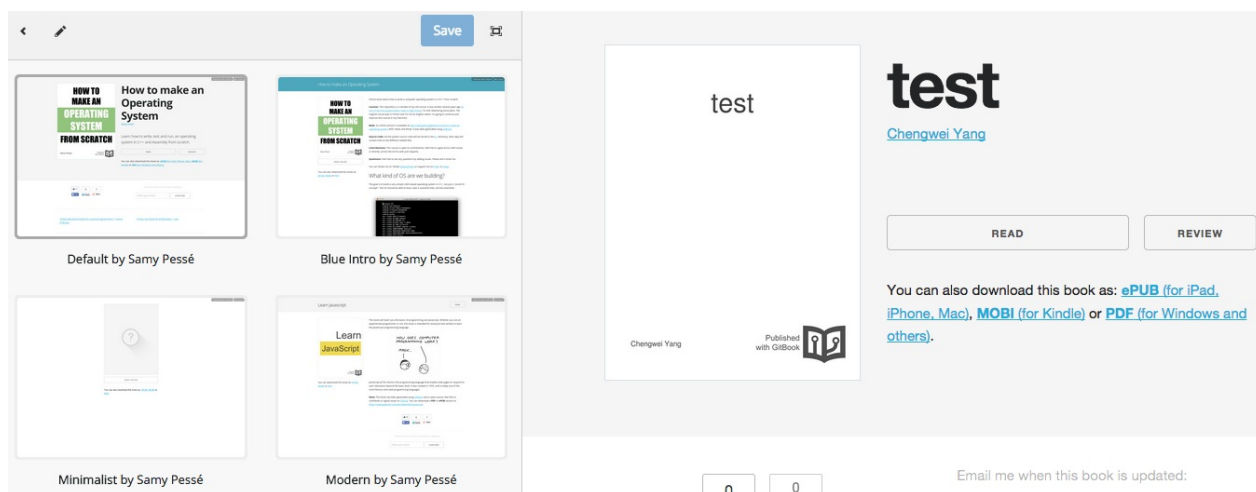
其它方面的配置比較簡單，請讀者自行摸索！

書籍主題

GitBook.com 上的書籍可以自由配置主題，首先進入書籍的屬性頁面，下拉頁面找到 "Theme" 圖標，如下圖所示：



點擊 "Theme" 後，出現主題選擇頁面，目前提供了 4 種可選主題，用戶可以選擇其中一種作為書籍的主題，如下圖所示：



最後，點擊 "Save" 按鈕，保存即可！

GitHub 集成

GitBook.com 為每本書籍都創建了一個 Git 項目，並且使用這個 Git 項目來管理書籍源碼（注意：這裡的源碼是指所有用戶提交的內容）。正如在 [編輯書籍](#) 中介紹的那樣，我們可以通過向書籍的 Git 項目提交內容來更新書籍。

另外，GitBook.com 還可以集成 [GitHub](#)，所以用戶可以將書籍的源碼通過 GitHub 上的項目來管理，這樣可以使用 GitHub 帶來的各種優點，例如：

