

# Git: 解鎖工作效率的神器 (Git 概念簡介 + 延伸應用介紹)

## Git ≠ GitHub

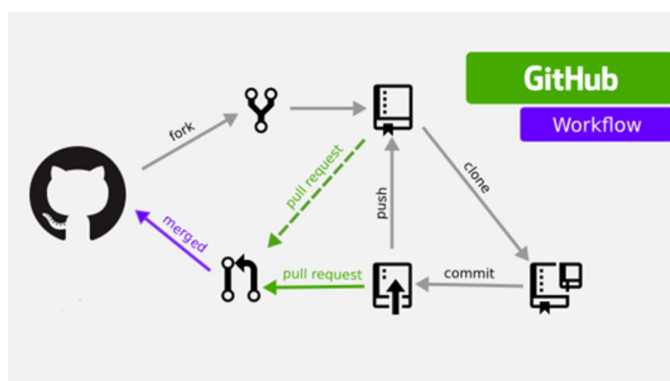
- Git：處理版本控制的軟體
  - 儲存程式、課本
  - 可只在本機端執行，不須開網站
- GitHub：將 Git 版本資訊儲放至雲端
  - 類似雲端硬碟、問題追蹤器
  - 附有 DevSecOps 的功能，可用於設計、實作程式
- 為何要使用Git / GitHub？
  - 版本控制
  - 容易傳遞資料
  - 人員合作容易
  - 自動化處理

## 版本控制

- 在Google文件上點擊版本紀錄，都是顯示「匿名使用者」
  - 無法得知是誰編輯文件 → Git有所有紀錄且不允許匿名
- 不怕電腦壞掉 → 可追溯各種紀錄
  - 可看到之前做的事情
  - 隨時有往後退一步的機會
  - 有結構的規畫完想完成的事

## 容易傳遞資料

- 過去在開發、管理時有太多資料，很難全部記住
- 上傳到雲端，需要雲端壓縮、本地解壓縮 → 耗時
- 在Git只需要上傳和下載



- 若需要變動，可先產生自己的版本（fork）
- Git clone：完全複製別人的內容，下載到本地端
- Git pull：下載別人更新的內容
- Git add：確定要更新的內容
- Git commit：標記更新內容
- Git push：上傳，送回雲端

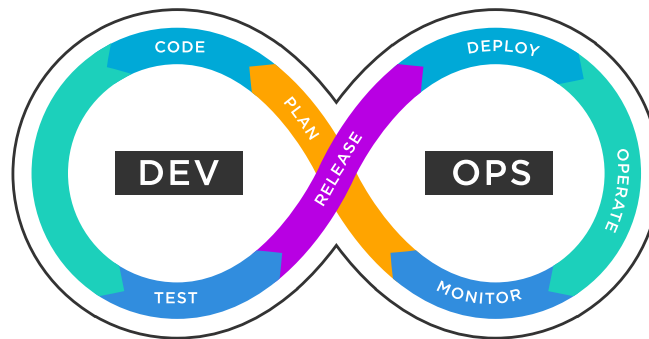
## 人員合作容易

- (情境題)甲等了很久，乙總算說：「我更新首頁囉！」
- 但甲不確定乙改了什麼
- 乙：「就檔案A呀！」
- 甲：「載下來跑怎麼有undefined？」
- 乙：「啊對...檔案B也有更動」
  - 溝通困難 → 只要Git pull即可更新
  - 可用Pull Request改別人的程式

- 若雲端會覆蓋到，則無法上傳
- 一個檔案兩個人同時變動→發生衝突→如何解衝突？

## 自動化處理

- 軟體工程的流程
  - Plan → Code → Build → Test → Release → Deploy → Operate → Monitor



- Git credential manager：和GitHub操作時，協助管理認證(帳號、密碼)
  - Assets：對於不同的電腦系統都做出實際的程式出來，測試程式是否正常
  - 由 GitHub Actions 協助
- [reactjs.org](https://reactjs.org/) 官網放在GitHub上面
  - 使用GitHub Action 和 Vercel完成，部屬到瀏覽器上，給一般人使用

