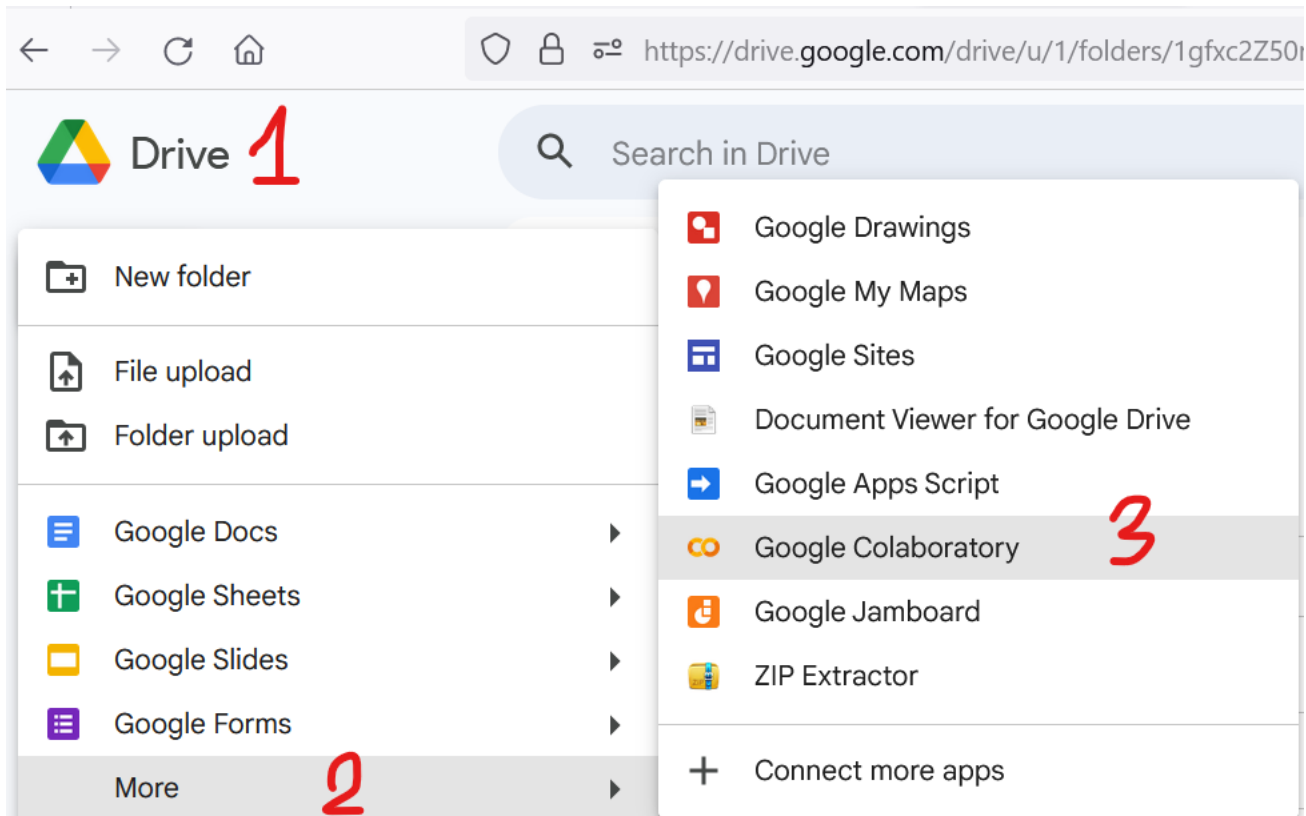




THỰC NGHIỆM SỬ DỤNG CÔNG CỤ

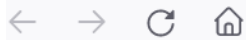


SỬ DỤNG PYTHON TRÊN GOOGLE COLAB





SỬ DỤNG PYTHON TRÊN GOOGLE COLAB



https://colab.research.google.com/drive/1SoJE2D9Ph

This notebook is open with private outputs. Outputs will n



Untitled1.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved



+ Code + Text



Start coding or generate with AI.



- **Kiểu số:** có 3 kiểu số trong python là int, float, complex.
- Int, hoặc integer, là một số nguyên, dương hoặc âm, không có số thập phân, có độ dài không giới hạn.



```
a = 1  
b = 1105200006022000  
c = -12345
```

- Float, hoặc "số dấu phẩy động" là một số, dương hoặc âm, chứa một hoặc nhiều số thập phân.



```
a = 1.23  
b = 2.0  
c = -1.23
```

- Complex Complex (số phức) được viết với "j" là phần ảo



```
a = 3 +5j;  
b = 5j;  
c = -5j
```

- **Kiểu dữ liệu Chuỗi:** Chuỗi là một dãy các ký tự. Python hỗ trợ các ký tự Unicode. Nói chung, các chuỗi được biểu thị bằng dấu ngoặc đơn hoặc dấu ngoặc kép.



```
a = "Hello World" # Cũng giống như  
a = 'Hello World'
```

- Chuỗi là mảng các ký tự, python không có kiểu dữ liệu ký tự, một ký tự đơn giản chỉ là một chuỗi có độ dài là 1. Sử dụng dấu [] để truy cập các phần tử của chuỗi.



