GIT&GITHUB

製作:JX



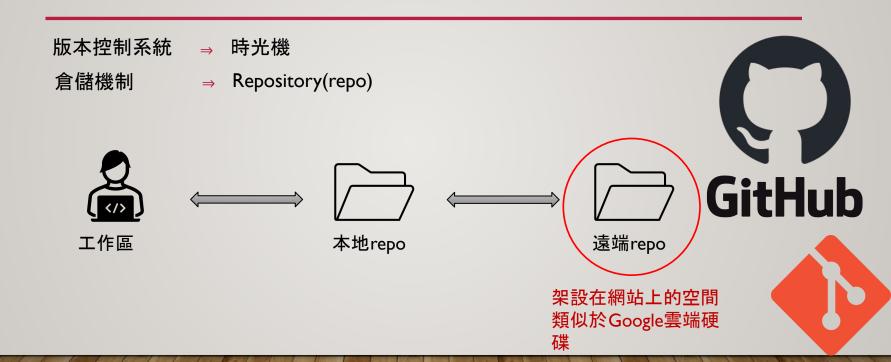
在開始之前...

是搶答環節!

- Q: 你和王小明、陳小華的期末分組報告需要製作一份投影片, 製作的過程三人都必須參與, 你會怎麼做?
- A:使用Google簡報共用編輯。
- Q:如果用Google簡報的合作模式套用到撰寫程式碼, 會遇到什麼困難?
- A:
 - · 出Bug了怎麼辦?
 - 程式規模越來越大, 怎能確保團隊合作無間?



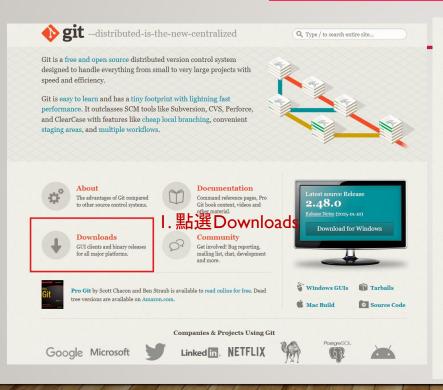
什麼是GIT?



GIT環境設定



首先, 前往 GIT官方網站 安裝程式





(Windows) 3. 直接點選下載

4. 一路 Next到底



Q Type / to search entire site...

About

Documentation

Downloads

GUI Clients

Logos

Community

The entire Pro Git book written by Scott Chacon and Ben Straub is available to read online for free, Dead tree versions are available on Amazon.com.

Download for Windows

Click here to download he latest (2.47.1(2)) 64-bit version of Git for Windows. This is the most recent maintained build. It was released yesterday, on 2025-01-14.

Other Git for Windows downloads

Standalone Installer

32-bit Git for Windows Setup.

64-bit Git for Windows Setup.

Portable ("thumbdrive edition") 32-bit Git for Windows Portable.

64-bit Git for Windows Portable.

Using winget tool

Install winget tool if you don't already have it, then type this command in command prompt or

winget install --id Git.Git -e --source winget

The current source code release is version 2.48.0. If you want the newer version, you can build it from the source code.

Now What?

Now that you have downloaded Git, it's time to start using it.



(Mac)

Install Homebrew

\$ /bin/bash -c "\$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"



Homebrew

Install homebrew if you don't already have it, then:

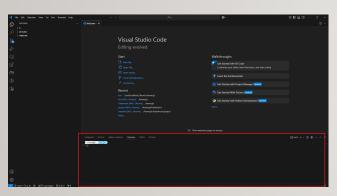
\$ brew install git



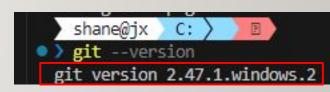
安裝完成後,於「終端機」中輸入命令

方法二(Windows): 搜尋cmd, 打開命令提示字元

方法一: VS Code的終端機區塊







代表已安裝完成

方法二(Mac):

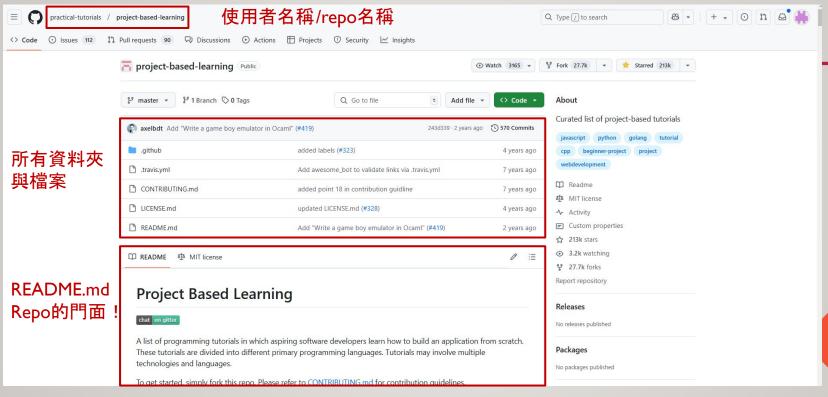
完command+空白鍵, 搜尋並打開「終端機」

實作時間(5min)

