

2021

PYTHON EXCEL

TÀI LIỆU TỰ HỌC (PHẦN CƠ BẢN)
(ĐẶNG THANH VŨ)

GIỚI THIỆU	0
GIẢI THÍCH TỪ NGỮ	1
CHƯƠNG 1: HƯỚNG DẪN CÀI ĐẶT PYTHON, VISUAL STUDIO CODE, XLWINGS	2
I. CÀI ĐẶT PYTHON	2
II. CÀI ĐẶT VISUAL STUDIO CODE	6
1. Cài Visual Studio code (VSC)	6
2. Cài đặt Extension:	8
3. Hướng dẫn soạn thảo Python trên Visual Studio cơ bản	9
III. CÀI ĐẶT THƯ VIỆN XLWINGS	12
CHƯƠNG 2: LÀM QUEN VỚI NGÔN NGỮ PYTHON.....	15
1. Python lists	15
2. Vòng lặp for trong list python.....	18
3. Vòng lặp While python	24
4. Hàm def trong python	25
5. Đối tượng Class Python	27
6. Đối tượng kế thừa Class Python	30
CHƯƠNG 3: THƯ VIỆN XLWINGS	32
I. ĐỐI TƯỢNG XLWINGS	32
II. ĐỐI TƯỢNG WORKBOOK (BẢNG TÍNH EXCEL)	34
1. Mở Workbook thông qua lớp ứng dụng xlwings.App	34
2. Mở 1 Workbook mới	35
3. Mở workbook bằng đường dẫn đầy đủ	36
4. Mở Workbook bằng xw.Book	38
5. Lưu, đóng Workbook và thoát App	44
6. Phản ôn tập đối tượng xlwings.....	46
7. Phản giải thích thêm về thuộc tính active	47
8. Ôn tập đối tượng workbook.....	49
9. Bài tập phản đối tượng workbook:.....	51
10. Đối tượng Sheet	53
CHƯƠNG 4: CÀI ĐẶT ADDIN SỬ DỤNG TRỰC TIẾP PYTHON TRÊN EXCEL	103

1. Kích hoạt quyền truy cập đối với VBA	103
2. Cách cài đặt AddIn.....	104
3. Cách kết nối folder AddIn.....	104
CHƯƠNG 5: ỦNG DỤNG PHỤC VỤ CÔNG VIỆC:	119
VIẾT ỦNG DỤNG TỔNG HỢP DỮ LIỆU.....	119
1. Tổng dữ liệu các sheets trên cùng 1 file Excel	119
2. Tổng hợp dữ liệu nhiều files Excel, mỗi file có nhiều sheets	121
CHƯƠNG 6: GIAO DIỆN ĐỒ HỌA VỚI TKINTER	125
I. GIAO DIỆN ĐỒ HỌA TKINTER	125
II. GIAO DIỆN ĐỒ HỌA PyQT5:	126
CHƯƠNG 7: XUẤT ỦNG DỤNG	140

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 cù thê 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 chưa hiểu có thể tham gia thảo luận cùng với group tại đây: <https://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 gửi lời cảm ơn đến:

- + Tất cả bạn đọc.
- + Các bạn tham gia trong Group cùng những lời động viên tinh thần, góp ý tích cực trong suốt quá trình chỉnh sửa và hoàn thiện.
- + Nhà biên tập: **Nguyễn Thế Anh** (nguyên Phó Tổng biên tập của Tạp chí Kinh tế xây dựng).

Mọi thông tin góp ý xin liên hệ <https://vunghixuan.github.io>

GIẢI THÍCH TỪ NGỮ

- **Python:** là ngôn ngữ lập trình, chức năng dùng để biên dịch mã code. Như chúng ta biết, máy tính chỉ đọc được 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.

- **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)

- **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,...

- **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

I. CÀI ĐẶT PYTHON

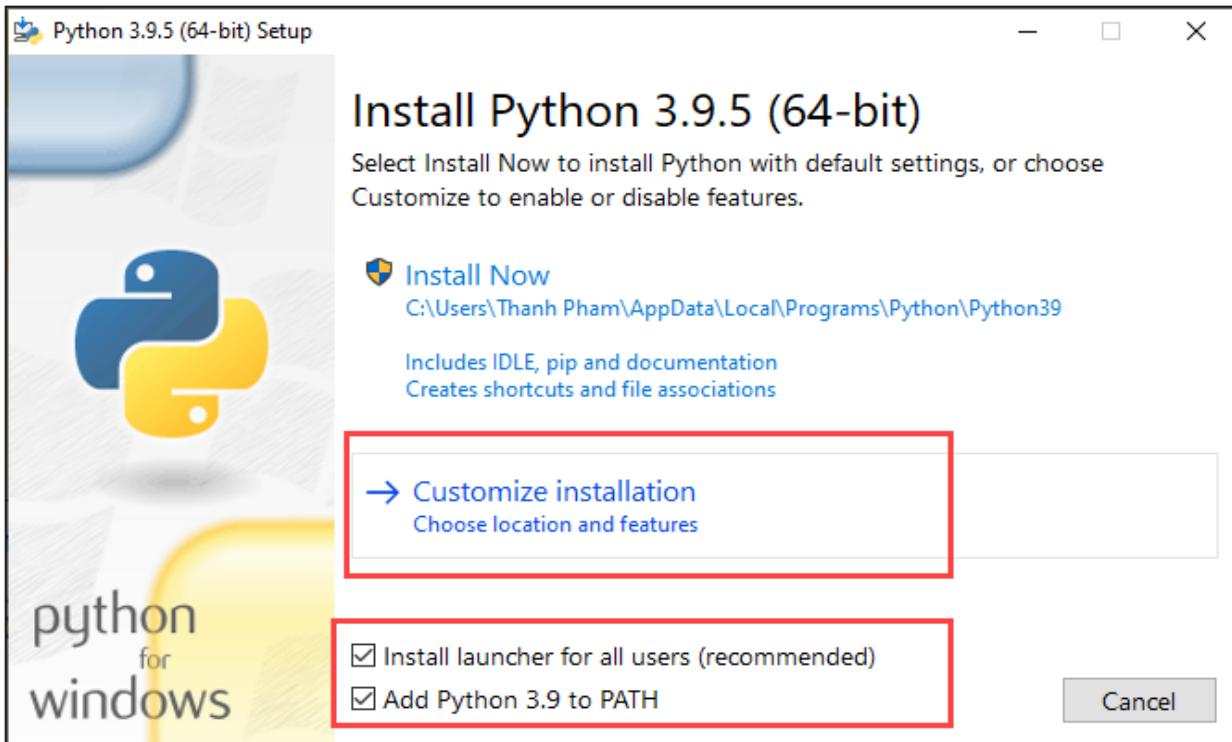
Truy cập vào trang: <https://python.org/downloads> để cài Python cho hệ điều hành bạn đang sử dụng.

Đây là phiên bản mới nhất của Python tại thời điểm mình cập nhật, khi các bạn đọc tài liệu này có thể Python đã cập nhật phiên bản mới hơn chặng hạn “python-3.9.7”.

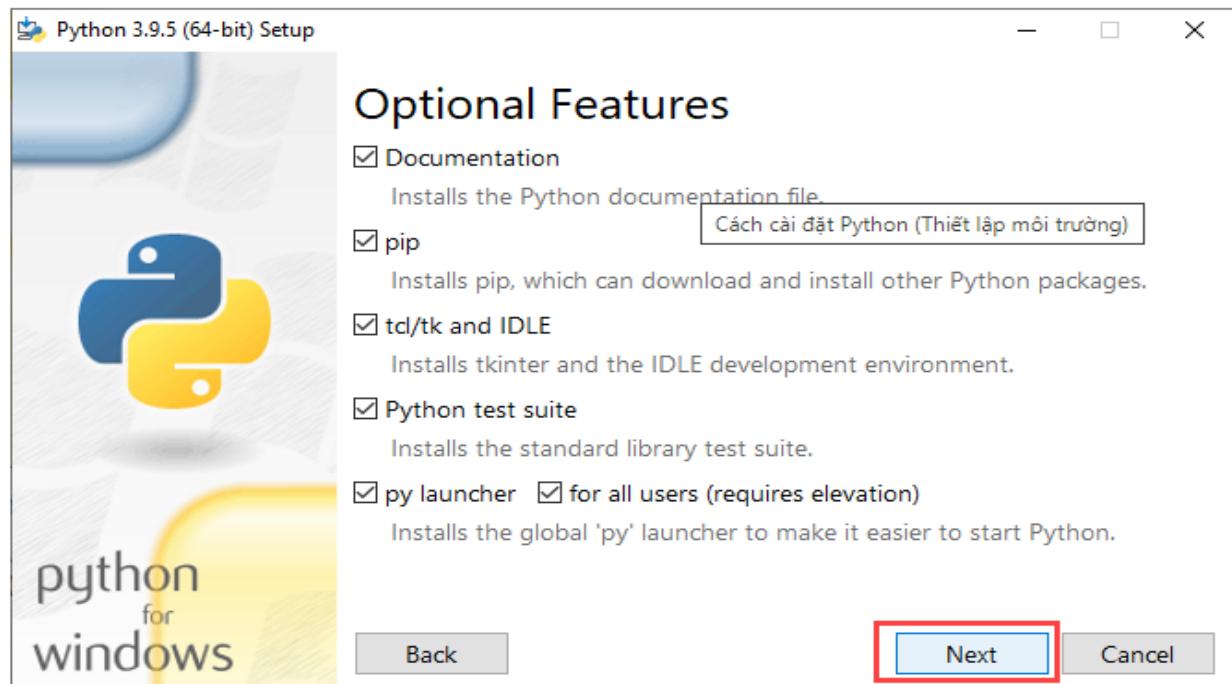


Click đúp chuột vào file vừa tải xuống (python-3.9.5-amd64.exe) để tiến hành cài đặt Python: Phần màu đỏ cho windows, các hệ điều hành khác là màu xanh.

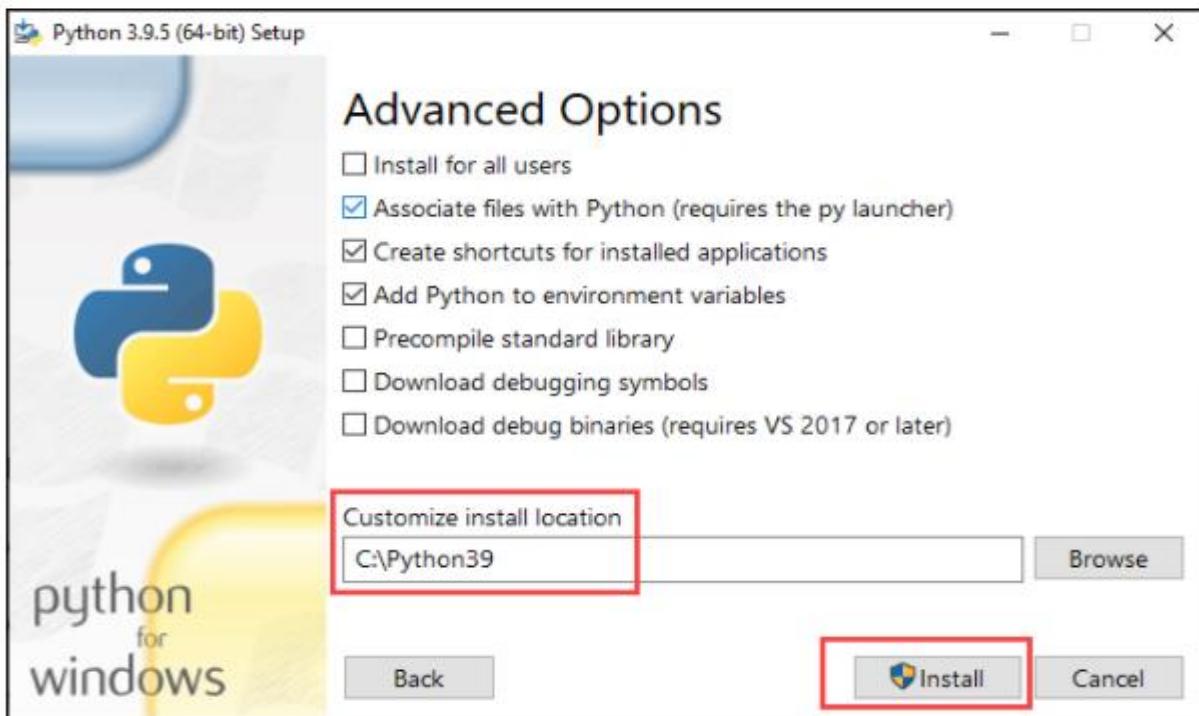
Tiếp theo Chọn **Customize installation**:



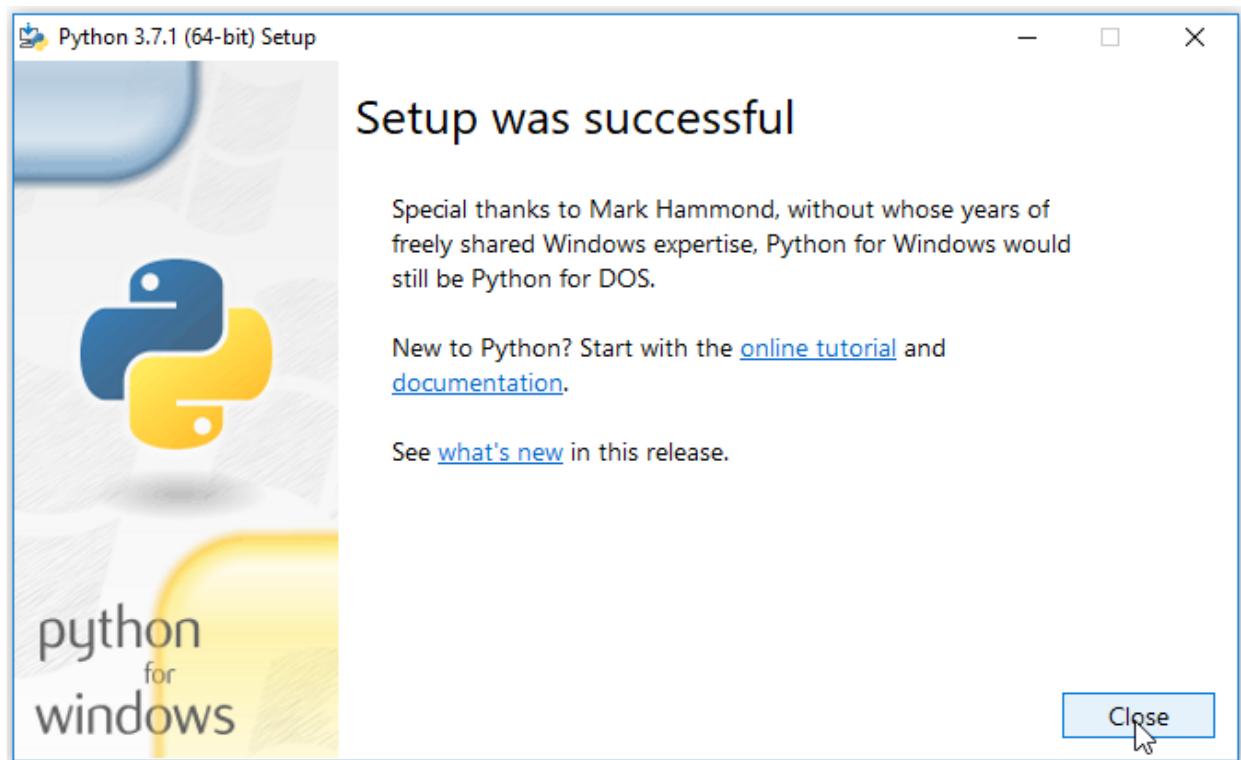
Chọn tất cả các tính năng tùy chọn, click Next:



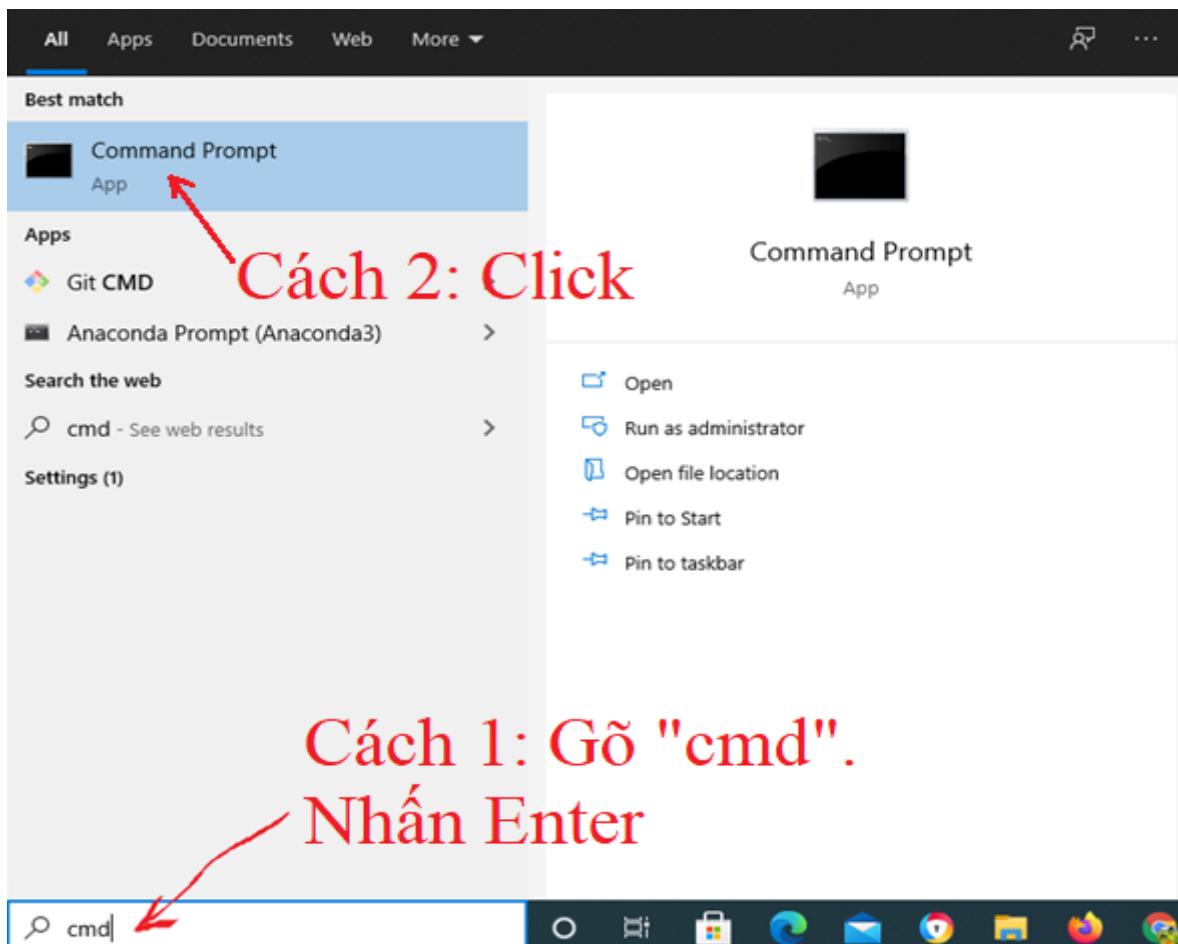
Chọn các tính năng tùy chọn như trong hình, click **Install**:



Nếu bạn muốn để mặc định như mình thì không phải chọn button **Browse**, nhấp luôn vào **Install**. Kết quả sau khi cài đặt Python:



Để kiểm tra việc cài thành công hay không bạn mở 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

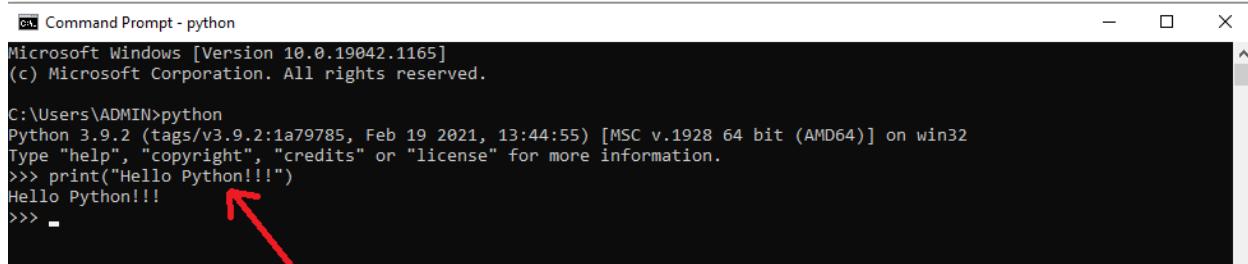


Sau khi **Command Prompt** xuất hiện, bạn gõ “python”. Kết quả như sau:

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 tỏ cho ngôn ngữ Python: `print("Hello Python!!!")`



```
Command Prompt - python
Microsoft Windows [Version 10.0.19042.1165]
(c) Microsoft Corporation. All rights reserved.

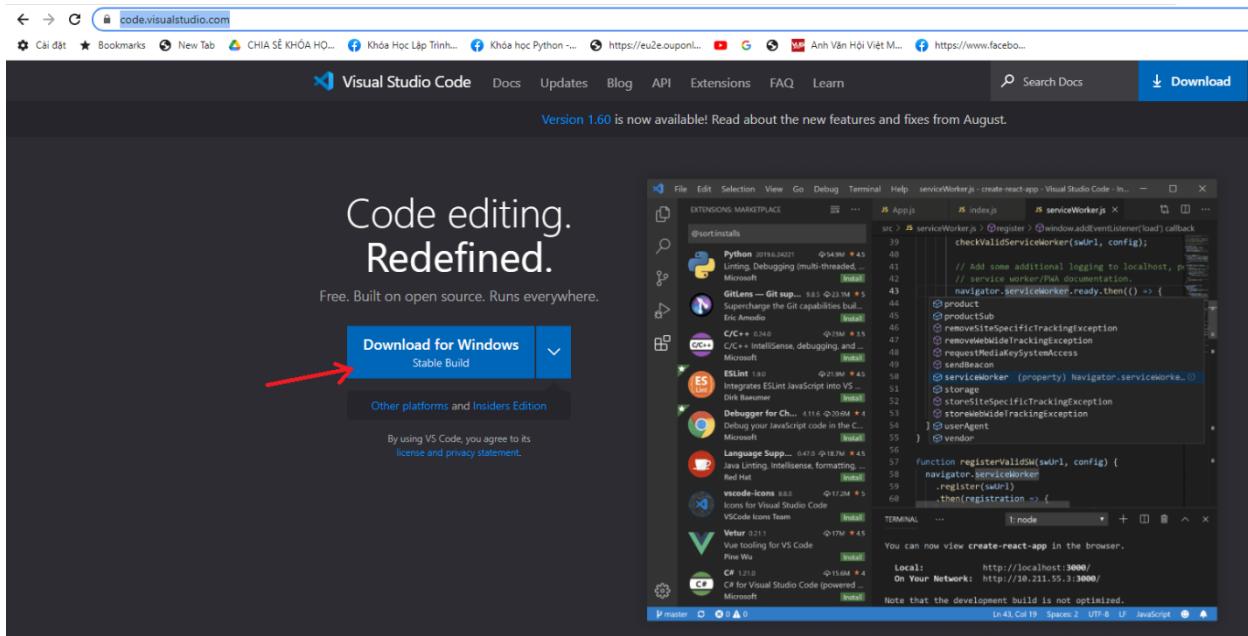
C:\Users\ADMIN>python
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello Python!!!")
Hello Python!!!
>>> -
```

II. CÀI ĐẶT VISUAL STUDIO CODE

1. Cài Visual Studio code (VSC)

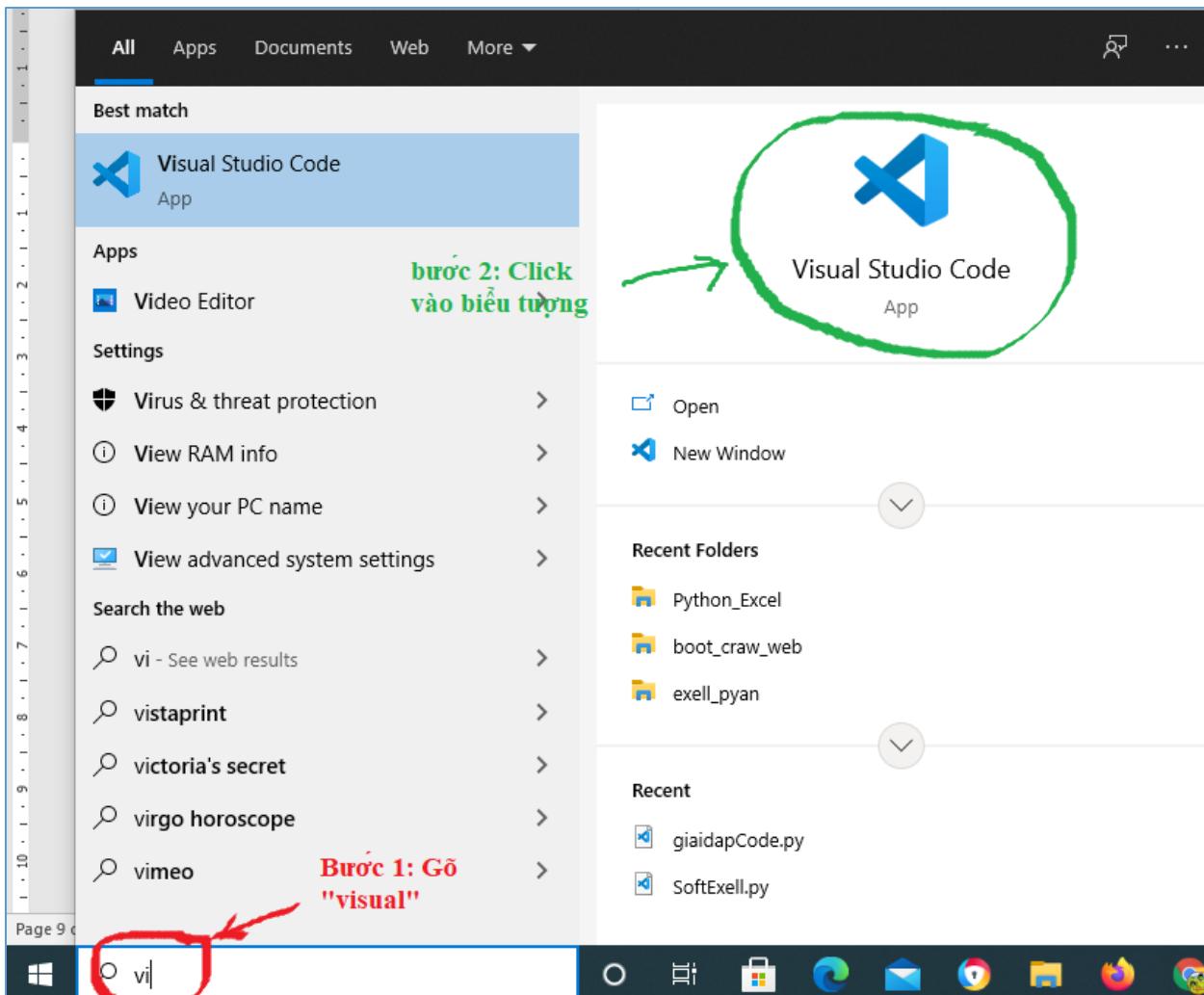
Visual Studio Code là bản soạn thảo mã code (tương tự như bạn dùng Word để soạn thảo văn bản) và chỉnh sửa code (ngắn gọn là biên tập code, còn gọi là IDE VSC). VSC có ưu điểm nhẹ, tiện lợi và phổ biến.

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> để tải bộ cài. 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.

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.

2. 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).



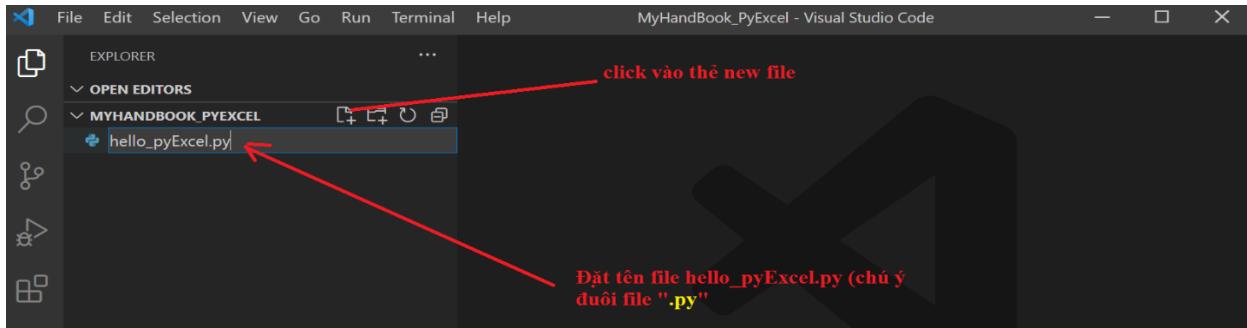
Ngoài ra, VSC còn tích hợp nhiều ứng dụng giúp cho quá trình soạn thảo đẹp hơn, nhanh chóng và tiện ích (có thể google để tìm ra những thứ cần thiết cho chính bạn)

3. 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 gõ thêm đuôi “.py”, nếu không gõ VSC sẽ hiểu là file txt. Nếu bạn trót quên gõ đuôi .py cho file thì bạn có thể đổi lại tên và thêm đuôi vào cho file nhé).

❖ *Tạo File script python*

- **Bước 1:** Tạo file **hello_pyExcel.py**, chú ý đuôi file là “.py”

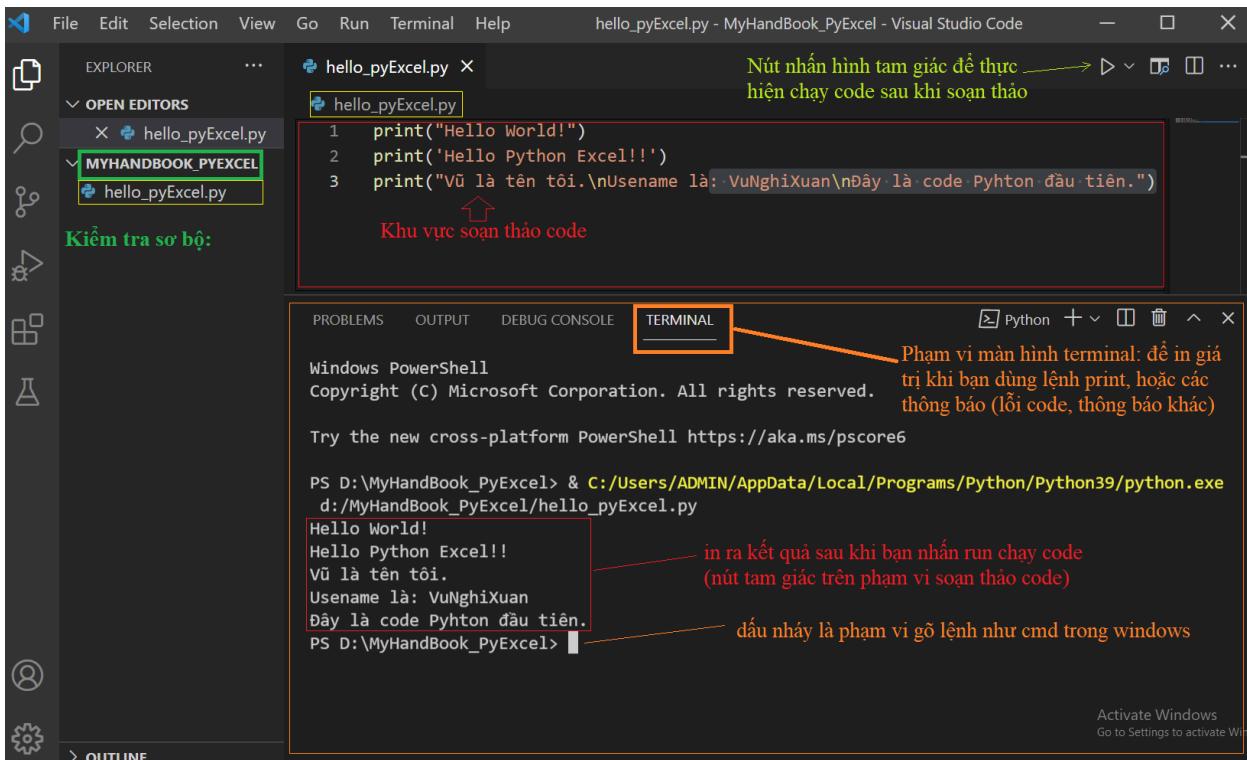


- **Bước 2.** Code. Sau khi tạo file như trên, bạn sẽ thấy màn hình soạn thảo Visual Studio Code (VSC), bạn hãy gõ code vào như hình dưới:

```
print("Hello world!")
```

```
print('Hello Python Excel!!')
```

print("Vũ là tên tôi. \nUsername là: VuNghiXuan\nĐây là code Python đầu tiên.")



Bạn có thể tải file code ở đường link sau:
<https://github.com/VuNghiXuan/giaotrinhPythonExcel/tree/main/C1>

- **Bước 3.** Chạy thử. Bạn gõ python **hello_pyExcel.py** ở phần Terminal

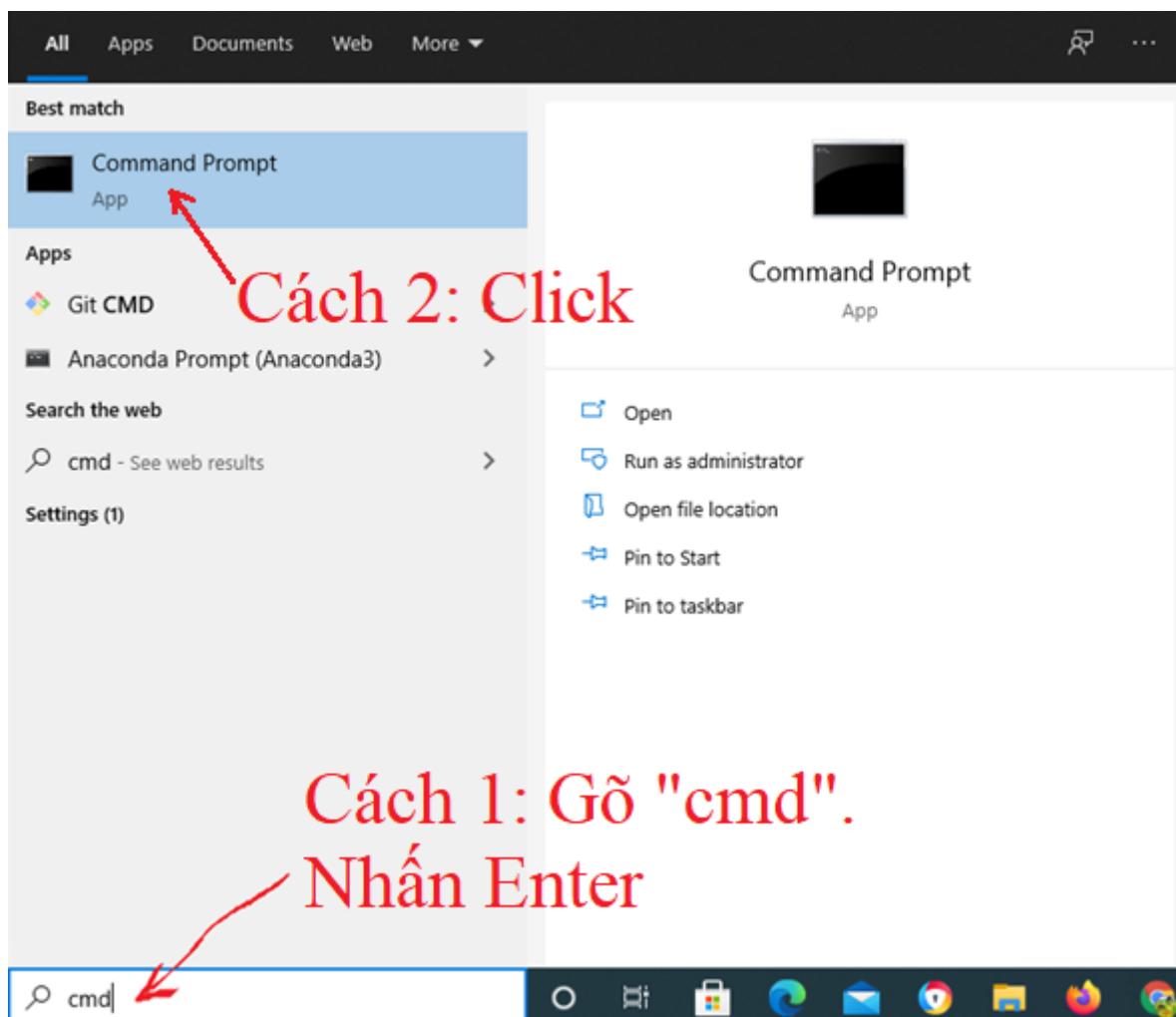
❖ *Giải thích code:*

	Dấu ngoặc đơn mở báo hiệu bắt đầu thân hàm
	Dấu ngoặc đơn đóng báo hiệu kết thúc hàm
print("Hello Word!")	
Hàm trong python	Tất cả phần nằm trong 2 dấu ngoặc đơn(): được gọi là tham số
	Trong đó : "Hello Word!" là tham số kiểu String

- + 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*).

