

# 商管程式設計 107-1

# Project Workshop

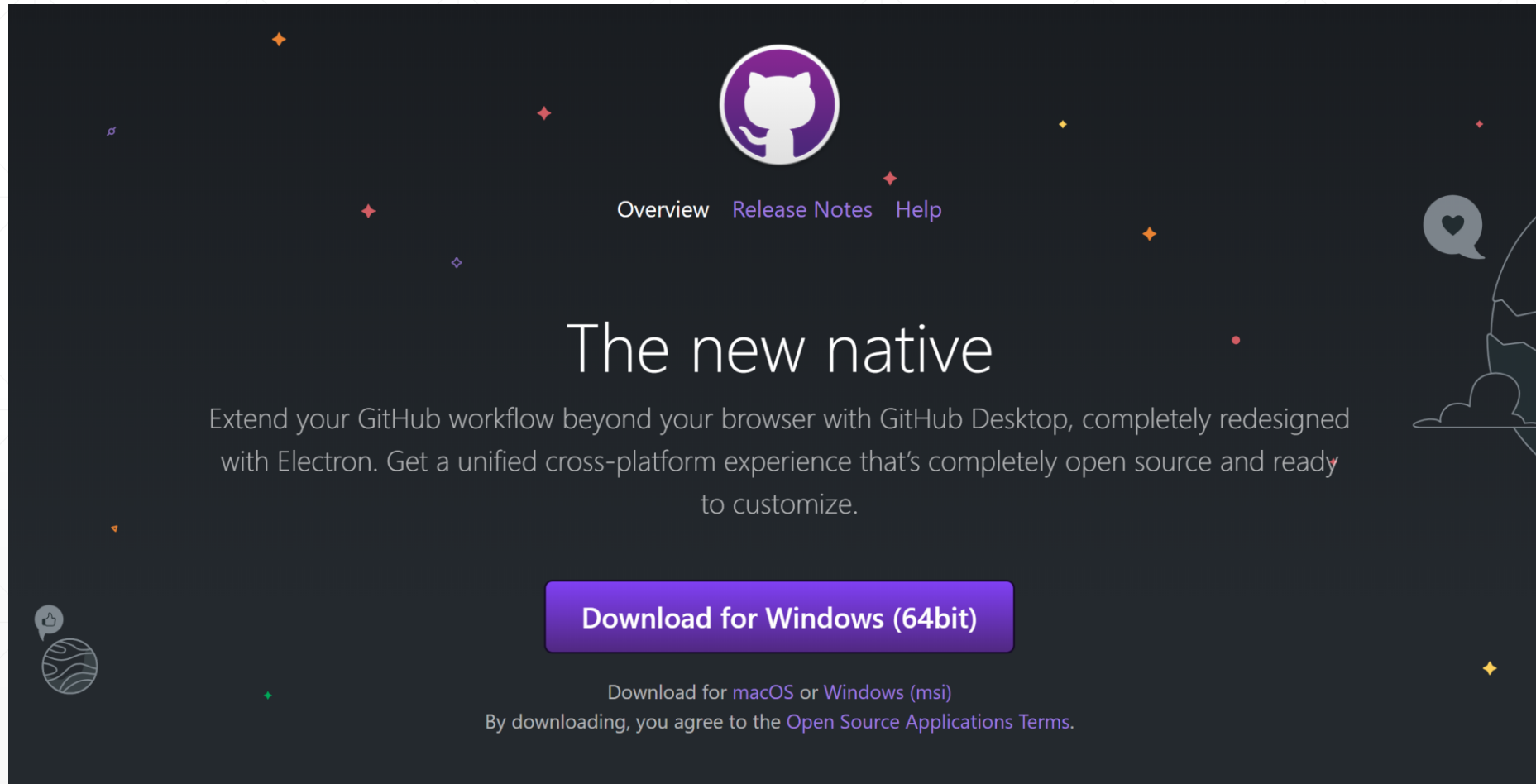
---

2018/12/19-20

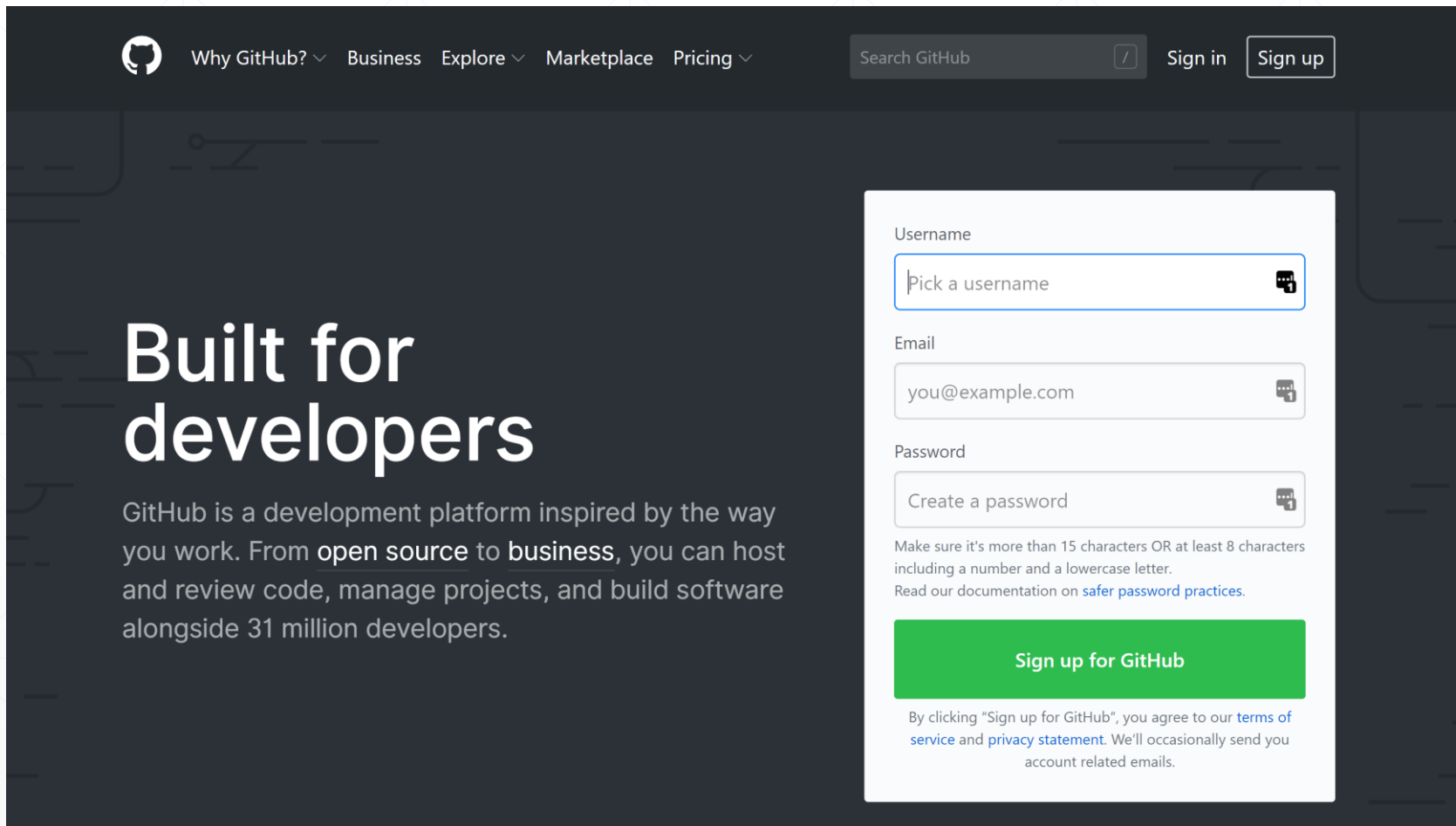
# Agenda

- 版本控制
- GitHub
  - 使用 GitHub
  - GitHub Desktop 操作
- 結語
- Discussion / QA Time

# 節省時間，先下載 <https://desktop.github.com/>



# 節省時間，先註冊 <https://github.com/>

A screenshot of the GitHub website's sign-up page. The background is dark with a faint pattern of code and icons. On the left, the text "Built for developers" is prominently displayed in white, followed by a paragraph describing GitHub as a development platform. On the right, a white sign-up form is overlaid. The form contains fields for "Username", "Email", and "Password", each with a placeholder text and a small icon. Below the password field, there is a green button labeled "Sign up for GitHub". At the bottom of the form, there is a line of text stating that by signing up, the user agrees to the terms of service and privacy statement.

Why GitHub? ▾ Business Explore ▾ Marketplace Pricing ▾ Search GitHub / Sign in Sign up

## Built for developers

GitHub is a development platform inspired by the way you work. From open source to business, you can host and review code, manage projects, and build software alongside 31 million developers.

Username  
Pick a username

Email  
you@example.com

Password  
Create a password

Make sure it's more than 15 characters OR at least 8 characters including a number and a lowercase letter.  
Read our documentation on [safer password practices](#).

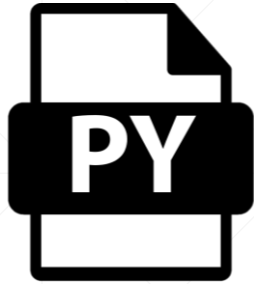
**Sign up for GitHub**

By clicking "Sign up for GitHub", you agree to our [terms of service](#) and [privacy statement](#). We'll occasionally send you account related emails.