安裝GIT



GITHUB REPO





操作時間 (5MIN)

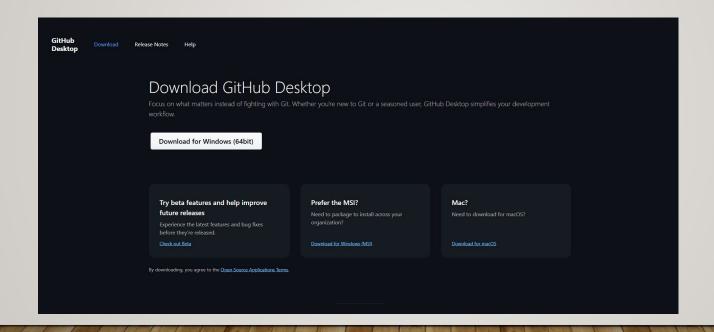
註冊GITHUB



HELLO, GIT!

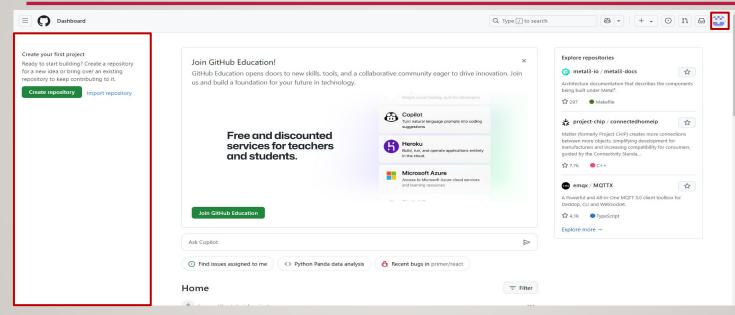


下載GITHUB DESKTOP並以GITHUB帳號登入





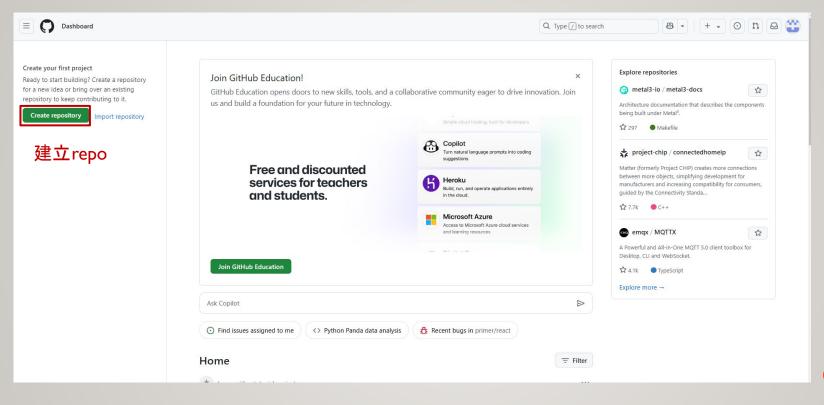
主畫面



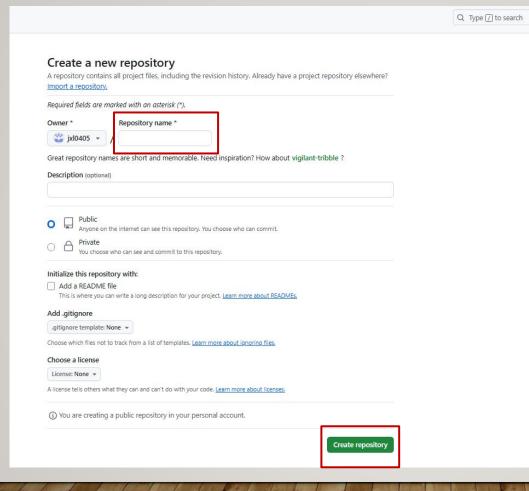
帳號中心



遠端REPO

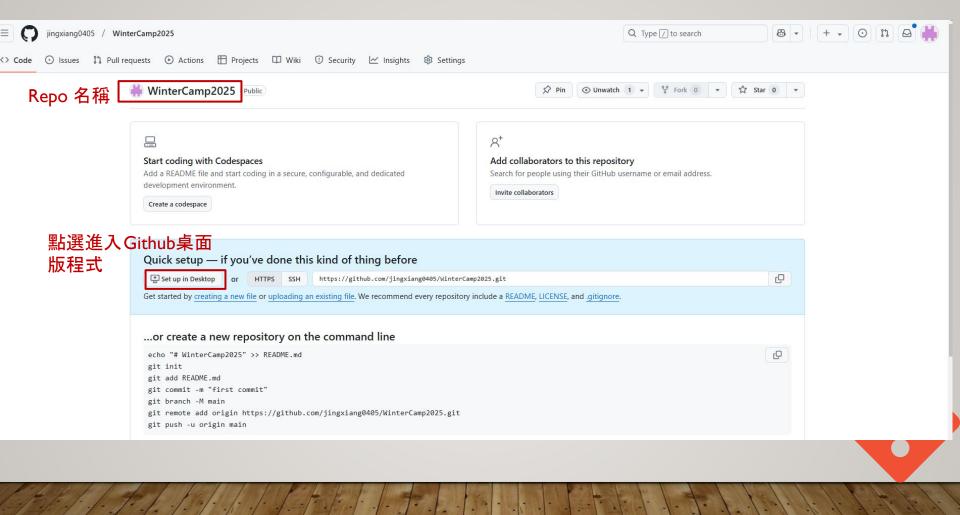




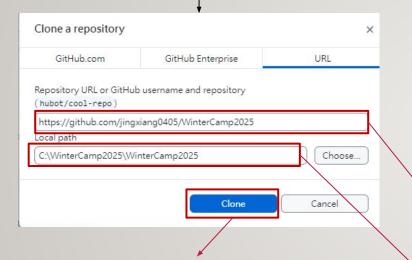




8 - | + - O n A 🖀



