

# **TÀI LIỆU**

**Hướng dẫn cài đặt và thiết lập môi trường  
cho phần thực hành**

# Môi trường phát triển tích hợp (IDEs) cho Python

+ **IDEs Offline:** Sử dụng bộ công cụ Anaconda



+ **IDEs Online:** Sử dụng Colab của Google






# 1. Hướng dẫn cài đặt, sử dụng Anaconda/Jupyter Notebook



# 1. Sử dụng Anaconda

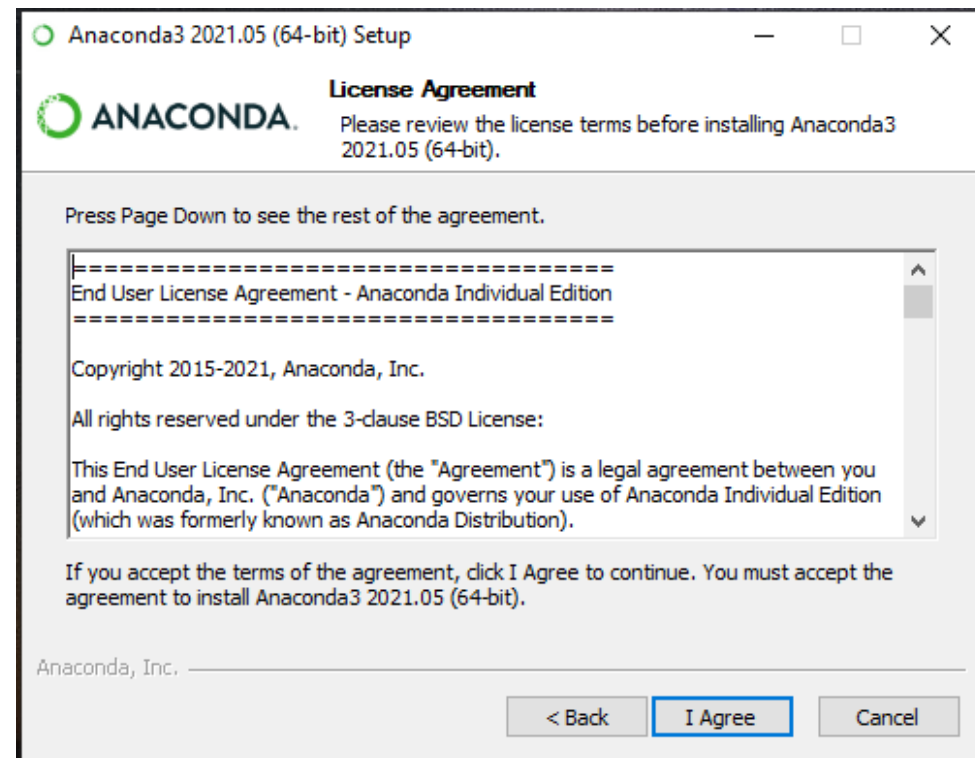
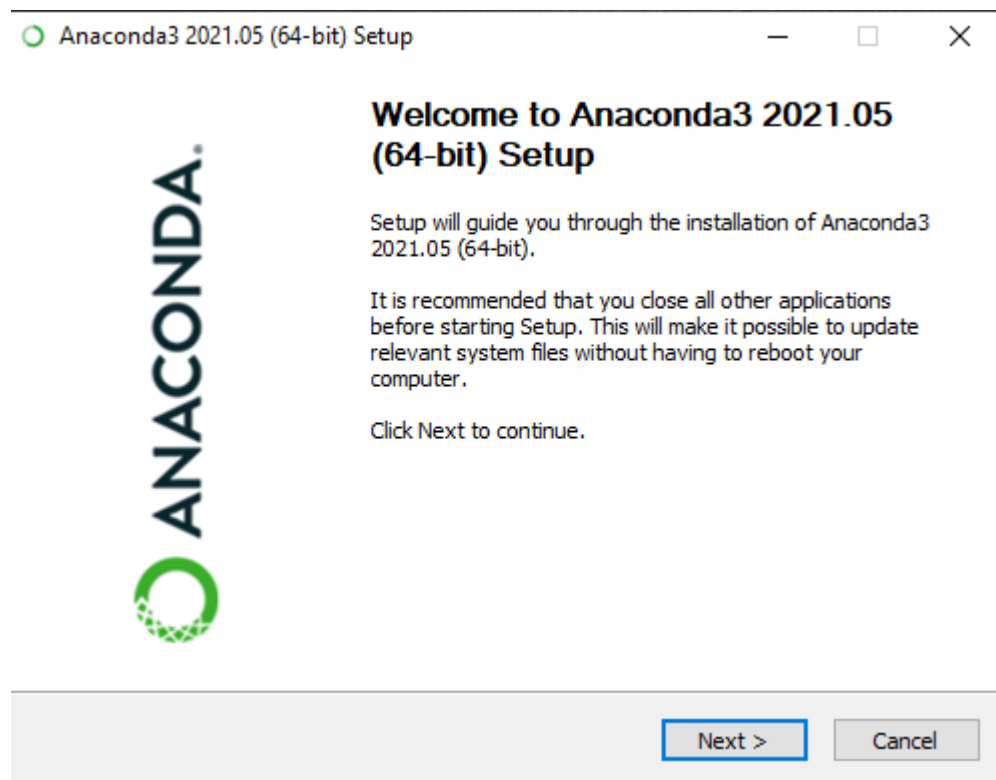
- Anaconda là một bản phân phối mã nguồn mở miễn phí của ngôn ngữ lập trình Python, để xử lý dữ liệu quy mô lớn, phân tích dự đoán và tính toán khoa học, nhằm mục đích đơn giản hóa việc quản lý và deploy các package.
  - <https://www.anaconda.com/products/individual#Downloads>
- ➡ Tải bộ cài và setup môi trường

Anaconda Installers

Windows 	MacOS 	Linux 
Python 3.8	Python 3.8	Python 3.8
64-Bit Graphical Installer (466 MB)	64-Bit Graphical Installer (462 MB)	64-Bit (x86) Installer (550 MB)
32-Bit Graphical Installer (397 MB)	64-Bit Command Line Installer (454 MB)	64-Bit (Power8 and Power9) Installer (290 MB)

# Cài đặt Anaconda

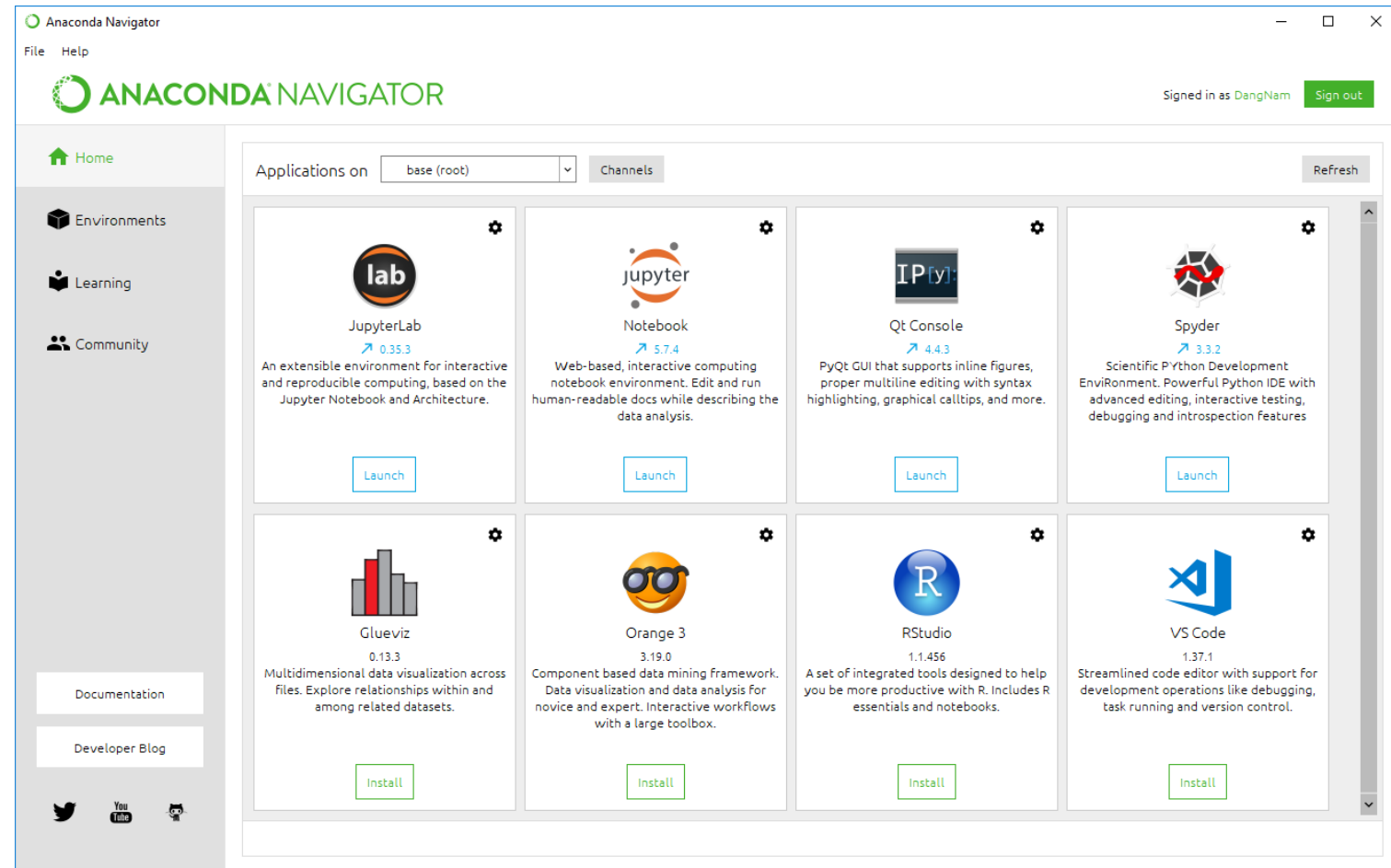
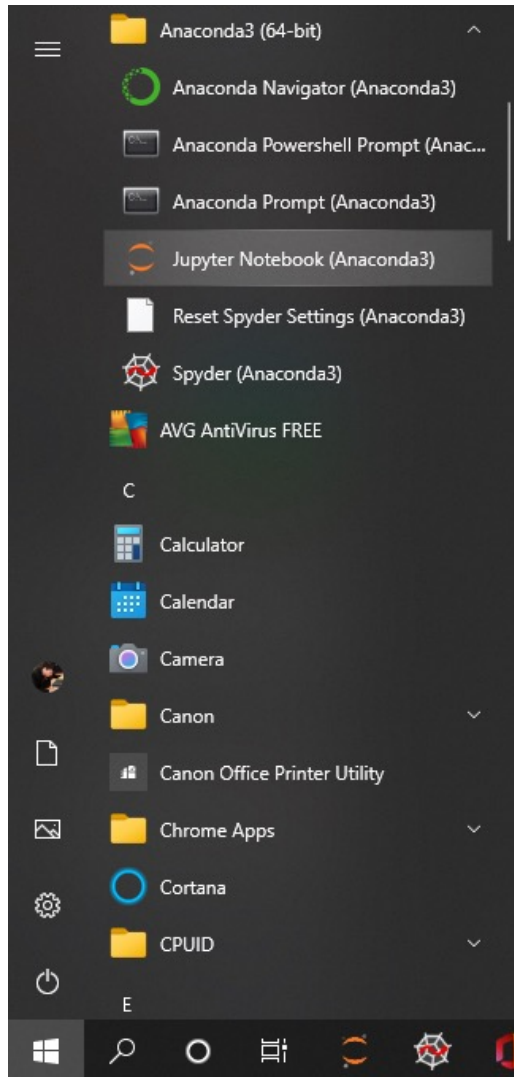
Chạy file setup đã tải và cài đặt theo các bước như với các phần mềm khác

