

# 社課紀錄 03/08

## 18分鐘帶你淺嚐Git與GitHub

講者：蘇胤翔

▼ ppt

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/84d51b7c-3786-42b5-9f1d-680e44a975ed/20230308%E7%A4%BE%E8%AA%B2%EF%BC%9AGit.pptx.pdf>

18分鐘帶你淺嚐Git與GitHub

07:15開始

介紹社課規劃

Git和GitHub是不一樣的東西

版本控制

容易傳遞資料

合作容易

自動化

好用資源

07:32 結束

07:39 實作開始

Terminal

Git

申請GitHub帳號

**Generating SSH keys**

08:05 停下來等待大家實作完成

8:24結束實作等待

在GitHub上建立新的repository

8:28 等待現場實作

8:34 講解GitHub

Git(Hub) 使用流程

Git基礎指令

示範

**P** 2023 <https://docs.google.com/presentation/d/1TUKTpGETx7RuCK8mcL6KYQLDSxPiC5Dn/edit?usp=drivesdk&oid=108462176772250182455&rtpof=true&sd=true>  
0308  
社  
課：G  
it.pptx

## 07:15開始

### 介紹社課規劃

- 體驗社課僅能體驗一堂，之後須繳費
- 前18分鐘類似上學期LightTalk，後面時間為工作坊

### Git和GitHub是不一樣的東西

- Git：處理版本控制
  - 儲存程式、課本
  - 可只限本機端執行
- GitHub：Git加上雲端內容
  - 類似雲端硬碟、問題追蹤器
  - 附有 DevSecOps 的功能
- 使用Git / GitHub原因
  - 版本控制
  - 傳遞資料方便
  - 人員合作容易
  - 資料處理

### 版本控制

- 共編按「查看版本紀錄」— 但都是「匿名使用者」
  - 不知道是誰編輯的
  - 有顯示各自的修改內容，但顏色都一樣

- 幫你提神的不是咖啡，是打翻並弄壞筆電的咖啡
  - 有版本控制，就不必怕電腦壞掉
    - 可隨時看之前做的事情
    - 隨時有往後退一步的機會
    - 有結構的規畫完成你想完成的事
    - 可追殺做錯事的人

## 容易傳遞資料

- 難以記得更動內容
  - 檔案太多太大
- 上傳到雲端，需要等雲端壓縮以及本地解壓縮

## 圖例

- 若變動，可先產生自己的版本（fork）
- 載到本地端（pull）
- 複製一份（clone）
- 留下註記（commit）
- 送回網站（push）

## 合作容易

- A等很久，B總算說：「我更新首頁了！」
- 但A不確定B改了什麼
- B就說：「就檔案A啊！」
- A：「怎麼有undefined？」
- B後來才說：「啊...檔案B也有改」
  - 只要下Git pull即可更新
  - 若雲端會覆蓋到，則無法上傳
  - 可用Pull Request改別人的程式

## 自動化

- 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
  - Assets：對於不同的電腦系統都做出實際的程式出來
  - 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

## 好用資源

- 猴子也能懂得Git入門指南
- NewPipe
  - 音樂撥放器，可看yt影片，可擋廣告
  - 也是在GitHub上面

## 07:32 結束

---

## 07:39 實作開始

---

- 環境設定
  - Git & GitHub
  - 憑證
- 同步檔案庫
  - \$git clone
  - \$git pull
  - \$git push
- Git基礎指令
  - 紀錄的建立、還原
  - 狀況查看
  - 分支

- Git branch
  - 新建、切換、刪除branch
  - 合併branch

## Terminal

---

- iTerm2
- PowerShell
- CMD
- Mac Terminal

## Git

---

- 確認版本

```
git --version
```

- 下載Git
- Windows
  - 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
  - CentOS
    - `yum install git`
- brew
  - `brew install git`

- （不建議新手用vim）

## Download GitHub Desktop

- 圖形使用者介面、不用輸入指令

## 申請GitHub帳號

- 到GitHub首頁 註冊帳號
- 讓Git與GitHub連線

### 憑證建立方法

1. 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 ← 這要放哪?? 蛤?不能放大也 好奇怪！（嘆氣）
2. ssh key for GitHub

