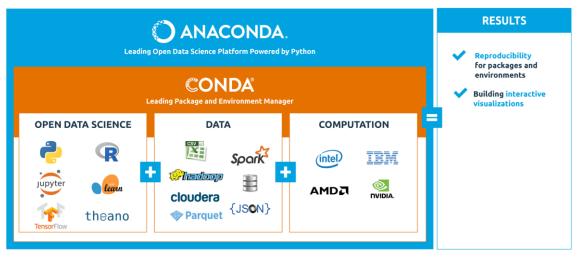


CHƯƠNG 2 Ngôn ngữ và công cụ lập trình trong khoa học dữ liệu

Nội dung chương 2

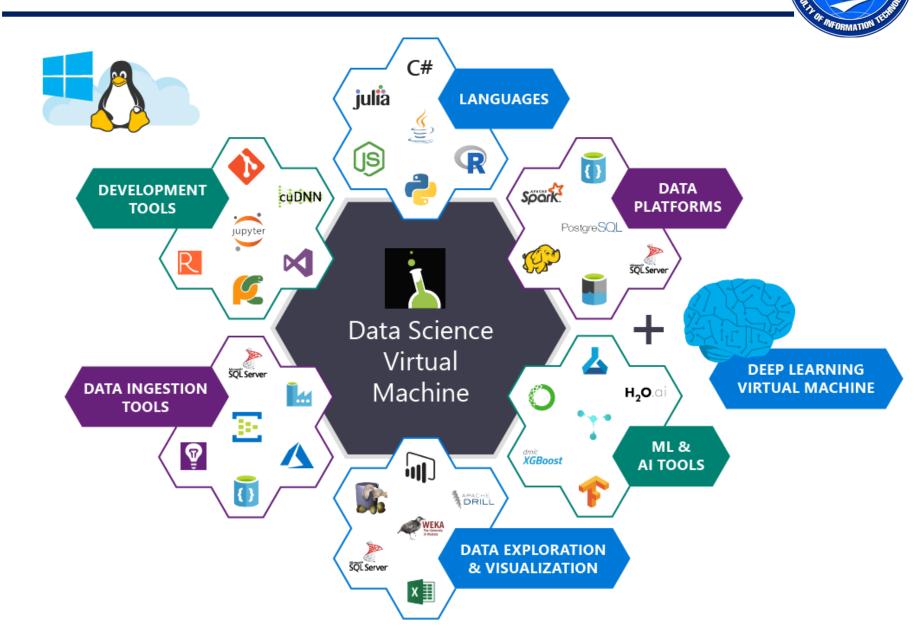


- 2.1 Giới thiệu một số ngôn ngữ và công cụ lập trình trong KHDL
- 2.2 Ngôn ngữ lập trình Python trong KHDL
- 2.3 Cài đặt và thiết lập môi trường lập trình
- 2.4 Xây dựng ứng dụng đầu tiên với python
- 2.5 Bài tập chương 2