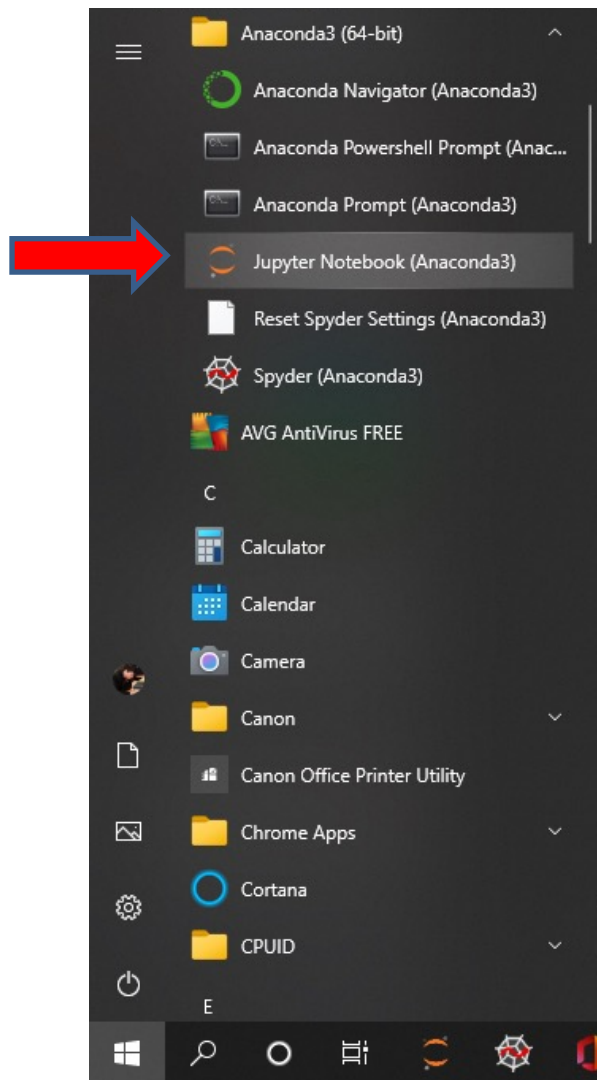


Youtube hướng dẫn setup

[https://www.youtube.com/watch?v=g6ln1dAt-RI&ab\\_channel=Ritman](https://www.youtube.com/watch?v=g6ln1dAt-RI&ab_channel=Ritman)

# Cài đặt Anaconda thành công!



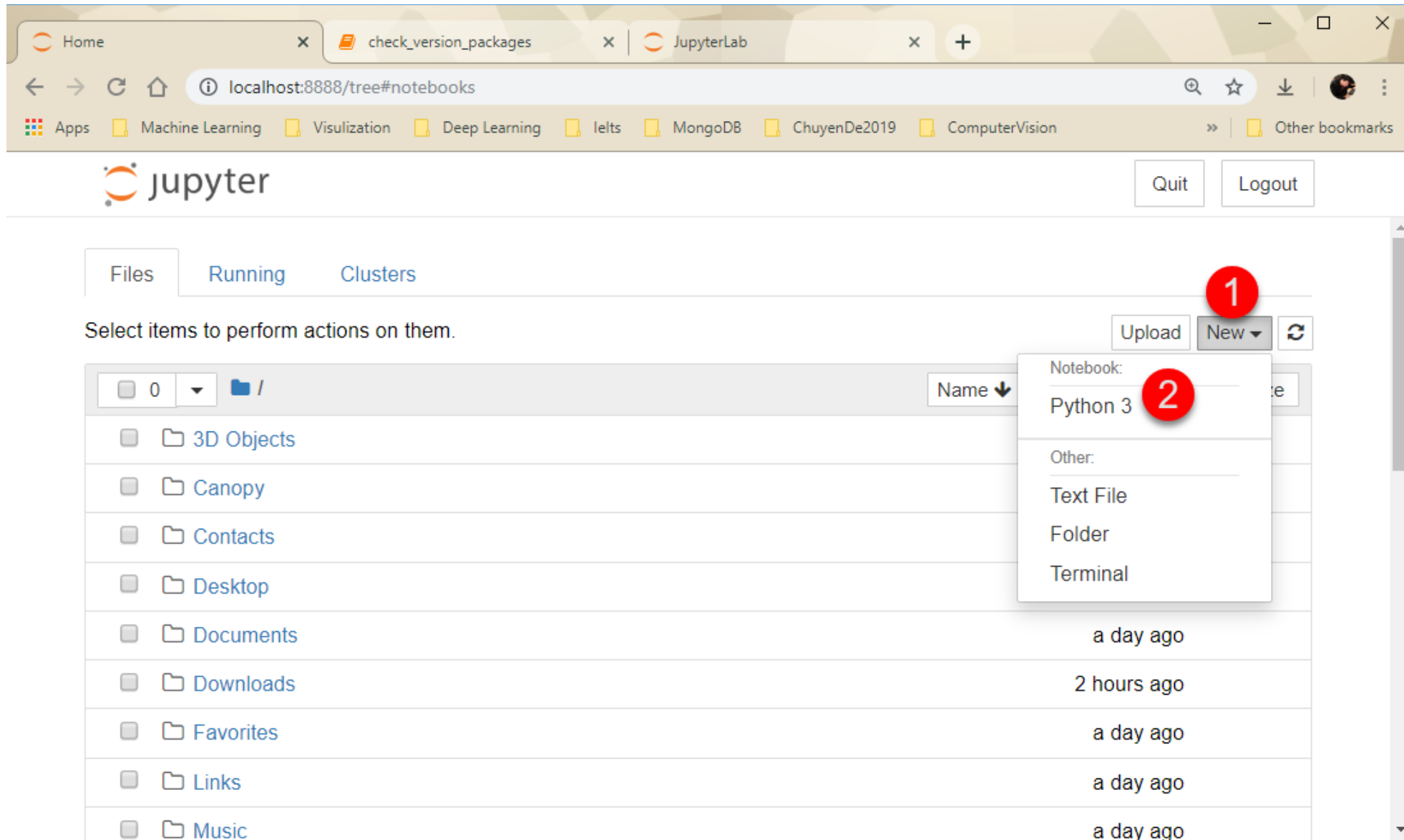


Jupyter notebook là môi trường chạy code python ở dạng interactive.

Interactive có điểm mạnh là cho ta biết kết quả từng bước trong chương trình.

Điều này sẽ giúp ích rất nhiều cho lập trình viên, đặc biệt cho những người mới học Python.

# Tạo file Python mới với Jupyter Notebook



The screenshot displays the JupyterLab web interface in a browser. The browser's address bar shows the URL `localhost:8888/tree#notebooks`. The JupyterLab header includes the logo, a 'Quit' button, and a 'Logout' button. Below the header, there are tabs for 'Files', 'Running', and 'Clusters'. The 'Files' tab is active, showing a file browser with a list of folders. A red circle with the number '1' highlights the 'New' button in the top right corner of the file browser. A dropdown menu is open, showing options for creating new files or folders. A red circle with the number '2' highlights the 'Python 3' option under the 'Notebook:' section. The dropdown menu also includes options for 'Text File', 'Folder', and 'Terminal' under the 'Other:' section. The file browser list shows folders like '3D Objects', 'Canopy', 'Contacts', 'Desktop', 'Documents', 'Downloads', 'Favorites', 'Links', and 'Music'.