- 其它用戶可以 fork
- 用戶可以點贊，獲得更新提醒
- 用戶可以貢獻自己的內容

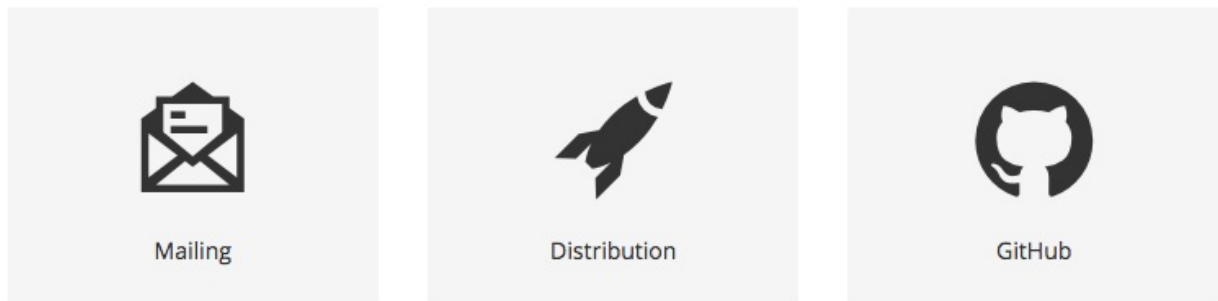
等等。

這裡，接著之前的實例，將使用 GitHub 中的項目來替代 GitBook.com 上的項目

`https://git.gitbook.com/chengweiv5/test.git`。

使用 GitHub 上的項目

首先，將書籍的 Git 項目設置為 GitHub 上的項目，進入書籍屬性頁面，找到 "GitHub" 圖標，如下圖：



輸入將要使用的 GitHub 上的項目，注意是公開的項目，如下圖：

Tools

Traffic

Distribution

History

Mailing

Book settings

Options


Collaborators 0

Domain Names

GitHub Integration

Save

GitHub Repository

 ex: Username/Repo

Link your book to a GitHub repository to edit this repository directly from the editor.

Integration

Before setting up the integration, enter a GitHub repository in the text input above.

Before connecting a webhook, you need to connect a GitHub account to your GitBook account in the [settings](#).

It will automatically updates & builds your book whenever you push to this repository (only public repos).

Add a deployment webhook

保存後，可以看到之前不可點擊的 "Add a deployment webhook" 按鈕已經可以點擊了，這個按鈕表示：每當用戶配置的 GitHub 上的項目更新時，自動更新書籍！

Tools

Traffic

Distribution

History

Mailing

Book settings

Options

Collaborators 0


Domain Names

GitHub Integration

Save

Book configuration has been updated!

GitHub Repository

 chengweiv5/test

Link your book to a GitHub repository to edit this repository directly from the editor.

Integration


Before connecting a webhook, you need to connect a GitHub account to your GitBook account in the [settings](#).

It will automatically updates & builds your book whenever you push to this repository (only public repos).

Add a deployment webhook

創建 GitHub 項目

假設上面配置的 GitHub 中的項目並不存在，那麼，首先，登陸 GitHub，然後創建一個 test 項目，如下圖所示：

Owner:  chengweiv5 / Repository name: test

Great repository names are short and memorable. Need inspiration? How about **secret-nemesis**.

Description (optional)

☒ **Public**
Anyone can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

☐ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** | Add a license: **None** ⓘ

Create repository

導入代碼

現在，將書籍原來的 Git 項目內容導入到新建的 GitHub 中的項目中，假設已經將 `https://git.gitbook.com/chengweiv5/test.git` 項目克隆到了 `test` 目錄：

```
$ cd test

$ git remote add github https://github.com/chengweiv5/test.git

$ git push -u github master
Counting objects: 31, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (26/26), done.
Writing objects: 100% (31/31), 6.58 KiB | 0 bytes/s, done.
Total 31 (delta 7), reused 0 (delta 0)
To https://github.com/chengweiv5/test.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from github.
```

`git push` 命令中的 `-u` 表示將本地 `master` 分支的上游分支設置為 `github/master`，所以以後修改了本地 `master` 分支後，`git push` 將推送到 `github` 上，而非原來的 `https://git.gitbook.com/chengweiv5/test.git`。

更新書籍

現在，可以通過提交變更到 GitHub 項目中來更新書籍了，例如：

```
$ git diff
diff --git a/README.md b/README.md
```

```
index e7b10b8..c776a52 100644
--- a/README.md
+++ b/README.md
@@ -2,3 +2,5 @@ GitBook Example
=====

Welcome in your GitBook!
+
+Hello from GitHub!

$ git commit -asm "hello from github"
[master d985a79] hello from github
1 file changed, 2 insertions(+)

$ git push
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 392 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/chengweiv5/test.git
b6a8b3f..d985a79 master -> master
```

[Return to book](#)

[Review this book](#)

[About the author](#)

[Introduction](#) ✓

1. Chapter1

1.1. Section1.1

1.2. Section1.2

2. Chapter2

Published using GitBook

☰ 🔍 A

🐦 f 8x

GitBook Example

Welcome in your GitBook!

Hello from GitHub!

