



GIT 工具實習

2016/10/21

Te-Hsiu, Lu

GIT 簡介

- 分散式的版本控制系統
- 無窮無盡的開 branch (分支)
- 使用於產品開發
 - 可以把一份程式碼前後不同之處儲存下來，以便於讓你回復到任一版本

GIT 全域指令

- 在開始使用Git前，加入以下資訊

```
$ git config --global user.name "Your Name"
```

```
$ git config --global user.email "Your Email"
```

- 將git輸出可加入顏色以方便辨認

```
$ git config --global color.ui true
```

檢視 GIT 全域設定

- 螢幕輸出設定

\$ git config --list

- 顯示git設定檔案

\$ cat ~/.gitconfig

```
lthsIU101u@csie0[9:26pm]~>git config --global  
user.name "lthsIU101u"
```

```
lthsIU101u@csie0[9:26pm]~>git config --global  
user.email "lthsIU101u@cs.ccu.edu.tw"
```

```
lthsIU101u@csie0[9:27pm]~>git config --global  
color.ui true
```

```
lthsIU101u@csie0[9:27pm]~>git config --list  
user.name=lthsIU101u
```

```
user.email=lthsIU101u@cs.ccu.edu.tw
```

```
color.ui=true
```

```
lthsIU101u@csie0[9:27pm]~>
```

```
lthsui101u@csie0[9:29pm]~>git config --global
user.name "lthsui101u"
lthsui101u@csie0[9:29pm]~>git config --global
user.email "lthsui101u@cs.ccu.edu.tw"
lthsui101u@csie0[9:29pm]~>git config --global
color.ui true
lthsui101u@csie0[9:30pm]~>cat ~/.gitconfig
[user]
    name = lthsui101u
    email = lthsui101u@cs.ccu.edu.tw
[color]
    ui = true
lthsui101u@csie0[9:30pm]~>
```

開始使用 GIT

- 首先開一個資料夾

```
$ mkdir Fruit
```

- 並進入該資料夾

```
$ cd Fruit
```

- 建立一個新的 Repository

```
$ git init
```

```
lthsIU101u@csie0[9:32pm]~>mkdir Fruit
lthsIU101u@csie0[9:32pm]~>cd Fruit/
lthsIU101u@csie0[9:32pm]~/Fruit>git init
Initialized empty Git repository in /.amd_mnt
/cs1/host/csdata/home/under/u101/lthsIU101u/Fruit/.git/
lthsIU101u@csie0[9:32pm]~/Fruit>
```


GIT 基本指令-STATUS

- 檢查目前 Git 的狀態

\$ git status

- 在Fruit下新增編輯 apple.txt 檔案並觀察目前git狀態

```
lthsIU101u@csie0[9:41pm]~/Fruit>vi apple.txt
```

```
lthsIU101u@csie0[9:42pm]~/Fruit>git status
```

```
On branch master
```

```
Initial commit
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what  
will be committed)
```

```
apple.txt
```

```
nothing added to commit but untracked files p  
resent (use "git add" to track)
```

```
lthsIU101u@csie0[9:42pm]~/Fruit>
```

GIT 基本指令-ADD

- 使git來追蹤我們的檔案

\$ git add <檔案>

- 取消目前追蹤檔案版本

\$ git rm --cached <檔案>

```
lthsIU101u@csie0[9:47pm]~/Fruit>git add apple.txt
```

```
lthsIU101u@csie0[9:47pm]~/Fruit>git status
On branch master
```

```
Initial commit
```

```
Changes to be committed:
  (use "git rm --cached <file>..." to unstage
  )
```

```
    new file:   apple.txt
```

```
lthsIU101u@csie0[9:47pm]~/Fruit>
```

GIT 基本指令-COMMIT

- 將此次追蹤的檔案作註解

```
$ git commit
```

```
$ git commit -m "輸入此次變更記錄"
```

- 修改當前commit文字

```
$ git commit --amend
```

```
lths1u101u@csie0[9:56pm]~/Fruit>git commit -m  
"add apple.txt in 2016.10.21"  
[master (root-commit) e1b9193] add apple.txt  
in 2016.10.21  
1 file changed, 1 insertion(+)  
create mode 100644 apple.txt  
lths1u101u@csie0[9:56pm]~/Fruit>
```

修改流程



GIT 基本指令-LOG

- 查看commit的歷史紀錄

\$ git log

\$ git log --stat

\$ git log -p


```
lthsui101u@csie0[9:57pm]~/Fruit>git log  
commit e1b9193533ddee5f4c440ba468eb026b7be418  
ba
```

```
Author: lthsui101u <lthsui101u@cs.ccu.edu.tw>
```

```
Date: Thu Oct 20 21:56:46 2016 +0800
```

```
    add apple.txt in 2016.10.21
```

```
lthsui101u@csie0[9:57pm]~/Fruit>
```