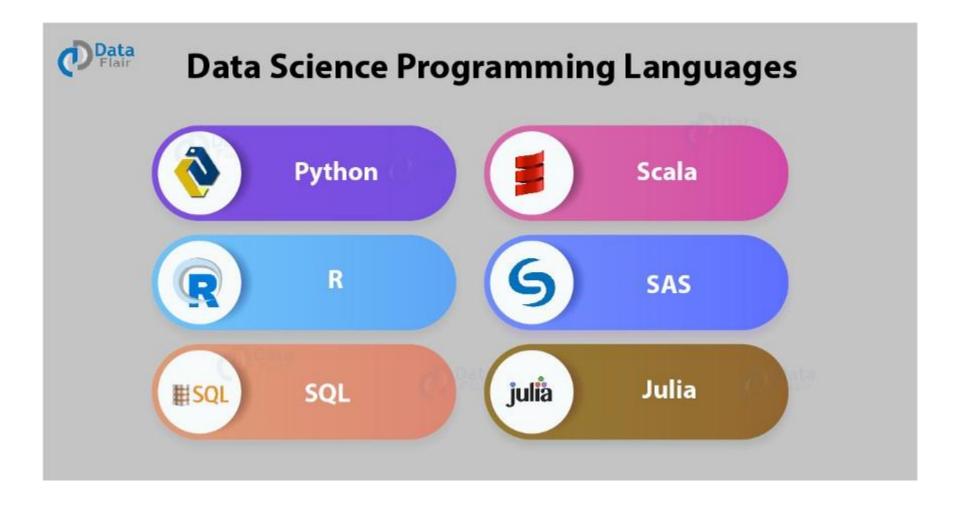
1. Giới thiệu một số ngôn ngữ và công

CÔNG NGHỆ THÔNG TIN



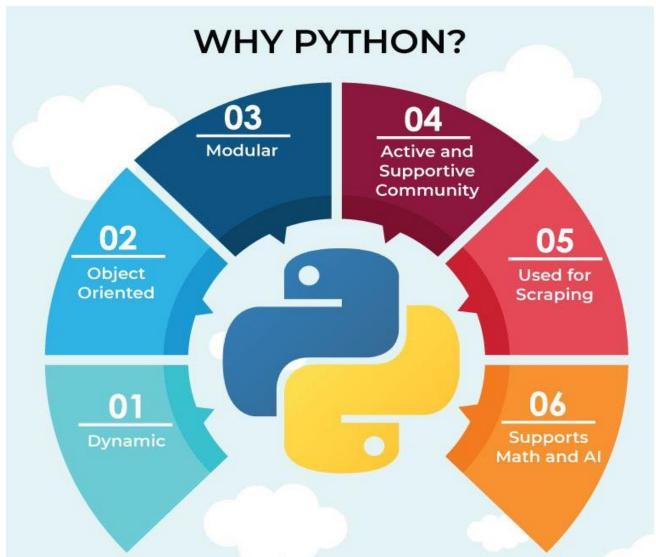
1.Giới thiệu một số ngôn ngữ và công





2. Ngôn ngữ lập trình Python

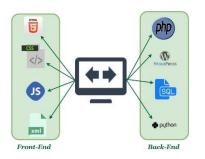




KHOA
CÔNG NGHỆ THÔNG TIN

KHOA
CÔNG NGHỆ THÔNG THÔNG THÔNG THÔNG THÀ
CÔNG NGHỆ THÔNG THÀ
CÔNG NGH THÀ
CÔNG NGH THÔNG THÀ
CÔNG NGH THÀ
CÔNG NGH THÔNG THÀ
CÔNG NGH THÔNG THÀ
CÔNG NGH
CÔNG NGH THÀ
CÔNG NGH TH

- Ngôn ngữ lập trình có thể sử dụng để phát triển từ các ứng dụng cho desktop, web đến các ứng dụng cho di động.
- Phù hợp nhất để phát triển các ứng dụng cho backend web development (server side), phân tích dữ liệu, trí thông minh nhân tạo (AI), máy học (machine learning), big data và hiển thị dữ liệu (data visualization).
- Nhiều lập trình viên sử dụng Python để phát triển game và các ứng dụng trên destop.
- Phù hợp cho việc xây dựng các script nhỏ để tự đống hóa các công việc đơn giản, ví dụ như viết 1 đoạn script để đếm số lượng email có chứa các từ khóa cho trước.











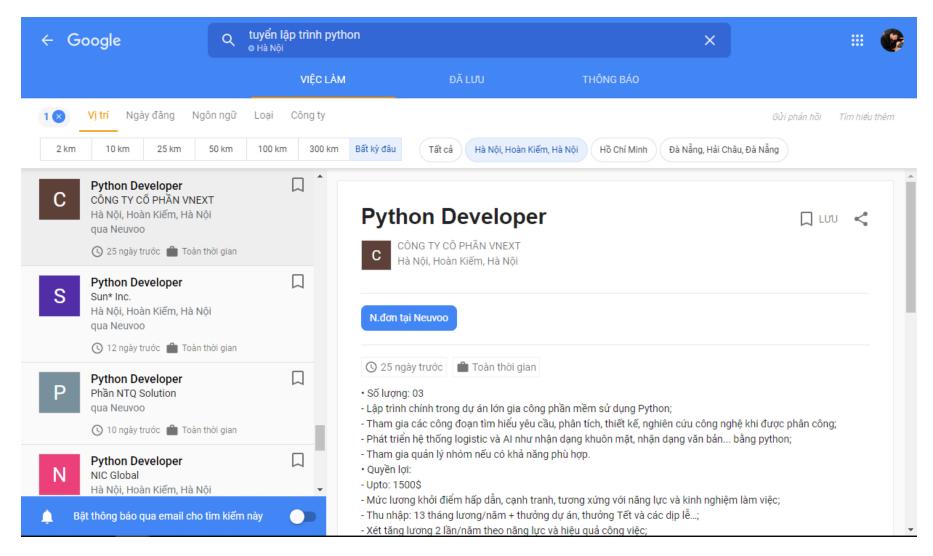


Easy	01 Extensible	07
Expressive 0	2 Embeddable	08 Features of
Free and Open Source	Interpreted	09
High-Level 04	Large Standard Library	10
Portable 05	GUI Programming	11 Python
Object Oriented 06	Dynamically Typed	12









