

Lab0-2

Git - Branch Merge Conflict

本節目的：

- 連結 GitHub Classroom
- 創立新的分支(Branch) Develop。
- 合併(Merge)分支 Develop → Master。
- 了解如何解決分支合併時的衝突(Conflict)。

0.1 實戰演練

- 連接 Github Classroom
- 創立新的分支
- 合併(Merge)分支
- 解決分支合併時的衝突(Conflict)

0.1.1 連接 Github Classroom

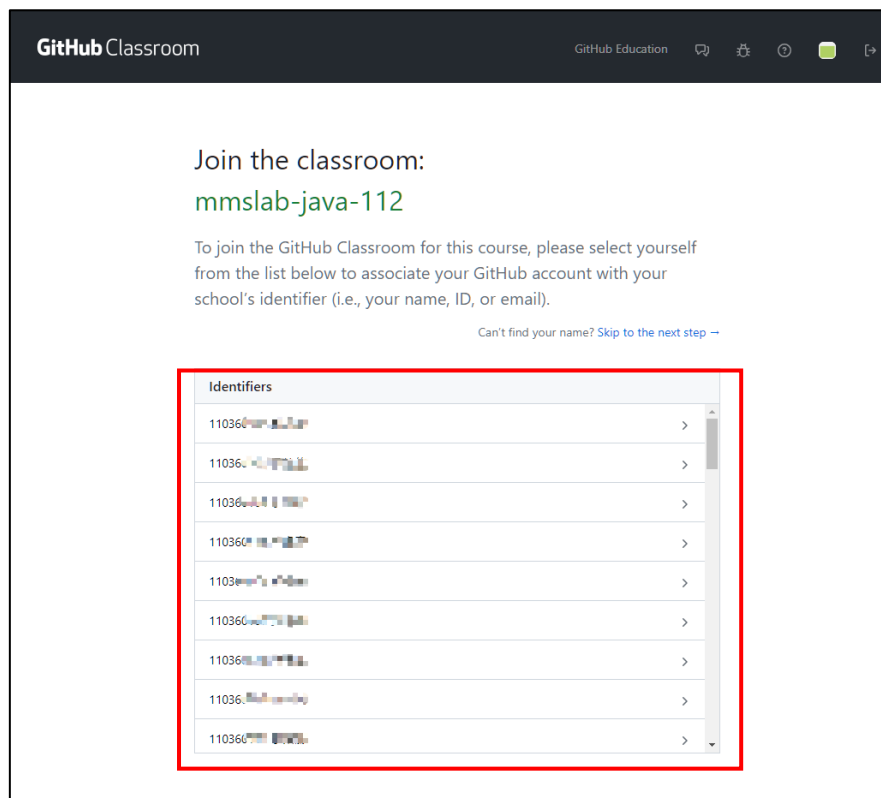
Step1 點擊實驗/作業繳交連結

Lab0-2 Github Classroom Assignment: 連結在上課前公布至課堂群組及 <https://7333.italkutalk.com/>

