(Đặng Thanh Vũ)

Vunghixuan.info@gmail.com

PYTHON CƠ BẢN

**Kèm “Tự Học Python Excel”**

**GIỚI THIỆU**

Trong quá trình đọc tài liệu Tự Học Python Excel, một số kiến thức về code pyhton cơ bản mà các bạn chưa được trang bị sẽ gây lúng túng, hiểu nhầm, hoặc dễ sinh ra lỗi khi soạn thảo code. Vì thế, tôi xin đính kèm phần tài liệu này để bạn đọc củng cố phần kiến thức này.

Tham gia thảo luận cùng group các vấn đề khó khăn trong quá trình học tại: <https://www.facebook.com/groups/1015080755701155>

*Thông tin góp ý xin liên hệ* [*https://vunghixuan.github.io/*](https://vunghixuan.github.io/)

Chân thành cảm ơn!

**Giải thích một số từ ngữ:**

1. **Python:** là ngôn ngữ lập trình, chức năng dùng để biên dịch mã code. Như chúng ta biết, máy tính chỉ đọc dc các dãy số (0 0 0 1 0 1 1).

Python có chức năng biên dịch những gì chúng ta viết tại các bản trình soạn thảo code (cụ thể các bạn đang sử dụng là Visual Code) và chuyển chúng thành các đoạn mã (0 0 0 1 0 1 1) nói trên

1. **Trình soạn thảo:** là bản viết code của chúng ta, hiện có rất nhiều trình soạn thảo như Visual Studio Code, Sublime Text, Jupyter notebooks,…) Trong tải liệu này, hướng dẫn sử dụng theo trình soạn thảo Visual Studio Code (gọi tắt là VSC).

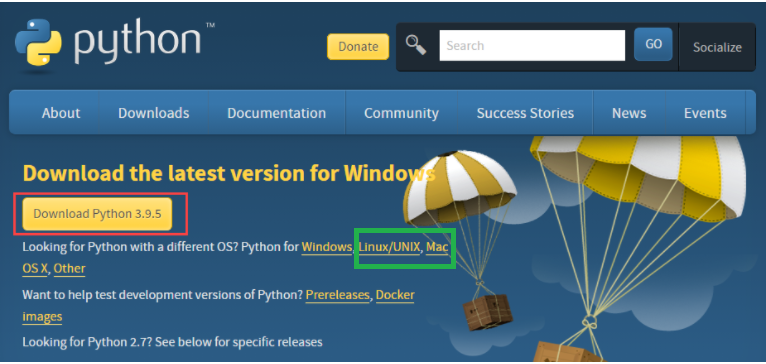
**CHƯƠNG 1:**

**HƯỚNG DẪN CÀI ĐẶT PYTHON, VISUAL STUDIO CODE, XLWINGS**

1. **HƯỚNG DẪN CÀI ĐẶT PYTHON**

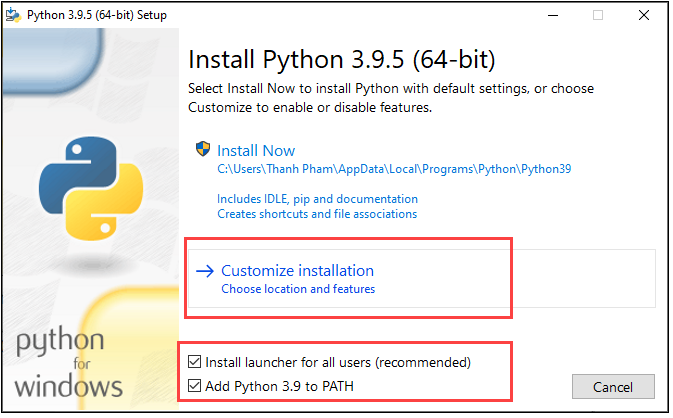
Truy cập vào trang: <https://www.python.org/downloads/> để cài đặt Python cho hệ điều hành bạn đang sử dụng.

Đây là phiên bản mới nhất của Python tại thời điểm mình cập nhật, các bạn cài sau có thể đã cập nhật phiên bản mới hơn chẳng hạn là “python-3.9.7”

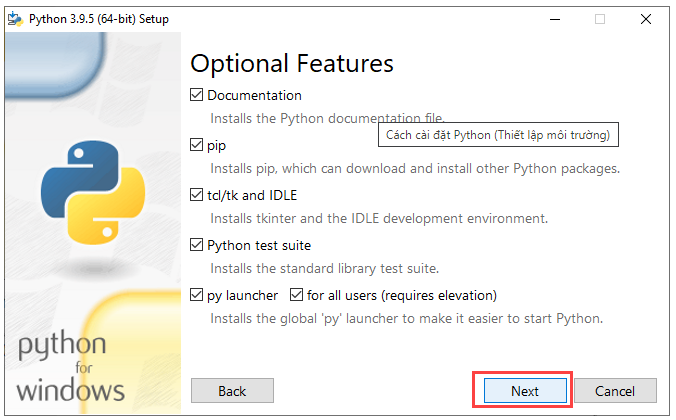


Click đúp chuột vào file vừa tải xuống (python-3.9.5-amd64.exe) để tiến hành cài đặt Python: Phần màu đỏ cho windows, các hệ điều hành khác là màu xanh.

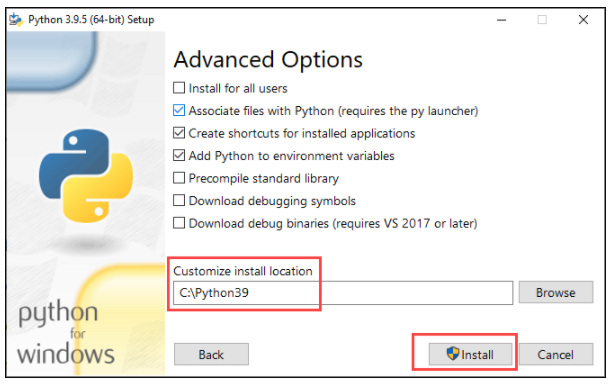
Tiếp theo Chọn **Customize installation**:



Chọn tất cả các tính năng tùy chọn, click Next:

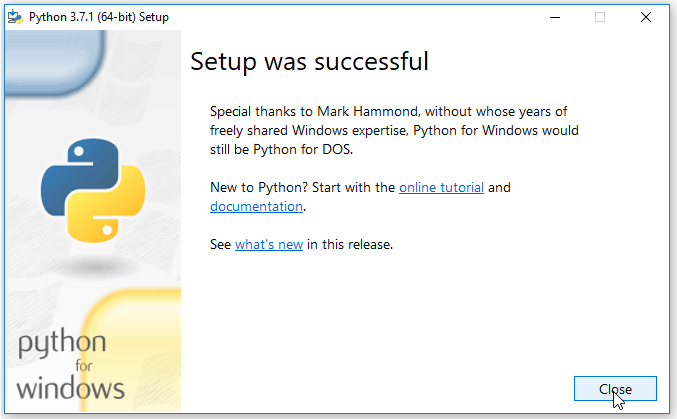


Chọn các tính năng tùy chọn như trong hình, click Install:

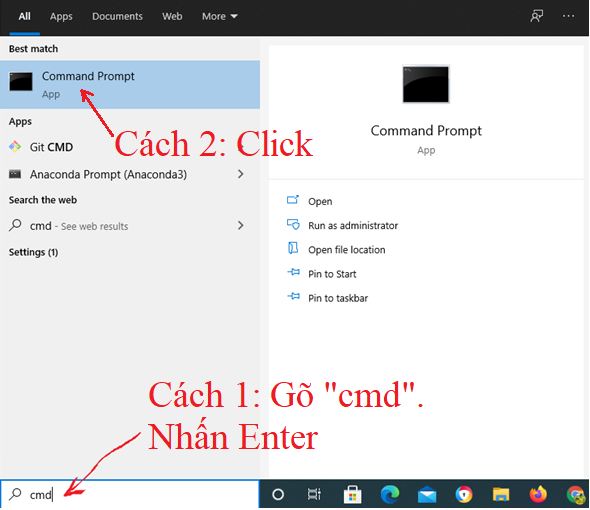


Nếu bạn muốn để mặc định như mình thì không phải chọn button Browse, nhấn luôn 🡪**Install**

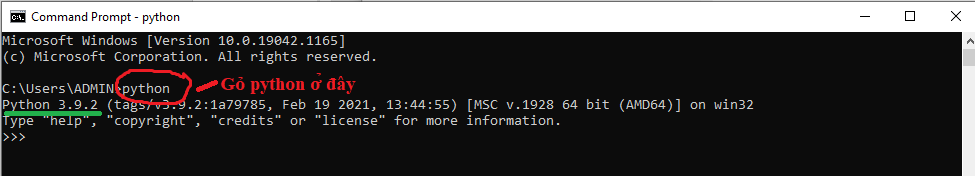
Kết quả sau cài đặt:

****

Để kiểm tra việc cài thành công hay không bạn màn hình windows, góc trái dưới cùng, cạnh biểu tượng Windows, là form text của windows tại dòng chữ “Type here to search” gõ “cmd”, tức là khởi động **Command Prompt**, bạn nhấn Enter hoặc click để chọn



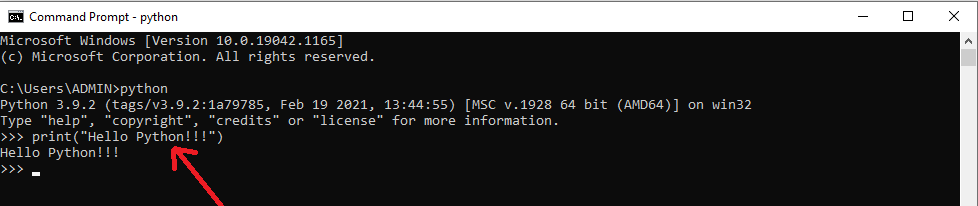
Sau khi **Command** **Prompt** xuất hiện, bạn gõ “python”. Kết quả như sau:



Chỗ màu xanh là version bạn đã cài. Vì mình cài trước rồi nên phiên bản là python 3.9.2

Bạn có thể học code python bằng các câu lệnh đơn giản trực tiếp ở đây như sau:

Chẳng hạn như gõ những câu để bái tổ cho ngôn ngữ Python: *print(“Hello Python!!!”)*



