

# Ngôn Ngữ Lập Trình Python

## Giới Thiệu

Trịnh Tấn Đạt

Đại Học Sài Gòn

[trinhtandat@sgu.edu.vn](mailto:trinhtandat@sgu.edu.vn)

<http://sites.google.com/site/ttdat88>

# Thông Tin Môn Học

- 4 tín chỉ: 75 tiết (LT: 45; TH: 30)
- Nội dung chính
  - (1) Giới thiệu ngôn ngữ Python,
  - (2) Cấu trúc và cú pháp trong ngôn ngữ Python,
  - (3) Nguyên lý hướng đối tượng trong Python,
  - (4) Các thư viện phổ biến trong Python,
  - (5) Xây dựng các ứng dụng

# Mục Đích Môn Học

- **Về kiến thức:**

- Nắm vững kiến thức về ngôn ngữ Python.
- Định hướng cho sinh viên hoàn thành các dự án môn học, nâng cao kỹ năng lập trình.

- **Về kỹ năng:**

- Sinh viên sử dụng thành thạo ngôn ngữ Python để viết các chương trình ứng dụng.

- **Về thái độ:**

- Rèn luyện thái độ và tư duy làm việc nghiêm túc, khoa học.

# Tài Liệu Học Tập

- Các tài liệu tham khảo
  - Mark Lutz, Learning Python: Powerful Object-Oriented Programming. O'Reilly Media, Inc. 2013
  - Think Python: How to think like a computer scientist
  - A Byte of Python
  - Python cơ bản- Võ Duy Tuấn
- Các nguồn từ Internet
  - <http://scipy-lectures.org/>
  - Python tutorial: <https://www.programiz.com/python-programming/first-program>
  - Github
  - ...

# Nội Dung

- Chương 1: Giới thiệu
- Chương 2: Cấu trúc và cú pháp
- Chương 3: Nguyên lý hướng đối tượng
- Chương 4: Các thư viện phổ biến trong python
- **Chương 5: Các xử lý cơ bản trong python (Đồ án môn học)**
- **Chương 6: Xây dựng các ứng dụng (Đồ án môn học)**

# Đồ án môn học

Sinh viên làm việc theo nhóm (tối đa 04 thành viên).

- Tìm hiểu và xây dựng các ứng dụng minh họa
- Viết báo cáo (Yêu cầu tối thiểu 25 trang và có trích dẫn tài liệu tham khảo đầy đủ)
- Chương trình demo
- Thuyết trình + Trả lời câu hỏi
- Yêu cầu có ý tưởng cải tiến (tính mới)

- Ứng dụng các thư viện máy học
- Ứng dụng phân tích dữ liệu
- Ứng dụng IoT
- Ứng dụng lập trình web
- Ứng dụng xử lý ngôn ngữ tự nhiên
- Ứng dụng xử lý ảnh và thị giác máy tính
- Ứng dụng lập trình game
- Các ứng dụng khác ...

# Đánh giá

- Điểm quá trình (50%)
  - Điểm danh (10%)
  - Kiểm tra giữa kỳ (40%): Trắc nghiệm + Tự luận
- Đồ án môn học (50%): Báo cáo + Thuyết trình cuối kỳ

# Giới thiệu ngôn ngữ python

- Python
  - ngôn ngữ thông dịch
  - ngôn ngữ lập trình hướng đối tượng.
  - hoàn toàn tạo kiểu động và sử dụng cơ chế cấp phát bộ nhớ tự động.
  - có cấu trúc dữ liệu cấp cao mạnh mẽ.
  - cú pháp lệnh là điểm cộng vô cùng lớn vì sự rõ ràng, dễ hiểu và cách gõ linh động.
  - thuận tiện cho người mới học lập trình.
- Có thể chạy trên nhiều hệ điều hành khác nhau : *Unix, Windows, Mac OS, Linux...*



# Giới thiệu ngôn ngữ python

- <https://www.24h.com.vn/kham-pha-cong-nghe/top-5-cong-nghe-lap-trinh-giup-sinh-vien-it-vuot-bao-cach-mang-40-c675a989128.html>
- <https://www.tiobe.com/tiobe-index/>

May 2019	May 2018	Change	Programming Language	Ratings	Change
1	1		Java	16.005%	-0.38%
2	2		C	14.243%	+0.24%
3	3		C++	8.095%	+0.43%
4	4		Python	7.830%	+2.64%
5	6	▲	Visual Basic .NET	5.193%	+1.07%
6	5	▼	C#	3.984%	-0.42%
7	8	▲	JavaScript	2.690%	-0.23%
8	9	▲	SQL	2.555%	+0.57%
9	7	▼	PHP	2.489%	-0.83%
10	13	▲	Assembly language	1.816%	+0.82%
11	15	▲	Objective-C	1.626%	+0.69%
12	12		Delphi/Object Pascal	1.406%	+0.39%
13	18	▲	Perl	1.394%	+0.48%
14	16	▲	MATLAB	1.366%	+0.44%
15	10	▼	Ruby	1.343%	+0.16%

# Giới thiệu ngôn ngữ python

- Ví dụ : một đoạn python script

```
import math
import os
from copy import deepcopy

import cv2
import numpy as np
from PIL import Image, ImageEnhance
from numpy import asarray, mean, array, argmax
from skimage.transform import radon

def distance(v1, v2):
    dist = np.sqrt(np.sum(np.square(np.subtract(v1, v2))))
    return dist

def box_area(box):
    return (int(box[2]) - int(box[0])) * (int(box[3]) - int(box[1]))
```

# Giới thiệu ngôn ngữ python

- Được tạo ra bởi Guido Van Rossum (sinh năm 1956). Thiết kế bắt đầu vào cuối những năm 1980 và được phát hành lần đầu tiên vào tháng 2 năm 1991.
- Python kế thừa từ ngôn ngữ ABC
- Tại sao lại có tên là Python?
  - Rossum là fan của một chương trình hài cuối những năm 1970, và cái tên “Python” được lấy từ tên một phân trong đó **“Monty Python’s Flying Circus”**.
- Python 2 được giới thiệu năm 2000
  - Hỗ trợ unicode
  - Mã python 2 rất phổ biến
- Python 3 được phát hành năm 2008
  - Hiện đã có phiên bản 3.7.3
- Bản phân phối python Anaconda
  - hỗ trợ rất nhiều thư viện (numpy, scipy, matplotlib , sklearn)
  - dùng nhiều trong machine learning, data science, AI



# Tại Sao Sử Dụng Python

Tính năng chính của Python (Ưu điểm)

- **Ngôn ngữ lập trình đơn giản, dễ học:** Python có cú pháp rất đơn giản, rõ ràng.
- **Miễn phí, mã nguồn mở:** Python có một cộng đồng rộng lớn, không ngừng cải thiện nó mỗi lần cập nhật.
- **Khả năng di chuyển:** Các chương trình Python có thể di chuyển từ nền tảng này sang nền tảng khác và chạy nó mà không có bất kỳ thay đổi nào.
- **Khả năng mở rộng và có thể nhúng:** Giả sử một ứng dụng đòi hỏi sự phức tạp rất lớn, bạn có thể dễ dàng kết hợp các phần code bằng C, C++ (**module ctypes**) vào code Python.
- **Ngôn ngữ thông dịch cấp cao:** Không giống như C/C++ (**ngôn ngữ biên dịch**), với Python, bạn không phải lo lắng những nhiệm vụ khó khăn như quản lý bộ nhớ, dọn dẹp những dữ liệu.
- **Thư viện tiêu chuẩn lớn để giải quyết những tác vụ phổ biến**
- **Hướng đối tượng:** Mọi thứ trong Python đều là hướng đối tượng

# Hạn chế

Ngôn ngữ này có những mặt hạn chế nhất định như sau:

- Python không có các thuộc tính như: protected, private hay public, không có vòng lặp do...while và switch....case.
- Python mặc dù nhanh hơn so với PHP, nhưng lại không nhanh hơn so với C/C++, Java.

# Sử Dụng Python Để Làm Gì

Python được dùng ở đâu?

- **Lập trình ứng dụng web:** Bạn có thể tạo web app có khả năng mở rộng (scalable). Các trang như Mozilla, Reddit, Instagram đều được viết bằng Python.
- **Khoa học và tính toán:** Có nhiều thư viện trong Python cho khoa học và tính toán số liệu, như SciPy và NumPy. Ngoài ra, Python còn được sử dụng nhiều trong machine learning, data mining và deep learning (scikit-learn, tensorflow , keras, pytorch, ...)
- **Tạo nguyên mẫu phần mềm:** Python là ngôn ngữ tuyệt vời để tạo những nguyên mẫu (bản chạy thử - prototype). Ví dụ, bạn có thể sử dụng Pygame (thư viện viết game) để tạo nguyên mẫu game trước. Nếu thích nguyên mẫu đó có thể dùng C++ để viết game thực sự.
- **Ngôn ngữ tốt để dạy lập trình:** Python được nhiều công ty, trường học sử dụng để dạy lập trình cho trẻ em và những người mới lần đầu học lập trình.

# Môi trường cài đặt

# Cài Đặt

1. Cài đặt Python 3 + IDE Pycharm

**2. Cài đặt Python 3 dùng bản phân phối Anaconda (RAM > 4GB)**

- **Cài đặt IDE Pycharm**
- Anaconda hỗ trợ IDE Spyder
- Jupyter Notebook

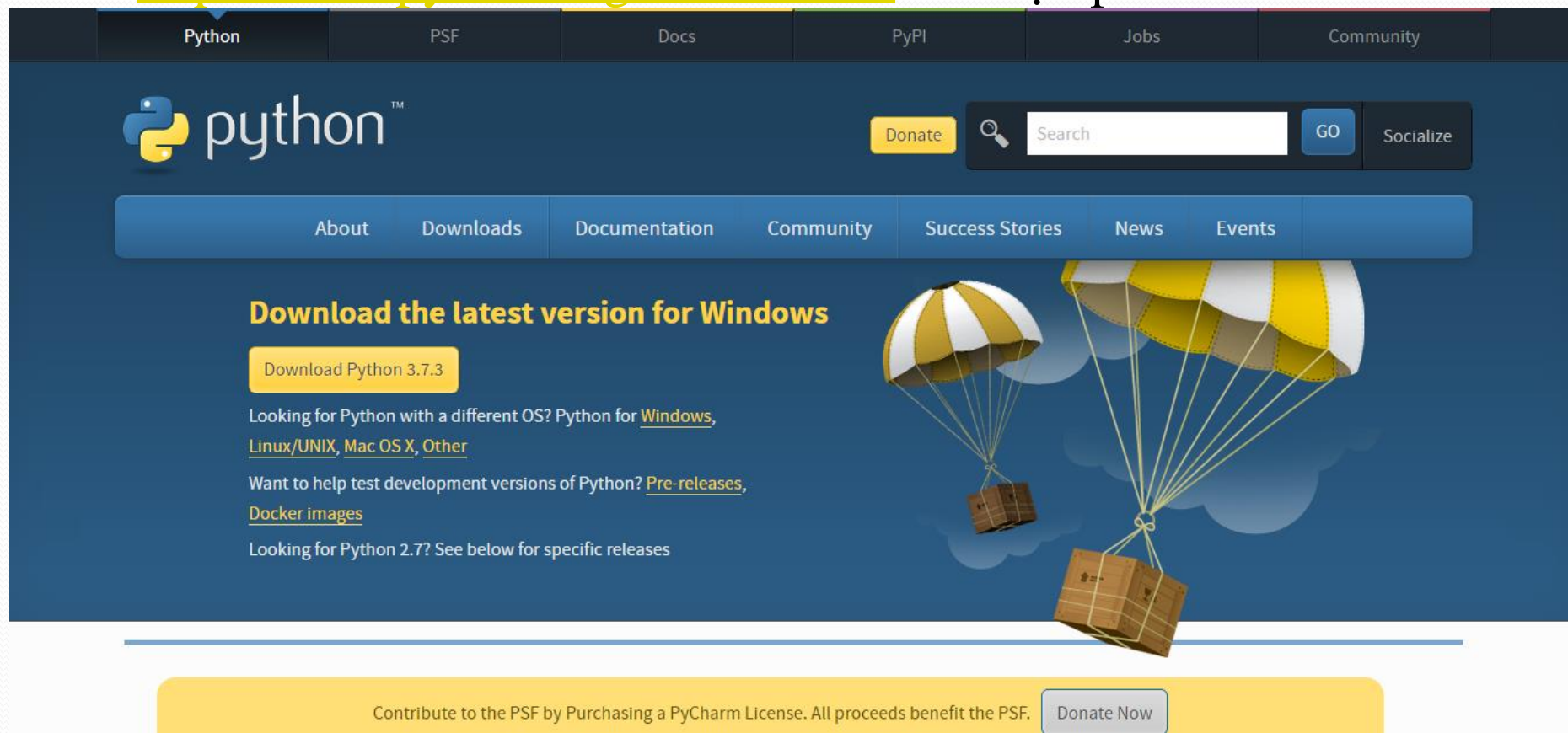
3. Sử dụng Google Colab, Kaggle

4. Visualize code execution : <https://pythontutor.com/>



# Cài đặt Python 3

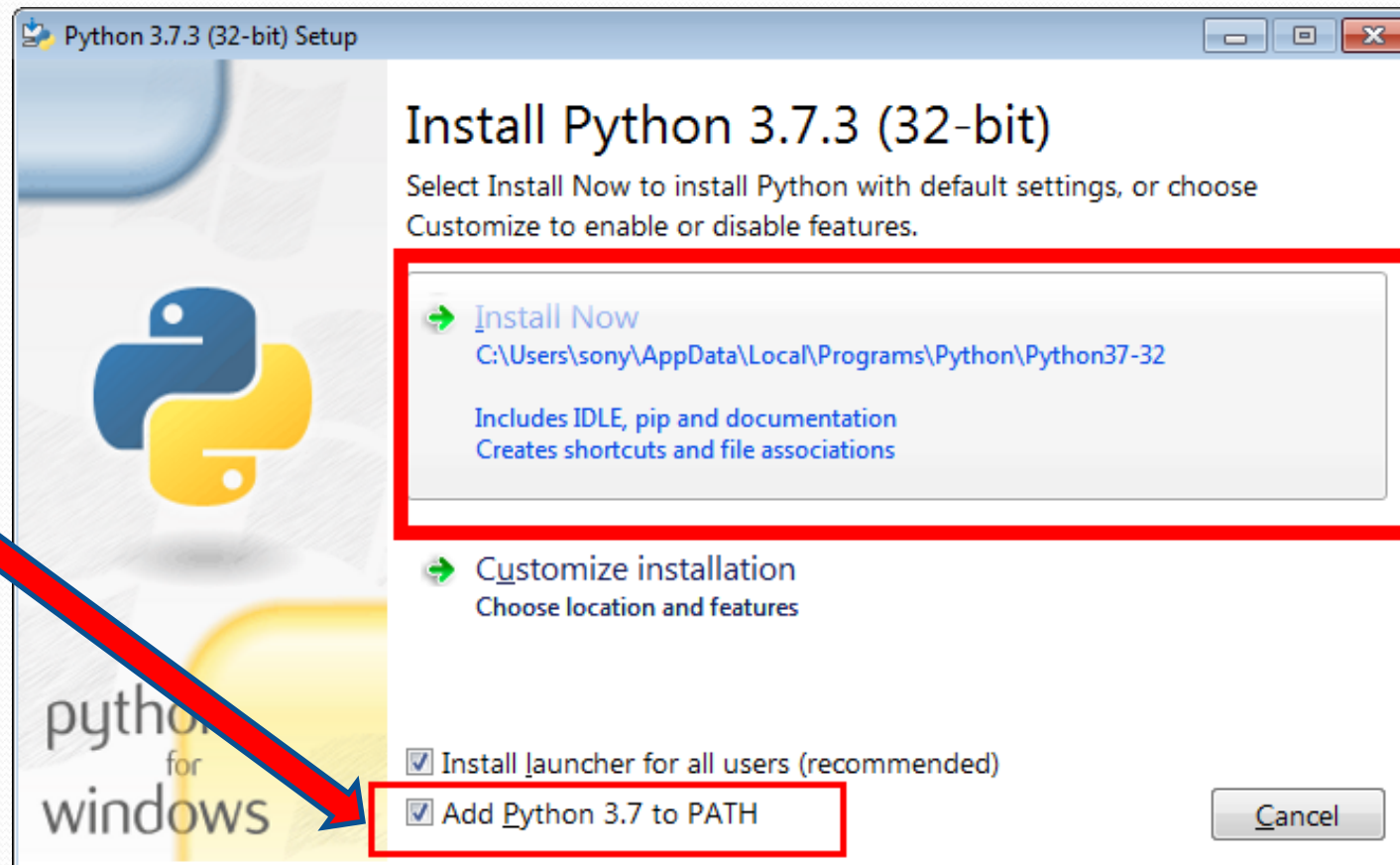
- Bước 1: Để tải và cài đặt Python 3 hãy truy cập vào trang web chính thức của Python: <http://www.python.org/downloads/> và chọn phiên bản