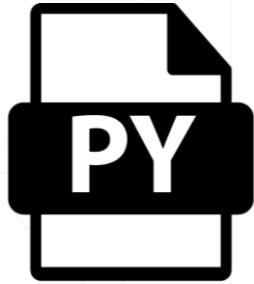
# 版本控制

---

# 你自己寫 code



Project.py



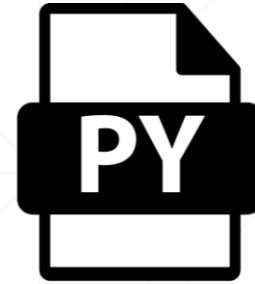
Project\_2.py



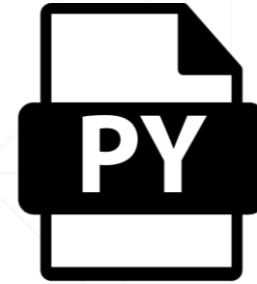
Project\_3.py



Project\_4.py



Project\_5.py



Project\_final  
.py

# 大家一起寫 code



Project.py



Project\_2.py



Project\_3.py



Project  
(2).py



Project (複製)  
.py



Project\_final  
.py



Project\_final  
\_2.py



Project\_final  
\_修正.py



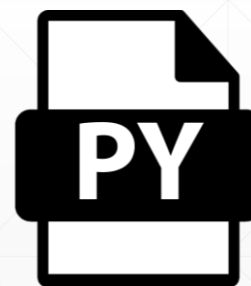
Project\_final  
\_2 (複製).py



Project\_3  
(複製) fix.py



Project 做完  
了.py



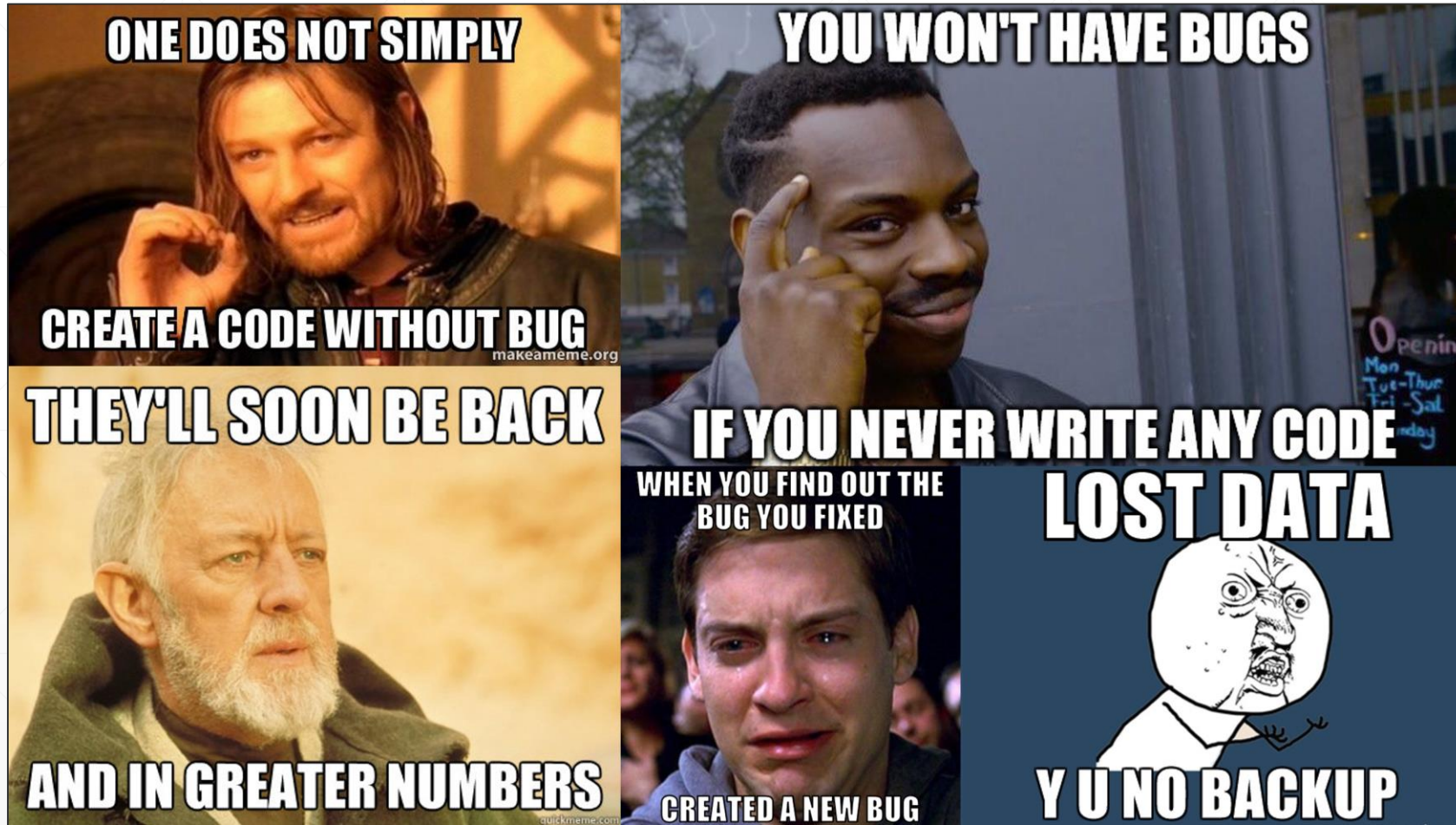
Project\_final  
\_修正 (2).py

# 大家一起寫 code (進階)

- 我改了昨天晚上的 code，你也改了，都傳到群組
- 隔天又有人改了，但忘記是用誰的版本下去改的
- 一個禮拜後發現 code 寫錯了，不知道誰寫的
- 為了要把 code 改過來，改掉某一段
- 又過了一個禮拜，發現改掉的那一段造成另一個功能壞掉
- 去群組找備份，裡面有 20 個檔案，不知道哪個是哪個
- .....



# 你的 code 一塌糊塗，而且沒有人愛你



# 版本控制的用途

- 記錄每一次更新的內容
- 檢查新舊版本之間的差異、衝突
- 保存從舊到新的每一份 code，隨時回退版本
- 有效的群組共同編輯
- ~~抓出是哪個老鼠屎把 code 寫爆了~~

# VCS – 版本控制系統

- 2005 年，一群寫 code 的人 (linux devs) 為了避免悲劇頻繁發生而催生了 git
  - 免費開源！(a.k.a. 任何人都可以用)
- 伺服器上放一份 code
  - 每個人電腦再存一份
- 要寫 code 前 → 從伺服器上下載
- 寫完 code 後 → 更新上去伺服器



# 應用 git 的服務



GitHub



GitLab



BitBucket



SourceForge

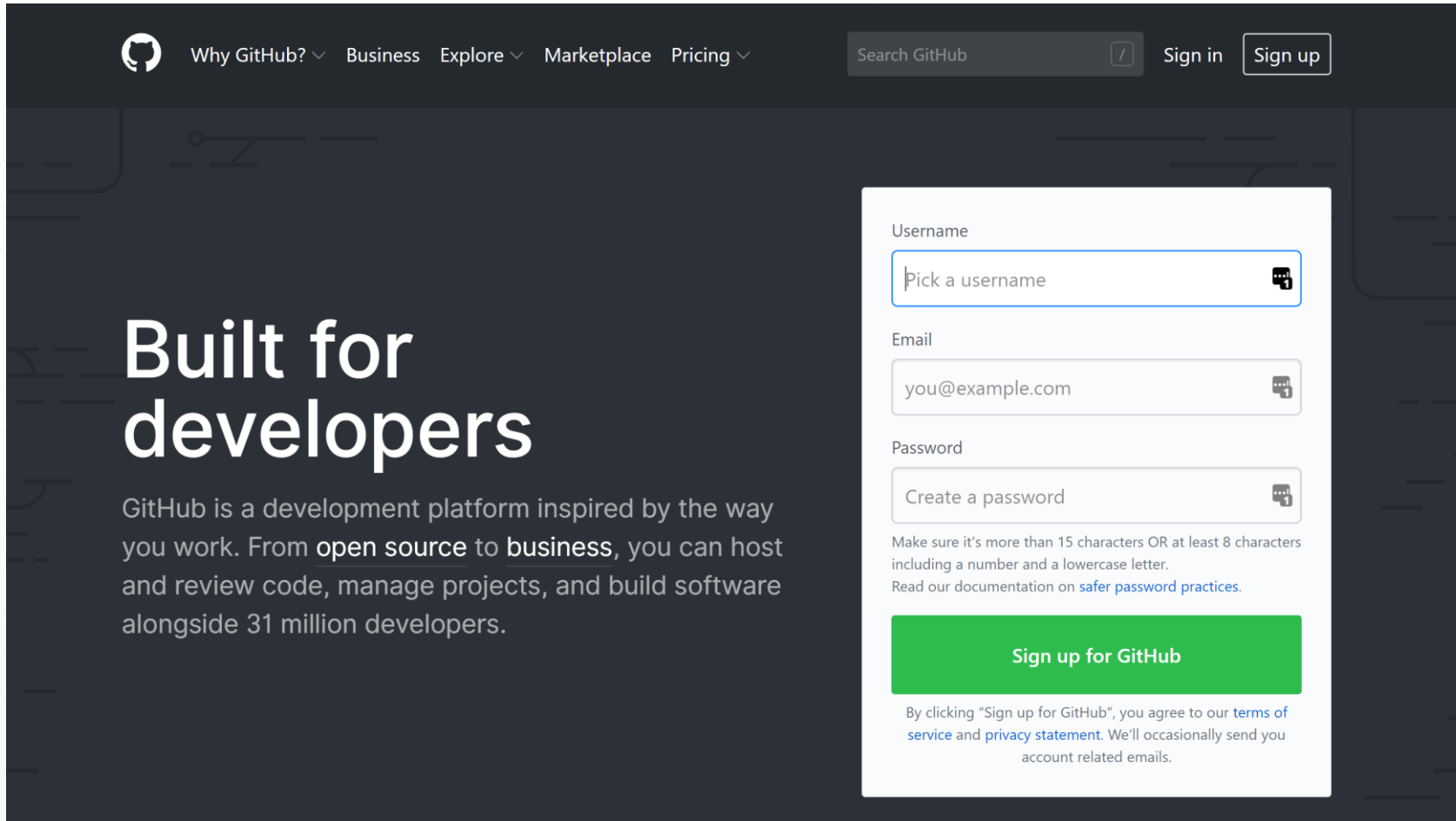
# 為甚麼今天教 GitHub ?

- 最多人用，具有豐富的社群和功能
  - 共同開發同一個大專案
  - 合理的借用別人的 code，發展自己的版本 (然後被採納回去)
  - 和別人線上討論、追蹤議題
  - 建立專案的線上 wiki 說明文件
  - 建立自己的 (靜態) 網站
- 當作自己的求職履歷

# 使用 GitHub

---

註冊 <https://github.com/>

A screenshot of the GitHub website's sign-up page. The page has a dark blue header with the GitHub logo, navigation links (Why GitHub?, Business, Explore, Marketplace, Pricing), a search bar, and 'Sign in' and 'Sign up' buttons. The main content area is dark blue with the text 'Built for developers' and a description of GitHub. On the right, there is a white sign-up form with fields for Username, Email, and Password, each with a strength indicator icon. Below the form is a green 'Sign up for GitHub' button and a disclaimer about terms of service and privacy statement.

Why GitHub? ▾ Business Explore ▾ Marketplace Pricing ▾ Search GitHub / Sign in Sign up

# Built for developers

GitHub is a development platform inspired by the way you work. From open source to business, you can host and review code, manage projects, and build software alongside 31 million developers.

Username  
Pick a username

Email  
you@example.com

Password  
Create a password

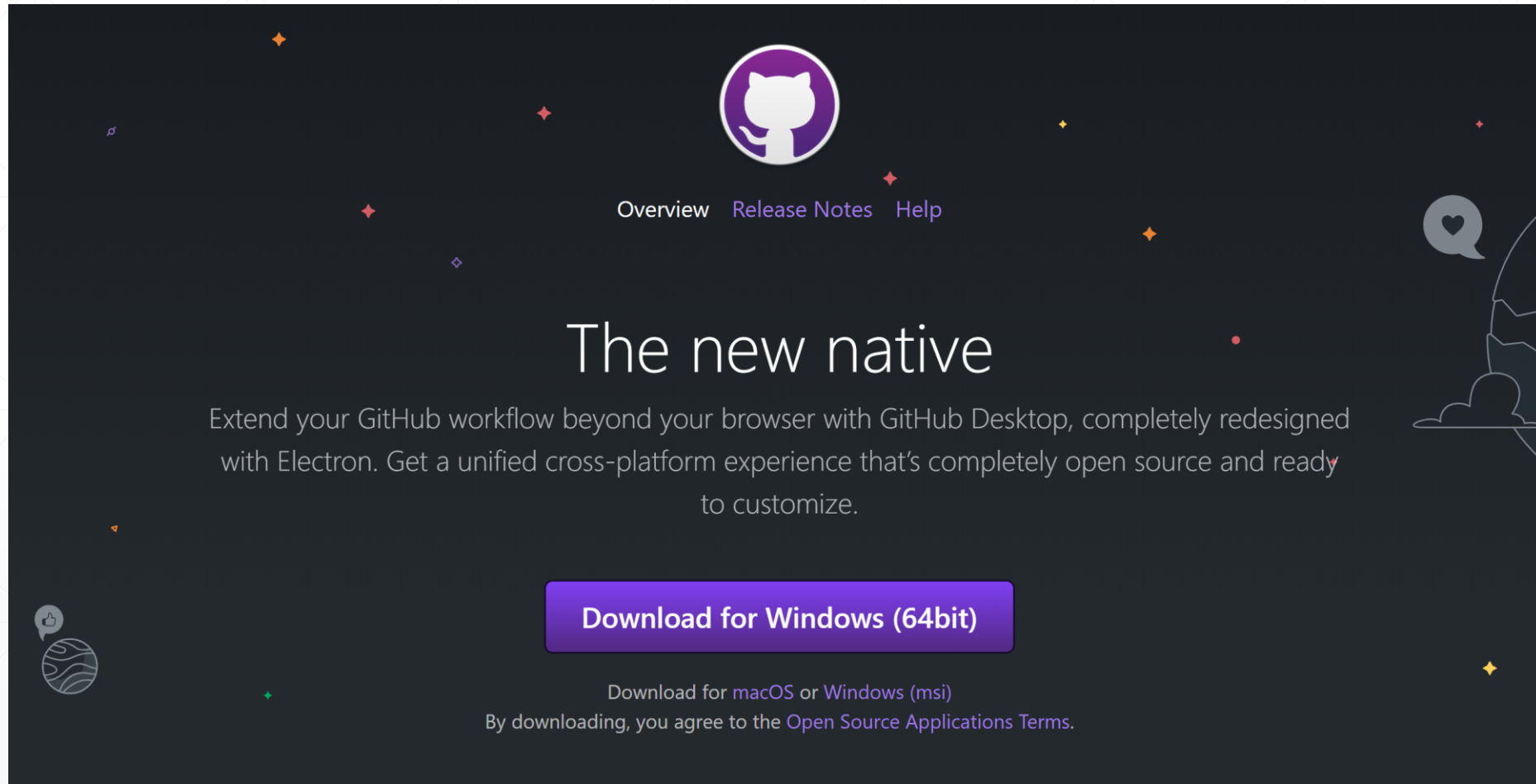
Make sure it's more than 15 characters OR at least 8 characters including a number and a lowercase letter.  
Read our documentation on [safer password practices](#).