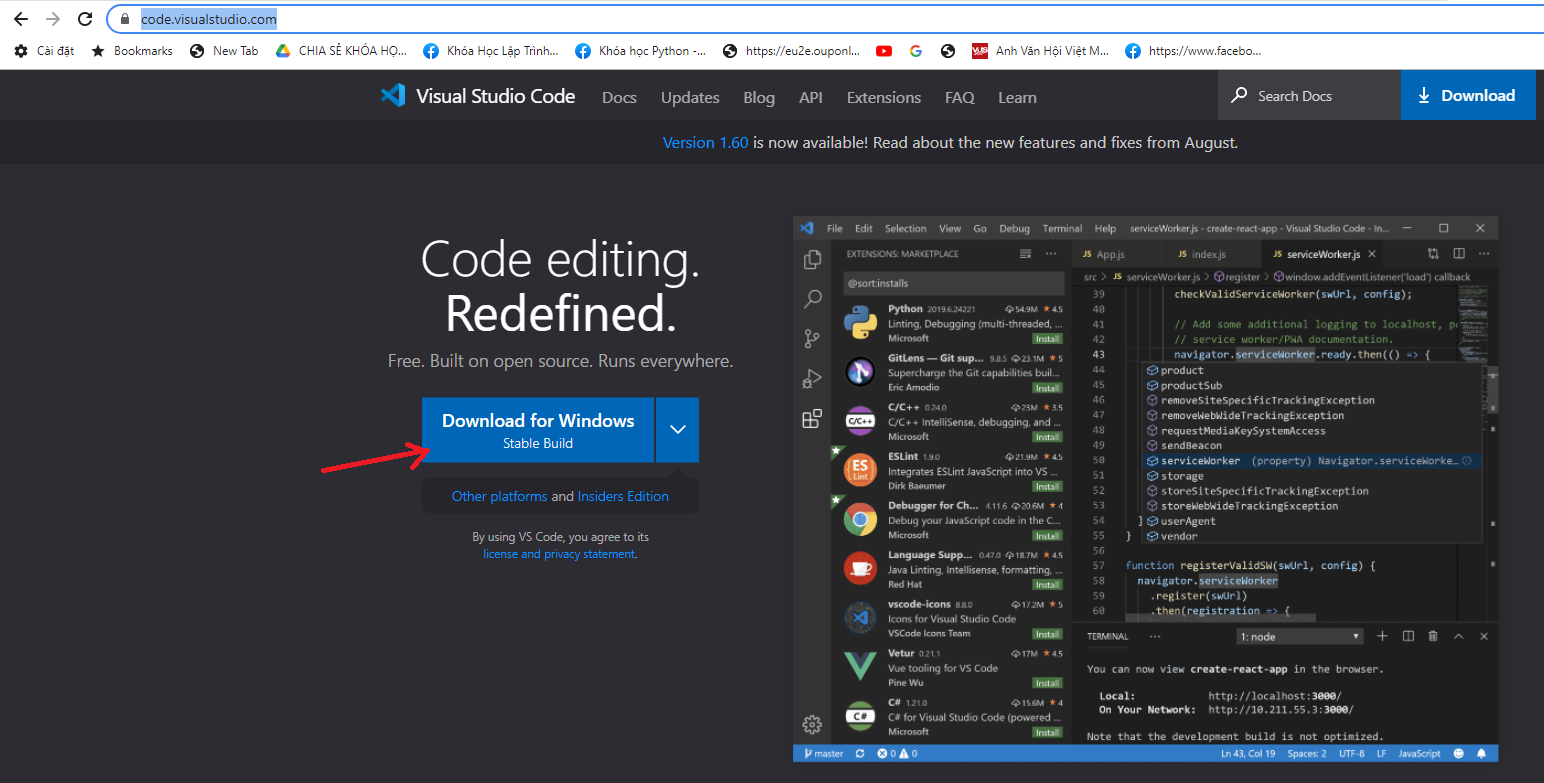
1. **HƯỚNG DẪN CÀI ĐẶT VISUAL STUDIO CODE**

Visual Studio Code là bản soạn thảo (giống như word), dùng để trình bày soạn thảo và chỉnh sửa code, ưu điểm nhẹ, tiện lợi và phổ biến.

Bạn có thể tải xuống và cài đặt miễn phí trên Windows, nhưng nó cũng có sẵn trên Linux và macOS.

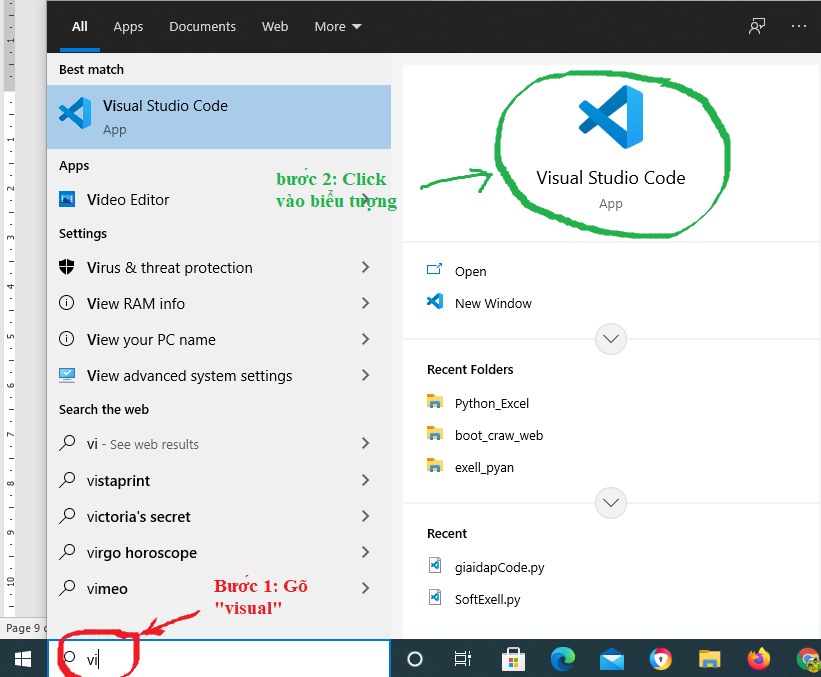
Truy cập vào trang: <https://code.visualstudio.com/>

Website tự Recommend phiên bản thích hợp với máy của bạn, hoặc nếu muốn cài bản khác có thể kéo xuống nhé



Sau khi tải được file VSCodeUserSetup-x64-1.60.0.exe *(cái này tùy thuộc hệ điều hành máy bạn)*, bạn đúp click vào file này để cài đặt. Các thao tác còn lại giống như cài đặt Python ở trên.

Sau khi cài đặt thành công, trên màn hình windows, góc trái dưới cùng, cạnh biểu tượng Windows, là form text của windows tại dòng chữ **“Type here to search”** gõ 1 phần trong đoạn chữ “Visual Studio Code”, windows sẽ gợi ý chương trình bạn cần tìm theo từ khóa bạn gõ vào, màn hình xuất hiện như sau:



Bạn click vào biểu tượng Visual Studio Code (màu xanh) để khởi động, sau Visual Studio Code hiện lên

**Cài đặt Extension:**

Extension được Microsoft phát triển, có hỗ trợ gợi nhớ code (suggestion), tự động hoàn thành code, chạy ứng dụng Python và debug.

Tiếp theo để thuận tiện trong việc code, xử lý lỗi, gợi ý tính năng mà Visual Studio Code (mình gọi tắt là VSC)

Ví dụ mình chọn gói IntelliSense (Pylance), Linting, Debugging (multi-threaded, remot…) và thực hiện 4 bước theo hướng dẫn (đính kèm ảnh bên dưới).

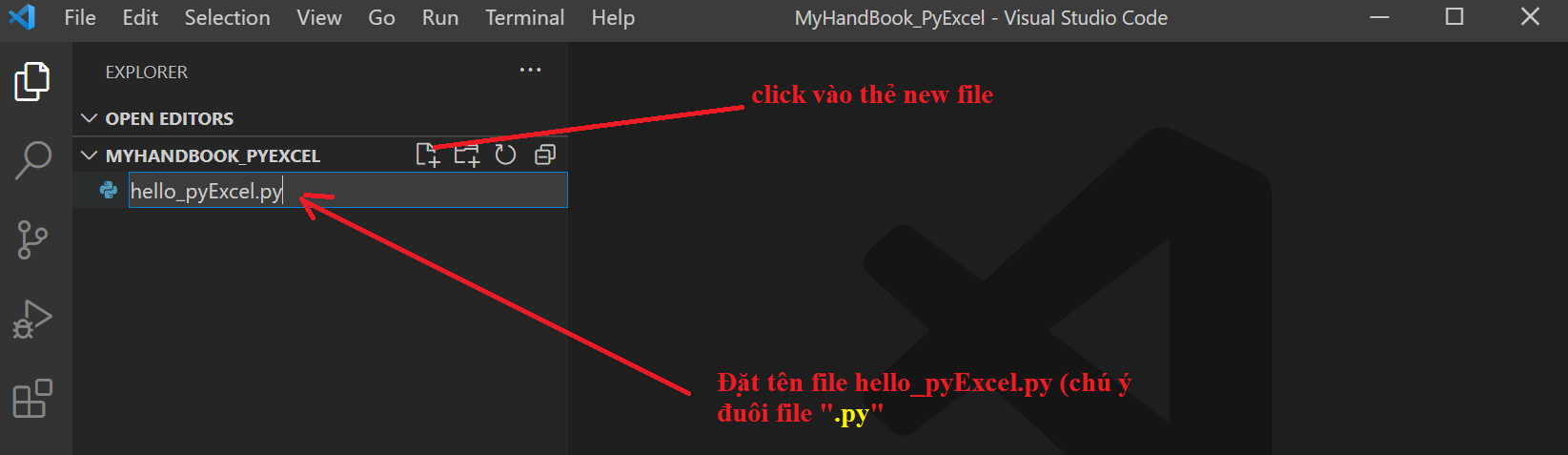


Ngoài ra, VSC còn tích hợp nhiều ứng dụng giúp cho quá trình soạn thảo đẹp hơn, nhanh chóng và tiện ích (có thể google để tìm ra những thứ cần thiết cho chính bạn)

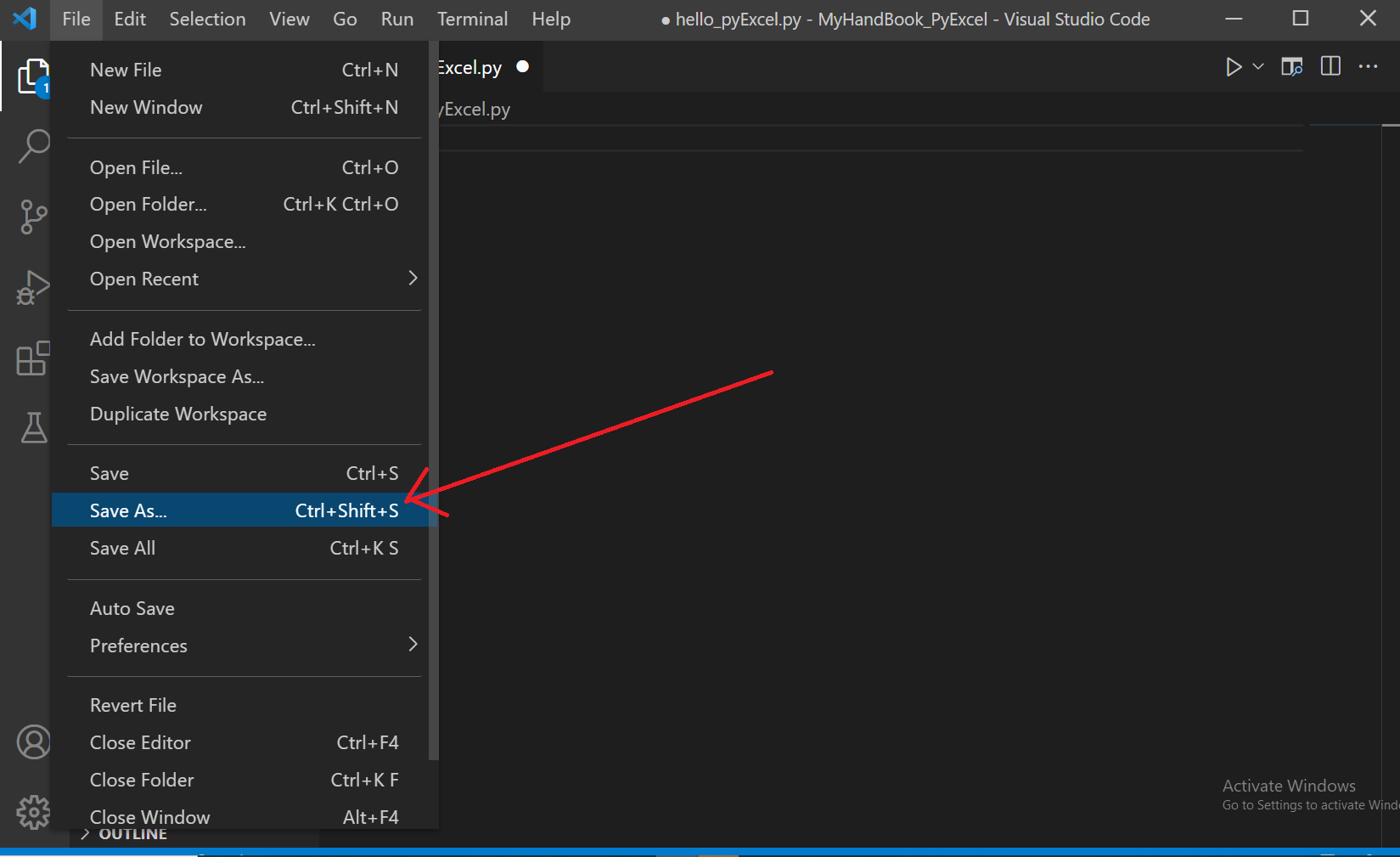
**Hướng dẫn soạn thảo Python trên Visual Studio cơ bản**