0s



```
a = "Stringee"  
print(a[1])
```

t

- **Kiểu dữ liệu boolean:** Cho kết quả là True hoặc False



```
print(1 > 2) # False  
print(1 == 2) # False  
print(1 < 2) # True
```

```
False  
False  
True
```

- **Kiểu dữ liệu danh sách(list):**
 - Danh sách là một kiểu dữ liệu linh hoạt độc quyền trong Python.
 - Giống như mảng trong các ngôn ngữ lập trình khác.
 - Nhưng nó có thể chứa đồng thời nhiều loại dữ liệu khác nhau.



```
list = [1, 3.5, "Hello"]  
print(list) # In ra list  
print(list[2]) # In ra "Hello"
```

```
[1, 3.5, 'Hello']  
Hello
```


- **Kiểu dữ liệu Tuple:**

- Tuple là một kiểu dữ liệu được sử dụng để lưu trữ các đối tượng (chẳng hạn như hằng số) sẽ không bị sửa đổi sau này.
- Phần lưu trữ còn lại rất giống với kiểu dữ liệu danh sách (list)
- tuple được viết bằng dấu ngoặc tròn:



```
thistuple = ("apple", "banana", "cherry") # Cách khai báo tương tự với list  
print(thistuple)
```

```
('apple', 'banana', 'cherry')
```

- **Kiểu dữ liệu Tuple:**

- Tạo Tuple với một mục: Để tạo một bộ dữ liệu tuple chỉ có một phần tử, ta phải thêm dấu phẩy sau phần tử đó, nếu không Python sẽ không nhận ra nó là một tuple.



```
# Đây là cách khởi tạo đúng  
thistuple = ("apple",)  
print(type(thistuple))  
# Đây là cách khởi tạo sai  
thistuple = ("apple")  
print(type(thistuple))
```

```
<class 'tuple'>  
<class 'str'>
```

- **Kiểu dữ liệu Tuple:**

- Có thể dùng hàm tuple() để tạo một tuple

```
▶ thistuple = tuple(("apple", "banana", "cherry")) # lưu ý 2 cặp dấu ngoặc tròn  
print(thistuple)
```

```
('apple', 'banana', 'cherry')
```

- **Kiểu dữ liệu Dictionaries**

- Dictionaries là kiểu dữ liệu dạng bảng, các phần tử được sắp xếp dựa trên khóa (key) thay vì chỉ số như trong danh sách (list).
- Python dictionaries bao gồm: các cặp khóa-giá trị, trong đó mỗi khóa là duy nhất và không thay đổi, và tương ứng với mỗi khóa là một giá trị có thể thay đổi.
- Các khóa trong từ điển là không có thứ tự, có nghĩa là việc truy cập các phần tử của từ điển không được thực hiện dựa trên chỉ số như trong danh sách mà dựa trên khóa của chúng.

- **Kiểu dữ liệu Dictionaries**

- Để tạo một Python dictionaries, ta có thể sử dụng cú pháp dưới đây:

```
my_dict = {"key1": value1, "key2": value2, ...}
```

- Cũng có thể sử dụng hàm tạo dict() để tạo dictionary mới:



```
pairs = [("key1", "value1"), ("key2", "value2")]  
my_dict = dict(pairs)
```

- **Python sets:**

- Python sets là một kiểu dữ liệu được sử dụng để lưu trữ các phần tử không có thứ tự và duy nhất.
- Sets là một tập hợp không chứa các phần tử trùng lặp và không có thứ tự cụ thể.
- Sets trong Python được định nghĩa bằng cặp ngoặc nhọn {} hoặc hàm set().



```
my_set = {"element1", "element2", "element3", "..."}  
my_set = {1, 2, 3, 4, 5}
```

- Toán tử số học

Toán tử	Mô tả
+	Toán tử cộng 2 giá trị
-	Toán tử trừ 2 giá trị
*	Toán tử nhân 2 giá trị
/	Toán tử chia 2 giá trị (chia ra số thập phân)
%	Toán tử chia 2 giá trị lấy phần dư
//	Toán tử chia 2 giá trị, làm tròn xuống
**	Toán tử mũ

- Toán tử quan hệ

Toán tử	Mô tả
$==$	So sánh giá trị của các đối số xem có bằng nhau hay không
$!=$	So sánh giá trị của các đối số xem có khác nhau hay không.
$<$	Dấu $<$ đại diện cho phép toán nhỏ hơn
$>$	Dấu $>$ đại diện cho phép toán lớn hơn
$>=$	Dấu $>$ đại diện cho phép toán nhỏ hơn hoặc bằng
$<=$	Dấu $>$ đại diện cho phép toán lớn hơn hoặc bằng

- Toán tử gán

Toán tử	Mô tả
=	Toán tử gán cho 1 biến khác
+=	Toán tử này cộng rồi gán lại cho biến đó
-=	Toán tử này trừ rồi gán lại cho biến đó
*=	Toán tử này nhân rồi gán lại cho biến đó
/=	Toán tử này chia rồi gán lại cho biến đó
%=	Toán tử này chia lấy dư rồi gán lại cho biến đó
**=	Toán tử này tính mũ rồi gán lại cho biến đó
//=	Toán tử này chia làm tròn xuống rồi gán lại cho biến đó

- Toán tử logic

Toán tử	Mô tả
and	Giống toán tử &&, đúng nếu 2 vế của and là True, còn lại là False
or	Giống toán tử ??, đúng nếu 1 trong 2 vế của or là True, còn lại là False
not	Giống toán tử !, dạng phủ định
in	Nếu 1 đối số thuộc 1 tập nào đó thì trả về là True, ngược lại là False
not in	Ngược lại của in
is	Toán tử này trả về nếu 2 vế của toán tử: $a == b$
not is	Ngược lại của is