接著會直接打開GITHUB DESKTOP



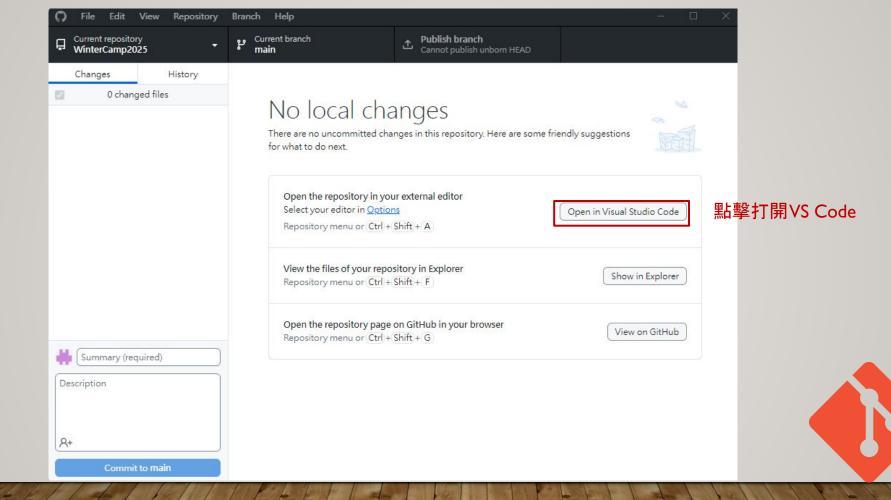
選擇完成後點「Clone」 意為「把遠端repo複製到本地repo」



(已經自動填入)Repo網址(URL): https://github.com/{擁有者帳號名稱}/{Repo名稱}

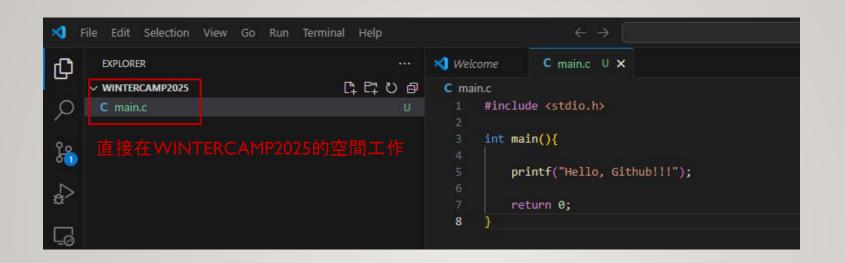
選擇剛剛創建的資料夾





CODING時間到!

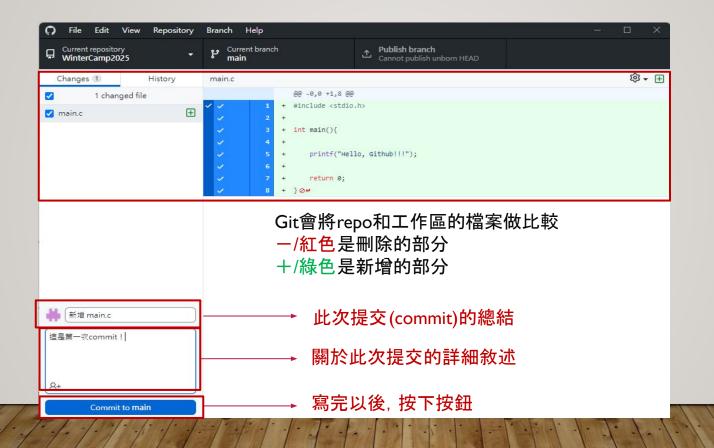




綠色的U代表Untracked(未追蹤), 意為本地repo並沒有此檔案的紀錄

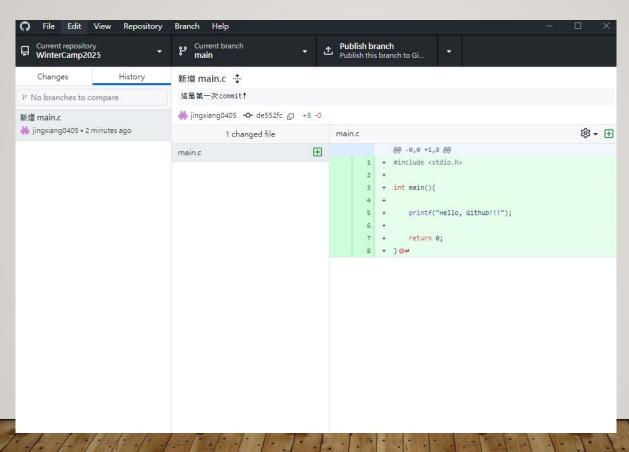


回到GITHUB DESKTOP, 你會發現...