2. Môi trường phát triển tích hợp (IDEs) cho Python





2. Môi trường phát triển tích hợp (IDEs) cho Python



+ IDEs Offline: Sử dụng bộ công cụ Anacoda

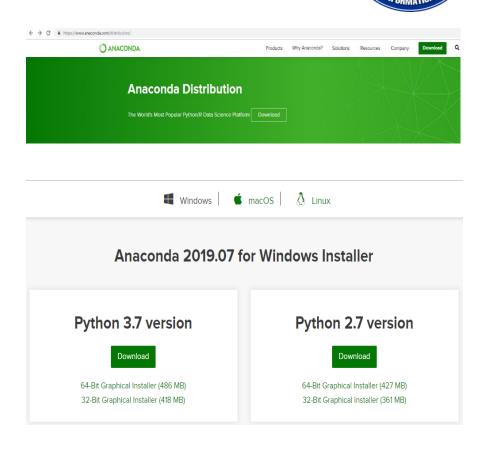


+ IDEs Online: Sử dụng Colab của Google



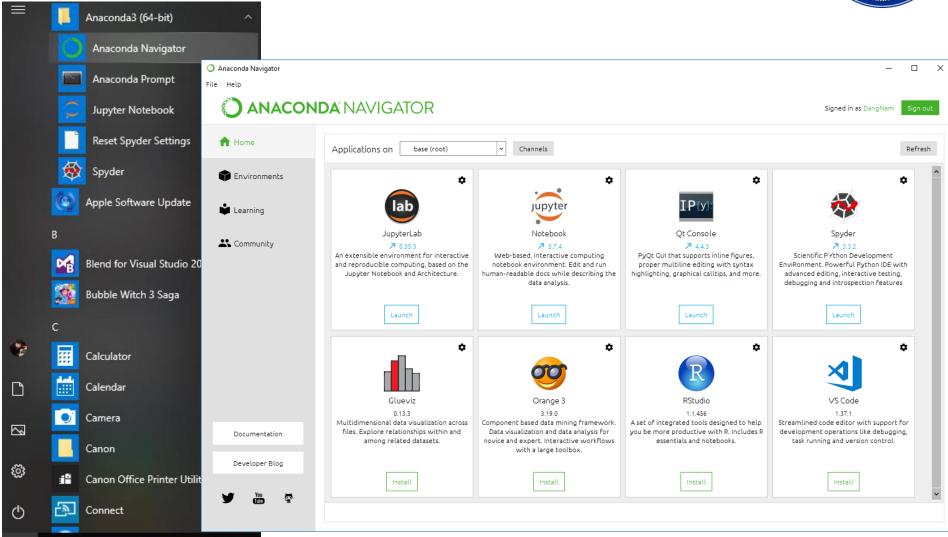
3.1 Sử dụng Anaconda

- 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.co m/distribution/#downloadsection -> download



Cài đặt Anaconda





Link hướng dẫn cài đặt Anaconda và sử dụng Jupyter!



JUPYTER NOTEBOOK



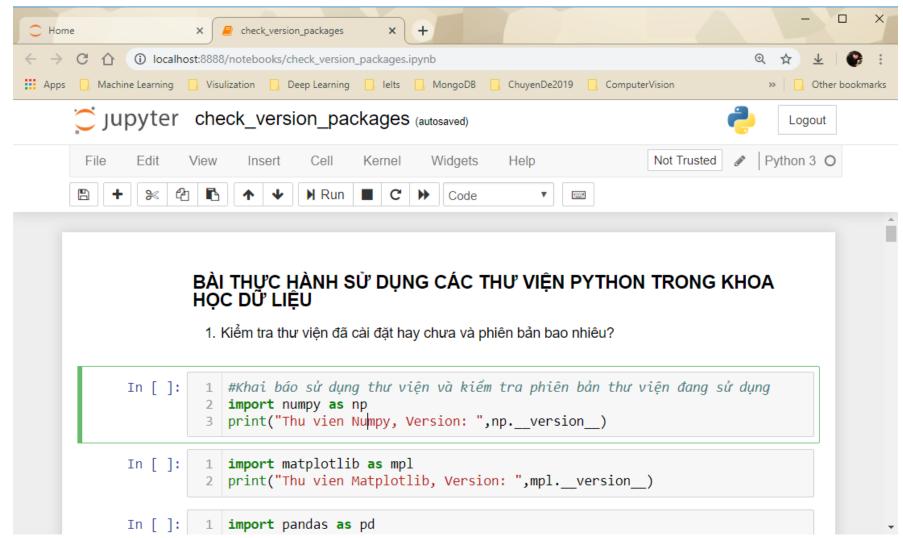
Jupyter notebook là môi trường chạy code python ở dạng interactive.

