

Git與GitHub 軟體創作歷程管理與協同開發

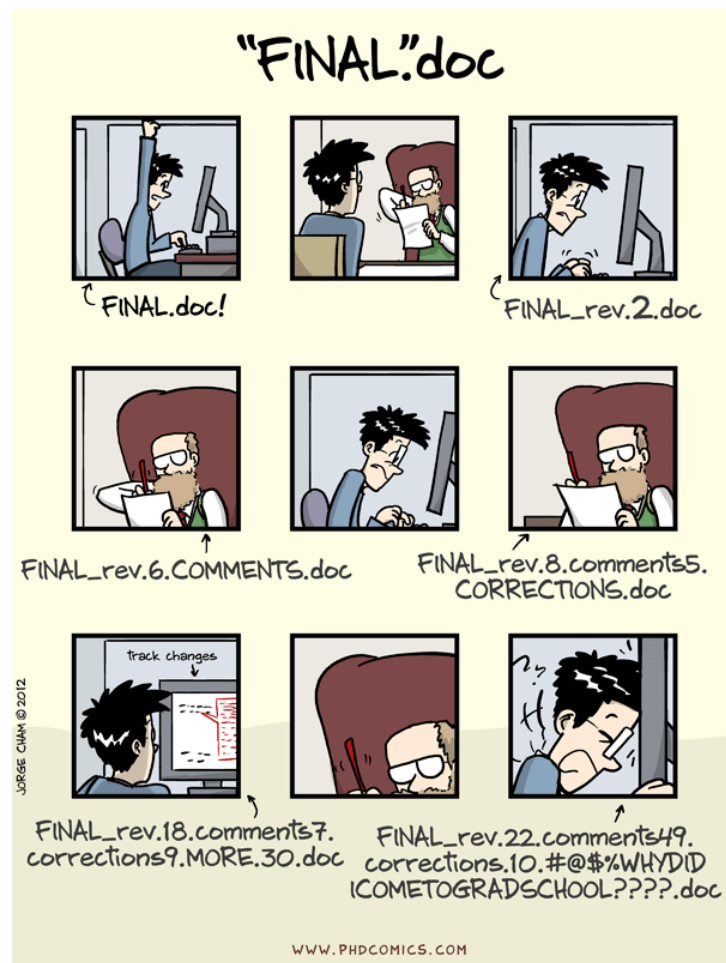
原始教材編撰(SourceTree) :
逢甲大學資訊工程學系 陳錫民 教授

課程改編與講授(GitHub Desktop) :
臺灣海洋大學資訊工程系 馬尚彬 教授

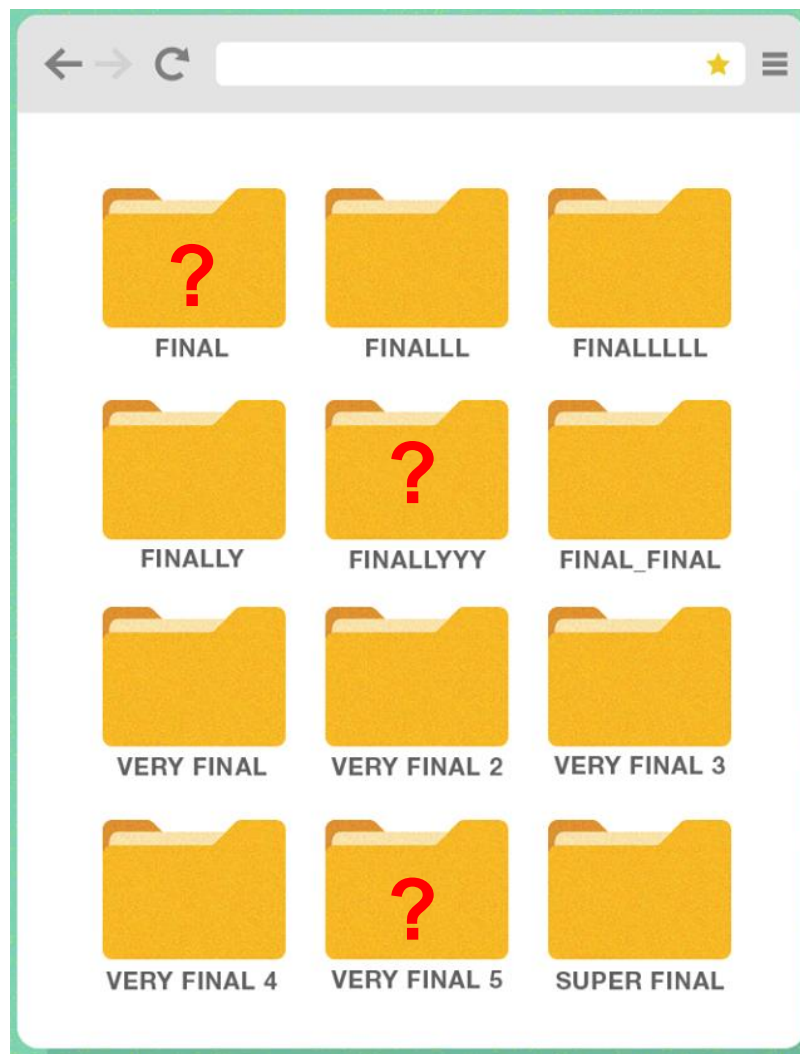
2024/5/9

- 為什麼需要版本控制？

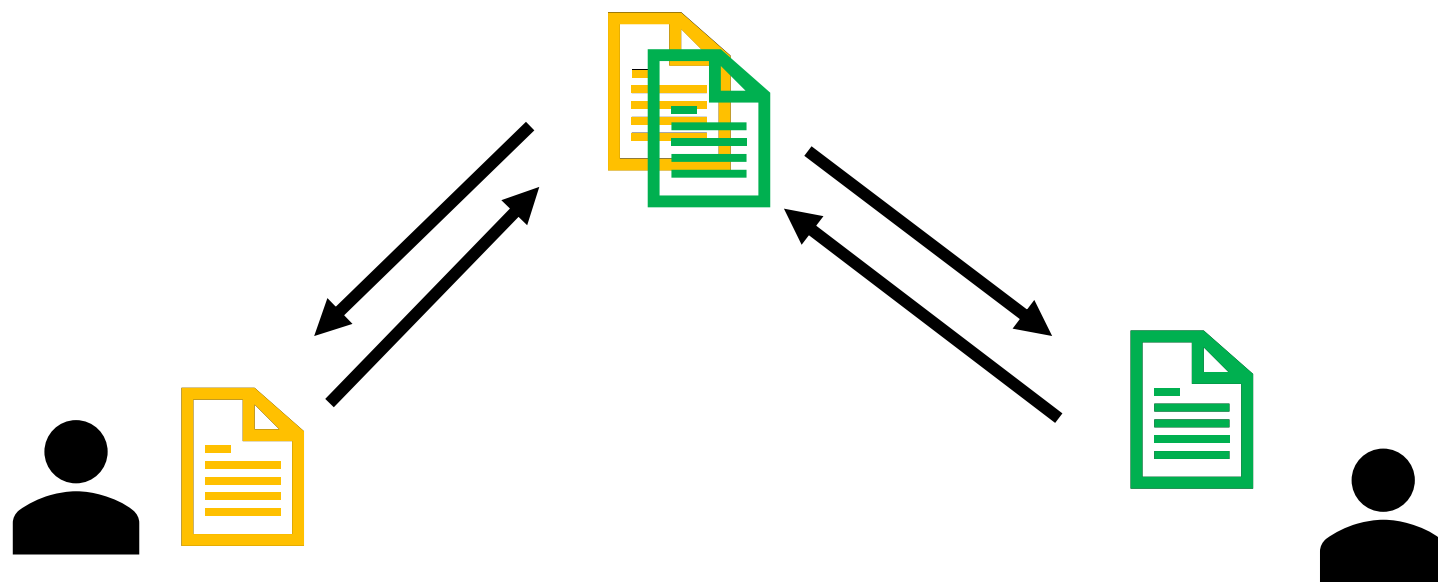
當你撰寫報告時，如果你想備份，你會怎麼做？



- 會發生什麼事？
 - 很多版本檔案



- 會發生什麼事？
 - 跟別人合作時.....

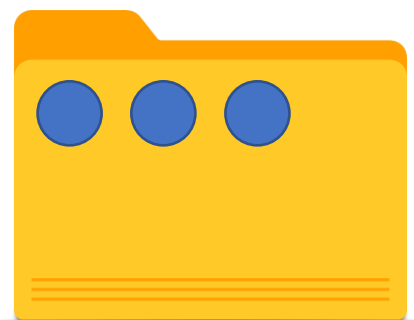


- 為什麼要版本控制

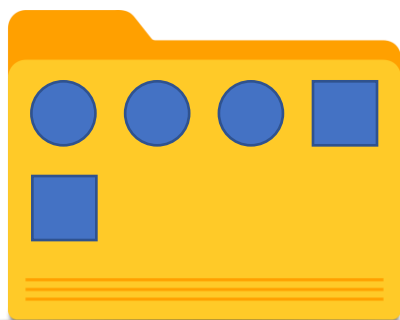
- 「凡走過必留下痕跡」
 - 追蹤歷程
 - 改了東西，不會改不回來
- 「三個臭皮匠勝過一個諸葛亮」
 - 大家一起改，不會互相干擾
 - 大家一起改，還能清楚知道對方改了什麼

- 什麼是「版本」？

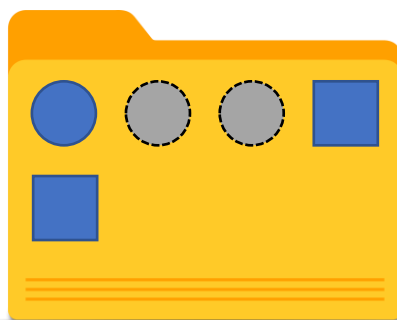
版本 1



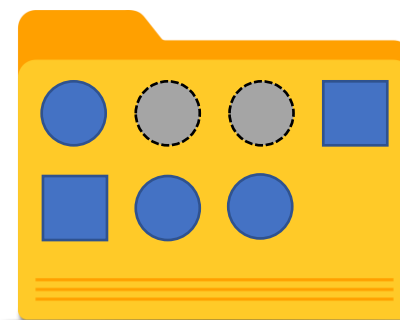
版本 2



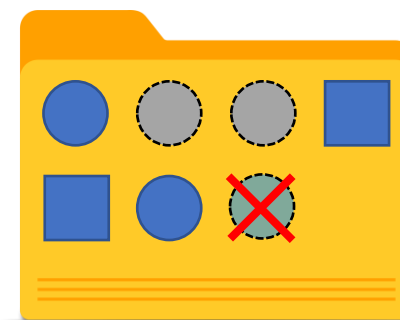
版本 3



版本 4



版本 5



Day 1

Day 2

Day 3

Day 4

Day 5

- 版本控制

- 什麼類型的內容適合版本控制
 - 程式原始碼
 - 測試程式碼
 - 伺服器的設定檔
 - 文件
 - 書籍
 - 網站內容



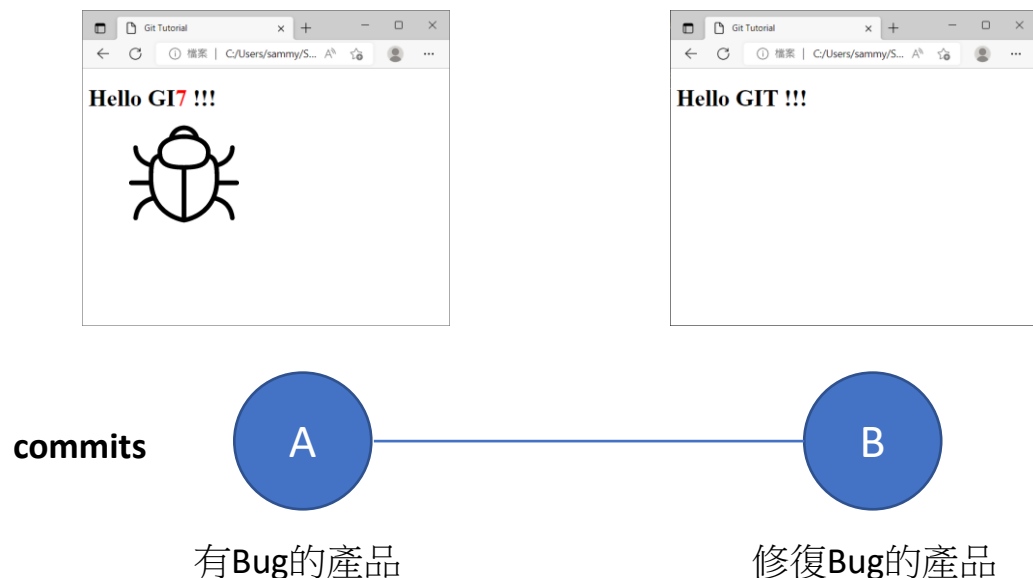
- **Git是什麼？**

- 分散式的版本控制系統
- 開源軟體(open-source software)
 - 活躍的開源社群
 - 豐富的生態系支持
- 適合於多種不同類型的專案與工作流程
 - Linux作業系統開發即運用Git進行管理
- 是大部分 IT公司都會運用的工具