**Sign up for GitHub**

By clicking "Sign up for GitHub", you agree to our [terms of service](#) and [privacy statement](#). We'll occasionally send you account related emails.

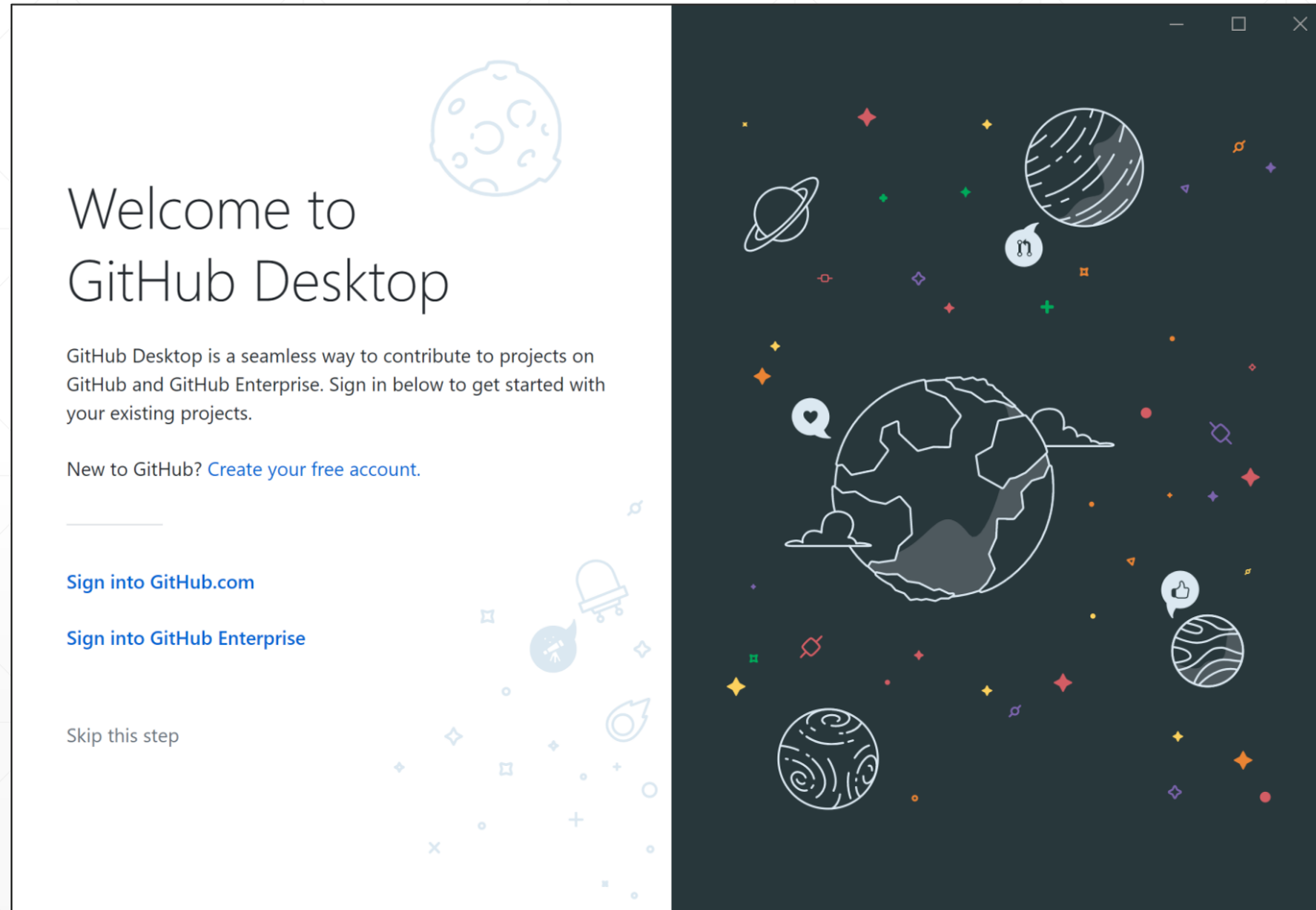


下載 <https://desktop.github.com/>

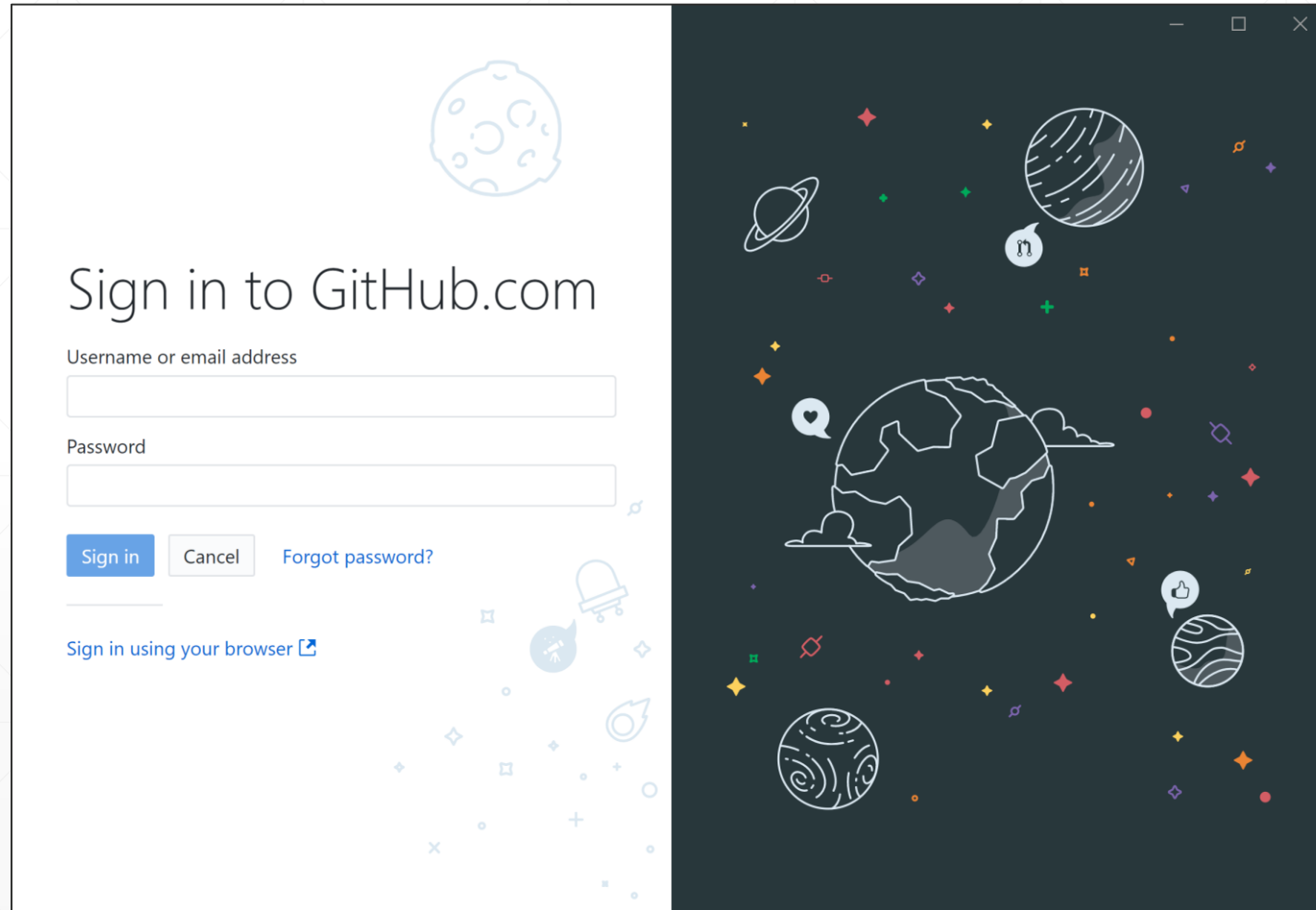




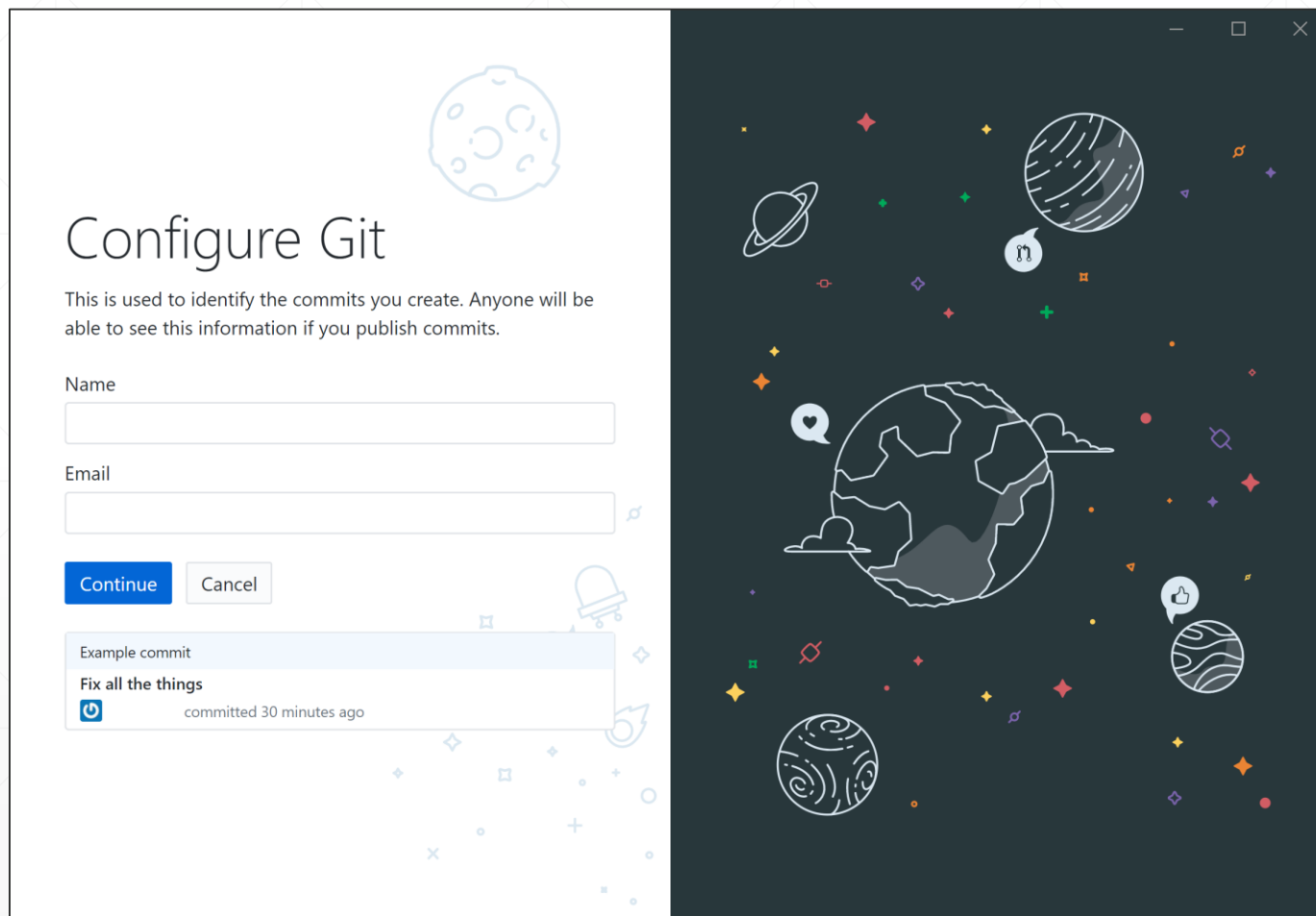
# 打開



# 登入



# 設定名字



The image shows a 'Configure Git' dialog box on the left and a space-themed illustration on the right. The dialog box has a light blue header with a moon icon. Below the title, it explains that the information is used to identify commits. It contains input fields for 'Name' and 'Email', 'Continue' and 'Cancel' buttons, and an 'Example commit' section showing a commit message 'Fix all the things' with a commit icon and timestamp.

