



<http://vku.udn.vn>

ĐẠI HỌC ĐÀ NẴNG

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT - HÀN  
Vietnam - Korea University of Information and Communication Technology

# Python cơ bản

Python cho AI



## VUI MỪNG

- Giới thiệu
- Biến và toán tử
- Chức năng
- Cấu trúc lựa chọn
- Cấu trúc vòng lặp
- Các lỗi thường gặp



## VUI MỪNG

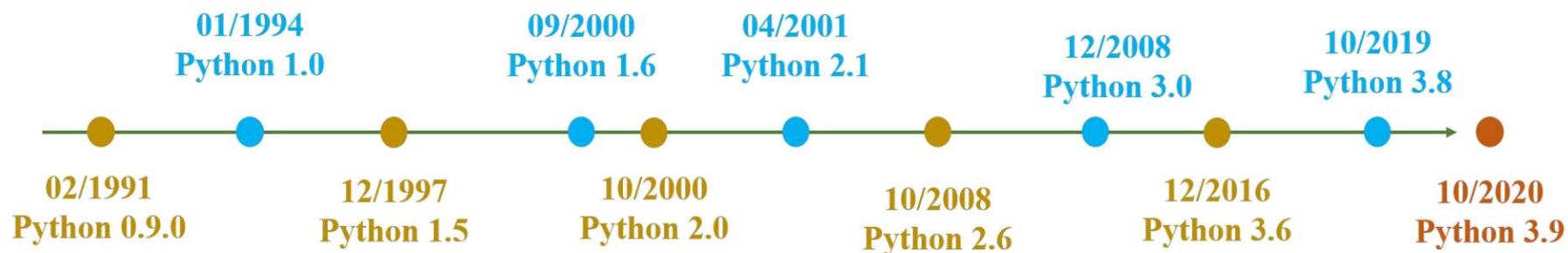
### • Giới thiệu

- Biến và toán tử
- Chức năng
- Cấu trúc lựa chọn
- Cấu trúc vòng lặp
- Các lỗi thường gặp



# Giới thiệu

## Lịch sử Python



Ý tưởng từ 1980s



Được đặt tên theo  
nhóm hài Monty Python



Bắt đầu cài đặt  
từ 12/1989



Guido van Rossum



228,855 python packages  
(PyPI)



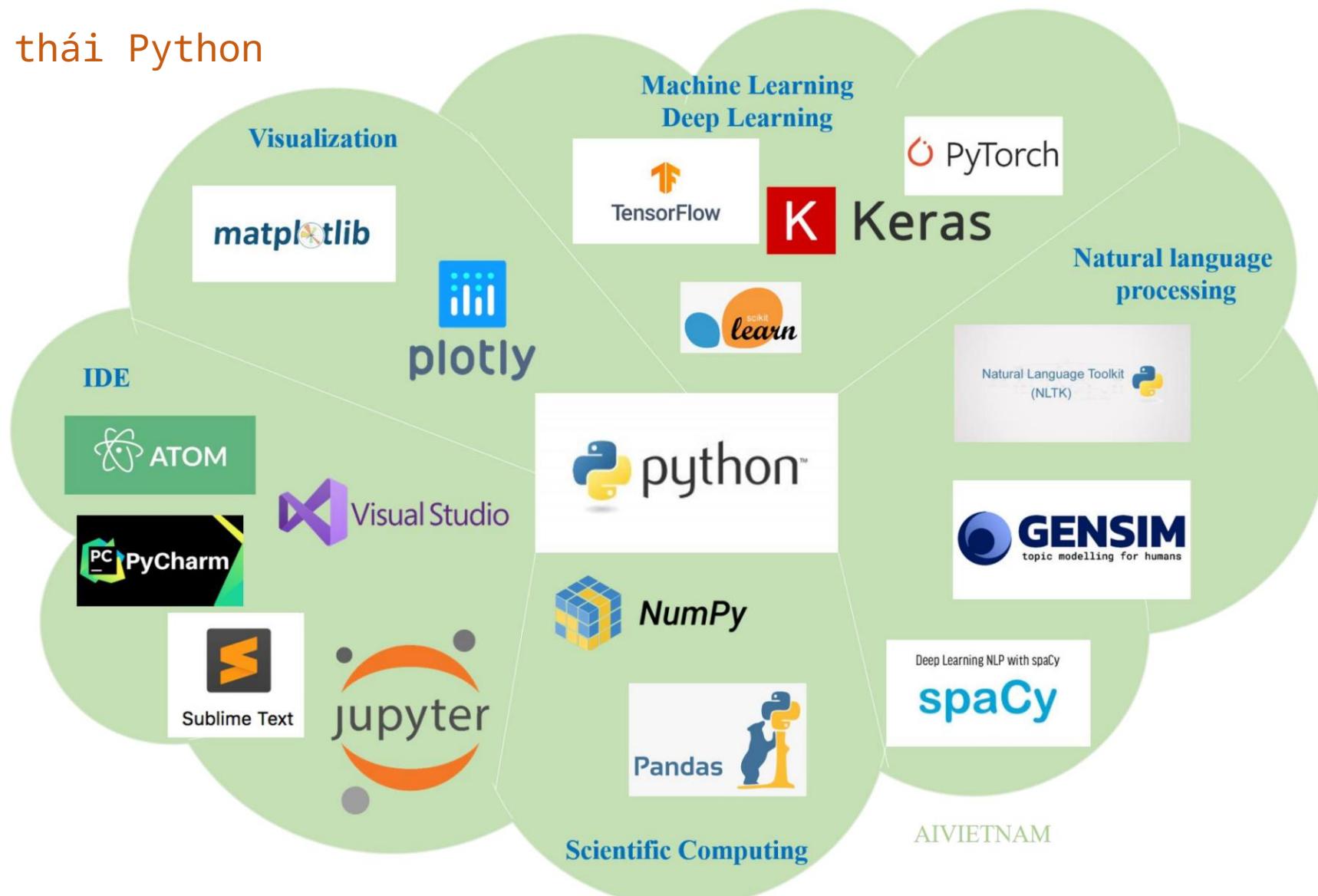
AIVIETNAM

Hỗ trợ rất mạnh cho  
Data Science và Machine Learning



# .Giới thiệu

## Hệ sinh thái Python





## Giới thiệu

### Ngôn ngữ lập trình hàng đầu (2020)

**Choose a Ranking**

**Language Types**

IEEE Spectrum	Trending	Web	Enterprise	
Jobs	Open	Custom	Mobile	Embedded
<a href="#">Create custom ranking</a>				(Click to hide)

Language Ranking: **IEEE Spectrum**

[https://spectrum.ieee.org/static/  
interactive/the-top-programming-languages-2020](https://spectrum.ieee.org/static/interactive/the-top-programming-languages-2020)

Rank	Language	Type	Score
1	Python▼		100.0
2	Java▼		95.3
3	C▼		94.6
4	C++▼		87.0
5	JavaScript▼		79.5
6	R▼		78.6
7	Arduino▼		73.2
8	Go▼		73.1



## .Giới thiệu

Chương trình Python đầu tiên

Tạo tệp .py

Viết mã Python

Chạy tập tin



## .Giới thiệu

### Chương trình Python đầu tiên - Sử dụng Google Colab

<https://colab.research.google.com/notebooks/intro.ipynb>

The screenshot shows a Google Colab interface. At the top, there's a logo consisting of three overlapping circles in orange, yellow, and blue, followed by the text "python.ipynb" and a star icon. Below the logo is a menu bar with "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help". A horizontal line separates the menu from the main workspace. On the left side of the workspace, there are two buttons: "+ Code" and "+ Text". The main area contains a code cell with the following content:

```
<> [1] print('my first program !!!')
```

Below the code cell, the output is displayed in a separate cell:

```
my first program !!!
```



## .Giới thiệu

### Chương trình Python đầu tiên - Sử dụng Anaconda + Jupyter Notebook

The screenshot shows a Jupyter Notebook interface with the following details:

- Title Bar:** jupyter Python Last Checkpoint: vài giây trước (unsaved changes)
- Toolbar:** File, Edit, View, Insert, Cell, Kernel, Widgets, Help. Below the toolbar are icons for Save, New, Close, Cell, Run, Cell Kernel, Cell Kernel, and Code.
- Code Cell:** In [1]:  
1 print('Hello!')  
Hello!
- Empty Cell:** In [ ]: 1



## VUI MỪNG

- Giới thiệu
- Biến và toán tử
- Chức năng
- Cấu trúc lựa chọn
- Cấu trúc vòng lặp
- Các lỗi thường gặp



# Biến và toán tử

tên biến = giá trị biến

số nguyên	1, 2, 3, 0, -1, -2
Trôi nổi	1,5, 0,5, -3,21, 1,0
Sợi dây	'Joe', 'Schmoe', 'Joe', 'Schmoe'
Boolean	Đúng sai

```

1. # tạo biến number_of_days có giá trị là 10
2. number_of_days = 10
3.
4. # tạo biến số distance có giá trị là 20.5
5. distance = 20.5
6.
7. # tạo biến chuỗi greeting có giá trị là "Hello AI-VIET-NAM"
8. greeting = "Hello AI-VIET-NAM"
9.
10. # tạo biến boolean is_student có giá trị là True
11. is_student = True

```



# .Biến và toán tử

## tên biến

Nên có ý nghĩa

Không thể sử dụng từ khóa

and	exec	not
as	finally	or
assert	for	pass
break	from	print
class	global	raise
continue	if	return
def	import	try
del	in	while
elif	is	with
else	lambda	yield
except	True	False

## Ví dụ

1. | `return = 5`

Kết quả

```
File "/tmp/sessions/90e4ade1bc0a8087/main.py", line 1
    return = 5
          ^
SyntaxError: invalid syntax
```



## .Biến và toán tử

Toán Tử Số Học		
Toán Tử	Ý nghĩa	Ví dụ
+	Phép cộng	$2 + 3 = 5$
-	Phép trừ	$3 - 2 = 1$
*	Phép nhân	$2 * 2 = 4$
/	Phép chia	$10 / 2 = 5$
%	Phép chia lấy số dư	$10 \% 4 = 2$
//	Phép chia lấy số nguyên	$18 // 5 = 3$
**	Phép lũy thừa	$3^{**}5 = 243$

```

main.py
1 # aivietnam.ai
2
3 # Khai báo biến x với giá trị là 9
4 x = 3
5
6 # Khai báo biến y với giá trị là 6
7 y = 2
8
9 # Lấy giá trị của x cộng giá trị của y rồi in ra kết quả
10 print(x + y)
11
12 # Lấy giá trị của x trừ giá trị của y rồi in ra kết quả
13 print(x - y)
14
15 # Lấy giá trị của x nhân với giá trị của y rồi in ra kết quả
16 print(x * y)
17
18 # Lấy giá trị của x chia với giá trị của y rồi in ra kết quả
19 print(x / y)
20
21 # Lấy giá trị của x chia với giá trị của y chỉ lấy phần nguyên
22 print(x // y)
23
24 # Lấy giá trị x mũ y (x^y)
25 print(x ** y)

```



## .Biến và toán tử

### Ví dụ

Chuyển đổi độ C sang độ F

$$= \frac{9}{5} + 32$$



```

1. # input
2. temp_c = float(input('Nhập nhiệt độ theo độ C: '))
3.
4. # process
5. temp_f = ((9 / 5) * temp_c) + 32
6.
7. # output
8. print('Nhiệt độ F là: ', temp_f)

```

Chuyển đổi độ F sang độ C

$$= ( -32 ) \frac{9}{5}$$



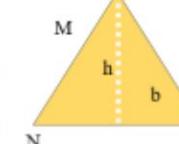
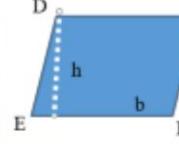
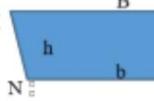
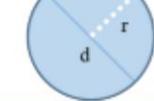
???