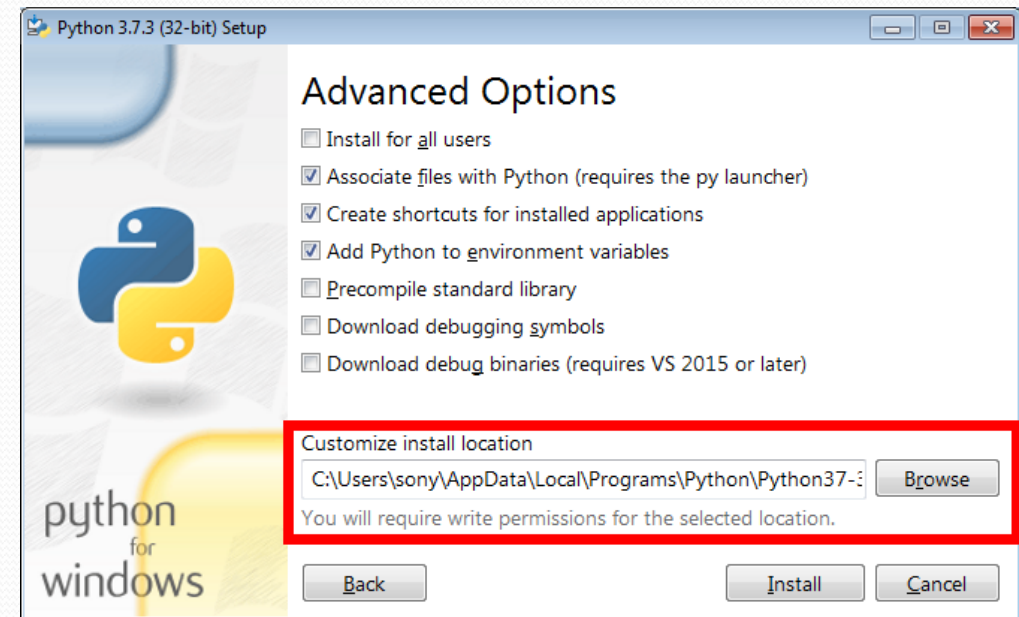
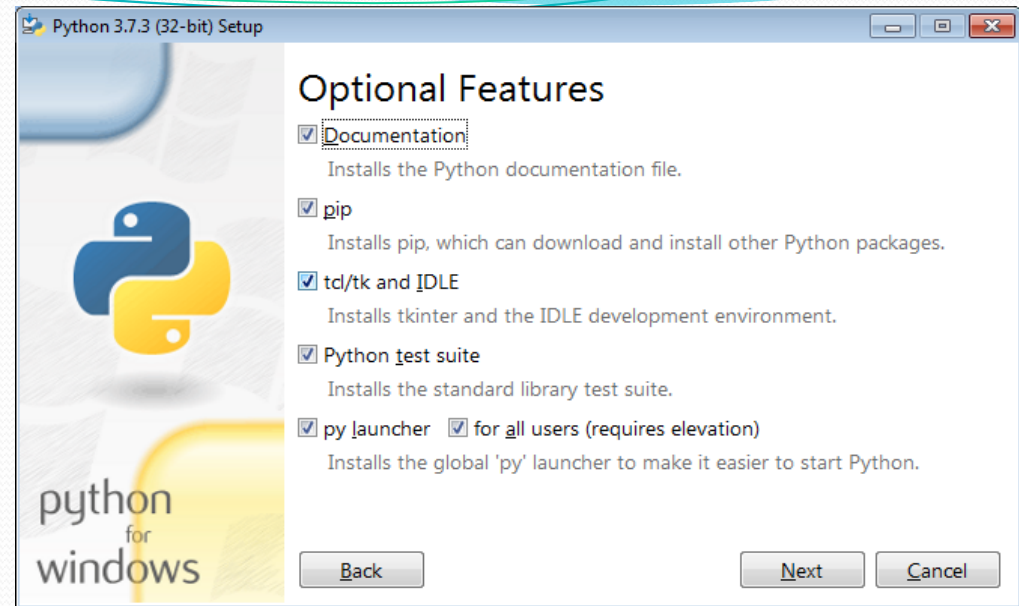
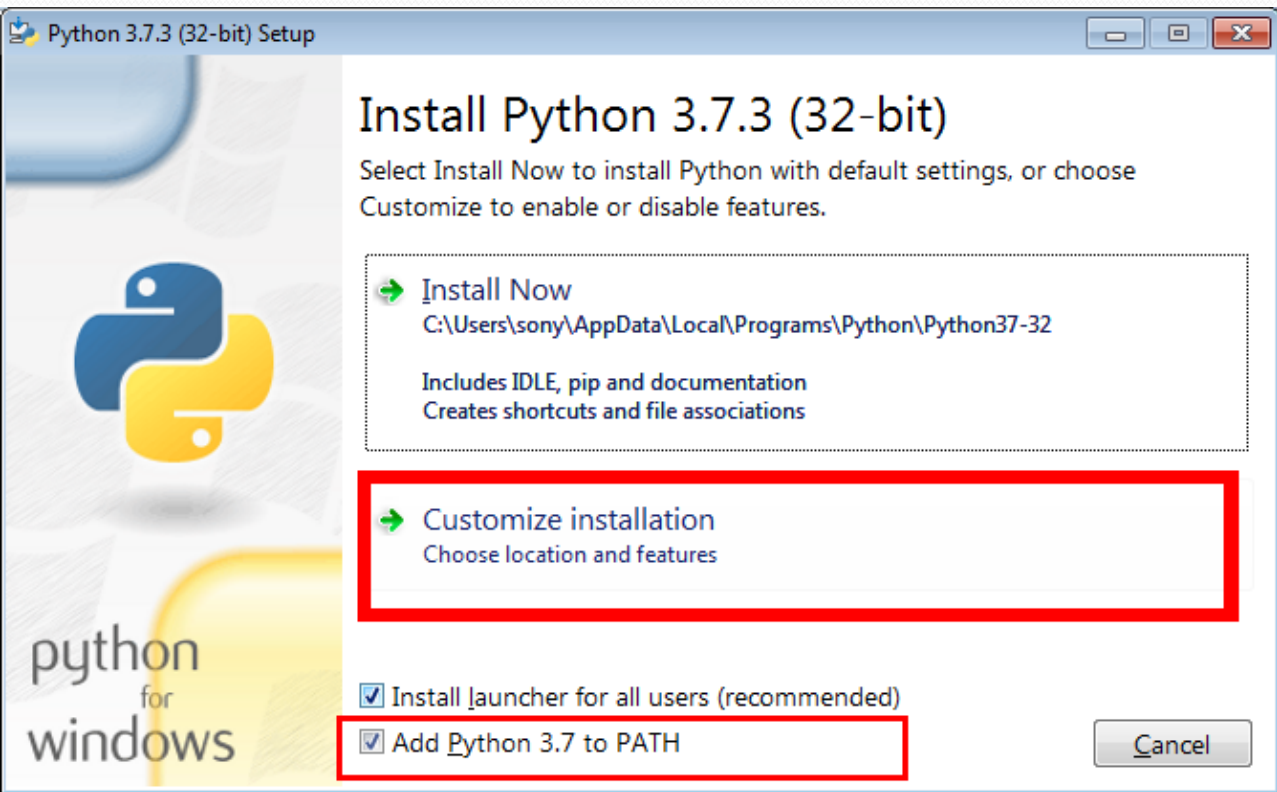


# Cài đặt Python 3

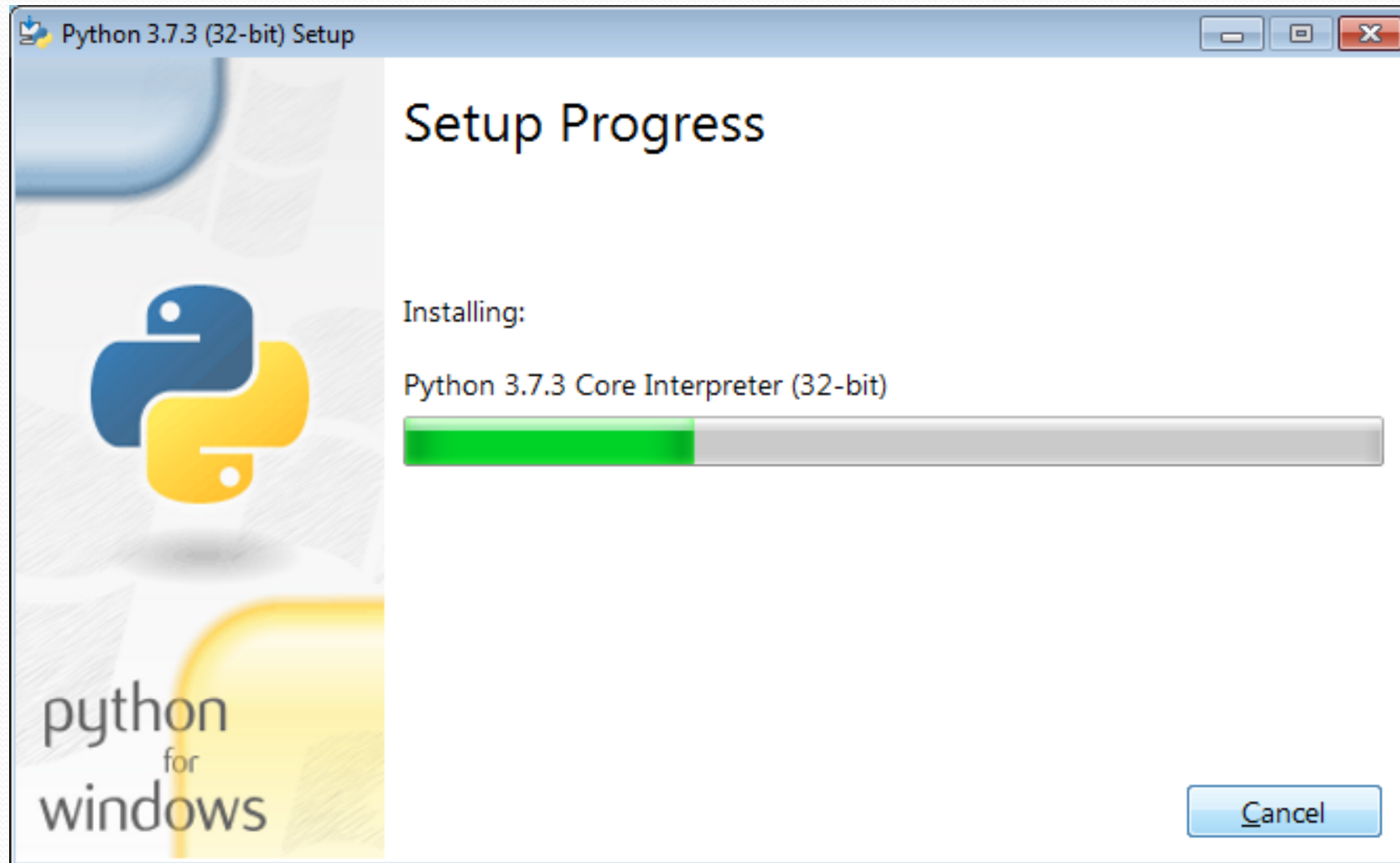
- Bước 2: Sau khi quá trình tải xuống đã hoàn tất, chạy tập tin exe để cài đặt Python. Giờ hãy click vào Install Now.

\* Note: Chọn Add Python to PATH



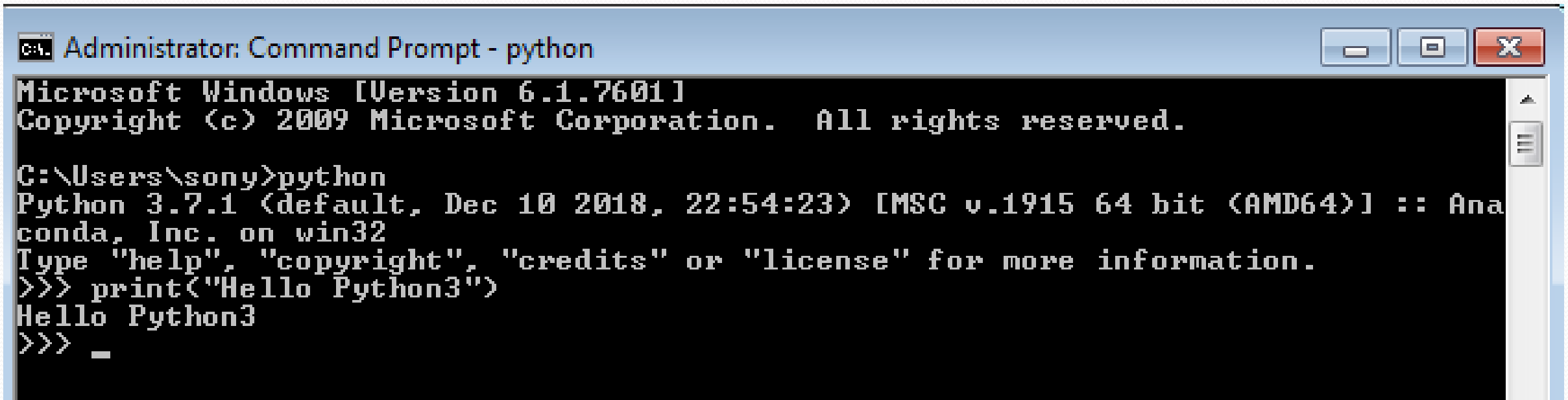


# Cài đặt Python 3



# Cài đặt Python 3

- Run và kiểm tra quá trình cài đặt thành công
  1. Mở Command Prompt (cmd)
  2. Type : python --version hoặc python



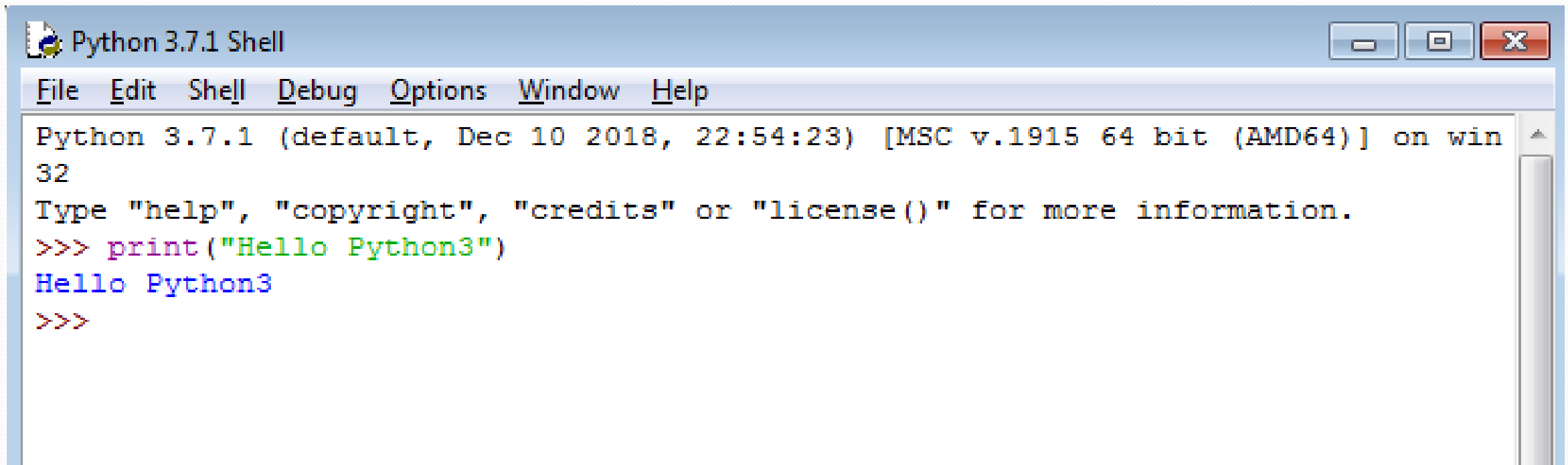
```
Administrator: Command Prompt - python
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\sony>python
Python 3.7.1 (default, Dec 10 2018, 22:54:23) [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello Python3")
Hello Python3
>>> _
```

# Cài đặt Python 3

**Hoặc** dùng Python Shell

- Trên “Start menu” search “idle” and run file “idle.exe”

The image is a screenshot of a Windows application window titled "Python 3.7.1 Shell". The window has a standard Windows interface with a title bar, a menu bar (File, Edit, Shell, Debug, Options, Window, Help), and a main text area. The text area shows the following content: "Python 3.7.1 (default, Dec 10 2018, 22:54:23) [MSC v.1915 64 bit (AMD64)] on win32", "Type 'help', 'copyright', 'credits' or 'license()' for more information.", ">>> print('Hello Python3')", and "Hello Python3". The prompt ">>>" is shown at the end of the line, indicating the shell is ready for the next command. The text is color-coded: "Python" is blue, "3.7.1" is red, "default" is green, "Dec 10 2018, 22:54:23" is blue, "[MSC v.1915 64 bit (AMD64)]" is red, "on win32" is blue, "Type" is blue, "help", "copyright", "credits", and "license()" are in quotes and blue, "for more information." is blue, ">>>" is red, "print" is blue, "Hello Python3" is blue, and "Hello Python3" is blue.