Home x check\_version\_packages x JupyterLab x +

localhost:8888/tree#notebooks

Apps Machine Learning Visualization Deep Learning Ielts MongoDB ChuyenDe2019 ComputerVision Other bookmarks

jupyter Quit Logout

Files Running Clusters

Select items to perform actions on them.

Upload New 1

Notebook: 2  
Python 3

Other:  
Text File  
Folder  
Terminal

0 /

Name

3D Objects	
Canopy	
Contacts	
Desktop	
Documents	a day ago
Downloads	2 hours ago
Favorites	a day ago
Links	a day ago
Music	a day ago



# Giao diện cơ bản của Jupyter Notebook

Home x check\_version\_packages x +

localhost:8888/notebooks/check\_version\_packages.ipynb

Apps Machine Learning Visualization Deep Learning Ielts MongoDB ChuyenDe2019 ComputerVision Other bookmarks

jupyter check\_version\_packages (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

Save Add Split Cell Merge Cell Up Down Run Stop Restart Code

**BÀI THỰC HÀNH SỬ DỤNG CÁC THƯ VIỆN PYTHON TRONG KHOA HỌC DỮ LIỆU**

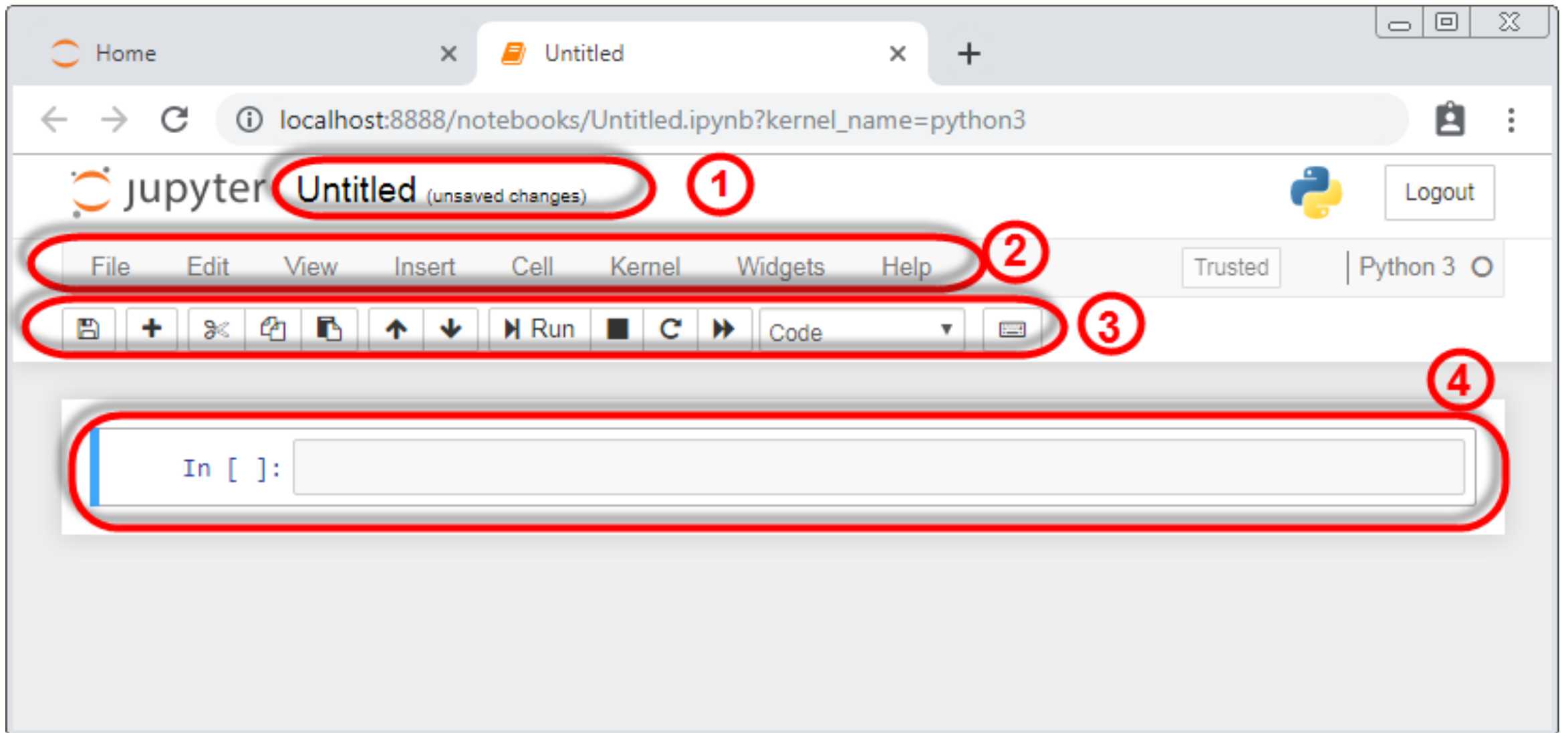
1. Kiểm tra thư viện đã cài đặt hay chưa và phiên bản bao nhiêu?

```
In [ ]: 1 #Khai báo sử dụng thư viện và kiểm tra phiên bản thư viện đang sử dụng
        2 import numpy as np
        3 print("Thư viện Numpy, Version: ",np.__version__)
```

```
In [ ]: 1 import matplotlib as mpl
        2 print("Thư viện Matplotlib, Version: ",mpl.__version__)
```

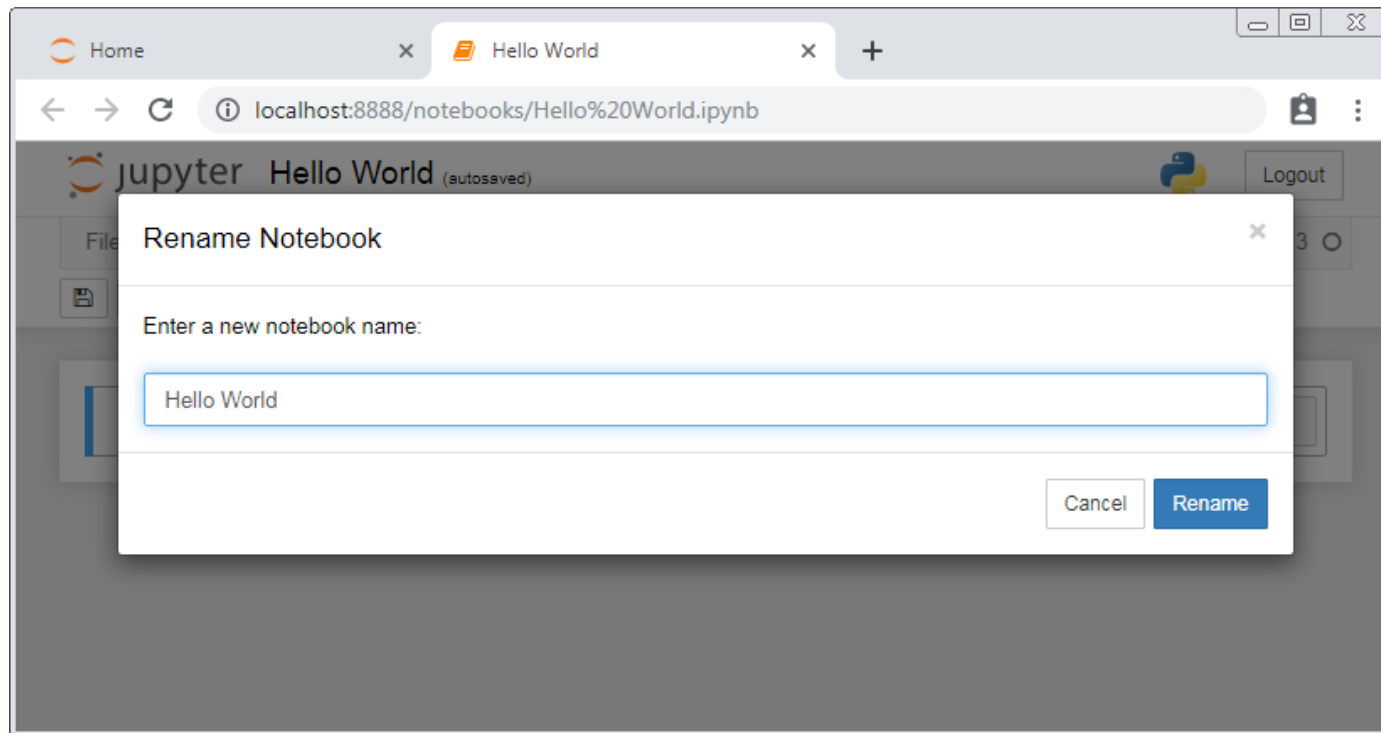
```
In [ ]: 1 import pandas as pd
```

# Sử dụng công cụ Jupyter notebook



# Phần 1

- Tiêu đề tài liệu, mặc định chưa nhập tiêu đề là Untitled.
- Khi nhấp chuột vào phần này, một cửa sổ sẽ hiện ra để bạn nhập tên file.



# Phần 2

File Edit View Insert Cell Kernel Widgets Help

- **File:** các menu thao tác với file như tạo mới, lưu trữ, save as...
- **Edit:** các menu thao tác với nội dung file như copy, paste văn bản, cell, tìm kiếm và thay thế một nội dung trong file, chèn hình ảnh...
- **View:** menu điều chỉnh giao diện, hiển thị hoặc ẩn các thành phần giao diện trong Jupyter Notebook.
- **Insert, Cell:** các hành động liên quan đến Cell là thành phần cơ bản của một tài liệu.
- **Kernel:** các thao tác với kernel hiện hành như thay đổi nếu muốn sử dụng ngôn ngữ khác, restart, shutdown
- Một số các menu khác...