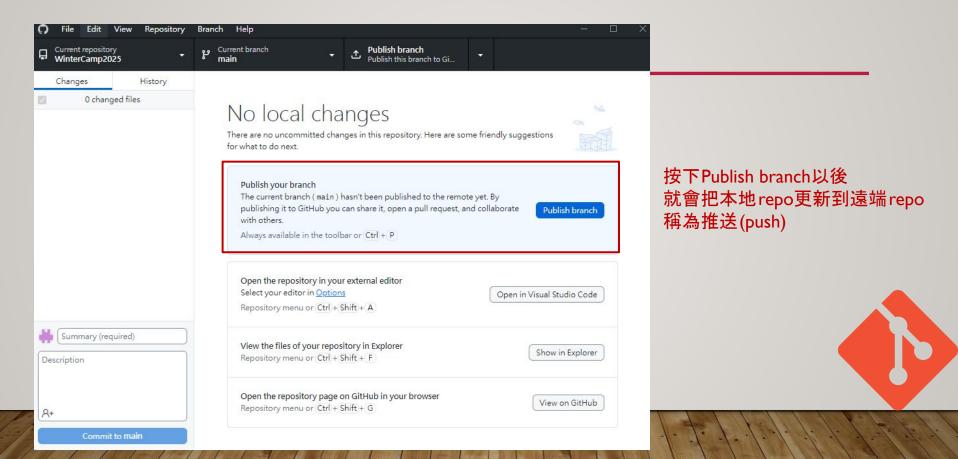


你可以在HISTORY檢視所有變更

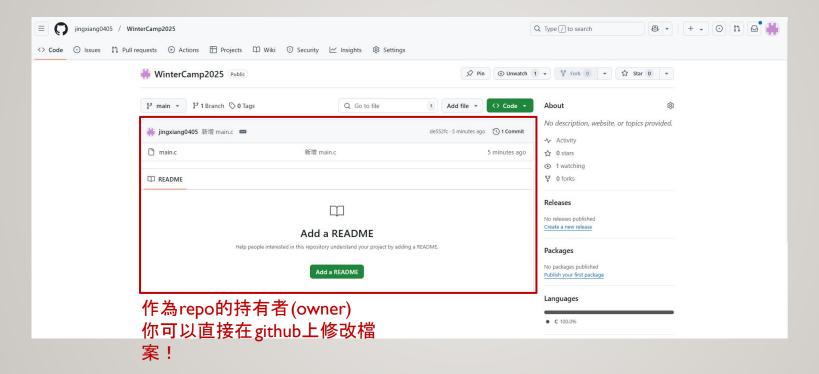




如果所有更改都已經提交(COMMIT), 你會看到...



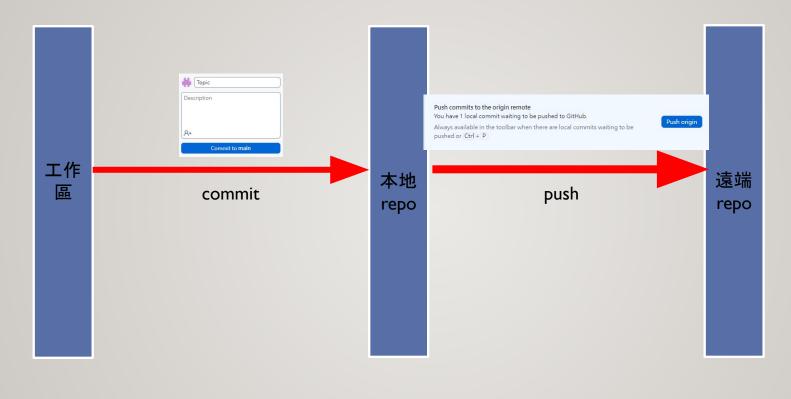
回到GITHUB, 重新整理一下網頁



完成!



剛剛做了什麼?