**Configure Git**

This is used to identify the commits you create. Anyone will be able to see this information if you publish commits.

Name

Email

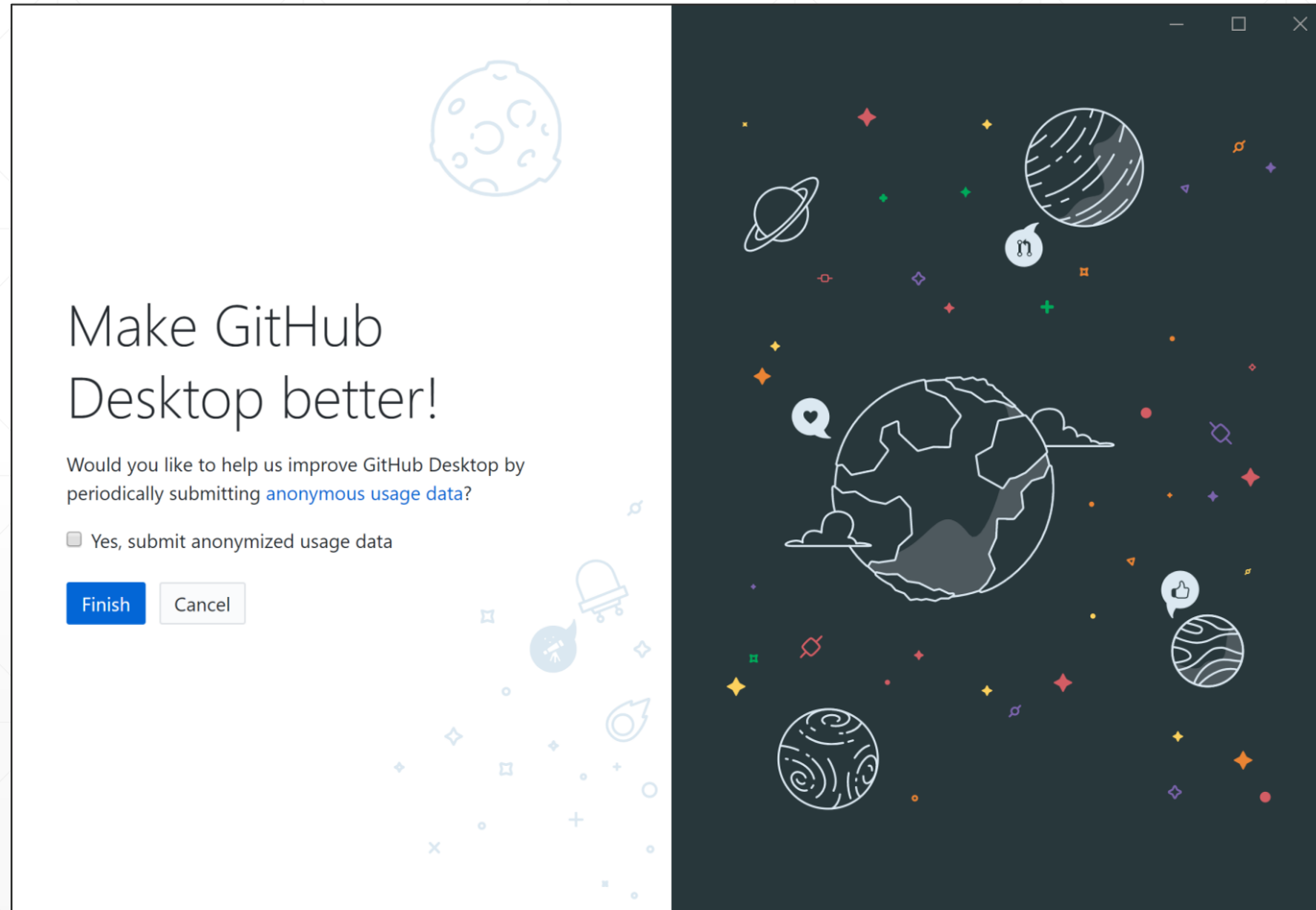
**Continue** **Cancel**

Example commit

**Fix all the things**  
committed 30 minutes ago

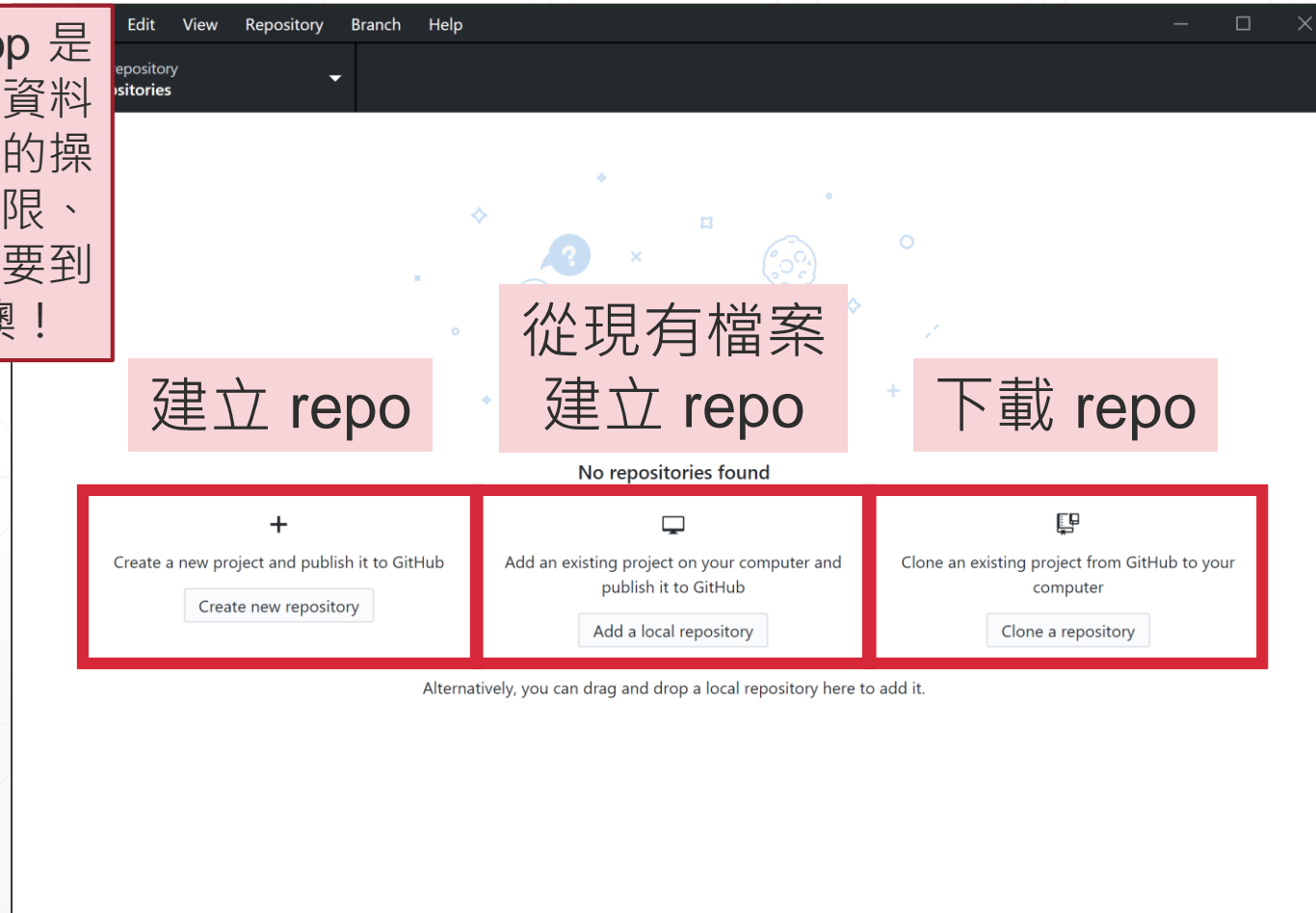
The background illustration on the right is a dark space scene with a large Earth in the center, a ringed planet (Saturn) at the top left, and several other planets and stars. There are also small icons of a heart, a thumbs up, and a person, suggesting a social or collaborative theme.

# 設定結束



# 主畫面

Note: GitHub desktop 是幫助掌管電腦上同步資料的程式，有些純雲端的操作 (ex. 設定上傳權限、刪除雲端資料) 仍然要到 GitHub 網站上操作噢！



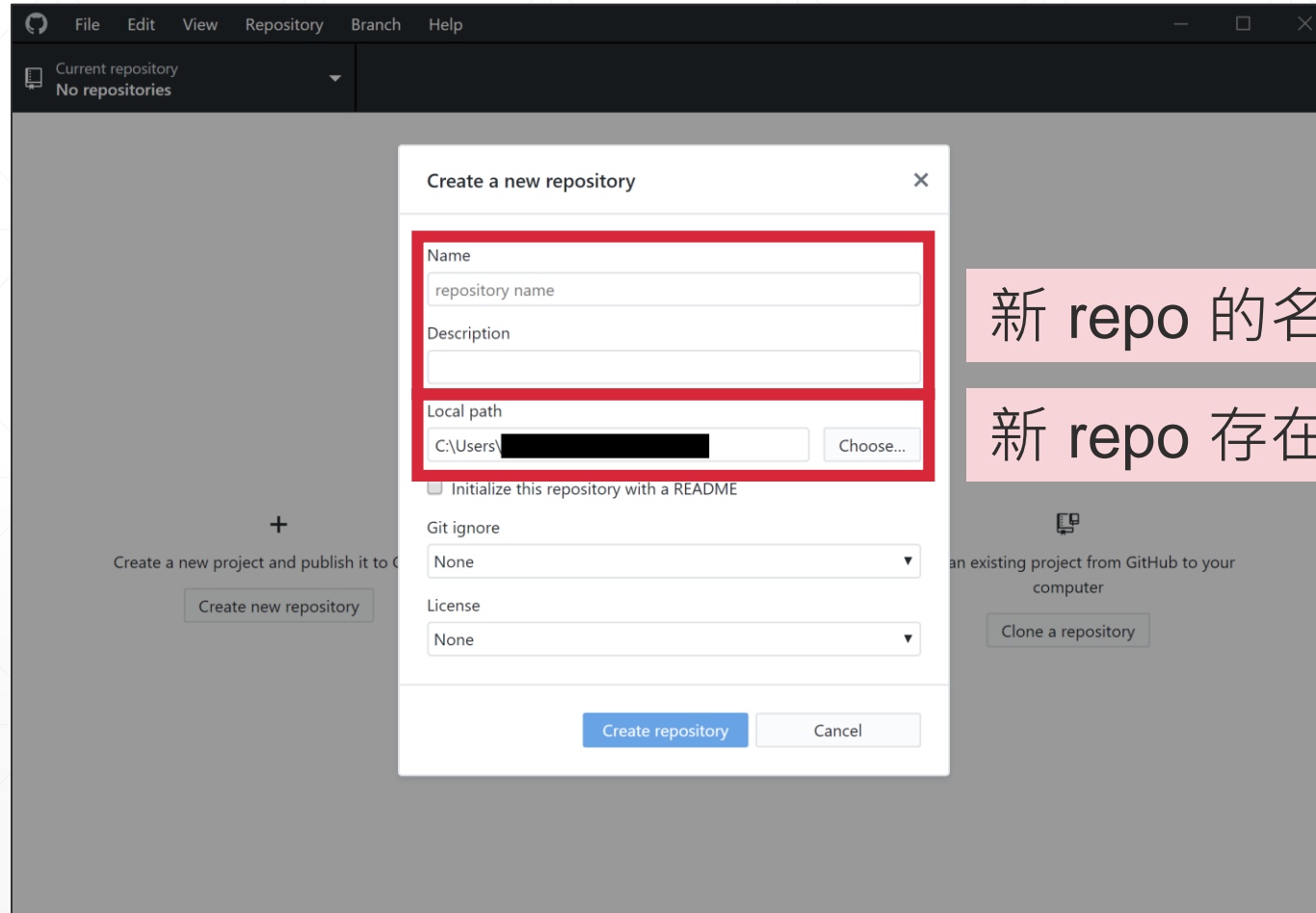
# 甚麼是 repository (repo) ?

