

Windows Using GitHub

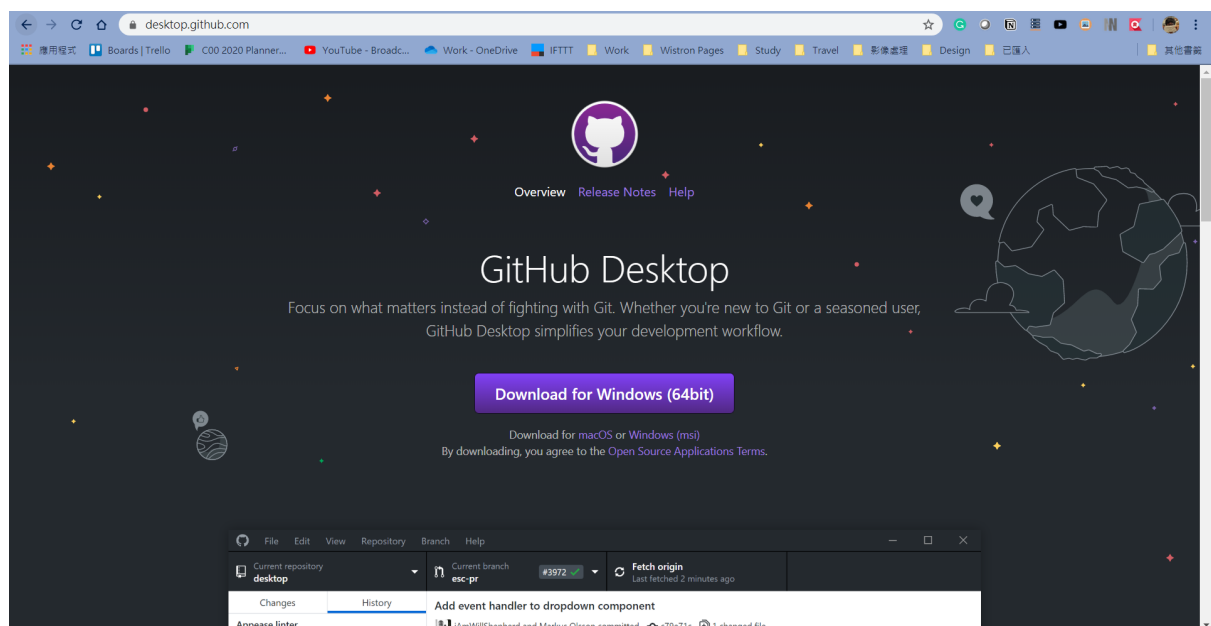
2020/3/6 Josh Hsi

這份文件分兩個部分說明Windows上怎麼使用GitHub 來放寫好的程式

1. 如何新增資料夾 or 檔案到GitHub上?
2. 如何使用Visual Studio Code寫Python & 把程式放上GitHub?

1. 如何新增資料夾 or 檔案到GitHub上?

Step1: 到網站下載GitHub Desktop版本



Step2: 開啟程式 & 執行



Sign in to GitHub.com

Username or email address

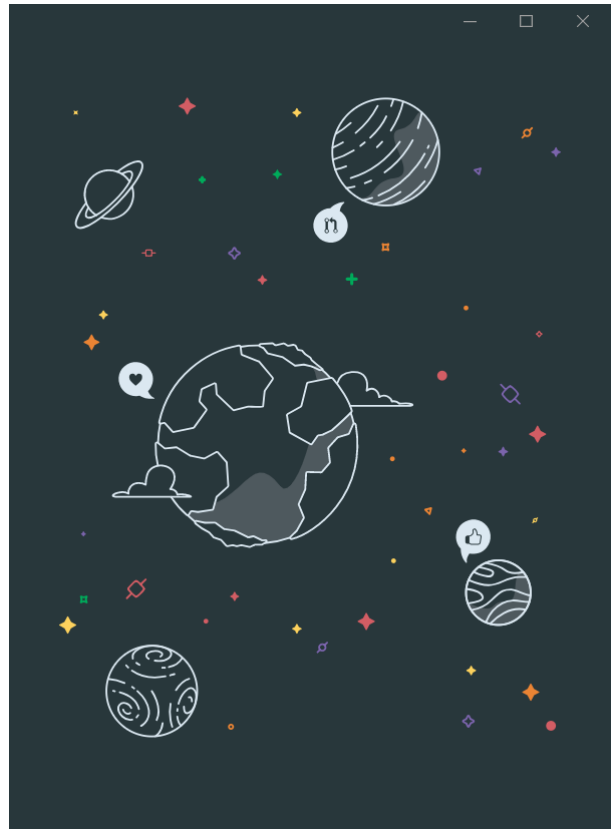
Password

[Sign in](#)

[Cancel](#)

[Forgot password?](#)

[Sign in using your browser](#)



Ste3: 登入



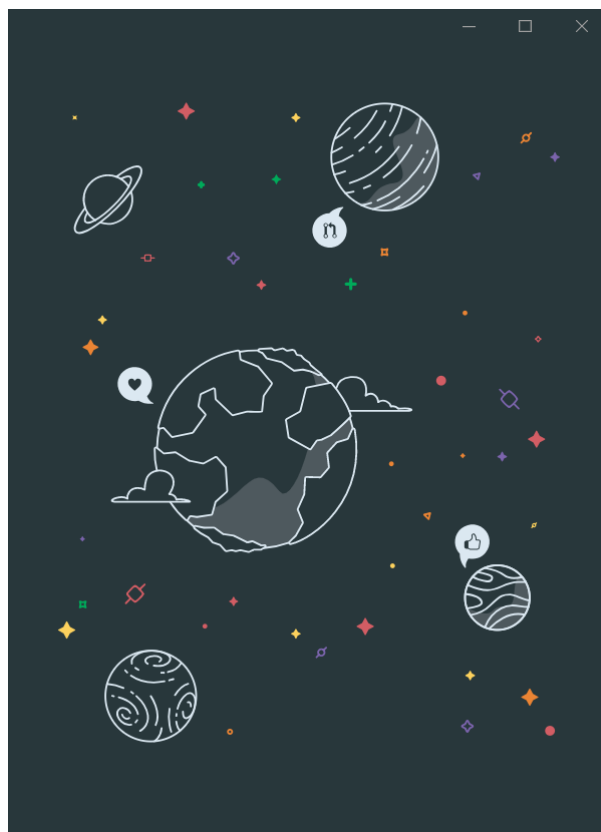
Make GitHub Desktop better!

Would you like to help us improve GitHub Desktop by periodically submitting [usage stats](#)?