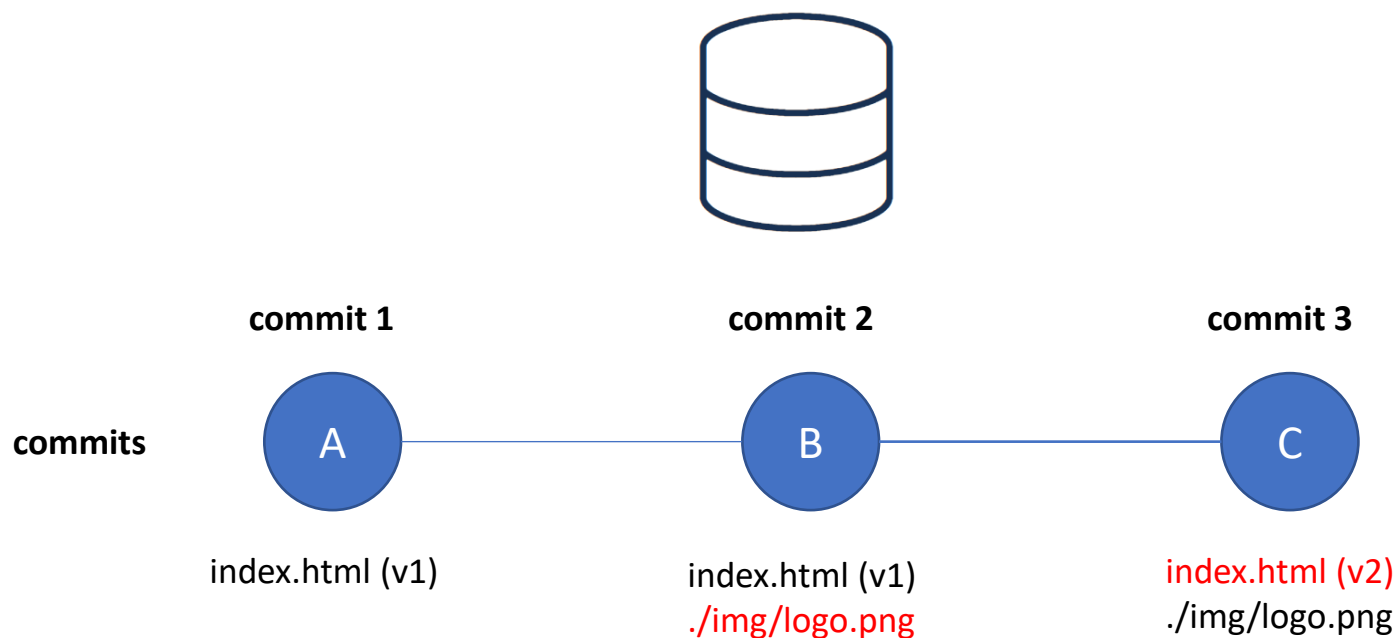
• Git Commits

- Git記錄專案在不同時間的狀態，構成一個專案開發的**歷程**
- 專案中每個記錄版本稱為一個交付(commit)
- 每一次commit可視為專案在一個給定的時間點上的快照(snapshot)



- **Git Repository (Repo) 是什麼?**

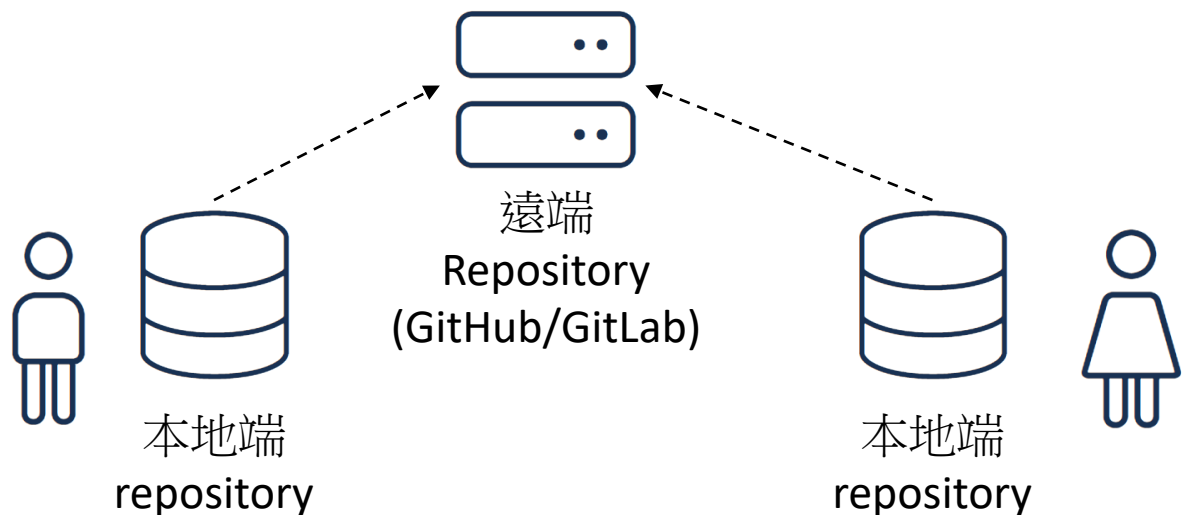
- 一連串的專案快照(snapshot)，包含多個交付(commit)



- 分散式版本控制系統

- 分散式版本控制系統具有三特性

- 每一位使用者皆有自己本地端的專案歷程
- 使用者可以離線使用版控系統
- 可以方便地進行儲存庫(Repository)內容的整合工作



- 遠端儲存庫(Remote Repository)

