TỰ HỌC PYTHON EXCEL

Dành cho người MỚI BẮT đầu

(đặng thanh vũ)

2021

**GIỚI THIỆU**

Tài liệu này đúng nghĩa là 1 cuốn nhật ký trong quá trình tự học của group Python Excel của tụi mình. Nó là ý tưởng, được ghi chép lại cách 1 cách cụ thể và chi tiết các vấn đề đã học, những gút mắc và hướng giải quyết trong quá trình tiếp cận với 1 ngôn ngữ lập trình hiện đại.

Do đó, nếu bạn chỉ biết cơ bản về Excel *(các bạn là dân văn phòng, công sở, sinh viên, học sinh, ...ở các lĩnh vực khác)*, thì bạn cũng có thể đọc và thực hành từng bước theo tài liệu này. Đoạn, phần nào bạn không hiểu có thể tham gia thảo luận cùng với group tại đây: <https://www.facebook.com/groups/1015080755701155>

Cuối cùng, vì nó không phải là sản phẩm thương mại, lại là những người ngoại đạo về lập trình, nên từ ngữ trong tài liệu mang tính chất nông dân học vụ. Các bạn đọc tham gia góp ý tích cực để mình chỉnh sửa ngày một hoàn thiện hơn. Bấy nhiêu đó cũng là niềm động viên quý báu nhất mà mình mong nhận được từ các bạn.

Chân thành cảm ơn!

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

**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. **import:** là 1 từ khóa (nên hiểu là 1 hàm) báo hiệu cho Python biết tui bắt đầu nhập vào thư viện (ở bài học các bạn đang nhập vào là thư viện xlwings). Ngoài thư viện sau này các bạn còn phải nhập vào gói, modul (tạm thời chưa vội đề cập ở đây)
2. **xlwings:** là 1 thư viện chứa các modul (hàm) giúp chúng ta tương tác với Excel, như gọi và đọc dữ liệu (data) file excel, xử lý dữ liệu, điều khiển bảng tính excel,…
3. **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**

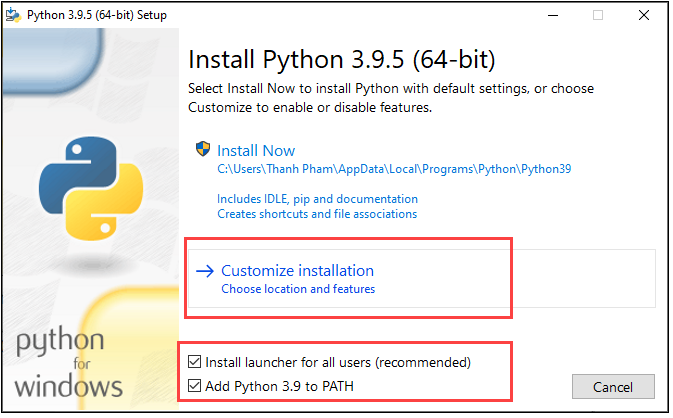
Truy cập vào trang: <https://www.python.org/downloads/> để cài đặt Python cho hệ điều hành bạn đang sử dụng.

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”

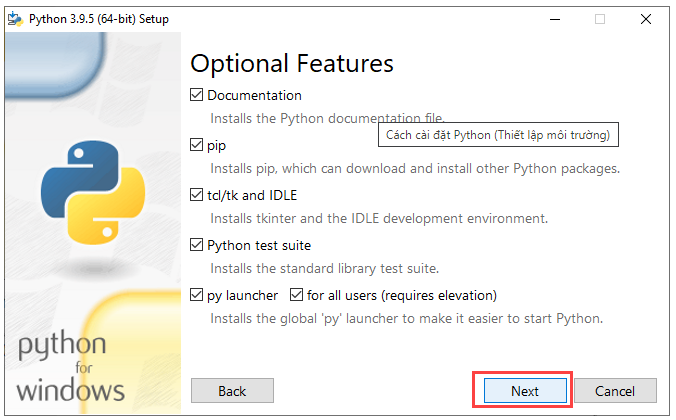


Click đúp chuột vào file vừa tải xuống (python-3.9.5-amd64.exe) để tiến hành cài đặt Python: Phần màu đỏ cho windows, các hệ điều hành khác là màu xanh.

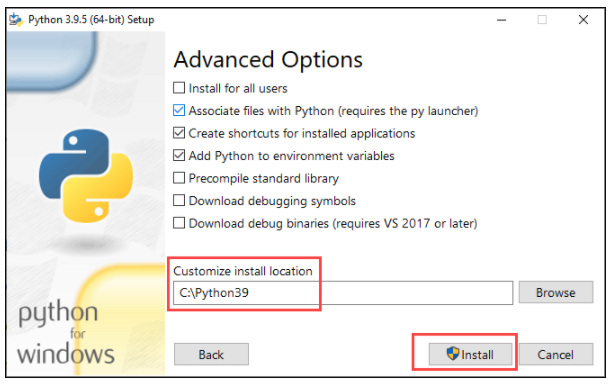
Tiếp theo Chọn **Customize installation**:



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

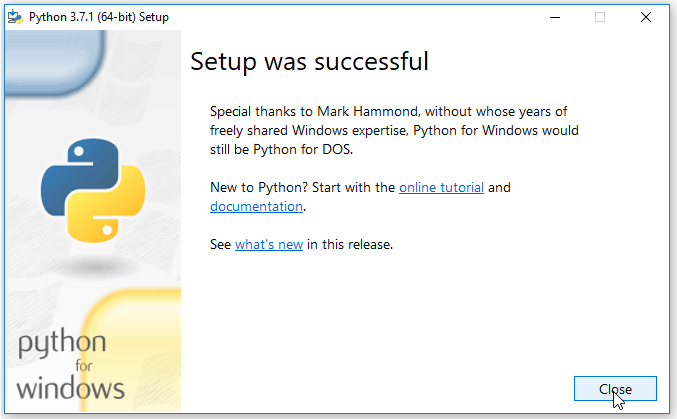


Chọn các tính năng tùy chọn như trong hình, click Install:

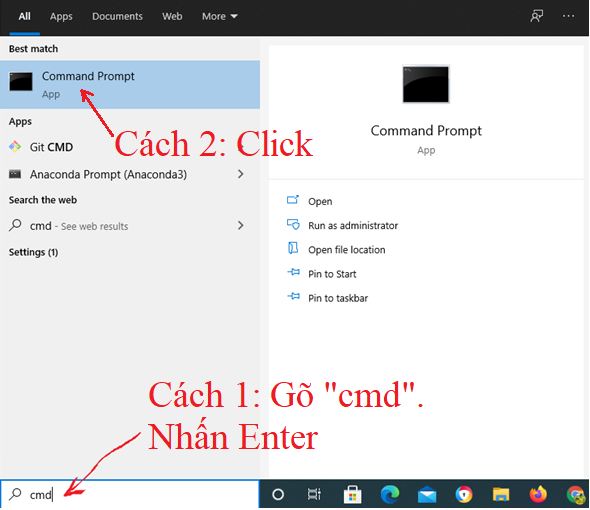


Nếu bạn muốn để mặc định như mình thì không phải chọn button Browse, nhấn luôn 🡪**Install**

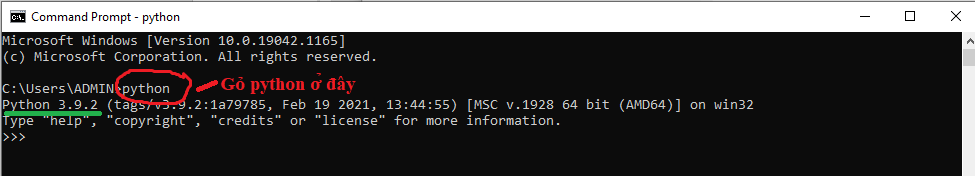
Kết quả sau cài đặt:

****

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



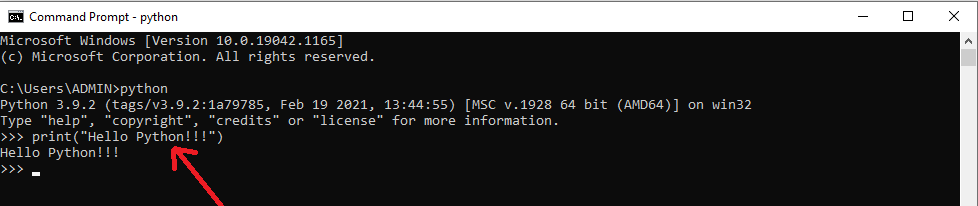
Sau khi **Command** **Prompt** xuất hiện, bạn gõ “python”. Kết quả như sau:



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!!!”)*



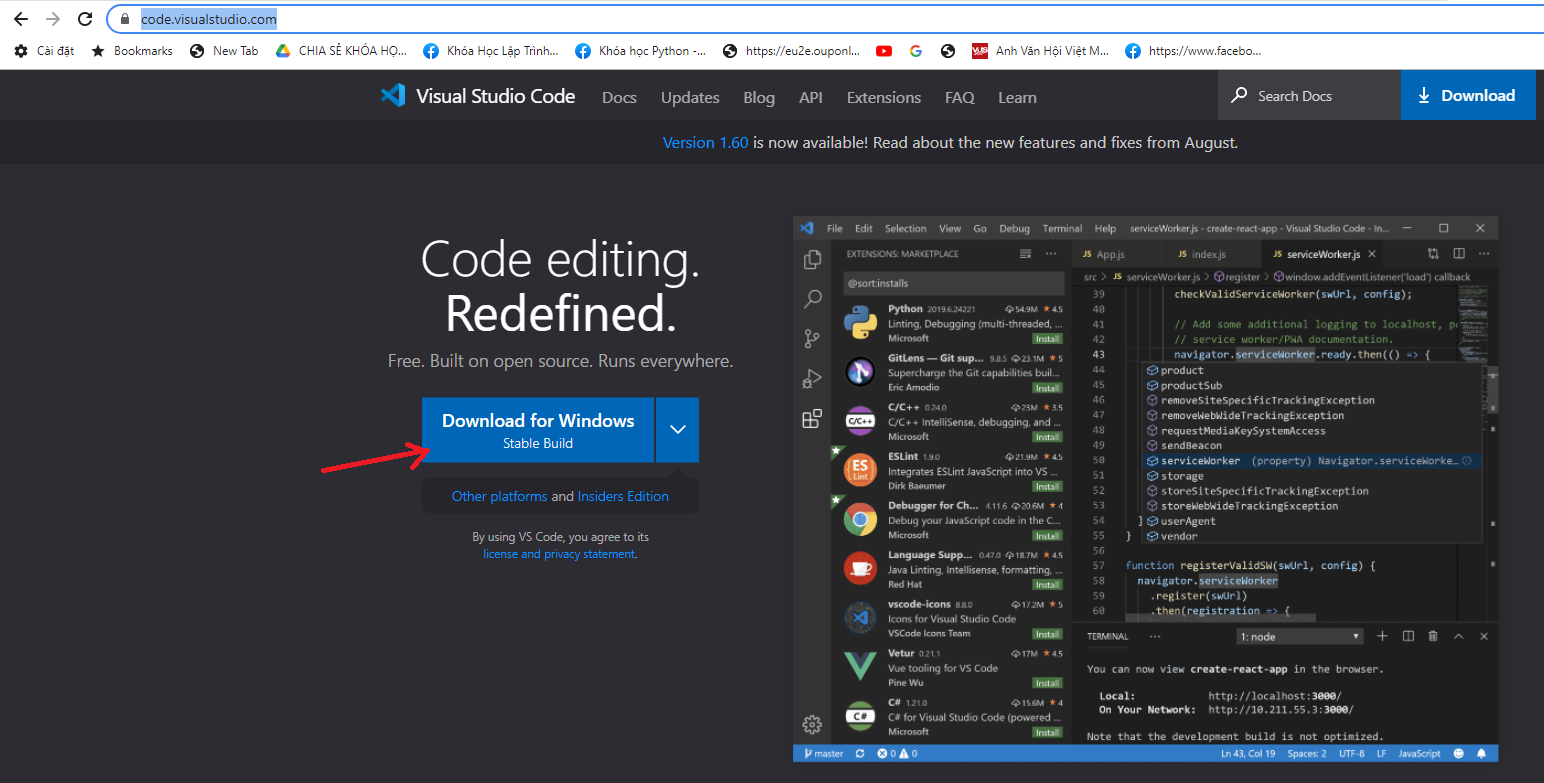
1. **HƯỚNG DẪN CÀI ĐẶT VISUAL STUDIO CODE**

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.

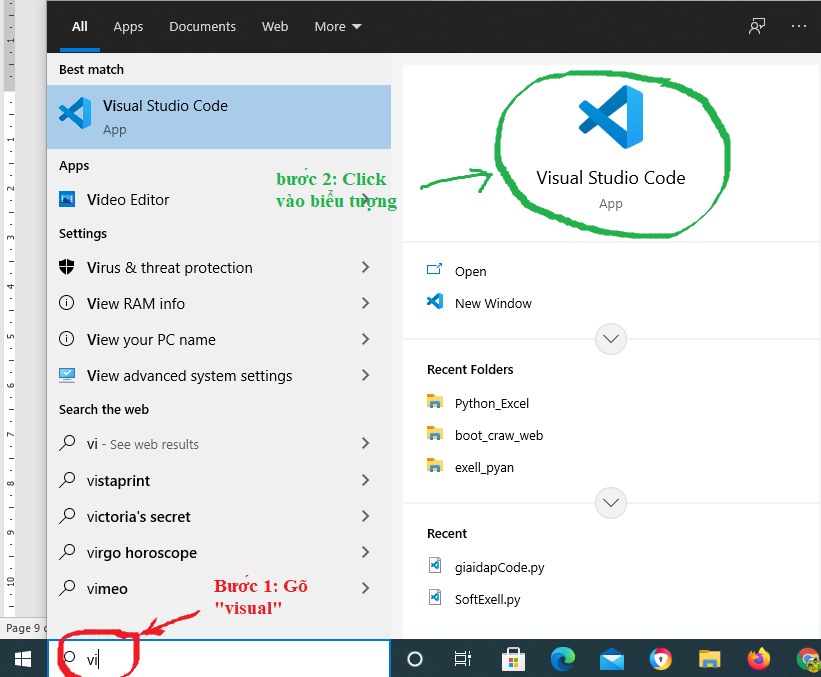
Truy cập vào trang: <https://code.visualstudio.com/>

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).

