

# **Buổi 2: Công cụ & ngôn ngữ Python trong Khoa học dữ liệu #1**

---

1. Công cụ Khoa học dữ liệu
2. Python trong Khoa học dữ liệu #1
3. Hướng dẫn làm game Lô Đề Học
4. Tổng kết

# 1. CÔNG CỤ KHOA HỌC DỮ LIỆU

Ngôn ngữ chính:



Ưu điểm của Python:

- Ít cú pháp.
- Nhiều thư viện mạnh mẽ.



## Thư viện của Python:



- **Pandas** : cung cấp các cấu trúc dữ liệu và các phép toán để thao tác với các bảng số và chuỗi thời gian.



- **NumPy** : xử lý các mảng và ma trận lớn, đa chiều với các hàm toán học cao cấp.



- **Matplotlib** : trực quan hoá dữ liệu, tạo biểu đồ, đồ thị.



- **Scikit-learn** : thư viện học máy (classification, regression, clustering, ...).



- **PyTorch** : thư viện chuyên sâu, xây dựng mô hình cho Deep Learning.

# 1. CÔNG CỤ KHOA HỌC DỮ LIỆU / SQL

## Ưu điểm của SQL:

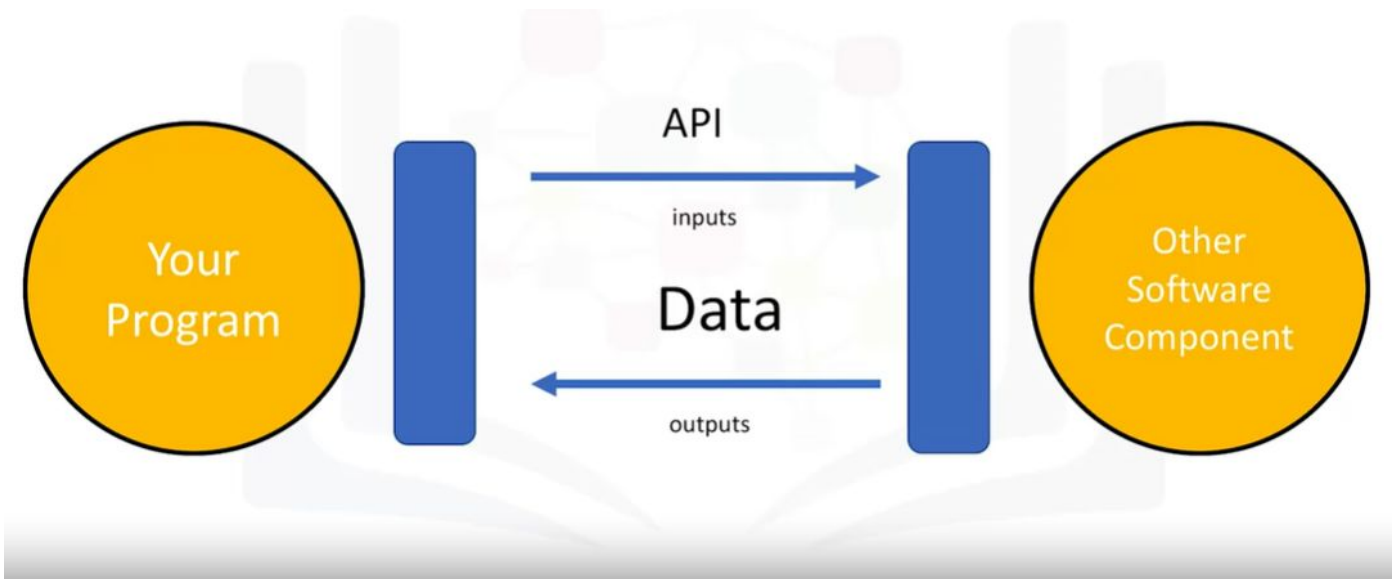
- Ngôn ngữ xử lý dữ liệu có cấu trúc.
- Cần thiết cho việc lưu trữ và phân tích dữ liệu.
- Truy cập dữ liệu hiệu quả.



# 1. CÔNG CỤ KHOA HỌC DỮ LIỆU / API

**API Interface** : Cho phép các phần mềm giao tiếp với nhau.

VD: Pandas là tập hợp các thành phần phần mềm được kết hợp lại với nhau, Pandas không chỉ được viết bởi Python. Nhưng khi cần sử dụng, bạn chỉ cần gọi API của Pandas để tương tác với dữ liệu và thực thi yêu cầu của bạn.