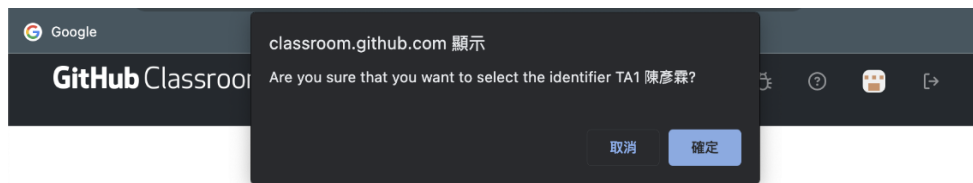
Step2 將自己的 Github 帳號與 Classroom 學生連結

同一門課程 Github Classroom 的作業或實驗僅需連結一次 (若曾經連結過，可以略過此步驟)。

在學生清單中，會列出本門課程尚未被連結的學生，請找到並點擊自己的學號/姓名



點擊後會跳出確認提示，確認無誤點擊確定。

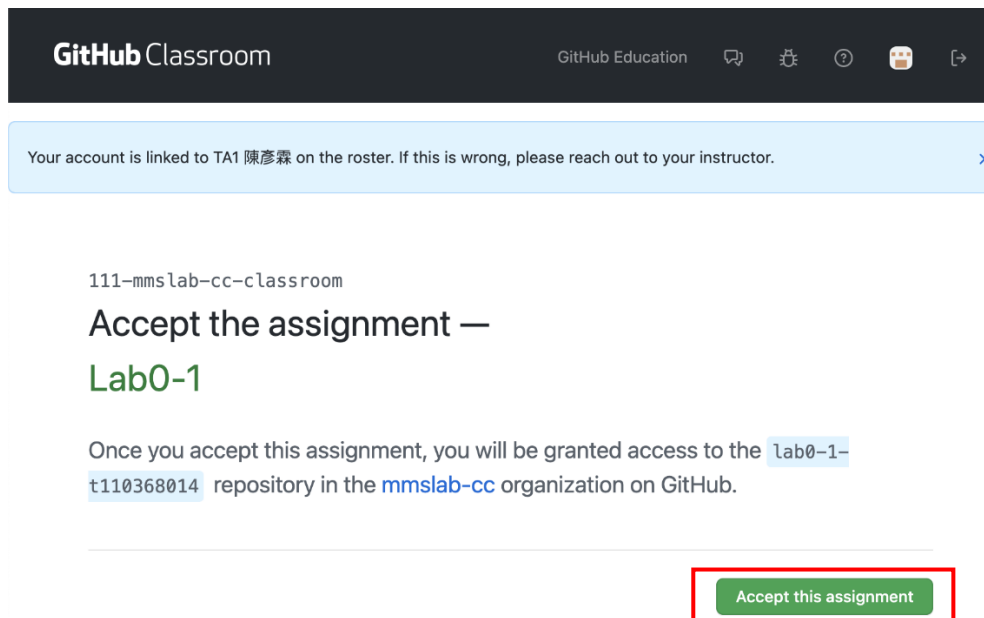


如果遇到以下問題：

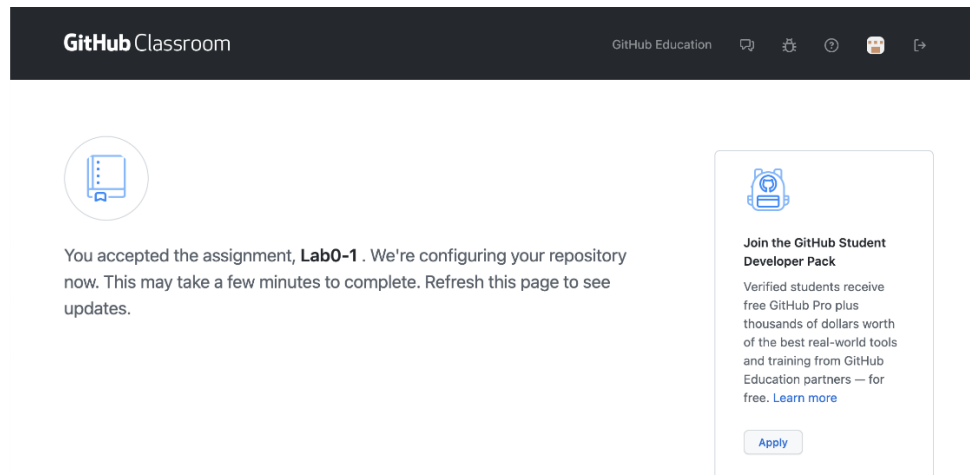
1. 名單中找不到自己的學號姓名
2. 選擇錯人
3. 學號姓名錯誤

請與課堂助教反應，助教將會協助處理。

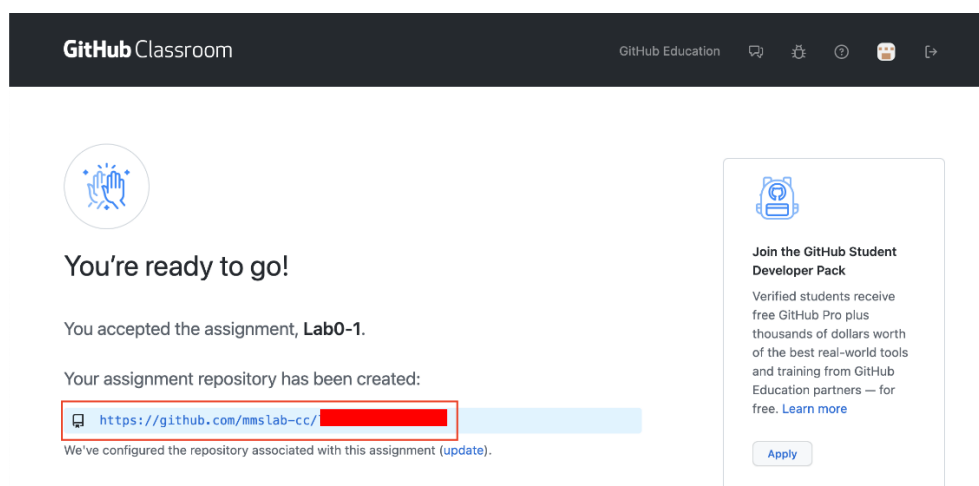
Step3 接受 Assignment，點擊 Accept this assignment



接受 Assignment 後，Github classroom 會幫你建立專屬的 repository，而建立專屬的 repository 需要一段時間，請等 10 秒左右刷新此頁面



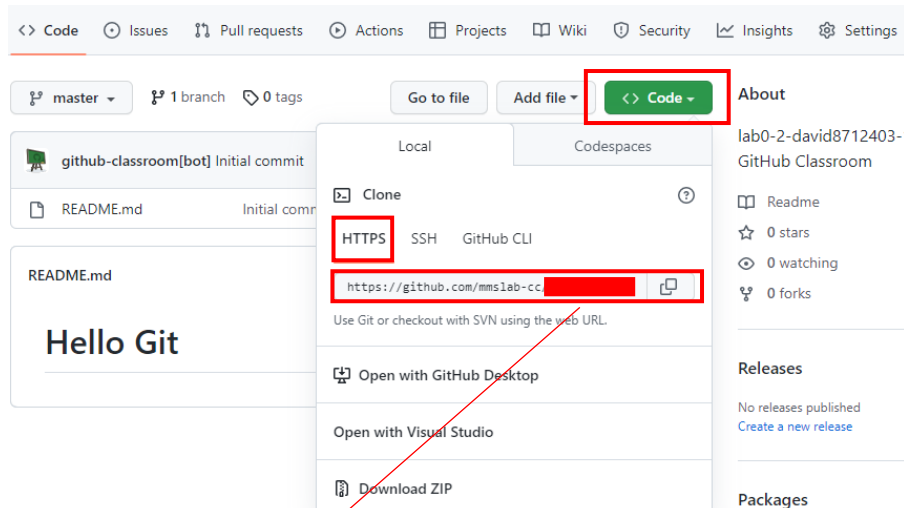
刷新頁面後，將會看到屬於自己的 repository 連結，並點擊該連結。



Step4 將專案 Clone 到自己電腦
到桌面開啟 Git Bash



複製專案在遠端資料庫的位置，並在 Git bash 輸入指令進行下載專案(命令列
點擊右鍵可以選貼上)



```
$ git clone https://github.com/xxx/xxx.git (記得改成自己的資料庫連結網址)
```

輸入指令情況：

```
Jdway@DESKTOP-GBM49C1 MINGW64 ~/Desktop
$ git clone https://github.com/XXXXXXXXXXXXXXXXXXXX.git
Cloning into 'XXXXXXXXXXXX'...
remote: Counting objects: 7, done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 7 (delta 0), reused 7 (delta 0), pack-reused 0
Unpacking objects: 100% (7/7), done.

Jdway@DESKTOP-GBM49C1 MINGW64 ~/Desktop
$ |
```

