

AI VIETNAM
All-in-One Course

Python

An Introduction

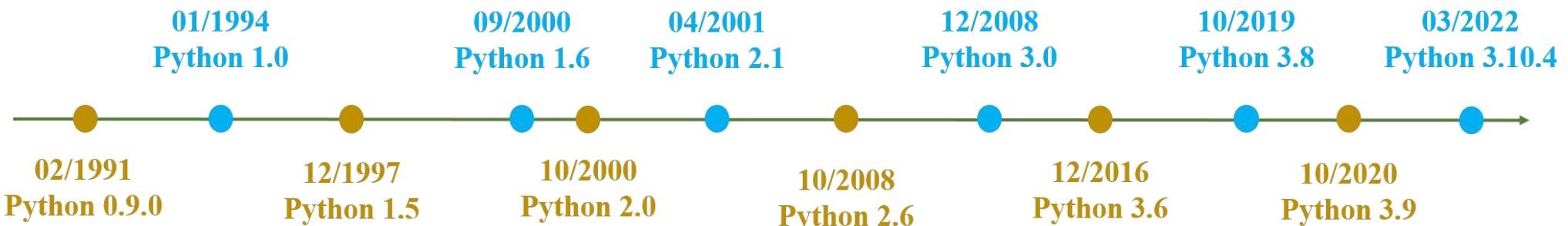
Quang-Vinh Dinh
Ph.D. in Computer Science

Outline

- Python Environment
- Basic Data Types
- Functions
- Conditions
- Common Errors

Introduction

❖ Python history



Ý tưởng từ 1980s



Được đặt tên theo
nhóm hài Monty Python



Bắt đầu cài đặt
từ 12/1989



Guido van Rossum



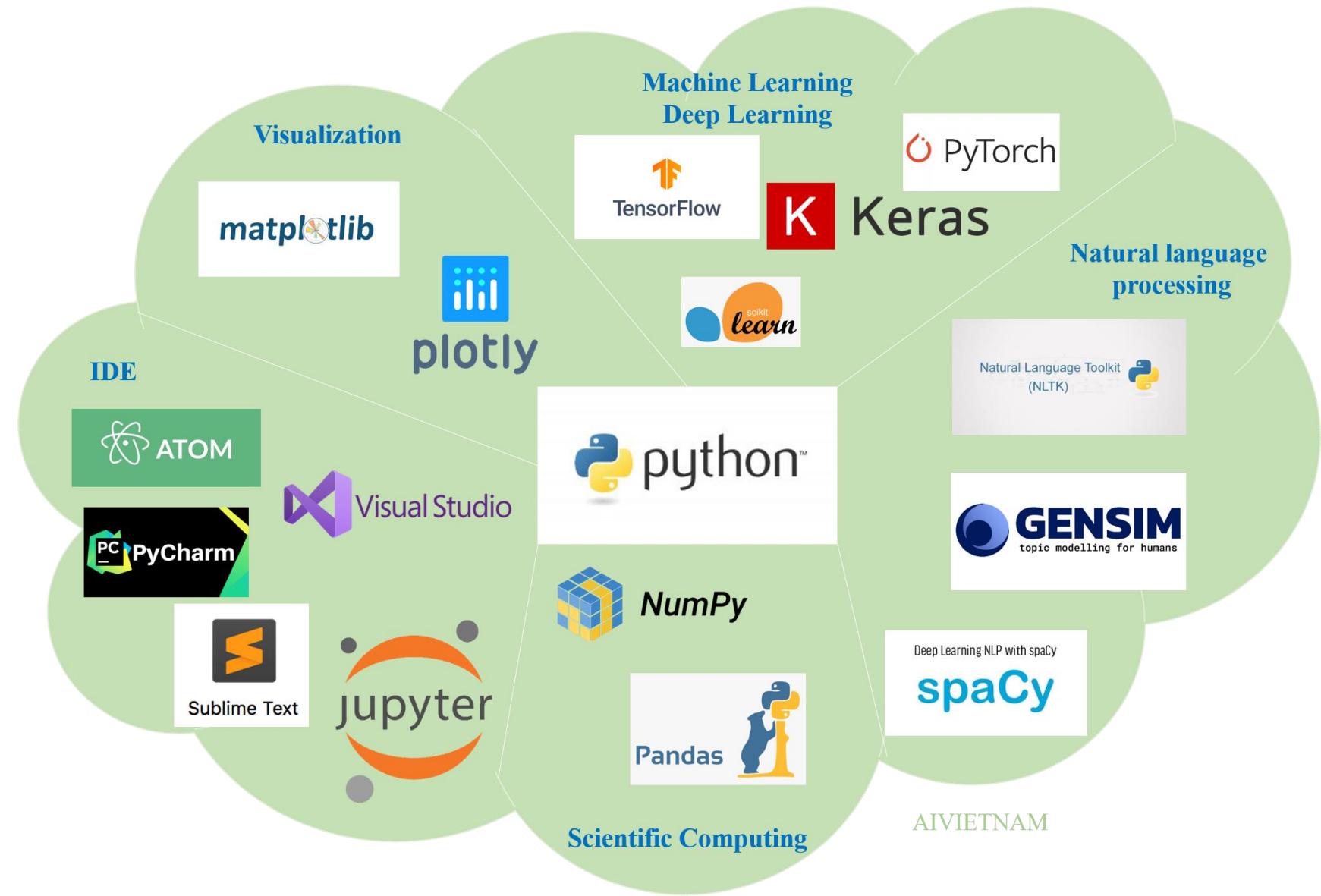
228,855 python packages
(PyPI)



Hỗ trợ rất mạnh cho
Data Science và Machine Learning

Introduction

❖ Python ecosystem



Introduction

❖ Top Programming Languages (2021)

Choose a Ranking

IEEE Spectrum Trending

Jobs Open Custom

[Create custom ranking](#)

Language Types

Web  Enterprise 

Mobile  Embedded 

(Click to hide)

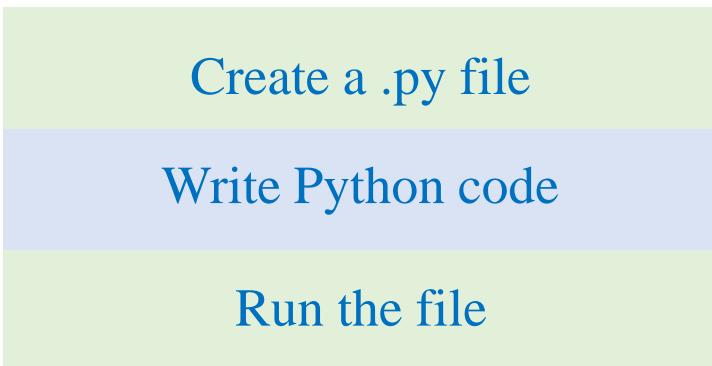
Language Ranking: **IEEE Spectrum**

<https://spectrum.ieee.org/top-programming-languages/>

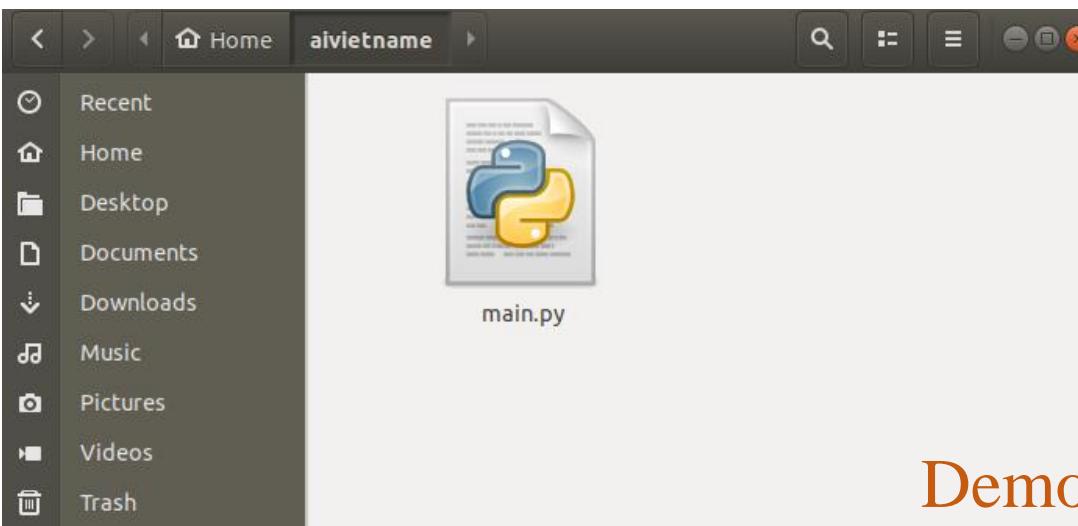
Rank	Language	Type	Score
1	Python 	  	100.0
2	Java 	  	95.4
3	C 	  	94.7
4	C++ 	  	92.4
5	JavaScript 		88.1
6	C# 	   	82.4
7	R 		81.7
8	Go 	 	77.7
9	HTML 		75.4
10	Swift 	 	70.4

Introduction

❖ First Python program



```
1 print("My first program !!!")
```

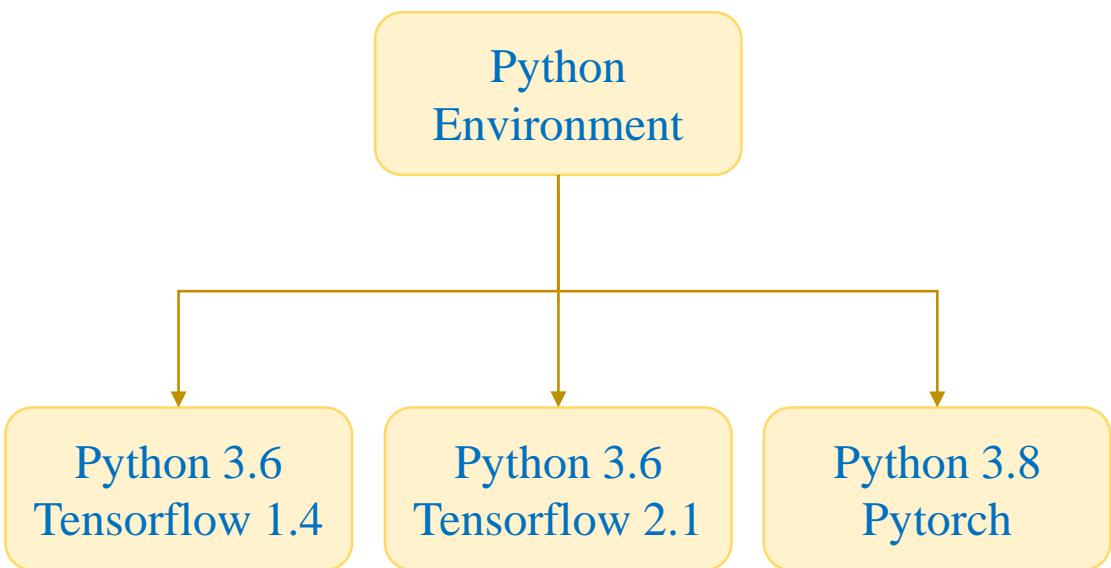


Demo

```
quangvinhproject@quangvinhproject-desktop:~/aivietname$ cd aivietname/
quangvinhproject@quangvinhproject-desktop:~/aivietname$ python main.py
My first program !!!
quangvinhproject@quangvinhproject-desktop:~/aivietname$
```

Introduction

❖ Virtual environment



```
sudo apt-get install -y python3-venv
```

```
python3 -m venv a_name
```

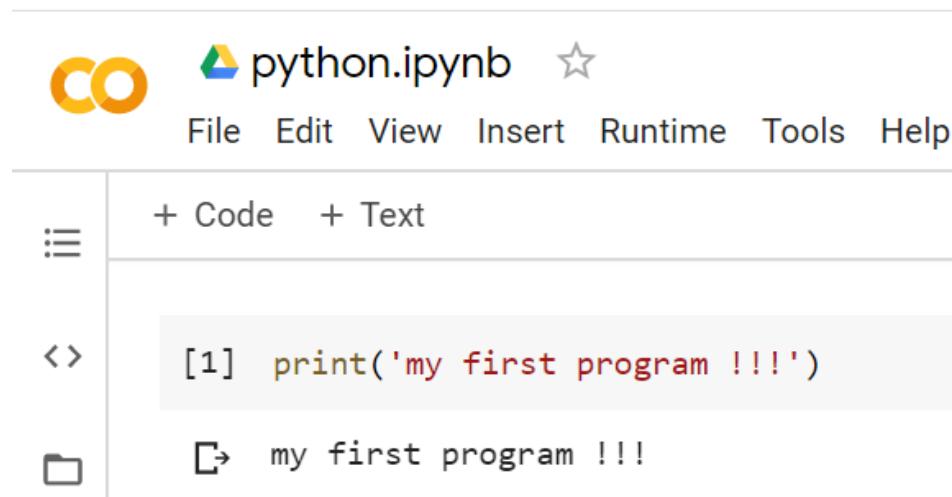
```
source a_name/bin/activate
```

```
server2@server2: ~
File Edit View Search Terminal Help
server2@server2:~$ 
server2@server2:~$ 
server2@server2:~$ 
server2@server2:~$ source vinh_env/bin/activate
(vinh_env) server2@server2:~$ python
Python 3.6.9 (default, Jan 26 2021, 15:33:00)
[GCC 8.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print('hello')
hello
>>> quit()
(vinh_env) server2@server2:~$ deactivate
server2@server2:~$ 
server2@server2:~$ 
server2@server2:~$ 
server2@server2:~$ 
server2@server2:~$ 
```

Introduction

- ❖ First Python program
- ❖ Using Google Colab

<https://colab.research.google.com/notebooks/intro.ipynb#recent=true>

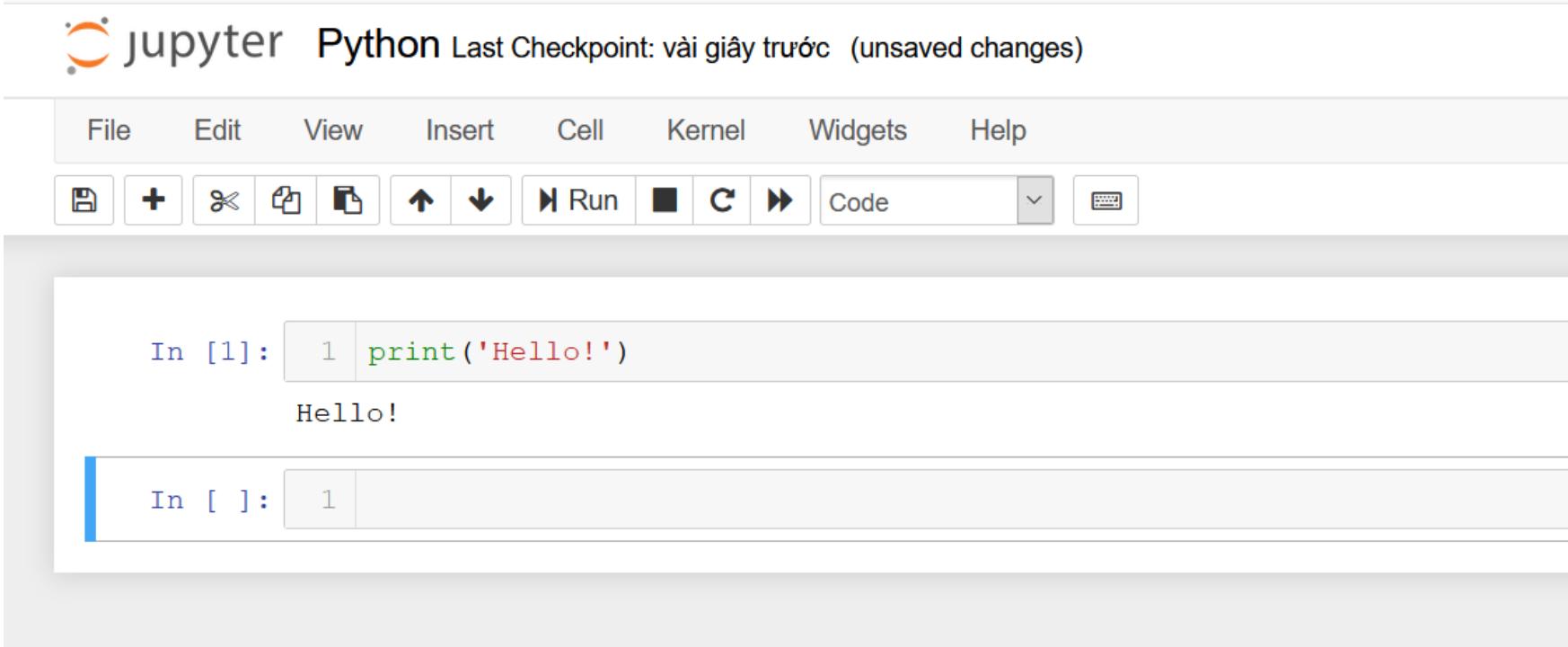


The screenshot shows a Google Colab interface. At the top, there's a logo consisting of orange 'CO' letters, the file name 'python.ipynb', and a star icon. Below the logo is a menu bar with 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. On the left side, there's a sidebar with icons for file navigation and a search bar. The main area contains a code cell with the command `[1] print('my first program !!!')`. The output of this cell, `my first program !!!`, is displayed below it in a larger font.

Demo

Introduction

- ❖ First Python program
 - ❖ Using Anaconda + Jupyter Notebook



The screenshot shows a Jupyter Notebook interface. At the top, there's a toolbar with various icons for file operations, cell selection, and execution. Below the toolbar, the main area displays a code cell labeled "In [1]". Inside the cell, the Python code `print('Hello!')` is written. When the cell is run, the output "Hello!" is displayed directly below it. A second, empty code cell labeled "In []:" is visible below the first one.