操作時間 (I5MIN)



同步

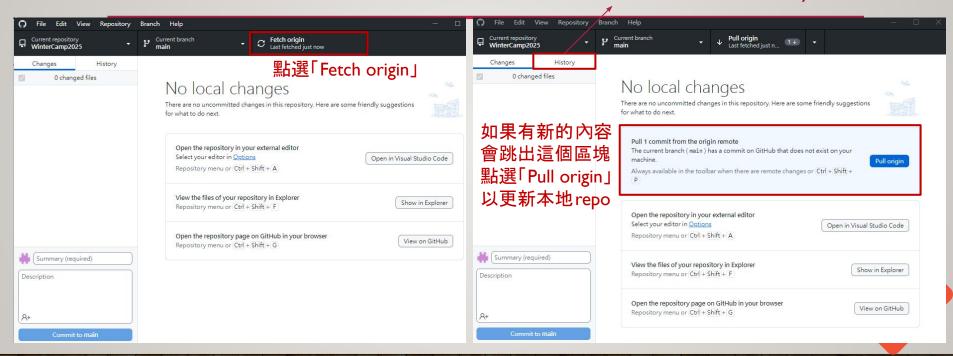
你不是一個人在寫 CODE!





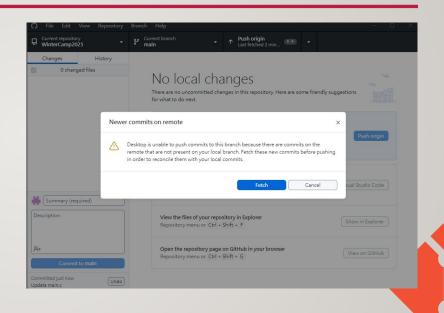
PULL: 讓本地REPO保持最新狀態

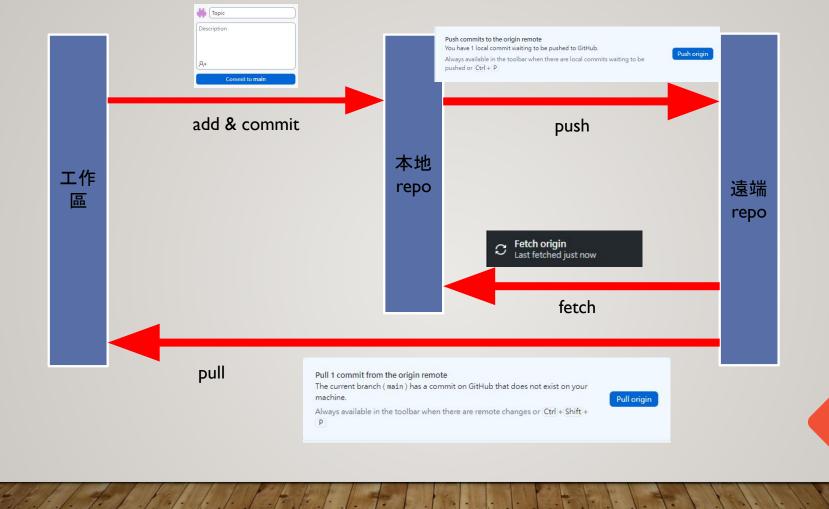
完成以後可以在History查看紀錄



GITHUB DESKTOP防呆機制

 如果想要推送(push)的版本和遠端 repo有衝突, Github Desktop會要 求你先Pull!



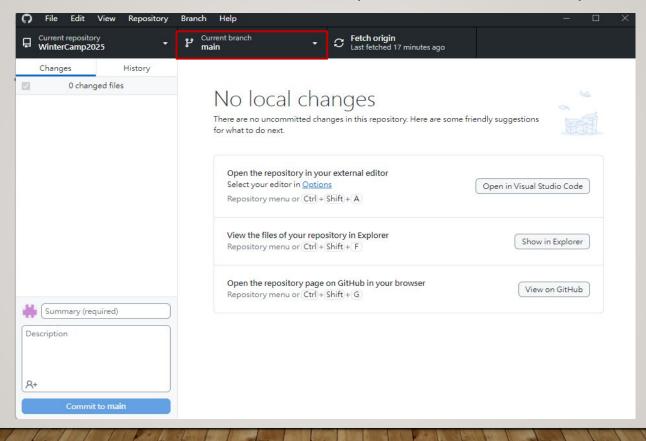


分支(BRANCH)