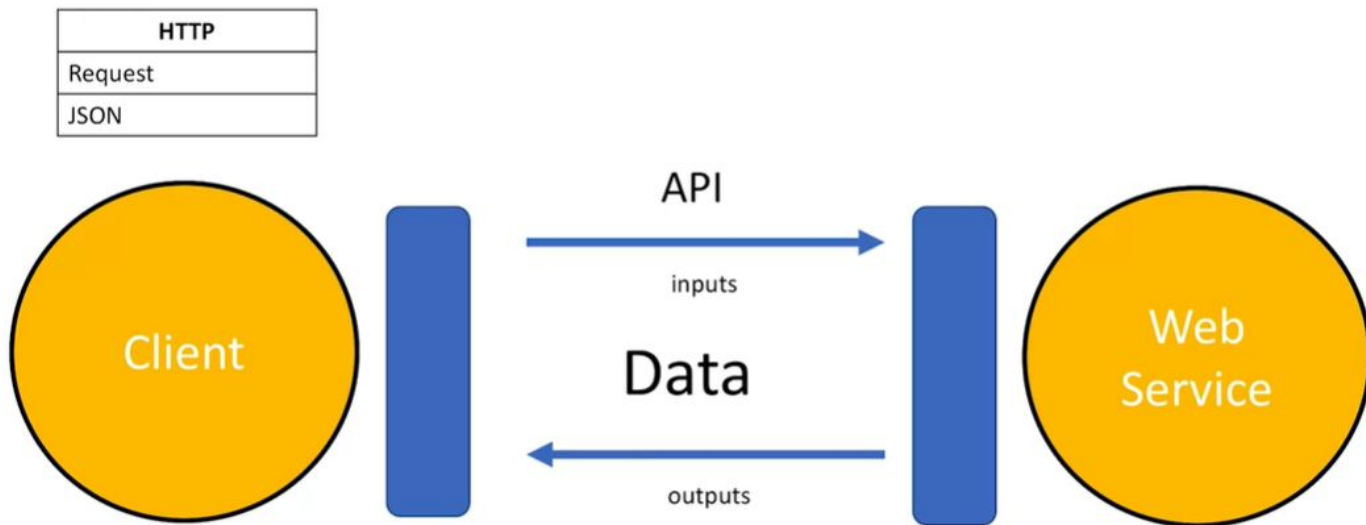
# 1. CÔNG CỤ KHOA HỌC DỮ LIỆU / API

**Rest API** : Cho phép các phần mềm giao tiếp với nhau thông qua Internet.

VD: Watson Speech to Text API là ví dụ về REST API, chuyển đổi lời nói thành văn bản.

Trong lệnh gọi API, bạn gửi tệp âm thanh qua Internet tới dịch vụ web của Watson (Request).

API trả lại văn bản chuyển đổi từ âm thanh (Response).



# 1. CÔNG CỤ KHOA HỌC DỮ LIỆU / LOẠI DỮ LIỆU

**Dữ liệu có cấu trúc:** Loại dữ liệu được tổ chức rõ ràng (như chia thành các hàng và cột).

VD: các bảng Excel, các bảng trong cơ sở dữ liệu SQL, ...

	A	B	C
1	Sản phẩm	Quý 1	Quý 2
2	Sô cô la	\$744,60	\$162,56
3	Keo dẻo	\$5.079,60	\$1.249,20
4	Bánh mì dài Scotland	\$1.267,50	\$1.062,50
5	Bánh Sir Rodney's Scones	\$1.418,00	\$756,00
6	Bánh Tarte au sucre	\$4.728,00	\$4.547,92
7	Bánh quy Sô cô la	\$943,89	\$349,60
8	Tổng cộng	\$14.181,59	\$8.127,78

Cust_No	Cust_Name	Phone No
002	David Gordon	0231-5466356
003	Prince Fernandes	0221-5762382
003	Charles Yale	0321-8734723
002	Ryan Ford	0241-2343444
005	Bruce Smith	0241-8472198

Customer

Item_No	Description	Price
HW1	Power Supply	4000
HW2	Keyboard	2000
HW3	Mouse	800
SW1	Office Suite	15000
SW2	Payroll Software	8000

Items

Ord_No	Item_No	Qty
101	HW3	50
101	SW1	150
102	HW2	10
103	HW3	50
104	HW2	25
104	HW3	100
105	SW1	100

Order\_Details

Ord_No	Ord_Date	Cust_No
101	02-08-12	002
102	11-08-12	003
103	21-08-12	003
104	28-08-12	002
105	30-08-12	005