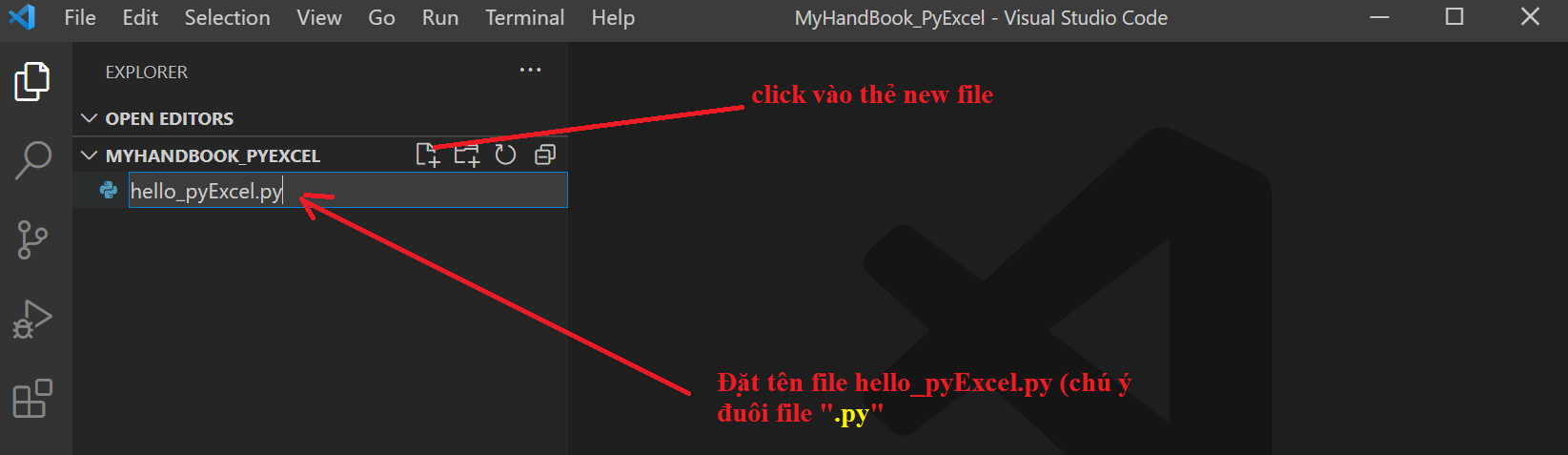


Ngoài ra, VSC còn tích hợp nhiều ứng dụng giúp cho quá trình soạn thảo đẹp hơn, nhanh chóng và tiện ích (có thể google để tìm ra những thứ cần thiết cho chính bạn)

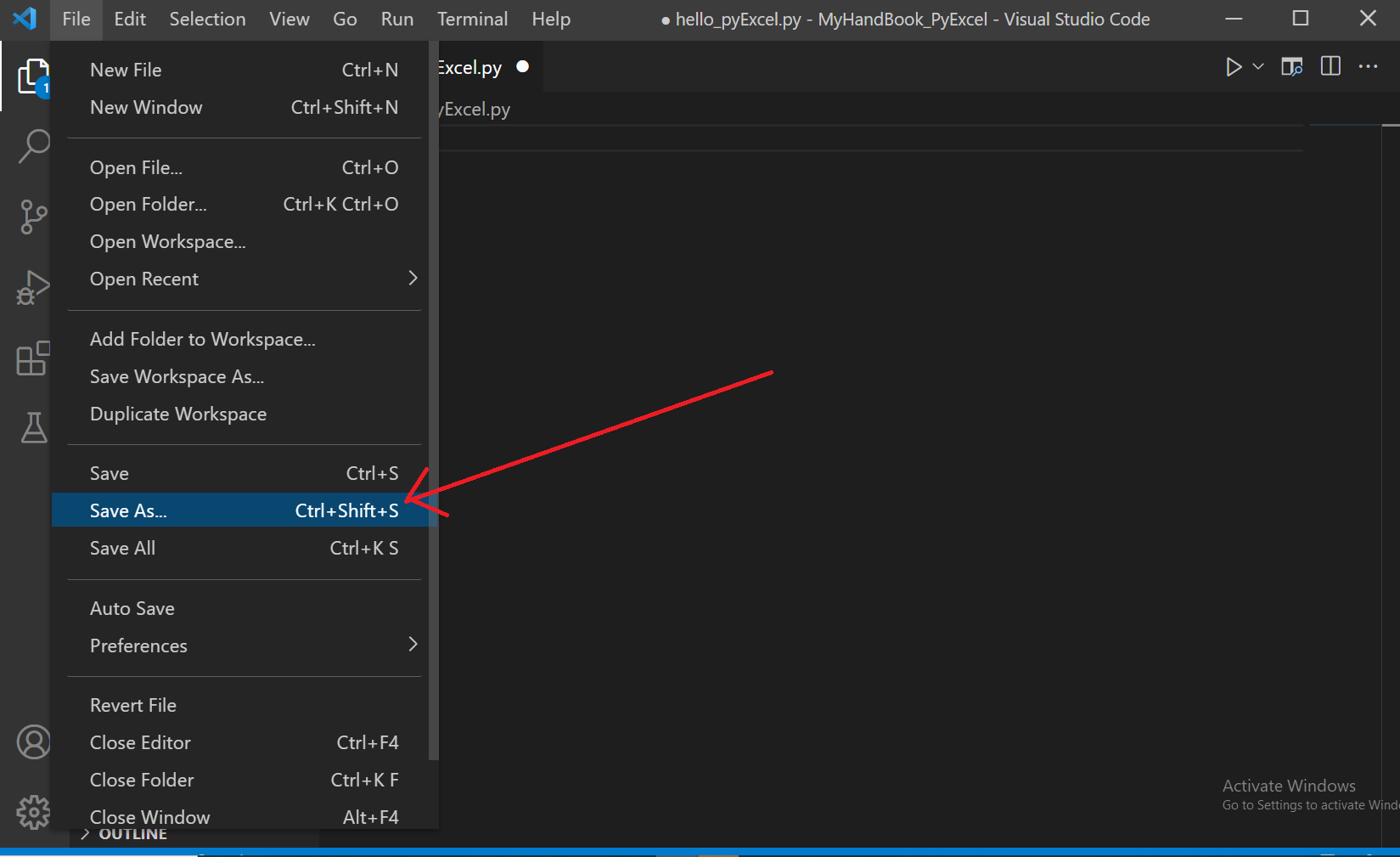
**Hướng dẫn soạn thảo Python trên Visual Studio cơ bản**

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.

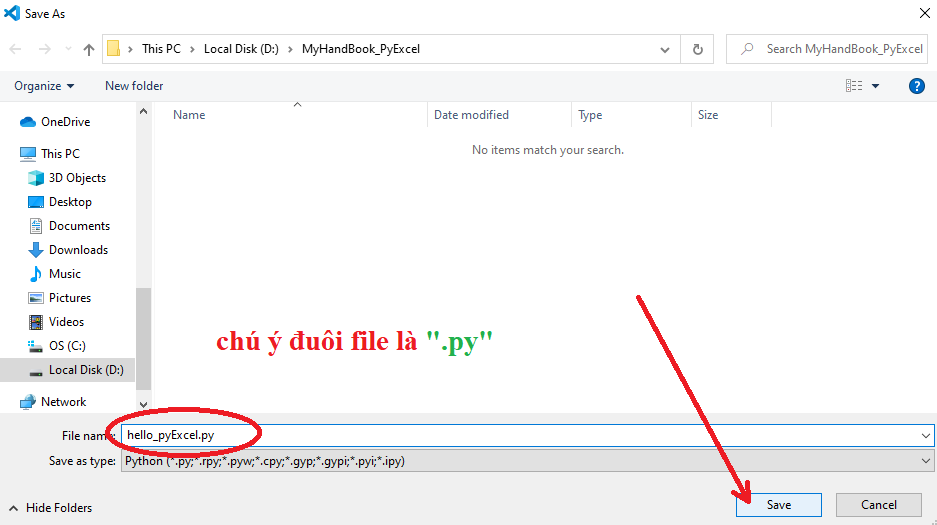
* **Tạo File script python**
* Bước 1: Tạo file **hello\_pyExcel.py**, chú ý đuôi file là **“.py”**



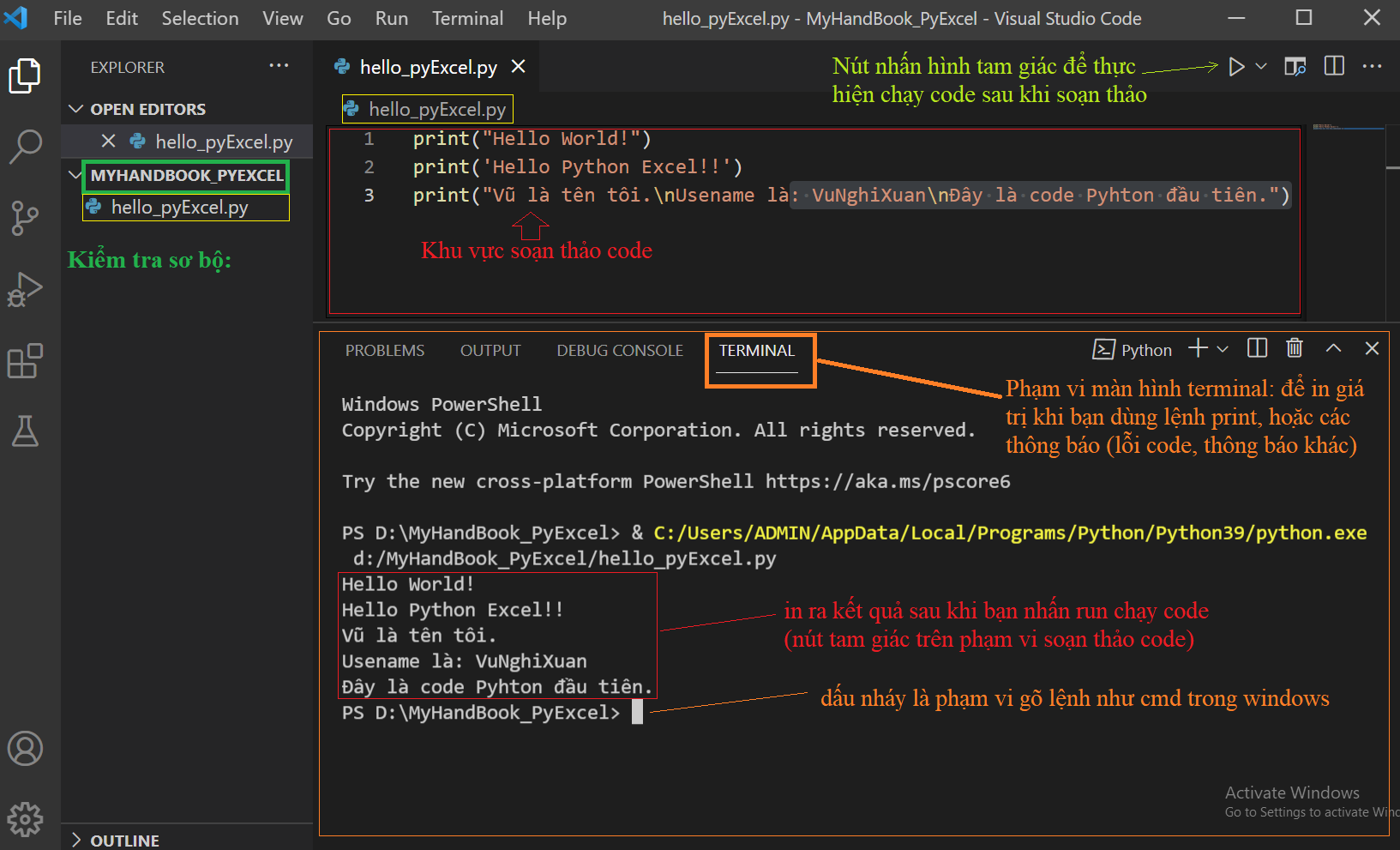
Thực hiện lưu file



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

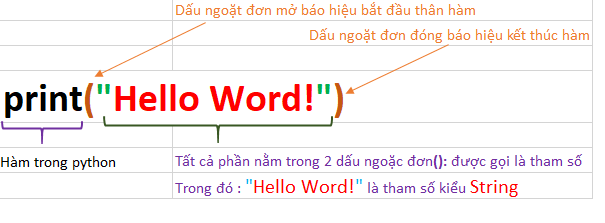


Link code: <https://github.com/VuNghiXuan/giaotrinhPythonExcel/tree/main/C1>

**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:**

****

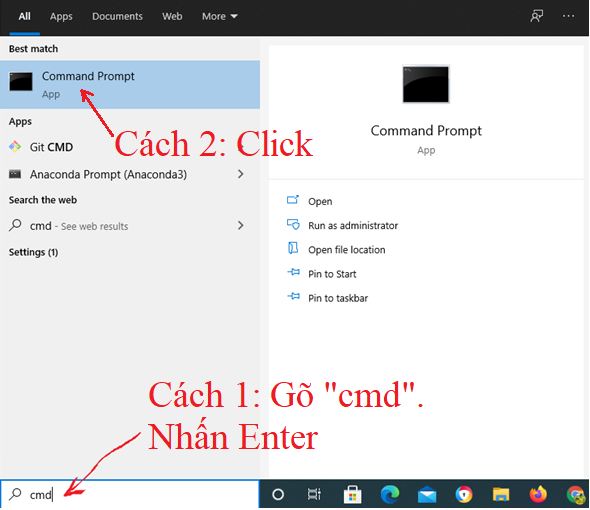
+ Hàm **print** trong python: dùng để in ra màn hình Terminal.