# .Biến và toán tử

## Ví dụ

Tính toán  
Chu vi và diện tích  
các hình thường gặp

TÊN	HÌNH DẠNG	DIỆN TÍCH	CHU VI
HÌNH TAM GIÁC		$A = \frac{b \times h}{2}$	$P = MN + NP + PM$
HÌNH BÌNH HÀNH		$A = b \times h$	$P = DE + EF + FG + GD$
HÌNH CHỮ NHẬT		$A = L \times w$	$P = L+w+L+w$ $P = 2L+2w$
HÌNH VUÔNG		$A = l^2$	$P = l + l + l + l$ $P = 4l$
HÌNH THAN		$A = \frac{(B+b) \times h}{2}$	$P = MN + NP + PR + RM$
HÌNH TRÒN		$A = \pi r^2$	$C = 2\pi r \times \pi d$



## .Biến và toán tử

# Toán tử so sánh

TOÁN TỬ SO SÁNH	
TOÁN TỬ	Ý NGHĨA
<code>==</code>	Bằng
<code>!=</code>	Khác
<code>&gt;</code>	Lớn hơn
<code>&lt;</code>	Nhỏ hơn
<code>&gt;=</code>	Lớn hơn hoặc bằng
<code>&lt;=</code>	Nhỏ hơn hoặc bằng

```

1 a = 5
2 b = 8
3
4 # hỏi biến a có bằng biến b hay không?
5 print(a == b)
6
7 # hỏi biến a có khác biến b hay không?
8 print(a != b)
9
10 # hỏi biến a có lớn hơn biến b hay không?
11 print(a > b)
12
13 # hỏi biến a có lớn hơn hoặc bằng biến b hay không?
14 print(a >= b)
15
16 # hỏi biến a có nhỏ hơn biến b hay không?
17 print(a < b)
18
19 # hỏi biến a có lớn hơn hoặc bằng biến b hay không?
20 print(a <= b)

```

False  
True  
False  
False  
True  
True



## VUI MỪNG

- Giới thiệu
- Biến và toán tử
- **Chức năng**
- Cấu trúc lựa chọn
- Cấu trúc vòng lặp
- Các lỗi thường gặp



## Chức năng

### Chức năng tích hợp sẵn in (tham số)

```

1. # tạo biến number_of_days có giá trị là 10
2. number_of_days = 10
3. print(number_of_days)

4.

5. # tạo biến số distance có giá trị là 20.5
6. distance = 20.5
7. print(distance)

8.

9. # tạo biến chuỗi greeting có giá trị là "Hello AI-VIET-NAM"
10. greeting = "Hello AI-VIET-NAM"
11. print(greeting)

12.

13. # tạo biến boolean is_student có giá trị là True
14. is_student = True
15. print(is_student)

```

10  
20.5  
Hello AI-VIET-NAM  
True

### loại (tham số)

```

1. # tạo biến number_of_days có giá trị là 10
2. number_of_days = 10
3. data_type_of_number_of_days = type(number_of_days)
4. print(data_type_of_number_of_days)

5.

6. # tạo biến số distance có giá trị là 20.5
7. distance = 20.5
8. data_type_of_distance = type(distance)
9. print(data_type_of_distance)

10.

11. # tạo biến chuỗi greeting có giá trị là "Hello AI-VIET-NAM"
12. greeting = "Hello AI-VIET-NAM"
13. data_type_of_greeting = type(greeting)
14. print(data_type_of_greeting)

15.

16. # tạo biến boolean is_student có giá trị là True
17. is_student = True
18. data_type_of_is_student = type(is_student)
19. print(data_type_of_is_student)

```

1. <class 'int'>
2. <class 'float'>
3. <class 'str'>
4. <class 'bool'>

## Chức năng tích hợp sẵn đầu vào (prompt)

```

1. # yêu cầu user nhập thông tin từ bàn phím
2. input_data = input("Hãy nhập tên của bạn!")
3.
4. # lấy kiểu dữ liệu của input_data
5. data_type_input_data = type(input_data)
6.
7. print("-----")
8. print(data_type_input_data)
9. print(input_data)

```

Hãy nhập tên của bạn! AIVIETNAM

<class 'str'>  
AIVIETNAM

Hãy nhập tên của bạn! 32

<class 'str'>  
32

Hãy nhập tên của bạn! 70.5

<class 'str'>  
70.5

## Chức năng tích hợp - Chuyển đổi loại

`int()`

```

1. # yêu cầu user nhập thông tin từ bàn phím
2. input_data = input("Hãy nhập tuổi của bạn!")
3.
4. # ép kiểu dữ liệu từ string về int
5. input_data_int = int(input_data)
6.
7. # lấy kiểu dữ liệu của input_data
8. data_type = type(input_data_int)
9.
10. print("-----")
11. print(data_type)
12. print(input_data_int)

```

`trôi nổi()`

```

1. # yêu cầu user nhập thông tin từ bàn phím
2. input_data = input("Hãy nhập cân nặng của bạn!")
3.
4. # ép kiểu dữ liệu từ string về float
5. input_data_float = float(input_data)
6.
7. # lấy kiểu dữ liệu của input_data
8. data_type = type(input_data_float)
9.
10. print("-----")
11. print(data_type)
12. print(input_data_float)

```



## .Chức năng

### Mô-đun ngẫu nhiên và toán học

```

1 # absolute value of x
2
3 import math
4
5 n1 = 1
6 n2 = -2
7
8 print(math.fabs(n1))
9 print(math.fabs(n2))

```

1.0  
2.0

```

1 # log(x)
2
3 import math
4
5 x = 4
6 print(math.log(x))
7 print(math.log(math.e))

```

1.3862943611198906  
1.0

```

1 # sine of x
2
3 import math
4
5 x = 2
6 print(math.sin(x))

```

0.9092974268256817

```

1 # the e number
2
3 import math
4
5 print(math.e)

```

2.718281828459045

```

1 # exponential of x
2
3 import math
4
5 x = 2
6 print(math.exp(x))

```

7.38905609893065

```

1 # square root of x
2
3 import math
4
5 x = 4
6 print(math.sqrt(x))

```

2.0

```

1 # cosine of x
2
3 import math
4
5 x = 2
6 print(math.cos(x))

```

-0.4161468365471424

```

1 # the PI number
2
3 import math
4
5 print(math.pi)

```

3.141592653589793

<https://docs.python.org/3/library/math.html>



## Mô-đun ngẫu nhiên và toán học

```

1 # generate random floating-point numbers in [0, 1)
2
3 import random
4
5 print(random.random())
6 print(random.random())
7 print(random.random())
8 print(random.random())

```

0.6408053793939149  
0.5857197983520188  
0.8967201901385281  
0.8720695344313283

```

1 # generate random integer numbers in [a, b]
2
3 import random
4
5 print(random.randint(1, 9))
6 print(random.randint(1, 9))
7 print(random.randint(1, 9))
8 print(random.randint(1, 9))

```

7  
3  
6  
4

<https://docs.python.org/3/library/random.html>

## Xây dựng chức năng

## Xác định tên hàm

Chữ thường có dấu gạch dưới và bắt đầu bằng một động từ

## Thụt lè

Sử dụng 4 dấu cách để thụt lè

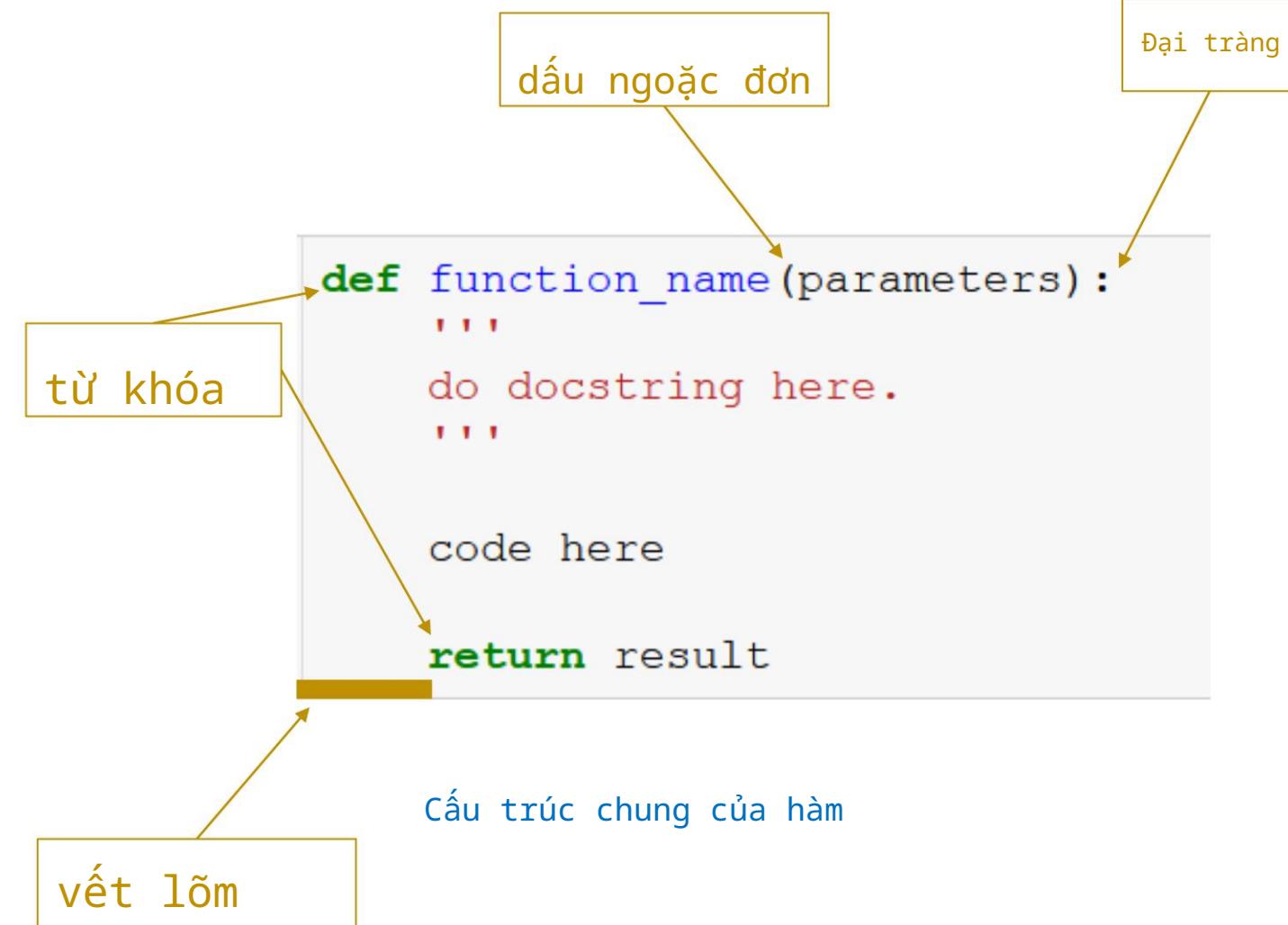
## Xác định tham số hàm

Dữ liệu đầu vào giúp thực hiện chức năng

## Làm chuỗi tài liệu

Giải thích và mô tả chức năng

## Đầu ra của hàm





## .Chức năng

### Xây dựng chức năng

```
def function_name(parameters):
    """
        do docstring here.
    """

    code here

    return result
```

Tên	khu vực tính toán_hình chữ nhật
Tham số	chiều cao chiều rộng
đầu ra	khu vực

```
1 def compute_rectangle_area(height, width):
2     """
3         This function aims to compute area for a rectangle.
4
5         height -- the height of the rectangle
6         width -- the width of the rectangle
7
8         This function returns the area of the rectangle
9     """
10
11     area = height*width
12
13     return area
```

## Giá trị mặc định

### Giá trị mặc định



```

1 def compute_rectangle_area(height=0, width=0):
2     """
3         This function aims to compute area for a rectangle.
4
5         height -- the height of the rectangle
6         width -- the width of the rectangle
7
8         This function returns the area of the rectangle
9     """
10
11    area = height*width
12
13    return area

```

```

1 areal = compute_rectangle_area(5, 6)
2 print('area 1: ', areal)
3
4 area2 = compute_rectangle_area(height=5, width=6)
5 print('area 2: ', area2)
6
7 area3 = compute_rectangle_area(width=6, height=5)
8 print('area 3: ', area3)
9
10 area4 = compute_rectangle_area(width=6,
11                                height=5)
12 print('area 4: ', area4)
13
14 area5 = compute_rectangle_area()
15 print('area 5: ', area5)

area 1: 30
area 2: 30
area 3: 30
area 4: 30
area 5: 0

