

AI VIETNAM  
All-in-One Course

# Python

## An Introduction

Quang-Vinh Dinh  
Ph.D. in Computer Science

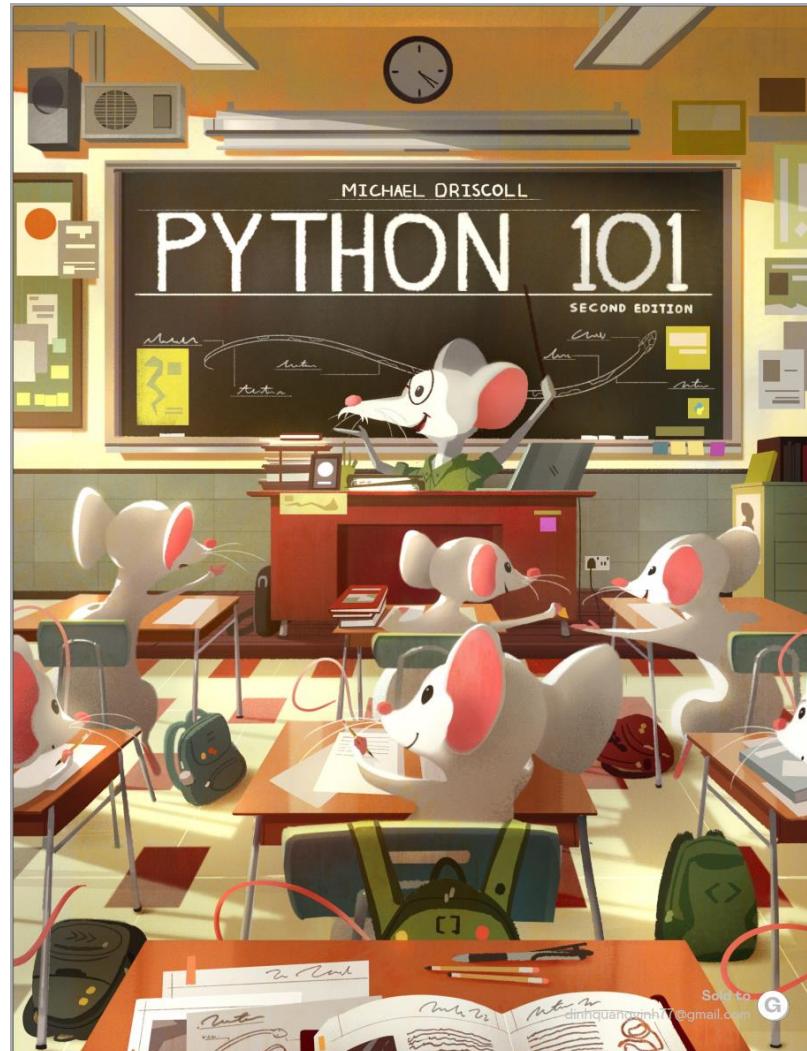
# Outline

- Programming Languages
- Python Environment
- Basic Data Types
- Functions
- Conditions
- Case Studies

