

# GitHub 及 GitHub Desktop 使用教學

## 目錄

一、簡介.....	2
二、使用步驟.....	2
1. 申請 GitHub 帳號.....	2
2. 加入 ai-lab 專案.....	2
3. 下載 GitHub Desktop.....	2
4. 將 ai-lab 專案 clone (克隆) 至本機.....	2
三、原理及術語.....	2
四、GitHub Desktop 介紹.....	3
五、提交修改.....	3
1. Changes.....	4
2. 檔案內容.....	4
3. Git commit message header (提交訊息標題).....	5
4. Git commit message description (提交訊息描述).....	5
5. 提交本次修改.....	5
6. Push.....	5
7. Create Pull Request.....	6

## 一、簡介

Git：分散式版本控制系統 (Decentralized Version Control System)

GitHub：提供 Git 代管服務的平台

GitHub Desktop：GitHub 桌面版

## 二、使用步驟

### 1. 申請 GitHub 帳號

GitHub 網址：<https://github.com/>

### 2. 加入 ai-lab 專案

向專案持有者申請 Collaborators 加入連結

### 3. 下載 GitHub Desktop

GitHub Desktop 下載網址：<https://desktop.github.com/>

### 4. 將 ai-lab 專案 clone (克隆) 至本機

開啟 GitHub Desktop

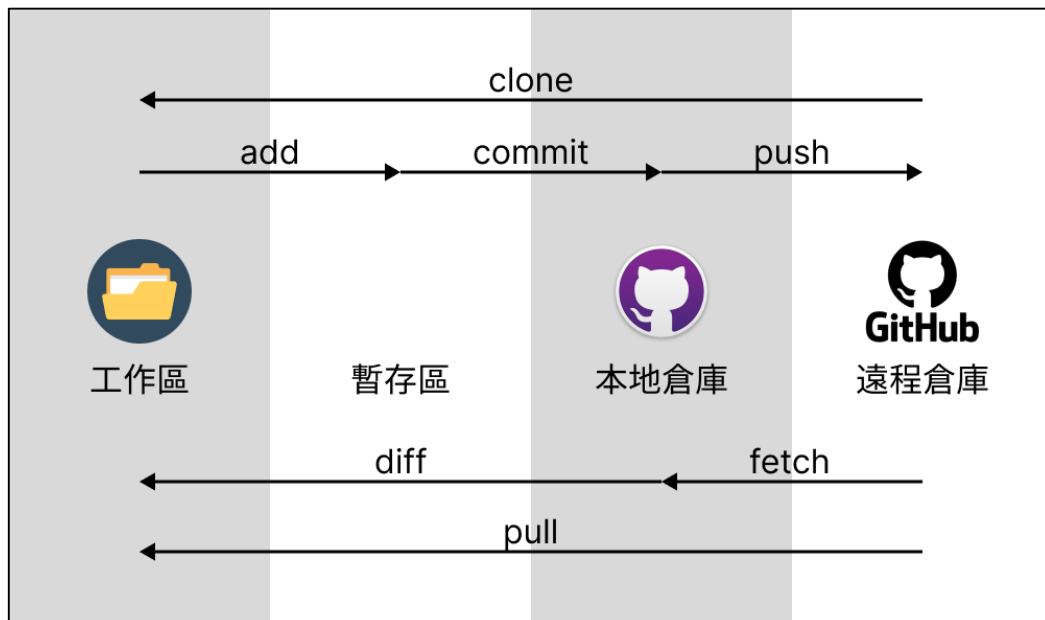
選擇 Clone repository

找到 ai-lab 專案

選擇要存放檔案的本機資料夾 (必須為空資料夾)