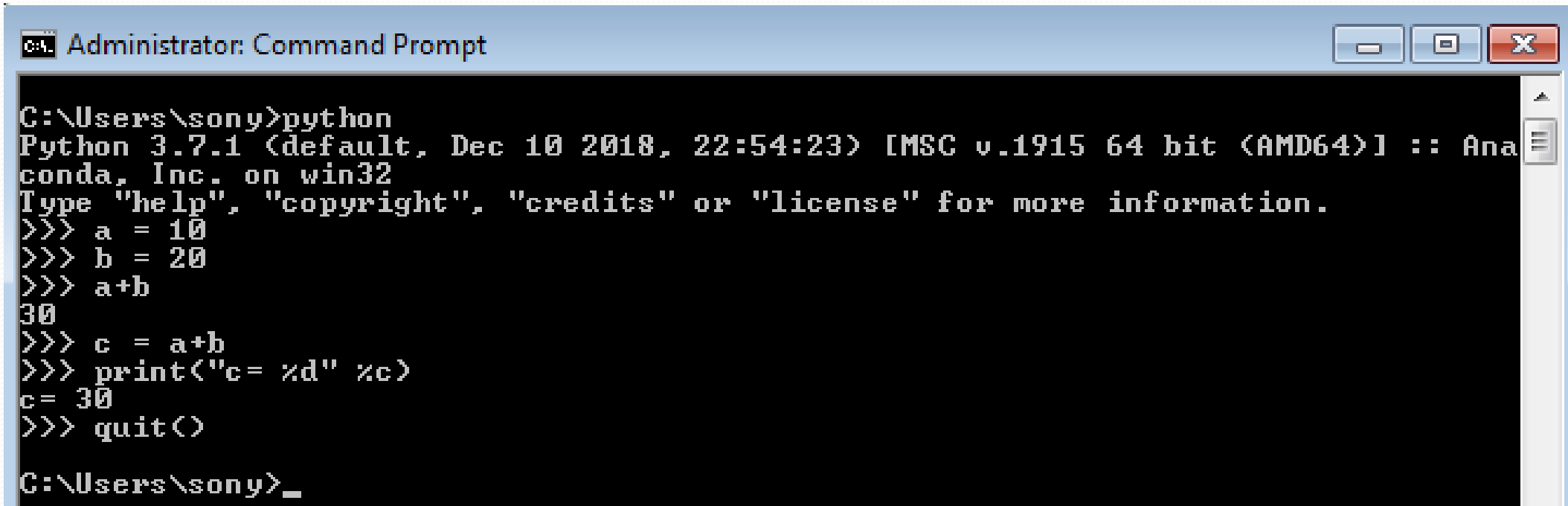
# Cài đặt Python 3

- Thực thi: Python có 2 chế độ thực thi
  - Chế độ dòng lệnh: chạy từng lệnh một
  - Chế độ thực thi: chỉ ra chương trình cần thực hiện (file script \*.py)
    - Trình dịch python sẽ nạp, dịch và chạy chương trình đó

# Cài đặt Python 3

Chế độ dòng lệnh: open cmd -> type python / hoặc open Python Shell

- Lúc này trình thông dịch python sẽ chờ người dùng gõ từng dòng lệnh
- Gõ dòng lệnh nào xong, python chạy liền dòng đó
- Chấm dứt chế độ này bằng cách gõ lệnh: “quit()” hoặc Ctrl+Z



```
Administrator: Command Prompt

C:\Users\sony>python
Python 3.7.1 (default, Dec 10 2018, 22:54:23) [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> a = 10
>>> b = 20
>>> a+b
30
>>> c = a+b
>>> print("c= %d" %c)
c= 30
>>> quit()

C:\Users\sony>_
```



# Cài đặt Python 3

Chế độ thực thi trên file script \*.py

- Ví dụ: Dùng Notepad (hoặc các text editor) tạo file test.py (lưu file dưới dạng mở rộng \*.py)
  - Open Command Prompt (cmd)
  - type: python <đường dẫn tới file test.py>
  - Ví dụ: python C:\Users\sony\Desktop\python\_co\_ban\test\test.py

test.py

```
# -*- coding: utf-8 -*-  
print("Hello world")  
a = 1  
b = 2  
c = a+b  
print("c=%d" %c)
```

```
C:\Users\sony>python C:\Users\sony\Desktop\python_co_ban\test\test.py  
Hello world  
c=3
```

Hoặc di chuyển đến thư mục chứa file cần thực thi

Type: cd <đường dẫn tới thư mục chứa file >

Ví dụ: cd C:\Users\sony\Desktop\python\_co\_ban\test  
python test.py

```
C:\Users\sony>cd C:\Users\sony\Desktop\python_co_ban\test  
C:\Users\sony\Desktop\python_co_ban\test>python test.py  
Hello world  
c=3
```

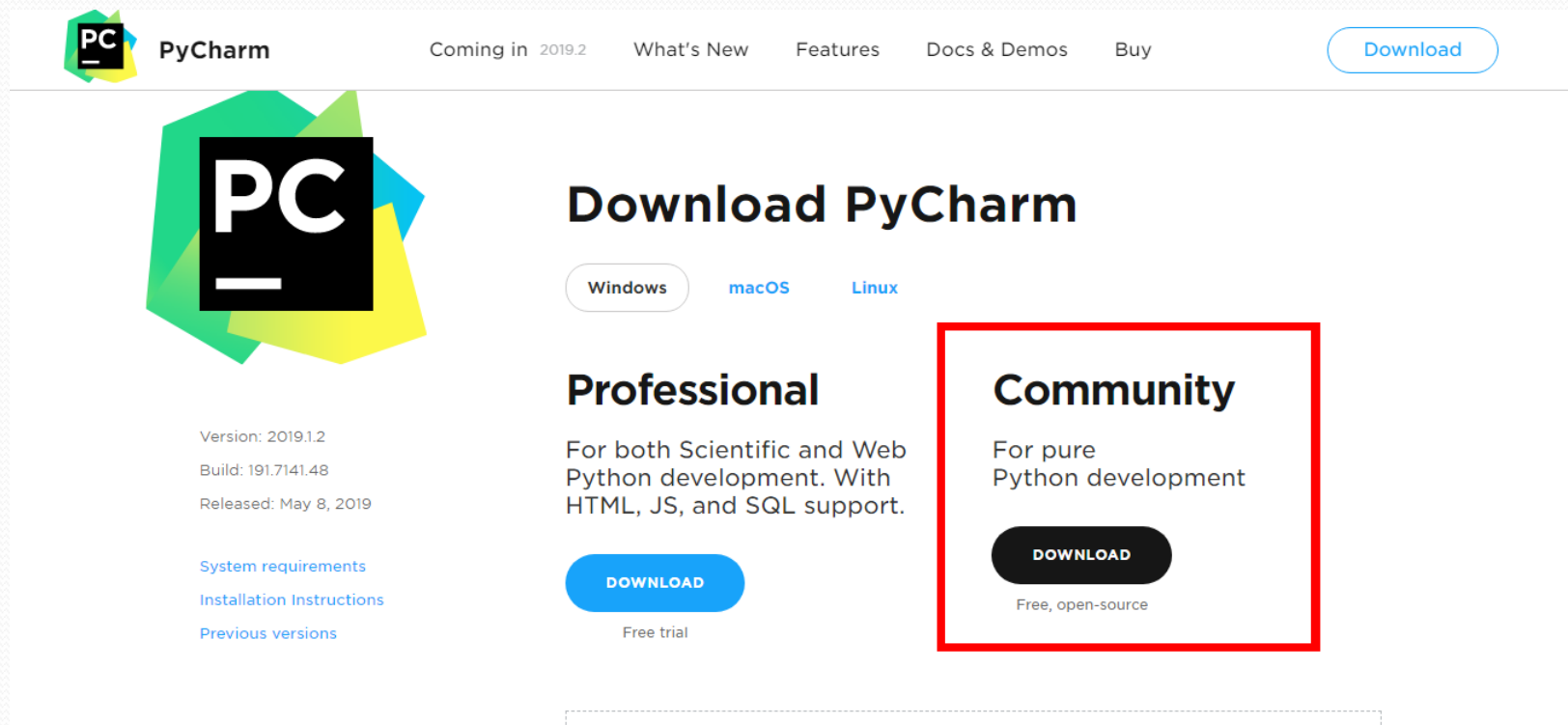
# Cài đặt Python 3

- Hạn chế khi dùng các text editors (notepad, notepad++, wordpad, gedit, vi, ...)
  - Không có tính năng tự động nhắc và sửa lỗi cú pháp (thừa thiếu space/tab , viết sai tên hàm, sai tên thư viện, ...)
  - Không có tính năng debug
  - ...
- Sử dụng môi trường phát triển tích hợp (Integrated Development Environment-**IDE**)
  - **PyCharm**
  - Spyder
  - PyDev
  - Eclipse
  - ...

# Cài đặt Python 3

Cách cài đặt Pycharm IDE sau khi đã cài đặt python

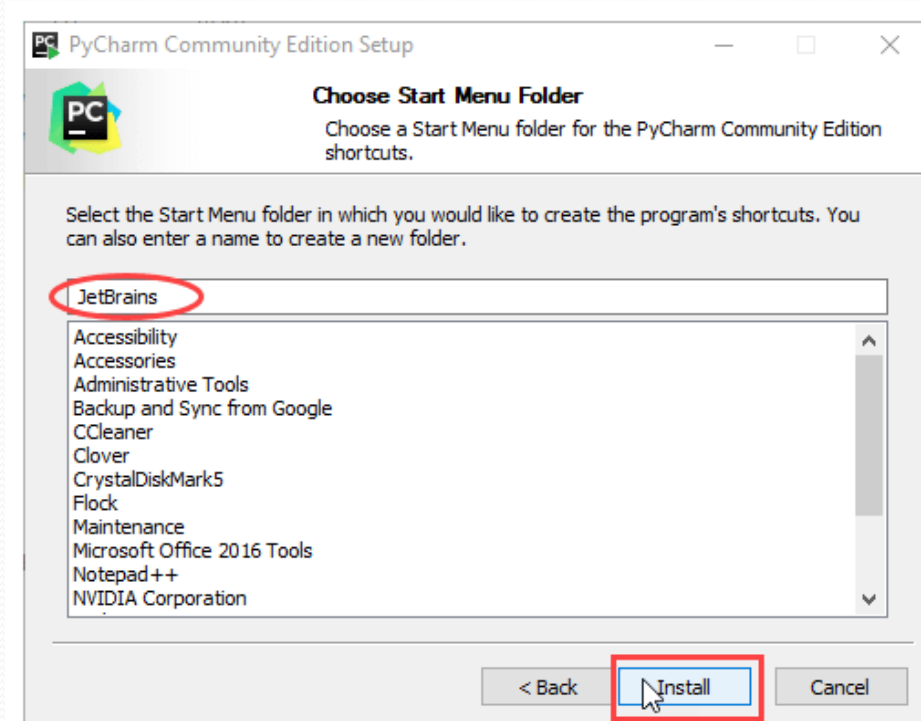
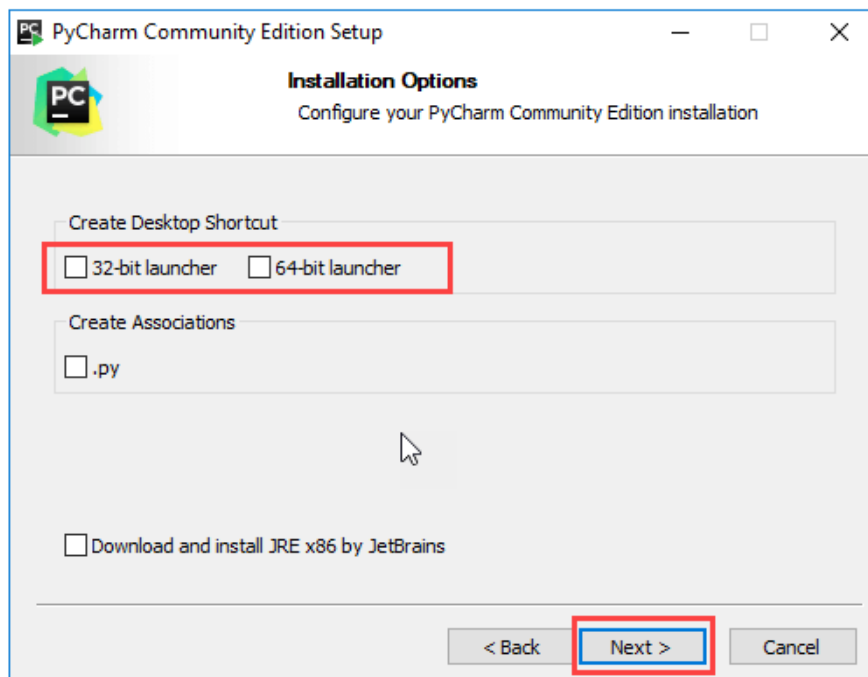
- Bước 1: Để tải PyCharm, hãy truy cập vào trang web: <https://www.jetbrains.com/pycharm/download/> sau đó click vào đường dẫn “Download” nằm dưới mục Community:



# Cài đặt Python 3

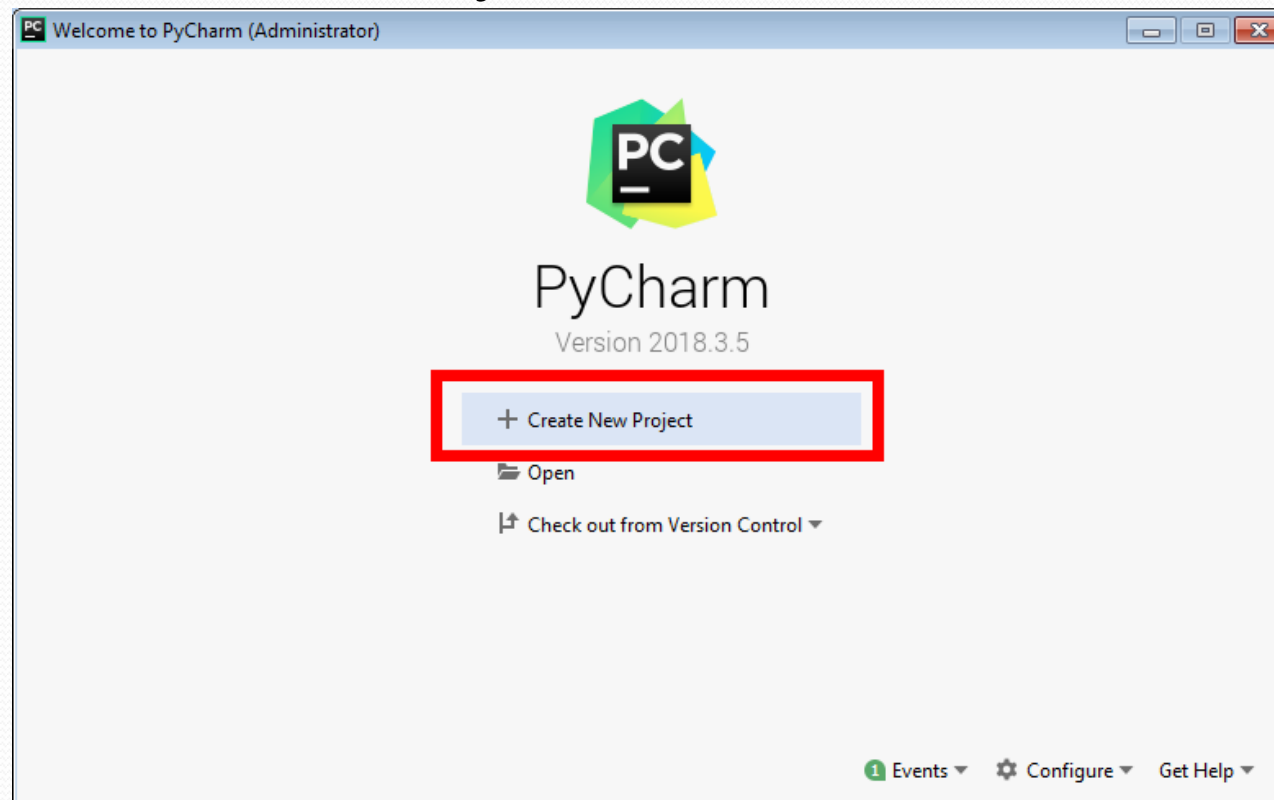
Cách cài đặt Pycharm IDE sau khi đã cài đặt python

- Bước 2: Khi quá trình download đã hoàn tất, chạy tập tin exe để cài đặt PyCharm.