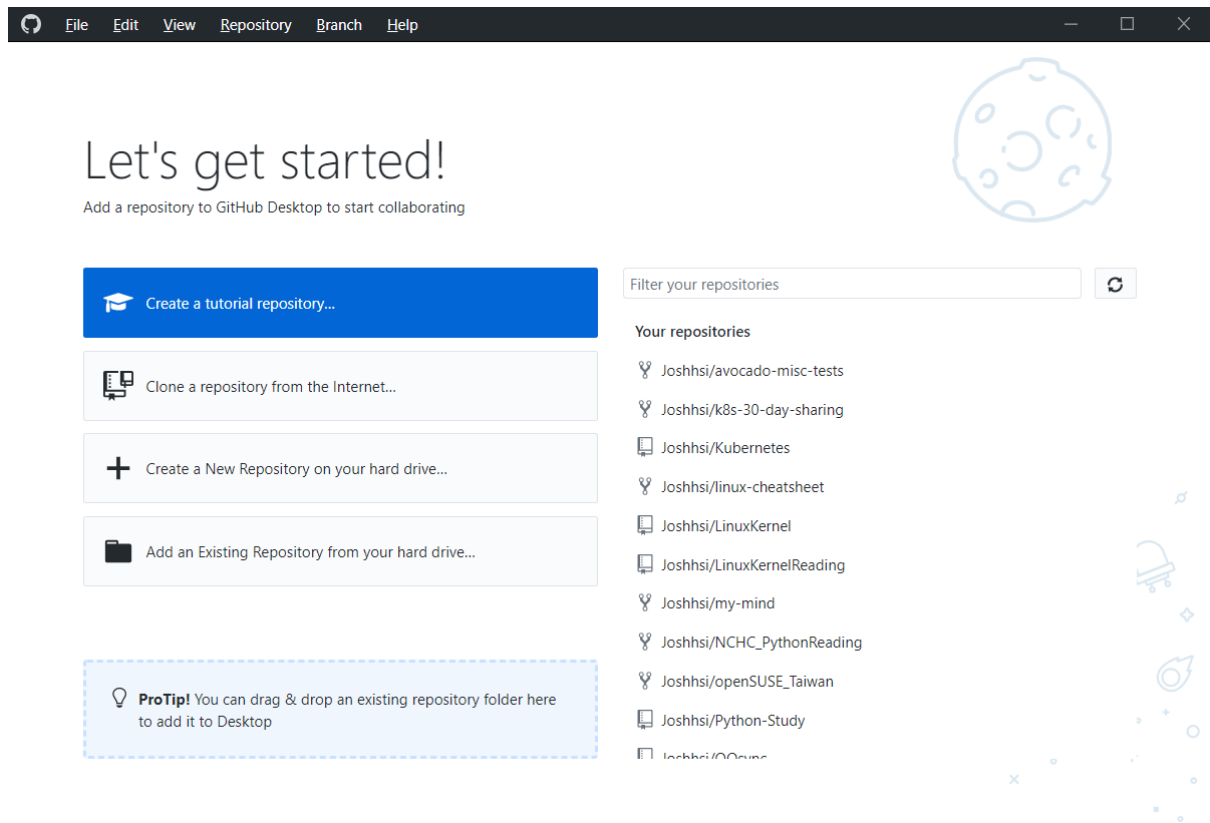
☒ Yes, submit periodic usage stats

Finish

Cancel

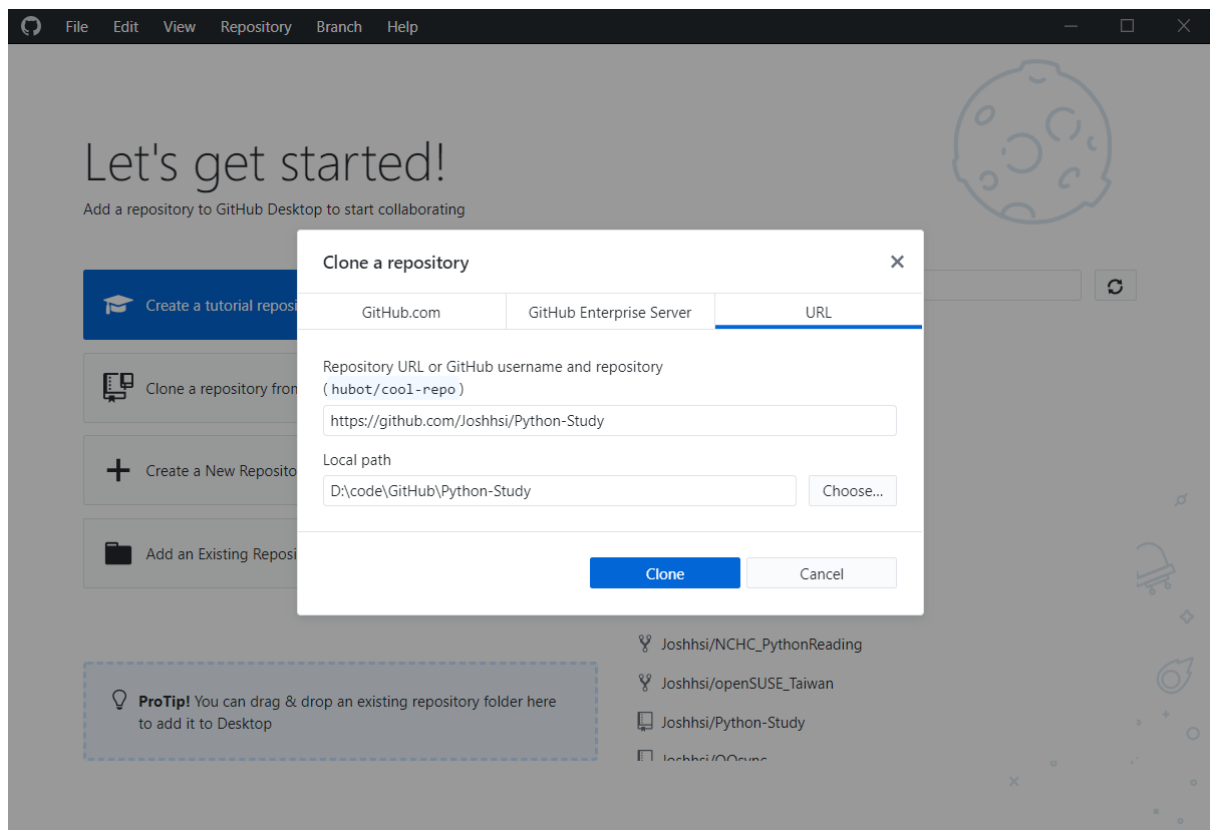


Step 4: 登入完成後會要你建立repo, 這邊選第二個"Clone a repo from the Internet"



Step5 : 選URL的方式Clone, 並且複製上想要clone的repo網址, 選擇Local要存放的位置

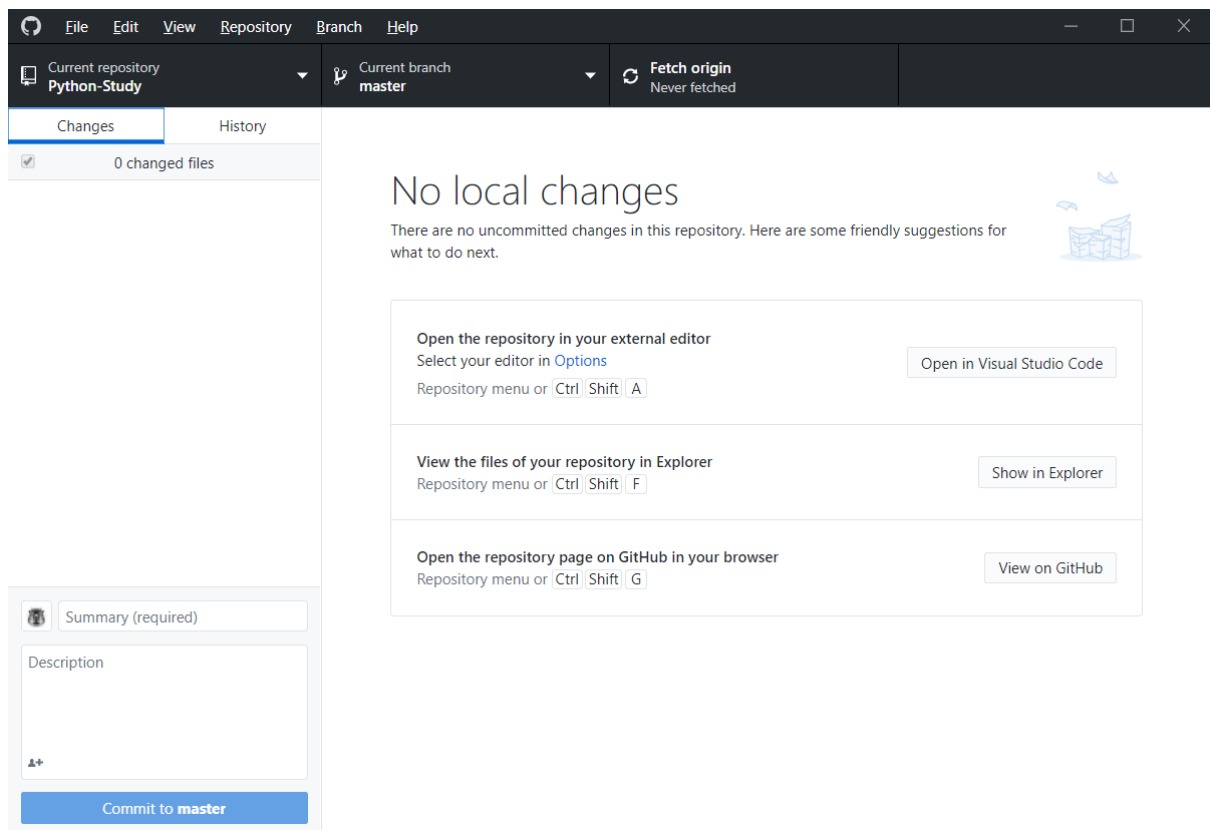
i.e. <https://github.com/Joshhsi/Python-Study>.



