

Git & Github

曹又霖

October 14, 2015

大綱

- 1 版本控制
 - 簡介
 - 版本控制種類
- 2 Git
 - 簡介
 - 使用教學
- 3 Github
 - 簡介
 - 使用說明
- 4 參考資料
 - 參考資料

為什麼要版本控制

- 改錯程式或誤刪檔案
- 還原之前寫的版本
- 團隊合作開發
- 軟體發行時，要分成穩定版和開發版

本地端版本控制

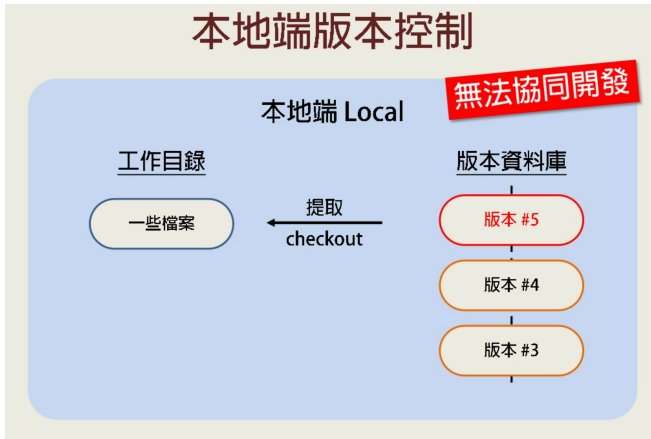


Figure 1: 本地端版本控制

中心式版本控制

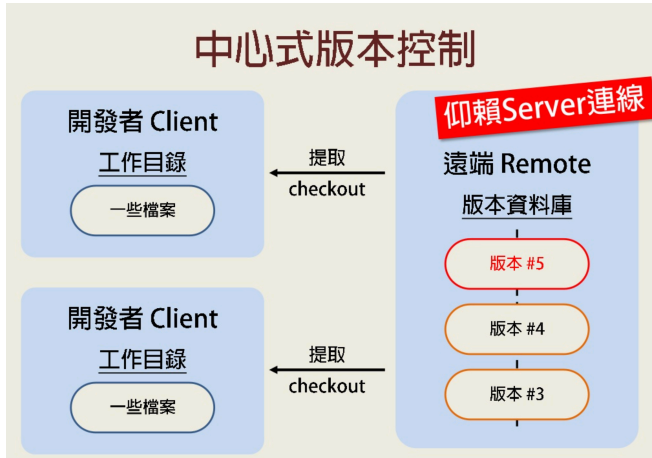


Figure 2: 中心式版本控制

分散式版本控制

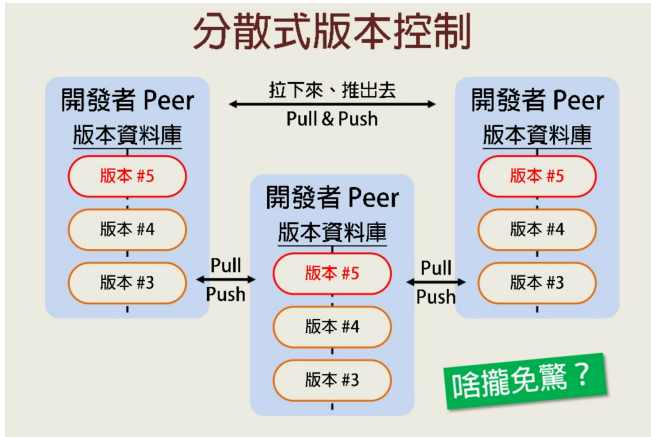


Figure 3: 分散式版本控制

Git

- 發明人 Linus Torvalds (即 Linux 創始人)
- 目的為管理 Linux 程式碼
- 是一分散式版本控制

Git 目標

- 速度快
- 簡單使用
- 非線性開發
- 分散式開發

安裝 Git

- 安裝 Git
 - <https://help.github.com/articles/set-up-git/>
- 安裝 Git GUI
 - GitX (Mac OS X)
 - TortoiseGit (Windows)

設定 Git

設定名稱

```
git config --global user.name "Maplewing"
```

設定 Email

```
git config --global user.email "sinmaplewing@gmail.com"
```

建立一個 Git Repository

- Repository(倉庫)：版本控制系統的管轄之下的儲存位置。
- 有兩種方法可以建立 Git Repo：

創建新的 Git Repo

```
git init
```

複製一個現有的 Git Repo

```
git clone  
https://github.com/sinmaplewing/Mini\_Monopoly.git
```