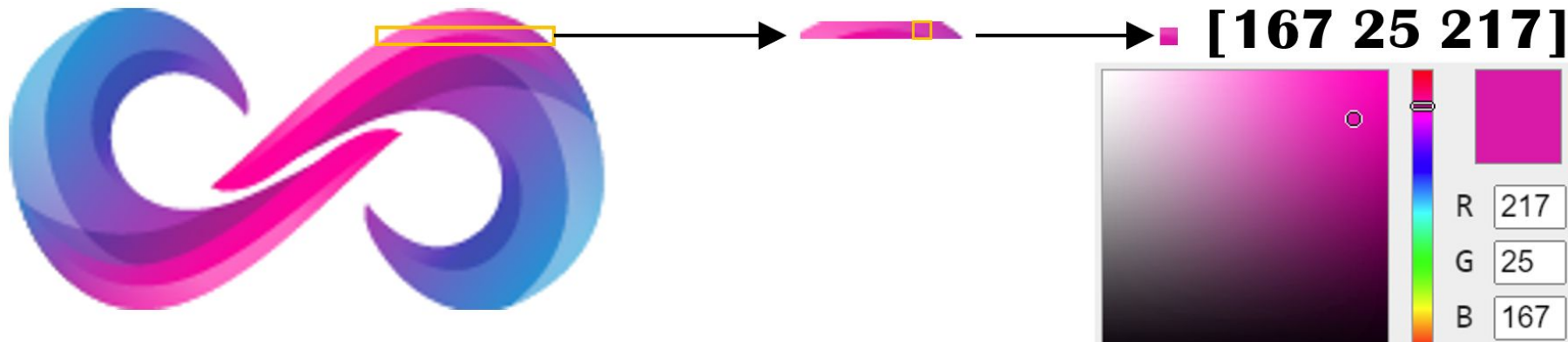
Order\_August



# 1. CÔNG CỤ KHOA HỌC DỮ LIỆU / LOẠI DỮ LIỆU

**Dữ liệu phi cấu trúc:** Loại dữ liệu không thể tổ chức, sắp xếp theo một cách định sẵn.

VD: văn bản, video, hình ảnh, dữ liệu âm thanh, ...



# 1. CÔNG CỤ KHOA HỌC DỮ LIỆU / LOẠI DỮ LIỆU

**Dữ liệu bán cấu trúc:** Loại dữ liệu không tuân theo một cấu trúc cụ thể nhưng vẫn chứa một số cấu trúc mô tả, cho phép tổ chức và lưu trữ một cách linh hoạt.

VD: JSON, XML, tệp nén, ...

JSON ▾

```
{
  "id": "12345",
  "created_at": "2024-03-21T12:34:56Z",
  "text": "Hello, Meta Mind!",
  "likes": 10000,
  "shares": 9999,
}
```

XML ▾

```
<status>
  <group name="30 BUỔI HỌC VỚI TEAM NGUYỄN HẢI ĐĂNG">
    <author>Nguyễn Hải Đăng</author>
    <date>Thứ Hai, 18 Tháng 3, 2024 lúc 09:01</date>
    <description>Chào mọi người, Em là Đăng...</description>
    <likes>1000</likes>
    <comments>999</comments>
  </group>
</status>
```

### Comment

#### What?

- Một đoạn văn bản, code bắt đầu bằng #, hoặc nằm trong “...”.
- Không được thực thi khi chương trình chạy.

#### Why?

- Mô tả về một khối mã.
- Ghi chú thêm thông tin.
- Tắt một dòng code.

### Comment

```
# Đây là comment 1 dòng
name = "Dang"
...

Đây là comment nhiều dòng
...

print(f"Hello {name}")
```

Hello Dang

```
# Đây là ví dụ về comment
name = "Dang"
age = 18
...

print(f"Hello {age}")
...

print(f"Hello {name} {age} tuoi")
```

Hello Dang 18 tuoi

### Data types: Kiểu dữ liệu

**What?** Các biến, hằng số và các giá trị cụ thể có một "kiểu dữ liệu".

- **Integer:**
  - Kí hiệu: int
  - Kiểu số nguyên không giới hạn.
- **Float:**
  - Kí hiệu: float
  - Kiểu dữ liệu số thực.
- **String:**
  - Kí hiệu: str
  - Kiểu dữ liệu chuỗi, là một chuỗi ký tự Unicode.
- **Boolean:**
  - Kí hiệu: bool
  - Kiểu dữ liệu đại diện cho các giá trị logic.
  - Có 2 giá trị: True, False.