# Phần 3

Thanh menu nhanh (shortcut menu) bao gồm các thao tác nhanh mà bạn thường làm với một tài liệu, có một số nút trên đó chúng ta hay dùng nhất như:

- Save file
- Copy, paste
- Thay đổi thứ tự các cell trong tài liệu
- Chạy code trong tài liệu
- Lựa chọn loại văn bản trong cell:
- Code: Cell này chứa mã Python và có thể thực hiện được.
- Markdown: Cell này chứa văn bản diễn giải



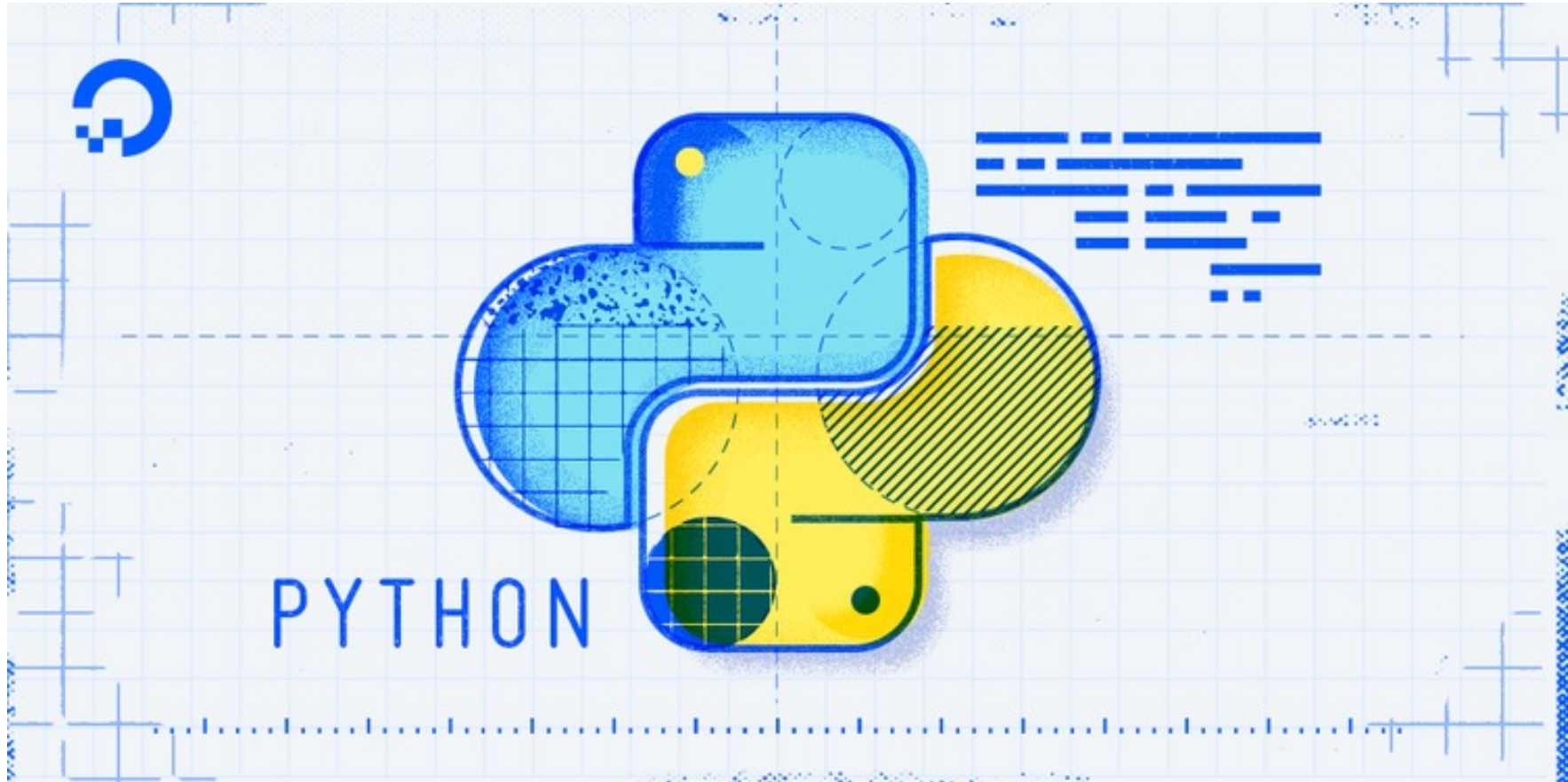
## Phần 4

Nội dung tài liệu chứa các cell có nội dung là code hoặc markdown, đây là vùng chúng ta sẽ làm việc chính.

```
In [7]: f = open("HUMG.txt",'r',encoding = 'utf-8')
```

```
In [8]: a = f.read(12) # đọc 12 kí tự đầu tiên
        print('Nội dung 11 kí tự đầu là:\n', (a))
        b = f.read(35) # đọc 35 kí tự tiếp theo
        print('Nội dung 35 kí tự tiếp theo là:\n', (b))
        c = f.read() # đọc phần còn lại
        print('Nội dung phần còn lại là:\n', (c))
```

# Xây dựng ứng dụng đầu tiên “Hello world” với Python



# Xây dựng chương trình đầu tiên với Jupyter notebook

---

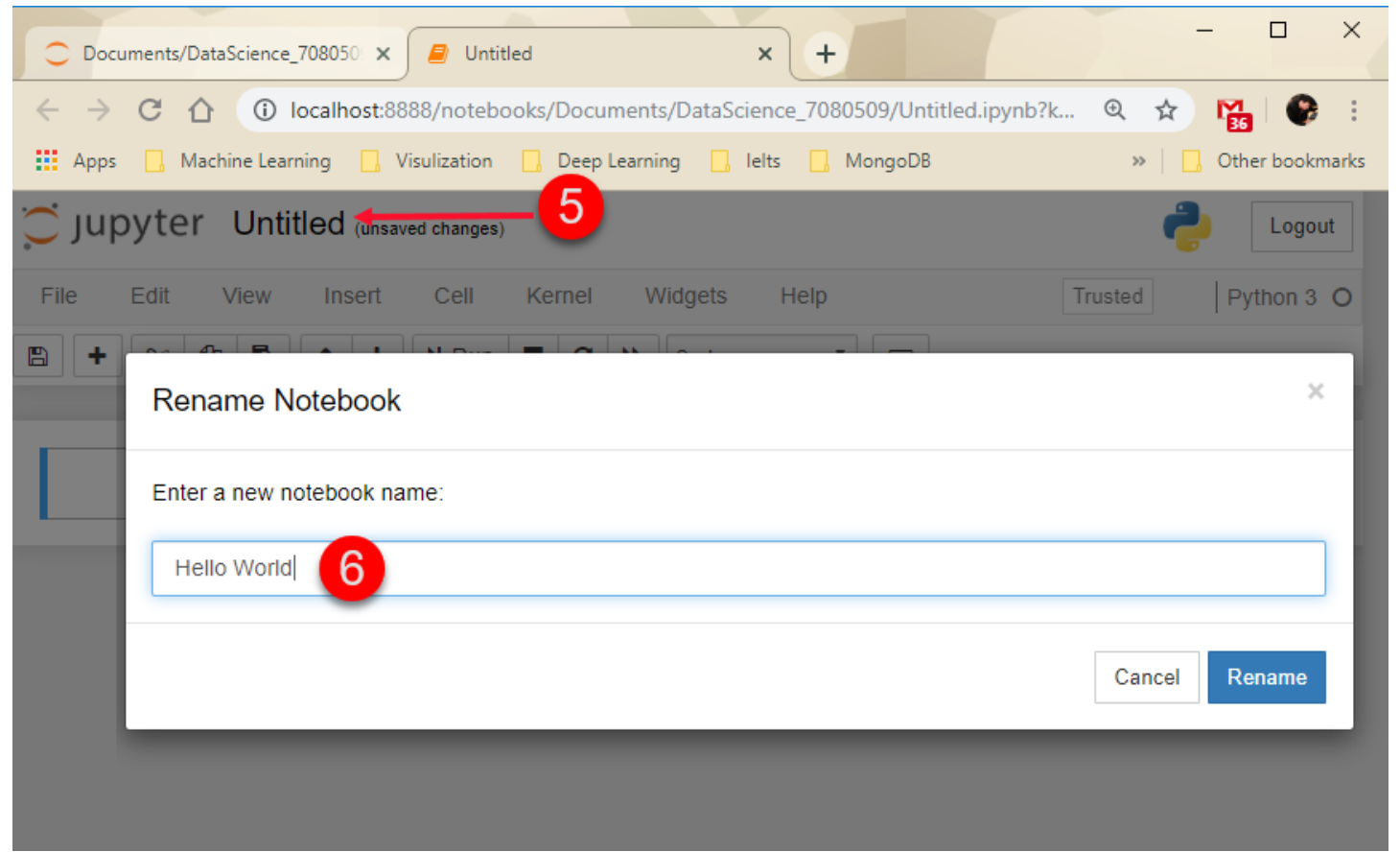
- **Bước 1:** Tạo thư mục **Practice\_Course** trong **Document** để lưu trữ toàn bộ file mã nguồn và dữ liệu của khóa học.
- **Bước 2:** Chạy Jupyter Notebook
- **Bước 3:** Truy cập tới thư mục **Practice\_Course** đã tạo
- **Bước 4:** Chọn New/Python3 để tạo file code mới