# Cài đặt Python 3

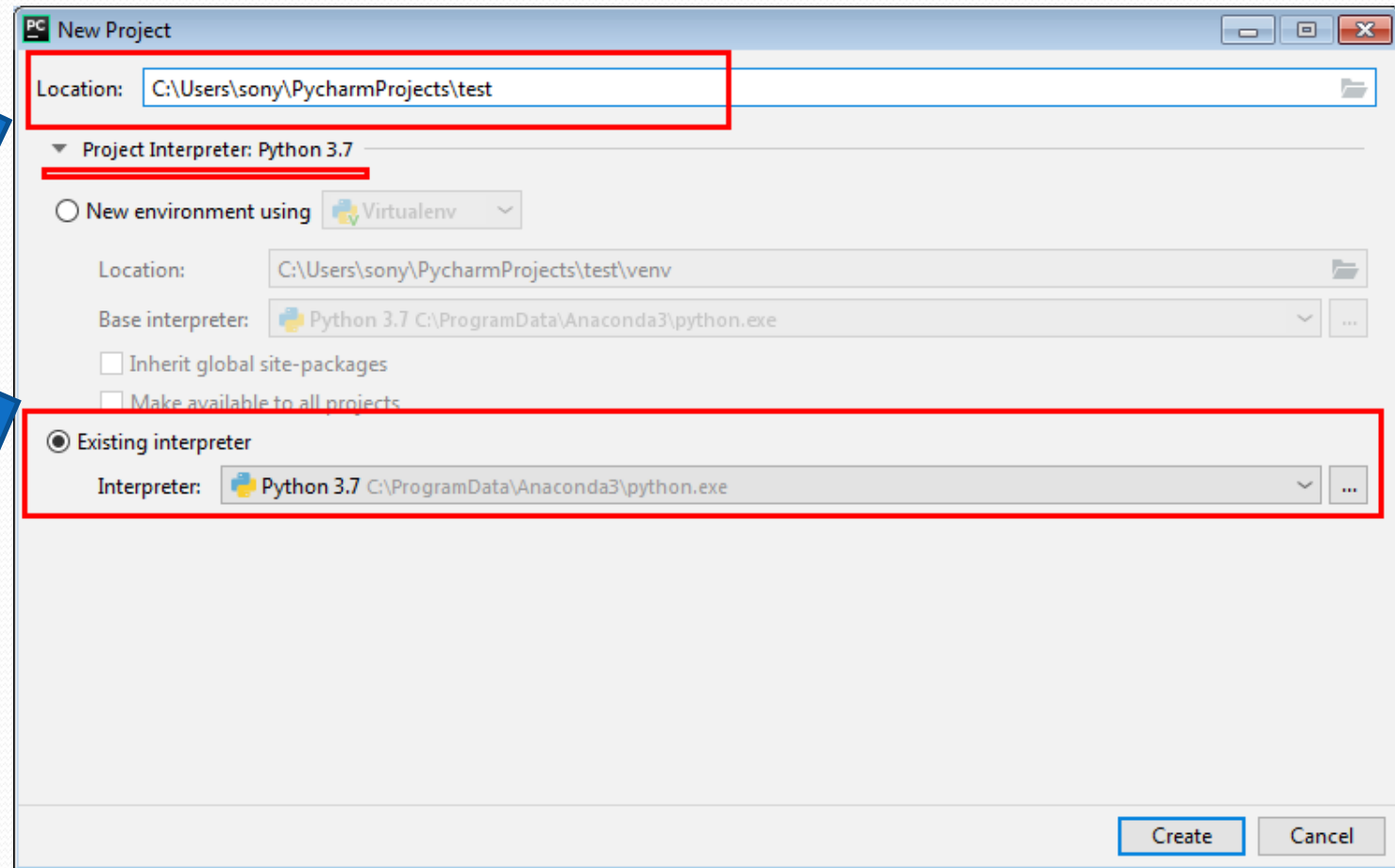
- Tạo một chương trình minh họa dùng pycharm
- B1 : Mở trình soạn thảo PyCharm. Bạn có thể thấy màn hình giới thiệu khi mở PyCharm. Để tạo một dự án mới, bấm vào “Create New Project”.



# Cài đặt Python 3

- B2: Chọn thư mục và tên của project để lưu (Location)

- PyCharm sẽ tìm thấy trình thông dịch Python bạn đã cài đặt trước đó.
- (đường dẫn đến file python.exe – khi đã cài đặt python)
  - Sau đó chọn “Create”

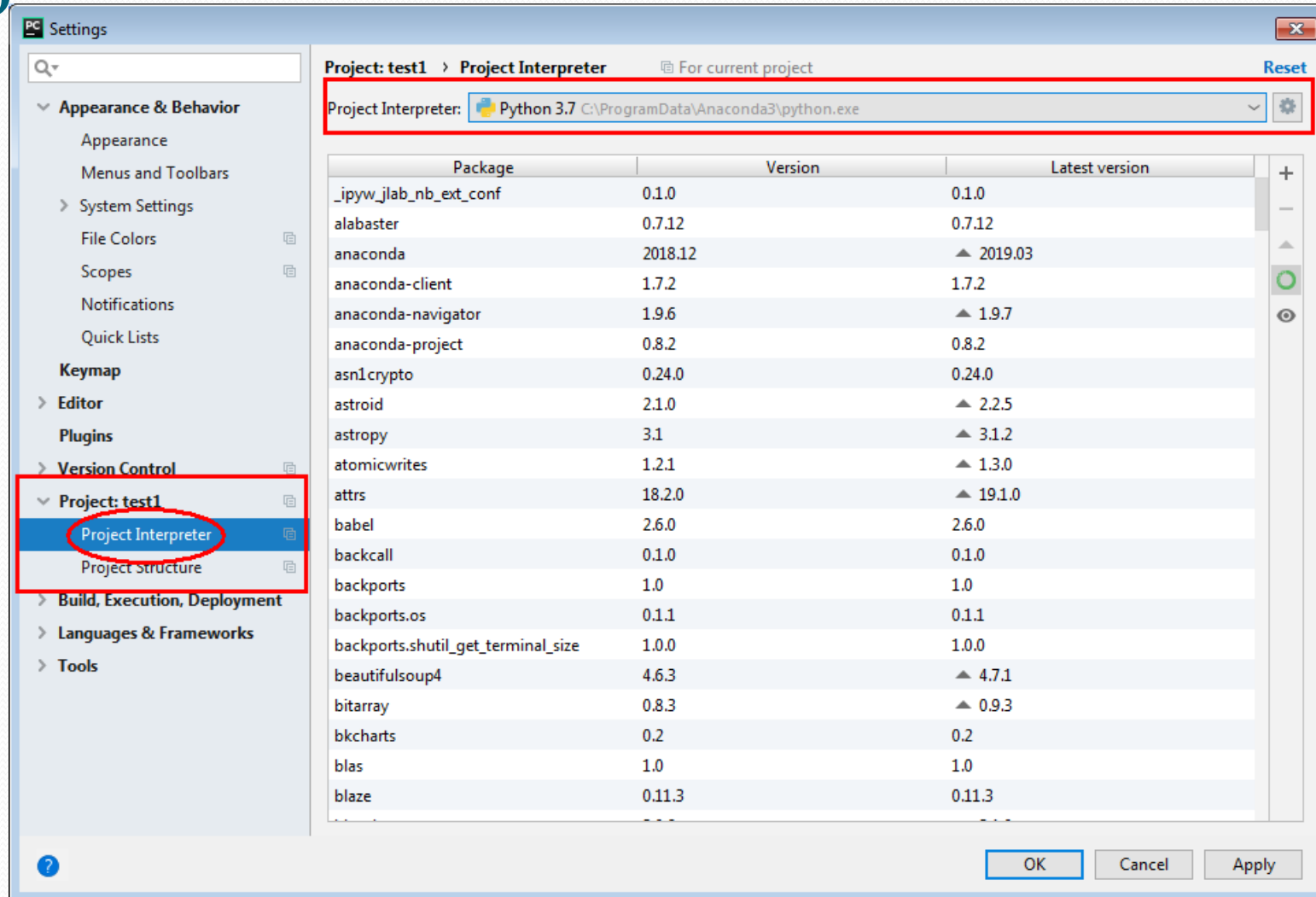
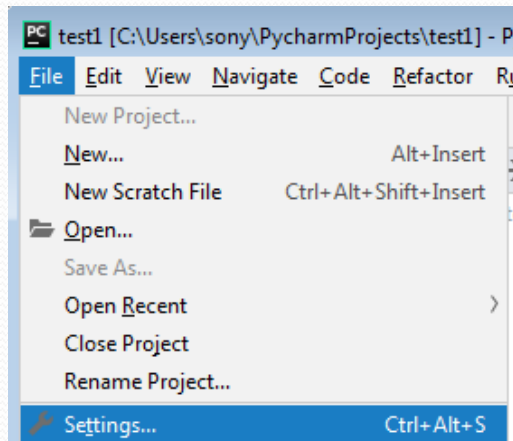


# Cài đặt Python 3

(đường dẫn đến file python.exe – trong thư mục cài đặt python)

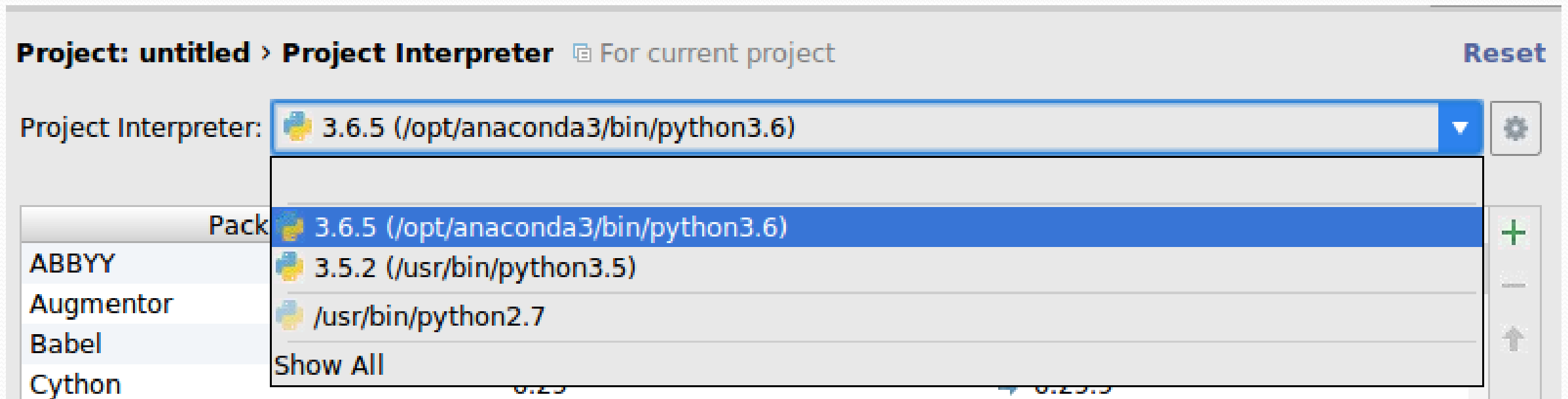
- B3: Load package và update trình thông dịch python cho Pycharm.

• Chọn File ->



# Cài đặt Python 3

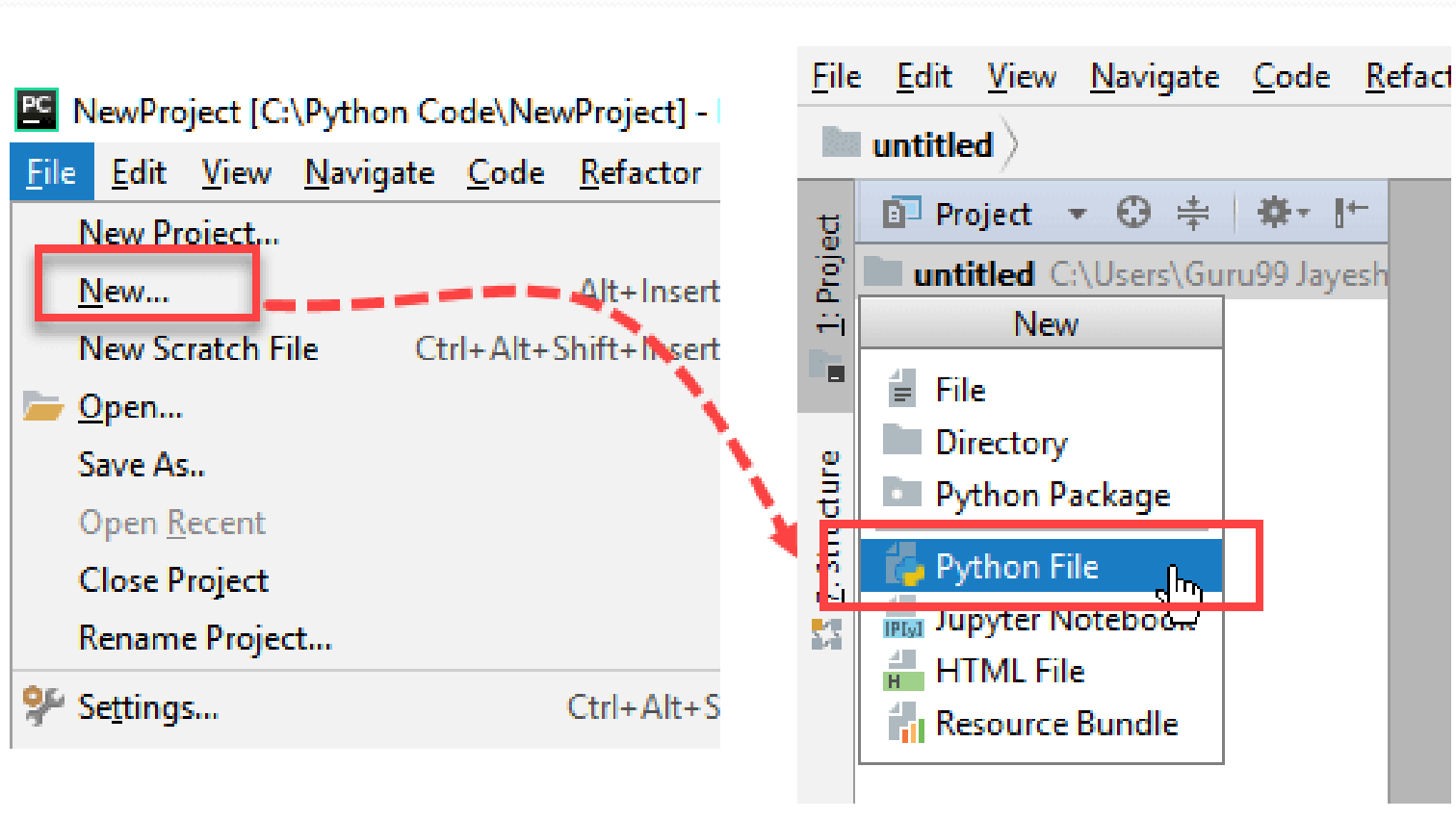
- Ví dụ: có thể cài đặt cùng lúc nhiều python version khác nhau và tùy vào project chọn trình thông dịch và môi trường thích hợp





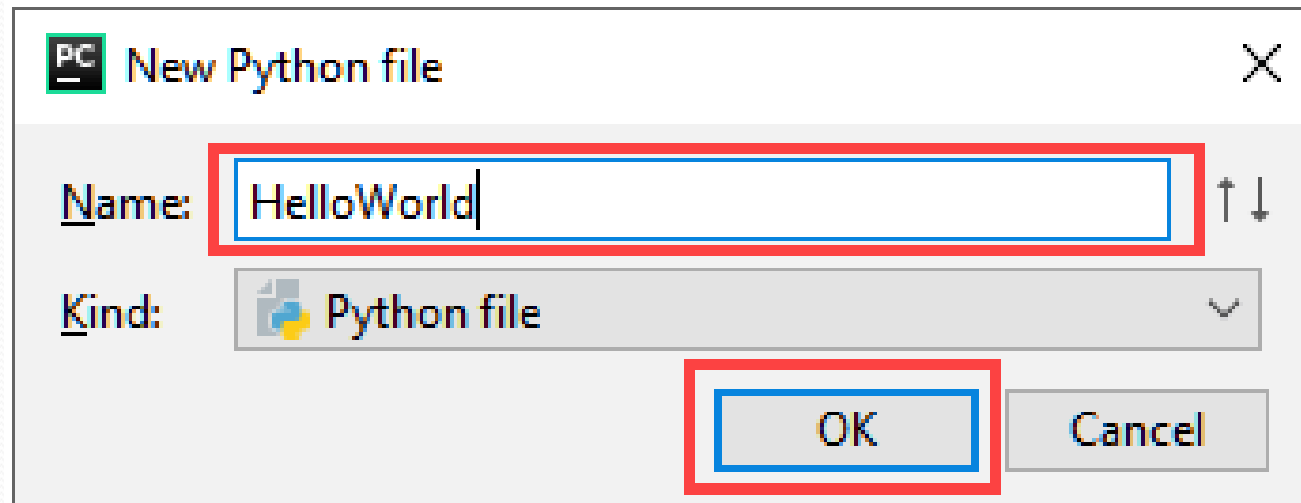
# Cài đặt Python 3

- B4: Vào menu “File” và chọn “New”. Tiếp theo, chọn “Python File”.



