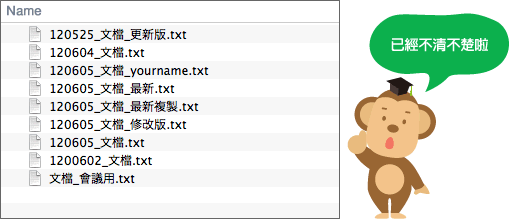
**Git & GitHub**

1. 版本控制

Q:開發程式時需要將檔案回復上一次編輯的狀態或備份，正常會?

A:複製檔案，並將檔名附上日期或說明。

但…這樣會很麻煩而且還會很雜亂，如下圖。



1. 使用Git管理檔案版本:

\*Git是什麼?

(1).為分散式版本控制系統

(2).可以把檔案的狀態作為更新歷史記錄保存起來。因此可以把編輯過的檔案復原到以前的狀態，也可以顯示編輯過內容的差異。

1. Github

What is Github

1. 是一個 git 的托管網站，你可以在本地端數據庫使用Git版本控制功能。但如果想要公開在本地端數據庫的修改內容，就需要將內容上傳到遠端數據庫(Github)，透過遠端數據庫(Github)還可以取得其他人修改的內容。
2. 可以讓開發者將自己的專案儲存到網路上，與人分享
3. 我們可以方便的使用 Git 在本機進行專案管理。
4. Github教學