Demo

Introduction

❖ Install new package

```
1 import matplotlib.pyplot as plt  
2  
3 plt.plot([1, 2, 3, 4])
```

```
ModuleNotFoundError Traceback (most recent call last)  
<ipython-input-1-64dc63c31929> in <module>  
----> 1 import matplotlib.pyplot as plt  
      2  
      3 plt.plot([1, 2, 3, 4])  
  
ModuleNotFoundError: No module named 'matplotlib'
```

Introduction

❖ Install new package

pip install matplotlib

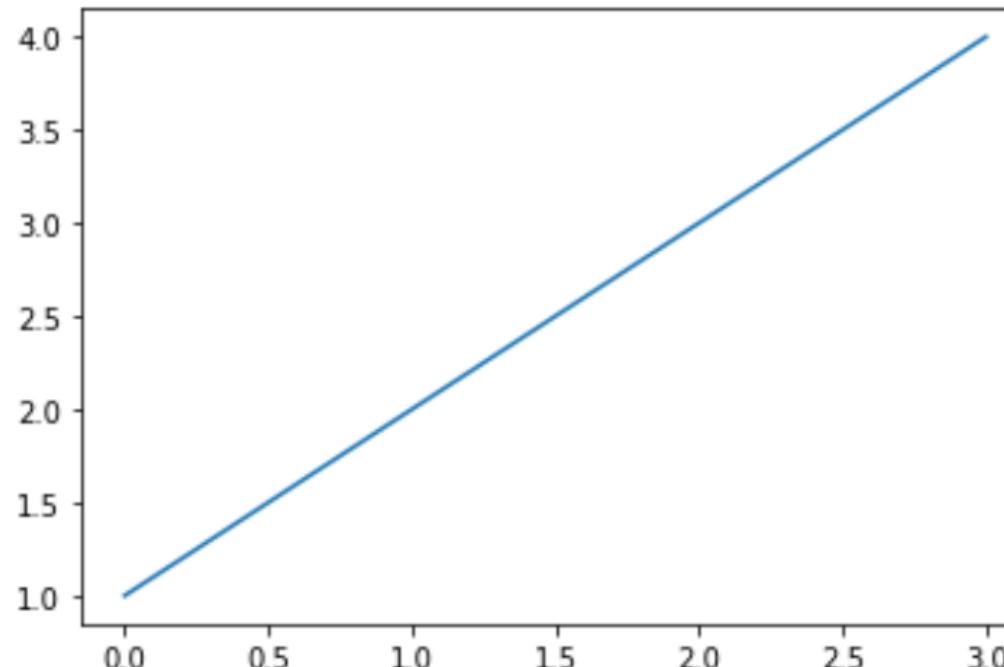
```
1 !pip install matplotlib

Collecting matplotlib
  Using cached matplotlib-3.3.4-cp36-cp36m-manylinux1_x86_64.whl (11.5 MB)
Requirement already satisfied: python-dateutil>=2.1 in ./lib/python3.6/site-packages (from matplotlib) (2.8.1)
Collecting cycler>=0.10
  Using cached cycler-0.10.0-py2.py3-none-any.whl (6.5 kB)
Collecting kiwisolver>=1.0.1
  Using cached kiwisolver-1.3.1-cp36-cp36m-manylinux1_x86_64.whl (1.1 MB)
Requirement already satisfied: pyparsing!=2.0.4,!>=2.1.2,!>=2.1.6,>=2.0.3 in ./lib/python3.6/site-packages (from matplotlib) (2.4.7)
Collecting numpy>=1.15
  Using cached numpy-1.19.5-cp36-cp36m-manylinux2010_x86_64.whl (14.8 MB)
Collecting pillow>=6.2.0
  Downloading Pillow-8.3.1-cp36-cp36m-manylinux_2_5_x86_64.manylinux1_x86_64.whl (3.0 MB)
    |██████████| 3.0 MB 2.6 MB/s eta 0:00:01
Requirement already satisfied: six in ./lib/python3.6/site-packages (from cycler>=0.10->matplotlib) (1.16.0)
Installing collected packages: pillow, numpy, kiwisolver, cycler, matplotlib
Successfully installed cycler-0.10.0 kiwisolver-1.3.1 matplotlib-3.3.4 numpy-1.19.5 pillow-8.3.1
```

Introduction

❖ Install new package

```
1 import matplotlib.pyplot as plt  
2  
3 plt.plot([1, 2, 3, 4])  
4 plt.show()
```



Variable

variable_name = variable_value

```
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
```

variable_value	
Integer	1, 2, 3, 0, -1, -2
Float	1.5, 0.5, -3.21, 1.0
String	'Joe', 'Schmoe', "Joe", "Schmoe"
Boolean	True, False

Variable

variable_name = variable_value

Demo

variable_name

Should has a meaning

Cannot use keywords

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

Example

1. | return = 5

Kết quả

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

Built-in Functions

print(parameters)

```
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

type(parameter)

```
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'>
```

Built-in Functions

input(promt)

```
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
```

Built-in Functions

Type Conversion

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)
```

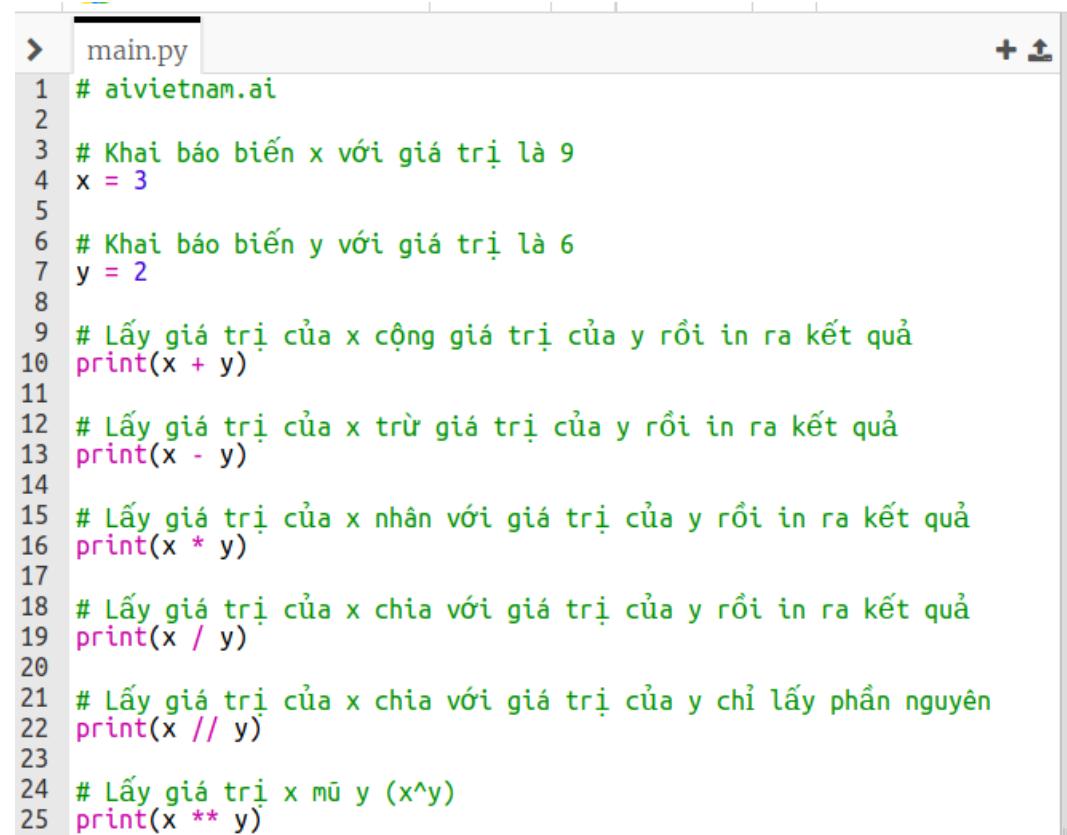
float()

```
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)
```

Demo

Basic Operators

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	$3^{**}5 = 243$



The screenshot shows a code editor window with a file named 'main.py'. The code demonstrates various arithmetic operations using variables x and y. The code is as follows:

```
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)
```

Demo

Example

Celsius to Fahrenheit conversion

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

Celsius to Fahrenheit conversion

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

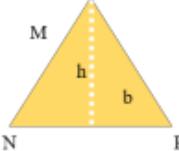
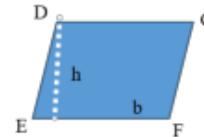
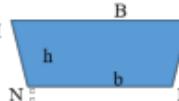
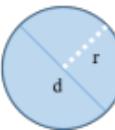
```
1. # input
2. temp_c = float(input('Nhập nhiệt độ theo độ C: '))
3.
4. # process
5. temp_f = ((9 / 5) * temp_c) + 32
6.
7. # output
8. print('Nhiệt độ F là: ', temp_f)
```

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

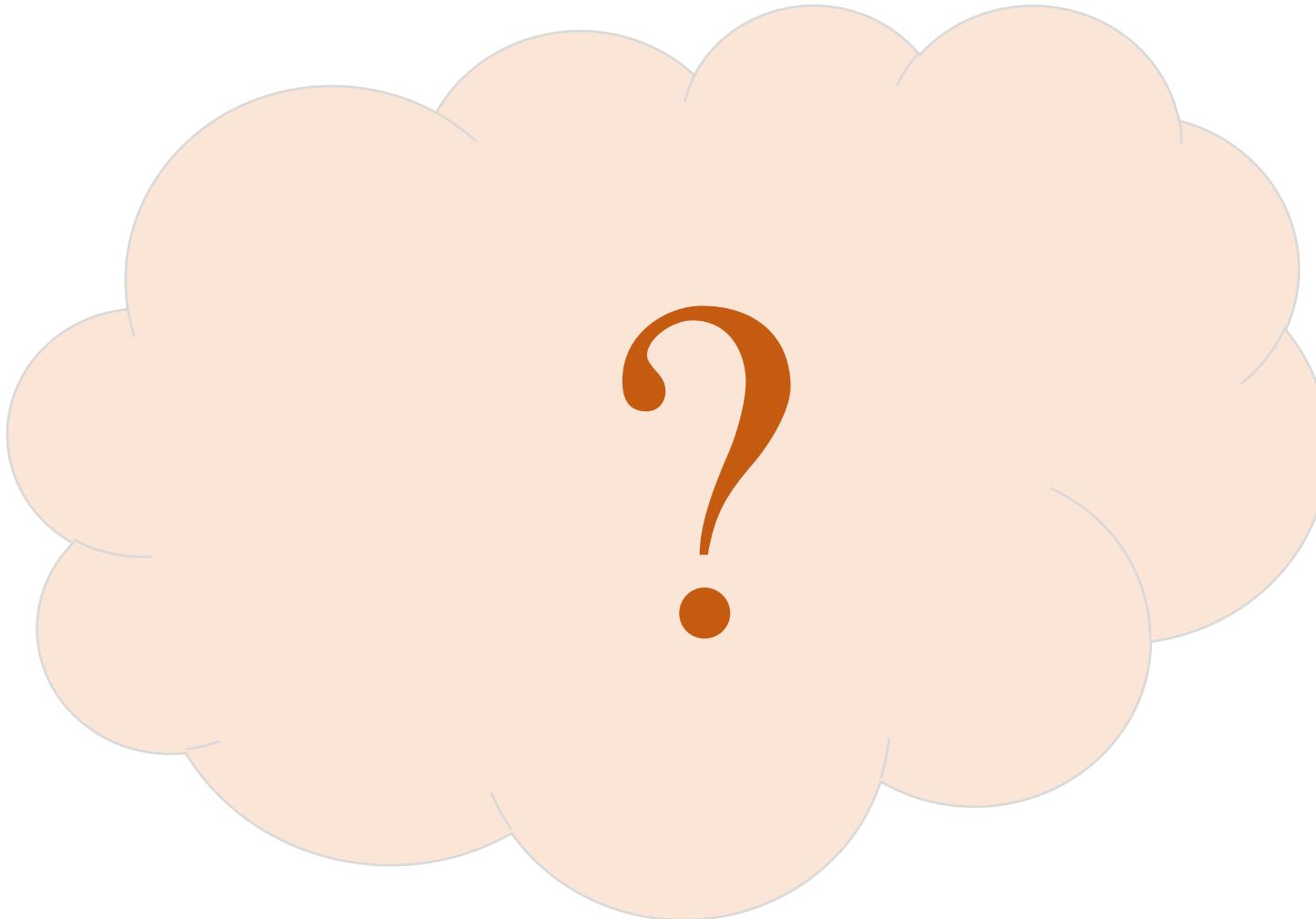
Example

Circumference and area of common shapes

Demo

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		$A = \pi r^2$	$C = 2\pi r \times \pi d$

Questions



Function

Notes for function construction

Define function name

Lowercase with underscores
and begin with a verb

Indentation

Use 4 spaces for indentation

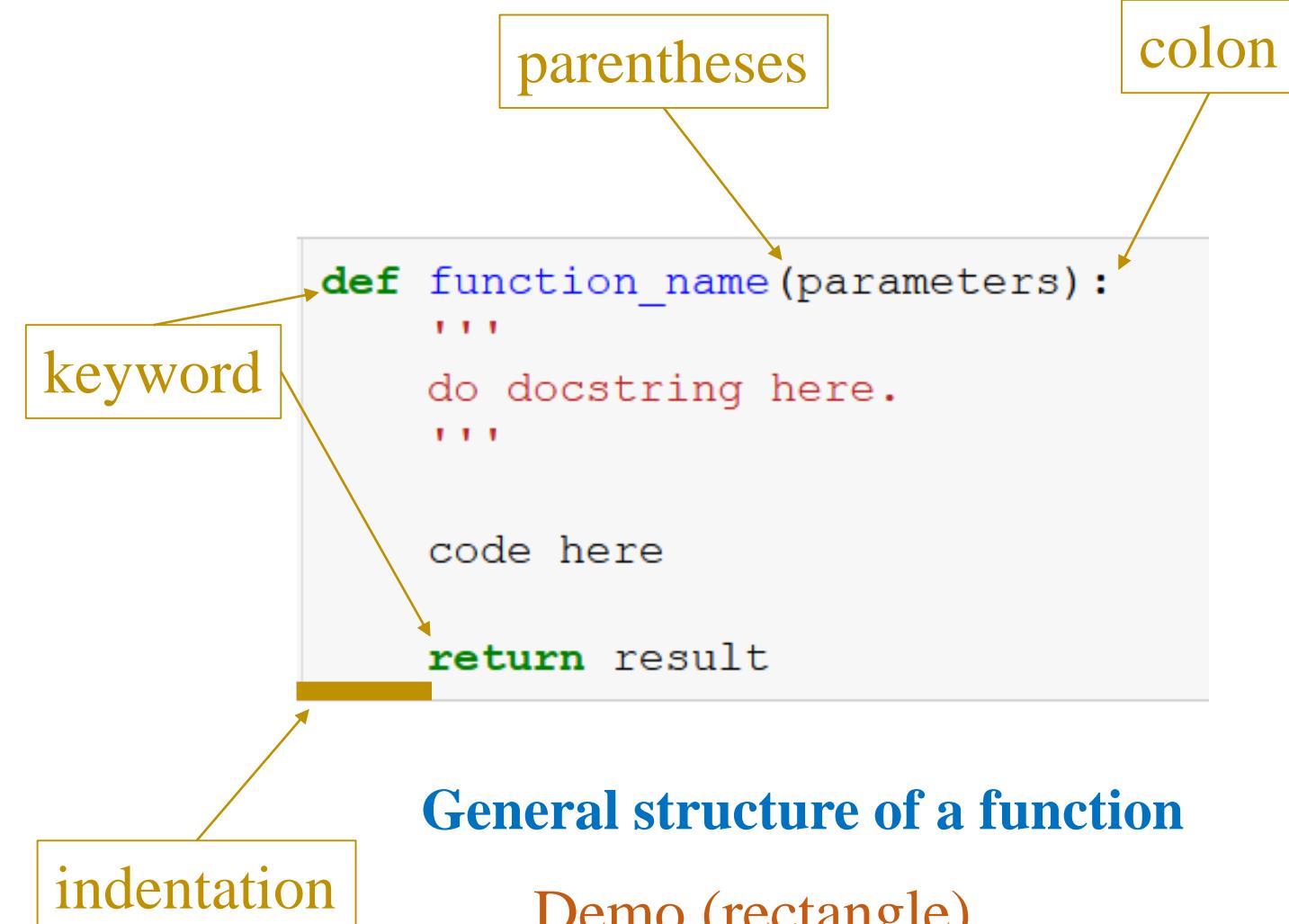
Determine function parameters

Input data help the function done

Do docstring

Explain and describe the function

Output of the function



General structure of a function

Demo (rectangle)

Function

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

    code here

    return result
```

Define function name
Lowercase with underscores,
usually begin with a verb

Determine function parameters
Input data help the function done

Do docstring
Explain and describe the function

Output of the function

TÊN	HÌNH DẠNG	DIỆN TÍCH	CHU VI
HÌNH CHỮ NHẬT		$A = L \times w$	$P = L+w+L+w$ $P=2L+2w$

Name	compute_rectangle_area
Parameter	height, width
Output	area

```
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
```

Function

❖ Default values

Default values



```
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
```

Overflow and Underflow

❖ Why?