- 遠端儲存庫代管服務

- GitHub

- <https://github.com/>
 - 最多人使用的平台
 - 發佈的repo可成為你的履歷
 - 可以在上面找到很多很棒的開源程式碼，甚至電子書

- GitLab

- <https://about.gitlab.com/>

- Bitbucket

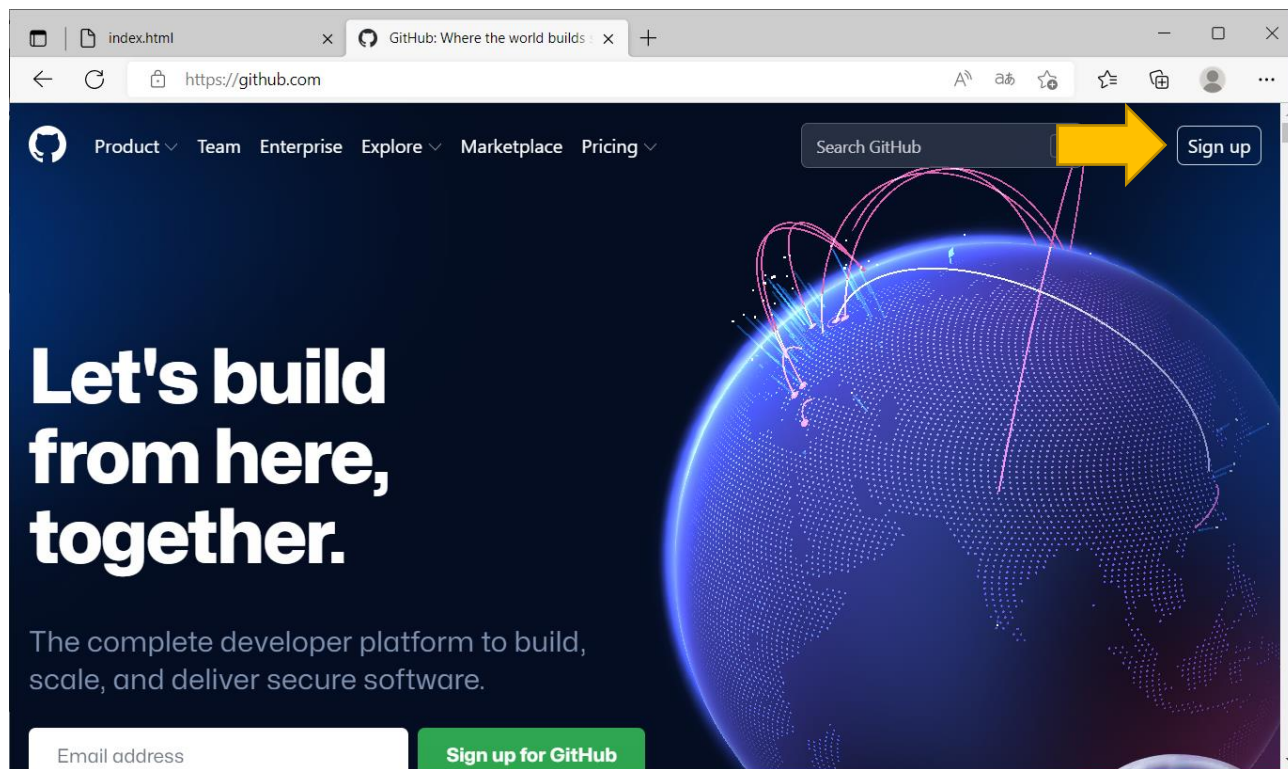
- <https://bitbucket.org/>

-

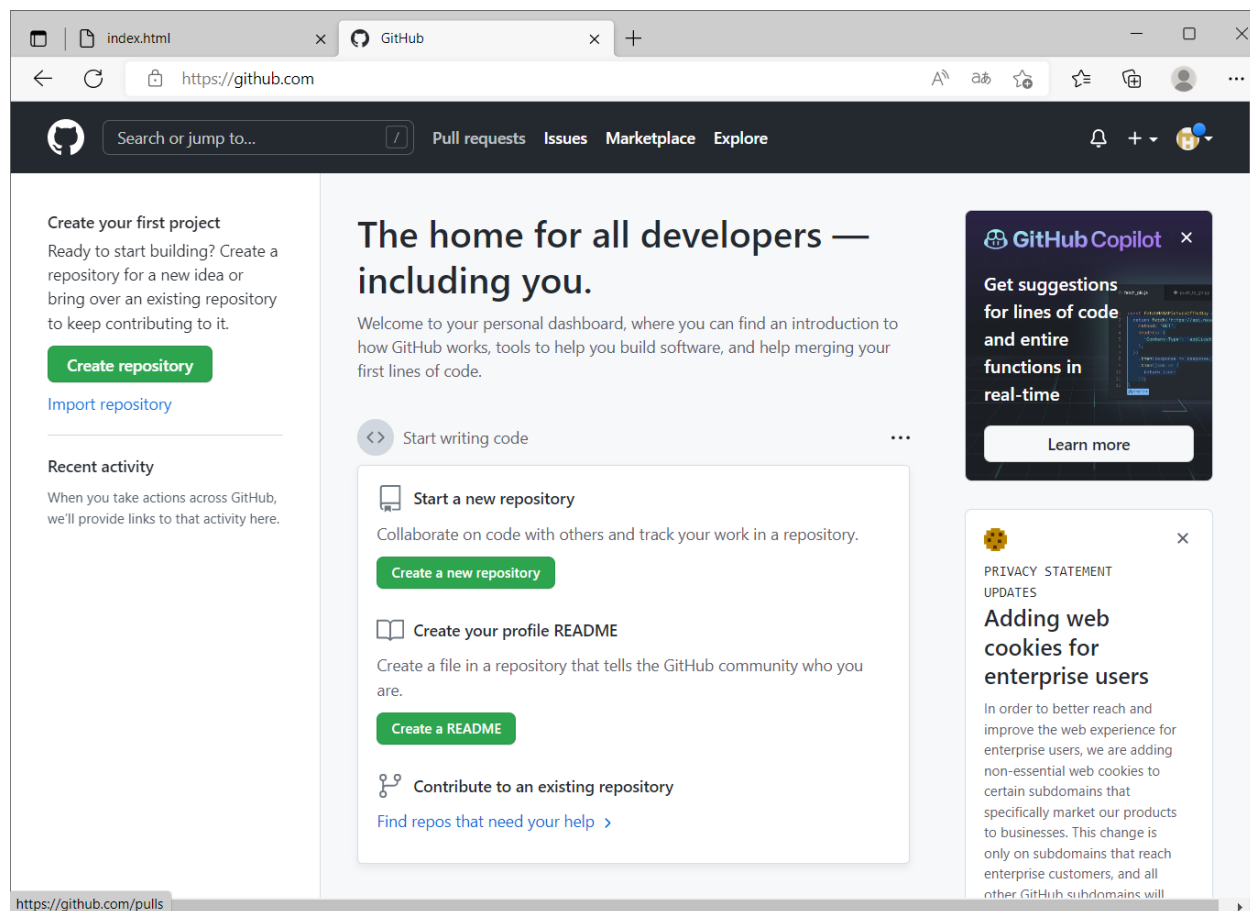
- GitHub遠端儲存庫

- 申請GitHub帳號

- <https://github.com/>

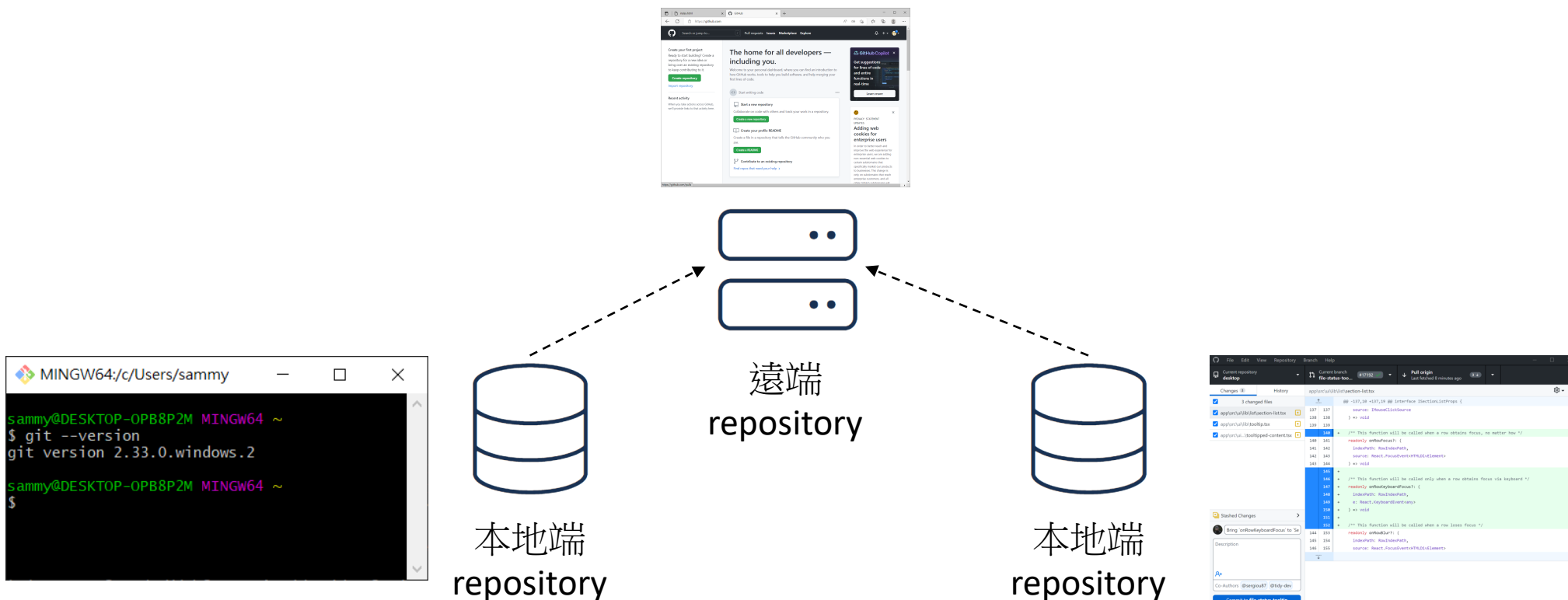


- GitHub遠端儲存庫
 - 註冊成功

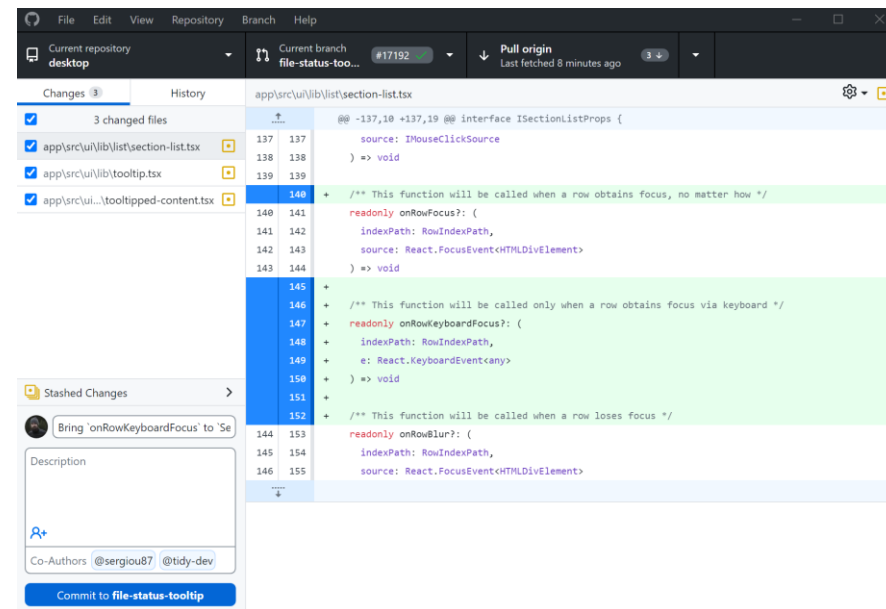


• Git工具

- 指令(command line)模式與使用者介面(GUI)模式

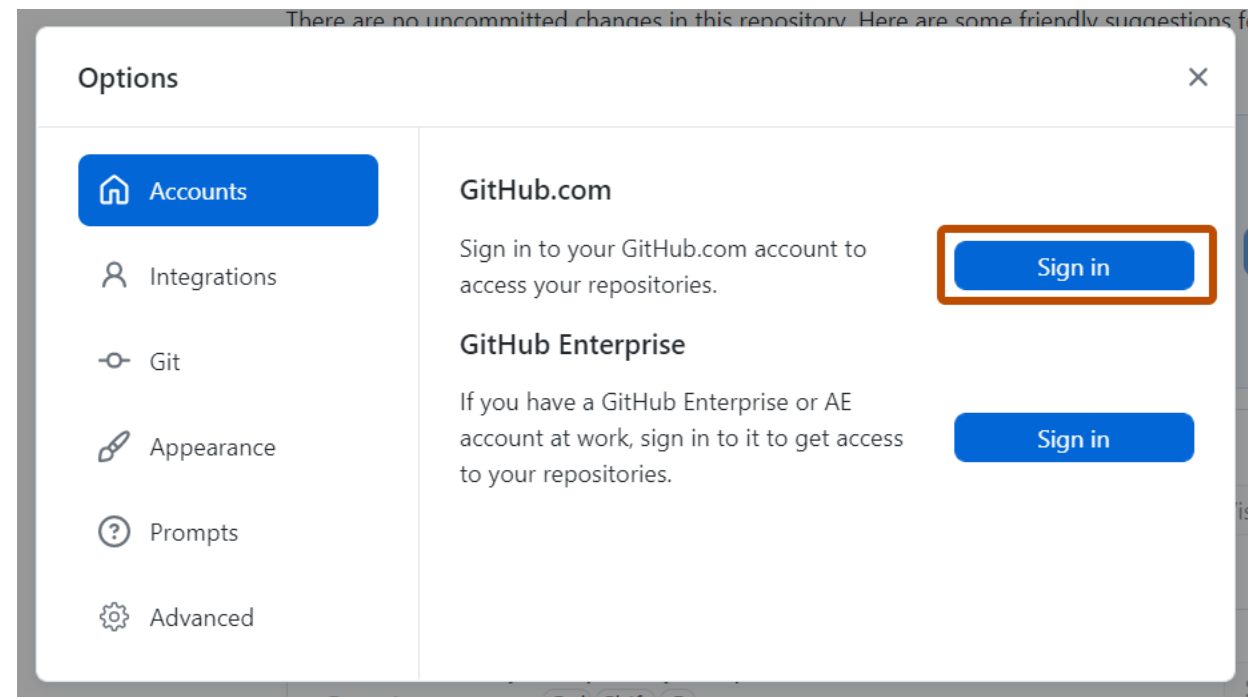


- 安裝圖形化Git客戶端工具
 - GitHub Desktop
 - <https://desktop.github.com/>
 - 原生支援與GitHub之串連
 - SourceTree
 - <https://www.sourcetreeapp.com/>
 - 其他圖形化Git客戶端工具
 - <https://git-scm.com/downloads/guis/>



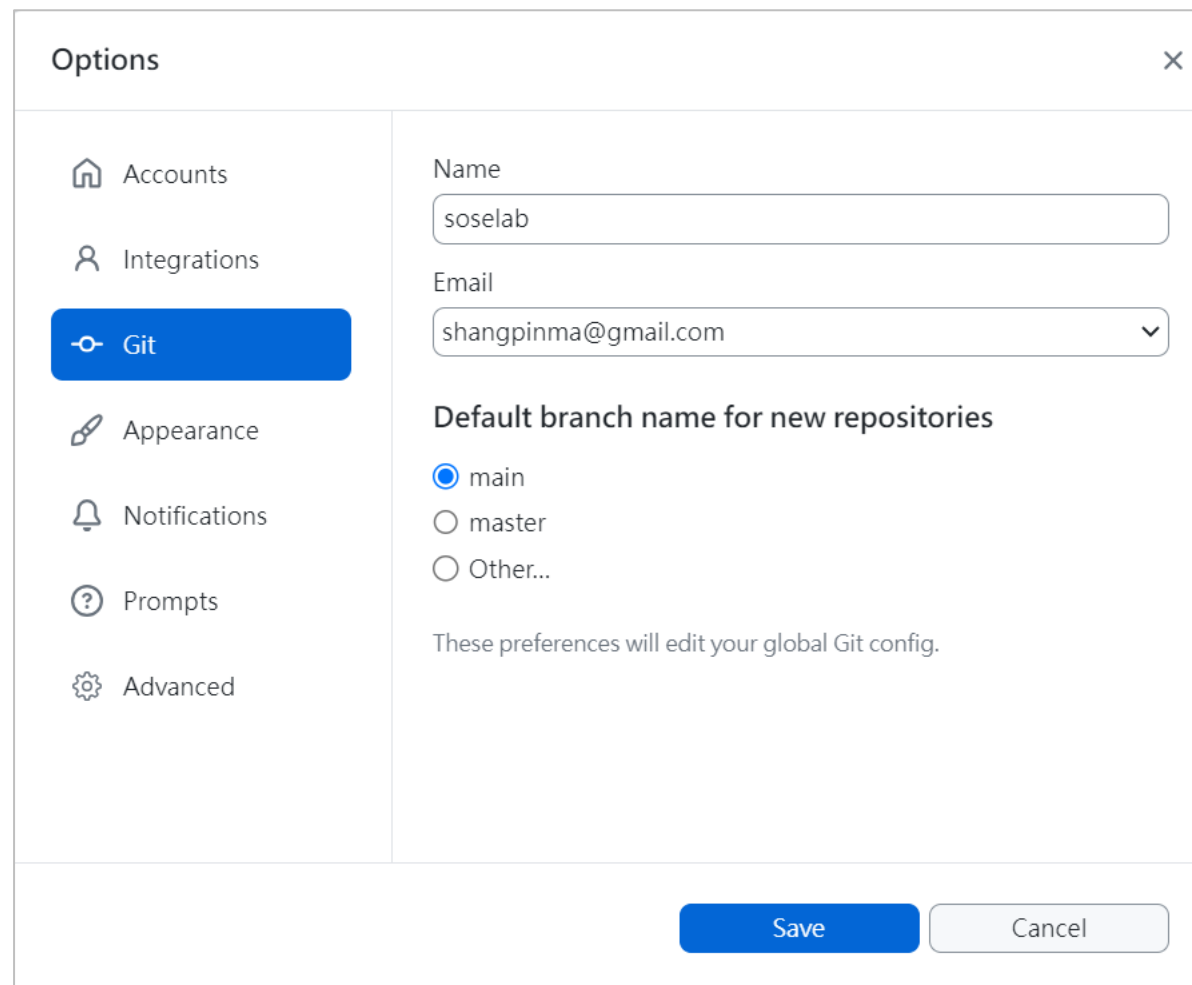
- **GitHub Desktop使用者帳號設定**

- 安裝 GitHub Desktop 後，可以使用 **GitHub 帳號** 對應用程式進行驗證。
 - 透過身份驗證，你可以連接到 GitHub 上的遠端儲存庫。
- 操作方式：[File]->[Options]->[Accounts]，並透過 GitHub 帳號登入。
 - 如果還沒有 GitHub 帳號，請立即申請：<https://github.com/>。



• GitHub Desktop使用者帳號設定

- 在"Git"頁籤的設定部分，請設定與GitHub帳號相同的Email，以確保之後的commit之作者資訊正確。
- 此外，預設分支名稱請選擇"main"，讓Git與GitHub都具有相同的預設分支名稱。
 - 同學們可於此門課程後再去了解分支的運用。



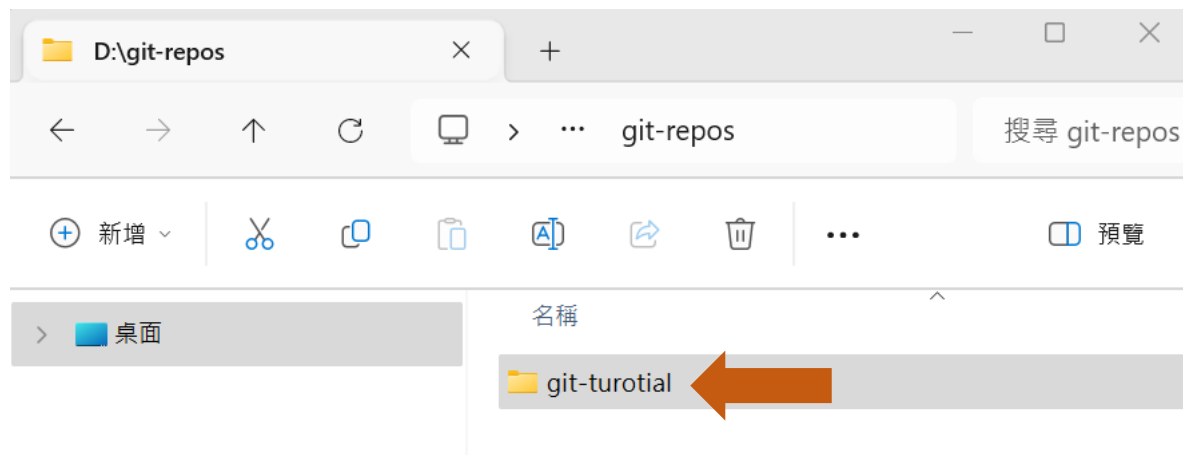
The screenshot shows the 'Options' dialog box in GitHub Desktop, with the 'Git' tab selected. The left sidebar lists various settings categories: Accounts, Integrations, Git (selected), Appearance, Notifications, Prompts, and Advanced. The main area displays the following configuration options:

- Name:** A text input field containing 'soselab'.
- Email:** A dropdown menu showing 'shangpinma@gmail.com'.
- Default branch name for new repositories:** Three radio button options: 'main' (selected), 'master', and 'Other...'.