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

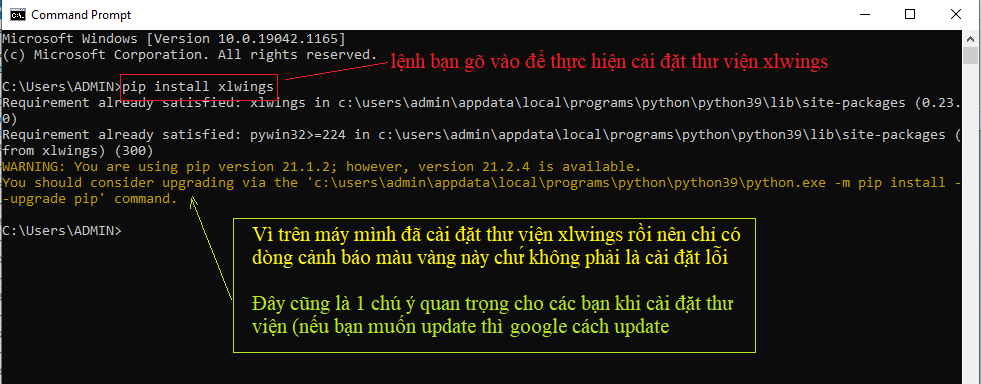
+ "Hello Word!": là 1 tham số của hàm, kiểu String. Đối với kiểu String có thể dùng dấu ‘Hello Word!’ thay cho “Hello Word!” *(hãy xem dòng code thứ 2 của trình soạn thảo phía trên).*

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)**



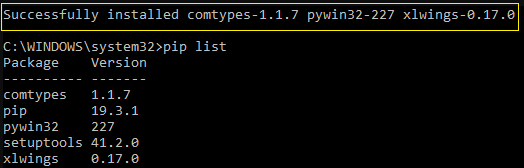
Khi màn hình **Command Prompt** xuất hiện tại vị trí chuột nhấp nháy là phần gõ lệnh, bạn gõ: **pip install xlwings**



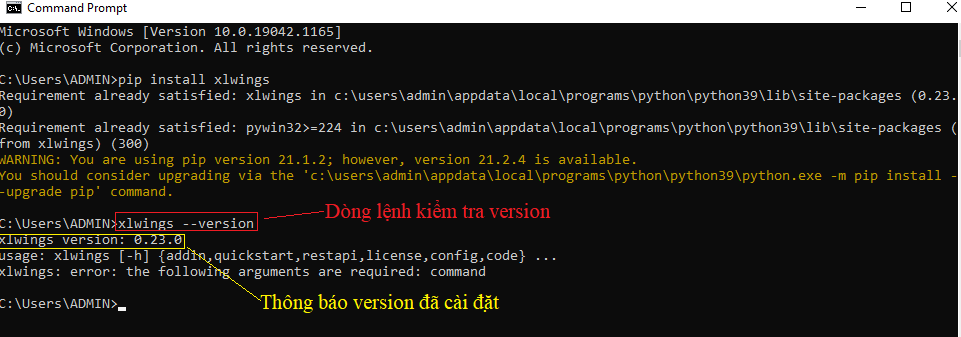
Để cập nhật thay vì cài đặt bạn dùng lệnh: pip install --upgrade xlwings

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

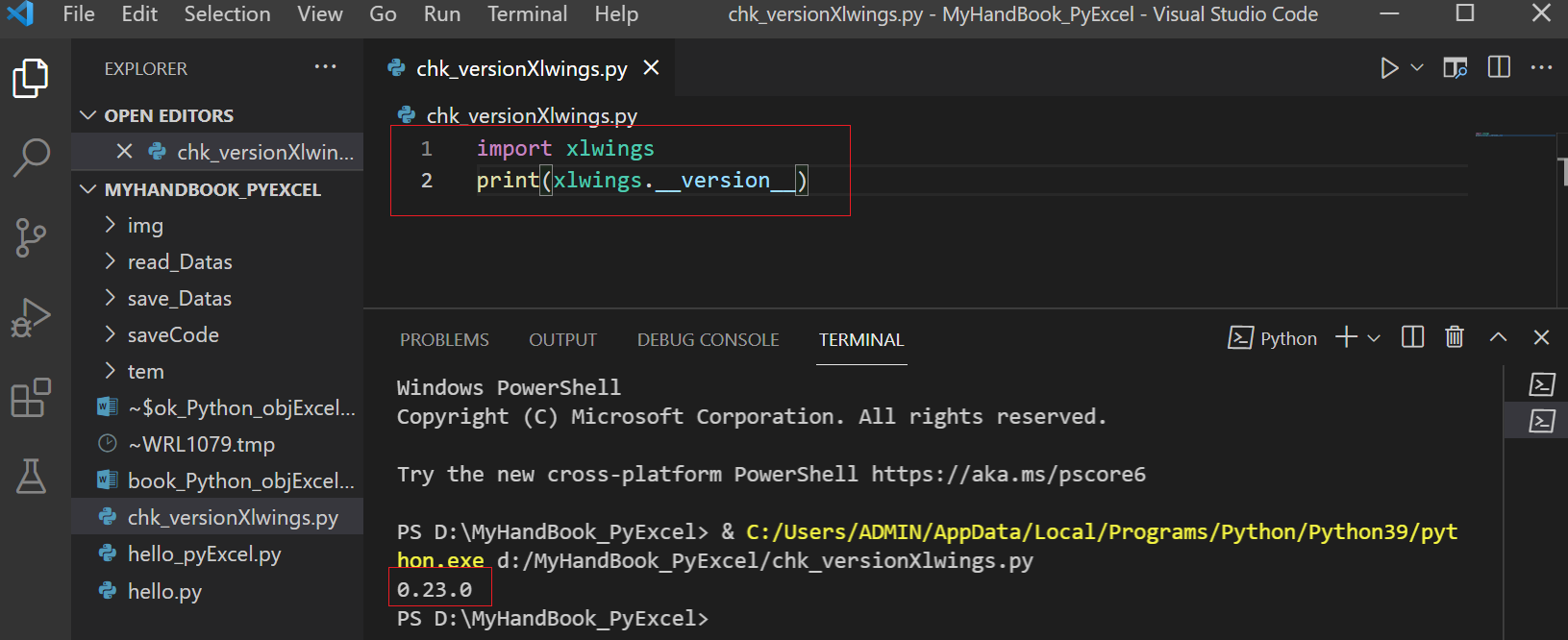


**. Cách 3:** Trong trình soạn thảo VSC bạn thực hiện 2 dòng lệnh

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:**

## Trong chương này mình chỉ gói gọn 1 số đối tượng, phương thức (hàm) và 1 số thuộc tính thông dụng trong python để phục vụ cho bạn lập trình trong thư viện xlwings (thư viện giao tiếp, điều khiển excel). Khuyến khích bạn google thêm để học hết cơ bản về python, hoặc bạn có thể down tài liệu ("Python rat la co ban - Vo Duy Tuan 2.pdf “) tại đây:

## <https://github.com/VuNghiXuan/document_python_basic>

***(Tài liệu này mình tham khảo từ các bạn học Python cơ bản trong các group, mình chưa đọc nhé! Vấn đề bản quyền bạn nên liên hệ chính tác giả để biết thông tin)***

**I. LÀM QUEN VỚI NGÔN NGỮ PYTHON:**

* **Python lists:**

Trong python, **list** được xem là 1 danh sách (giống như 1cái thực đơn món ăn), được ký hiệu bằng 2 dấu ngoặc vuông **[ ]**, để dễ hình dung bạn hãy xem nó như 1 chiếc thùng, cái túi khổng lồ chứa nhiều đồ vật dụng (bút, viết, kẹo bánh, tiền…). Pyhton cho phép bạn chứa tất cả các kiểu, thậm chí khi cần bạn có thể nới rộng cái túi ấy lớn hơn bằng cách thêm vào (append).

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"]

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

# Giả sử có thêm 1 bạn vào tổ 3; Ta sử dụng phương thức append() để nới rộng list như sau:

To\_3.append("Dần")

# 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

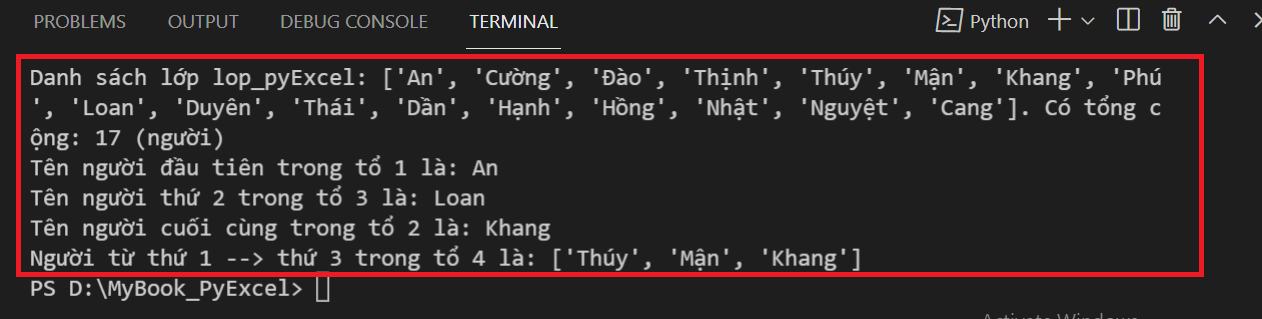
print("Tên người cuối cùng trong tổ 2 là:", To\_2[-1])

# 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])

link code: <https://github.com/VuNghiXuan/giaotrinhPythonExcel/tree/main/C2>

Kết quả xuất ra ở Terminal:



* **Python lists trong list:**

Khi tiếp cận với Excel, hoặc các dạng bảng dữ liệu, phân tích hình ảnh. thì list còn chứa trong nó các list con gọi là “list trong list”, nghĩa là list lồng với list, ký hiệu **[**["a"], [1,5.6,3], ["Jerry” ,2, "“Tom“]**]**. Ví dụ như: **list** túi lớn khủng lồ chứa nhiều loại vật dụng nói trên, trong cái túi đó lại chứa 1 cái túi nhỏ hơn dựng nhiều loại bánh kẹo, thì túi lớn chứa túi nhỏ này được gọi là **list trong list**.

# list và list trong list

list\_DanhSach = ["An", "Cường", "Đào", "Thịnh"]

list\_int = [1, 2, 3, 2.5, 5, 5.6] # Số nguyên (kiểu int) và số thực (kiểu float)

list\_str\_num = [ 5, "Hồng", 126.4554]

"Kiểu list trong list, trong bảng Ecxel chứa hàng và cột dữ liệu "

data\_Excel = [["STT", "Họ và tên", "Năm sinh"],

              [1, "Đặng Vũ Nghi Xuân", 2009],

              [2, 'Đặng Diệp Quân', 2012]]

print(list\_DanhSach)

print(list\_int)

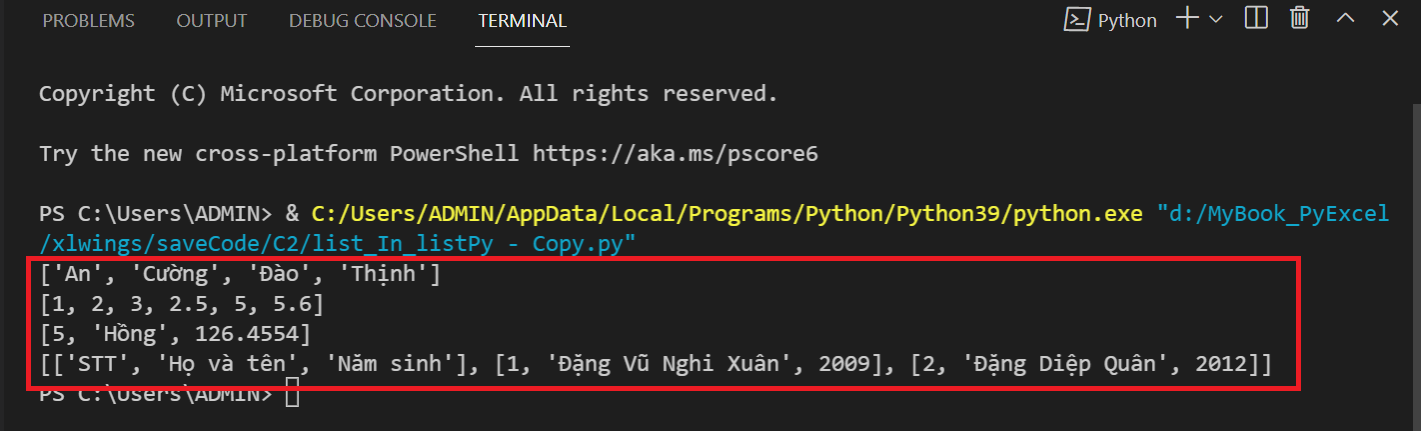
print(list\_str\_num)

print(data\_Excel)

link code tham khảo:

<https://github.com/VuNghiXuan/giaotrinhPythonExcel/blob/main/C2/list_In_listPy.py>

Kết quả chạy code in ra màn hình Terminal



**. Vòng lặp for trong list python:**

**a/ Cách lấy element trong list**

Như các ngôn ngữ khác, Python cung cấp cho bạn vòng lặp bắt đầu bằng từ khóa (key) là **for** kết hợi với **in,** cuối cùng là dấu **:**

Cú pháp:

**for** element\_list **in** list**:**

<nhấn phím Tab> Khối lệnh của for

Bạn tưởng tượng **list**  là 1 cái hộp chứa nhiều vật dụng trong đó. Mỗi 1 vật đồ dùng là 1 **element\_list (**từng phần tử trong list**).** Vòng lặp for được hiểu làđi đến từng phần tử trong hộp, bạn có thể đọc đọc được tên nó, lấy số thứ tự (vị trí nó trong cái hộp) ra và biến nó thành 1 cái khác chẳng hạn.

Dưới đây là đoạn code ví dụ:

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

hop = ["Bánh", "Kẹo", "Viết", "Cục tẩy"]

# Dùng vòng lặp for lấy tên các loại chứa trong cái hộp

for element\_Hop in hop:

    print(element\_Hop)

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

hop = ["Bánh", "Kẹo", "Viết", "Cục tẩy"]

# Dùng vòng lặp for lấy tên các loại chứa trong cái hộp

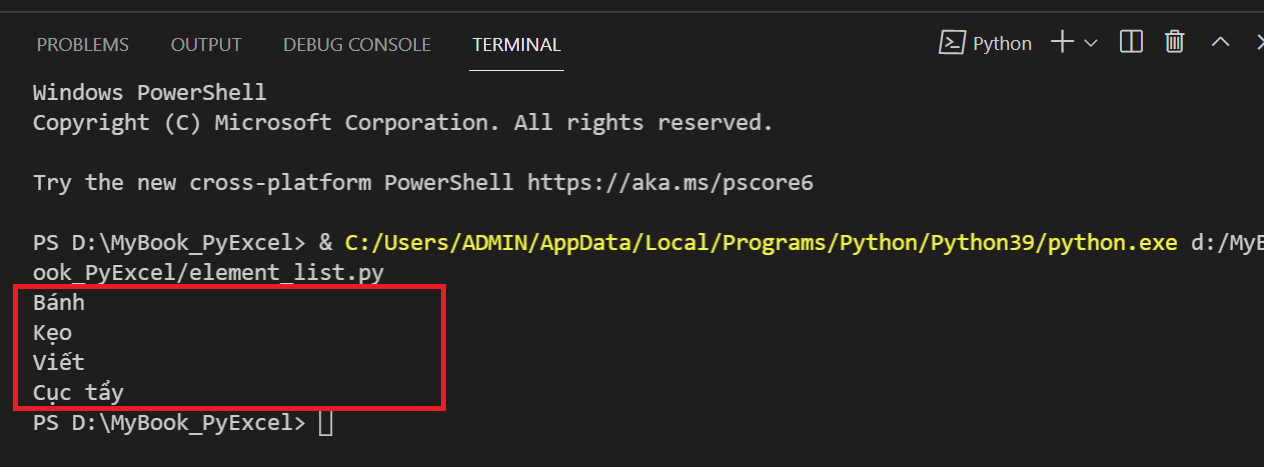
for element\_Hop in hop:

    print(element\_Hop)