III. CÀI ĐẶT THU VIỆN Xlwings

Khởi động **Command Prompt** bằng cách gõ “cmd” nhấn **Enter** tại hộp text “Type here to search”, nằm sát góc trái màn hình Windows (**nếu bạn 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**

```
Microsoft Windows [Version 10.0.19042.1165]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ADMIN>pip install xlwings
Requirement already satisfied: xlwings in c:\users\admin\appdata\local\programs\python\python39\lib\site-packages (0.23.0)
Requirement already satisfied: pywin32>=224 in c:\users\admin\appdata\local\programs\python\python39\lib\site-packages (from xlwings) (300)
WARNING: You are using pip version 21.1.2; however, version 21.2.4 is available.
You should consider upgrading via the 'c:\users\admin\appdata\local\programs\python\python39\python.exe -m pip install --upgrade pip' command.

C:\Users\ADMIN>
```

Vì trên máy mình đã cài đặt thư viện xlwings rồi nên chỉ có dòng cảnh báo màu vàng này chứ không phải là cài đặt lỗi

Đây cũng là 1 chú ý quan trọng cho các bạn khi cài đặt thư viện (nếu bạn muốn update thì google cách update)

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 “Successfully installed ...”, (như hình dưới)

```
Successfully installed comtypes-1.1.7 pywin32-227 xlwings-0.17.0

C:\WINDOWS\system32>pip list
Package      Version
-----  -----
comtypes    1.1.7
pip        19.3.1
pywin32     227
setuptools  41.2.0
xlwings    0.17.0
```

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

```

C:\Users\ADMIN>pip install xlwings
Requirement already satisfied: xlwings in c:\users\admin\appdata\local\programs\python\python39\lib\site-packages (0.23.0)
Requirement already satisfied: pywin32>=224 in c:\users\admin\appdata\local\programs\python\python39\lib\site-packages (from xlwings) (300)
WARNING: You are using pip version 21.1.2; however, version 21.2.4 is available.
You should consider upgrading via the 'c:\users\admin\appdata\local\programs\python\python39\python.exe -m pip install --upgrade pip' command.

C:\Users\ADMIN>xlwings --version
xlwings version: 0.23.0
Usage: xlwings [-h] {addin,quickstart,restapi,license,config,code} ...
xlwings: error: the following arguments are required: command
C:\Users\ADMIN>

```

Dòng lệnh kiểm tra version
Thông báo version 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

chk_versionXlwings.py - MyHandBook_PyExcel - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER OPEN EDITORS MYHANDBOOK_PYEXCEL chk_versionXlwings.py

chk_versionXlwings.py

```

1 import xlwings
2 print(xlwings.__version__)

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Windows PowerShell

Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

PS D:\MyHandBook_PyExcel> & C:/Users/ADMIN/AppData/Local/Programs/Python/Python39/python.exe d:/MyHandBook_PyExcel/chk_versionXlwings.py

0.23.0

PS D:\MyHandBook_PyExcel>

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://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: LÀM QUEN VỚI NGÔN NGỮ PYTHON

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

Cũng như những thành viên khác, để làm quen với ngôn ngữ lập trình bạn thử code chương trình đầu tiên **Hello Wold!** như sau:

```
# Dấu "#" dùng để ghi chú Python không phiên dịch các dòng này
# Hoặc bạn cũng có thể ghi chú như thế, chú ý phải có dấu ; cuối câu";
# Kiểu string (kiểu chữ, ký tự), được đặt trong dấu nháy ', hoặc dấu """
# \n: ký tự xuống dòng
# print: là phương thức, hàm để in ra màn hình Terminal. Ghi chú nhớ viết đúng chữ print và không viết Hoa

"Dưới đây là code để in ra màn hình Teminal";
print("Hello Word!")
print('Hello Python Excel!!!')
print("Vũ là tên là tôi.\nUsername là: VuNghiXuan\nĐây là code Python đầu tiên.")
```

Nếu như lần đầu bạn trong thành công thì đây là đường Tải file code để bạn có thể copy về thực hiện để hiểu nó.

Tải file: <https://github.com/VuNghiXuan/giaotrinhPythonExcel/tree/main/C1>

1. Python lists

➤ *list*:

Trong python, **list** được xem là 1 danh sách (giống như 1 cái 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...). Python 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])

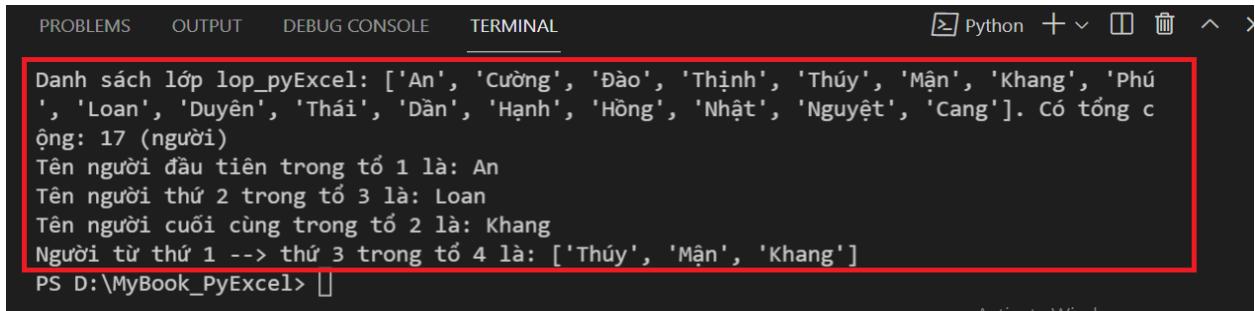
# In ra Tên người cuối cùng trong tổ 2
print("Tên người cuối cùng trong tổ 2 là:", To_2[-1])

# 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])
```

Tải file code:

<https://github.com/VuNghiXuan/giaotrinhPythonExcel/blob/main/C2/1.listPy.py>

Kết quả xuất ra ở Terminal:



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Python + v □ ^ < >
Danh sách lớp lop_pyExcel: ['An', 'Cường', 'Đào', 'Thịnh', 'Thúy', 'Mận', 'Khang', 'Phú ', 'Loan', 'Duyên', 'Thái', 'Dần', 'Hạnh', 'Hồng', 'Nhật', 'Nguyệt', 'Cang']. Có tổng c ộng: 17 (người)
Tên người đầu tiên trong tổ 1 là: An
Tên người thứ 2 trong tổ 3 là: Loan
Tên người cuối cùng trong tổ 2 là: Khang
Người từ thứ 1 --> thứ 3 trong tổ 4 là: ['Thúy', 'Mận', 'Khang']
PS D:\MyBook_PyExcel> []
```

➤ *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 khổng lồ chứa nhiều loại vật dụng nói trên, trong cái túi đó lại chứa 1 cái túi nhỏ hơn đựng nhiều loại bánh kẹo, thì túi lớn chứa túi nhỏ này được gọi là **list trong list**.

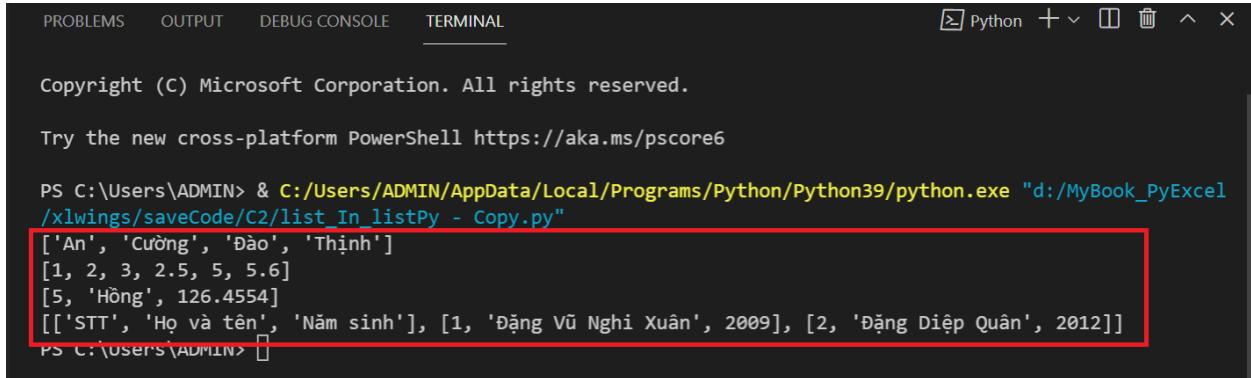
```
# list và list trong list
list_DanhSach = ["An", "Cường", "Đào", "Thịnh"]
list_int = [1, 2, 3, 2.5, 5, 5.6] # Số nguyên (kiểu int) và số thực (kiểu float)
list_str_num = [ 5, "Hồng", 126.4554]

"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)
```

Tải file code tham khảo:

https://github.com/VuNghiXuan/giaotrinhPythonExcel/blob/main/C2/2.list_In listPy.py

Kết quả chạy code in ra màn hình Terminal



The screenshot shows a terminal window with the following output:

```
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\ADMIN & C:/Users/ADMIN/AppData/Local/Programs/Python/Python39/python.exe "d:/MyBook_PyExcel
/xlwings/saveCode/C2/list_In_listPy - Copy.py"
['An', 'Cường', 'Đào', 'Thịnh']
[1, 2, 3, 2.5, 5, 5.6]
[5, 'Hồng', 126.4554]
[['STT', 'Họ và tên', 'Năm sinh'], [1, 'Đặng Vũ Nghi Xuân', 2009], [2, 'Đặng Diệp Quân', 2012]]
PS C:\Users\ADMIN>
```

A red box highlights the last four lines of the output, which are the list and its elements.

2. Vòng lặp for trong list python

- *Cách lấy giá trị phần tử trong list (element):*

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

Cú pháp:

```
for element_list in list:  
    <nhấn phím Tab> Khởi 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 là đi đến từng phần tử trong hộp, bạn có thể 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"]

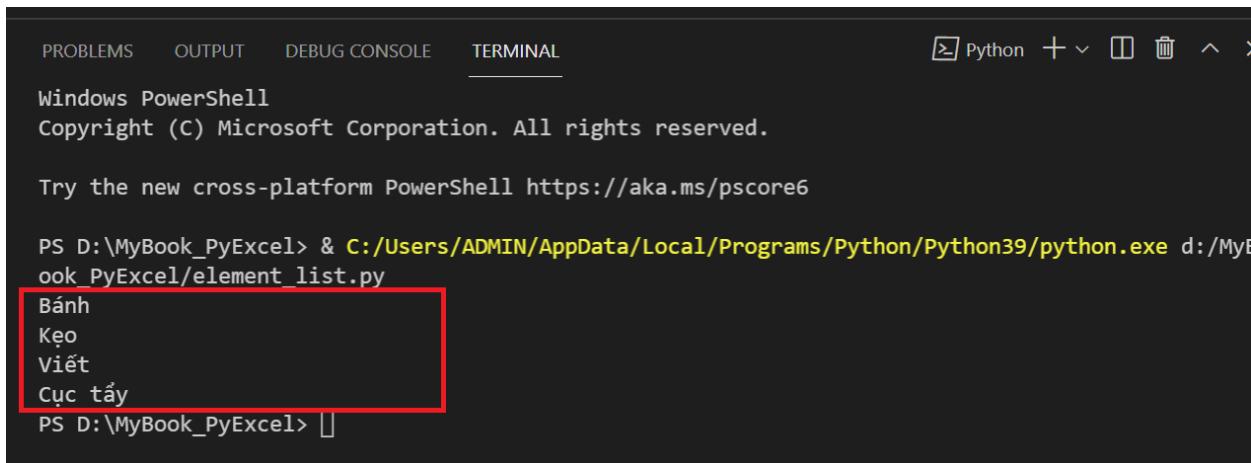
# Dùng vòng lặp for lấy tên các loại chứa trong cái hộp
for element_Hop in hop:
    print(element_Hop)
# Nhập vào các danh sách học sinh Lớp Python_Excel gồm có 4 tổ
hop = ["Bánh", "Kẹo", "Viết", "Cục tẩy"]
```

```
# Dùng vòng lặp for lấy tên các loại chứa trong cái hộp
for element_Hop in hop:
    print(element_Hop)
```

Tải file code:

https://github.com/VuNghiXuan/giaotrinhPythonExcel/blob/main/C2/3.Vonglap_For.py

Kết quả:



The screenshot shows a terminal window with the following output:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\MyBook_PyExcel> & C:/Users/ADMIN/AppData/Local/Programs/Python/Python39/python.exe d:/MyBook_PyExcel/element list.py
Bánh
Kẹo
Viết
Cục tẩy
PS D:\MyBook_PyExcel> |
```

A red rectangular box highlights the output text "Bánh", "Kẹo", "Viết", and "Cục tẩy".

Lưu ý về lỗi:

The screenshot shows a Visual Studio Code interface with a dark theme. On the left is a sidebar with various icons. The main area has a tab bar with "element_list.py - MyBook_PyExcel - Visual Studio Code", "xlwings_objSheet.py", "element_list.py 9", "hello.py", and "hell". The "element_list.py" tab is active, displaying the following code:

```
# Nhập vào các danh sách học sinh Lớp Python_Excel gồm có 4 tố
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)
```

A yellow box highlights the word "For" in the fifth line. The status bar at the bottom shows "Ln 5, Col 2".

The terminal window below shows the command prompt "PS D:\MyBook_PyExcel> & C:/Users/ADMIN/AppData/Local/Programs/Python/Python39/python.exe d:/MyBook_PyExcel/element_list.py" followed by the error message:

```
File "d:\MyBook_PyExcel\element_list.py", line 5
  For element_Hop in hop:
          ^
SyntaxError: invalid syntax
PS D:\MyBook_PyExcel>
```

The status bar at the bottom of the terminal shows "Ln 5, Col 2 Spaces: 4 UTF-8 CRLF Python".

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 “**:**” sẽ có thông báo lỗi giống vậy.

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

```

element_list.py - MyBook_PyExcel - Visual Studio Code
File Edit Selection View Go Run Terminal Help
RUN AND DEBUG: RUN ...
Welcome xlwings_objSheet.py element_list.py 1 hello.py ...
element_list.py > ...
1 # Nhập vào các danh sách học sinh Lớp Python_Excel gồm có 4 tố
2 hop = ["Bánh", "Kẹo", "Viết", "Cục tẩy"]
3
4 # Dùng vòng lặp for lấy tên các loại chứa trong cái hộp
5 for element_Hop in hop:
6     print(element_Hop)

Sửa lỗi: Nhập 1 lần phím Tab trên bàn phím thực vào như word

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\MyBook_PyExcel> & C:/Users/ADMIN/AppData/Local/Programs/Python/Python39/python.exe d:/MyBook_PyExcel/element_list.py
File "d:/MyBook_PyExcel/element_list.py", line 6
    print(element_Hop)
          ^
IndentationError: expected an indented block
PS D:\MyBook_PyExcel>

```

Activate Windows
Go to Settings to activate Windows.

Python 3.9.2 64-bit 1 △ 0 🔍

Type here to search

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Try the new cross-platform PowerShell https://aka.ms/pscore6
PS D:\MyBook_PyExcel> & C:/Users/ADMIN/AppData/Local/Programs/Python/Python39/python.exe d:/MyBook_PyExcel/element_list.py
File "d:/MyBook_PyExcel/element_list.py", line 6
 print(element_Hop)
 ^
IndentationError: expected an indented block
PS D:\MyBook_PyExcel>

➤ Cách lấy giá trị từ index_list (chỉ số mục) trong list:

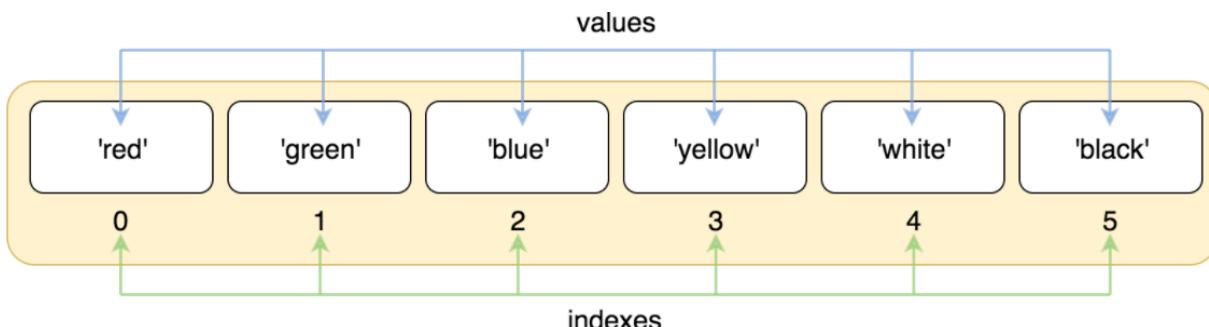
Cú pháp:

```

for index_list in range(len(list)):
    <nhấn phím Tab> Khởi 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 element có gán với index trong list

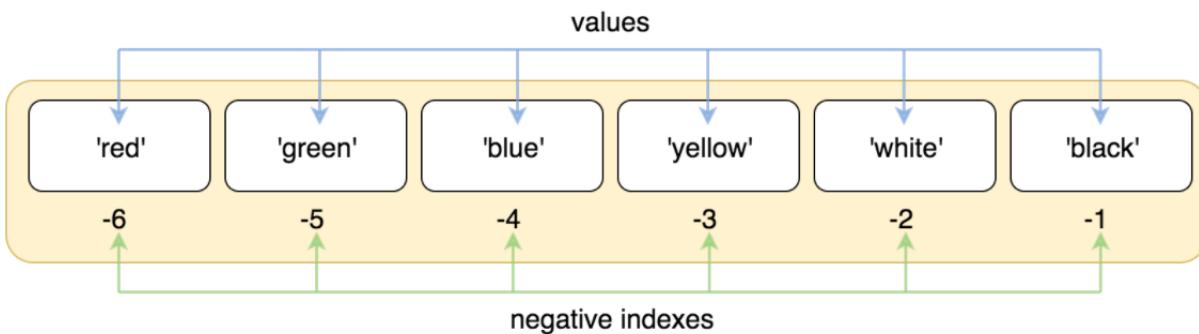


Sơ đồ cấu trúc trong 1 list (Phần màu sắc là giá trị, mỗi giá trị gán 1 index khác nhau)

Cách lấy giá trị trong list dựa theo index

```
1. >>> colors = ['red', 'green', 'blue', 'yellow', 'white', 'black']
2. >>> colors[0]
3. 'red'
4. >>> colors[1]
5. 'green'
6. >>> colors[5]
7. 'black'
```

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



Chỉ số index trong list theo chiều từ phần tử cuối ngược về phần tử đầu tiên

Cách lấy giá trị của phần tử trong list dựa theo index chiều ngược lại

```
1. >>> colors = ['red', 'green', 'blue', 'yellow', 'white', 'black']
2. >>> colors[-1]
3. 'black'
4. >>> colors[-2]
5. 'white'
6. >>> colors[-6]
7. 'red'
```

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
"""

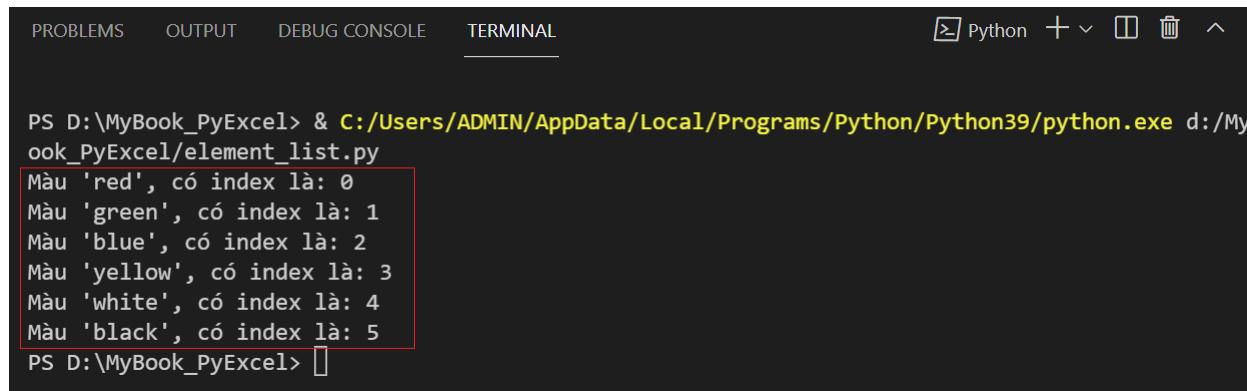
# In ra giá trị màu xem có đúng theo chỉ số index trên không?
for i_color in range(len(colors)):
    print(f"Màu '{colors[i_color]}', có index là: {i_color}")

```

Tải file code:

https://github.com/VuNghiXuan/giaotrinhPythonExcel/blob/main/C2/4.Vong_lap_For_index.py

Kết quả được xuất ra Terminal



```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Python + ▾ ^

PS D:\MyBook_PyExcel> & C:/Users/ADMIN/AppData/Local/Programs/Python/Python39/python.exe d:/MyBook_PyExcel/element_list.py
Màu 'red', có index là: 0
Màu 'green', có index là: 1
Màu 'blue', có index là: 2
Màu 'yellow', có index là: 3
Màu 'white', có index là: 4
Màu 'black', có index là: 5
PS D:\MyBook_PyExcel>

```

➤ 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).

➤ *Cách lấy và thay đổi giá trị trong list:*

```
# Nhập vào các danh sách học sinh Lớp Python_Excel gồm có 4 tố
hop = ["Bánh", "Kẹo", "Viết", "Cục tẩy"]

# Số phần tử trong list
num_index = len(hop)

# Dùng vòng lặp for thay thế giá trị trong list
if hop[idx] == "Kẹo":
    hop[idx] = "Kim cương"
    break # break: Nghĩa là, khi tìm thấy điều kiện thì dừng và thoát
khỏi vòng lặp
print(hop)
```

Bạn chú ý phần tử thứ 2 trong list, lúc đầu có index = 1, value = “Kẹo”, qua vòng lặp giá trị được thay thế là “Kim cương”.

Tải code:

https://github.com/VuNghiXuan/giaotrinhPythonExcel/blob/main/C2/3.a.ChangeValue_for.py

Kết quả thực thi code:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\MyBook_PyExcel\xlwings\saveCode\C2> & C:/Users/ADMIN/AppData/Local/Programs/Python/Python39/python.exe d:/MyBook_PyExcel/xlwings/saveCode/C2/3.a.ChangeValue_for.py
['Bánh', 'Kim cương', 'Viết', 'Cục tẩy']
PS D:\MyBook_PyExcel\xlwings\saveCode\C2>
```

3. Vòng lặp While python

Cú pháp:

```
while điều_kiện_kiểm_tra:  
    Khối lệnh của while
```

Để mô phỏng cách hoạt động trong **vòng lặp While**, mình lấy ví dụ cô giáo dặn học sinh như sau:

```
Trong khi cô_giảng_bài_10phút:  
    các_em_phải_yên_lặng
```

Như vậy, trong thời gian 10 phút thì lời dặn cô, có hiệu lực (tức là điều_kiện_kiểm_tra =True) thì Khối lệnh là **các_em_phải_yên_lặng** được thực hiện.

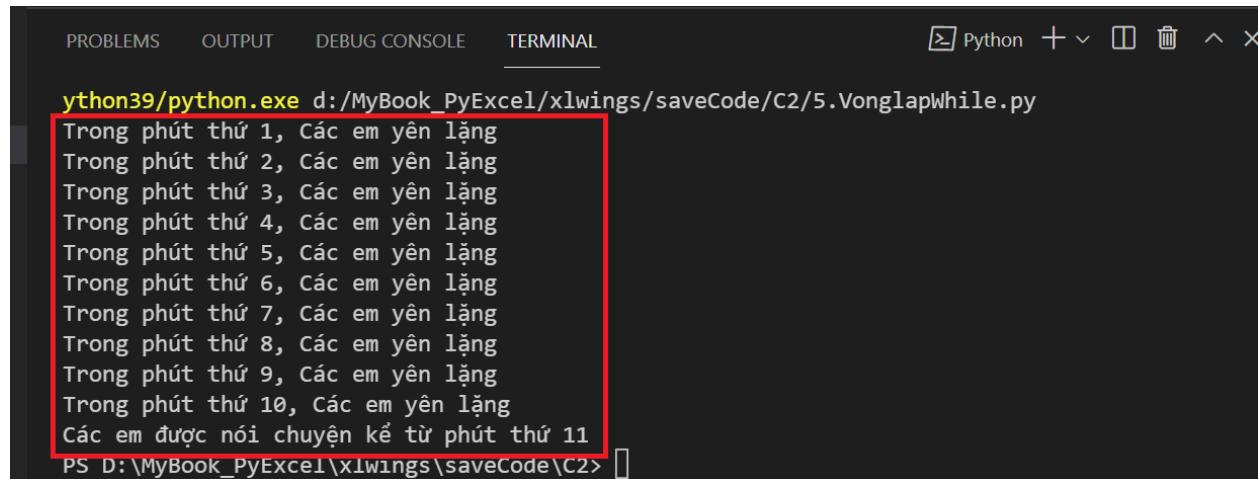
Code:

```
# Thời gian bắt đầu  
thoi_gian = 0  
while thoi_gian <10:  
    print(f"Trong phút thứ {thoi_gian+1}, Các em yên lặng")  
    thoi_gian += 1 # thoi_gian tăng lên mỗi phút  
print(f"Các em được nói chuyện kể từ phút thứ {thoi_gian+1}")
```

Tải code:

<https://github.com/VuNghiXuan/giaotrinhPythonExcel/blob/main/C2/5.VonglapWhile.py>

Kết quả thực thi code:



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
```

```
python39/python.exe d:/MyBook_PyExcel/xlwings/saveCode/C2/5.VonglapWhile.py  
Trong phút thứ 1, Các em yên lặng  
Trong phút thứ 2, Các em yên lặng  
Trong phút thứ 3, Các em yên lặng  
Trong phút thứ 4, Các em yên lặng  
Trong phút thứ 5, Các em yên lặng  
Trong phút thứ 6, Các em yên lặng  
Trong phút thứ 7, Các em yên lặng  
Trong phút thứ 8, Các em yên lặng  
Trong phút thứ 9, Các em yên lặng  
Trong phút thứ 10, Các em yên lặng  
Các em được nói chuyện kể từ phút thứ 11
```

4. Hàm def trong python

Người dùng tự xây dựng hàm để thực hiện 1 chức năng nào đó theo chủ ý của

mình và bắt đầu bằng từ khóa **def** (hàm tự định nghĩa của người dùng).

Cú pháp:

```
def ten_ham(các tham số/đôi số):  
    Các câu lệnh
```

Để dễ hiểu về **def trong python**, bạn xem toàn bộ hàm như một cỗ máy cỗ sản xuất bò viên (*mình dùng từ cỗ vì bây giờ không còn ai dùng máy này nữa*).

Hãy nhìn bức tranh bao quát lại như sau:

```
def may_che_bien(thit_bo, gia_vị):  
    bo_vien = thịt_bo + gia_vị  
    return bo_vien
```

Trong đó:

- **def**: như 1 chiếc chìa khóa khởi động máy
- **may_che_bien**: tên hàm đặt tên là máy chế biến
- **thit_bo**: nguyên liệu chính, trong hàm là **tham số 1**
- **gia_vị**: là gia vị, bao gồm: muối, ớt, tiêu, mắm, bột ngọt
- **return**: là chìa khóa mở cửa của đầu ra sản phẩm
- **bo_vien**: là sản phẩm đầu ra (giá trị hàm trả về)

Để mô phỏng bài toán mình đi code bài tính tổng 2 phép cộng với đoạn code:

```
# Xây dựng hàm để tính tổng 2 số  
def tong2so(a, b):  
    tong = a + b  
    return tong
```

Tiếp theo bạn truyền 2 số a=5, b=2.5 và truyền vào hàm

```
# Xây dựng hàm để tính tổng 2 số  
def tong2so(a, b):  
    tong = a + b  
    return tong  
  
# Cho a=5, b=2.5  
a = 5  
b = 2.5
```

```

# Truyền tham số vào hàm như sau:
tong_a_b = tong2so(a, b)

# In ra kết quả:
print("Tổng a + b là: ", tong_a_b)

```

Tải code:

https://github.com/VuNghiXuan/giaotrinhPythonExcel/blob/main/C2/6.def_Tong2so.py

Kết quả thực thi code:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\MyBook_PyExcel\xlwings\saveCode\C2> & C:/Users/ADMIN/AppData/Local/Programs/Python/Python39/python.exe d:/MyBook_PyExcel/xlwings/saveCode/C2/6.def_Tong2so.py
Tổng a + b là: 7.5
PS D:\MyBook_PyExcel\xlwings\saveCode\C2>

```

5. Đối tượng Class Python

Python là một ngôn ngữ lập trình hướng đối tượng. **Đối tượng (Object)** chỉ đơn giản là một tập hợp các dữ liệu (các biến) và các phương thức (các hàm) hoạt động, thuộc tính trên các dữ liệu đó, tạo thành 1 lớp gọi là (**class**).

Cấu trúc khai báo 1 class:

```

class Tên_Class:
    '''Đây là docstring. Một lớp mới vừa được khai báo.'''
    def __init__(self):
        pass

```

Tên_Class: Là tên đối tượng.

__int__: Là khởi tạo các thuộc tính, phương thức của đối tượng.

self: nghĩa là chính đối tượng (**Tên_Class**), quy ước này nhằm dễ phân biệt với các tham số (thuộc tính) truyền vào. Bạn có thể thay bằng từ khác như (object, a, b,

c,...) vẫn không ảnh hưởng đến cấu trúc, hoạt động.

pass: là 1 từ khóa, khi gặp từ này Python sẽ thực hiện công việc gì cả, chuyển đến câu lệnh tiếp theo. Người lập trình có thể tận dụng từ khóa này để tạm dừng phần code mà không bị báo lỗi trong khi chưa nghĩ ra đoạn code tiếp theo.

The screenshot shows the Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help
- Editor Bar:** 7.class.py - C2 - Visual Studio Code
- Explorer:** Shows a list of files: 1.listPy.py, 2.list_In_listPy.py, 3.a.ChangeValue_for.py, 3.Vonglap_For.py, 4.Vonglap_For_index.py, 5.VonglapWhile.py, 6.def_Tong2so.py, 7.class.py (highlighted with a green box), 8.class_hocsinh.py
- Editor Content:** A Python class definition:

```
1. class hocsinh():
2.     # Khởi tạo các phương thức, thuộc tính của học sinh
3.     def __init__(self):
4.         pass # Tạm dừng để suy nghĩ đoạn code tiếp theo
```
- Terminal:** Displays command-line output:

```
Bác Kiều Phong. Có điểm trung bình 9.333. Kết quả: Đậu
Nam Mộ Dung. Có điểm trung bình 5.333. Kết quả: Đậu
PS D:\MyBook_PyExcel\xlwings\saveCode\C2> & C:/Users/ADMIN/AppData/Local/Programs/Python/Python39/python.exe d:/MyBook_PyExcel\xlwings\saveCode/C2/7.class.py
  File "d:/MyBook_PyExcel\xlwings\saveCode\C2/7.class.py", line 7
    ^
IndentationError: expected an indented block
PS D:\MyBook_PyExcel\xlwings\saveCode\C2>
```

Phản lỗi sinh ra khi không dùng từ khóa pass (tạm dừng)

Sau đây là cách khai báo 1 đối tượng học sinh đơn giản:

Code:

```
class hocsinh():

    # Khởi tạo các phương thức, thuộc tính của học sinh
    def __init__(self):
        pass # Tạm dừng để suy nghĩ đoạn code tiếp theo
```

Sau khi suy nghĩ sau các thuộc tính đối tượng cần khai báo mình bỏ từ khóa **pass** và thêm vào các thuộc tính của đối tượng **hocsinh** như như **tên học sinh**:

Code:

```
class hocsinh():

    def __init__(self, ten):
        self.ten = str(ten)

    # Cách truyền tham số vào đối tượng
```

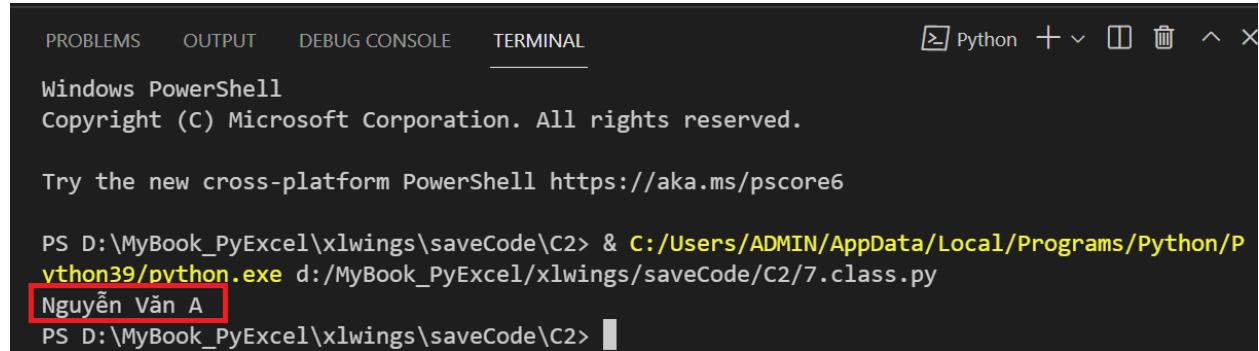
```
hs_1 = hocsinh("Nguyễn Văn A")

# Lấy thông tin đối tượng và in ra tên đối tượng
print(hs_1.ten)
```

Tải code:

<https://github.com/VuNghiXuan/giaotrinhPythonExcel/blob/main/C2/7.class.py>

Kết quả thực thi code:



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\MyBook_PyExcel\xlwings\saveCode\C2> & C:/Users/ADMIN/AppData/Local/Programs/Python/Python39/python.exe d:/MyBook_PyExcel/xlwings/saveCode/C2/7.class.py
Nguyễn Văn A
PS D:\MyBook_PyExcel\xlwings\saveCode\C2>
```

Tiếp theo thêm 1 số thuộc tính cho đối tượng như **điểm toán, lý, hóa** và hàm tính **điểm trung bình**, hàm in ra **kết quả thi** như sau:

```
class hocsinh():

    # Khởi tạo 1 đối tượng học sinh
    def __init__(self, ten, toan, ly, hoa):
        self.ten = str(ten)
        self.toan = float(toan)
        self.ly = float(ly)
        self.hoa = float(hoa)

    # Phương thức tính điểm trung bình
    def diem_TB(self):
        self.avrg_score = round((self.toan + self.ly + self.hoa)/3,3)
        return self.avrg_score

    # Thuộc tính (Chiều cao, cân nặng)
    def ketqua(self):
```

```

if self.avrg_score > 5:
    return "Đậu"
else: return "Rớt"

hs = [hocsinh("Bắc Kiều Phong", 9, 10, 10),
      hocsinh("Nam Mộ Dung", 5, 6, 7),
      hocsinh("Nam Đế", 4, 5, 1),
      hocsinh("Bắc Cái", 10, 0, 1)]

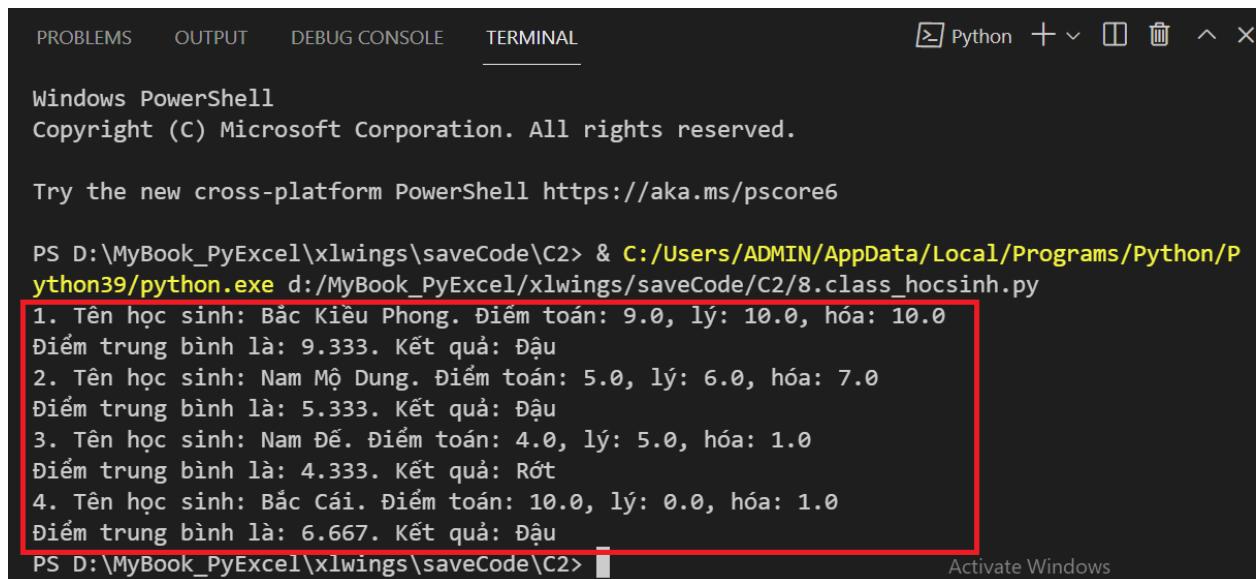
for i_hs in range(len(hs)):
    print(f'{hs[i_hs].ten}. Có điểm trung bình {hs[i_hs].diem_TB()}. Kết quả: {hs[i_hs].ketqua()}')

```

Tải code:

https://github.com/VuNghiXuan/giaotrinhPythonExcel/blob/main/C2/8.class_hocsinh.py

Kết quả thực thi code:



```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Python + ▾ □ ^ ×

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\MyBook_PyExcel\xlwings\saveCode\C2> & C:/Users/ADMIN/AppData/Local/Programs/Python/Python39/python.exe d:/MyBook_PyExcel/xlwings/saveCode/C2/8.class_hocsinh.py
1. Tên học sinh: Bắc Kiều Phong. Điểm toán: 9.0, lý: 10.0, hóa: 10.0
Điểm trung bình là: 9.333. Kết quả: Đậu
2. Tên học sinh: Nam Mộ Dung. Điểm toán: 5.0, lý: 6.0, hóa: 7.0
Điểm trung bình là: 5.333. Kết quả: Đậu
3. Tên học sinh: Nam Đế. Điểm toán: 4.0, lý: 5.0, hóa: 1.0
Điểm trung bình là: 4.333. Kết quả: Rớt
4. Tên học sinh: Bắc Cái. Điểm toán: 10.0, lý: 0.0, hóa: 1.0
Điểm trung bình là: 6.667. Kết quả: Đậu
PS D:\MyBook_PyExcel\xlwings\saveCode\C2>

```

6. Đối tượng kế thừa Class Python

Để giúp bạn hiểu rõ về kế thừa trong Python, vận dụng khi làm việc với GUI (giao diện đồ họa), phần này sẽ gặp nhiều khi bạn sử dụng các Gui Tkinter, pyQt.