## 分享：Git好用的學習資源

- 猴子也能懂得Git入門指南 [https://backlog.com/git-tutorial/tw/intro/intro1\\_1.html](https://backlog.com/git-tutorial/tw/intro/intro1_1.html)
- NewPipe - a free YouTube client <https://newpipe.net/>
  - 音樂撥放器，看Youtube影片、聽音樂，可以擋廣告

## Terminal

---

學習 Git 時，常常都是在終端機（Terminal）操作，而Terminal又可分為以下幾種

- iTerm2
- PowerShell
- CMD
- Mac Terminal

## Git

---

- 確認版本 (\$ 錢號+輸入的指令，後面直接輸入)

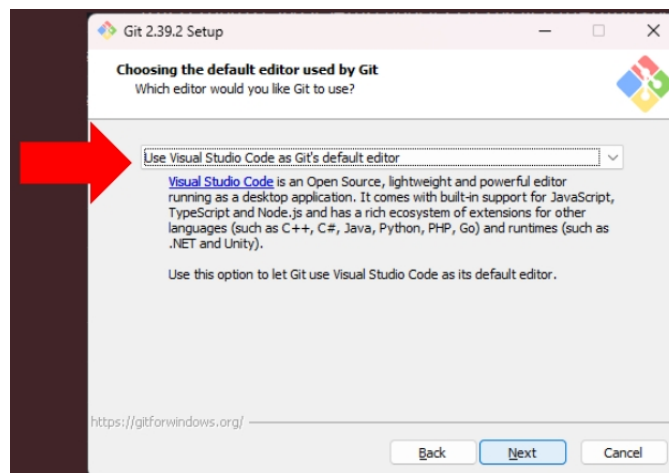
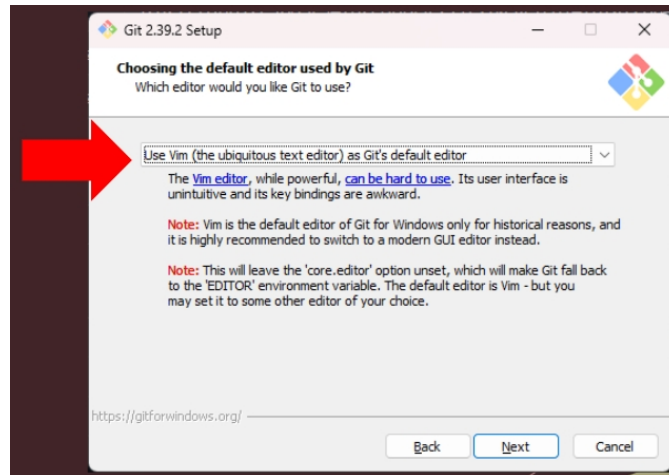
```
$ git --version
```

- 下載Git
- Windows
  - Google 搜尋「git windows」→ 官網提供的Setup
  - `winget install -e --id Git.Git`