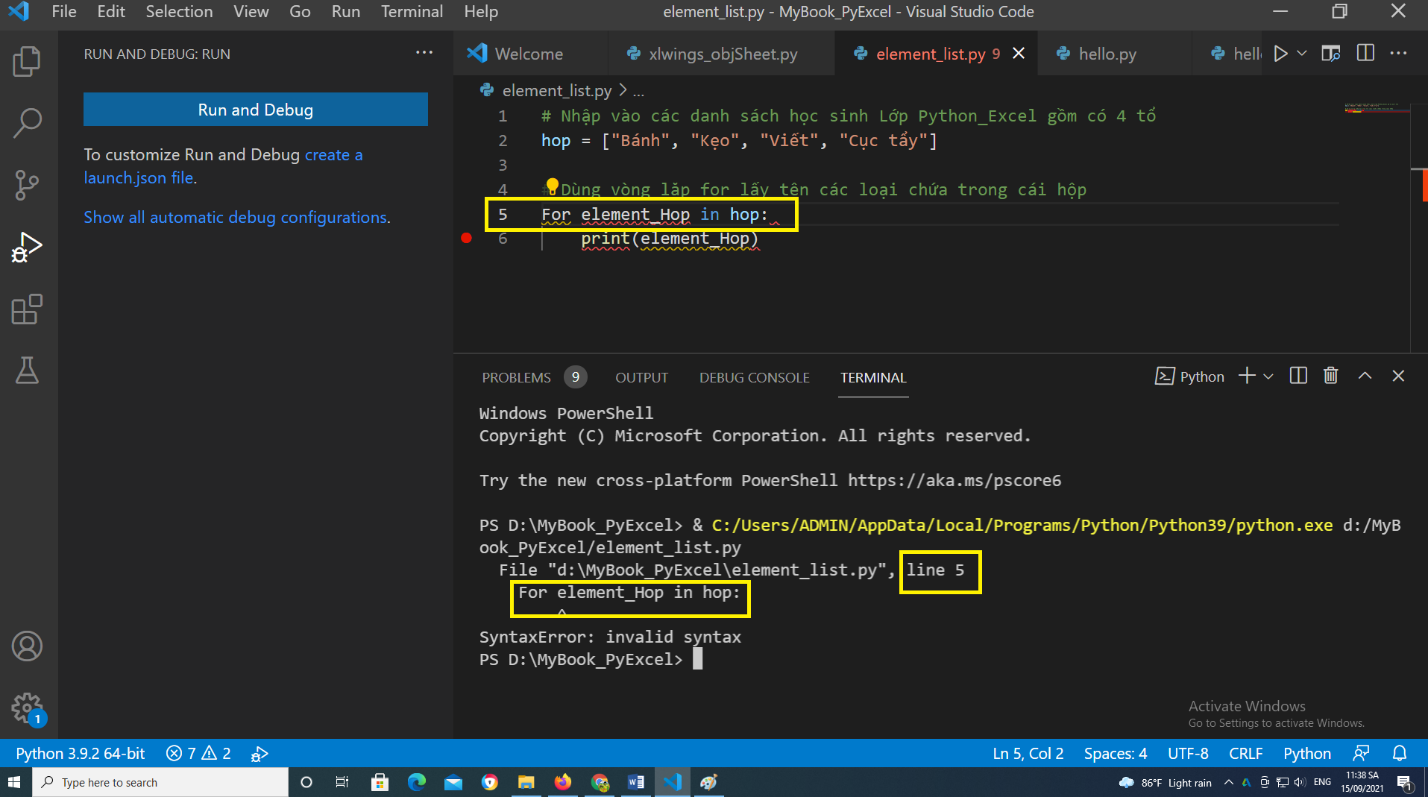
link code:

<https://github.com/VuNghiXuan/giaotrinhPythonExcel/blob/main/C2/Vonglap_For.py>

Kết quả:

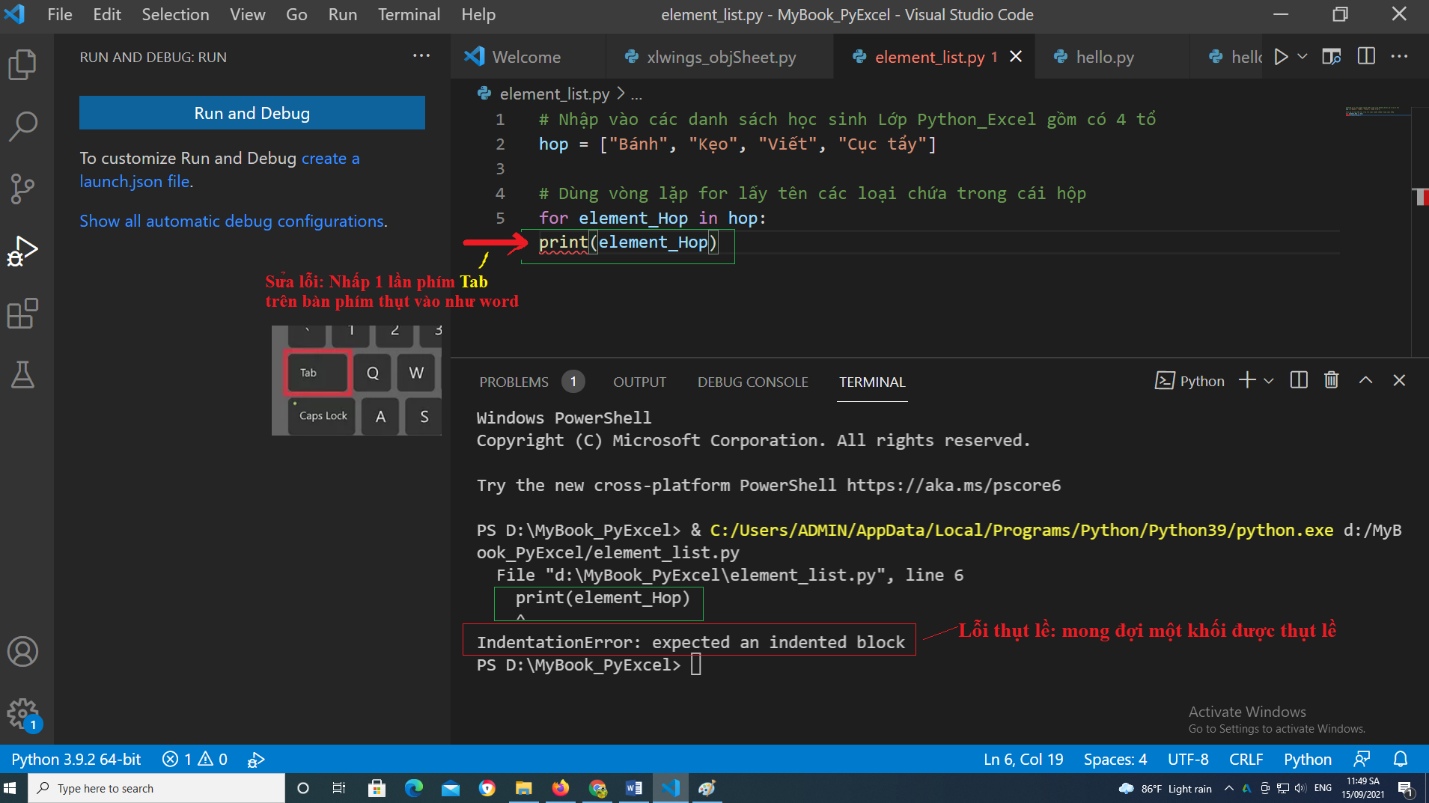


Lưu ý về lỗi:

****

Lỗi này phát sinh tại dòng 5 của đoạn code, lý do là bạn viết sai từ khóa **“For”**, từ khóa đúng là **“for”**, tức là viết thường không viết hoa. Ngoài ra: nếu bạn thiếu, hay sai từ khóa **“in”** hoặc dấu **“:“** sẽcó thông báo lỗi giống vậy.

Tiếp theo là bị lỗi không phím **“Tab”** để khai báo thân hàm vòng lặp:



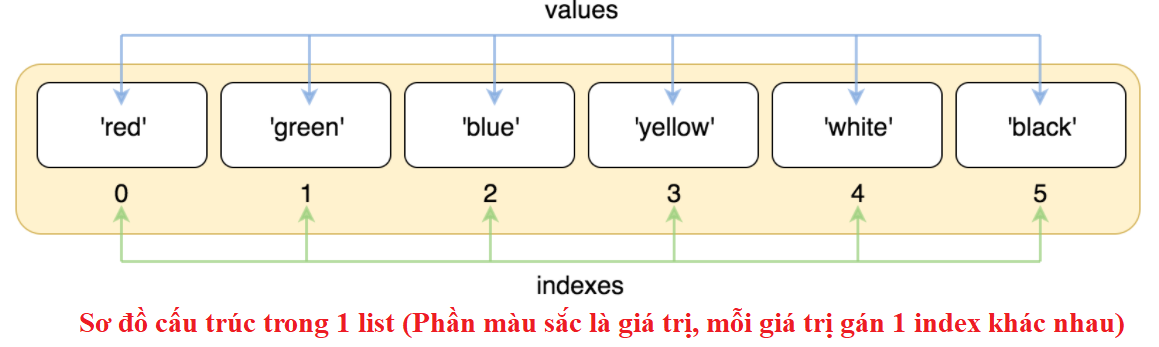
**b/ Cách lấy index\_list trong list**

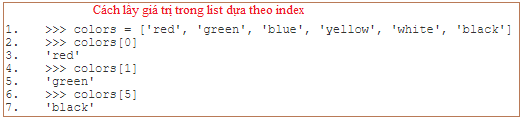
Cú pháp:

**for** index\_list **in** range(len(list))**:**

<nhấn phím Tab> Khối lệnh của for

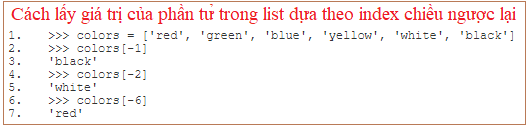
**index\_list (**chỉ số mục trong list**):** Giống như mã ID cho password hay số chứng minh nhân dân của các bạn, index\_list trong list chứa các phần tử (element), mỗi elment có gán với index trong list





Hoặc bạn có thể lấy giá trị các phẩn tử trong list, bắt đầu từ chỉ số index cuối list ngược về phần tử đầu tiên.





Trong list phần tử đầu tiên có chỉ số index = 0 và phần tử cuối cùng bắt đầu đếm từ phần tử đầu tiên này cộng dồn về đến phần tử cuối cùng. Ví dụ list **colors** trên phần tử đầu tiên có giá trị là “red” ứng với index = 0. Phần tử cuối cùng là "“back"”, tương ứng với index = 5.

Để bạn dễ hình dung về giá trị và index trong list mình tiến hành trực tiếp trên code:

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

colors = ["red", "green", "blue", "yellow", "white", "black"]

"""

list colors màu trên có các chỉ số index lần lượt là:

    + Giá trị "red" -->index= 0

    + Giá trị "green" -->index= 1

    + Giá trị "blue" -->index= 2

    + Giá trị "yellow" -->index= 3

    + Giá trị "white" -->index= 4

    + Giá trị "black" -->index= 5

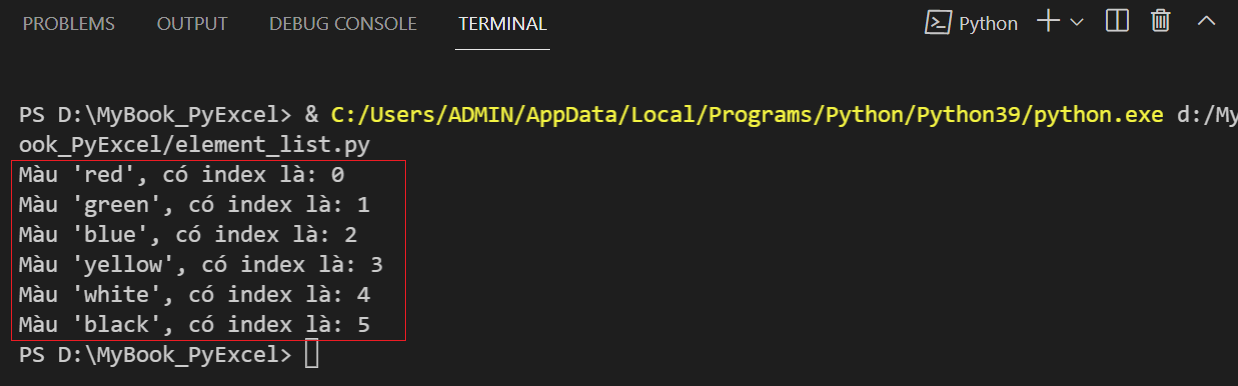
"""

# In ra giá trị màu xem có đúng theo chỉ số index trên không?

for i\_color in range(len(colors)):

    print(f"Màu '{colors[i\_color]}', có index là: {i\_color}")

Kết quả được xuất ra Terminal



**Giải thích các hàm mới cho đoạn code trên:**

**len(colors):**  Hàm trả về tổng số phần tử (element) của 1 list colors. Mình có 6 màu thì **len(colors) = 6.** Giá trị này trả về cho hàm range bên trong nó là số 6, tức là **range(6)**

**range:** là 1 hàm trong python, nghĩa là phạm vi, khoảng giới hạn của các phần tử trong list **và giới hạn cho index trong list tại vị trí cuối cùng** (nghĩa là không thực hiện chạy code cho index cuối cùng). Ví dụ list **colors** có **(len(colors)) =6**, nghĩa là index **i\_color** chạy liên tục thứ tự từ 0🡪6, tuy nhiên khi chạy đến **i\_color** = 5 thì tổng số phần tử đã là 6 (là màu "**black"**) là đã hết giá trị trong list **colors** (Bạn nhìn lại tại khung màu đỏ mà Terminal trả về dòng cuối cùng ở hình trên sẽ rõ).

**CHƯƠNG 3:**

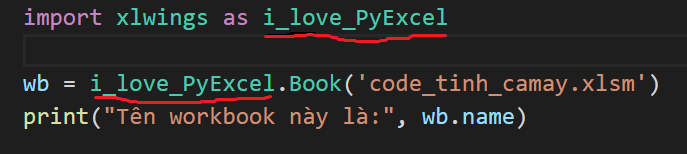
1. **TÌM HIỂU VỀ THƯ VIỆN XLWINGS:**

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.

Hãy khởi động 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.