Dưới đây là ví dụ kế thừa trong Python.

```
class Base(object):
    def __init__(self, property):
        self.property = property

class Child(Base):
    def __init__(self, childProperty):
        super(Child, self).__init__()
        self.childProperty = childProperty
```

Khi khởi tạo Child bạn sẽ truyền vào 2 biến:

```
childInstance = Child(property, childProperty)
```

Lưu ý: Base class phải inherit (kế thừa) từ object thì các class khác mới inherit (kế thừa) từ nó được. Trong python 3 bạn chỉ cần viết: super().__init__() thay vì super(Child, self).__init__().

CHƯƠNG 3: THU VIỆN XLWINGS

Thu 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.

I. ĐỐI TƯỢNG XLWINGS

Phần này cực kỳ quan trọng, nó giúp bạn hiểu rõ bản chất của lập trình hơn. Một số bạn code được rất nhiều thứ nhưng thường xuyên bị lỗi và không biết làm thế nào để gỡ lỗi, lý do là vì chưa hiểu rõ về đối tượng. Điều này, giống như bạn biết về 1 người nào đó mà chỉ biết tên của họ, còn họ thường làm gì, thích cái gì bạn đều không biết.

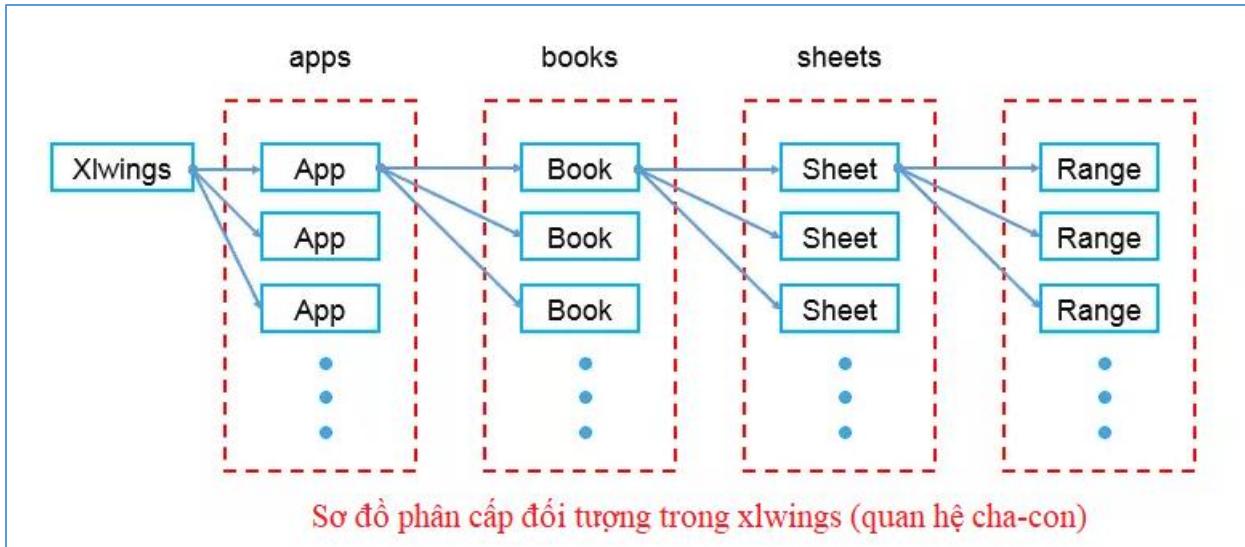
Python cũng thế, nó là ngôn ngữ lập trình theo hướng đối tượng (*tiếng Anh: Object-oriented programming, viết tắt: OOP*). Đối tượng tức là những vật thể tồn tại (dùng để phân biệt đối tượng này với đối tượng khác), có trạng thái và hành vi...

Mỗi **đối tượng** đều có các phương thức, thuộc tính khác nhau. Trong đó, **phương thức** là chỉ hành động: đi, đứng, chạy,... **thuộc tính** chỉ tính chất, đặc điểm như là: màu sắc, tính cách, mô tả từng bộ phận của đối tượng đó.

Ví dụ: **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** của con mèo là: có râu, móng vuốt, màu lông...

Đối tượng được phân cấp rõ ràng, mắc xích quan hệ với nhau là cha con (trong nhà ông A, có thằng con tên B, con thằng B là thằng C, cứ thế cho đến thế hệ cuối cùng). Đối tượng xlwings cũng vậy, bạn xem sơ đồ bên dưới.



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

Hãy khởi động VSC và nhập đoạn code sau:

```
import xlwings as xw
```

Giải thích về code:

- **import:** là 1 từ khóa trong python, chức năng báo cho chương trình biết tôi đang nhập 1 thư viện từ bên ngoài.
- **xlwings:** là tên thư viện (hay **đối tượng**) dùng để giao tiếp, điều khiển file Excel
- **as:** là 1 từ khóa, dùng để gán tên cho thư viện, giống như dấu “ = ” trong toán học (ý muốn nói xlwing = xw)
- **xw:** là tên biến (gọi tắt) cho xlwing. Ví dụ tên của bạn đầy đủ là Nguyễn Văn Tèo, người ta có thể gọi bạn là Tèo.

Dòng lệnh trên có nghĩa là nhập vào thư viện xlwings và gán cho nó cái tên là xw (còn gọi là biến), bạn có thể gán là xls, hay tên 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,... Bạn có thể google để tìm hiểu cách đặt tên biến.

Lưu ý xảy ra các lỗi sau:

1. Nếu cửa sổ dòng lệnh xuất hiện dòng lỗi "**ModuleNotFoundError: No module named ‘xlwings‘**", có nghĩa là bạn chưa cài đặt thư viện xlwing. Bạn quay lại Chương 1 để được hướng dẫn cài đặt thư viện này.
2. Các từ khóa "**“import”, “as”**" viết đúng theo giáo trình và không được viết Hoa

II. ĐỐI TƯỢNG WORKBOOK (BẢNG TÍNH EXCEL)

Python là ngôn ngữ lập trình thông dịch (khác với ngôn ngữ biên dịch khác), giao tiếp với Excel thông qua thư viện bên ngoài là xlwings, do đó cần phải cài đặt và “import” thư viện bên ngoài thì mới hoạt động được.

Phần cài đặt mình đã nói ở Chương 1 rồi, bây giờ bạn hãy mở trình soạn thảo VSC để tiến hành import thư viện xlwings và gọi bảng tính Excel.

1. Mở Workbook thông qua lớp ứng dụng xlwings.App

App là một ứng dụng, tương ứng với một phiên bản Excel, được sử dụng làm trình quản lý đảm bảo rằng mọi thứ được làm sạch lại đúng cách và ngăn chặn các quy trình zombie (*về cơ bản là các phần còn lại của các quá trình chết chưa được dọn sạch. Một chương trình tạo ra các quy trình zombie không được lập trình đúng*) thì các phiên bản Excel mới có thể được kích hoạt.

Bạn có thể quản lý được các file, phiên bản khác nhau của Excel khi bạn đang tương tác với chúng một cách dễ dàng (*trên Mac, xlwings cho phép bạn chạy nhiều phiên bản Excel*).

```
xlwings.App(visible=None, spec=None, add_book=True, impl=None)
```

Tham số:	<ul style="list-style-type: none"> - visible (kiểu <i>bool</i>, mặc định là <i>None</i>) – visible=True: Cho phép hiện bảng tính, ngược lại = False. - spec (kiểu <i>str</i>, mặc định là <i>None</i>) – Cho phép mở nhiều phiên bản Excel, tuy nhiên chỉ có Mac mới sử dụng được, khi sử dụng phải khai báo rõ đường dẫn đầy đủ của phiên bản cài đặt. <p style="margin-top: 10px;">/Applications/Microsoft Office 2011/Microsoft Excel hoặc /Applications/Microsoft Excel</p> <p>Trên Windows, khi thay đổi phiên bản Excel bạn phải vào Control Panel > Programs and Features và chạy Repair bộ cài Office để thay đổi mặc định phiên bản này.</p> <ul style="list-style-type: none"> - Add_book (kiểu <i>bool</i>, mặc định là <i>None</i>) –True: Tạo thêm 1 workbook thứ 2, ngược lại = False.
-----------------	--

2. Mở 1 Workbook mới

Code như dưới đây sẽ thực hiện mở Excel và 1 workbook mới nạp vào Excel, cho phép hiển thị, nhìn thấy và hiện ra Terminal phiên bản của Excel.

```
import xlwings as xw

# Khởi tạo App cho phép hiện ra màn hình
app = xw.App(visible = True)

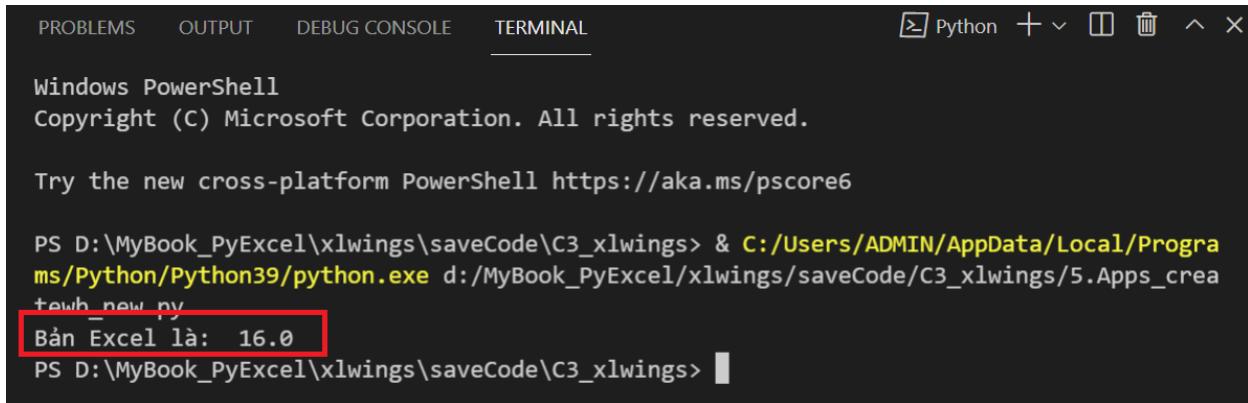
"""
Ghi chú:
+ visible=True: Nghĩa là cho phép hiện bảng tính mới vừa tạo
+ add_book = False": Nghĩa là không tạo thêm 01 bảng tính mới nữa
+ add_book: Không khai báo mặc định là False
"""

# In ra phiên bản app
print("Bản Excel là: ", app.version)
```

Tải code:

https://github.com/VuNghiXuan/giaotrinhPythonExcel/blob/main/C3_xlwings/5.Apps_createwb_new.py

Kết quả được in ra trên Terminal



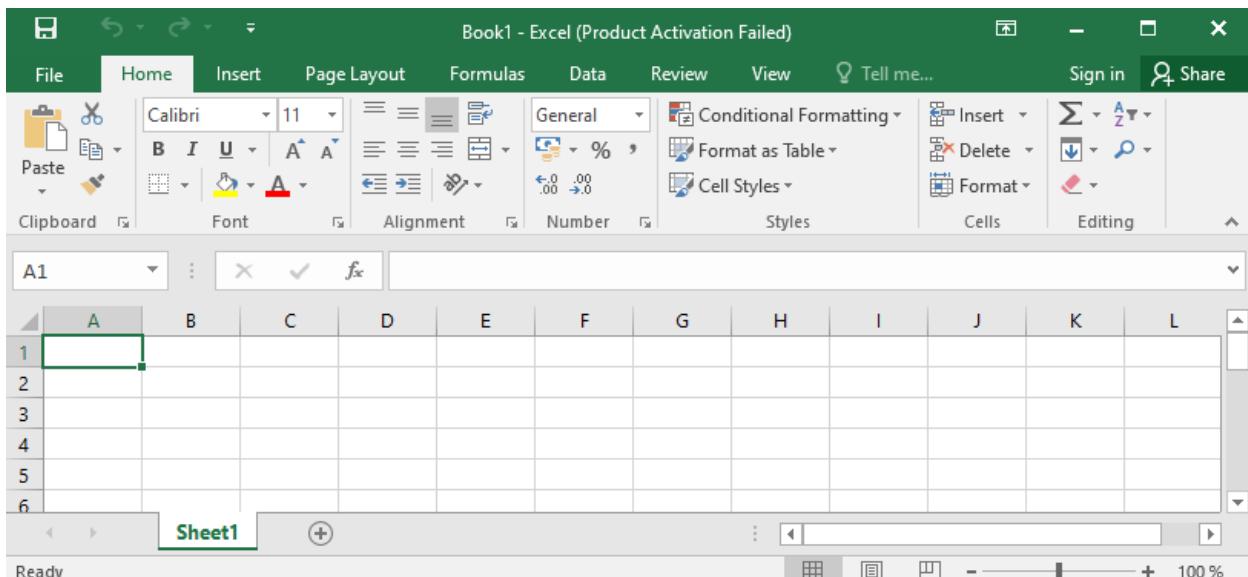
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Python + × □ ^ ×

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\MyBook_PyExcel\xlwings\saveCode\C3_xlwings> & C:/Users/ADMIN/AppData/Local/Programs/Python/Python39/python.exe d:/MyBook_PyExcel/xlwings/saveCode/C3_xlwings/5.Apps_createwb_new.py
Bản Excel là: 16.0
PS D:\MyBook_PyExcel\xlwings\saveCode\C3_xlwings>
```

Một bảng tính Excel tên “**Book1**” trong có 1 sheet tên là “**Sheet1**” xuất hiện trên màn hình windows như sau:



3. Mở workbook bằng đường dẫn đầy đủ

Mình nghĩ đây là cách tốt nhất cho các bạn mới học lập trình. Vì các bạn có thể hiểu rõ ràng file Excel được gọi từ đâu trong hệ thống, lưu nó như thế nào và trong quá trình code lỡ cúp điện bạn vẫn mở được sau này.

Nếu bạn chưa chuẩn bị file Excel thì hãy tải file movies.xls tại đây để thực hành:

https://github.com/VuNghiXuan/giaotrinhPythonExcel/blob/main/C3_xlwings/movies.xls

Giả sử bạn đã tải file “movies.xls”, về máy và lưu tại đường dẫn là: **D:\MyBook_PyExcel\xlwings\read_Datas**. Ta thực hiện mở lại file “movies.xls” với đường dẫn đầy đủ là **D:\MyBook_PyExcel\xlwings\read_Datas\movies.xls**

```
import xlwings as xw

# Đặt tên biến app = App Excel
app = xw.App()

# Đặt biến path_full làm tên đường dẫn đầy đủ
path_full = 'D:\\MyBook_PyExcel\\xlwings\\read_Datas\\movies.xls'
"Đường dẫn thay bằng '\\\' để tránh lỗi (mình đã giải thích bài dưới)";

# Đặt biến wb (bảng tính Excell), mở bảng đoạn code sau
wb = app.books.open(path_full)

# In ra Terminal tên bảng tính, và đường dẫn đầy đủ file
print(f'Tên bảng tính là:{wb.name}\\nĐường dẫn file:{wb.fullname}')

# Thoát khỏi ứng dụng
app.quit()
```

Tải file code:

https://github.com/VuNghiXuan/giaotrinhPythonExcel/blob/main/C3_xlwings/6.Apps_readBook_pathFull.py

Kết quả được in ra trên Terminal và Bảng tính “**movies.xls**” được mở ra trên windows:

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

PS D:\MyBook_PyExcel\xlwings\saveCode\C3_xlwings> & C:/Users/ADMIN/AppData/Local/Programs/Python/Python39/python.exe d:/MyBook_PyExcel/xlwings/saveCode/C3_xlwings/6.Apps_readBook pathFull.py

Tên bảng tính là: movies.xls
Đường dẫn file:D:\MyBook_PyExcel\xlwings\read_Datas\movies.xls

PS D:\MyBook_PyExcel\xlwings\saveCode\C3_xlwings>

4. Mở Workbook bằng xw.Book

Có các cách mở worbook như sau:

	xw.Book	xw.books
New book	<code>xw.Book()</code>	<code>xw.books.add()</code>
Unsaved book	<code>xw.Book('Book1')</code>	<code>xw.books['Book1']</code>
Book by (full)name	<code>xw.Book(r'C:/path/to/file.xlsx')</code>	<code>xw.books.open(r'C:/path/to/file.xlsx')</code>

4.1. Mở nhanh 1 workBook mới

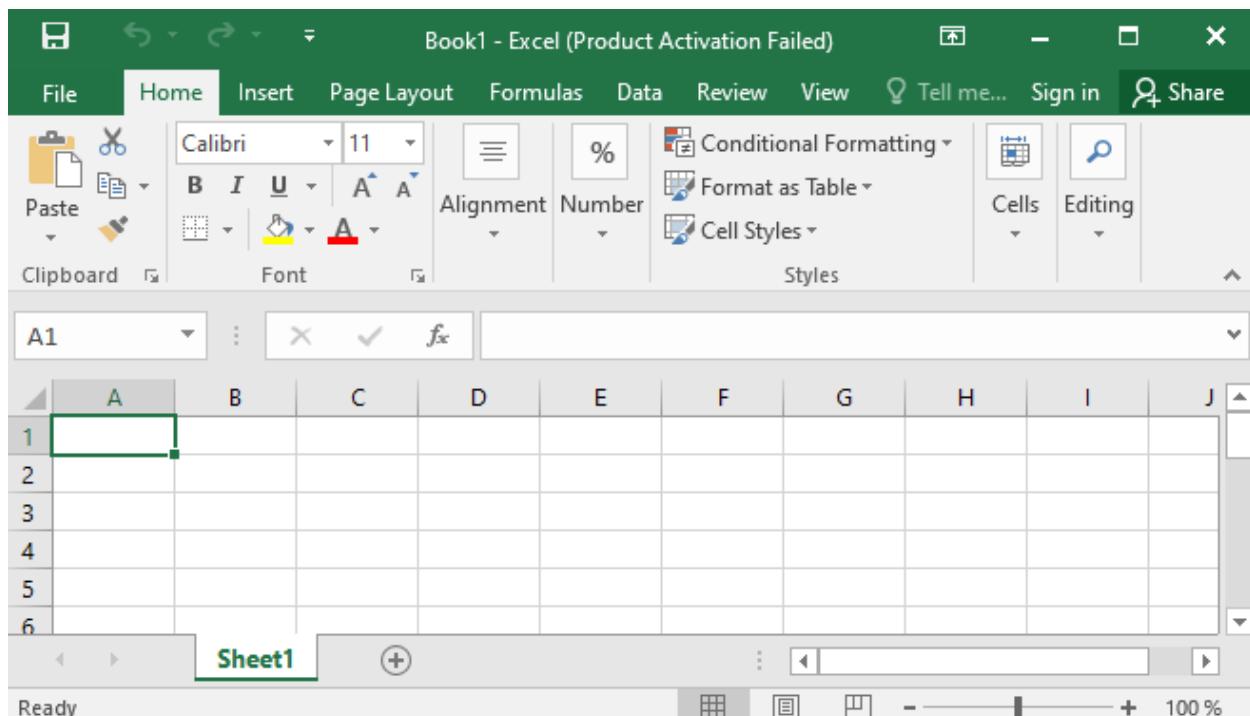
```
# import xlwings as xw
import xlwings as xw

# Mở mới 1 workbooks
xw.Book()
```

Tải file code:

https://github.com/VuNghiXuan/giaotrinhPythonExcel/blob/main/C3_xlwings/1.xw_Book.py

Nhấn nút run () ở góc trên bên phải màn hình soạn thảo VSC, hoặc tổ hợp phím Ctrl +F5 để thực hiện đoạn lệnh. Kết quả 1 workbook mới vừa được tạo ra có tên là “Book1” và trong có sheet tên là “Sheet1”.



Bạn đã thành công khi đã mở được 1 bảng tính, bây giờ bạn muốn mở thêm 1 workbook nữa thì phải dùng phương thức **xw.books.add()**. Cụ thể code như sau:

```
# import xlwings as xw
import xlwings as xw

# Mở mới 1 workbooks
xw.Book()

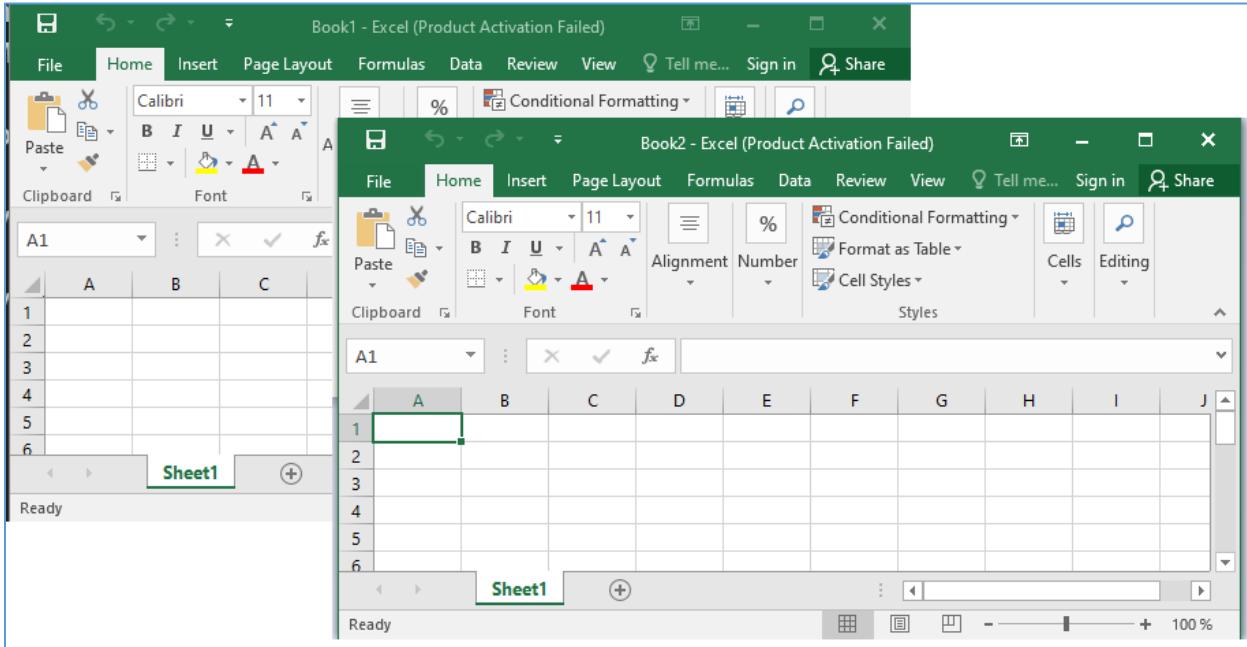
# Từ 1 workbooks trên mở thêm 1 workbooks bằng phương thức
xw.books.add()
```

Chú ý: Để tránh lỗi, trước khi thực hiện câu lệnh **xw.books.add()**, phải đảm bảo rằng trước đó bạn đã mở mới 1 Book mới (chẳng hạn đoạn code **xw.Book()** phải được thực hiện trước).

Tải file code:

https://github.com/VuNghiXuan/giaotrinhPythonExcel/blob/main/C3_xlwings/2.xw_books.add.py

Kết quả sau khi nhấn phím run chạy code như sau:

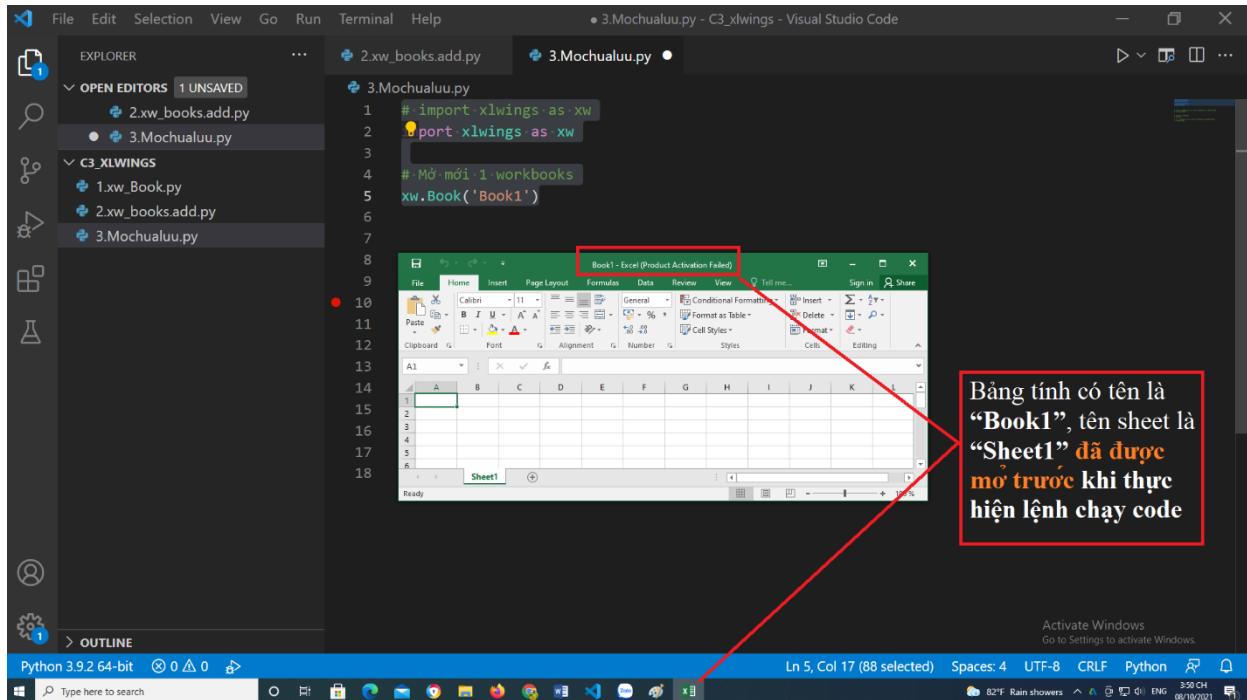


Kết quả là bạn có 02 workbooks có tên là “**Book1**”, “**Book2**” và cả 02 workbooks đều có **Sheet1**.

4.2. Mở chưa lưu

Cách này chỉ thực hiện được khi màn hình Window của bạn đã mở sẵn 1 Workbook có tên là “**Book1**” và chưa lưu nó bất kỳ vào 1 đường dẫn khác.

Giả sử màn hình của mình đã mở sẵn bảng tính có tên là “**Book1**”, tên sheet là “**Sheet1**” bằng đoạn code ở trên kết quả như hình dưới:



Tiếp theo bạn comment lại dòng code `xw.Book()` và thêm vào đoạn code `xw.Book('Book1')`

```
# import xlwings as xw
import xlwings as xw

"Comment lại đoạn code xw.Book()";
# xw.Book()

"Để thực hiện đoạn code dưới đây";
xw.Book('Book1')
```

Tải file code:

https://github.com/VuNghiXuan/giaotrinhPythonExcel/blob/main/C3_xlwings/3.Mochualuu.py

Sau đó, bạn mới chạy code. Kết quả code chạy không báo lỗi. Ngược lại khi chưa mở file, hoặc file đã được lưu vào đường dẫn khác rồi, khi chạy code bạn sẽ được báo lỗi `FileNotFoundException: No such file: 'book1'`, nghĩa là không tìm thấy file có tên là 'Book1'.

Cả 2 cách trên mình không khuyến khích bạn thực hiện theo, trừ khi bạn đã hiểu sâu về nó. Lý do, cực kỳ đơn giản bảng tính Excel là đối tượng và đã là đối tượng thì bạn cần phải đặt tên cho nó, lưu nó vào 1 địa chỉ rõ ràng để quản lý và thuận tiện gọi ra khi cần thiết.

4.3. Mở nhanh workbook bằng đường dẫn đầy đủ

Thực hiện đoạn code:

```
import xlwings as xw

# Đặt đường dẫn đầy đủ tên file là
path_full = "D:\MyBook_PyExcel\xlwings\read_Datas\movies.xls"

# Mở file bằng phương thức xw.Book từ đường dẫn path_full và đặt tên biến là Excel_movies
Excel_movies = xw.Book(r'path_full')

print(Excel_movies.name)
```

Khi chạy code mình gặp 1 lỗi như thế này:

Dòng code gây lỗi → path_full = "D:\\MyBook_PyExcel\\xlwings\\read_Datas\\movies.xls"

Nguyên nhân gây lỗi →

Dòng gây lỗi → SyntaxError: (unicode error) 'unicodeescape' codec can't decode bytes in position 17-18: truncated \xxx escape

Hình dòng code gây ra lỗi do: "\r" trùng với ký tự trong ngôn ngữ Python

Lỗi này gây ra do đường dẫn **D:\MyBook_PyExcel\xlwings\read_Datas\movies.xls** tại vị trí có các dấu “\”. Vì trong ngôn ngữ Python, cũng như một số ngôn ngữ khác, các ký tự này dùng để thực hiện một số chức năng như: \r – đưa con trỏ văn bản console về đầu dòng; \t – Thực hiện thụt đầu dòng (Tab), \n: nghĩa là ký tự xuống dòng. Trong bài code Hello World !!! ở chương 2 mình cũng có nhắc đến ký tự \n này.

Có nhiều cách sửa lỗi đoạn code trên. Mình đưa 2 cách cơ bản như sau:

- **Cách 1: Bạn thêm 1 dấu “\” để trở thành “\\” như sau:**

```
import xlwings as xw

# Đặt đường dẫn đầy đủ tên file là
# path_full = "D:\MyBook_PyExcel\xlwings\read_Datas\movies.xls"

# Cách 1: Thêm "\" vào trở thành "\\"
path_full = "D:\\MyBook_PyExcel\\xlwings\\read_Datas\\movies.xls"

# Mở file bằng phương thức xw.Book từ đg dãy path_full và đặt tên biến là Excel_movies
# Excel_movies = xw.Book(r'D:\MyBook_PyExcel\xlwings\read_Datas\movies.xls')
Excel_movies = xw.Book(path_full)

print(Excel_movies.name)
```

Tải file sửa lỗi code **cách 1**:

https://github.com/VuNghiXuan/giaotrinhPythonExcel/blob/main/C3_xlwings/4.opend_wb_pathfull.py

- **Cách 2: thêm ký tự r trước đường dẫn và gán trực tiếp vào xw.Book()**

```
import xlwings as xw

# Đặt đường dẫn đầy đủ tên file là
# path_full = "D:\MyBook_PyExcel\xlwings\read_Datas\movies.xls"

# Cách 2: thêm ký tự r và gán trực tiếp phương thức Book
```

```

Excel_movies = xw.Book(r"D:\MyBook_PyExcel\xlwings\read_Datas\movies.xls")

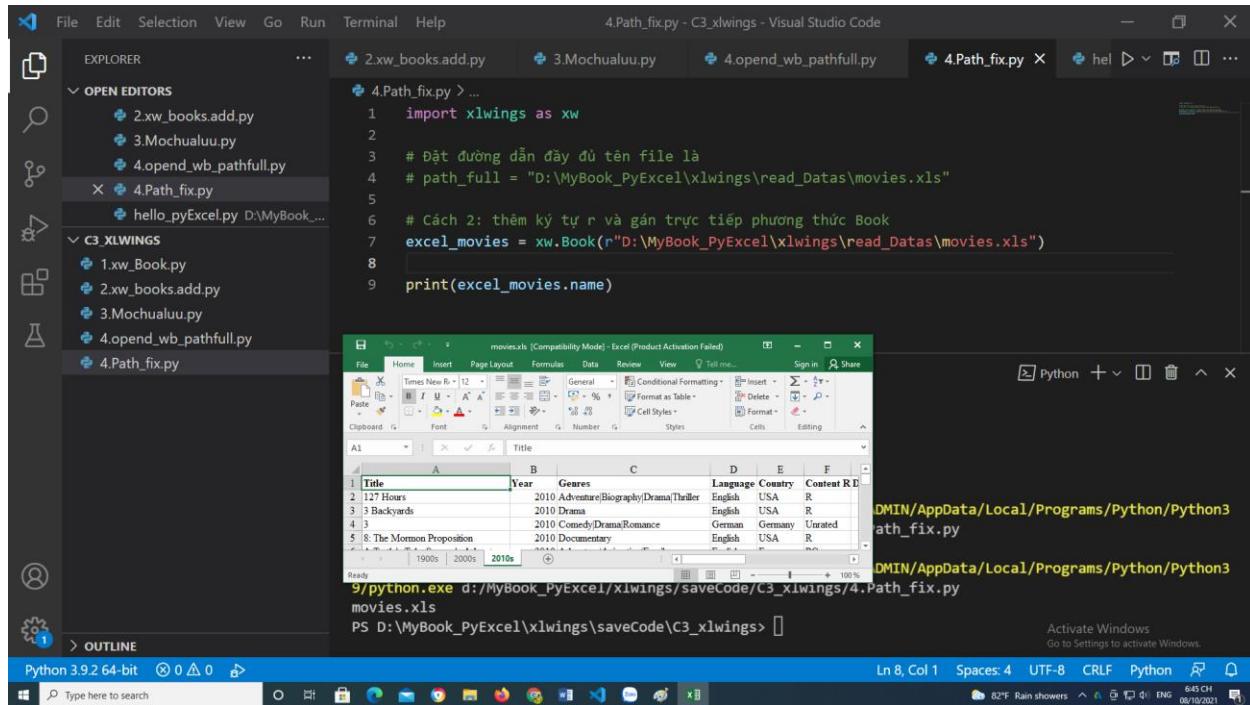
print(Excel_movies.name)

```

Tải file sửa lỗi code cách 2:

https://github.com/VuNghiXuan/giaotrinhPythonExcel/blob/main/C3_xlwings/4.Path_fix.py

Sau khi sửa lỗi thành công file “movies.xls” được mở như hình bên dưới



Khi đã thực hành đến đây, bạn đã biết cách đọc và mở 1 file Excel bảng tính bằng nhiều cách. Tùy theo tình huống mà bạn vận dụng cho phù hợp với công việc, dự án của mình.

5. Lưu, đóng Workbook và thoát App

Để thuận tiện cho việc quản lý các file một cách khoa học, cũng như so sánh được kết quả đầu vào và đầu ra dữ liệu, bạn cần tạo cho mình 2 folder (01 folder dùng trong đó chứa các file input bạn cần đọc vào [file “movies.xls”], folder còn lại là output đầu ra chứa các file sau khi xử lý bằng code và lưu lại).

Code:

```
import xlwings as xw

# Khai báo đường dẫn cụ thể file input và output;

# Đường dẫn file input (file "movies.xls" chứa trong thư mục "read_Datas")
path_input = 'D:\\MyBook_PyExcel\\xlwings\\read_Datas\\movies.xls'

# Đường dẫn file output, file lưu (file "C3_save_movies.xls" chứa trong thư mục "read_Datas")
path_output = 'D:\\MyBook_PyExcel\\xlwings\\save_Datas\\C3_save_movies.xls'

# Đặt tên biến app = App Excel
app = xw.App()

# Mở đọc file input (movies.xls)
wb = app.books.open(path_input)

# Lưu file output (C3_save_movies.xls)
wb.save(path_output)

# In ra thông tin file output
print(f'Tên bảng tính là: {wb.name}\nĐường dẫn file: "{wb.fullname}"')

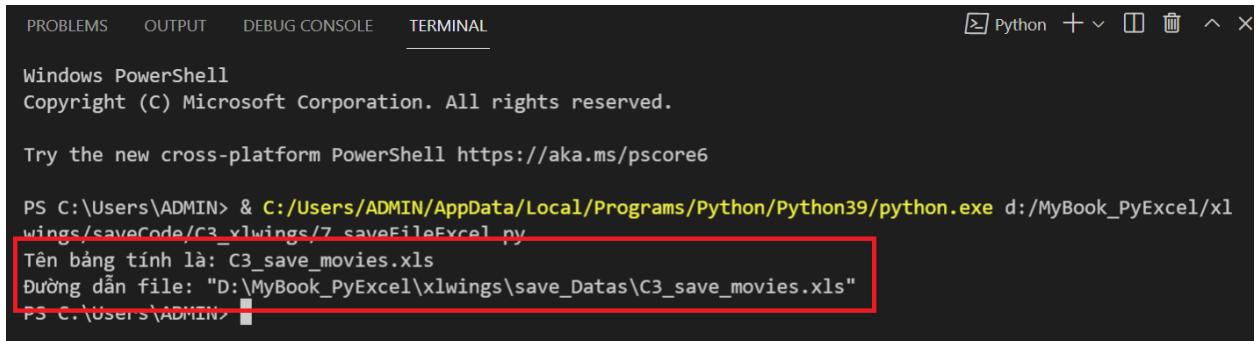
# Đóng workbook
wb.close()

# Thoát khỏi 1 ứng dụng Excel hiện hành.
app.quit()
```

Tải code:

https://github.com/VuNghiXuan/giaotrinhPythonExcel/blob/main/C3_xlwing s/7.saveFileExcel.py

Kết quả thực thi code:

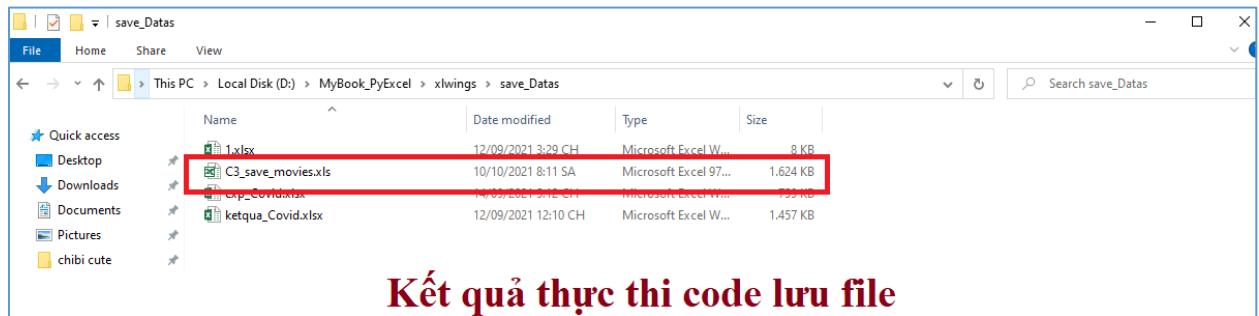


```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\ADMIN> & C:/Users/ADMIN/AppData/Local/Programs/Python/Python39/python.exe d:/MyBook_PyExcel/xlwings/saveCode/C3_xlwings/7_saveFileExcel.py
Tên bảng tính là: C3_save_movies.xls
Đường dẫn file: "D:\MyBook_PyExcel\xlwings\save_Datas\C3_save_movies.xls"
PS C:\Users\ADMIN>
```

Kết quả in ra file Output và đường dẫn trên cửa sổ Terminal



Kết quả thực thi code lưu file

6. Phần ôn tập đối tượng xlwings

6.1. Tại sao phải gán tên biến?

Tên biến là để đặt tên cho đối tượng đó (gắn liền 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 không, có học giỏi hay không?)

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*). Mục đích gắn tên (đặt tên biến) để bạn dễ dàng gọi, xử lý và điều khiển chúng bất cứ lúc nào.