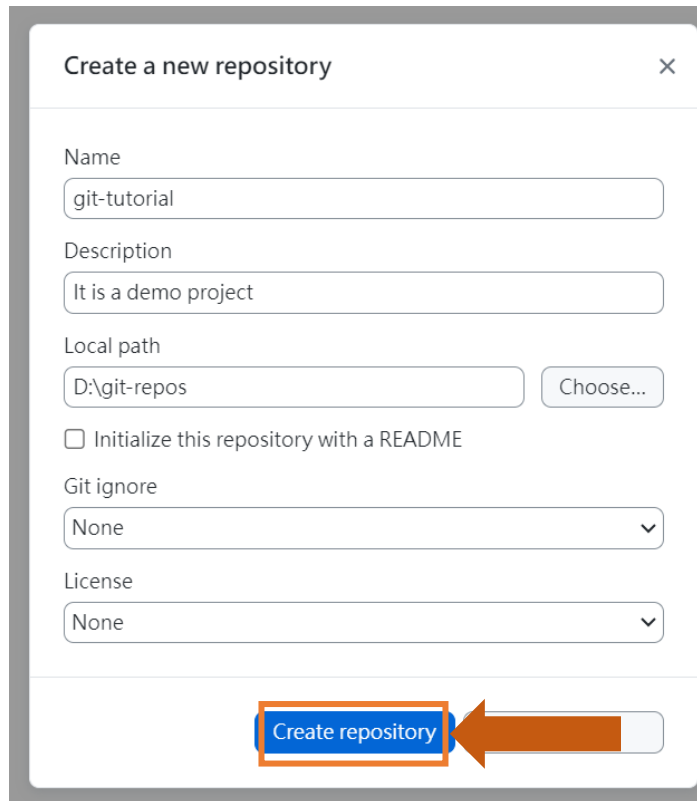
At the bottom of the dialog, there is a note: 'These preferences will edit your global Git config.' and two buttons: 'Save' and 'Cancel'.

- 建立本地端儲存庫(Local Repository)

- 開啟檔案總管，建立版控專案根目錄
 - 用以管理所有的Git專案
 - 例如：目錄 *git-repos*
- 接著建立版控專案目錄(Working Tree)
 - 例如：目錄 *git-tutorial*



- 建立本地端儲存庫(Local Repository)
 - 透過GitHub Desktop設定版控工作目錄(working tree)
 - [File]->[New Repository]



Create a new repository

Name
git-tutorial

Description
It is a demo project

Local path
D:\git-repos Choose...

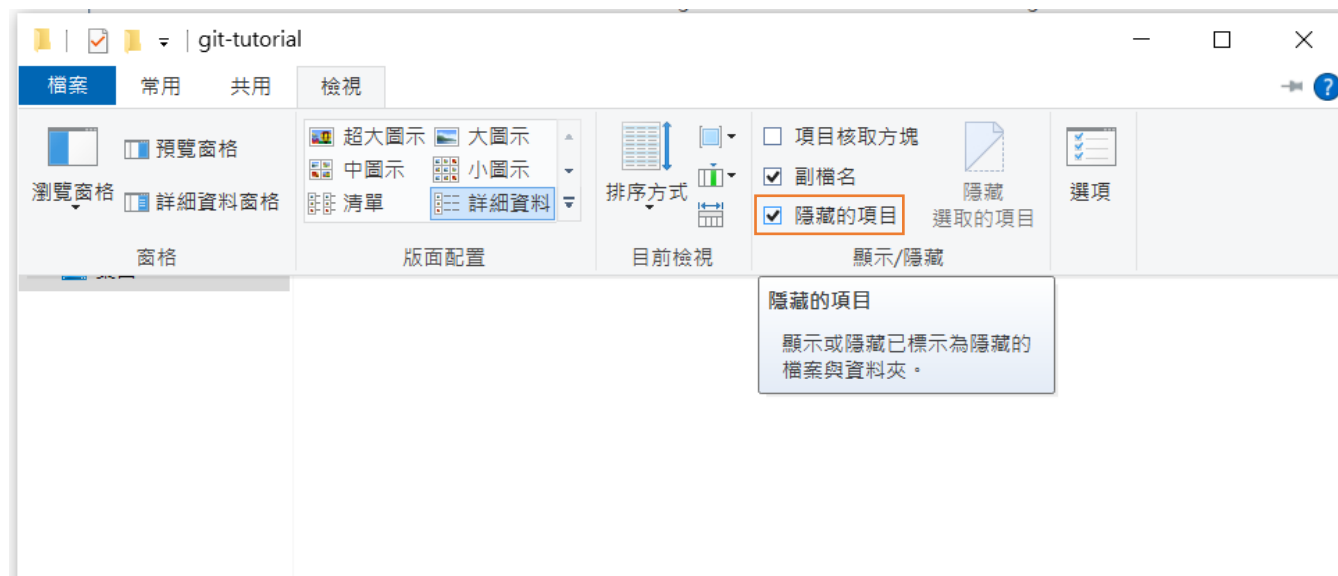
☐ Initialize this repository with a README

Git ignore
None

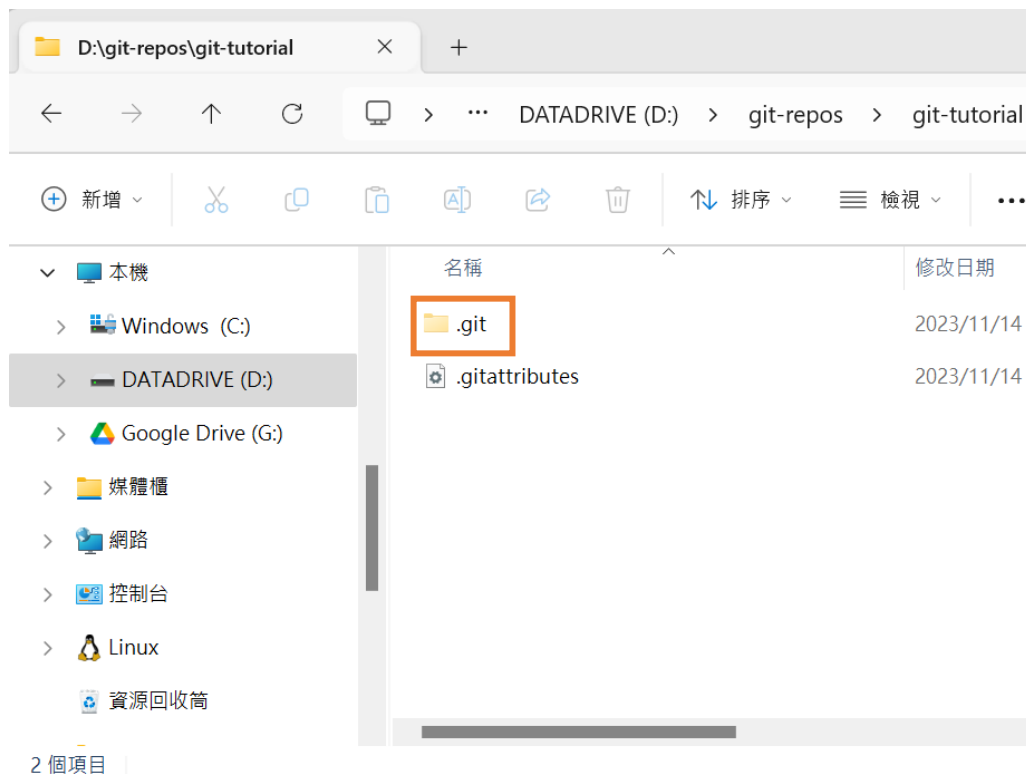
License
None

Create repository

- 建立本地端儲存庫(Local Repository)
 - 開啟檔案總管，開啟顯示隱藏項目



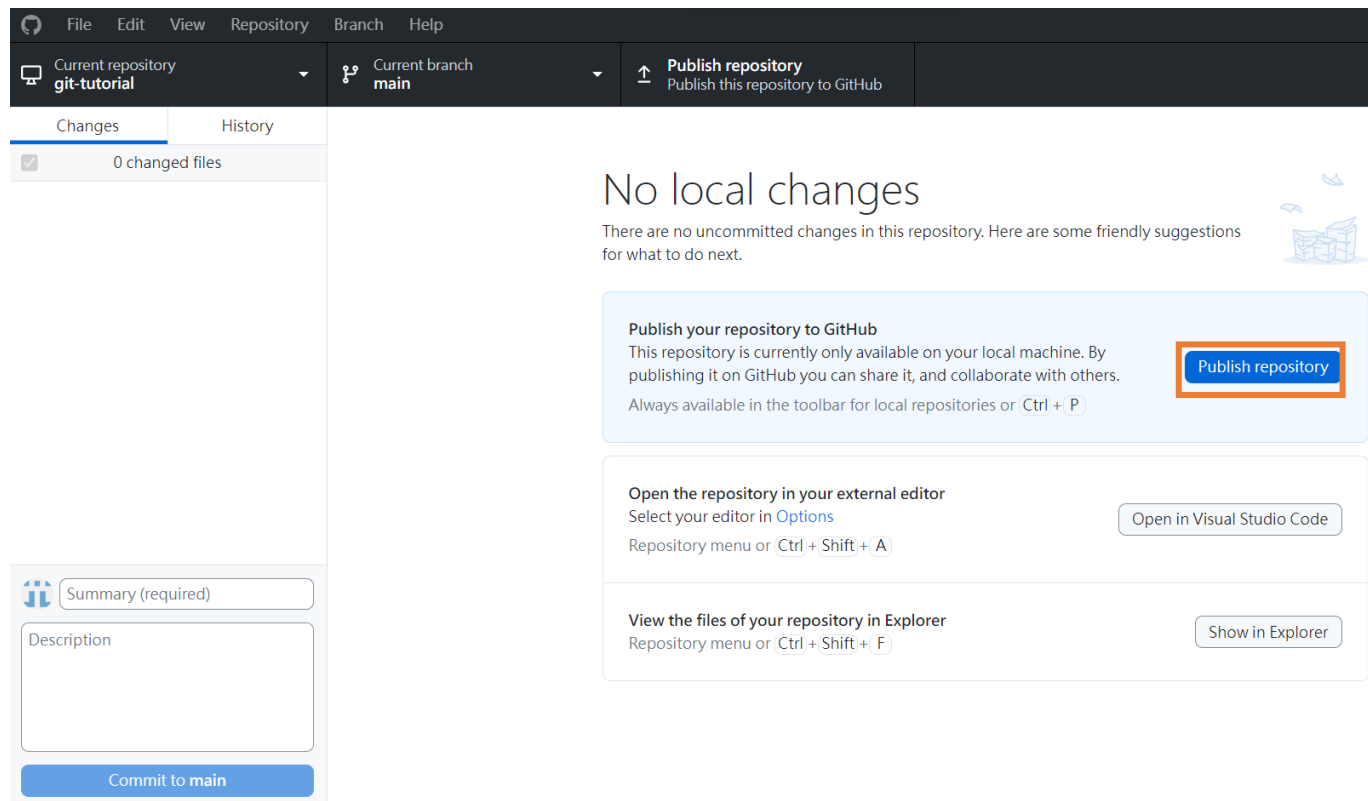
- 建立本地端儲存庫(Local Repository)
 - 開啟檔案總管，查看本地端存庫(.git)



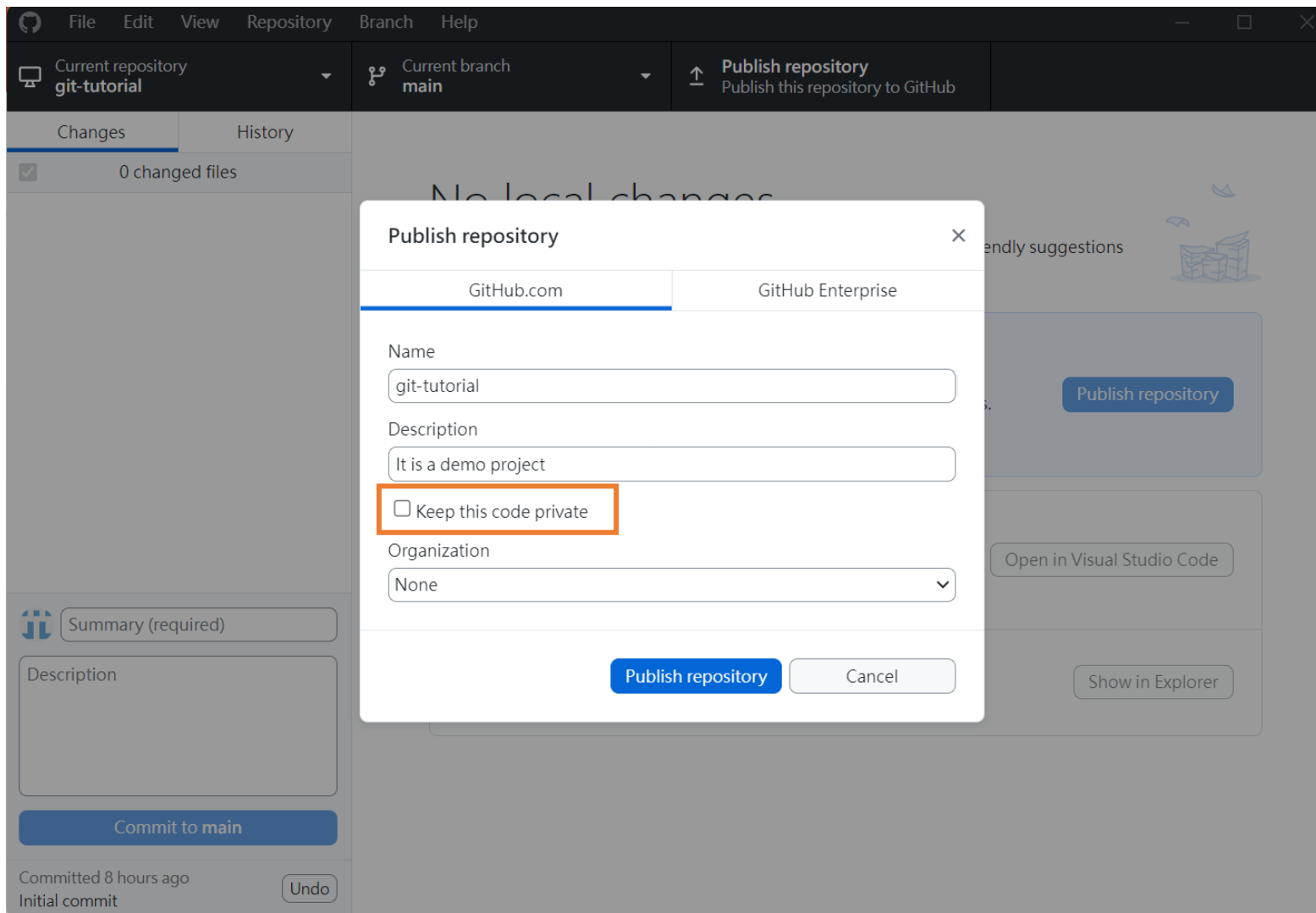
A screenshot of the contents of the .git directory. The path shown is > ... git-repos > git-tutorial > .git >. The table lists various files and folders with their modification dates.