GIT 基本指令- BRANCH

- 從主要 branch 再開出一條新的 branch 來做，這支新開的 branch 會帶著原本 branch 目前狀態，當完成所要開發的新功能或是bug 修正後確認沒問題就再把它 合併回主 branch
- 開新branch
\$ git branch <名稱>
- 列出所有branch
\$ git branch

```
lthsIU101u@csie0[9:59pm]~/Fruit>git branch ba  
nana
```

```
lthsIU101u@csie0[9:59pm]~/Fruit>git branch  
    banana
```

```
* master
```

```
lthsIU101u@csie0[9:59pm]~/Fruit>
```

GIT 基本指令- CHECKOUT

- 切換至其他branch

\$ git checkout <名稱>

- 切換至master branch

\$ git checkout master

```
lthsui101u@csie0[9:59pm]~/Fruit>git branch ba  
nana
```

```
lthsui101u@csie0[9:59pm]~/Fruit>git branch  
banana
```

```
* master
```

```
lthsui101u@csie0[9:59pm]~/Fruit>git checkout  
banana
```

```
Switched to branch 'banana'
```

```
lthsui101u@csie0[DING!]~/Fruit>git branch  
* banana
```

```
master
```

```
lthsui101u@csie0[10:00pm]~/Fruit>
```

GIT 基本指令- MERGE

- 修改branch banana 下的檔案 apple.txt，內容更改為
this is not an apple
it is a banana
- add、commit

```
lths1u101u@csie0[10:12pm]~/Fruit>git checkout  
banana  
Switched to branch 'banana'  
lths1u101u@csie0[10:12pm]~/Fruit>vim apple.tx  
t  
lths1u101u@csie0[10:13pm]~/Fruit>git add appl  
e.txt  
lths1u101u@csie0[10:13pm]~/Fruit>
```

```
lthsui101u@csie0[10:13pm]~/Fruit>git status
```

```
On branch banana
```

```
Changes to be committed:
```

```
(use "git reset HEAD <file>..." to unstage)
```

```
    modified:   apple.txt
```

```
lthsui101u@csie0[10:13pm]~/Fruit>git commit -  
m "edit apple.txt in 2016.10.26"
```

```
[banana f138b0f] edit apple.txt in 2016.10.26  
1 file changed, 2 insertions(+), 1 deletion(-)
```

```
lthsui101u@csie0[10:14pm]~/Fruit>
```


GIT 基本指令- MERGE (CONT.)

- 將branch合併起來

\$ git merge <branch名稱>

```
lthsui101u@csie0[10:14pm]~/Fruit>git checkout  
master
```

```
Switched to branch 'master'
```

```
lthsui101u@csie0[10:14pm]~/Fruit>git merge ba  
nana
```

```
Updating 64806ea..f138b0f
```

```
Fast-forward
```

```
apple.txt | 3 ++-
```

```
1 file changed, 2 insertions(+), 1 deletion(  
-)
```

```
lthsui101u@csie0[10:14pm]~/Fruit>cat ./apple.  
txt
```

```
this not an apple
```

```
it is a banana
```

```
lthsui101u@csie0[10:14pm]~/Fruit>
```

GIT 基本指令-CLONE

- 如果在網路上看到別人開源的專案想要複製下來，更改做自己的版本

\$ git clone <位置> <名稱>

```
lthsiu101u@csie0[1:17am]~>git clone https://github.com/gogojimmy/Animal.git MyAnimal
```

```
Cloning into 'MyAnimal'...
```

```
remote: Counting objects: 8, done.
```

```
remote: Total 8 (delta 0), reused 0 (delta 0), pack-reused 8
```

```
Unpacking objects: 100% (8/8), done.
```

```
Checking connectivity... done.
```

```
lthsiu101u@csie0[1:18am]~>ls
```

```
Maildir@  MyAnimal/  TMP/  WWW/
```

```
lthsiu101u@csie0[1:18am]~>
```

參考網頁

- 連猴子都能懂的Git入門指南
- <https://backlogtool.com/git-guide/tw/>
- Git 教學(1)：Git的基本使用
- <http://blog.gogojimmy.net/2012/01/17/how-to-use-git-1-git-basic/>

GIT TOOL LAB HOMEWORK

- 請列出“輸入姓名、信箱至git全域變數”的兩道指令，並使用git config --list指令顯示你在git裡面已經加入你的全域變數。(20%)
- 延續課程內容，用 git reset --hard HEAD~ 退回合併，建立一個新的branch名稱叫做orange，並切換至此branch，將apple.txt內容變更為

this is not an apple

it is an orange

提交並附註“edit to orange”，master依序合併banana及orange，請說明為什麼會Automatic merge failed，並擷取此時的apple.txt。(50%)

- branch、checkout、add、commit、merge的使用各5分
- apple.txt內容 10分
- 說明合併失敗原因10分
- 如何排除5分

- 比較“git log”、“git log --stat”、“git log -p”三個指令之間的差異。(30%)