1. 將現有專案放到Gitlab上
2. 在Gitlab上建立專案
3. 在Gitlab建立分支init
4. 在本機專案上建立專案

git init

1. 變更預設分支為main

git branch -m master main

1. 設定同步遠端儲藏庫

git remote add origin <http://twtp1apl115.local.wonderland.com.tw/pmic/contentsystem.git>

1. 同步遠端記錄到本地

git fetch

1. 切換到init分支

git checkout init

1. 建立.gitignore

<https://www.toptal.com/developers/gitignore>

1. 將檔案簽入index

git add .

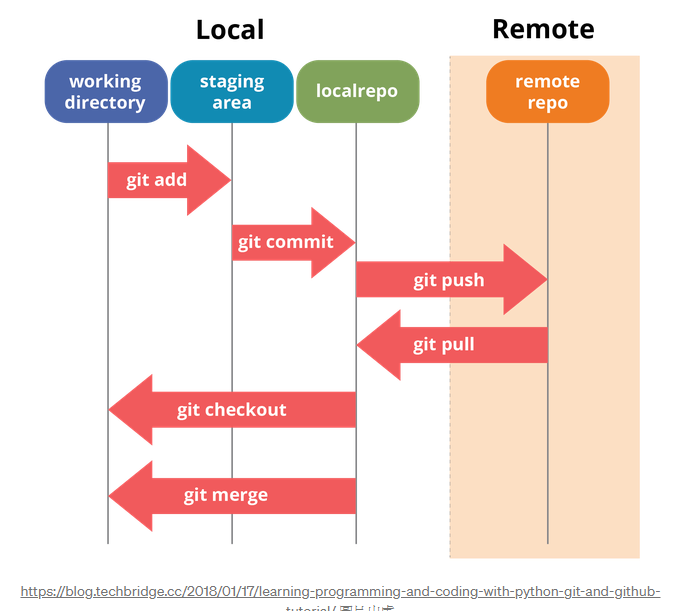
1. 將index 簽入本機儲藏庫

git commit -m "專案建立"

1. 將branch(init)的異動記錄推送到遠端儲藏庫

git push origin init

1. 在Gitlab提交merge request，並同意合併到main
2. Git流程圖



1. 常用指令
   1. 建立版本控管

git init

* 1. 查看目前版本狀態

git status

* 1. 將檔案加入index

git add filename #將特定檔案加入

git add . #將所有變更過的檔案加入

* 1. 將某個檔案移除版本控制

git rm --cached filename

* 1. 將index簽入版控

git commit -m "說明"

git commit --amend #在最新的commit異動檔案，不會產生新的commit，但commit id會變

* 1. 分支

git branch #查看目前有哪些分支

git branch -v #查看目前所有分支詳細訊息

git branch feature-new-user-status #建立分支feature-new-user-status

git branch -D feature-new-user-status #刪除分支feature-new-user-status

git branch -m abc cde #將分支abc更名為cde

* 1. 切換

git checkout feature-new-user-status #切換到分支feature-new-user-status

git checkout -b feature-new-user-status main #從main分支建立新分支feature-new-user-status，並切換到分支feature-new-user-status

git checkout a107cd6 #切換到commit(a107cd6)

git checkout -- filename #將此檔案的狀態還原到本機儲藏庫版本

git checkout HEAD #將所有變更還原到本機儲藏庫版本，HEAD指的是目前所在分支

* 1. 版本歷程

git show #顯示最新的一版的異動

git show ae9a #顯示commit 編號為 ae9a4a287ba6e21663e74ce8b27d74db9405ba22

git show HEAD #顯示此版的異動

git log #顯示出每個commit版本的comment及編號

git log --oneline #精簡log

git reflog #列出commit清單

git fetch #將遠端的分支清單sync到本端(注意只是清單,不包含檔案), 後續可以快速使用checkout切換branch

* 1. 還原

git reset HEAD^ #目前分支的commit退回前一版

git reset feature-new-user-status^ #分支feature-new-user-status的commit退回前一版

git reset HEAD^^ #目前分支的commit退回前兩版

git reset HEAD~2 #目前分支的commit退回前兩版

git reset 85e7e30 #目前分支的commit退回到commit(85e7e30)

git reset --mixed #git reset的預設模式，假設目前的commit是72a0315，執行git --mixed 85e7e30，則儲藏庫版本會退回到85e7e030，72a0315的異動內容會保留在working directory

git reset --mixed #git reset的預設模式，假設目前的commit是72a0315，執行git --mixed 85e7e30，則儲藏庫版本會退回到85e7e030，72a0315的異動內容會保留在working directory

git reset --hard #假設目前的commit是72a0315，執行git --hard 85e7e30，則儲藏庫版本會退回到85e7e030，index為空，working directory也變成85e7e030的內容