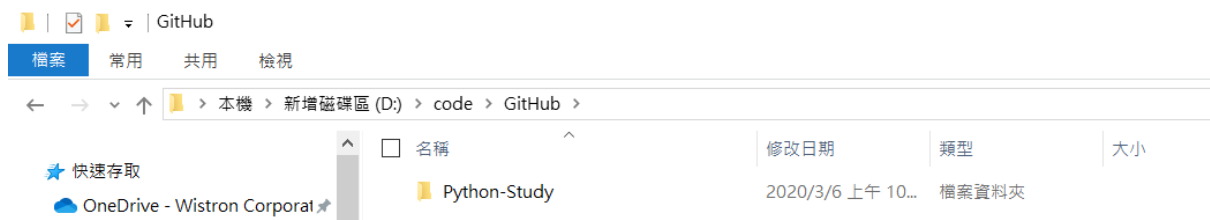
Step6: 按下Clone後就開始複製repo了!



Step7: Clone完成後就會看到這個畫面, 左上角的Current Repository可以看到 "Python-Study"



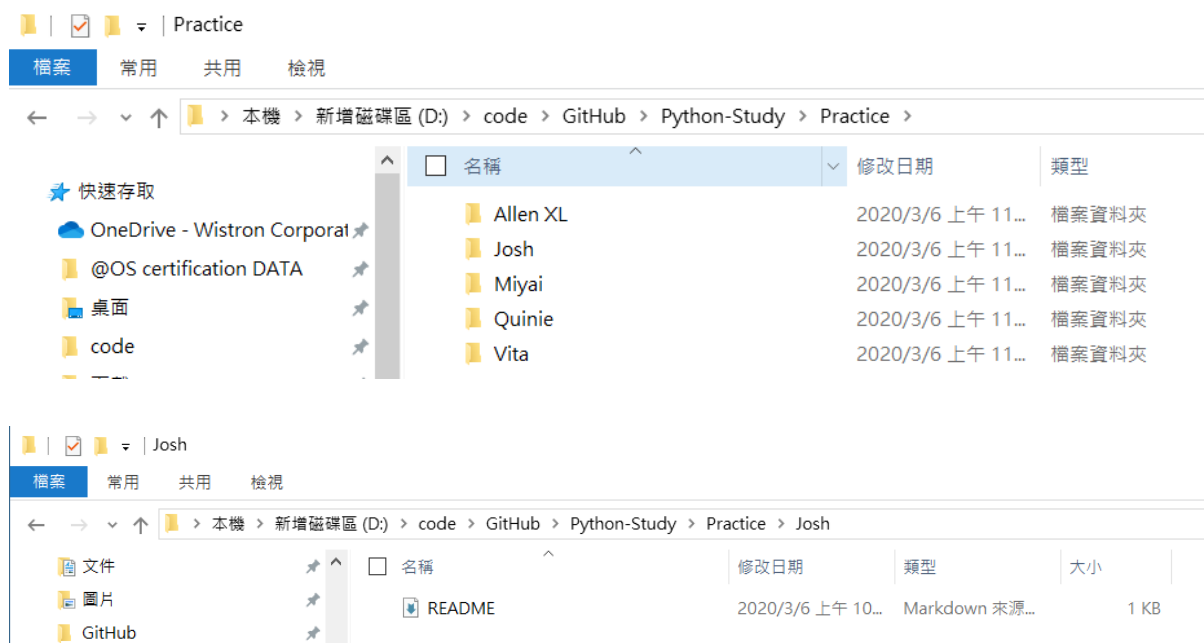
Step 8: 同時local的資料夾也可以看到"Python-Study"