名稱	修改日期
hooks	2023/11/14
info	2023/11/14
logs	2023/11/14
objects	2023/11/14
refs	2023/11/14
COMMIT_EDITMSG	2023/11/14
config	2023/11/14
description	2023/11/14
HEAD	2023/11/14
index	2023/11/14

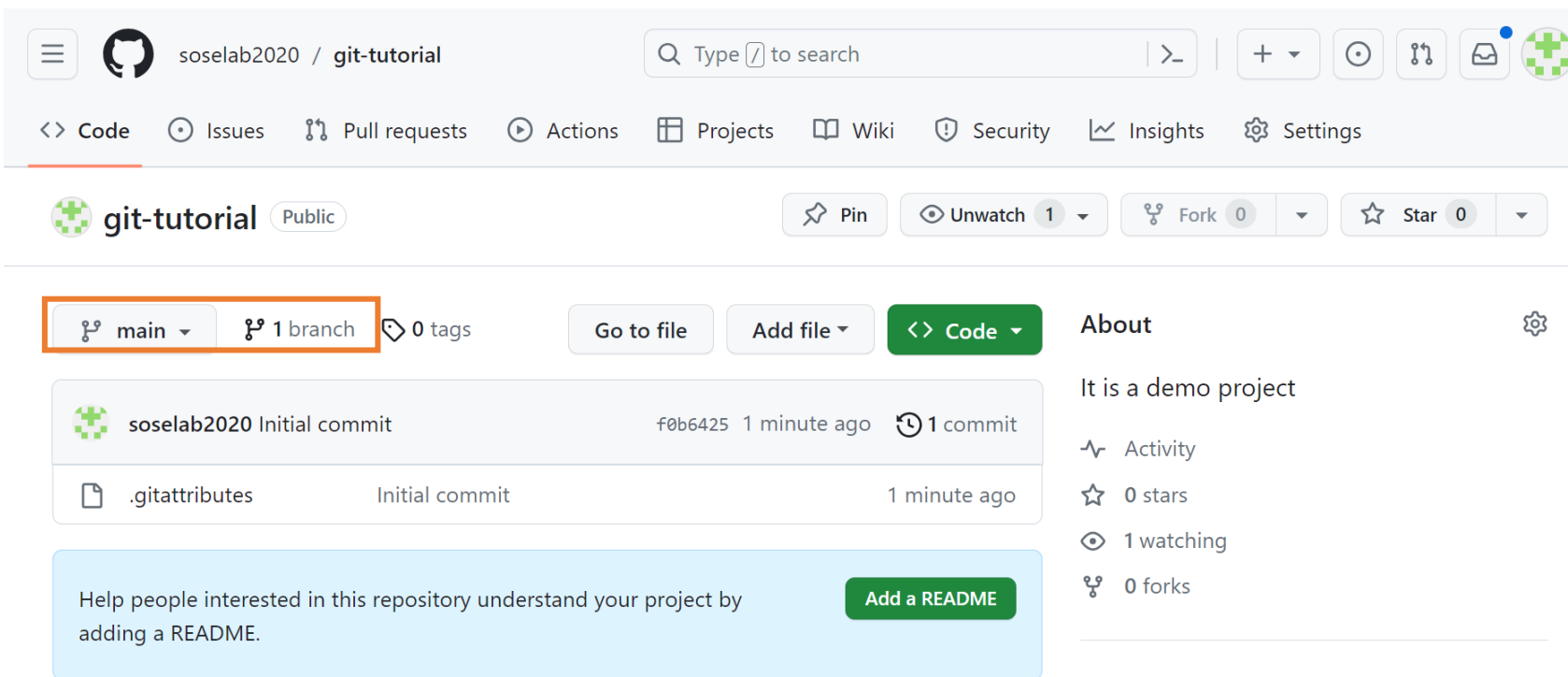
- 建立與連結遠端儲存庫(Remote Repository)
 - 我們可直接將本地端儲存庫與GitHub遠端儲存庫串連起來
 - 點擊"Publish Repository"功能即可完成
 - 等同將專案發布於Internet，日後可方便與他人協同合作



- 建立與連結遠端儲存庫(Remote Repository)



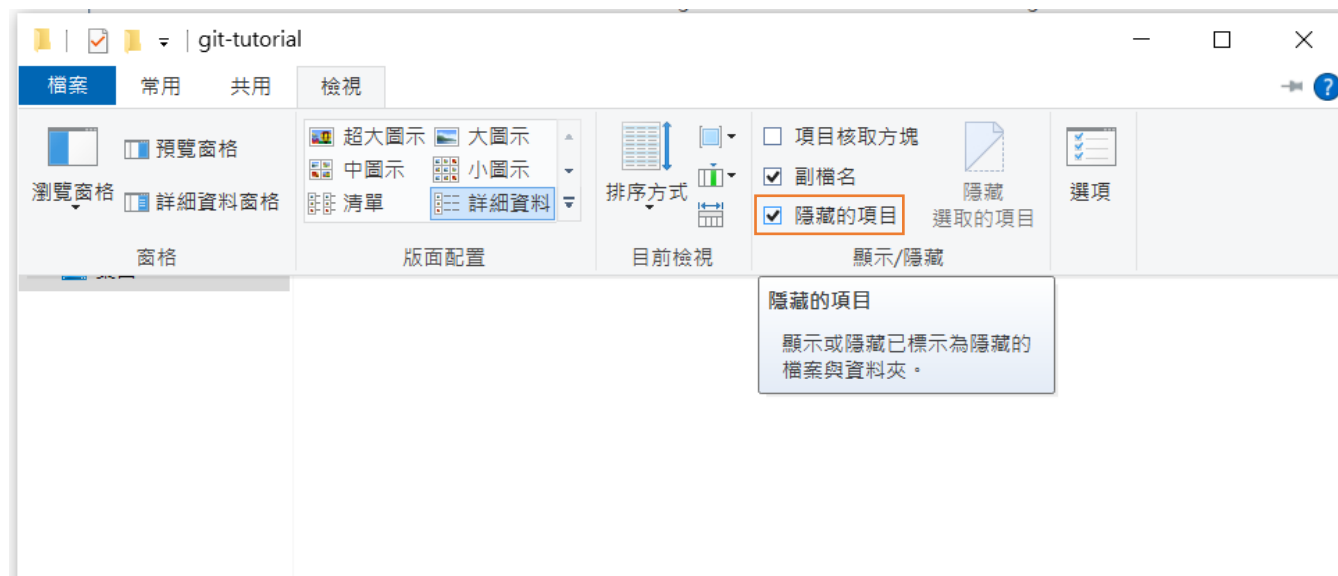
- 建立與連結遠端儲存庫(Remote Repository)
 - 連結GitHub，將會看到遠端儲存庫也成功地被建立出來。
 - 一個分支(branch)可視為專案的獨立開發支線
 - 以目前來說，可先都以main分支為主，日後請大家務必要再了解分支的處理方式



- Lab 1

- 請完成GitHub Desktop之安裝與設定。
 - 請完成與GitHub帳號的串連。
 - 請完成Git所需的使用者帳號與email設定。
- 請透過GitHub Desktop建立一個空的Local Repository (本地儲存庫)，並透過"Publish Repository"功能建立對應的Remote Repository (遠端儲存庫)。
 - 此本地儲存庫我們後續將其稱之為LR1。
 - 遠端儲存庫我們後續將其稱之為RR。

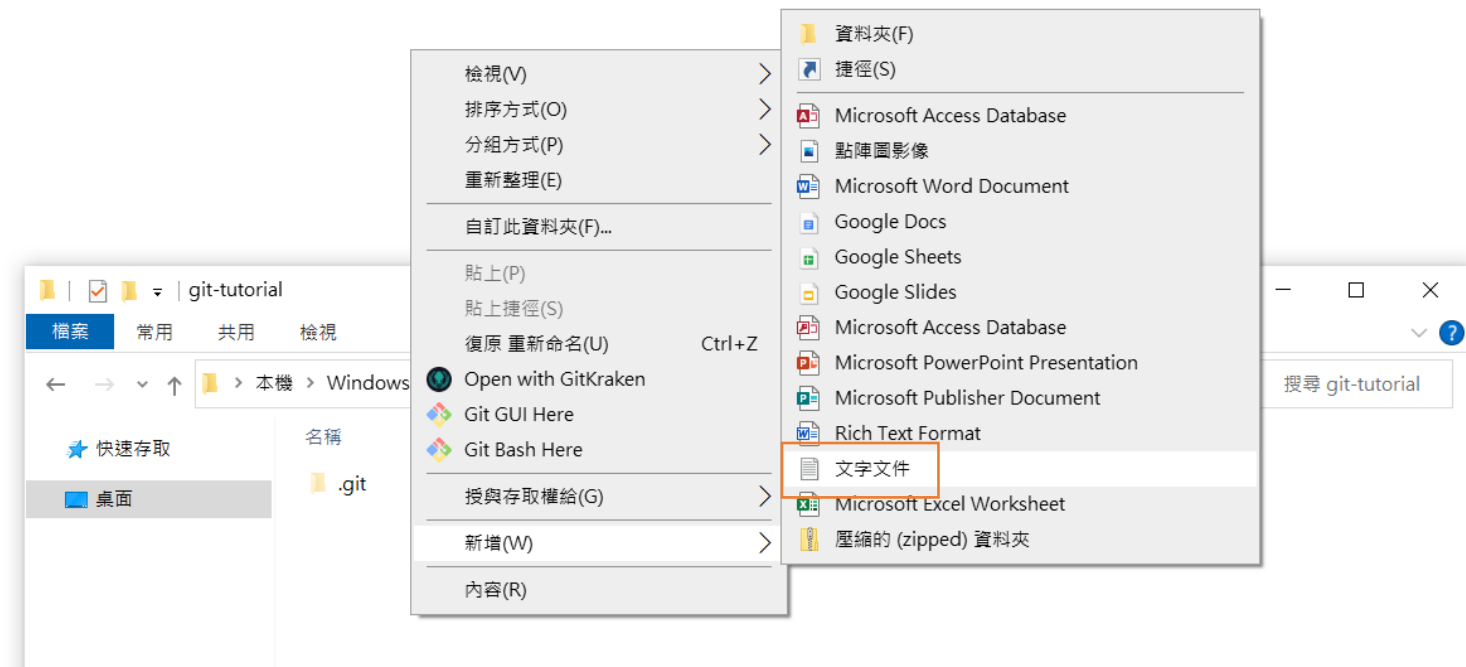
- 交付檔案至本地端儲存庫
 - 首先開啟檔案總管，開啟顯示附檔名。



- 交付檔案至本地端儲存庫

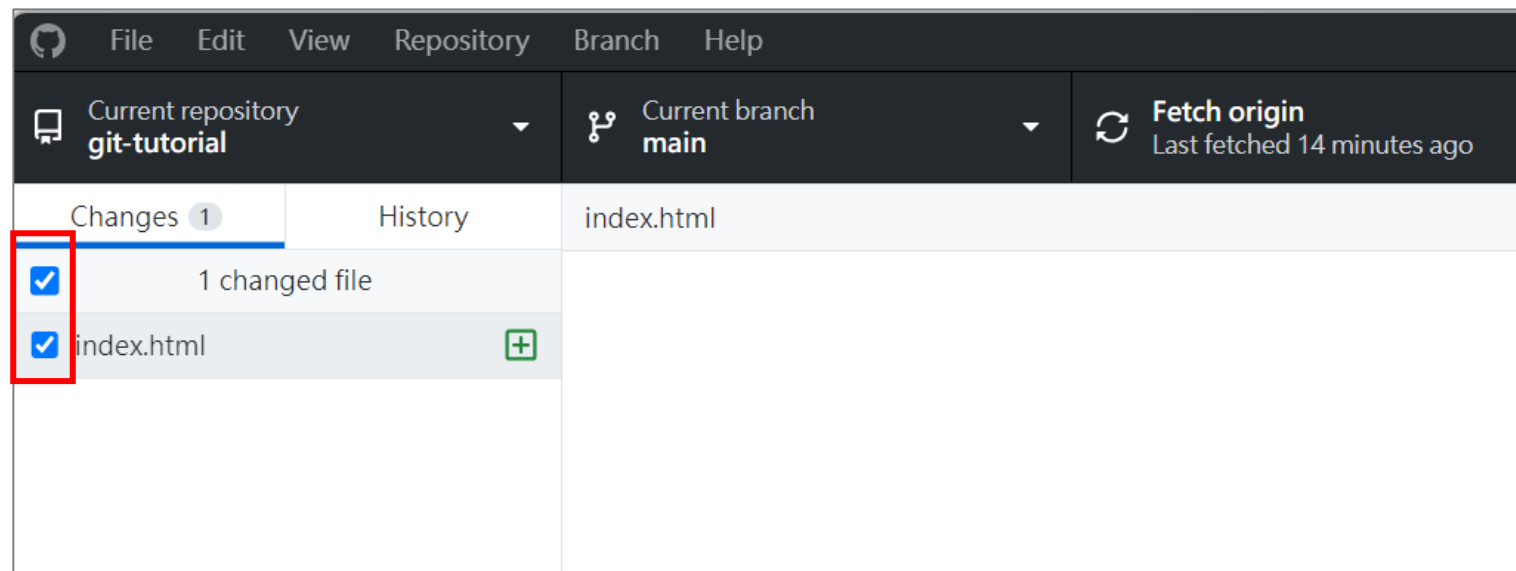
- 加入未追蹤檔案至工作目錄

- e.g., *index.html*

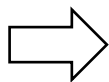


- 檢視檔案狀態

GitHub Desktop會預設勾選所有新加入檔案，以讓我們將未追蹤檔案加入待交付區域



工作樹
(Working Tree)