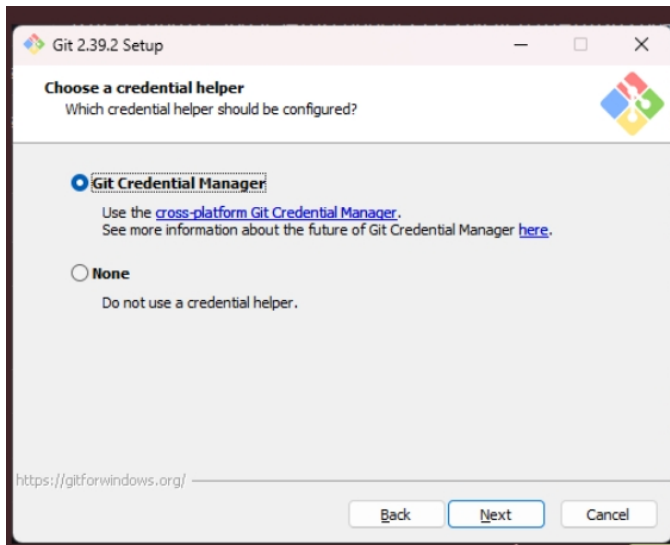
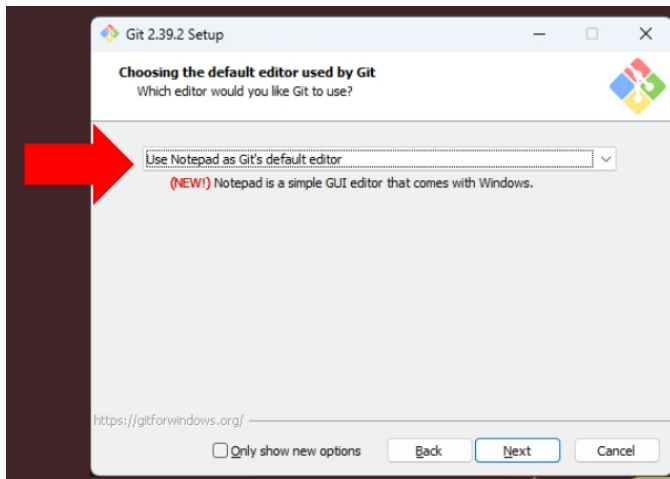
## Linux/WSL

- apt
  - Ubuntu、Debian
    - `apt install git`
- yum (視管理器)
  - CentOS
    - `yum install git`

- brew
  - `brew install git`
- 選擇Visual Studio Code 或是記事本（不建議新手用Vim，除非很熟悉）



- 選擇(預設的)Git Credential Manager



## Download GitHub Desktop

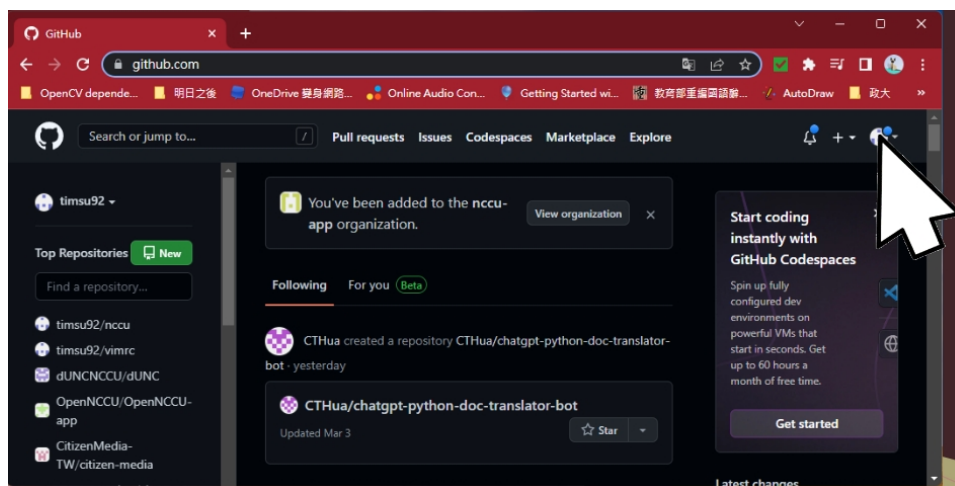
- 圖形使用者介面、不用輸入指令就可以完成80~90%的Git指令
- Google 搜尋第一個下載