Step1 :安裝git

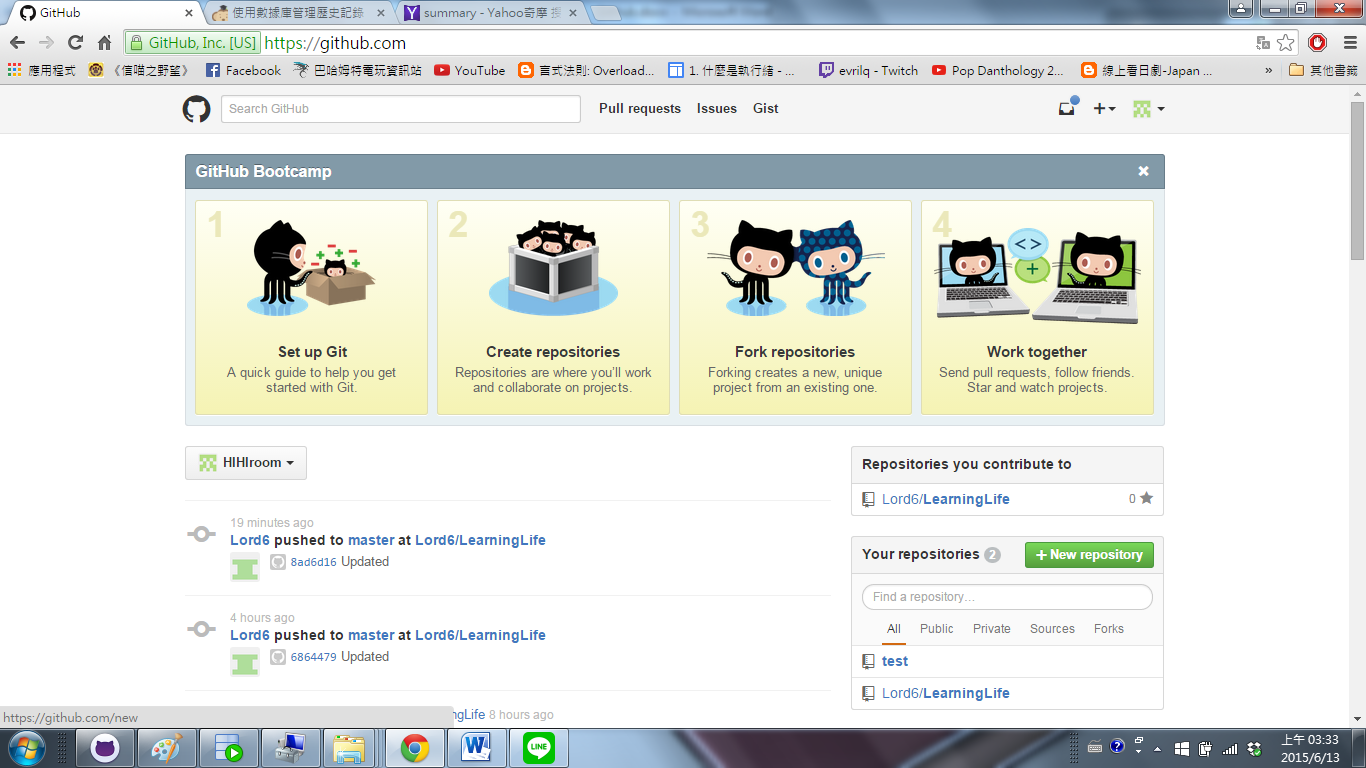
Step2 :到Github註冊帳號

Step3 :安裝Github軟體

