

版本控制 Git & GitHub

王維駿
Build School 助教

請尊重講師的著作權及智慧財產權！

Build School 課程之教材、程式碼等、僅供課程中學習用、請不要任意自行散佈、重製、分享，謝謝

Git 的優點

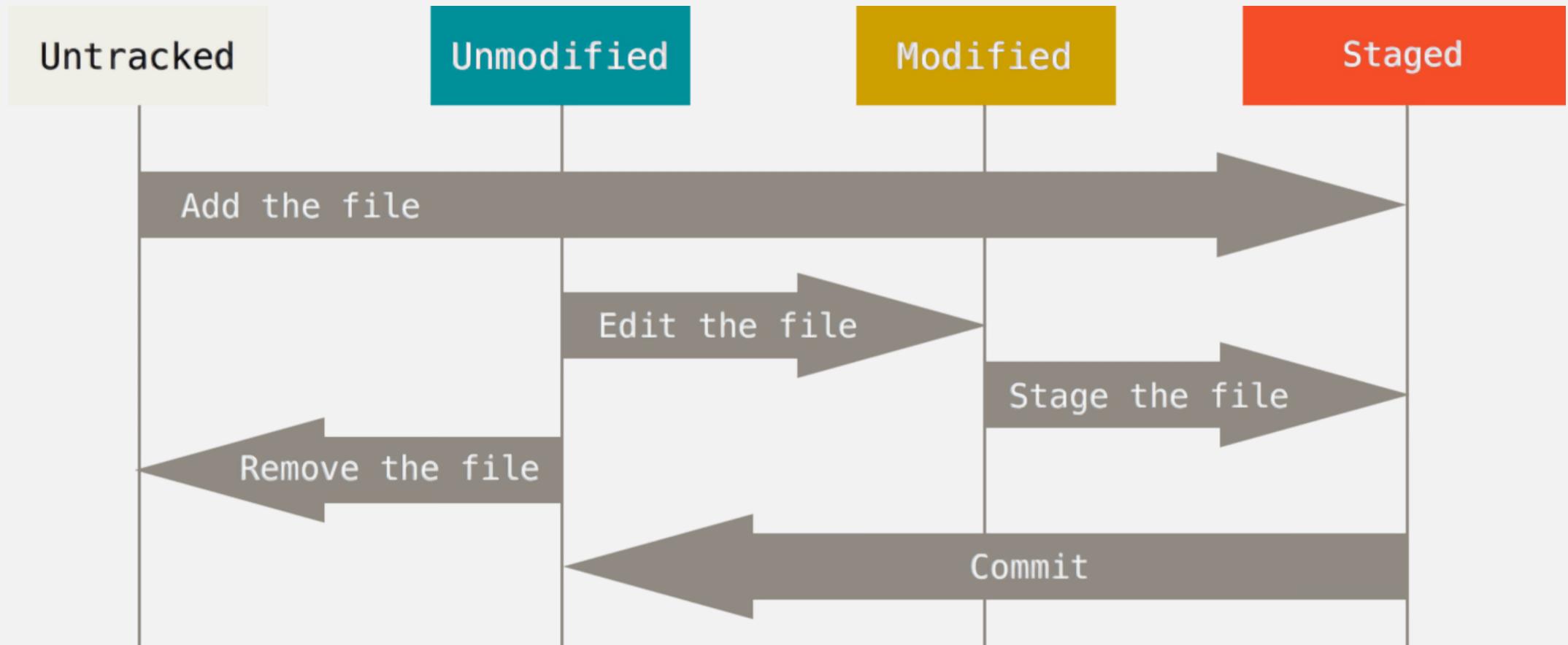
- 免費、開源
- 速度快、體積小
- 分散式系統

安裝

- <https://git-scm.com/>

The screenshot shows the official Git website (<https://git-scm.com/>). At the top left is the Git logo and the tagline "fast-version-control". A search bar is at the top right. The main content area features a diagram of five computer stacks connected by red and blue arrows, symbolizing distributed version control. Below the diagram, a paragraph explains Git's purpose: "Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency." Another paragraph highlights Git's performance: "Git is easy to learn and has a **tiny footprint with lightning fast performance**. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, convenient **staging areas**, and **multiple workflows**." The page also includes sections for "About", "Documentation", "Downloads", and "Community", each with a corresponding icon. A prominent "Latest source Release" section shows "2.23.0" (Release Notes: 2019-08-16) with a "Download 2.23.0 for Mac" button. At the bottom, links are provided for "Mac GUIs", "Tarballs", "Windows Build", and "Source Code".

