Lab0
Git - Branch Merge Conflict

本節目的：

- 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

- 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 創立新的分支

Step1 開啟 Lab0-1 的 MyProject 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
```

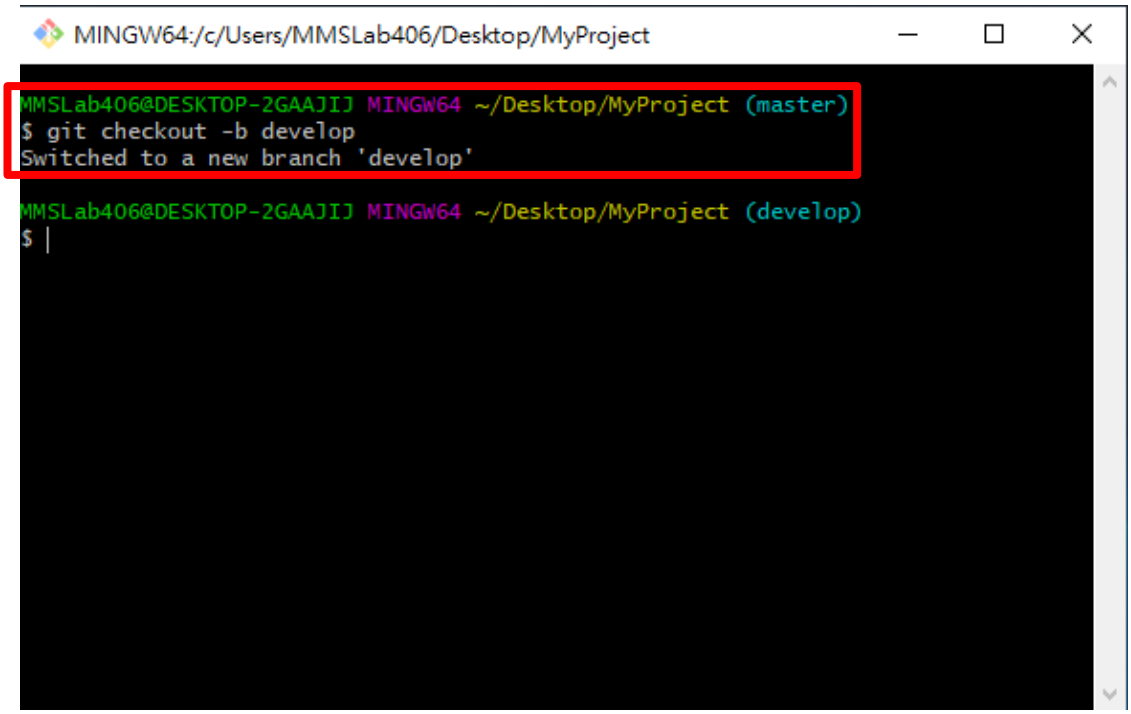

圖 0-1 新增分支並切換

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

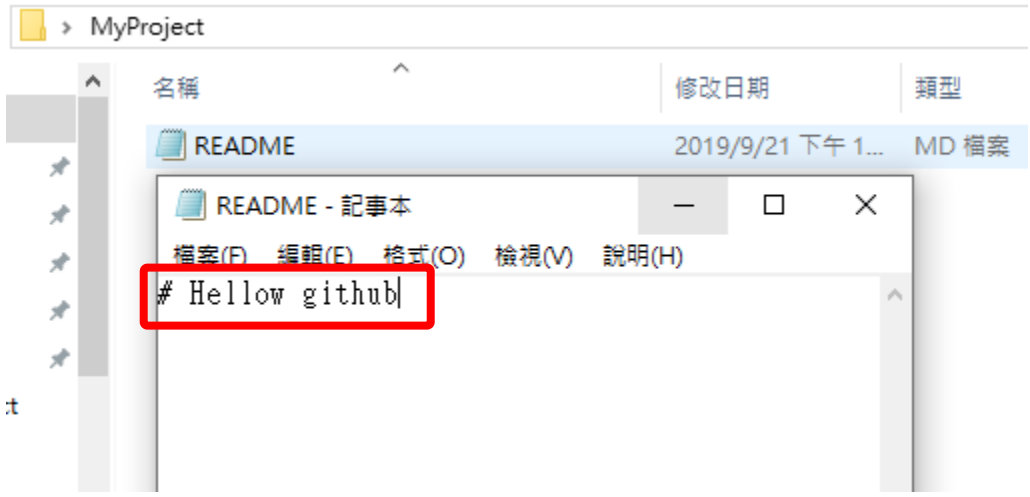

圖 0-2 修改 README 檔

Step3 檢查資料狀態，有一個修改的檔案，如圖 0-3 所示。