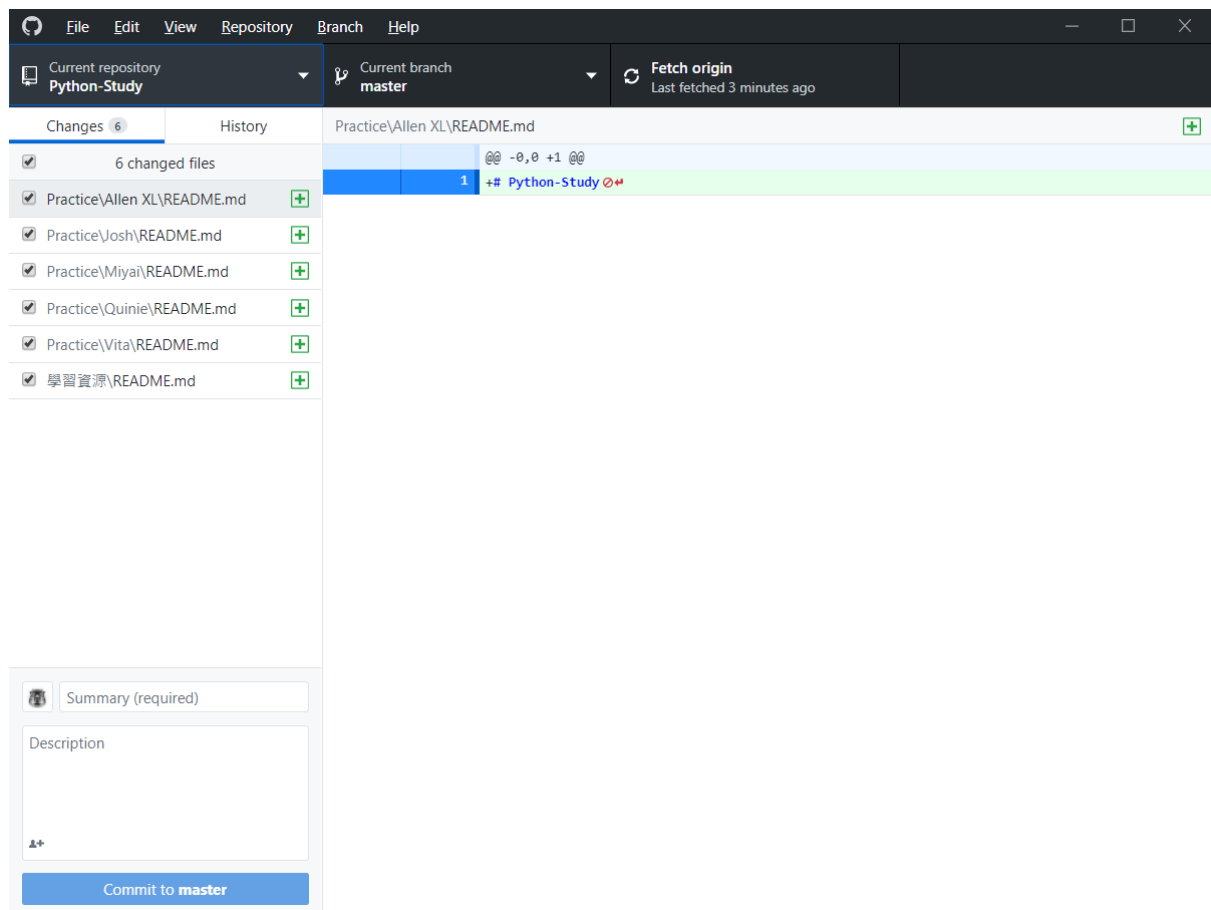
Step 9: 這時候就可以建立想要建立的資料夾目錄



目錄內不可以沒有檔案, 所以一開始會先放一個Readme在裡面



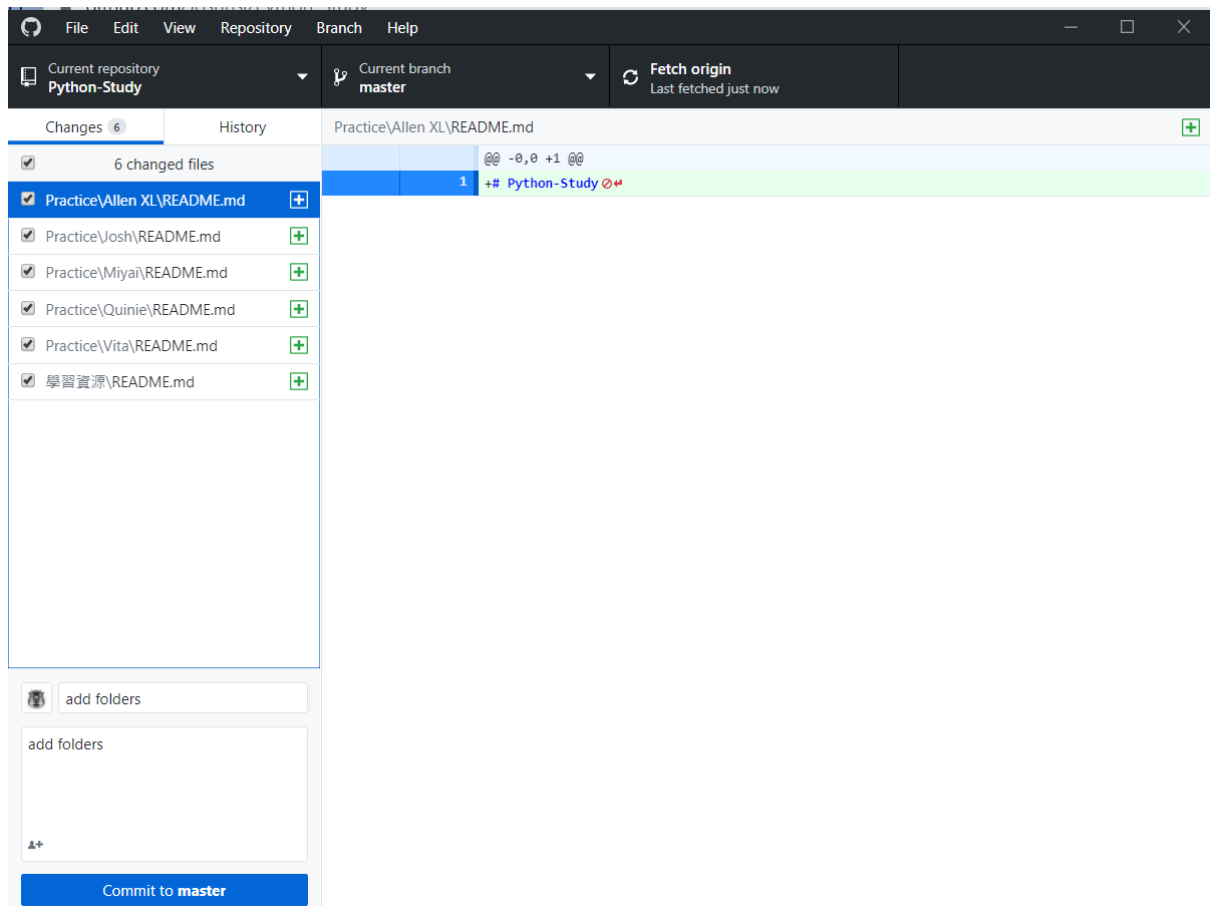
Step 10: 建立完資料夾回"GitHub Desktop"可以看到左邊的Changes多了幾個紀錄, 這是因為有檔案異動的關係



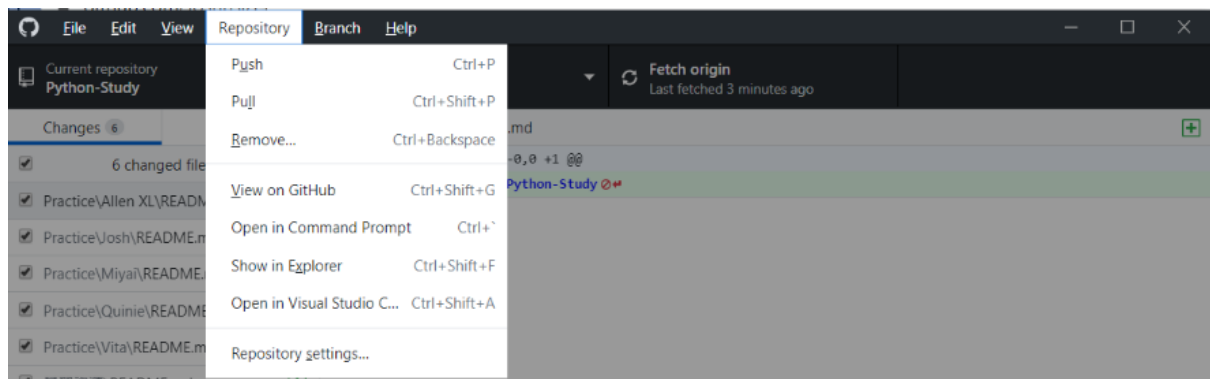
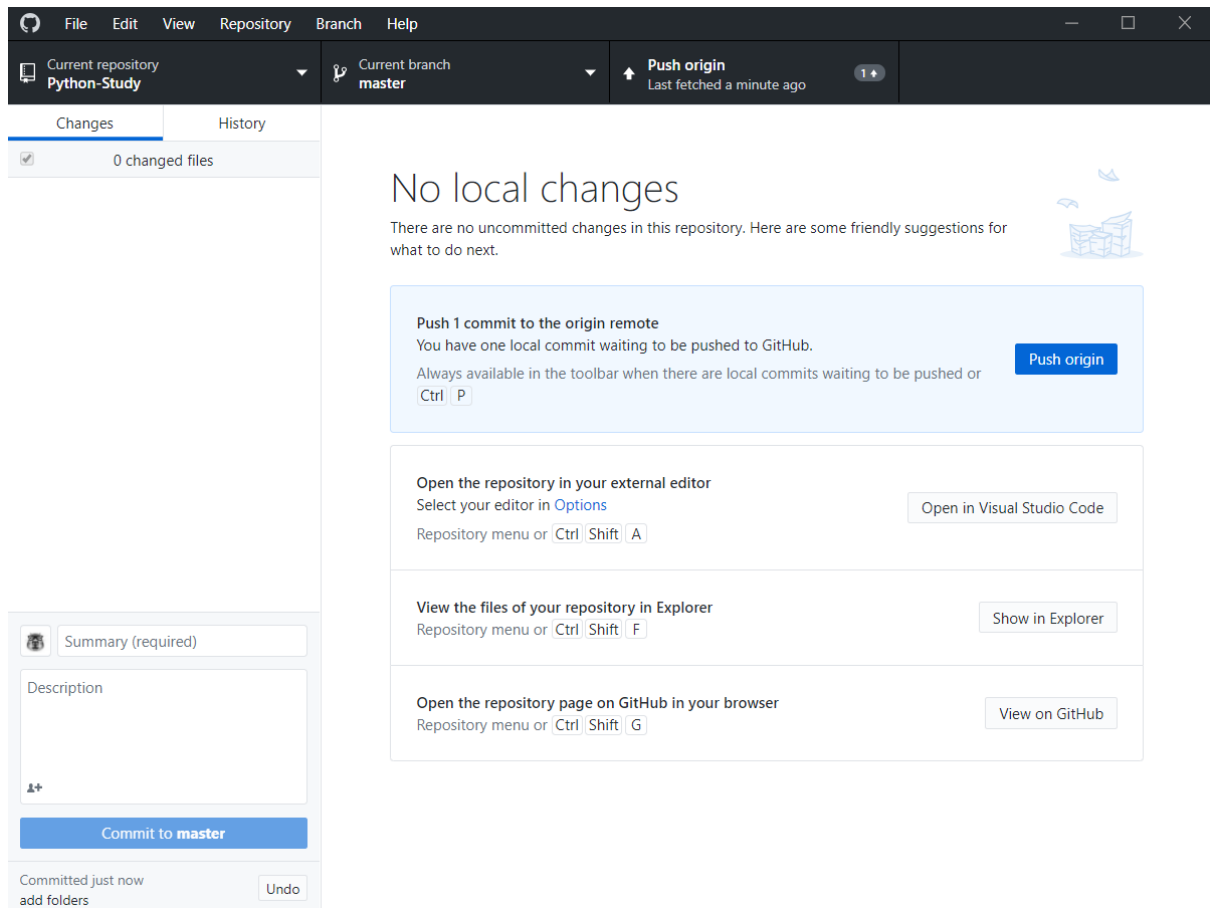
Step 11: 這時候就可以按下左下角的Commit