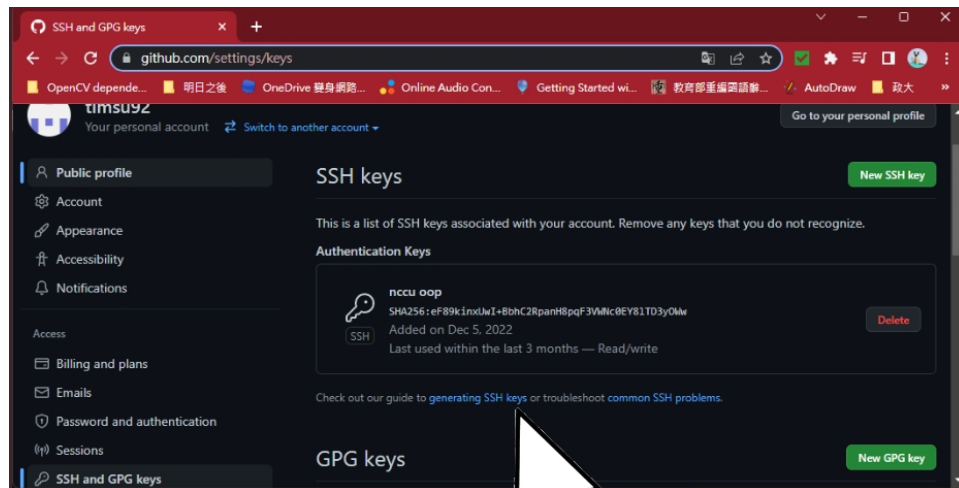
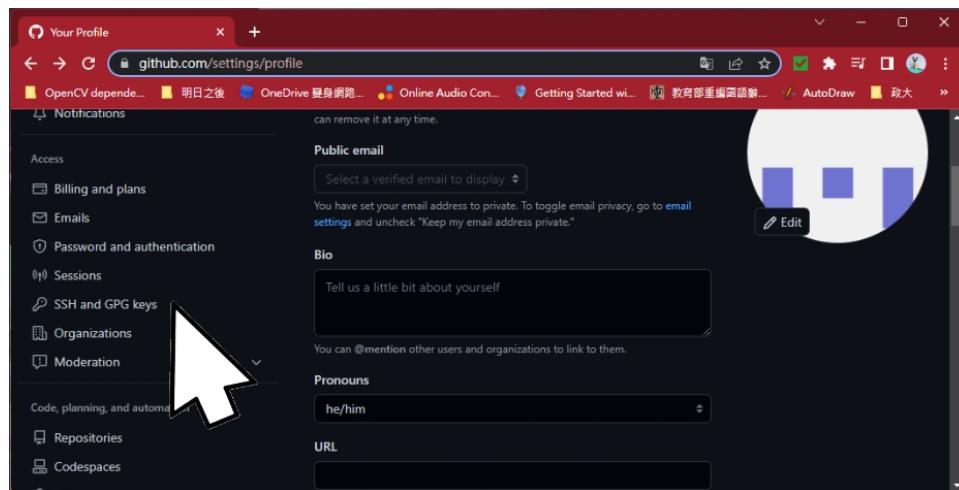
## 申請GitHub帳號(才可跟別人合作!)

- 到GitHub首頁註冊帳號 <https://github.com/>
- Q. 如何讓Git與GitHub連線?
  - A. 憑證建立，方法有兩種，如下
    1. Git credential manager <https://github.com/git-ecosystem/git-credential-manager>
      - 會跟Unity的Mono衝突
      - 進到GitHub頁面 → Install
      - 根據不同系統下載
      - WSL的欄位在windows底下
    2. 建立SSH key for GitHub (較常見，以下有詳細步驟)

## Generating SSH keys

(到GitHub，點擊頭像 > Settings > SSH and GPG keys > Generating SSH keys)







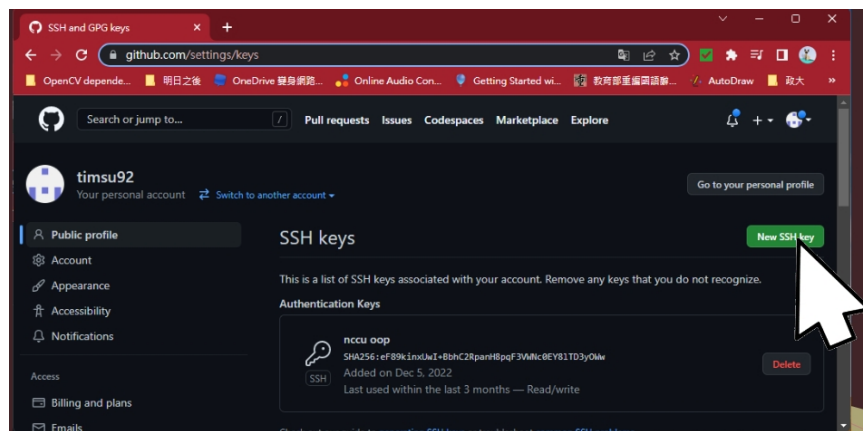
## 建立 ssh key 以連線GitHub

```
$ ssh-keygen -t ed25519 -C "你的電郵"
$ git config --global user.name "你想取的名稱"
$ git config --global user.email "你的電郵"
# 複製使用者家目錄/.ssh/id_ed25519.pub的內容
```