```
$ git status
```

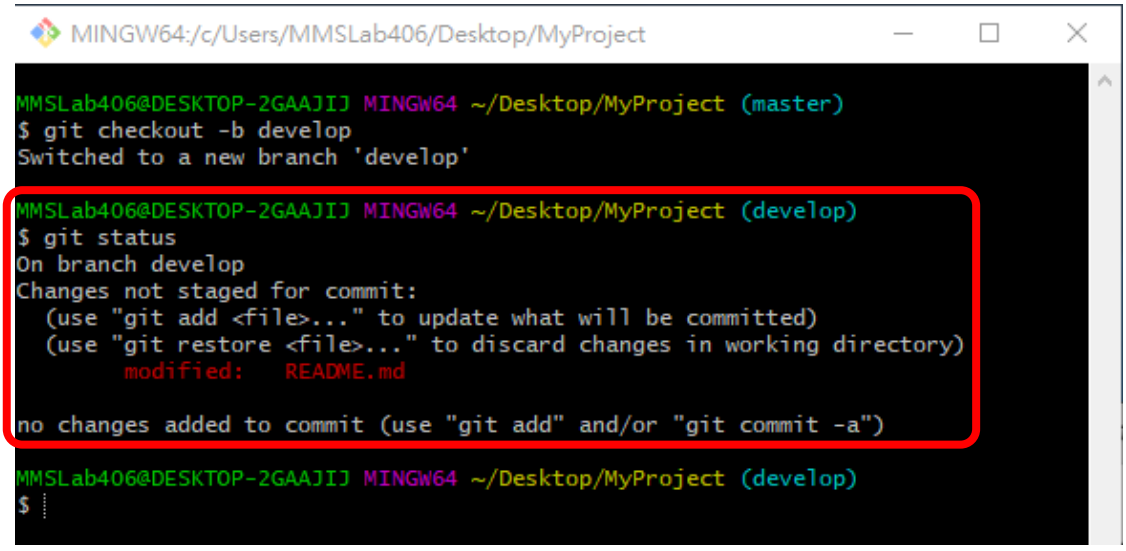

圖 0-3 檢查資料狀態

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”
```