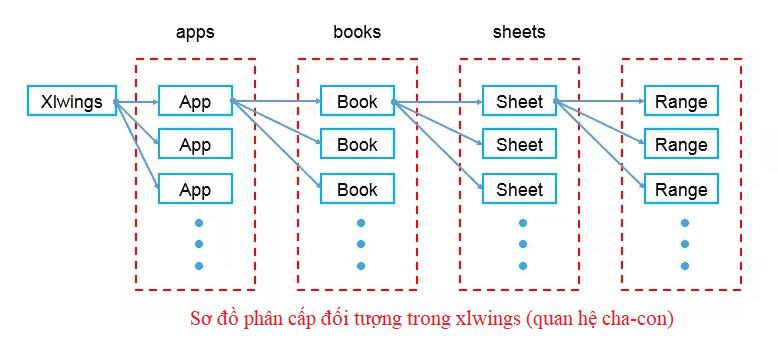


Mình đặt tên biến “i\_love\_PyExcel” thay cho cách dùng “xw”, chương trình vẫn hoạt động bình thường.

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), …



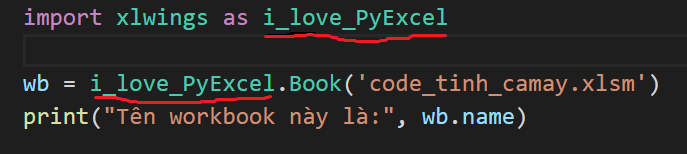
Túm lại, thuộc tính là chỉ hành động: đi, đứng, chạy,..thuộc tính là: màu sắc, tính cách, mô tả từng bộ phận của đối tượng đó.

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.

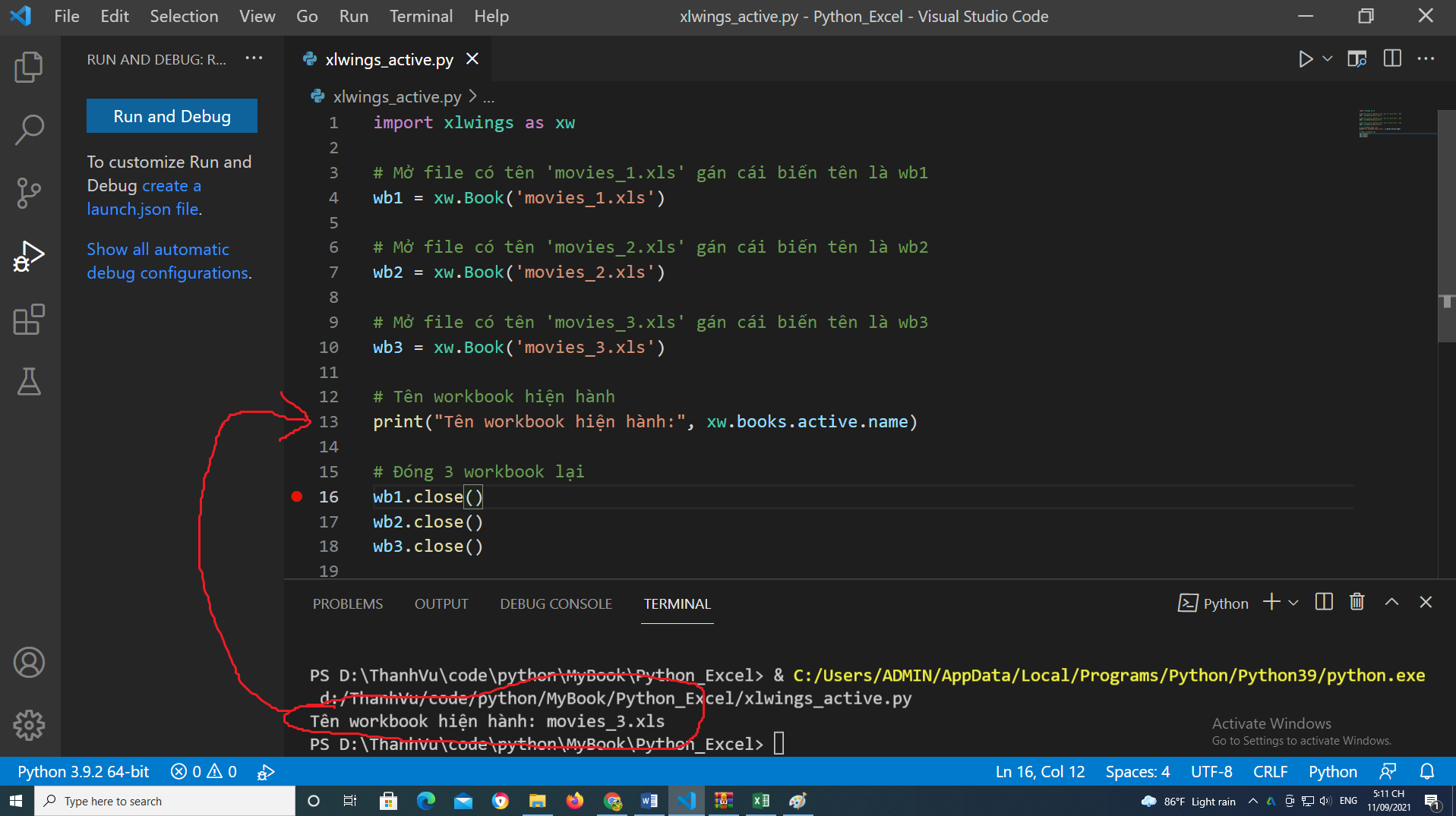
**\* Phần giải thích thêm về thuộc tính active**

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)

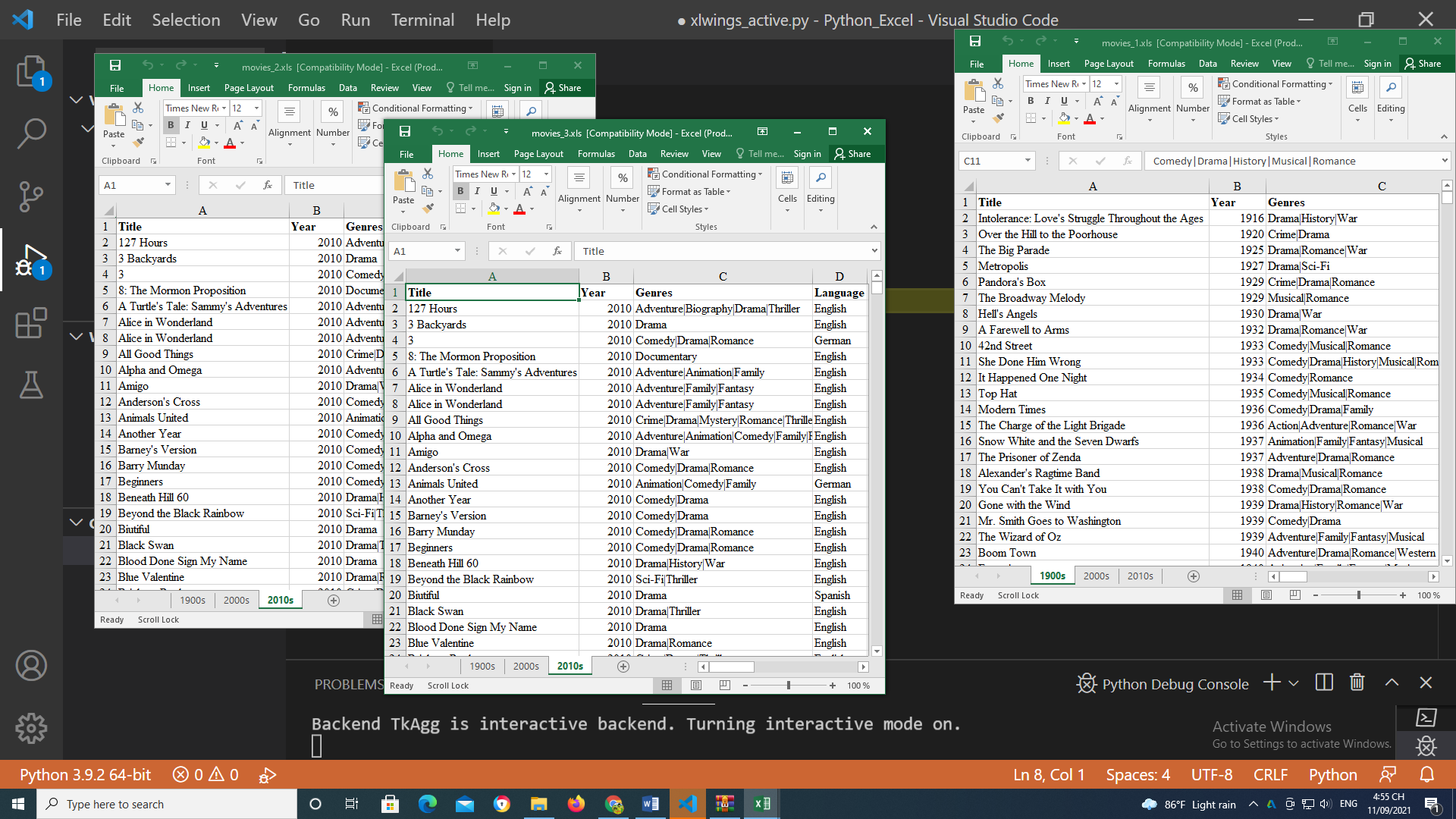
**Còn trong bảng tính excel thì sao?**

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



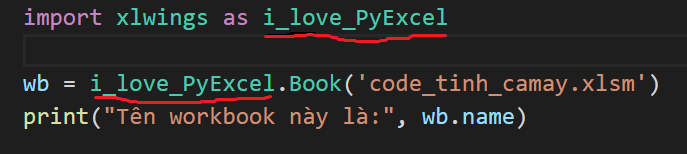
Tại terminal do đoạn code dòng thứ 13, thực hiện in ra tên của workbook đang được kích hoạt (hiện hành) là: “movies\_3”

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.

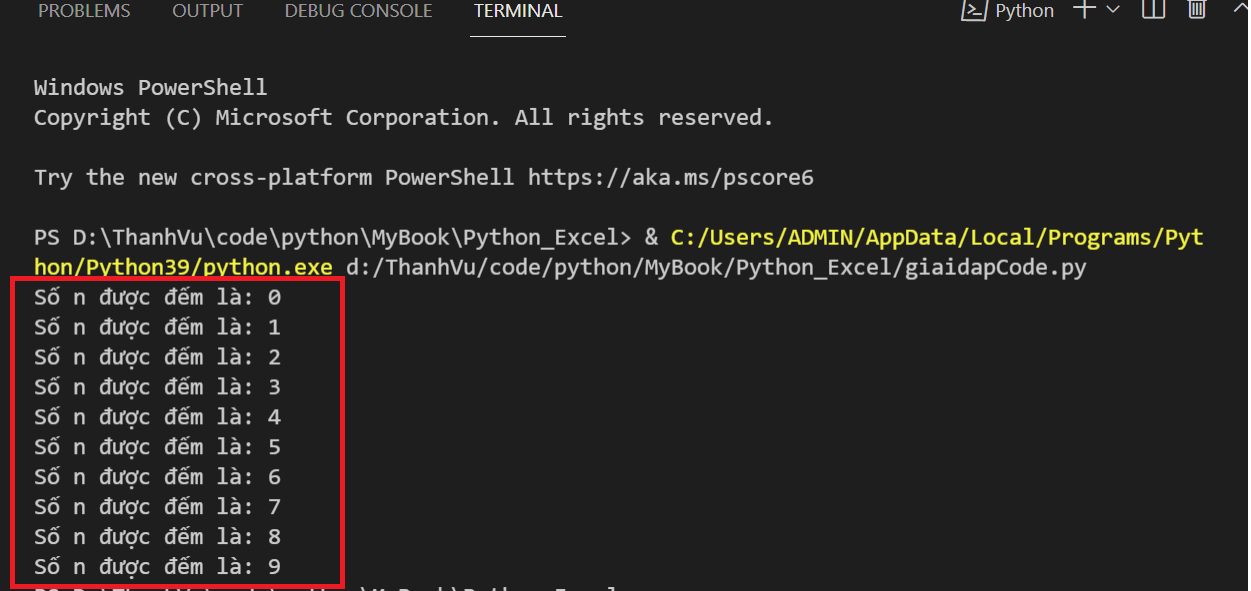
# Cho biến n là số đếm từ 0->10

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 (Đạon này mình có nói ở phần vòng lặp for trên phần python cơ bản trên)

**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>

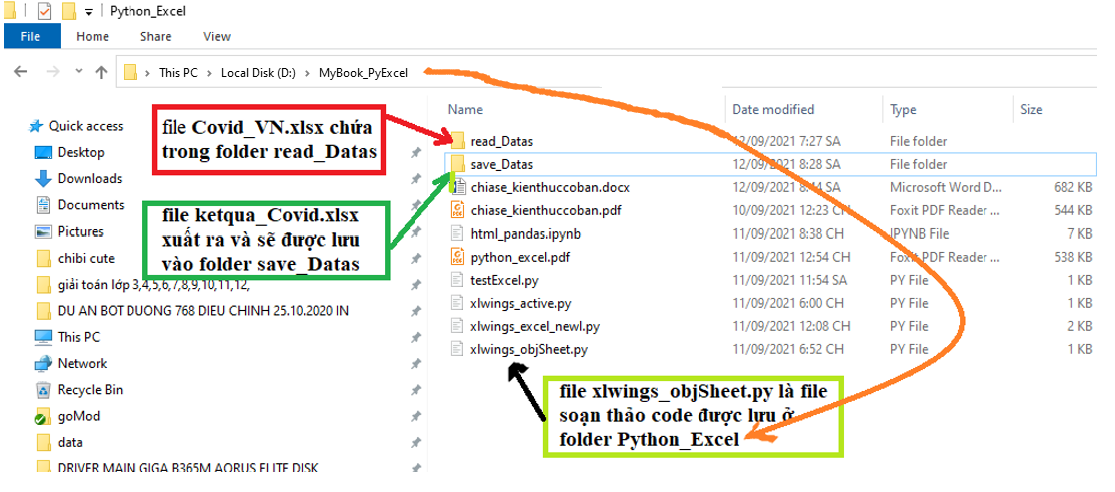
Để thuận tiện mình đã tải file này về trên máy mình và lưu nó tại ổ đĩa D, đường dẫn đây đủ là : **D:\** **MyBook\_PyExcel\** **read\_Datas\Covid\_VN.xlsx**