---

Examples	
<hr/>	
Integer	0, 12, 5, -5, 10000000....
Float	4.5, 3.99, 0.1
String	"Hi", "Hello", "Hi there!"
Boolean	True, False

---

### Variables: Biến trong Python

- Biến lưu trữ dữ liệu có thể thay đổi.
- Lưu trữ số, chuỗi, và các loại dữ liệu khác.
- Không cần khai báo kiểu dữ liệu trước khi dùng.

#### In biến

```
str = "Hello World"  
print(str)
```

Hello World

### String: Chuỗi trong Python

- Lưu và xử lý các ký tự.
- Đặt trong dấu nháy đơn ' hoặc nháy kép ".
- Cung cấp nhiều phương thức xử lý chuỗi.

#### In chuỗi

```
print("Hello World")
```

Hello World

#### In chuỗi + biến

```
name = "Dang"  
print("Hello", name)
```

Hello Dang

```
name = "Dang"  
print(f"Hello {name}")
```

Hello Dang

### Global variable: Biến toàn cục

- Các biến được tạo bên ngoài hàm được gọi là biến toàn cục.
- Có thể sử dụng ở cả bên trong và ngoài hàm.

```
▶ # Global variable
name = "Đăng"

def show():
    print(name)

# Trong hàm
show()
# Ngoài hàm
print(name)
```



Đăng  
Đăng

***"global"***

Sử dụng biến  
toàn cục

```
▶ name = "Đăng"

def show():
    global name
    name = "Vẫn là Đăng"
    print(name)

show()
print(name)
```



Vẫn là Đăng  
Vẫn là Đăng

### Local variable: Biến cục bộ

- Các biến bên trong hàm.
- Chỉ có thể sử dụng bên trong hàm đó.

```
▶ # Global variable
name = "Đăng"

def show():
    # Local variable
    name = "Khôi"
    print(name)

show()
print(name)
```



Khôi  
Đăng

## 2. PYTHON TRONG KHOA HỌC DỮ LIỆU #1

### Operations: Toán tử

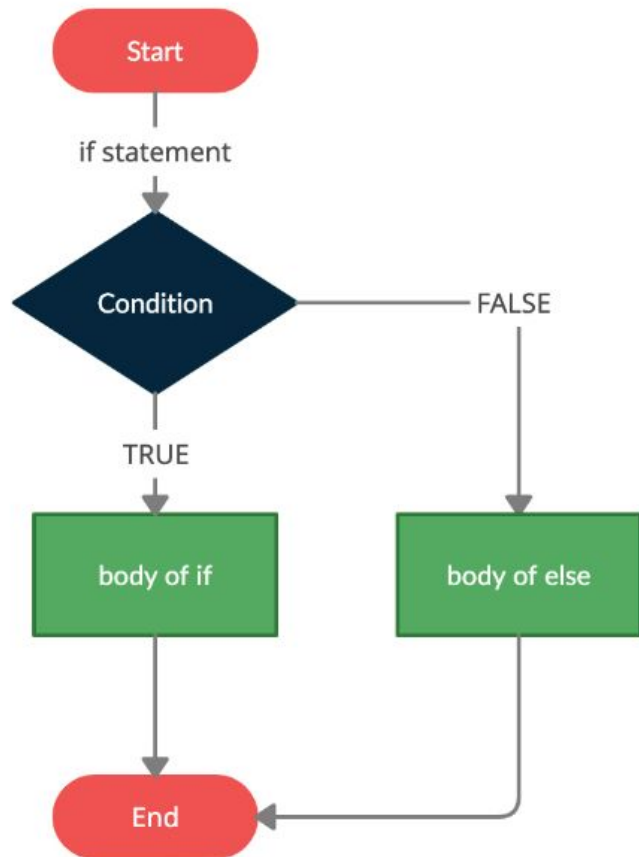
Số học	Ví dụ
Phép cộng (+)	$x+y$
Phép trừ (-)	$x-y$
Phép nhân (*)	$x*y$
Phép chia (/)	$x/y$
Phép chia lấy phần dư (%)	$x\%y$
Toán tử mũ(**)	$x**y$
Chia làm tròn xuống (//)	$x//y$

Quan hệ	Ví dụ
Bằng (==)	$x==y$
Khác (!=)	$x!=y$
Khác (<>)	$x<>y$
Lớn hơn (>)	$x>y$
Nhỏ hơn (<)	$x<y$
Lớn hơn hoặc bằng (>=)	$x>=y$
Nhỏ hơn hoặc bằng (<=)	$x<=y$