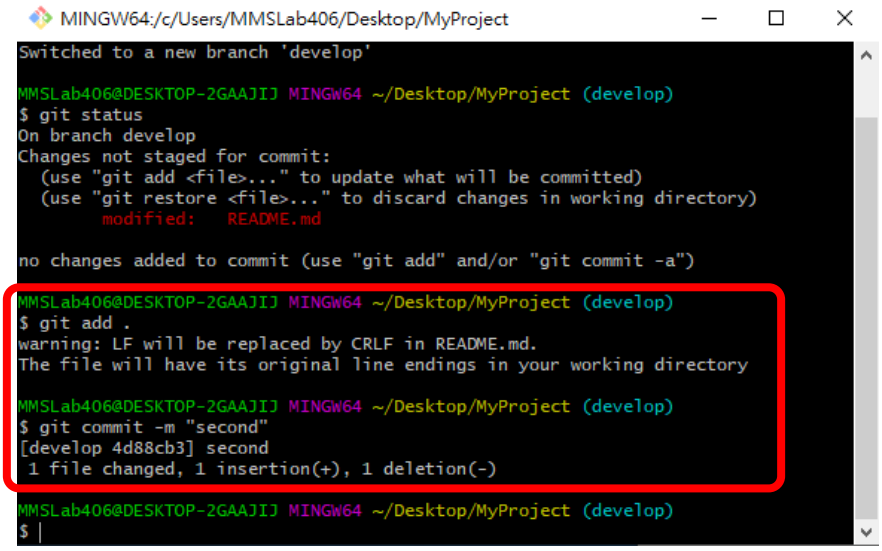

圖 0-4 加入並提交

Step5 提交到遠端資料庫（GitHub）上，如圖 0-5 所示。

```
$ git push origin develop
```

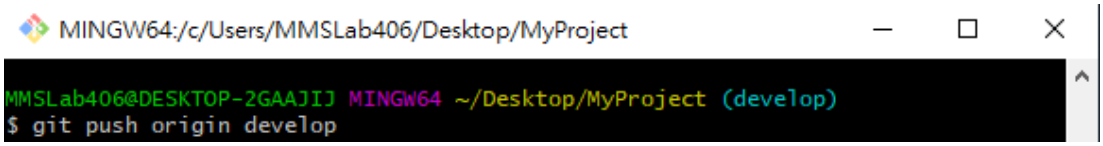

圖 0-5 提交到遠端資料庫

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
```

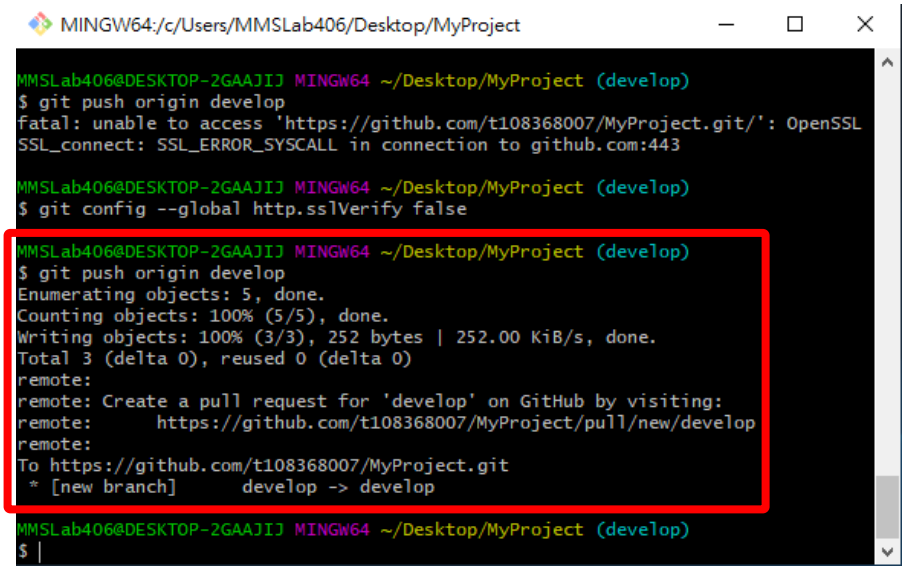

圖 0-6 檔案上傳成功

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。