- 等於一個**線上倉庫**
  - 一個專案 or 一個共享資料夾
- 把需要版本控制的東西通通丟上去：.py .html.....
  - 不用版本控制的也通通丟上去：.mp3 .jpg .csv.....
  - 反正倉庫裡面想放甚麼就放甚麼 (放不用錢)
- 所有 repo 都是**公開**的
  - GitHub 上不公開的要收費 or 不要用 GitHub

# GitHub Desktop 操作

---

# New (建立新的 repo)

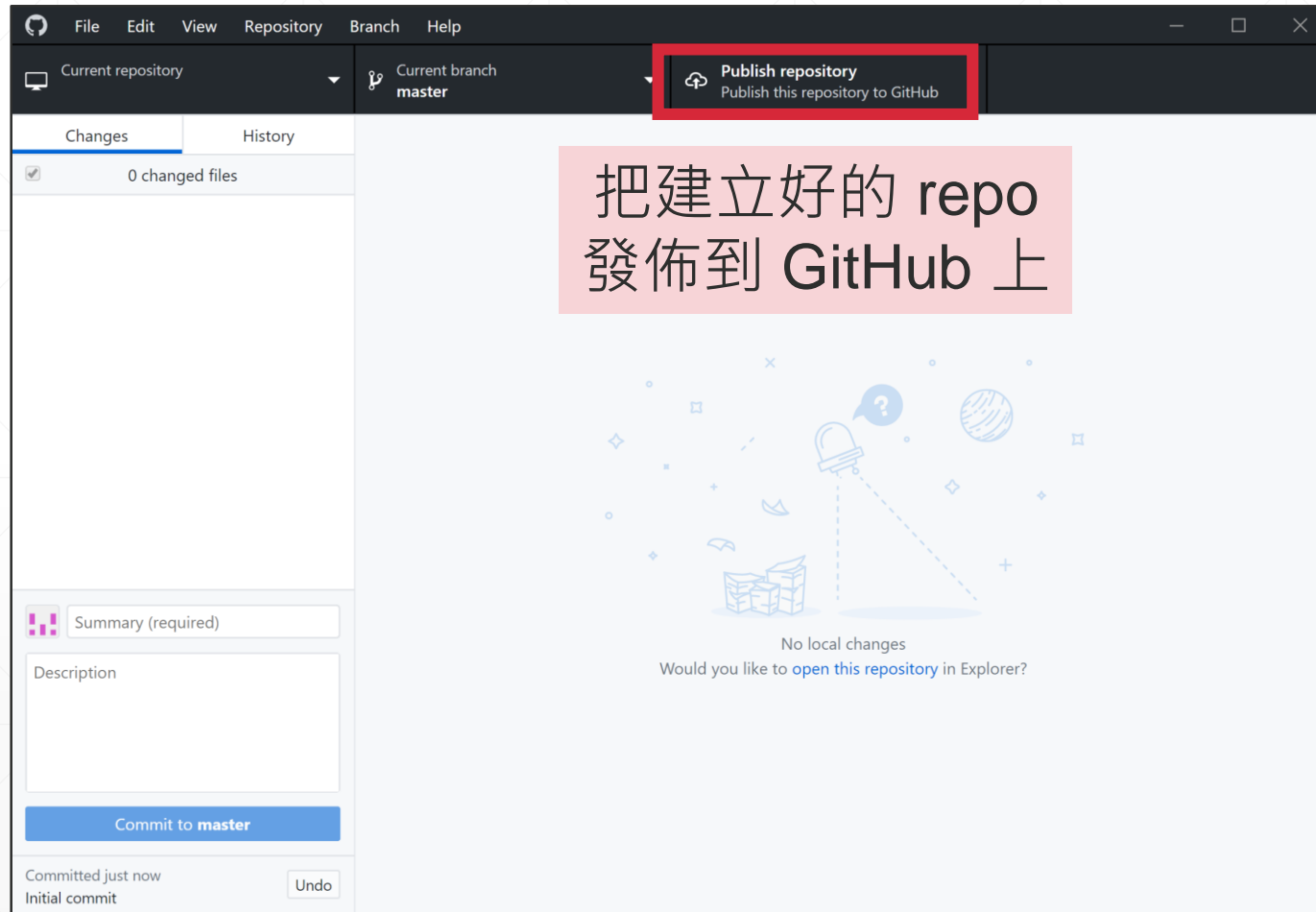


新 repo 的名字和敘述

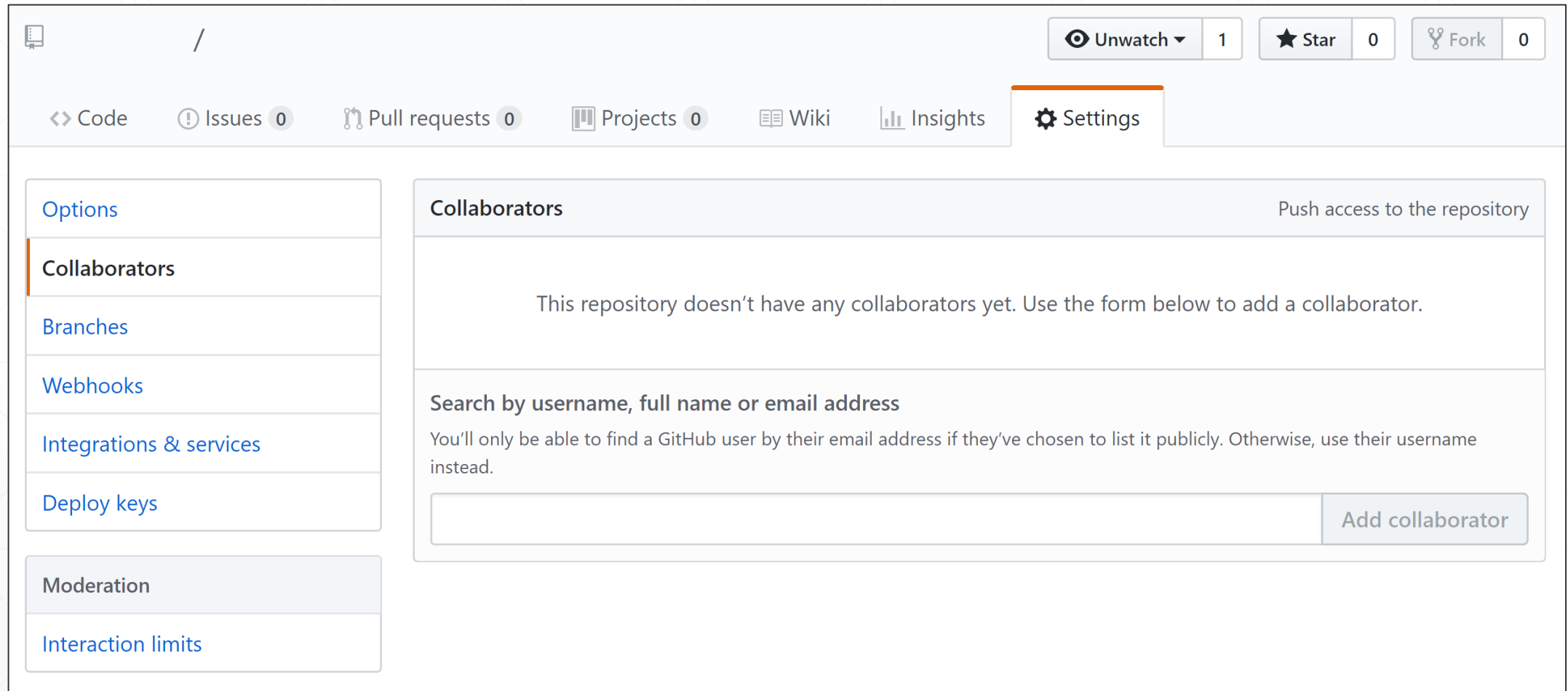
新 repo 存在電腦的哪裡



# Publish (發布到雲端)



# Collaborators (共編成員)



The screenshot shows the GitHub repository settings page for a repository. The top navigation bar includes links for Code, Issues (0), Pull requests (0), Projects (0), Wiki, Insights, and Settings (selected). On the right, there are buttons for Unwatch (1), Star (0), and Fork (0). The left sidebar contains a list of settings: Options, Collaborators (selected), Branches, Webhooks, Integrations & services, Deploy keys, Moderation, and Interaction limits. The main content area is titled 'Collaborators' and includes a sub-header 'Push access to the repository'. Below this, a message states: 'This repository doesn't have any collaborators yet. Use the form below to add a collaborator.' A search bar is provided with the label 'Search by username, full name or email address' and a note: 'You'll only be able to find a GitHub user by their email address if they've chosen to list it publicly. Otherwise, use their username instead.' An 'Add collaborator' button is located to the right of the search bar.

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Options

**Collaborators**

Branches

Webhooks

Integrations & services

Deploy keys

Moderation

Interaction limits

**Collaborators** Push access to the repository

This repository doesn't have any collaborators yet. Use the form below to add a collaborator.

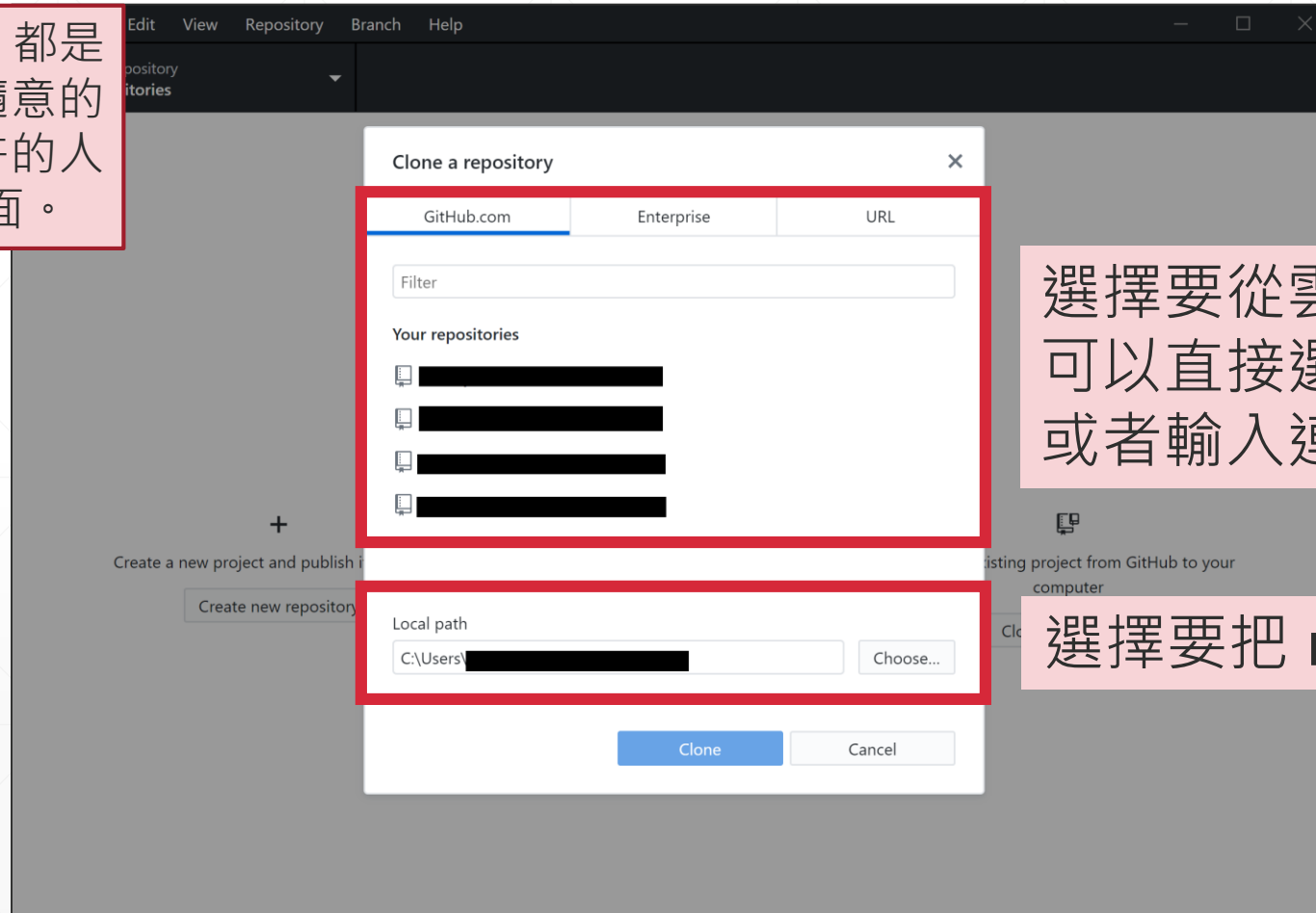
**Search by username, full name or email address**

You'll only be able to find a GitHub user by their email address if they've chosen to list it publicly. Otherwise, use their username instead.