各司其職

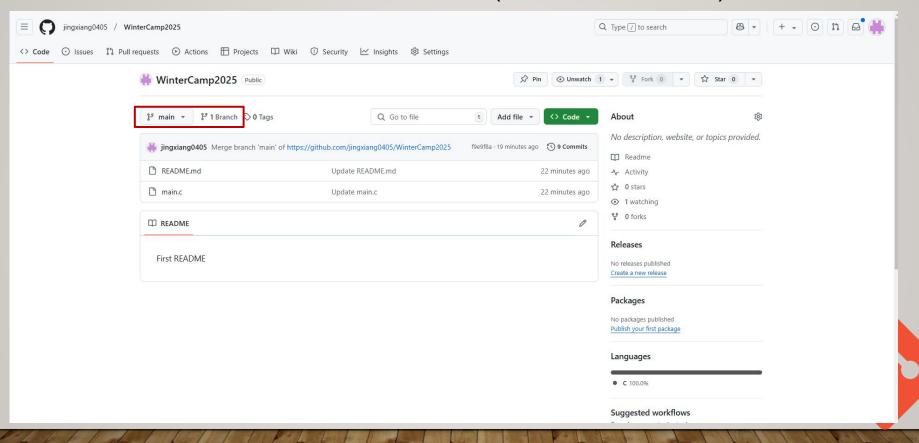


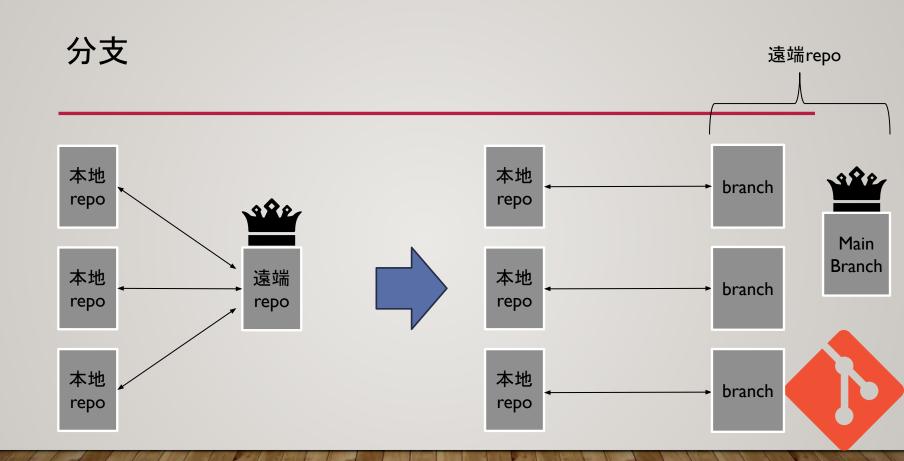
之前都是在唯一的主要分支(MAIN BRANCH)上工作

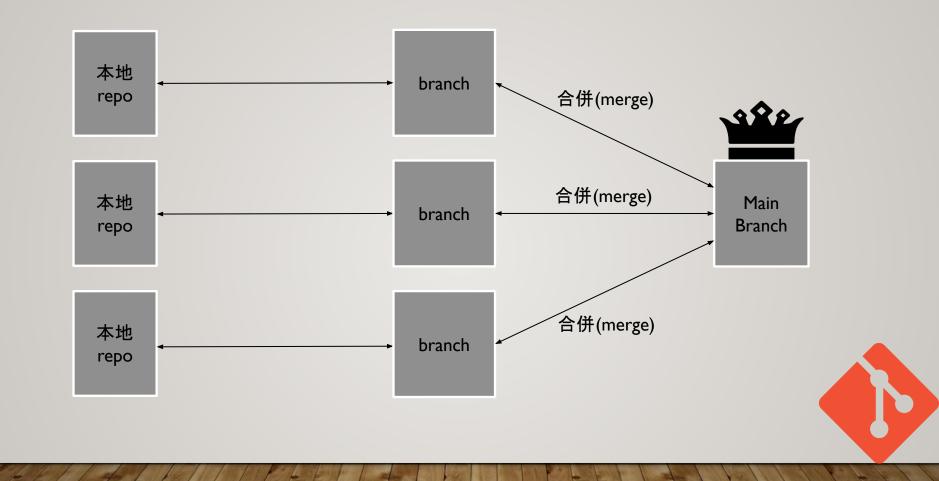




之前都是在唯一的主要分支(MAIN BRANCH)上工作



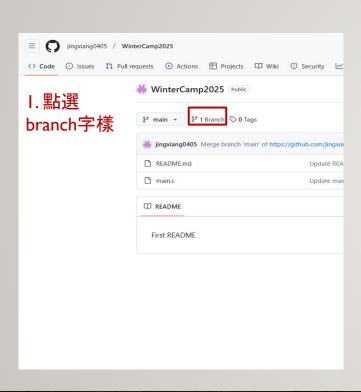


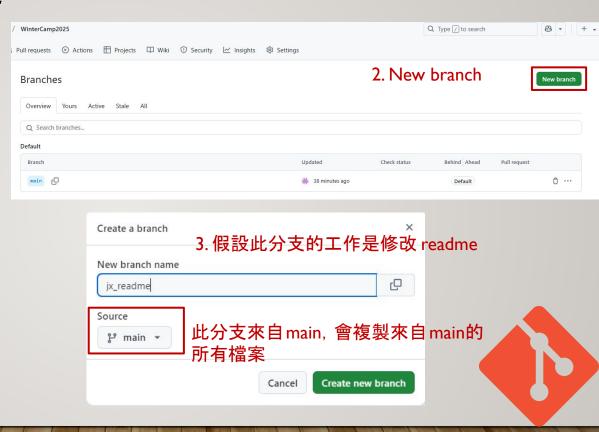


分支操作

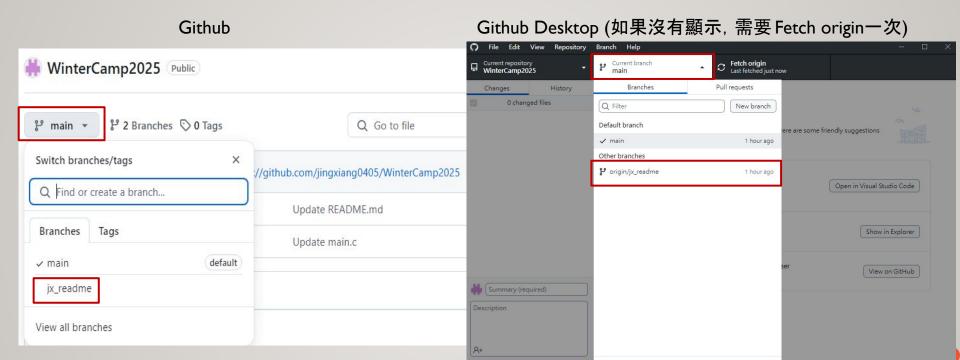