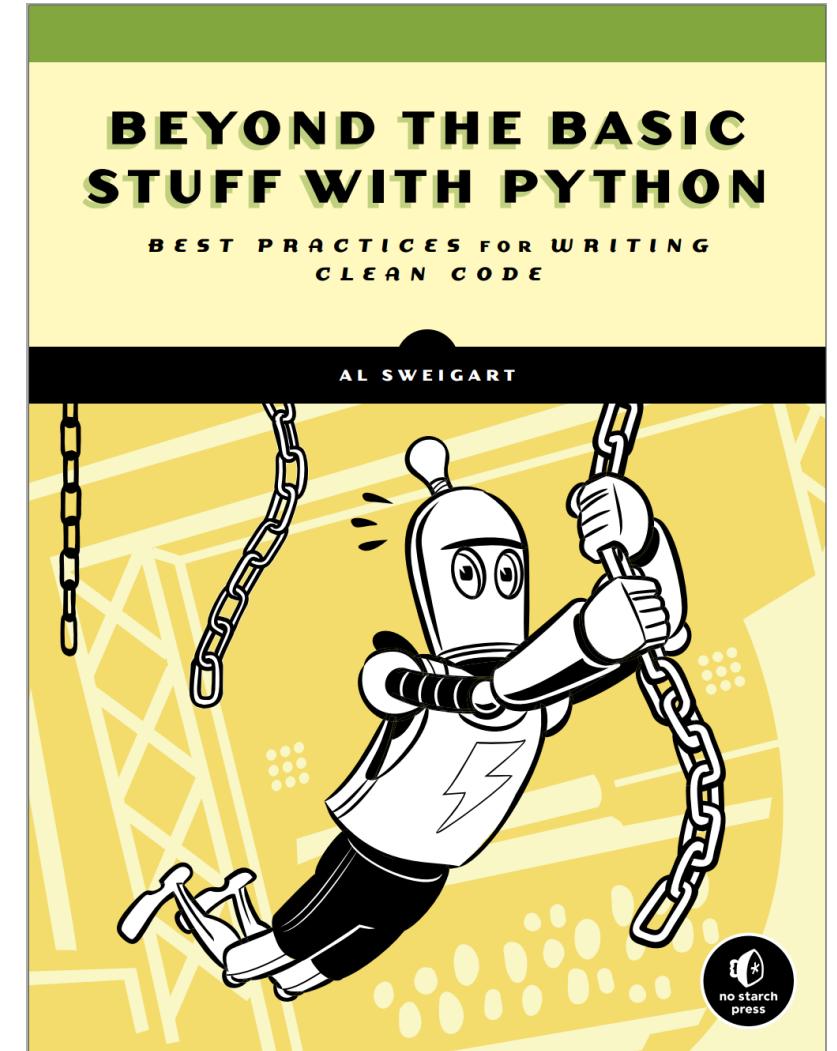
# References

## ❖ Books

<https://www.dropbox.com/sh/j8nvrijaw46tkgjx/AAA10WcRf2CMMmnW3gX4bVQta?dl=0>



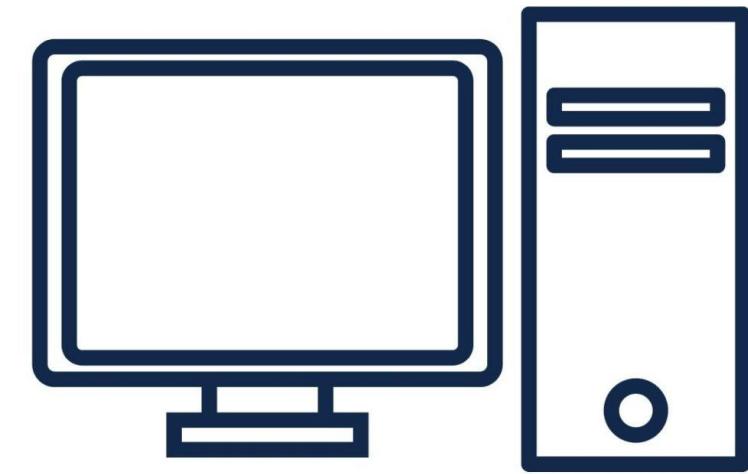
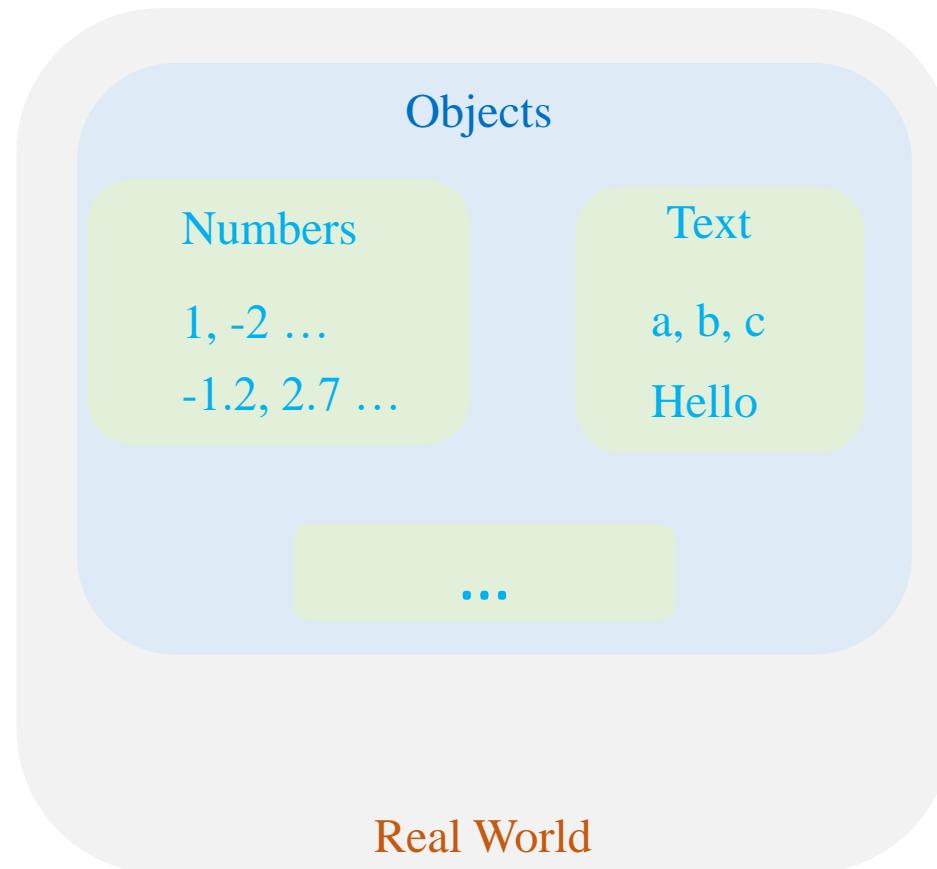
Python 101 - 2020