Add collaborator

# Clone (從雲端下載 repo)

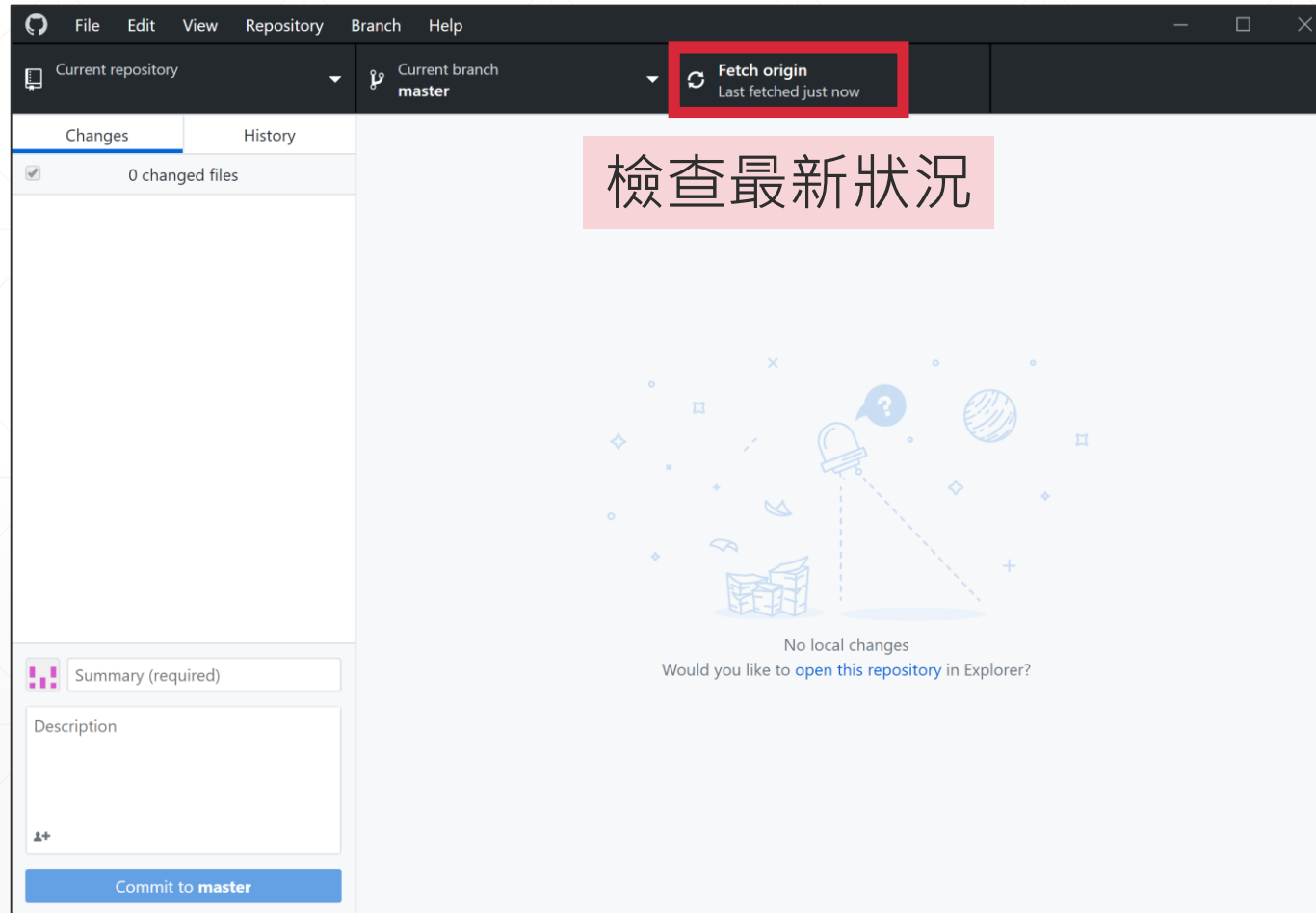
Note: 因為所有 repo 都是公開的，所以可以隨意的下載，但只有被允許的人才能上傳到 repo 裡面。