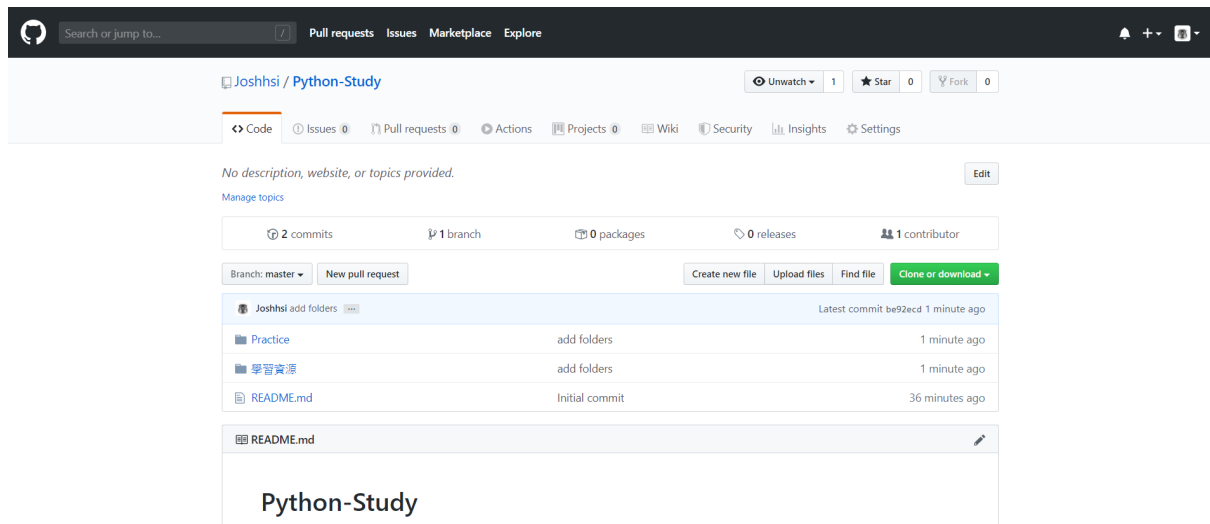
這一步是確認要把Local有異動的檔案更新到repo裡，但是還沒按下Push前還不會真的更新



Step 12: 按下Commit後會看到左邊的changes已經變成空的了, 但時這時候repo還沒被更新, 這時候可以按下上面Tab出現的 "Push origin", 或是更上面的頁籤"Repository" 裡面的 "Push" 來正式把更動的檔案更新到repo裡