>

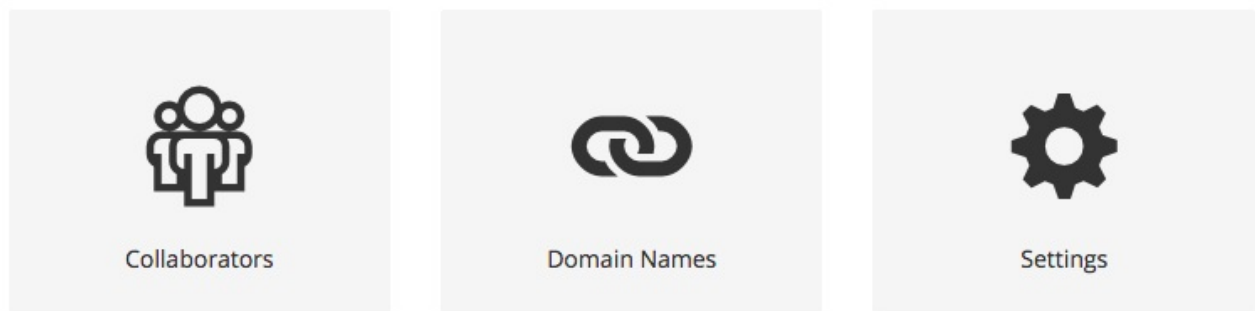
綁定域名

除了可以使用類似 `http://chengweiv5.gitbooks.io/test/content/index.html` 地址訪問用戶的書籍外，還可以為每本書綁定域名，前提是用戶有自己的域名。

這裡繼續以 test book 為例，將其綁定到 `test.chengweiyang.cn`，這樣，用戶就可以通過新的域名訪問本書！

修改書籍的域名配置

首先，在書籍的屬性頁面找到 "Domain Names" 屬性，如下圖：



點擊 "Domain Names"，進入配置頁面，如下圖：

Tools	<p>🔔 Quick tip — Add your custom domain names here then point your DNS to GitBook.</p>
Traffic	
Distribution	
History	
Mailing	
Book settings	<p>Custom Domains</p>
Options	<p>For homepage</p> <input type="text" value="www.example.com"/> <p>The homepage of your book is currently accessible at: http://chengweiv5.gitbooks.io/test/.</p>
Collaborators 0	<p>For content</p> <input type="text" value="www.example.com"/> <p>The content of your book is currently accessible at: http://chengweiv5.gitbooks.io/test/content/.</p>
Domain Names	
GitHub Integration	
<p>Save</p>	

分別為書籍的封面和內容設置一個域名，如下圖所示：

Tools

Traffic

Distribution

History

Mailing

Book settings

Options

Collaborators 0

Domain Names

GitHub Integration

Save

✔ Book configuration has been updated!

💡 Quick tip — Add your custom domain names here then [point your DNS to GitBook](#).

Custom Domains

For homepage

test.chengweiyang.cn

The homepage of your book is currently accessible at: <http://test.chengweiyang.cn/>.

For content

content.test.chengweiyang.cn

The content of your book is currently accessible at: <http://content.test.chengweiyang.cn/>.

需要注意的是：這裡只能是一個合法的域名，不能是域名下的某個地址，例如：`chengweiyang.cn/test`，`chengweiyang.cn/test/content`。

配置域名

將書籍的訪問地址修改為個人的域名後，還需要讓個人域名能夠找到書籍的內容，這裡參考 [gitbook 文檔](#) 來配置。主要的是要為你的域名添加子域名 CNAME，並且指向 `gitbook.com`，然後等待域名生效，可能需要幾分鐘到幾小時不等。