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

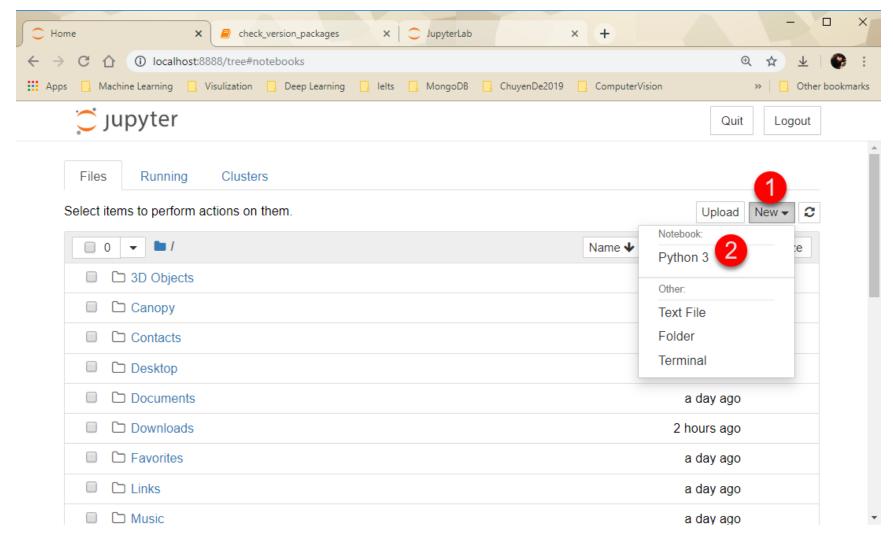
Điều này sẽ giúp ích rất nhiều cho lập trình viên, đặc biệt cho những người mới học Python.

Giao diện cơ bản của Jupyter Notbook



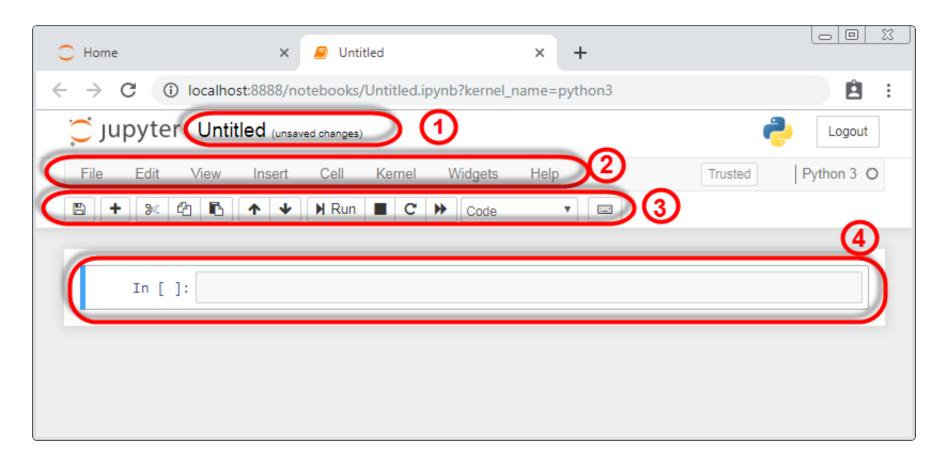


Tạo file Python mới với Jupyter Notebo

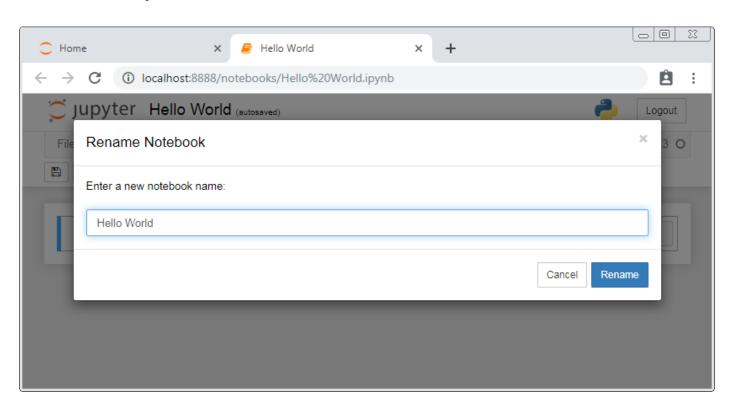


Sử dụng công cụ Jupyter notebook





- 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.



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...