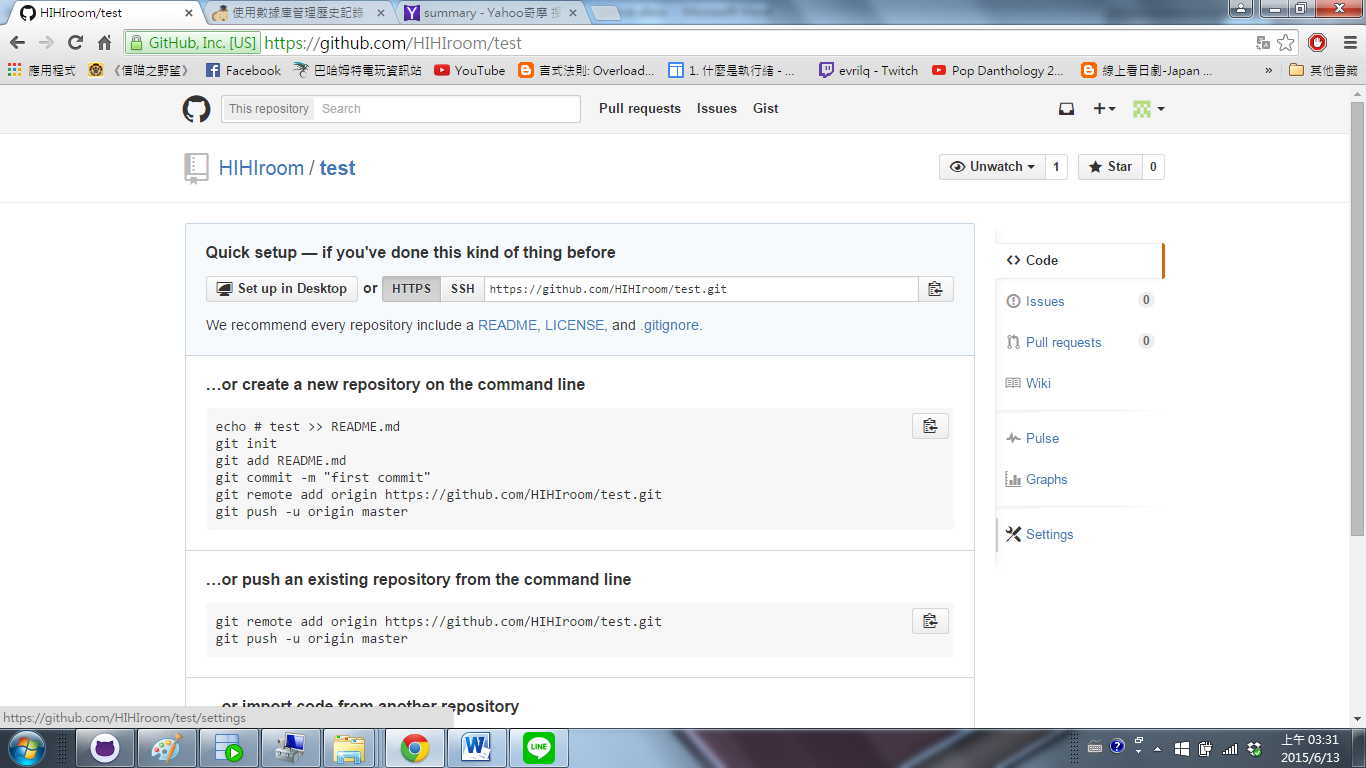
**利用github與人分享專案**

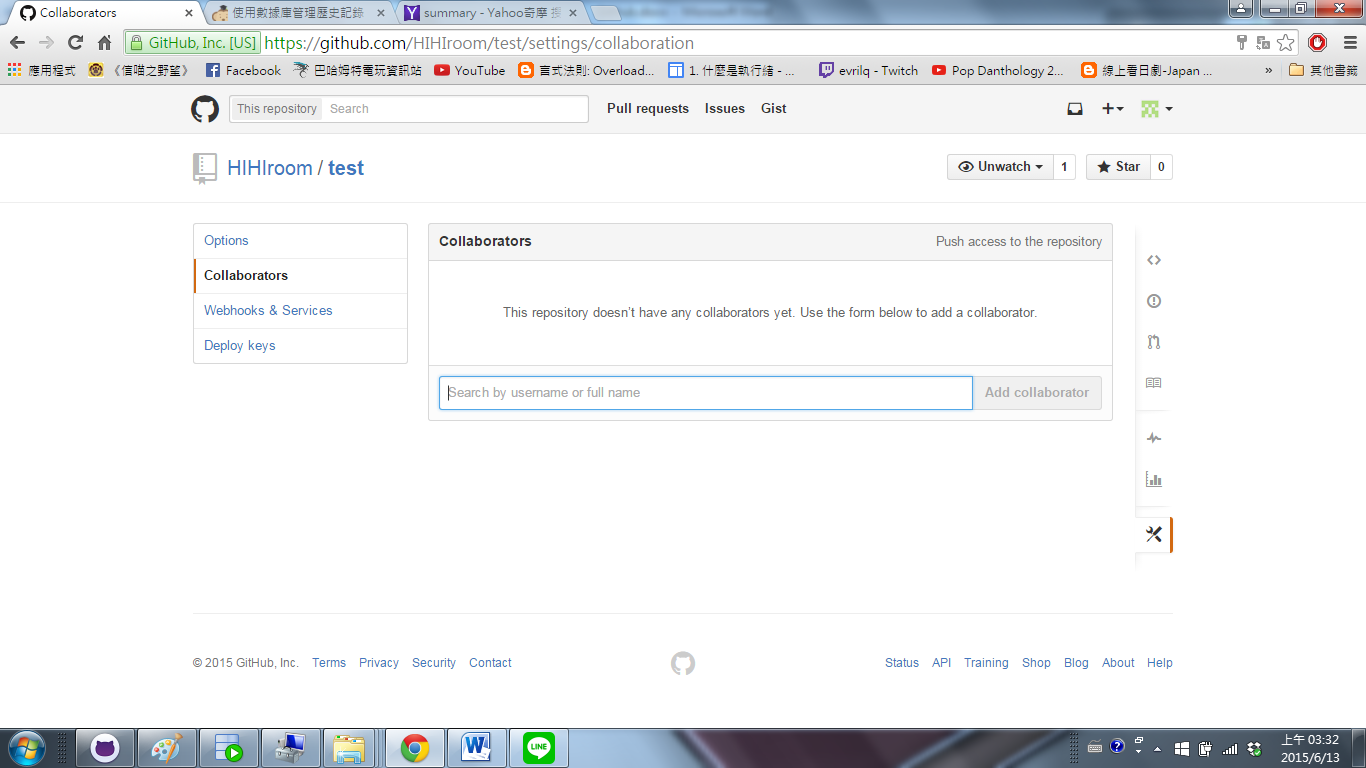
專案主:

1. 先在github建立新的repository



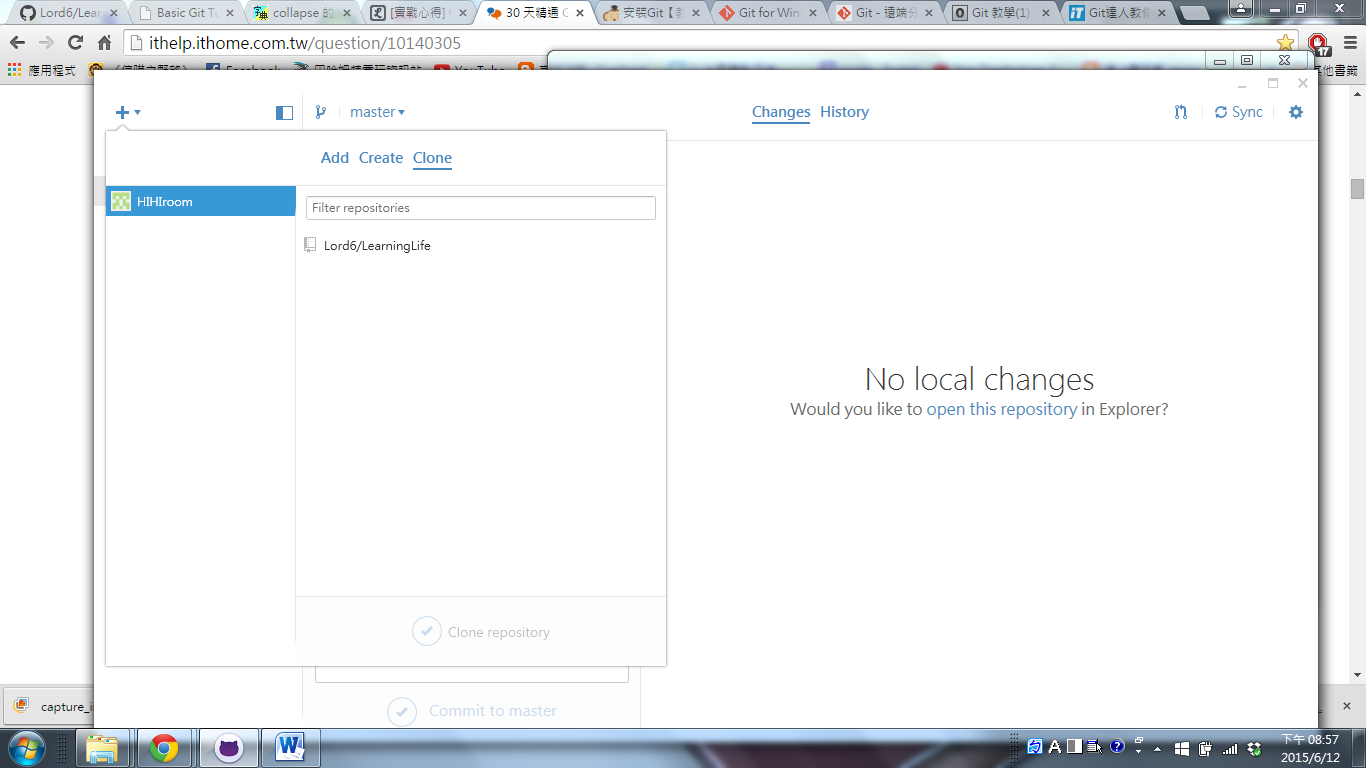
1. 在欲分享repository頁面上->settings->Collaborators->新增組員的github帳號