Logic	Ví dụ
Trả về True nếu cả hai vế toán hạng đúng (and)	$x<5$ and $x<10$
Trả về True nếu một trong hai toán hạng đúng (or)	$x<5$ or $x<4$
Đảo ngược kết quả, nếu kết quả True thì sẽ trả về False (not)	not( $x<5$ and $x<10$ )

Gán	Ví dụ	Gán	Ví dụ
=	$x=5$ same as $x = 5$	&=	$x\&=3$ same as $x = x\&3$
+=	$x+=3$ same as $x = x+3$	=	$x =3$ same as $x = x 3$
-=	$x-=3$ same as $x = x-3$	^=	$x^-=3$ same as $x = x^3$
*=	$x*=3$ same as $x = x*3$	>>=	$x>>=3$ same as $x = x>>3$
/=	$x/=3$ same as $x = x/3$	<<=	$x<<=3$ same as $x = x<<3$
%=	$x\%=3$ same as $x = x\%3$	//=	$x//=3$ same as $x = x//3$
**=	$x**=3$ same as $x = x**3$		

## 2. PYTHON TRONG KHOA HỌC DỮ LIỆU #1



```
x = 10

# Check if x is even or odd
if x % 2 == 0:
    print("x is even")
else:
    print("x is odd")
```

if...else thường

```
1 def maximum(a,b):
2     return a if a > b else b
```

```
1 def check(num):
2     if(num % 2 == 0):
3         return "Even"
4     else:
5         return "Odd"
```

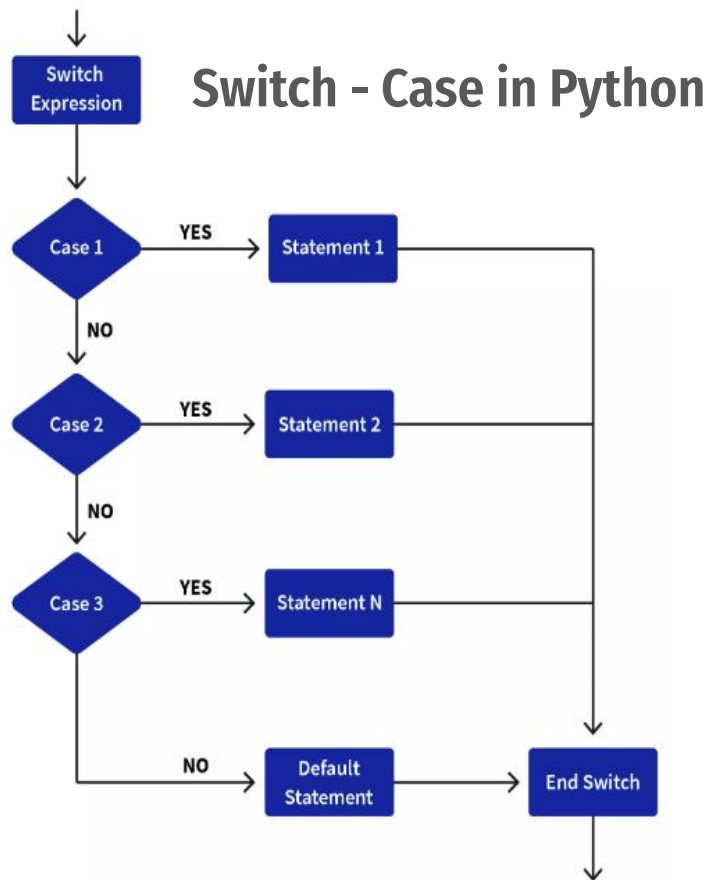
if...else PROMAX

```
1 def check(num):
2     return "Even" if (num % 2 == 0) else "Odd"
```

```
1 age = 9
2 drink = "beer" if age > 22 else "coke" if age > 10 else "juice"
3 print(drink)
```



# 2. PYTHON TRONG KHOA HỌC DỮ LIỆU #1



```
def describe_number(x):  
    match x:  
        case 0:  
            return "Zero"  
        case 1:  
            return "One"  
        case 2:  
            return "Two"  
        case _:  
            return "Other"  
  
print(describe_number(0)) # Output: Zero  
print(describe_number(1)) # Output: One  
print(describe_number(5)) # Output: Other
```