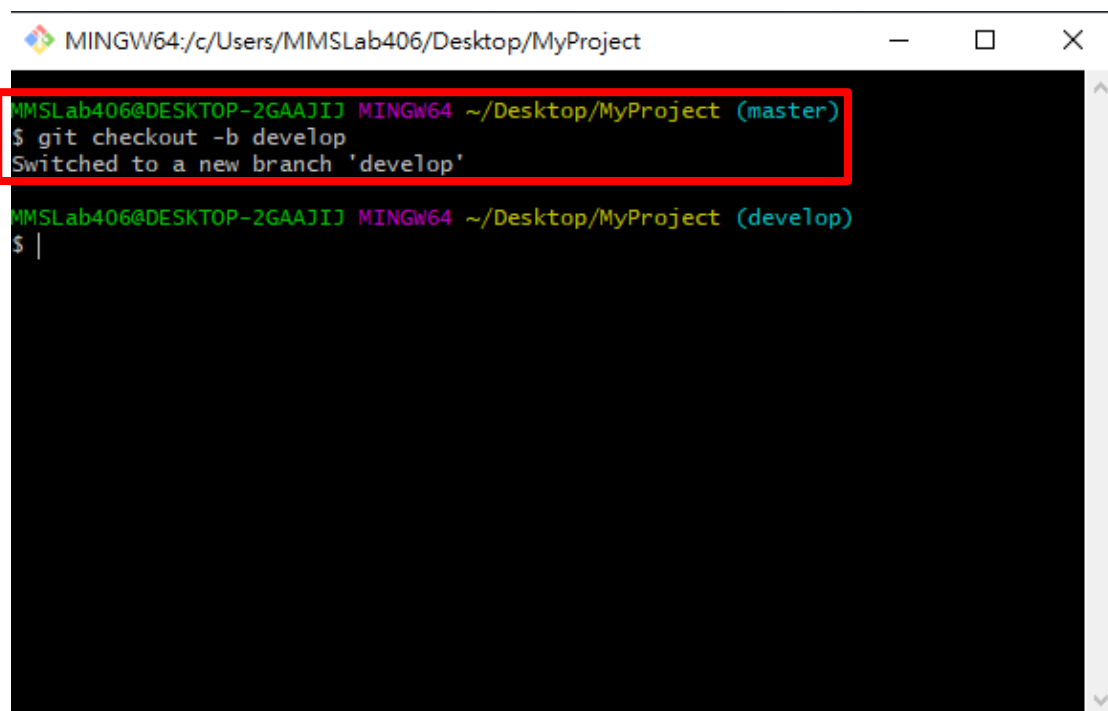
完成下載專案後請到桌面開啟剛下載的專案資料夾，再開啟一次 Git Bash 繼續下面演練。



0.1.2 創立新的分支

Step1 開啟剛剛 clone 下來的專案資料夾，並開啟 Bash，輸入指令新增分支，並切換到新分支，如圖 0-1 所示。

```
$ git checkout -b develop
```



```
MINGW64:/c/Users/MMSLab406/Desktop/MyProject
MMSLab406@DESKTOP-2GAAJIJ MINGW64 ~/Desktop/MyProject (master)
$ git checkout -b develop
Switched to a new branch 'develop'
MMSLab406@DESKTOP-2GAAJIJ MINGW64 ~/Desktop/MyProject (develop)
$ |
```

圖 0-1 新增分支並切換

Step2 用記事本開啟 README，並修改內容後儲存，如圖 0-2 所示。



圖 0-2 修改 README 檔

Step3 檢查資料狀態，有一個修改的檔案，如圖 0-3 所示。

```
$ git status
```

```
MINGW64: /c:/Users/MMSLab406/Desktop/MyProject

MMSLab406@DESKTOP-2GAAJIJ MINGW64 ~/Desktop/MyProject (master)
$ git checkout -b develop
Switched to a new branch 'develop'

MMSLab406@DESKTOP-2GAAJIJ MINGW64 ~/Desktop/MyProject (develop)
$ git status
On branch develop
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md
no changes added to commit (use "git add" and/or "git commit -a")

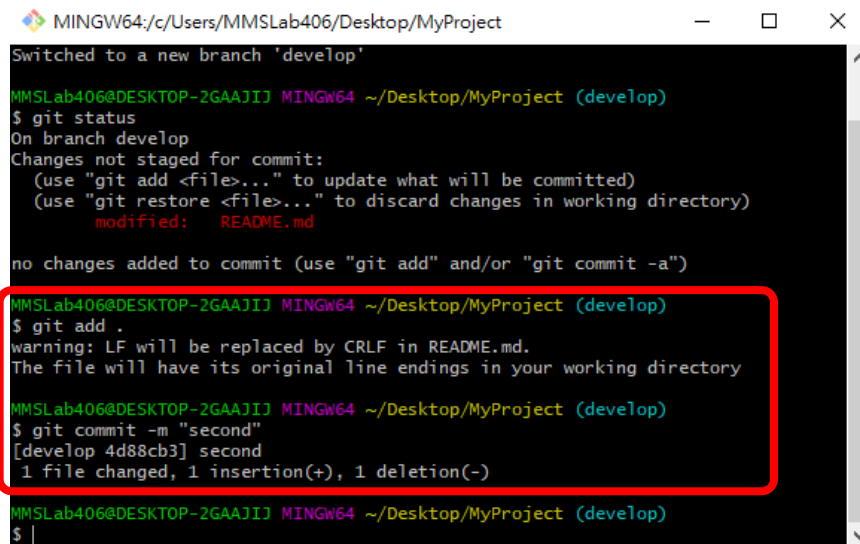
MMSLab406@DESKTOP-2GAAJIJ MINGW64 ~/Desktop/MyProject (develop)
$
```

圖 0-3 檢查資料狀態

Step4 將檔案加入提交名單並提交，如圖 0-4 所示。

```
$ git add .
```

```
$ git commit -m "second"
```



```
MINGW64:/c:/Users/MMSLab406/Desktop/MyProject
Switched to a new branch 'develop'

MMSLab406@DESKTOP-2GAAJIJ MINGW64 ~/Desktop/MyProject (develop)
$ git status
On branch develop
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")

MMSLab406@DESKTOP-2GAAJIJ MINGW64 ~/Desktop/MyProject (develop)
$ git add .
warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working directory

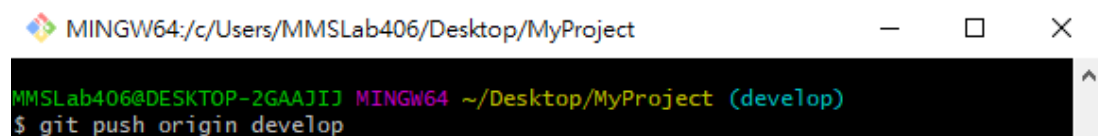
MMSLab406@DESKTOP-2GAAJIJ MINGW64 ~/Desktop/MyProject (develop)
$ git commit -m "second"
[develop 4d88cb3] second
1 file changed, 1 insertion(+), 1 deletion(-)

MMSLab406@DESKTOP-2GAAJIJ MINGW64 ~/Desktop/MyProject (develop)
$ |
```

