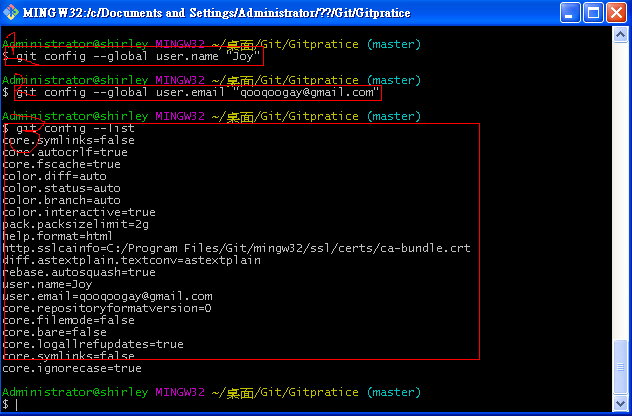
**Git 、 GitHub 以及TortoiseGit**

本文件的內容為介紹以及教學Git、GitHub以及TortoiseGit的使用以及應用。

1. **下載Git**

****

1. **設置你的Git**

****

先設定你的基本資料(之後commit的時候才容易知道作者是誰)

1.就是設置你的姓名

git config - -global user.name “你的姓名”

2.就是設置信箱

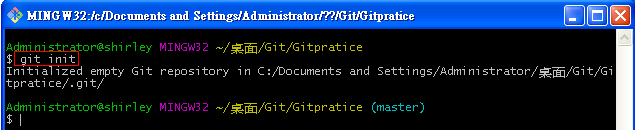
git config - -global user.email. “你的信箱”

3.檢查你的狀態，看你信箱設置多少跟姓名設置什麼

git config –list

**3、如何將自己的資料夾變為儲存庫**

****

****

輸入git init，就可以把資料夾變成儲存庫了

**Git的基本介紹**

首先我們來介紹Git，也就是自己的儲存庫的一些狀態，這樣才可以管理自己的儲存庫，有分四種狀態

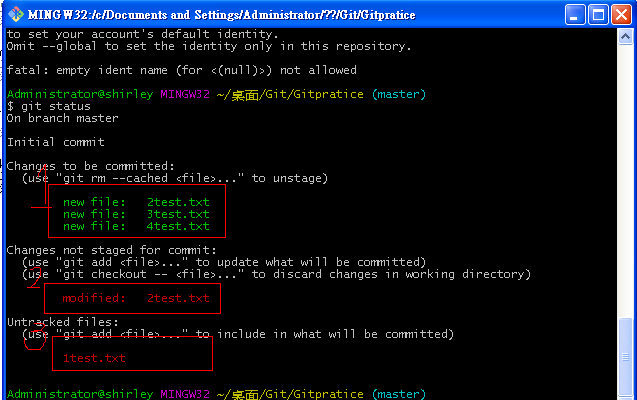
* 未追蹤(Untracked files)：在版本提交後才又加進來的檔案，
* 這些檔案並沒有被GIT所追蹤控管，簡單來說就是在這個文件夾變成儲存庫後面才新加的。
* 已更改(Changes not staged for commit)：已提交版本後，卻又再次修改，這些檔案會被丟回工作目錄(WD)，簡單來說當檔案從Changes to be committed or Committed **只要這個檔案被修改都會變成** Changes not staged for commit
* 等待提交(Changes to be committed)：在工作目錄（未追蹤Untracked files）的檔案執行git add後，會放在暫存區(Stage)。這些放在暫存區的檔案狀態便是等待提交囉！
* 已提交(Committed)：在暫存區(Stage)的檔案執行git commit後，檔案便置於儲存區(Repo)，這些放在儲存區的檔案即是已提交的狀態
* commit 是寫程式時一個很重要的動作，一個 commit 在 Git 中就是一個節點，這些 commit 的節點就是未來你可以回朔及追蹤的參考，你可以想像就像是電玩遊戲時的存檔，每一個

**Git的基本介紹**

commit 就是一次存檔，讓我們未來在需要的時候都可以回到這些存檔時的狀態。因此我們將剛剛做的修改提交：

**4、儲存庫的四種型態**

在此之前要先提到一個指令”git status” 可以看到這檔案中所有的檔案的狀態



以上圖為例

* 就是已經已提交(Committed) or 等待提交(Changes to be committed) 。
* 你看到2為什麼已提交的檔案又變成是modified 2test.txt 這表示這個檔案被修改過了 你要 git add 2test.txt 在 git commit 2test.txt 在把它變成已提交的檔案。
* 就是新加入的檔案,不是儲存庫裡面的檔案。

**4、儲存庫的四種型態**

問:那如何分已提交(Committed) or 等待提交(Changes to be committed)?