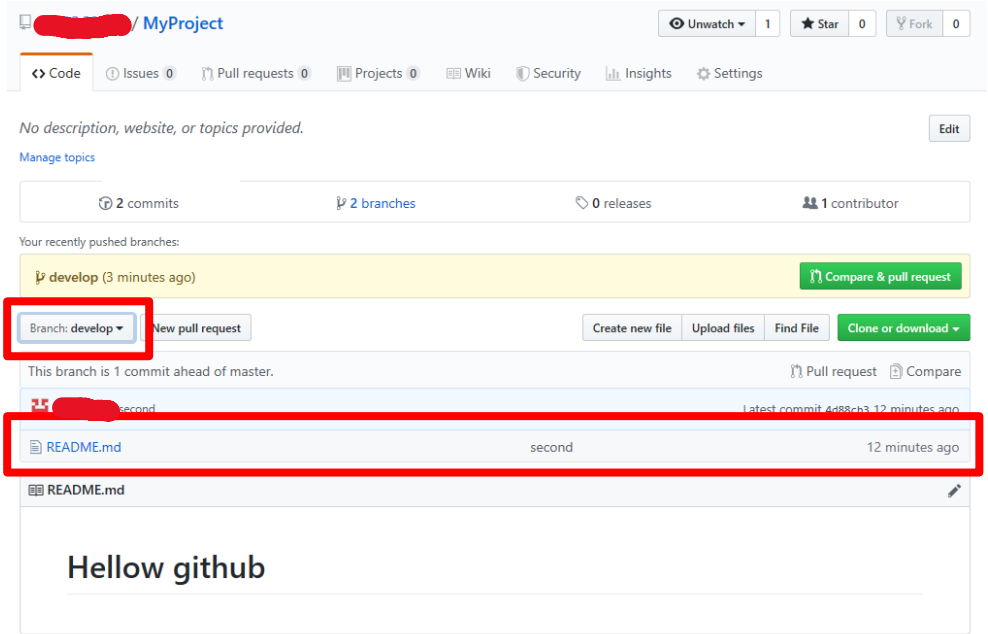

圖 0-7 查看 GitHub 上的專案

0.1.2 合併(Merge)分支

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

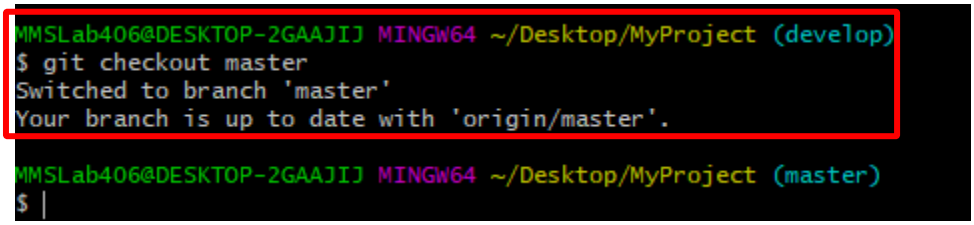

圖 0-8 切換分支

Step2 打開資料夾確認 README 目前的資料，如圖 0-9 所示。

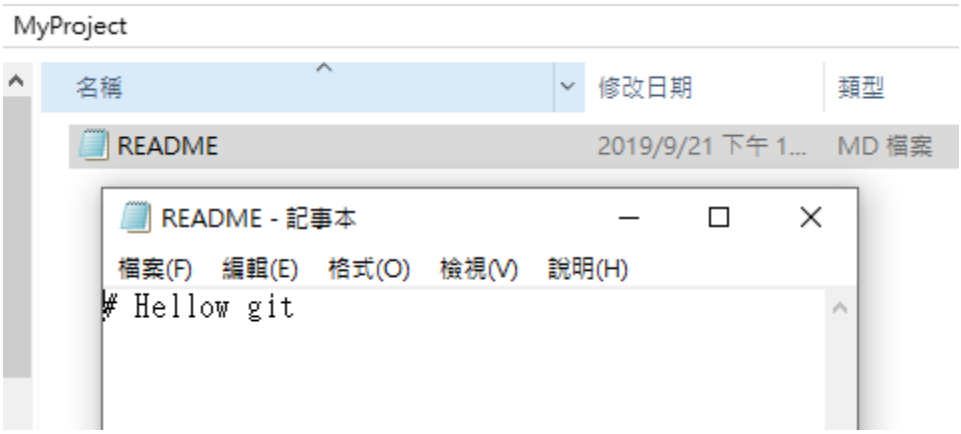

圖 0-9 確認目前資料

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
```