Phần này là bước đầu tiên thực hiện nên các bạn phải thật để ý, dễ sinh ra lỗi **(Chú ý** sau khi đặt tên file phải thêm đuôi **“.py”**, còn không nó sẽ hiểu là file txt.

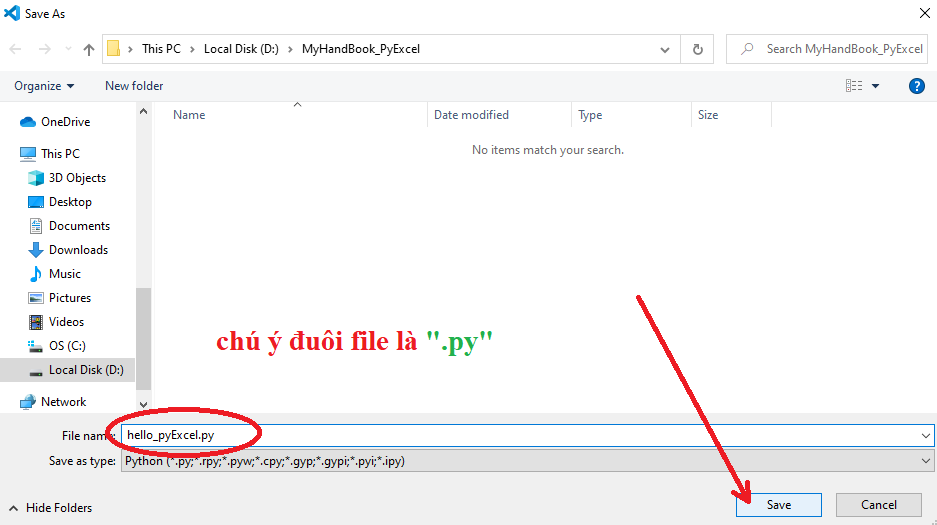
* **Tạo Workspace**
* Bước 1: Tạo file **hello\_pyExcel.py**, chú ý đuôi file là **“.py”**



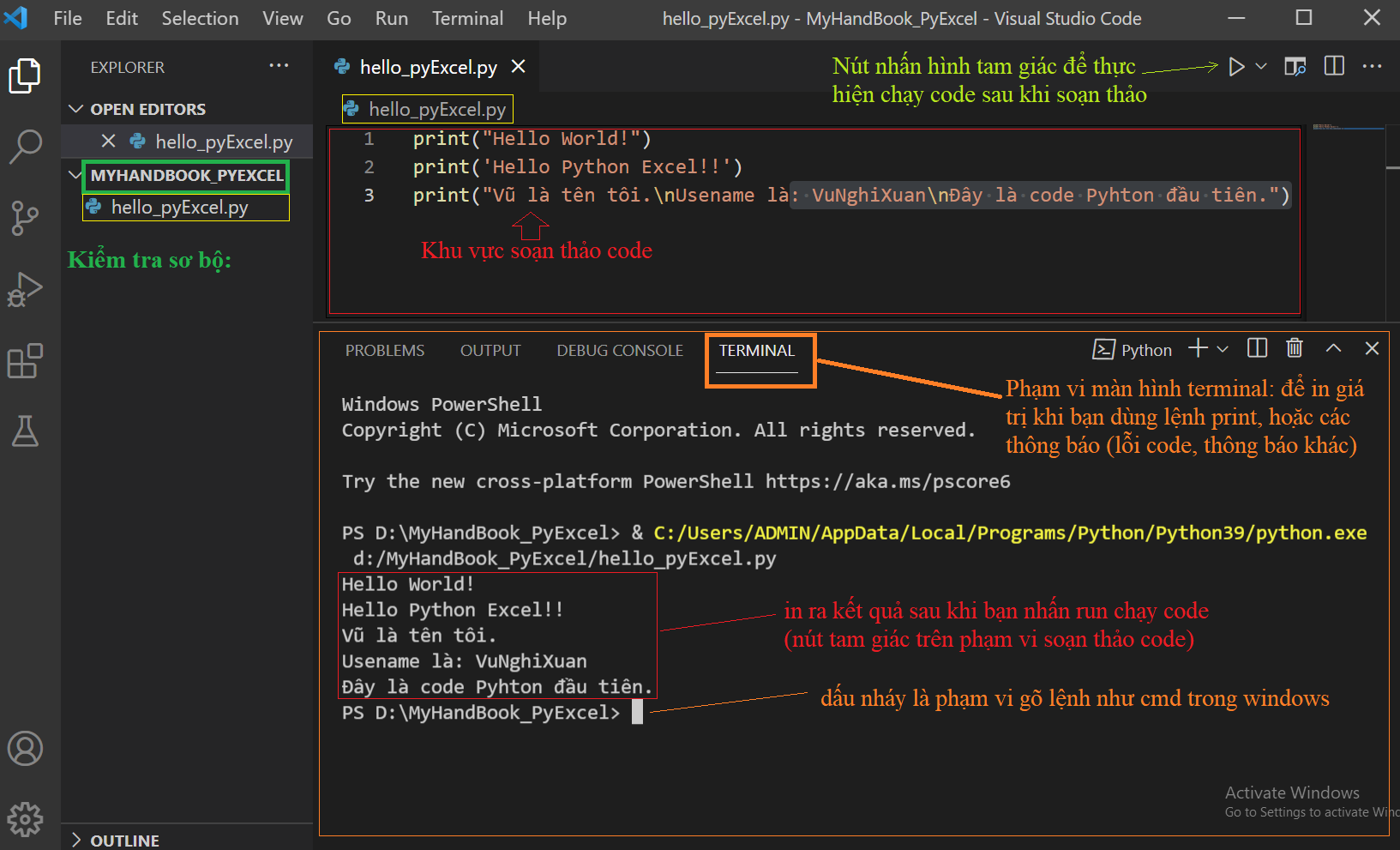
Thực hiện lưu file



Hộp thoại Save As được mở ra, bạn chú ý đuôi file phải “**.py” .** Nhấn chọn nút **Save** (hình dưới)



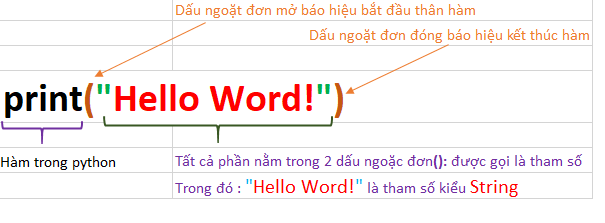
Giới thiệu phạm vi các khu vực màn hình soạn thảo VSC



**LÀM QUEN VỚI PYHTON TRÊN TRÌNH SOẠN THẢO (VSC)**

Quay lại trình soạn thảo chứa file **hello\_pyExcel.py.** Bạn thử gõ theo đoạn code print(“Hello world!”), print(‘Hello Python Excel!!’) trong phạm vi soạn thảo code ở hình trên (phần giới thiệu màn hình VSC) xem có bị lỗi gì hay không?

**. Giải thích code:**

****

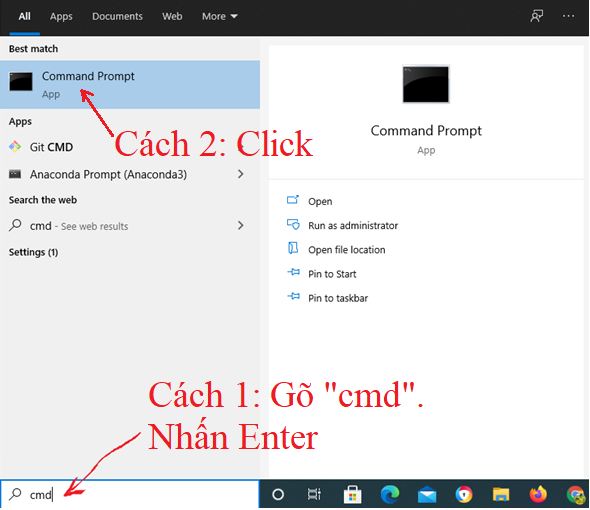
+ Hàm **print** trong python: dùng để in ra màn hình Terminal.

+ Dấu “**(** “dùng để báo hiệu bắt đầu của hàm. Dấu “ **)**“ là dấu hiệu kết thúc hàm

+ "Hello Word!": là 1 tham số của hàm, kiểu String. Đối với kiểu String có thể dùng dấu ‘Hello Word!’ thay cho “Hello Word!” *(hãy xem dòng code thứ 2 của trình soạn thảo phía trên).*

1. **HƯỚNG DẪN CÀI ĐẶT THƯ VIỆN XLWINGS**

Khởi động **Command Prompt** bằng cáchgõ “cmd” nhấn **Enter** tại hộp text **“Type here to search”**, nằm sát góc trái màn hình Windows **(nếu bạn thì nhìn vào hình dưới)**



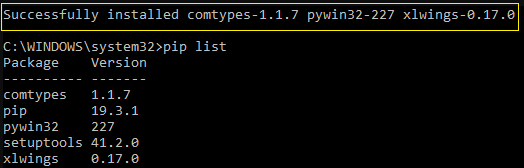
Khi màn hình **Command Prompt** xuất hiện tại vị trí chuột nhấp nháy là phần gõ lệnh, bạn gõ: **pip install xlwings**



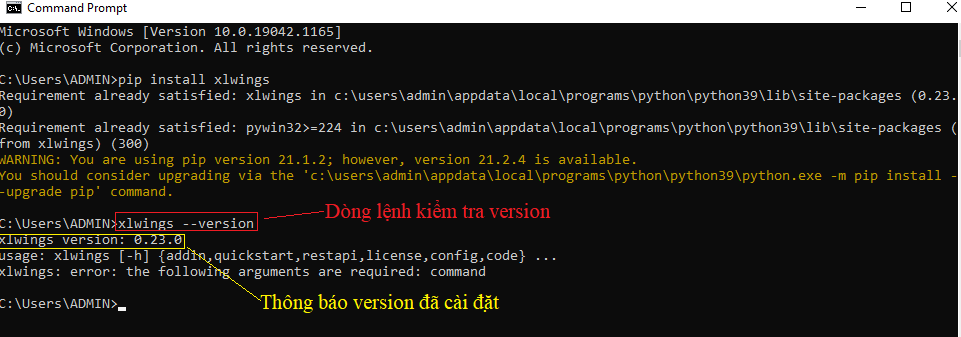
Để cập nhật thay vì cài đặt bạn dùng lệnh: **pip install --upgrade xlwings**

Cách kiểm tra thư viện đã cài đặt thành công hay chưa?