```
1 # aivietnam
2
3 result = 1e-100
4 print(result)
```

1e-100

```
1 # aivietnam
2
3 result = 1e-1000
4 print(result)
```

0.0

```
1 # aivietnam
2
3 result = 1e100
4 print(result)
```

1e+100

```
1 # aivietnam
2
3 result = 1e1000
4 print(result)
```

inf

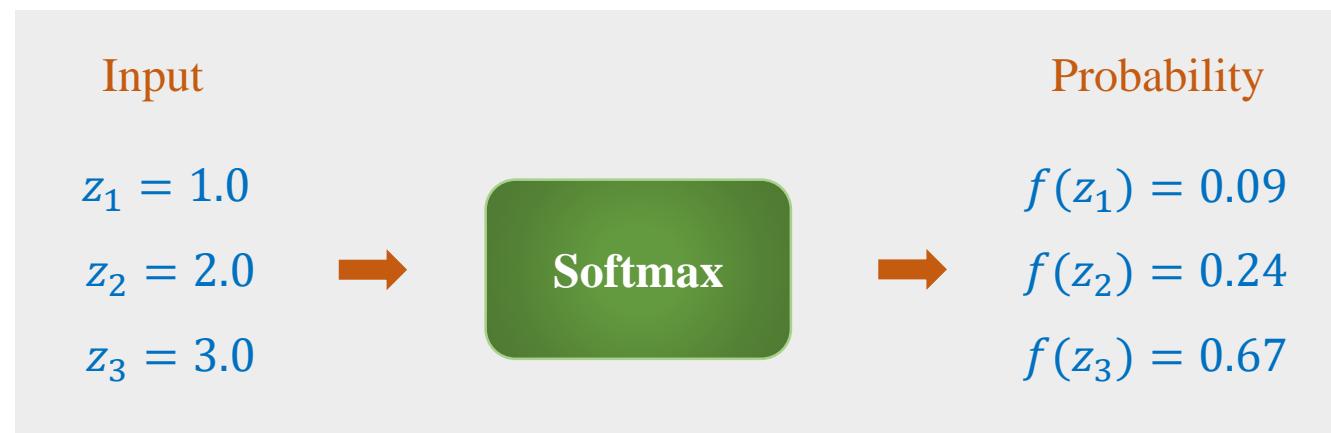
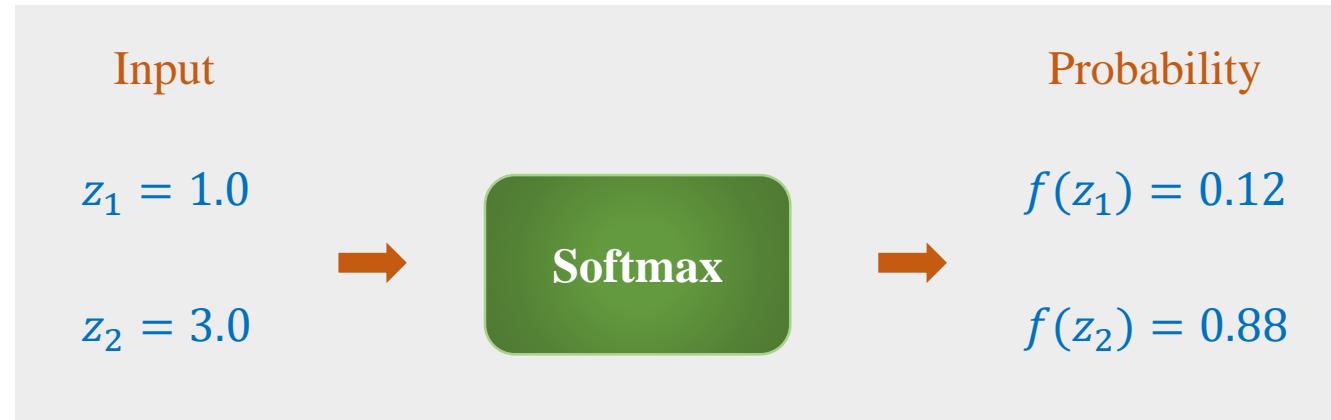
Overflow and Underflow

Softmax function

$$P_i = f(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

$$0 \leq f(z_i) \leq 1$$

$$\sum_i f(z_i) = 1$$



Softmax function

Chuyển các giá trị của một vector thành các giá trị xác suất

Formula

$$f(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$$

$$0 \leq f(x_i) \leq 1$$

$$\sum_i f(x_i) = 1$$

Input

$$x_1 = 1.0$$

$$x_2 = 2.0$$

$$x_3 = 3.0$$

Softmax

Probability

$$f(x_1) = 0.09$$

$$f(x_2) = 0.24$$

$$f(x_3) = 0.67$$

```

1 import math
2
3 # Given three values
4 v1 = 1.0
5 v2 = 2.0
6 v3 = 3.0
7
8 # compute softmax
9 total = math.exp(v1) + math.exp(v2) + math.exp(v3)
10
11 s1 = math.exp(v1)/total
12 s2 = math.exp(v2)/total
13 s3 = math.exp(v3)/total
14
15 # print out
16 print(f"{s1:.5f} {s2:.5f} {s3:.5f}")
17

```

0.09003 0.24473 0.66524

```

1 import math
2
3 # Given three values
4 v1 = 1001.0
5 v2 = 1002.0
6 v3 = 1003.0
7
8 # compute softmax
9 total = math.exp(v1) + math.exp(v2) + math.exp(v3)
10
11 s1 = math.exp(v1)/total
12 s2 = math.exp(v2)/total
13 s3 = math.exp(v3)/total
14
15 # print out
16 print(f"{s1:.5f} {s2:.5f} {s3:.5f}")
17

```

OverflowError
C:\Users\ DINHVI~1\AppData\Local\Temp\ipython-7
8 # compute softmax
----> 9 total = math.exp(v1) + math.exp(v2) + math.exp(v3)
10
11 s1 = math.exp(v1)/total

OverflowError: math range error

OverflowError

Traceback (most recent call last)

Softmax function

Chuyển các giá trị của một vector thành các giá trị xác suất

Formula

$$f(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$$

$$0 \leq f(x_i) \leq 1$$

$$\sum_i f(x_i) = 1$$

Input

$$x_1 = 1.0$$

$$x_2 = 2.0$$

$$x_3 = 3.0$$

Softmax

Probability

$$f(x_1) = 0.09$$

$$f(x_2) = 0.24$$

$$f(x_3) = 0.67$$

Giá trị nan vì
 e^x vượt giới
hạn lưu trữ của
biến

```

1 import math
2
3 # Given three values
4 v1 = 1001.0
5 v2 = 1002.0
6 v3 = 1003.0
7
8 # compute softmax
9 total = math.exp(v1) + math.exp(v2) + math.exp(v3)
10
11 s1 = math.exp(v1)/total
12 s2 = math.exp(v2)/total
13 s3 = math.exp(v3)/total
14
15 # print out
16 print(f"{s1:.5f} {s2:.5f} {s3:.5f}")
17

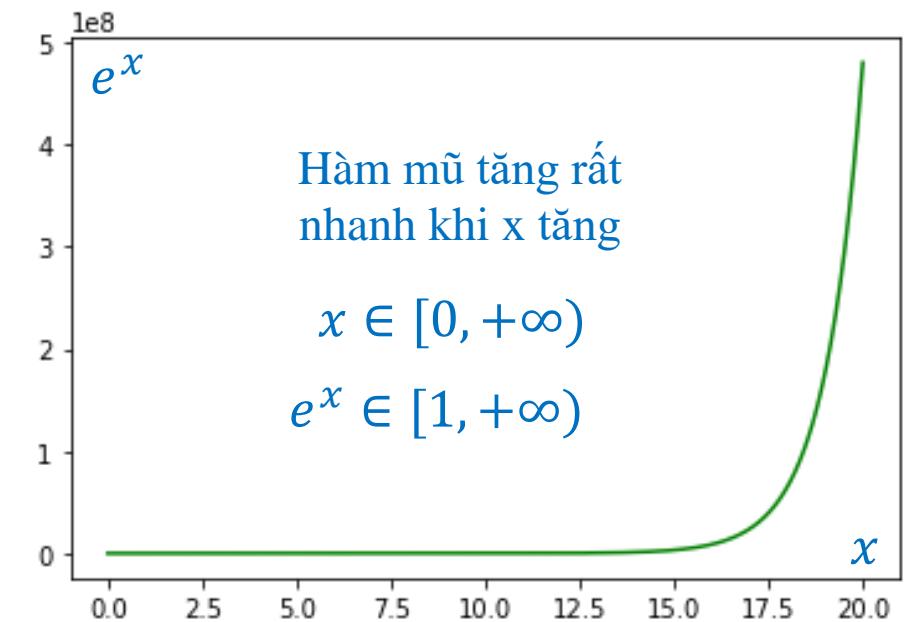
```

```

OverflowError
C:\Users\ DINHVI~1\AppData\Local\Temp\ipy
    7
    8 # compute softmax
----> 9 total = math.exp(v1) + math.exp(
    10
    11 s1 = math.exp(v1)/total

OverflowError: math range error

```



Implementation (stable)

Softmax function (stable)

(Stable) Formula

$$m = \max(x)$$

$$f(x_i) = \frac{e^{(x_i-m)}}{\sum_j e^{(x_j-m)}}$$

X

$$x_1 = 1.0$$

$$x_2 = 2.0$$

$$x_3 = 3.0$$

X-m

$$x_1 = -2.0$$

$$x_2 = -1.0$$

$$x_3 = 0$$

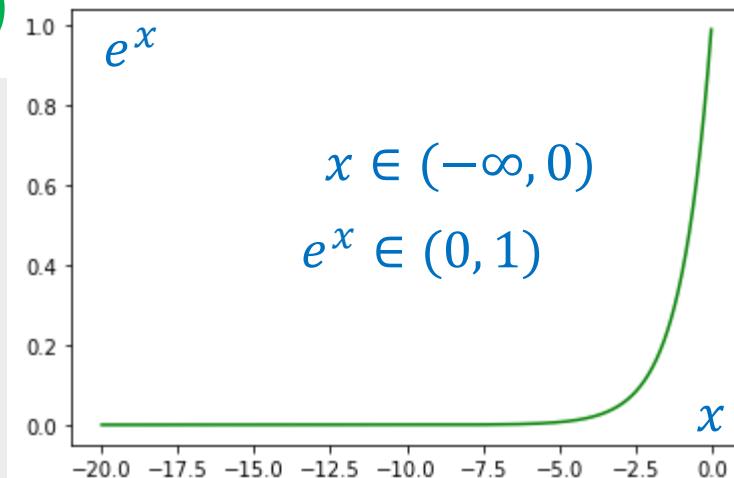
Softmax

Probability

$$f(x_1) = 0.09$$

$$f(x_2) = 0.24$$

$$f(x_3) = 0.67$$



```

3 # Given three values
4 v1 = 1.0
5 v2 = 2.0
6 v3 = 3.0
7
8 # get max
9 max_value = v3
10
11 # compute stable softmax
12 e_v1 = math.exp(v1 - max_value)
13 e_v2 = math.exp(v2 - max_value)
14 e_v3 = math.exp(v3 - max_value)
15
16 total = e_v1 + e_v2 + e_v3
17
18 s1 = e_v1/total
19 s2 = e_v2/total
20 s3 = e_v3/total
21
22 # print out
23 print(f"{s1:.5f} {s2:.5f} {s3:.5f}")

```

```

3 # Given three values
4 v1 = 1001.0
5 v2 = 1002.0
6 v3 = 1003.0
7
8 # get max
9 max_value = v3
10
11 # compute stable softmax
12 e_v1 = math.exp(v1 - max_value)
13 e_v2 = math.exp(v2 - max_value)
14 e_v3 = math.exp(v3 - max_value)
15
16 total = e_v1 + e_v2 + e_v3
17
18 s1 = e_v1/total
19 s2 = e_v2/total
20 s3 = e_v3/total
21
22 # print out
23 print(f"{s1:.5f} {s2:.5f} {s3:.5f}")

```

```

3 # Given three values
4 v1 = 1.0
5 v2 = 1001.0
6 v3 = 1002.0
7
8 # get max
9 max_value = v3
10
11 # compute stable softmax
12 e_v1 = math.exp(v1 - max_value)
13 e_v2 = math.exp(v2 - max_value)
14 e_v3 = math.exp(v3 - max_value)
15
16 total = e_v1 + e_v2 + e_v3
17
18 s1 = e_v1/total
19 s2 = e_v2/total
20 s3 = e_v3/total
21
22 # print out
23 print(f"{s1:.5f} {s2:.5f} {s3:.5f}")

```

0.09003 0.24473 0.66524

0.09003 0.24473 0.66524

0.00000 0.26894 0.73106

Condition

❖ Comparison operators

TOÁN TỬ SO SÁNH	
TOÁN TỬ	Ý NGHĨA
==	Bằng
!=	Khác
>	Lớn hơn
<	Nhỏ hơn
>=	Lớn hơn hoặc bằng
<=	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

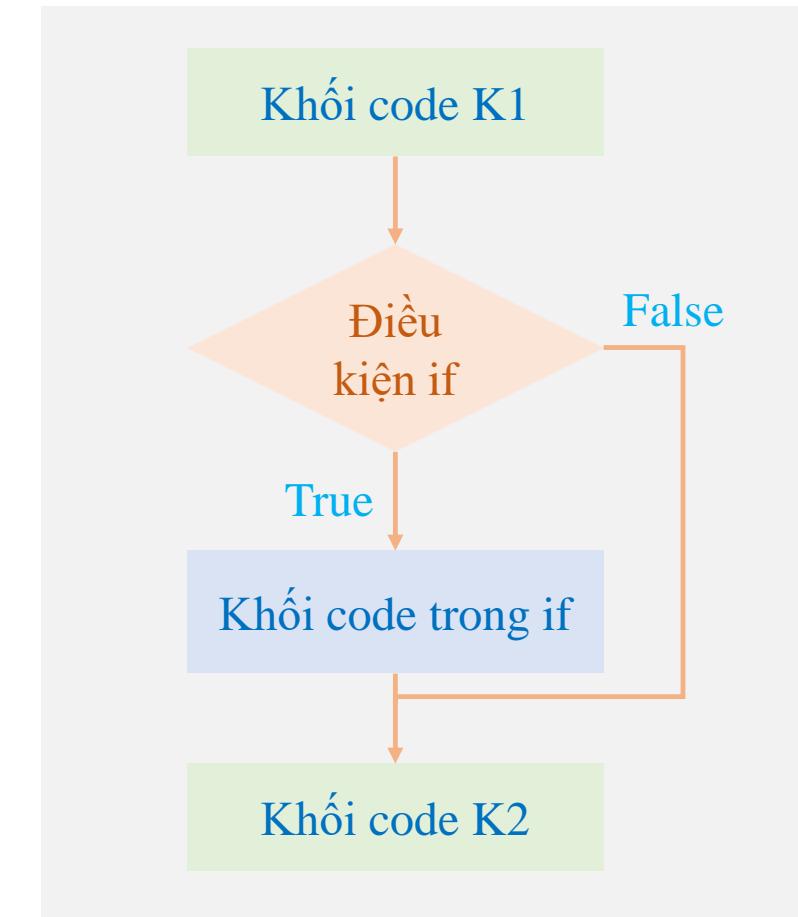
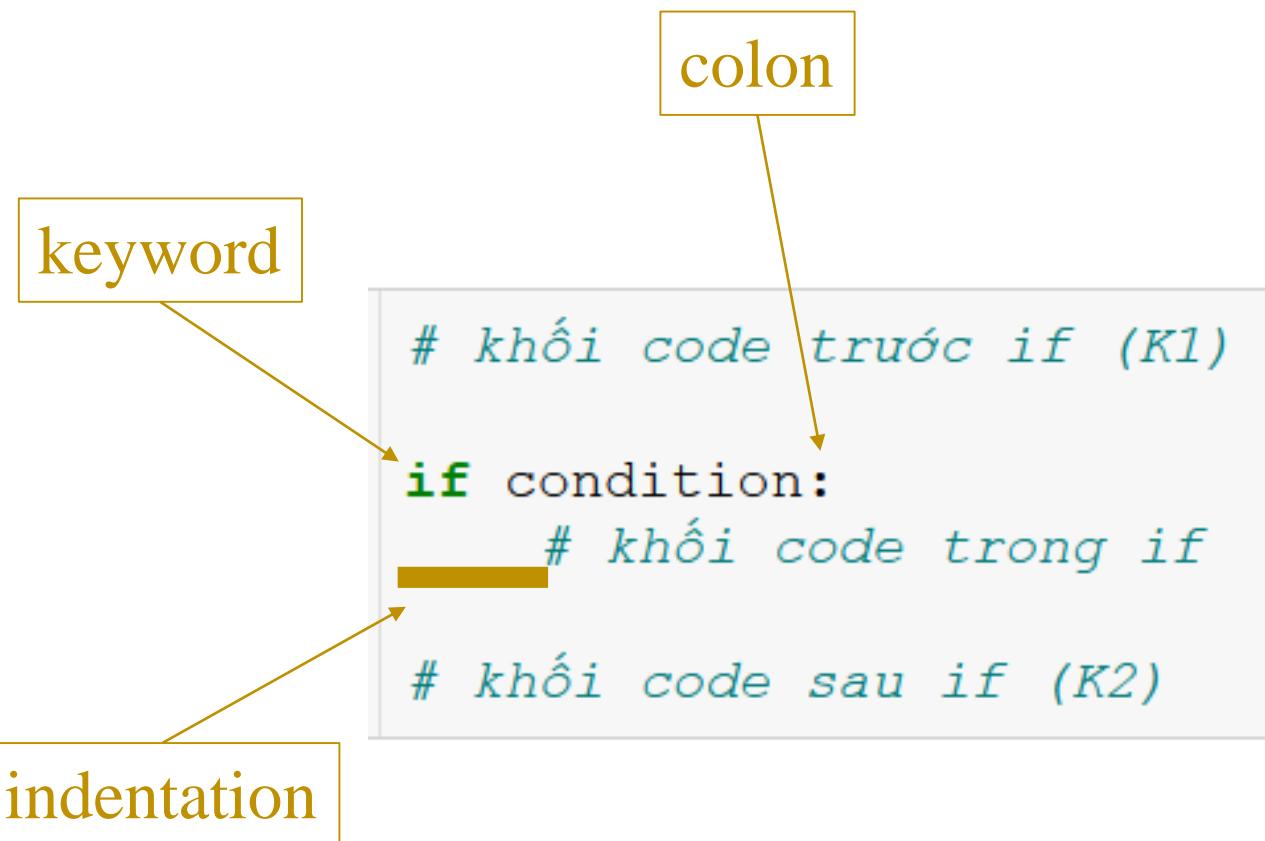
False

True

True

Condition

❖ if condition



Condition

if condition

$$\text{ReLU}(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases}$$

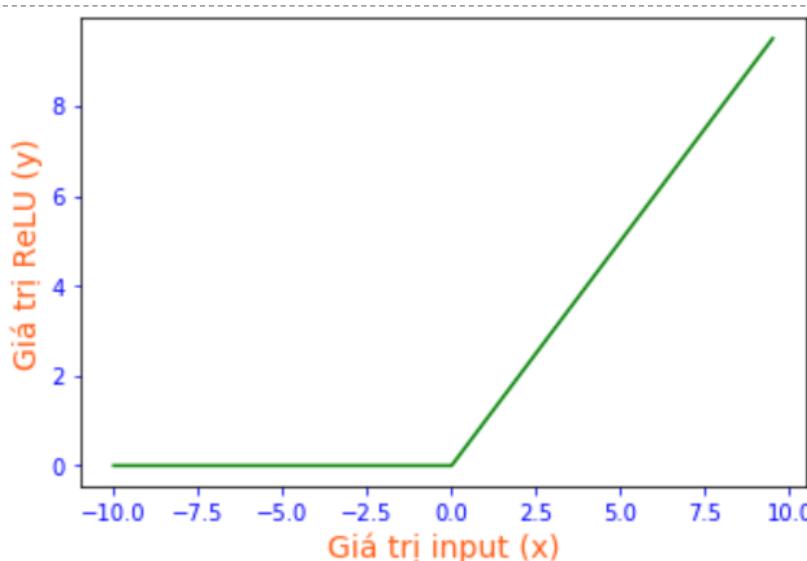
data =

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)