先在GITHUB上創建分支



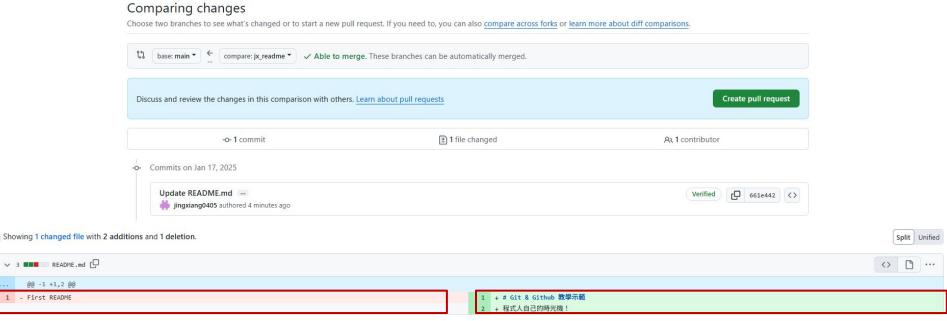


切換分支



Choose a branch to merge into main

任意修改

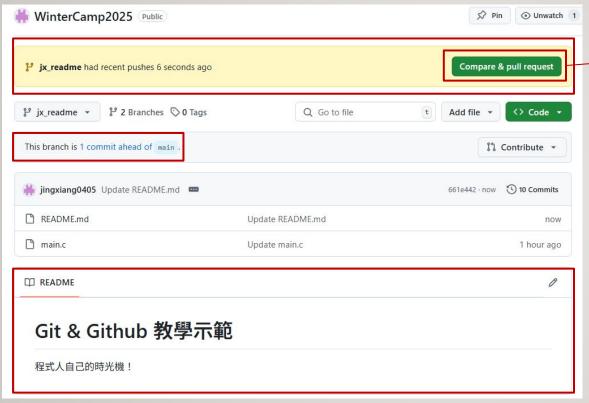


舊的內容

新的修改



提交一次改動後的REPO主頁



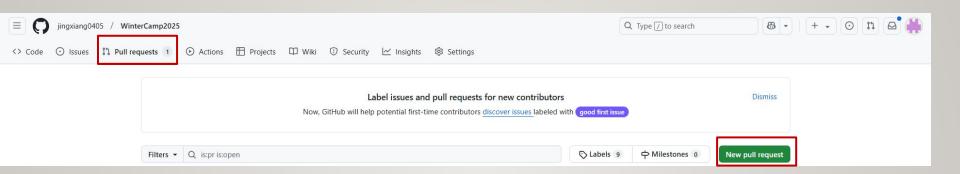
可以直接點這裡 建立Pull Request(PR)

PR: 將分支A合併至分支B

此時切換分支到 main 你會發現什麼都沒變!

