git init 建立資料庫和主要分支

git init --help 該指令的網頁說明檔

git add poem.txt 將檔案增加到索引中

git status 目前git的狀態

git commit -m ‘這次作業的說明’ --author=’操作者姓名<email>’ 送進git資料庫

git commit --amend -m ‘新的作業說明’ 修改上述設定

gitk 啟動圖形檢視模式

exit 離開圖形檢視模式

git config -l 顯示git設定值

git config user.name ‘操作者姓名’ 指定使用者姓名到C組態

git config user.email ‘操作者email’ 指定使用者email到C組態

git config --global user.name ‘操作者姓名’ 指定使用者姓名到B組態

git config --global user.email ‘操作者email’ 指定使用者email到B組態

git config --system user.name ‘操作者姓名’ 指定使用者姓名到A組態

git config --system user.email ‘操作者email’ 指定使用者email到A組態

git config --unset user.name 移除指定的組態

git config alias.指定別名 ‘正式的指令和選項’ 新增別名指令

git config --unset alias.指定別名 移除別名指令

git config --global core.editor notepad 以記事本取代vi編輯器

git diff 將資料庫或者索引檔案與資料夾檔案做比對

git difftool 使用自訂工具比對程式

git config --global diff.tool kdiff3 定義工具名稱

git config --global difftool.kdiff3.cmd \ 延續到下一行

“’C:\Users\blue\Desktop\KDiff3Portable\X-KDiff3.exe’ \”\$LOCAL\” \”\$REMOTE\”” 新增檔案位置

git config --global difftool.prompt false 關閉提示功能

touch .gitignore 新增可忽略資料夾(可將資料夾或者檔案做忽略追蹤動作)

git rm --cached 檔案名稱 刪除索引的資料

git reset HEAD 檔案名稱 刪除資料庫資料

git show HEAD 顯示最新節點資料

git tag 自訂標籤名稱 commit 節點標籤 自訂標籤名稱

git tag -d commit 節點標籤 移除自訂標籤

git reset 選項 commit 節點標籤 回復到某一節點狀態

--soft 資料庫變更

--mixed 索引變更 (預設)

--hard 全部都會變更

git checkout commit 或標籤 或檔案 取出資料庫檔案

git grep ‘要找得字串’ commit 節點 取出字串檔案

git checkout master 回到主分支狀態

git blame -L 起始行,結束行 檔案名稱 顯示字串編輯者資訊

git mv 原檔案名稱 新檔案名稱 變更資料庫檔案名稱

git stash save 暫存目前工作狀態

git stash list 顯示目前暫存資料

git checkout 識別碼 取出暫存資料

git stash pop 合併目前資料夾檔案

git gc 清理資料庫

--aggressive 完整清理

--auto 判斷清理

--no-prune 整理清理

git branch 分支名稱 從最新資料長出分支

git branch 分支名稱 commit節點標籤 從節點長出分支

git checkout 分支名稱 切換操作分支

git branch 顯示所有分支

gitk --all 顯示所有分支流程圖

git branch -d 要刪除的分支名稱 刪除分支

git branch -D 要刪除的分支名稱 強制刪除分支

git branch -m 新的分支名稱 更改分支名稱

git merge 分支名稱 將分支合併目前分支

git merge --no-ff 分支名稱 留下合併紀錄

git merge --abort 放棄這次合併(衝突時使用)

git mergetool 打開比kdiff3做修正

git cherry-pick commit標籤 將標籤合併目前標籤(形成新標籤)

git cherry-pick --abort 放棄這次合併

git revert HEAD 回到指定節點前一個節點

git reset HEAD 回到指定節點

git rebase master 合併原理同merge

git reflog HEAD 或是任何分支名稱 查詢分支歷史紀錄

git reset --hard HEAD@{3} 回復rebase動作

git add . 將新增檔案、修改檔案加入索引

git add -u 將修改檔案、刪除檔案加入索引

git add -A 將新增檔案、修改檔案、刪除檔案加入索引

git checkout -b 新分支名稱 同下指令

------------------------------------------

git branch 新分支名稱

git checkout 新分支名稱

------------------------------------------

git init --bare Git 建立遠端資料庫

git clone 複製git資料庫

設定遠端Github資料庫上傳

ssh-keygen 設定SSH金鑰

git clone 遠端ssh地址 資料夾名稱 複製遠端資料庫到本地資料夾

git push origin 分支名稱 上傳到遠端伺服器

更新遠端Github資料庫到本地資料夾

git fetch origin master:temp 下載遠端資料庫最新資料

git diff temp 比較最新資料和本地資料

git merge temp 合併temp分支到master分支

git branch -d temp 刪除temp分支

衝突發生

|  |
| --- |
| error: failed to push some refs to 'git@github.com:awdr74100/Git\_test.git'  hint: Updates were rejected because the tip of your current branch is behind  hint: its remote counterpart. Integrate the remote changes (e.g.  hint: 'git pull ...') before pushing again.  hint: See the 'Note about fast-forwards' in 'git push --help' for details.    Blue@DESKTOP-GLQF21T MINGW64 ~/Desktop/GitHub (master) |

解決方法：

先拉在推，difftool 檢視。

push -f 強推

更改commit紀錄內容

git log --oneline 查看歷史紀錄

git commit --amend -m “更改內容” 修改最後一次提交名稱

新增分支上傳GitHub

git branch gh-pages 本地端新增gh-pages分支

git checkout gh-pages 切換到gh-pages分支

git push --set-upstream origin try1 將本地 ty1分支上傳到github(方法一)

git push -u origin try1 將本地 ty1分支上傳到github(方法二)

git checkout -b try1 創建分支並切換到新分支

git push origin :try 刪除遠端分支

本地端增加遠端儲存庫

git remote add origin 遠端ssh地址 增加遠端儲存庫

修改remote遠端數據庫位置

git remote set-url origin 遠端ssh地址 修改遠端儲存庫位置

刪除remote遠端數據庫位置

git remote set-url origin 遠端ssh地址 刪除遠端儲存庫位置