```

Condition

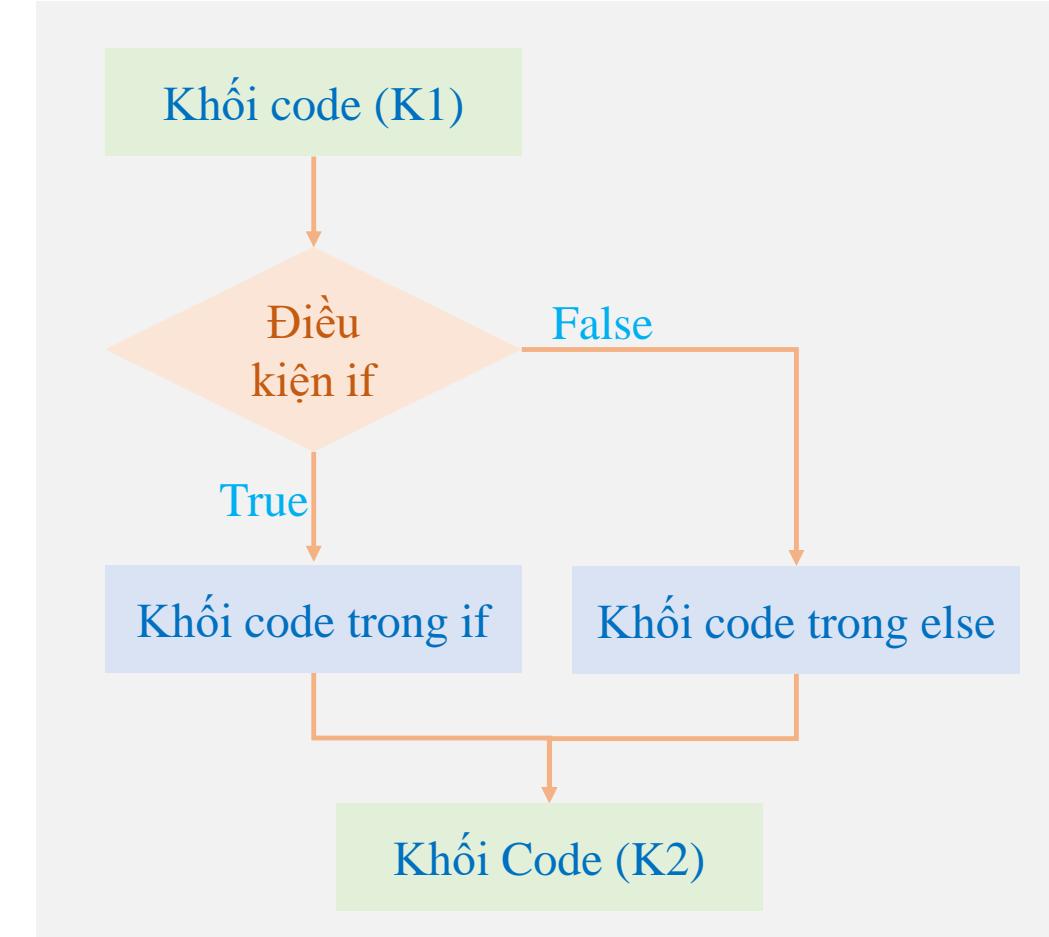
❖ if-else condition

```
# kh i code tr c if (K1)
if condition:
    # kh i code trong if
else:
    # kh i code trong else
# kh i code sau if (K2)
```

keyword

colon

indentation



Condition

❖ if-else condition

if
condition

$$\text{ReLU}(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases}$$

data = 

data_a = ReLU(data)

data_a = 

Demo

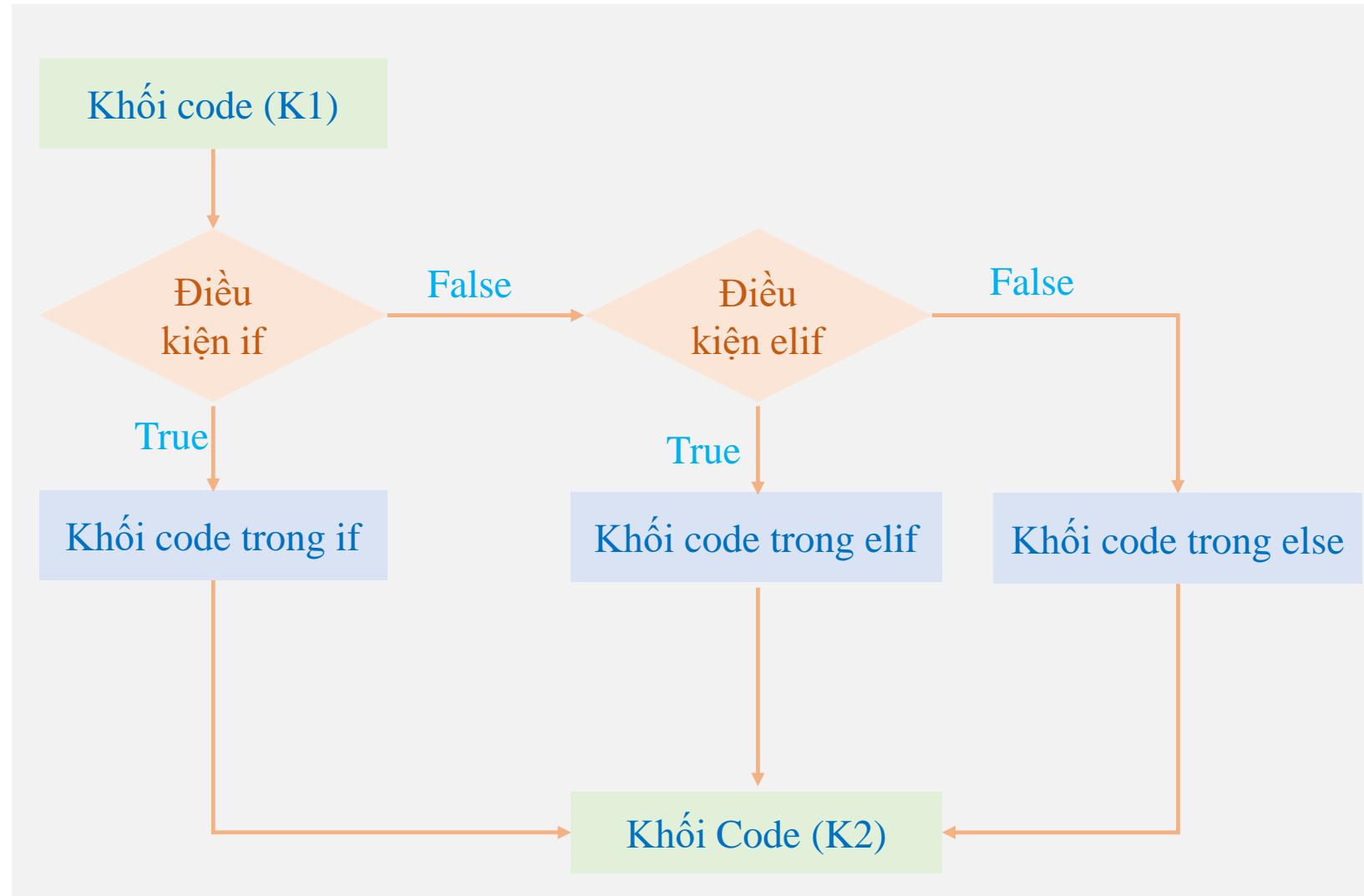
Condition

❖ if-elif-else condition

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

colon

indentation



Condition

❖ if-elif-else condition

$$\text{LTP}(x, y) = \begin{cases} 0 & \text{if } x = y \\ 1 & \text{if } x > y \\ -1 & \text{if } x < y \end{cases}$$

Demo

Condition

❖ if-elif-else condition

$$\text{FuzzyEncoding}(x, y, T) = \begin{cases} 5, & \text{if } x + T \leq y \\ 4, & \text{if } x < y < x + T \\ 3, & \text{if } x = y \\ 2, & \text{if } y < x < y + T \\ 1, & \text{if } y + T \leq x \end{cases}$$

Random and Math Modules

```
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

Demo

Random and Math Modules

```
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

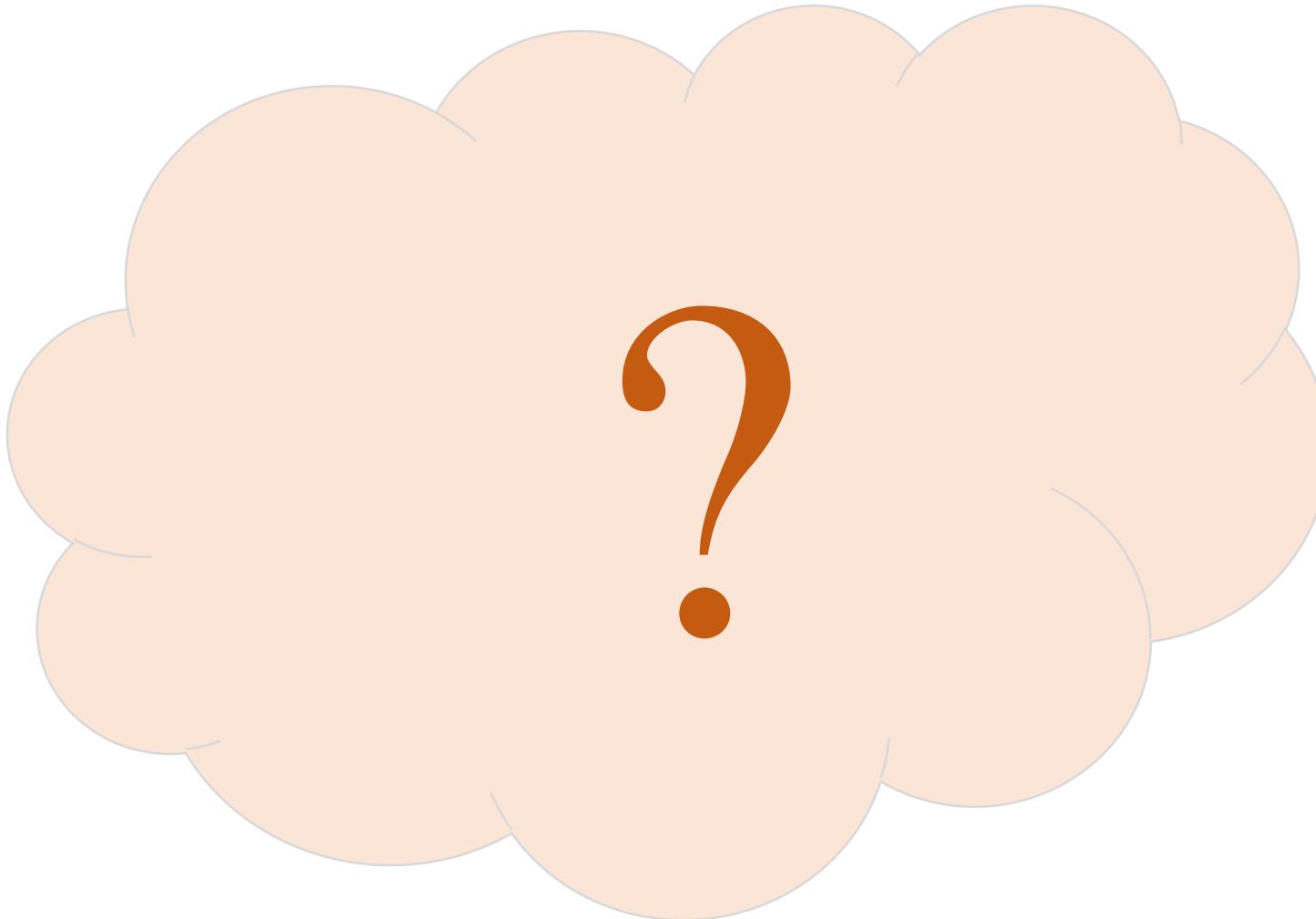
Demo

```
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))
```

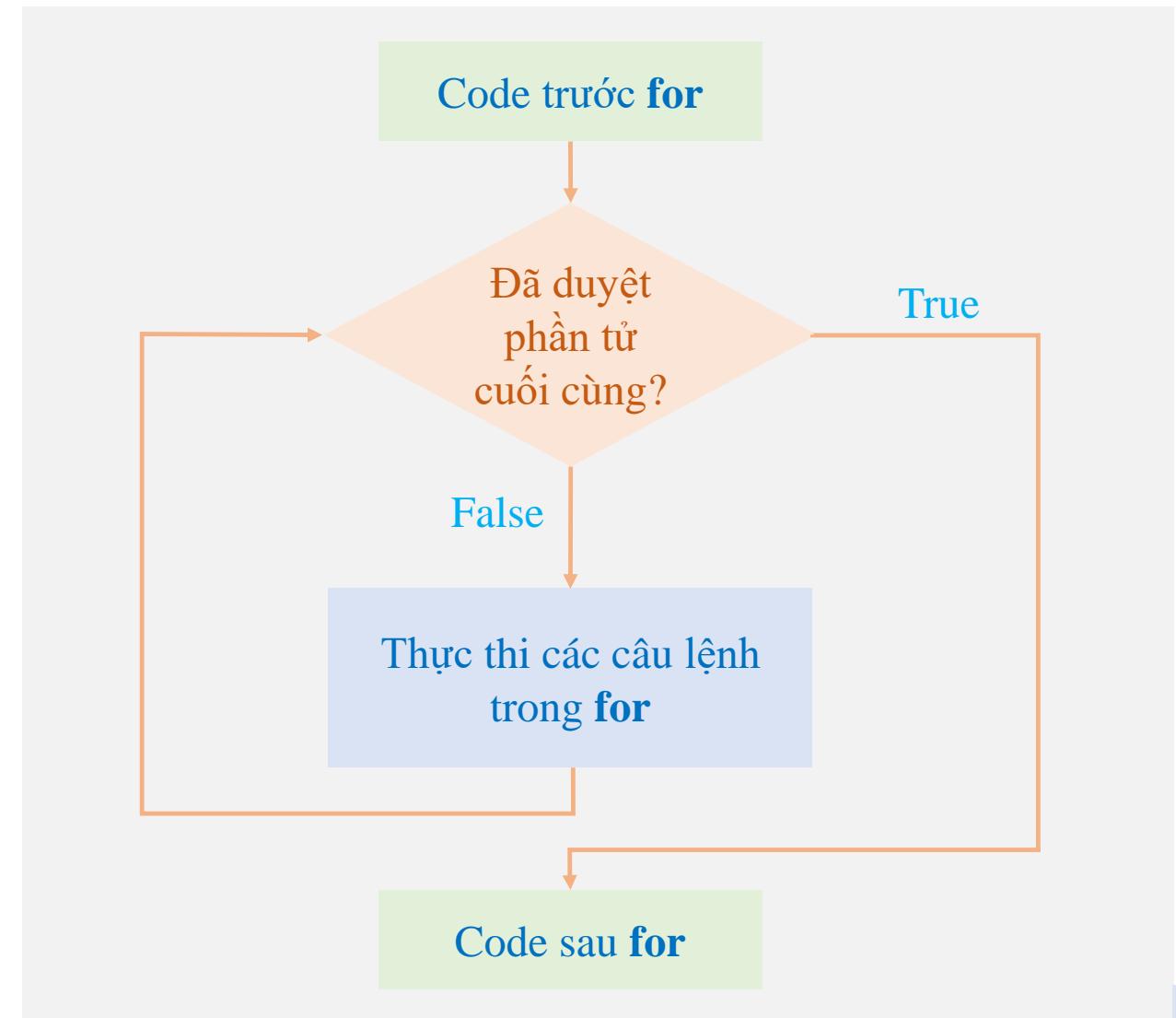
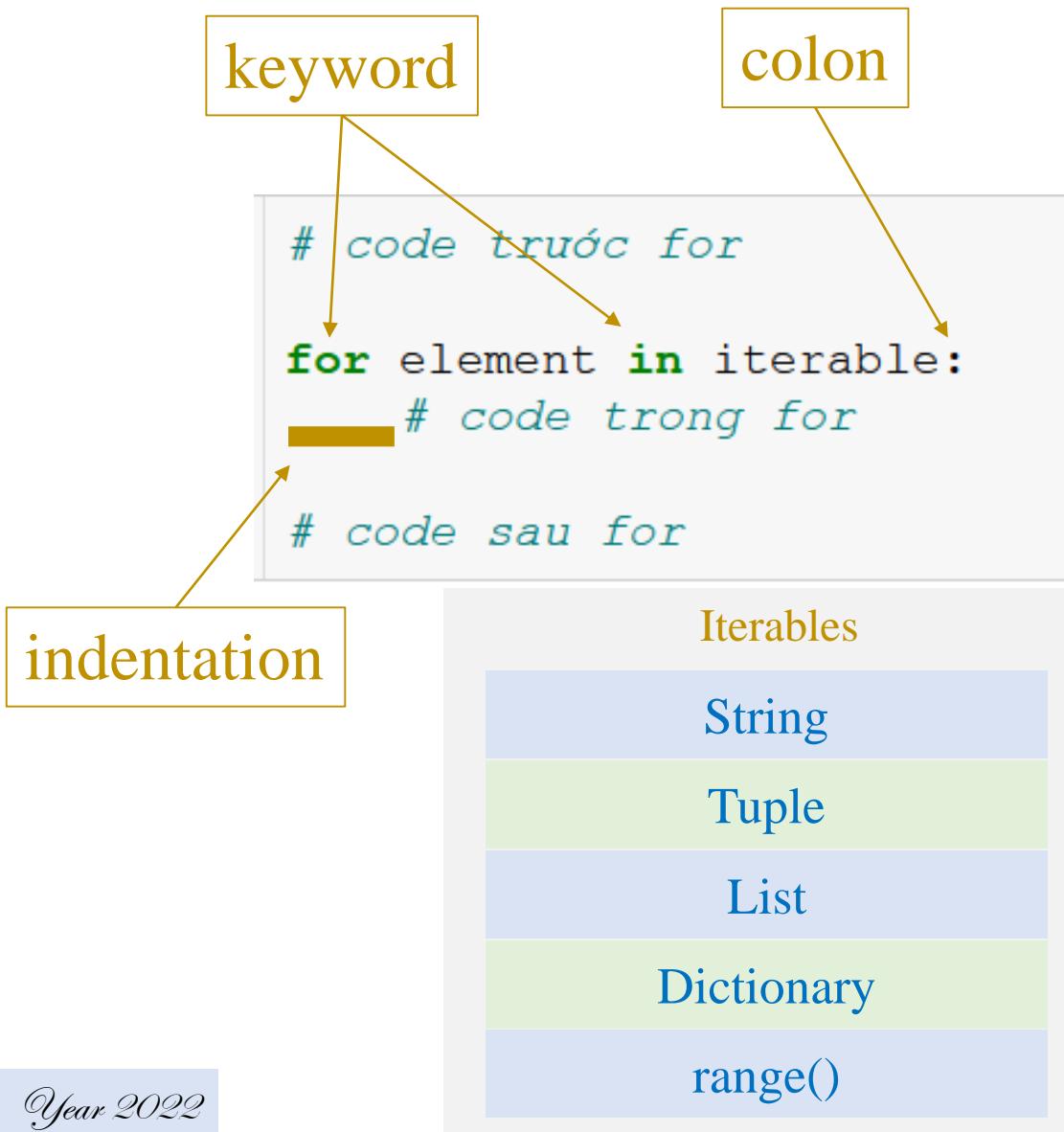
<https://docs.python.org/3/library/random.html>