## Mac & Linux

```
ssh-keygen -t ed25519 -C "你的電郵"
eval "$(ssh-agent -s)"
> Agent pid 59566
$ sudo -s -H
$ exec ssh-agent bash
$ exec ssh-agent zsh
ssh-add ~/.ssh/id_ed25519
# 複製使用者家目錄/.ssh/id_ed25519.pub的內容
```

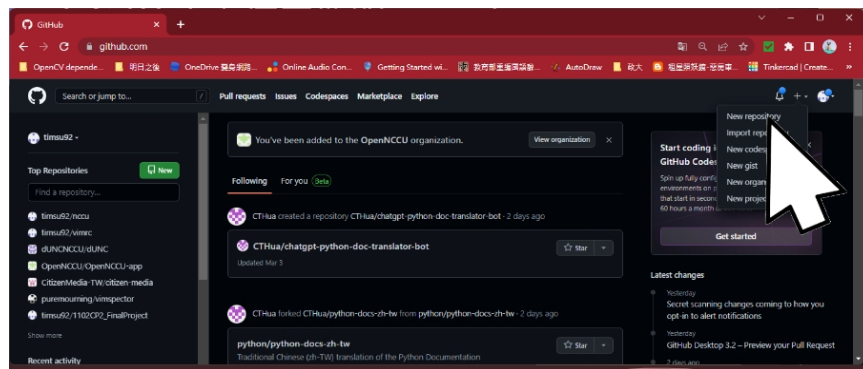
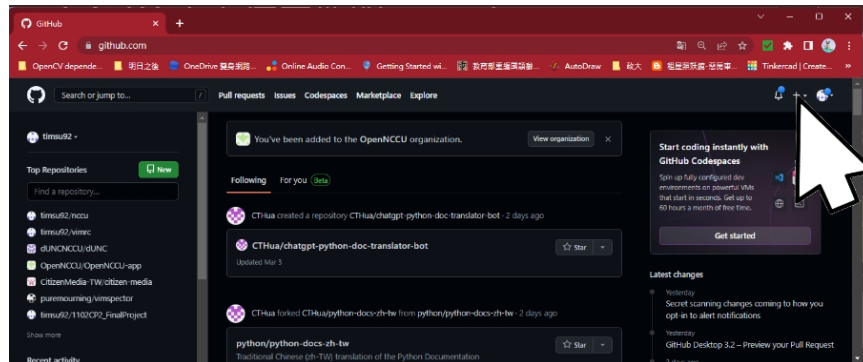
- 點開New SSH key



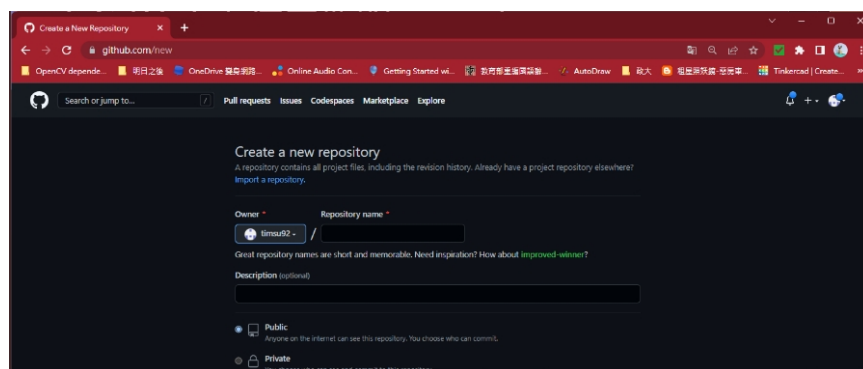
- Add new → 輸入Title(你可辨識的名稱) → 貼上剛剛複製的Key