圖 0-4 加入並提交

Step5 提交到遠端資料庫（GitHub）上，如圖 0-5 所示。

```
$ git push origin develop
```



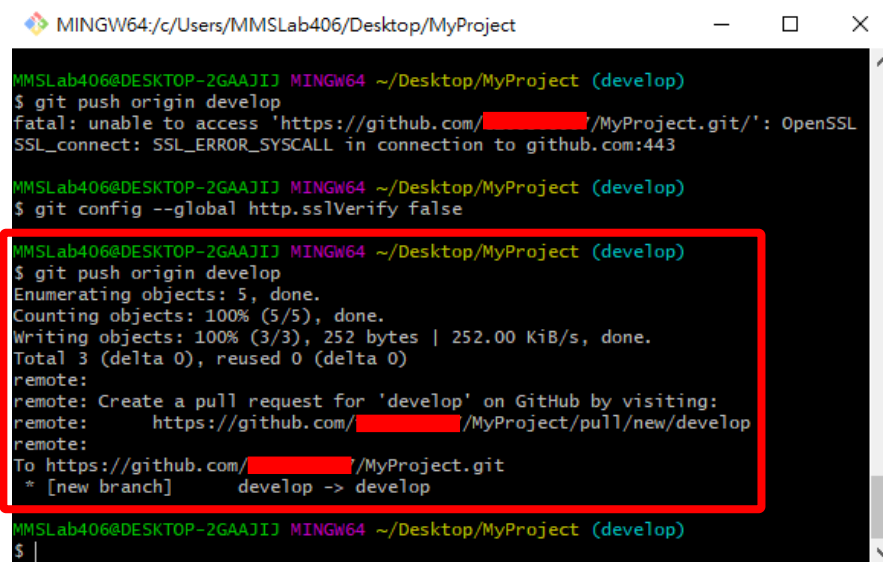
```
MINGW64:/c:/Users/MMSLab406/Desktop/MyProject

MMSLab406@DESKTOP-2GAAJIJ MINGW64 ~/Desktop/MyProject (develop)
$ git push origin develop
```

圖 0-5 提交到遠端資料庫

Step6 檔案成功推至遠端的 GitHub，如圖 0-6 所示，若遇到 unable to access 的問題，請輸入以下指令，再執行一次 Step5。

```
$ git config --global http.sslVerify false
```



```
MINGW64~/c/Users/MMSLab406/Desktop/MyProject
MMSLab406@DESKTOP-2GAAJ1J MINGW64 ~/Desktop/MyProject (develop)
$ git push origin develop
fatal: unable to access 'https://github.com/[redacted]/MyProject.git/': OpenSSL
SSL_connect: SSL_ERROR_SYSCALL in connection to github.com:443

MMSLab406@DESKTOP-2GAAJ1J MINGW64 ~/Desktop/MyProject (develop)
$ git config --global http.sslVerify false

MMSLab406@DESKTOP-2GAAJ1J MINGW64 ~/Desktop/MyProject (develop)
$ git push origin develop
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Writing objects: 100% (3/3), 252 bytes | 252.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:   https://github.com/[redacted]/MyProject/pull/new/develop
remote:
To https://github.com/[redacted]/MyProject.git
 * [new branch]      develop -> develop

MMSLab406@DESKTOP-2GAAJ1J MINGW64 ~/Desktop/MyProject (develop)
$ |
```

圖 0-6 檔案上傳成功

Step7 回瀏覽器，F5 重新整理 GitHub 的頁面，並切換到 develop 分支，可以看到剛剛上傳的檔案，如圖 0-7 所示。

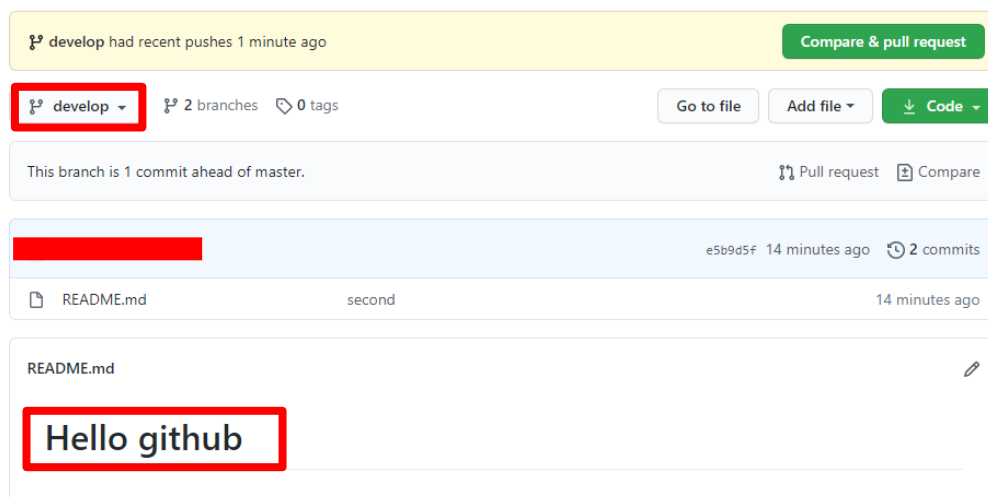


圖 0-7 查看 GitHub 上的專案

0.1.3 合併(Merge)分支

Step1 切換分支從 develop 到 master，如圖 0-8 所示。

```
MMSLab406@DESKTOP-2GAAJIJ MINGW64 ~/Desktop/MyProject (develop)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

MMSLab406@DESKTOP-2GAAJIJ MINGW64 ~/Desktop/MyProject (master)
$ |
```