Beyond the Basic Stuff with Python

# Programming Languages

## ❖ What should a programming language provide?



A 10x10 grid of binary digits (0s and 1s) representing the encoded data. The 5th row contains several 1s highlighted in green, indicating specific data being transmitted or processed.

0	0	0	1	0	1	0	0	1	0
1	0	0	1	0	1	0	0	1	0
0	1	0	0	1	1	0	0	1	0
0	1	0	1	0	0	0	1	0	1
1	0	0	1	1	0	1	0	0	0
0	1	0	1	1	0	1	0	0	0
0	1	0	1	0	0	1	0	0	1
1	0	0	1	0	0	1	0	1	0
1	0	1	0	1	0	0	0	0	1
0	0	1	0	0	0	1	0	0	1

# Programming Languages

## ❖ What should a programming language provide?

A way to represent numbers using bits

With three bits

0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

8 different cases

Positive numbers

[0, 7]  
[8, 15]

Negative numbers

[-1, -8]  
[...]

Mixed

[-3, 4]

# Programming Languages

## ❖ What should a programming language provide?

A way to represent numbers using bits

With three bits

0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

8 different  
cases

Positive numbers

[0, 7]  
[8, 15]

Negative numbers

[-1, -8]  
[...]

Mixed

[-3, 4]

How to distinguish the cases?

# Programming Languages

## ❖ What should a programming language provide?

A way to represent numbers using bits

With three bits

+ Metadata

0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

8 different cases

Type 1  
Positive numbers

[0, 7]

Type 2  
Negative numbers

[-1, -8]

Type 3  
Mixed

[-3, 4]

# Programming Languages

## ❖ What should a programming language provide?

A way to represent numbers using bits

With eight bits

+ Metadata

256  
different  
cases

Type 1  
unsigned char

[0, 255]

Type 2  
char

[-127, 128]

0 | 0 | 0 | 0 | 0 | 1 | 1 | 0

Syntax for programmer

char a\_number = 6;

# Programming Languages

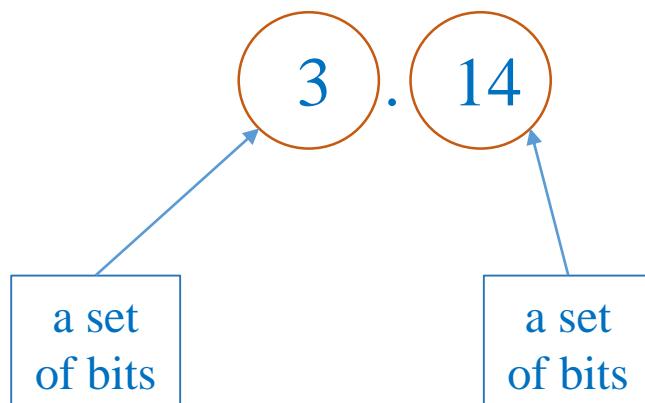
## ❖ What should a programming language provide?

How to represent floating-point numbers?

3.14

With 4 bytes  
(32 bits)  
float

With 8 bytes  
(64 bits)  
double



Syntax for programmer

float a\_number = 3.14;

double another\_number = 2.71;

# Programming Languages

## ❖ What should a programming language provide?

How to represent/store text using bits?

Aa Bb Cc Dd Ee Ff

iStock  
by Getty Images

iStock  
by Getty Images

Gg Hh Ii Jj Kk Ll Mm

Nn Oo Pp Qq Rr Ss Tt

Uu Vv Ww Xx Yy Zz

iStock  
by Getty Images

1 2 3 4 5 6 7 8 9 0

( - + = % @ " " ? ! & . , )

817933444

Need a mapping table!

# Programming Languages

## ❖ What should a programming language provide?

How to represent/store text using bits?