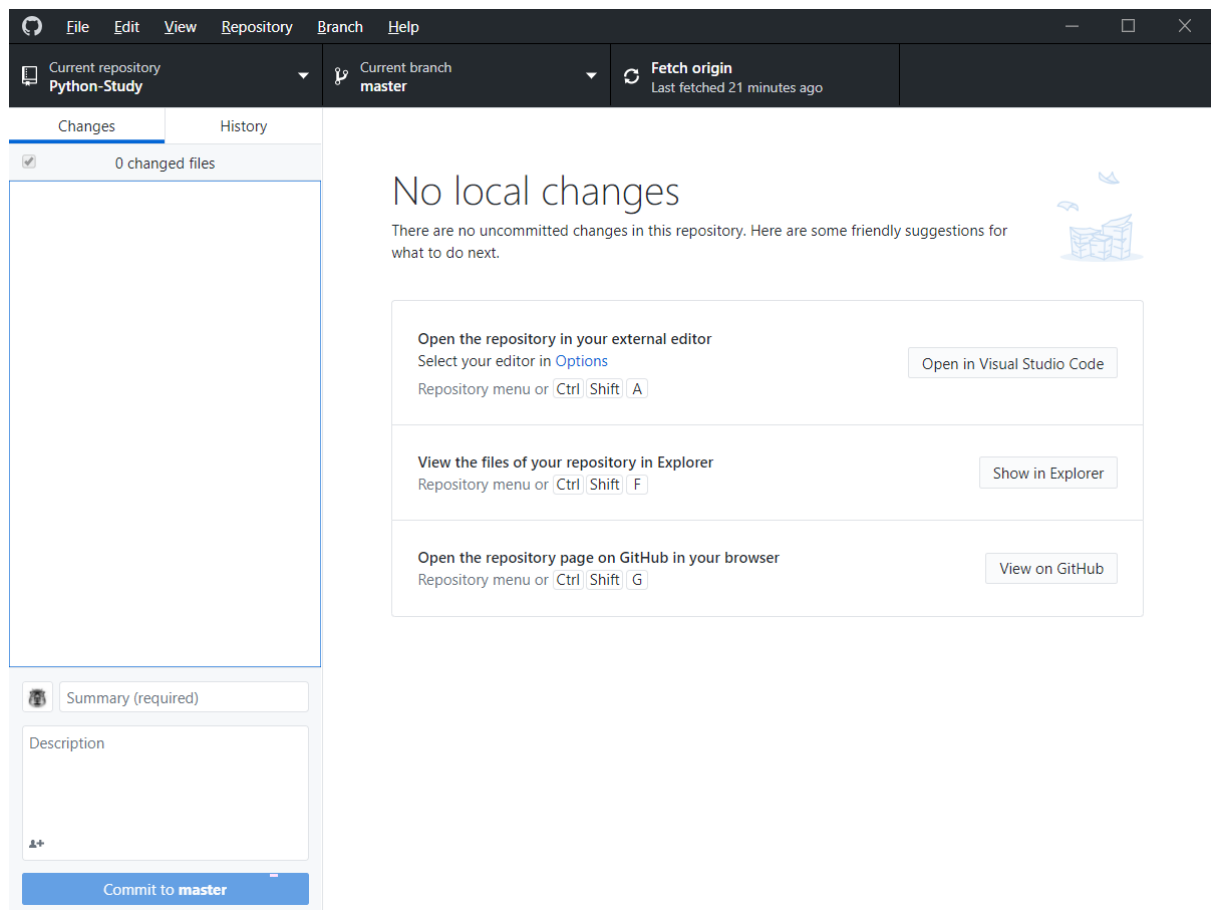


Step 13: Push完之後，回到網頁上看就會發現Local建立的資料夾已經放上GitHub了

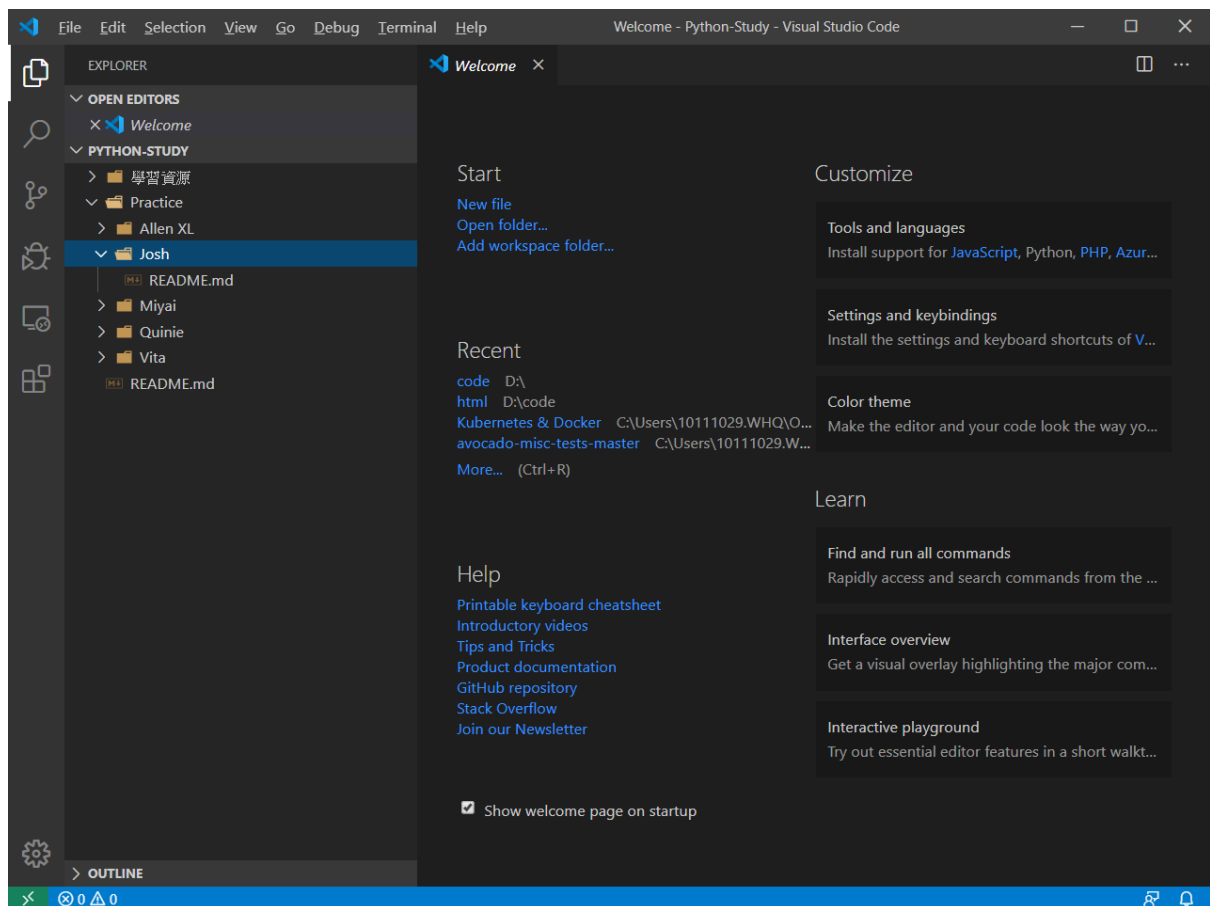


2. 如何使用Visual Studio Code寫Python & 把程式放上GitHub?

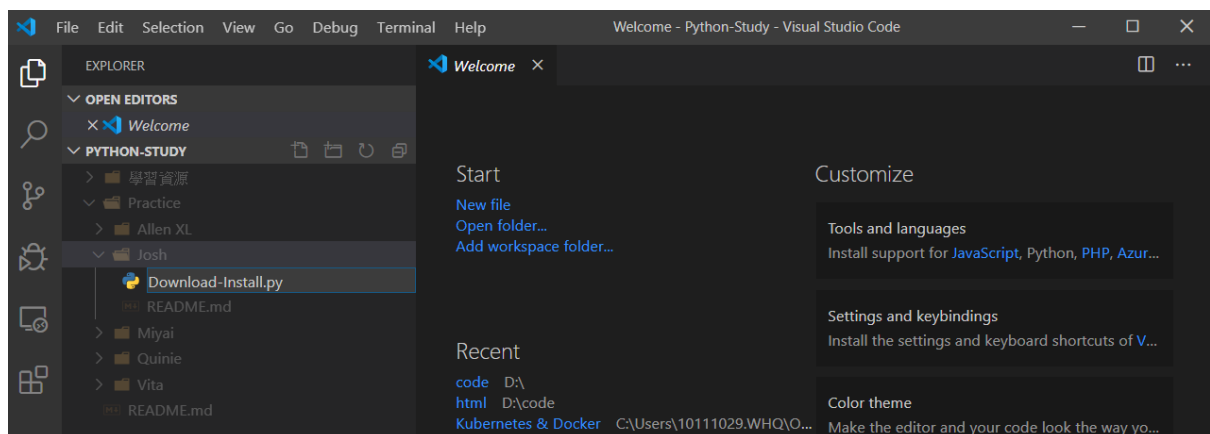
Step 1: 在"GitHub Desktop"的主頁面點選"Open in Visual Studio Code"



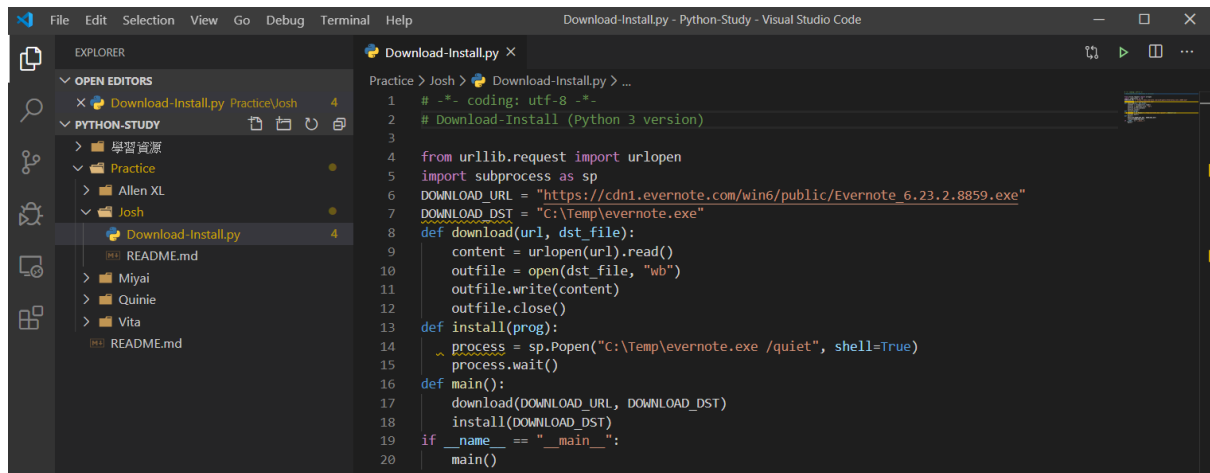
Step 2: 這時候會自動幫你開啟 VS Code, 並且會開啟GitHub的資料夾