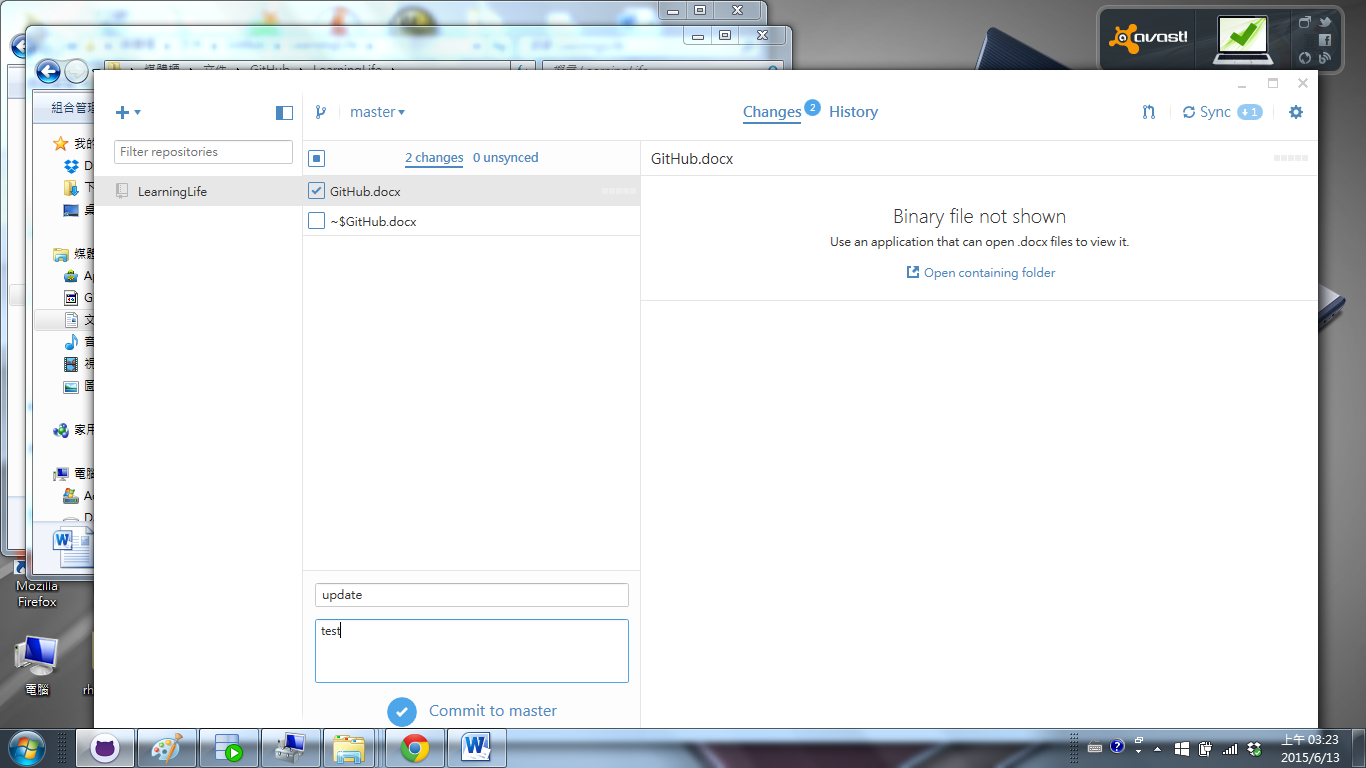




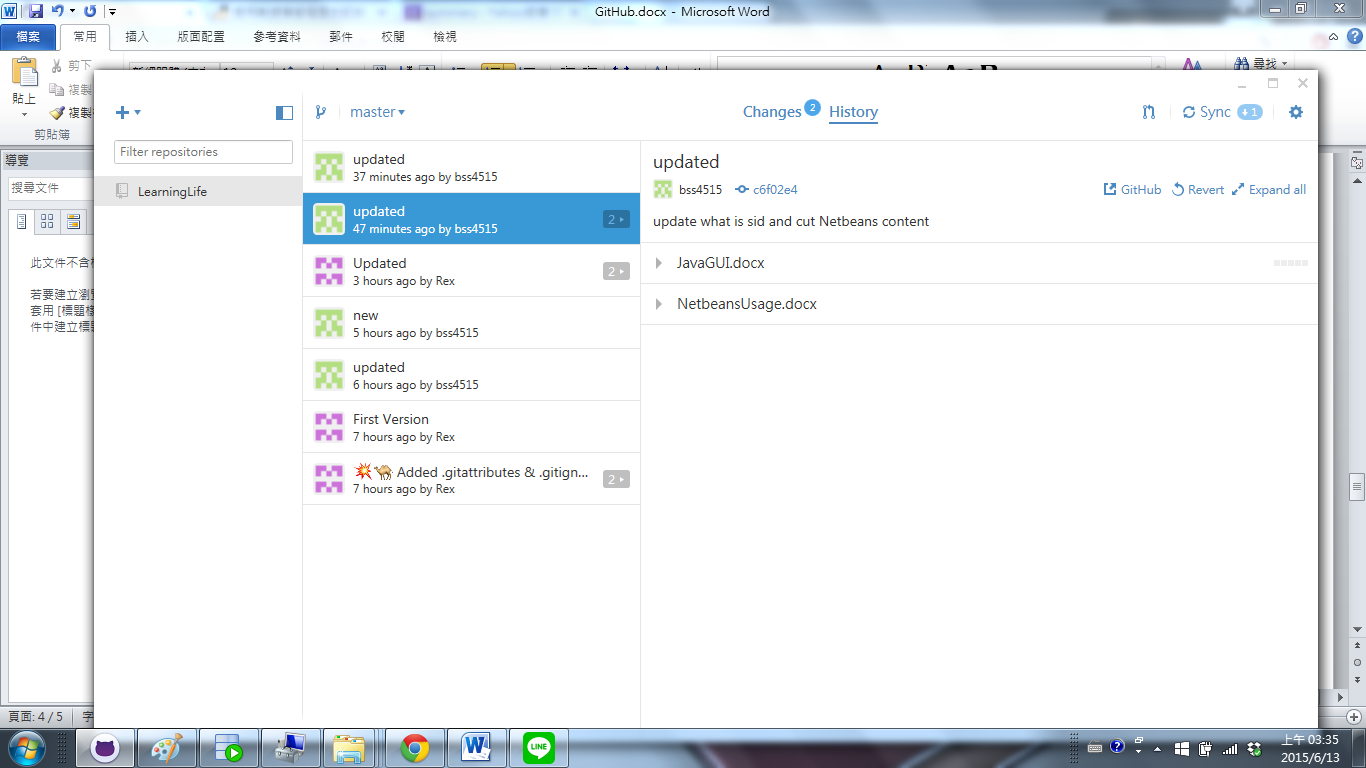
組員:

被加入的組員在github軟體上會出現專案主的repository，組員要做修改和上傳時:

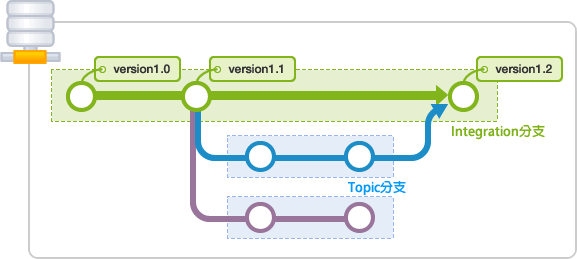
1. 如下圖點clone，下載repository至本機端
2. 在本機端更改後的文件會出現在github軟體上，此時需清楚輸入summary與description後即可commit to master(本機端檔案版本更新)，之後sync即可與專案主同步檔案版本(遠端檔案版本更新)