7
3
6
4

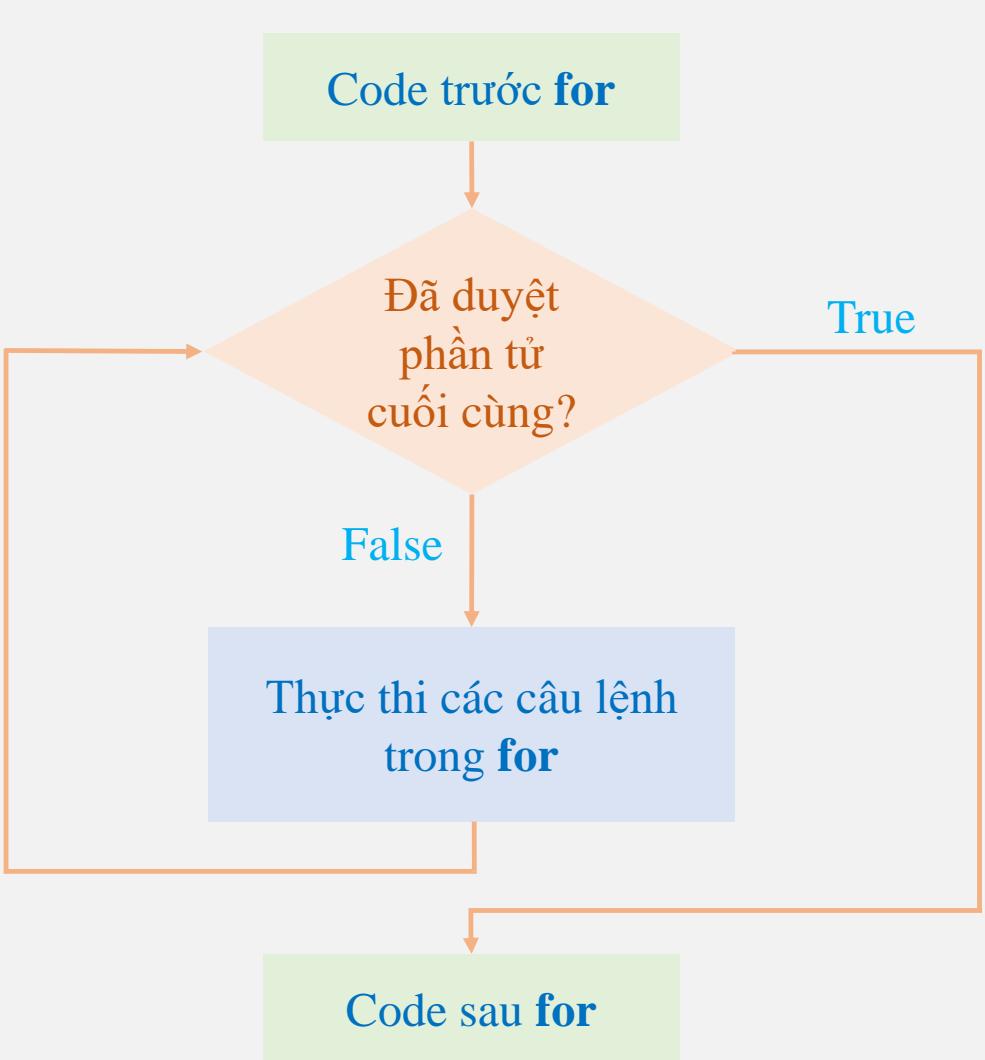
Questions



For Loop



For Loop



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

apple
banana
melon
peach

```
1 # iterate a dictionary
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

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

apple
banana
melon

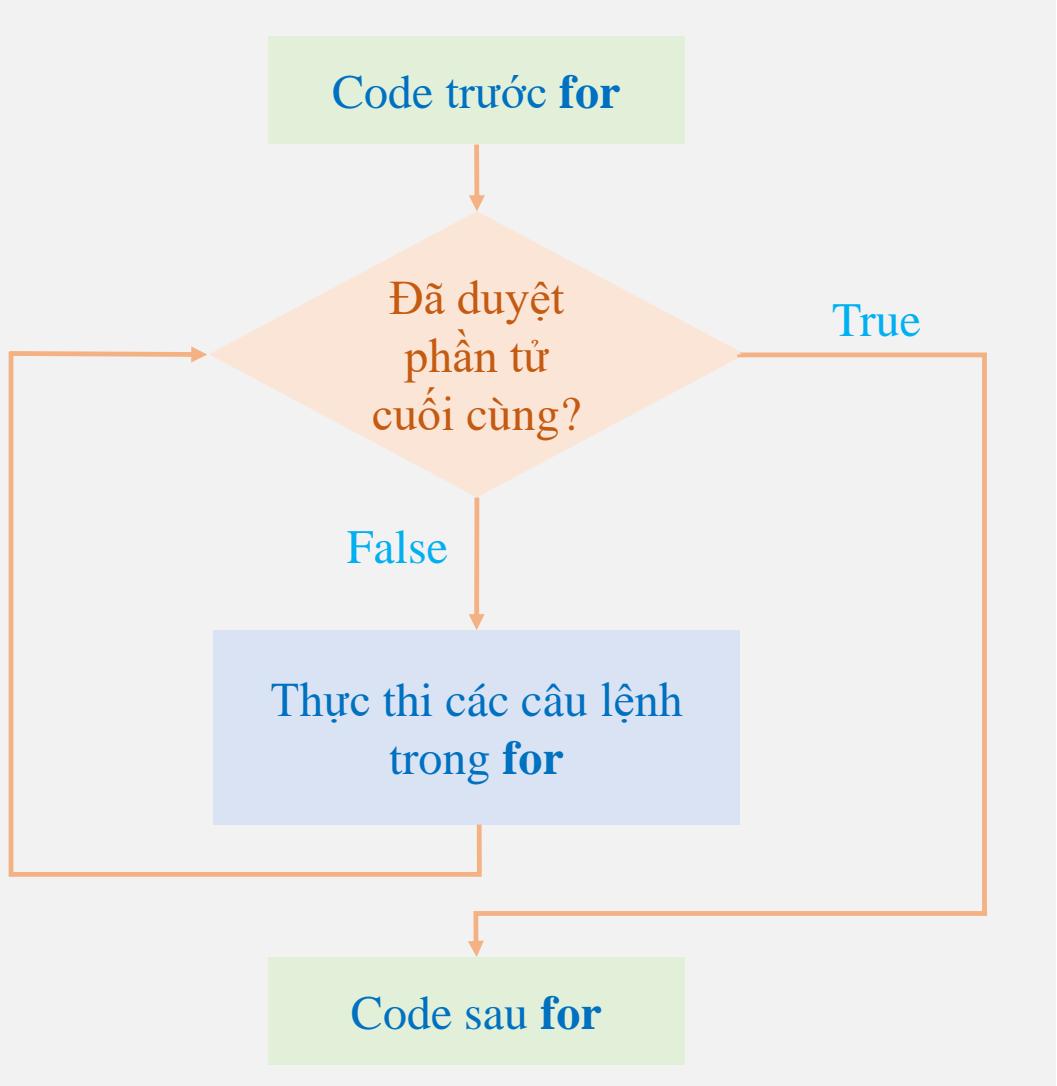
```
1 # iterate a string
2
3 greeting = 'Hello'
4
5 for char in greeting:
6     print(char)
```

H
e
l
l
o

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

0
1
2
3
4

For Loop



range(start=0, stop, step=1)

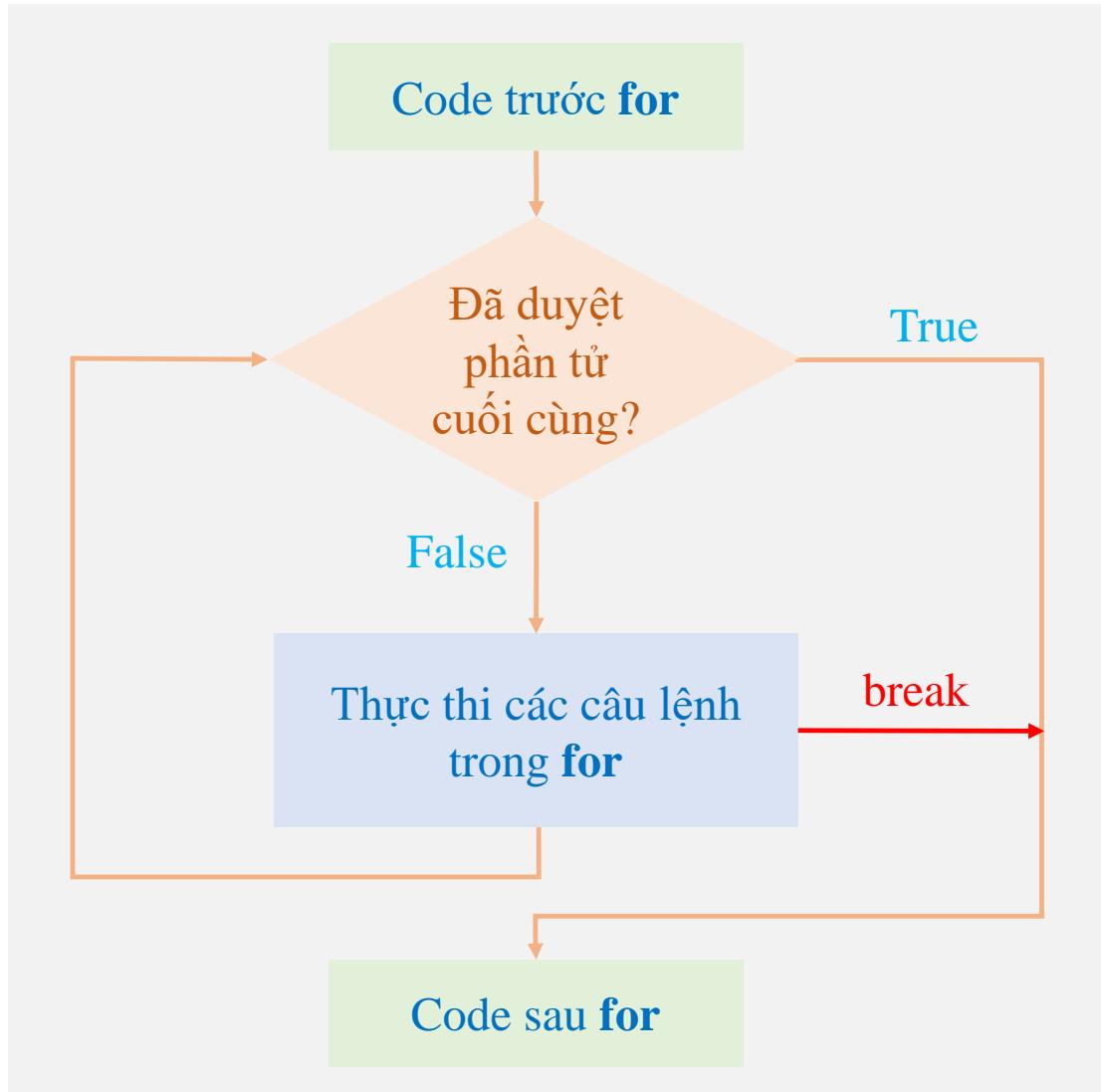
range(start=0, stop=5, step=1)

0, 1, 2, 3, 4

range(5)

0, 1, 2, 3, 4

For Loop

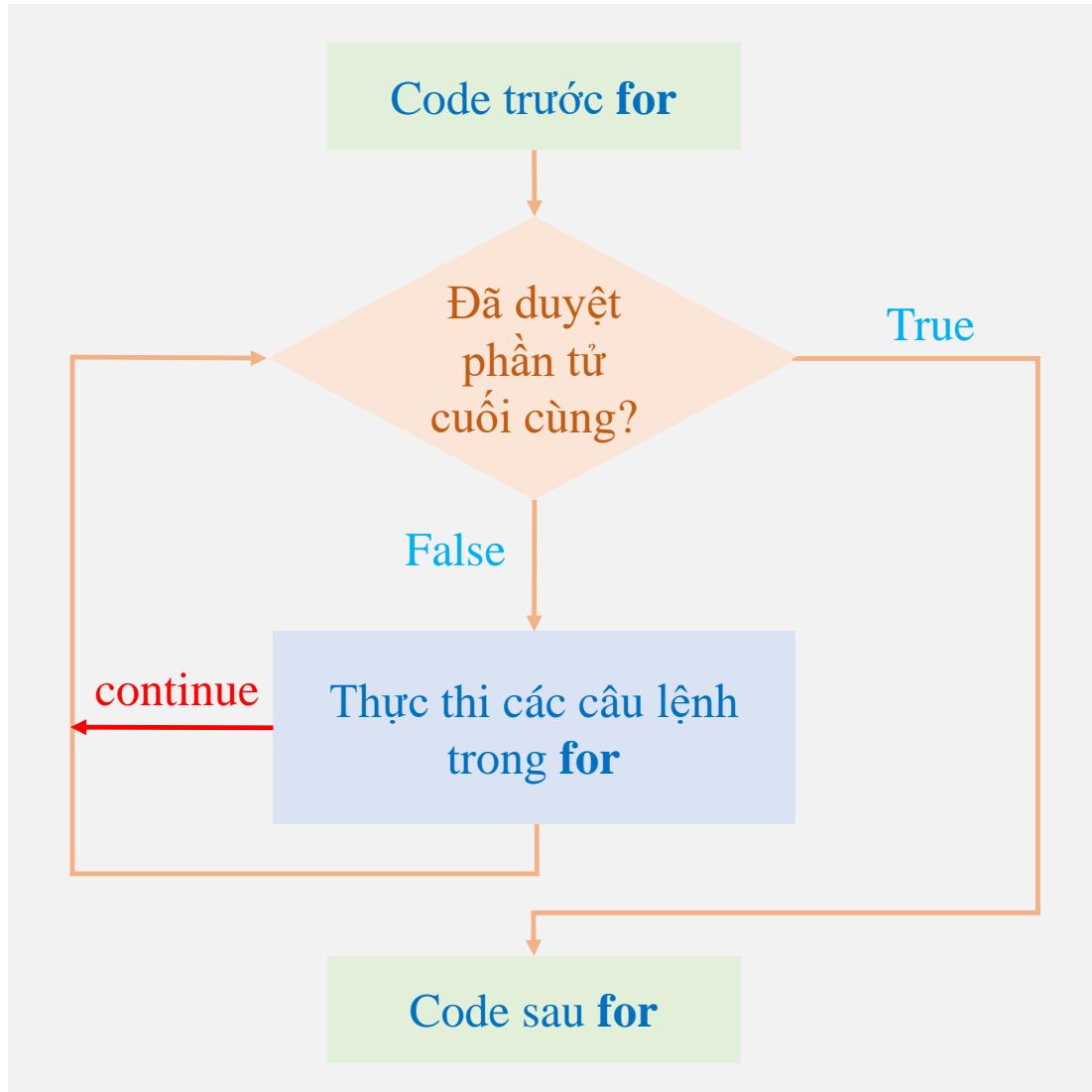


break keyword