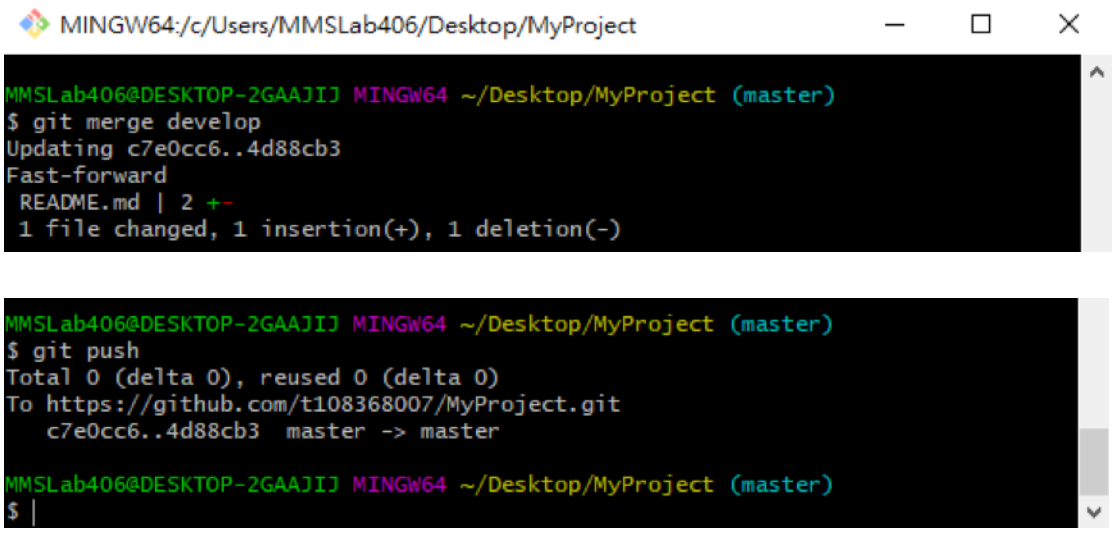

圖 0-10 合併、提交、上傳

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。

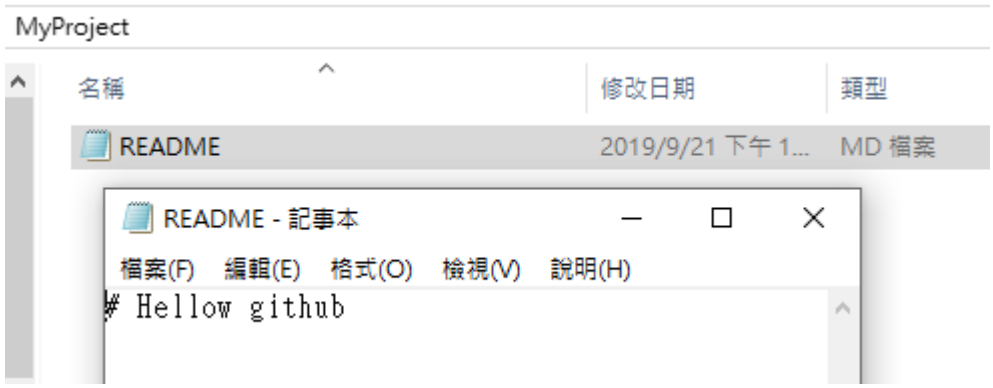

圖 0-11

0.1.3 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”
```

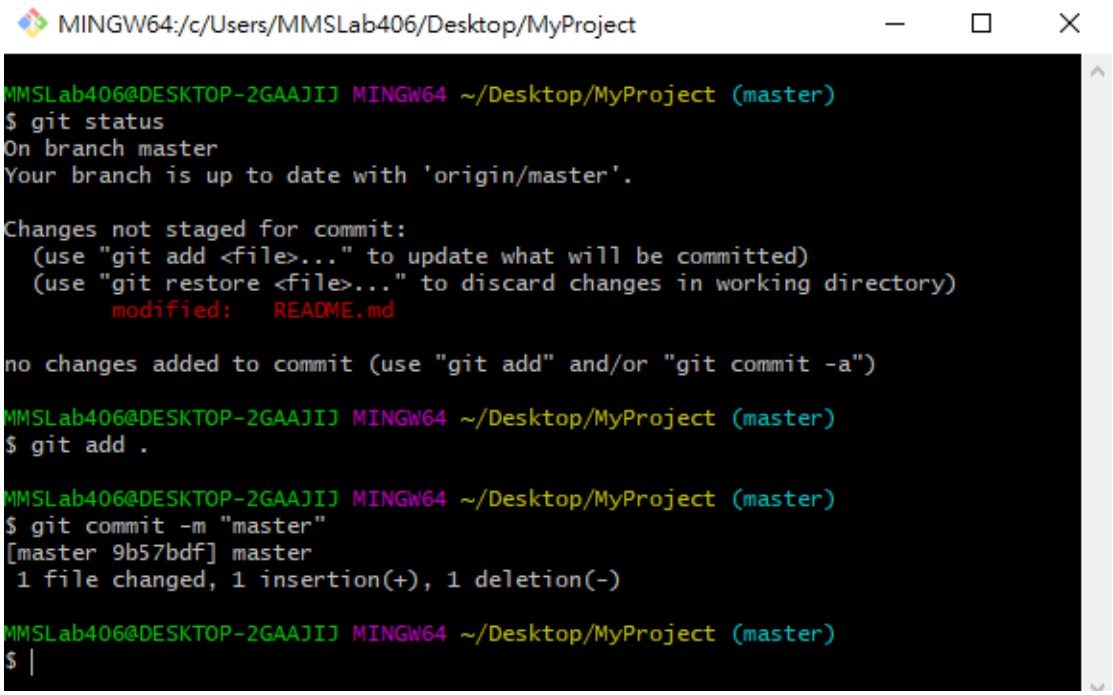

圖 0-13 檢查、加入、提交

Step3 切換分支到 develop，如圖 0-14 所示。