```

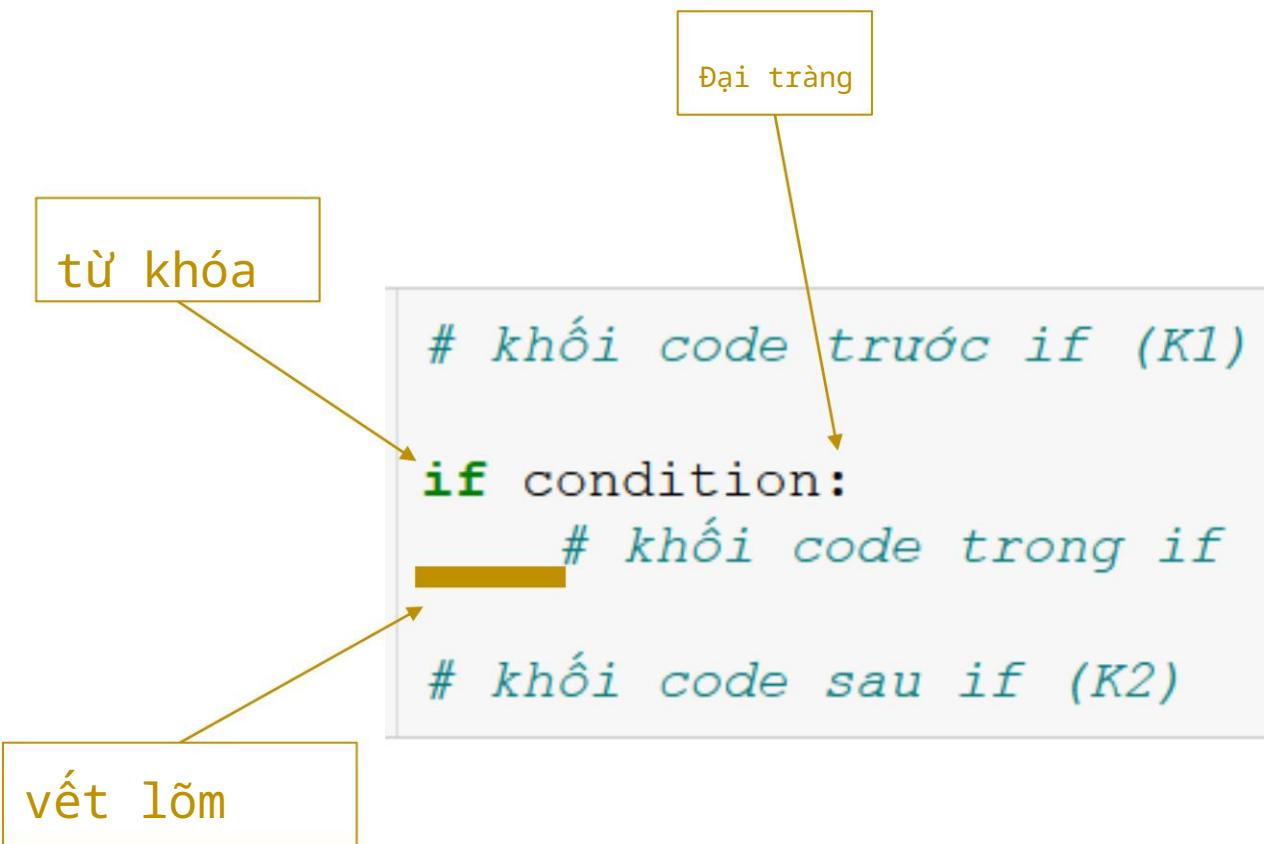


## VUI MỪNG

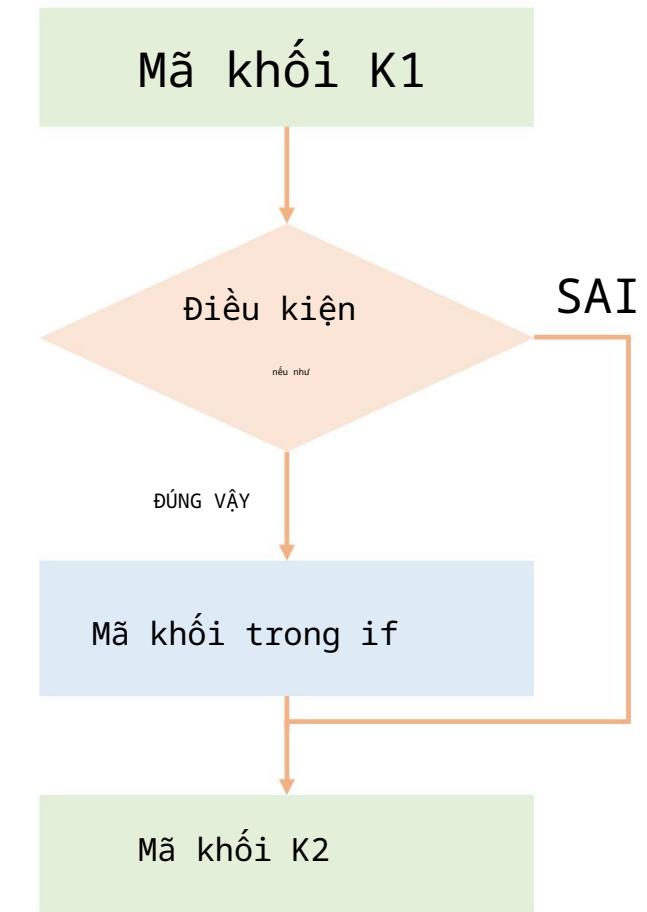
- Giới thiệu
- Biến và toán tử
- Chức năng
- Cấu trúc lựa chọn
  - Cấu trúc vòng lặp
  - Các lỗi thường gặp



nếu có điều kiện



## Cấu trúc lựa chọn





## Cấu trúc lựa chọn

nếu có điều kiện

$$\text{ReLU} \in \begin{cases} 0 & \text{nếu } \leq 0 \\ & \text{nếu } > 0 \end{cases}$$

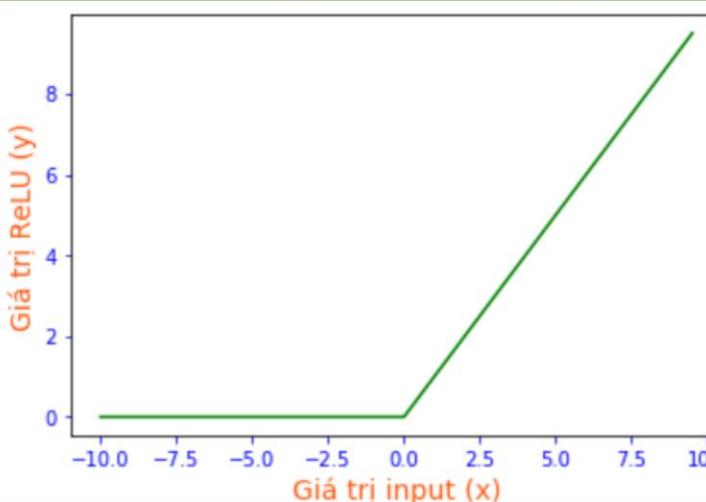
**data** = 

1	5	-4	3	-2
---	---	----	---	----

**data\_a** = **ReLU(data)**

**data\_a** = 

1	5	0	3	0
---	---	---	---	---



```

1 def ReLU(x):
2     """
3         This function aims to compute ReLU for a value x.
4
5         x -- an input value
6
7         This function returns the ReLU of x
8         """
9
10    result = 0
11
12    if x > 0:
13        result = x
14
15    return result

```

```

1 value1 = ReLU(x=5)
2 value2 = ReLU(x=-2)
3
4 print(value1)
5 print(value2)

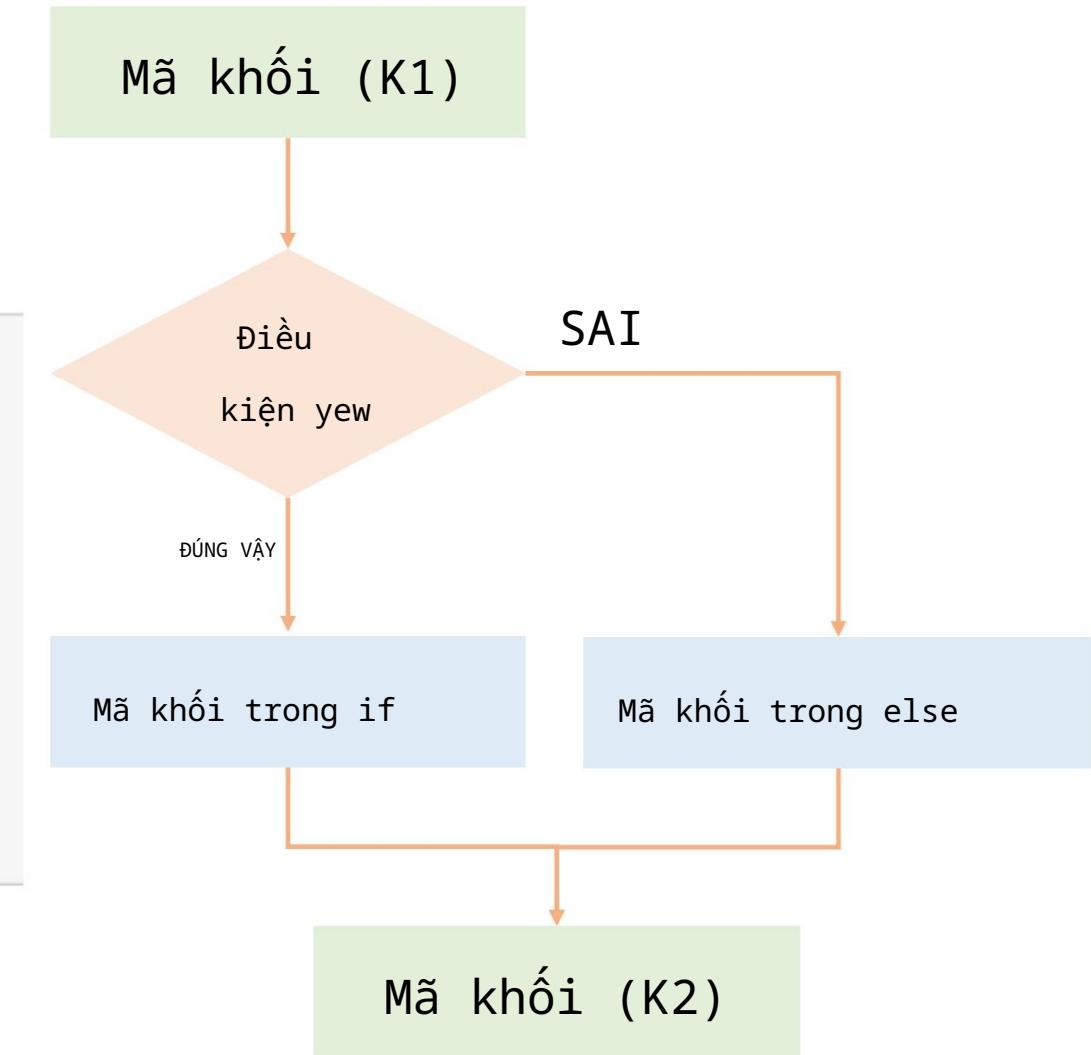
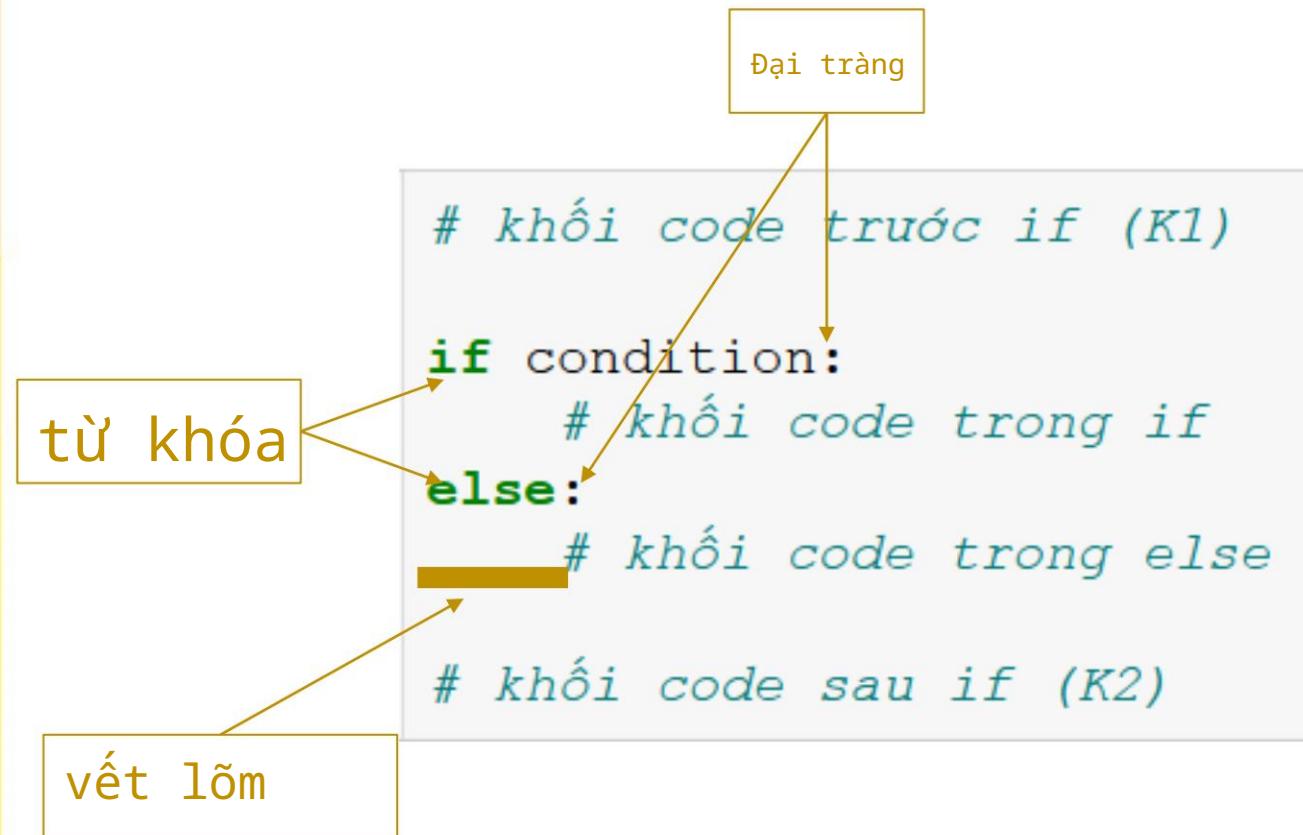
```