Syntax for programmer

```
char a_number = 'A';
```

A mapping table

Decimal: 65

0	1	0	0	0	0	0	1
---	---	---	---	---	---	---	---

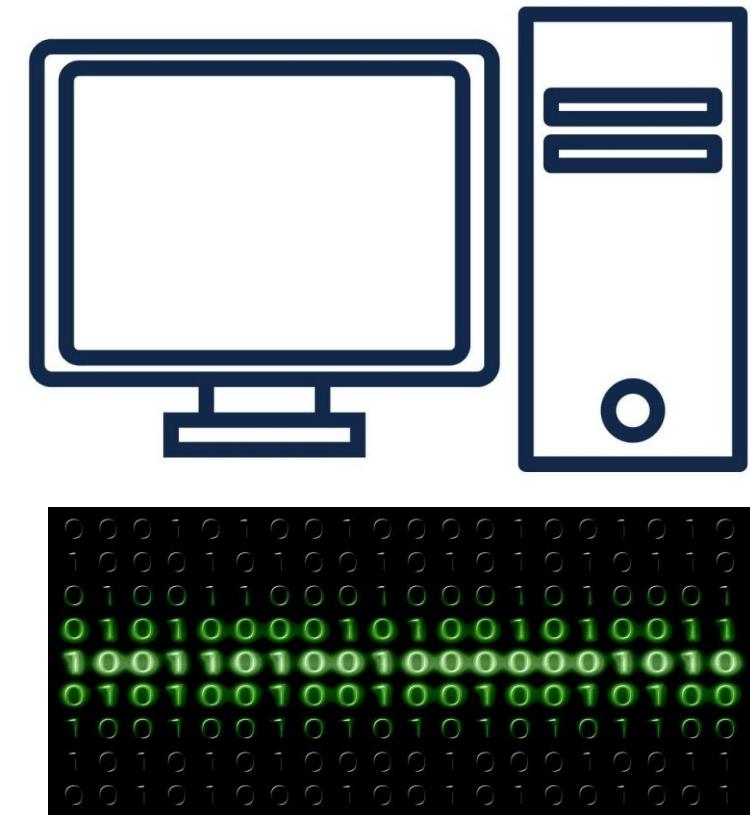
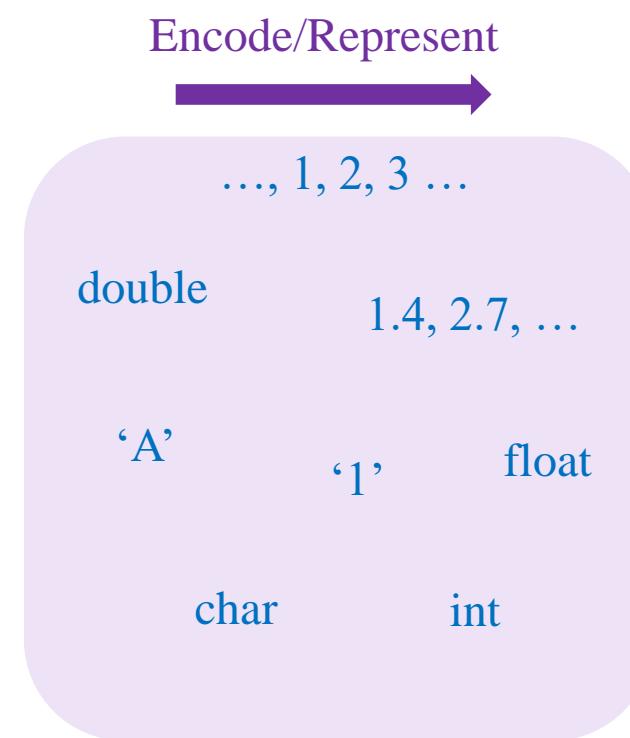
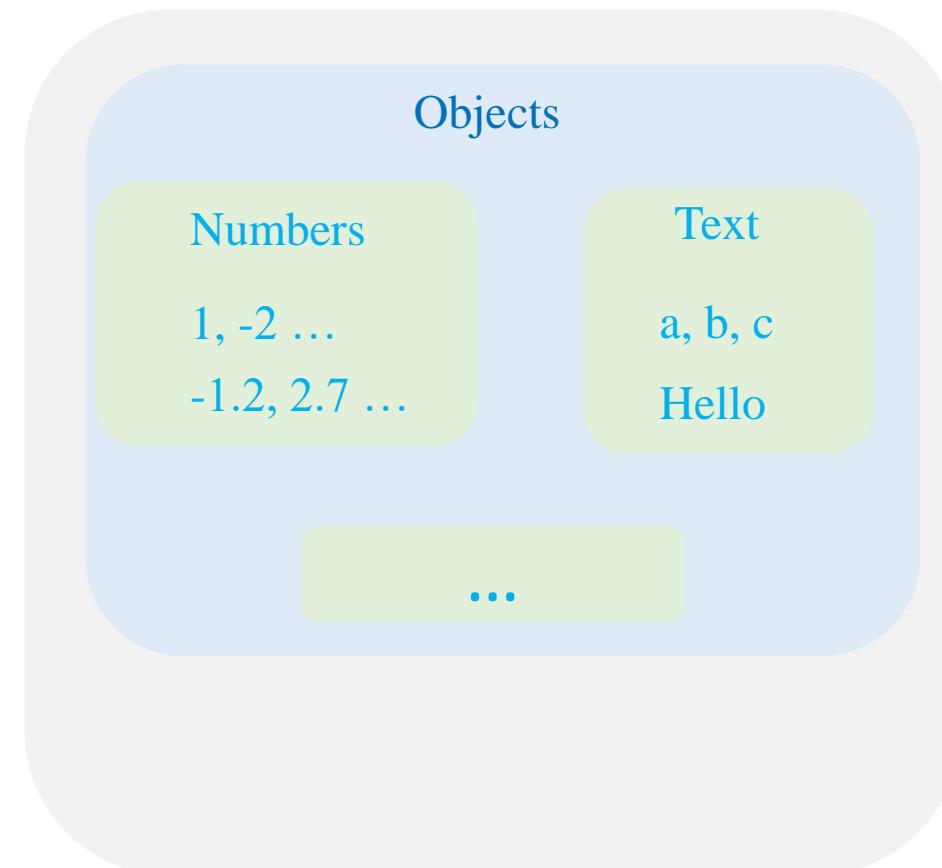
# ASCII TABLE