這裡不再演示怎樣為你的域名添加 CNAME 記錄，各個域名服務商提供的工具可能不同，所以用戶可以諮詢域名提供商獲取幫助。

驗證

可以使用 `nslookup` 來驗證是否解析成功：

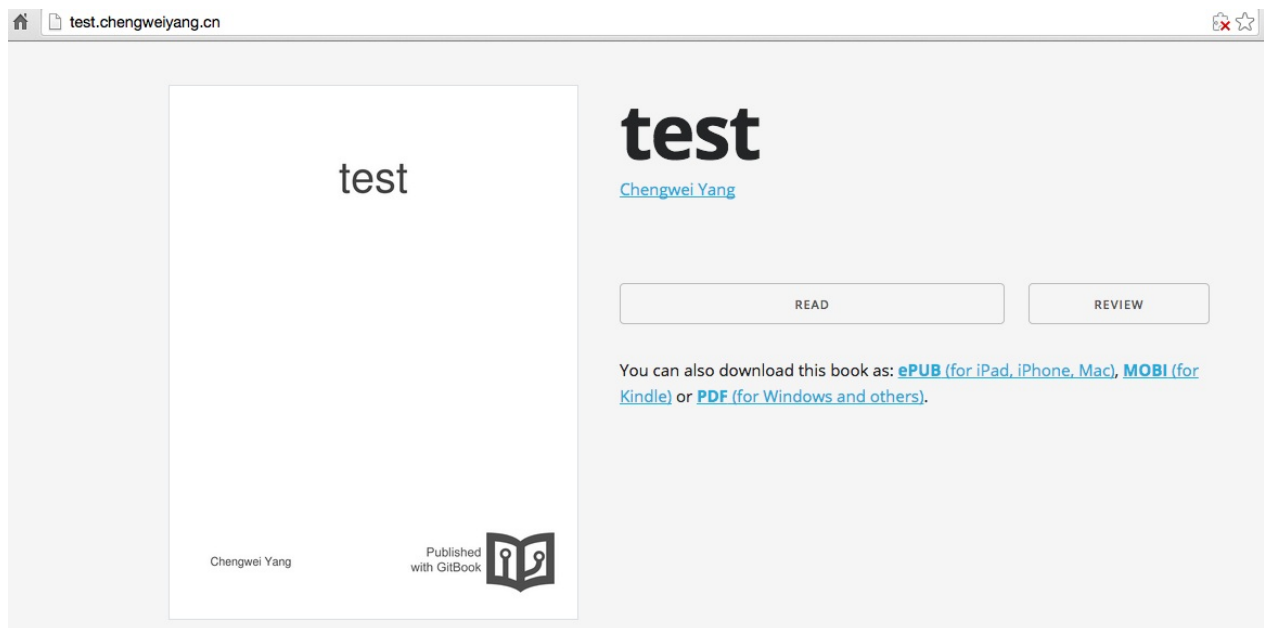
```
$ nslookup test.chengweiyang.cn
Server:      10.11.50.66
Address:     10.11.50.66#53

Non-authoritative answer:
test.chengweiyang.cn canonical name = www.gitbook.com.
www.gitbook.com canonical name = saga-4658.herokussl.com.
saga-4658.herokussl.com canonical name = elb049689-1604000308.us-east-1.elb.amazonaws.com
Name:   elb049689-1604000308.us-east-1.elb.amazonaws.com
Address: 23.21.177.67
Name:   elb049689-1604000308.us-east-1.elb.amazonaws.com
Address: 23.21.212.67
Name:   elb049689-1604000308.us-east-1.elb.amazonaws.com
Address: 23.21.54.142

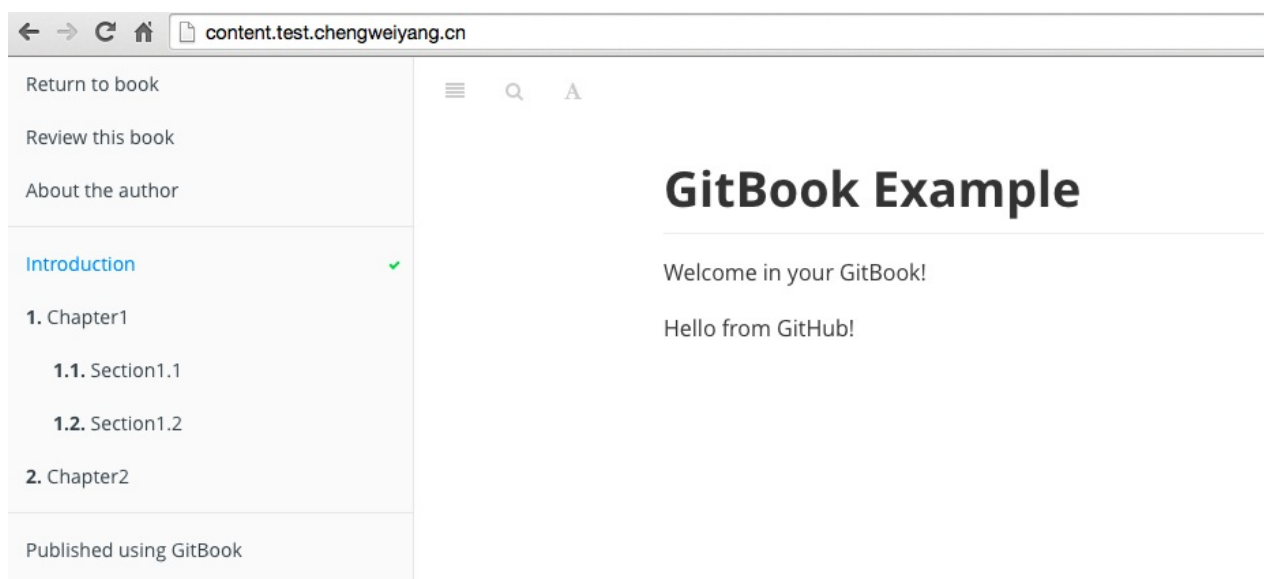
$ nslookup content.test.chengweiyang.cn
Server:      10.11.50.65
Address:     10.11.50.65#53
```

```
Non-authoritative answer:
content.test.chengweiyang.cn    canonical name = www.gitbook.com.
www.gitbook.com canonical name = saga-4658.herokuapp.com.
saga-4658.herokuapp.com canonical name = elb049689-1604000308.us-east-1.elb.amazonaws.com
Name:   elb049689-1604000308.us-east-1.elb.amazonaws.com
Address: 23.21.212.67
Name:   elb049689-1604000308.us-east-1.elb.amazonaws.com
Address: 23.21.54.142
Name:   elb049689-1604000308.us-east-1.elb.amazonaws.com
Address: 23.21.177.67
```

