課程目標

理解 git 版本控制的基本觀念

熟悉 git 基本版控指令

熟悉 github 服務

熟悉 github flow 軟體開發流程

熟悉 markdown 語言語法

心理建設

git 是工具

免費的工具最貴

不要過度依賴圖形化介面工具

不要害怕使用終端機打指令

你經常在寫程式時遇到...

檔案同時被別⼈或⾃⼰覆蓋，為了避免...

file1.rb

file1\_1.rb

file1\_2.rb

你經常在寫程式時遇到...

需求修改，想復原前幾天寫的版本...

file\_20160911.rb

file\_20160920.rb

file\_20160921.rb

你經常在寫程式時遇到...

需要知道某一段程式碼是誰改的...

bad\_code.rb

你經常在寫程式時遇到...

軟體發⾏，需要分成維護版跟開發版

released.rb

development.rb

什麼是版本控制？

可以隨時復原修改，回到之前的版本

多⼈協作時，不會把別⼈的東⻄蓋掉

保留修改歷史記錄，以供查詢

軟體發⾏時，可以⽅便管理不同版本

What is git?

一個開放源碼的版本管理系統(VCS)

發明⼈ Linus Torvalds (Linux 的作者)

原本的是為了管理 Linux Kernel 原始碼

What is github?

一個基於 git 開發出來的服務

讓你能透過 git 在遠端備份程式碼

一般的使用免費但必須公開程式碼

廣受大量開源專案的喜愛

是程式設計師的寶庫 (可以在裡面查詢各式各樣的程式與實作的程式碼)

Register for github

[www.github.com](http://www.github.com)

請到以下連結註冊(或登入)github

Install git (Windows)

git for windows 連結

1. 打開 git 的彩色輸出：

git config --global color.ui true

2. 設定 git 的 user 和 email （用註冊 github 的資訊）

git config --global user.name "你的名稱"

git config --global user.email "你的email"

2. 接下來打開 git bash：

1. 下載並安裝 git for windows:

Repository (儲存庫)

我們一旦要對一個專案裡的檔案進行版本控管，我們一定需要創造一個儲存庫來記錄這些版本的訊息

假定說現在我想對 /lesson1 這個路徑底下的檔案做版本控制

mkdir lesson1

cd lesson1

建立儲存庫的方式有很多種，最簡單的方式就是切換到想要控管的程式碼路徑 (資料夾) 下，輸入：

確認你現在已經在 /lesson1 這個路徑底下，然後輸入：

git init

Initialized empty Git repository in /lesson1

Working Directory (工作目錄)

接下來這個路徑 (資料夾) 就成為我們的工作目錄 (Working Directory)，之後無論是新增、修改、刪除檔案，或是下任何 git 指令，都是在該工作目錄底下進行

一旦你下了 git init 後看到這樣的訊息：

Initialized empty Git repository in /lesson1

代表以後在 /lesson1 資料夾底下的檔案就會被列入版本控管

git 的指令也只能在 /lesson1 底下才能運作

markdown 語法

markdown 是一個用於產生顯示於網頁內容的寫作用語法

github 會自動把 markdown 語法轉成我們看得懂的網頁內容

簡單易學

markdown 教學1

https://markdown.tw/

markdown 教學2

<https://wastemobile.gitbooks.io/gitbook-chinese/content/format/markdown.html>

Visual Studio Code

推薦使用的編輯器，是一個由微軟起頭、開源的程式碼編輯器，功能強大，並且內建 markdown 文件的預覽功能

新增第一筆檔案

現在我們來新增一個檔案，lesson1.md，並用 git 對該檔案進行版本控制

我在 /lesson1 這個路徑底下建立一個叫 lesson1.md 的檔案

touch lesson1.md

code .

接下來可以用任何你喜歡的編輯器打開 lesson1.md

練習一

# NTU GIT CLASS 練習一

姓名

----

日期

----

yyyy/mm/dd

Github Username

---------------

我最喜歡的三件事

---------------

請依照以下格式，在 lesson1.md 填入你的資料：

練習一

git add lesson1.md

填完資料後，接下來就把 lesson1.md 目前的進度納入版本控制吧，先用 git add 指令把 lesson1.md 放入一個暫存區：

git add lesson1.md

git status

接下來執行 git commit -m "一段你對這個版本的備註"，這時放在站存區裡的進度就會被儲存成一個版本

git commit -m "added lesson1.md"

git status

接下來用 git log 檢查一下，確認我們的 commit 有成功：

git log

在 github 建立一個遠端儲存庫

今天你寫好的程式碼，你當然不希望只放在本機端，因為本機端要是發生任何天災人禍，你的程式碼就消失了，所以遠端備份是比較保險的作法

在 github 建立一個遠端儲存庫

先登入 github，然後在首頁按下 new repository 的按鈕，或直接連至 這個頁面

在 github 建立一個遠端儲存庫

在這裡輸入你的儲存庫名稱，記得要勾選 public 選項，代表你放上去的程式碼是會被公開出來的 (private 是要錢的！！！)

在 github 建立一個遠端儲存庫

現在你已建立起一個「沒有版本」的空白 git 儲存庫

Github 也很佛心的幫你產生上傳本機端 git 儲存庫的指令教學

在 github 建立一個遠端儲存庫

接下來我們就準備把方才 commit 完的 lesson1.md 放上 github 的儲存庫吧！

git push origin master

git push 如同字義，可以想像成把本機端的版本 "推" 到雲端上 github 的儲存庫內

成功後請回到 github 的頁面並且重新載入，檢視一下 lesson1.md，你的程式碼是否以很清楚的方式呈現出來了呢？