Match - Case

```
print("Choices:\n 1 - Email\n 2 - SMS\n 3 - Push Notification")  
x = input("Please enter your choice:\n")
```

```
x = int(x)
```

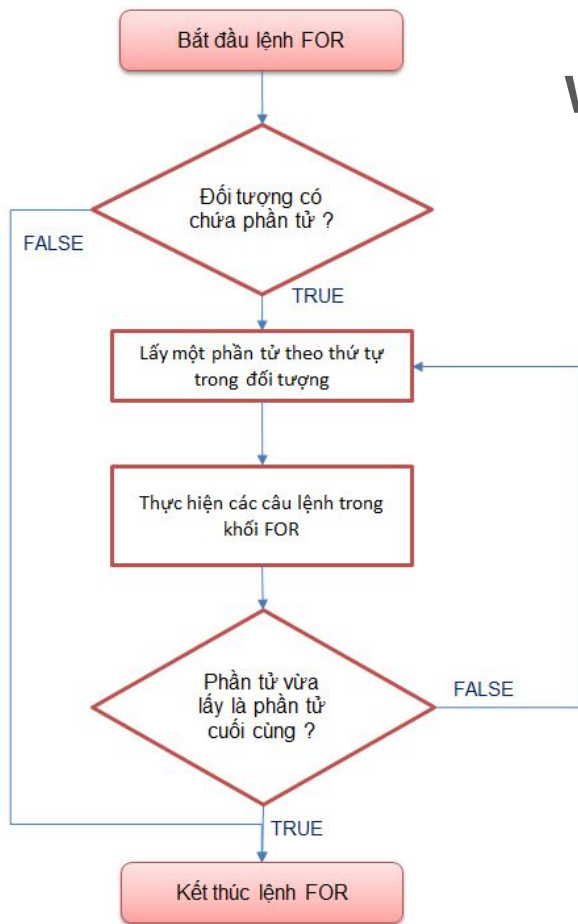
```
if x is 1:  
    print('sending email')  
elif x is 2:  
    print('sending sms')  
elif x is 3:  
    print('sending push notification')  
else:  
    print('wrong choice')
```

If - elif - else

```
choices_dict = {1: 'sending email', 2: 'sending sms', 3: 'sending push notification'}  
  
print(choices_dict.get(x, 'wrong choice'))
```

## 2. PYTHON TRONG KHOA HỌC DỮ LIỆU #1

### Vòng lặp for



```
numbers = [1, 2, 3, 4, 5]

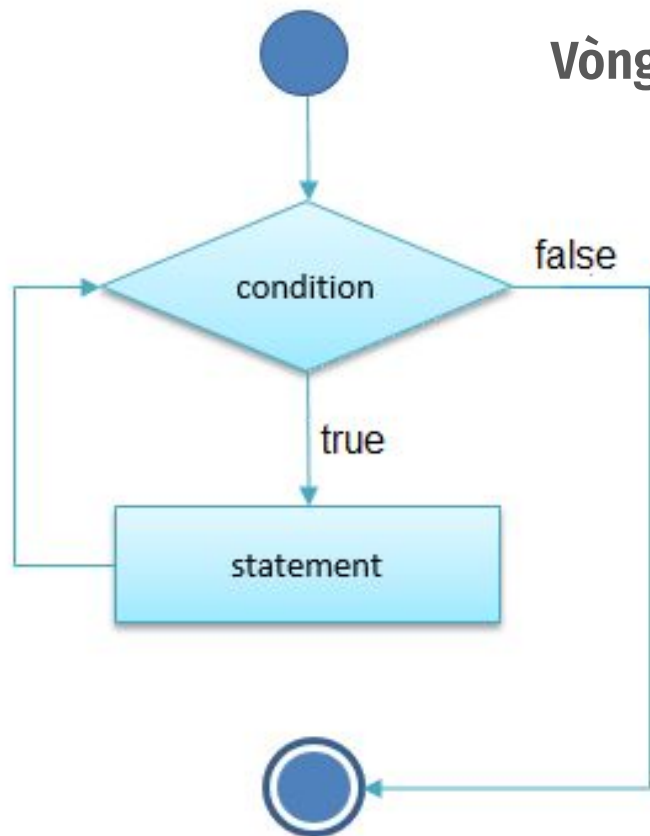
for number in numbers:
    print(number)
```