1.第一種狀態是未追蹤(Untracked files)，也就是當檔案變儲存庫，在加入的文件

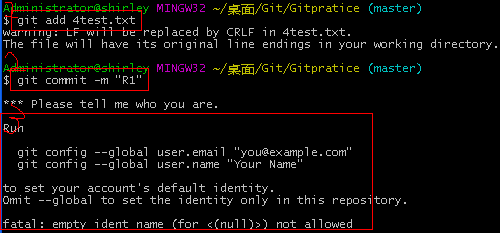
2.先說第三種狀態是等待提交(Changes to be committed)輸入git add (檔案名稱)，就可以把未追蹤(Untracked files)－＞等待提交(Changes to be committed) 。

像下面那張圖我的檔案就叫3test.txt所以我就輸入git add 3test.txt

C:\Documents and Settings\Administrator\桌面\Git\Gitpiture\Gitstatus3.bmp

3.在來說第四種狀態是已提交(Committed),就是一個準備上傳的檔案了

**4、儲存庫的四種型態**



以上圖為例

1就是把4test.txt變成等待提交(Changes to be committed),意思就是準備可以提交到自己儲存庫了。

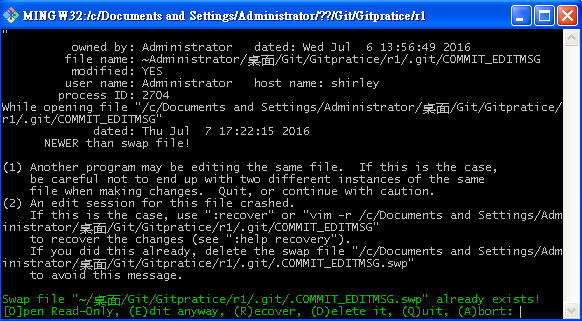
2就是把等待提交(Changes to be committed)-> 已提交(Committed),表示這份檔案已經傳到到自己儲存庫了　–m 就是快速提交“R1”,而這裡面的””,就是註解就是提交的這個節點。

3就是告訴我們的信箱並沒有設,他不知道自己儲存庫當上傳到雲端儲存庫時要叫做什麼信箱，姓名要叫什麼，因為當遠端儲存庫要跟自己的儲存庫作push跟pull時他要知道你是誰,才能夠給你push or pull 不然,隨便一個人都可以push or pull那就誰都可以動到你的遠端儲存庫。

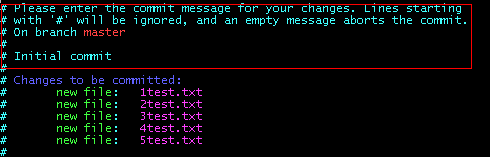
**4、儲存庫的四種型態**

在前面提到 commit 在 Git 中就是一個節點，一個很重要的概念就是commit是有順序的,順序可以變一張圖

假如是用　git commit 就會跑出



這時可以按q離開



這個視窗中最上面的空行是要給你寫下你這次 commit 的訊息，例如在這裡寫下 “Add test.rb file to test git function”來表示這一次提交的目的，**強烈建議在 commit 的時候要盡量清楚表達這次 commit 的內容為何**，因為這會讓你未來要回頭看 code 的時候能讓你快速的找到你想要找的內容，也能對團隊中其他成員了解你在做什麼。

**4、儲存庫的四種型態**

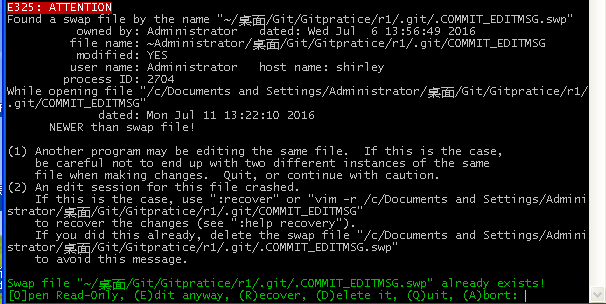
再說另一個指令git commit --amend

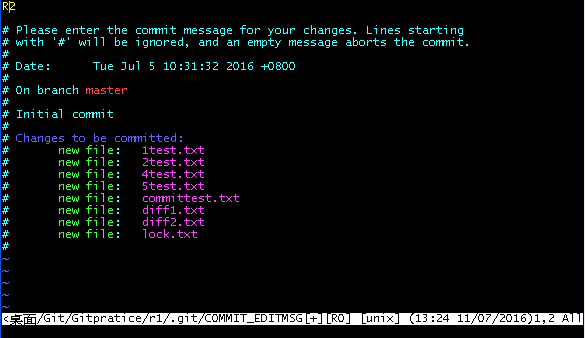
這是用來修改上一個commit註解的指令

我們來把R1改成R2

輸入git commit –amend

然後就會跑出像之前git commit 的這個畫面



按下i跑到編輯模式然後把R1改成R2 按下Esc 打上:wq!就修改歷史記錄了

**vim 幾個簡單指令**

*i 進入編輯模式(自由打字啦)  
Esc 退出編輯模式(回到一般模式)  
dd 刪除一行(一般模式下)  
:wq! 存檔離開(一般模式下)  
:q! 不存檔直接離開 (一般模式下)*