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

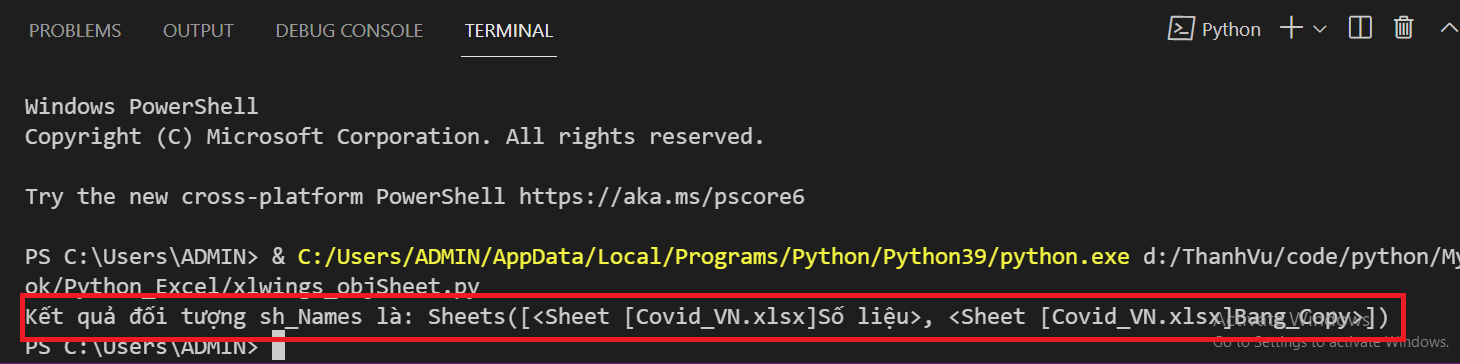
wb.close()

# 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()

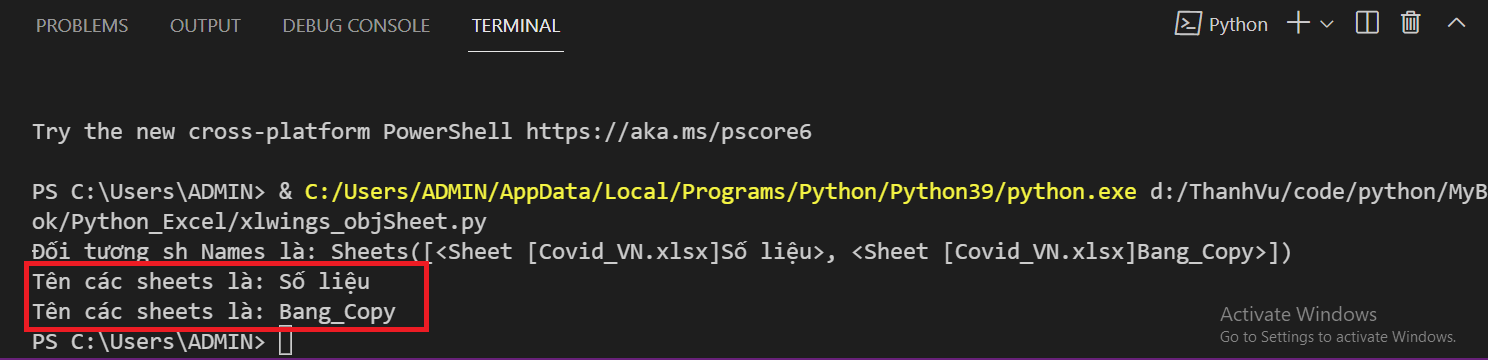
Một số chú ý dễ xảy ra lỗi khi dùng **vòng lặp for**:

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')

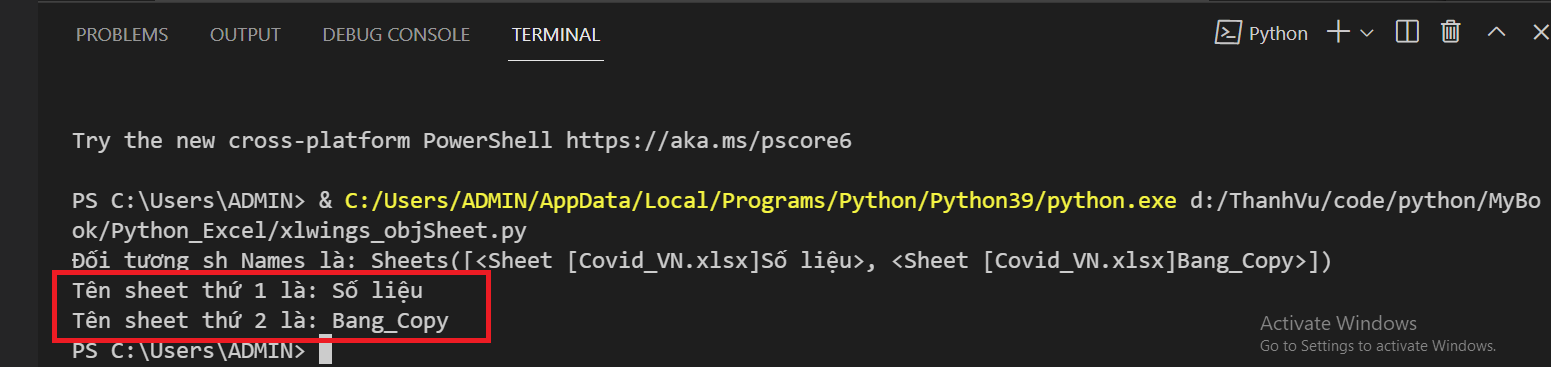
# Đoạn code này đóng file excel

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

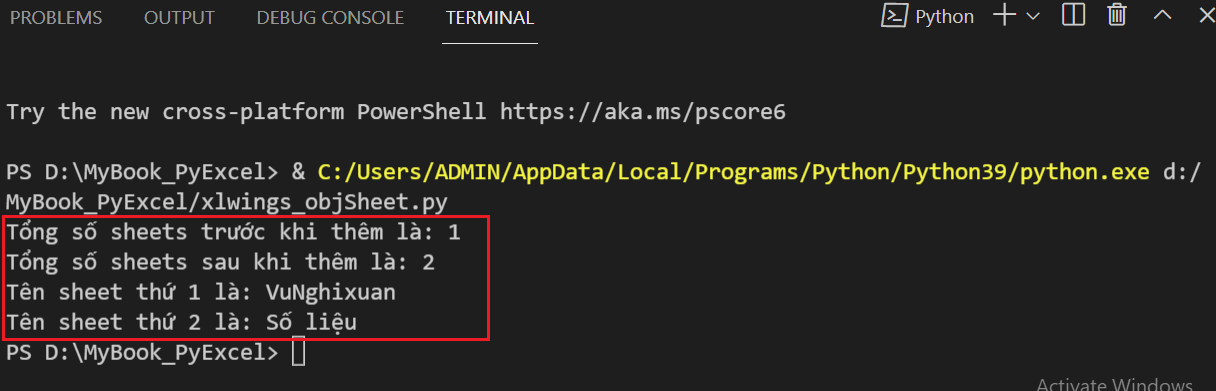
wb.close()

# 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>

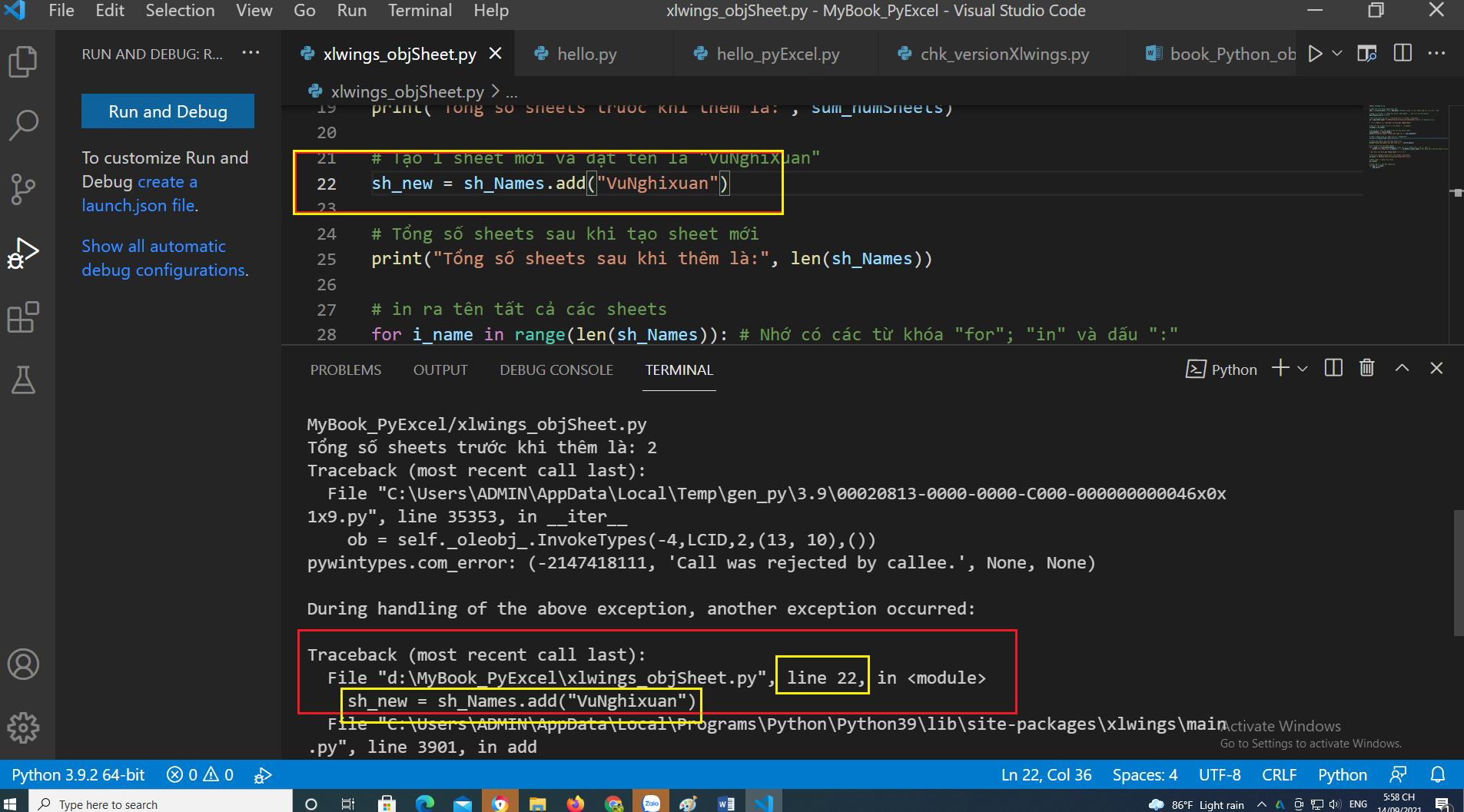
**sh\_new = sh\_Names.add("VuNghixuan")**

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

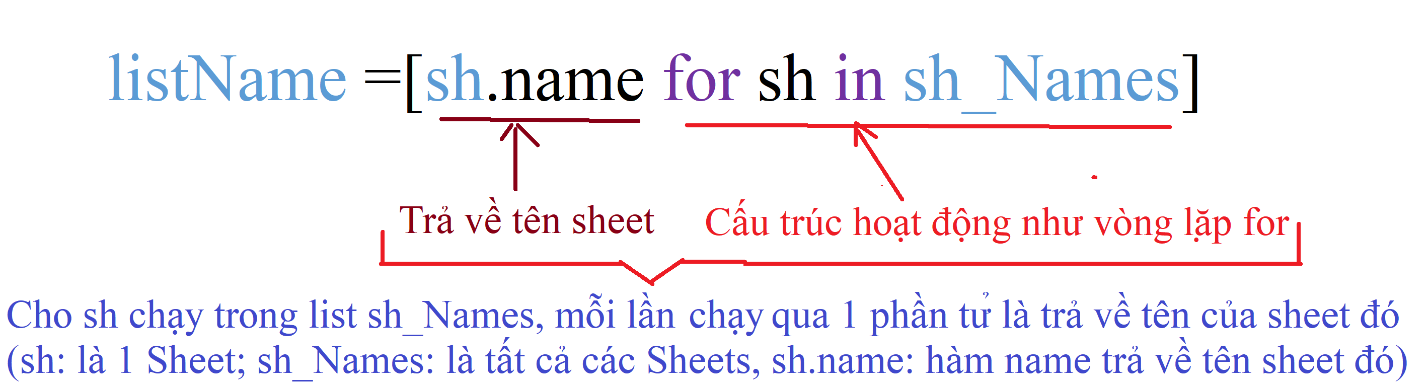
wb.close()

# Đoạn code này tắt đối tượng app

for app in xw.apps:

    app.quit()

Giải thích code mới:



**. Đổi tên tên sheets**

Ở trên bạn đã tạo ra 1 sheet mới có tên là **“VuNghixuan”**  bây giờ mình muốn đổi tên sheet này cho gọn hơn là **“NghiXuan”** thì bạn suy nghĩ tìm cách lấy chỉ số index của sheet đó (đoạn code ở dưới index chính là **i\_name**), sau đó dùng hàm **.name**  gán cho nó cái tên khác.

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

# Cho biến i\_name chạy trong list tổng số phần tử sh\_Names

for i\_name in range(len(sh\_Names)):

    # Nếu tại index của sh\_Names nào có tên là 'VuNghixuan' thì

    if sh\_Names[i\_name].name == 'VuNghixuan':

        # Thì đổi tên là "NghiXuan"

        sh\_Names[i\_name].name = "NghiXuan"

" 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

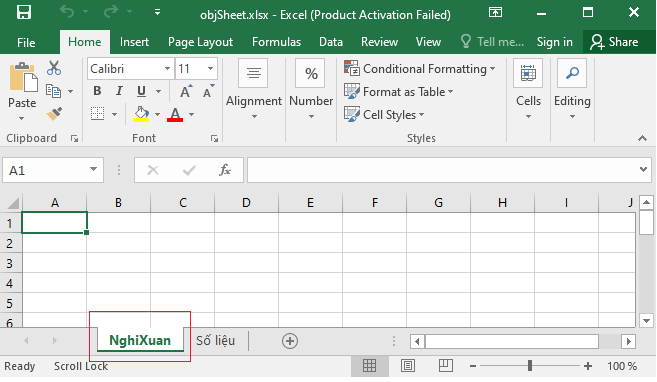
wb.close()

# Đoạn code này tắt đối tượng app

for app in xw.apps:

    app.quit()

Kết quả sau khi thực hiện code



Dựa vào ví dụ trên bạn có thể tự mình tạo ra đoạn code đổi tên hàng loạt sheet trong file excel. Đây cũng là 1 bài tập dành cho bạn để ôn lại kiến kiến phần này

1. **Đối tượng Range (Range là 1 cấu trúc gồm nhiều dòng và nhiều cột):**

Xlwings xem bảng tính Excel là 1 list trong python, có 2 chiều (Với Excel 2016 là: 1.048.576 Rows và 16.384 Columns), chính xác là 1 ma trận có: m (dòng) x n (cột) trong toán học:

Sau đây là các đoạn code thêm vào giá trị, dữ liệu cho bảng tính Excel

import xlwings as xw

from xlwings.main import Sheets

# Đoạn code này tắt các file Excel do trong quá trình code bị lỗi mà bạn quên tắt file

for app in xw.apps:

    app.quit()

# Khởi tạo file excel bằng modul App, chọn visible = False: Ko mở file

app = xw.App(visible = True, add\_book = True) # 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

# Đặt tên cho workbook vừa mới tạo là: wb,

wb = app.books[0] # Có thể dùng: wb = app.books['Book1']