# Cài đặt Python 3

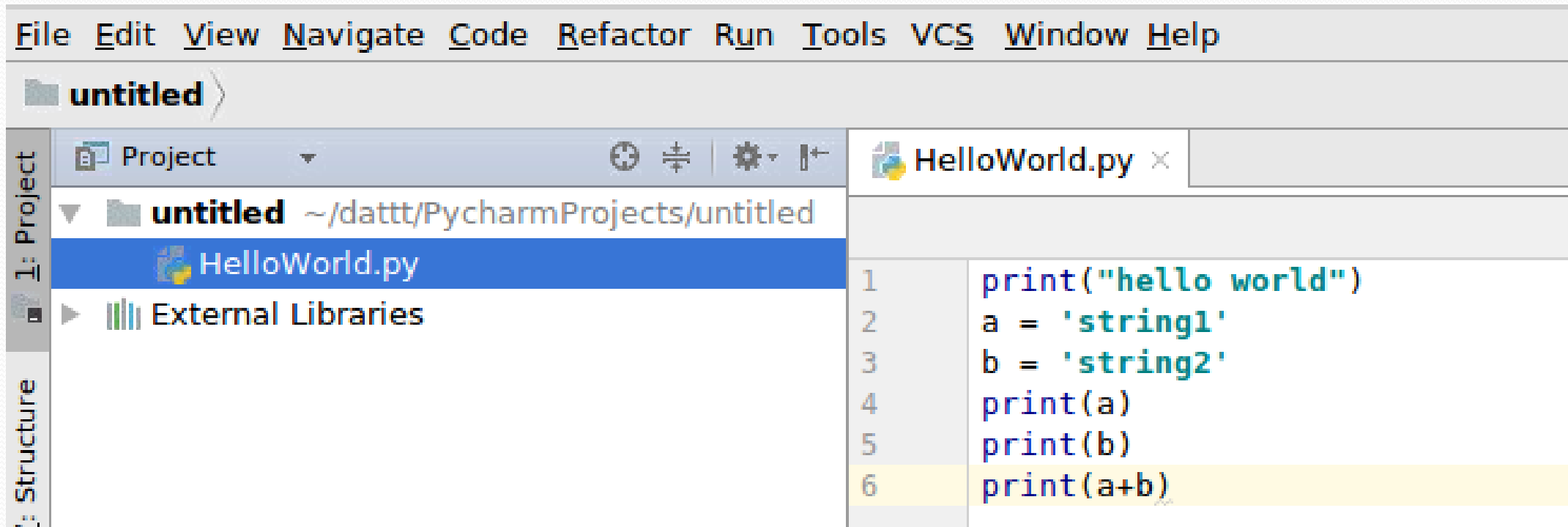
- B5: Một cửa sổ mới sẽ xuất hiện. Giờ hãy nhập tên bạn muốn đặt cho tệp



# Cài đặt Python 3

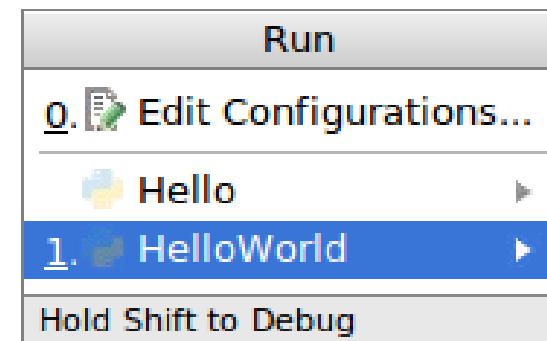
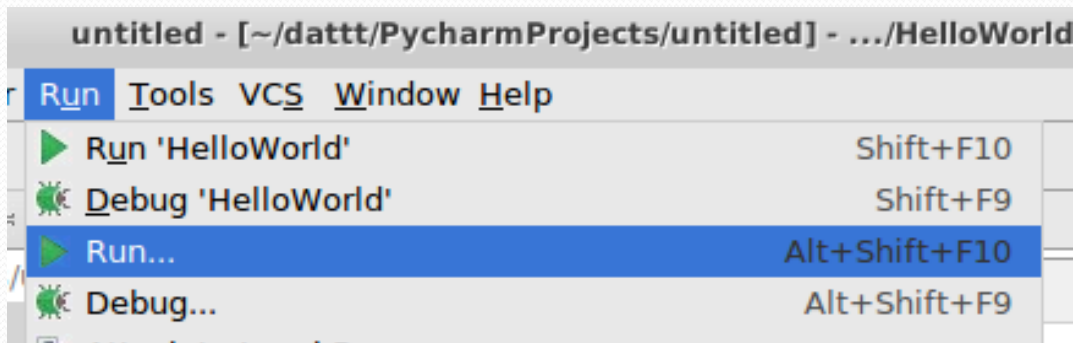
- B6: Viết một chương trình đơn giản

```
print("hello world")  
a = 'string1'  
b = 'string2'  
print(a)  
print(b)  
print(a+b)
```

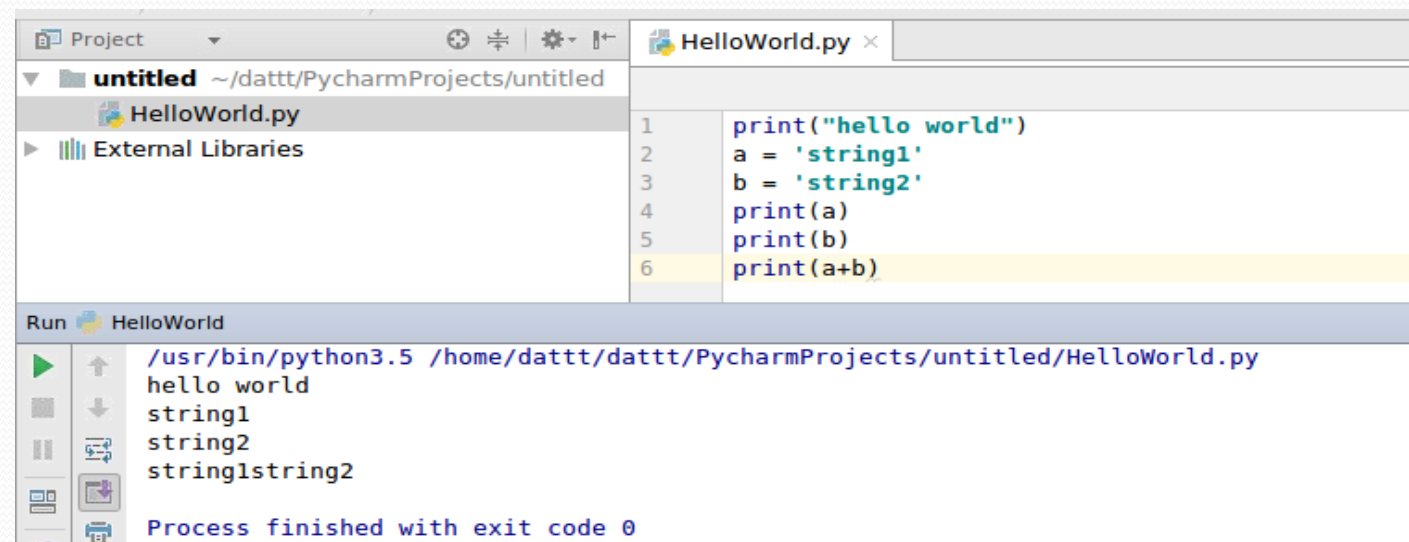


# Cài đặt Python 3

- B7 : Vào menu “Run” và chọn “Run” để chạy chương trình của bạn.



- B8: xem kết quả

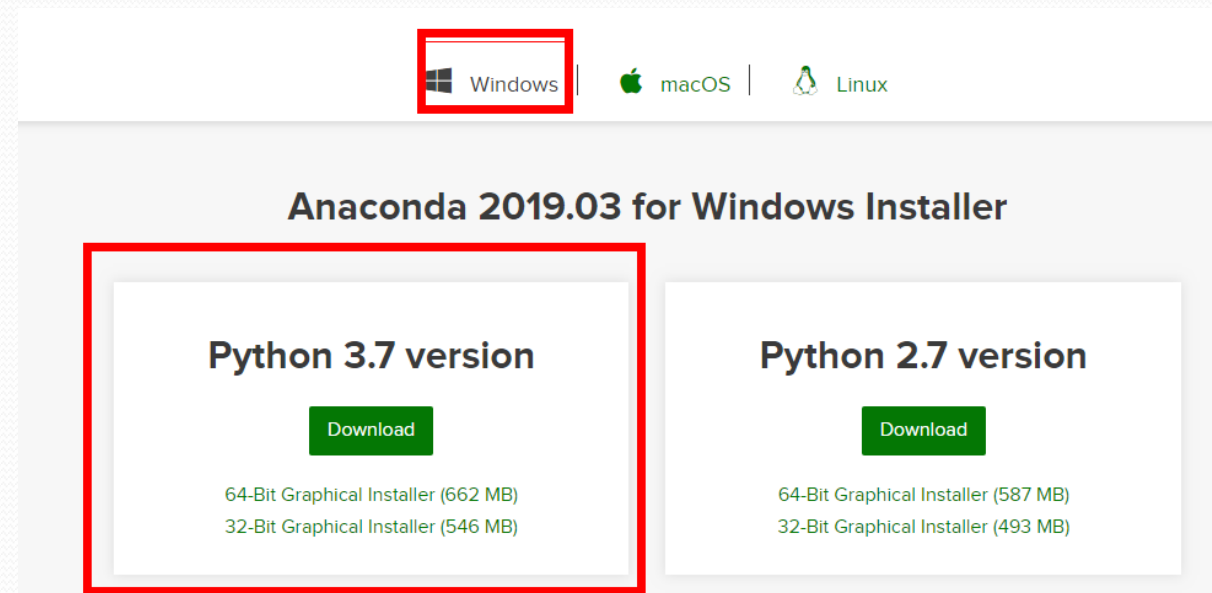


# Cài đặt Python 3 dùng bản phân phối Anaconda

- Anaconda là một bản phân phối mã nguồn mở miễn phí của ngôn ngữ lập trình Python  
(<https://www.anaconda.com/distribution/>)
  - tích hợp sẵn các phiên bản python.
  - phục vụ cho machine learning, deep learning, xử lý dữ liệu quy mô lớn, phân tích dự đoán và tính toán khoa học.
  - được hơn 6 triệu người sử dụng.
  - hơn 1,500+ Python/R data science packages

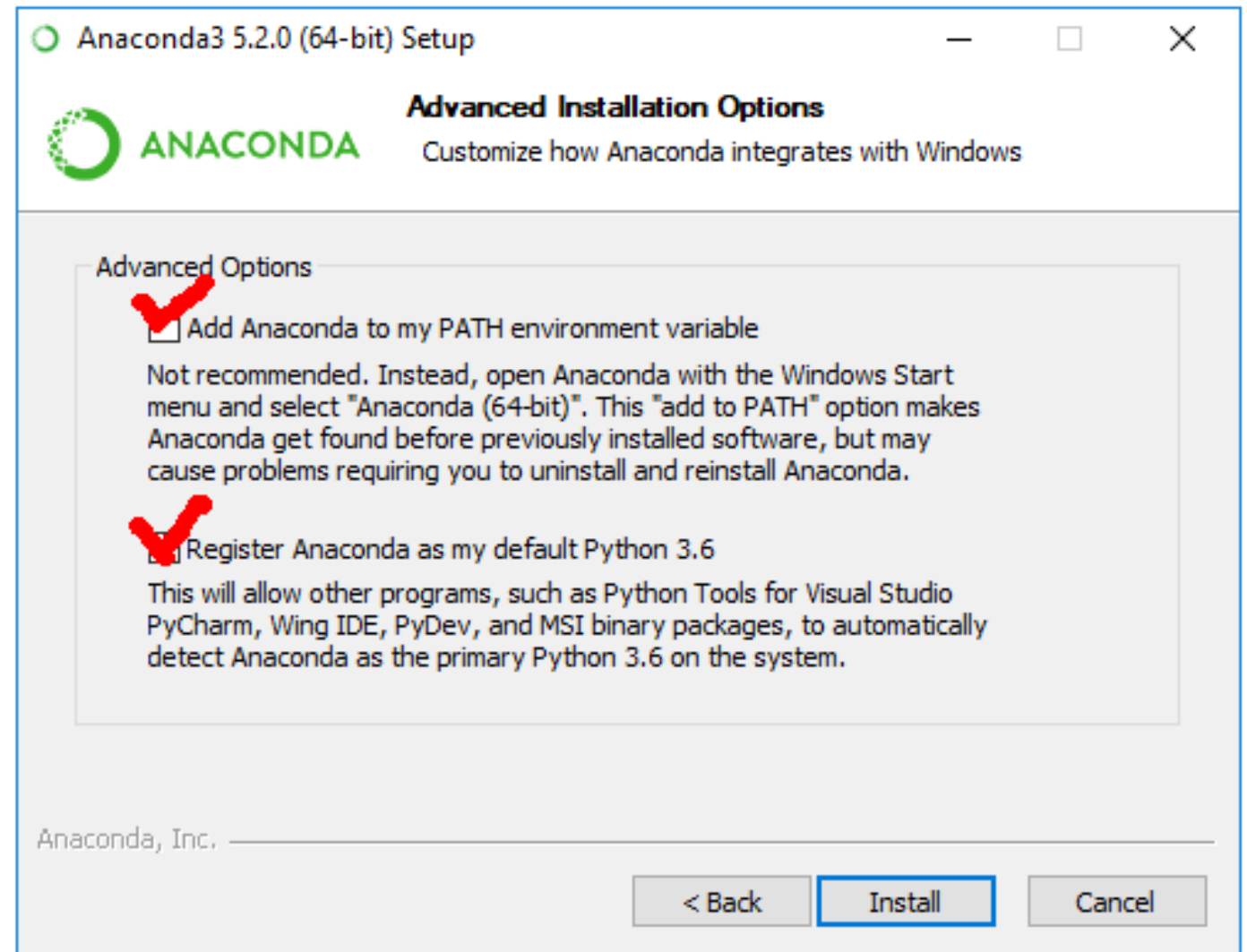
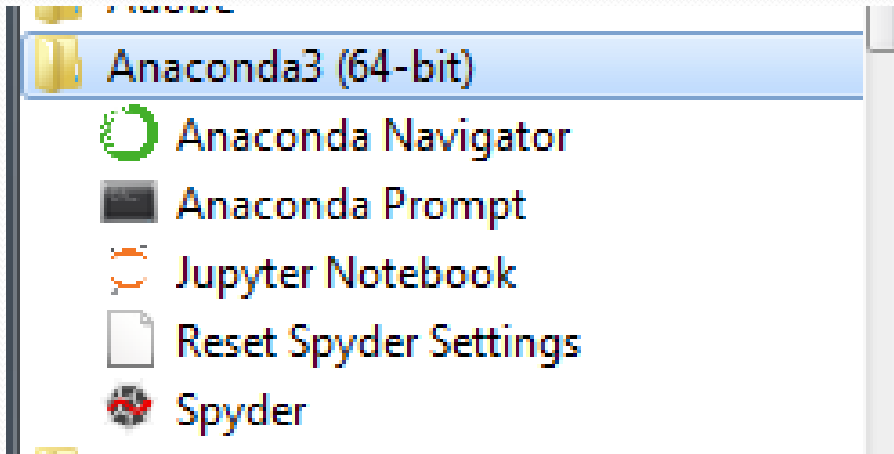
# Cài đặt Python 3 dùng bản phân phối Anaconda