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





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))
```

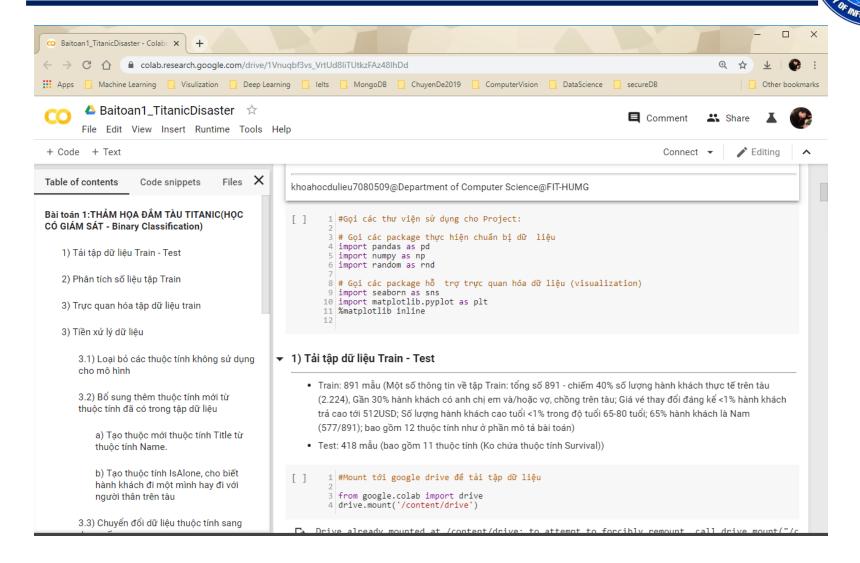
3.2 Sử dụng Google Colab

- 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 kĩ 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



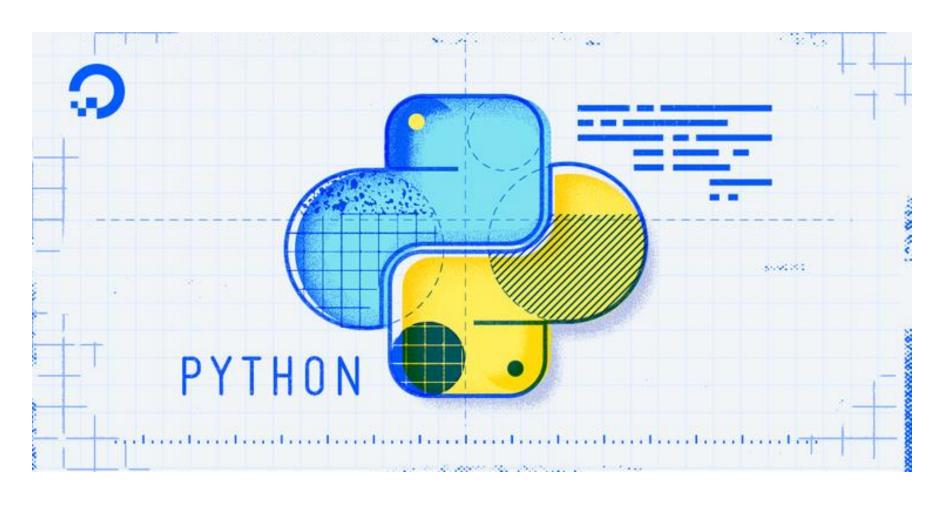


3.2 Sử dụng Google colab

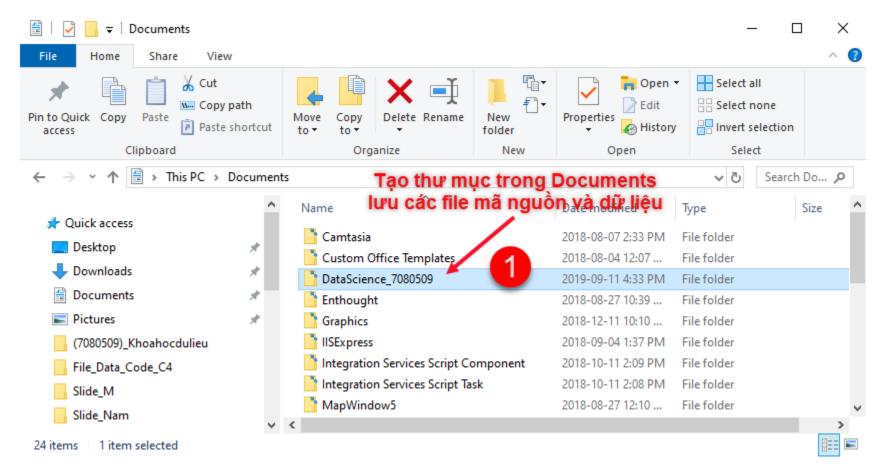


4. "Hello world" với Python

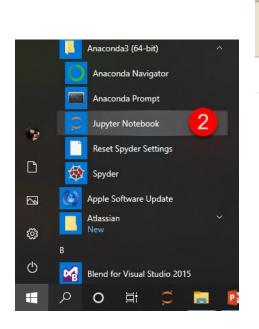


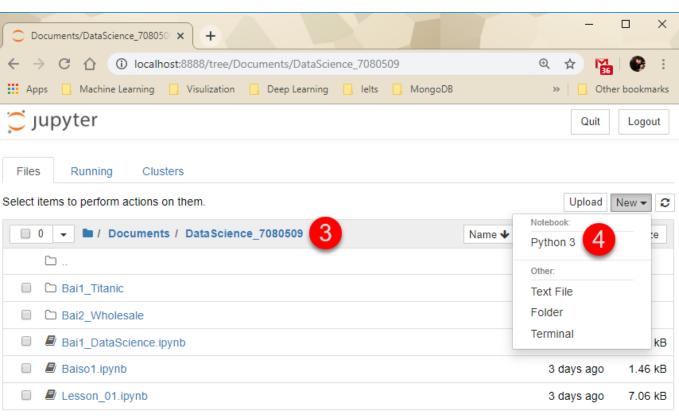


 Bước 1: Tạo thư mục 7080509_DataScience trong Document để lưu trữ toàn bộ file mã nguồn và dữ liệu.



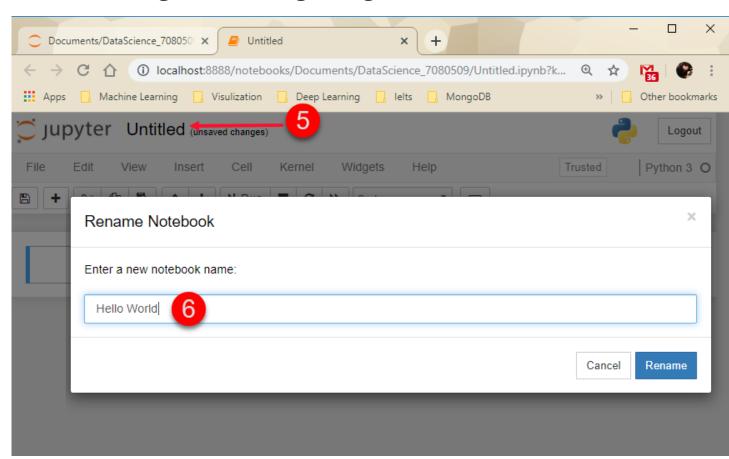
- Bước 2: Chạy Jupyter Notebook
- Bước 3: Truy cập tới thư mục DataScience_7080509 đã tạo
- Bước 4: Chọn New/Python3 để tạo file code mới



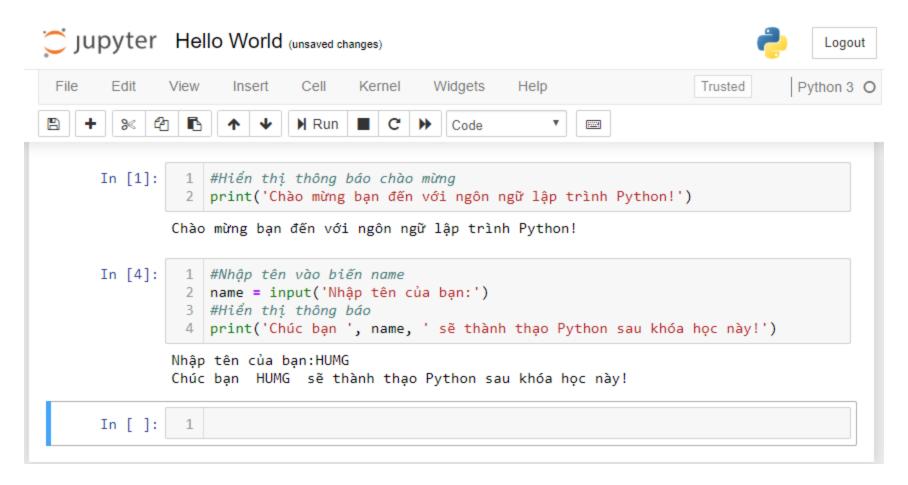


- Bước 5: Tab giao diện code mới mở ra, kích vào tên mặc định Untitled để đặt lại tên cho file code.