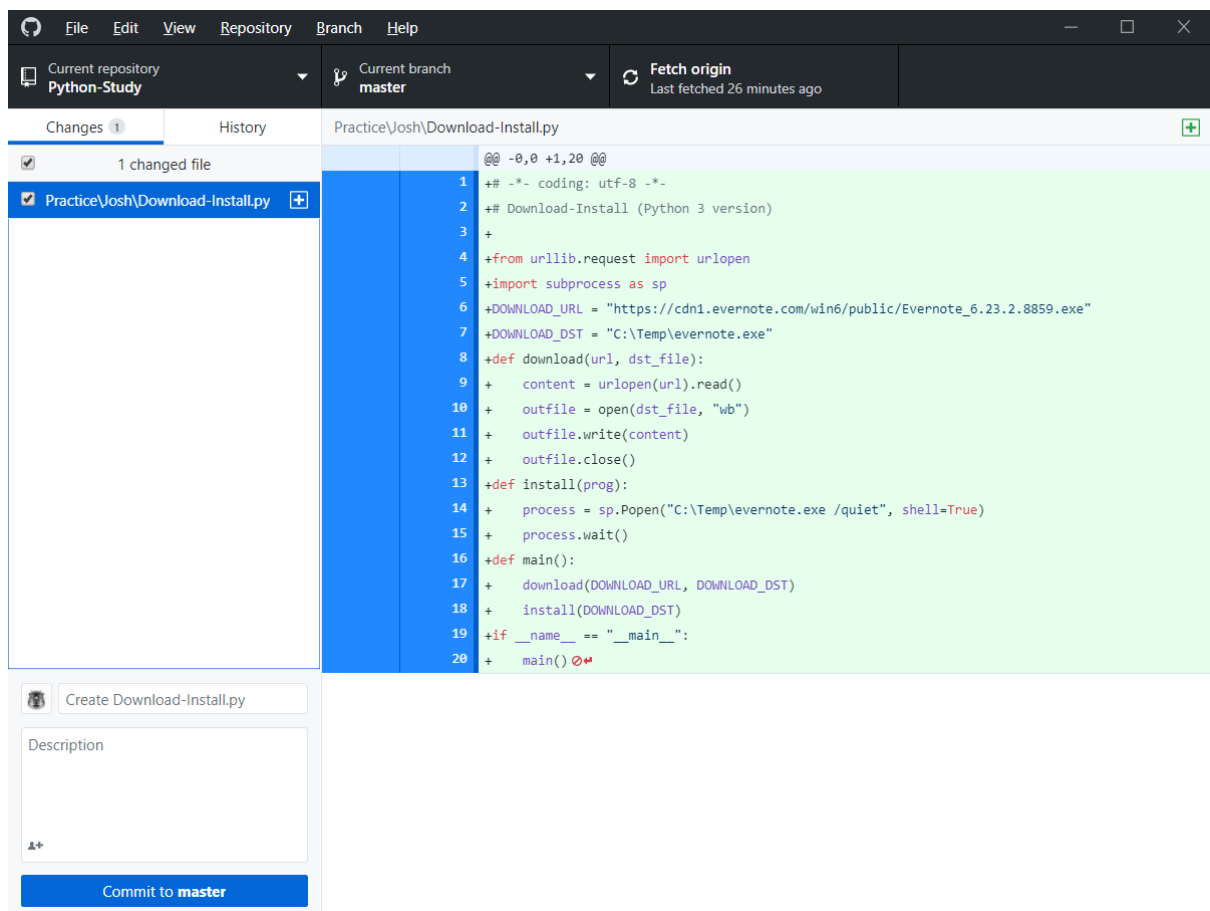
Step3 :這邊可以建立你的python程式



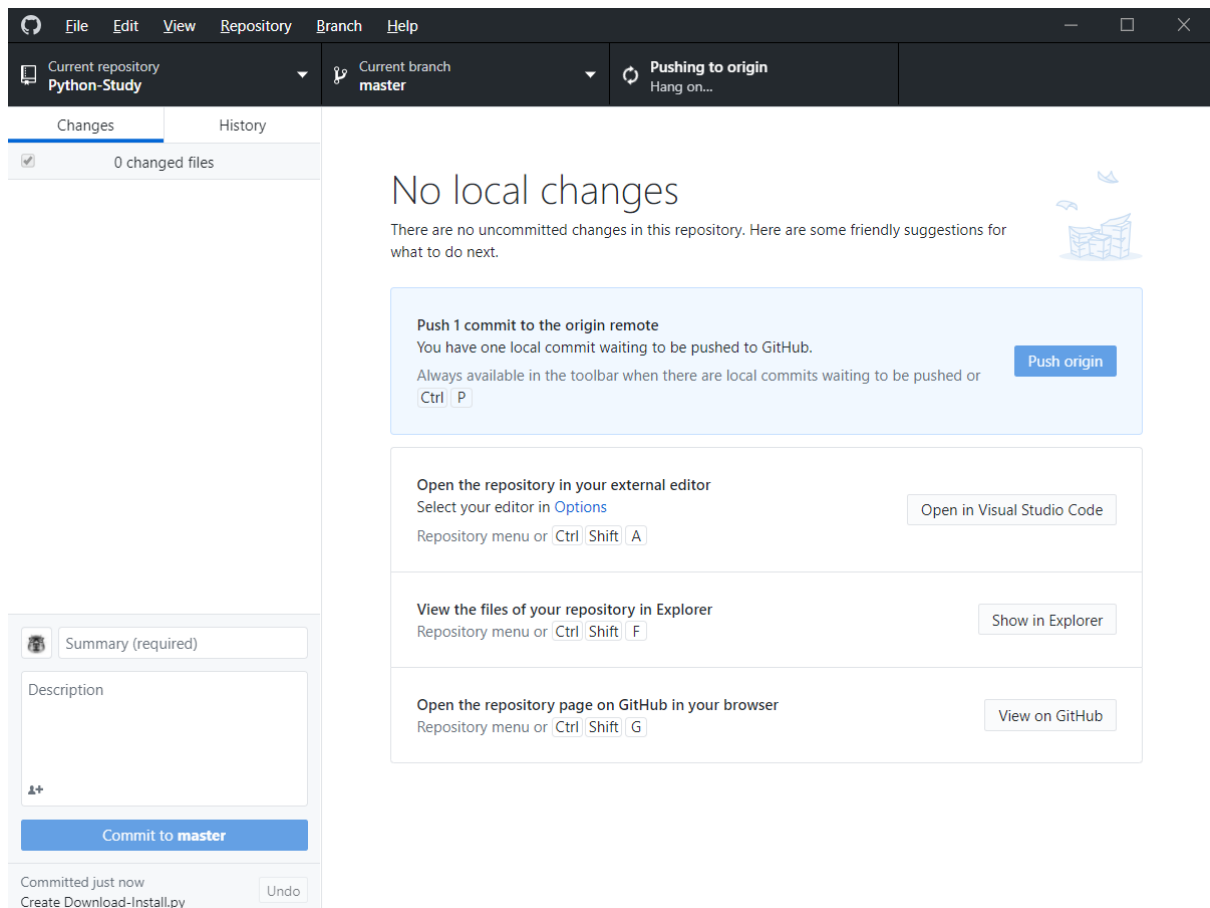
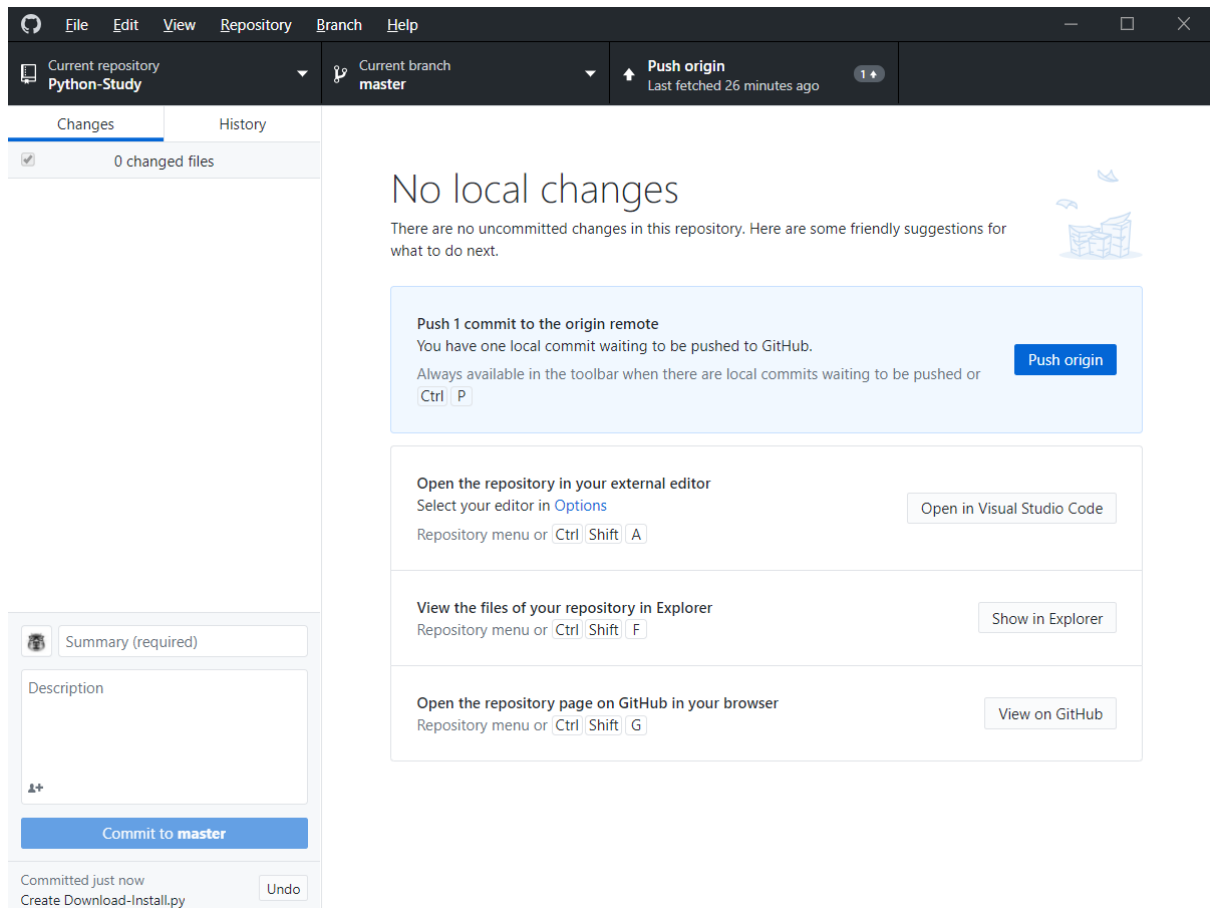
Step4 : 這邊我寫了Download & Install的python code