<https://commons.wikimedia.org/wiki/File:ASCII-Table-wide.svg>

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[END OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

# Programming Languages

## ❖ What should a programming language provide?



What's more???

# Programming Languages

## ❖ C++ and Python

```
1 #include <iostream>
2 #include <string>
3
4 int main() {
5     int number = 5;
6     printf("Number is %d\n", number);
7
8     char* greeting = "Hello";
9     printf("Greeting is %s\n", greeting);
10
11    return 0;
12 }
```

/tmp/f4nWZVcHql.o  
Number is 5  
Greeting is Hello

C++ example

```
1 variable = 5
2 print("Number is", variable)
3
4 variable = "Hello"
5 print("Greeting is", variable)
6
```

Number is 5  
Greeting is Hello

```
1 variable = 5
2 print(type(variable))
3
4 variable = "Hello"
5 print(type(variable))
```

<class 'int'>  
<class 'str'>

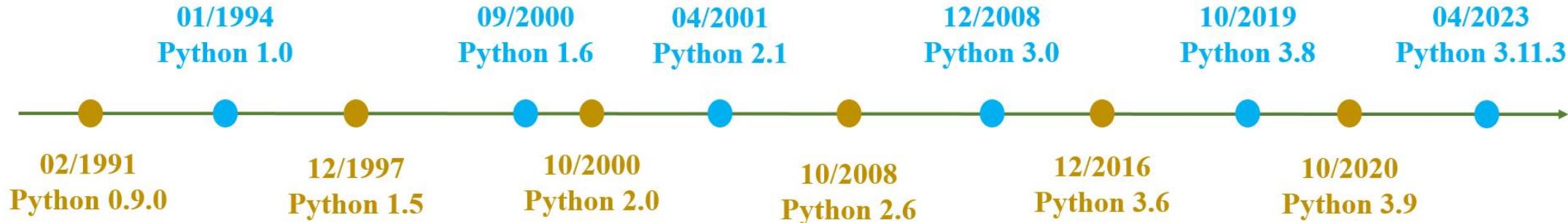
Python example

# Outline

- Programming Languages
- Python Environment
- Basic Data Types
- Functions
- Conditions
- Case Studies