Git 檔案狀態



Git 身份設定

- 使用者名稱 username
 - \$ git config user.name "Mona Lisa"
 - \$ git config user.name (確認使用者名稱)
- 電子信箱 email address
 - \$ git config --global user.email "email@example.com"
 - \$ git config --global user.email (確認email)

Repository (repo)

- 存放所有檔案變更資料

全新的開始

- 在專案目錄下啟用git
 - 輸入\$ git init

這個命令將會建立一個名為 `.git` 的子資料夾，其中包含 Git 所有必需的倉儲檔案，也就是 Git 倉儲的骨架。到現在這步驟為止，倉儲預設沒有追蹤任何檔案。

看看目前檔案狀態

- \$ git status

決定哪一些檔案要被追蹤

- \$ git add [指定檔案]

[指定檔案] 常用幾種方式：

- 直接寫檔名
- 指定路徑表示該路徑之下所有檔案
- 使用 . 表示所有尚未被放入暫存區的變更

留下紀錄並且放上註解

- \$ git commit -m "[註解內容]"

停止追蹤指定檔案

- \$ git rm [指定檔案]

[指定檔案] 常用幾種方式：

- 直接寫檔名
- 指定路徑表示該路徑之下所有檔案

查看過往的紀錄和註解

- \$ git log ([指定範圍])

常用 `-n` 查看最新 `n` 次的記錄(`n` 為正整數)

ex: `git log -5` 可查看最新五次記錄

也可用 `--since`, `--after` / `--until`, `--before` 指定時間區段

ex: `git log --since="2019-01-01"` 可查看2019年起建立的記錄

比較檔案的不同

- \$ git diff ([想比較的對象])

完全不指定比較對象時，會顯示目前尚未 staged 的檔案變更

想看已 staged 但尚未 commit 的變更需使用 git diff --staged

也可用 hash 值指定兩筆記錄用於比較

ex: git diff 1397a9b 52ec22c (新的記錄通常放後面)

人生就是不斷的後悔

註解打錯字/檔案遺漏

- 改變最後一次記錄
 - \$ git commit --amend

使用方式：

使用 git add / git rm 將要加入/移除記錄的檔案處理好，

(若沒有遺漏檔案則不需上述步驟)

再使用 git commit --amend ，

會跳出編輯器畫面，重寫註解後存檔即完成。

新增(add)錯檔案

- 已經 add 但尚未 commit 時可以使用 reset 處理
 - \$ git reset [指定檔案]

git reset: 將檔案狀態從 staged 變回 modified

git rm: 將檔案從 tracked 變成 untracked

將已經變更的檔案直接回到上個記錄的狀態

- \$ git checkout [指定檔案]

只是看看某個版本時專案的狀況

- \$ git checkout [commit]

想直接回到某個版本並放棄該版本之後的所有記錄

- \$ git reset --hard [commit]

有些檔案不想被 track

- <https://github.com/github/gitignore>
- 增加 .gitignore 檔案之後還要先 add 並 commit
- 已經加入 tracked 的檔案要另外手動移除

使用分支(branch)

使用分支

- 建立新分支
 - \$ git branch [分支名稱]
- 查詢現在有哪些分支
 - \$ git branch

合併分支

- 先切換到想用來合併的分支
 - \$ git checkout [分支名稱]
- 指定要合併哪一個分支
 - \$ git merge [分支名稱]

多人協作

設定遠端Repository

- 你想要把專案放到哪個位置
 - \$ git remote add [路徑識別名稱] [URL]

將檔案丟上雲端

- \$ git push ([指定參數]) [遠端路徑的識別名稱] [branch 名稱]

同步他人進度到本機

- 只取得遠端版本
 - \$ git fetch
- 取得最新版本並合併
 - \$ git pull

GitHub

- 最大的Git Server
- 開發者最好的履歷

GitHub – 建立新的Repo

Screenshot of the GitHub 'Create a new repository' page.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Owner **Repository name ***

 yoruwing / Practice-Github ✓

Great repository names are short and memorable. Need inspiration? How about [stunning-umbrella](#)?