```
$ git checkout develop
```

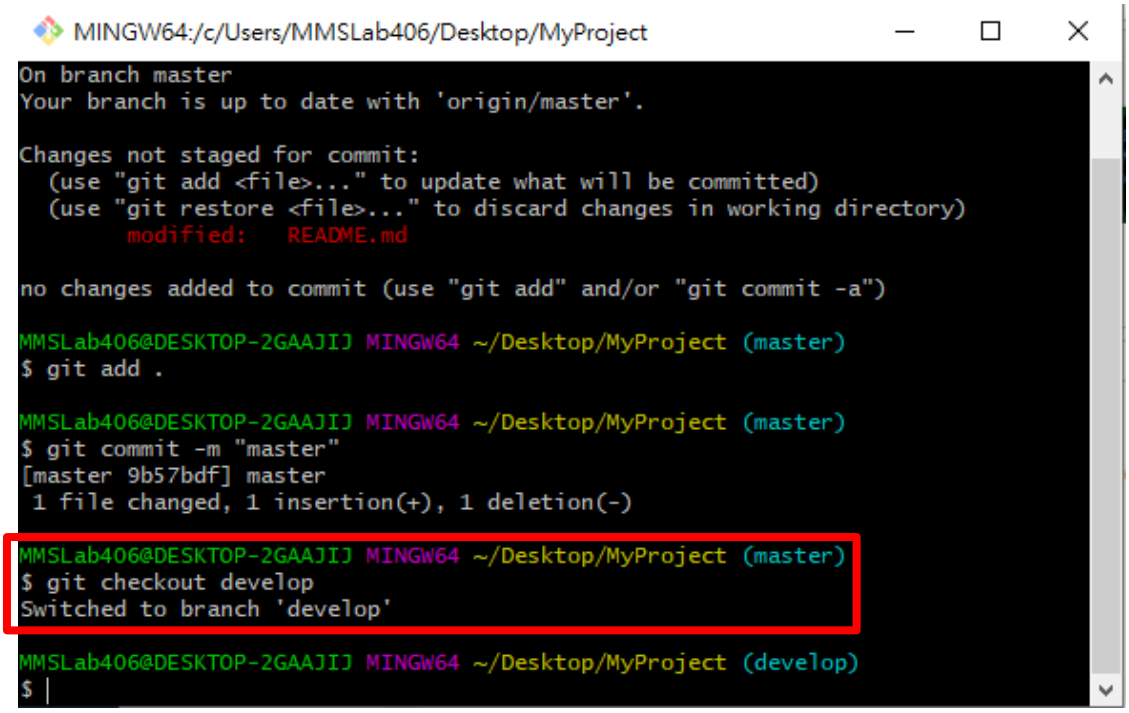

圖 0-14 切換分支

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”
```