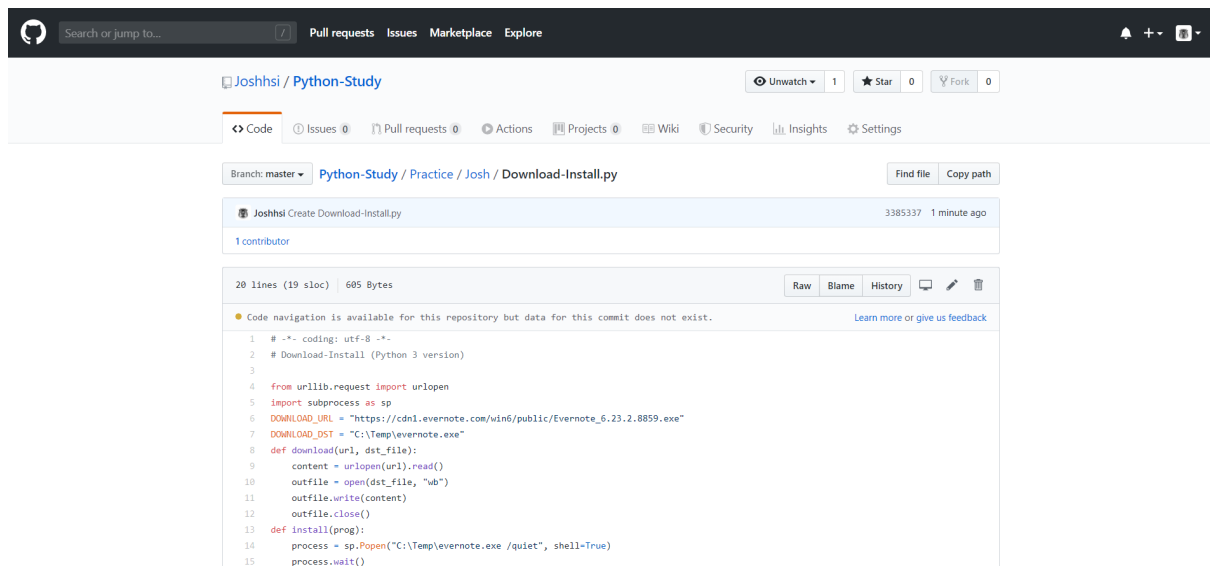
Step5: 這時候回到GitHub Desktop就會看到剛剛更新的檔案內容, 如同之前做過的, 按下右下角的commit確認異動



Step 6 : Commit完會看到左邊的Changes已經被清空, 再按右邊的"Push origin", 把更新Push到GitHub上



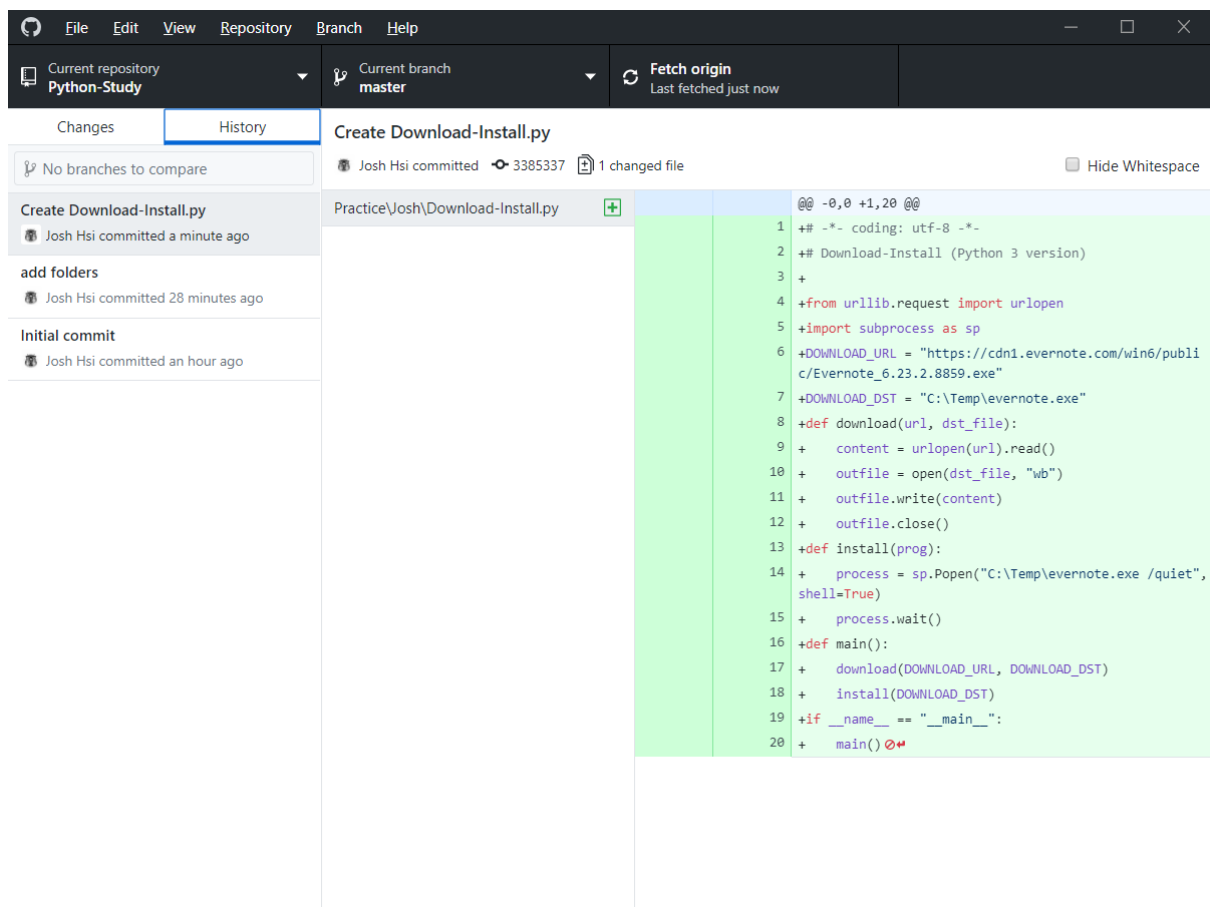
Step7: Push完之後就可以看到Python的程式已經放在GitHub上了~~



The screenshot shows the GitHub web interface for the repository 'Joshhsi / Python-Study'. The repository has 1 star and 0 forks. The 'Code' tab is selected, showing the file 'Download-Install.py' on the 'master' branch. The commit history shows a single commit by Joshhsi. The code content is as follows:

```
1 # -*- coding: utf-8 -*-
2 # Download-Install (Python 3 version)
3
4 from urllib.request import urlopen
5 import subprocess as sp
6 DOWNLOAD_URL = "https://cdn1.evernote.com/win6/public/Evernote_6.23.2.8859.exe"
7 DOWNLOAD_DST = "C:\\Temp\\evernote.exe"
8 def download(url, dst_file):
9     content = urlopen(url).read()
10    outfile = open(dst_file, "wb")
11    outfile.write(content)
12    outfile.close()
13 def install(prog):
14     process = sp.Popen("C:\\Temp\\evernote.exe /quiet", shell=True)
15     process.wait()
```

P.S Push完之後也可以在GitHub Desktop上看到之前commit & push的紀錄



The screenshot shows the GitHub Desktop application interface. The 'Current repository' is 'Python-Study' and the 'Current branch' is 'master'. The 'Fetch origin' button is visible. The 'History' tab is selected, showing a list of commits. The commit history shows two commits: 'Create Download-Install.py' and 'Initial commit', both by Josh Hsi. The 'Create Download-Install.py' commit is selected, showing the file 'Practice\\Josh\\Download-Install.py' and the commit message 'Josh Hsi committed 3385337 1 changed file'. The code content is as follows:

```
@@ -0,0 +1,20 @@
1 +# -*- coding: utf-8 -*-
2 +# Download-Install (Python 3 version)
3 +
4 +from urllib.request import urlopen
5 +import subprocess as sp
6 +DOWNLOAD_URL = "https://cdn1.evernote.com/win6/public/Evernote_6.23.2.8859.exe"
7 +DOWNLOAD_DST = "C:\\Temp\\evernote.exe"
8 +def download(url, dst_file):
9 +    content = urlopen(url).read()
10 +    outfile = open(dst_file, "wb")
11 +    outfile.write(content)
12 +    outfile.close()
13 +def install(prog):
14 +    process = sp.Popen("C:\\Temp\\evernote.exe /quiet", shell=True)
15 +    process.wait()
16 +def main():
17 +    download(DOWNLOAD_URL, DOWNLOAD_DST)
18 +    install(DOWNLOAD_DST)
19 +if __name__ == "__main__":
20 +    main()
```