# 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



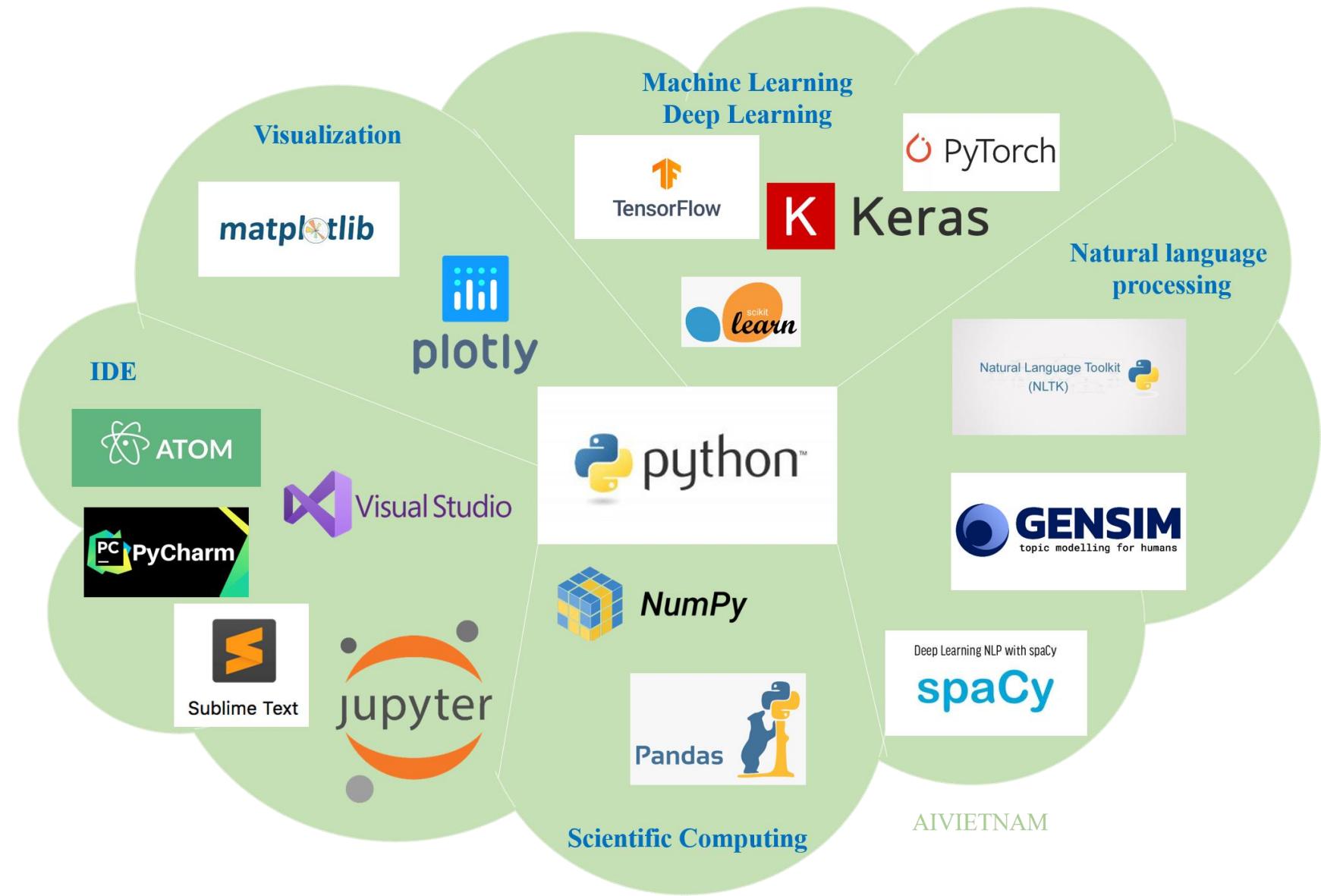
228,855 python packages  
(PyPI)