5  
0



## Cấu trúc lựa chọn

### điều kiện if - else





## Cấu trúc lựa chọn

điều kiện if - else

$$\text{ReLU } \in ) \begin{array}{l} 0 \text{ nếu } \leq 0 \\ \text{nếu } > 0 \end{array}$$

**data =** 

**data\_a = ReLU(data)**

**data\_a =** 

## Cấu trúc lựa chọn

### điều kiện if - Elif - else

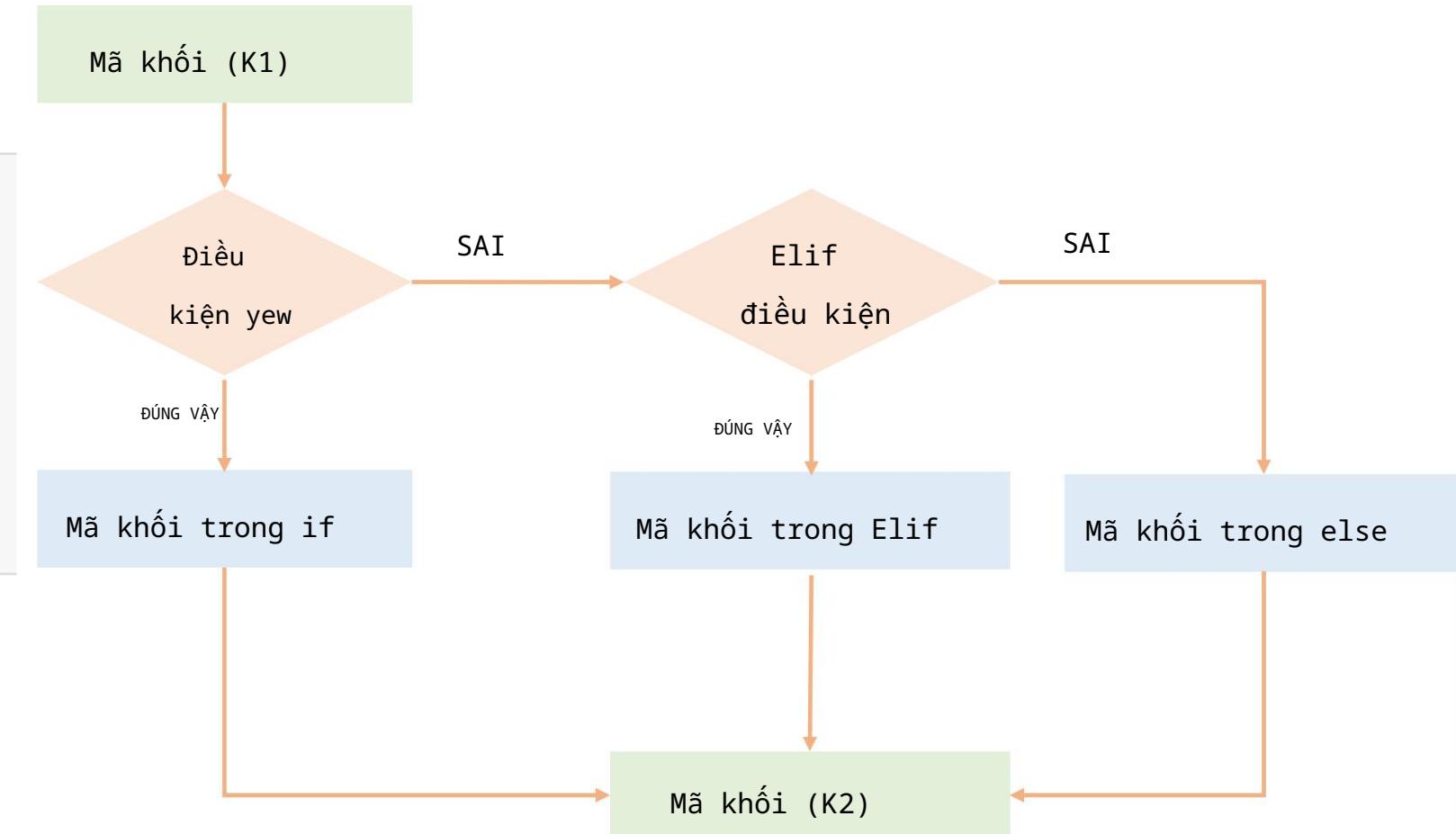
```

1  # khối code trước if-else (K1)
2
3  if condition1:
4      # khối code trong if
5  elif condition2:
6      # khối code trong elif
7  else:
8      # khối code trong else
9
10 # khối code sau if-else (K2)

```

Đại tràng

vết lõm





## Cấu trúc lựa chọn

điều kiện if - Elif - else

$$\text{LBP} ( , ) = \begin{cases} 0 & \text{nếu } = 1 \\ & \text{nếu } > 1 \\ & \text{nếu } < \end{cases}$$



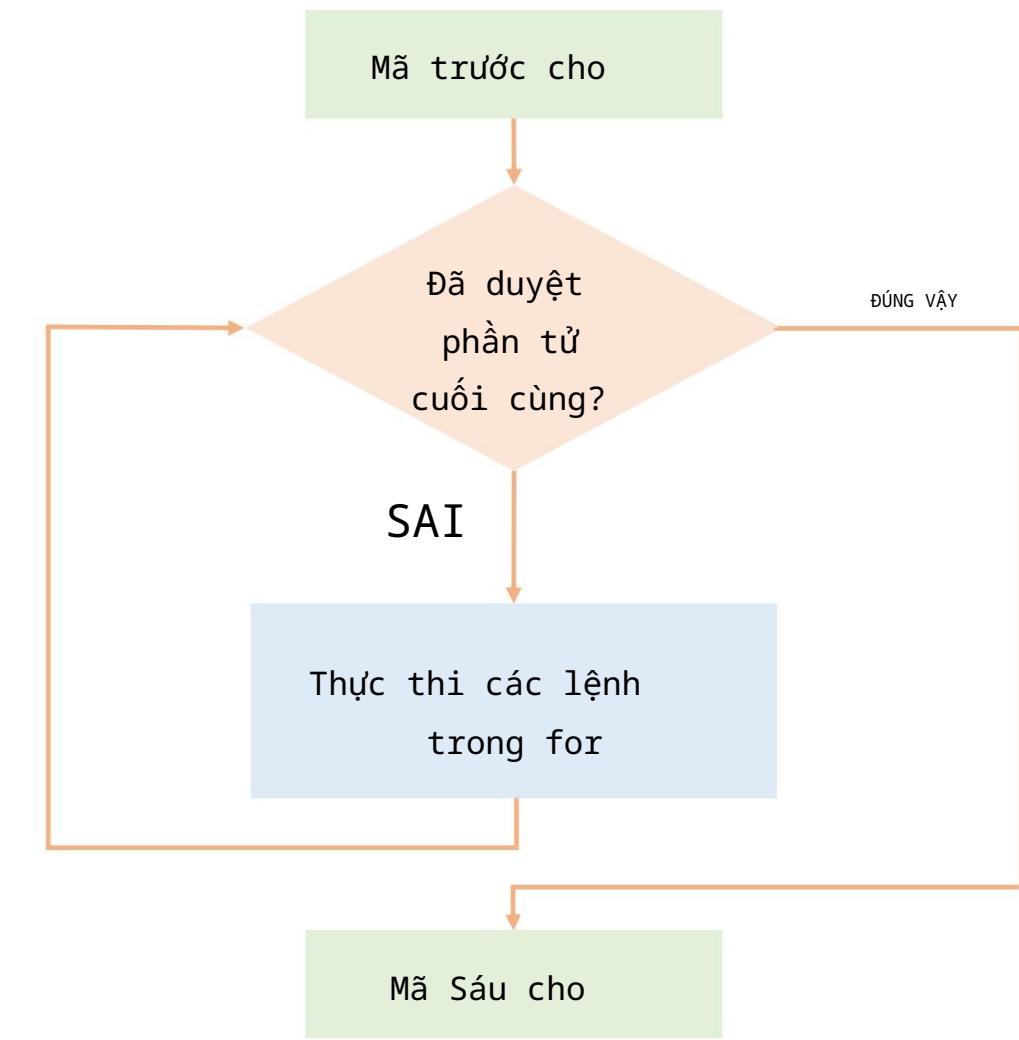
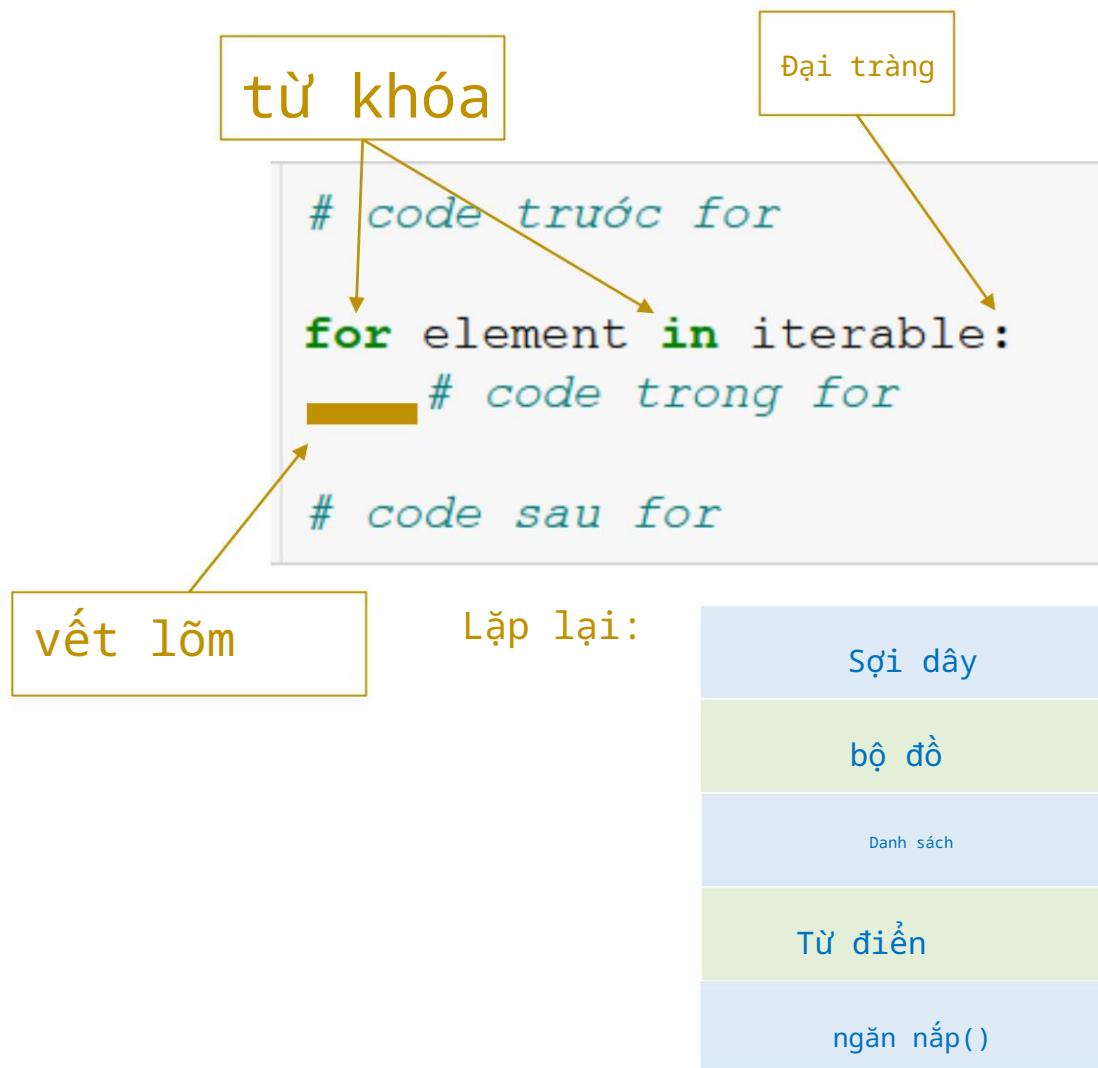
## VUI MỪNG

- Giới thiệu
- Biến và toán tử
- Chức năng
- Cấu trúc lựa chọn
- **Cấu trúc vòng lặp**
- Các lỗi thường gặp



# Cấu trúc vòng lặp

Vòng lặp cho





## Vòng lặp cho

```

1 # iterate a list
2
3 fruits = ['apple', 'banana', 'melon', 'peach']
4
5 for fruit in fruits:
6     print(fruit)

```

apple  
banana  
melon  
peach

```
# iterate a dictionary
```

```

1
2
3 parameters = {'learning_rate': 0.1,
4                      'optimizer': 'Adam',
5                      'metric': 'Accuracy'}
6
7 for key in parameters:
8     print(key, parameters.get(key))

```

learning\_rate 0.1  
optimizer Adam  
metric Accuracy

```
# iterate a tuple
```

```

1
2
3 fruits = ('apple', 'banana', 'melon')
4
5 for fruit in fruits:
6     print(fruit)