**. Cách 1:** Thay vì dòng cảnh báo màu vàng hình trên máy bạn sẽ có dòng **“Successflly installed …”, (**như hình dưới)



**. Cách 2:** gõ tiếp lệnh trong cmd dòng lệnh: **xlwings –version** để kiểm tra phiên bản đã cài đặt

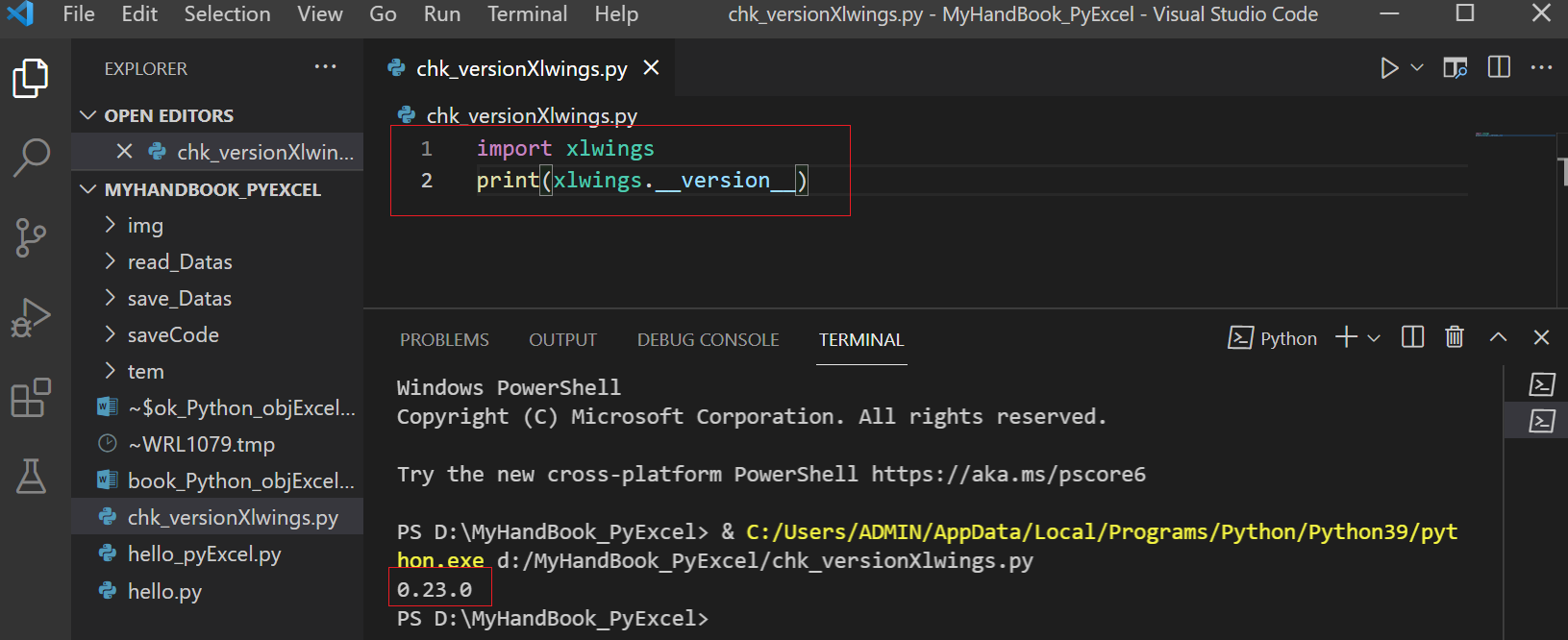


**. Cách 3:** Trong trình soạn thảo VSC bạn thực hiện 2 dòng lệnh

import xlwings

print(xlwings.\_\_version\_\_)

Kết quả xem hình bên dưới



Các bước hướng dẫn trên là mình thực hiện trên máy Windows 10, 64-bit. Các hệ điều hành khác cũng tương tự, nếu quá trình cài đặt bị lỗi bạn chia sẻ trên group (tại : <https://www.facebook.com/groups/1015080755701155> ) để mọi người có thể giúp bạn (hoặc có hướng dẫn truy cập web để sửa lỗi), đồng thời đây cũng là cách để mình bổ sung thêm phần lỗi cho cuốn tài liệu này.

**CHƯƠNG 2:**

**I. PYTHON CƠ BẢN:**

**CHƯƠNG 3:**

1. **TÌM HIỂU VỀ THƯ VIỆN XLWINGS:**

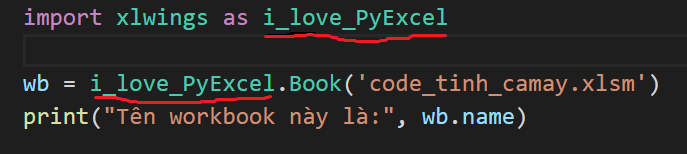
Thư viện xlwings là thư viện chứa các modul, hàm giúp chúng ta có thể giao tiếp và điều khiển được với file excel, có thể thêm hoặc lấy được dữ liệu, xử lý, phân tích và lưu lại trên file excel đó, hoặc 1 file, nhiều file excel khác.

**TÌM HIỀU VỀ THƯ VIỆN XLWINGS**

Tại màn hình soạn thảo Visual Studio Code (gọi tắt là VSC), chúng ta nhập đoạn code sau:

import xlwings as xw

Dòng lệnh này có nghĩa là nhập vào thư viện xlwings và gán cho nó cái tên là xw (còn gọi là biến), thực ra bạn có thể gán cho nó từ xls, hay tên của bạn của bạn cũng được, miễn là tuân theo một số quy tắc đặt tên biến, không có dấu gạch ngang, không dùng dấu “\*”, không có dấu cách,…Các bạn có thể google để tìm hiểu cách đặt tên biến.

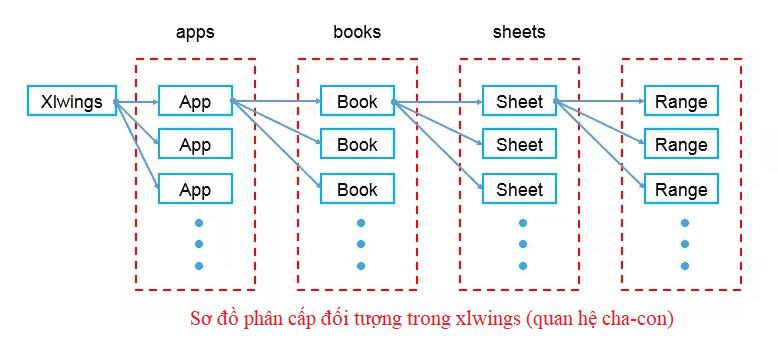


Mình đặt tên biến “i\_love\_PyExcel” thay cho cách dùng “xw”, chương trình vẫn hoạt động bình thường.

1. **Đối tượng workbook:**

Python là ngôn ngữ lập trình theo hướng đối tượng *(tiếng Anh: Object-oriented programming, viết tắt: OOP)*. Đối tượng tức là những vật thể tồn tại (dùng để phân biệt đối tượng này với đối tượng khác), có trạng thái và hành vi…Tỉ dụ như: **Đối tượng** là 1 con mèo. Các hành vi: leo, trèo cây, kêu meo meo… (trong lập trình người ta gọi là phương thức). Còn thuộc tính con mèo là: có râu, móng vuốt, màu lông…

Cụ thể, bảng tính Excel là 1 đối tượng. Có các phương thức: mở (open), đóng (close), lưu, đọc, …Thuộc tính như: Tên bảng tính, chứa dữ liệu (data), …



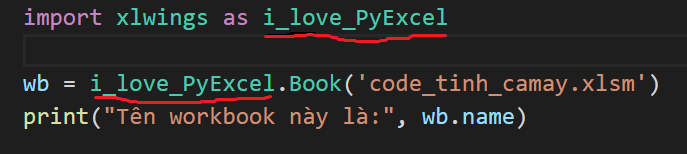
Túm lại, thuộc tính là chỉ hành động: đi, đứng, chạy,..thuộc tính là: màu sắc, tính cách, mô tả từng bộ phận của đối tượng đó.

Vì Python cũng như những ngôn ngữ khác hiện nay đều nhắm đến hướng đối tượng nên để dễ dàng gọi chúng, người ta gán cho nó 1 cái tên gọi là biến.

**a/ Tại sao phải gán tên biến?**

Tên biến là để đặt tên cho đối tượng đó (gắn liện với thuộc tính và tính cách). Giống như mỗi người phải có 1 cái tên *(chỉ cần gọi tên là người khác có thể hình dung bạn có đặc điểm cao hay thấp, có chạy được ko, có học giỏi hay ko?)*

Vì thế khi lập trình bạn cũng nên gán cho nó 1 cái tên là vậy *(dĩ nhiên tên biến cũng phải tuân theo quy tắc chung như đã nói trên)*.



Như vậy trong phạm vi bản code tên, có thể hiểu i\_love\_PyExcel: là tên biến cho thư viện xlwings, hay wb là tên biến cho file bảng tính “code\_tinh\_camay.xlsm”.

**b/ Không đặt tên và gán biến?**

Câu trả lời là được nhé, chỉ có điều khi sử dụng mình phải gọi nó thông qua thuộc tính active (nghĩa là bảng tính hiện hành, đang được kích hoạt). Vấn đề sẽ xảy ra khi bạn mở cùng 1 lúc với nhiều file bảng tính thì làm sao để phân biệt được cái nào là cái hiện hành, tốn nhiều thời gian cho việc này quá nên tốt nhất bạn cứ đặt cho nó 1 cái tên.

