

Terminal 介紹

絕對路徑 vs. 相對路徑

- 相對路徑：相對於現在目錄的路徑表示法
 - 會隨著現在目錄的不同而改變
- 絕對路徑：不是相對路徑的，都屬於絕對路徑。
 - 一個絕對的位置，並不會隨著現在目錄的改變而改變

絕對路徑 vs. 相對路徑

- 「.」：當前目錄
- 「..」：前一層目錄
- 「~」：home目錄
- 「/」：根目錄

絕對路徑 vs. 相對路徑

- 相對路徑：相對於現在目錄的路徑表示法
 - 會隨著現在目錄的不同而改變

```
~/test1 ➤ ls  
test2 test3
```

```
~/test1/test2 ➤ cd ../test3  
~/test1/test3
```

- 絕對路徑：不是相對路徑的，都屬於絕對路徑。
 - 一個絕對的位置，並不會隨著現在目錄的改變而改變

```
~/test1/test2 ➤ pwd  
/Users/hackerifyouwant/test1/test2
```

課堂練習 絕對 or 相對

- /Users/hackerifyouwant/test1/test3
- ../test2
- ~/test1/test3

常用指令

Windows	MacOS / Linux	說明
cd	cd	切換目錄
cd	pwd	取得目前所在的位置
dir	ls	列出目前的檔案列表
mkdir	mkdir	建立新的目錄
無	touch	建立檔案
copy	cp	複製檔案
move	mv	移動檔案
del	rm	刪除檔案
cls	clear	清除畫面上的內容

課堂練習

- 新增一個 cpp檔案 test (code 檔名.副檔名)
- 新增一個 cpp檔案 test2
- 建立一個資料夾 test_file
- 進入資料夾並查看資料夾內有什麼檔案？
- 資料夾的位置？
- 複製 test 到 test_file 裡面 (用相對路徑)
- 刪除 test
- 移動 test2 到 test_file 裡面 (用絕對路徑)

ls -al

- ls 指令可列出在目前目錄所有的檔案及目錄，後面接的 -al 參數，a 是指連小數點開頭的檔案（例如.gitignore）也會顯示，l 則是完整檔案的權限、擁有者以及建立、修改時間
- 「.» 開頭的檔案，正常狀況是看不到，是隱藏起來得

萬用字元

- 萬用字元：一種特殊字元，可代表文字值中的未知字元，很適合尋找具有類似但並非相同資料的多個專案。萬用字元也可以協助根據指定的模式比對來取得資料。

字元	描述	範例
*	比對任何數目的字元。星號 (*) 可以用於字元字串中的任何位置。	wh* 會找出 what、white 和 why，但是不會找出 awhile 或 watch。
?	比對特定位置的單一字母。	b?ll 會找到 ball、bell 和 bill。
[]	比對括弧內的字元。	b[ae]ll 會找出 ball 和 bell，但不會找到 bill。
!	排除括弧內的字元。	b[!ae]ll 會找出 bill 和 bull，但是不會找出 ball 或 bell。 Like "[!a]*" 會尋找所有不是以字母 a 開頭的專案。
-	符合字元範圍。請記得以遞增順序指定字元 (A 到 Z，而不是 Z 到 A)。	b[a-c]d 會找出 bad、bbd 和 bcd。
#	比對任何單一數字字元。	1#3 會找出 103、113 和 123。

課堂練習

- 進到 test_file
- 新增 3個 cpp 檔「file1,file2, file3」
- 新增隱藏檔案 git
- 用一行指令，刪除所有的 cpp 檔
- 看資料夾中所有的檔案及權限

回家作業

- 以後盡量用終端機做事情

補充資料

- <https://zhung.com.tw/article/mac-terminal-often-used-commands/>
- <https://gitbook.tw/chapters/command-line/command-line.html>

Github

GitHub ?

- GitHub 是一個「透過 Git 進行版本控制」的「原始碼代管服務平台」
- 簡言之，就是各種開源軟體(專案)的聚集地，只要專案作者公開這個專案，任何人都能存取、使用、甚至對專案的內容做出修改，而專案作者能對這些改動用 Git (等等會介紹的版本控制工具)來進行控管，進而達成一種「共同創作」的理念。