Description (optional)

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

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

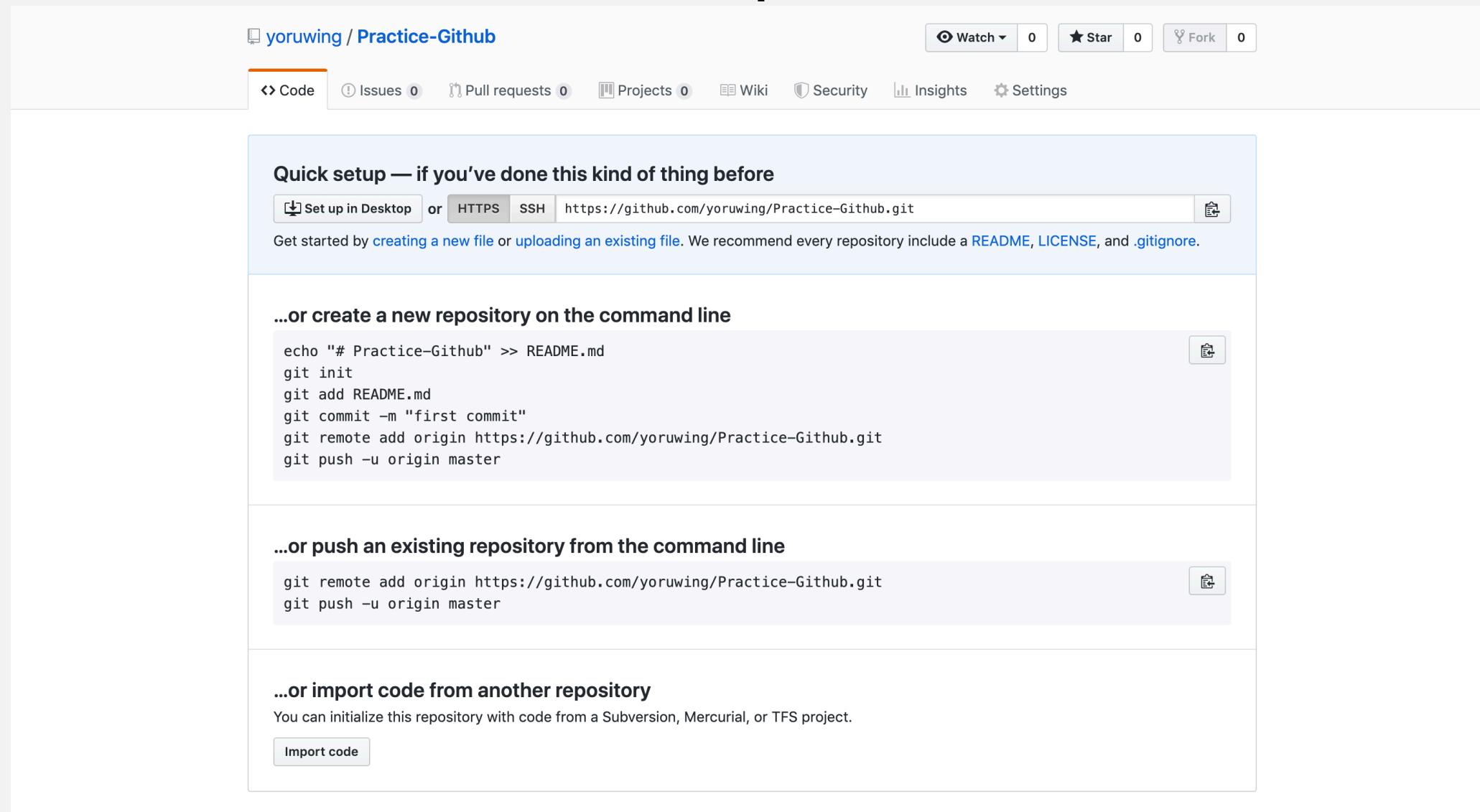
Skip this step if you're importing an existing repository.

Initialize this repository with a README
This will let you immediately clone the repository to your computer.

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

Create repository

GitHub – 設定新的Repo



The screenshot shows the GitHub interface for setting up a new repository. At the top, it displays the repository name "yoruwing / Practice-Github". Below the header, there are buttons for "Watch" (0), "Star" (0), and "Fork" (0). A navigation bar includes "Code" (selected), "Issues 0", "Pull requests 0", "Projects 0", "Wiki", "Security", "Insights", and "Settings".

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH <https://github.com/yoruwing/Practice-Github.git>