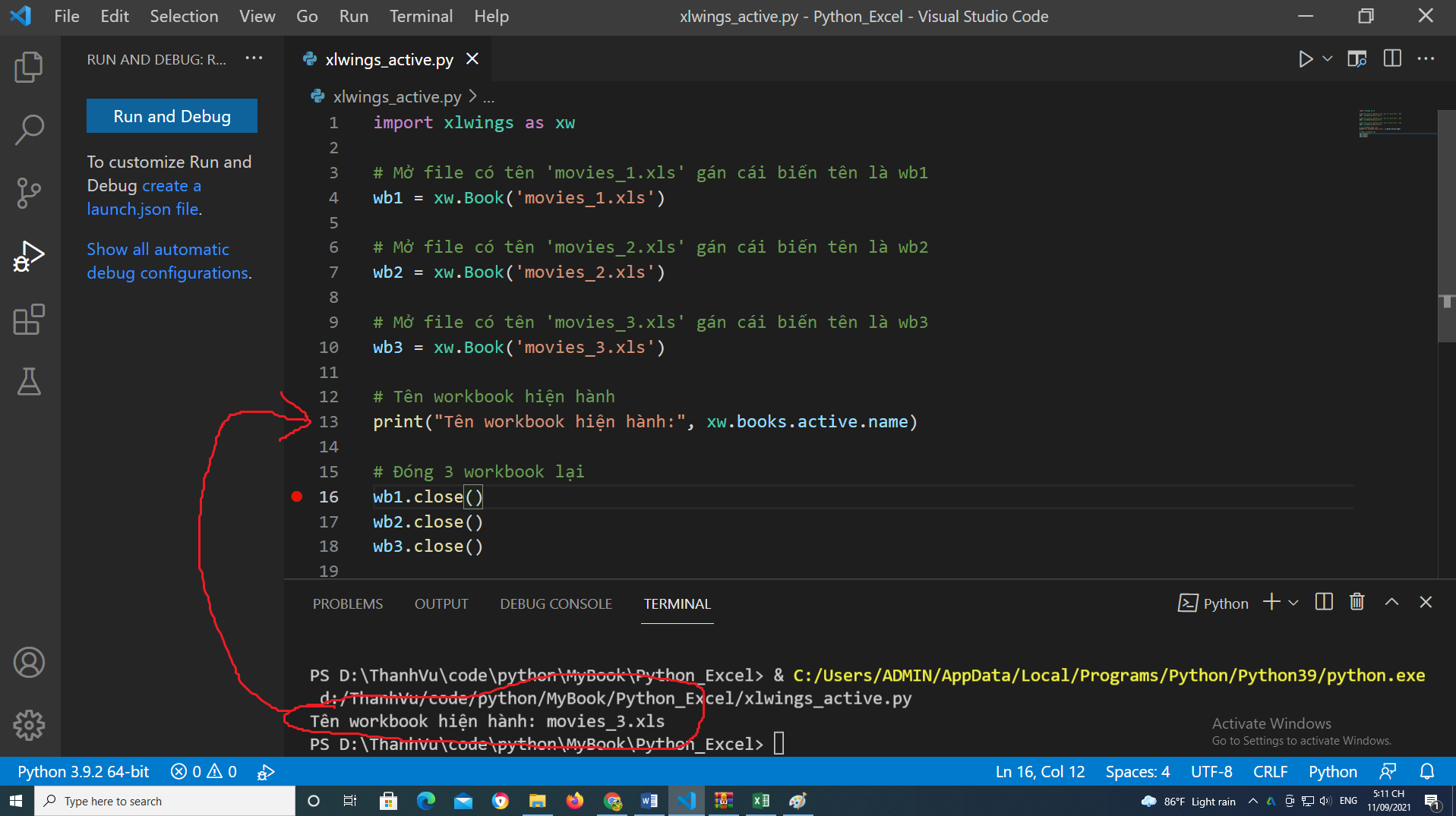
**\* Phần giải thích thêm về thuộc tính active**

Vì có bạn nhắn tin chưa hiểu về actice, nên mình giải thích kỹ hơn về phần này, mặc dù nó không quan trọng lắm (mình đã nói ở trên rổi).

Lấy 1 ví dụ: tổ 1 có các bạn A, B, C, D. Khi cô giáo gọi tên bạn D, được hiểu là bạn D đang được “active”, đang được kích hoạt (điều này giống như bạn đang kích hoạt 1 tài khoản vậy)

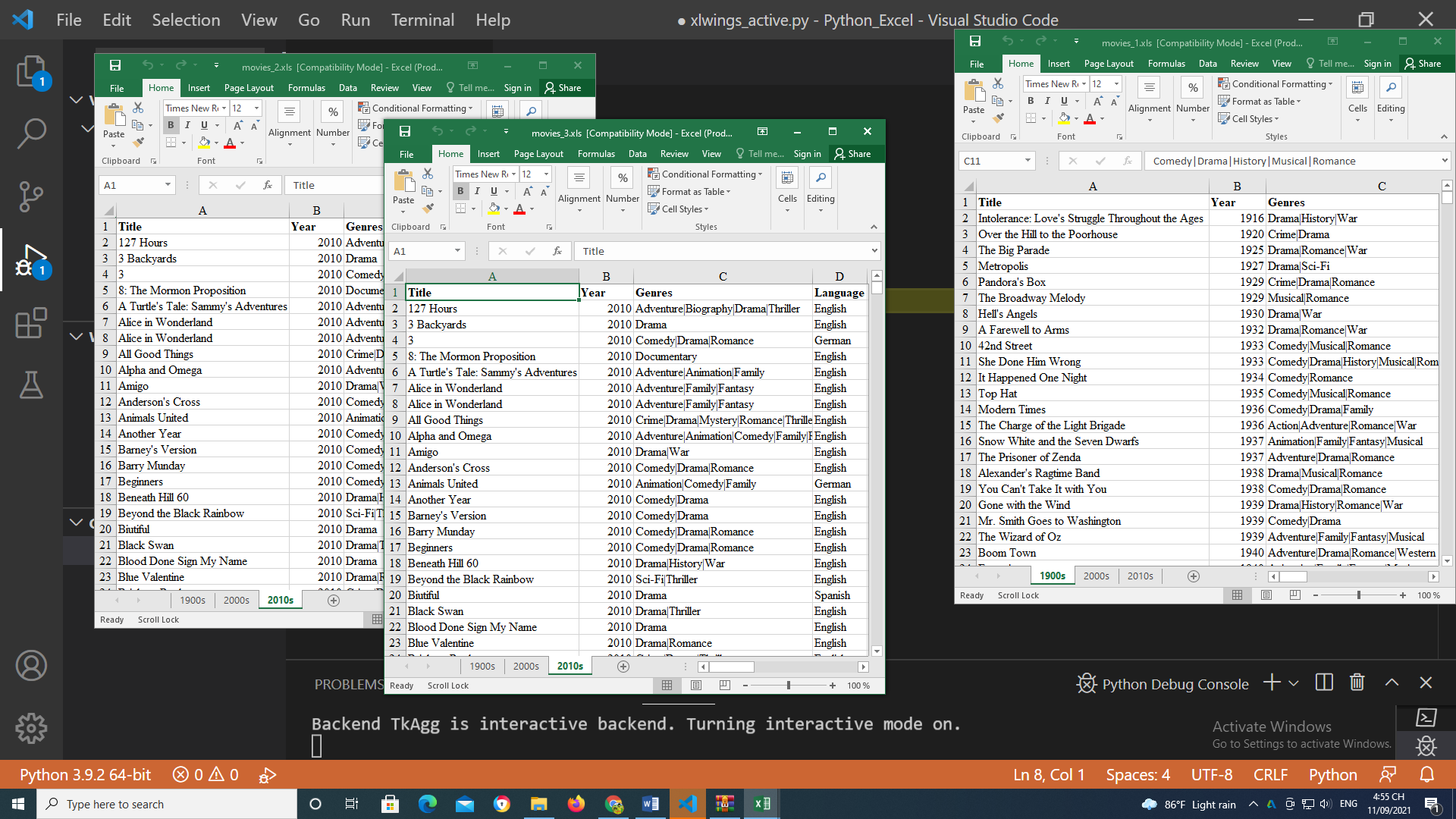
**Còn trong bảng tính excel thì sao?**

Mình mở 3 bảng tính gồm: movies\_1, movies\_2, movies\_3



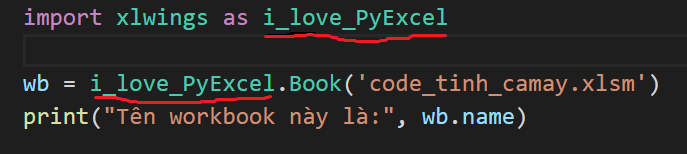
Tại terminal do đoạn code dòng thứ 13, thực hiện in ra tên của workbook đang được kích hoạt (hiện hành) là: “movies\_3”

Song song đó, màn hình window xuất hiện bảng tính “movies\_3” trên cùng so với 2 bảng tính còn lại. Điều này chứng minh là bảng tính nào mở ra sau cùng nhất là bảng tính active (hiện hành).



----------------------------------------------------------------------------------------------------

**Ôn Kiến Thức Về Thư Viện Xlwings Và Đối Tượng Tượng Workboks**



**Nhập thư viện xlwings**

**import xlwings as i\_love\_PyExcel #** Nhập vào thư viện xlwings và đặt tên là **i\_love\_PyExcel**

-

**wb = i\_love\_PyExcel.Book(“code\_tinh\_camay.xlsm”) #** Gán và đặt tên biến cho toàn bộ file của bảng tính **“code\_tinh\_camay.xlsm”** bằng cái tênlà **“wb”.** Trong đó, **Book** đượclà phương thức **(hàm) thông qua thư viện xlwings** *(mà ở trên ta đặt tên là i\_love\_PyExcel)*

Các dấu #; “”” “””; ” ” : nằm ở đầu dòng code dùng để ghi chú, python sẽ không thực hiện biên dịch các dòng này

Để gọi phương thức hay thuộc tính của đối tượng bạn dùng dấu **“.”** Và viết đúng tên cũng như thuộc tính hay phương thức để tránh lỗi.

Ngoài ra, bạn tham khảo và làm quen với cách gọi và làm việc 1 file excel thông qua 1 modul bằng phương thức xw.App(), như sau:

* Cách tạo ra file excel mới toanh:

import xlwings as xw

# Khởi tạo file excel bằng modul App

app = xw.App(visible=True, add\_book=False) # code này gán modul App với cái tên là app

# Tạo ra file excel mới gán tên là wb\_new

wb\_new = app.books.add()

""" \*Lưu ý tại đoạn code thứ 4, có 2 tham số:

    + "visible=True" : Nghĩa là cho phép hiện bảng tính mới vừa tạo

    + "add\_book=False": Nghĩa là không tạo thêm 01 bảng tính mới nữa

    --> Nếu bỏ tham số "add\_book" hoặc cho nó = True, nó sẽ sinh ra 2 workbooks (Trong đó tại dòng thứ 7: 01 workbook đầu tiên sẽ sinh ra do dòng lệnh app.books. Khi bạn thêm add vào nữa sẽ sinh thêm 1 workbook nữa)

"""

# lưu file mới đến thư mục "D:\MyBook\_PyExcel\"

wb\_new.save(r'D:\MyBook\_PyExcel\moitoanh.xls')

# Đoạn code này đóng file excel mới toanh đó

wb\_new.close()

# Đoạn code này tắt đối tượng app

app.quit()

* Cách đọc 1 file excel có tên lả “movies\_1.xls”:

import xlwings as xw

# Khởi tạo file excel bằng modul App

app = xw.App(visible=True, add\_book=False) # code này gán modul App với cái tên là app

# Tạo ra file excel mới gán tên là wb\_new

wb = app.books.open('movies\_1.xls')  #('movies\_1.xls')

# lưu file mới đến thư mục "D:\ThanhVu\code\python\Data\_Science\"

wb.save(r'D:\MyBook\_PyExcel\objSheet.xls')

# Đoạn code này đóng file excel mới toanh đó

wb.close()

# Đoạn code này tắt đối tượng app

app.quit()

**Bài tập phần đối tượng workbook:**

Hãy tạo ra 10 workbook mới đặt tên file từ 1🡪10, sau đó lưu file tại folder chứa cri[t python (file trình soạn thảo), cuối cùng tắt tất cả các apps của chương trình.