- **Bước 6:** Nhập tên mới cho file

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



Hãy nhập và chạy những dòng code Python đầu tiên!

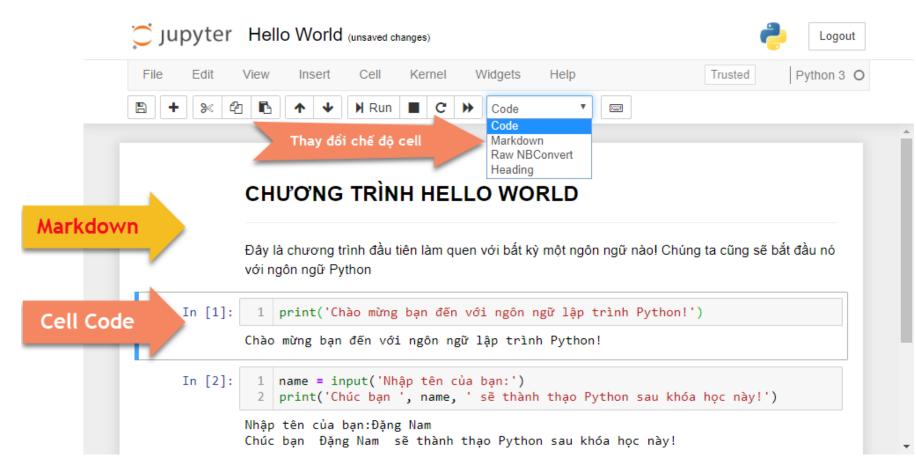


2 loại cell căn bản: Markdown – Code

KHOA
CÔNG NGHỆ THÔNG TIN

KHOA
CÔNG NGHỆ THÔNG THƠN THỐNG THỐNG THÔNG THÔNG THÔNG THỐNG THỐNG

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



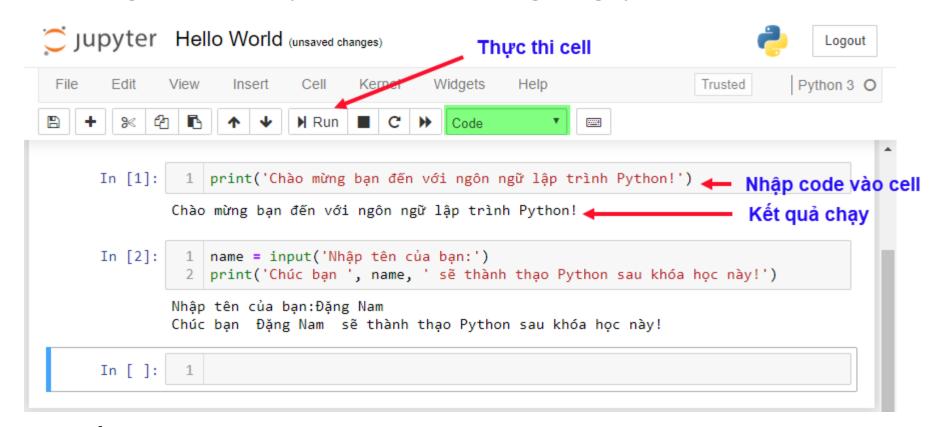
2 loại cell căn bản: Markdown – Code

KHOA
CÔNG NGHỆ THÔNG TIN

KHOA
CÔNG NGHỆ THÔNG THÔNG TIN

KHOA
CÔNG NGHỆ THÔNG THỐNG THÔNG T

 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.



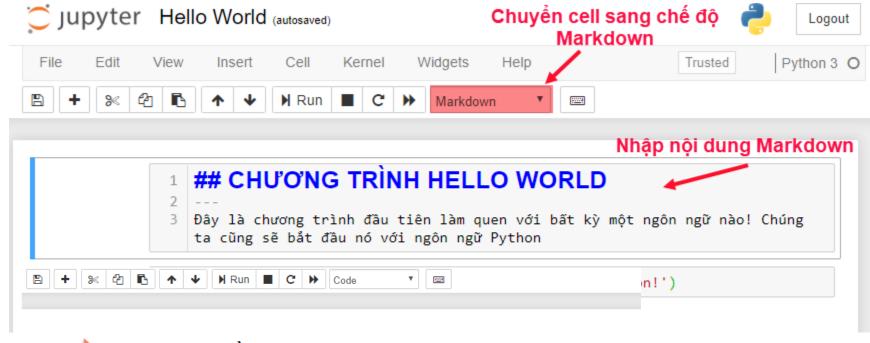
2 loại cell căn bản: Markdown – Code

KHOA

CONG NGHE THÔNG TIN

THE TAY MATORIMATION TELEMON

 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 chay 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

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

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")
                                          In [1]: print('Tôi yêu HUMG')
> print (biến)
                                                  Tôi yêu HUMG
Print('ký tự', biến)
                                          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
                                                  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 chương 2 - 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
        3 A = 0
        4 for i in range(1,6):
         5 print(i)
           a = a + i
        8 #Hiển thi kết quả biến a
           print('Giá trị của biến a = ',a)
           print('----')
        11
        12 for i in range (1,6,2):
        13 print(i)
           A = A + i
        14
        15
        16 #HIển thị kết quả biến A
           print('Giá trị của biến A = ',A)
        18
```