```

apple  
banana  
melon

## .Cấu trúc vòng lặp

```

1 # iterate a string
2
3 greeting = 'Hello'
4
5 for char in greeting:
6     print(char)

```

H  
e  
l  
l  
o

```

1 # use range()
2
3 for i in range(5):
4     print(i)

```

0  
1  
2  
3  
4



## Cấu trúc vòng lặp

Vòng lặp cho

phạm vi (bắt đầu = 0, dừng, bước = 1)

phạm vi (bắt đầu=0, dừng=5, bước=1)



0, 1, 2, 3, 4

phạm vi(5)



0, 1, 2, 3, 4

# .Cấu trúc vòng lặp

## Vòng lặp cho

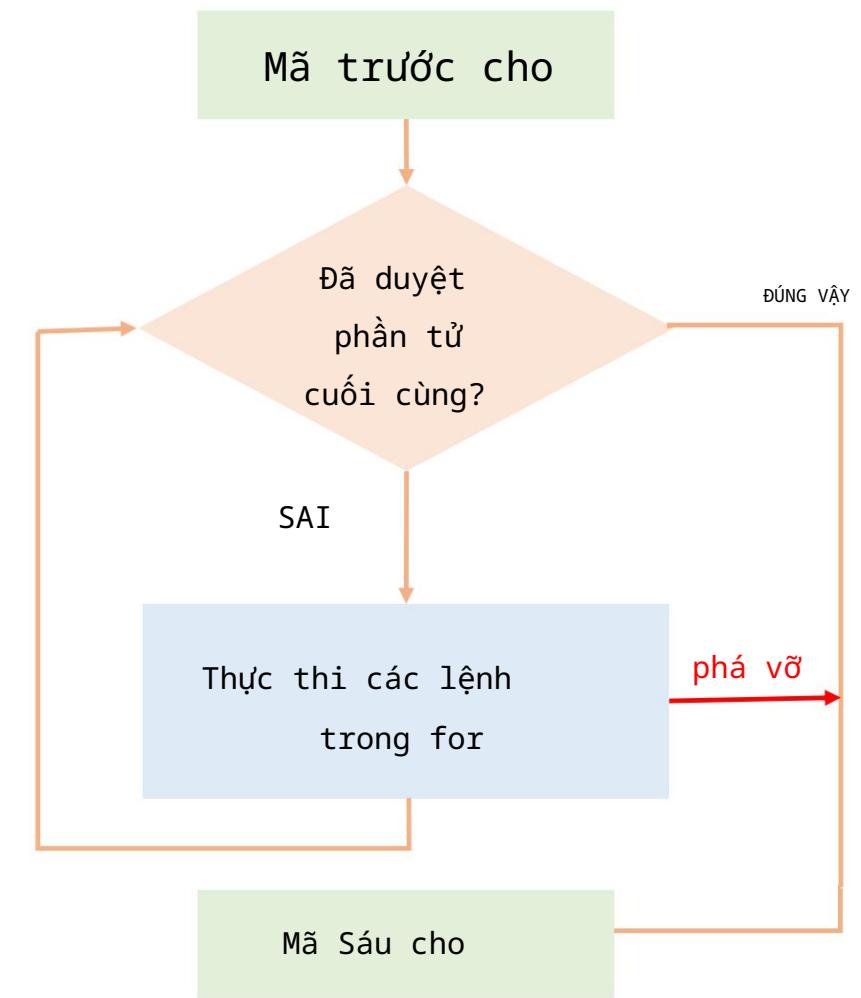
### phá vỡ từ khóa

```

1 # duyệt phần tử trong range(10)
2 for i in range(10):
3     # hỏi phần tử i có bằng 5 không?
4     if i == 5:
5         # nếu bằng thì thoát vòng lặp for này
6         break
7
8     # làm gì đó với i
9     print('Giá trị i là', i)

```

Giá trị i là 0  
 Giá trị i là 1  
 Giá trị i là 2  
 Giá trị i là 3  
 Giá trị i là 4





# .Cấu trúc vòng lặp

## Vòng lặp cho

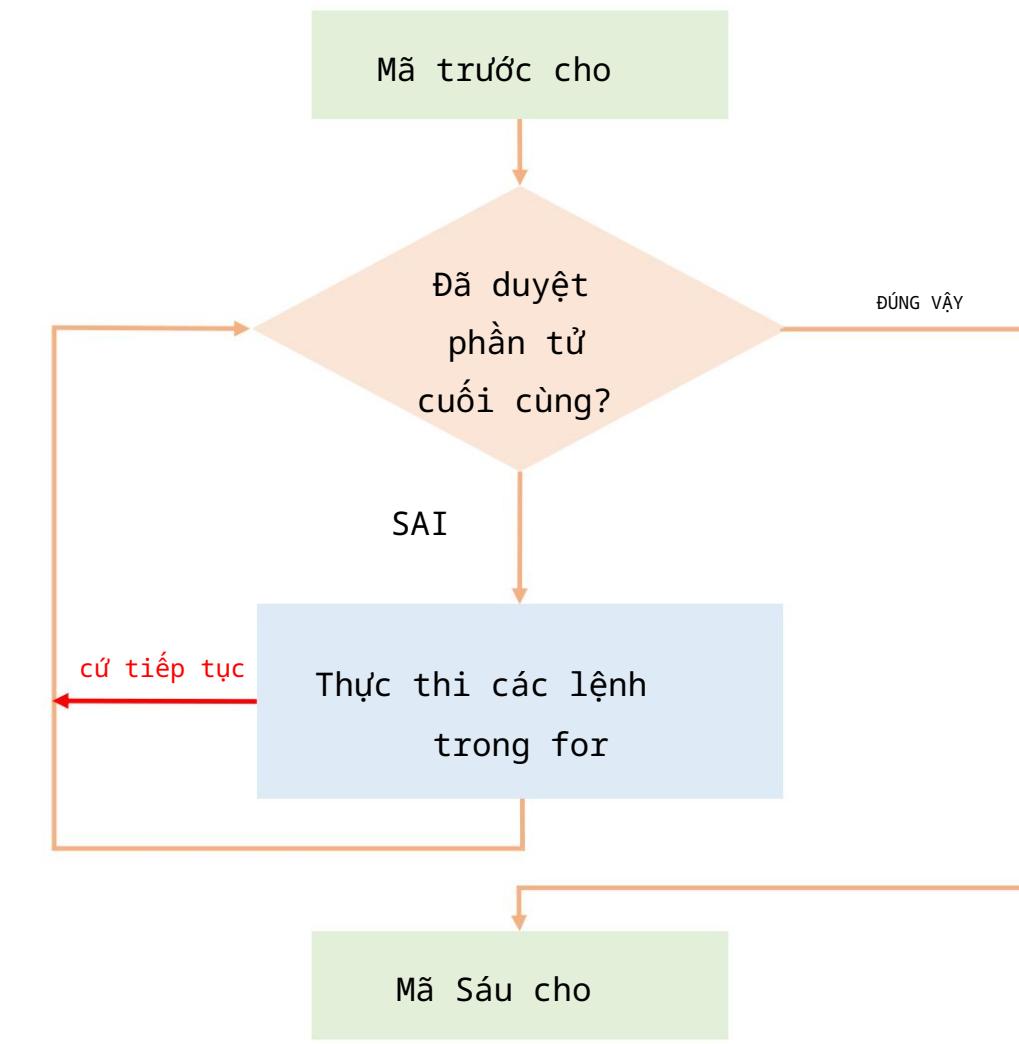
### tiếp tục từ khóa

```

1. # duyệt phần tử trong range(10)
2. for i in range(10):
3.     # hỏi phần tử i có bằng 5 không?
4.     if i == 5:
5.         # nếu bằng thì gọi continue
6.         # phần code sau continue sẽ không
7.         # được thực thi trong lần lặp này
8.         continue
9.
10.    # làm gì đó với i
11.    print('Giá trị i là', i)

```

Giá trị i là 0  
 Giá trị i là 1  
 Giá trị i là 2  
 Giá trị i là 3  
 Giá trị i là 4  
 Giá trị i là 6  
 Giá trị i là 7  
 Giá trị i là 8  
 Giá trị i là 9



## Cấu trúc vòng lặp

### Ví dụ - Mô phỏng việc tung đồng xu

Sự kiện: bất kỳ sự kiện nào xảy ra

Thử nghiệm: thử nghiệm cho kết quả

Không gian mẫu: tập hợp tất cả các kết quả có thể xảy ra của một thử nghiệm

Biến ngẫu nhiên: không có vấn đề gì, nhưng kết quả của sự kiện này

Xác thực

$$(sự kiện) = \frac{|sự kiện|}{||}$$



Thí nghiệm: Ném đồng xu

Không gian mẫu = Đầu, Đuôi }

Event: Gọi A là sự kiện mà kết quả của một lần ném là head

( ) Xác thực để sự kiện xảy ra

Tiền xu có 2 mặt  
(đầu và đuôi)

Xác định kết quả của một lần ném là head

A={đầu}

S={đầu, đuôi}

$$( ) = \frac{1}{||} \quad \frac{1}{2}$$

# .Cấu trúc vòng lặp

## Ví dụ - Mô phỏng việc tung đồng xu



Đếm #đầu

Đếm #đuôi

Kiểm tra xem hai số  
có giống nhau không

```

1.     # aivietnam.ai
2.     import random
3.
4.     # Tổng số lần búng đồng xu
5.     total_flips = 0
6.
7.     # số lần mặt sau xuất hiện
8.     num_tails = 0
9.
10.    # số lần mặt trước xuất hiện
11.    num_heads = 0
12.
13.    for _ in range(1000):
14.        # sinh số ngẫu nhiên nằm trong khoảng [0,1)
15.        n = random.random()
16.        if n < 0.5:
17.            num_tails = num_tails + 1
18.        else:
19.            num_heads = num_heads + 1
20.
21.        # code ở vị trí này không thuộc khối else
22.        total_flips = total_flips + 1

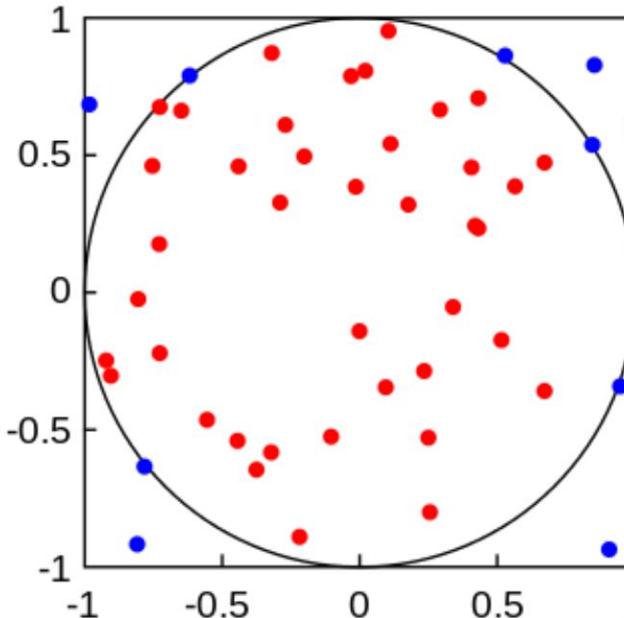
```