點擊 clone

## 三、原理及術語



clone：第一次獲取遠程倉庫時使用

add：使用 GitHub Desktop 可以省略此步驟

commit：提交修改至本地倉庫

push：推送本地倉庫的所有 commit 至遠程倉庫

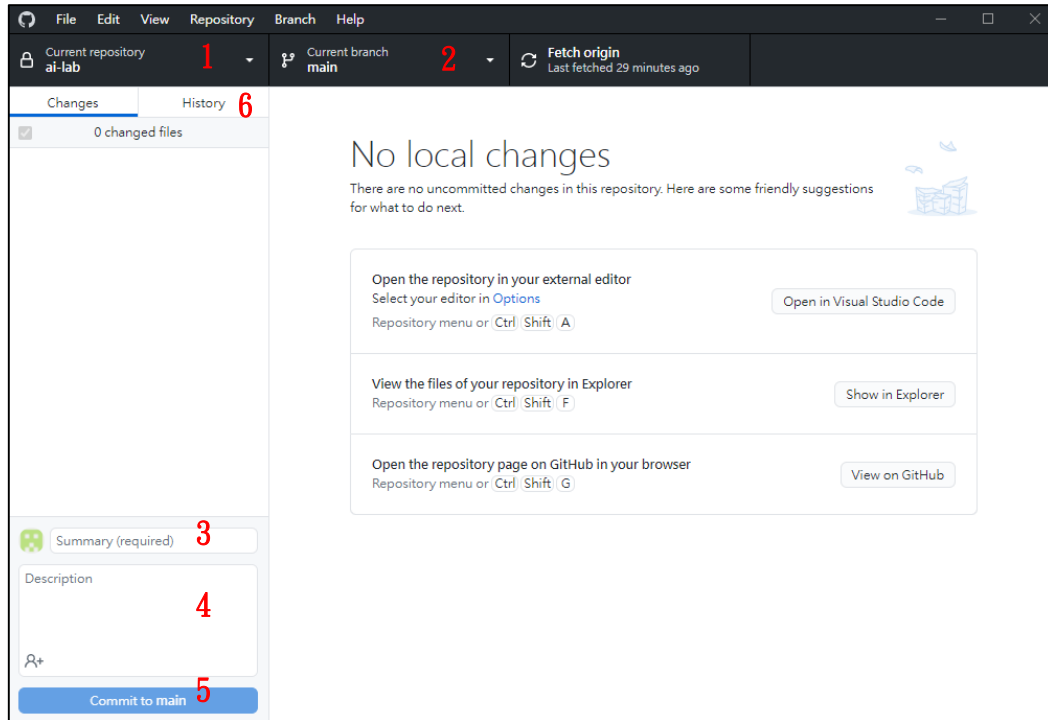
pull：遠程倉庫的內容覆蓋本機的檔案 (謹慎使用)