```
import xlwings as i_love_PyExcel
wb = i_love_PyExcel.Book('code_tinh_camay.xls')
print("Tên workbook này là:", wb.name)
```

Như vậy trong phạm vi đoạn code trên, 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.xls`”.

6.2. Không đặt tên và gán biến

Câu hỏi: Không đặt tên và gán biến có được không? 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? Bạn sẽ tốn rất nhiều thời gian cho việc này nên tốt nhất bạn cứ đặt cho nó 1 cái tên.

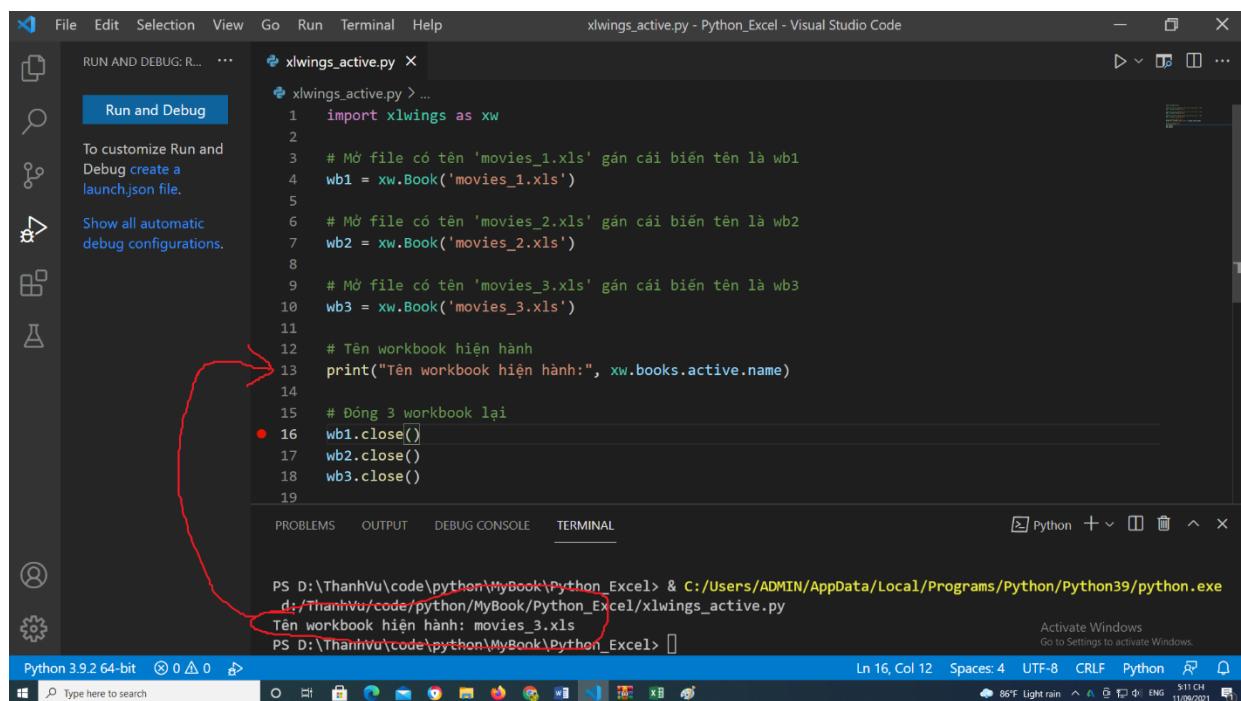
7. Phần giải thích thêm về thuộc tính active

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

Lấy 1 ví dụ: tôi 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 hay đang được chọn.

➤ Còn trong bảng tính Excel thì sao?

Mình mở 3 bảng tính gồm: movies_1, movies_2, movies_3



The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** Shows a folder structure with 'xlwings_active.py' highlighted.
- Code Editor:** Displays the Python code:

```
xlwings_active.py > ...
1 import xlwings as xw
2
3 # Mở file có tên 'movies_1.xls' gán cái biến tên là wb1
4 wb1 = xw.Book('movies_1.xls')
5
6 # Mở file có tên 'movies_2.xls' gán cái biến tên là wb2
7 wb2 = xw.Book('movies_2.xls')
8
9 # Mở file có tên 'movies_3.xls' gán cái biến tên là wb3
10 wb3 = xw.Book('movies_3.xls')
11
12 # Tên workbook hiện hành
13 print("Tên workbook hiện hành:", xw.books.active.name)
14
15 # Đóng 3 workbook lại
16 wb1.close()
17 wb2.close()
18 wb3.close()
19
```
- Terminal:** Shows the command run in the terminal: `PS D:\ThanhVu\code\python\MyBook\Python_Excel> & C:/Users/ADMIN/AppData/Local/Programs/Python/Python39/python.exe d:\ThanhVu\code\python\MyBook\Python_Excel\xlwings_active.py`. The output shows: `Tên workbook hiện hành: movies_3.xls`.
- Status Bar:** Shows system information like Python 3.9.2 64-bit, battery level, and network status.

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. Tức là bảng tính nào mở ra sau cùng sẽ là bảng tính active (hiện hành).

The screenshot shows a Visual Studio Code interface with two Excel windows and a Python Debug Console.

- Excel Windows:**
 - movies.xls [Compatibility Mode] - Excel (Prod...)**: This window contains a table with columns: Title, Year, Genres, and Language. It lists various movies from 1916 to 2010, categorized by year and genre.
 - movies_3.xls [Compatibility Mode] - Excel (Prod...)**: This window also contains a table with columns: Title, Year, Genres, and Language. It lists the same set of movies as the first window.
- Python Debug Console:**

```
Backend TkAgg is interactive backend. Turning interactive mode on.
```
- Taskbar:**
 - Python 3.9.2 64-bit
 - Type here to search
 - Windows Start button
 - Icons for File Explorer, Mail, Edge, etc.
 - System tray showing battery level (86%), language (ENG), and date (11/09/2021)

8. Ôn tập đối tượng workbook

8.1. Nhập vào thư viện xlwings

```
import xlwings as i_love_PyExcel  
  
wb = i_love_PyExcel.Book('code_tinh_camay.xlsm')  
print("Tên workbook này là:", wb.name)
```

➤ *Giải thích lại code:*

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 tên là "wb". Trong đó, Book là 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 thông dịch các dòng code 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:

8.2. 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()

```

8.3. Cách đọc 1 file Excel có tên là “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()
```

9. Bài tập phần đối tượng workbook:

- **Đề bài:** Hãy tạo ra 10 workbook mới đặt tên file từ 1 → 10, sau đó lưu file tại folder chứa script 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 quả sẽ không thay đổi. Bạn google để tìm hiểu thêm về vòng lặp for.

Kết quả in ra Terminal như sau:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Σ Python + ▾ ^

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\ThanhVu\code\python\MyBook\Python_Excel> & C:/Users/ADMIN/AppData/Local/Programs/Python/Python39/python.exe d:/ThanhVu/code/python/MyBook/Python_Excel/giaidapCode.py
Số n được đếm là: 0
Số n được đếm là: 1
Số n được đếm là: 2
Số n được đếm là: 3
Số n được đếm là: 4
Số n được đếm là: 5
Số n được đếm là: 6
Số n được đếm là: 7
Số n được đếm là: 8
Số n được đếm là: 9
```

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 (Đoạn 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 tên file sẽ = book+1, tức
là 0+1=1, đổi sang bằng chữ là str(book+1) và thêm đuôi fil
e sẽ là "1.xlsx"

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

+ Lần cuối cùng, book = 9 tương tự file sẽ là "9.xlsx"
-> Kết thúc vòng lặp. range trong Python không 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 (không dùng code)
    for app in xw.apps:
        app.quit()

```

10. 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.

10.1. Dữ liệu để thực hành

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://facebook.com/groups/1015080755701155>

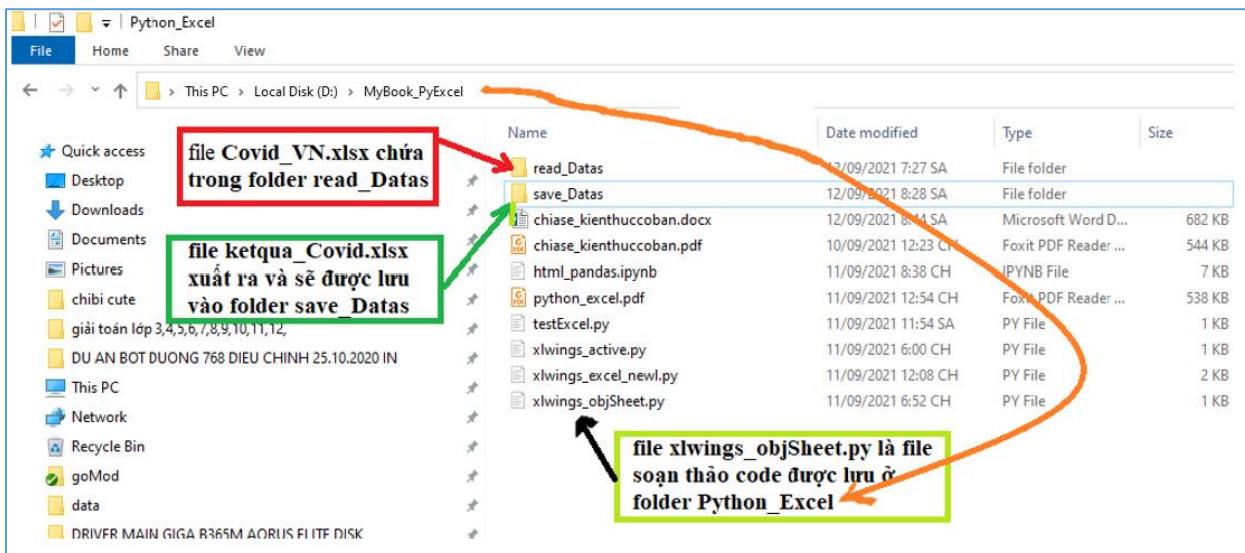
Để thuận tiện mình đã tải file này về trên máy mình và lưu nó tại ổ đĩa D, đường dẫn đầy đủ là: **D:\ MyBook_PyExcel\ read_Datas\Covid_VN.xlsx**

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 là file **ketqua_Covid.xlsx** (*file này xuất ra khi xử lý code sẽ nói sau phần dưới*).



10.2. Đọc file Excel

Quay lại VSC ta nhập 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 "rea
d_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 bạn 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:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\ADMIN> & C:/Users/ADMIN/AppData/Local/Programs/Python/Python39/python.exe d:/ThanhVu/code/python/Myok/Python_Excel/xlwings_objSheet.py
Kết quả đối tượng sh_Names là: Sheets([<Sheet [Covid_VN.xlsx]Số liệu>, <Sheet [Covid_VN.xlsx]Bang_Copy>])
PS C:\Users\ADMIN>
```

Kết quả in ra đọc hơi khó hiểu, nhưng cũng nhìn thấy được tên file là: “Covid_VN.xlsx”, còn “Số liệu” và “Bang_Copy” là tên của các Sheets đó.

10.3. Lấy tên sheet

➤ *Bước tiếp 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 code như sau:

```
import xlwings as xw

# Khởi tạo file Excel bằng modul App
app = xw.App(visible=True, add_book=False) # code này gắn
modul App với cái tên là app

# Đọc file Excel có tên là "Covid_VN.xlsx" từ folder "rea
d_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 p
hím 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 không 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**”

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Python + ⌂ ⌂ ⌂ ⌂

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\ADMIN> & C:/Users/ADMIN/AppData/Local/Programs/Python/Python39/python.exe d:/ThanhVu/code/python/MyBook/Python_Excel/xlwings_objSheet.py
Đối tượng sh_Names là: Sheets([<Sheet [Covid_VN.xlsx]Số liệu>, <Sheet [Covid_VN.xlsx]Bang_Copy>])
Tên các sheets là: Số liệu
Tên các sheets là: Bang_Copy
PS C:\Users\ADMIN>

```

Activate Windows
Go to Settings to activate Windows.

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 -----"

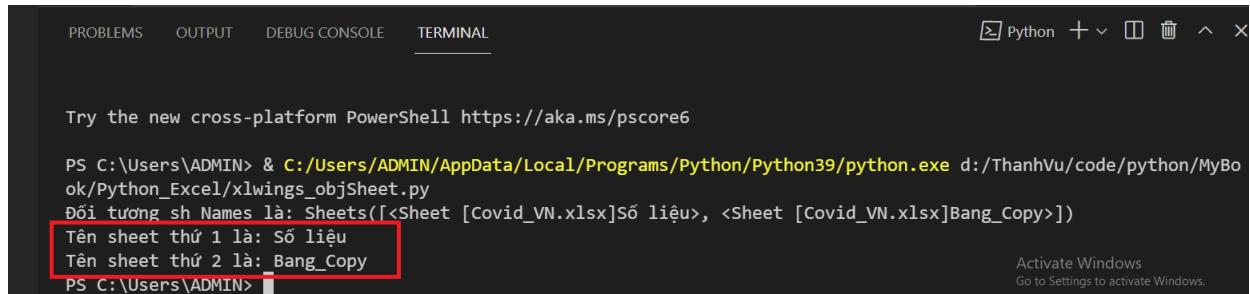
# lưu file "ketqua_Covid.xlsx" tại folder "save_Data"
wb.save(r'D:\MyBook_PyExcel\save_Datas\ketqua_Covid.xlsx'
)

# Đoạn code này đóng file Excel
wb.close()

# Đoạn code này tắt đối tượng app
app.quit()

```

Kết quả được in ra tại Terminal



```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Try the new cross-platform PowerShell https://aka.ms/pscore6
PS C:\Users\ADMIN> & C:/Users/ADMIN/AppData/Local/Programs/Python/Python39/python.exe d:/ThanhVu/code/python/MyBook/Python_Excel/xlwings_objSheet.py
Đổi tương sh Names là: Sheets([<Sheet [Covid_VN.xlsx]Số liệu>, <Sheet [Covid_VN.xlsx]Bang_Copy>])
Tên sheet thứ 1 là: Số liệu
Tên sheet thứ 2 là: Bang_Copy
PS C:\Users\ADMIN>

```

- 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
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

The screenshot shows a terminal window with the following content:

```
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\MyBook_PyExcel> & C:/Users/ADMIN/AppData/Local/Programs/Python/Python39/python.exe d:/MyBook_PyExcel/xlwings_objSheet.py
Tổng số sheets trước khi thêm là: 1
Tổng số sheets sau khi thêm là: 2
Tên sheet thứ 1 là: VuNghixuan
Tên sheet thứ 2 là: Số liệu
PS D:\MyBook_PyExcel>
```

The output is highlighted with a red box around the lines: "Tổng số sheets trước khi thêm là: 1", "Tổng số sheets sau khi thêm là: 2", "Tên sheet thứ 1 là: VuNghixuan", and "Tên sheet thứ 2 là: Số liệu".

➤ 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 (**phản 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 bạn hãy nhìn qua hình để hiểu rõ vị trí mình tìm để sửa lỗi:

```

xlwings_objSheet.py - MyBook_PyExcel - Visual Studio Code
xlwings_objSheet.py X hello.py hello_pyExcel.py chk_versionXlwings.py book_Python_ob D ...
RUN AND DEBUG: R... ...
Run and Debug
To customize Run and Debug create a launch.json file.
Show all automatic debug configurations.
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Python + ⚡ Python - x
MyBook_PyExcel/xlwings_objSheet.py
Tổng số sheets trước khi thêm là: 2
Traceback (most recent call last):
  File "C:\Users\ADMIN\AppData\Local\Temp\gen_py\3.9\00020813-0000-0000-C000-000000000046x0x1x9.py", line 3533, in __iter__
    ob = self._oleobj_.InvokeTypes(-4,LCID,2,(13, 10),())
pwintypes.com_error: (-2147418111, 'Call was rejected by callee.', None, None)

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "d:\MyBook_PyExcel\xlwings_objSheet.py", line 22, in <module>
    sh_new = sh_Names.add("VuNghixuan")
  File "C:\Users\ADMIN\AppData\Local\Programs\Python\Python39\lib\site-packages\xlwings\main.py", line 3901, in add

```

➤ *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**”

Lưu ý nhỏ: Trong phần này thay vì file “**Covid_VN.xlsx**”, bạn có thể sử dụng 1 file khác thay vào cũng được, mình sử dụng file có tên là “**objSheet.xlsx**” để làm ví dụ minh họa.

Code:

```

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 "objSheet.xlsx" từ folder "read_Data"
wb = app.books.open(r'D:\MyBook_PyExcel\read_Datas\objSheet.xlsx')

" -----> 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
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:

listName =[sh.name for sh in sh_Names]

Trả về tên sheet Cấu trúc hoạt động như vòng lặp for

Cho sh chạy trong list sh_Names, mỗi lần chạy qua 1 phần tử là trả về tên của sheet đó (sh: là 1 Sheet; sh_Names: là tất cả các Sheets, sh.name: hàm name trả về tên sheet đó)

10.4. Đổi 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 đổi 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"

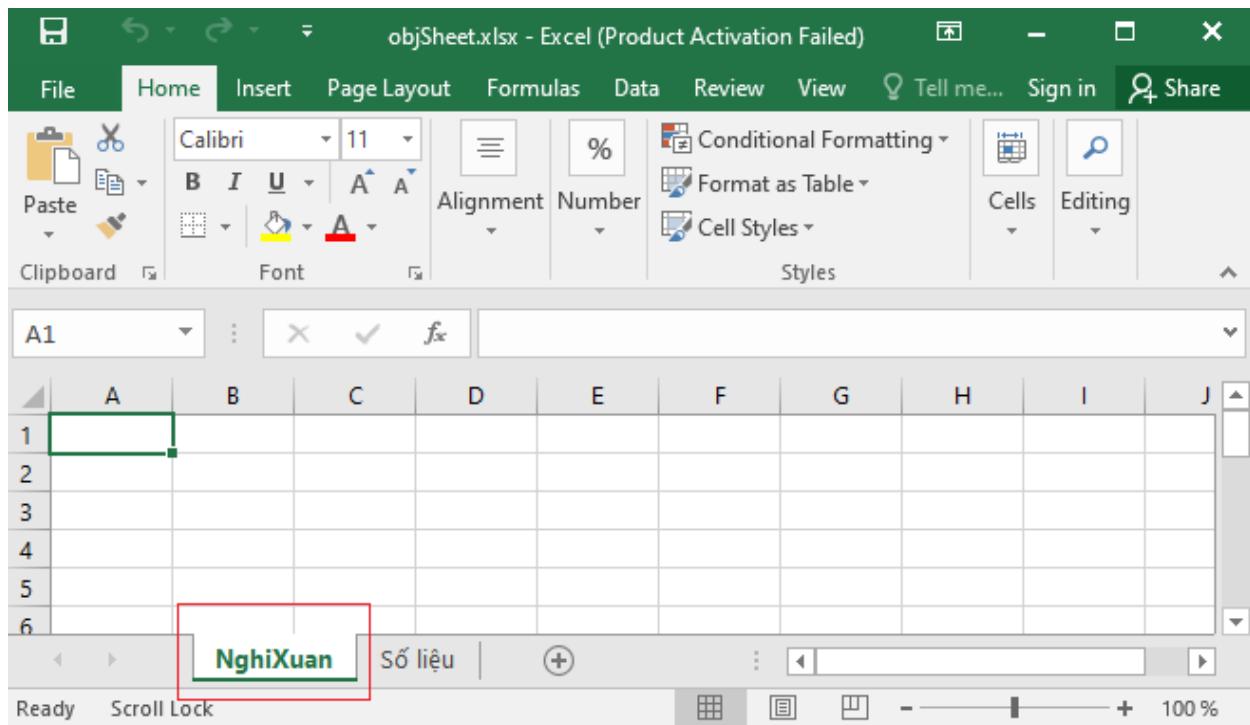
" Kết thúc tìm hiểu đổi tượng Sheet <-----"

# lưu file
wb.save(r'D:\MyBook_PyExcel\read_Datas\objSheet.xlsx')

# Đoạn code này đóng file Excel
wb.close()

# Đoạn code này tắt đổi 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 thức phần này.

10.5. 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: Không 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

1	Range	Range
2	Range	Range
3	Range	Range
4		
5		
6		

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:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL Python + ⌂ ⌂ ⌂ X

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\MyBook_PyExcel\xlwings> & C:/Users/ADMIN/AppData/Local/Programs/Python/Python39/python.exe d:/MyBook_PyExcel/xlwings/xlwings_valuesOfColRow.py
Giá trị của cột 1 là: Toán
Giá trị của cột 2 là: Lý
Giá trị của cột 3 là: Hóa
Giá trị của cột 4 là: Sinh
Giá trị của cột 5 là: Ngoại Ngữ
PS D:\MyBook_PyExcel\xlwings>
```

➤ **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:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL Python + ⌂ ⌂ ⌂ X

PS D:\MyBook_PyExcel\xlwings> & C:/Users/ADMIN/AppData/Local/Programs/Python/Python39/python.exe d:/MyBook_PyExcel/xlwings/xlwings_valuesOfColRow.py
Giá trị của hàng thứ 1 là: [1]
Giá trị của hàng thứ 2 là: [2]
Giá trị của hàng thứ 3 là: [3]
Giá trị của hàng thứ 4 là: [4]
Giá trị của hàng thứ 5 là: [5]
PS D:\MyBook_PyExcel\xlwings>
```

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:
')

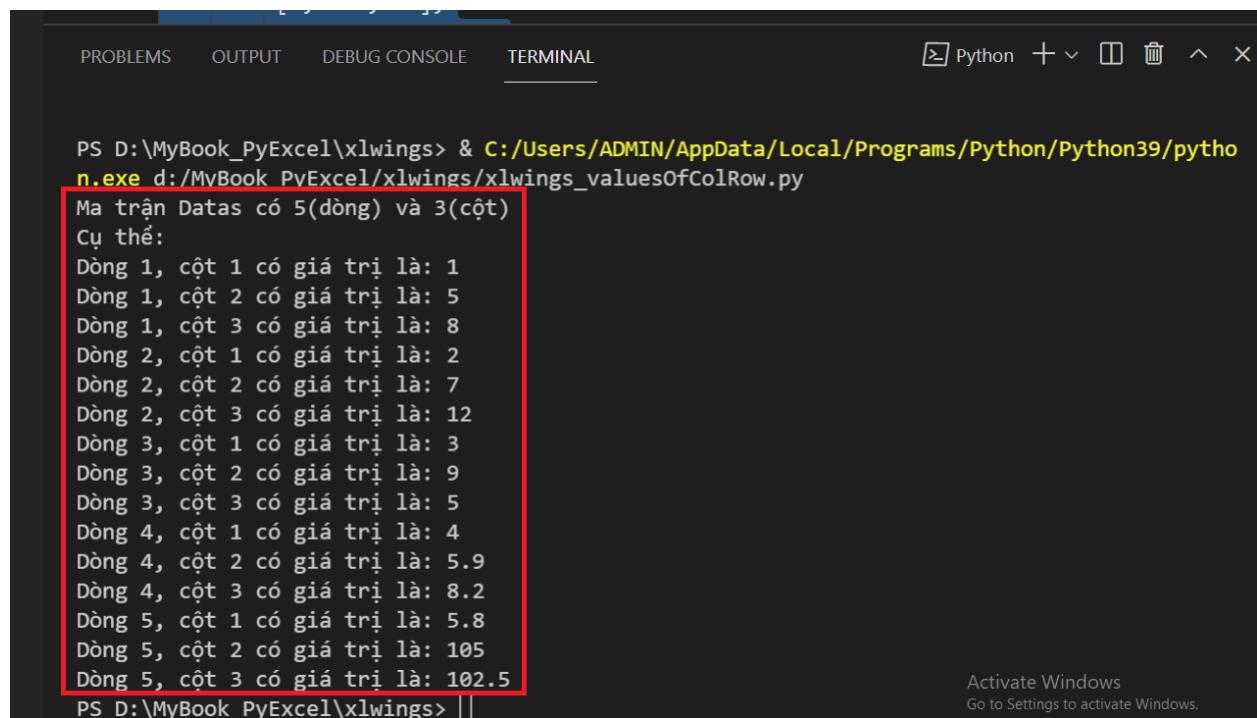
for i_row in range(total_rows):
    for i_col in range(total_columns):
        print(f'Dòng {i_row+1}, cột {i_col+1} có giá trị là: {datas[i_row][i_col]}')

```

Tải file code:

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

Kết quả:



```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Python + ▾ ⌂ ⌄ ^ ×

PS D:\MyBook_PyExcel\xlwings> & C:/Users/ADMIN/AppData/Local/Programs/Python/Python39/python.exe d:/MyBook_PyExcel/xlwings/xlwings_valuesOfColRow.py
Ma trận Datas có 5(dòng) và 3(cột)
Cụ thể:
Dòng 1, cột 1 có giá trị là: 1
Dòng 1, cột 2 có giá trị là: 5
Dòng 1, cột 3 có giá trị là: 8
Dòng 2, cột 1 có giá trị là: 2
Dòng 2, cột 2 có giá trị là: 7
Dòng 2, cột 3 có giá trị là: 12
Dòng 3, cột 1 có giá trị là: 3
Dòng 3, cột 2 có giá trị là: 9
Dòng 3, cột 3 có giá trị là: 5
Dòng 4, cột 1 có giá trị là: 4
Dòng 4, cột 2 có giá trị là: 5.9
Dòng 4, cột 3 có giá trị là: 8.2
Dòng 5, cột 1 có giá trị là: 5.8
Dòng 5, cột 2 có giá trị là: 105
Dòng 5, cột 3 có giá trị là: 102.5
PS D:\MyBook_PyExcel\xlwings> ||

```

Activate Windows
Go to Settings to activate Windows.

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

Code:

```
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: Không 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:

	A	B	C	D	E	F	G	H	I	J
1	Toán	Lý	Hóa	Sinh	Ngoại Ngữ					
2		1								
3		2								
4		3	1	5	8					
5		4	2	7	12					
6		5	3	9	5					
7			4	5,9	8,2					
8			5,8	105	102,5					
9										

10.6. Đọc dữ liệu từ bảng tính Excel

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

STT	Họ và tên	Toán	Lý	Hóa	
1	Thiên Văn	8	9	10	
2	Lỗ Chí	6	5	2	
3	Trần Nhất	2	10	8	
4	Tôn Khổng	8	6	7	

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

Code:

```
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]
```

```

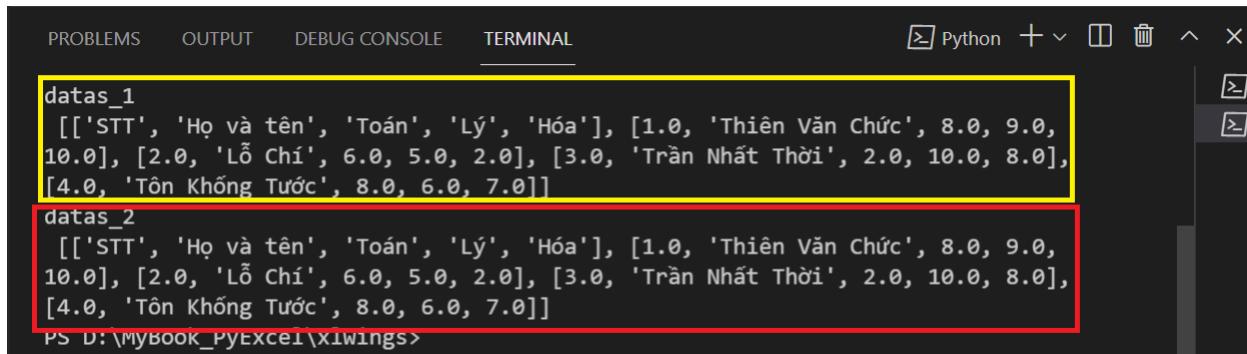
# Cách 1: Đọc dữ liệu bằng địa chỉ
datas_1 = sh.range("A1:E5").value

# 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.



```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Python + ×

datas_1
[['STT', 'Họ và tên', 'Toán', 'Lý', 'Hóa'], [1.0, 'Thiên Văn Chức', 8.0, 9.0, 10.0], [2.0, 'Lỗ Chí', 6.0, 5.0, 2.0], [3.0, 'Trần Nhất Thời', 2.0, 10.0, 8.0], [4.0, 'Tôn Khổng Tước', 8.0, 6.0, 7.0]]
datas_2
[['STT', 'Họ và tên', 'Toán', 'Lý', 'Hóa'], [1.0, 'Thiên Văn Chức', 8.0, 9.0, 10.0], [2.0, 'Lỗ Chí', 6.0, 5.0, 2.0], [3.0, 'Trần Nhất Thời', 2.0, 10.0, 8.0], [4.0, 'Tôn Khổng Tước', 8.0, 6.0, 7.0]]
PS D:\MyBOOK_PyExcel\X1wings>

```

10.7. 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 dưới.

Bạn tải file data gián đoạn tại địa chỉ sau:

https://github.com/VuNghiXuan/Mybooks/blob/main/read_Datas/1.BangDieu.xlsx

	A	B	C	D	E	F	G	H	I	J
1	STT	Họ và tên	Toán	Lý	Hóa			Tổng	Điểm TB	
2	1	Thiên Văn	8	9	10			27	9	
3	2	Lỗ Chí	6	5	2			13	4,333333	
4	3	Trần Nhất	2	10	8			20	6,666667	
5	4	Tôn Khổng	8	6	7			21	7	
6										
7		6 Thùy Linh	10	9	9,5			28,5	9,5	
8										

Bảng dữ liệu bị gián đoạn tại dòng số 6 và cột "F", "G"

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

The screenshot shows a terminal window with the following content:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\ Open file in editor (ctrl + click) C:/Users/ADMIN/AppData/Local/Programs/Python/Python39/python.exe d:/MyBook_PyExcel/xlwings/xlwings_readDataToExcel.py
datas_2
[['STT', 'Họ và tên', 'Toán', 'Lý', 'Hóa'], [1.0, 'Thiên Văn Chức', 8.0, 9.0, 10.0],
[2.0, 'Lỗ Chí', 6.0, 5.0, 2.0], [3.0, 'Trần Nhất Thời', 2.0, 10.0, 8.0], [4.0, 'Tôn K
hông Tước', 8.0, 6.0, 7.0]]
PS D:\myBook_PyExcel\xlwings>
```

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 sót 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 áy 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.

Phương thức expand()

Tìm dòng cuối cùng có chứa dữ liệu

Tìm cột cuối cùng chưa dữ liệu

1048576 Đây là dòng cuối của bảng tính

STT	Họ và tên	Tiến	Lý	Hoá	G	H	I	J	K	L	XFB	XFC	XFD
					Tổng			Điểm TB					16384
1	Thiên Văn	8	9	10		27		9					
2	Lỗ Chí	6	5	2		13		4,333333					
3	Trần Nhất	2	10	8		20		6,666667					
4	Tôn Khôn	8	6	7		21		7					
6	Thùy Linh	10	9	9,5		28,5		9,5					
7													
1048571													
1048572													
1048573													
1048574													
1048575													
1048576	Đây là dòng cuối của bảng tính												

Phương thức tìm dòng cuối, cột cuối có chứa dữ liệu (đường màu xanh và màu tím)

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**.