# Xây dựng chương trình đầu tiên với Jupyter notebook

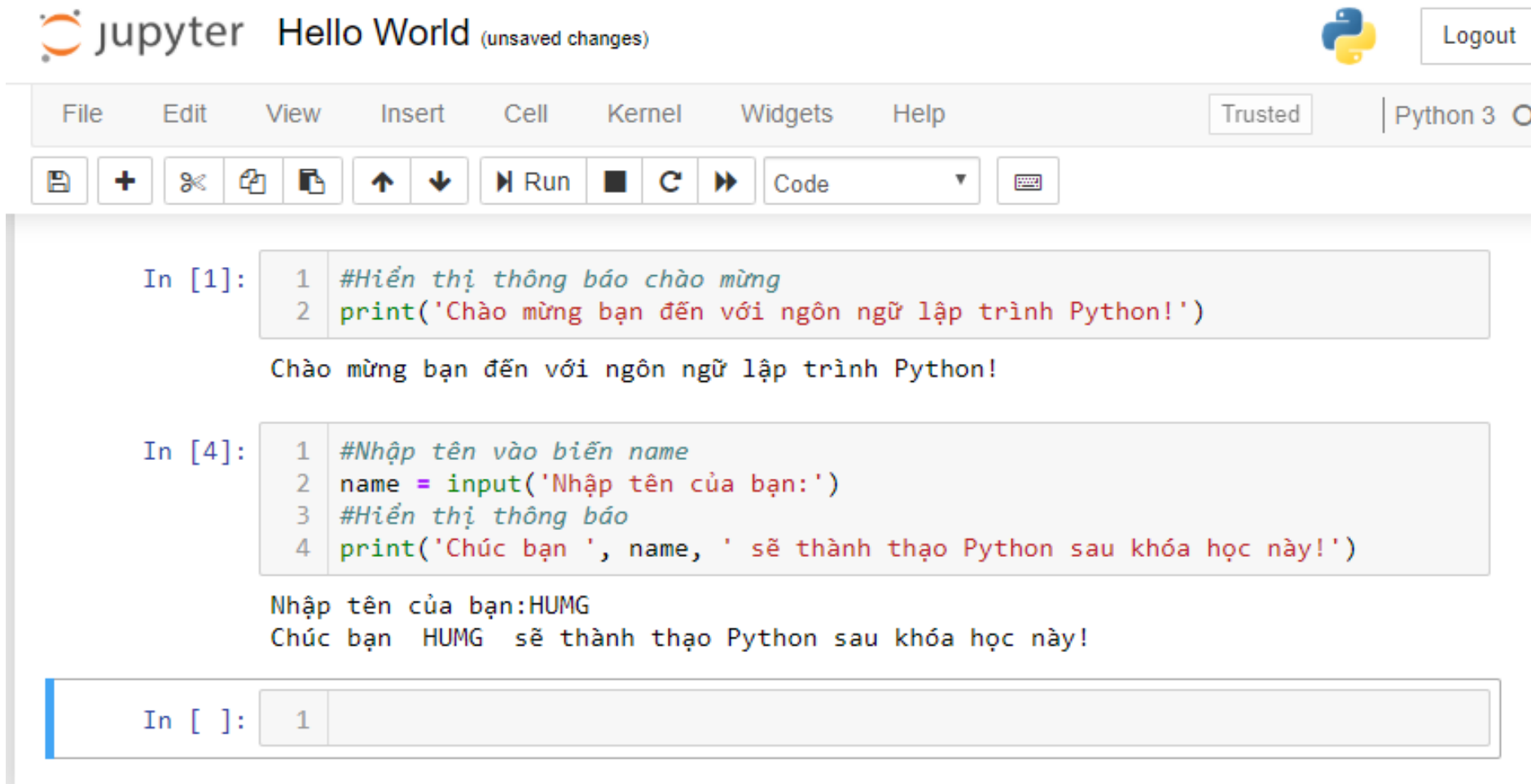
- **Bước 5:** Tab giao diện code mới mở ra, kích vào tên mặc định Untitled để đặt lại tên cho file code.
- **Bước 6:** Nhập tên mới cho file

**Bây giờ, Chúng ta đã sẵn sàng cho những dòng code đầu tiên!**




# Xây dựng chương trình đầu tiên với Jupyter notebook













Hãy nhập và chạy những dòng code Python đầu tiên!



The screenshot displays the Jupyter Notebook interface. At the top, the title bar shows "jupyter Hello World (unsaved changes)" and a "Logout" button. Below the title bar is a menu bar with options: File, Edit, View, Insert, Cell, Kernel, Widgets, Help. To the right of the menu bar are "Trusted" and "Python 3" indicators. Below the menu bar is a toolbar with icons for saving, adding, deleting, and running code. The main area contains two code cells. The first cell, labeled "In [1]:", contains two lines of Python code: `#Hiển thị thông báo chào mừng` and `print('Chào mừng bạn đến với ngôn ngữ lập trình Python!')`. The output of this cell is the text "Chào mừng bạn đến với ngôn ngữ lập trình Python!". The second cell, labeled "In [4]:", contains four lines of Python code: `#Nhập tên vào biến name`, `name = input('Nhập tên của bạn:')`, `#Hiển thị thông báo`, and `print('Chúc bạn ', name, ' sẽ thành thạo Python sau khóa học này!')`. The output of this cell shows the user input "HUMG" and the resulting message "Chúc bạn HUMG sẽ thành thạo Python sau khóa học này!". A third, empty code cell labeled "In [ ]:" is visible at the bottom.

jupyter Hello World (unsaved changes)  Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

           Code 

In [1]: 

```
1 #Hiển thị thông báo chào mừng
2 print('Chào mừng bạn đến với ngôn ngữ lập trình Python!')
```

Chào mừng bạn đến với ngôn ngữ lập trình Python!

In [4]: 

```
1 #Nhập tên vào biến name
2 name = input('Nhập tên của bạn:')
3 #Hiển thị thông báo
4 print('Chúc bạn ', name, ' sẽ thành thạo Python sau khóa học này!')
```

Nhập tên của bạn:HUMG  
Chúc bạn HUMG sẽ thành thạo Python sau khóa học này!

In [ ]: 

```
1
```

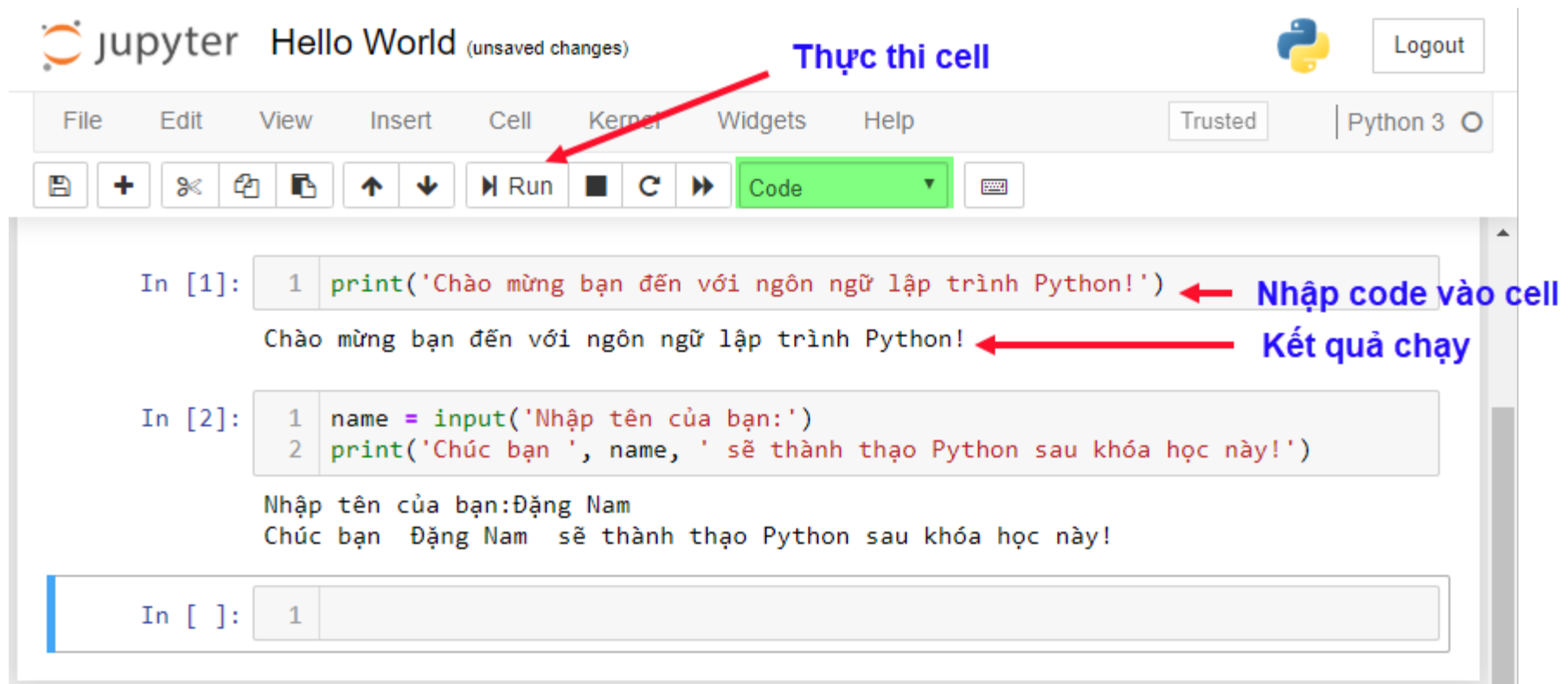
## 2 loại cell căn bản: Markdown – Code

- Cell:** Là thành phần căn bản nhất trong một Notebooks và được chia làm 2 loại là Code và Markdown.