1  
2  
3  
4  
5



## 2. PYTHON TRONG KHOA HỌC DỮ LIỆU #1

### Vòng lặp while



```
numbers = [1, 2, 3, 4, 5]

i = 0
while i < len(numbers):
    print(numbers[i])
    i += 1
```

1  
2  
3  
4  
5



## 2. PYTHON TRONG KHOA HỌC DỮ LIỆU #1

### Break - continue

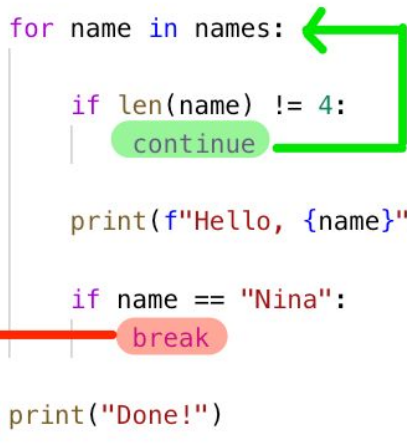
```
names = ["Jimmy", "Rose", "Max", "Nina", "Phillip"]
```

```
for name in names:
    if len(name) != 4:
        continue

    print(f"Hello, {name}")

    if name == "Nina":
        break

print("Done!")
```



### Assert

```
x = "python"
```

```
# if condition is true, it returns nothing
assert x == "python"
```

```
# if condition is not true, then an assertion error is thrown
assert x == "community"
```

```
-----
AssertionError                                Traceback (most recent call last)
<ipython-input-8-e587b3c8e281> in <module>
----> 1 assert x == "community"

AssertionError:
```

## 2. PYTHON TRONG KHOA HỌC DỮ LIỆU #1

### Khai báo hàm

```
def functionname(param, param2,...):  
    statements(s)
```

Hàm nếu không trả dữ liệu mặc định sẽ trả về giá trị None

### Ví dụ khai báo hàm

```
def sum(a, b):  
    return (a+b)
```

Hàm có hỗ trợ giá trị mặc định cho tham số khi không truyền vào.

```
def plus(c, d = 10):  
    return (c+d)
```

Nếu gọi hàm trên như sau:

```
plus(2)  
(kết quả trả về là 12)
```

### Ví dụ gọi hàm

```
sum(1, 2)  
(trả về giá trị là 3)
```

Một khác biệt trong cách gọi hàm của Python là chúng ta có thể thay đổi thứ tự tham số truyền vào bằng cách đặt tên tham số khi gọi hàm. Ví dụ ta có thể gọi hàm `sum(a,b)` ở ví dụ trên bằng cách truyền tham số `b` trước `a` như sau:

```
sum(b = 1, a = 10)
```

## 2. PYTHON TRONG KHOA HỌC DỮ LIỆU #1

### Xử lý chuỗi

Một chuỗi có thể khai báo bằng dấu nháy đôi " hoặc đơn '. Ví dụ các chuỗi sau:

```
str1 = "Hello"  
str2 = 'world'
```

Có thể truy xuất từng ký tự trong một chuỗi theo hình thức index, ví dụ: `str1[0]`, `str1[1]` ...

Có thể sử dụng 3 dấu nháy (đôi hoặc đơn) để khai báo chuỗi trên nhiều dòng. Ví dụ:

```
paragraph = """This is line 1  
This is line 2  
This is line 3"""
```

### Nối chuỗi

Có thể tạo một chuỗi dài từ việc nối các chuỗi lại theo cú pháp:

```
str = str1 + " " + str2
```

### Trích xuất chuỗi con

Truy xuất thông qua toán tử lấy khoảng `[start:end]` (range).

Mặc định start là từ vị trí đầu chuỗi (0) và end là đến vị trí cuối chuỗi. Ví dụ:

```
[2] str = "Hello World"
```

```
print(str[0:4])
```

```
Hell
```

```
[3] print(str[:4])
```

```
Hell
```

```
[4] print(str[-3:])
```

```
rl d
```

```
▶ print(str[6:-3])
```

```
Wo
```

## 2. PYTHON TRONG KHOA HỌC DỮ LIỆU #1

### Lấy độ dài của chuỗi

```
count = len("Hello world")  
(count có giá trị 11)
```

### Tìm & thay thế nội dung

Có thể tìm và thay thế trong chuỗi bằng cách gọi phương thức `replace(search, replace[, max])` của một chuỗi.

```
str = 'Hello world'  
newstr = str.replace('Hello', 'Bye')
```

### Tìm vị trí chuỗi con

Có thể tìm vị trí của một chuỗi con trong chuỗi lớn bằng cách gọi phương thức `find(str, beg=0 end=len(string))`. Bắt đầu là vị trí 0, nếu không tìm ra thì trả về -1. Ví dụ:

```
str = "Hello World"  
print(str.find('world'))  
  
-1
```

Hàm `find()` sẽ tìm theo thứ tự từ trái qua phải của chuỗi, tức là từ lần xuất hiện đầu tiên. Có thể dùng hàm `rfind()` để tìm theo vị trí từ cuối chuỗi về phía trước.