現在，將瀏覽器指向 `http://test.chengweiyang.cn` 就可以看到書籍了。



點擊 "READ" 按鈕，或者直接將瀏覽器指向 `http://content.test.chengweiyang.cn` 就可以閱讀書籍內容了！



個性化配置

除了修改書籍的主題外，還可以通過配置 `book.json` 文件來修改 gitbook 在編譯書籍時的行為，例如：修改書籍的名稱，顯示效果等等。

book.json

gitbook 在編譯書籍的時候會讀取書籍源碼頂層目錄中的 `book.js` 或者 `book.json`，這裡以 `book.json` 為例，參考 [gitbook 文檔](#) 可以知道，`book.json` 支持如下配置：

```
{
  // Folders to use for output
  // Caution: it overrides the value from the command line
  // It's not advised this option in the book.json
  "output": null,

  // Generator to use for building
  // Caution: it overrides the value from the command line
  // It's not advised this option in the book.json
  "generator": "site",

  // Book metadats (somes are extracted from the README by default)
  "title": null,
  "description": null,
  "isbn": null,

  // For ebook format, the extension to use for generation (default is detected from c
  // "epub", "pdf", "mobi"
  // Caution: it overrides the value from the command line
  // It's not advised this option in the book.json
  "extension": null,

  // Plugins list, can contain "-name" for removing default plugins
  "plugins": [],

  // Global configuration for plugins
  "pluginsConfig": {
    "fontSettings": {
      "theme": "sepia", "night" or "white",
      "family": "serif" or "sans",
      "size": 1 to 4
    }
  },

  // Variables for templating
  "variables": {},

  // Links in template (null: default, false: remove, string: new value)
  "links": {
    // Custom links at top of sidebar
    "sidebar": {
      "Custom link name": "https://customlink.com"
    },

    // Sharing links
    "sharing": {
      "google": null,
      "facebook": null,
      "twitter": null,
      "weibo": null,
      "all": null
    }
  }
}
```

```

    },
  },

  // Options for PDF generation
  "pdf": {
    // Add page numbers to the bottom of every page
    "pageNumbers": false,

    // Font size for the field content
    "fontSize": 12,

    // Paper size for the pdf
    // Choices are [u'a0', u'a1', u'a2', u'a3', u'a4', u'a5', u'a6', u'b0', u'b1', u'b2', u'b3', u'b4', u'b5', u'b6', u'b7', u'b8', u'b9']
    "paperSize": "a4",

    // Margin (in pts)
    // Note: 72 pts equals 1 inch
    "margin": {
      "right": 62,
      "left": 62,
      "top": 36,
      "bottom": 36
    },
  },

  //Header HTML template. Available variables: _PAGENUM_, _TITLE_, _AUTHOR_ and _S
  "headerTemplate": null,

  //Footer HTML template. Available variables: _PAGENUM_, _TITLE_, _AUTHOR_ and _S
  "footerTemplate": null
}
}

```

注意：上面的內容直接從 [gitbook 文檔](#) 中複製，所以可能過期！

首先，將這個文件放到書籍代碼頂層目錄中，命名為 `book.json`，然後編譯書籍：

```
$ gitbook build
```

可以看到，編譯完成，使用

```
$ gitbook serve
```

然後將瀏覽器指向 `http://127.0.0.1:4000`，可以看到，什麼都沒有改變！

是的，雖然這裡 `book.json` 文件非法，但是 `gitbook build` 並沒有報錯！所以，用戶需要自己準備工具來保證 `book.json` 必須是一個合法的 JSON 文件，並且不能含有非法配置項。