- Cài đặt tương tự như cài đặt python 3
- Download : <https://www.anaconda.com/>
  - <https://www.anaconda.com/distribution/>



# Cài đặt Python 3 dùng bản phân phối Anaconda

- Lúc cài đặt lưu ý:



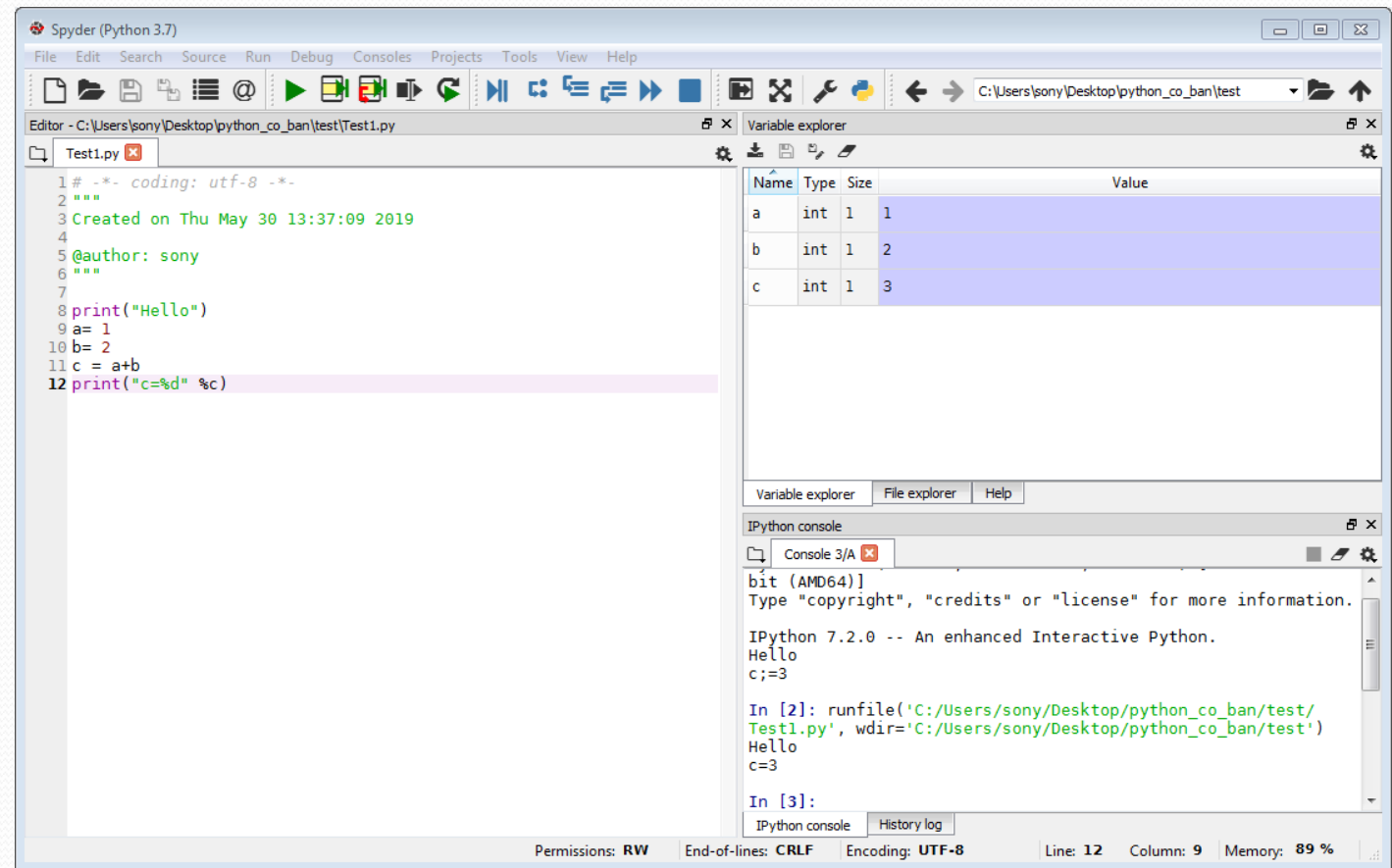
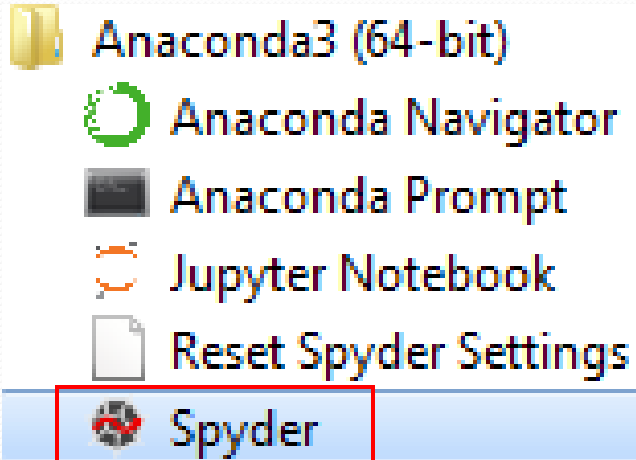
# Cài đặt Python 3 dùng bản phân phối Anaconda

- Kiểm tra cài đặt:
  - Open cmd
  - Type:
    - conda list: nếu cài đặt thành công sẽ thấy danh sách những package đã được cài đặt kèm theo version của chúng
    - python: nếu cài đặt thành công sẽ thấy version của python kèm theo Anaconda
- Cài đặt IDE Pycharm dùng Anaconda tương tự như cài đặt cho python 3 (như các slide trước)



# Cài đặt Python 3 dùng bản phân phối Anaconda

- Ngoài ra, Anaconda tích hợp sẵn một IDE Spyder

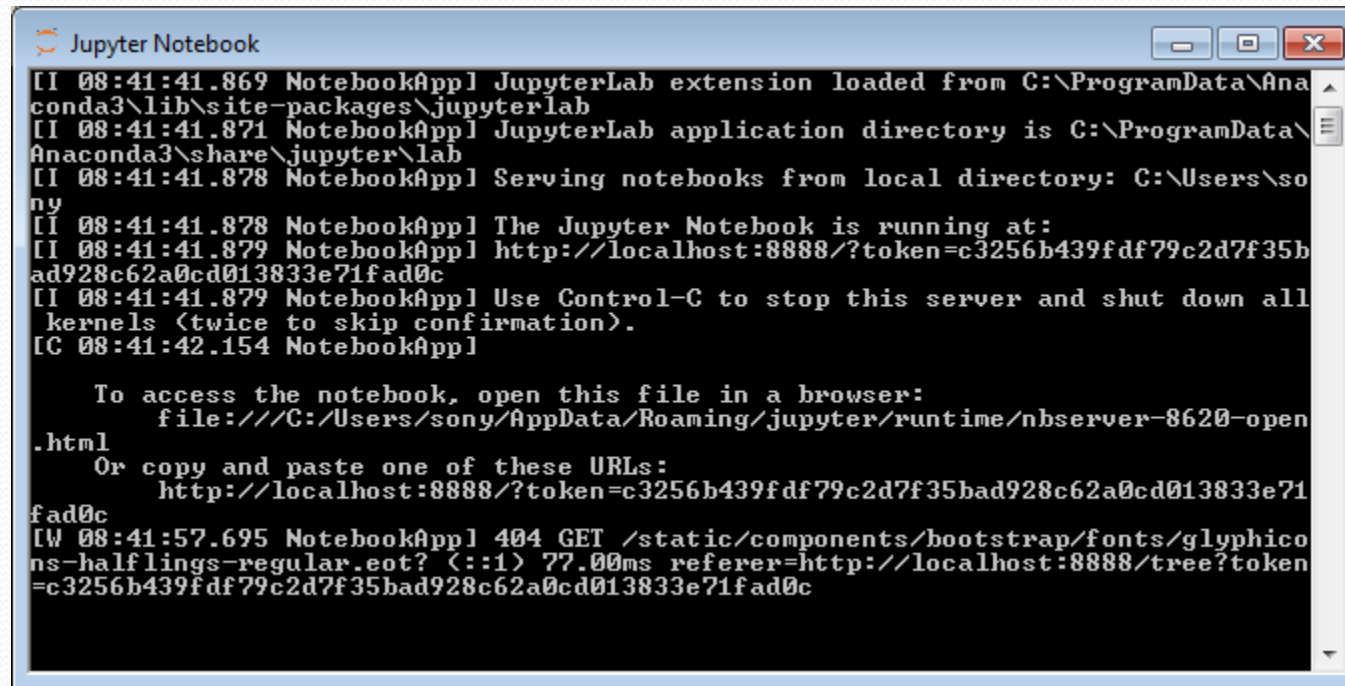


# Jupyter Notebook

- Là một ứng dụng web mã nguồn mở cho phép bạn trình bày như một quyển vở khoa học và cho tương tác trực tiếp với nó
  - code
  - mô phỏng
  - văn bản diễn giải
- Là một công cụ tuyệt vời để:
  - Học về Python
  - Data processing / transformations
  - Vẽ biểu đồ
  - Machine learning
  - Data science

# Jupyter Notebook

- Khởi động: Từ Anaconda Prompt chỉ cần gõ jupyter notebook



```
Jupyter Notebook
[I 08:41:41.869 NotebookApp] JupyterLab extension loaded from C:\ProgramData\Anaconda3\lib\site-packages\jupyterlab
[I 08:41:41.871 NotebookApp] JupyterLab application directory is C:\ProgramData\Anaconda3\share\jupyter\lab
[I 08:41:41.878 NotebookApp] Serving notebooks from local directory: C:\Users\sony
[I 08:41:41.878 NotebookApp] The Jupyter Notebook is running at:
[I 08:41:41.879 NotebookApp] http://localhost:8888/?token=c3256b439fdf79c2d7f35bad928c62a0cd013833e71fad0c
[I 08:41:41.879 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 08:41:42.154 NotebookApp]

To access the notebook, open this file in a browser:
    file:///C:/Users/sony/AppData/Roaming/jupyter/runtime/nbserver-8620-open.html
Or copy and paste one of these URLs:
    http://localhost:8888/?token=c3256b439fdf79c2d7f35bad928c62a0cd013833e71fad0c
[W 08:41:57.695 NotebookApp] 404 GET /static/components/bootstrap/fonts/glyphicons-halflings-regular.eot? (<:1) 77.00ms referer=http://localhost:8888/tree?token=c3256b439fdf79c2d7f35bad928c62a0cd013833e71fad0c
```

# Jupyter Notebook

[Quit](#)[Logout](#)[Files](#)[Running](#)[Clusters](#)


Select items to perform actions on them.

[Upload](#)[New ▾](#)☐ 0 ▾  /[Name ▾](#)[Last Modified](#)[File size](#)☐  [Contacts](#)

9 months ago

☐  [Corel](#)

7 years ago

☐  [Desktop](#)

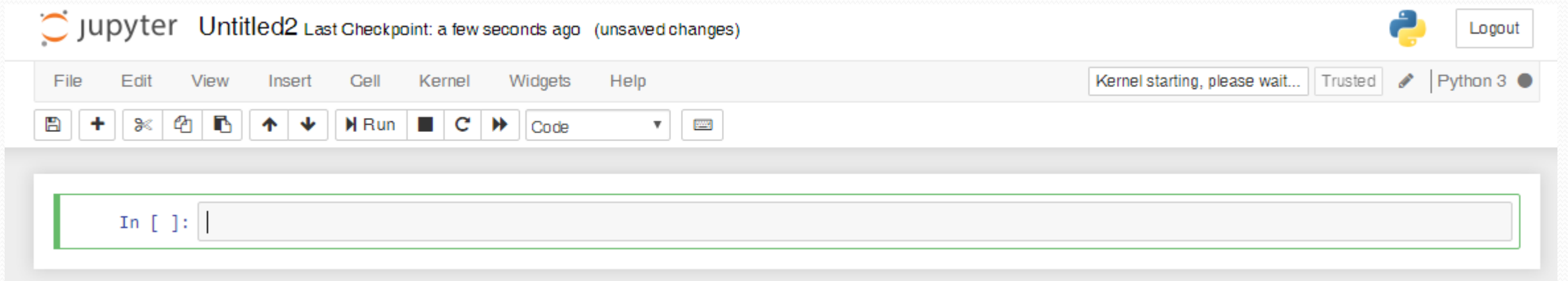
22 minutes ago

# Jupyter Notebook

- Tạo một notebook python mới



# Jupyter Notebook



# Jupyter Notebook

- Một notebook bao gồm nhiều cell (ô). Khi tạo mới một notebook, một cell rỗng đầu tiên được tạo sẵn.

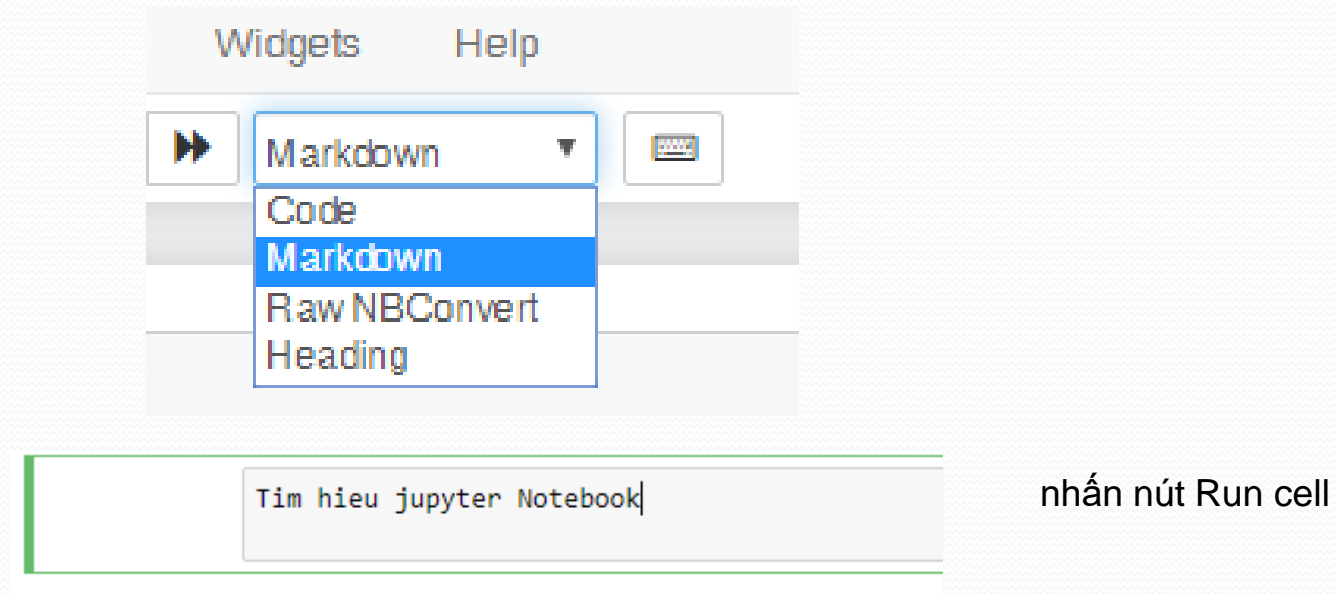
```
In [ ]: |
```

- Cell trên có kiểu là “Code”, điều đó có nghĩa là chúng ta có thể gõ code Python và thực thi ngay lập tức. Để thực thi code, chúng ta có thể nhấn nút Run cell hoặc nhấn Ctrl + Enter

```
In [1]: print("hello world")  
hello world
```

# Jupyter Notebook

- Người dùng có thể chuyển loại cell từ Code thành Markdown để viết những đoạn văn bản giải thích code