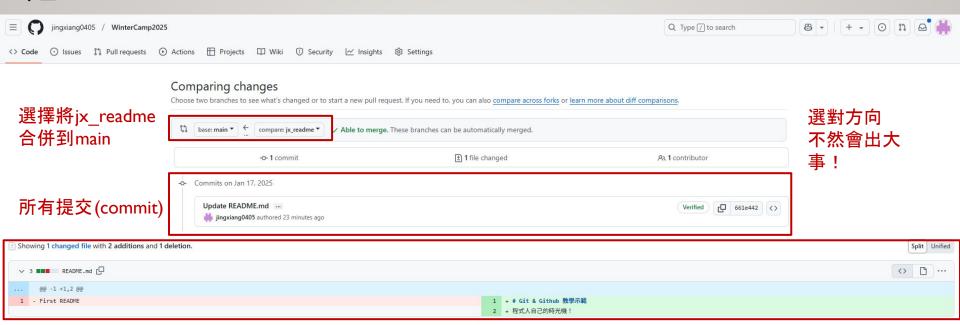


建立PR





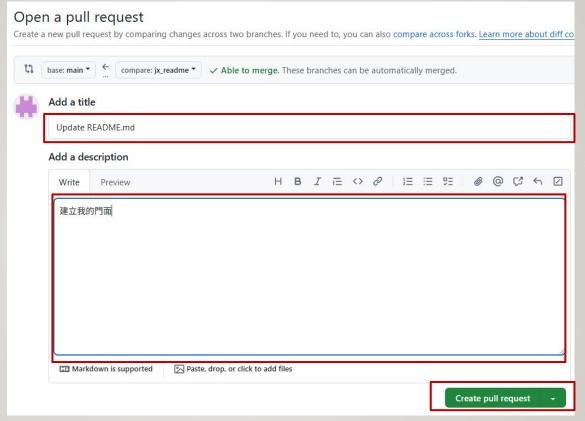
建立PR



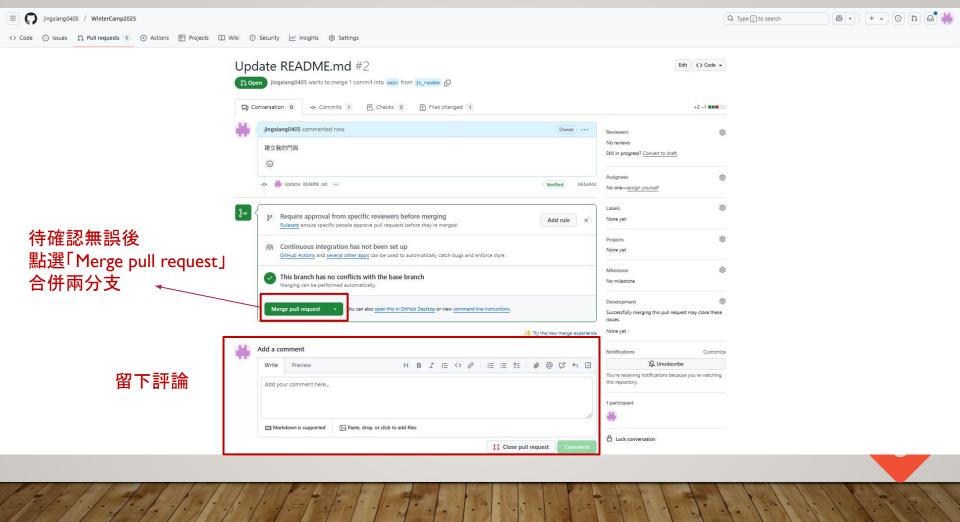
所有更改的檔案



PULL REQUEST







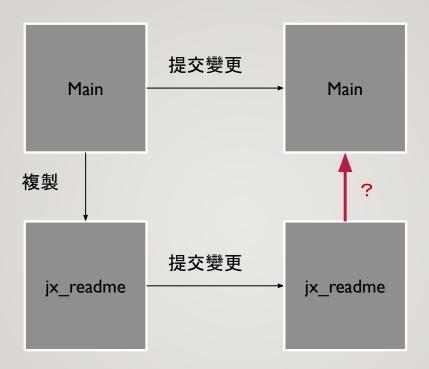
操作時間(IOMIN)



確認無誤?

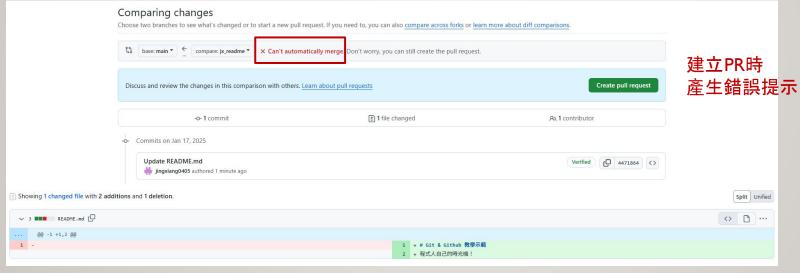
設想一個狀況...





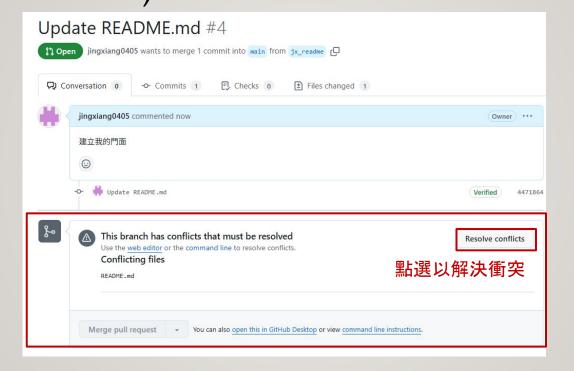








建立PR後... 產生衝突(CONFLICT)





Update README.md #4

Resolving conflicts between jx readme and main and committing changes \rightarrow jx readme

1 conflicting file README.md <<<<<< jx_readme README.md # Git & Github 教學示範 README.md 程式人自己的時光機! 3 4 偷偷地改一點東西 5 >>>>> main

Update README.md #4

Resolving conflicts between jx readme and main and committing changes > jx readme

1 conflicting file

README.md RFADMF.md

README.md

- # Git & Github 教學示範
- 程式人自己的時光機!

刪除其他部分 留下需要的區塊



一切就緒之後...

README.md

README.md



程式人自己的時光機!