圖 0-8 切換分支

Step2 打開資料夾確認 README 目前的資料，如圖 0-9 所示。

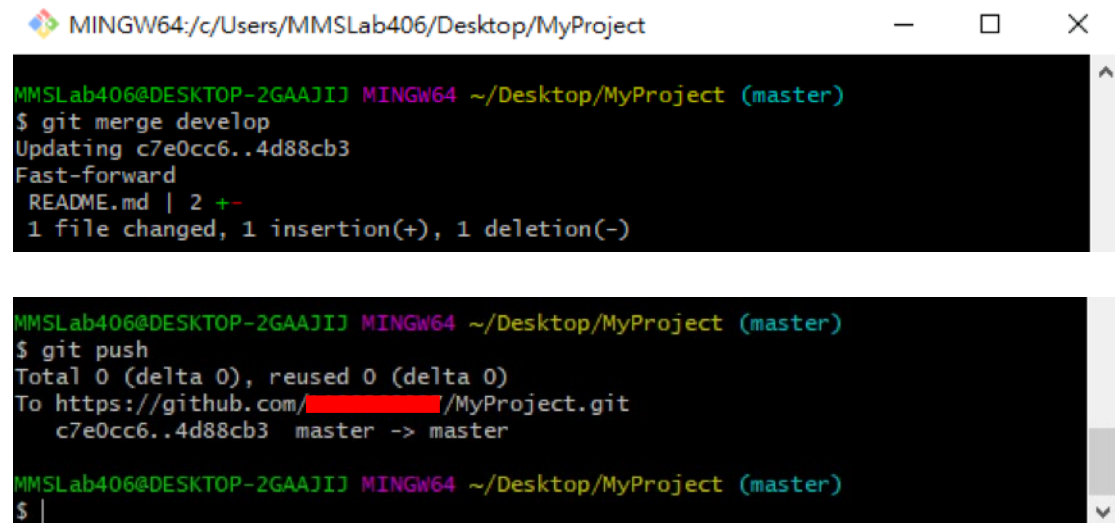


圖 0-9 確認目前資料

Step3 合併分支、提交版本、上傳到遠端資料庫，如圖 0-10 所示。

```
$ git merge develop
```

```
$ git push
```



The screenshot shows a terminal window titled 'MINGW64:/c:/Users/MMSLab406/Desktop/MyProject'. It displays the execution of two Git commands. The first command is 'git merge develop', which results in a fast-forward merge of 'c7e0cc6..4d88cb3' and updates 'README.md' with 2 changes (1 insertion, 1 deletion). The second command is 'git push', which successfully pushes the changes to the remote repository at 'https://github.com/[redacted]/MyProject.git'.

```
MINGW64:/c:/Users/MMSLab406/Desktop/MyProject
MMSLab406@DESKTOP-2GAAJIJ MINGW64 ~/Desktop/MyProject (master)
$ git merge develop
Updating c7e0cc6..4d88cb3
Fast-forward
 README.md | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

MMSLab406@DESKTOP-2GAAJIJ MINGW64 ~/Desktop/MyProject (master)
$ git push
Total 0 (delta 0), reused 0 (delta 0)
To https://github.com/[redacted]/MyProject.git
 c7e0cc6..4d88cb3 master -> master

MMSLab406@DESKTOP-2GAAJIJ MINGW64 ~/Desktop/MyProject (master)
$
```

圖 0-10 合併、提交、上傳

Step4 打開資料夾確認 README 目前的資料，會發現已經變成 develop 分支的資料，這就是將 develop 合併到 master，如圖 0-11 所示。



圖 0-11

0.1.4 解決分支合併時的衝突(Conflict)

Step1 打開資料夾修改 README 的資料並儲存，如圖 0-12 所示。

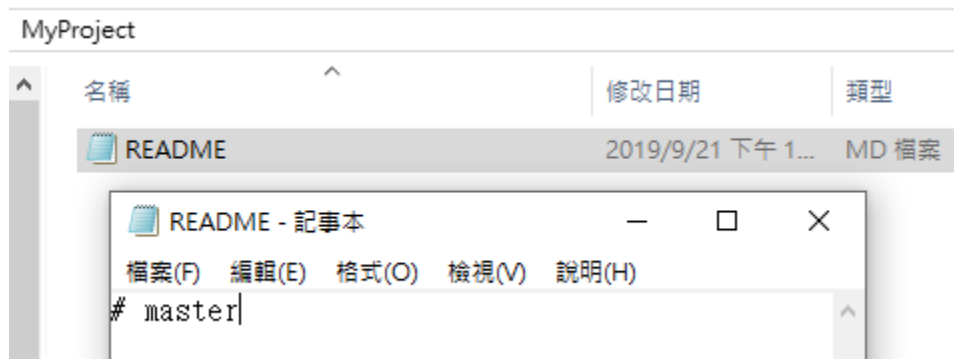


圖 0-12 修改資料

Step2 檢查狀態、加入提交名單、提交，如圖 0-13 所示。

```
$ git status  
  
$ git add .  
  
$ git commit -m "master"
```