```
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
```

For Loop



continue keyword

```
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
```

Demo

Example

❖ Simulation of coin tossing

Event: bất kỳ sự kiện gì xảy ra

Experiment: sự thử nghiệm cho ra kết quả

Sample space: tập hợp tất cả các kết quả có thể xảy ra của một experiment

Random variable: biến ngẫu nhiên, mô tả outcome của một event



Tiền xu có 2 mặt
(head và tail)

Experiment: ném đồng xu

Sample space $S = \{\text{Head}, \text{Tail}\}$

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

$P(A)$ Xác suất để event A xảy ra

Xác suất để kết quả của một lần ném là head

Xác suất

$$P(\text{event}) = \frac{|\text{event}|}{|S|}$$

$A = \{\text{head}\}$

$S = \{\text{head}, \text{tail}\}$

$$P(A) = \frac{|A|}{|S|} = \frac{1}{2}$$

Example

❖ Simulation of coin tossing



Count #heads

Count #tails

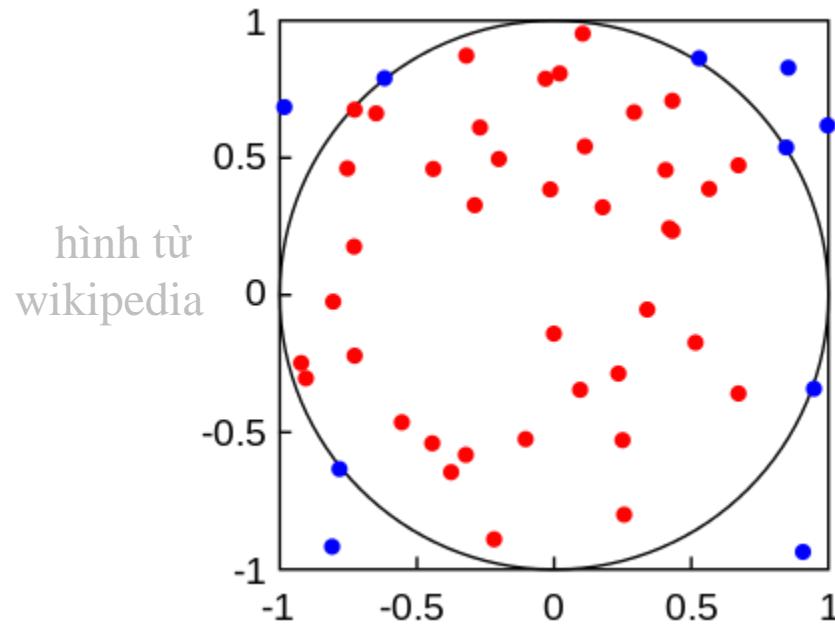
Check if the two numbers are similar

Demo

```
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 khôi else
22.     total_flips = total_flips + 1
```

Example: PI Estimation

❖ PI estimation



N_s is #random samples within the square generated according to uniform distribution

N_c is #random samples within the circle generated according to uniform distribution

circle radius $r = 1$

$$\text{circle_area } A_c = \pi r^2$$

square side $s = 2$

$$\text{square_area } A_s = s^2$$

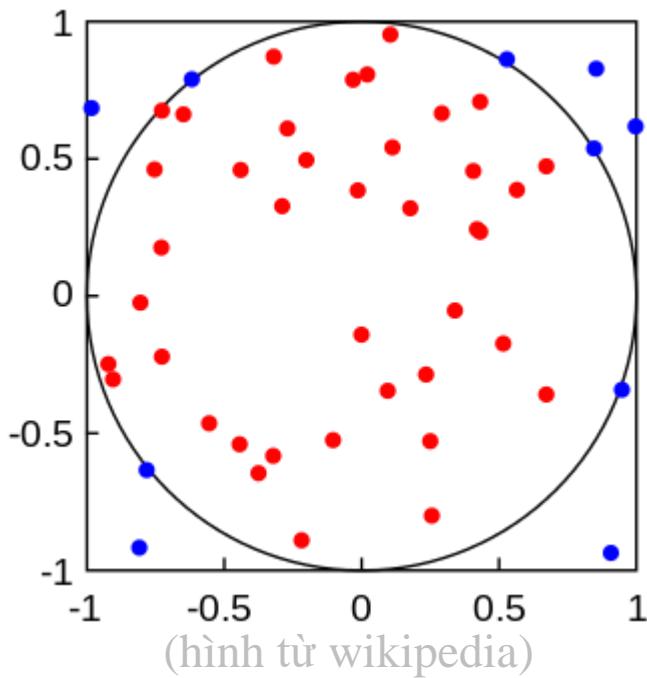
$$\frac{A_s}{A_c} \approx \frac{N_s}{N_c}$$

$$\frac{s^2}{\pi r^2} \approx \frac{N_s}{N_c}$$

$$\pi \approx \frac{s^2 N_c}{N_s}$$

Example

❖ PI estimation



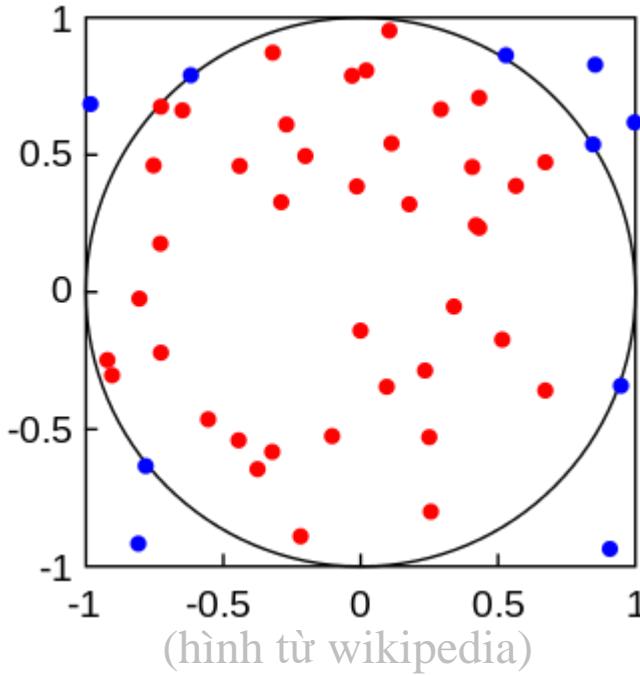
$$\pi \approx \frac{s^2 N_c}{N_s}$$

Random numbers $\in [-1,1]$

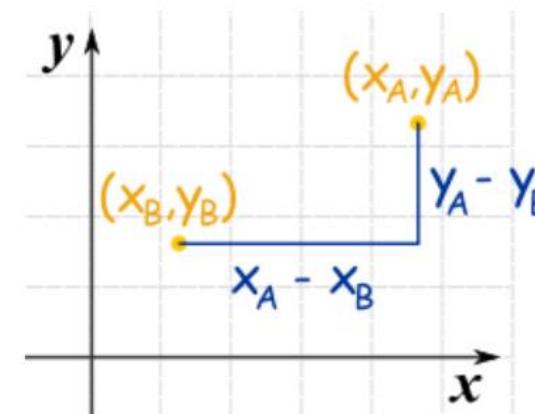
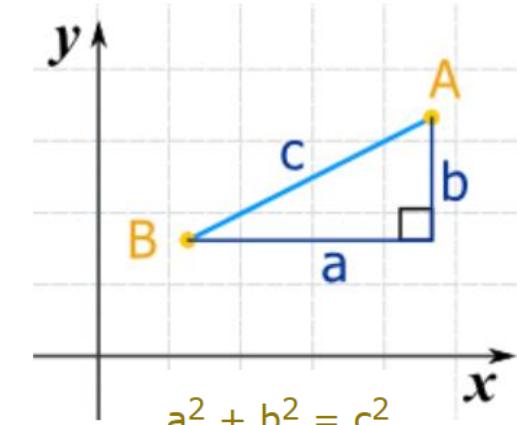
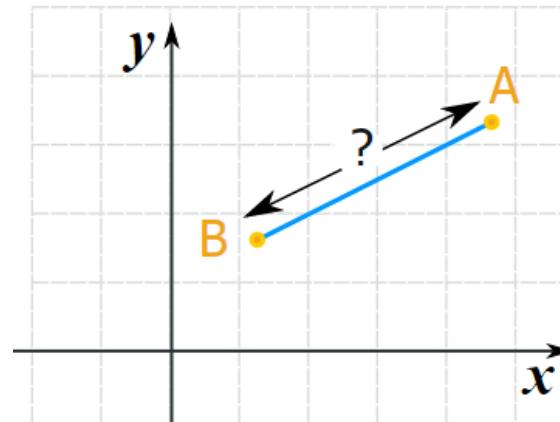
Check if a point is
within the circle

Example

❖ PI estimation



$$\pi \approx \frac{s^2 N_c}{N_s}$$

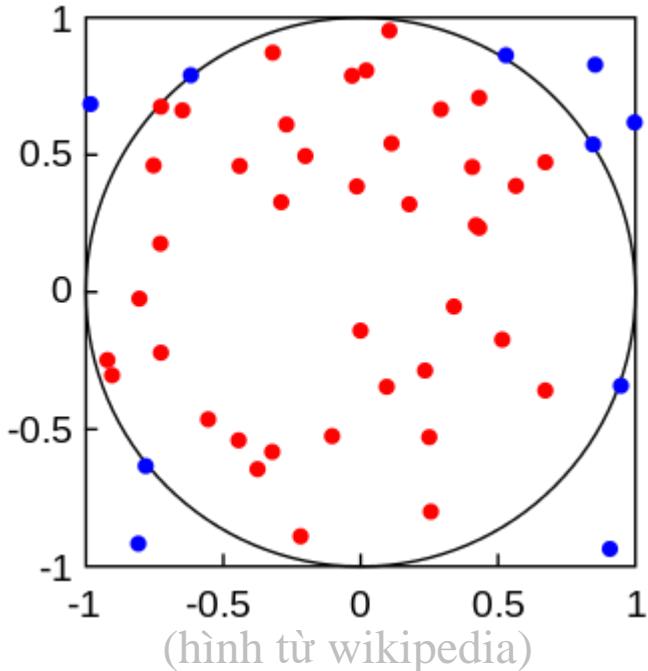


$$c = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$$

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

Example

❖ PI estimation



$$\pi \approx \frac{s^2 N_c}{N_s}$$

```
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)
```

Example

❖ PI estimation

Gregory-Leibniz Series

$$\text{PI} \approx 4 \sum_{i=1}^n \frac{(-1)^{i+1}}{2i - 1}$$

Nilakantha Series

$$\text{PI} \approx 3 + 4 \sum_{i=0}^n \frac{-1^i}{(2i+2)(2i+3)(2i+4)}$$

```
1 # Gregory-Leibniz Series
2
3 n = 1000
4 PI = 0
5 for i in range(1, n):
6     PI = PI + (-1)**(i+1) / (2*i - 1)
7 PI = PI*4
8
9 print('Estimated PI is ', PI)
```

Estimated PI is 3.142593654340044

```
1 # Nilakantha Series
2
3 n = 1000
4 PI = 0
5 for i in range(n):
6     PI = PI + (-1)**(i) / ((2*i+2)*(2*i+3)*(2*i+4))
7 PI = 3 + 4*PI
8
9 print('Estimated PI is ', PI)
```

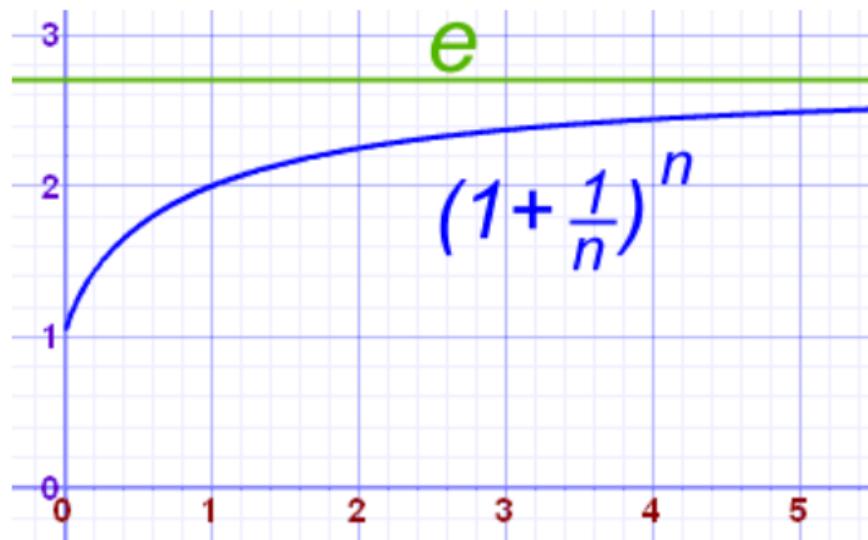
Estimated PI is 3.1415926533405423

Example

❖ Euler's number

$$e \approx \left(1 + \frac{1}{n}\right)^n$$

$$\lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n = e$$



n	$(1 + 1/n)^n$
1	2.00000
2	2.25000
5	2.48832
10	2.59374
100	2.70481
1,000	2.71692
10,000	2.71815
100,000	2.71827

Example

❖ Euler's number

$e = 2.71828$

Formula

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

1) Compute factorial

2) Compute sum

Example

❖ Euler's number

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

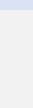
```
1. # aivietnam.ai
2.
3. # hàm tính giai thừa
4. def factorial(n):
5.     result = 1
6.
7.     for i in range(2, n+1):
8.         result = result*i
9.
10.    return result
11.
12. # hàm ước lượng số e
13. def estimate_e(n):
14.     result = 1
15.
16.     for i in range(1, n+1):
17.         result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

Example: Quadratic Root

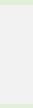
❖ Compute quadratic root for the number N

Newton Method

Set a value for x_0 ; $n = 0$
($x_0 = N/2$)



$$x_{n+1} = \frac{x_n + \frac{N}{x_n}}{2}$$



$n = n + 1$

Compute $\sqrt{9}$

$$N = 9$$

$$\text{set } x_0 = \frac{9}{2} = 4.5$$

$$n = 0$$

$$n = 0$$

$$x_1 = \frac{x_0 + \frac{N}{x_0}}{2} = \frac{4.5 + \frac{9}{4.5}}{2} = \frac{6.5}{2} = 3.25$$

$$n = 1$$

$$x_2 = \frac{x_1 + \frac{N}{x_1}}{2} = \frac{3.25 + \frac{9}{3.25}}{2} = \frac{6.019}{2} = 3.009$$

$$n = 2$$

$$x_3 = \frac{x_2 + \frac{N}{x_2}}{2} = \frac{3.009 + \frac{9}{3.009}}{2} = 3.00001$$

Example: Quadratic Root

❖ Compute quadratic root for the number N

Newton Method

Set a value for x_0 ; $n = 0$
 $(x_0 = N/2)$

$$x_{n+1} = \frac{x_n + \frac{N}{x_n}}{2}$$

$n = n + 1$

```
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

If-Else Condition and Math (1)

Input: a and b

if $a = 0$ then $y = b^2$
if $a = K$ then $y = \sqrt{b}$



Input: a and b

$$y = \frac{a}{K}(\sqrt{b}) + \frac{(K-a)}{K}b^2$$

```
1 import math
2
3 def function_1(a,b):
4     # a constant
5     K = 10
6
7     result = 0
8     if a==0:
9         result = b*b
10    elif a==K:
11        result = math.sqrt(b)
12
13    return result
14
15 print(function_1(0, 4))
16 print(function_1(10, 4))
```

16
2.0

```
1 import math
2
3 def function_2(a,b):
4     # a constant
5     K = 10
6
7     result = (a/K)*math.sqrt(b) + ((K-a)/K)*b*b
8
9     return result
10
11 print(function_2(0, 4))
12 print(function_2(10, 4))
```

16.0
2.0

If-Else Condition and Math (2)

Input: a and b

if $a > 0$ then

$$y = b^2$$

if $a < 0$ then

$$y = \sqrt{b}$$



Input: a and b

$$y = \frac{(b^2 + \sqrt{b})}{2} + \frac{(b^2 - \sqrt{b})}{2} |a|$$

```
1 import math
2
3 def function_3(a,b):
4     result = 0
5
6     if a>0:
7         result = b*b
8     elif a<0:
9         result = math.sqrt(b)
10
11    return result
12
13 print(function_3(2, 4))
14 print(function_3(-2, 4))
```

16
2.0

```
1 import math
2
3 def function_4(a,b):
4     result = (b*b + math.sqrt(b))/2
5     + (b*b - math.sqrt(b))/2 * (abs(a)/a)
6
7     return result
8
9 print(function_4(2, 4))
10 print(function_4(-2, 4))
```

16.0
2.0

If-Else Condition and Math (3)

Input: a and b

if $a = 0$ then $y = b^2$

if $a = 1$ then $y = \sqrt{b}$

if $a = 2$ then $y = b$



Input: a and b

Convert a into a one-hot vector v

$$u = [b^2 \quad \sqrt{b} \quad b]$$

$$y = v \cdot u$$

$$\begin{aligned}v &= [1 \quad 0 \quad 0] \\v &= [0 \quad 1 \quad 0] \\v &= [0 \quad 0 \quad 1]\end{aligned}$$

```

1 import math
2
3 def function_5(a,b):
4     result = 0
5
6     if a==0: result = b*b
7     elif a==1: result = math.sqrt(b)
8     elif a==2: result = b
9
10    return result
11
12 print(function_5(0, 4))
13 print(function_5(1, 4))
14 print(function_5(2, 4))

```

16
2.0
4