Code:

```
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:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS D:\MyBook_PyExcel\xlwings> & C:/Users/ADMIN/AppData/Local/Programs/Python/Python39/python.exe d:/MyBook_PyExcel/xlwings/xlwings_readDataToExcel_With_LastRowLastCol.py
Dòng cuối cùng bảng tính là: 1048576
Cột cuối cùng bảng tính là: 16384
PS D:\MyBook_PyExcel\xlwings>

```

Chú ý: Kết quả trên là dòng cuối, cột cuối của bảng tính excel, 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:

Code:

```

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(lc_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)

```

Tải code:

https://github.com/VuNghiXuan/Mybooks/blob/main/Python_Excel/xlwings/readDataToExcel_With_lastrowLastCol.py

Kết quả chạy code:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\MyBook_PyExcel\xlwings> & C:/Users/ADMIN/AppData/Local/Programs/Python/Python39/python.exe d:/MyBook_PyExcel/xlwings/xlwings_readDataToExcel_With_lastrowLastCol.py
Đòng cuối cùng bảng tính là: 1048576
Cột cuối cùng bảng tính là: 16384
Đòng cuối cùng có chứa dữ liệu là: 7
Cột cuối cùng có chứa dữ liệu là: 10
Bảng dữ liệu có địa chỉ là: $A$1:$J$7
Giá trị data là:
[[{'STT': 'Họ và tên', 'Toán': 'Lý', 'Hóa': None, 'None': None, 'Tổng': 'Điểm TB'}, {'STT': 1.0, 'Toán': 'Thiên Văn Chức', 'Lý': 8.0, 'Hóa': 9.0, 'None': 10.0, 'None': 27.0, 'Tổng': 9.0}, {'STT': 2.0, 'Toán': 'Lỗ Chí', 'Lý': 6.0, 'Hóa': 5.0, 'None': 2.0, 'None': None, 'Tổng': 13.0}, {'STT': 3.0, 'Toán': 'Trần Nhất Thời', 'Lý': 2.0, 'Hóa': 10.0, 'None': 8.0, 'None': None, 'Tổng': 20.0, 'None': 6.666666666666667}, {'STT': 4.0, 'Toán': 'Tôn Không Tước', 'Lý': 8.0, 'Hóa': 6.0, 'None': 7.0, 'None': 21.0, 'Tổng': 7.0}, {'STT': 5.0, 'Toán': 'None', 'Lý': None, 'Hóa': None, 'None': None, 'Tổng': None}, {'STT': 6.0, 'Toán': 'Thùy Linh', 'Lý': 10.0, 'Hóa': 9.0, 'None': 9.5, 'Tổng': 28.5}], [{"text": "Kết quả chạy code tìm được dòng cuối, cột cuối và địa chỉ bảng dữ liệu (bao gồm cả phần gián đoạn)"}]

```

10.8. Cách thay đổi chiều dữ liệu ghi bảng tính

Trong quá trình xử lý dữ liệu bạn có nhu cầu cần phải đổi chiều dữ liệu cột thành dòng và ngược lại thì **xlwings** cung cấp cho bạn hàm **options** với tham số **transpose = True** để thực hiện thay đổi 1 cách nhanh chóng.

Đầu tiên, bạn nên dùng code để mở 1 bảng tính xem dữ liệu của mình cần lấy, mình thực hiện mở file “**1.BangDiem.xlsx**”.

Code:

```
import xlwings as xw

path_input = 'D:\\MyBook_PyExcel\\xlwings\\read_Datas\\1.BangDiem.xlsx'

# Mở workbook để xem trước dữ liệu
xw.Book(path_input)
```

Sau đó comment lại các dòng này (đừng tắt workbook) và thực hiện kích hoạt workbook này bằng **active**, để thuận tiện thao tác mà không phải tắt mở file lại.

Code:

```
import xlwings as xw

# Đoạn comment lại"
# path_input = 'D:\\MyBook_PyExcel\\xlwings\\read_Datas\\1.BangDiem.xlsx'

# # Mở workbook để xem trước dữ liệu
# xw.Book(path_input)
"-----"

# Sau khi wb được mở comment 2 dòng trên và thực hiện code sau
wb = xw.books.active
sh1 = wb.sheets('DS_xeploai')
rng1 = sh1.range('A1:E5')

# Kiểm tra file có tên sheet là "test", nếu không thì tạo sheets("test")
```

```
"Đoạn code trong vòng for này giúp bạn không bị lỗi khi chạy code nhiều lần  
với tên sheet 'test' đã tạo trước đó"  
  
for sh in wb.sheets:  
    if sh.name == "test":  
        sh_test = sh.name  
        print("Đã tồn tại sheet có tên: 'test'")  
        break  
    else:  
        # Tạo 1 bảng tính có tên là "test"  
        sh_test = wb.sheets.add("test")  
  
    # Ghi dữ liệu theo chiều gốc của bảng tính  
    sh_test.range("A1").value = rng1.value  
  
    # Đổi chiều dòng thành cột và ngược lại  
    sh_test.range("A8").options(transpose=True).value = rng1.value
```

Tải code:

https://github.com/VuNghiXuan/giaotrinhPythonExcel/blob/main/C3_xlwing/s/11.transpose_range.py

Kết quả thực thi code:

STT	Họ và tên	Toán	Lý	Hóa
1	Thiên Văn	8	9	10
2	Lỗ Chí	6	5	2
3	Trần Nhất	2	10	8
4	Tôn Khổng	8	6	7
5				
6				
7				
8	STT	1	2	3
9	Họ và tên	Thiên Văn	Lỗ Chí	Trần Nhất
10	Toán	8	6	2
11	Lý	9	5	10
12	Hóa	10	2	8
13				

STT	1	2	3	4
1	Thiên Văn	Lỗ Chí	Trần Nhất	Tôn Khổng
2	8	6	2	8
3	9	5	10	6
4	10	2	8	7

10.9. Cách ghi dữ liệu list 1D và vùng dữ liệu vào bảng tính

Code ghi dữ liệu vào bảng tính theo dòng

```
import xlwings as xw

"Đoạn comment lại"
# path_input = 'D:\\MyBook_PyExcel\\xlwings\\read_Datas\\1.BangDiem.xlsx'

# # Mở workbook để xem trước dữ liệu
# xw.Book(path_input)
"-----"

# Sau khi wb được mở comment 2 dòng trên và thực hiện code sau
wb = xw.books.active

# Kiểm tra file có tên sheet là "test", nếu không thì tạo sheets("test")
"Đoạn code trong vòng for này giúp bạn không bị lỗi khi chạy code nhiều lần với
tên sheet 'test' đã tạo trước đó"
# Kiểm tra file có tên sheet là "test", nếu không thì tạo sheets("test")
```

```

"Đoạn code trong vòng for này giúp bạn không bị lỗi khi chạy code nhiều lần với
tên sheet 'test' đã tạo trước đó"
sheetNames = ""
for sh in wb.sheets:
    # Nếu tên sheet không phải là "test"
    if sh.name != "test":
        sheetNames += sh.name
    else:
        # Nếu tên sheet là "test"
        sheetNames += sh.name
        print("Đã tồn tại sheet có tên: 'test'")
        break

# Kiểm tra sự tồn tại sheet test
if "test" not in sheetNames:
    sh_test = wb.sheets.add("test") #Tạo sheet("test") mới
else: sh_test = wb.sheets("test") # Gán tên sheet("test")

list_1D = [1,2,3]
datas = ['Nguyễn Văn A', 'Nguyễn Văn B']

# Cách 1: Ghi dữ liệu trên 1 dòng
sh_test.range("A1").value = list_1D
sh_test.range("A2").value = list_1D

# Cách 2: Ghi dữ liệu trên 1 dòng
sh_test.range("A3").value = datas

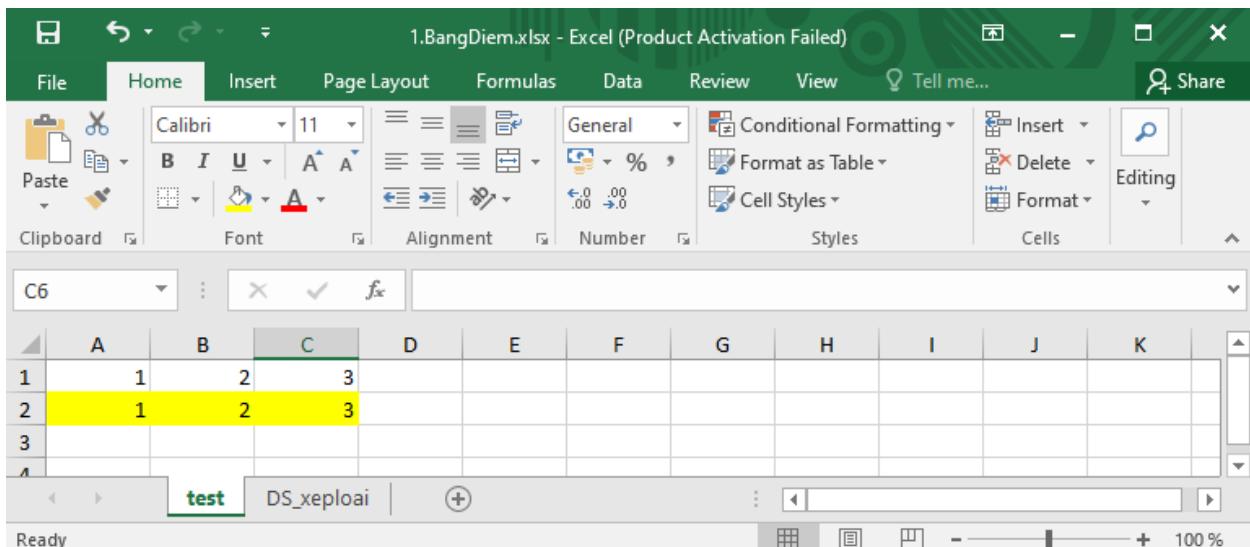
# Cách 3: Ghi theo vùng dữ liệu (range: nhiều cells)
sh_test.range("E1").options(ndim=1).value = sh_test.range('A1:C3').value

```

Tải code:

https://github.com/VuNghiXuan/giaotrinhPythonExcel/blob/main/C3_xlwing_s/12.ghi%201D.py

Kết quả thực thi code:



10.10. Cách ghi dữ liệu list 2D vào bảng tính

Code:

```
import xlwings as xw

# Đoạn comment lại
# path_input = 'D:\\MyBook_PyExcel\\xlwings\\read_Datas\\1.BangDiem.xlsx'

# # Mở workbook để xem trước dữ liệu
# xw.Book(path_input)
# -----

# Sau khi wb được mở comment 2 dòng trên và thực hiện code sau
wb = xw.books.active

# Kiểm tra file có tên sheet là "test", nếu không thì tạo sheets("test")
# Đoạn code trong vòng for này giúp bạn không bị lỗi khi chạy code nhiều lần
# với tên sheet 'test' đã tạo trước đó
sheetNames = ""
for sh in wb.sheets:
    # Nếu tên sheet không phải là "test"
    if sh.name != "test":
        sheetNames += sh.name
    else:
```

```
# Nếu tên sheet là "test"
sheetNames += sh.name
print("Đã tồn tại sheet có tên: 'test'")
break

# Kiểm tra sự tồn tại sheet test
if "test" not in sheetNames:
    sh_test = wb.sheets.add("test") #Tạo sheet("test") mới
else: sh_test = wb.sheets("test") # Gán tên sheet("test")

list_2D = [[4,5],
           [6,7],
           [8,9]]

# Ghi dữ liệu list 2D
sh_test.range("A2").value = list_2D
```

Tải code:

https://github.com/VuNghiXuan/giaotrinhPythonExcel/blob/main/C3_xlwing_s/13.ghiDulieu_2D.py

Kết quả thực thi code:

1									
2	4		5						
3	6		7						
4	8		9						
5									

10.11. Cách ghi công thức vào bảng tính

Mình có dữ liệu bảng tính như hình:

STT	Họ và tên	Toán	Lý	Hóa					
1	Thiên Văn	8	9	10					
2	Lỗ Chí	6	5	2					
3	Trần Nhất	2	10	8					
4	Tôn Khổng	8	6	7					

Tiếp theo, tính tổng điểm và điểm trung bình học sinh cho cột F cột G:

Code:

```
import xlwings as xw

"Đoạn comment lại"
# path_input = 'D:\\MyBook_PyExcel\\xlwings\\read_Datas\\1.BangDiem.xlsx'

# # Mở workbook để xem trước dữ liệu
# xw.Book(path_input)
"-----"

# Sau khi wb được mở comment 2 dòng trên và thực hiện code sau
wb = xw.books.active
sh = wb.sheets["DS_xeploai"]

# Ghi vào cột F, G tiêu đề cột
sh.range("F1").value = ['Tổng', 'Điểm TB']

# Điền công thức tính tổng ô F2
sh.range("F2").value = "=sum(C2:E2)"

# Điền công thức tính Trung bình ô G2
sh.range("G2").value = "=average(C2:E2)"

# Sao chép công thức ô F2:G2 cho các dòng bên dưới
func_rng = sh.range("F2:G2").formula
sh.range("F2:G5").formula = func_rng
```

Tải code:

https://github.com/VuNghiXuan/giaotrinhPythonExcel/blob/main/C3_xlwing_s/14.fomular_bangtinh.py

Kết quả thực thi code:

STT	Họ và tên	Toán	Lý	Hóa	Tổng	Điểm TB
1	Thiên Văn	8	9	10	27	9
2	Lỗ Chí	6	5	2	13	4,333333
3	Trần Nhất	2	10	8	20	6,666667
4	Tôn Khổng	8	6	7	21	7

10.12. Cách chèn dòng, cột bảng tính

Đây là các thao tác giống như bạn sử dụng trên bảng tính Excel, do đó trong quá trình code bạn nên thực hiện từng dòng 1 và xem kết quả thay đổi trên bảng tính.

```
import xlwings as xw

wb = xw.books.active
sh = wb.sheets["DS_xeploai"]

"Chèn dòng và cột bảng tính"
# sh.range("3:3").insert() # Chèn 1 dòng
sh.range("3:7").insert() # Chèn nhiều dòng
# sh.range("F:F").insert() # Chèn 1 cột
sh.range("F:G").insert() # Chèn nhiều cột
```

Tải code:

https://github.com/VuNghiXuan/giaotrinhPythonExcel/blob/main/C3_xlwings/14.1.Add_Col.text.py

10.13. Cách thêm, gộp, xóa nội dung trên bảng tính

Code:

```
import xlwings as xw

wb = xw.books.active
sh = wb.sheets["DS_xeploai"]

"Cách thêm nội dung";
sh.range('A1').value = "Thêm nội dung ô A1"
sh.range('A2').value = "Thêm nội dung ô A2"

# Nối nội dung ô A1 và ô A2 thành chuỗi mới
sh.range('A3').value = sh.range('A1').value + " " + sh.range('A2').value
sh.range('A4').value = "=Concatenate(A1,char(32),A2)" # char(32): khoảng
trắng
sh.range('A5').value = "=Concatenate(A1,char(10),A2)" # char(10): Xuống dòng
# ô A3, A4 cùng kết quả >>> "Thêm nội dung ô A1 Thêm nội dung ô A2"

"Xóa nội dung";
sh.range('A4').clear() #Xóa nội dung ô A4
```

Tải code:

https://github.com/VuNghiXuan/giaotrinhPythonExcel/blob/main/C3_xlwings/14.2%20addClearContent_Range.py

10.14. Thay đổi bộ cục, định dạng trên bảng tính Excel:

➤ *Phương thức API là gì?*

API là viết tắt của Application Programming Interface – phương thức trung gian kết nối các ứng dụng và thư viện khác nhau.

Nó cung cấp khả năng truy xuất đến một tập các hàm hay dùng, từ đó có thể trao đổi dữ liệu giữa các ứng dụng.

Thi thoảng vẫn có người làm tưởng API là một ngôn ngữ lập trình nhưng thực ra, API chỉ là các hàm hay thủ tục thông thường. Các hàm này được viết trên nhiều ngôn ngữ lập trình khác nhau

➤ *Cách xóa dòng, cột bảng tính*

Để thực hiện xóa dòng, cột trong bảng tính Excel, bạn phải phuong thức kết nối là API

Bạn mở lại bảng tính có kết quả sau khi thực thi code thêm công thức tính tổng và trung bình điểm phần trên, mình tiếp tục thực hành.

Đến đây bạn hãy nhập vào file **DeleteShiftDirection** chứa trong folder **constants** của thư viện **xlwings**

Code:

```
import xlwings as xw

# Nhập hàm từ thư viện xlwings, chứa trong folder
from xlwings.constants import DeleteShiftDirection

# Kích hoạt wb
wb = xw.books.active
sh = wb.sheets["DS_xeploai"]

# Xóa 1 dòng số 2
sh.range('2:2').api.Delete(DeleteShiftDirection.xlShiftUp)

# Xóa 1 cột F (cột tổng)
sh.range('F:F').api.Delete(DeleteShiftDirection.xlShiftUp)
```

Tải code:

https://github.com/VuNghiXuan/giaotrinhPythonExcel/blob/main/C3_xlwings/15.api.delRowCol.py

Kết quả thực thi code

STT	Họ và tên	Toán	Lý	Hóa	Điểm TB
2	Lỗ Chí	6	5	2	4,333333
3	Trần Nhất	2	10	8	6,666667
4	Tôn Khổng	8	6	7	7
5					
6					

➤ **Cách thay đổi, chỉnh sửa bối cảnh và định dạng dữ liệu bảng tính**

Đây là các thao tác chỉnh sửa bối cảnh giống như bạn sử dụng trên bảng tính Excel, do đó trong quá trình code bạn nên thực hiện từng dòng 1 và xem kết quả thay đổi trên bảng tính.

Code:

```
import xlwings as xw
from xlwings import constants

# Nhập hàm từ thư viện xlwings, chứa trong folder
from xlwings.constants import DeleteShiftDirection

# Kích hoạt wb
wb = xw.books.active
sh = wb.sheets["DS_xeploai"]

# Bỏ in đậm ô A1
sh.range('A1').api.Font.Bold = False
```

```

# In đậm dòng 1
sh.range('1:1').api.Font.Bold = True

"Tương tự bạn thực hiện tìm hiểu các đoạn code sau:";

sh.range('B:B').api.Font.Bold = True #In đậm 1 cột
sh.range('B:B').api.Font.Italic = True #In nghiêng 1 cột
sh.range('B:B').api.Font.Underline = True #In nghiêng 1 cột
sh.range('B:B').api.Font.Name = "Tahoma" #Thay kiểu chữ
sh.range('B:B').api.Font.Size = 15 #Thay kiểu chữ
sh.range('B:B').api.Font.ColorIndex = 5 #Thay màu chữ
sh.range('B:B').api.ShrinkToFit = True # Tự động giãn cho cột (số)
sh.range('B:B').api.WrapText = True # Tự động xuống dòng

" Canh chỉnh lề theo chiều ngang";
sh.range('B:B').api.HorizontalAlignment = constants.HAlign.xlHAlignLeft # Canh trái
sh.range('B:B').api.HorizontalAlignment = constants.HAlign.xlHAlignCenter # Canh giữa
sh.range('B:B').api.HorizontalAlignment = constants.HAlign.xlHAlignRight # Canh phải

" Canh chỉnh lề theo chiều dọc";
sh.range('B:B').api.VerticalAlignment = constants.VAlign.xlVAlignTop # Canh trên
sh.range('B:B').api.VerticalAlignment = constants.VAlign.xlVAlignCenter # Canh giữa
sh.range('B:B').api.VerticalAlignment = constants.VAlign.xlVAlignBottom # Canh dưới

" Xoay chữ theo chiều";
sh.range('B1').api.Orientation = constants.Orientation.xlVertical # Xoay chiều dọc
sh.range('B2').api.Orientation = constants.Orientation.xlHorizontal # Xoay chiều ngang

```

```

    sh.range('B3').api.Orientation = constants.Orientation.xlDownward # Xoay
xuống dưới
    sh.range('B4').api.Orientation = constants.Orientation.xlUpward # Xoay lên
trên

    "Thiết lập vùng in ấn"
sh.api.PageSetup.PrintArea = 'A1:E7' # Setup phạm vi in
sh.api.PageSetup.PrintTitleRows = '1:1' # Setup phạm vi in
sh.api.PageSetup.CenterHorizontally = True # Canh giữa theo chiều ngang
sh.api.PageSetup.CenterVertically = False # Canh giữa theo chiều dọc

# Header
sh.api.PageSetup.LeftHeader = "VuNghiXuan" # Thêm thông tin Bên trái header
# sh.api.PageSetup.CenterHeader = "Trang" # Thêm thông tin Giữa header
sh.api.PageSetup.RightHeader = "Python Excel" # Thêm thông tin Bên phải header

# Footer
sh.api.PageSetup.LeftFooter = "VuNghiXuan" # Thêm thông tin Bên trái Footer
# sh.api.PageSetup.CenterFooter = "Trang" # Thêm thông tin Giữa Footer
sh.api.PageSetup.RightFooter = "Python Excel" # Thêm thông tin Bên phải Footer

# Định dạng khổ trang A4: letter =1, A3 = 8, A4 = 9
# sh.api.PageSetup.PageSize = 9

# Chiều của trang: Portrait =1 (đứng), Lanscaps = 2 (ngang)
sh.api.PageSetup.Orientation = 2

sh.api.PrintOut(Preview = True, Copies =1) # Ché độ xem trước
# sh.api.PrintOut(Preview = True, Copies = 1, From = 1, To = 1) # In trang 1

    "Định dạng xuất --> DPF";
    # Tham khảo: https://docs.microsoft.com/en-
us/office/vba/api/Excel.workbook.exportasfixedformat
    # wb.api.ExportAsFixesFormat(0, 'new_FileName.pdf')
    # sh.api.ExportAsFixesFormat(0, 'name.dpf')

```

```

    "Một số thuộc tính không cần thực qua API";
sh.range('B:B').color = (255,255,0) # Thay màu nền
sh.range('B:B').autofit() # Tự động dãn cột vừa chữ
sh.range('F1:G1').merge() # Trộn 2 hay nhiều ô thành 1
sh.range('F1:G1').unmerge() # Bỏ trộn ô

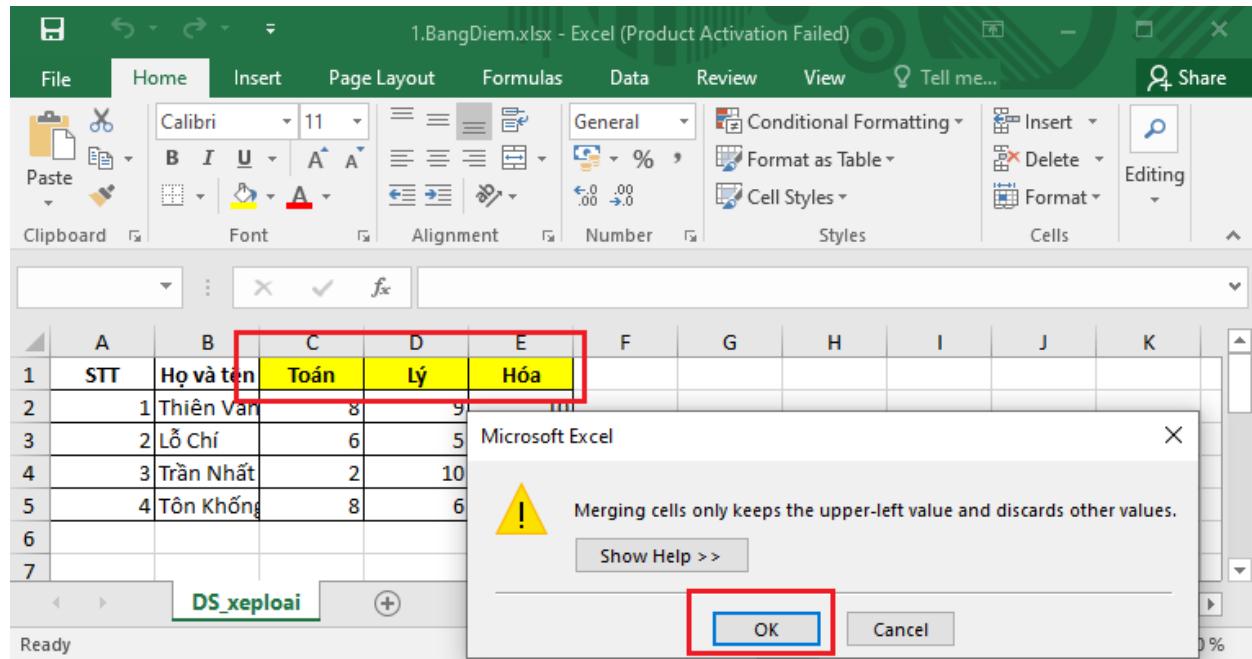
```

Tải code:

https://github.com/VuNghiXuan/giaotrinhPythonExcel/blob/main/C3_xlwing_s/16.api.dinhdang.py

➤ **Cách xử lý thông báo khi dùng gộp ô dữ liệu (merge) trên bảng tính**

Giả sử bạn mở lại bảng tính “1.BangDiem.xlsx” như sau:



Bây giờ bạn muốn dùng gộp 3 ô từ C1→E1 có giá trị lần lượt là “Toán”, “Lý”, “Hóa”. Khi bạn sử dụng dòng code `sh.range('C1:E1').merge()` để nhập các ô dữ liệu thành 1 ô, Excel sẽ có 1 cảnh báo đưa ra, nếu bạn nhấn OK thì Excel sẽ hiểu bạn chỉ cần dữ liệu cột đầu tiên bên trái tức là “Toán“. Kết quả dữ liệu của bạn như sau:

	A	B	C	D	E
1	STT	Họ và tên	Toán		
2	1	Thiên Văn	8	9	10
3	2	Lỗ Chí	6	5	2
4	3	Trần Nhất	2	10	8
5	4	Tôn Khổng	8	6	7
6					

Để khắc phục điều này bạn cần lấy dữ liệu 3 ô này lưu vào chỗ khác đồng thời xóa dữ liệu 3 ô này, sau khi thì hiện merge rồi mới dán kết quả vào ô này, viết code như sau:

Code:

```
import xlwings as xw
from xlwings import constants

# Nhập hàm từ thư viện xlwings, chứa trong folder
from xlwings.constants import DeleteShiftDirection

# Kích hoạt wb
wb = xw.books.active
sh = wb.sheets["DS_xeploai"]
rngCells = sh.range('C1:E1')

# Bước 1: Lưu dữ liệu vào 1 cái biến có giá trị 3 ô
valueCells = ', '.join(rngCells.value)

# Nước 2: Xóa dữ liệu để merge không có thông báo
rngCells.clear()

# Bước 3: Thực hiện trộn ô merge
rngCells.merge() # Trộn 2 hay nhiều ô thành 1

# Bước 4: Dán lại dữ liệu
```

```

rngCells.value = valueCells
rngCells.color = (255,255,0) # Cho màu vàng ô
rngCells.api.Font.Bold = True
rngCells.api.HorizontalAlignment = constants.HAlign.xlHAlignCenter # Canh
giữa

```

Tải code

https://github.com/VuNghiXuan/giaotrinhPythonExcel/blob/main/C3_xlwings/16b%20fixMergeError.py

Kết quả sau khi thực thi code:

STT	Họ và tên	Toán, Lý, Hóa
1	Thiên Văn	8 9 10
2	Lỗ Chí	6 5 2
3	Trần Nhất	2 10 8
4	Tôn Khổng	8 6 7
6		
7		

➤ *Cách ghi và xử lý kiểu ngày tháng trên bảng tính*

Sau đây là cách bạn định dạng ghi kiểu ngày tháng và ghi vào bảng tính Excel, so sánh ngày tháng và đánh dấu các ô có giá trị lớn hơn ngày hiện tại.

Code:

```

import xlwings as xw
from datetime import datetime as dt

wb = xw.books.active
sh = wb.sheets["DS_xeploai"]

from datetime import datetime

```

```

"Có các kiểu định dạng sau:"  

# sh.range('B1').options(numbers=str).value # Định dạng kiểu String  

# sh.range('C1').options(numbers=int).value # Định dạng integer  

# sh.range('D1').options(numbers=float).value # Định dạng float  

# sh.range('E1').options(dates=dt.date).value # Định dạng kiểu ngày tháng  
  

"Ví dụ định dạng và xử lý kiểu ngày tháng giữa Python và Bảng tính Excel"  

list_day = [['19/08/2020'], ['30/12/2022'], ['30/5/2021'], ['30/11/2021']]  
  

"Ghi định dạng kiểu ngày tháng vào bảng tính";  

sh.range('A11').options(dates=dt.date).value = list_day  
  

"Định dạng ngày tháng list Python";  

to_day = datetime.strptime('15/10/2021', '%d/%m/%Y')  
  

# So sánh các ô trong bảng tính và ngày hiện tại
for i_cell in range(len(list_day)):
    rng = sh.range(f'A{11 + i_cell}')
    covertRngTime = datetime.strptime(rng.value, '%d/%m/%Y')
    if covertRngTime > to_day:
        # Đánh dấu ô có giá trị > ngày '14/10/2021' trong bảng tính
        rng.color = (255,0,0)

```

Tải code:

https://github.com/VuNghiXuan/giaotrinhPythonExcel/blob/main/C3_xlwing_s/17.%20formatTypeData.py

➤ **Xử lý các giá trị lỗi trong bảng tính Excel (#NA, #DIV/0!, #VALUE!)**

Dữ liệu đầu vào của mình tại cột F có chứa các giá trị lỗi sau:

	A	B	C	D	E	F	G	H	I
1	STT	Họ và tên	Toán	Lý	Hóa	Tổng			
2	1	Thiên Văn	8	9	10	19			
3	2	Lỗ Chí	6	5	2	#VALUE!			
4	3	Trần Nhất	2	10	8	#REF!			
5	4	Tôn Khổng	8	6	7	#N/A			
6									
7									

Code:

```
import xlwings as xw

wb = xw.books.active
sh = wb.sheets["DS_xeploai"]

rng_value = sh.range('F1:F5').value
print("Giá trị trong list Python là:", rng_value)
```

Kết quả thực thi code:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Python + ×

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\MyBook_PyExcel\xlwings\saveCode\C3_xlwings> & C:/Users/ADMIN/AppData/Local/Programs/Python/Python39/python.exe d:/MyBook_PyExcel/xlwings/saveCode/C3_xlwings/18.duplicate_Data.py
Giá trị trong list Python là: ['Tổng', 19.0, None, None, None]
PS D:\MyBook_PyExcel\xlwings\saveCode\C3_xlwings>
```

List trong python sẽ không nhận các giá trị lỗi này và trả về **None**, điều này sẽ rất nguy hiểm trong quá trình xử lý dữ liệu, vì biết đâu đó trong quá trình tính toán sai, nhập hoặc chép dữ liệu (các giá trị chia cho 0, giá trị không tìm thấy, lỗi,...). Nếu bạn hiểu được dữ liệu không cần thiết phải lấy giá trị này thì bỏ qua thì không sao, còn giả sử bạn tính lương mà ra bằng None, tức là tiền lương trả cho bạn là 0 đồng thì hậu quả nhỏ nhất là bạn phải tự bỏ tiền túi của mình để bù rồi. Do đó, kiểm tra dữ liệu với các lỗi (Error) là điều cực kỳ cần thiết.

Viết đoạn code kiểm tra lỗi như sau:

```
import xlwings as xw

wb = xw.books.active
sh = wb.sheets["DS_xeploai"]

rng = sh.range('F1:F5')
valueErr = rng.options(empty='NA').value # cho phép lấy giá trị Lỗi

numErr = 0
for err in range(len(valueErr)):
    if valueErr[err] == 'NA':
        numErr += 1
        print("Giá trị lỗi ô:", rng[err].address)

print('Tổng các ô có lỗi là:', numErr)
```

Tải code:

https://github.com/VuNghiXuan/giaotrinhPythonExcel/blob/main/C3_xlwing_s/18.chkCellsErrors.py

Kết quả thực thi code:

The screenshot shows a terminal window with the following text:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\MyBook_PyExcel\xlwings\saveCode\C3_xlwings> & C:/Users/ADMIN/AppData/Local/Programs/Python/Python39/python.exe d:/MyBook_PyExcel/xlwings/saveCode/C3_xlwings/18.chkCellsErrors.py
Giá trị lỗi ô: $F$3
Giá trị lỗi ô: $F$4
Giá trị lỗi ô: $F$5
Tổng các ô có lỗi là: 3
PS D:\MyBook_PyExcel\xlwings\saveCode\C3_xlwings>
```

Activate Windows
Go to Settings to activate Windows.

Từ các địa chỉ liệt kê bạn mới quay lại tìm hiểu các giá trị lỗi của mình là gì để sửa lỗi, hoặc bỏ qua.

➤ *Cách tạo drop down validation*

Thông qua API kết nối với Excel để tạo ra phương thức cho Validation.Add method (Excel). Để hiểu thêm về cú pháp và cách hoạt động phương thức này bạn có thể tìm đọc thêm tại: <https://docs.microsoft.com/en-us/office/vba/api/Excel.validation.add>

Code:

```
import xlwings as xw
from xlwings import constants
def set_validation_list(rng, validation_list):
    if validation_list is None:
        validation_list = [' ']
        print('validation_list là giá trị None')
    if not isinstance(validation_list, list):
        raise TypeError('Biến validation_list không phải là list?')

    # Excel cung cấp 3 tham số cho Validation như sau:
    dv_type = constants.DVType.xlValidateList
    dv_alertstyle = constants.DVAlertStyle.xlValidAlertStop
    dv_operator = constants.FormatConditionOperator.xlEqual

    rng.api.Validation.Delete() #Xóa Validation chạy code trước đó
    rng.api.Validation.Add(dv_type,      dv_alertstyle,      dv_operator,
';'.join(validation_list))
```

```

wb = xw.books.active
sh = wb.sheets["DS_xeploai"]
rngGetData = sh.range('B2:B13')
rng_droplist = sh.range('F1')

# Lọc lấy giá trị duy nhất
value = rngGetData.value
unique_value = list(set(rngGetData.value)) # Hàm set: cho giá trị duy nhất
set_validation_list(rng_droplist, unique_value) # cho value vào cũng được

```

Tải code:

https://github.com/VuNghiXuan/giaotrinhPythonExcel/blob/main/C3_xlwing/s/19.duplicate_Data.py

Kết quả thực thi code:

The screenshot shows a Microsoft Excel spreadsheet titled "1.BangDiem.xlsx". The Data tab is selected in the ribbon. A dropdown menu is open over cell F1, displaying the unique values from column F: "Thiên Văn Chú", "Tôn Không Tứ", "Trần Nhất Thời", and "Lỗ Chí". The spreadsheet contains student marks for subjects Toán, Lý, and Hóa. The sheet tab at the bottom is labeled "DS_xeploai".

STT	Họ và tên	Toán	Lý	Hóa
1	Thiên Văn	8	9	10
2	Lỗ Chí	6	5	2
3	Trần Nhất	2	10	8
4	Tôn Khổng	8	6	7
5	Thiên Văn	8	9	10
6	Lỗ Chí	6	5	2
7	Trần Nhất	2	10	8
8	Tôn Khổng	8	6	7
9	Thiên Văn	8	9	10
10	Lỗ Chí	6	5	2
11	Trần Nhất	2	10	8
12	Tôn Khổng	8	6	7

CHƯƠNG 4: CÀI ĐẶT ADDIN SỬ DỤNG TRỰC TIẾP PYTHON TRÊN EXCEL

Giống như phương thức API, ADD-IN (Nghĩa tiếng anh thêm vào, bổ sung) là 1 phương thức trung gian cho phép người dùng triển khai hàm, module từ Python trực tiếp trên bảng tính Excel. Điều này chỉ thực hiện được qua các bước sau:

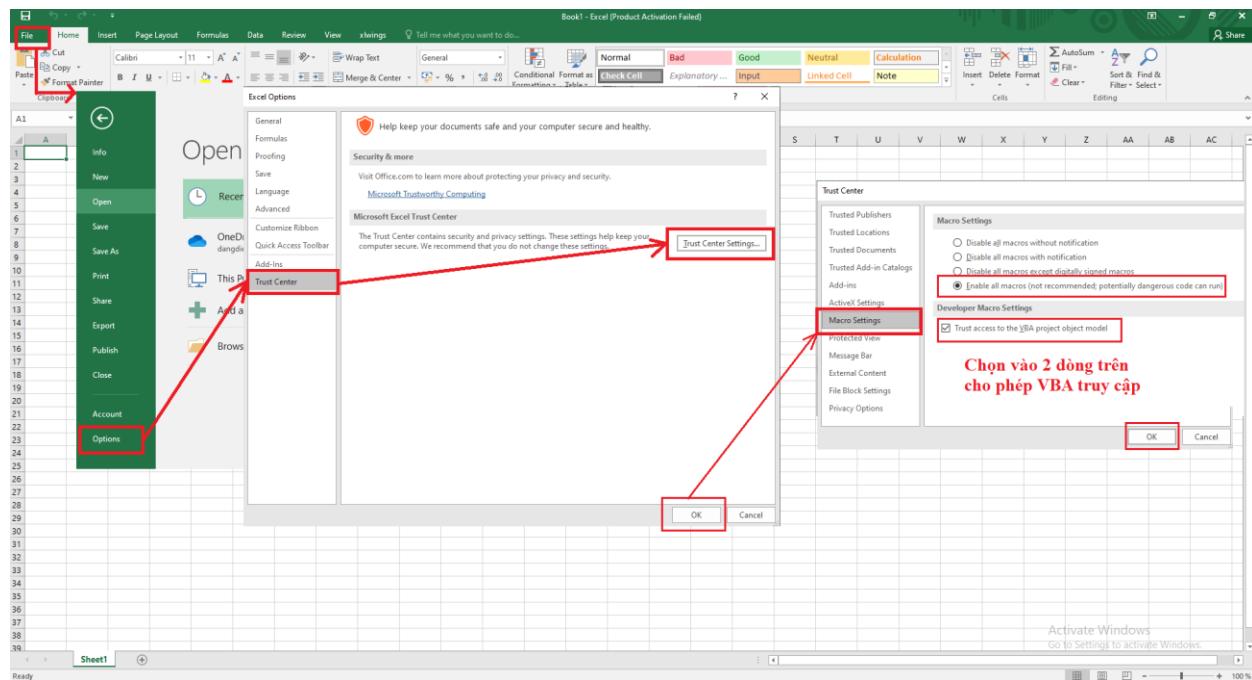
1. Kích hoạt quyền truy cập đối với VBA

VBA là ngôn ngữ lập trình có mã nguồn mở của Microsoft, cho người sử dụng nhúng mã code để điều khiển và phát triển các ứng dụng. Do đó, xlwings addin cũng dựa trên phương thức này gọi VBA khởi động để làm cầu nối liên kết với Python trong bảng tính Excel.

Mặc định Excel để bảo vệ dữ liệu, thông tin trên các file nên muốn VBA truy cập (nhúng mã vào) bạn phải thực hiện khai báo bằng các bước sau đây: Bạn thao tác lần lượt theo trình tự sau:

File > Options > Trust Center > Trust Center Settings > Macro Settings

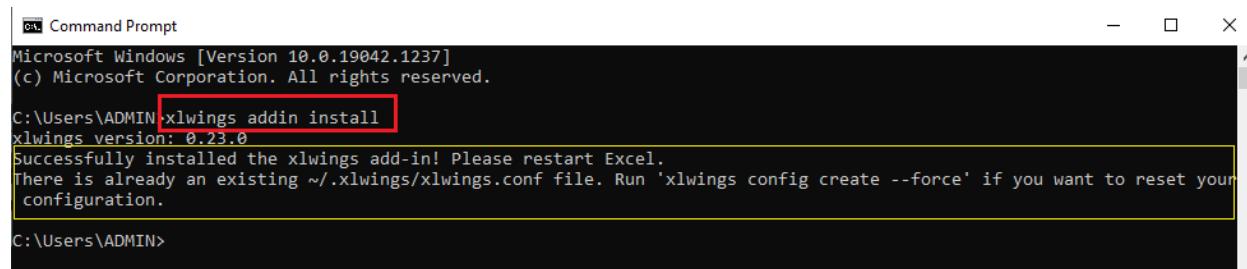
và nhấn dấu tick (✓) vào Trust access to the VBA project object model



Bước này mục đích thay đổi mặc định từ không cho phép thành cho phép VBA truy cập và chỉ cần thực hiện 1 lần trên hệ thống máy tính của bạn.

2. Cách cài đặt AddIn

Hãy khởi động chương trình **Command Prompt** trong windows, tại dấu nhắc chuột gõ nhập lệnh: **xlwings addin install** → Nhấn Enter bạn được thông báo cài đặt thành công phần khung viền màu vàng hình bên dưới.