# books[0]: Nghĩa là book đầu tiên

# Gán biến sh1 cho sheet đầu tiên

sh1= wb.sheets[0] # In ra tên sheet

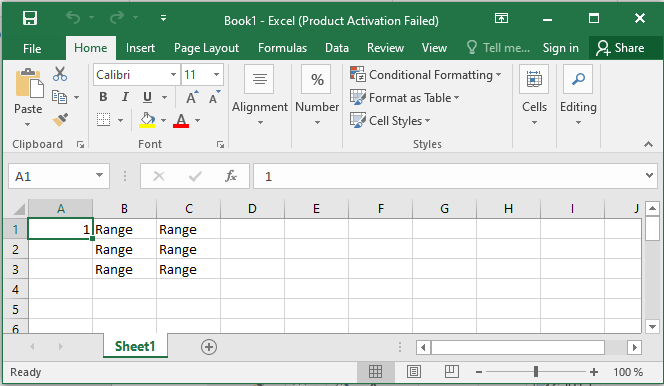
"Phần code tìm hiểu đối tượng Range"

# Cách thực hiện thêm giá trị, dữ liệu vào bảng tính

sh1.range('A1').value = 1 # Điền số 1 vào ô A1

sh1.range("A2:D5").value = "Range" # giá trị "A2:D5" = "Range"

Kết quả sau khi thực hiện code



Tiếp theo, để làm quen cách sắp xếp dữ liệu theo dòng và cột trong bảng tính Excel, bạn cần phân biệt chiều trong bảng tính Excel. Trước tiên bạn tìm hiểu cách hoạt động của 1 **list đơn** và 1 **list trong list**, qua đoạn code sau:

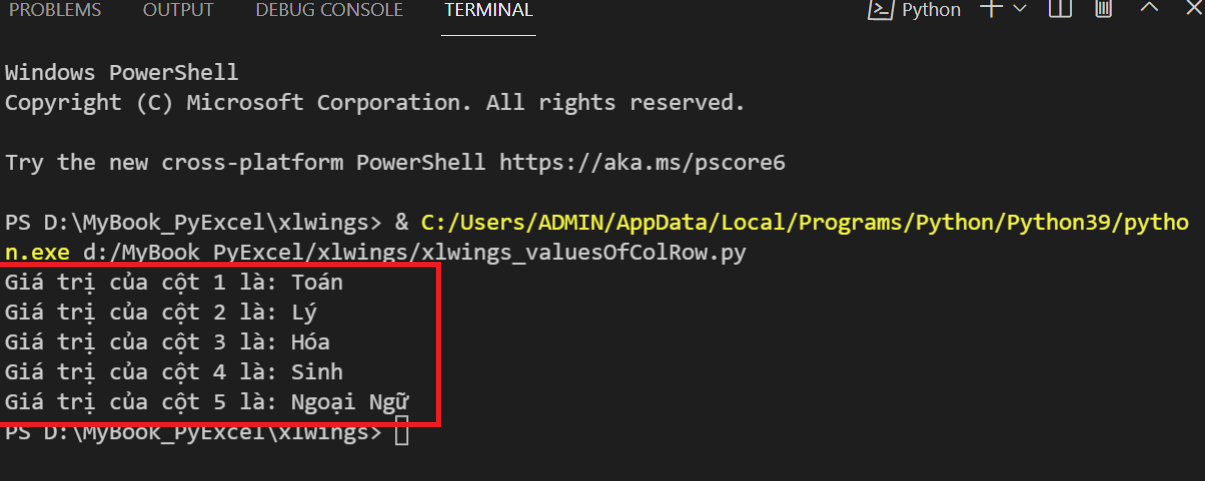
* **list đơn (List 1D):** Cách ghi giá trị từng cột trên 1 dòng bảng tính

columns = ["Toán","Lý", "Hóa", "Sinh", "Ngoại Ngữ"]

for i\_col in range(len(columns)):

    print(f'Giá trị của cột {i\_col+1} là: {columns[i\_col]} ')

Kết quả:



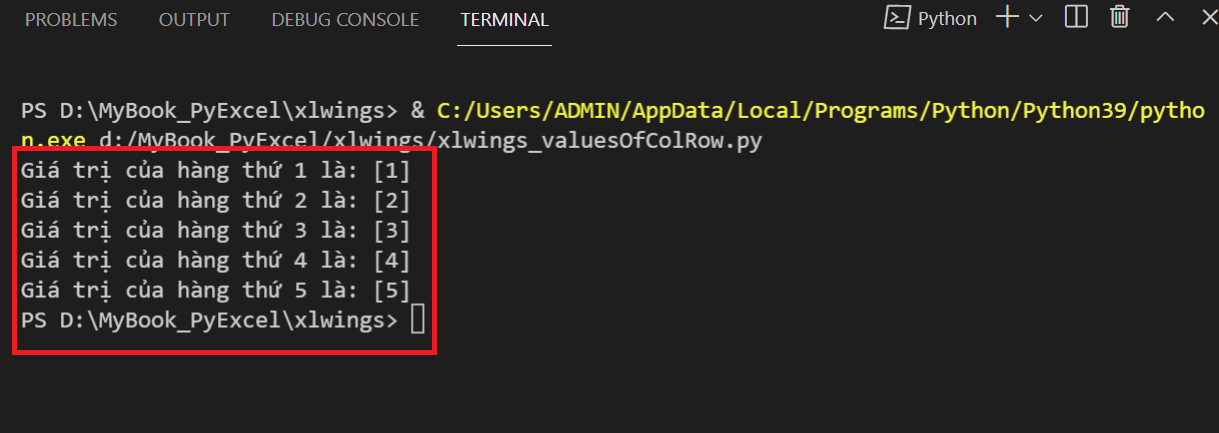
* **list trong list List 2D):** là cách ghi giá trị từng hàng trên 1 cột bảng tính

rows = [[1],[2],[3],[4],[5]]

for i\_row in range(len(rows)):

    print(f'Giá trị của hàng thứ {i\_row+1} là: {rows[i\_row]} ')

Kết quả:



Kết hợp kiểu list trong list ghi ra 1 data dữ liệu:

datas = [[1,5,8],

         [2,7,12],

         [3,9,5],

         [4, 5.9,8.2],

         [5.8, 105,102.5]]

total\_rows = len(datas)

total\_columns = len(datas[0]) # Tức là tổng số phần tử hàng đầu tiên

print(f'Ma trận Datas có {total\_rows}(dòng) và {total\_columns}(cột)\nCụ thể: ')

for i\_row in range(total\_rows):

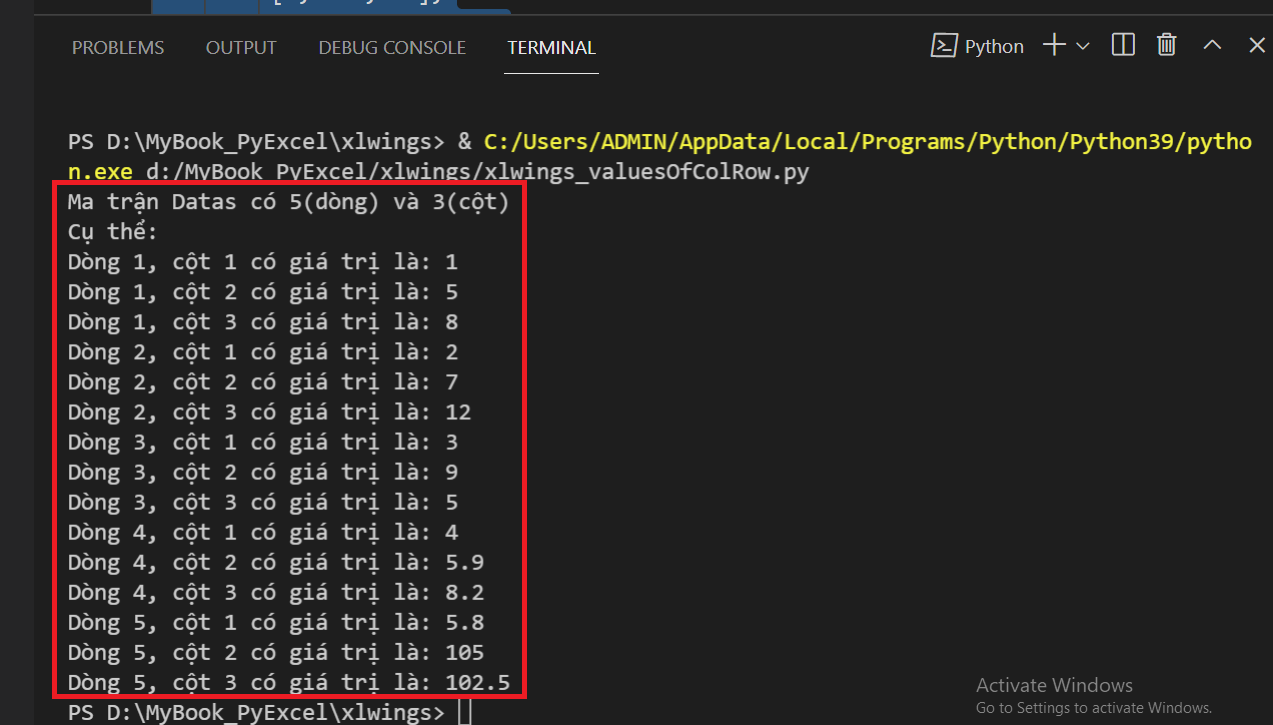
    for i\_col in range(total\_columns):

        print(f'Dòng {i\_row+1}, cột {i\_col+1} có giá trị là: {datas[i\_row][i\_col]}')

link code:

<https://github.com/VuNghiXuan/Mybooks/blob/fe7e03d40bc28139c55d7ac7395a6a545aee3df4/Python_Excel/xlwings_valuesOfColRow.py>

Kết quả:



Bây giờ hãy đưa hết các **list** và **list trong list** ghi ra bảng tính Excel

import xlwings as xw

from xlwings.main import Sheets

# Đoạn code này tắt các file Excel do trong quá trình code bị lỗi mà bạn quên tắt file

for app in xw.apps:

    app.quit()

# Khởi tạo file excel bằng modul App, chọn visible = False: Ko mở file

app = xw.App(visible = True, add\_book = True) # 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

# Đặt tên cho workbook vừa mới tạo là: wb,

wb = app.books[0] # Có thể dùng: wb = app.books['Book1']

# books[0]: Nghĩa là book đầu tiên

# Gán biến sh1 cho sheet đầu tiên

sh1= wb.sheets[0] # In ra tên sheet

"Phần code tìm hiểu đối tượng Range"

columns = ["Toán","Lý", "Hóa", "Sinh", "Ngoại Ngữ"]

rows = [[1],[2],[3],[4],[5]]

datas = [[1,5,8],

         [2,7,12],

         [3,9,5],

         [4, 5.9,8.2],

         [5.8, 105,102.5]]

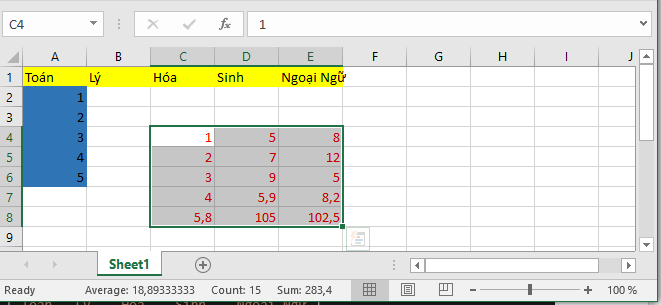
# Cách thực hiện thêm giá trị, dữ liệu vào bảng tính

sh1.range('A1').value = columns # Điền số 1 vào ô A1

sh1.range('A2').value = rows

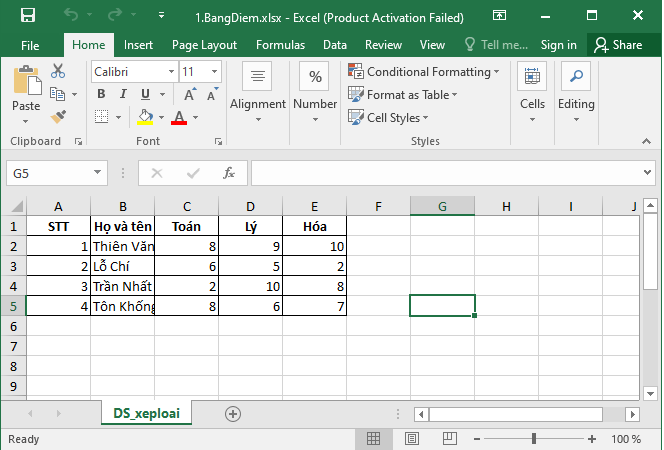
sh1.range("C4").value = datas

Xem giá trị ghi ra bảng tính Excel:



* **Đọc dữ liệu từ bảng tính Ecxel:**

Bạn hãy tạo 1 file Excel chứa các điểm thi các môn như sau:

****

Bạn tham khảo đoạn code dưới đây để lấy dữ liệu từ bảng tính có tên “1.BangDiem.xlsx”

import xlwings as xw

from xlwings.main import Sheets

# Đoạn code này tắt các file Excel do trong quá trình code bị lỗi mà bạn quên tắt file

for app in xw.apps:

    app.quit()

# Đọc dữ liệu từ bảng tính Excel

wb = xw.Book(r'D:\MyBook\_PyExcel\xlwings\read\_Datas\1.BangDiem.xlsx')

sh = wb.sheets[0]

# Cách 1: Đọc dữ liệu bằng địa chỉ

datas\_1 = sh.range("A1:E5").value

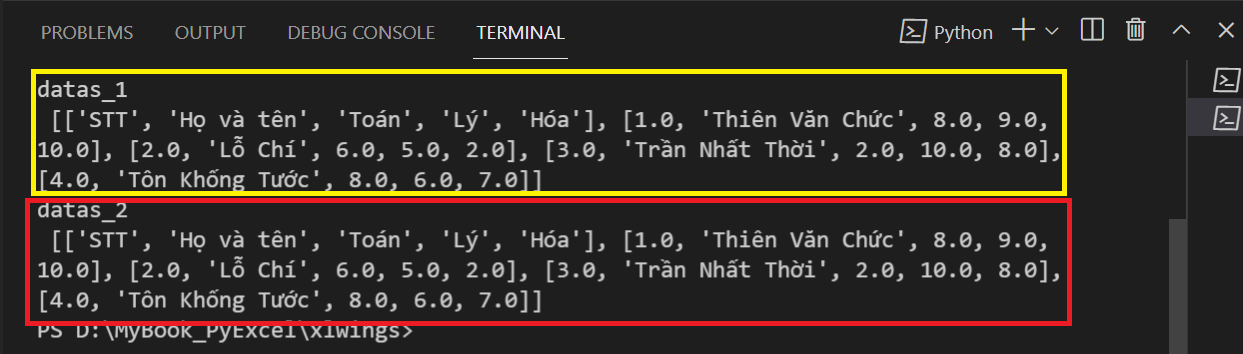
# Cách 2: Đọc dữ liệu từ ô A1 --> sử dụng phương thức expand mở rộng phạm vi

datas\_2 = sh.range("A1").expand().value

print("datas\_1\n", datas\_1)

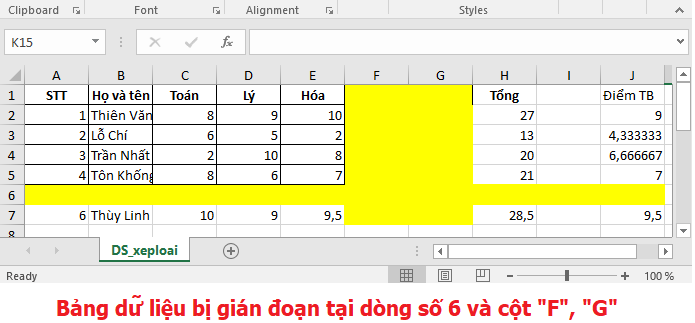
print("datas\_2\n", datas\_2)

Sau khi in ra cửa sổ Terminal cho thấy Datas\_1 và Datas\_2 đều cho kết quả giống nhau. Do đó, tùy theo đặc tính và yêu cầu bạn có thể vận dụng sao cho hợp lý.