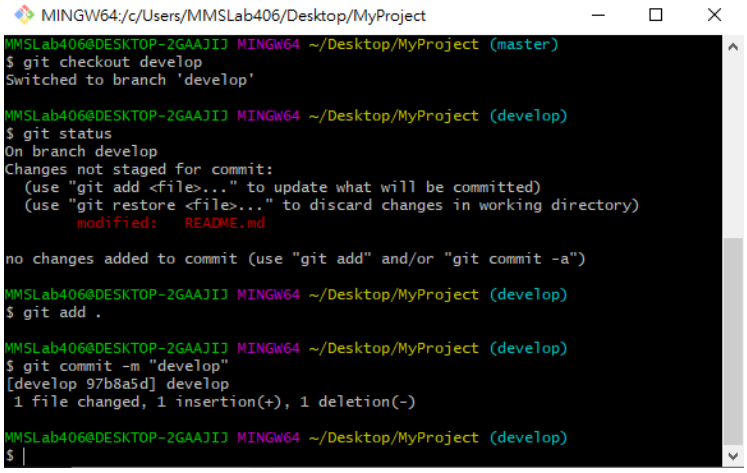

圖 0-16 檢查、加入、提交

Step6 切換分支到 master 並合併 develop，會發現產生衝突，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
```

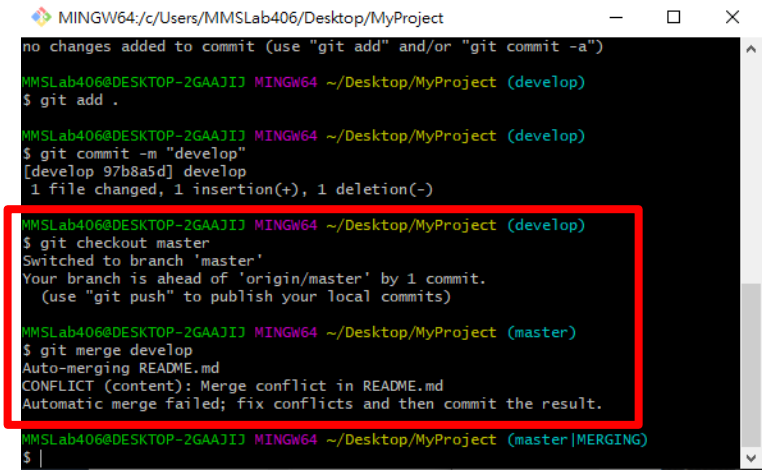

圖 0-17 產生衝突

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。

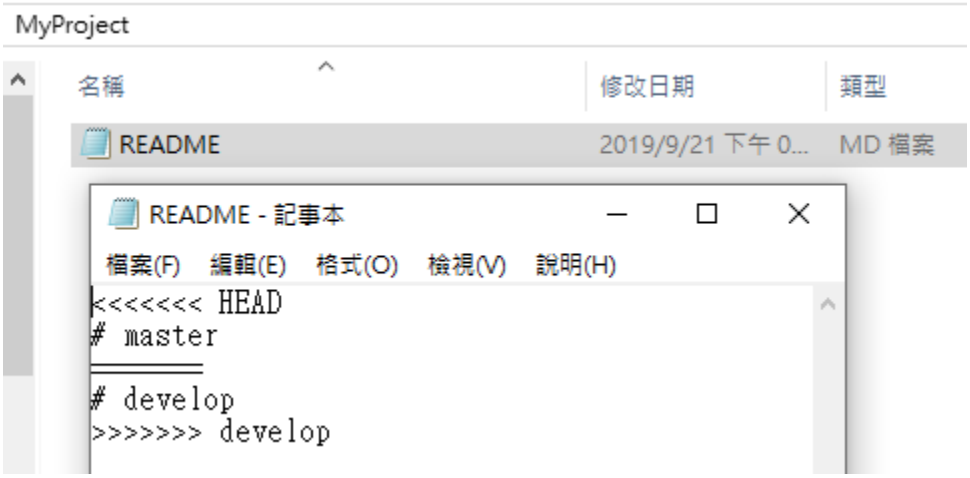

圖 0-18 衝突標籤

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”
```