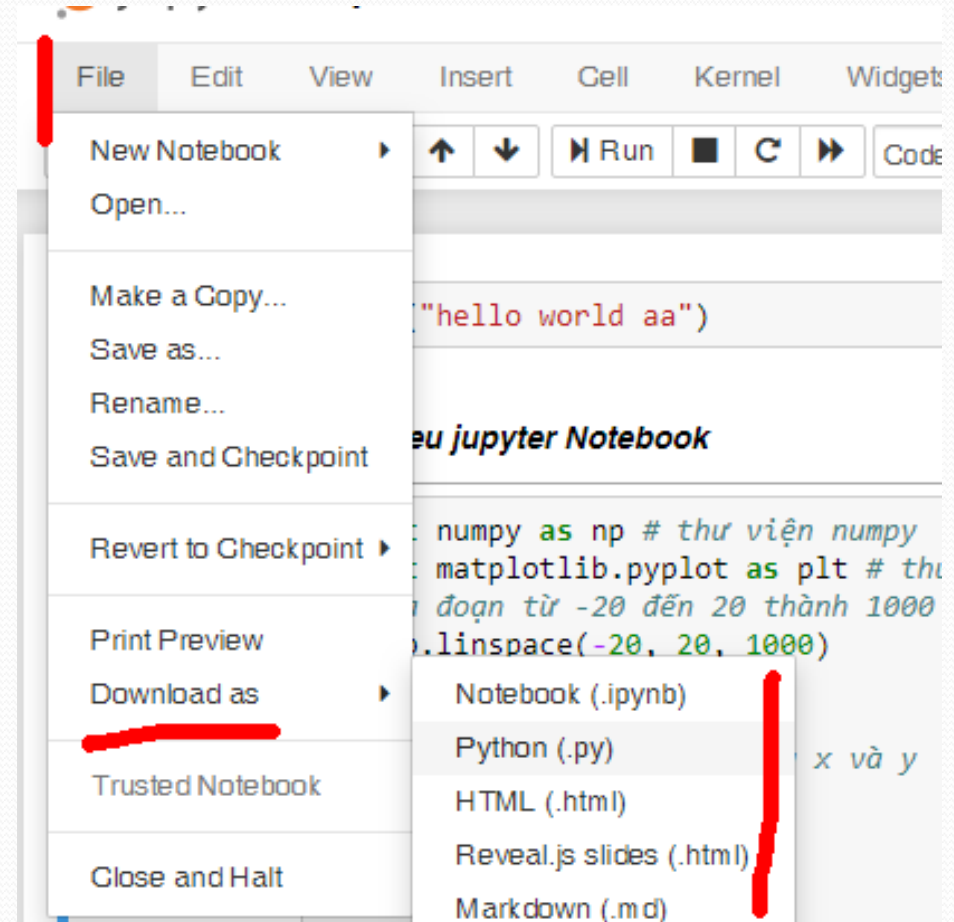


# Jupyter Notebook

- Checkpoints: Bằng cách tạo các Checkpoints lưu trạng thái hiện tại của notebook, Jupyter Notebook cho phép người dùng có thể quay lại thời điểm tạo Checkpoints để kiểm tra hoặc hoàn tác trước đó.
- Để tạo Checkpoint, chọn *File -> Save and Checkpoint*. Nếu chúng ta muốn xem lại các Checkpoints trước đó thì chọn *File -> Revert to Checkpoint*.

# Jupyter Notebook

- Chức năng Export notebook



# Code Python trên trình duyệt

- <http://pythonfiddle.com/>

The screenshot shows the Python Fiddle web application interface. At the top, there is a navigation bar with buttons for 'Run', 'Reset', 'Share', 'Import', and 'Dashboard', along with a 'Language' dropdown menu. The title 'Python Fiddle' and the 'Python Cloud IDE' logo are on the right. The main area is divided into three sections: a left sidebar with a 'Featured' list of links, a central code editor with a Python script, and a right sidebar for tags. The output area at the bottom displays the result of the code execution.

test1

Python Fiddle Python Cloud IDE

Featured

- [Euler Problem Fifth teen Lattice Path](#)
- [DFA Simple Implementation](#)
- [Cartesian to Polar](#)
- [Parse GPX file](#)
- [Knockout sequence sorting](#)
- [Battleships Game in Python](#)
- [Temperature convert program](#)
- [Greatest Common Divisor](#)
- [Pig latin Translator for Python](#)

```
1 print("Hello")
2 a =4
3 b=25
4 c= a+b
5 print("c== %d" %c)
```

Tags:

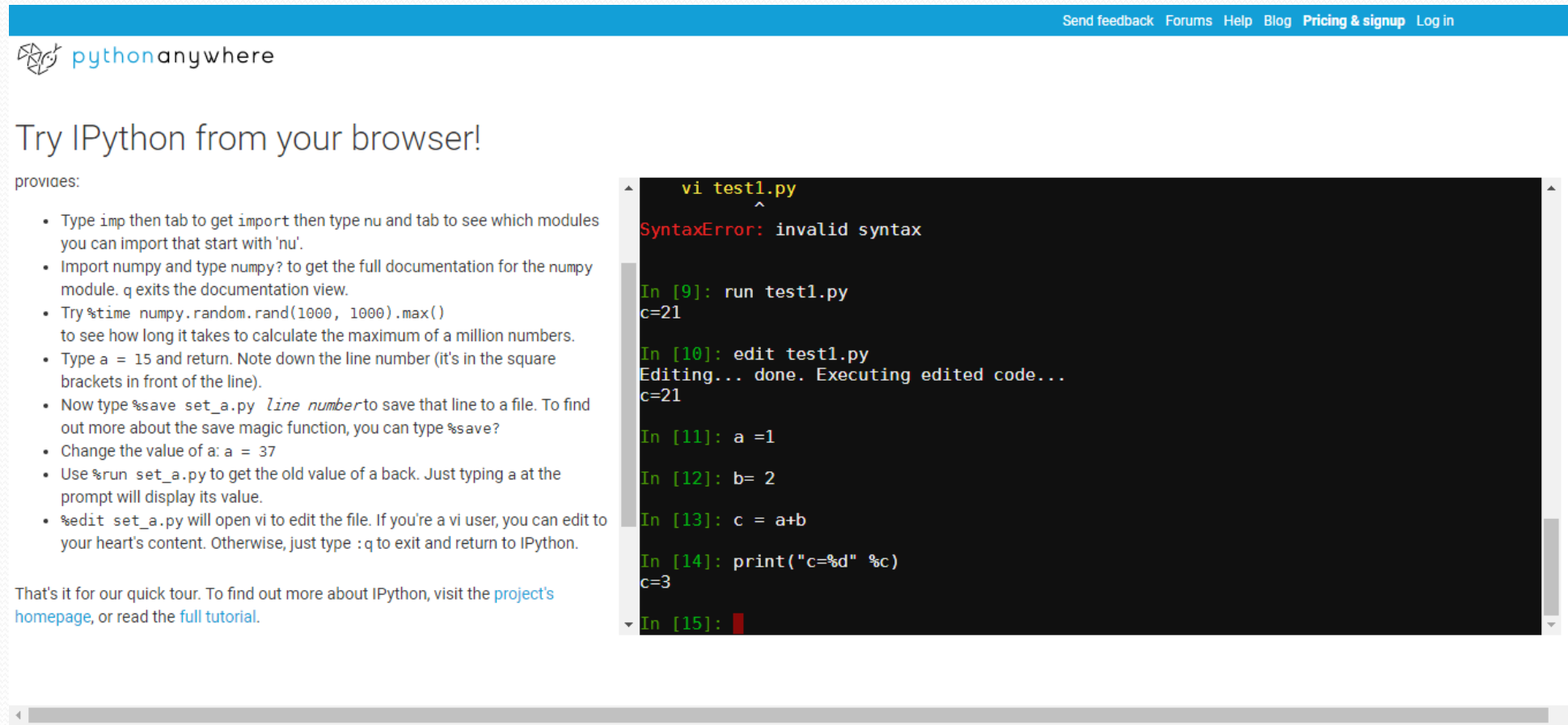
A comma-separated list of tags.

Save

Hello  
c== 29

# Code Python trên trình duyệt

- <https://www.pythonanywhere.com/try-ipython/>



pythonanywhere

Send feedback Forums Help Blog Pricing & signup Log in

## Try IPython from your browser!

provides:

- Type `imp` then `tab` to get `import` then type `nu` and `tab` to see which modules you can import that start with 'nu'.
- Import `numpy` and type `numpy?` to get the full documentation for the `numpy` module. `q` exits the documentation view.
- Try `%time numpy.random.rand(1000, 1000).max()` to see how long it takes to calculate the maximum of a million numbers.
- Type `a = 15` and `return`. Note down the line number (it's in the square brackets in front of the line).
- Now type `%save set_a.py line number` to save that line to a file. To find out more about the `save` magic function, you can type `%save?`
- Change the value of `a`: `a = 37`
- Use `%run set_a.py` to get the old value of `a` back. Just typing `a` at the prompt will display its value.
- `%edit set_a.py` will open `vi` to edit the file. If you're a `vi` user, you can edit to your heart's content. Otherwise, just type `:q` to exit and return to IPython.

That's it for our quick tour. To find out more about IPython, visit the [project's homepage](#), or read the [full tutorial](#).

```
vi test1.py
^
SyntaxError: invalid syntax

In [9]: run test1.py
c=21

In [10]: edit test1.py
Editing... done. Executing edited code...
c=21

In [11]: a = 1

In [12]: b= 2

In [13]: c = a+b

In [14]: print("c=%d" %c)
c=3

In [15]:
```

# Visualize code execution

- <https://pythontutor.com/>

You can also embed visualizations into any webpage. Here is a Python example:

Python 3.6

```
1 def listSum(numbers):
2     if not numbers:
3         return 0
4     else:
5         (f, rest) = numbers
6         return f + listSum(rest)
7
8 myList = (1, (2, (3, None)))
9 total = listSum(myList)
```

[Edit this code](#)

→ line that just executed  
→ next line to execute

Step 11 of 22

Rendered by [Python Tutor](#)  
[Customize visualization](#)

Frames

Global frame

listSum

myList

listSum

numbers

f

rest

Objects

function

listSum(numbers)

tuple

tuple

tuple

0 1

1

0 1

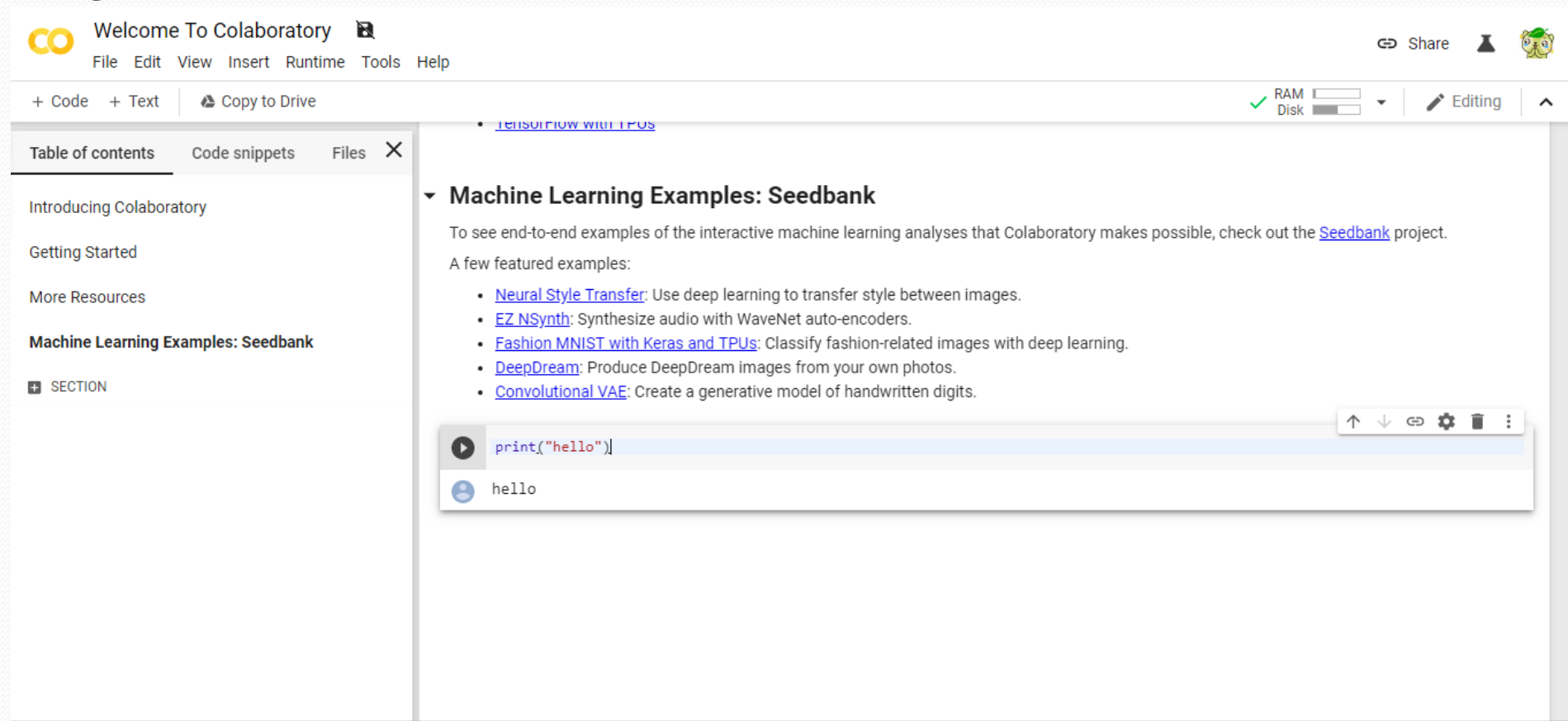
2

0 1

3 None

# Dịch Vụ Hỗ Trợ Deep Learning Và Machine Learning

- Google Colab <https://colab.research.google.com/>



The screenshot displays the Google Colaboratory web interface. At the top, there's a header with the Colab logo, 'Welcome To Colaboratory', and a menu bar (File, Edit, View, Insert, Runtime, Tools, Help). Below the header, there's a toolbar with options like '+ Code', '+ Text', and 'Copy to Drive'. The main content area is divided into a left sidebar and a main panel. The sidebar contains a 'Table of contents' with links to 'Introducing Colaboratory', 'Getting Started', 'More Resources', and 'Machine Learning Examples: Seedbank'. The main panel shows the 'Machine Learning Examples: Seedbank' section, which includes a description of the Seedbank project and a list of featured examples: 'Neural Style Transfer', 'EZ NSynth', 'Fashion MNIST with Keras and TPUs', 'DeepDream', and 'Convolutional VAE'. Below the text, there's a code editor showing a single line of Python code: `print("hello")`. The output of this code is displayed below the editor as 'hello'.

CO Welcome To Colaboratory

File Edit View Insert Runtime Tools Help

+ Code + Text Copy to Drive

Table of contents Code snippets Files X

Introducing Colaboratory

Getting Started

More Resources

Machine Learning Examples: Seedbank

SECTION

TensorFlow with TPUs

Machine Learning Examples: Seedbank

To see end-to-end examples of the interactive machine learning analyses that Colaboratory makes possible, check out the [Seedbank](#) project.

A few featured examples:

- [Neural Style Transfer](#): Use deep learning to transfer style between images.
- [EZ NSynth](#): Synthesize audio with WaveNet auto-encoders.
- [Fashion MNIST with Keras and TPUs](#): Classify fashion-related images with deep learning.
- [DeepDream](#): Produce DeepDream images from your own photos.
- [Convolutional VAE](#): Create a generative model of handwritten digits.

```
print("hello")
```

hello

# Google Colab

- Google Colab is a free tool for training neural network.
- Run on any browser (Chrome, Internet Explorer, Firefox, etc.)
- Train the model with NVIDIA Tesla K80

<https://trituenhantao.io/tutorial/lam-quen-voi-google-colab/>

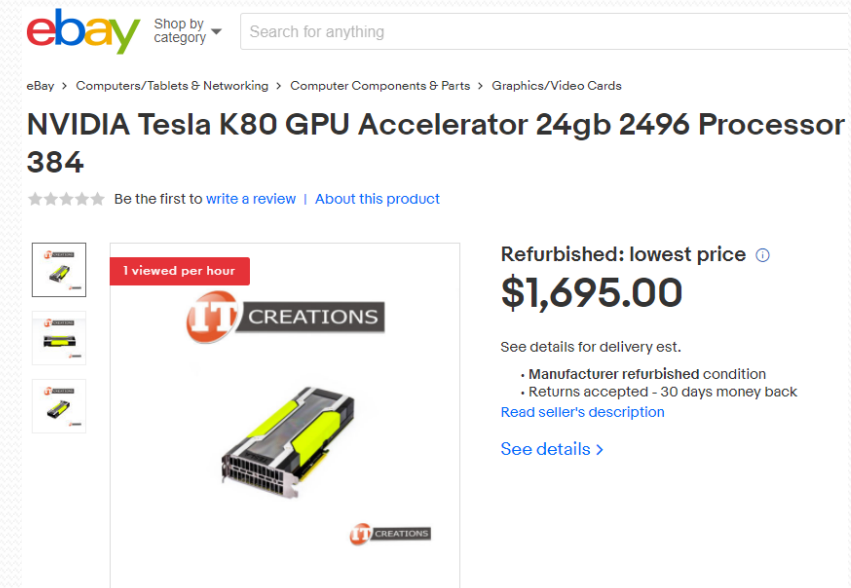
Language program: Python

IDE: Jupyter Notebook

Train on data uploaded to Google Drive (default 15GB)

Maximum time used: 12 hours

Maximum RAM: 13GB



The image shows an eBay product listing for an NVIDIA Tesla K80 GPU Accelerator. The listing includes the eBay logo, a search bar, and the product title "NVIDIA Tesla K80 GPU Accelerator 24gb 2496 Processor 384". Below the title, there are five stars and a link to write a review. The product is shown in a 3D rendering, and the price is listed as \$1,695.00. The listing also includes a "Refurbished: lowest price" badge and a "See details" link.

ebay Shop by category Search for anything

eBay > Computers/Tablets & Networking > Computer Components & Parts > Graphics/Video Cards

**NVIDIA Tesla K80 GPU Accelerator 24gb 2496 Processor 384**

★★★★★ Be the first to [write a review](#) | [About this product](#)

1 viewed per hour

CREATIONS

Refurbished: lowest price ⓘ

**\$1,695.00**

See details for delivery est.