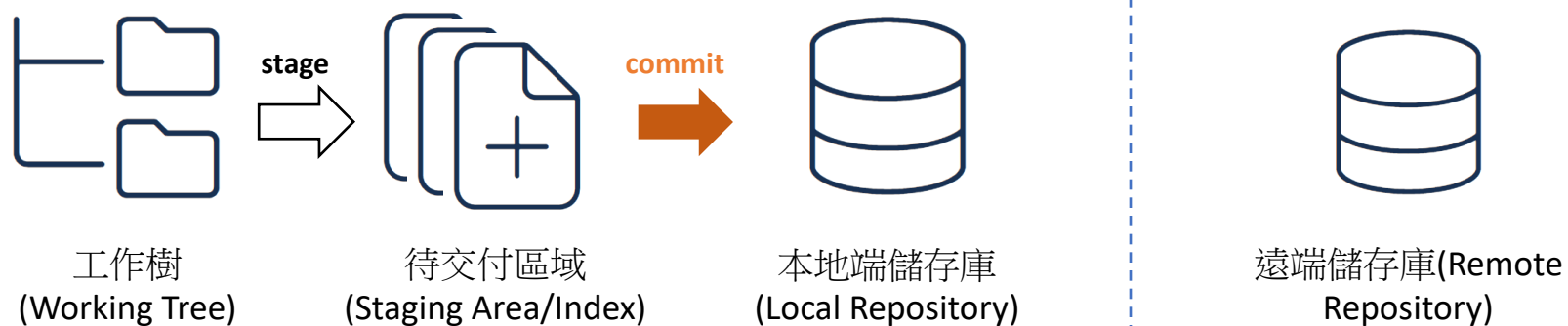


stage



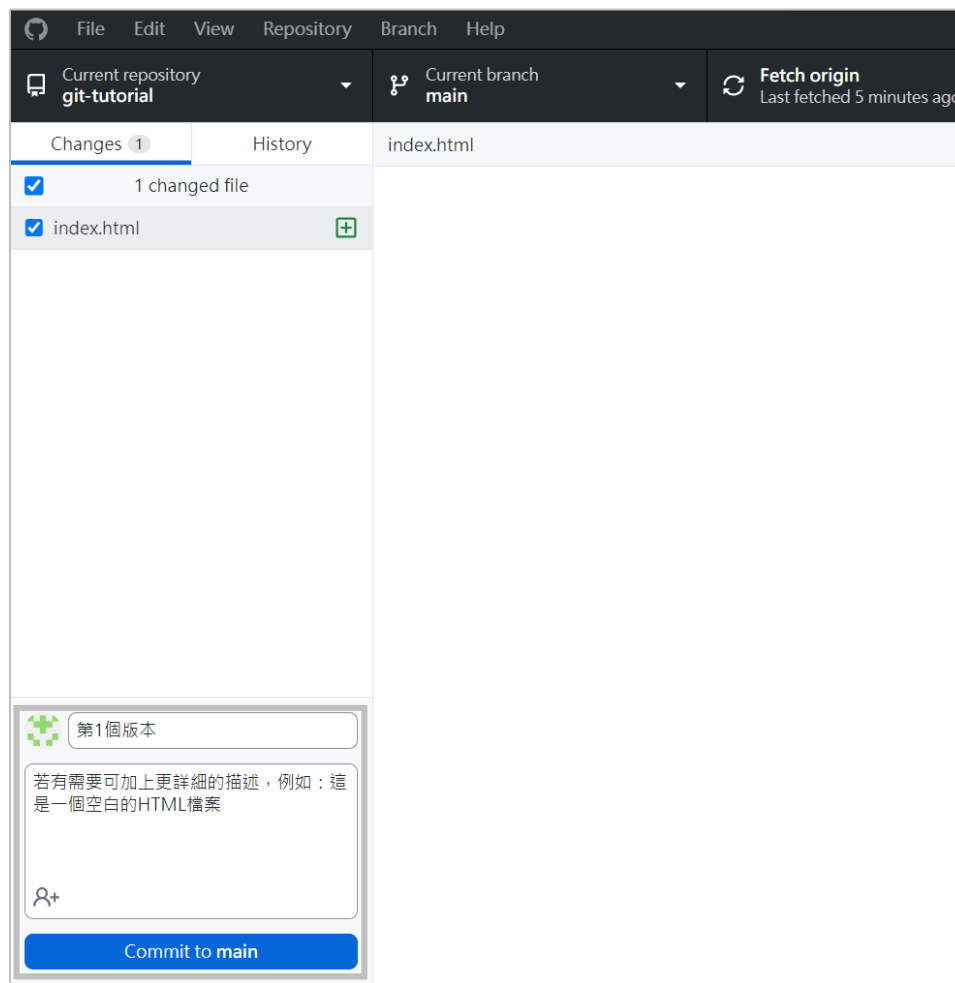
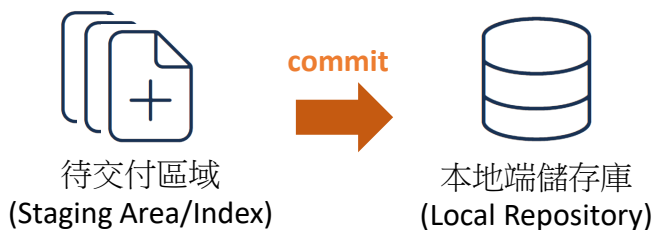
待交付區域
(Staging Area/Index)

- 交付檔案至本地端儲存庫

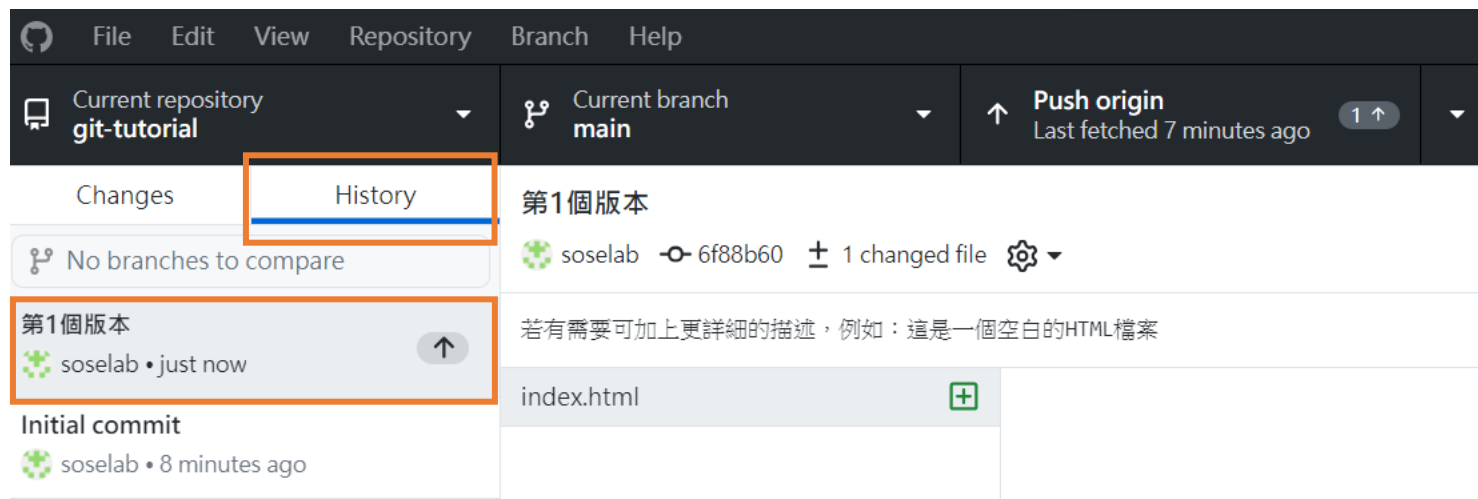
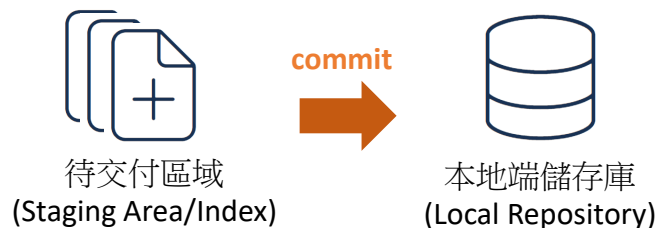


- 交付檔案至本地端儲存庫

- 我們要撰寫交付訊息(commit message)，實際建立commit。



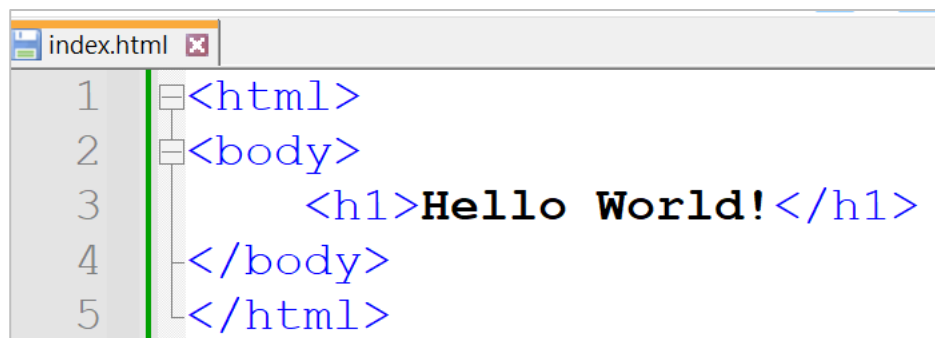
- 交付檔案至本地端儲存庫
 - 檢視交付歷程(history，也就是log)



- 小量改進 - 修改內容

- 修改/變更工作目錄(working tree)中的 *index.html*

```
<h1>Hello World!</h1>
```



```
1 <html>
2 <body>
3     <h1>Hello World!</h1>
4 </body>
5 </html>
```

(Notepad++)



(Chrome)

- 小量改進 - 修改內容

- 可查看差異，
並交付新版

File Edit View Repository Branch Help

Current repository: git-tutorial Current branch: main Push origin Last fetched 10 minutes a... 1 ↑

Changes 1 History index.html

1 changed file

index.html

```
@@ -0,0 +1,5 @@
1 + <html>
2 + <body>
3 +   <h1>Hello World!</h1>
4 + </body>
5 + </html>
```

顯示與前版的差異(diff)