## .Cấu trúc vòng lặp

### Ví dụ - Ước tính PI

hình từ  
wikipedia



là #mẫu ngẫu nhiên trong ô vuông  
được tạo theo phân bố đồng đều

là #mẫu ngẫu nhiên trong vòng tròn  
được tạo theo phân bố đồng đều

$$\text{bán kính hình tròn} = 1$$

$$\text{khu vực vòng tròn} = \pi$$

$$\text{cạnh hình vuông} = 2$$

$$\text{diện tích hình vuông} = 4$$

$$— \approx —$$

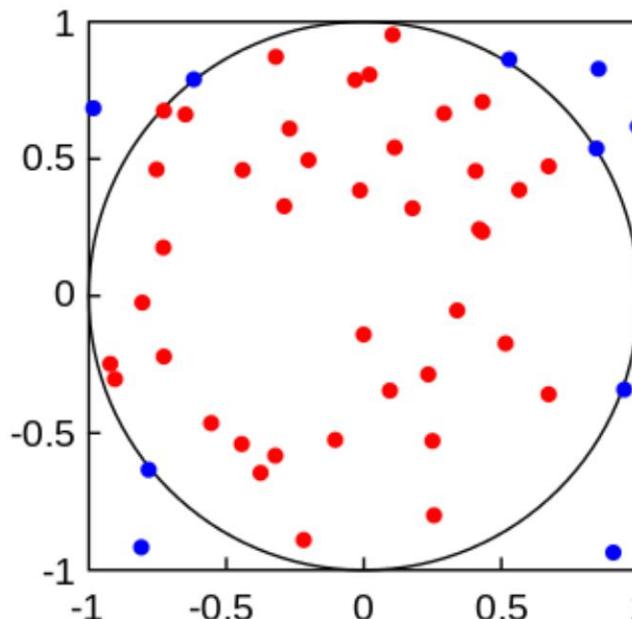
$$\frac{\pi}{4} \approx$$

$$\approx —$$



## .Cấu trúc vòng lặp

### Ví dụ - Ước tính PI



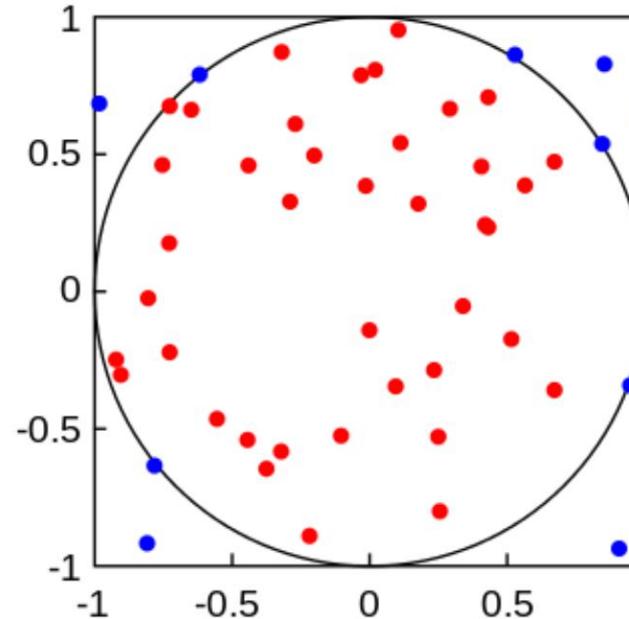
$\approx$  —

Số ngẫu nhiên  $[-1, 1]$

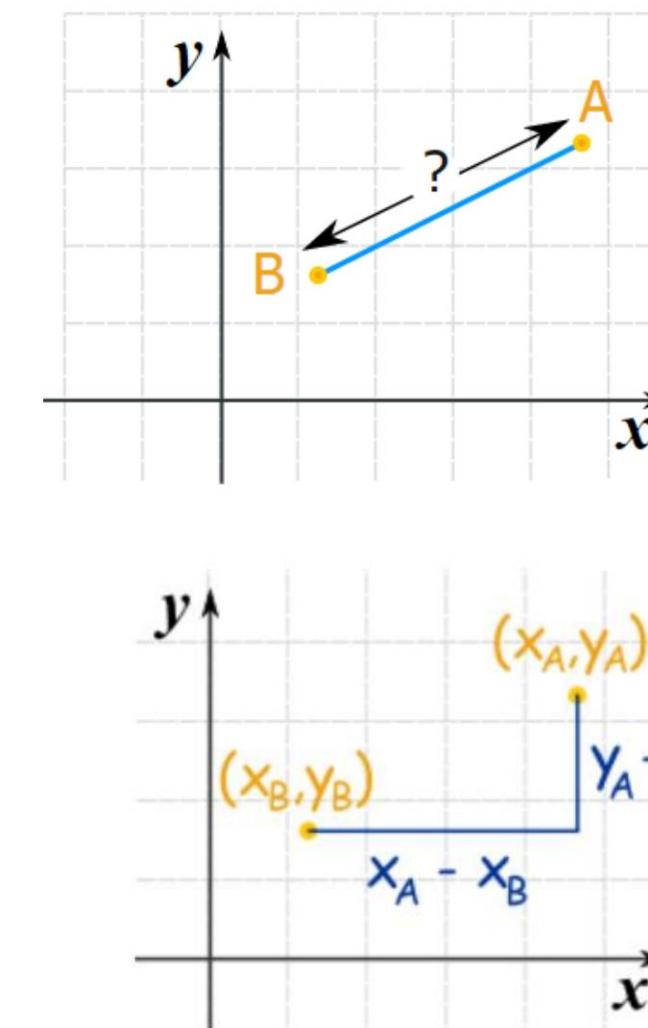
Kiểm tra xem một điểm có nằm trong vòng tròn không



## Ví dụ - Ước tính PI

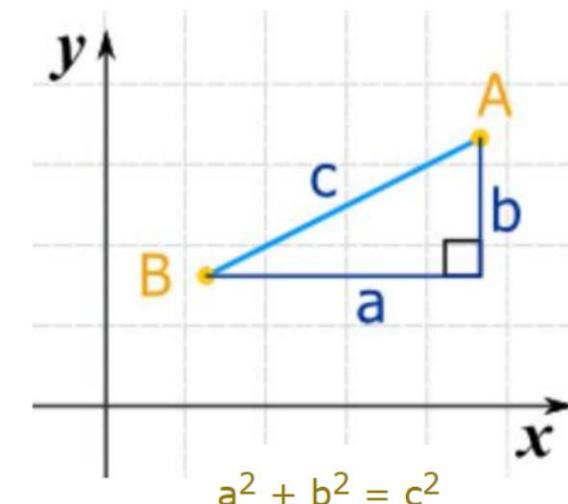


$\approx$  —



<https://www.mathsisfun.com/algebra/distance-2-points.html>

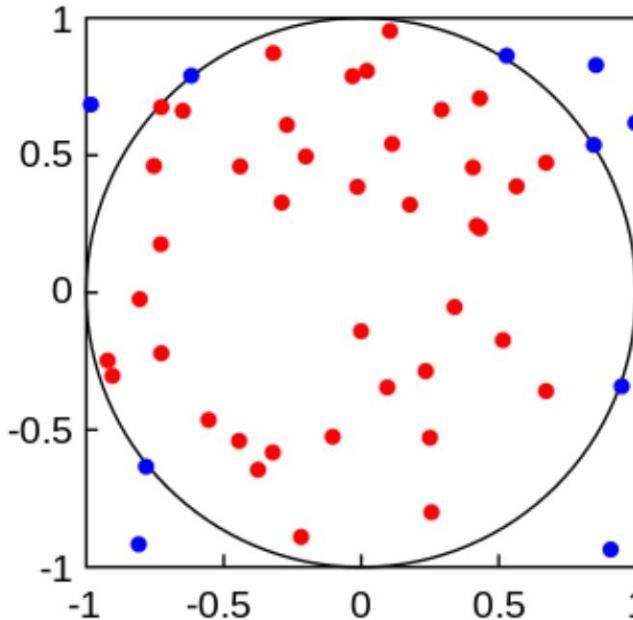
## Cấu trúc vòng lặp



$$c = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$$



## Ví dụ - Ước tính PI



$\approx$  —

```

1. # aivietnam.ai
2. import random
3. import math
4.
5.
6. # Tổng số điểm p được sinh ra
7. N = 100000
8.
9. # số điểm thuộc hình tròn
10. N_T = 0
11.
12. # Sinh ra N điểm ngẫu nhiên
13. for i in range(N):
14.     # sinh ra x, y thuộc [-1, 1].
15.     x = random.random()**2 - 1
16.     y = random.random()**2 - 1
17.
18.     x2 = x**2
19.     y2 = y**2
20.
21.     # kiểm tra p có nằm trong đường tròn
22.     if math.sqrt(x2 + y2) <= 1.0:
23.         N_T = N_T + 1
24.
25. # tính PI
26. pi = (N_T / N) * 4
27. print(pi)

```

.Cấu trúc vòng lặp



## .Cấu trúc vòng lặp

### Ví dụ - Số Euler

= 2,71828

công thức

$$e \approx 1 + \frac{1}{1!} + \frac{1}{2!} + \dots + \frac{1}{n!}$$



1. Tính giải thừa
2. Tính toán



## Ví dụ - Số Euler

= 2,71828

công thức

$$e \approx 1 + \frac{1}{1!} + \frac{1}{2!} + \dots + \frac{1}{n!}$$

## .Cấu trúc vòng lặp

```

1.      # aivietnam.ai
2.
3.      # hàm tính giai thừa
4.      def factorial(n):
5.          result = 1
6.
7.          for i in range(2, n+1):
8.              result = result*i
9.
10.         return result
11.
12.     # hàm ước lượng số e
13.     def estimate_e(n):
14.         result = 1
15.
16.         for i in range(1, n+1):
17.             result = result + 1/factorial(i)
18.
19.         return result
20.
21.     # ước lượng số e với n = 10
22.     print(estimate_e(10))

```

## .Cấu trúc vòng lặp

### Ví dụ - Tính căn bậc hai của số N

Phương pháp Newton

Đặt giá trị cho  $\theta_0$ ;  $n = 0$   
 $(\theta_0 = N/2)$

$$+1 = \frac{+ -}{2}$$

$n = n + 1$

$\sqrt{\quad}$

= 9

$\theta_0 = \frac{9}{2} = 4,5$

= 0

= 0

$$\theta_1 = \frac{\theta_0 + \frac{N}{\theta_0}}{2} = \frac{9 + \frac{4,5}{4,5}}{2} = \frac{6,5}{2} = 3,25$$

= 1

$$\theta_2 = \frac{\theta_1 + \frac{N}{\theta_1}}{2} = \frac{9 + \frac{3,25}{3,25}}{2} = \frac{6,019}{2} = 3,009$$

= 2

$$\theta_3 = \frac{\theta_2 + \frac{N}{\theta_2}}{2} = \frac{9 + \frac{3,009}{3,009}}{2} = \frac{3,00001}{2} = 3,00001$$

## Ví dụ - Tính căn bậc hai của số N

Phương pháp Newton

Đặt giá trị cho 0;  $n = 0$   
 $(\theta = N/2)$



$$+1 = \frac{+ -}{2}$$



$n = n + 1$

```

1 def compute_square_root(N, num_loops):
2     """
3         This function aims to compute square root for the number N
4
5         N -- the number needs to take the square root
6         num_loops -- number of loops used for this optimization
7     """
8
9     x_n = N/2.0
10
11    for i in range(num_loops):
12        x_np1 = (x_n + N/x_n) / 2.0
13        x_n = x_np1
14
15    return x_np1
16
17 print(compute_square_root(N=9, num_loops=10))
18 print(compute_square_root(N=2, num_loops=10))

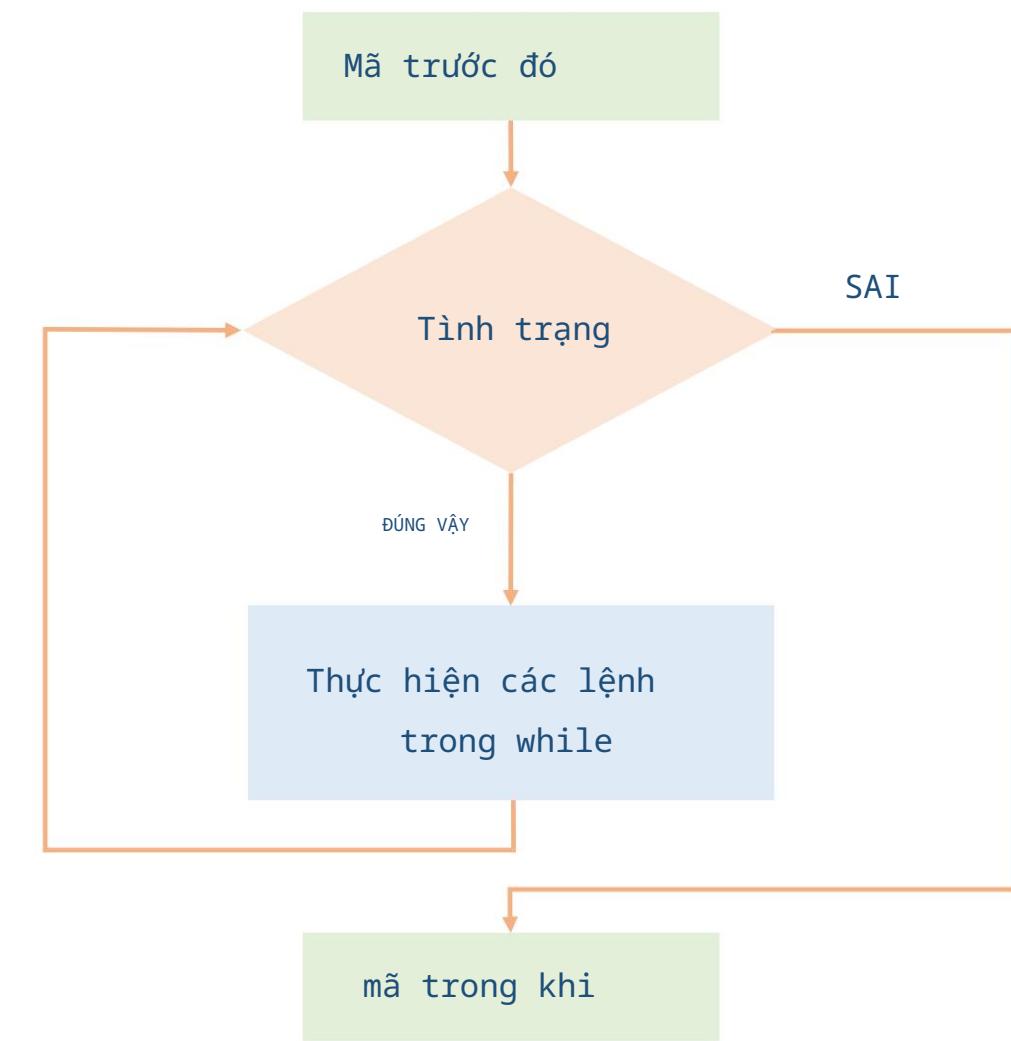
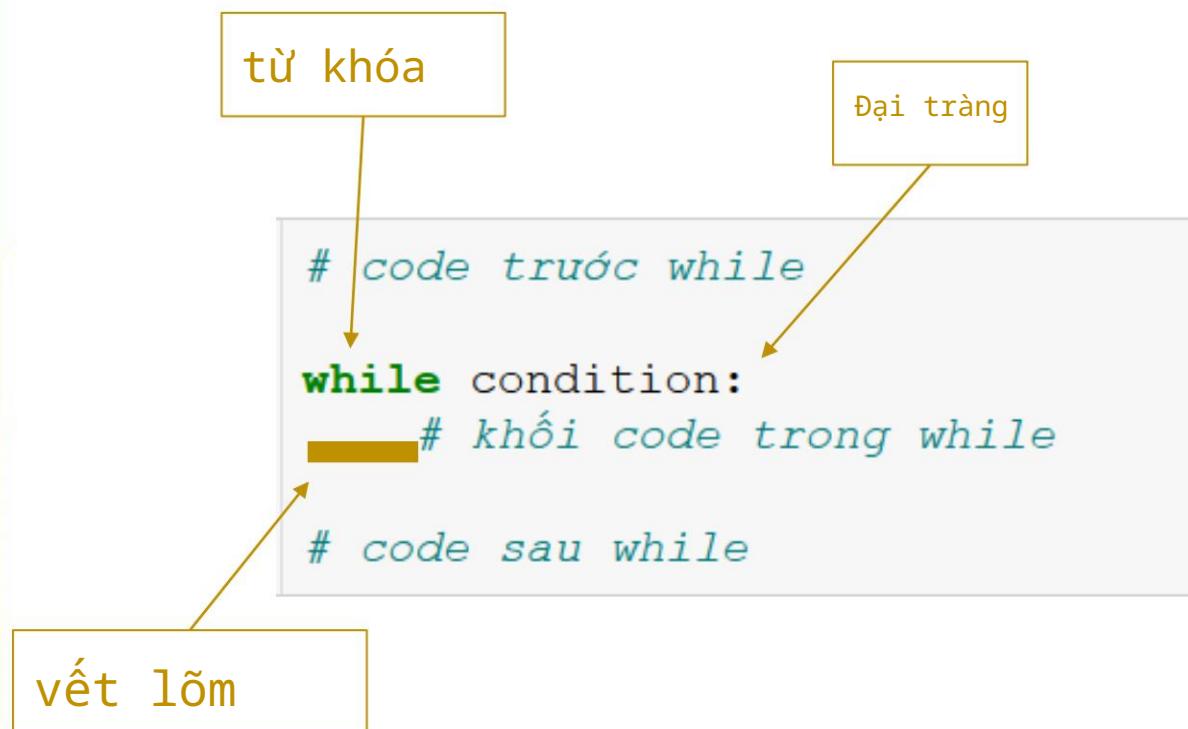
```