首先，刪除註釋項，以及空行，如果是在 vim 中，可以執行下面的命令：

```
:%g/\s*\n//d
:%g/^\s*$//d
```

然後，使用 python 來檢查 book.json 是否合法，同樣，在 vim 中執行下面的命令：

```
:%!python -m json.tool
```

很顯然，下面的配置不能通過，所以刪去（注：但是默認主題卻是使用的這個配置！）。

```
"pluginsConfig": {  
  "fontSettings": {  
    "theme": "sepia", "night" or "white",  
    "family": "serif" or "sans",  
    "size": 1 to 4  
  }  
},
```

最後，剩下的內容如下：

```
{  
  "description": null,  
  "extension": null,  
  "generator": "site",  
  "isbn": null,  
  "links": {  
    "sharing": {  
      "all": null,  
      "facebook": null,  
      "google": null,  
      "twitter": null,  
      "weibo": null  
    },  
    "sidebar": {}  
  },  
  "output": null,  
  "pdf": {  
    "fontSize": 12,  
    "footerTemplate": null,  
    "headerTemplate": null,  
    "margin": {  
      "bottom": 36,  
      "left": 62,  
      "right": 62,  
      "top": 36  
    },  
    "pageNumbers": false,  
    "paperSize": "a4"  
  },  
  "plugins": [],  
  "title": null,  
  "variables": {}  
}
```

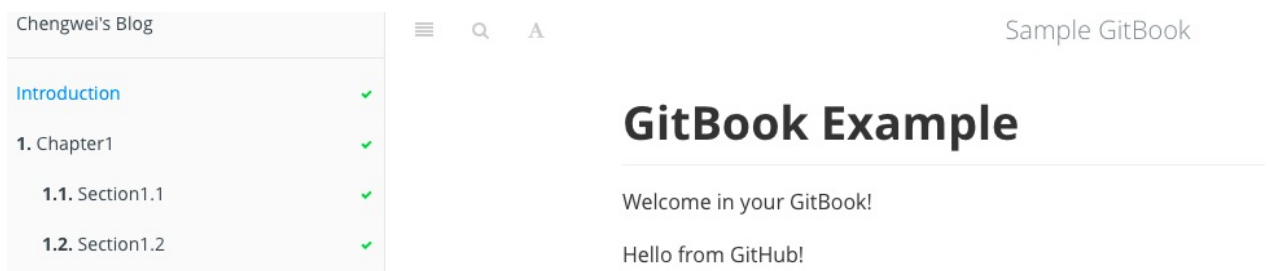
現在，修改一些配置，修改後為：

```

{
  "author": "Chengwei Yang <me@chengweiyang.cn>",
  "description": "This is a sample book created by gitbook",
  "extension": null,
  "generator": "site",
  "isbn": null,
  "links": {
    "sharing": {
      "all": null,
      "facebook": null,
      "google": null,
      "twitter": null,
      "weibo": null
    },
    "sidebar": {
      "Chengwei's Blog": "http://www.chengweiyang.cn"
    }
  },
  "output": null,
  "pdf": {
    "fontSize": 12,
    "footerTemplate": null,
    "headerTemplate": null,
    "margin": {
      "bottom": 36,
      "left": 62,
      "right": 62,
      "top": 36
    },
    "pageNumbers": false,
    "paperSize": "a4"
  },
  "plugins": [],
  "title": "Sample GitBook",
  "variables": {}
}

```

現在，重新編譯書籍，預覽效果，如下圖所示：



可以看到，書籍的標題變成了 "Sample GitBook"，而且在左邊的導航欄中添加了一個鏈接！

需要注意的是：GitBook.com 上的書籍標題經試驗不能通過配置 `book.json` 的方式修改 `title`，需要在書籍的屬性頁面中的 'Settings' 中進行修改！

插件

gitbook 還支持許多插件，用戶可以從 [NPM](#) 上搜索 gitbook 的插件，[gitbook 文檔](#) 推薦插件的命名方式為：

- gitbook-plugin-X: 插件
- gitbook-theme-X: 主題

所以，可以通過以上兩種方式來搜索 gitbook 的插件或者主題。

主題插件

雖然 gitbook 默認的主題已經非常不錯，但是，還有一些非常值得推薦的主題，用戶可以通過在 [NPM](#) 上搜索 `gitbook-theme` 來查找主題插件。