加入檔案

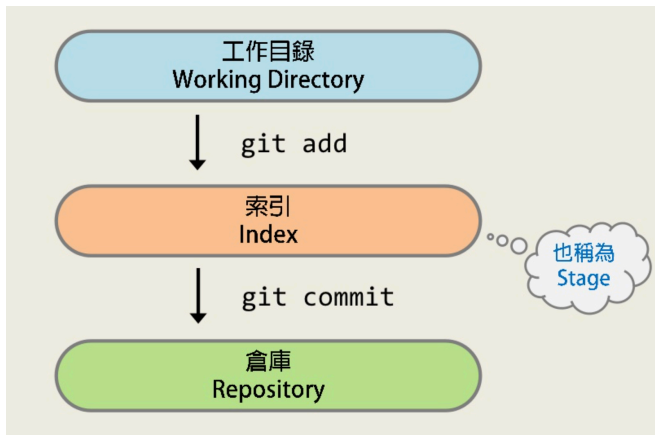


Figure 4: 加入檔案流程

加入檔案

加入檔案進入 Stage

```
git add code.c  
git add .
```

將變更記錄送入 Repo

```
git commit -m "Add Code.c"  
git commit -a
```

檔案管理

- Git 會管理空檔案，但不理會空資料夾。

改名檔案或是移動檔案

```
git mv code.c code.cpp
```

刪除檔案

```
git rm code.c
```

.gitignore

- 告訴 Git 哪些檔案不用加入。
- <https://github.com/github/gitignore>：這裡有大部分專案的.gitignore template 可以參考。

狀態與紀錄

觀看目前狀態

```
git status
```

觀看 commit 紀錄

```
git log  
git log --oneline  
git log --oneline --decorate --graph
```

觀看差別

```
git diff
```


復原檔案

復原至上次 commit 的內容

```
git checkout HEAD^  
git checkout HEAD^ code.c
```

將檔案變回 unstage 狀態，不更動檔案

```
git reset HEAD code.c
```

使用分支

- 預設分支為 master

查看目前分支

```
git branch
```

創建新分支並移過去

```
git checkout -b newbranch
```

移動分支

```
git checkout master
```

合併指定分支到目前所在分支

```
git merge newbranch
```

Github

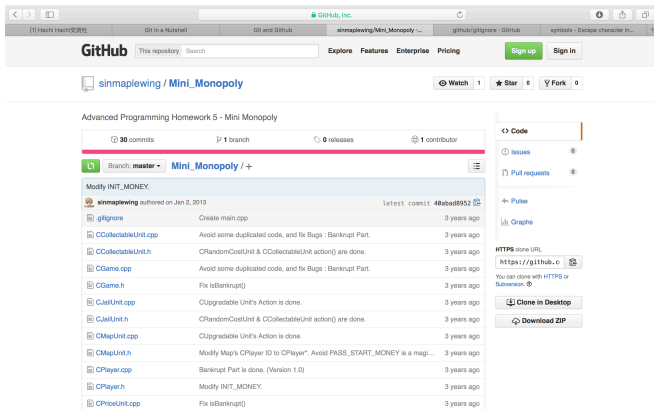


Figure 5: Github

Github

- `http://www.github.com`
- 線上的 Repository
- 可創建 Repo、fork Repo 或是共同合作

使用 Github

設定 Remote

```
git remote add origin Repo_url
```

將 Local 的變更推到 Remote 上

```
git push -u origin master
```

將 Remote 的變更拉到 Local 上

```
git pull  
git pull origin master
```

Demo

Demo!

參考資料

- Git in a Nutshell: <http://www.slideshare.net/Dannvix/git-in-a-nutshell>
- Git & Github: <http://www.slideshare.net/ihower/git-and-github-7306407>
- 連猴子都能懂的 Git 入門指南:
<https://backlogtool.com/git-guide/tw/>
- Git 版本控制系統 | ihower 的 Git 教室:
<https://ihower.tw/git/>
- Git 初學筆記 - 指令操作教學: <http://blog.longwin.com.tw/2009/05/git-learn-initial-command-2009/>