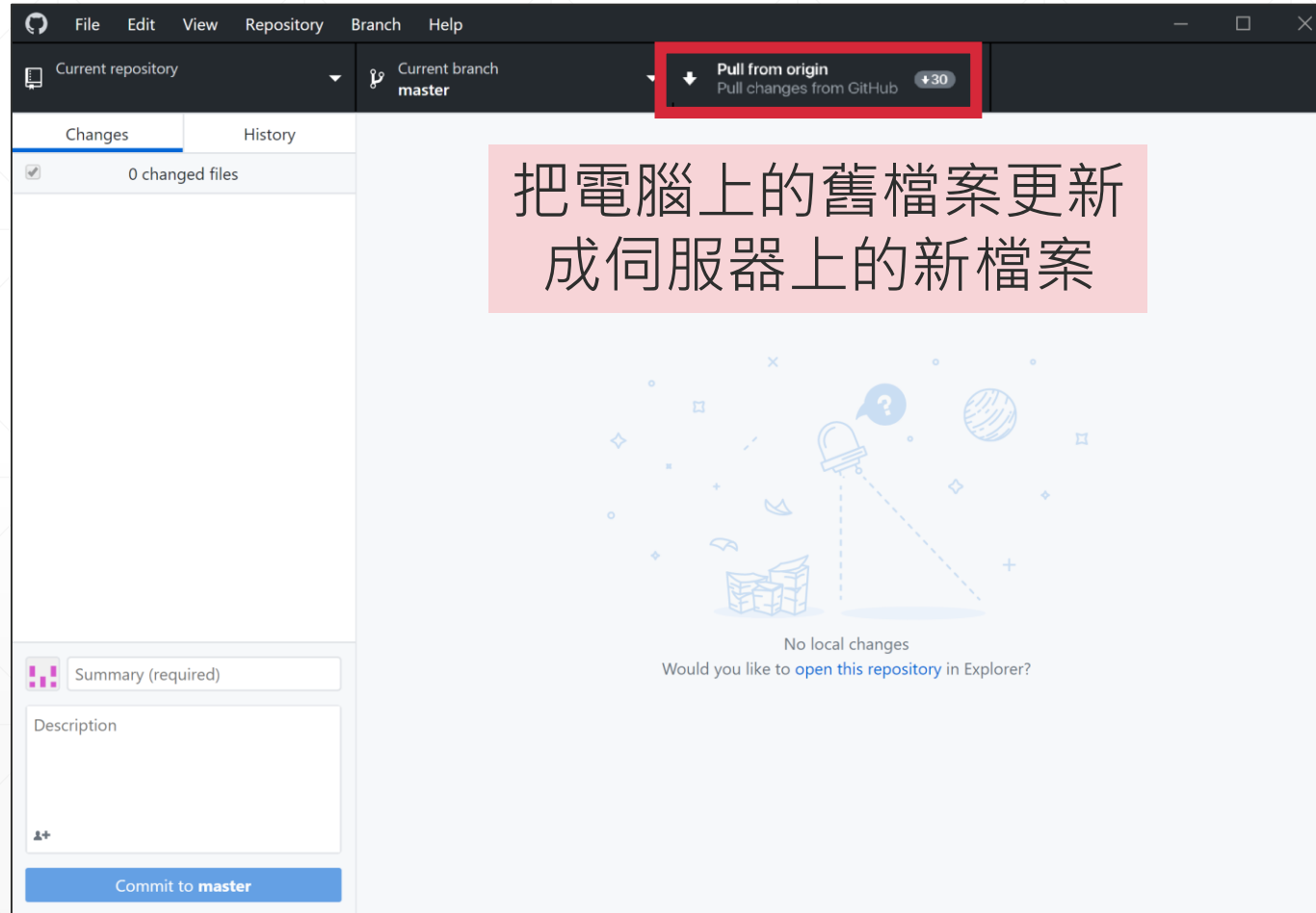
選擇要從雲端下載的 repo 可以直接選取自己的，或者輸入連結下載別人的

選擇要把 repo 存在哪裡

# Fetch (檢查他人進度)



# Pull (從雲端更新到電腦)

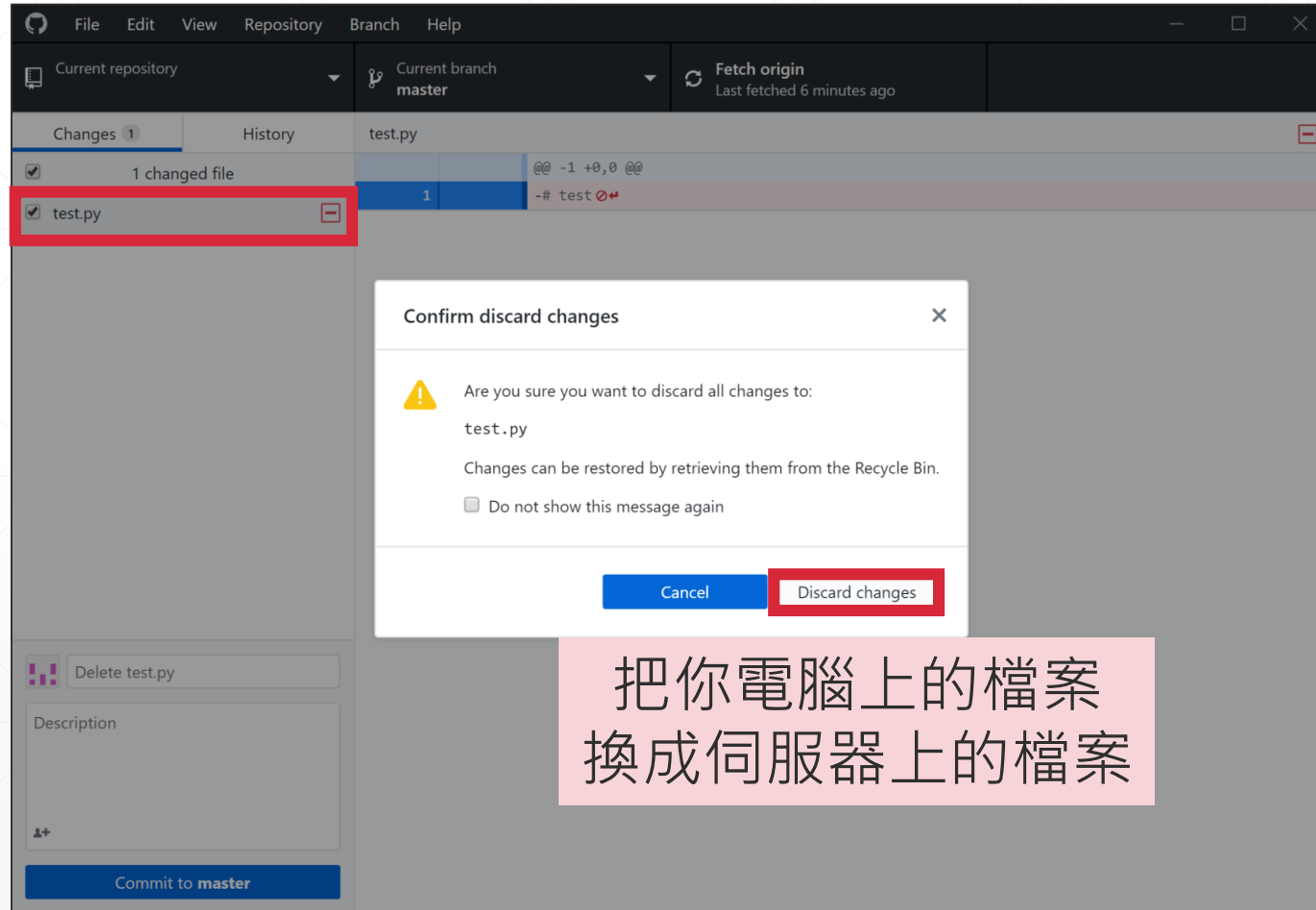


# 開始打 code

```
31     def __init__(self, settings):
32         self.file = None
33         self.fingerprints = set()
34         self.logdupes = True
35         self.debug = debug
36         self.logger = logging.getLogger(__name__)
37         if path:
38             self.file = open(os.path.join(path, 'requests.log'),
39                             'a')
40             self.file.seek(0)
41             self.fingerprints.update(self.request_fingerprint(request) for request in self.requests)
42
43     @classmethod
44     def from_settings(cls, settings):
45         debug = settings.getbool('SUPERFILTER_DEBUG')
46         return cls(job_dir(settings), debug)
47
48     def request_seen(self, request):
49         fp = self.request_fingerprint(request)
50         if fp in self.fingerprints:
51             return True
52         self.fingerprints.add(fp)
53         if self.file:
54             self.file.write(fp + os.linesep)
55
56     def request_fingerprint(self, request):
57         return request_fingerprint(request)
```

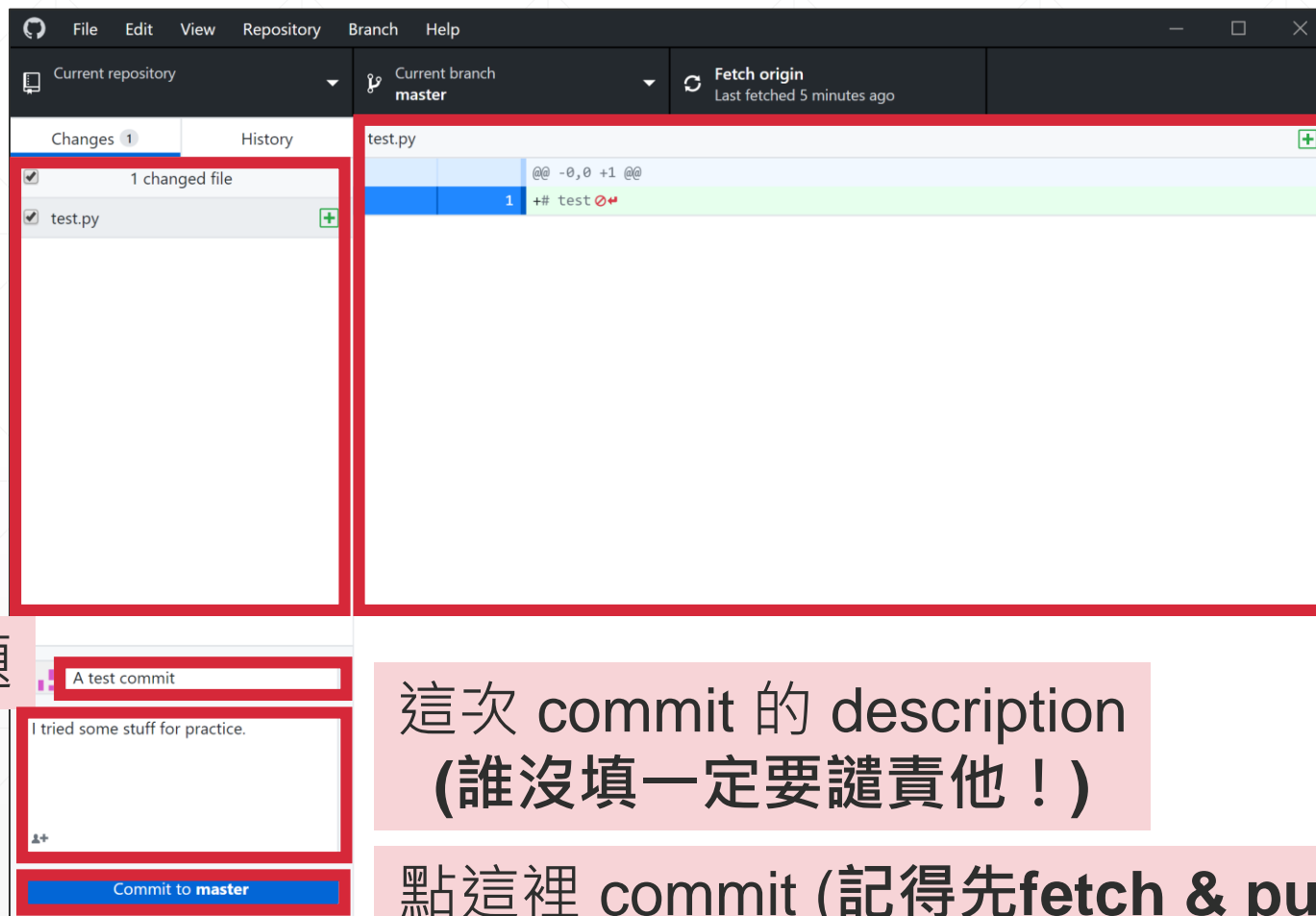
# Discard Changes (取消修改)

在要取消修改 (ex. 誤刪、改錯) 的檔案上按右鍵 → Discard Changes



把你電腦上的檔案  
換成伺服器上的檔案

# Commit (提交/設定存檔點)



這次 commit 更新的內容清單

檔案增刪狀況

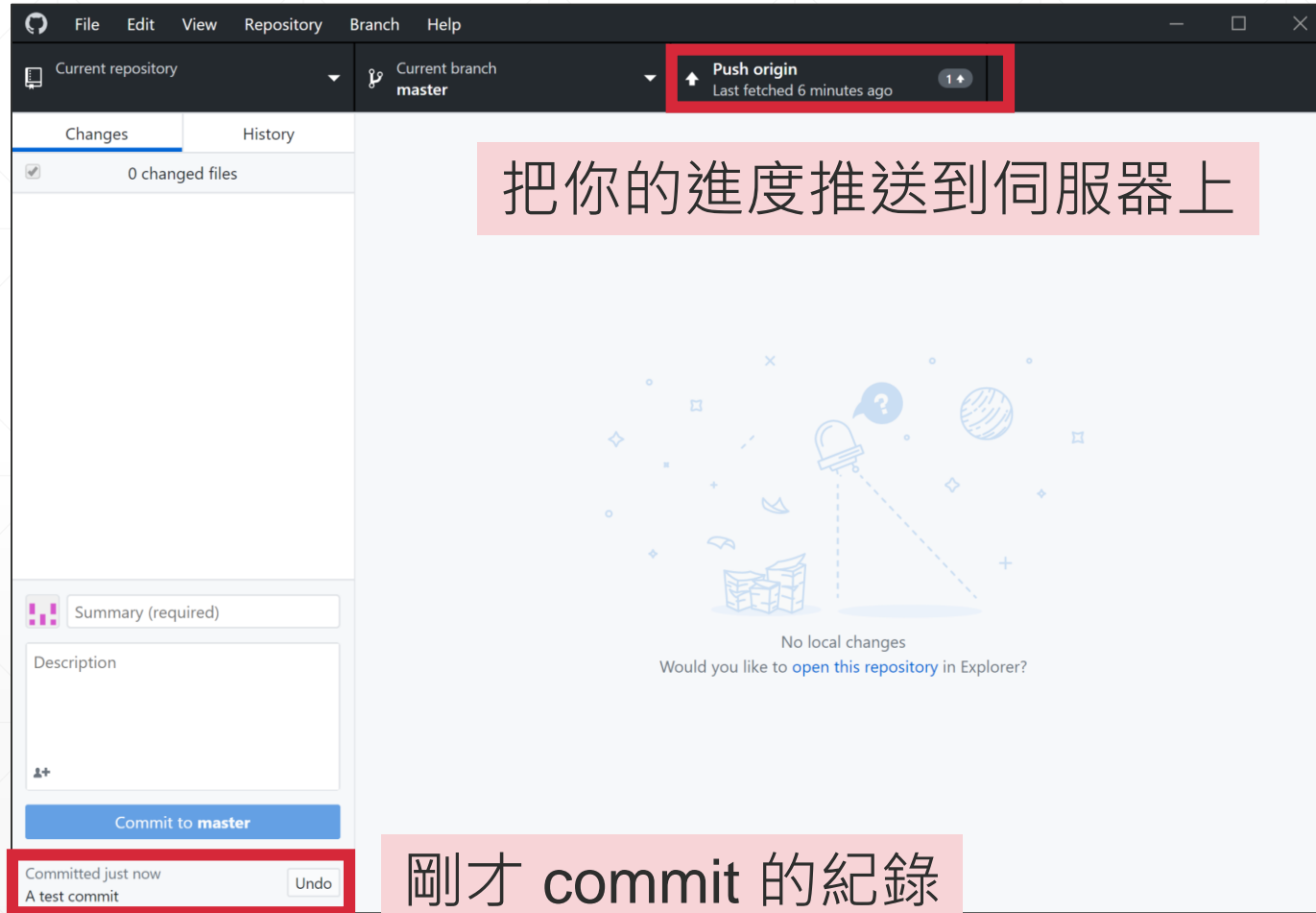
這次 commit 的標題

這次 commit 的 description  
(誰沒填一定要譴責他！)

點這裡 commit (記得先fetch & pull！)