## 在GitHub上建立新的repository

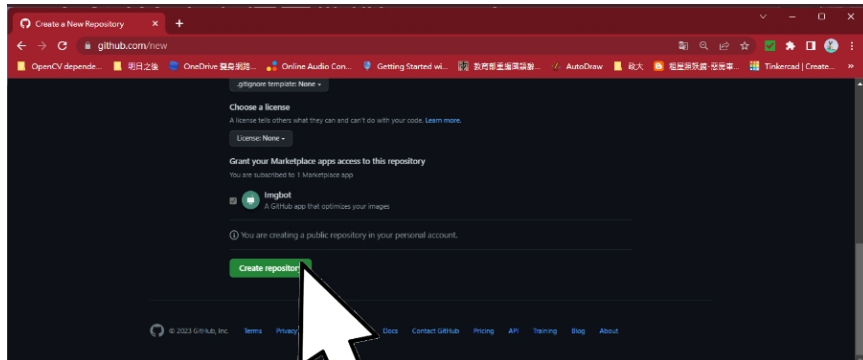
- 。 點擊右上角+號，點擊 New Repository



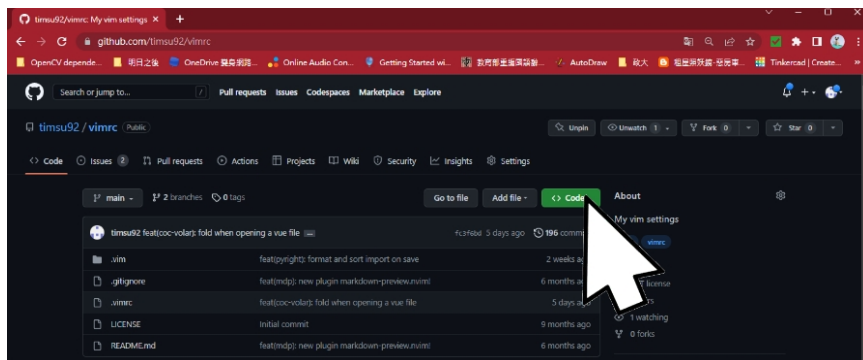
- 。 輸入名稱 (不可重複)



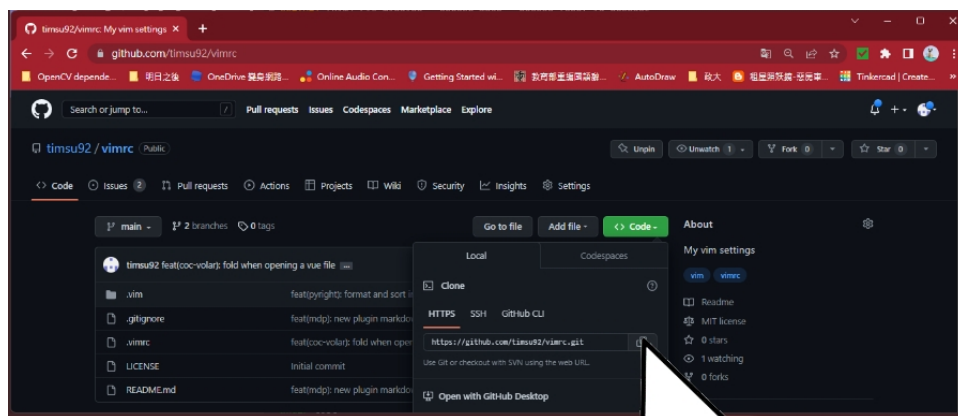
- 拉到底點擊Create repository



- 點開Code (綠色按鈕)



- 複製【Https】底下的網址 (此為本地抓資料的網址)



```
git clone 網址
cd <這個新repo的名稱>
# 在底下建立一個新檔案
git add <新檔案的檔名>
git commit -m '任意的文字'
git push
```