## Generating SSH keys

### 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
```

建立 `ssh key` 以連線GitHub (PPT 36頁)

```
$ 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
```

## 08:05 停下來等待大家實作完成

### 雅婷逐字稿

透過雅婷的AI語音辨識技術，無論會議記錄、訪談、課程都能被輕鬆紀錄、快速搜尋

 <https://asr.yating.tw/>



## 8:24結束實作等待

- 點開New ssh key
- Add new → 輸入title → 貼上剛剛的Key

## 在GitHub上建立新的repository

- 右上角點開 → New Repository
- 輸入名稱 → 捲到底按Create repository
- 點開Code → 複製【Https】底下的網址

```
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
```

## 8:28 等待現場實作

## 8:34 講解GitHub

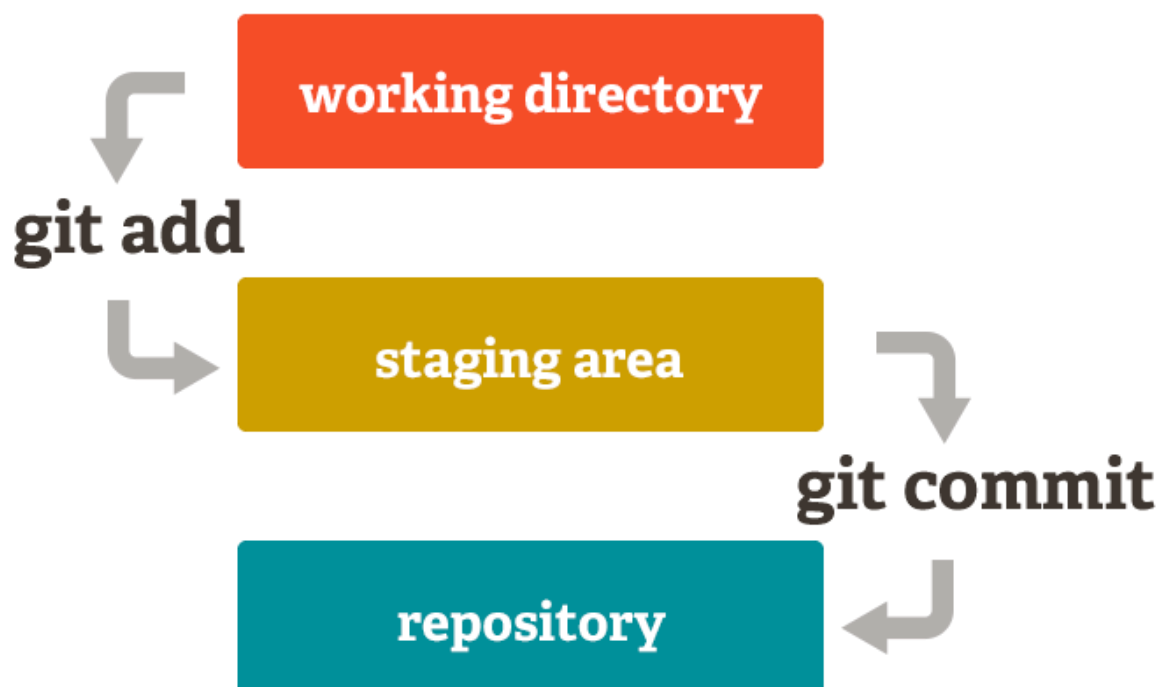
- 上完助教課後，寫作業聽YouTube → Lofi → 感覺他做的事跟GitHub操作很像
  - 書櫃放了那些書 → 想到什麼內容就把書放到桌上
  - 寫書 & 改內容，覺得差不多就放旁邊 → 想到什麼再寫 or 拿新書
    - 最後再把書放回書櫃
- Local Repository：儲存深層設定

- 存在 .git 裡面
- 書櫃的由來：git init
- 租屋 → 家裡有另一個遠端的書櫃
  - 同步過來：git fetch，抓到我們家
  - 需要一張桌子放書、工作：working directory
- 書櫃拿到桌子的方法：git checkout / git branch...
- 書桌東西寫一寫想先休息 → staging area
  - 放書：git add
  - 放錯位置：需使用 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
- 檯燈：git status（燈打下去，看有什麼東西）

## Git(Hub) 使用流程

---

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





## 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

## 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

## 8:48 結束

PPT到59頁