|  |  |  |
| --- | --- | --- |
| **準備使用 git** | | |
| git clone (git url) | 複製資料Online Data到 localtion端 | |
| git init | 使資料夾具有 git 同步 / 上傳的功能 | |
| **添加檔案** | | |
| git add  (檔案名稱 ex : content.txt) | 新增檔案到暫存檔 | |
| git add --all | 新增該資料夾的所有檔案 | |
| **暫存檔資料送出** | | |
| git commit -m  "(要備註的訊息)" | 將暫存檔的所有資料送出 | |
| **將檔案復原回上個 commit 的狀態** | | |
| git checkout (檔案名稱) |  | |
| **資料上傳** | | |
| git push | 提交所有暫存檔的資料到 Online | |
| git push -u origin  (分支名稱) | 提交所有暫存檔的資料到 Online ，並選擇要提交的 branch | |
| **更改檔案名稱** | | |
| git mv (content.txt ,test.txt) | ( 要更改的檔案名稱 , 要更改的名稱) | |
| **刪除檔案** | | |
| git rm (檔案名稱) | 刪除 localtion端的檔案 | |
| git rm --cached  (檔案名稱) | 刪除 Online端的檔案 | |
| **獲取** | | |
| git fetch | 並不想抓檔案下來同步，只是想確認本地和線上的版本是否有差異 | |
| git pull | 將 Online端 的資料 Download 同步下來。 ( pull的內容就是 fetch + merge 所組成的 ) | |
| **查看內容** | | |
| git status | 檢查當前資料的狀態。( 可看到刪除 / 新增的資料有哪些 ) | |
| git log --oneline | 檢查過去所有的 Commit 紀錄 | |
| **分支 (branch)** | | |
| git branch | 檢查目前在哪個branch 上 | |
| git checkout (分支名稱) | 切換到另外一個 branch | |
| git branch (分支名稱) | 建立一個新 branch | |
| git push -u origin (分支名稱) | 當存在兩個以上 branch 時，檔案必須選定要上傳的 branch | |
| git branch (分支名稱) -d | 刪除 branch 。 (master 不可刪除) | |
| git merge (別的分支名稱) | Branch 合併，將別的 branch 資料合併過來，注意可能會有 “衝突” 。 | |
| ***額外狀況 :*** | | |
| **git commit 指令下完，想要收回** | | |
| git reset HEAD content.txt  or  git reset HEAD^^  git checkout (檔案名稱) | 取離暫存檔 (單個 / 全部) | |
| git checkout => 復原回上個 commit | |
| **git commit ( HEAD )收回時，又想要復原** | | |
| git reset (commit 版本號) --hard | 復原上一次的 Commit | |
|  | **\* 若忘記查看版本號 : git reflog** | |
| **若想更改最後一筆 git commit 所傳送的訊息** | | |
| git commit --amend -m "(重新輸入要傳送的訊息)" | |  |

**互動模式 操作 :**

|  |
| --- |
| 離開 Esc + : |
| 編輯 : i |
| 復原 : u |
| 存檔 + 離開 : Esc + : -> wq |