The screenshot shows the Jupyter Notebook interface with the title "Hello World (unsaved changes)". The top menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. The "Cell" menu is open, showing options: Code, Code, Markdown, Raw NBConvert, and Heading. An orange arrow labeled "Thay đổi chế độ cell" points to the "Cell" menu. A yellow arrow labeled "Markdown" points to a Markdown cell containing the text "CHƯƠNG TRÌNH HELLO WORLD" and a paragraph: "Đây là chương trình đầu tiên làm quen với bất kỳ một ngôn ngữ nào! Chúng ta cũng sẽ bắt đầu nó với ngôn ngữ Python". Two orange arrows labeled "Cell Code" point to two Code cells. The first Code cell contains the code `print('Chào mừng bạn đến với ngôn ngữ lập trình Python!')` and its output is "Chào mừng bạn đến với ngôn ngữ lập trình Python!". The second Code cell contains the code `name = input('Nhập tên của bạn:')` and `print('Chúc bạn ', name, ' sẽ thành thạo Python sau khóa học này!')` and its output is "Nhập tên của bạn:Đặng Nam" followed by "Chúc bạn  Đặng Nam  sẽ thành thạo Python sau khóa học này!".

## 2 loại cell căn bản: Markdown – Code

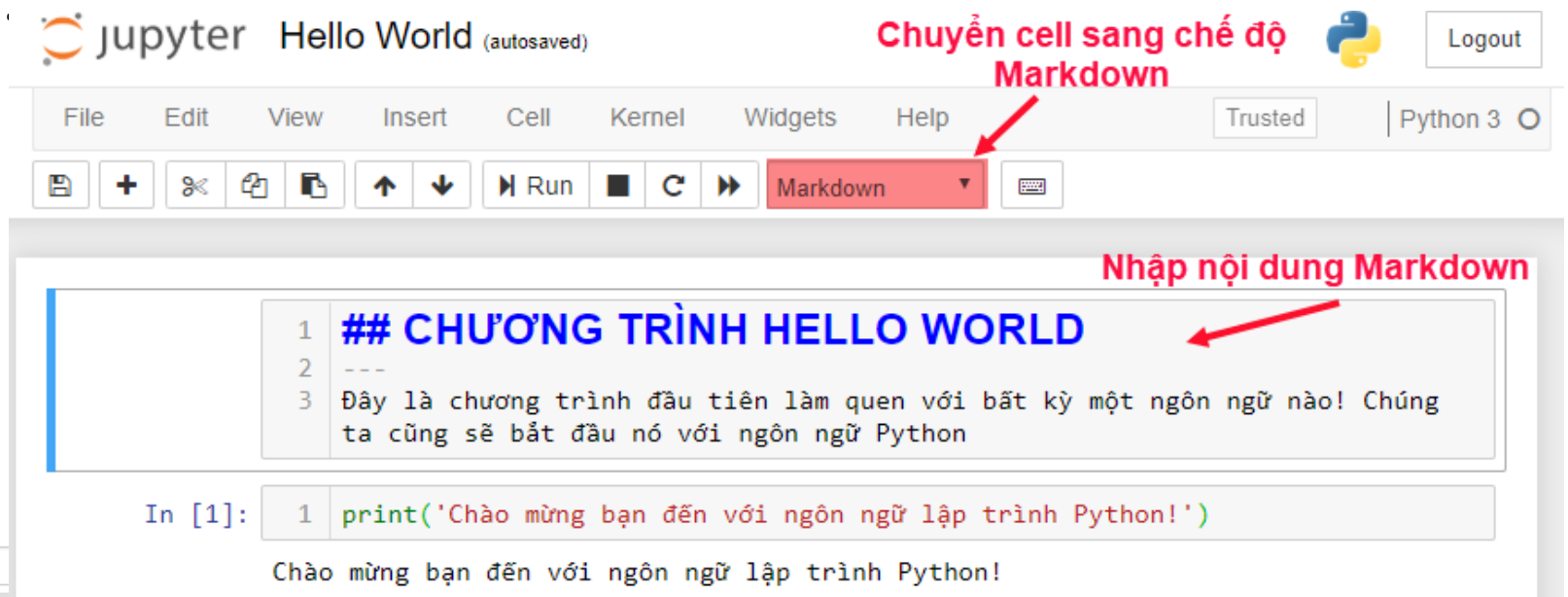
- **Code (mặc định khi thêm mới 1 cell):** Cho phép người dùng thực hiện việc lập trình và thực thi các câu lệnh được viết trong 1 cell. Kết quả trả về sẽ được ghi ngay bên dưới cell đó.



Phím tắt để thực thi một cell: **Shift + Enter** hoặc **Ctrl + Enter**

## 2 loại cell căn bản: Markdown – Code

- **Markdown:** Cho phép người dùng có thể dễ dàng soạn thảo, đọc và trang trí đoạn văn bản, có thể chèn hình ảnh, video....là công cụ hữu ích cho phép ghi chú lại các nội dung bài học, chú thích và tài liệu đi kèm...



Kết quả hiển thị khi chạy cell Markdown

### CHƯƠNG TRÌNH HELLO WORLD

Đây là chương trình đầu tiên làm quen với bất kỳ một ngôn ngữ nào! Chúng ta cũng sẽ bắt đầu nó với ngôn ngữ Python

```
In [1]: 1 print('Chào mừng bạn đến với ngôn ngữ lập trình Python!')
```

# Hàm nhập dữ liệu từ bàn phím: input()

- Trong Python để nhập liệu từ bàn phím ta dùng hàm **input()**
- Cú pháp:
  - `input('chuỗi hiển thị:')`
  - `Biến = input("chuỗi hiển thị")`

**Lưu ý:** Kết quả trả về của câu lệnh này sẽ là một giá trị thuộc kiểu **dữ liệu chuỗi**

```
In [2]: input()
```

4

```
Out[2]: '4'
```

```
In [3]: input('Hãy nhập 1 số bất kỳ')
```

Hãy nhập 1 số bất kỳ4

```
Out[3]: '4'
```

```
In [4]: a = input('Hãy nhập số a: ')
```

Hãy nhập số a: 5

# Hàm xuất dữ liệu ra màn hình: print()

- Hàm **print()** dùng để xuất dữ liệu ra màn hình trên các dòng khác nhau
- Cú pháp:
  - `print("ký tự")`
  - `print(biến)`
  - `print('ký tự', biến)`

In [1]: `print('Tôi yêu HUMG')`

Tôi yêu HUMG

In [2]: `a=1`  
`print(a)`

1

In [4]: `k=63`  
`print("Chúng tôi là SV k", k)`

Chúng tôi là SV k 63

In [5]: `k=63`  
`ns=2001`  
`print("Chúng tôi là SV k", k, 'Chúng tôi sinh năm', ns)`

Chúng tôi là SV k 63 Chúng tôi sinh năm 2001

# Chú thích trong Python

- Chương trình càng lớn thì càng phức tạp, từ đó việc đọc code sẽ khó khăn hơn.
- Đặc biệt với các chương trình mã nguồn mở, việc code dễ đọc và hiểu là rất quan trọng.  
=> sử dụng chú thích (comments) trong chương trình là một việc làm cần thiết.
- Trong Python: Chú thích được bắt đầu sau ký tự **#** (một dòng).

```
In [1]: list1 = [] # list rỗng  
list2 = [1, 2, 3] # list số nguyên  
list3 = [1, "Hello", 3.4] # list với kiểu dữ liệu hỗn hợp
```



# Bài tập 1

- Thêm một cell code mới, nhập đoạn mã dưới đây, chạy và cho biết kết quả!

```
In [3]: 1  #khởi tạo biến a = 0; A = 0
        2  a = 0
        3  A = 0
        4  for i in range(1,6):
        5      print(i)
        6      a = a + i
        7
        8  #Hiển thị kết quả biến a
        9  print('Giá trị của biến a = ',a)
       10  print('-----')
       11
       12  for i in range (1,6,2):
       13      print(i)
       14      A = A + i
       15
       16  #HIển thị kết quả biến A
       17  print('Giá trị của biến A = ',A)
       18
```

# Bài tập 2

- Tạo một file mới, đặt tên: Calculator.ipynb
- Viết code: nhập vào 2 số a, b và thực hiện để hiển thị kết quả như sau:

In [1]:

```
1  #-----  
2  #{Code của bạn ở đây}  
3  #-----  
4
```

Nhập vào số a:9

Nhập vào số b:5

-----

Tổng a + b = 14

Hiệu a - b = 4

Tích a \* b = 45

Thương a / b = 1.8

**Lưu ý:** lệnh `num_a = int(a)` sẽ chuyển đổi kiểu dữ liệu của a sang kiểu int (số nguyên) và gán giá trị cho biến num\_a

## 2. Hướng dẫn Sử dụng Google colab



# Google colab

- Google colab là một dịch vụ đám mây miễn phí, hiện nay có hỗ trợ GPU (Tesla K80) và TPU (TPUv2).
- Google colab được phát triển dựa trên Jupiter Notebook nên việc sử dụng Google Colab cũng tương tự như việc sử dụng Jupyter Notebook.
- Google Colab là một công cụ lý tưởng để rèn luyện kỹ năng lập trình với ngôn ngữ Python thông qua các thư viện của deep learning. Google Colab cài đặt sẵn cho chúng ta những thư viện rất phổ biến trong nghiên cứu Deep Learning như PyTorch, TensorFlow, Keras và OpenCV.