第2版：新增內容之標題

Description

Commit to main

Committed 3 minutes ago
第1個版本

- 小量改進 - 修改內容
 - 檢視交付歷程

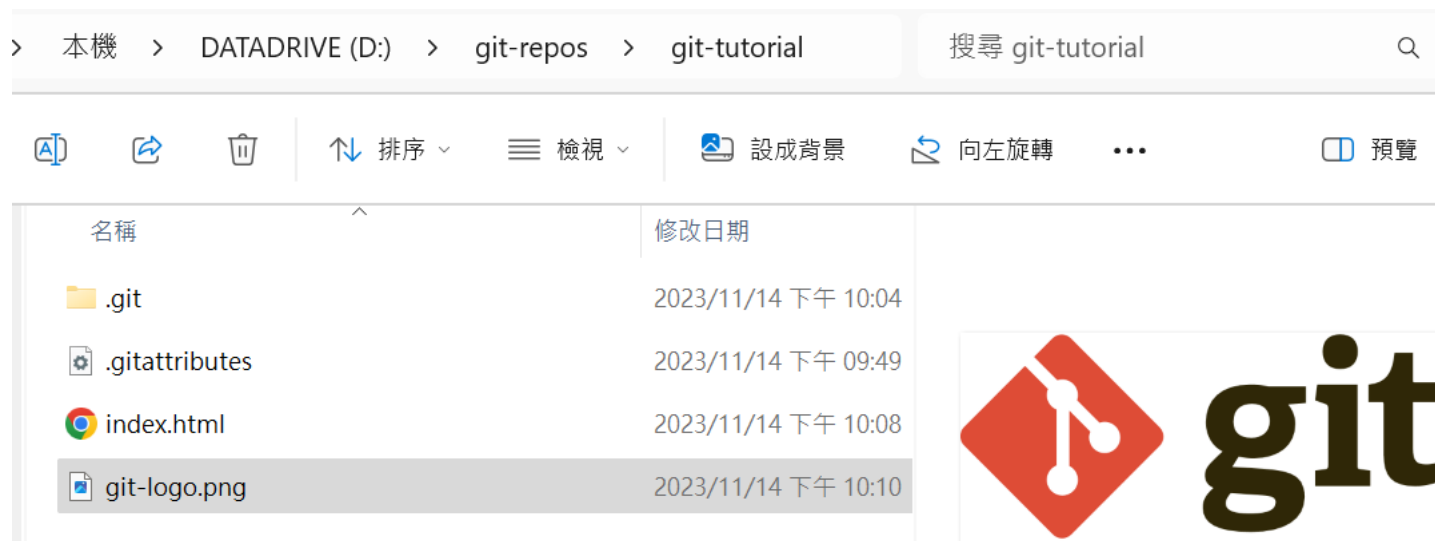
The screenshot shows the Visual Studio Code Git interface. The top bar includes menus for File, Edit, View, Repository, Branch, and Help. Below this, the 'Current repository' is 'git-tutorial' and the 'Current branch' is 'main'. A 'Push origin' button is visible, indicating the last fetch was 14 minutes ago. The 'History' tab is selected, showing a list of commits. The most recent commit is '第2版：新增內容之標題' by 'soselab' just now. Below it is '第1個版本' by 'soselab' 8 minutes ago, and the 'Initial commit' by 'soselab' 15 minutes ago. To the right, a diff view for 'index.html' is shown, highlighting the changes in the second commit. The diff shows the addition of an HTML document structure with a 'Hello World!' message.

Commit	Author	Time	Changes
第2版：新增內容之標題	soselab	just now	1 changed file +5 -0
第1個版本	soselab	8 minutes ago	
Initial commit	soselab	15 minutes ago	

Diff for index.html:

```
@@ -0,0 +1,5 @@
1 + <html>
2 + <body>
3 +     <h1>Hello World!</h1>
4 + </body>
5 + </html>
```

- 小量改進 - 新增檔案
 - 加入未追蹤圖片至工作目錄
 - e.g., *git-logo.png*



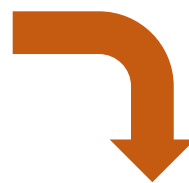
- 小量改進 - 新增檔案
 - 修改/變更工作目錄中的 *index.html*

```
<h1>Hello World!</h1>  

```



```
1 <html>  
2 <body>  
3   <h1>Hello World!</h1>  
4     
5 </body>  
6 </html>
```



- 小量改進 - 新增檔案

- 可再查看差異，
並交付新版

File Edit View Repository Branch Help

Current repository **git-tutorial** Current branch **main** **Push origin** Last fetched 24 minutes a... 2 ↑

Changes 2 History index.html

2 changed files

- git-logo.png
- index.html

@@ -1,5 +1,6 @@

```
1 <html>
2 <body>
3     <h1>Hello World!</h1>
4 +   
5 </body>
6 </html>
```

新增圖片於網頁

Description

Commit to main

Committed 9 minutes ago
第2版：新增內容之標題

Undo

- 小量改進 - 新增檔案
 - 檢視交付歷程
 - 我們可透過此模式逐步新增原始碼與建立交付版本

The screenshot shows the Git GUI interface with the following components:

- Menu Bar:** File, Edit, View, Repository, Branch, Help.
- Repository Info:** Current repository: git-tutorial, Current branch: main, Push origin (Last fetched 28 minutes ago).
- History Tab:** Shows a list of commits with descriptions and timestamps.

Commit Description	Timestamp
新增圖片於網頁	soselab • 1 minute ago
第2版：新增內容之標題	soselab • 13 minutes ago
第1個版本	soselab • 21 minutes ago
Initial commit	soselab • 28 minutes ago
- Diff View:** Shows the changes in the current commit (75c4d83) compared to the previous one. The diff shows two files: git-logo.png and index.html.

File	Changes
git-logo.png	Added (+)
index.html	Modified (yellow square)
- Code Diff:** Shows the diff for index.html, highlighting the addition of an image tag.

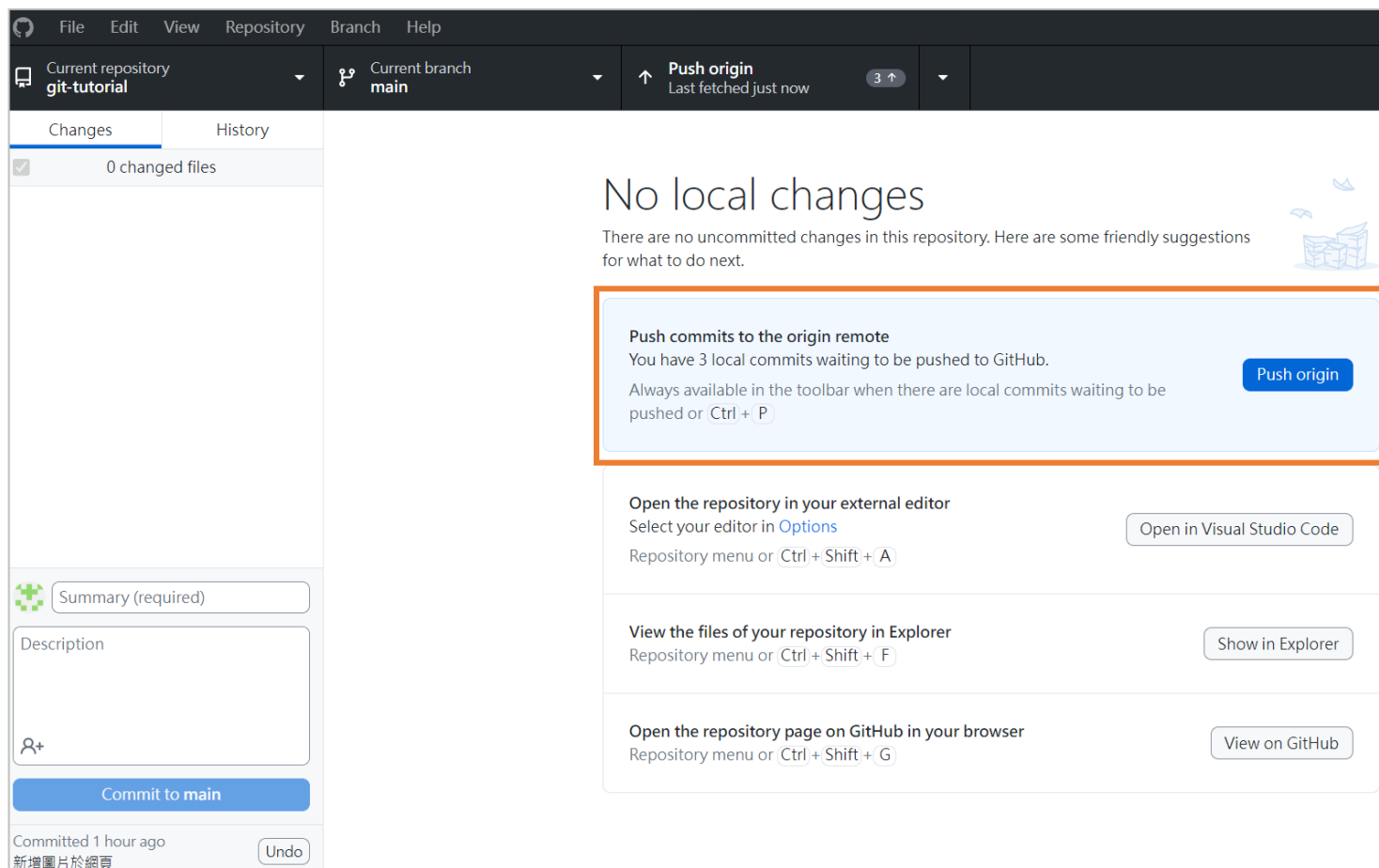
```
@@ -1,5 +1,6 @@
1 1 <html>
2 2 <body>
3 3     <h1>Hello World!</h1>
4 4 +   
5 5 </body>
6 6 </html>
```

- 同步遠端儲存庫

- 我們可將本地端儲存庫的一個交付歷程**推送(push)**至遠端儲存庫



- 同步遠端儲存庫
 - 點選[Push origin]



- 完成遠端儲存庫同步

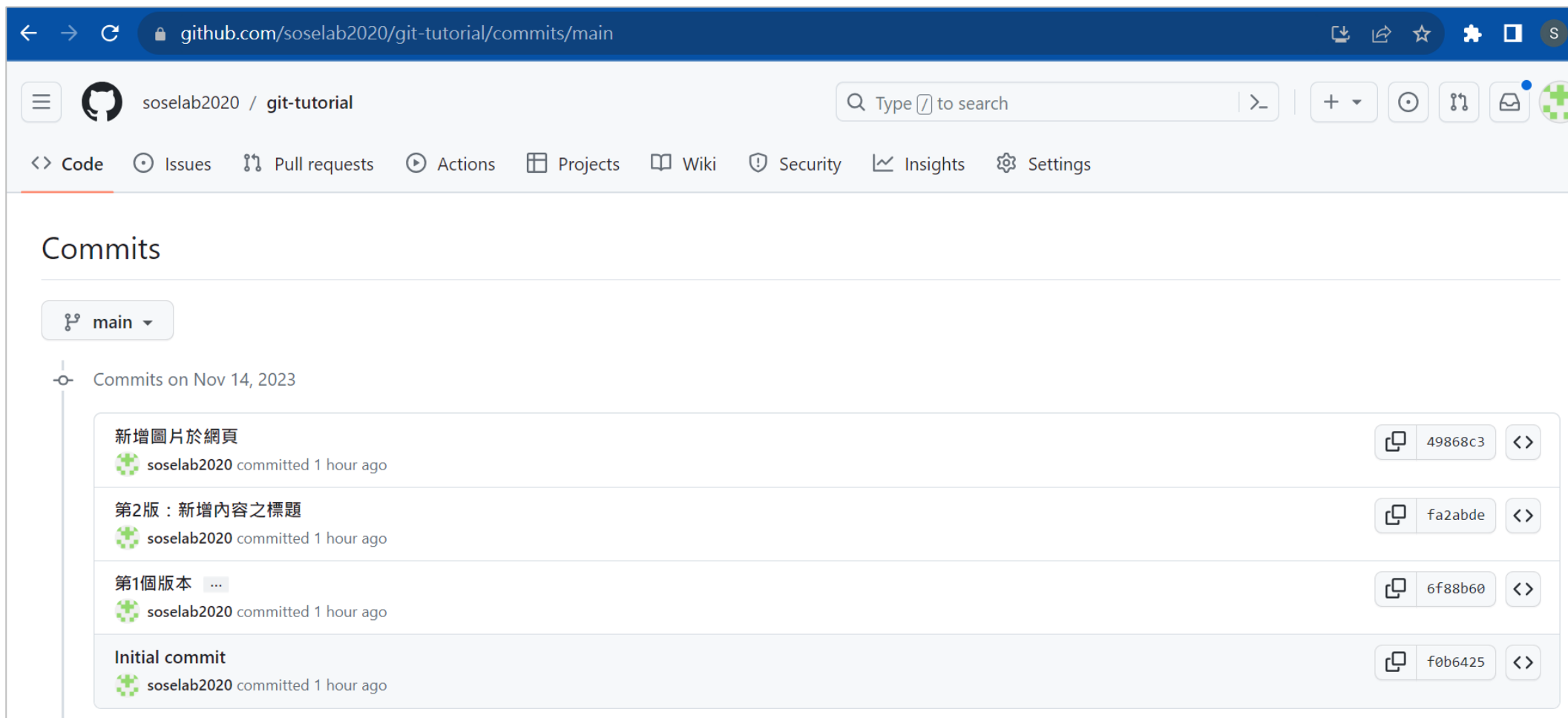
- 重整遠端儲存庫

- 可看到所有commit均以推送至遠端儲存庫

The screenshot shows a GitHub repository page for 'git-tutorial' by user 'soselab2020'. The repository is public and has 1 branch, 0 tags, and 4 commits. The commit history table lists three files: .gitattributes, git-logo.png, and index.html, all committed 1 hour ago. The '4 commits' link is highlighted with an orange box. The right sidebar shows the repository is a demo project with 0 stars, 1 watcher, and 0 forks.

File	Commit Message	Time
.gitattributes	Initial commit	1 hour ago
git-logo.png	新增圖片於網頁	1 hour ago
index.html	新增圖片於網頁	1 hour ago

- 查看遠端儲存庫上的交付歷程



The screenshot displays the GitHub interface for the repository `soselab2020 / git-tutorial`. The page is titled "Commits" and shows a list of commits on the `main` branch. The commits are ordered from most recent to oldest. Each commit entry includes a description, the user `soselab2020`, the time "committed 1 hour ago", a copy icon, a hexadecimal commit hash, and a comparison icon (`<>`).