```
Command Prompt
Microsoft Windows [Version 10.0.19042.1237]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ADMIN>xlwings addin install
xlwings version: 0.23.0
Successfully installed the xlwings add-in! Please restart Excel.
There is already an existing ~/.xlwings/xlwings.conf file. Run 'xlwings config create --force' if you want to reset your configuration.

C:\Users\ADMIN>
```

Phần cài đặt này chỉ thực hiện 1 lần cho gói **xlwings addin** trên máy bạn, để sử dụng cho máy khác bạn phải cài đặt lại.

3. Cách kết nối folder AddIn

Thông qua gói **xlwings addin** bạn đã cài đặt trên, bây giờ bạn phải tạo liên kết 1 thư mục chứa file Excel và code Python của bạn như sau:

Hãy tạo cho mình folder (đặt tên folder và địa chỉ tùy ý, miễn là bạn nhớ tên đường dẫn đầy đủ). Ví dụ, trên máy mình mình đặt tên là “**C4_addIn**”, có đường dẫn đầy đủ là: “**D:\MyBook_PyExcel\xlwings\saveCode\C4_addIn**”

Quay lại cửa sổ **Command Prompt** trong windows nhập các dòng lệnh:

- Bước 1: Di chuyển đến thư mục bạn cần kết nối: Mình chứa trên ổ D nên thực hiện:

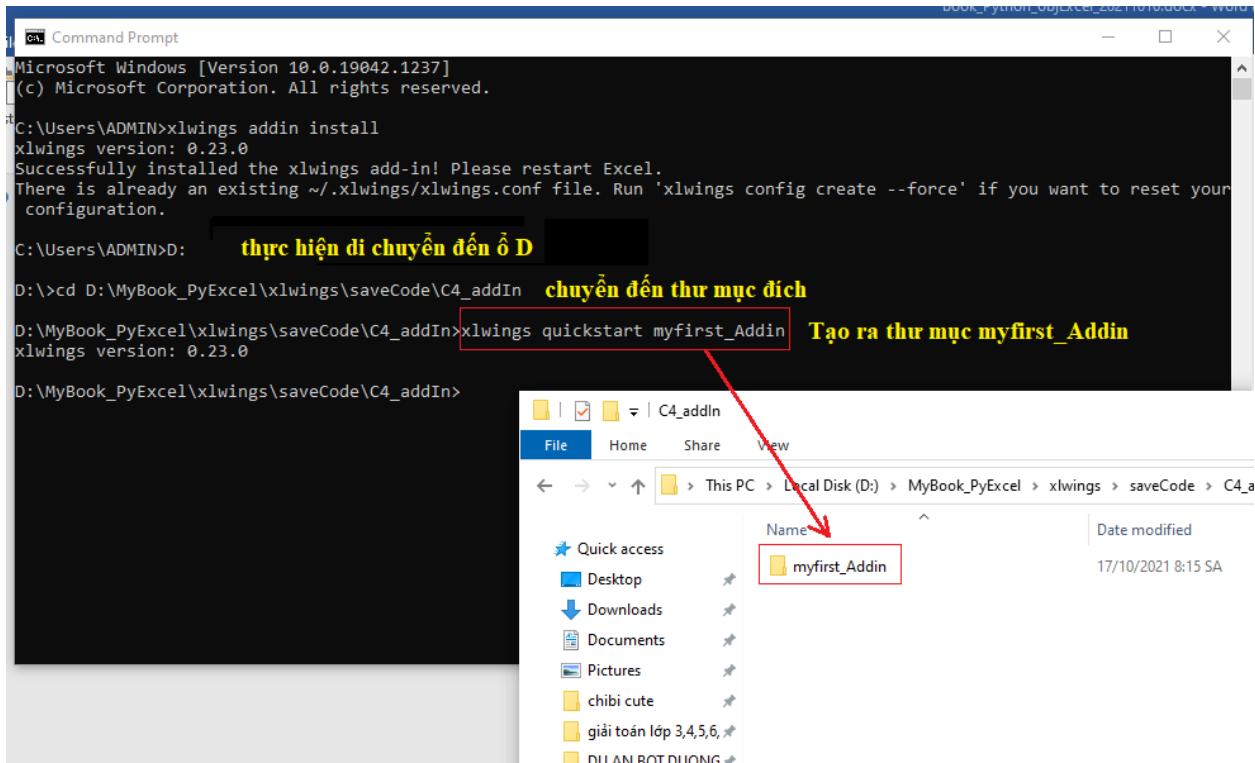
```
C:\Users\ADMIN> D:
```

```
D:>\> cd: D:\MyBook_PyExcel\xlwings\saveCode\C4_addIn
```

- Bước 2: Tạo liên kết với folder (tên folder này mình cũng tự đặt tên là **myfirst_Addin**

```
D:\MyBook_PyExcel\xlwings\saveCode\C4_addIn>xlwings quickstart myfirst_Addin
```

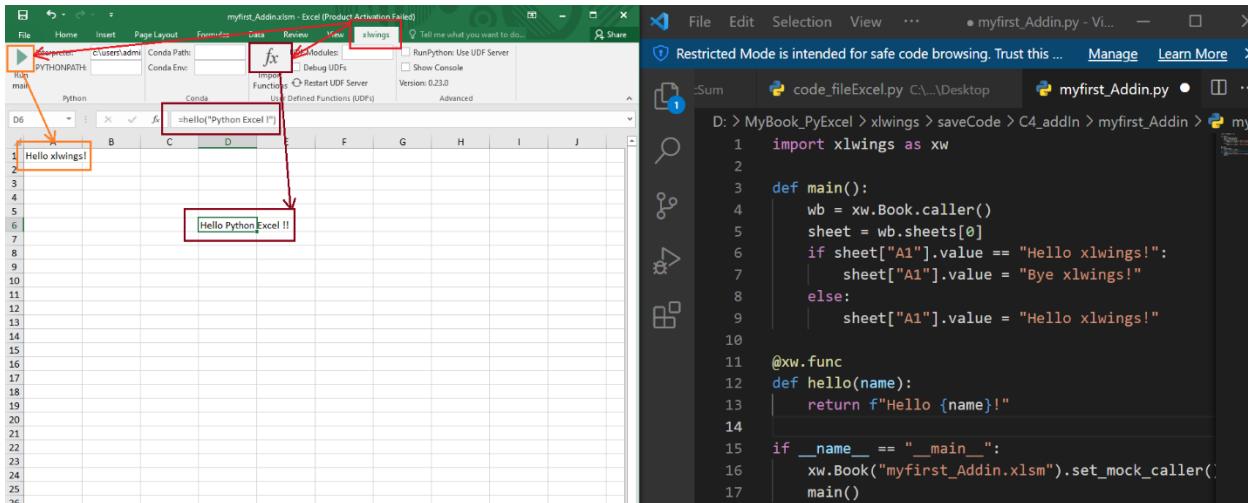
Sau khi kết thúc lệnh, một folder có tên là **myfirst_Addin** được tạo ra như hình bên dưới:



Hãy click vào thư mục **myfirst_Addin**, bạn thấy xuất hiện 01 file Python và 01 file Excel cùng mang tên là **myfirst_Addin** được tạo ra. Bạn mở 2 file này để tìm hiểu cách hoạt động của nó.

Tên file Excel tìm đến Menu Tab tên là **xlwings** (Nằm cùng dòng menu Tab với các nút File, Home, View,...), nhấn vào Run (nút tam giác ngược ➤), tại ô A1 sẽ xuất hiện dòng chữ "**Hello xlwings**"

Xem giao diện xlwings addin gọi code python

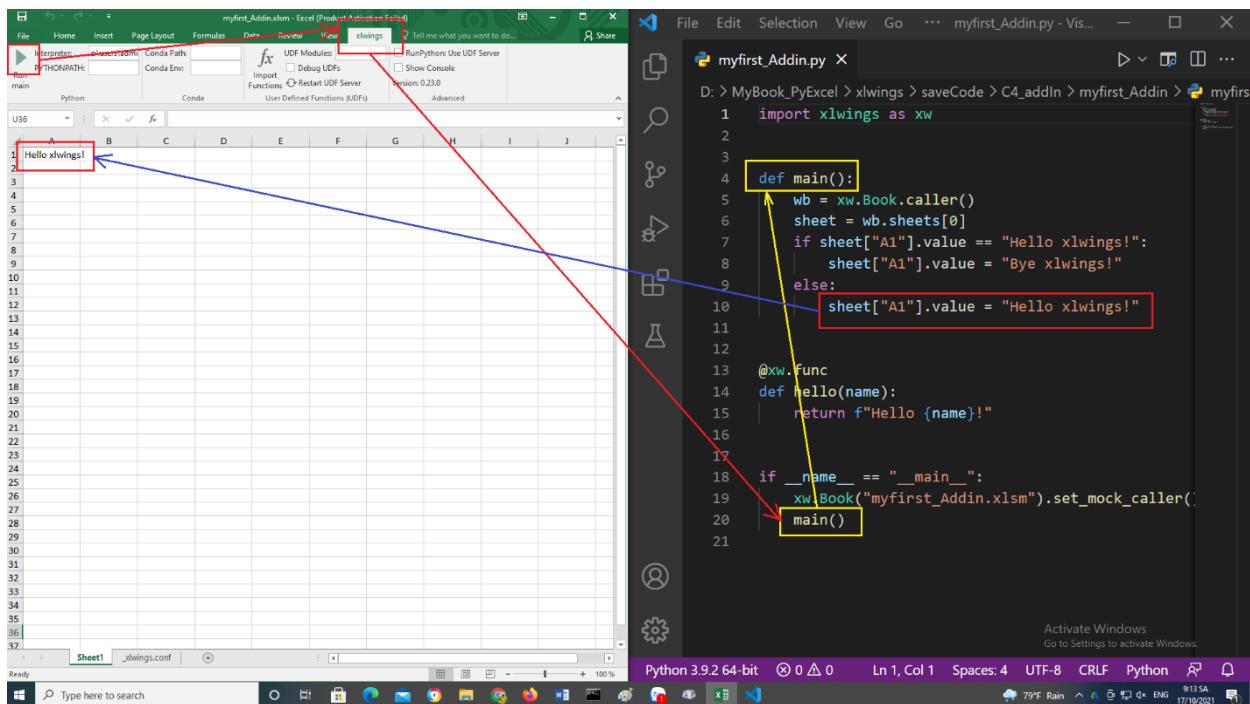


Giải thích cách hoạt động:

Qua phương thức Add-In nút Run được tạo ra để gọi đối tượng xlwings addin tìm đến hàm **main()** trong file script Python là **myfirst_Addin** để thực hiện các lệnh trong def main().

Chú ý:

Hàm **main()** được gọi là hàm chuẩn, kết thúc hàm không có return (giá trị trả về) thì nó chỉ thực hiện các lệnh bên trong nó. Bạn có thể thêm return và trả về 1 giá trị nào đó trong hàm này để tận dụng làm biến sử dụng cho các hàm khác.



➤ Sửa code trong hàm main() phương thức Add-In

Hàm **main()** trên chỉ là 1 ví dụ để bạn có thể tìm hiểu cách thức hoạt động của liên kết. Bạn hoàn toàn có thể sửa code trong hàm để thực hiện các mục đích khác nhau cho công việc của mình hoặc có thể lưu file Excel có đuôi file là (.xlsx) cho nhẹ hơn. **Tuy nhiên, muốn đổi tên file thì tên thư mục, và tên python cũng phải giống nhau**)

Thực hiện copy folder **myfisrt_Addin** → sửa tên folder vừa copy và **tên 02 file** (Excel và python) thành **VuNghiXuan** sửa code trong file **VuNghiXuan.py** như sau:

```
import xlwings as xw

def main():
    wb = xw.Book.caller()
    sheet = wb.sheets[0]
    list_RowsCols = [[15,21.56,32.98, 4.658],
                    ['Xuân', 'Lan', 'Thu', 'Cúc'],
                    ['Mặn', "chần", "cà", "hai"]]
    sheet.range('A1').value = list_RowsCols
```

```
@xw.func # Bộ chuyển hàm xlwings
def hello(name):
    return f"Hello {name}!"

if __name__ == "__main__":
    # set_mock_caller() dễ dàng chuyển qua lại gọi hàm giữa Python và Excel
    # Mã nguồn khuyến cáo file Excel và mã Python cùng folder.
    xw.Book("VuNghiXuan.xlsm").set_mock_caller()
    main()
```

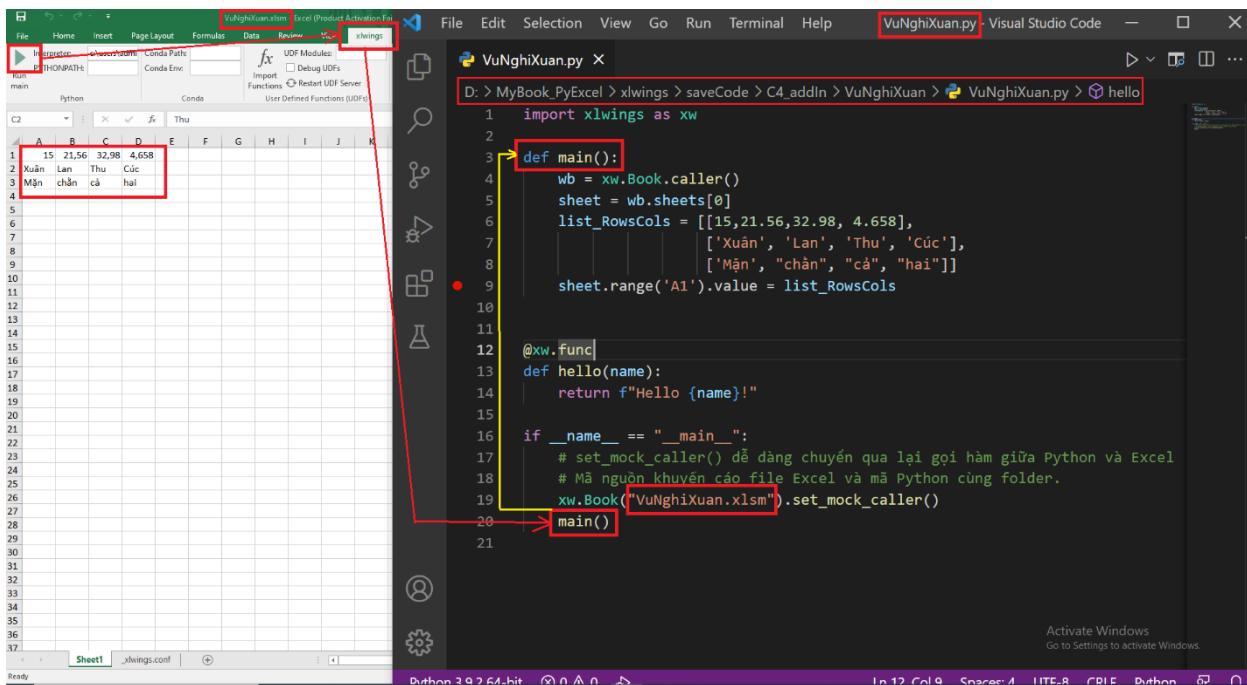
Tải code:

https://github.com/VuNghiXuan/giaotrinhPythonExcel/blob/main/C4_Add_in/VuNghiXuan.py

Tải file Excel:

https://github.com/VuNghiXuan/giaotrinhPythonExcel/blob/main/C4_Add_in/VuNghiXuan.xlsm

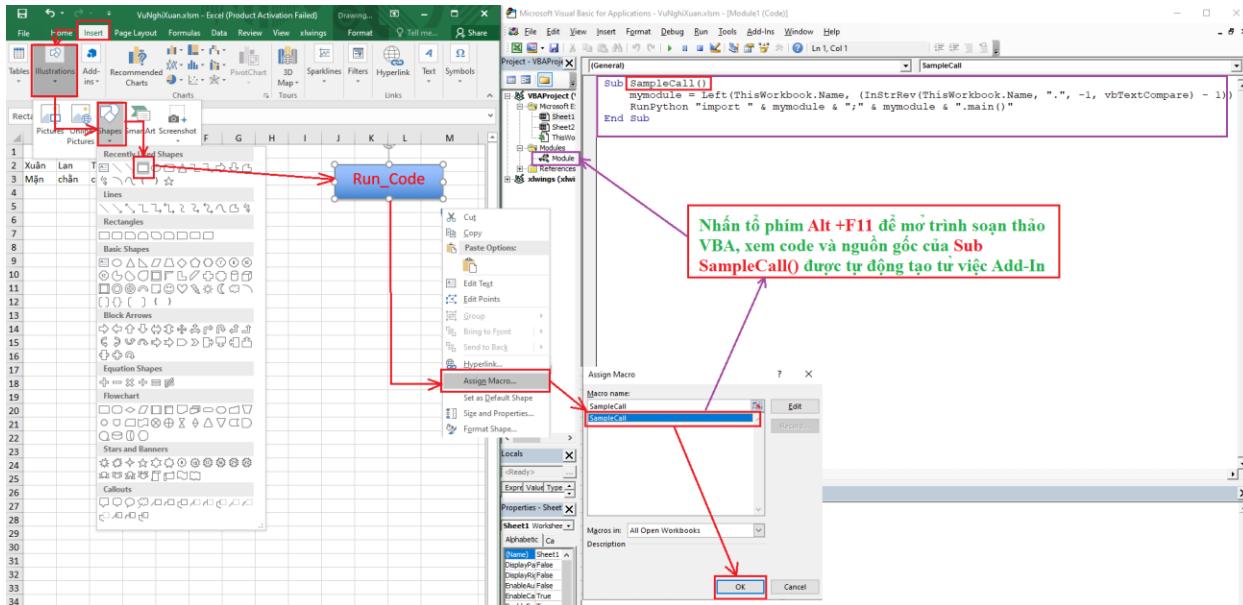
Nhấn Run trong file Excel **VuNghiXuan.xlsm** bạn xem kết quả:



Bạn chú ý các phần viền màu đỏ là phần được sửa để lúc chạy không bị lỗi hoặc kết quả ra không đúng với nội dung thay đổi code.

➤ *Tạo và gán nút Run_Code thực thi code với phương thức Add-In*

Để tạo ra nút **Run_Code** bạn làm theo hướng dẫn đường màu đỏ. Để xem code VBA và code **Sub SampleCall()** bạn nhấn tổ hợp phím Alt +F11 (đường màu tím) hình bên dưới. Sau khi tạo xong bạn xóa các nội dung trong bảng tính Excel và Click chuột vào nút **Run_Code** để xem kết quả.



➤ *Bổ sung hàm và thao tác trực tiếp trên bảng tính Excel*

Để minh họa mình đi viết hàm tính tổng 2 số và hàm tính tổng các số trong 1 dãy số:

Hàm **tong_2_so(x,y)** là tổng của giá trị $x+y$

Hàm **tong_cac_so (<Nhập dãy số>)** là tổng các số cộng lại với nhau. Ví dụ trong Excel ta nhập **tong_cac_so(1;2;3;4)**, tức là $(1+2+3+4) \rightarrow$ Kết quả = 10

Code:

```
import xlwings as xw

def main():
    wb = xw.Book.caller()
    sheet = wb.sheets[0]
    list_RowsCols = [[15,21.56,32.98, 4.658],
                     ['Xuân', 'Lan', 'Thu', 'Cúc'],
                     ['Mặn', "chẳng", "cả", "hai"]]
    sheet.range('A1').value = list_RowsCols

@xw.func
def tong_cac_so(val_1 ,*vals):
    tong = val_1
```

```

for val in vals:
    tong += val
return tong

@xw.func
def tong_2_so(x, y):
    return (x + y)

if __name__ == "__main__":
    # set_mock_caller() dễ dàng chuyển qua lại giữa Python và Excel
    # Mã nguồn khuyến cáo file Excel và mã Python cùng folder.
    xw.Book("VuNghiXuan.xlsx").set_mock_caller()
    main()

```

Kết quả:

The screenshot illustrates the integration of Python and Excel. On the left, an Excel spreadsheet displays a formula =tong_cac_so(1;2;3;4;5;6;10) in cell E8, which evaluates to the value 31. On the right, a Python code editor shows the implementation of the `tong_cac_so` function. The function takes multiple arguments and iterates through them to calculate their sum. A red box highlights the entire function definition.

Đối với hàm tính tổng 2 số bạn tự thao tác để làm quen

Chú ý:

Mỗi lần bạn thêm hàm vào thì phải thực hiện lưu lại (**nhấn save**) cho bảng tính và file code Python, đồng thời nhấn vào thẻ **Import Function** của thẻ xlwings để update và kích hoạt hàm mới.

➤ **Bổ sung hàm cho Excel theo dạng bảng 2 chiều (list 2D)**

Để minh họa mình viết 1 hàm lấy giá trị từ các ô (“B7:C8”) sau đó nhân tất cả các giá trị này cho 1 hệ số và trả về kết quả tại dãy ô (“F6:G7”)

Code:

```
import xlwings as xw

def main():
    wb = xw.Book.caller()
    sheet = wb.sheets[0]
    list_RowsCols = [[15,21.56,32.98, 4.658],
                     ['Xuân', 'Lan', 'Thu', 'Cúc'],
                     ['Mận', "chẳng", "cả", "hai"]]
    sheet.range('A1').value = list_RowsCols

"Bắt đầu viết hàm trực tiếp trên Excel -----";

# list 1D
@xw.func
def tong_cac_so(val_1 ,*vals):
    tong = val_1
    for val in vals:
        tong += val
    return tong

@xw.func
def tong_2_so(x, y):
    return (x + y)

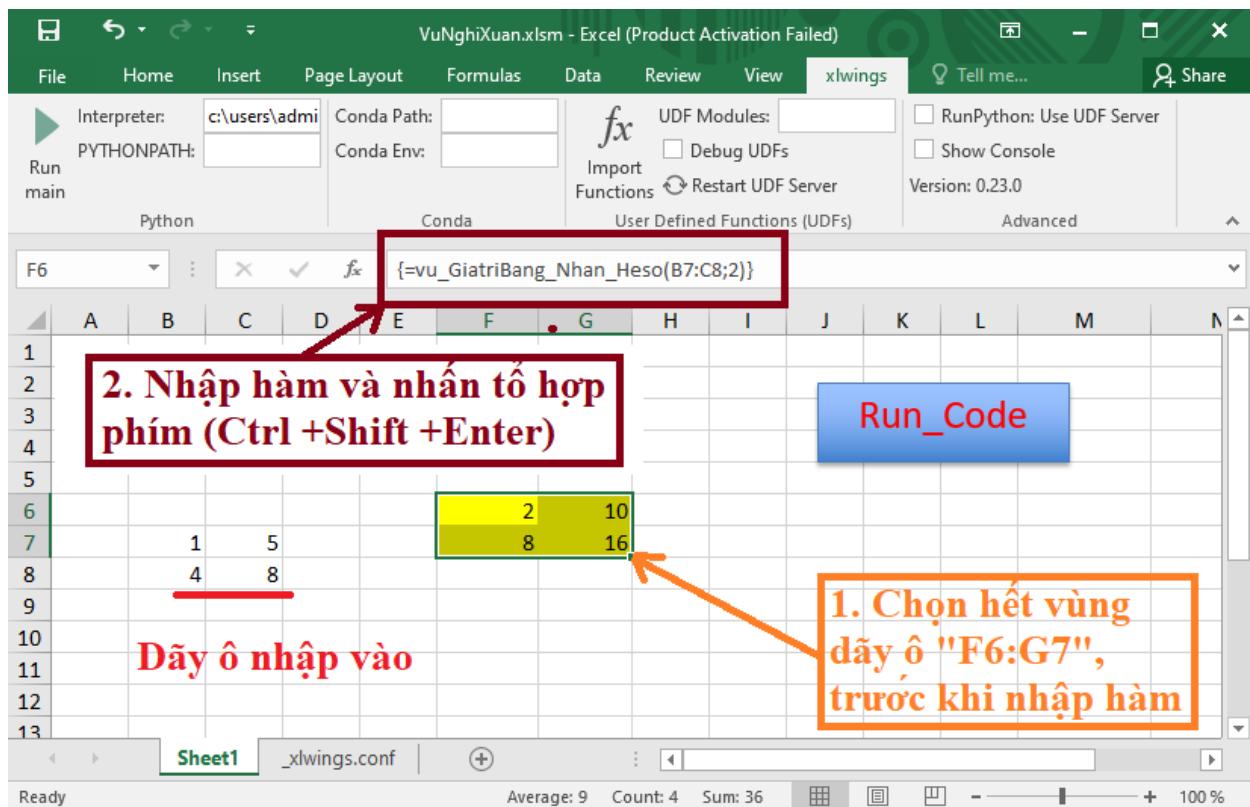
# list 2D (in ra 1 bảng gồm nhiều dòng và cột)
@xw.func
@xw.arg('data', ndim=2)
def vu_GiatriBang_Nhan_Heso(data, heso):
    return [[cell *heso for cell in row] for row in data]
```

```
"Kết thúc hàm-----"  
  
if __name__ == "__main__":  
    # set_mock_caller() dễ dàng chuyển qua lại gọi hàm giữa Python và Excel  
    # Mã nguồn khuyến cáo file Excel và mã Python cùng folder.  
    xw.Book("VuNghiXuan.xlsx").set_mock_caller()  
    main()
```

Sau khi bổ sung hàm bạn cần lưu (Ctrl+s) cả 02 file Excel và file code Python và thời kích hoạt thẻ  **Import Function**, tiếp đến bạn cần thực hiện các bước sau:

- ✓ Bôi chọn vùng cần in với số dãy ô tương đồng với dãy ô nhập, cụ thể là chọn dãy ô từ (“F6:G7”)
- ✓ Tại vùng nhập hàm của Bảng tính Excel, bạn nhập "**=vu_GiatriBang_Nhan_Heso(B7:C8;2)**". Trong đó:
 - **vu_GiatriBang_Nhan_Heso**: là tên hàm đã code trong Python
 - **B7:C8** là dãy ô tham số giá trị đầu vào
 - **2** là hệ số cần phải nhân cho từng giá trị. Ví dụ bạn cho vào là 3 thì $3 * 2$ = 6
- ✓ Để trả về kết quả cho hàng loạt dãy ô là 1 mảng trong bảng tính bạn nhấn tổ hợp phím (**Ctrl +Shift +Enter**) ngay trên vị trí nhập hàm.

Nếu bạn thực hiện đúng thì giống kết quả theo hình sau:



Bạn thử thay hệ số = 2 ở trên thành =3, =4 hoặc =0.5 và test lại xem kết quả.

➤ *Sửa tên file Excel và file Script Python với phương thức Add-In*

Để giúp bạn có thể hiểu hơn phương thức gọi trong Python và Excel (ngôn ngữ VBA) thì bạn có thể tham khảo văn bản tắt cách giải thích code bên dưới.

Giải thích dòng code VBA:

```
Sub SampleCall()
    mymodule = Left(ThisWorkbook.Name, (InStrRev(ThisWorkbook.Name, ".", -1,
vbTextCompare) - 1))
    RunPython "import " & mymodule & ";" & mymodule & ".main()"
End Sub
```

ThisWorkbook.Name: Chính là tên workbook Excel “VuNghiXuan.xls”.

Hàm InStrRev(string1,string2,[start,[compare]]): Hàm InStrRev() trong VBA trả về vị trí của lần xuất hiện đầu tiên của một chuỗi trong một chuỗi khác. Tìm kiếm xảy ra từ phải sang trái.

- string1 [Tham số bắt buộc]: Chuỗi được tìm kiếm, theo code trên chính là tên workbook “**VuNghiXuan.xlsm**”.
- string2: Tham số bắt buộc. Chuỗi mà String1 được tìm kiếm, là dấu “.”
- start: Tham số tùy chọn. Một tham số tùy chọn. Chỉ định vị trí bắt đầu cho tìm kiếm. Việc tìm kiếm bắt đầu ở vị trí đầu tiên từ phải sang trái. Code trên là giá trị **-1**, nghĩa là tìm từ cuối chuỗi về đầu chuỗi (phải qua trái).
- compare: Tham số tùy chọn. Chỉ định so sánh chuỗi được sử dụng. Nó có thể có các giá trị được đề cập sau đây.
 - ✓ 0 = vbBinaryCompare – Thực hiện so sánh nhị phân (mặc định).
 - ✓ 1 = vbTextCompare – Thực hiện so sánh kiểu văn bản.

Các bạn tự suy luận để có kết quả cuối cùng với mymodule là “**VuNghiXuan**” và dòng lệnh **RunPython "import " & mymodule & ";" & mymodule & ".main()"** trả về là **import VuNghiXuan ; VuNghiXuan.main()**. Nghĩa là, VBA gọi Python chạy hàm **main()** trong file Script Python **VuNghiXuan**.

Nếu bạn không muốn biết về VBA thì có thể bỏ qua, chỉ cần thực hiện mở được cửa sổ VBA bằng tổ hợp phím **Alt+F11** và thực hiện theo hướng dẫn hình bên dưới là cũng chạy được code.

Đầu tiên, thực hiện:

- ✓ Đổi tên file Excel là **file_ketqua.xlsm**.
- ✓ Đổi tên file script Python là **code_fileExcel.py**
- ✓ Thay hàm **main()** bằng hàm **tinh tong()**
- ✓ Trong cửa sổ VBA: Thực hiện comment hoặc xóa dòng **mymodule = Left(ThisWorkbook.Name, (InStrRev(ThisWorkbook.Name, ".", -1, vbTextCompare) - 1))** và gán trực tiếp tên file tên hàm như hình bên dưới.

```

code_fileExcel.py
1 import xlwings as xw
2
3 def tinh tong():
4     wb = xw.Book.caller()
5     sheet = wb.sheets[0]
6     list_RowsCols = [[15,21.56,32.98, 4.658],
7                      ['Xuân', 'Lan', 'Thu',
8                      'Cúc'],
9                      ['Mận', "chẳng", "cả",
10                     "hai"]]
11    sheet.range('A1').value = list_RowsCols
12
13 # def hello(name):
14 #     return f"Hello {name}!"
15
16 if __name__ == "__main__":
17     # set_mock_caller() dễ dàng chuyển qua
18     # lại gọi hàm giữa Python và Excel
19     # Mã nguồn khuyễn cáo file Excel và mã
20     # Python cùng folder.
21     xw.Book("file_ketqua.xlsm").set_mock_caller()
22     tinh tong()

```

```

Sub SampleCall()
    myModule = Left(ThisWorkbook.Name, InStrRev(ThisWorkbook.Name, ".xlsm") - 1) & "code_fileExcel"
    RunPython "import " & myModule & ";" & "code_fileExcel" & ".tinh tong()"
End Sub

```

Thực hiện thay tên file Script Python

Thay đổi tên hàm trong file Script Python

Kết quả thực thi code code VBA

Hướng dẫn thay đổi tên file Excel, Python và khai báo code trong VBA

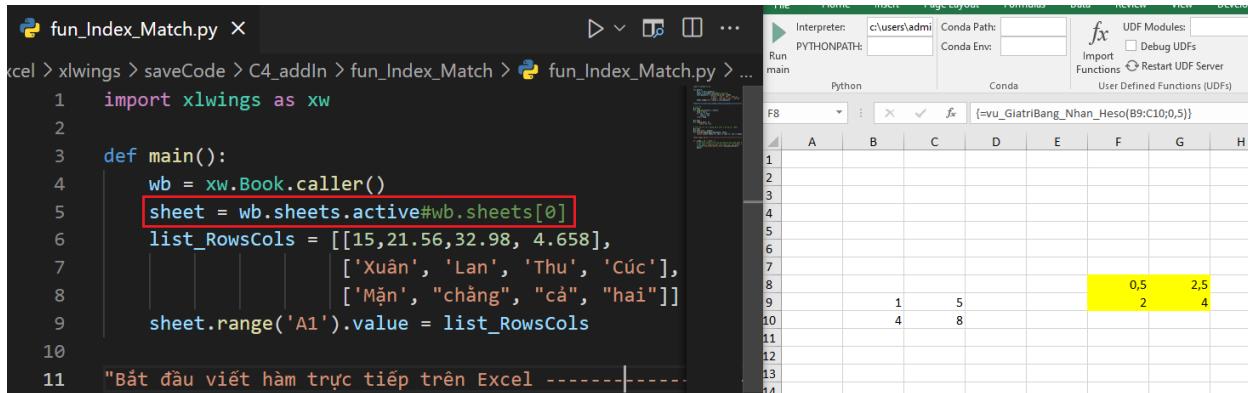
Sau khi thay đổi xong bạn có thể mở file Excel có tên là **file_ketqua.xlsm** và nhấn vào nút **Run_Code**. Nếu bạn thực hiện đúng theo hướng dẫn thì kết quả không có gì thay đổi, nghĩa là code bạn hoạt động rất tốt.

Đây là kết quả mình trên **file_ketqua.xlsm** sau nhiều lần thực hiện xóa nội dung bảng tính **Sheet1** và **Run_Code**.

	A	B	C	D	E
1	15	21,56	32,98	4,658	
2	Xuân	Lan	Thu	Cúc	
3	Mận	chẳng	cả	hai	
4					

➤ *Cách để các hàm sau khi AddIn hoạt động trong tất cả các sheets:*

Tại dòng code số 5 của hình trên ta thay đổi mặc định là **sheet = wb.sheets[0]** thành **sheet = wb.sheets.active**



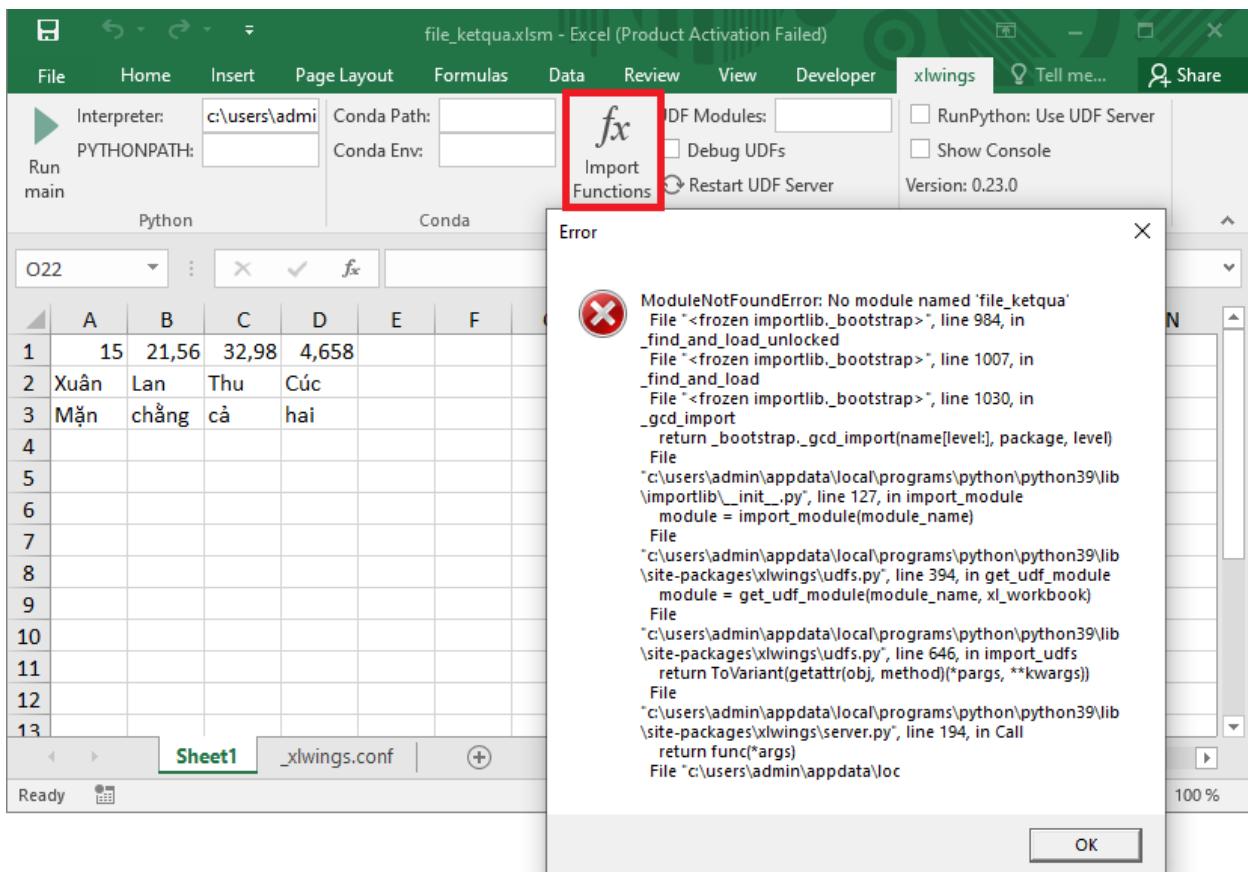
```

fun_Index_Match.py X
xlwings > saveCode > C4_addIn > fun_Index_Match > fun_Index_Match.py > ...
1   import xlwings as xw
2
3   def main():
4       wb = xw.Book.caller()
5       sheet = wb.sheets.active#wb.sheets[0]
6       list_RowsCols = [[15,21.56,32.98, 4.658],
7                         ['Xuân', 'Lan', 'Thu', 'Cúc'],
8                         ['Mận', "chẳng", "cà", "hai"]]
9       sheet.range('A1').value = list_RowsCols
10
11      "Bắt đầu viết hàm trực tiếp trên Excel -----"

```

Bây giờ hàm của bạn có thể hoạt động tại các sheets được kích hoạt (active)

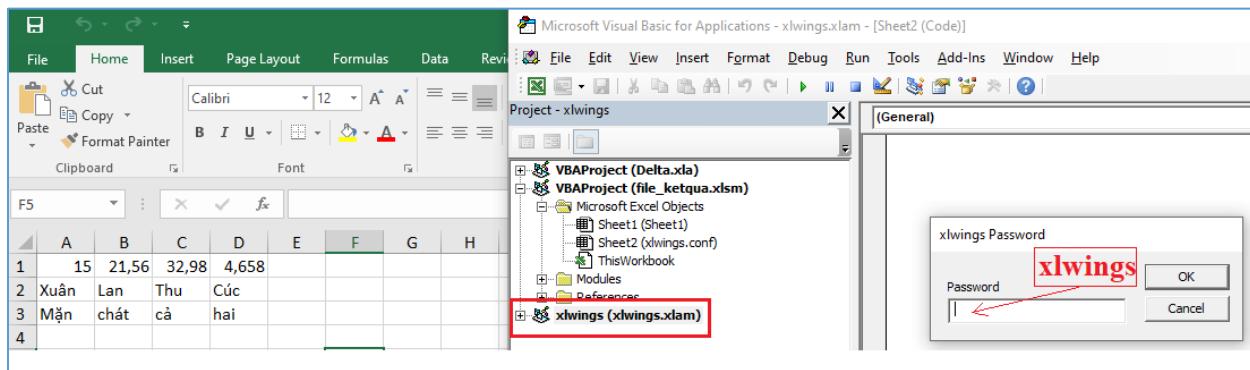
Chú ý: Khi thực hiện mở file lại cho những lần sau, đối với các file đổi tên mình bị báo lỗi sau:



Do đó, tốt nhất là bạn không đổi tên file nhé. Vì xài ứng dụng người khác có nhiều vấn đề bản quyền hay phải có hướng dẫn cụ thể. Mình cũng thấy dùng cái Addin này không thuận tiện cho lắm. Thời gian tới, mình sẽ hướng dẫn bạn tiếp cận với các gói GUI, bạn có thể tự viết app giao tiếp trực tiếp với Excell một cách xịn xò.

➤ *Mở password xlwings và tham khảo code trên VBA*

Nếu bạn biết về ngôn ngữ VBA và muốn tìm hiểu phương thức kết nối với Python thì bạn mở cửa sổ VBA (Alt + F11), click vào dự án xlwings (xlwings.xlam) và gõ password là: **xlwings**



Đây là những gì bạn tham khảo:

```

Option Explicit
#Const App = "Microsoft Excel" 'Adjust when using outside of Excel

Function GetDirectoryPath(Optional wb As Workbook) As String
    ' Leaving this here for now because we currently don't have #Const App in Utils
    Dim Path As String
    #If App = "Microsoft Excel" Then
        On Error Resume Next 'On Mac, this is called when exiting the Python interpreter
        Path = Getdirectory(GetFullName(wb))
        On Error GoTo 0
    #ElseIf App = "Microsoft Word" Then
        Path = ActiveDocument.Path
    #ElseIf App = "Microsoft Access" Then
        Path = CurrentProject.Path ' Won't be transformed for standalone module as ThisProject doesn't exit
    #ElseIf App = "Microsoft PowerPoint" Then
        Path = ActivePresentation.Path ' Won't be transformed for standalone module ThisPresentation doesn't exist
    #Else
        Exit Function
    #End If
    GetDirectoryPath = Path
End Function

Function GetConfigFilePath() As String
    #If Mac Then
        '/Library/Containers/com.microsoft.Excel/Data/xlwings.conf
        GetConfigFilePath = GetMacDir("$HOME", False) & "/" & PROJECT_NAME & ".conf"
    #Else
        GetConfigFilePath = Environ("USERPROFILE") & "\." & PROJECT_NAME & "\& PROJECT_NAME & ".conf"
    #End If
End Function

Function GetDirectoryConfigFilePath() As String
    Dim pathSeparator As String

```

CHƯƠNG 5: ỨNG DỤNG PHỤC VỤ CÔNG VIỆC:

VIẾT ỨNG DỤNG TỔNG HỢP DỮ LIỆU

1. Tổng dữ liệu các sheets trên cùng 1 file Excel

Code:

```
import xlwings as xw

# Xây dựng hàm đọc file--> Trả về đối tượng workbook
def readExcel(path_input):
    app = xw.App()
    wb = app.books.open(path_input)
    return wb

# Tìm dòng cuối, cột cuối dữ liệu:
def get_lrows_lcols(sh, i_sh):
    lr_table = sh.cells.last_cell.row # Dòng cuối bảng tính Excel: 65536 dòng
(bản Excel bị giới hạn)
    lc_table = sh.cells.last_cell.column # Cột cuối bảng tính Excel: 256 cột

    lr_data = sh.range('A'+ str(lr_table)).end('up').row # Dòng cuối dữ liệu
    lc_data = sh.range(1, lc_table).end('left').column # Cột cuối dữ liệu

    # Chỉ lấy dòng tiêu đề sheet đầu tiên
    if i_sh == 0:
        table_datas = sh.range((1,1), (lr_data,lc_data)).value # Data dữ liệu
từ dòng 1
    else:
        table_datas = sh.range((2,1), (lr_data,lc_data)).value # Data dữ liệu
lấy từ dòng 2
    lr_data = lr_data - 1
```

```

        print(f'file: {wb.name}. Sheet{i_sh+1} có: {lr_data} dòng và {lc_data} cột')
    return table_datas, lr_data, lc_data

# Xây dựng tìm hiểu dữ liệu các sheets
def getData_AllSheets(wb):
    num_shs = len(wb.sheets)
    datas_out = []
    sum_lr = 0
    for i_sh in range(num_shs):
        sh = wb.sheets[i_sh]

        # Dùng hàm để trả về giá data 1 sheet, tổng dòng, tổng cột
        datas, lr_data, lc_data = get_lrows_lcols(sh, i_sh)
        sum_lr += lr_data

        # Thêm dữ liệu vào datas mới
        datas_out.extend(datas) # datas.value: Lấy giá trị dữ liệu
    # print(f'file:{wb.name}, sheet{i_sh+1} có {sum_lr} dòng và {lc_data} cột')
    return datas_out, sum_lr, lc_data

"Khai báo đường dẫn cụ thể file input và output";

# Đường dẫn file input (file "movies.xls" chứa trong thư mục "read_Datas")
path_input = 'D:\\MyBook_PyExcel\\xlwings\\read_Datas\\movies.xls'

# Đường dẫn lưu (file "C4_tonghopdulieuSheets" chứa trong thư mục "read_Datas")
path_output = 'D:\\MyBook_PyExcel\\xlwings\\save_Datas\\C4_tonghopdulieuSheets.xls'

# Đọc và mở file bằng hàm
wb = readExcel(path_input)

# Lấy dữ liệu các sheet

```

```

datas_out, sum_lr, lc_data = getData_AllSheets(wb)

wb_new = xw.Book()
wb_new.sheets[0].range('A1').value = datas_out
wb_new.sheets[0].name = "Tonghop_dulieu_sheets"
wb_new.save(path_output)

# In ra thông tin file output
print(f'File: {wb.name}. Tổng cộng: {sum_lr} dòng và {lc_data} cột')

# Đóng workbook
wb.close()
wb_new.close()

# Thoát khỏi 1 ứng dụng Excel hiện hành.
for ap in xw.apps:
    ap.quit()

```

2. Tổng hợp dữ liệu nhiều files Excel, mỗi file có nhiều sheets

Code:

```

import xlwings as xw

# Xây dựng hàm đọc file--> Trả về đối tượng workbook
def readExcel(path_input):
    app = xw.App(visible=False)
    wb = app.books.open(path_input)
    return wb

# Tìm dòng cuối, cột cuối dữ liệu:
def get_lrows_lcols(i_file, sh, i_sh):
    lr_table = sh.cells.last_cell.row # Dòng cuối bảng tính Excel: 65536 dòng
(bản Excel bị giới hạn)
    lc_table = sh.cells.last_cell.column # Cột cuối bảng tính Excel: 256 cột

```

```

lr_data = sh.range('A'+ str(lr_table)).end('up').row # Dòng cuối dữ liệu
lc_data = sh.range(1, lc_table).end('left').column # Cột cuối dữ liệu

# Chỉ lấy dòng tiêu đề sheet đầu tiên
if i_file and i_sh == 0:
    table_datas = sh.range((1,1), (lr_data,lc_data)).value # Data dữ liệu
từ dòng 1
else:
    table_datas = sh.range((2,1), (lr_data,lc_data)).value # Data dữ liệu
lấy từ dòng 2
    lr_data = lr_data - 1
print(f'file: {wb.name}. Sheet{i_sh+1} có: {lr_data} dòng và {lc_data}
cột')
return table_datas, lr_data, lc_data

# Xây dựng tóm tắt dữ liệu các sheets
def getData_AllSheets(wb, i_file):
    num_shs = len(wb.sheets)
    datas_out = []
    sum_lr = 0
    for i_sh in range(num_shs):
        sh = wb.sheets[i_sh]

        # Dùng hàm để trả về giá trị data 1 sheet, tổng dòng, tổng cột
        datas, lr_data, lc_data = get_lrrows_lcols(i_file, sh, i_sh)
        sum_lr += lr_data

        # Thêm dữ liệu vào datas mới
        datas_out.extend(datas) # datas.value: Lấy giá trị dữ liệu
    # print(f'file:{wb.name}, sheet{i_sh+1} có {sum_lr} dòng và {lc_data}
cột')
    return datas_out, sum_lr, lc_data

"Khai báo đường dẫn cụ thể file input và output";

```

```

# Đường dẫn file input (file "movies.xls" chứa trong thư mục "read_Datas")
path_input = ['D:\\MyBook_PyExcel\\xlwings\\read_Datas\\movies.xls',
              'D:\\MyBook_PyExcel\\xlwings\\read_Datas\\1_movies.xls',
              'D:\\MyBook_PyExcel\\xlwings\\read_Datas\\2_movies.xls',
              'D:\\MyBook_PyExcel\\xlwings\\read_Datas\\3_movies.xls',
              'D:\\MyBook_PyExcel\\xlwings\\read_Datas\\4_movies.xls',
              'D:\\MyBook_PyExcel\\xlwings\\read_Datas\\5_movies.xls']

# Đường dẫn lưu (file "C4_tonghopdulieuSheets" chứa trong thư mục
"path_output"
"read_Datas") = 'D:\\MyBook_PyExcel\\xlwings\\save_Datas\\C4_tonghopdulieuFiles.xls'

# Lấy dữ liệu các sheet
data_files = []
sum_lr_files = 0
for i_file in range(len(path_input)):
    # Đọc và mở file bằng hàm
    wb = readExcel(path_input[i_file])
    datas_out, sum_lr, lc_data = getData_AllSheets(wb, i_file)
    data_files.extend(datas_out)

    sum_lr_files += sum_lr
    wb.close()

wb_new = xw.Book()
wb_new.sheets[0].range('A1').value = data_files
wb_new.sheets[0].name = "Tonghop_dulieu_files"
wb_new.save(path_output)

# In ra thông tin file output
print(f'File: {wb_new.name}. Tổng cộng: {sum_lr_files} dòng và {lc_data} cột')

# Đóng workbook
# wb.close()

```

```
wb_new.close()

# Thoát khỏi 1 ứng dụng Excel hiện hành.
for ap in xw.apps:
    ap.quit()
```

CHƯƠNG 6: GIAO DIỆN ĐỒ HỌA VỚI TKINTER

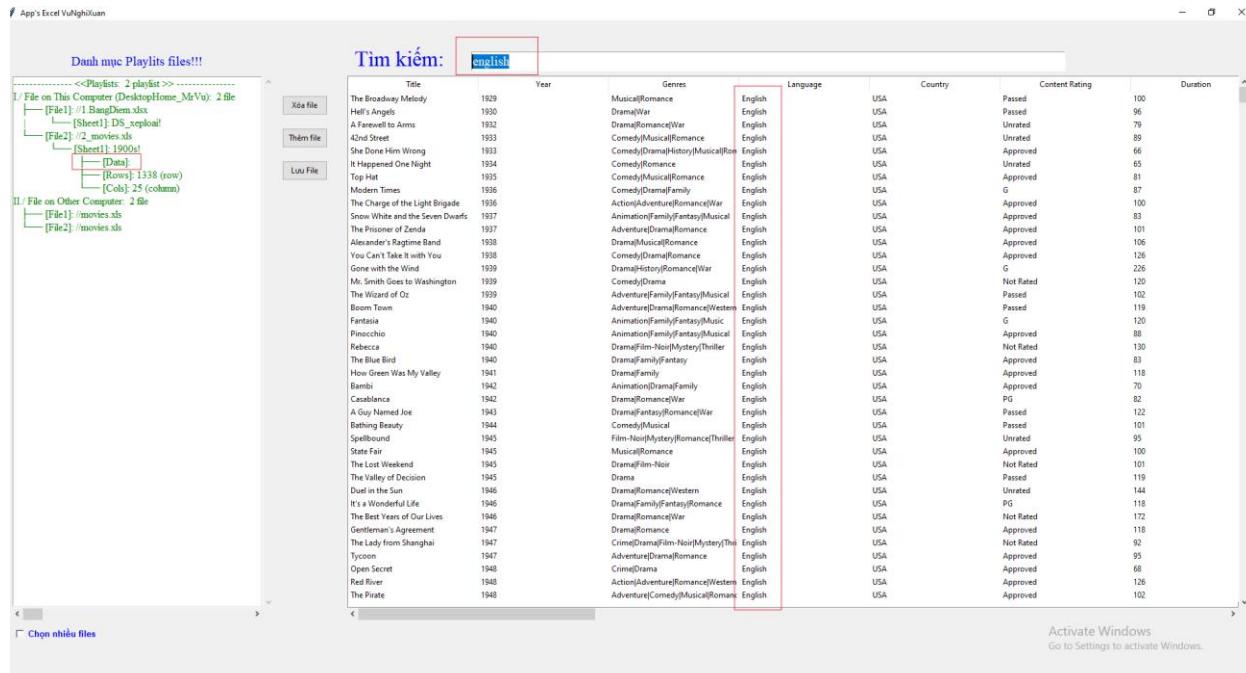
Chương này mình **chỉ giới thiệu qua** để các bạn tìm hiểu thêm và tự nâng cấp code của mình chuyển thành 1 phần mềm cho người dùng sử dụng.

Hiện nay, có rất nhiều loại giao diện đồ họa Graphic Uses Interface (gọi tắt là GUI). Mỗi loại đều có ưu nhược điểm riêng nên khó có thể trình bày hết ở đây. Tuy nhiên, đây là 02 loại có thể giúp bạn làm được các giao diện từ dễ đến một cách chuyên nghiệp, đặc biệt là PyQt.

I. GIAO DIỆN ĐỒ HỌA TKINTER

Tkinter là 1 giao diện đồ họa có sẵn trong Python mà không phải cài đặt, giúp bạn có thể tạo ra giao diện cho người dùng dựa theo logic code của mình.

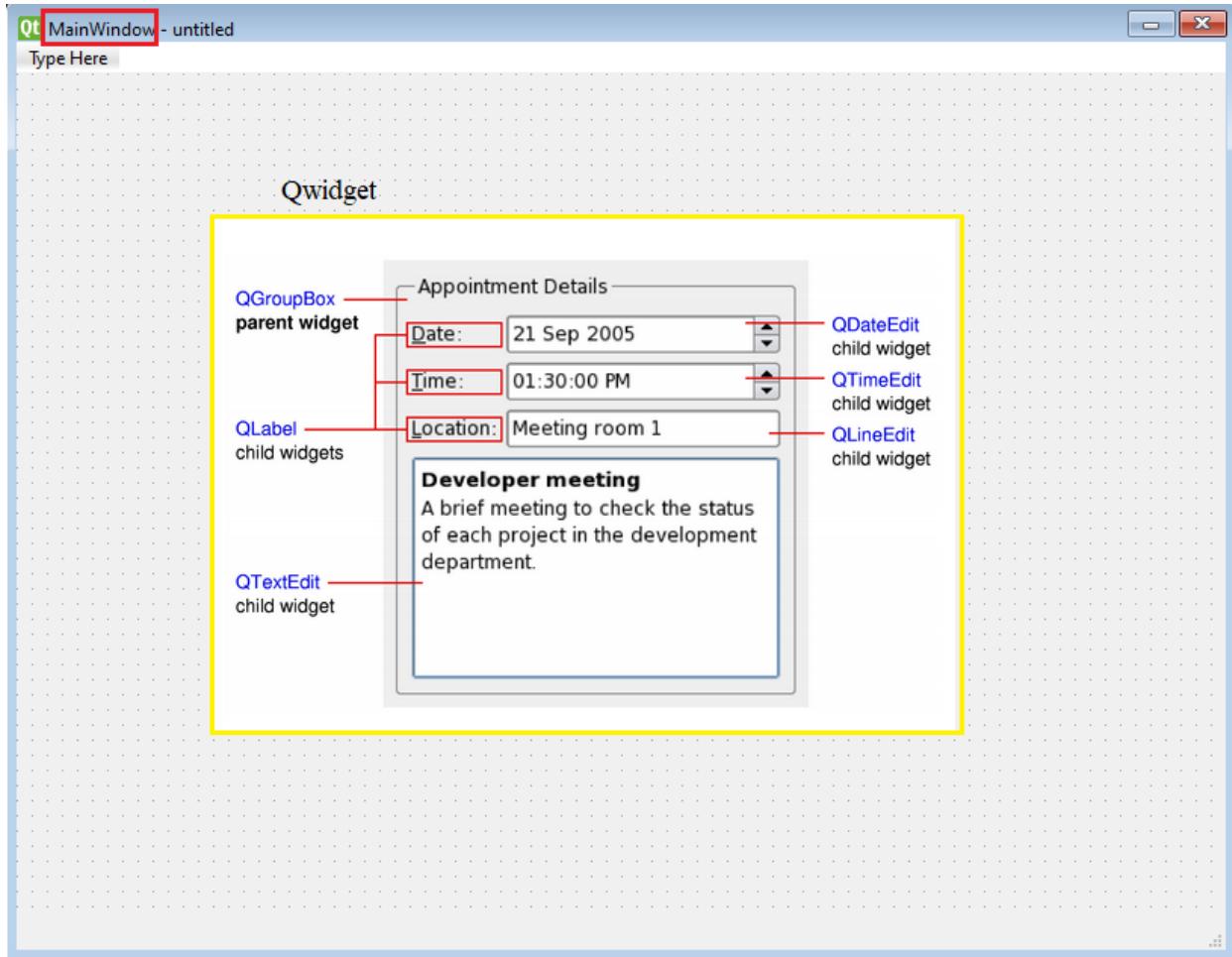
Đây là giao diện được sử dụng Thư viện Tkinter, phát triển thành **ứng dụng tìm kiếm chuỗi trong bảng tính file Excel**, bạn có thể nhập vào nhiều file và xem và tìm kiếm được chuỗi dữ liệu trên từng Sheets.



II. GIAO DIỆN ĐỒ HỌA PyQT5:

QT là bộ công cụ đồ họa, còn PyQt là 1 thư viện ngôn ngữ Python để lập trình giao diện, con số **5** là 1 phiên bản **version**.

Các thành phần trong 1 giao diện QT:



QWidget là lớp cơ sở cho tất cả các lớp có thể vẽ được trong Qt. Bất kỳ QWidget lớp dựa trên cơ sở nào cũng có thể được hiển thị dưới dạng cửa sổ bằng cách hiển thị nó khi nó không có cha.

Qdialog dựa trên QWidget, nhưng được thiết kế để hiển thị dưới dạng cửa sổ. Nó sẽ luôn xuất hiện trong một cửa sổ và có các chức năng để làm cho nó hoạt động tốt với các nút thông thường trên hộp thoại (chấp nhận, từ chối, v.v.).



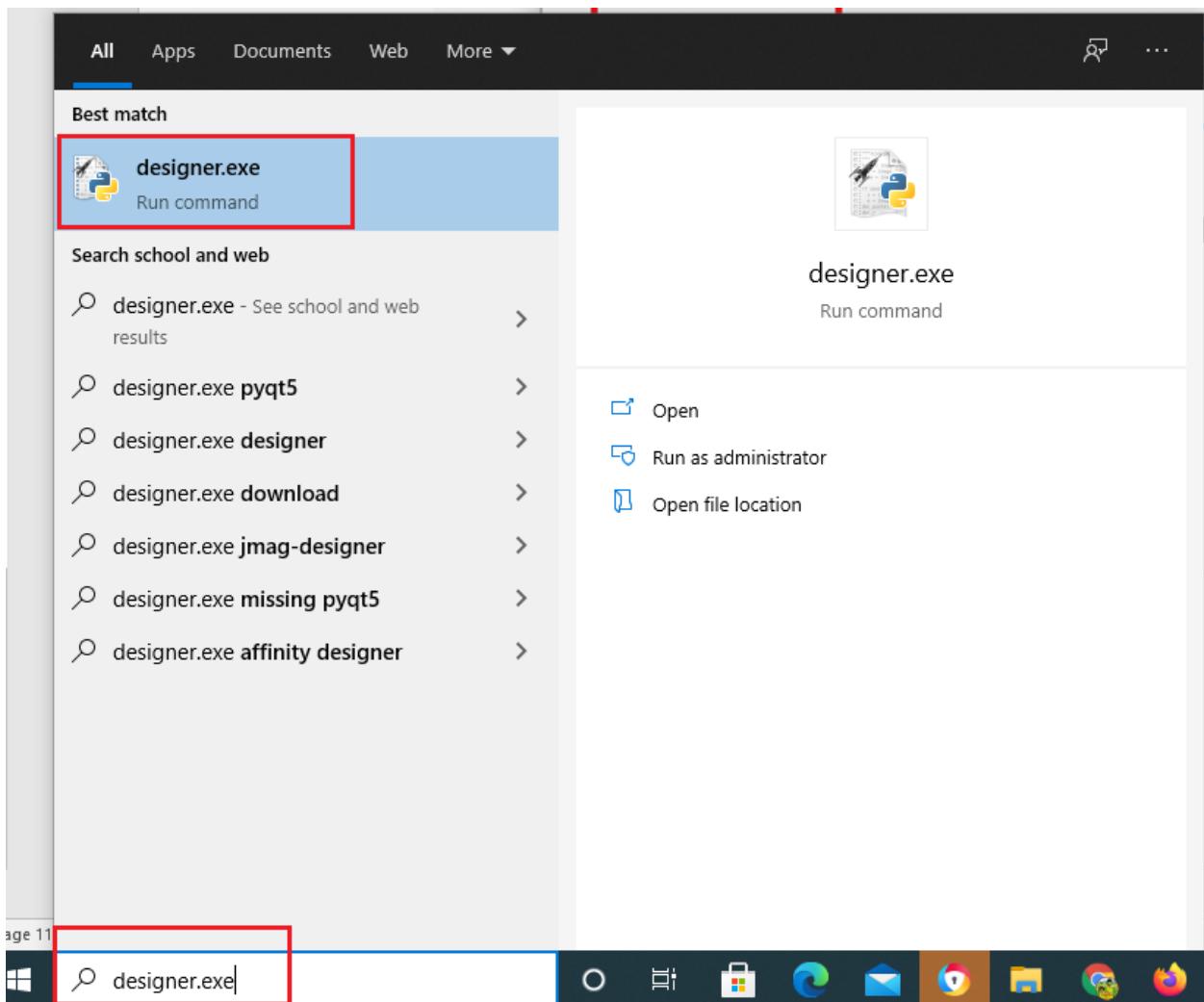
Qmainwindow được thiết kế xung quanh các nhu cầu thông thường đối với một cửa sổ chính. Nó có các vị trí được xác định trước cho thanh menu, thanh trạng thái, thanh công cụ (trên hình là nút **Close** màu đỏ) và các widget khác. Nó không có bất kỳ phụ cấp tích hợp nào cho các nút tương tự QDialog.

Để bắt đầu với giao diện, bạn khởi động cửa sổ **Command Prompt** và cài đặt với cú pháp: **pip install pyqt5designer** (phiên bản pyqt5)

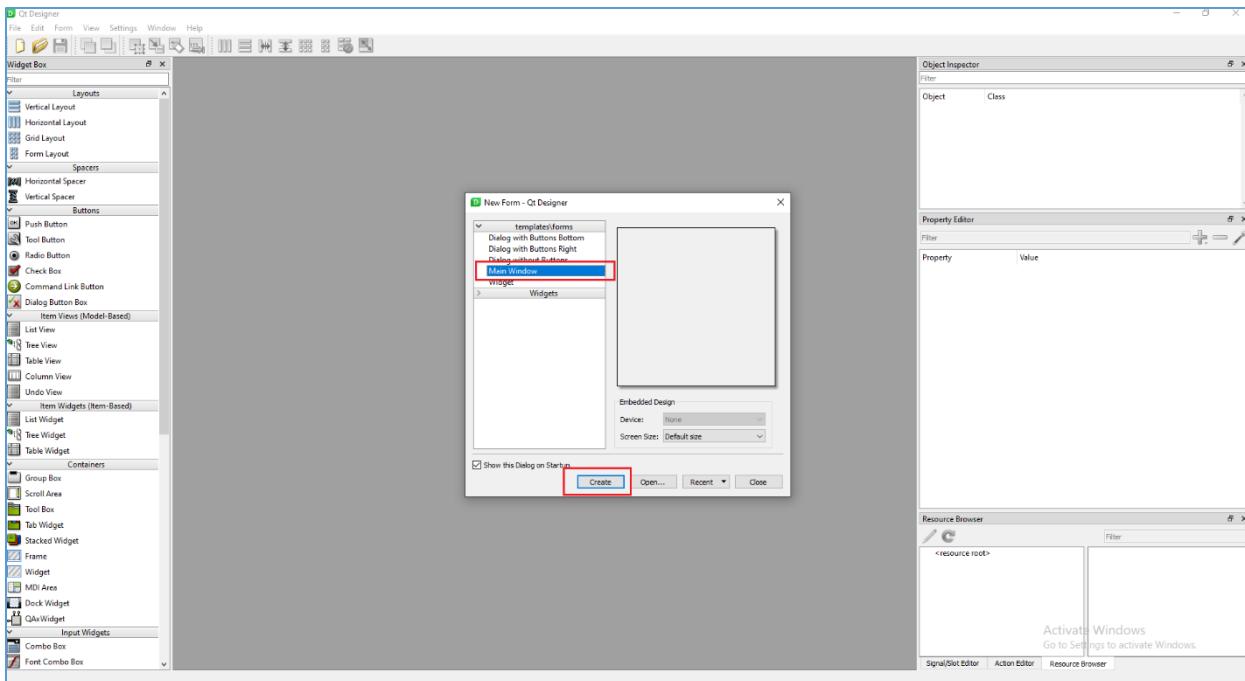
```
C:\Users\ADMIN>pip install pyqt5designer
Collecting pyqt5designer
  Downloading PyQt5Designer-5.14.1-py3-none-win_amd64.whl (40.8 MB)
    |██████████| 40.8 MB 6.4 MB/s
Installing collected packages: pyqt5designer
Successfully installed pyqt5designer-5.14.1
WARNING: You are using pip version 21.1.2; however, version 21.3 is available.
You should consider upgrading via the 'c:\users\admin\appdata\local\programs\python\python39\python.exe -m pip install --upgrade pip' command.

C:\Users\ADMIN>
```

Sau khi cài đặt thành công bạn gõ vào Entry tìm kiếm của Windows dòng chữ **designer.exe** và click vào phần mềm này như hình dưới.

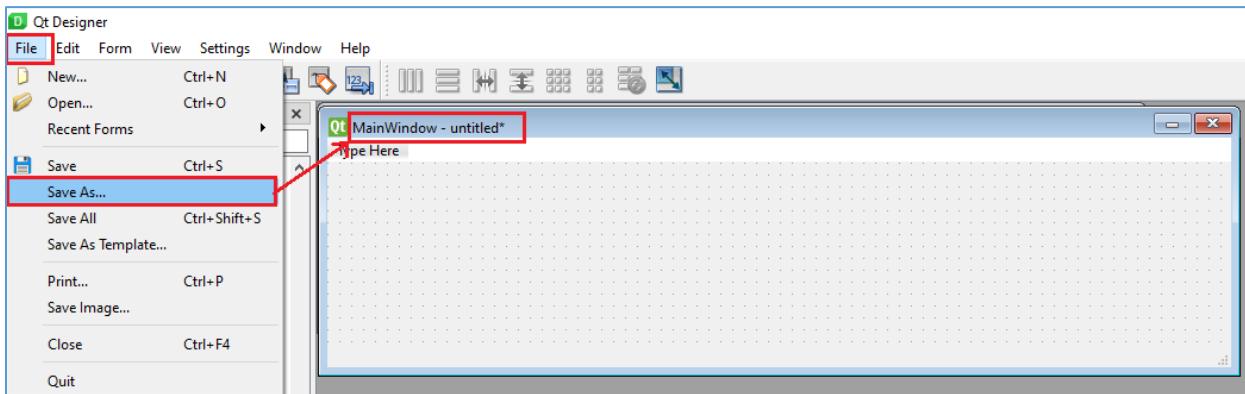


Đây chính là giao diện của Designer PyQt5.



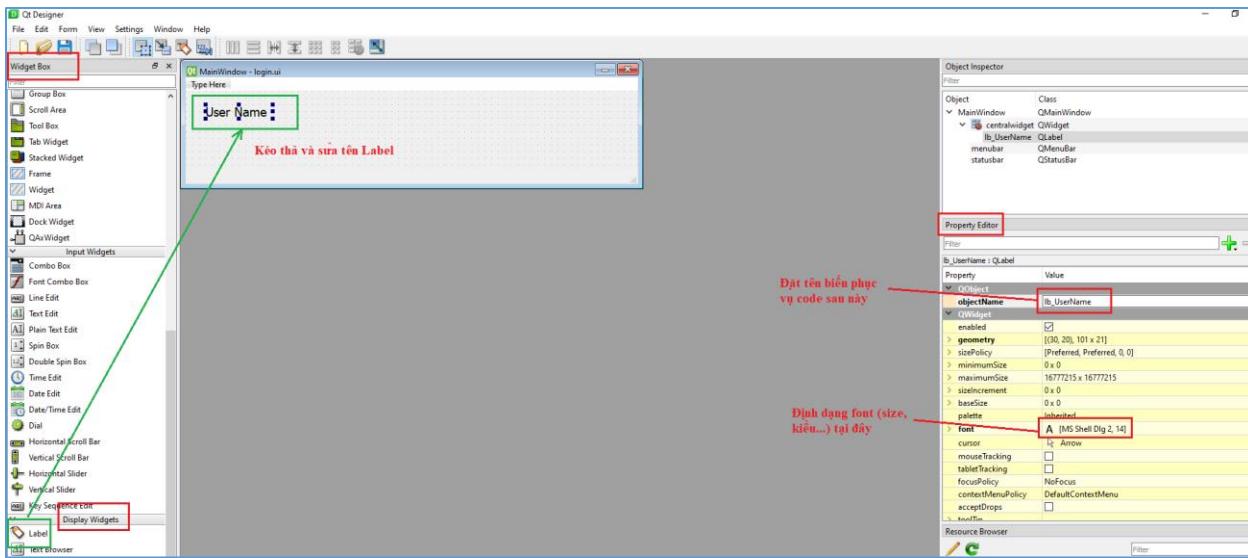
➤ Cách tạo ra nhãn Label trên Form QT5

Bạn chọn vào mục **Main Windows** và nhấn vào nút **Create** để tạo form mới, bạn vào **File>Save as...** để lưu và đổi tên file thành **login.ui**



Tiếp theo, trong menu **WidgetBox** tại mục **Display Widgets** bạn kéo thả đối tượng **Label** vào form và khai báo giống như hình bên dưới. Đặt tên đối tượng cho nhãn User Name này là **lb_UserName**, đó cũng là tên biến để dễ gọi và điều khiển nó trong bản thảo code.

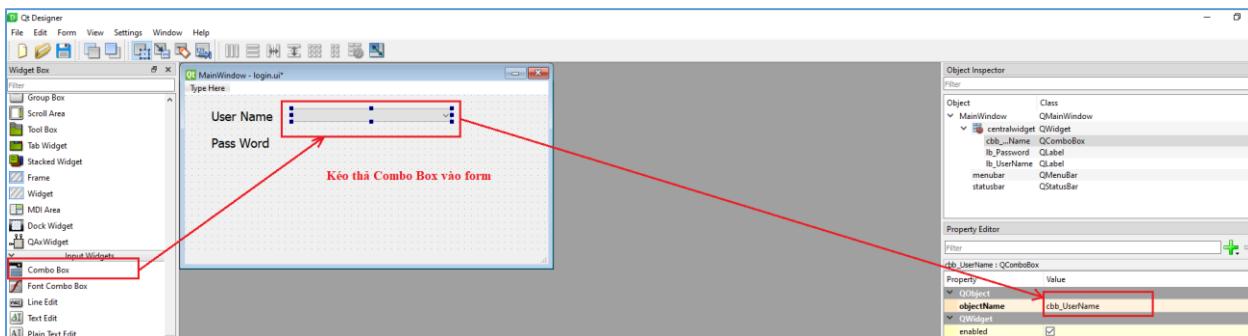
Vào thẻ **Property Editor** để thay đổi định dạng lại đối tượng label như: thay đổi cỡ chữ, màu chữ,...



Tương tự bạn có thể copy UserName và đổi tên hoặc thực hiện kéo thả như trên để tạo ra Password và đặt tên đối tượng này là **lb_PassWord**

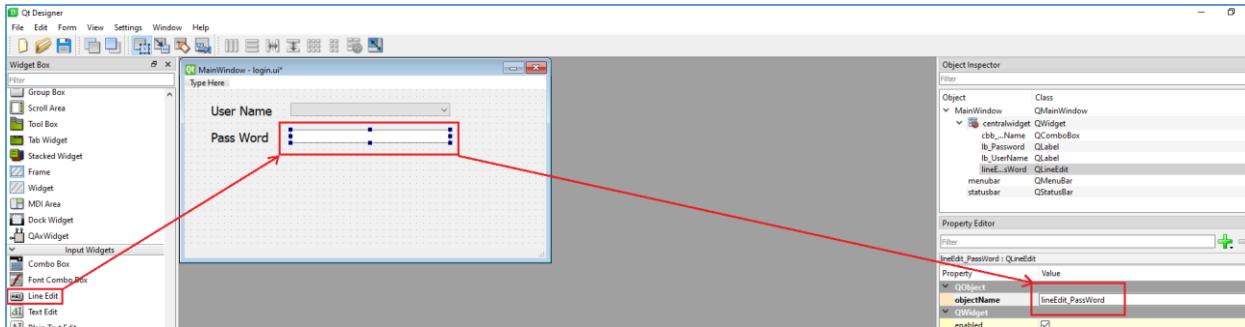
➤ Cách tạo Combo Box

Vào **Input Widgets**, chọn và kéo thả **Combo Box** vào form, đặt tên đối tượng **cbb_UserName**



➤ Cách tạo Line Edit

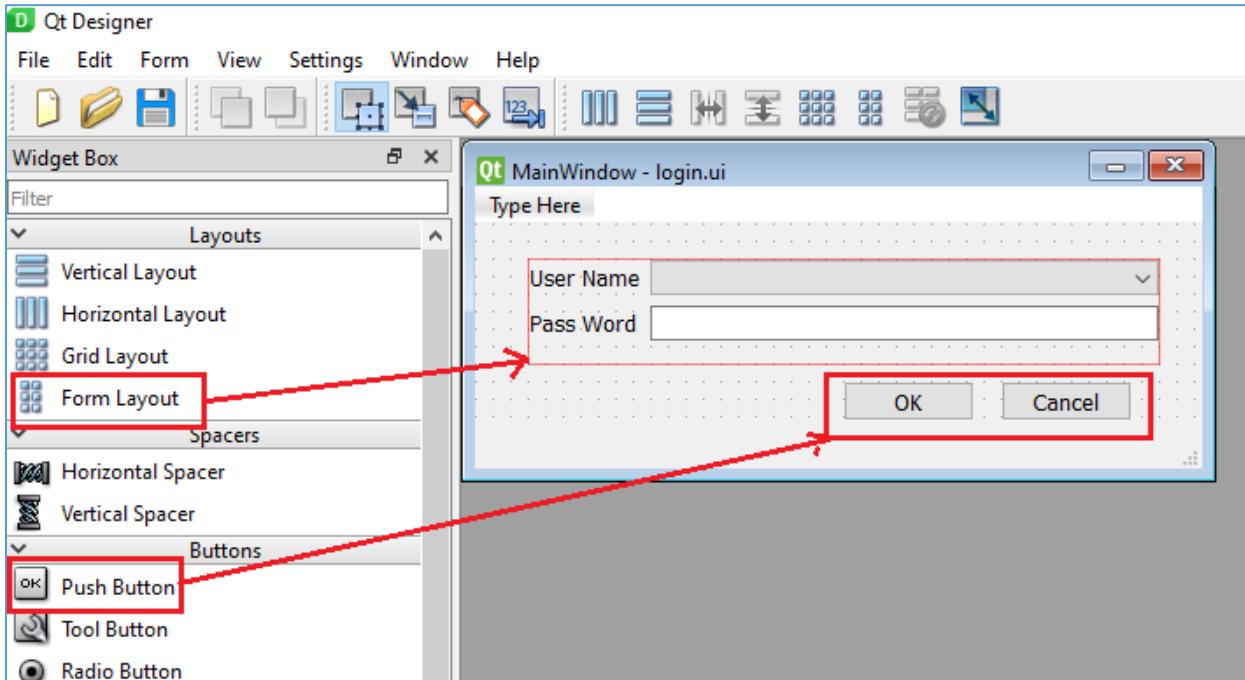
Vào **Input Widgets**, chọn và kéo thả **Line Edit** vào form, đặt tên đối tượng **lineEdit_PassWord**



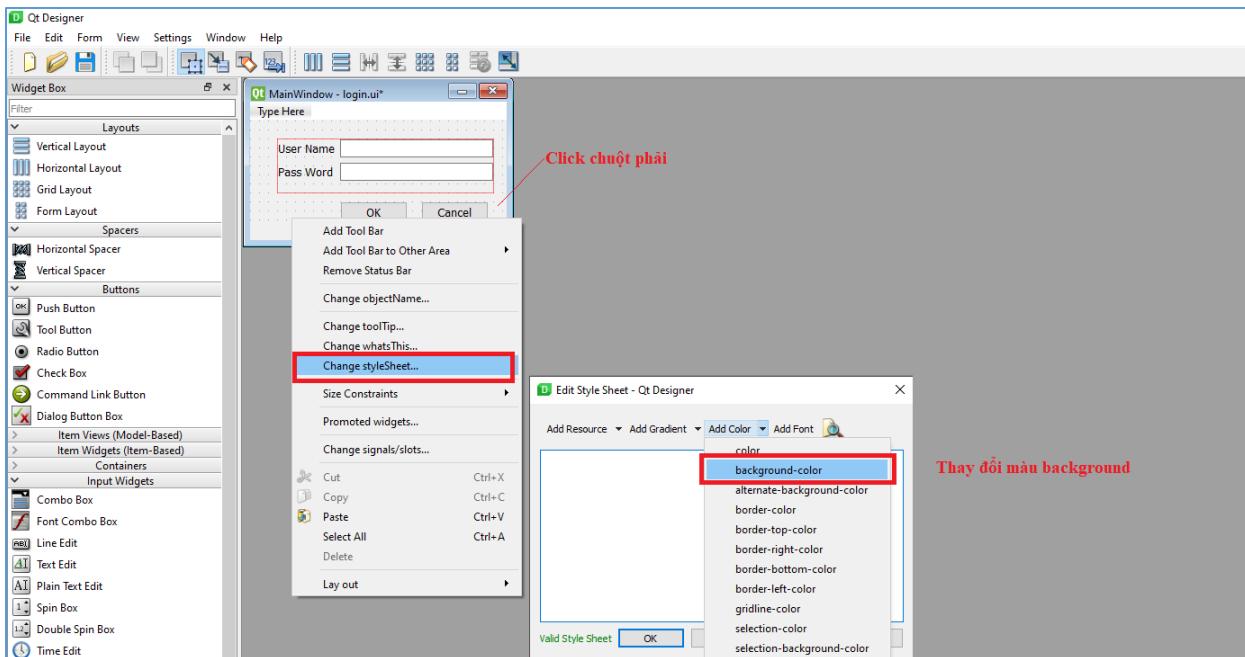
➤ Cách tạo pushButton

Tương tự cách kéo thả trên bạn đến **Input Widgets**, chọn **pushButton** và đặt tên đối tượng **pushBtn_Ok** và **pushBtn_Cancel**

Cuối cùng bạn kéo thả **01 Form Layout** đặt tên là **formLayout_longin** và kéo thả các đối tượng trên vào để tạo thành 1 trình quản lý riêng biệt đẹp mắt hơn.



➤ Cách thay đổi trên form chính (MainWindows)



➤ Cách chuyển file Ui thành file code Python

Khởi động Command Prompt và nhập vào dòng lệnh cài đặt **pip install PyQt5**

```
C:\> Command Prompt
Microsoft Windows [Version 10.0.19042.1237]
(c) Microsoft Corporation. All rights reserved.

