Git collaboration workflows

假設三個人在 clone 了同一個東西下來分頭做作業

A 先做了一些改變後 push 上去了，這時候 B 如果再 push 上去的話就會被通知說要先 pull 下來去做 merge 才能再 push 回去。

這時候會有一個問題，如果有一個人叫做 C，他的 code 遇到問題了那也是要 push 上去才能請別人處理，就像上述說的，必須要去 merge 然後可能要處理 conflict，這時候如果再 push 回去的話就會變成所有人都有爛掉的 code 了

所以說不能只 work 在 master branch!!!!因為要一直到處理 merge，每個人都工作都會干擾的 main code，這樣就不能實驗了。再來就是要合作的話就只能 push 尚未完成的

code 上去了，現在大家都只能拿到爛 code

Feature branches

就是沒有人在 main 工作，大家都在 feature branch 做工作啦，所有的東西應該要在不同的分離的 branch 工作。

這樣 master branch 就不會被汙染了。

這樣子要做實驗，測試之類的動作都會變得比較簡單。

一旦確定這個 branch 真的真的沒有問題了而且也滿意了，那就把他跟 master 去做 merge 的動作嘍

通常不會所有的 branch 都要一直 push 上去 github 啦，情況會是需要別人幫我檢查我的 code 的時候才會 push 上去

在檢查完都沒問題會將這個 branch 融入 master，最後再把這個 branch 刪除。

Merging In Feature Branches

Merge 之前要先討論過

或是說用 pull request

Pull Request

基本上就是在說”我這些東西要 merge 進 master 囉，你們覺得怎麼樣??”

Pull request 的按鈕可以在 github 上找到