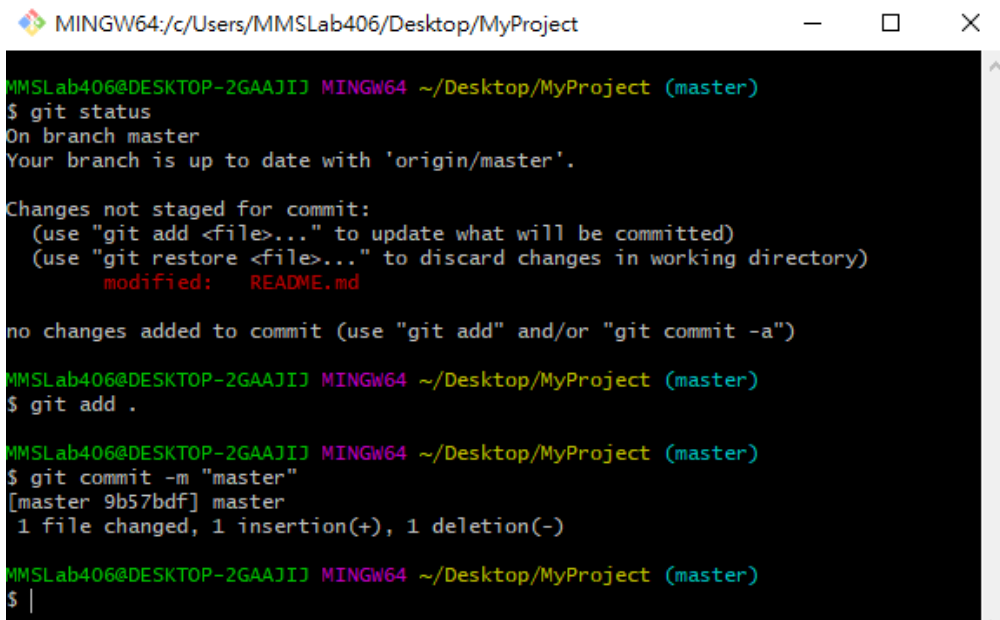
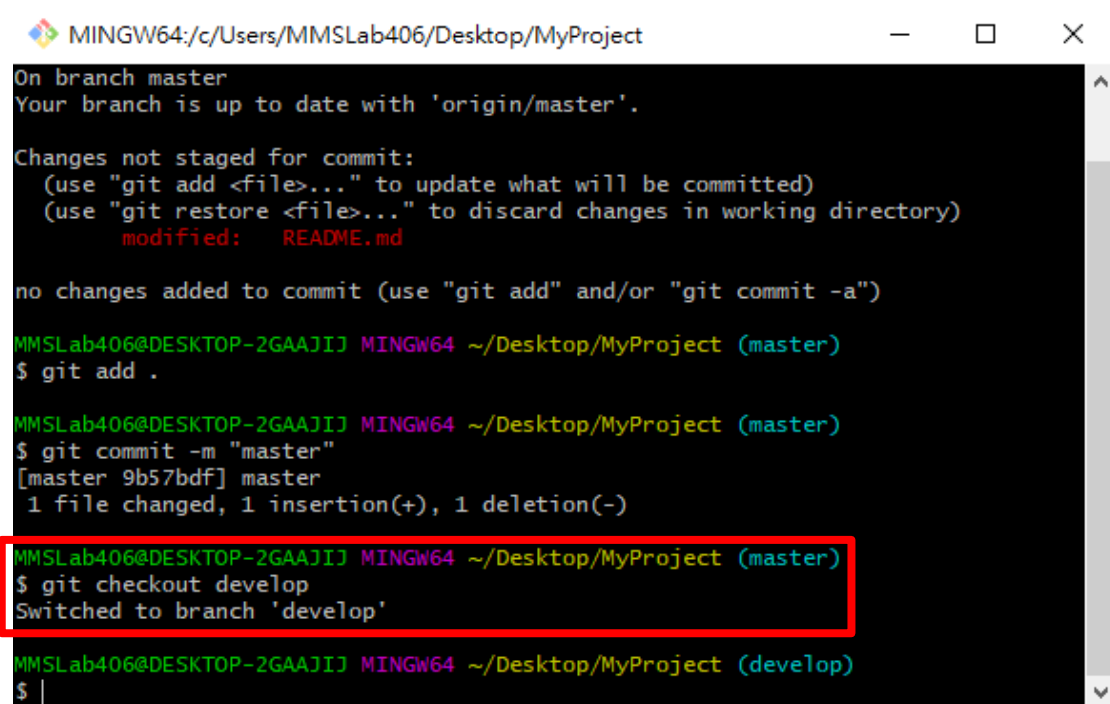


圖 0-13 檢查、加入、提交

Step3 切換分支到 develop，如圖 0-14 所示。

```
$ git checkout develop
```

A terminal window titled 'MINGW64:/c/Users/MMSLab406/Desktop/MyProject' with standard window controls. The terminal output shows the user on the 'master' branch. They run 'git add .' and 'git commit -m "master"', which creates a commit with hash '9b57bdf'. Then, they run 'git checkout develop', which switches them to the 'develop' branch. This last command and its output are highlighted with a red rectangular box.

```
MINGW64:/c/Users/MMSLab406/Desktop/MyProject
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")

MMSLab406@DESKTOP-2GAAJIJ MINGW64 ~/Desktop/MyProject (master)
$ git add .

MMSLab406@DESKTOP-2GAAJIJ MINGW64 ~/Desktop/MyProject (master)
$ git commit -m "master"
[master 9b57bdf] master
1 file changed, 1 insertion(+), 1 deletion(-)

MMSLab406@DESKTOP-2GAAJIJ MINGW64 ~/Desktop/MyProject (master)
$ git checkout develop
Switched to branch 'develop'

MMSLab406@DESKTOP-2GAAJIJ MINGW64 ~/Desktop/MyProject (develop)
$ |
```