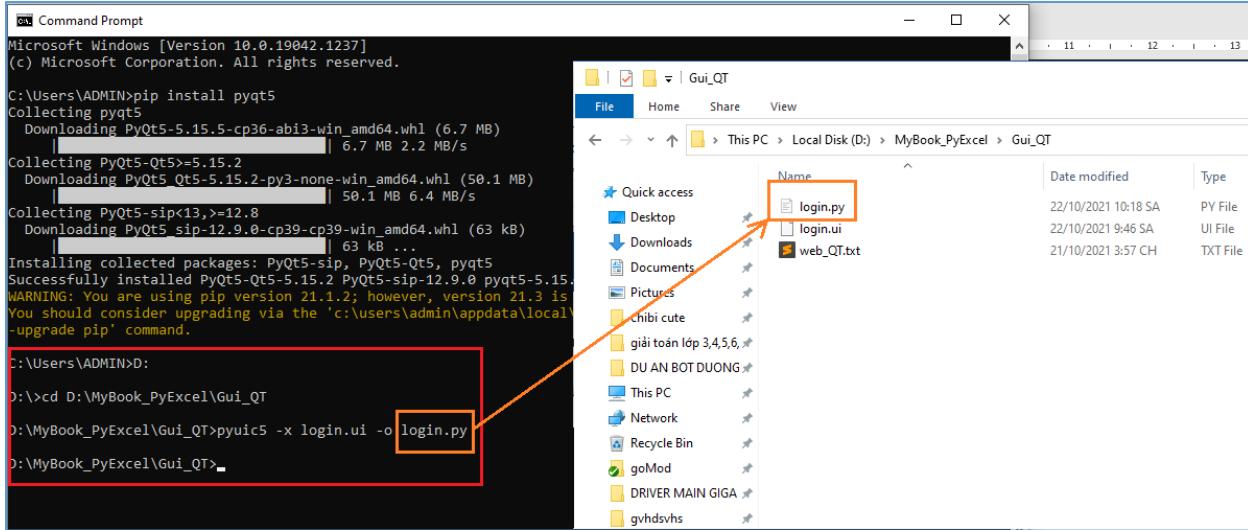
C:\Users\ADMIN>pip install PyQt5
Collecting PyQt5
  Downloading PyQt5-5.15.5-cp36-abi3-win_amd64.whl (6.7 MB)
    |██████████| 6.7 MB 2.2 MB/s
Collecting PyQt5-Qt5>=5.15.2
  Downloading PyQt5_Qt5-5.15.2-py3-none-win_amd64.whl (50.1 MB)
    |██████████| 50.1 MB 6.4 MB/s
Collecting PyQt5-sip<13,>=12.8
  Downloading PyQt5_sip-12.9.0-cp39-cp39-win_amd64.whl (63 kB)
    |██████████| 63 kB ...
Installing collected packages: PyQt5-sip, PyQt5-Qt5, PyQt5
Successfully installed PyQt5-Qt5-5.15.2 PyQt5-sip-12.9.0 PyQt5-5.15.5
WARNING: You are using pip version 21.1.2; however, version 21.3 is available.
You should consider upgrading via the 'c:\users\admin\appdata\local\programs\python\python39\python.exe -m pip install --upgrade pip' command.

C:\Users\ADMIN>
```

Sau đó bạn chuyển qua ổ đĩa có chứa file **login.ui**, trên máy mình là ổ D nên mình tiếp tục gõ **D:**

Copy đường dẫn tại thư mục có chứa file **login.ui** nhập vào dòng lệnh **cd <Tên đường dẫn>**

Tiếp theo là câu lệnh chuyển file **ui** thành file **py** (mình đặt tên file python là **login.py**). Lệnh của mình gõ vào là **pyuic5 -x login.ui -o login.py** như hình dưới.



Khi trên folder của bạn xuất hiện file **login.py** được tạo ra bạn click vào xem code đã được chuyển hóa từ **Gui** sang **python** như thế nào?

Code được tạo ra

```
# -*- coding: utf-8 -*-

# Form implementation generated from reading ui file 'login.ui'
#
# Created by: PyQt5 UI code generator 5.15.5
#
# WARNING: Any manual changes made to this file will be lost when pyuic5 is
# run again. Do not edit this file unless you know what you are doing.

from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(415, 161)
        self.centralwidget = QtWidgets.QWidget(MainWindow)
```

```

        self.centralwidget.setObjectName("centralwidget")
        self.formLayoutWidget = QtWidgets.QWidget(self.centralwidget)
        self.formLayoutWidget.setGeometry(QtCore.QRect(30, 20, 361, 61))
        self.formLayoutWidget.setObjectName("formLayoutWidget")
        self.formLayout_longin = QtWidgets.QFormLayout(self.formLayoutWidget)
        self.formLayout_longin.setContentsMargins(0, 0, 0, 0)
        self.formLayout_longin.setObjectName("formLayout_longin")
        self.lb_UserName = QtWidgets.QLabel(self.formLayoutWidget)
        font = QtGui.QFont()
        font.setPointSize(10)
        font.setBold(False)
        font.setWeight(50)
        self.lb_UserName.setFont(font)
        self.lb_UserName.setLayoutDirection(QtCore.Qt.LeftToRight)
        self.lb_UserName.setAutoFillBackground(False)
        self.lb_UserName.setTextFormat(QtCore.Qt.AutoText)
        self.lb_UserName.setObjectName("lb_UserName")
        self.formLayout_longin.setWidget(0, QtWidgets.QFormLayout.LabelRole,
self.lb_UserName)
        self.cbb_UserName = QtWidgets.QComboBox(self.formLayoutWidget)
        self.cbb_UserName.setObjectName("cbb_UserName")
        self.formLayout_longin.setWidget(0, QtWidgets.QFormLayout.FieldRole,
self.cbb_UserName)
        self.lb_Password = QtWidgets.QLabel(self.formLayoutWidget)
        font = QtGui.QFont()
        font.setPointSize(10)
        font.setBold(False)
        font.setWeight(50)
        self.lb_Password.setFont(font)
        self.lb_Password.setLayoutDirection(QtCore.Qt.LeftToRight)
        self.lb_Password.setAutoFillBackground(False)
        self.lb_Password.setTextFormat(QtCore.Qt.AutoText)
        self.lb_Password.setObjectName("lb_Password")
        self.formLayout_longin.setWidget(1, QtWidgets.QFormLayout.LabelRole,
self.lb_Password)
        self.lineEdit_Password = QtWidgets.QLineEdit(self.formLayoutWidget)

```

```

        self.lineEdit_Password.setObjectName("lineEdit_Password")
        self.formLayout_longin.setWidget(1, QtWidgets.QFormLayout.FieldRole,
self.lineEdit_Password)
        self.pushBtn_Ok = QtWidgets.QPushButton(self.centralwidget)
        self.pushBtn_Ok.setGeometry(QtCore.QRect(210, 90, 75, 23))
        font = QtGui.QFont()
        font.setPointSize(10)
        self.pushBtn_Ok.setFont(font)
        self.pushBtn_Ok.setObjectName("pushBtn_Ok")
        self.pushBtn_Cancel = QtWidgets.QPushButton(self.centralwidget)
        self.pushBtn_Cancel.setGeometry(QtCore.QRect(300, 90, 75, 23))
        font = QtGui.QFont()
        font.setPointSize(10)
        self.pushBtn_Cancel.setFont(font)
        self.pushBtn_Cancel.setObjectName("pushBtn_Cancel")
        MainWindow.setCentralWidget(self.centralwidget)
        self.menubar = QtWidgets.QMenuBar(MainWindow)
        self.menubar.setGeometry(QtCore.QRect(0, 0, 415, 21))
        self.menubar.setObjectName("menubar")
        MainWindow.setMenuBar(self.menubar)
        self.statusbar = QtWidgets.QStatusBar(MainWindow)
        self.statusbar.setObjectName("statusbar")
        MainWindow.setStatusBar(self.statusbar)

        self.retranslateUi(MainWindow)
        QtCore.QMetaObject.connectSlotsByName(MainWindow)

    def retranslateUi(self, MainWindow):
        _translate = QtCore.QCoreApplication.translate
        MainWindow.setWindowTitle(_translate("MainWindow", "MainWindow"))
        self.lb_UserName.setText(_translate("MainWindow", "User Name"))
        self.lb_Password.setText(_translate("MainWindow", "Pass Word"))
        self.pushBtn_Ok.setText(_translate("MainWindow", "OK"))
        self.pushBtn_Cancel.setText(_translate("MainWindow", "Cancel"))

```

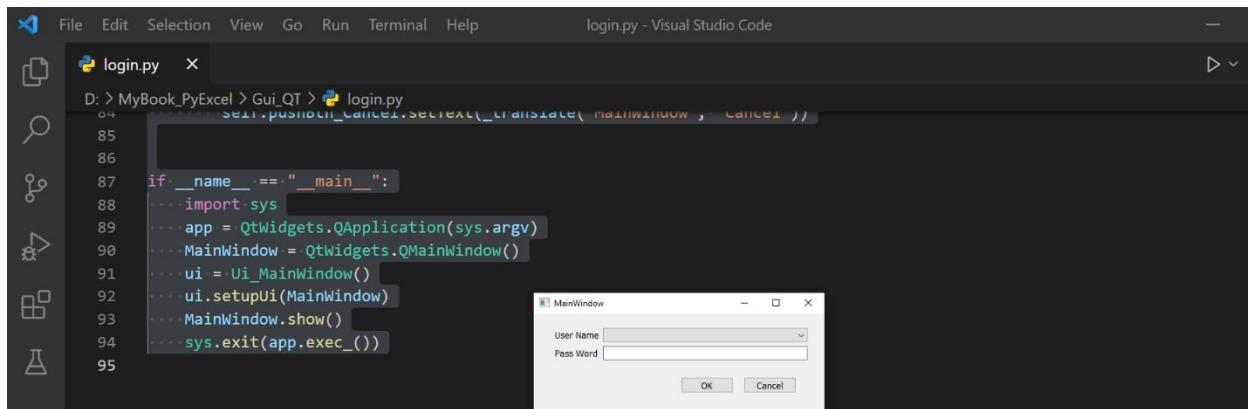
```

if __name__ == "__main__":
    import sys
    app = QtWidgets.QApplication(sys.argv)
    MainWindow = QtWidgets.QMainWindow()
    ui = Ui_MainWindow()
    ui.setupUi(MainWindow)
    MainWindow.show()
    sys.exit(app.exec_())

```

Bạn tự Breakpoint và tìm hiểu cách tạo code như trên nào nhé!

Kết quả thực thi code bằng nút Run:



Bây giờ là bảng đăng nhập đã hiện ra, nếu bạn muốn tạo thành công cụ hoàn chỉnh thì xem sửa code phần tạo ra **Combo Box** có tên là **cbb_UserName** thành Line Edit cho phù hợp để nhập vào thông tin.

Để giúp bạn hình dung và tìm hiểu phần cốt lõi trong code trên thì giao diện PyQt5 cấu thành từ các module, bao gồm:

- + **QtCore**: là module bao gồm phần lõi không thuộc chức năng GUI, ví dụ dùng để làm việc với thời gian, file và thư mục, các loại dữ liệu, streams, URLs, mime type, threads hoặc processes.
- + **QtGui**: bao gồm các class dùng cho việc lập trình giao diện (windowing system integration), event handling, 2D graphics, basic imaging, fonts và text.

- + QtWidgets: bao gồm các class cho widget, ví dụ: button, hộp thoại, ... được sử dụng để tạo nên giao diện người dùng cơ bản nhất.
- + QtMultimedia: thư viện cho việc sử dụng âm thanh, hình ảnh, camera,...
- + QtBluetooth: bao gồm các class giúp tìm kiếm và kết nối với các thiết bị có giao tiếp với phần mềm.
- + QtNetwork: bao gồm các class dùng cho việc lập trình mạng, hỗ trợ lập trình TCP/IP và UDP client , server hỗ trợ việc lập trình mạng.
- + QtPositioning: bao gồm các class giúp việc hỗ trợ xác định vị.
- + Enginio: module giúp các client truy cập các Cloud Services của Qt.
- + QtWebSockets: cung cấp các công cụ cho WebSocket protocol.
- + QtWebKit: cung cấp các class dùng cho làm việc với các trình duyệt Web , dựa trên thư viện WebKit2.
- + QtWebKitWidgets: các widget cho WebKit.
- + QDom: các class dùng cho làm việc với XML file.
- + QtSvg: dùng cho hiển thị các thành phần của SVG file.
- + QSql: cung cấp các class dùng cho việc làm việc với dữ liệu.
- + QTest: cung cấp các công cụ cho phép test các đơn vị của ứng dụng với PyQt5.

➤ Cách gọi trực tiếp file ui

Ngoài cách gọi file trên bạn cũng có thể gọi trực tiếp bằng đoạn code sau:

```
import sys
from PyQt5 import QtCore, QtGui, QtWidgets, uic
pyQTfileName = "login.ui"

Ui_MainWindow, QtBaseClass = uic.loadUiType(pyQTfileName)

class MyApp(QtWidgets.QMainWindow, Ui_MainWindow):
    def __init__(self):
        QtWidgets.QMainWindow.__init__(self)
```

```

Ui_MainWindow.__init__(self)
self.setupUi(self)

if __name__ == "__main__":
    app = QtWidgets.QApplication(sys.argv)
    window = MyApp()
    window.show()
    sys.exit(app.exec_())

```

Hoặc cách đơn giản nhất:

```

import sys
from PyQt5 import uic
from PyQt5.QtWidgets import QMainWindow
from PyQt5.QtWidgets import QApplication,QMainWindow

class formUi(QMainWindow):
    def __init__(self):
        super(formUi,self).__init__()
        # Load Ui file
        uic.loadUi("login.ui", self)
        # Show App
        self.show()

    # Initialize app
app = QApplication(sys.argv)
uiWindows = formUi()
app.exec_()

```

➤ Cài đặt PyQt5 tools

Ngoài ra, bạn cần cài đặt thêm **pyqt5 tools** công cụ hỗ trợ phát triển. Trong Command Prompt bạn dùng lệnh **pip install pyqt5 tools**

```
Command Prompt
Microsoft Windows [Version 10.0.19042.1237]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ADMIN>pip install pyqt5 tools
Requirement already satisfied: PyQt5-sip<13,>=12.8 in c:\users\admin\appdata\local\programs\python\python39\lib\site-packages (5.15.5)
Requirement already satisfied: PyQt5 in c:\users\admin\appdata\local\programs\python\python39\lib\site-packages (from PyQt5) (12.9.0)
Requirement already satisfied: PyQt5-Qt5>=5.15.2 in c:\users\admin\appdata\local\programs\python\python39\lib\site-packages (from PyQt5) (5.15.2)
Collecting pytils
  Downloading pytils-0.3.tar.gz (89 kB)
    |██████████| 89 kB 3.0 MB/s
Requirement already satisfied: six in c:\users\admin\appdata\local\programs\python\python39\lib\site-packages (from pytils) (1.15.0)
Requirement already satisfied: lxml in c:\users\admin\appdata\local\programs\python\python39\lib\site-packages (from pytils) (4.6.3)
Building wheels for collected packages: tools, pytils
  Building wheel for tools (setup.py) ... done
  Created wheel for tools: filename=tools-0.1.9-py3-none-any.whl size=46758 sha256=d880b0bdb4a7bfb6d7d2673568ecf8b017b779c2b410638feb7ce2ce9cf893a
  Stored in directory: c:\users\admin\appdata\local\pip\cache\wheels\77\13\56\879617e6017b5dde1711ca8c9ee5e9838e913c85ae22fd96c9
  Building wheel for pytils (setup.py) ... done
  Created wheel for pytils: filename=pytils-0.3-py3-none-any.whl size=40355 sha256=b896d30c23bb473479992fae91e586d1d5e5e999f12d2db51261c92d3a50f71f
  Stored in directory: c:\users\admin\appdata\local\pip\cache\wheels\03\ee\4e1162bc3b1a0bc3f7a7be9eadze1c73bec57364efd416f8d
Successfully built tools pytils
Installing collected packages: pytils, tools
Successfully installed pytils-0.3 tools-0.1.9
WARNING: You are using pip version 21.1.2; however, version 21.3 is available.
You should consider upgrading via the 'c:\users\admin\appdata\local\programs\python\python39\python.exe -m pip install --upgrade pip' command.

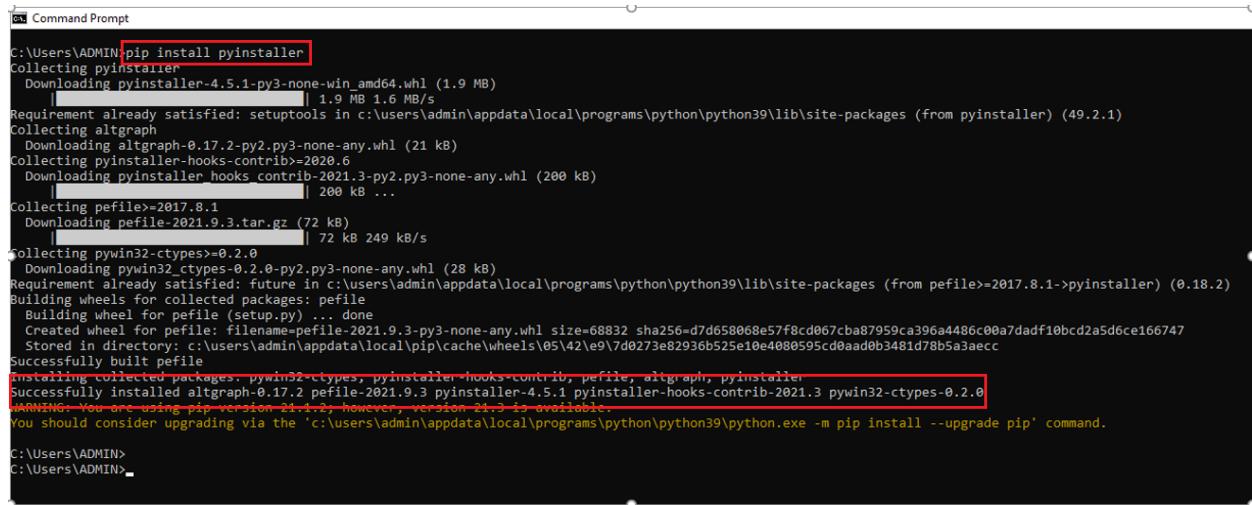
C:\Users\ADMIN>
```

Đó tất cả là những gì mình giới thiệu phần Gui để các bạn có thể tự tìm hiểu và phát triển phần mềm cho chính bạn.

CHƯƠNG 7: XUẤT ỦNG DỤNG

Sau khi bạn đã thành công về giao diện đồ họa thì đã đến lúc bạn cần phải xuất ứng dụng mình đã viết từ code đến giao diện cho người dùng sử dụng, các bạn cài đặt cho mình gói **pyinstaller** như sau:

Sau khi khởi động Command Prompt bạn nhập pip install pyinstaller

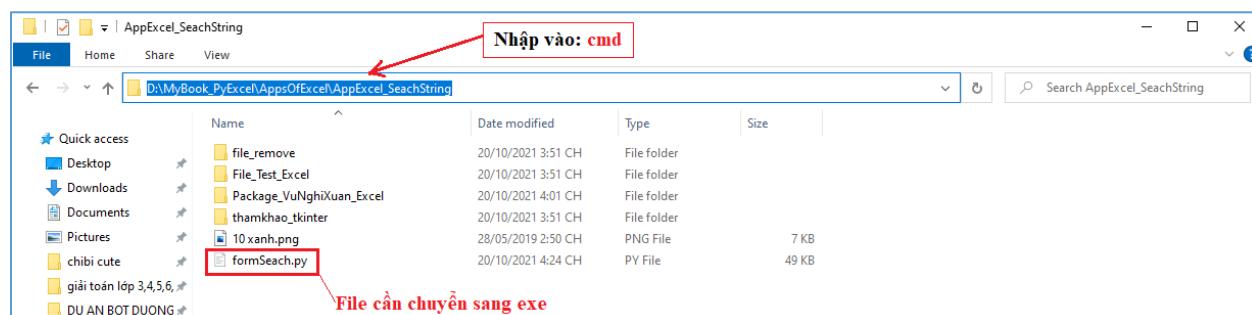


```
C:\Users\ADMIN>pip install pyinstaller
Collecting pyinstaller
  Downloading pyinstaller-4.5.1-py3-none-win_amd64.whl (1.9 MB)
    |████████| 1.9 MB 1.6 MB/s
Requirement already satisfied: setuptools in c:\users\admin\appdata\local\programs\python\python39\lib\site-packages (from pyinstaller) (49.2.1)
Collecting altgraph
  Downloading altgraph-0.17.2-py2.py3-none-any.whl (21 kB)
Collecting pyinstaller-hooks-contrib==2020.6
  Downloading pyinstaller_hooks_contrib-2021.3-py2.py3-none-any.whl (200 kB)
    |████████| 200 kB ...
Collecting pefile>=2017.8.1
  Downloading pefile-2021.9.3.tar.gz (72 kB)
    |████████| 72 kB 249 kB/s
Collecting pywin32-ctypes>=0.2.0
  Downloading pywin32_ctypes-0.2.0-py3-none-any.whl (28 kB)
Requirement already satisfied: future in c:\users\admin\appdata\local\programs\python\python39\lib\site-packages (from pyinstaller) (0.18.2)
Building wheels for collected packages: pefile
  Building Wheel for pefile (setup.py) ... done
    Created wheel for pefile: filename=pefile-2021.9.3-py3-none-any.whl size=68832 sha256=d7d658068e57f8cd067cba87959ca396a4486c00a7dadf10bcd2a5d6ce166747
  Stored in directory: c:\users\admin\appdata\local\pip\cache\wheels\85142\ea9\7d0273e82936b525e10e4080595cd0aad0b3481d78b5a3aecc
Successfully built pefile
Installing collected packages: pywin32-ctypes, pyinstaller-hooks-contrib, pefile, altgraph, pyinstaller
Successfully installed altgraph-0.17.2 pefile-2021.9.3 pyinstaller-4.5.1 pyinstaller-hooks-contrib-2021.3 pywin32-ctypes-0.2.0
WARNING: You are using pip version 21.1.3; however, version 21.3 is available.
You should consider upgrading via the 'c:\users\admin\appdata\local\programs\python\python39\python.exe -m pip install --upgrade pip' command.

C:\Users\ADMIN>
C:\Users\ADMIN>
```

➤ Cách chuyển file Python thành file .exe

Đầu tiên bạn vào thư mục chứa Script Python chứa code cần chuyển sang file .exe, trên máy mình là file **formSeach.py** nằm trong đường dẫn (hình dưới) và nhập **cmd** để chuyển đường dẫn trực tiếp vào cửa sổ **Command Prompt**



Sau khi cửa sổ **Command Prompt** khởi động bạn gõ dòng lệnh: **pyinstaller –onefile –w tên file.py**

Kết quả chuyển file thành công

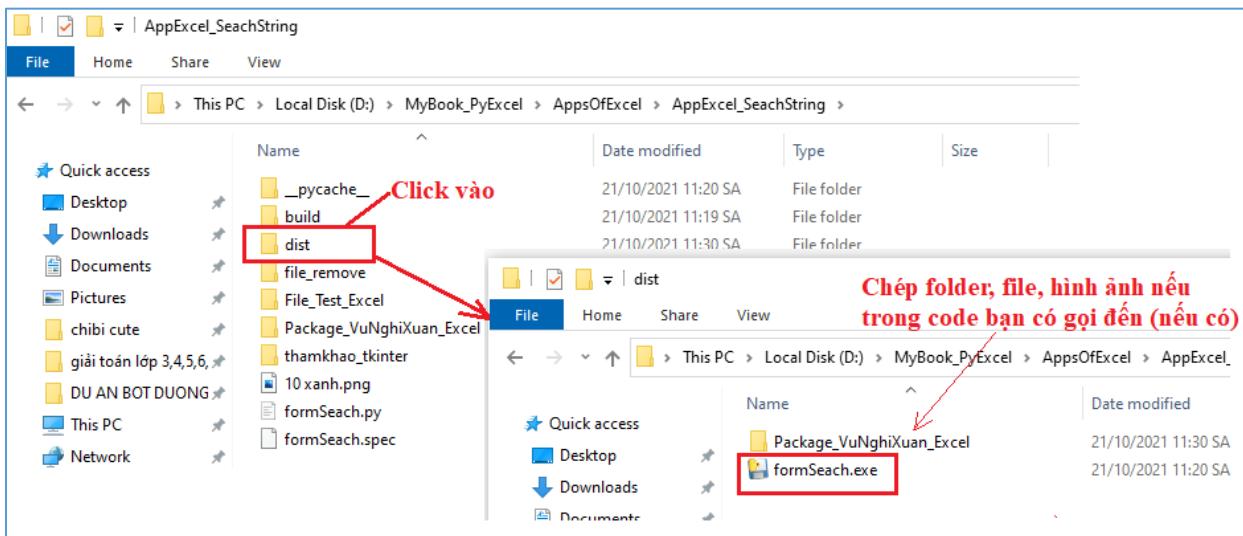
```

C:\Windows\System32\cmd.exe
D:\MyBook_PyExcel\AppsOfExcel\AppExcel_SearchString>pyinstaller --onefile -w formSearch.py
62 INFO: PyInstaller: 4.5.1
63 INFO: Python: 3.9.2
63 INFO: Platform: Windows-10-10.0.19041-SP0
63 INFO: wrote D:\MyBook_PyExcel\AppsOfExcel\AppExcel_SearchString\formSearch.spec
65 INFO: UPX is not available.
66 INFO: Extending PYTHONPATH with paths
['D:\\MyBook_PyExcel\\AppsOfExcel\\AppExcel_SearchString',
 'D:\\MyBook_PyExcel\\AppsOfExcel\\AppExcel_SearchString']
506 INFO: checking Analysis
507 INFO: Building Analysis because Analysis-00.toc is non existent
507 INFO: Initializing module dependency graph...
509 INFO: Caching module graph hooks...
73413 INFO: Copying icons from 'c:\\users\\admin\\appdata\\local\\programs\\python\\python39\\lib\\site-packages\\PyInstaller\\bootloader\\images\\icon-windowed.ico'
73425 INFO: Writing RT_GROUP_ICON 0 resource with 104 bytes
73426 INFO: Writing RT_ICON 1 resource with 3752 bytes
73427 INFO: Writing RT_ICON 2 resource with 2216 bytes
73428 INFO: Writing RT_ICON 3 resource with 1384 bytes
73429 INFO: Writing RT_ICON 4 resource with 38188 bytes
73427 INFO: Writing RT_ICON 5 resource with 9640 bytes
73427 INFO: Writing RT_ICON 6 resource with 4264 bytes
73427 INFO: Writing RT_ICON 7 resources with 1128 bytes
73431 INFO: Updating manifest in D:\MyBook_PyExcel\AppsOfExcel\AppExcel_SearchString\build\formSearch\runw.exe.qbxqdu0p
73431 INFO: Updating resource type 24 name 1 language 0
73441 INFO: Appending archive to EXE D:\MyBook_PyExcel\AppsOfExcel\AppExcel_SearchString\dist\formSearch.exe
85364 INFO: Building EXE from EXE-00.toc completed successfully.

D:\MyBook_PyExcel\AppsOfExcel\AppExcel_SearchString>

```

Vì trong file code của mình có gọi hàm hỗ trợ được xây dựng từ 1 thư viện riêng chứa trong folder VuNghiXuan, nên mình copy nó vào cùng file formSearch.exe để chương trình chạy không bị báo lỗi. Nếu file code của bạn có sử dụng file hình ảnh, hoặc file để trích xuất dữ liệu thì bạn làm giống mình.



Cuối cùng mình khởi động file **formSearch.exe** xem nó có hoạt động thế nào? Đây là kết quả phần mềm của mình.

App's Excel VuNghiXuan

Danh mục Playlists files!!!

Tìm kiếm: English

I/ File on This Computer (DesktopHome_MrVu). 2 file

- [File1] /BachDem.xlsx
 - [Sheet1] - 19000+ rows
 - [File2] /2_movies.xls
 - [Sheet1] 1338 (row)
 - [Rows]: 1338 (row)
 - [Cells]: 25 (column)
- II/ File on Other Computer: 2 file
 - [File1] /movies.xls
 - [File2] /movies.xls

Xóa file Thêm file Lưu file Chọn nhiều files

Activate Windows
Go to Settings to activate Windows.

Title	Year	Genres	Language	Country	Content Rating	Duration
The Broadway Melody	1929	Musical Romance	English	USA	Passed	100
He's Angel	1930	Drama War	English	USA	Passed	96
A Farewell to Arms	1932	Drama Romance War	English	USA	Unrated	79
42nd Street	1933	Comedy Musical Romance	English	USA	Unrated	89
Don't Give Him Writing	1933	Comedy Romance Musical Romance	English	USA	Approved	66
It Happened One Night	1934	Comedy Romance	English	USA	Unrated	65
Top Hat	1935	Comedy Musical Romance	English	USA	Approved	81
Modern Times	1936	Comedy Drama Family	English	USA	G	87
The Charge of the Light Brigade	1936	Action Adventure Romance War	English	USA	Approved	100
Snow White and the Seven Dwarfs	1937	Animation Family Fantasy Musical	English	USA	Approved	83
The Prisoner of Zenda	1937	Adventure Drama Romance	English	USA	Approved	101
Alexander's Ragtime Band	1938	Drama Musical Romance	English	USA	Approved	106
You Can't Take It with You	1938	Comedy Drama Romance	English	USA	Approved	126
Gone with the Wind	1939	Drama History Romance War	English	USA	G	226
Mr. Smith Goes to Washington	1939	Comedy Drama	English	USA	Not Rated	120
The Wizard of Oz	1939	Adventure Family Fantasy Musical	English	USA	Passed	102
Boom Town	1940	Adventure Drama Romance Western	English	USA	Passed	119
Fantasia	1940	Animation Family Fantasy Music	English	USA	G	120
Priscilla	1940	Animation Family Fantasy Musical	English	USA	Approved	88
Robesco	1940	Drama Film-Noir Mystery Thriller	English	USA	Not Rated	130
The Blue Bird	1940	Drama Family Fantasy	English	USA	Approved	83
How Green Was My Valley	1941	Drama Family	English	USA	Approved	118
Bambi	1942	Animation Drama Family	English	USA	Approved	70
Casablanca	1942	Drama Romance War	English	USA	PG	82
A Guy Named Joe	1943	Drama Fantasy Romance War	English	USA	Passed	122
Bathing Beauty	1944	Comedy Musical	English	USA	Passed	101
Spellbound	1945	Film-Noir Mystery Romance Thriller	English	USA	Unrated	95
State Fair	1945	Musical Romance	English	USA	Approved	100
The Lost Weekend	1945	Drama Film-Noir	English	USA	Not Rated	101
The Valley of Decision	1945	Drama	English	USA	Passed	119
Duel in the Sun	1946	Drama Romance Western	English	USA	Unrated	144
It's A Wonderful Life	1946	Drama Family Romance	English	USA	PG	118
The Best Years of Our Lives	1946	Drama Family War	English	USA	Not Rated	172
Confidential Agreement	1947	Drama Romance	English	USA	Approved	118
The Lady from Shanghai	1947	Crime Drama Film-Noir Mystery Thriller	English	USA	Not Rated	92
Tycoon	1947	Adventure Drama Romance	English	USA	Approved	95
Open Secret	1948	Crime Drama	English	USA	Approved	68
Red River	1948	Action Adventure Romance Western	English	USA	Approved	126
The Pirate	1948	Adventure Comedy Musical Romance	English	USA	Approved	102

Phần này, tạm thời chỉ để lại code cho các bạn tham khảo trước. Phần sau, mình sẽ giải thích chi tiết các đoạn code, thông qua đó xây dựng từng hàm một, mỗi hàm 1 chức năng riêng biệt, thuận tiện cho việc update phát triển thành 1 phần mềm, đồng thời dễ kiểm soát, nhanh chóng tìm ra lỗi và sửa lỗi.

Thân chào, hẹn gặp lại !!!

Tp.Thủ Đức, ngày 11/10/2021

Biên tập thêm tại Tp.Hà Nội, ngày 22/11/2021.