Hỗ trợ rất mạnh cho  
Data Science và Machine Learning

# Introduction

## ❖ Python ecosystem



# Introduction

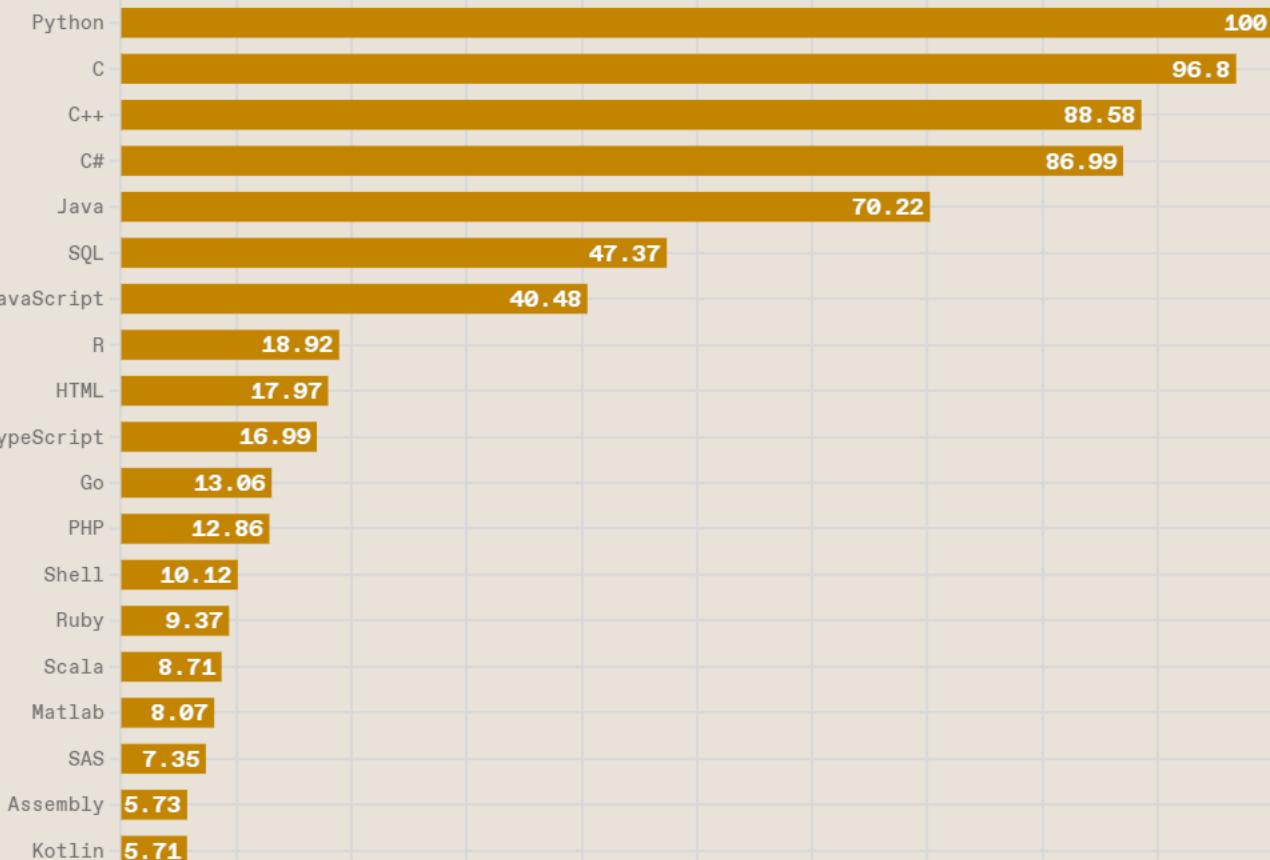
## Top Programming Languages (08/2022)

<https://spectrum.ieee.org/top-programming-languages-2022>

## Top Programming Languages 2022

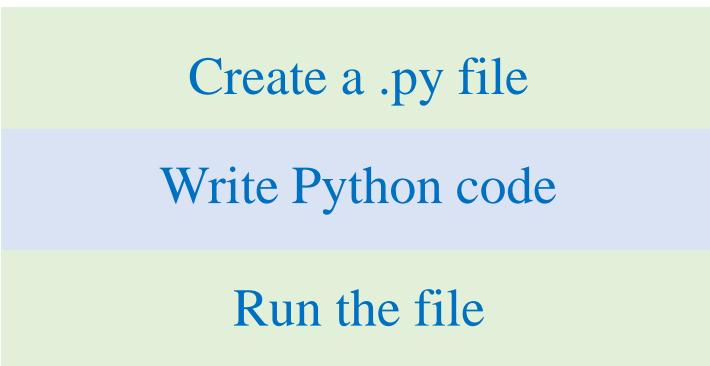
Click a button to see a differently weighted ranking