**Gợi ý:** Tìm hiểu về vòng lặp for trong python để tạo file và tắt các apps bằng vòng lặp file.

Đoạn code sau sử dụng “range(start, stop, step)”, sẽ giúp bạn hiểu vòng lặp for làm việc như thế nào. Trong đó, step là bước nhảy ví dụ bạn cho bằng 2, thì từ 1 nó sẽ đếm tiếp là 3.

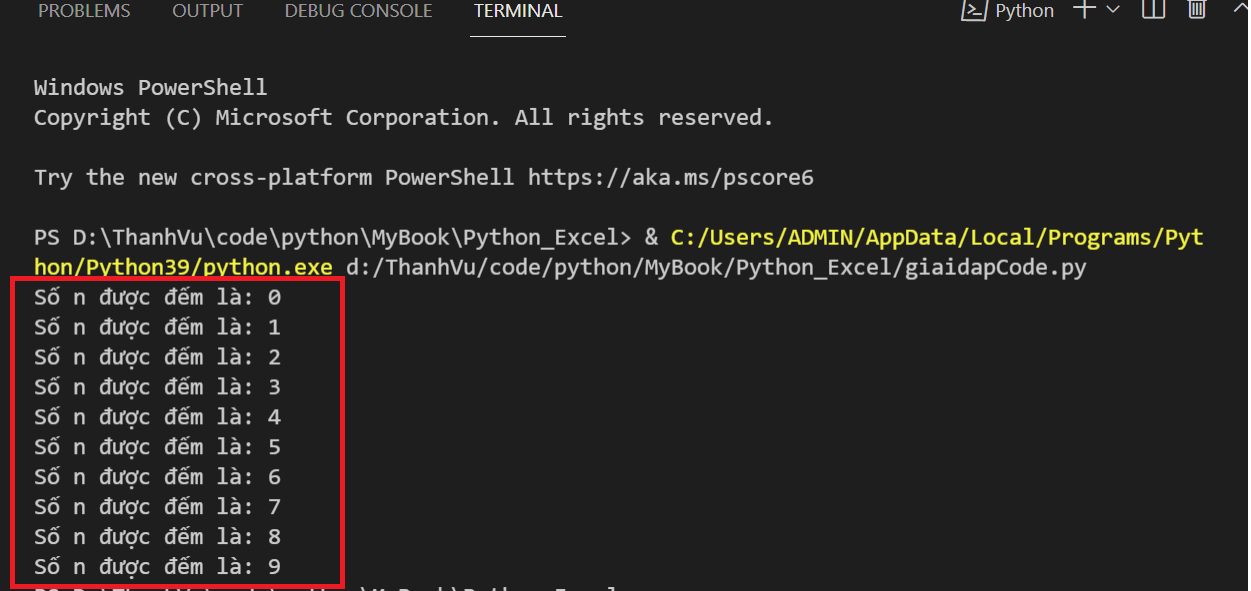
# Cho biến n là số đếm từ 0->10

for n in range(0, 10):

  print("Số n được đếm là:", n)

*Bạn cũng có thể thay code: range(0,10) bằng range(10), kết qua sẽ không thay đổi. Bạn google để tìm hiểu thêm về vòng lặp for.*

Kết quả in ra Terminal như sau:



Nhận xét: Số n được gọi là index (chỉ mục), chứ không phải số thứ tự như mình đếm. Nó bắt đầu từ con số 0,1,2,3,4,5,6,7,8,9 là kết thúc. Nghĩa là đã có tổng cộng 10 số rồi. Python không đếm phần tử index cuối cùng mà mình cho vào range, cụ thể là con số 10.

**Giải bài tập**

import xlwings as xw

import os

# Khởi tạo file excel bằng modul App

app = xw.App(visible= False, add\_book= False) # code này gán modul App với cái tên là app

# Dòng này sẽ Tắt các thông báo excel (như update,... xảy ra 1 số file excel)

app.display\_alerts=False

""" Giải thích code vòng lặp for bên dưới:

Cho biến book chạy trong phạm vi từ 0->2

Lần chạy đầu tiên book = 0, thêm vào 1 workbook và lưu nó vào tại folder chứa file code.

    + Lần đầu tiên do book =0, nên lần đầu tiên tên file sẽ = book+1, tức là 0+1=1, đổi sang bằng chữ là str(book+1) và thêm đuôi file sẽ là "1.xlsx"

+ Tương tự cho đến khi book=8, lúc này tạo ra file có tên "8.xlsx"

+ Lần thứ cuối cùng, book = 9 tương tự file sẽ là "9.xlsx"

    ->Kết thúc vòng lặp. range trong Python ko đếm giá trị cuối của vòng lặp

"""

for book in range(10):

    book = app.books.add().save(str(book+1) + ".xlsx")

# Tắt tất cả các apps Excel đang hiển thị, kể cả file mở = tay (ko dùng code)

for app in xw.apps:

    app.quit()

1. **Đối tượng Sheet:**

Sheet là 1 trang bảng tính excel, nó là đối tượng con của 1 file Excel. Có nhiều Sheets trong 1 file excel. Cách giao tiếp và điều khiển được nó thì đầu tiên chúng ta phải gọi nó ra và đặt cho nó 1 cái tên, khi nào cần thì gọi nó thì được hiểu nó đang được kích hoạt (active), đó cũng là lý do mà tại sao ở trên mình không muốn nói sâu về thuộc tính active là vậy.

Hãy tham khảo dữ liệu từ bảng excel có tên là **“Covid\_VN.xlsx”** tình hình covid Việt Nam, cũng như các bản code trong bài viết, bạn có thể tải dữ liệu về theo đường link này:

<https://github.com/VuNghiXuan/Mybooks/tree/main/Python_Excel>

Nếu các bạn không tải được file thì liên lạc với mình tại group này nhé:

<https://www.facebook.com/groups/1015080755701155>

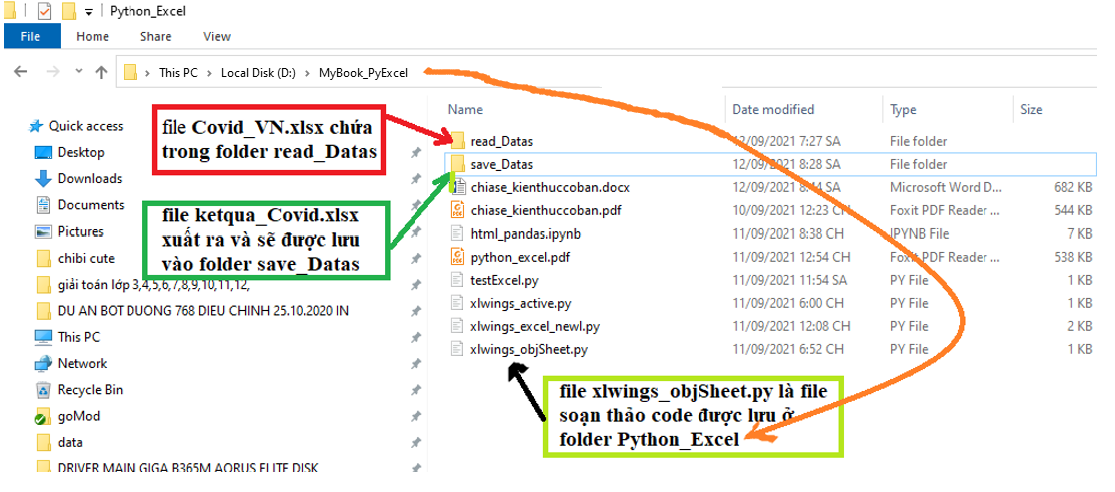
Để thuận tiện mình đã tải file này về trên máy mình và lưu nó tại ổ đĩa D, đường dẫn đây đủ là : **D:\** **MyBook\_PyExcel\** **read\_Datas\Covid\_VN.xlsx**

Bạn có thể tham khảo cách lưu dữ liệu và quy trình xử lý các file trong máy của mình như sau:

Tại Folder **read\_Datas:** mình sẽ lưu các file chứa dữ liệu (dữ liệu thô) chưa được đọc, chưa được phân tích xử lý. Cụ thể là file **Covid\_VN.xlsx** được chứa trong folder này.

File trình soạn thảo code chứa trong folder **Python\_Excel** *(kèm theo hình bên dưới)*:

Folder **save\_Datas** sẽ chứa các file sau khi xử lý code (tại trình soạn thảo VSC). Cụ thể làfile **ketqua\_Covid.xlsx** *(file này xuất ra khi xử lý code sẽ nói sau phần dưới)*

**

Quay lại trình soạn thảo VSC, chúng ta nhập các đoạn code sau

import xlwings as xw

# Khởi tạo file excel bằng modul App

app = xw.App(visible=True, add\_book=False) # code này gán modul App với cái tên là app

# Đọc file excel có tên là "Covid\_VN.xlsx" từ folder "read\_Data"

wb = app.books.open(r'D:\MyBook\_PyExcel\read\_Datas\Covid\_VN.xlsx')  #('movies\_1.xls')

" -----> Đoạn này là bắt đầu tìm hiểu đối tượng Sheet"

# Đặt tên cho 1 danh sách các tên Sheets là "sh\_Names":

sh\_Names = wb.sheets

print ("Kết quả đối tượng sh\_Names là:", sh\_Names)

" Kết thúc tìm hiểu đối tượng Sheet <---------"

# lưu file "ketqua\_Covid.xlsx" tại folder "save\_Data"

wb.save(r'D:\MyBook\_PyExcel\save\_Datas\ketqua\_Covid.xlsx')

# Đoạn code này đóng file excel

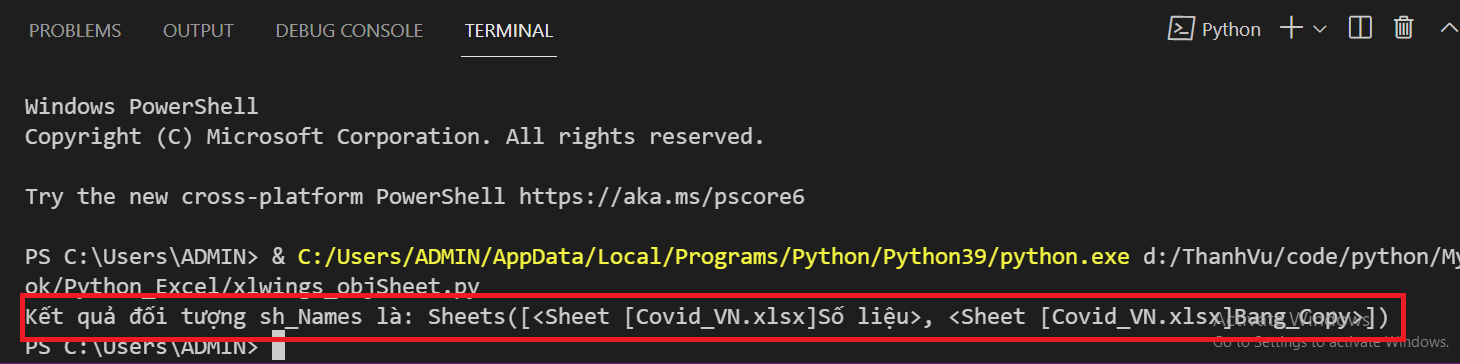
wb.close()

# Đoạn code này tắt đối tượng app

app.quit()

Như chúng ta đã biết, Excel là đối tượng cha (xem nó như 1 cái thùng chứa), bên trong nó là các đối tượng con là các Sheet, do đó thông qua đối tượng cha thì mới đi được đến đối tượng con, đó là tại sao phải có dòng lệnh **wb.sheets** (ở đoạn code sh\_Names = wb.sheets). Chú ý những chữ “s” đằng sau chữ “”sheets” nhé, nó tượng trưng cho số nhiều trong tiếng anh, nghĩa là gồm nhiều sheets trong 1 bảng tính.