圖 0-14 切換分支

Step4 打開資料夾修改 README 的資料並儲存，如圖 0-15 所示。

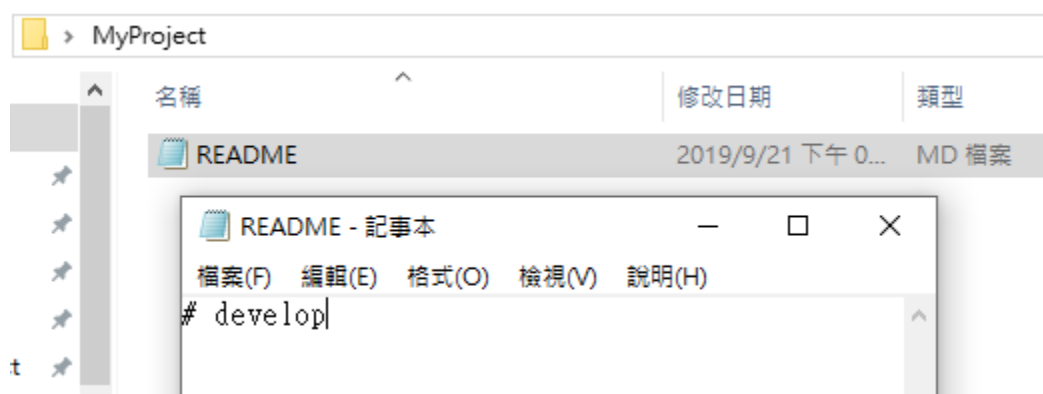


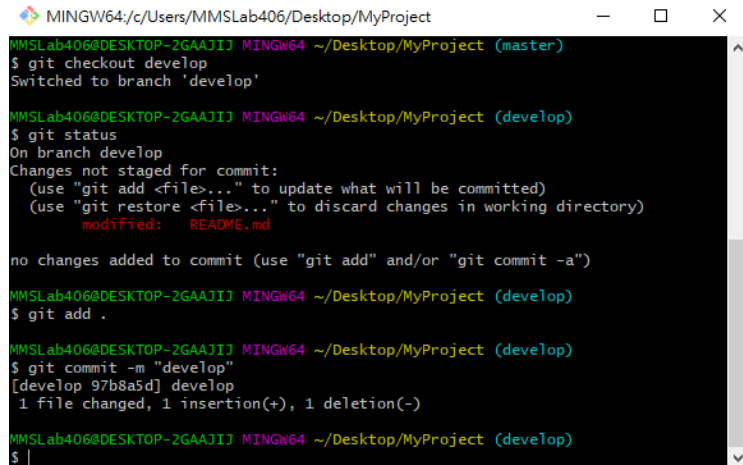
圖 0-15 修改資料

Step5 檢查狀態、加入提交名單、提交，如圖 0-16 所示。

```
$ git status
```

```
$ git add .
```

```
$ git commit -m "develop"
```



```
MINGW64/c/Users/MMSLab406/Desktop/MyProject
MMSLab406@DESKTOP-2GAAJIJ MINGW64 ~/Desktop/MyProject (master)
$ git checkout develop
Switched to branch 'develop'

MMSLab406@DESKTOP-2GAAJIJ MINGW64 ~/Desktop/MyProject (develop)
$ git status
On branch develop
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")

MMSLab406@DESKTOP-2GAAJIJ MINGW64 ~/Desktop/MyProject (develop)
$ git add .

MMSLab406@DESKTOP-2GAAJIJ MINGW64 ~/Desktop/MyProject (develop)
$ git commit -m "develop"
[develop 97b8a5d] develop
1 file changed, 1 insertion(+), 1 deletion(-)

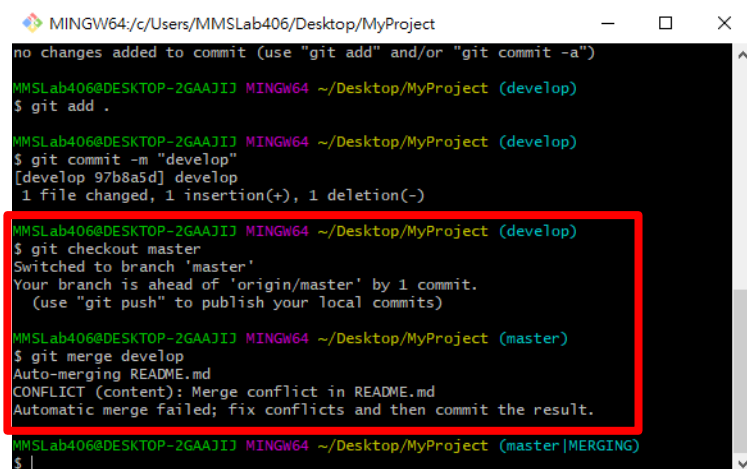
MMSLab406@DESKTOP-2GAAJIJ MINGW64 ~/Desktop/MyProject (develop)
$
```

圖 0-16 檢查、加入、提交

Step6 切換分支到 master 並合併 develop，會產生衝突，因為不同的分支修改了相同的地方，這時就需要解開衝突，保留必要的資料，如圖 0-17 所示。

```
$ git checkout master
```

```
$ git merge develop
```



```
MINGW64/c/Users/MMSLab406/Desktop/MyProject
no changes added to commit (use "git add" and/or "git commit -a")

MMSLab406@DESKTOP-2GAAJIJ MINGW64 ~/Desktop/MyProject (develop)
$ git add .

MMSLab406@DESKTOP-2GAAJIJ MINGW64 ~/Desktop/MyProject (develop)
$ git commit -m "develop"
[develop 97b8a5d] develop
1 file changed, 1 insertion(+), 1 deletion(-)

MMSLab406@DESKTOP-2GAAJIJ MINGW64 ~/Desktop/MyProject (develop)
$ git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

MMSLab406@DESKTOP-2GAAJIJ MINGW64 ~/Desktop/MyProject (master)
$ git merge develop
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.

MMSLab406@DESKTOP-2GAAJIJ MINGW64 ~/Desktop/MyProject (master|MERGING)
$
```