## 2. PYTHON TRONG KHOA HỌC DỮ LIỆU #1

### Tách chuỗi

Có thể tách chuỗi dựa theo một chuỗi delimiter bằng cách gọi phương thức:

`split(str="", num=string.count(str))`

```
str = "Hello World"

print(str.split(' '))

['Hello', 'World']
```

Có thể sử dụng hàm `splitlines()` để tách chuỗi theo từng hàng và loại bỏ ký tự `NEWLINE`.

### Xóa khoảng trắng dư thừa

Có thể loại bỏ các ký tự (mặc định là ký tự khoảng trắng) trước và sau một chuỗi, bằng cách gọi các phương thức sau:

- `strip([chars])` : loại bỏ trước và sau chuỗi.
- `lstrip([chars])` : loại bỏ phía trước chuỗi.
- `rstrip([chars])` : loại bỏ phía sau chuỗi.

### Một số hàm xử lý chuỗi

- `isnumeric()` : Kiểm tra một chuỗi có phải là chuỗi số.
- `lower()` : Chuyển chuỗi hết thành chữ thường.
- `upper()` : Chuyển chuỗi hết thành chữ HOA.



### 3. HƯỚNG DẪN LÀM GAME LÔ ĐỀ

**Đề bài:** Lô đề một loại hình cá cược dựa trên kết quả xổ số với điểm quy đổi 1:70 (trúng 1 ăn 70). Mỗi ngày sẽ có 7 giải từ giải Nhất đến giải Bảy. Người chơi được tính là trúng lô khi cược số lô trùng với bộ giải trên. Nếu trúng, số tiền thưởng sẽ bằng tiền cược nhân với 70, thua sẽ bị mất tiền cược. Người chơi mỗi lần vào chơi sẽ có trong tay 1.000.000 đồng. Yêu cầu của bạn là mô phỏng lại game Lô Đề Học theo mô tả trên. Game sẽ có chức năng chính:

- **Chơi lô:** Người dùng cược 1 con số có 2 chữ số, sau đó hệ thống quay ngẫu nhiên 7 giải và hiển thị ra màn hình (số lô và số quay ngẫu nhiên là số có 2 chữ số từ 10-99).
  - Nếu trúng lô, sẽ hiển thị số lô trúng, tiền thưởng và tổng tiền.
  - Nếu người chơi tách lô sẽ hiển thị tách lô và số tiền còn lại.
  - Người chơi sẽ được lựa chọn chơi tiếp hay dừng cuộc chơi trở về màn hình chính.
  - Nếu người chơi không còn tiền, hệ thống sẽ hiển thị thông báo phá sản và không được chơi nữa.
- **Nạp tiền:** Người chơi sẽ được nạp tiền để chơi lô. Được phép quay trở về màn hình chính để chọn các chức năng khác.
- **Thoát game:** Người chơi khi lựa chọn chức năng này sẽ thoát khỏi game.

**Lưu ý:** Cần xử lý đầy đủ các trường hợp ngoại lệ có thể xảy ra. Ví dụ: người dùng nhập sai định dạng thì yêu cầu nhập lại,... Sử dụng hàm để xử lý.

# 4. TỔNG KẾT

Hoàn thành Form điểm danh sau (10'): [Form điểm danh](#)

Bài tập sau buổi học:

- [Lab 7.1: Thực hành viết code Python đầu tiên](#)
- [Lab 7.2: Các thao tác với String](#)
- [Lab 7.3: Nhánh và câu điều kiện](#)
- [Lab 7.4: Vòng lặp](#)
- [Lab 7.5: Hàm](#)
- [Bài tập tổng hợp](#)
- [Game Lô Đề Học](#)

Yêu cầu:

- Đặt tên folder theo mẫu DSMMC\_<Buổi học>\_<Tên> (Ví dụ: *DSMMC\_Buoi2\_DangNH*)
- Hoàn thành các bài lab sau đó nộp lên Git chung của lớp: [Link Git](#) , sau đó submit link folder git trên [Form bài tập](#)
- Hạn nộp: Trước 12 giờ của **BUỔI HỌC THỨ 4**

***THANK YOU !***