版本控制？

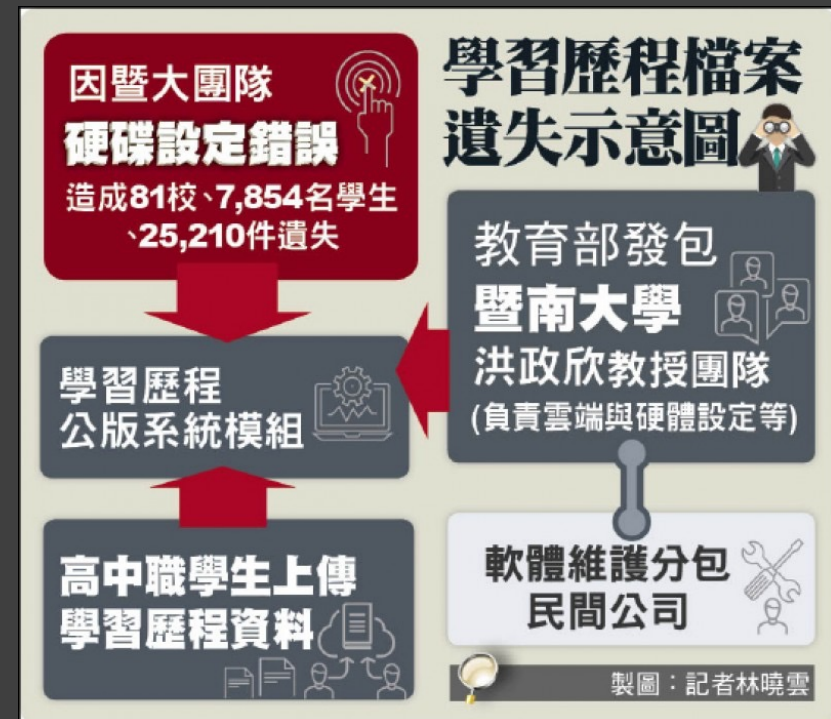
- 為了防止資料不見，會怎麼做？
 - 備份
 - 這就是最原始的版控系統 VCS (Version Control System)
 - 缺點：只在本地端（自己的電腦上）備份並控管，只要本地設備一出狀況，就會血本無歸。

版本控制？

- 公司裡儲存大型數據的資料庫，或是開發部門的專案，也需要備份。對於公司而言，維持這些資料的完整可說是至關重要，千萬不能有「一跳電資料就全部消失還還原不了」這類情況發生。

版本控制？

- 公司裡儲存大型數據的資料庫，或是開發部門的專案，也需要備份。對於公司而言，維持這些資料的完整可說是至關重要，千萬不能有「一跳電資料就全部消失還還原不了」這類情況發生。



版本控制？

- 共用

- 在一間 100 人的小公司裡，一次只有一台電腦能連上資料庫，如果遇到問題需要更改，就得排隊排好長一段時間，等到排到時可能問題已經擴散到其他檔案了...
- 為了方便多人共用，便出現了**備份 x 共用的版控，CVCs (Centralized Version Control System)**，比起 VCS，可以接受多人共同使用，但還是繼承了上一代的缺點，**只要本地設備一出狀況，就會血本無歸。**

Git 與 版本控制？

- 解決了「單一儲存區」這項最致命的缺點，改為「分散式儲存」，DVCs (**Distributed Version Control System**) 將資料分散於不同設備上(就是儲存資料的本地端)，如此一來，就算其中一處設備損壞，也不影響其他用戶使用。

Git 與 版本控制？

- GitHub 便是採用 Git 的理念，除了專案作者本人的電腦上，也在 GitHub 官網上儲存一份備份，而且每個曾進行過更動的本地端(就是改過檔案的人的電腦)也能重新放回官網上，形成新的備份。

實際案例

- Git 的開發者 林納斯·托瓦茲 (Linus Torvalds ，當今最著名的電腦工程師之一) ，當初並不是為了開發版本控制系統而開發 Git 的，而是為了方便他現有的開源專案——Linux 作業系統的合作開發所做的工具。
- 由於作業系統的程式碼隨隨便便都是百萬行起跳，因此，良好的版控工具變得至關重要，除了在錯誤出現時，能有效的還原回正確的版本，還要能將各種程式碼反覆在各裝置上測試與轉移。但是 Linus 對當時的版控工具抱有疑慮，而當時的版控工具公司又拒絕與 Linus 團隊合作改進版控工具，無耐只好自己做一個了，Git 才終於問世。

不是工程師也能用的 GitHub

- GitHub 是公開的線上平台，因此很適合拿來放一些作品集，或著是共同編輯的文案等，有著類似端硬碟的功能(而且還是免費的，只要不介意會公開的話)。而且每當有人對文案進行改動，GitHub 還會記下改動時間、編輯者等資訊，又能在更動時檢查後再與原版合併，因此很方便團隊進行共同編輯。
- 就有日輕小說的漢化組(翻譯網路小說的團體)在 GitHub 上進行作業，除了共同翻譯以外，翻譯後的校對又能讓所有人看過再進行更動，有效提昇了作業效率。