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# Practice-Github" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/yoruwing/Practice-Github.git
git push -u origin master
```

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

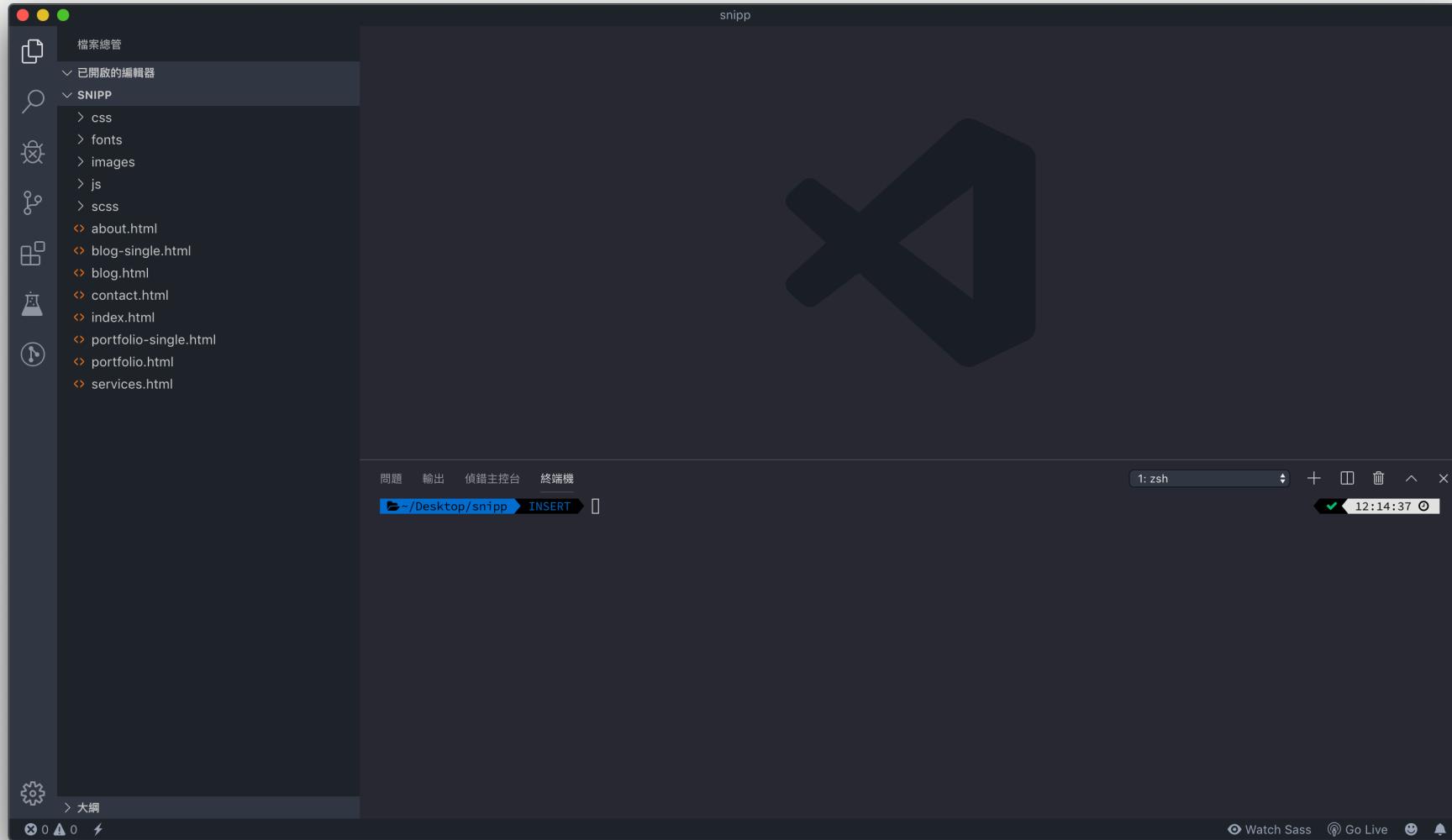
```
git remote add origin https://github.com/yoruwing/Practice-Github.git
git push -u origin master
```

...or import code from another repository

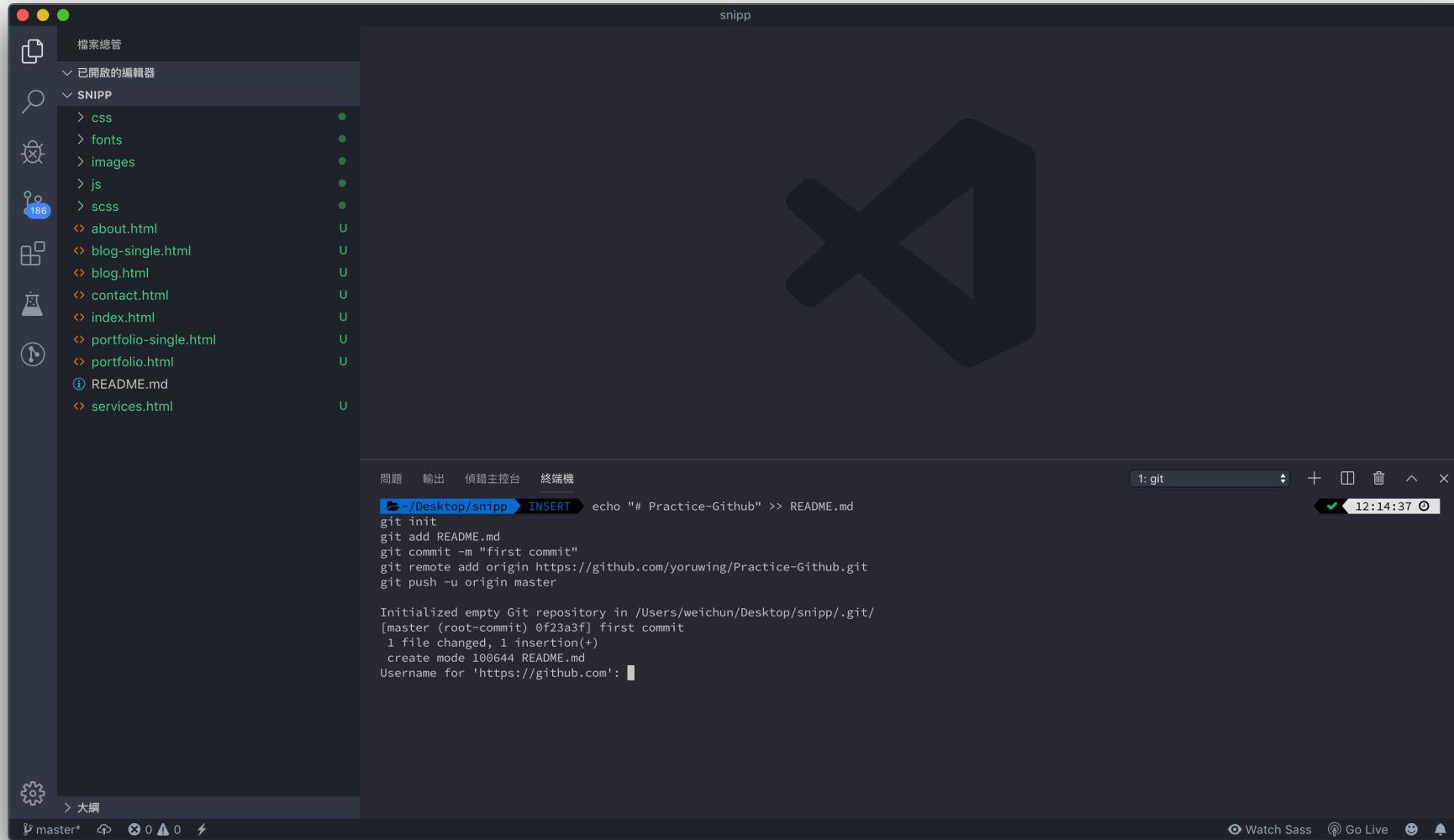
You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Import code

GitHub - 在專案目錄中設置Git



GitHub - 完成身份驗證



GitHub - 回到你的GitHub頁面

The screenshot shows a GitHub repository page for the user 'yoruwing' named 'Practice-Github'. The page includes a navigation bar with links for Pull requests, Issues, Marketplace, and Explore. On the right side of the header are icons for notifications, a plus sign, and a user profile. Below the header, the repository name 'yoruwing / Practice-Github' is displayed, along with statistics: 1 commit, 1 branch, 0 releases, and 1 contributor. There are buttons for Unwatch, Star, Fork, and Edit. A 'Clone or download' button is also present. The main content area shows a single commit from 'yoruwing' titled 'first commit' made 3 minutes ago. The commit message is 'first commit'. At the bottom of the page, there is a 'README.md' file containing the text 'Practice-Github'. The footer contains links for Contact GitHub, Pricing, API, Training, Blog, and About, along with the GitHub logo.

Lab 將其它的檔案Push上GitHub

yoruwing / Practice-Github

Code Issues 0 Pull requests 0 Projects 0 Wiki Security Insights Settings

No description, website, or topics provided.

Manage topics

2 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find File Clone or download

File	Commit Project	Time
css	Commit Project	36 seconds ago
fonts	Commit Project	36 seconds ago
images	Commit Project	36 seconds ago
js	Commit Project	36 seconds ago
scss	Commit Project	36 seconds ago
.DS_Store	Commit Project	36 seconds ago
README.md	first commit	7 minutes ago
about.html	Commit Project	36 seconds ago
blog-single.html	Commit Project	36 seconds ago
blog.html	Commit Project	36 seconds ago
contact.html	Commit Project	36 seconds ago
...