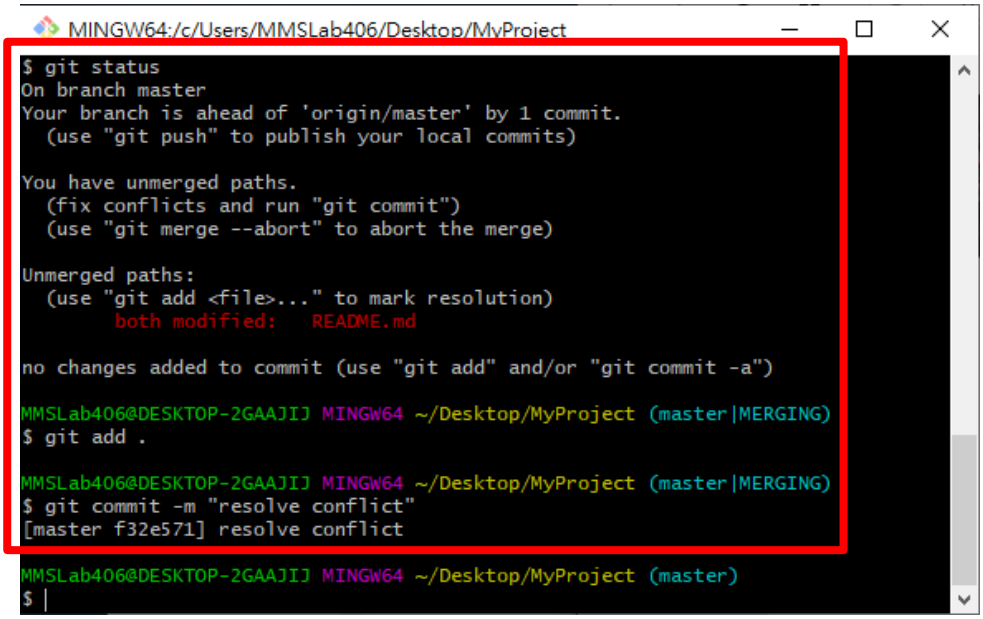

圖 0-20 檢查、加入、提交

Step10 檔案推至遠端的 GitHub，如圖 0-21 所示。

```
$ git push origin master
```

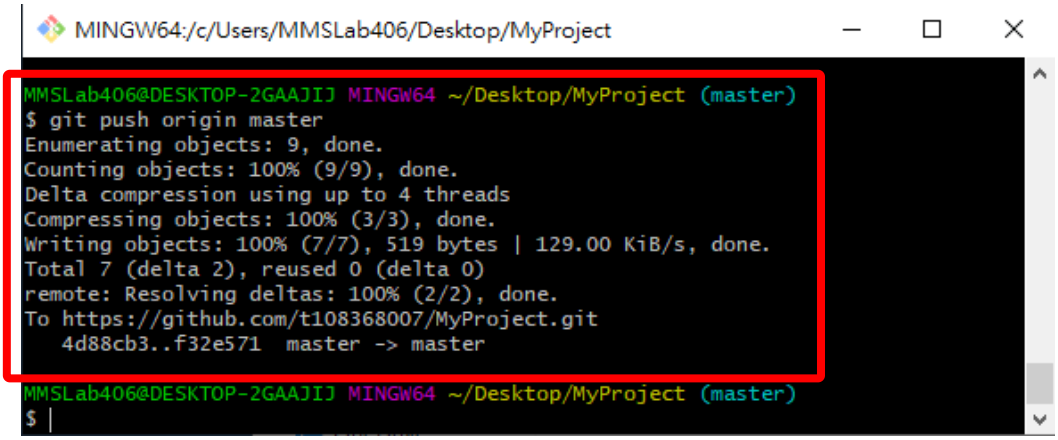

圖 0-21 檔案上傳成功

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。

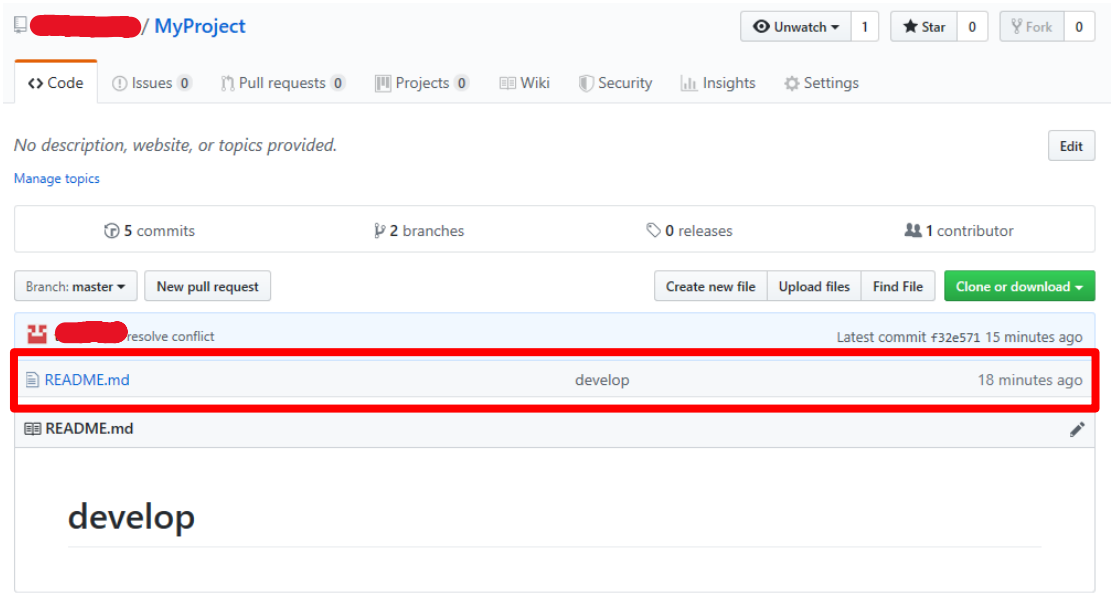

圖 0-22 查看 GitHub 上的專案