Bây giờ, bạn chú ý kết quả in ra tại dòng Terminal như sau:



Kết quả in ra là cái gì đó đọc chả hiểu gì cả, nhưng cũng nhìn thấy được tên file là: “Covid\_VN.xlsx”, còn “Số liệu” và “Bang\_Copy” là tên của các Sheets đó.

**\*Bước tiếp theo là làm sao để lấy ra từng cái tên Sheet?**

Kỹ thuật này dùng vòng lặp (**for i\_name in sh\_Names**) liên quan đến kiến thức của bài tập trước. Nghĩa là, thằng **for** (gọi là từ khóa) đóng vai trò như 1 ông tổ trưởng khu phố, chạy qua từng nhà (là từng sheet), ghi lại và viết ra từng cái tên chủ hộ, còn **i\_name** là cái tên chủ hộ (tên cái sheet đó), **sh\_Names** được xem như 1 tổ dân phố vậy.Ok! chúng ta bắt đầu thêm **vòng lặp for** cho codenhư sau:

importimport xlwings as xw

# Khởi tạo file excel bằng modul App

app = xw.App(visible=True, add\_book=False) # code này gán modul App với cái tên là app

# Đọc file excel có tên là "Covid\_VN.xlsx" từ folder "read\_Data"

wb = app.books.open(r'D:\MyBook\_PyExcel\read\_Datas\Covid\_VN.xlsx')  #('movies\_1.xls')

" -----> Đoạn này là bắt đầu tìm hiểu đối tượng Sheet"

# Đặt tên cho 1 danh sách các tên Sheets là "sh\_Names":

sh\_Names = wb.sheets

# In ra terminal đối tượng sh\_Names

print ("Đối tượng sh\_Names là:", sh\_Names)

# Dùng vòng lặp for đi qua từng sheet và in ra tên sheet như sau:

for i\_name in sh\_Names: # Nhớ có các từ khóa "for"; "in" và dấu ":"

    print("Tên các sheets là:", i\_name.name) # Nhớ dùng phím tab phía đầu hàm print để thụt vào

" Kết thúc tìm hiểu đối tượng Sheet <---------"

# lưu file "ketqua\_Covid.xlsx" tại folder "save\_Data"

wb.save(r'D:\MyBook\_PyExcel\save\_Datas\ketqua\_Covid.xlsx')

# Đoạn code này đóng file excel

wb.close()

# Đoạn code này tắt đối tượng app

app.quit()

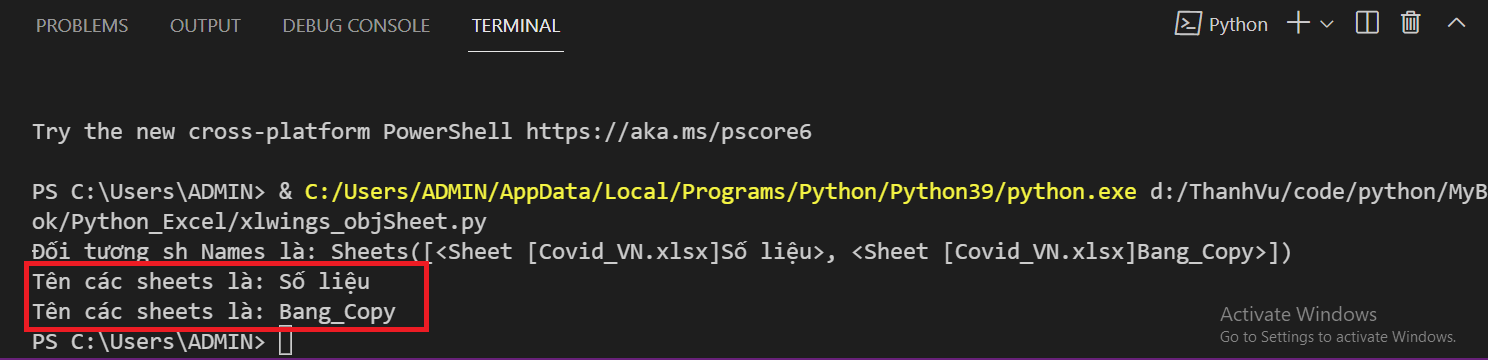
Một số chú ý dễ xảy ra lỗi khi dùng **vòng lặp for**:

Các từ khóa **“for”**, **“in”**, và dấu **“:”** là những từ khóa bắt buộc, không viết hoa. Đồng thời khi xuống dòng báo hiệu là thân hàm phải có **“Tab”** để thụt vào (Báo hiệu tất cả dòng lệnh trong thân hàm for.

+ Hàm “**name”** trong đoạn code “i\_name**.name”** là hàm lấy tên của đối tượng sheet, là hàm thuộc thư viện xlwings.

+ Hàm **“print”,** là hàm của python chứ ko phải là hàm thư viện xlwings

Xem kết quả xuất ra trên Terminal chúng ta đã có được tên 2 cái sheet là “Số liệu” và sheet “Bang\_copy”



Thay đổi cấu trúc vòng lặp for để in được giá trị sheet thứ mấy như sau:

import xlwings as xw

# Khởi tạo file excel bằng modul App

app = xw.App(visible=True, add\_book=False) # code này gán modul App với cái tên là app

# Đọc file excel có tên là "Covid\_VN.xlsx" từ folder "read\_Data"

wb = app.books.open(r'D:\MyBook\_PyExcel\read\_Datas\Covid\_VN.xlsx')  #('movies\_1.xls')

" -----> Đoạn này là bắt đầu tìm hiểu đối tượng Sheet"

# Đặt tên cho 1 danh sách các tên Sheets là "sh\_Names":

sh\_Names = wb.sheets

# In ra terminal đối tượng sh\_Names

print ("Đối tượng sh\_Names là:", sh\_Names)

# Dùng vòng lặp for đi qua từng sheet và in ra tên sheet như sau:

for i\_name in range(len(sh\_Names)): # Nhớ có các từ khóa "for"; "in" và dấu ":"

    print(f"Tên sheet thứ {i\_name+1} là:", sh\_Names[i\_name].name) # Nhớ dùng phím tab phía đầu hàm print để thụt vào

" Kết thúc tìm hiểu đối tượng Sheet <---------"

# lưu file "ketqua\_Covid.xlsx" tại folder "save\_Data"

wb.save(r'D:\MyBook\_PyExcel\save\_Datas\ketqua\_Covid.xlsx')

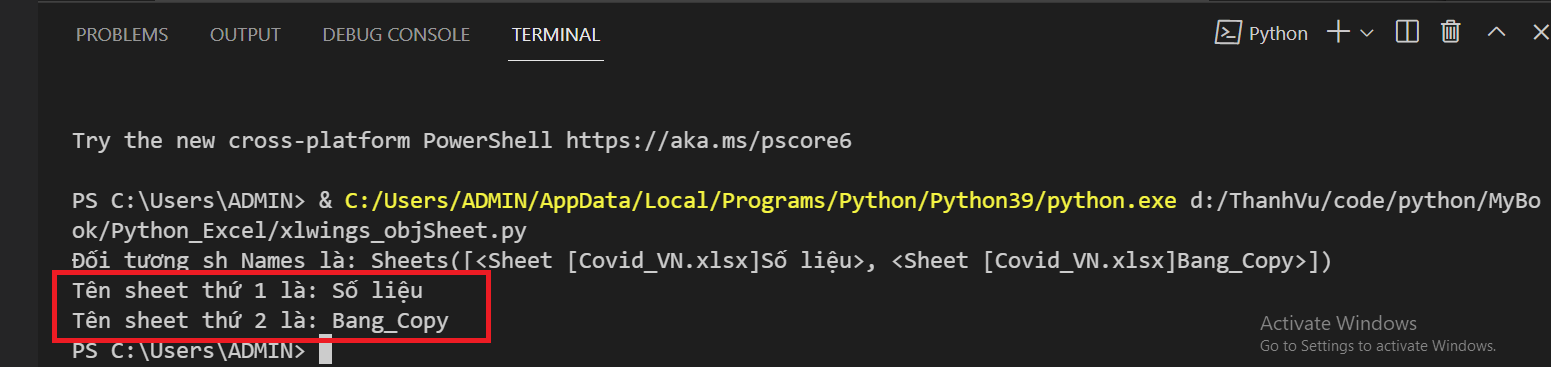
# Đoạn code này đóng file excel

wb.close()

# Đoạn code này tắt đối tượng app

app.quit()

Kết quả được in ra tại Terminal



**. Đếm tổng số sheets, tạo 1 sheet mới và đặt tên Sheet**

import xlwings as xw

# Khởi tạo file excel bằng modul App

app = xw.App(visible = True, add\_book = False) # code này gán modul App với cái tên là app

# Dòng này sẽ Tắt các thông báo excel (như update,... xảy ra 1 số file excel)

app.display\_alerts = False

# Đọc file excel có tên là "Covid\_VN.xlsx" từ folder "read\_Data"

wb = app.books.open(r'D:\MyBook\_PyExcel\read\_Datas\Covid\_VN.xlsx')  #('movies\_1.xls')

" -----> Đoạn này là bắt đầu tìm hiểu đối tượng Sheet"

# Đặt tên cho 1 danh sách các tên Sheets là "sh\_Names":

sh\_Names = wb.sheets

# Dùng hàm "len" để đếm trong có bao nhiêu sheet

sum\_numSheets = len(sh\_Names)

print("Tổng số sheets trước khi thêm là:", sum\_numSheets)

# Tạo 1 sheet mới và đặt tên là "VuNghiXuan"

sh\_new = sh\_Names.add("VuNghixuan")

# Tổng số sheets sau khi tạo sheet mới

print("Tổng số sheets sau khi thêm là:", len(sh\_Names))

# in ra tên tất cả các sheets

for i\_name in range(len(sh\_Names)): # Nhớ có các từ khóa "for"; "in" và dấu ":"

    print(f"Tên sheet thứ {i\_name+1} là:", sh\_Names[i\_name].name) # Nhớ dùng phím tab phía đầu hàm print để thụt vào

" Kết thúc tìm hiểu đối tượng Sheet <---------"

# lưu file "ketqua\_Covid.xlsx" tại folder "save\_Data"

wb.save(r'D:\MyBook\_PyExcel\save\_Datas\exp\_Covid.xlsx')

# Đoạn code này đóng file excel

wb.close()

# Đoạn code này tắt đối tượng app

for app in xw.apps:

    app.quit()

Kết quả sau khi chạy code



**Xử lý lỗi:**

Giả sử bạn chạy thêm 1 code nữa ở đoạn code phía trên, bạn sẽ thấy xuất hiện hàng loạt dòng lỗi.