- Manufacturer refurbished condition
- Returns accepted - 30 days money back

[Read seller's description](#)

[See details >](#)

# Kaggle

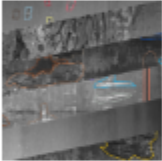







- <https://www.kaggle.com/>



- Tạo tài khoản



# Kaggle Competitions

19 Active Competitions		
	<b>Severstal: Steel Defect Detection</b> Can you detect and classify defects in steel? <i>Featured</i> · Kernels Competition · 3 months to go ·  manufacturing, image data	\$120,000 100 teams
 TWO SIGMA	<b>Two Sigma: Using News to Predict Stock Movements</b> Use news analytics to predict stock price performance <i>Featured</i> · Kernels Competition · 9 days to go ·  news agencies, time series, finance, money	\$100,000 2,927 teams
	<b>APTOS 2019 Blindness Detection</b> Detect diabetic retinopathy to stop blindness before it's too late <i>Featured</i> · Kernels Competition · a month to go ·  healthcare, medicine, image data, multiclass classi...	\$50,000 1,892 teams
	<b>SIIM-ACR Pneumothorax Segmentation</b> Identify Pneumothorax disease in chest x-rays <i>Featured</i> · a month to go ·  image data, object segmentation	\$30,000 1,182 teams

# Kaggle Datasets


## Datasets

[Documentation](#)[New Dataset](#)


[Feedback](#)[Filter](#)


[Public](#)[Your Datasets](#)[Favorites](#)


Sort by: **Hottest** ▼





**NYS Environmental Remediation Sites**

 State of New York

 19 hours


 3 MB

 8.5


 4 Files (other, JSON, CSV)


▲


52





**Google-Landmarks Dataset**

 Google

 15 hours


 107 MB

 8.2


 5 Files (CSV)


▲


330





**New York City Current Job Postings**

 City of New York

 6 days


 3 MB

 8.5


 3 Files (other, JSON, CSV)

▲

103



**Crimes in Boston**

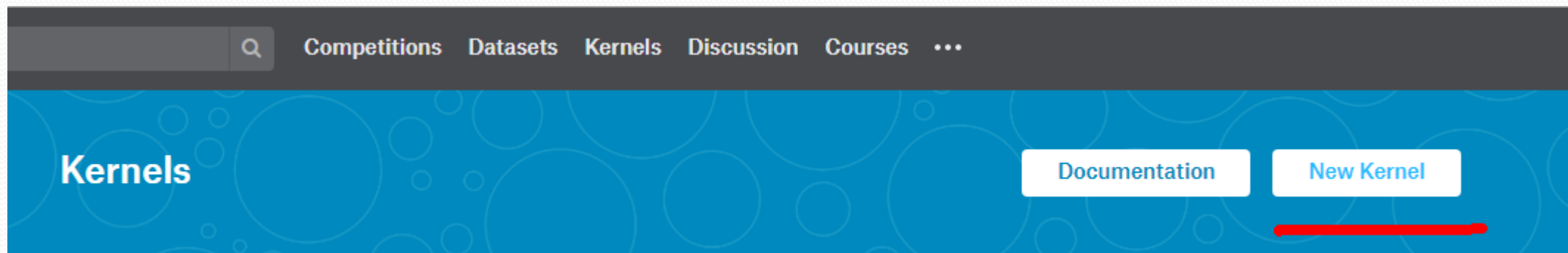
 Analyze Boston

▲

113

# Kaggle Kernels

- Kaggle Kernels là một free platform để chạy môi trường Jupyter notebooks trên trình duyệt.
- Những xử lý trên notebooks của bạn được thực hiện trên server, do đó sẽ không ảnh hưởng tới hiệu năng của máy tính.



<https://towardsdatascience.com/introduction-to-kaggle-kernels-2ad754ebf77>

# Kaggle Kernels

Select Kernel Type

Script

```
import numpy as np # linear algebra
import pandas as pd # data processing,
# Input data files are available in the
from subprocess import check_output
print(check_output(["ls", "../input"]))
# Any results you write to the current
```

- Python, R, RMarkdown
- Runs all the code, every time
- Ideal for fitting a model and competition submissions
- Shares code for review and RMarkdown reports









Notebook

Introduction

```
# Loading in the training data
train = pd.read_csv("../train.csv")
```

- Jupyter Notebooks in Python or R
- Runs cells of code and Markdown
- Ideal for interactive data exploration and polished analysis
- Shares insights through code & commentary

# Kaggle Kernels

<i>Old Resources</i>	<i>New Resources</i>
 2 CPUs	 4 CPUs
 8 GB RAM	 16 GB RAM
 512 MB Disk Space	 1 GB Disk Space
 20 min execution time	 60 min execution time

# Kaggle Kernels

k

kernel6278361ab8 *Draft saved*

+ Add Dataset

✓ Commit

→

File Run Help

```
4
5 import numpy as np # linear algebra
6 import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
7
8 # Input data files are available in the "../input/" directory.
9 # For example, running this (by clicking run or pressing Shift+Enter) will list the files in the input directory
10
11 import os
12 print(os.listdir("../input"))
13
14 print("hello world")
15 # Any results you write to the current directory are saved as output.
```

▶ Console

● Draft Session | CPU 0% | RAM 245.6MB/16GB | GPU Off | Disk 266.3MB/4.9GB

↕ CLEAR

Your Kernel is now running in the cloud.

Use the Play button or [Shift+Enter] to execute the current line of your script (or whatever's highlighted).

Enter some code at the bottom of this Console tab and press [Enter].

```
[ ]
hello world
[ ]
hello world
[ ]
hello world
```

> Enter console command here

Sessions

↕

● Draft Session

5m/9h

GPU Off

CPU

0.01%

RAM

245.6MB/16GB

Disk

266.3MB/4.9GB

Workspace

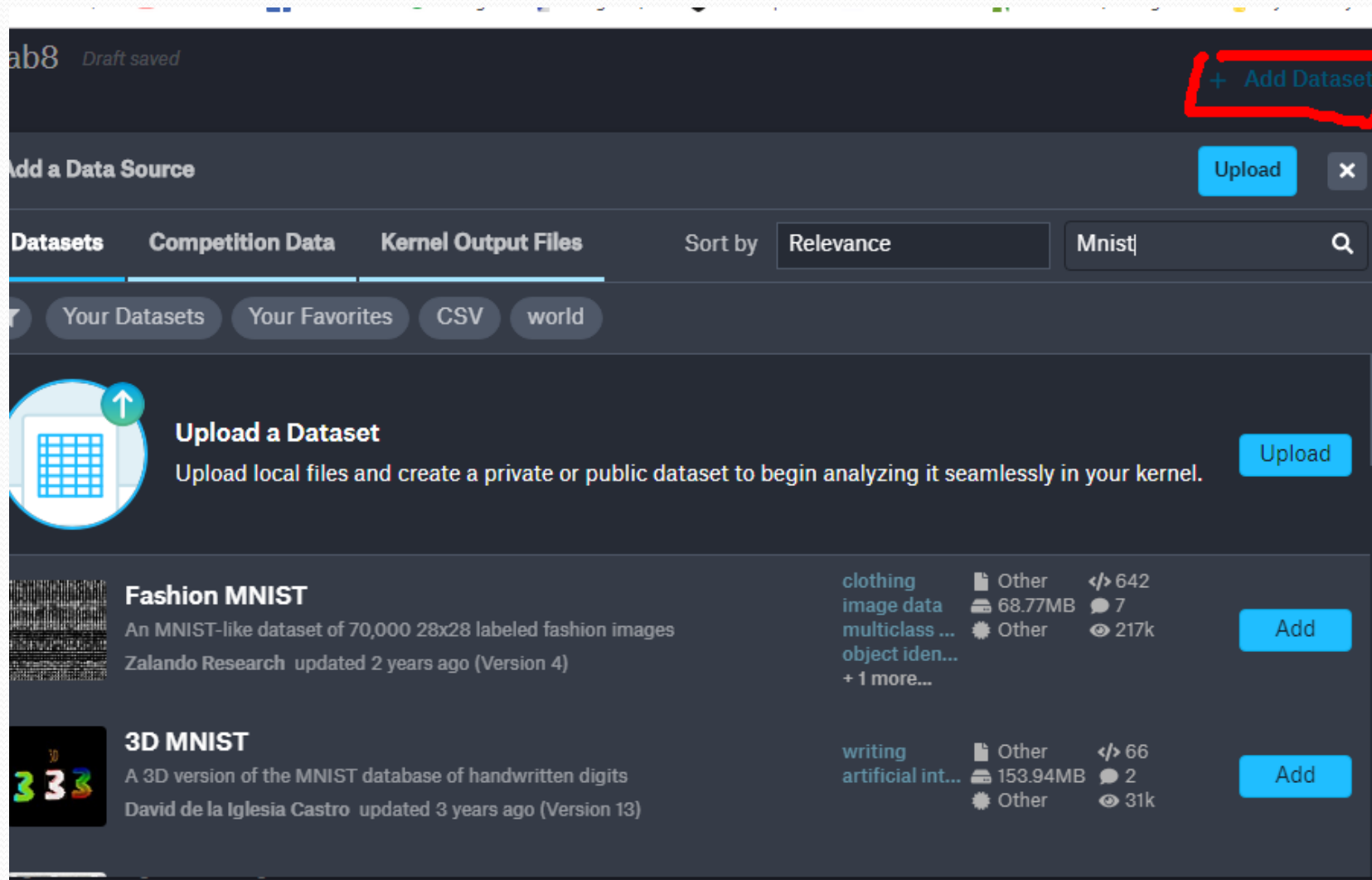
↕

input (read-only data)

Versions

↕

# Kaggle Kernels



The screenshot shows the Kaggle Kernels interface. At the top, there's a header with 'ab8 Draft saved' and a '+ Add Dataset' button highlighted with a red box. Below this is a section titled 'Add a Data Source' with an 'Upload' button and a close button. The main navigation bar includes 'Datasets', 'Competition Data', and 'Kernel Output Files'. The 'Datasets' tab is active, showing a search bar with 'Mnist' and a 'Sort by Relevance' dropdown. Below the navigation bar, there are tabs for 'Your Datasets', 'Your Favorites', 'CSV', and 'world'. The main content area features an 'Upload a Dataset' section with an 'Upload' button and a description: 'Upload local files and create a private or public dataset to begin analyzing it seamlessly in your kernel.' Below this, there are two dataset listings: 'Fashion MNIST' and '3D MNIST'. Each listing includes a thumbnail, a title, a description, the creator's name, the update date, and a list of tags. The 'Fashion MNIST' dataset is by Zalando Research, updated 2 years ago (Version 4), and includes tags like 'clothing', 'image data', 'multiclass ...', 'object iden...', and '+ 1 more...'. The '3D MNIST' dataset is by David de la Iglesia Castro, updated 3 years ago (Version 13), and includes tags like 'writing', 'artificial int...', and '+ 1 more...'. Both listings have an 'Add' button.

ab8 Draft saved

+ Add Dataset

Add a Data Source

Upload

Datasets Competition Data Kernel Output Files

Sort by Relevance

Mnist

Your Datasets Your Favorites CSV world

Upload a Dataset

Upload local files and create a private or public dataset to begin analyzing it seamlessly in your kernel.

Upload

**Fashion MNIST**

An MNIST-like dataset of 70,000 28x28 labeled fashion images

Zalando Research updated 2 years ago (Version 4)

clothing image data multiclass ... object iden... + 1 more...

Other 68.77MB 642 7 217k

Add

**3D MNIST**

A 3D version of the MNIST database of handwritten digits

David de la Iglesia Castro updated 3 years ago (Version 13)

writing artificial int... + 1 more...

Other 153.94MB 66 2 31k

Add

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import os
print(os.listdir("../input"))
print("hello world")
# Any results you write to the current directory are saved as output.
import tensorflow as tf
print(tf.__version__)
data_train_file = "../input/fashion-mnist_train.csv"
data_test_file = "../input/fashion-mnist_test.csv"
df_train = pd.read_csv(data_train_file)
df_test = pd.read_csv(data_test_file)
df_train.head()
```





# Những Lỗi Thường Gặp

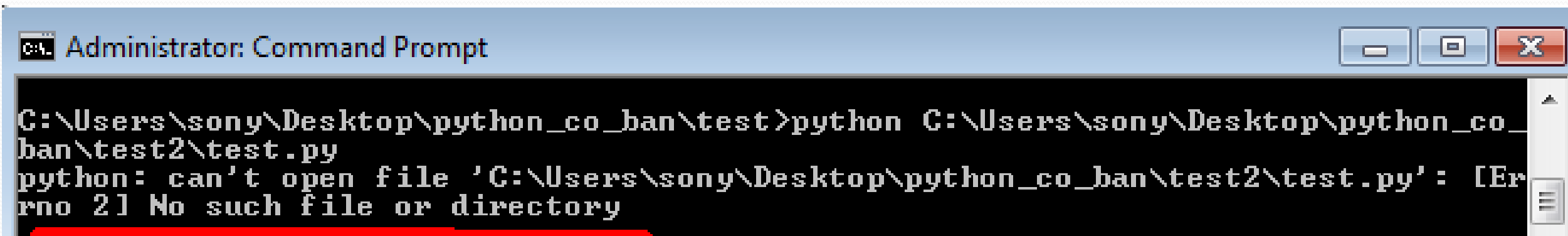
- Lỗi thiếu/ thừa khoảng trắng hoặc tab

```
# -*- coding: utf-8 -*-  
print("Hello world")  
  a = 1 # thừa khoảng trắng  
b = 2  
c = a+b  
print("c=%d" %c)
```

```
File "E1.py", line 3  
  a = 1  
  ^  
IndentationError: unexpected indent
```

# Những Lỗi Thường Gặp

- Sai đường dẫn tới thư mục chứa file cần thực thi \*.py hoặc file không tồn tại

A screenshot of a Windows Command Prompt window titled "Administrator: Command Prompt". The window has a blue title bar with standard minimize, maximize, and close buttons. The command prompt shows the following text:

```
C:\Users\sony\Desktop\python_co_ban\test>python C:\Users\sony\Desktop\python_co_ban\test2\test.py  
python: can't open file 'C:\Users\sony\Desktop\python_co_ban\test2\test.py': [Errno 2] No such file or directory
```

The error message is displayed in white text on a black background. A red horizontal line is drawn under the error message.

# Bài tập Thực hành

1. Cài đặt python 3 và IDE Pycharm

**Hoặc**

Cài đặt Anaconda và IDE Pycharm (RAM >4GB)

2. Tạo 1 python script vidu.py (dùng notepad)

- Mở Command Prompt (cmd) và thực thi file vidu.py

3. Tạo một chương trình đơn giản dùng Pycharm

```
a = 10
b = 2
c = a**b
print("c=%d" %c)
```

vidu.py

```
print("hello world")
a = 'string1'
b = 'string2'
print(a)
print(b)
print(a+b)
```

```
import time

def countdown(time_sec):
    while time_sec:
        mins, secs = divmod(time_sec, 60)
        timeformat = '{:02d}:{:02d}'.format(mins, secs)
        print(timeformat, end='\r')
        time.sleep(1)
        time_sec -= 1

    print("stop")

countdown(20)
```