# Google colab

The screenshot shows a Google Colab notebook interface. The browser address bar displays the URL: `colab.research.google.com/drive/1Vnuqb3vs_VrtUd8liTUtKzFAz48lhDd`. The notebook title is "Baitoan1\_TitanicDisaster". The left sidebar contains a "Table of contents" with the following items:

- Bài toán 1: THẨM HỌA ĐẦM TÀU TITANIC (HỌC CÓ GIÁM SÁT - Binary Classification)
- 1) Tải tập dữ liệu Train - Test
- 2) Phân tích số liệu tập Train
- 3) Trực quan hóa tập dữ liệu train
- 3) Tiền xử lý dữ liệu
  - 3.1) Loại bỏ các thuộc tính không sử dụng cho mô hình
  - 3.2) Bổ sung thêm thuộc tính mới từ thuộc tính đã có trong tập dữ liệu
    - a) Tạo thuộc tính mới từ thuộc tính Title từ thuộc tính Name.
    - b) Tạo thuộc tính IsAlone, cho biết hành khách đi một mình hay đi với người thân trên tàu
  - 3.3) Chuyển đổi dữ liệu thuộc tính sang

The main content area shows the following code snippets:

```
[ ] 1 #Gọi các thư viện sử dụng cho Project:
2
3 # Gọi các package thực hiện chuẩn bị dữ liệu
4 import pandas as pd
5 import numpy as np
6 import random as rnd
7
8 # Gọi các package hỗ trợ trực quan hóa dữ liệu (visualization)
9 import seaborn as sns
10 import matplotlib.pyplot as plt
11 %matplotlib inline
12
```

Below the code, the section "1) Tải tập dữ liệu Train - Test" is expanded, showing the following details:

- Train: 891 mẫu (Một số thông tin về tập Train: tổng số 891 - chiếm 40% số lượng hành khách thực tế trên tàu (2.224), Gần 30% hành khách có anh chị em và/hoặc vợ, chồng trên tàu; Giá vé thay đổi đáng kể <1% hành khách trả cao tới 512USD; Số lượng hành khách cao tuổi <1% trong độ tuổi 65-80 tuổi; 65% hành khách là Nam (577/891); bao gồm 12 thuộc tính như ở phần mô tả bài toán)
- Test: 418 mẫu (bao gồm 11 thuộc tính (Ko chứa thuộc tính Survival))

Below the details, the following code snippet is shown:

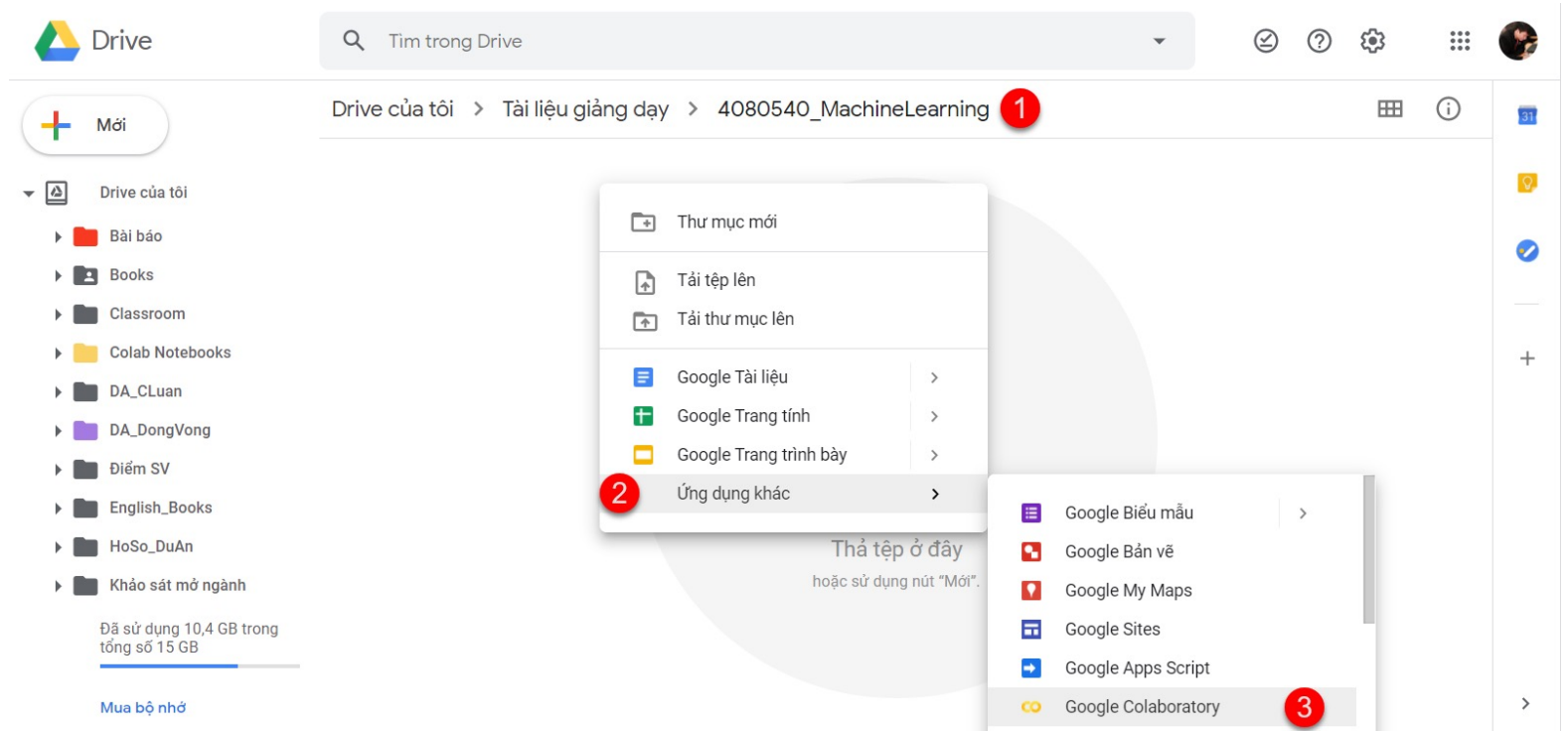
```
[ ] 1 #Mount tới google drive để tải tập dữ liệu
2
3 from google.colab import drive
4 drive.mount('/content/drive')
```

At the bottom, a message indicates: "Drive already mounted at /content/drive: to attempt to forcibly remount, call drive.mount('/c".

[Link hướng dẫn sử dụng Google colab!](#)

# Google colab

- Truy cập vào Google Drive của bạn để bắt đầu!
- **Bước 1:** Tạo thư mục **Practice\_Course** để lưu trữ toàn bộ file mã nguồn và dữ liệu.
- **Bước 2:** Kích chuột phải vào thư mục vừa tạo chọn “Ứng dụng khác”
- **Bước 3:** Chọn Google Colaboratory để thực hiện viết chương trình đầu tiên!



# Google colab

- Hãy viết chương trình đầu tiên của bạn!
- **Bước 1:** Đặt tên file: 'HelloWorld'
- **Bước 2:** Nhập đoạn lệnh vào Cell
- **Bước 3:** Kích chọn nút Run để chạy mã lệnh

The screenshot displays the Google Colab web interface. At the top, the file is named 'HelloWorld.ipynb' with a red circle '1' next to it. The menu bar includes 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. On the right, there are buttons for 'Comment', 'Share', and a user profile icon. Below the menu, there are tabs for '+ Code' and '+ Text', and a status bar showing 'RAM' and 'Disk' usage, along with an 'Editing' mode indicator. The main workspace shows a code cell titled 'Chương trình đầu tiên của tôi - Hello World!'. A red circle '3' is next to the 'Run' button (a play icon). The code inside the cell is as follows:

```
1 #Dòng chú thích:  
2 #Bạn hãy Nhập vào tên của bạn  
3  
4 st_name = input('Tên bạn là gì?')  
5  
6 #Hiển thị câu chào mừng!  
7  
8 print('CHÀO MỪNG BẠN ',st_name, ' ĐẾN VỚI LẬP TRÌNH PYTHON TRONG KHOA KHỌC DỮ LIỆU!')  
9
```

A red circle '2' is next to the input function in line 4. Below the code cell, the output is displayed: 'Tên bạn là gì?Đặng Nam' followed by the printed message 'CHÀO MỪNG BẠN Đặng Nam ĐẾN VỚI LẬP TRÌNH PYTHON TRONG KHOA KHỌC DỮ LIỆU!'.