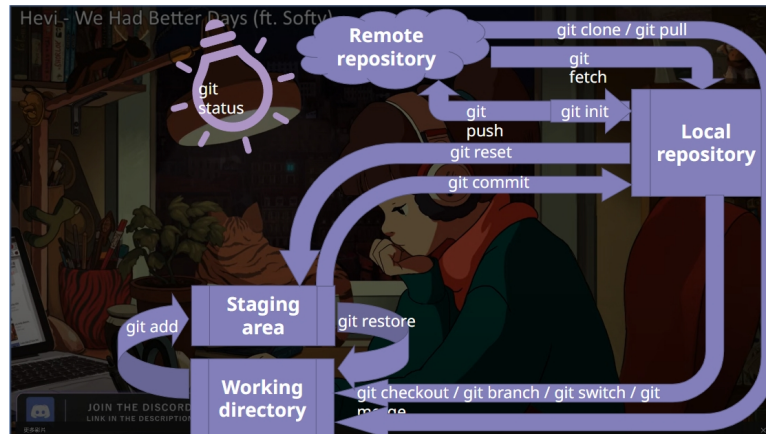
## Git基礎指令

---

- `git clone`
  - 將遠端的repository複製下來
  - 記得要cd到clone完畢的資料夾內
- `git pull`
  - 將遠端的更新同步到本地(下載)
- `git add`
  - 將手邊的工作內容複製到旁邊的書堆
- `git commit`
  - 紀錄一段工作內容
- `git push`
  - 將本地的更新同步到遠端

## GitHub使用流程

- 將資料與GitHub想像成書與書櫃！



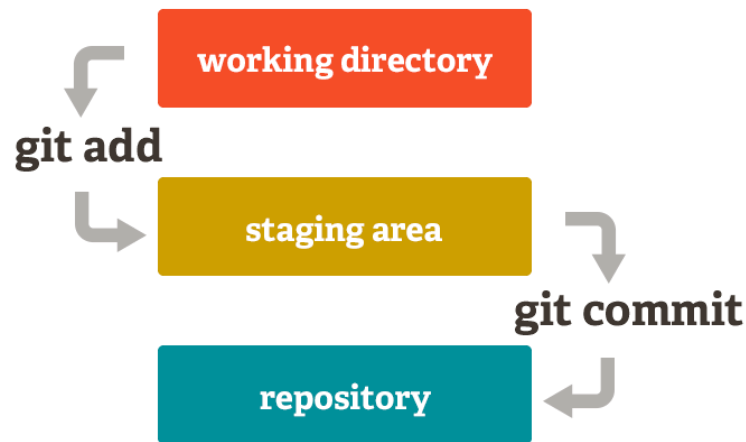
- (情境題)書櫃放了一些書 → 坐在書桌前想到什麼內容，就去書櫃拿相關的書放到桌上
- 寫書 & 改內容，覺得完成的差不多就暫時放旁邊，之後想到什麼再寫，或到書櫃上拿新書
- 最後把書放回書櫃
- Local Repository：儲存深層設定 (就是書櫃)
  - 存在 .git 裡面
  - 書櫃的由來：git init
- 租屋處的書櫃 → 家裡的另一個遠端的書櫃
  - 同步過來：git fetch，連結到家裡的書櫃
  - 需要一張桌子放書，或是用來工作：Working directory
- 從書櫃拿書到桌子：git checkout / git branch...
- 書桌東西寫一寫想先休息 → Staging area
  - 把書放回書櫃：git add
  - 放錯位置了：使用 git restore
- 覺得寫夠了做紀錄 → git commit

- 做錯紀錄了 → git reset
- 做完許多事情，放回書櫃 → git push
- 獨立出來的檯燈：git status（燈打下去，可以一目瞭然看有什麼東西）

## 精簡的Git(Hub) 使用流程

---

- 只需會git add 和 git commit就可以了
- 只適用於最簡單的情況



## 示範

---

- 打開資料夾
- 新增一個 diary.txt
- `git add diary.txt`
  - 複製檔案
- `git commit -m '0308diary'`
  - 紀錄檔案
- `git push`
  - 更新同步檔案