這裡將分別介紹幾種主題插件的使用方法，拋磚引玉！

ComScore

ComScore 是一個彩色主題，默認的 gitbook 主題是黑白的，也就是標題和正文都是黑色的，而 ComScore 可以為各級標題添加不同的顏色，更容易區分各級標題，效果如下圖所示：

编辑书籍

在创建了书籍后，可以使用免费的在线编辑器进行编辑，也可以使用 [gitbook editor](#) 编辑，甚至使用任何喜欢的文本编辑器来编辑，例如：Vim。

在线编辑

进入到书籍的属性页面后，点击 "Edit Book" 按钮即可打开在线编辑器。

GitBook 的在线编辑器对于国内用户来说，很可能不能访问，所以最好还是下载 [gitbook editor](#) 到本地，安装后使用，或者使用自己喜欢的文本编辑器直接编辑。

gitbook editor

gitbook editor 实际上就是一个本地应用版的在线编辑器，使用方式和在线编辑器类似，所见即所得，这里不再介绍，读者可以参考 [gitbook 使用](#) 中的内容。

Git & Markdown

另一种方式，是直接使用文本编辑器，编写 Markdown 文档，然后，使用 Git 提交到书籍的远程项目，当然，提交前，最好在本地使用 `gitbook` 预览效果；提交后，GitBook.com 会自动生成更新书籍的内容。

克隆书籍源代码

GitBook.com 上的每本书都使用 Git 项目来管理，所以，这里首先需要克隆需要编辑书籍的 Git 项目，登陆 GitBook.com 后，跳转到书籍的属性页面，如下图所示：

實用插件

除了主題插件意外，gitbook 也有很多實用性插件，用戶可以在 [GitHub](#) 或者 [NPM](#) 上搜索 `gitbook-plugin` 來查找。

裡將介紹其中幾種插件的使用，包括：

- `disqus`, 集成用戶評論系統

Disqus

[Disqus](#) 是一個非常流行的為網站集成評論系統的工具，同樣，[gitbook](#) 也可以集成 [disqus](#) 以便可以和讀者交流。

首先，需要在 [disqus](#) 上註冊一個賬號，然後添加一個 [website](#)，這會獲得一個關鍵字，然後在集成時配置這個關鍵字即可。

安裝 disqus 插件

可以參考 [插件項目主頁](#) 來安裝，命令如下：

```
$ npm install gitbook-plugin-disqus -g
```

然後，修改 `book.json` 配置文件，添加插件的配置內容：

```
{
  "plugins": ["disqus"],
  "pluginsConfig": {
    "disqus": {
      "shortName": "introducetogitbook"
    }
  }
}
```

注意：上面的 `shortName` 的值就是你在 [disqus](#) 上創建的 [website](#) 獲得的唯一關鍵字。

效果如下圖所示：



multipart

multipart 插件可以將書籍分成幾個部分，例如：

- GitBook Basic
- GitBook Advanced

對有非常多章節的書籍非常有用，分成兩部分後，各個部分的章節都從 1 開始編號。

安裝及配置

和安裝其它插件一樣，執行以下命令：

```
$ npm install gitbook-plugin-multipart -g
```

然後編輯 `book.json` 添加 `multipart` 到 `plugins` 中：

```
"plugins": [  
  "multipart"  
],
```

toggle-chapters

[toggle-chapters](#) 插件的效果是：默認只在目錄導航中顯示章的標題，而不會顯示小節的標題，點擊每一章或者每一節會顯示當前章或節的子目錄，如果有的話，但是同時會收起其它之前展開的章節。所以，個人認為不是非常實用，因為這樣子用戶不能快速跳轉到沒有展開的章節！

安裝和使用非常簡單，參考官方文檔即可！

codeblock-filename

`codeblock-filename` 可以為代碼添加一個文件名，以便顯示當前代碼段屬於的文件。

例如：

hello.py

```
print "hello"
```

發佈到 GitHub Pages

除了能夠將書籍發佈到 [GitBook.com](#) 外，還可以將書籍發佈到 GitHub Pages，由於沒有找到官方文檔，所以這裡記錄的是我自己正在使用的一種方法。

如果讀者不瞭解 GitHub Pages 為何物，簡單說就是一個可以託管靜態網站的 Git 項目，支持使用 markdown 語法以及 Jekyll 來構建，或者直接使用已經生成好的靜態站點。詳細可以參考 [GitHub Pages 主頁](#)。