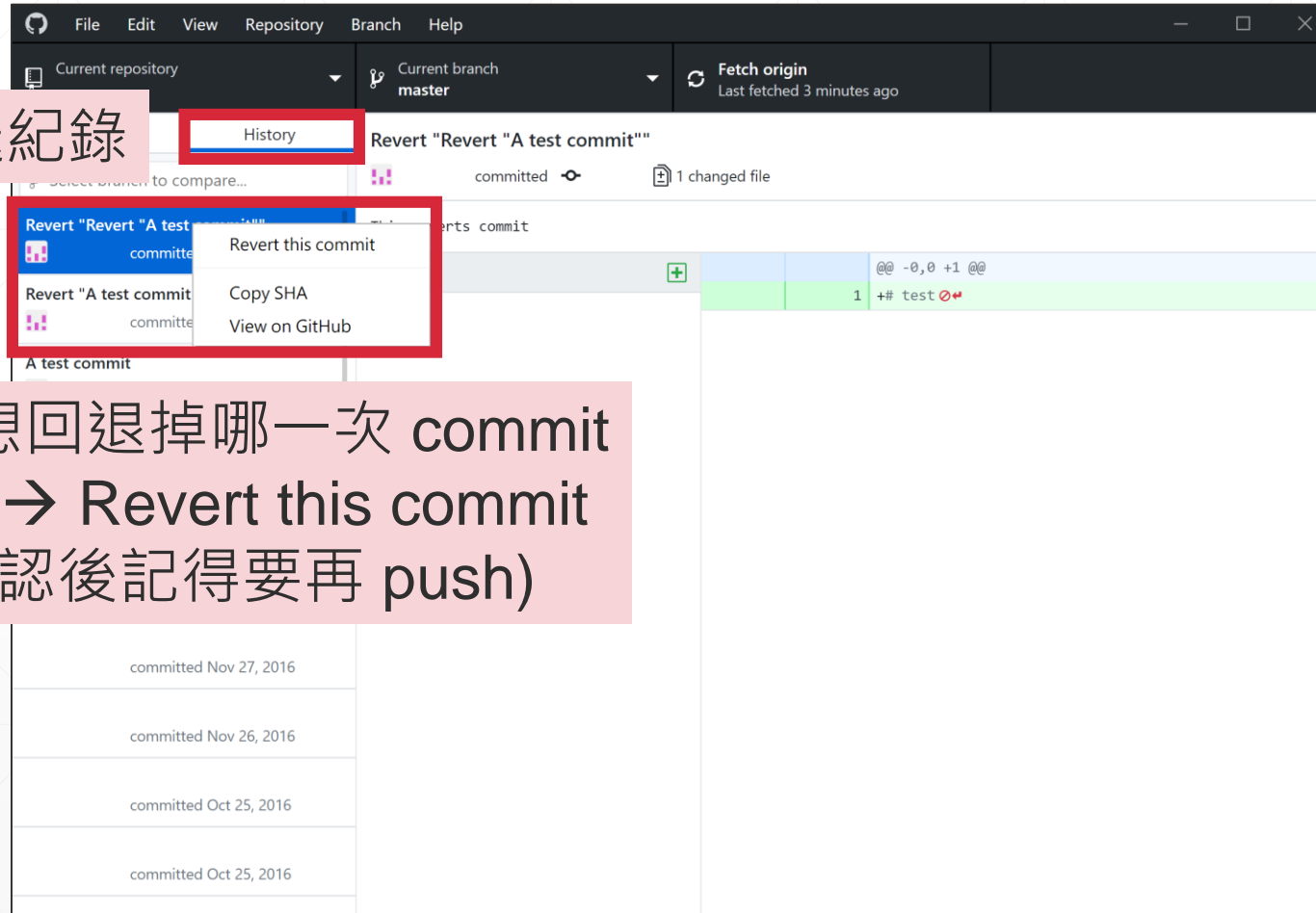


# Push (推送)



# Revert (回退 commit)

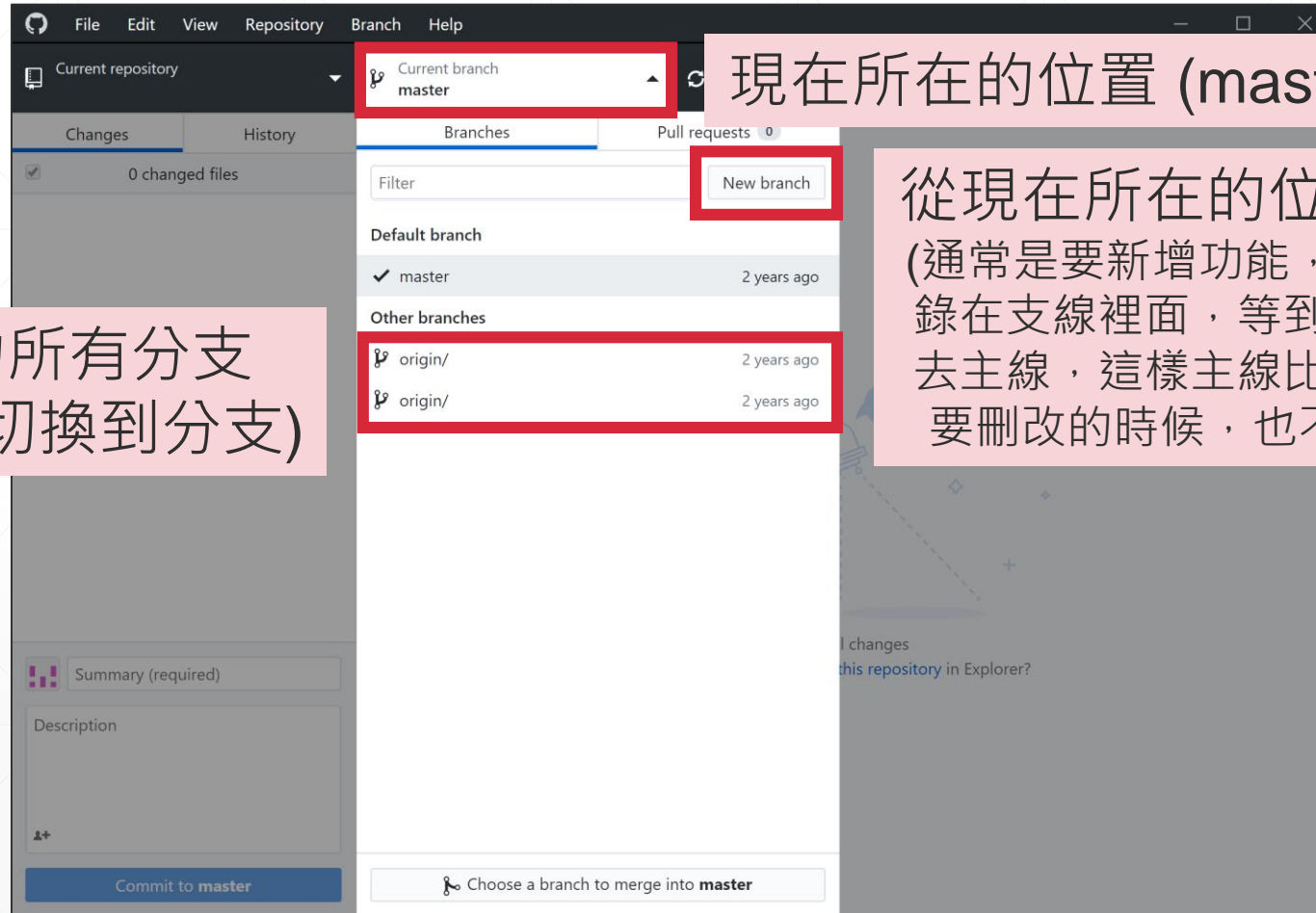
切換到推送紀錄



選擇想回退掉哪一次 commit  
右鍵 → Revert this commit  
(確認後記得要再 push)

# Branch (分支/不想打擾別人用)

現有的所有分支  
(點選可切換到分支)

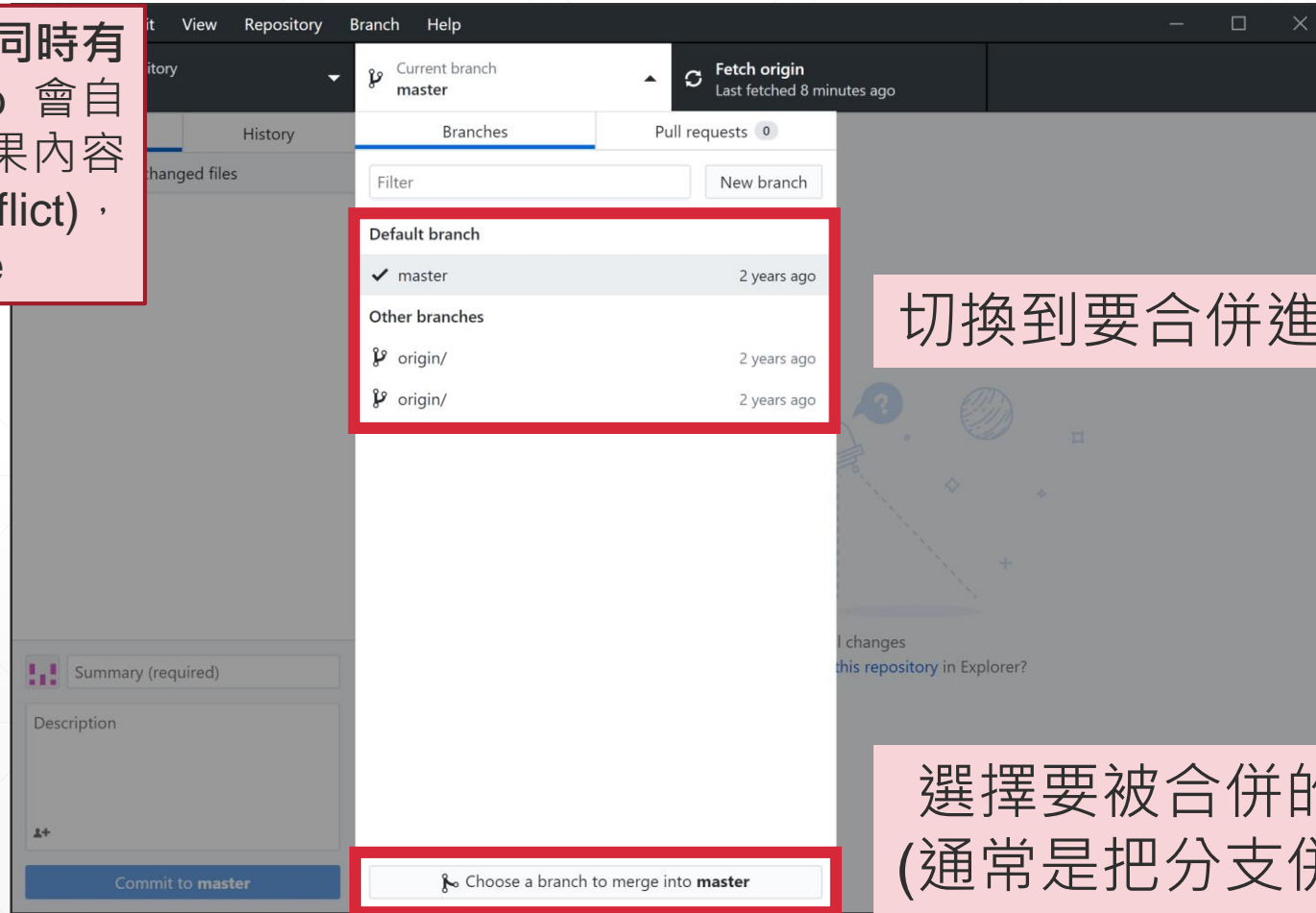


現在所在的位置 (master = 主線)

從現在所在的位置分出去一條線  
(通常是要新增功能，把寫新東西的過程紀錄在支線裡面，等到新東西寫好了再併回去主線，這樣主線比較不會亂；萬一寫錯要刪改的時候，也不會動到主線的內容)

# Merge (合併)

Note: 如果你和別人同時有兩個 **push**，GitHub 會自動幫你 **merge**；如果內容互相衝突 (產生 **conflict**)，那就需要手動 **merge**



切換到要合併進入的主分支

選擇要被合併的旁分支  
(通常是把分支併入主線)

# 其他功能

- Fork

- 把別人的 repo 分一份出去，變成自己的 repo 自己的進度線

- Pull Request

- 當你 fork 了別人的 repo、並且做出了修訂之後，可以發送 pull request 給原本的主人，請他參考你的版本
  - 如果他覺得你寫得很好，就可以把你的版本 pull 回去合併

# 請 (主動) 做好版本控制



# 結語

---

# 期末報告的建議

- 一定要寫**好**你的 code
  - 要有註解，變數和 class 名稱不要亂取
- 做好版本控制
- 多和組員溝通討論，口頭報告要好好準備
  - 組內互評估 20% 報告成績，全班互評估 15%
- 善用 Google 搜尋和 Stack Overflow 網站
  - 用英文搜尋 python3 *your question*
- **注意死線，不要遲交** (也不能遲交)



# 更多 GitHub 參考資料

- 連猴子都懂的 Git 入門指南  
<https://backlog.com/git-tutorial/tw/>
- 五倍紅寶石 - 為你自己學 Git  
<https://gitbook.tw/>
- Mr. Opengate - Git 與 GitHub 版本控制超簡易教學  
<http://mropengate.blogspot.com/2015/04/git-GitHub.html>
- Will 保哥 - 30 天精通 Git 版本控管  
<https://GitHub.com/doggy8088/Learn-Git-in-30-days>

# Discussion / Q&A Time

---