fetch：將遠程倉庫的內容獲取至本地倉庫

diff：對本地倉庫與工作區的內容進行差異對比

在 push 前可以先使用 fetch 以減少檔案發生 conflict（衝突）的情況

#### 四、GitHub Desktop 介紹



1. Current repository（當前專案）

2. Current branch（當前分支）

盡量避免在 main 分支上進行改動

選擇不同分支時，本機資料夾裡的檔案也會跟著改變成該分支的版本

3. Git commit message header（提交訊息標題）

此欄位為**必填**

4. Git commit message description（提交訊息描述）

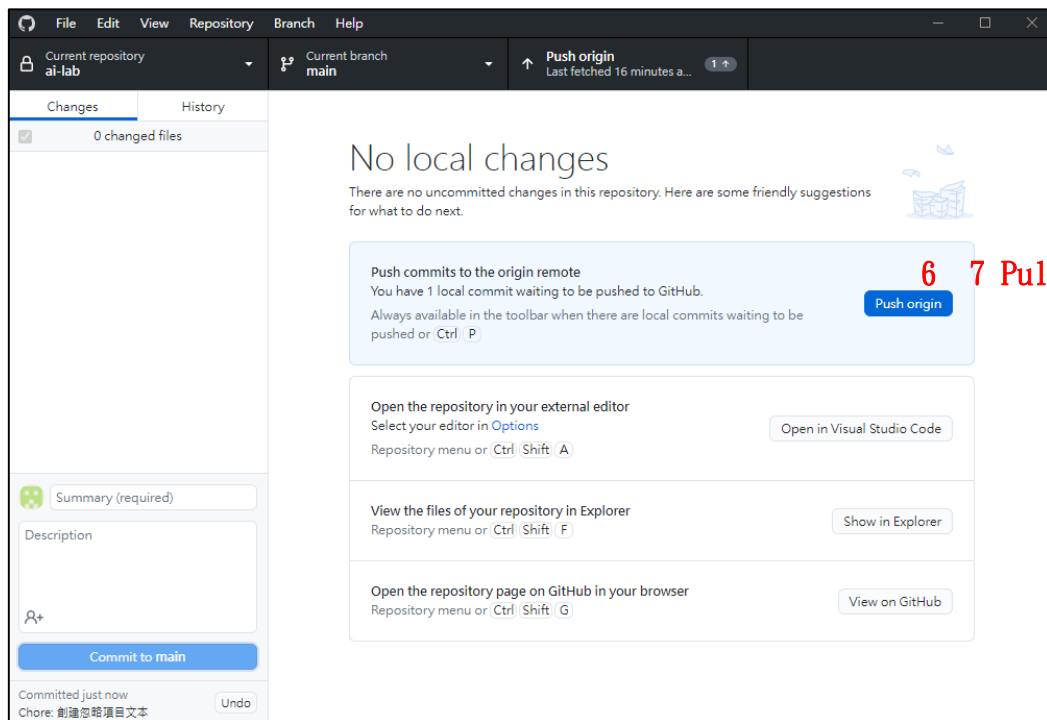
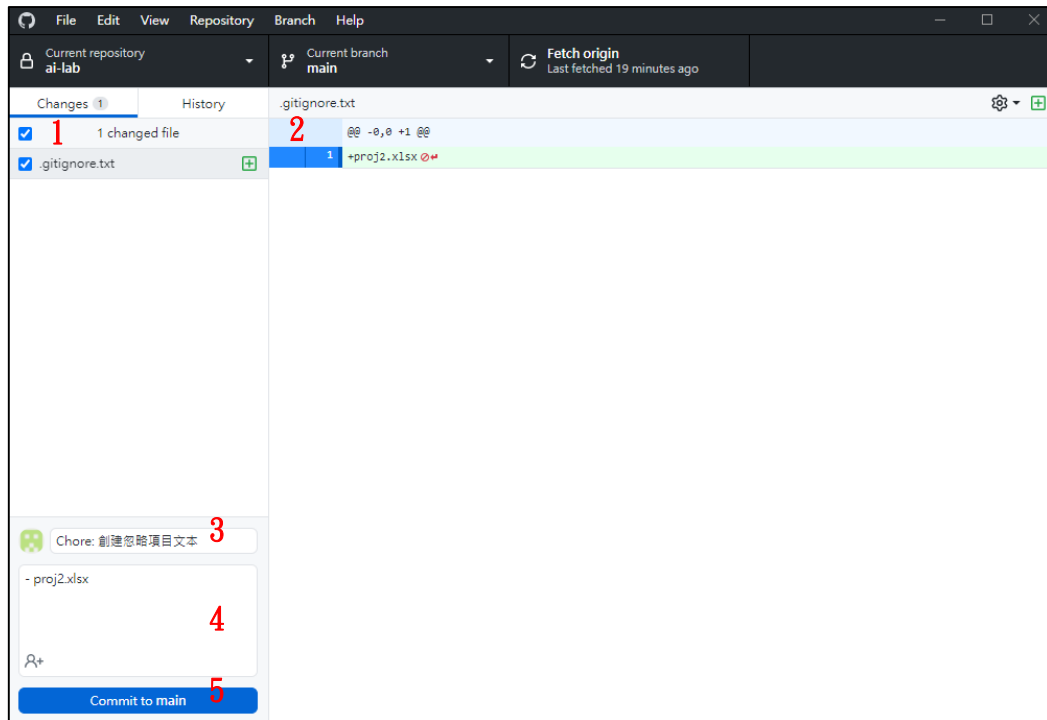
此欄位為**選填**

5. Commit（提交）

6. History（提交的歷史紀錄）

可以在此回復到之前的版本

#### 五、提交修改



## 1. Changes

左側欄位會顯示有修改的檔案的名稱

如果要忽略單個檔案，讓該檔案不被提交，可以對檔案點擊右鍵，選擇第一個選項 Discard changes

## 2. 檔案內容

右側欄位會顯示修改的檔案的內容

綠底表示增加行，紅底表示刪除行

點擊右上角的齒輪，在 Diff display 中選擇 Split 可以切換顯示模式

### 3. Git commit message header (提交訊息標題)

此欄位為**必填**

格式：**TYPE** + 冒號 (半形) + 空格 (半形) + 主旨

例一：Chore: 創建忽略項目文本

例二：Fix: 兩個單位題修正

主旨無特別規定，盡量簡潔即可

TYPE 類型遵循以下表格分類：

類型	說明	有無程式碼改動
Feat	新功能。	有
Fix	錯誤修正。	有
Docs	更新文件。	沒有
Style	程式碼格式調整。	沒有
Refactor	重構。	有
Test	測試、新增測試。	沒有
Chore	更新專案設定、版本號。	沒有
Revert	回復到之前的版本。	有

### 4. Git commit message description (提交訊息描述)

此欄位為**選填**

可以為多行

格式：減號 (半形) + 空格 (半形) + 描述

例一：- proj2.xlsx

例二：- 498.txt

- 505.txt

描述無特別規定，盡量簡潔即可，通常為註記

### 5. 提交本次修改

點擊之後 Git 就會將本次修改紀錄到**本地倉庫**中，稱為一個 Commit

可以提交多個 Commit 再推送至遠程倉庫

### 6. Push

點擊之後 Git 會將**本地倉庫**中**所有** Commit 上傳至**遠程倉庫**的指定分支  
(不建議直接將修改推送至 main 分支)

## 7. Create Pull Request

在非 main 分支時會出現 Create Pull Request 選項

想要將目前分支與其他分支合併時使用

點擊之後會跳轉至網頁版的 GitHub Pull Request 介面

此處可以查看目前分支與想合併的分支之間的差異以及是否有衝突

選擇下方的 Pull Request 就可以申請合併分支了