Bạn tìm đến dòng **Traceback (most recent call last)**. Sau đó để ý 2 dòng kế tiếp (phần mình cho chữ màu đỏ), chẳng hạn, trên máy mình là:

File "d:\MyBook\_PyExcel\xlwings\_objSheet.py", **line 22**, in <module>

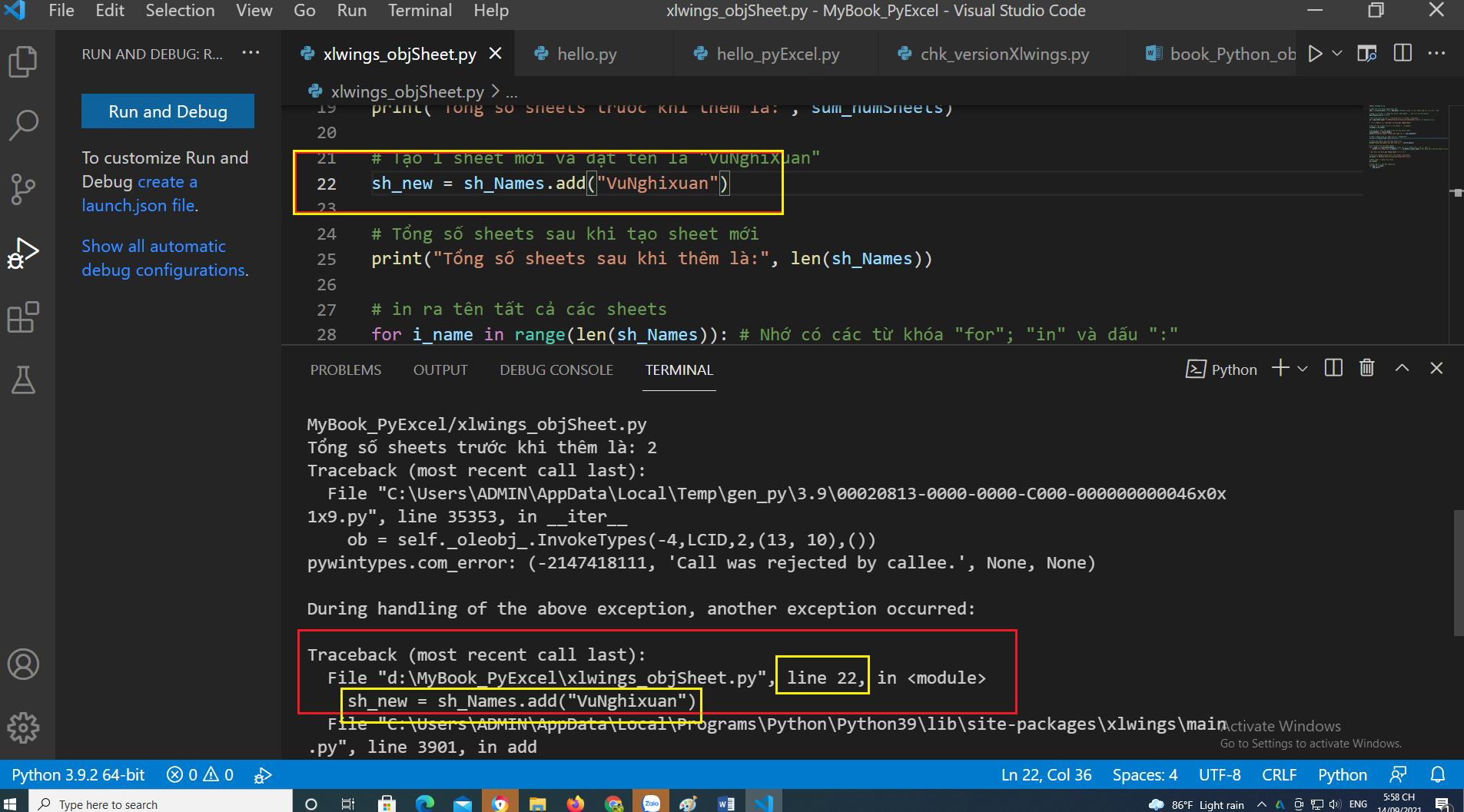
**sh\_new = sh\_Names.add("VuNghixuan")**

Đó là dòng bị lỗi. Tại sao vậy, bạn cố tình tiếp tục nhấn **run** tiếp (nút tam giác trên trình soạn thảo), vẫn là dòng lỗi ấy? Bạn dừng lại 1 chút nào, suy nghĩ **“line 22”** liên quan gì, dòng **sh\_new = sh\_Names.add("VuNghixuan")**

Có lẽ bạn đã đoán ra rồi phải không, **“line 22”** là dòng mình code trên trình soạn thảo chính là đoạn code: **sh\_new = sh\_Names.add("VuNghixuan")**

Nó phát sinh lỗi vì khi chạy lần đầu nó đã được tạo ra 1 Sheet có tên là " **VuNghixuan**” rồi, lần thứ 2 bị trùng tên nên gây ra lỗi (tức là không đồng nhất dữ liệu về tên).

Trước khi tìm cách sửa lỗi, bạn nhìn qua hình để hiểu rõ vị trí mình tìm để sửa lỗi



**Một số cách khắc phục vấn đề này như sau:**

**Cách 1:** Lấy tên toàn bộ danh sách tên các Sheets vào **list,** gán cái List này là listName, kiểm tra cái tên **“VuNghixuan”** có trong listName này không, nếu không thì tạo 1 sheet mới đặt tên là **“VuNghixuan”**

import xlwings as xw

from xlwings.main import Sheets

# Khởi tạo file excel bằng modul App

app = xw.App(visible = True, add\_book = False) # code này gán modul App với cái tên là app

# Dòng này sẽ Tắt các thông báo excel (như update,... xảy ra 1 số file excel)

app.display\_alerts = False

# Đọc file excel có tên là "Covid\_VN.xlsx" từ folder "read\_Data"

wb = app.books.open(r'D:\MyBook\_PyExcel\read\_Datas\objSheet.xlsx')  #('movies\_1.xls')

" -----> Đoạn này là bắt đầu tìm hiểu đối tượng Sheet"

# Đặt tên cho 1 danh sách các tên Sheets là "sh\_Names":

sh\_Names = wb.sheets

# Dùng hàm "len" để đếm trong có bao nhiêu sheet

sum\_numSheets = len(sh\_Names)

print("Tổng số sheets trước khi thêm là:", sum\_numSheets)

# Tạo 1 sheet mới và đặt tên là "VuNghiXuan"

listName = [sh.name for sh in sh\_Names]

if "VuNghixuan" not in listName:

    sh\_new = sh\_Names.add("VuNghixuan")

# Tổng số sheets sau khi tạo sheet mới

print("Tổng số sheets sau khi thêm là:", len(sh\_Names))

# in ra tên tất cả các sheets

for i\_name in range(len(sh\_Names)): # Nhớ có các từ khóa "for"; "in" và dấu ":"

    print(f"Tên sheet thứ {i\_name+1} là:", sh\_Names[i\_name].name) # Nhớ dùng phím tab phía đầu hàm print để thụt vào

" Kết thúc tìm hiểu đối tượng Sheet <---------"

# lưu file "ketqua\_Covid.xlsx" tại folder "save\_Data"

wb.save(r'D:\MyBook\_PyExcel\read\_Datas\objSheet.xlsx')

# Đoạn code này đóng file excel

wb.close()

# Đoạn code này tắt đối tượng app

for app in xw.apps:

    app.quit()

**Giải tích code:** Phần này liên quan đến code cơ bản Python

**list** trong python được xem là 1 danh sách (giống như mảng trong các ngôn ngữ khác), được ký hiệu bằng 2 dấu ngoặc vuông **[ ]**.

Ví dụ:

# Nhập vào các danh sách học sinh Lớp Python\_Excel gồm có 4 tổ

To\_1 = ["An", "Cường", "Đào", "Thịnh"]

To\_2 = ["Thúy", "Mận", "Khang"]

To\_3 = ["Phú", "Loan", "Duyên", "Thái", "Dần"]

To\_4 = ["Hạnh", "Hồng", "Nhật", "Nguyệt", "Cang"]

# Cách sắp xếp các tổ thành danh sách lớp

lop\_pyExcel = To\_1 + To\_2 + To\_3 + To\_4

# In ra danh sách Lớp Python\_Excel

print(f"Danh sách lớp lop\_pyExcel: {lop\_pyExcel}. Có tổng cộng: {len(lop\_pyExcel)} (người) ")

# In ra Tên người thứ 2 trong tổ 3

print("Tên người đầu tiên trong tổ 1 là:", To\_1[0])

# In ra Tên người thứ 2 trong tổ 3

print("Tên người thứ 2 trong tổ 3 là:", To\_3[1])

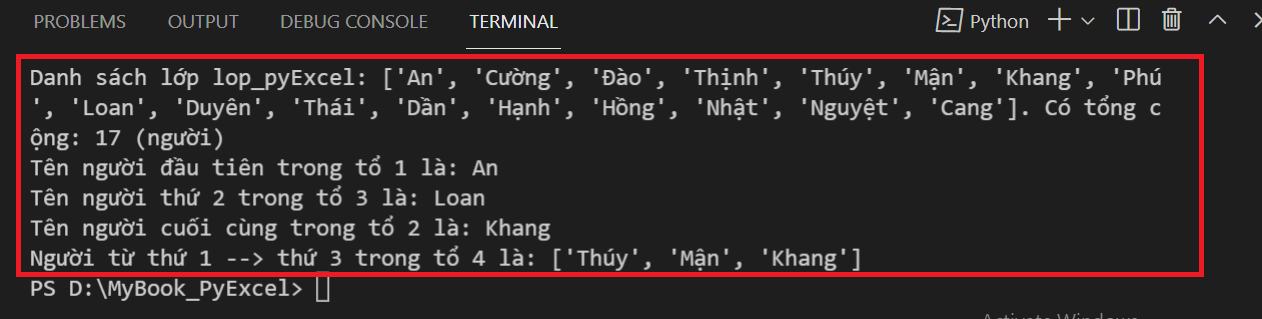
# In ra Tên người cuối cùng trong tổ 2

print("Tên người cuối cùng trong tổ 2 là:", To\_2[-1])

# In ra Người từ thứ 1 đến người thứ 3 trong tổ 4

print("Người từ thứ 1 --> thứ 3 trong tổ 4 là:", To\_2[0:4])

Kết quả xuất ra ở Terminal:



. Viết trong code python: **Nhom\_1 = [“An”, “Cường”, “Đào”, “Thịnh”]**

Tham khảo code nhập vào và xuất tên từng phần tử trong 1 **list** danh sách

**. Đổi tên hàng loạt tên các sheets**

1. **Thuộc tính Sheet**
2. **Lấy dữ liệu trong Sheet**
3. **Xử lý data**