Git instruction(in ubuntu):

* Git init(first cd to target folder)
* Git configure(follow below url)
  + <https://blog.techbridge.cc/2018/01/17/learning-programming-and-coding-with-python-git-and-github-tutorial/>
* Git remote add repo url
  + (example: git remote add disksim\_github “https…..”)
  + Repo is my github repository name
  + url is my github url(https or ssh)
* git remote remove repo
  + remove the repo you add
* git public key設定：
  + <https://blog.jaycetyle.com/2018/02/github-ssh/>
* git clone --single-branch --branch master url
  + for example: git clone –single-branch –branch simulation https:…
  + master is my github branch name
  + url is my github url(https or ssh)
* git push repo branch
  + 先clone
  + Git add . (追蹤所有檔案) or git add filename
  + Git commit -m <任何當下那個版本的標題---隨便打>
  + 然後才能push
  + 若還是無法，使用git push -f repo branch指令
  + Git commit -v：顯示所有的更動細節
  + Git log -p：與上述極為相似，可實際操作看看差異
* Git pull repo branch(同git push)
* Branch
  + <https://zlargon.gitbooks.io/git-tutorial/content/branch/create_delete.html>
* 無法更改branch name，只能透過以下做法達成：假設目標是要將simulation改成simulation\_new
  + 先新建一個branch，命名時，就以要更改的名稱命名(simulation\_new)。
  + 在新建時，可選擇依照某個branch來新建（這樣就會將指定branch(simulation)的歷史紀錄-commit一併載入）---就依照想要更改的branch(simulation)
  + 接著刪除舊的branch(simulation)
  + 這樣一來，就如同更改名稱一樣，將原本的simulation改成simulation\_new，歷史commit也都和原本的simulation一樣。
  + 之後要做什麼更動，就接續做下去，完全沒有任何問題
* 檔名無論如何絕對不能包含『(』or 『)』
* 解決無法上傳大檔案的方式：使用下列指令
  + 找出寫入大檔案的來源程式碼，然後在最後fclose後面加一行fopen，以w模式開啟，即可清掉所有寫過的內容
* 參考網站：
  + <https://stackoverflow.com/questions/23517464/error-cannot-pull-with-rebase-you-have-unstaged-changes>
  + <https://andy6804tw.github.io/2018/12/09/git-exceeds-size/>
* 登入http相關
  + Git config -l可解決登入的問題（在push or clone的時候）
  + 參考網站：
    - <https://stackoverflow.com/questions/68775869/support-for-password-authentication-was-removed-please-use-a-personal-access-to>
* 若用ssh無法push使用：(參考以下網址)
  + <https://medium.com/@wama.tw/%E6%96%B0%E5%A2%9E-github-ssh-key-3e941c281ffa>
* 若不小心commit & push一個大檔案，可靠下列方式撤回commit & push(其實push原本就不可能成功，所以是撤回commit)
  + Git reset HEAD~
    - 上述指令做一次就會將該commit相關資訊徹底清除，只需做到【包含大檔案的commit】之前即可，這個指令不會影響當下檔案，只會針對【.git】資料夾的檔案做刪除
  + Git reset –hard HEAD~
    - 這個指令就與上述不同，會影響當下檔案!!!!!