

我的 repository 網址：[syun413/Bonus-assignment \(github.com\)](https://github.com/syun413/Bonus-assignment)

學習到的 Git 指令：

- (1) git init：建立新的本地端 Repository。
- (2) git clone：複製遠端的 Repository 檔案到本地端。
- (3) git status：檢查本地端檔案異動狀態。
- (4) git add [檔案或資料夾] 將指定的檔案（或資料夾）加入版本控制。
- (5) git add . 將指定的檔案（或資料夾）加入全部。
- (6) git commit 提交目前的異動。
- (7) git commit -m "提交說明內容"：提交目前的異動並透過 -m 參數設定摘要說明文字。
- (8) git stash：獲取目前工作目錄的 dirty state，並保存到一個未完成變更的 stack，以方便隨時回復至當初的 state。
- (9) git log 查看先前的 commit 記錄。
- (10) git push 將本地端 Repository 的 commit 發佈到遠端。
- (11) git push origin [BRANCH_NAME] 發佈至遠端指定的分支（Branch）。
- (12) git branch 查看分支。
- (13) git branch [BRANCH_NAME] 建立分支。
- (14) git checkout [BRANCH_NAME] 取出指定的分支。
- (15) git checkout -b [BRANCH_NAME] 建立並跳到該分支。
- (16) git branch -D [BRANCH_NAME] 強制刪除指定分支（須先切換至其他分支再做刪除）。
- (17) git reset --hard [HASH] 強制恢復到指定的 commit（透過 Hash 值）。
- (18) git checkout [HASH] 切換到指定的 commit（與 git checkout [BRANCH_NAME] 相同）。
- (19) git branch -m <OLD_BRANCH_NAME> <NEW_BRANCH_NAME> 修改分支名稱。
- (20) git remote add origin <GitHub 提供的網址>：在 GitHub 提供網址的位置架一個代號為 origin 的遠端數據庫
- (21) git push -u origin master：在本地端資料推送到 GitHub 的 origin 數據庫 master 分支上。
- (22) git pull origin master：將遠端有更動的版本同步下來。
- (23) git checkout -b <branch-name>：新增並切換到新分支，修改檔案內容
- (24) git commit -am "update" git push origin <branch-name>：將有調整檔案的新分支推送到 GitHub
- (25) pull request：GitHub 會自動偵測有沒有新的分支，選擇提交要推送的分支併入遠端的 master 的請求。review 後沒問題可以將分支併入 master，並刪除分支。