1. 回復檔案:history->選擇commit->點Revert ，即會回復成所選commit的上一個版本

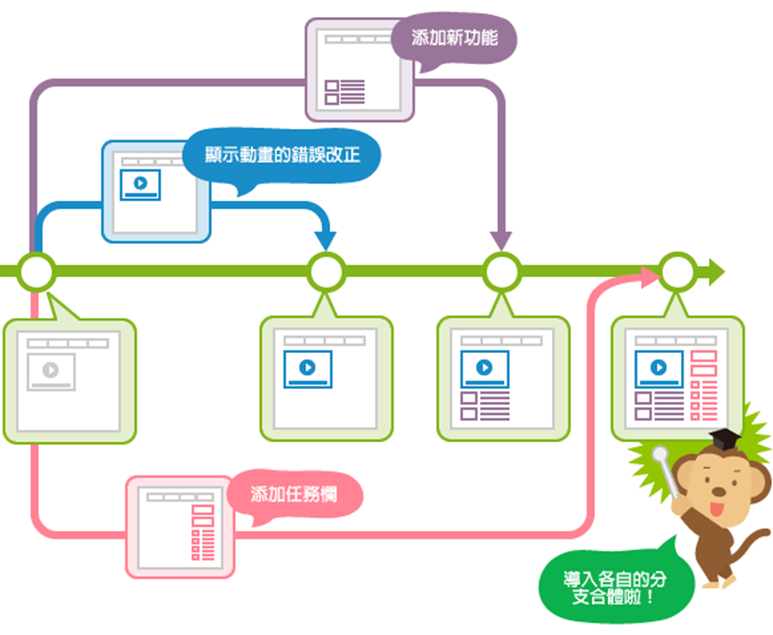


**Branch:**



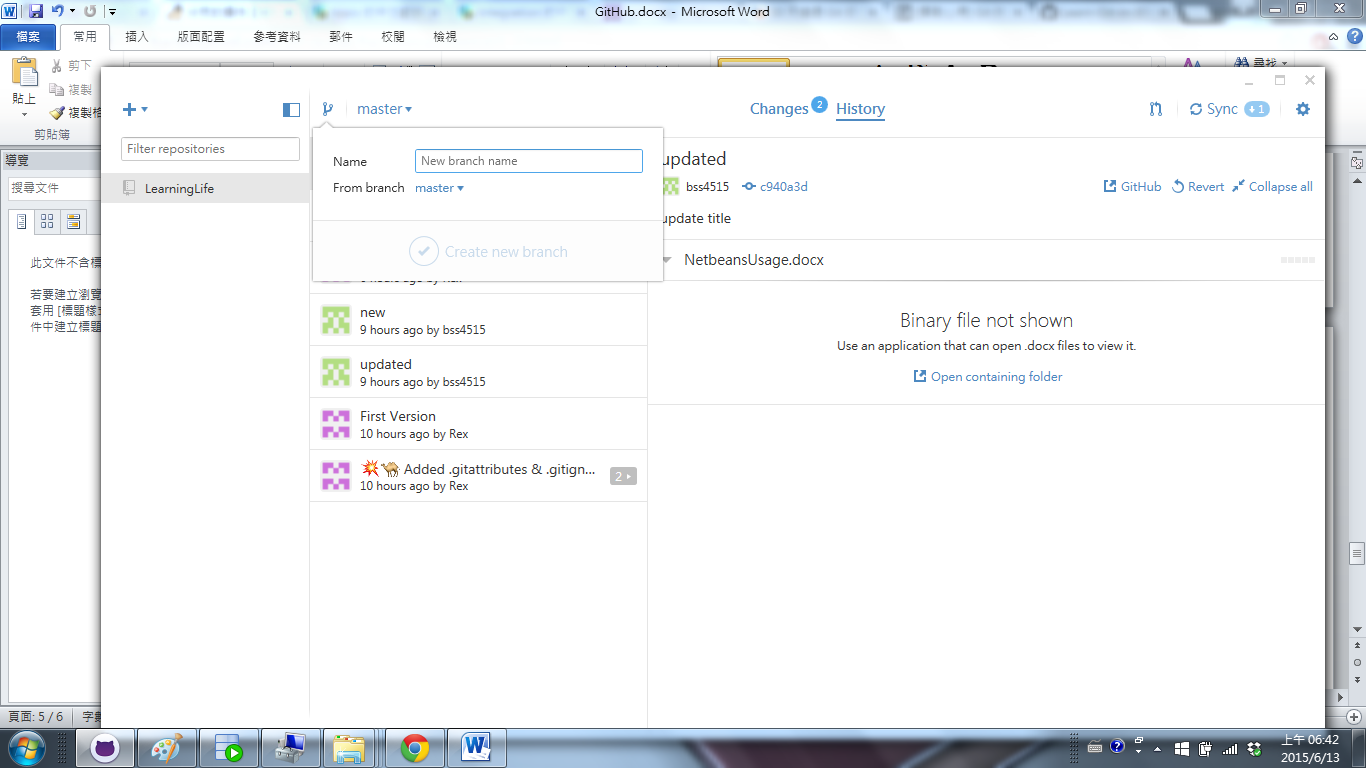
Integration分支: 為了可以隨時建立發布版本的分支， 通常會將系統預設的master分支當作Integration分支使用