```

1 import math
2 import numpy as np
3
4 def function_6(a,b):
5     vector_v = np.array([0, 0, 0])
6     vector_v[a] = 1
7
8     vector_u = np.array([b*b, math.sqrt(b), b])
9
10    result = np.dot(vector_v, vector_u)
11    return result
12
13 print(function_6(0, 4))
14 print(function_6(1, 4))
15 print(function_6(2, 4))

```

16.0
2.0
4.0

Sigmoid Function

Sigmoid function

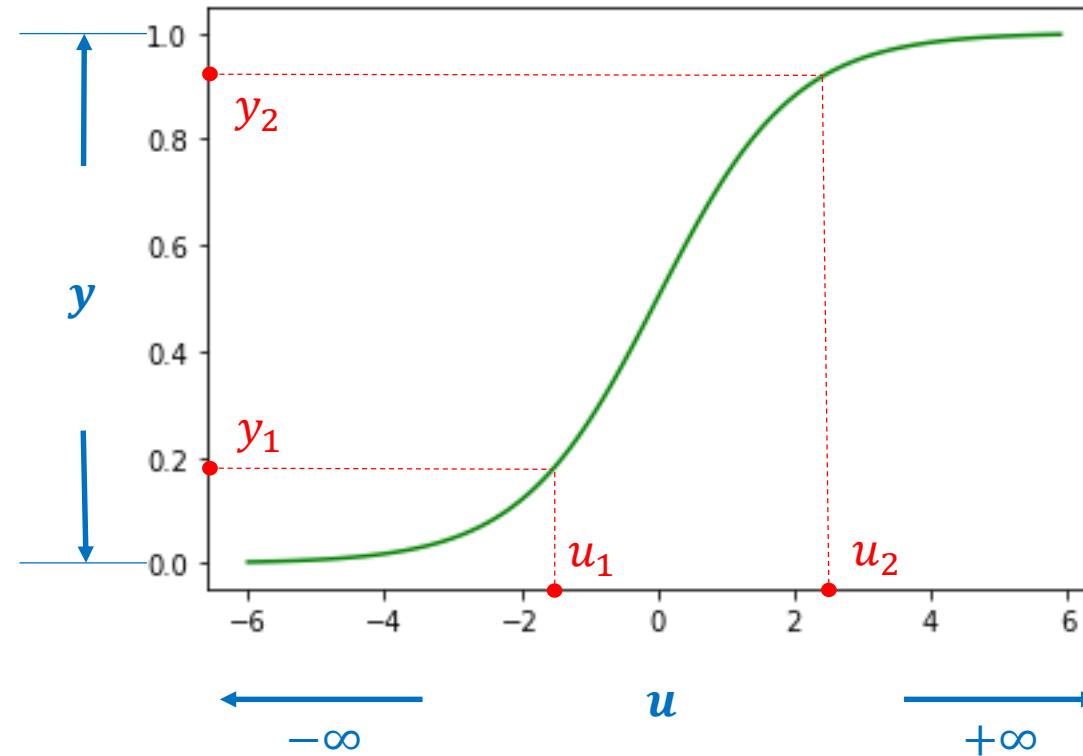
$$y = \sigma(u) = \frac{1}{1 + e^{-u}}$$

$$u \in (-\infty, +\infty)$$

$$y \in (0, 1)$$

Property

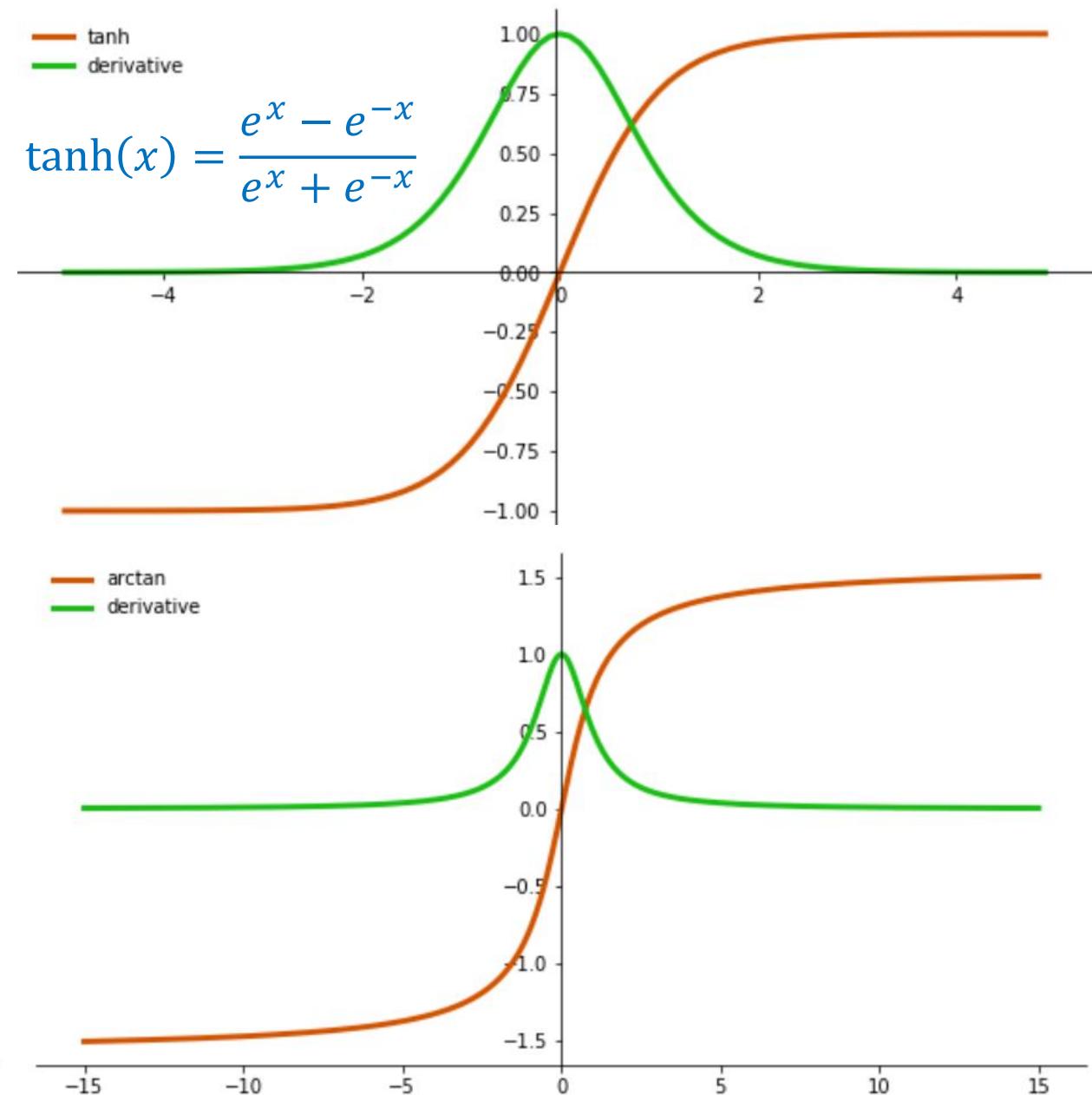
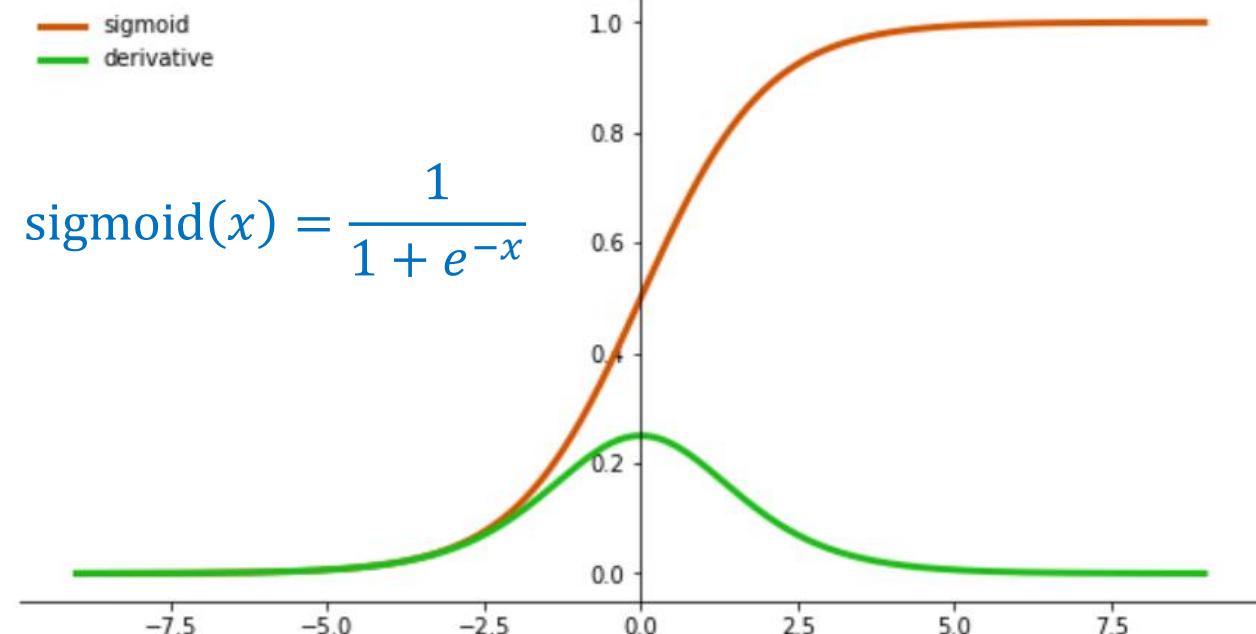
$$\forall u_1, u_2 \in [a, b] \text{ và } u_1 \leq u_2 \rightarrow \sigma(u_1) \leq \sigma(u_2)$$



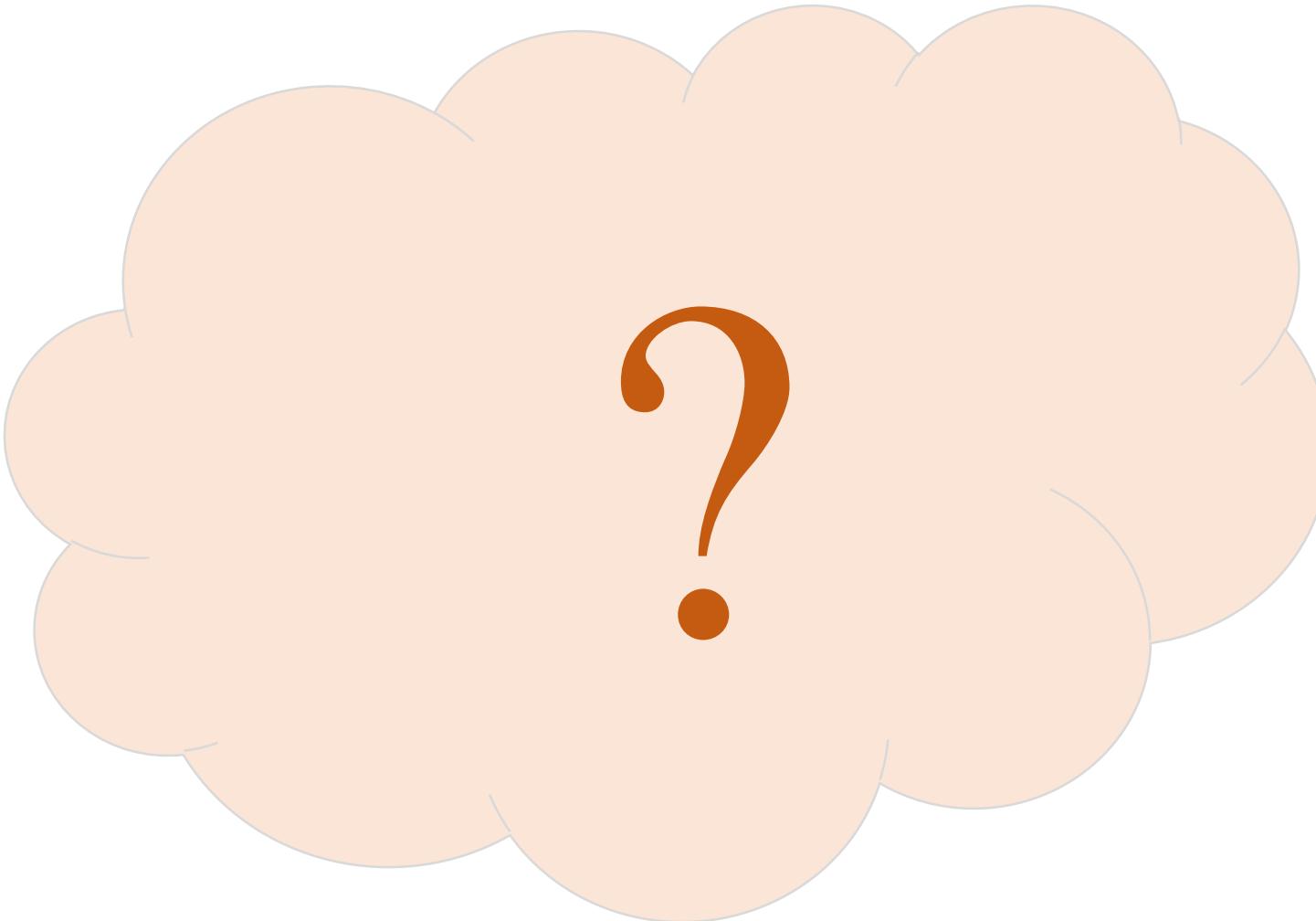
Activation Functions

S-shaped curves

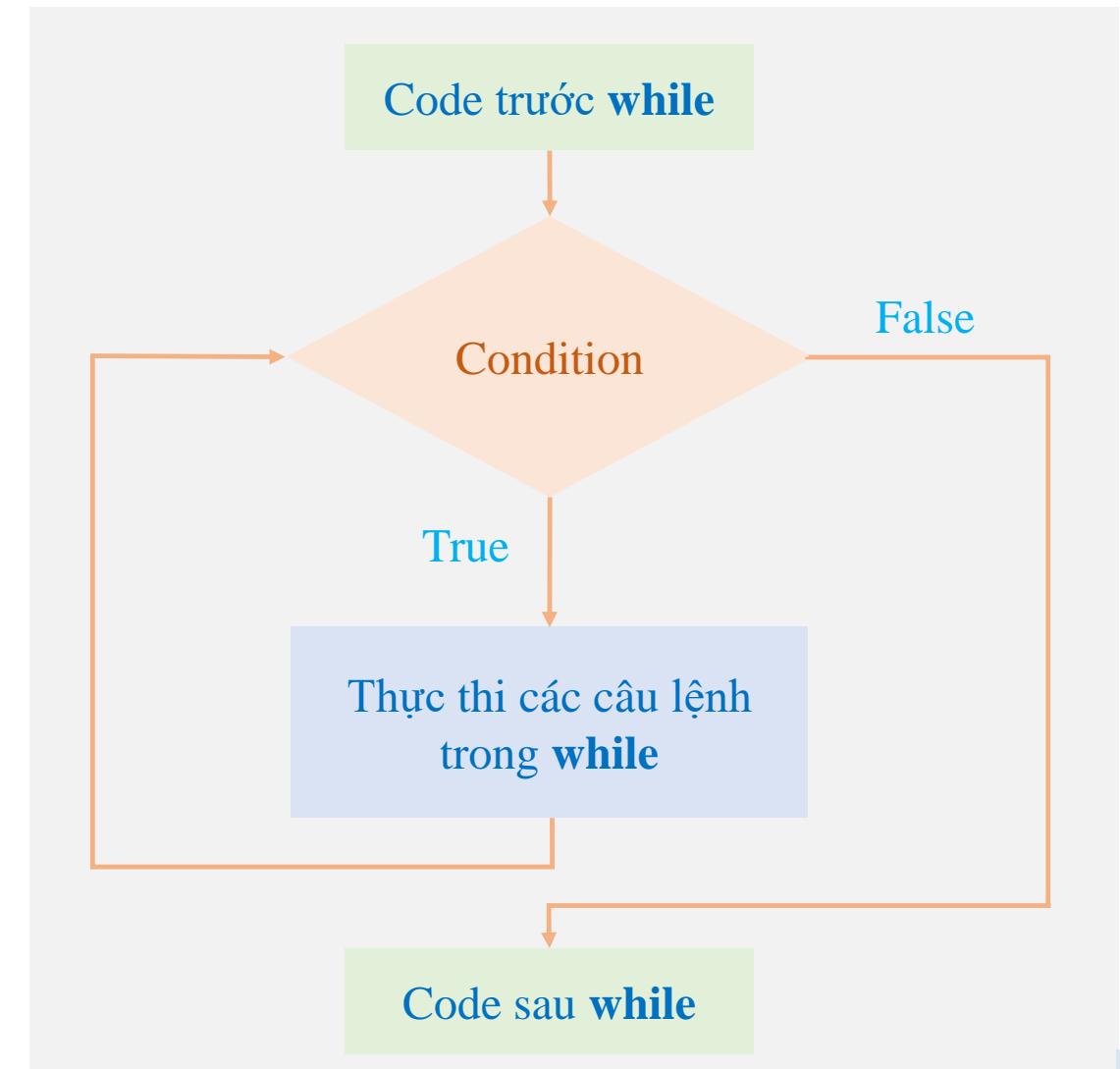
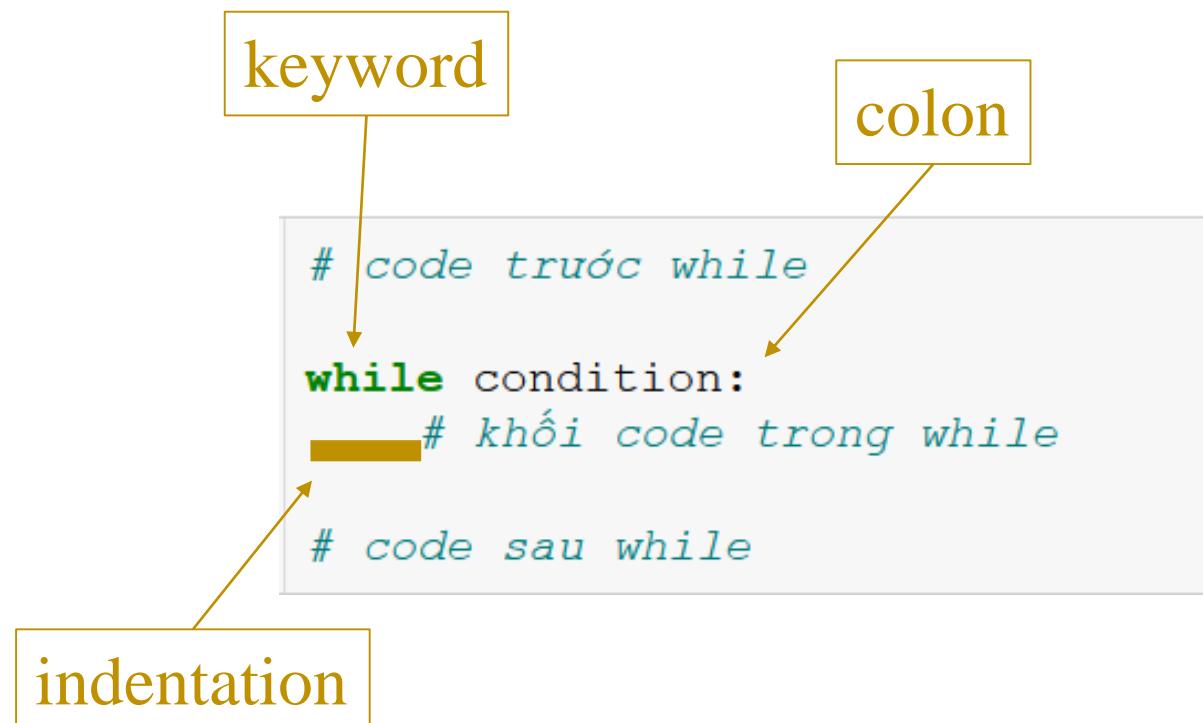
Map x values into small ranges



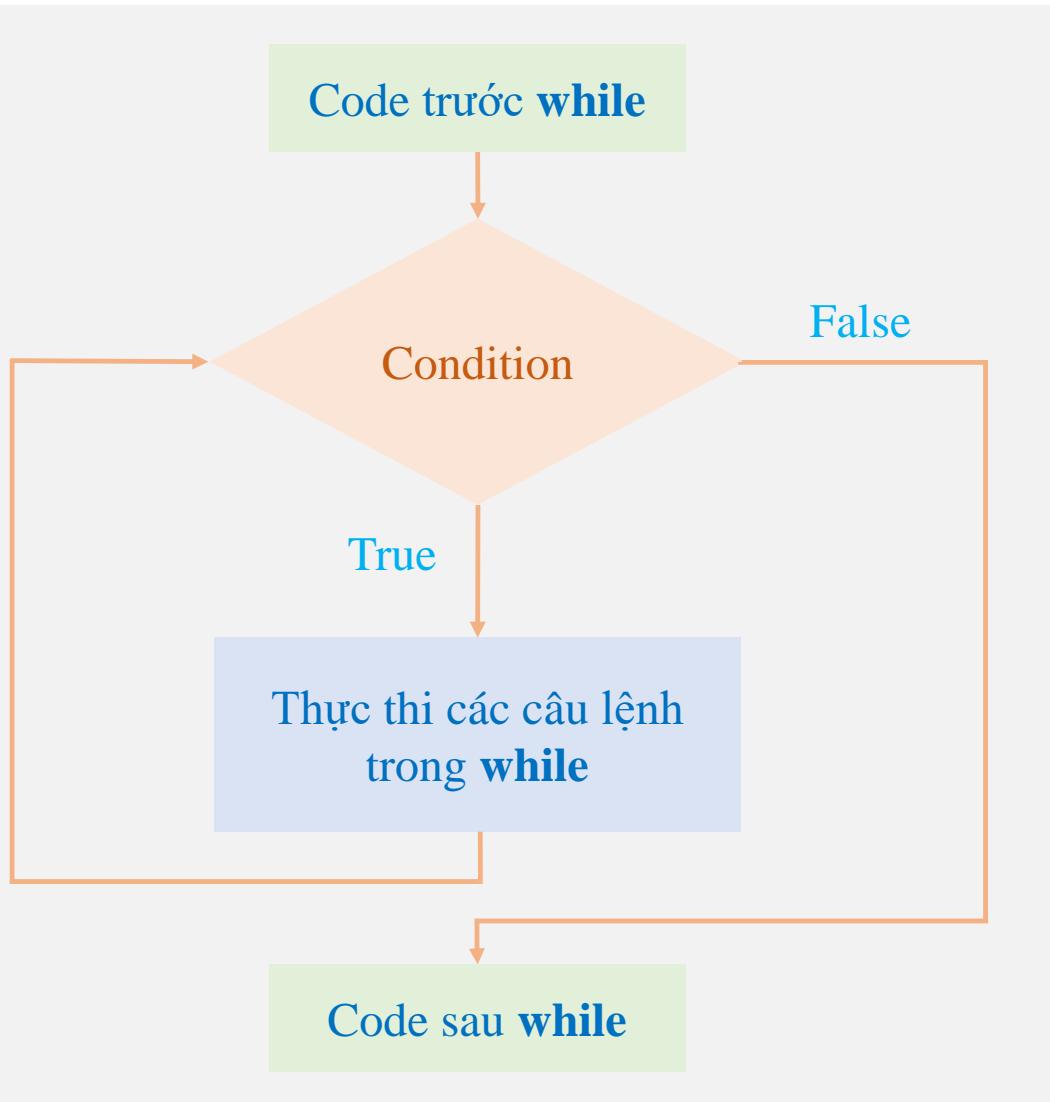
Questions



While Loop



While Loop