3.0  
1.414213562373095



## .Cấu trúc vòng lặp

Trong khi lặp lại



# .Cấu trúc vòng lặp

Trong khi lặp lại

```

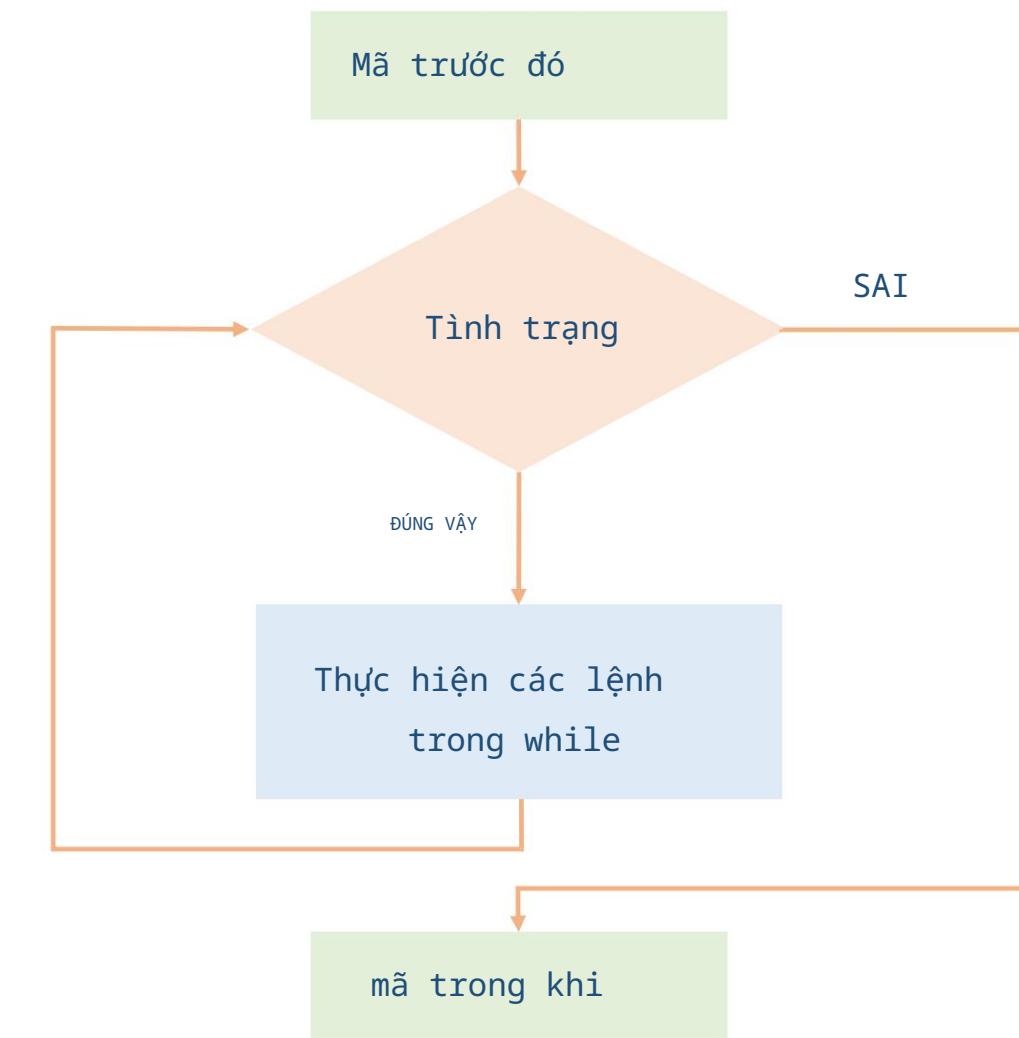
1 # tạo biến i
2 i = 0
3
4 # bắt đầu vòng lặp while
5 while i<5:
6     # code inside while
7     print(i)
8     i = i + 1
9
10 print('Phần code này khi đã thoát while')

```

```

0
1
2
3
4
Phần code này khi đã thoát while

```





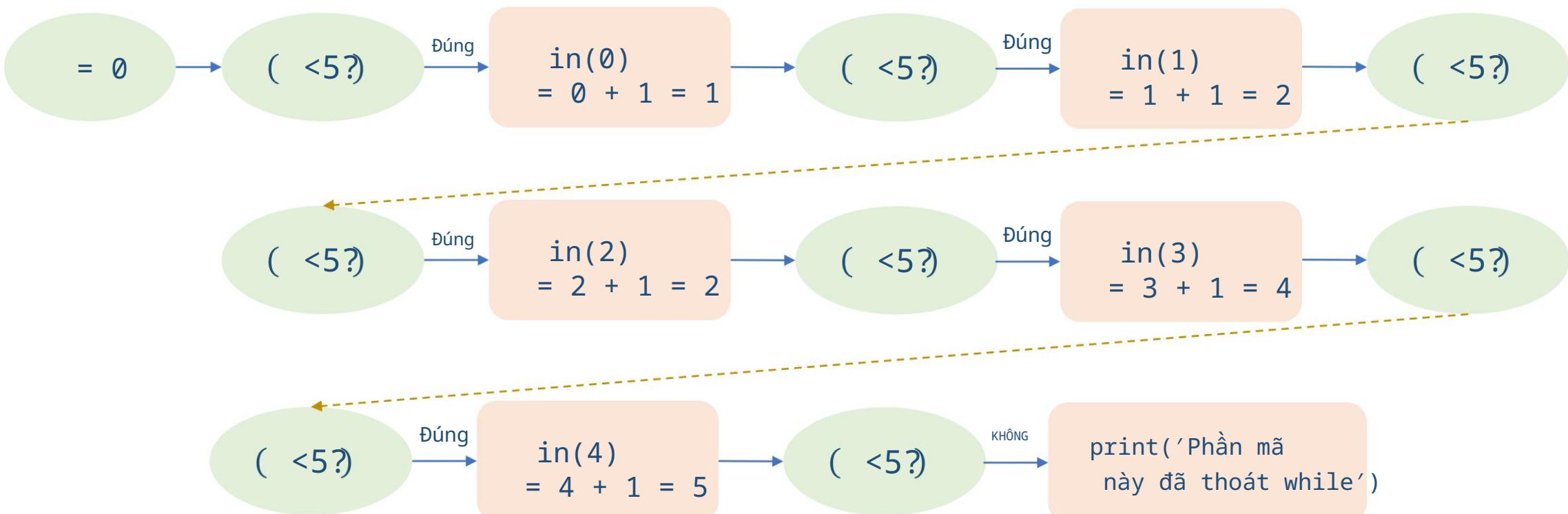
# Cấu trúc vòng lặp

Trong khi lặp lại

```

1 # tạo biến i
2 i = 0
3
4 # bắt đầu vòng lặp while
5 while i<5:
6     # code inside while
7     print(i)
8     i = i + 1
9
10 print('Phần code này khi đã thoát while')

```





## .Cấu trúc vòng lặp

Trong khi lặp lại

while-True-break

```

1. import random
2.
3. # cho vòng lặp chạy vô tận
4. while True:
5.     # sinh số ngẫu nhiên
6.     num = random.randint(0,10)
7.     print('Số sinh ra có giá trị là', num)
8.
9.     # kiểm tra num có bằng 5 hay không?
10.    if num == 5:
11.        # nếu có thì thoát khỏi while
12.        break;
13.    print('Đã thoát khỏi while')

```

```

Số sinh ra có giá trị là 4
Số sinh ra có giá trị là 3
Số sinh ra có giá trị là 8
Số sinh ra có giá trị là 1
Số sinh ra có giá trị là 0
Số sinh ra có giá trị là 5
Đã thoát khỏi while

```



## VUI MỪNG

- Giới thiệu
- Biến và toán tử
- Chức năng
- Cấu trúc lựa chọn
- Cấu trúc vòng lặp
- Các lỗi thường gặp



## Lỗi 1

```

2.     # Lỗi sử dụng biến chưa được khai báo
3.
4.     # khai báo biến a = 5
5.     a = 5
6.
7.     # thực hiện a + b, sau đó lưu vào biến c
8.     c = a + b
9.
10.    # in giá trị c
11.    print(c)

```

---

NameError Traceback (most recent call last)

```

<ipython-input-1-eae96ee94f9f> in <module>
      6
      7 # thực hiện a + b, sau đó lưu vào biến c
----> 8 c = a + b
      9
     10 # in giá trị c

NameError: name 'b' is not defined

```



## Lỗi 2

## Lỗi thông thường

```
2.     # Lỗi viết sai tên hàm
3.
4.     # khai báo biến a = 5
5.     a = 5
6.
7.     # in giá trị a
8.     Print(a)
```

```
NameError                                                 Traceback (most recent call last)
<ipython-input-2-f09db6b2bf7e> in <module>
      6
      7 # in giá trị a
----> 8 Print(a)

NameError: name 'Print' is not defined
```



## Lỗi 3

## Lỗi thông thường

```
2. # Lỗi khai báo chuỗi (string) sai
3.
4. # khai báo biến chuỗi s
5. s = 'Hello AIVIETNAM'
6.
7. # in giá trị s
8. print(s)
```

```
File "<ipython-input-3-96feed73c6b1>", line 5
    s = 'Hello AIVIETNAM'
          ^

```

```
SyntaxError: EOL while scanning string literal
```



## Lỗi 4

## Lỗi thông thường

```

2. # Lỗi chia cho số 0
3.
4. # khai báo biến a và b
5. a = 5
6. b = 0
7.
8. # tính giá trị c bằng a chia cho b
9. c = a / b
10.
11. # in giá trị c
12. print(c)

```

```

-----
ZeroDivisionError                                Traceback (most recent call last)
<ipython-input-4-298e1112d534> in <module>
      7
      8 # tính giá trị c bằng a chia cho b
----> 9 c = a / b
     10
     11 # in giá trị c

ZeroDivisionError: division by zero