Bài tập chương 2 - 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:

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

(Tham khảo:1) So sánh code C – Java - Python



"Hello, World"

```
#include <stdio.h>
 int main(int argc, char ** argv)
   printf("Hello, World!\n");
ava
 public class Hello
    public static void main(String argv[])
      System.out.println("Hello, World!");
```

now in Python print "Hello, World!"

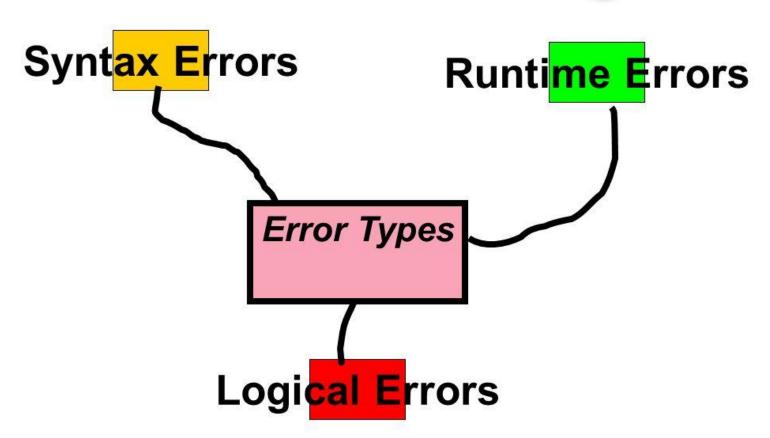
Trong các ngôn ngữ lập trình nói chung, dù là một lập trình viên chuyên nghiệp hay mới chập chững vào nghề đều có thể gặp lỗi trong quá trình lập trình.

Tuỳ 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.





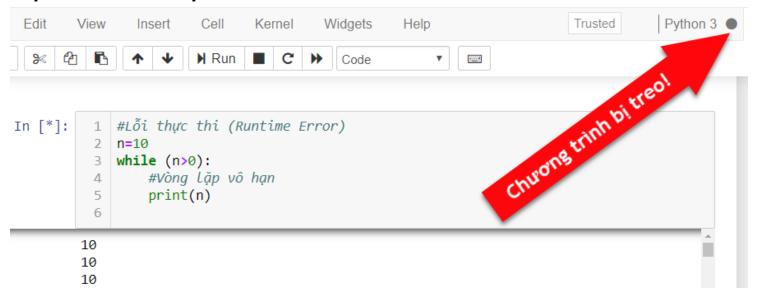


1) Lỗi cú pháp (Syntax Error): Đây là loại lỗi sơ đẳng nhất trong lập trình. Thường là do gõ sai cấu trúc của ngôn ngữ (ví dụ như thiếu dấu kết thúc một câu lệnh, một số ngôn ngữ từ khoá phân biệt chữ hoa, chữ thường thì lại gõ chữ hoa, v.v. gọi là lỗi chính tả).

Chương trình sẽ không thể biên địch 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.

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

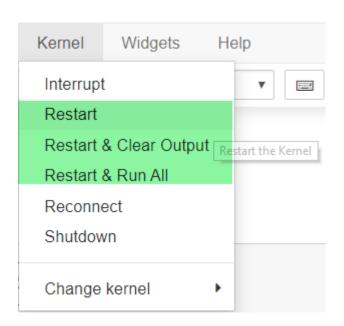
Nên bổ xung *cấu trúc xử lý ngoại lệ* **try ... catch ... finally...** để hạn chế lỗi thực thi



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)





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]:
             #Lỗi thuật toán (Logical Errors)
             #Tìm quãng đường di chuyển của xe biết Vân tốc của xe: 60 km/h
             thoigian = input('Nhập thời gian xe chạy (h):')
             #Tính quãng đường
                                                                    Chương trình chay nhưng
             quangduong = thoigian*60
                                                                      kết quả không đúng!
             #Hiển thi kết quả
             print('Quang drong xe chay droc la: ', quangduong, ' (Km)')
         10
        Nhập thời gian xe chạy (h):3
        Quãng đường xe chạy được là:
```

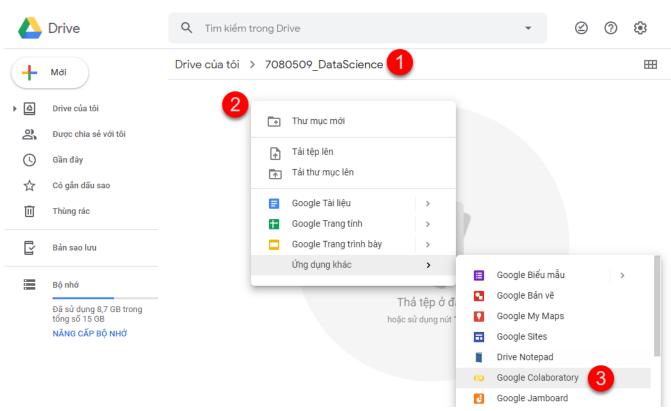
33333333333 (Km)

(Tham khảo 3) hướng dẫn xây dựng chương trình đầu tiên trong Google colab

- Truy cập vào Google Drive của bạn để bắt đầu!
- Bước 1: Tạo thư mục 7080509_DataScience để lưu trữ toàn bộ file mã nguồn và dữ liệu.

 Bước 2: Kích chuột phải và thư mực vừa tạo chọn "Ứng dụng khác"

• Bước 3: Chọn Google Colaboratory để thực hiện viết chương trình đầu tiên!



(Tham khảo 3) hướng dẫn xây dựng chương trình đầu tiên trong 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