我的解讀

- **git**就是版本控制系統，基本上就想像成幫你的備份資料做整理的一個系統
- 你如果某個版本的**code**寫到爛掉，就可以用**git**退到比較舊的版本（備份的概念），然後從那個舊版本開始寫
- 而**github**可以當作有整理備份資料功能的**google drive**

Git 指令

常用指令

init / clone

Git	zsh	do
<code>git clone</code>		抓遠端儲存庫下來
<code>git init</code>		Git 初始化
<code>rm -rf .git</code>		移除 Git

常用指令

- git add file (選取上傳的檔案)
- git commit -m 'message'
(紀錄更動什麼東西，給一個標題)
- git push (push到github上，
類似上傳到google drive的感覺)
- git pull (把文件抓下來，
從google drive下載檔案的概念)

基本版更 (pull / push / add / commit / status)

Git	zsh	do
git status	gst	
git add (file)	ga (~)	
git add .	ga .	
git commit -m'message'	gcmsg '(~)'	
git pull	gl	
git push (remote) (branch)	gp (~) (~)	
git push -u (remote) (branch)		
git restore --staged (file)		取消 git add
git pull --rebase (remote) (branch)	gl	

課堂練習

- 新增資料夾 Variable_Practice
- 進入Variable_Practice
- 把你的 Code 全都移到 Variable_Practice

課堂練習

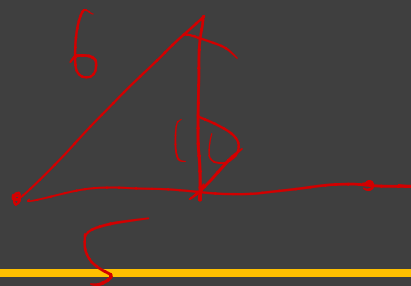
- 註冊github
- 建立新的 repo 「 Variable_Practice 」
- 在終端機初始 git 的環境

課堂練習

- 把本地的Code，上傳到 GitHub上

解題小撇步

解題步驟



長: 12

短: $6^2 - 5^2$

$$a^2 - c^2$$

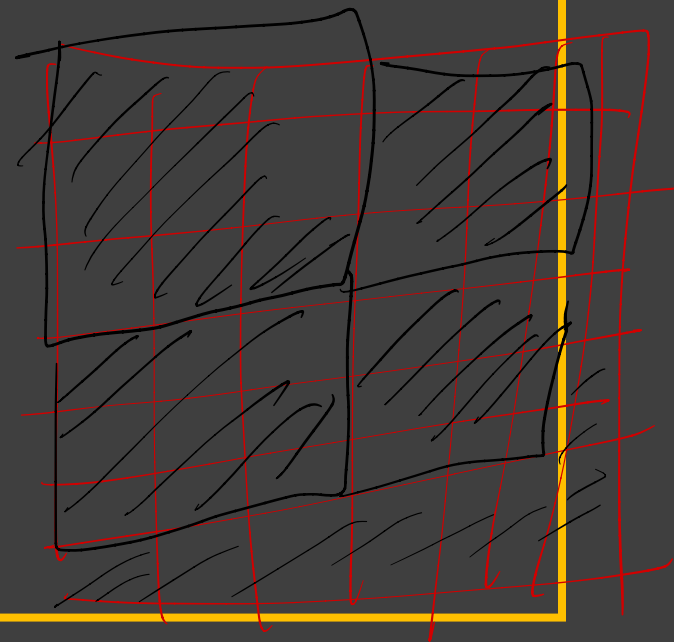
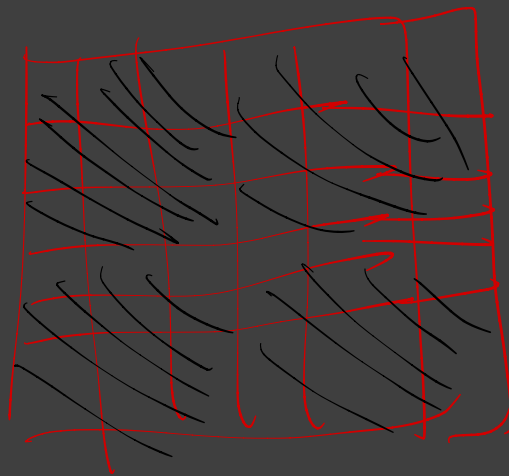
1. 寫出解題步驟及思路
2. 檢查步驟是否有問題
3. 將步驟轉為程式碼

課堂練習

- Zerojudge b004
- Zerojudge d039

輸入長寬

$n \times m \div 9$



課堂練習

- 第一個想法：除以9就可以
 - 但發現第二、三筆測資不符合這個狀況
- 第二個想法：邊長除以3，並相乘
 - 合理！！