圖 0-17 產生衝突

Step7 打開資料會有衝突標籤，可以看到不同分支衝突的範圍，如圖 0-18 所示。

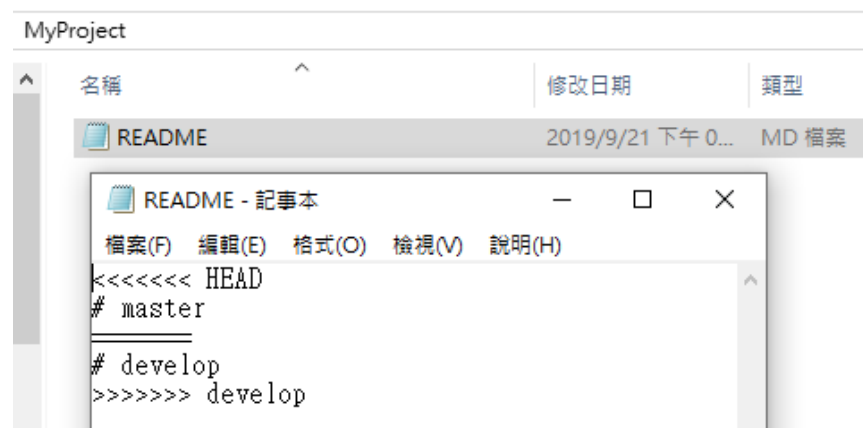


圖 0-18 衝突標籤

Step8 解決衝突保留 develop 的資料並儲存，如圖 0-19 所示。



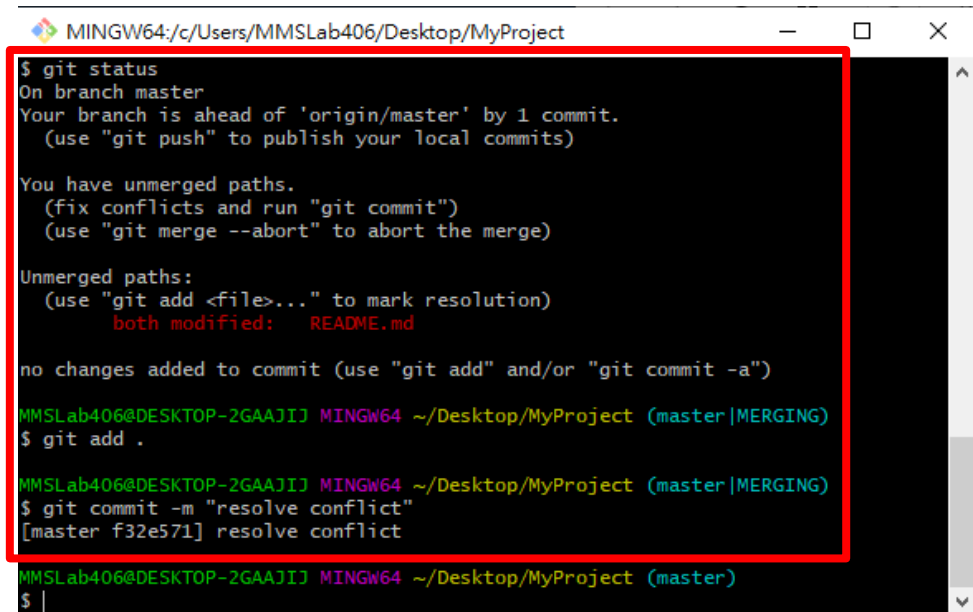
圖 0-19 解決衝突

Step9 檢查狀態、加入提交名單、提交，如圖 0-20 所示。

```
$ git status
```

```
$ git add .
```

```
$ git commit -m "resolve conflict"
```



```
MINGW64:/c/Users/MMSLab406/Desktop/MyProject
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

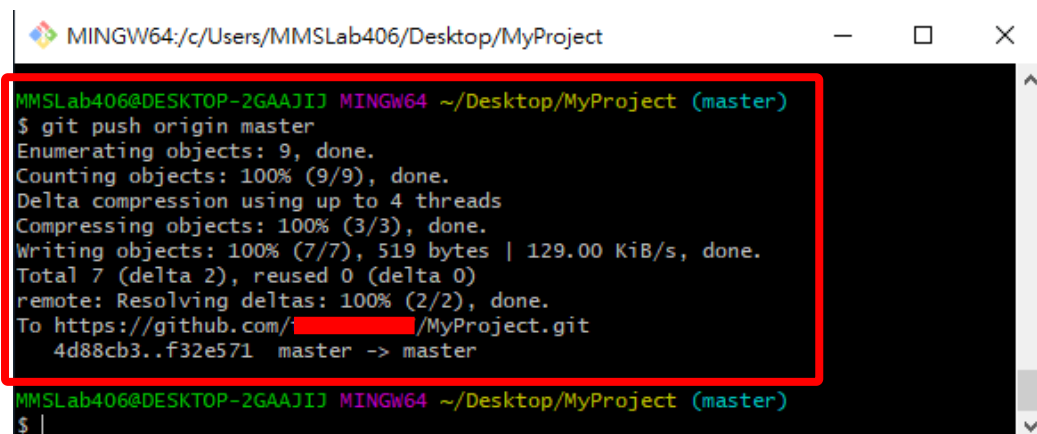
Unmerged paths:
  (use "git add <file>..." to mark resolution)
        both modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
MMSLab406@DESKTOP-2GAJJ1J MINGW64 ~/Desktop/MyProject (master|MERGING)
$ git add .
MMSLab406@DESKTOP-2GAJJ1J MINGW64 ~/Desktop/MyProject (master|MERGING)
$ git commit -m "resolve conflict"
[master f32e571] resolve conflict
MMSLab406@DESKTOP-2GAJJ1J MINGW64 ~/Desktop/MyProject (master)
$ |
```

圖 0-20 檢查、加入、提交

Step10 檔案推至遠端的 GitHub，如圖 0-21 所示。

```
$ git push origin master
```



```
MINGW64:/c/Users/MMSLab406/Desktop/MyProject
MMSLab406@DESKTOP-2GAJJ1J MINGW64 ~/Desktop/MyProject (master)
$ git push origin master
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (7/7), 519 bytes | 129.00 KiB/s, done.
Total 7 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/[redacted]/MyProject.git
  4d88cb3..f32e571  master -> master
MMSLab406@DESKTOP-2GAJJ1J MINGW64 ~/Desktop/MyProject (master)
$ |
```

圖 0-21 檔案上傳成功

Step11 回瀏覽器，F5 重新整理 GitHub 的頁面，可以看到剛剛上傳的檔案，如圖 0-22 所示。

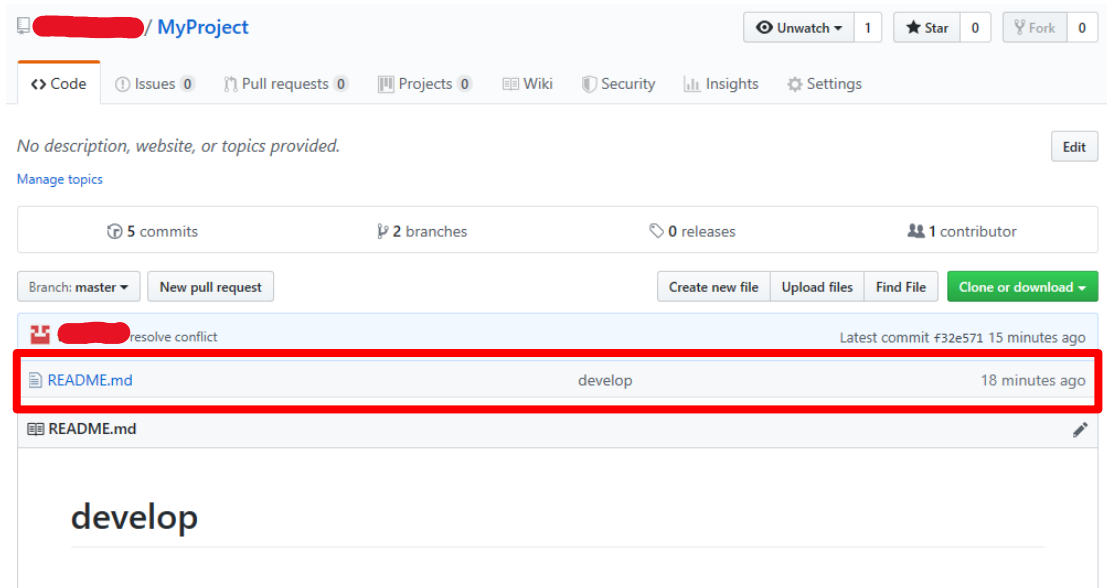


圖 0-22 查看 GitHub 上的專案

Step12 請開啟分支圖讓助教檢查，如圖 0-23 所示。

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) **[Insights](#)** [Settings](#)

[Pulse](#)
[Contributors](#)
[Community](#)
[Traffic](#)
[Commits](#)
[Code frequency](#)
[Dependency graph](#)
[Network](#)
[Forks](#)

Network graph

Timeline of the most recent commits to this repository and its network ordered by most

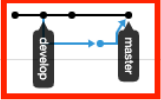
Owners	Sep
t108368007	10 

圖 0-23 查看分支圖