```
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

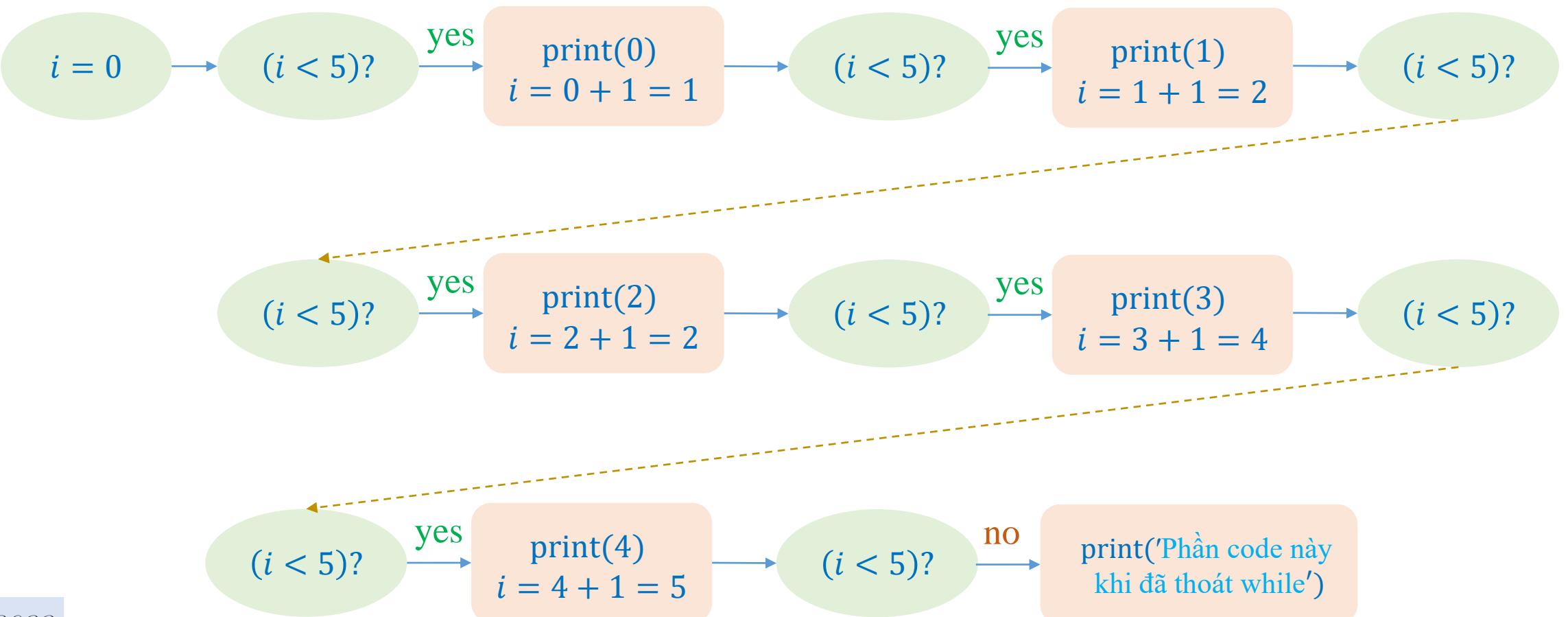
Phần code này khi đã thoát while

```

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')

```

While Loop



While Loop

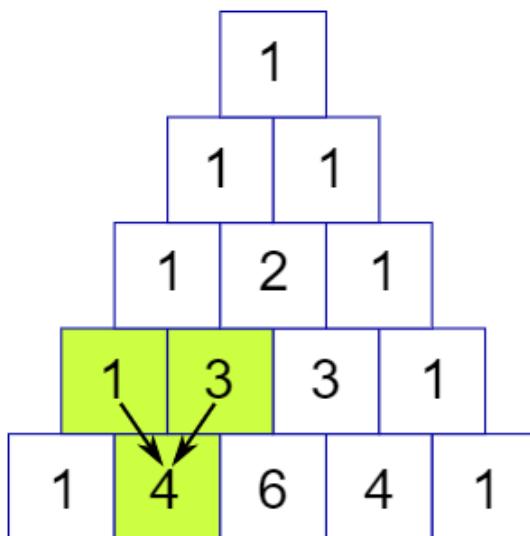
while-True-break

```
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
```

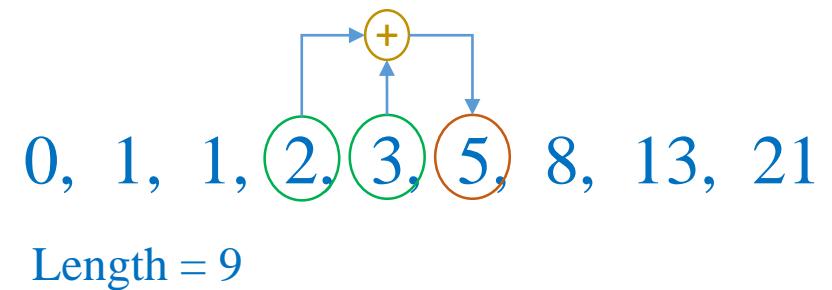
Exercises

Pascal's Triangle

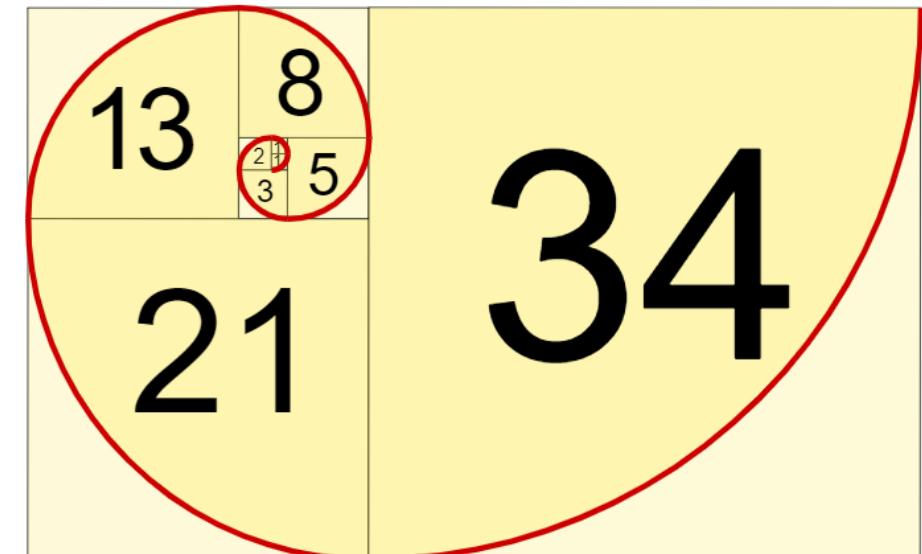


Level = 5

Fibonacci Sequence



Length = 9



Exercises

$$\sin(x) \approx \sum_{n=0}^{\infty} (-1)^n \frac{x^{(2n+1)}}{(2n+1)!} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \dots$$

$$\cos(x) \approx \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n}}{(2n)!} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \frac{x^{10}}{10!} + \dots$$

$$e^x \approx \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + \frac{x^1}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \dots$$

Exercises

$$\sinh(x) \approx \sum_{n=0}^{\infty} \frac{x^{(2n+1)}}{(2n+1)!} = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \frac{x^7}{7!} + \frac{x^9}{9!} + \dots$$

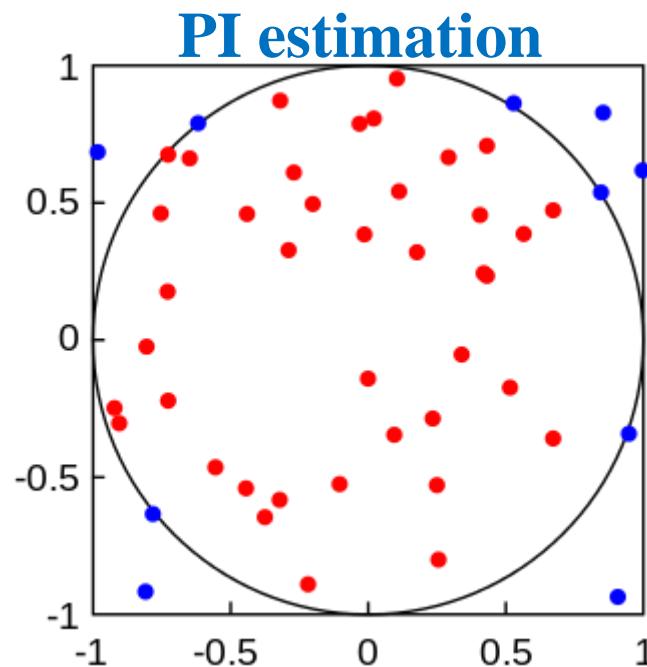
$$\cosh(x) \approx \sum_{n=0}^{\infty} \frac{x^{2n}}{(2n)!} = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \frac{x^6}{6!} + \frac{x^8}{8!} + \frac{x^{10}}{10!} + \dots$$

Exercises

E estimation

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

Simulation of coin tossing



Compute quadratic root for the number N

Newton Method

Set a value for x_0 ; $n = 0$
($x_0 = N/2$)

$$x_{n+1} = \frac{x_n + \frac{N}{x_n}}{2}$$

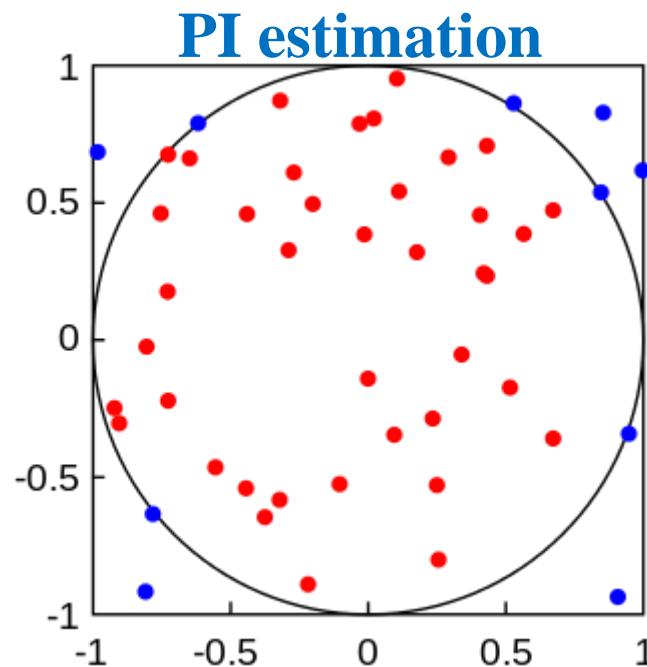
$$n = n + 1$$

Recursive

E estimation

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

Simulation of coin tossing



Compute quadratic root for the number N

Newton Method

Set a value for x_0 ; $n = 0$
($x_0 = N/2$)

$$x_{n+1} = \frac{x_n + \frac{N}{x_n}}{2}$$

$n = n + 1$

Common Errors

❖ Error 1

```
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

Common Errors

❖ Error 2

```
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
```

Common Errors

❖ Error 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
```

Common Errors

❖ Error 4

```
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`

Common Errors

❖ Error 5

```
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

```
<ipython-input-5-f1e2455fae51> in <module>
    9
   10 # tính giá trị c
--> 11 c = s + n
    12
   13 # in giá trị c
```

Traceback (most recent call last)

TypeError: must be **str**, not **int**

Common Errors

❖ Error 6

```
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'
```

Common Errors

❖ Error 7

```
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
```

Common Errors

❖ Error 8

```
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
```

Common Errors

❖ Error 9

```
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")?
```

Common Errors

❖ Error 10

```
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'
```

Common Errors

❖ Error 11

```
3.  
4.     l = [1, 2, 3, 4, 5]  
5.     print(l[0])  
6.     print(l[5])
```

```
1
```

```
-----  
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
```

Common Errors

❖ Error 12

```
4.     name = "aivietname.ai"  
5.     print(name[0])  
6.     print(name[50])
```

a

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

Common Errors

❖ Error 13

```
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
```

Common Errors

❖ Error 14

```
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
```

Common Errors

❖ Error 15

```
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
```

Common Errors

❖ Error 16

```
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'
```

Common Errors

❖ Error 17

```
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
```

Common Errors

❖ Error 18

```
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
```

Common Errors

❖ Error 19

```
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
```

Common Errors

❖ Error 20

```
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'
```

Common Errors

❖ Error 21

```
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
```