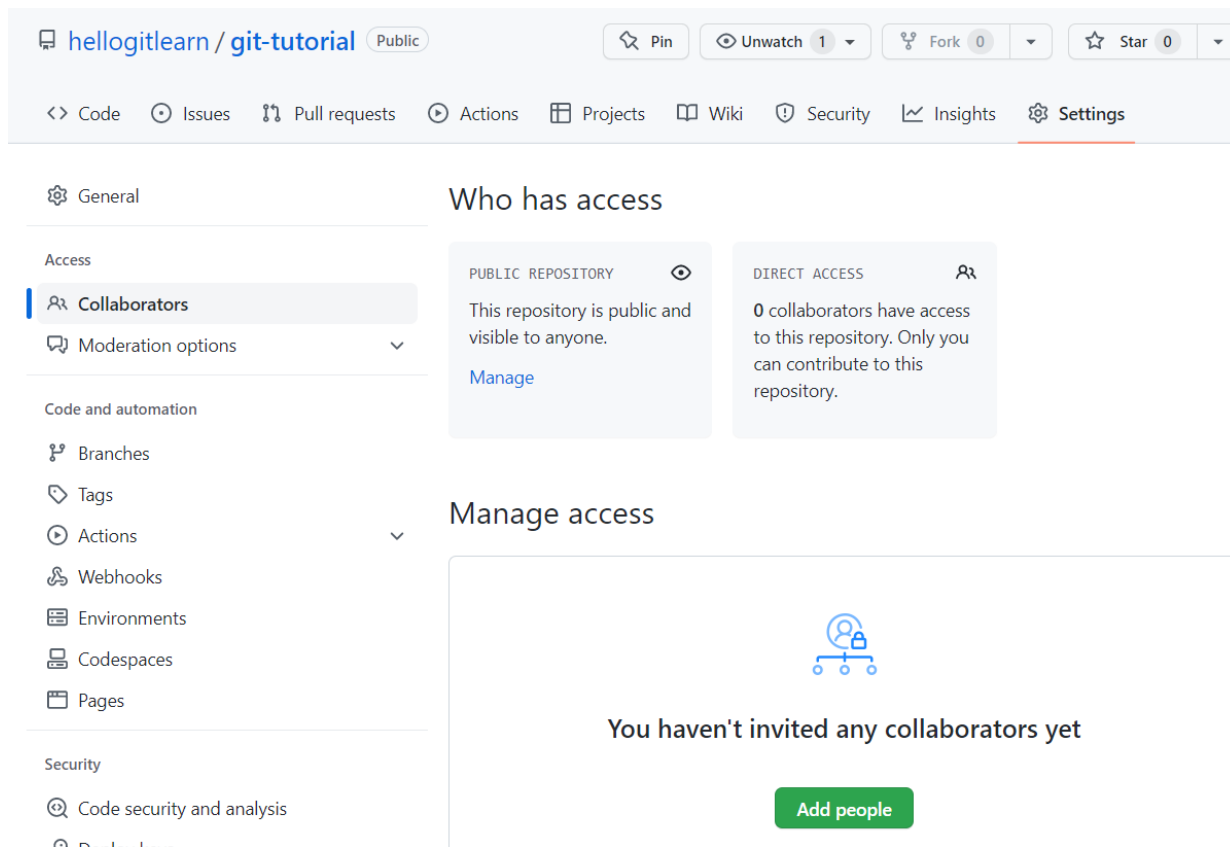
Commit Description	Commit Hash
新增圖片於網頁	49868c3
第2版：新增內容之標題	fa2abde
第1個版本 ...	6f88b60
Initial commit	f0b6425

• Lab 2

- 請選定任何一支程式(如HTML、Python、VBA)等，將其放入剛剛創建的空儲存庫，並對其進行修改，以建立多個commit。
 - 請隨時觀看History了解交付歷程。
- 將本地端的commit都推送到遠端GitHub儲存庫。
 - 請觀看GitHub之"Commits"內容，以觀察遠端儲存庫與本地儲存庫之交付版本內容是否一致。

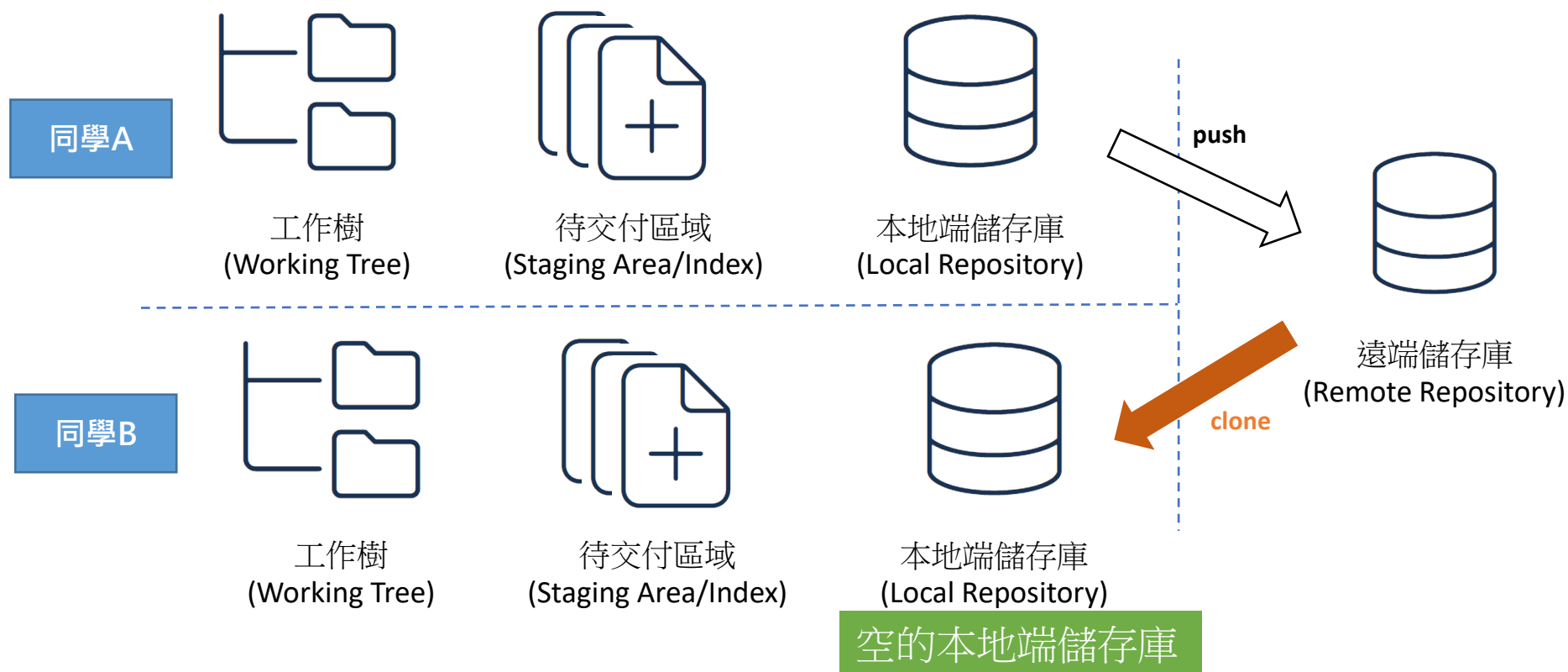
• GitHub 多人合作 - Collaborators

- Collaborators適用於一個team之間的合作
- 負責開這個repository的人，進入settings，加入collaborators的帳號



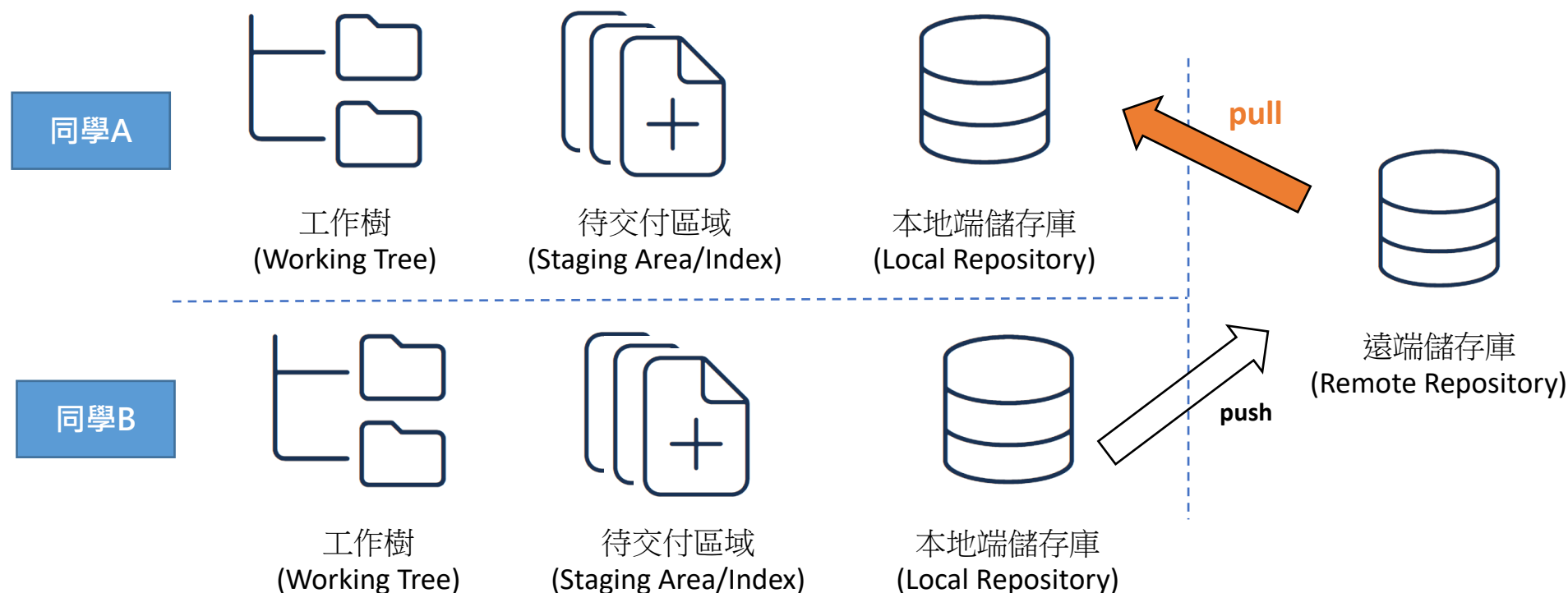
• GitHub 多人合作 - Collaborators

- 協同合作者可將遠端儲存庫(擁有者所維護)的交付歷程複製(clone)到本地端儲存庫



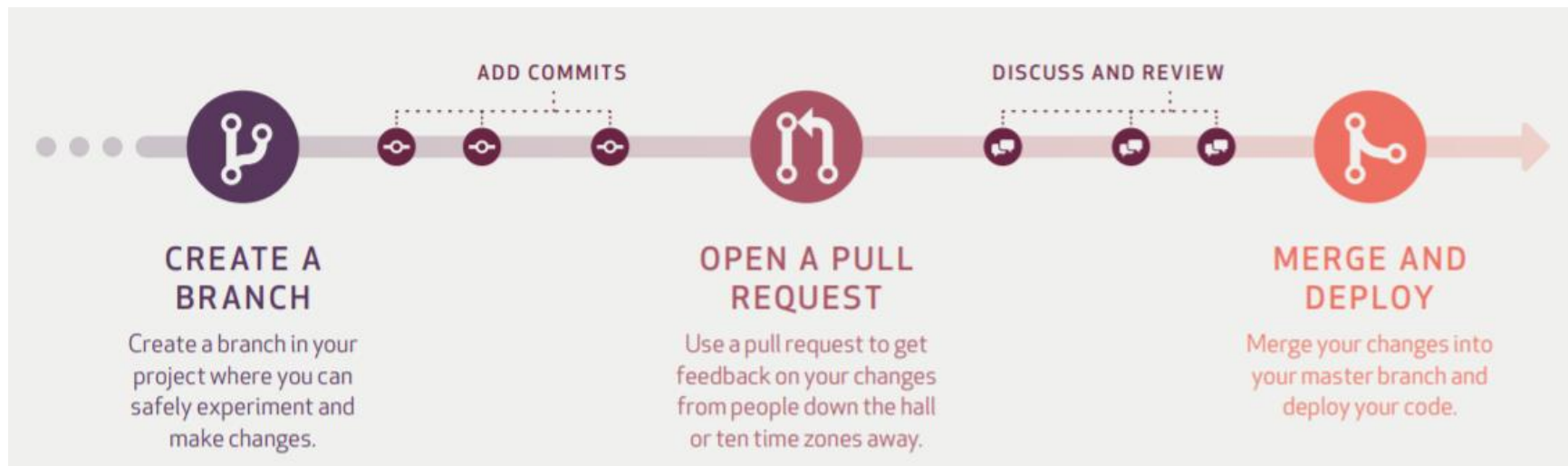
• GitHub 多人合作 - Collaborators

- 協同合作者可以將修改的交付透過push指令同步至遠端儲存庫
- 而其他團隊成員可以透過pull指令將遠端儲存庫的最新交付版本同步至本地端儲存庫



• GitHub 多人合作模式 – Push, Pull, Pull Request

- 後續可類似先前多工作環境的模式，在多人間對同一個儲存庫(同一個分支)持續Push與Pull。
- 若要更好的合作，應各自建立分支(branch)後，再進行合併(merge)。
- 更理想的合作模式可參考GitHub Flow，透過Pull Request來提醒團隊有分支合併之請求。





THANK YOU