1. 合併

git merge main feature-new-user-status #將分支feature-new-user-status合併到分支main，碰到有衝突時用GUI會比較容易比對

1. 檢視差異

git diff #建議用GUI會比較容易比對

1. 連結遠端儲藏庫

git remote add origin http://twtp1apl115.local.wonderland.com.tw/pmic/contentsystem.git #將本機儲藏庫與遠端儲藏庫連結

1. 拉回遠端儲藏庫

git clone http://twtp1apl115.local.wonderland.com.tw/pmic/contentsystem.git #將遠端儲藏庫的所有資料拉回本機，並建立對應的本機版控，通常只有第一次要開別人專案時會用

git fetch #下載遠端儲藏庫上面有但我本機目前還沒有的內容下來，並且在本地端形成相對應的分支，不過，fetch 指令只做下載，並不會進行合併

git pull #下載遠端儲藏庫上面有但我本機目前還沒有的內容下來，並且在本地端形成相對應的分支，並進行合併

1. 推送遠端儲藏庫

git push origin feature-new-user-status #在分支feature-new-user-status執行此語法，代表將分支feature-new-user-status的commit記錄推送到遠端儲藏庫的分支feature-new-user-status

git push -u origin feature-new-user-status #在分支feature-new-user-status執行此語法，將分支feature-new-user-status的commit記錄推送到遠端儲藏庫的分支feature-new-user-status，並且將本機與遠端的分支feature-new-user-status連結，之後只要執行git push就可以自動推送

1. 進階指令
   1. 整理commit

git rebase #有點複雜請直接參考，<https://medium.com/starbugs/use-git-interactive-rebase-to-organize-commits-85e692b46dd>，另外在使用時請特別注意副作用，<https://medium.com/@gliyao/%E7%82%BA%E4%BB%80%E9%BA%BC%E6%88%91%E4%B8%8D%E7%94%A8-squash-65a2ddaa83fc>

* 1. 尋找Bug第一次出現的commit

git bisect #請直接參考，<https://medium.com/%E9%80%B2%E6%93%8A%E7%9A%84-git-git-git/%E5%BF%AB%E9%80%9F%E6%89%BE%E5%87%BA%E9%8C%AF%E8%AA%A4%E9%BB%9E-%E4%BD%BF%E7%94%A8git-bisect-e1c3338e8db1>

修改BUG如何進行(在GIT專案都建置好的前題下, 參考GIT安裝與第一個專案.docx)

* 1. 遠端建立分支  
     Branch Name: Bug請填Bug20220516mars、新功能請填feature20220516mars  
     Create From請選main  
     一張含有 文字 的圖片

     自動產生的描述 一張含有 文字 的圖片

     自動產生的描述
  2. 本地端取得遠端最新分支的metadata  
     git fetch
  3. 本地端的工作區與新分支簽出編輯, 可使用第3個指令確定工作區目前的分支是哪個  
     git checkout bug20220516Mars =>表示本地端的工作區切換到bug20220516mars  
     git pull =>類似tfs的取得最新的版本  
     git branch =>確認自己目前在修改的分支是正確的
  4. 打開專案, 修改完畢之後, 加入修改的檔案, 並將修改簽入+註解  
     git add .  
     git commit -m "修改開案單"
  5. 將修改的檔案push到遠端(不用特地指定哪個分支, 因為目前你的工作區就是與bug20220516Mars這個綁在一起)  
     git push -u origin  
     一張含有 文字 的圖片

     自動產生的描述
  6. 回到網頁上, 就會立馬看到一個視窗, 跟你說你剛push, 可以建立Merge Request,  
     一張含有 文字 的圖片

     自動產生的描述   
     一張含有 文字 的圖片

     自動產生的描述
  7. 接著下來就可以自己按下Merge (如果專案的更新負責人是你自己)  
     像雅文主要更新的人員是Jimmy, 我就會通知jimmy幫忙按下Merge

一張含有 文字 的圖片

自動產生的描述

* 1. 等Merge完成後, 就可以刪除自己本地端的Branch  
     git branch -d bug20220516Mars  
     一張含有 文字 的圖片

     自動產生的描述