**MỘT SỐ NỘI DUNG THAM KHẢO THÊM!**



## “Hello, World”

- C

```
#include <stdio.h>

int main(int argc, char ** argv)
{
    printf("Hello, World!\n");
}
```

- Java

```
public class Hello
{
    public static void main(String argv[])
    {
        System.out.println("Hello, World!");
    }
}
```

- now in Python

```
print "Hello, World!"
```

## (Tham khảo 2) Các loại lỗi thường gặp trong lập trình

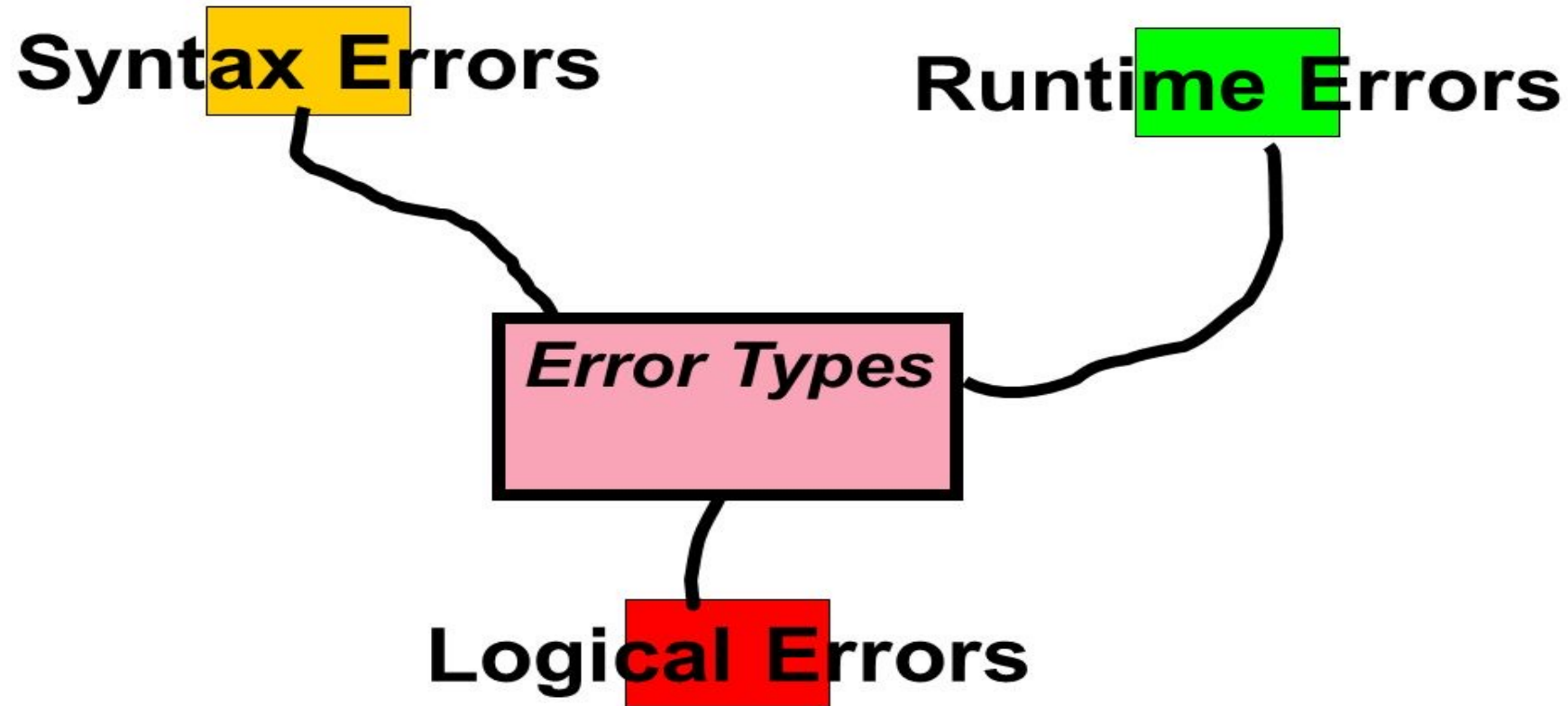


## (Tham khảo 2) Các loại lỗi thường gặp trong lập trình

- Trong các ngôn ngữ lập trình nói chung, dù là một lập trình viên chuyên nghiệp hay mới chập chững vào nghề đều có thể **gặp lỗi trong quá trình lập trình**.
- Tùy theo khả năng của từng lập trình viên mà có thể mắc lỗi nhiều hay ít.
- Đối với lập trình viên đã đi theo con đường lập trình, chắc chắn sẽ gặp lỗi, quan trọng là **khi gặp lỗi**, chúng ta sẽ **giải quyết lỗi đó như thế nào**.



## (Tham khảo 2) Các loại lỗi thường gặp trong lập trình



## (Tham khảo 2) Các loại lỗi thường gặp trong lập trình

**1) Lỗi cú pháp (Syntax Error):** Đây là loại lỗi sơ đẳng nhất trong lập trình. Thường là do gõ **sai cấu trúc của ngôn ngữ** (ví dụ như thiếu dấu kết thúc một câu lệnh, một số ngôn ngữ từ khoá phân biệt chữ hoa, chữ thường thì lại gõ chữ hoa, v.v. gọi là lỗi chính tả).

Chương trình sẽ không thể biên dịch được khi gặp lỗi này. Các trình biên dịch khi gặp lỗi ở dòng code nào thì sẽ báo lỗi. Việc tìm và sửa lỗi cú pháp rất đơn giản.

In [2]:

```
1 #Lỗi cú pháp
2 for i in range(1,e):
3     print('Biến :', i)
```

-----  
**NameError**

Traceback (most recent call last)

<ipython-input-2-e981b9106777> in <module>

1 #Lỗi cú pháp

----> 2 for i in range(1,e):

3 print('Biến :', i)

**NameError:** name 'e' is not defined

## (Tham khảo 2) Các loại lỗi thường gặp trong lập trình

**2) Lỗi thực thi (Runtime error):** Xảy ra bất ngờ khi chương trình đang chạy. Loại lỗi này thường xảy ra do người lập trình viết code ẩu, không lường hết các trường hợp xảy ra, khiến chương trình đang chạy thì bị lỗi **treo màn hình**, **thoát khỏi chương trình** hoặc thoát luôn chương trình, v.v. Lỗi này có thể dễ dàng phát hiện bằng cách Debug

Nên bổ xung *cấu trúc xử lý ngoại lệ* **try ... catch ... finally...** để hạn chế lỗi thực thi

```

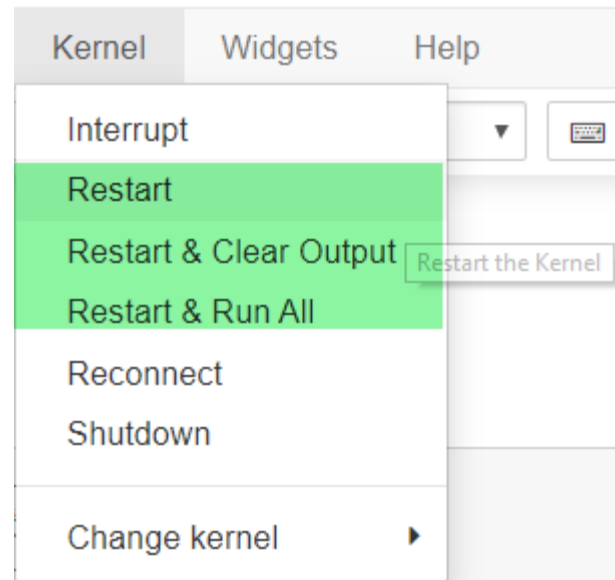
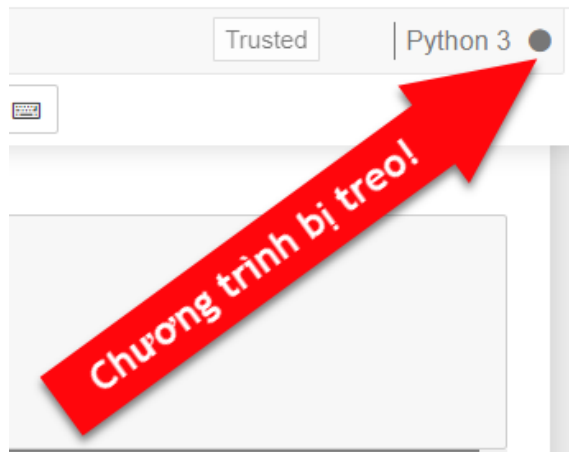
Edit View Insert Cell Kernel Widgets Help Trusted Python 3
[Icons] Run [Stop] [Refresh] [Next] Code
In [*]: 1 #Lỗi thực thi (Runtime Error)
        2 n=10
        3 while (n>0):
        4     #Vòng lặp vô hạn
        5     print(n)
        6
        10
        10
        10

```

Chương trình bị treo!

## (Tham khảo 2) Các loại lỗi thường gặp trong lập trình

- Trong trường hợp dấu tròn bên cạnh chữ Python 3 chuyển sang màu đen trong thời gian dài và không thể thực hiện được bất kỳ thao tác nào khác với cell, khi đó chương trình đang bị treo.
- Để thoát khỏi tình huống này có thể khởi động lại kernel (**restart the Kernel**)



## (Tham khảo 2) Các loại lỗi thường gặp trong lập trình

- **3) Lỗi thuật toán (Logical Error):** Đây là lỗi do tư duy sai, thuật toán sai dẫn đến sai kết quả
- Đây là loại lỗi **khó phát hiện nhất**, thường phải sử dụng chương trình, thậm chí là dùng lâu mới phát hiện ra được. Việc debug lỗi này cũng là một việc tốn khá nhiều thời gian và công sức. Để phát hiện lỗi này thì chương trình cần chạy nhiều lần với nhiều kết quả để xem nó có phù hợp hay không.

In [6]:

```
1 #Lỗi thuật toán (Logical Errors)
2 #Tìm quãng đường di chuyển của xe biết Vận tốc của xe: 60 km/h
3
4 thoigian = input('Nhập thời gian xe chạy (h):')
5 #Tính quãng đường
6 quangduong = thoigian*60
7
8 #Hiển thị kết quả
9 print('Quãng đường xe chạy được là: ', quangduong, ' (Km)')
10
```

## Chương trình chạy nhưng kết quả không đúng!

Nhập thời gian xe chạy (h):3

[illegible]



Thank You