由於 gitbook 書籍可以通過 `gitbook` 本地構建出 `site` 格式，所以可以直接將構建好的書籍直接放到 GitHub Pages 中託管，之後，可以通過如下地址訪問書籍：

```
<username>.github.io/<project>
```

例如：這本書中使用的例子 'test' 項目可以通過地址：`chengweiv5.github.io/test` 來訪問。

當訪問 `chengweiv5.github.io/test` 時，會訪問 `chengweiv5/test` 項目的 `gh-pages` 分支的內容，所以需要為項目創建一個 `gh-pages` 分支，並且將靜態站點內容放入其中。也就是說，`test` 項目將有如下兩個分支：

- `master`, 保存書籍的源碼
- `gh-pages`, 保存書籍編譯後的 HTML 文件

構建書籍

首先，使用 `gitbook build` 將書籍內容輸出到默認目錄，也就是當前目錄下的 `_book` 目錄。

```
$ gitbook build
Starting build ...
Successfully built!

$ ls _book
GLOSSARY.html      chapter1           chapter2           gitbook            glossary
```

創建 gh-pages 分支

執行如下命令來創建分支，並且刪除不需要的文件：

```
$ git checkout --orphan gh-pages
$ git rm --cached -r .
$ git clean -df
$ rm -rf *~
```

現在，目錄下應該只剩下 `_book` 目錄了，首先，忽略一些文件：

```
$ echo "*~" > .gitignore
$ echo "_book" >> .gitignore
$ git add .gitignore
$ git commit -m "Ignore some files"
```

然後，加入 `_book` 下的內容到分支中：

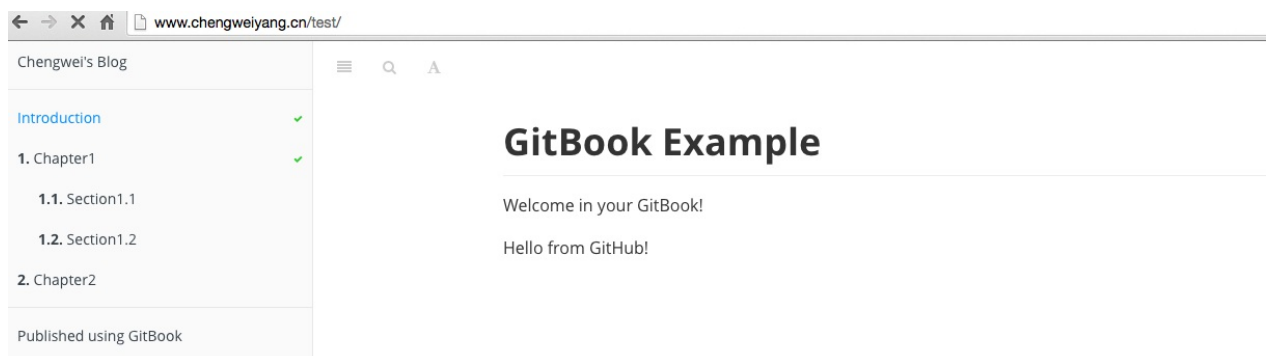
```
$ cp -r _book/* .
$ git add .
$ git commit -m "Publish book"
```

上傳書籍內容到 GitHub

現在，可以將編譯好的書籍內容上傳到 GitHub 中 `test` 項目的 `gh-pages` 分支了，雖然這裡還沒有創建分支，上傳和創建會一步完成！

```
$ git push -u origin gh-pages
Counting objects: 49, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (45/45), done.
Writing objects: 100% (49/49), 1.34 MiB | 131.00 KiB/s, done.
Total 49 (delta 5), reused 0 (delta 0)
To https://github.com/chengweiv5/test.git
 * [new branch]      gh-pages -> gh-pages
Branch gh-pages set up to track remote branch gh-pages from github.
```

現在，書籍的內容已經上傳到 GitHub 上，所以通過訪問 `chengweiv5.github.io/test` 就可以閱讀 `test` 這本書了！



注意：由於我將 `chengweiv5.github.io` 重定向到了個人站點 `www.chengweiyang.cn`，所以可以看到，瀏覽器中的 URL 自動變成了 `www.chengweiyang.cn/test`，非常 cool! 關於怎樣重定向 GitHub Pages 到個人域名，請參考博客：[怎樣使用 GitHub Pages 搭建個人站點](#)。