```



## Lỗi 5

### Lỗi thông thường

```

2. # Lỗi dùng phép toán không phù hợp với kiểu dữ liệu
3.
4. # khai báo biến chuỗi s
5. s = 'AI'
6.
7. # khai báo biến n có kiểu integer
8. n = 5
9.
10. # tính giá trị c
11. c = s + n
12.
13. # in giá trị c
14. print(c)

```

---

**TypeError** Traceback (most recent call last)

```

<ipython-input-5-fle2455fae51> in <module>
      9
     10 # tính giá trị c
--> 11 c = s + n
     12
     13 # in giá trị c

TypeError: must be str, not int

```



## Lỗi thông thường

### Lỗi 6

```

2. # Lỗi dùng phép toán không phù hợp với kiểu dữ liệu (2)
3.
4. # khai báo biến chuỗi s
5. s = 'AI'
6.
7. # khai báo biến n có kiểu integer
8. n = 5
9.
10. # tính giá trị c
11. c = n + s
12.
13. # in giá trị c
14. print(c)

```

**TypeError**

Traceback (most recent call last)

```

<ipython-input-8-cecd5289546d> in <module>
      9
     10 # tính giá trị c
--> 11 c = n + s
     12
     13 # in giá trị c

```

**TypeError:** unsupported operand type(s) for +: 'int' and 'str'



## Lỗi 7

## Lỗi thông thường

```
2.      # Lỗi indentation
3.
4.      # khai báo biến a và b
5.      a = 5
6.      b = 6
7.
8.      # thực hiện a + b, sau đó lưu vào biến c
9.      c = a + b
10.
11.     # in giá trị c
12.     print(c)
```

```
File "<ipython-input-7-31f64166c395>", line 6
    b = 6
^
```

**IndentationError:** unexpected indent



## Lỗi 8

## Lỗi thông thường

```
2. # Lỗi thiếu ngoặc đóng/mở
3.
4. import math
5.
6. number = 20.2
7. print(math.floor(number))
8. print(math.pi)
```

```
File "<ipython-input-15-920005110c33>", line 8
    print(math.pi)
               ^
SyntaxError: invalid syntax
```



## Lỗi 9

## Lỗi thông thường

```
2. # Lỗi thiếu ngoặc khi dùng print  
3.  
4. print "aivietnam.ai"
```

```
File "<ipython-input-3-a46b1c9e05ed>", line 4  
    print "aivietnam.ai"  
          ^  
SyntaxError: Missing parentheses in call to 'print'. Did you mean print("aivietnam.ai")?
```



## Lỗi 10

## Lỗi thông thường

```
2. # Lỗi khai báo module không tồn tại
3.
4. import mymodule
5.
6. print("aivietnam.ai")
```

```
-----
ModuleNotFoundError                                     Traceback (most recent call last)
<ipython-input-5-1b242c59080b> in <module>
      2 # Lỗi khai báo module không tồn tại
      3
----> 4 import mymodule
      5
      6 print("aivietnam.ai")

ModuleNotFoundError: No module named 'mymodule'
```



## Lỗi 11

## Lỗi thông thường

```
2.     # Lỗi out-of-range  
3.  
4.     l = [1, 2, 3, 4, 5]  
5.     print(l[0])  
6.     print(l[5])
```

```
-----  
IndexError                                                 Traceback (most recent call last)  
<ipython-input-6-b8c53176edc0> in <module>  
      4 l = [1, 2, 3, 4, 5]  
      5 print(l[0])  
----> 6 print(l[5])  
  
IndexError: list index out of range
```



## Lỗi 12

## Lỗi thông thường

```
2.     # Lỗi out-of-range  
3.  
4.     name = "aivietname.ai"  
5.     print(name[0])  
6.     print(name[50])
```

```
-----  
IndexError                                                 Traceback (most recent call last)  
<ipython-input-7-357ff533411a> in <module>  
      4 name = "aivietname.ai"  
      5 print(name[0])  
----> 6 print(name[50])  
  
IndexError: string index out of range
```



## Lỗi 13

## Lỗi thông thường

```
2.     # Lỗi key không tồn tại  
3.  
4.     dict = {'1':'Python', '5':'C++'}  
5.     print(dict['1'])  
6.     print(dict['5'])  
7.     print(dict['2'])
```

```
-----  
KeyError                                                 Traceback (most recent call last)  
<ipython-input-11-cd160b2a5e34> in <module>  
      5     print(dict['1'])  
      6     print(dict['5'])  
----> 7     print(dict['2'])  
  
KeyError: '2'
```



## Lỗi 14

## Lỗi thông thường

```
2.     # Lỗi thiếu dấu :
3.
4.     number = 15
5.     if number < 10
6.         print("A small number")
7.     else:
8.         print("A large number")
```

```
File "<ipython-input-15-fedf173614ac>", line 5
```

```
if number < 10
    ^
```

```
SyntaxError: invalid syntax
```



## Lỗi 15

## Lỗi thông thường

```
2. # Lỗi thiếu dấu :
3.
4. number = 15
5. if number < 10:
6.     print("A small number")
7. else
8.     print("A large number")
```

```
File "<ipython-input-16-699752908646>", line 7
  else
  ^
SyntaxError: invalid syntax
```



## Lỗi 16

## Lỗi thông thường

```
2. # Lỗi thay đổi giá trị của một tuple  
3.  
4. t = (1, 2, 3, 4)  
5. t[2] = 2
```

---

**TypeError**

```
<ipython-input-1-24607f50b9ee> in <module>  
      3  
      4 t = (1, 2, 3, 4)  
----> 5 t[2] = 2
```

**Traceback (most recent call last)**

```
TypeError: 'tuple' object does not support item assignment
```



## Lỗi 17

## Lỗi thông thường

```
2. # Lỗi cố gắng lấy phần tử từ từ dictionary rỗng
3.
4. dict = {'1':'Python', '5':'C++'}
5. item1 = dict.popitem()
6. item2 = dict.popitem()
7. item3 = dict.popitem()
```

```
-----
KeyError                                                 Traceback (most recent call last)
<ipython-input-6-40b67d40ba0f> in <module>
      5 item1 = dict.popitem()
      6 item2 = dict.popitem()
----> 7 item3 = dict.popitem()

KeyError: 'popitem(): dictionary is empty'
```



## Lỗi 18

## Lỗi thông thường

```
1. import math
2.
3. number = -4
4. print(math.sqrt(number))
```

```
-----
ValueError                                                 Traceback (most recent call last)
<ipython-input-8-f25b4b744f6e> in <module>
      2
      3 number = -4
----> 4 print(math.sqrt(number))

ValueError: math domain error
```



## Lỗi 19

## Lỗi thông thường

```
2.     # Lỗi đọc file không tồn tại  
3.  
4.     my_file = open("file.txt", "r")  
5.     print(my_file)
```

```
-----  
FileNotFoundError                      Traceback (most recent call last)  
<ipython-input-13-73d8e6dda2db> in <module>  
      2 # Lỗi đọc file không tồn tại  
      3  
----> 4 my_file = open("file.txt", "r")  
      5 print(my_file)  
  
FileNotFoundError: [Errno 2] No such file or directory: 'file.txt'
```



## Lỗi 20

## Lỗi thông thường

```
2. # Lỗi import thư viện chưa cài đặt
3.
4. # import pytorch
5. import torch
6.
7. print(torch.cuda.is_available())
```

```
-----
ModuleNotFoundError                                     Traceback (most recent call last)
<ipython-input-14-680f8ea2b256> in <module>
      1 # import thư viện pytorch
----> 2 import torch
      3
      4 print(torch.cuda.is_available())

ModuleNotFoundError: No module named 'torch'
```



## Lỗi 21

```
2. # Lỗi tìm vị trí của một string không có trong một string khác
3.
4. my_string = "Đây là bài học của AI VIETNAM"
5. my_string.index("hello")
```

```
-----
ValueError                                                 Traceback (most recent call last)
<ipython-input-1-9d5c7da27233> in <module>
      3
      4 my_string = "Đây là bài học của AI VIETNAM"
----> 5 my_string.index("hello")

ValueError: substring not found
```



## Lỗi 22

## Lỗi thông thường

```
2 # Lỗi sử dụng biến ngoài phạm vi (variable scope)
3 def a_function(x):
4     a_variable = 4
5     result = x*a_variable
6
7     return result
8
9 print(a_variable)
```

```
NameError Traceback (most recent call last)
<ipython-input-43-2cce0357dce7> in <module>
      6
      7
----> 8 print(a_variable)

NameError: name 'a_variable' is not defined
```



## Lỗi 23

### Lỗi thông thường

```

2 # Lỗi thay đổi giá trị của biến global
3
4 index = 5
5 def my_function():
6     index += 1
7     print(index)
8
9 my_function()

```

```

-----
UnboundLocalError                                     Traceback (most recent call last)
<ipython-input-4-63ee67900288> in <module>
      7     print(index)
      8
----> 9 my_function()

<ipython-input-4-63ee67900288> in my_function()
      4 index = 5
      5 def my_function():
----> 6     index += 1
      7     print(index)
      8

UnboundLocalError: local variable 'index' referenced before assignment

```



## Lỗi 24

## Lỗi thông thường

```
2 # Lỗi nối string và number
3
4 a_number = 5
5 a_string = 'value '
6 result = a_string + a_number
7
8 print(result)
```

---

**TypeError**

Traceback (most recent call last)

```
<ipython-input-7-9772d690ff0d> in <module>
      4 a_number = 5
      5 a_string = 'value '
----> 6 result = a_string + a_number
      7
      8 print(result)
```

**TypeError:** can only concatenate str (not "int") to str



## Lỗi 25

## Lỗi thông thường

```
2 # Lỗi ép kiểu string không phải số
3
4 str1 = '5'
5 str2 = 'hello'
6
7 value1 = int(str1)
8 value2 = int(str2)
```

```
-----
ValueError                                                 Traceback (most recent call last)
<ipython-input-9-1f0b23b26eb1> in <module>
      6
      7 value1 = int(str1)
----> 8 value2 = int(str2)

ValueError: invalid literal for int() with base 10: 'hello'
```



## Lỗi 26

```

1 # aivietnam.ai
2 # Lỗi stack-over-flow
3
4 def a_function(n):
5     return a_function(n)
6
7 a_function(5)

```

```

-----
RecursionError                                     Traceback (most recent call last)
<ipython-input-10-bda7ef50bf68> in <module>
      5     return a_function(n)
      6
----> 7 a_function(5)

<ipython-input-10-bda7ef50bf68> in a_function(n)
      3
      4 def a_function(n):
----> 5     return a_function(n)
      6
      7 a_function(5)

... last 1 frames repeated, from the frame below ...

<ipython-input-10-bda7ef50bf68> in a_function(n)
      3
      4 def a_function(n):
----> 5     return a_function(n)
      6
      7 a_function(5)

RecursionError: maximum recursion depth exceeded

```

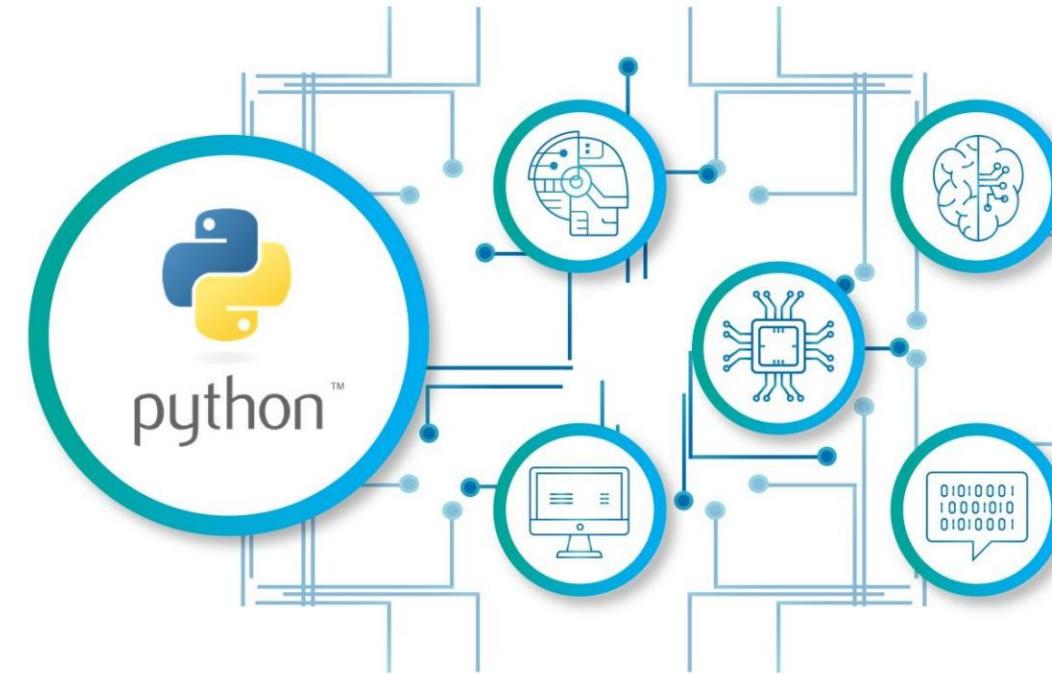


## Python cơ bản





# Python cho AI



Cảm ơn You...!