Topic分支: 為了開發功能或修復錯誤之類的任務所建立的分支。若同時進行多個任務時，您必須建立多個的Topic分支。完成作業後，要將Topic分支再合併到Integration分支。

當需要同時進行多個不同的修改，並讓分開的分支不受其他分支的影響，可以在主要分支建立自己專用的分支， 將自己分支的修改再導入到主要分支 

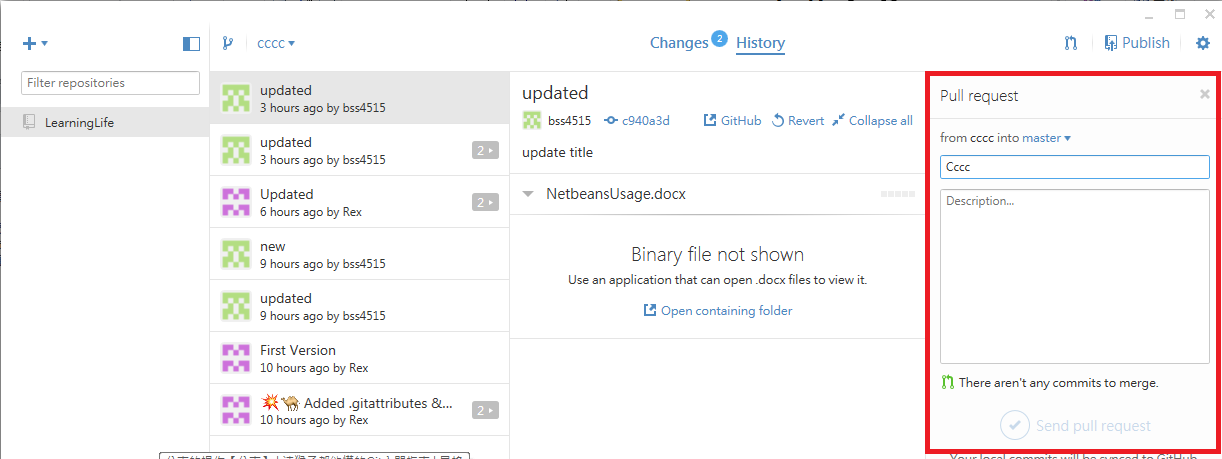
**建立新的branch**

在github軟體中建立branch如下圖



Pull request:

當新建立的分支完成，向專案主寄送Pull request，當專案主確認後即可合併，注意分支對應需設定正確



http://backlogtool.com/git-guide/tw/stepup/stepup1\_1.html

**Fork與pull request**

\*Git 儲存庫並沒有什麼「權限控管」的概念

\*在 GitHub 的個人帳戶下，只要你授權特定人為協同開發人員 (Collaborators)，他就能自由的

Push 與 Pull 專案原始碼。

\*GitHub 採用了 forks 與 pull request 的流程，可以做到基本的權限控管。

當專案主未授權時，就得在該專案使用forks，這樣就會在自己的github帳號下複製這份專案，並能下載到本機端作修改，但以遠端角度來看，這時只有唯讀權限，並沒有寫入，所以當你在本機端更改完成後，得向專案主寄送pull request ，待專案主確認完畢即可完成合併工作。

http://ithelp.ithome.com.tw/question/10140305