* **Cách lấy dữ liệu giới hạn dòng cuối, cột cuối cùng của bảng tính**

Giả sử dữ liệu bảng tính của mình không liên tục mà bị gián đoạn giữa các dòng trống, và cột trống như hình.



Bạn tải file data gián đoạn tại địa chỉ sau:

<https://github.com/VuNghiXuan/Mybooks/blob/main/read_Datas/1.BangDiem.xlsx>

Nếu ta lấy dữ liệu theo phương thức **expand()** mở rộng thì dữ liệu của bạn sẽ bị thiếu các cột cuối và dòng cuối.

import xlwings as xw

from xlwings.main import Sheets

# Đoạn code này tắt các file Excel do trong quá trình code bị lỗi mà bạn quên tắt file

for app in xw.apps:

    app.quit()

# Đọc dữ liệu từ bảng tính Excel

wb = xw.Book(r'D:\MyBook\_PyExcel\xlwings\read\_Datas\1.BangDiem.xlsx')

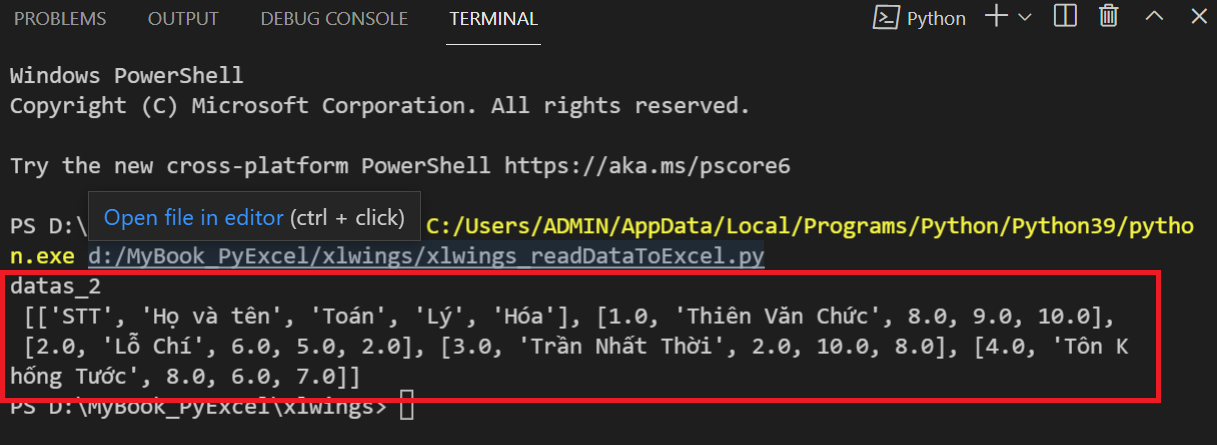
sh = wb.sheets[0]

# Cách 2: Đọc dữ liệu từ ô A1 --> sử dụng phương thức expand mở rộng phạm vi

datas\_2 = sh.range("A1").expand().value

print("datas\_2\n", datas\_2)

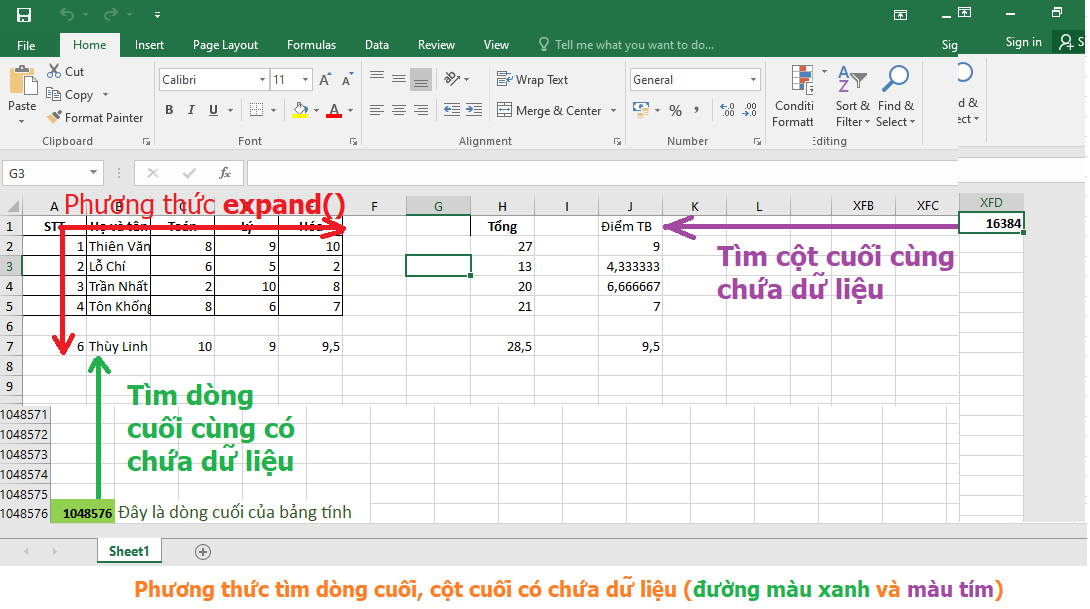
Kết quả Terminal

****

Quan sát dữ liệu phần các cột **(Tổng cộng, Điểm TB)** và dòng học sinh có tên **“Thùy Linh”** không nhận được phần dữ liệu này. Lý do, khi sử dụng phương thức **expand()**, phần dữ liệu lân cận **bị gián đoạn** ở các cột **“F”, “G”, “I”** và dòng số **“6”.**

Để khắc phục vấn đề này cũng như linh hoạt trong việc lấy dữ liệu, không bỏ xót các cột và dòng dữ liệu nằm sau phần bị gián đoạn, chúng ta đi tìm đi tìm dòng cuối và cột cuối cùng có chứa dữ liệu như sau:

Ở đoạn code **“**datas\_2 = sh.range(‘**A1**’).expand().value **“**, phía trên phương thức **expand()**,sẽ mở rộng vùng dữ liệu được ấy từ ô A1 (là ô ta định vị đầu tiên), **đi từ trên xuống dưới và từ trái qua phải**, sau đó dừng lại vùng dữ liệu bị gián đoạn.



Ngược lại, phương thức đi tìm dòng cuối dữ liệu ta bắt đầu từ dòng cuối cùng là **dòng 1048576** ta đi ngược lên trên và dừng lại dòng có chứa dữ liệu là **dòng số 6.** Còn đi tìm cột cuối cùng có chứa dữ liệu ta đi từ **cột 16384** đi từ phải sang trái và dừng lại ở dòng cuối cùng bên trái có chứa dữ liệu là **cột J**.

import xlwings as xw

from xlwings.main import Sheets

# Đoạn code này tắt các file Excel do trong quá trình code bị lỗi mà bạn quên tắt file

for app in xw.apps:

    app.quit()

# Đọc dữ liệu từ bảng tính Excel

wb = xw.Book(r'D:\MyBook\_PyExcel\xlwings\read\_Datas\1.BangDiem.xlsx')

sh = wb.sheets[0]

# Sử dụng hàm có sẵn last\_cell.row để tìm dòng cuối cùng của bảng tính

lr\_table = sh.cells.last\_cell.row

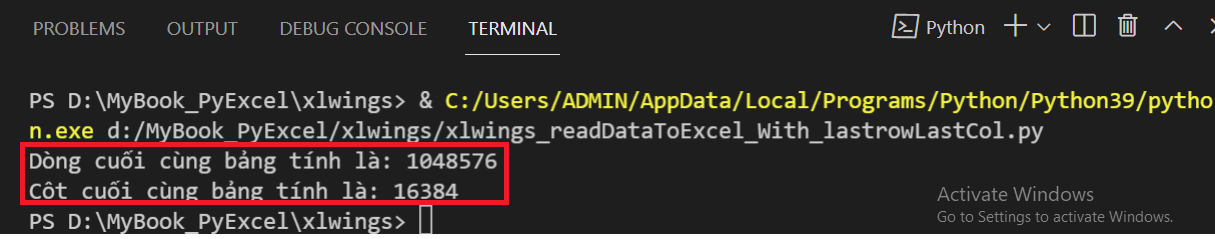
print('Dòng cuối cùng bảng tính là:', lr\_table)

# Sử dụng hàm có sẵn last\_cell.column để tìm cột cuối cùng của bảng tính

lc\_table = sh.cells.last\_cell.column

print('Cột cuối cùng bảng tính là:', lc\_table)

Kết quả chạy code:



Chú ý: Kết quả trên là dòng cuối, cột cuối của bảng tình execl, chứ chưa phải là dòng cuối dữ liệu.

Bước cuối cùng ta thêm đoạn code để đi tìm dòng cuối, cột cuối và bảng có chứa đầy đủ dữ liệu (bao gồm cả phạm vi phần gián đoạn) như sau:

import xlwings as xw

from xlwings.main import Sheets

# Đoạn code này tắt các file Excel do trong quá trình code bị lỗi mà bạn quên tắt file

for app in xw.apps:

    app.quit()

# Đọc dữ liệu từ bảng tính Excel

wb = xw.Book(r'D:\MyBook\_PyExcel\xlwings\read\_Datas\1.BangDiem.xlsx')

sh = wb.sheets[0]

# Sử dụng hàm có sẵn last\_cell.row để tìm dòng cuối cùng của bảng tính

lr\_table = sh.cells.last\_cell.row

print('Dòng cuối cùng bảng tính là:', lr\_table)

# Sử dụng hàm có sẵn last\_cell.column để tìm cột cuối cùng của bảng tính

lc\_table = sh.cells.last\_cell.column

print('Cột cuối cùng bảng tính là:', lc\_table)

" Thêm code để tìm dòng cuối, cột cuối có chứa dữ liệu";

lr\_data = sh.range('A'+ str(lr\_table)).end('up').row

print('Dòng cuối cùng có chứa dữ liệu là:', lr\_data)

# lcol = sh.range(row\_index, col).end("left").column

lc\_data = sh.range(1, lc\_table).end('left').column

print('Cột cuối cùng có chứa dữ liệu là:', lc\_data)

# Cuối cùng ta có bảng dữ liệu cần thu thập như sau:

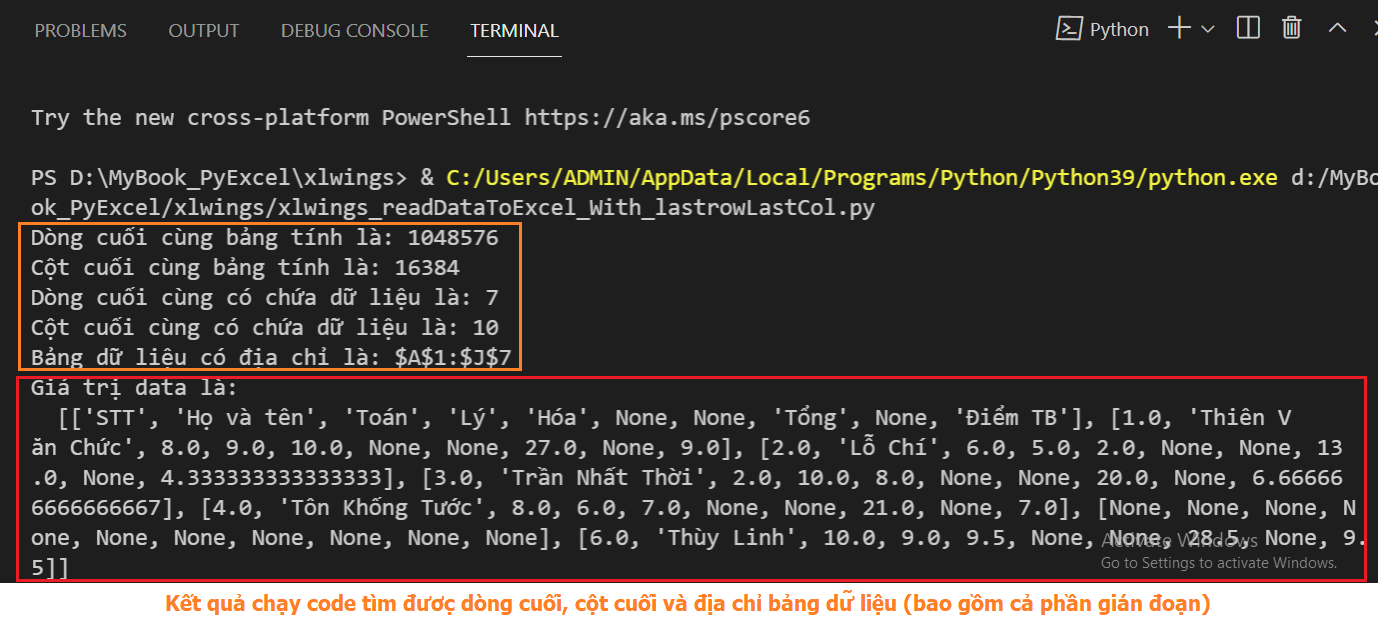
table\_datas = sh.range((1,1), (lr\_data,lc\_data))

print("Bảng dữ liệu có địa chỉ là:", table\_datas.address)

print("Giá trị data là:\n ",table\_datas.value)

linkcode: <https://github.com/VuNghiXuan/Mybooks/blob/main/Python_Excel/xlwings_readDataToExcel_With_lastrowLastCol.py>

Kết quả chạy code:



1. **Xử lý dữ liệu trong Sheet**
2. **Xử lý data**