Spectrum Jobs Trending



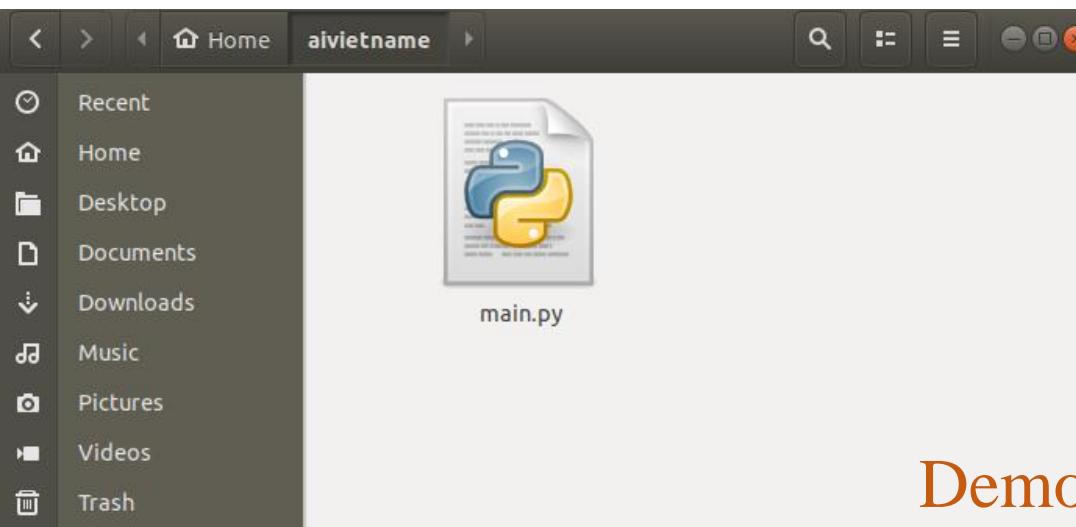
# Introduction

## ❖ First Python program



main.py

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

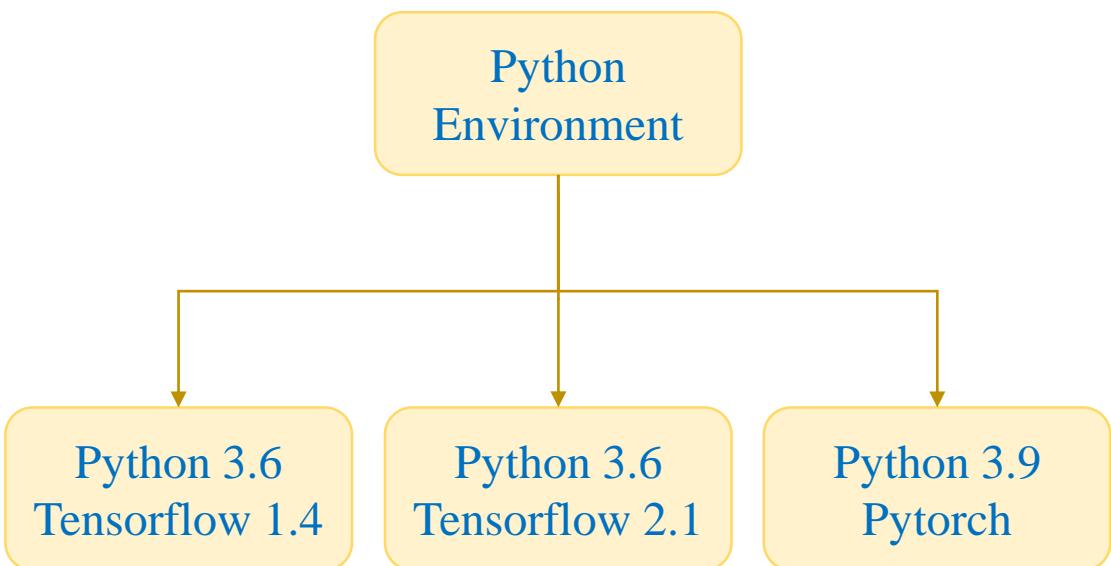


Demo

```
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
```

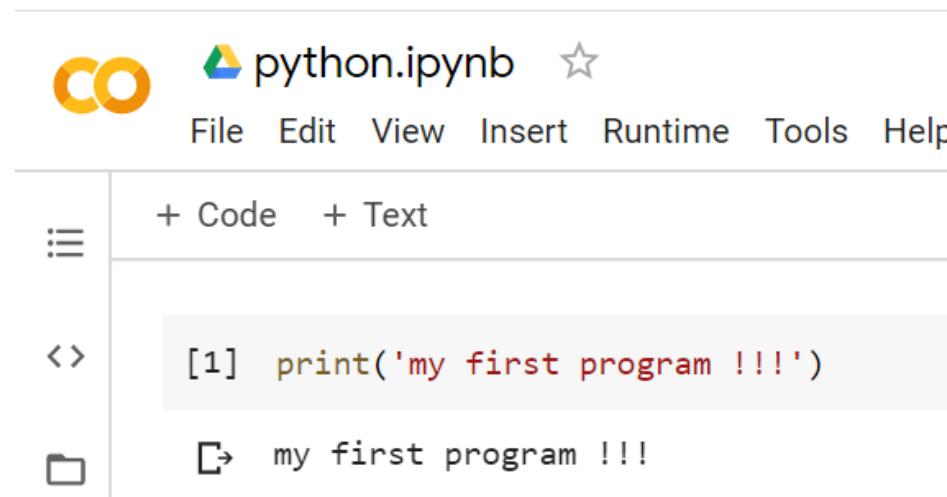
```
source a_name/bin/activate
```

```
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>

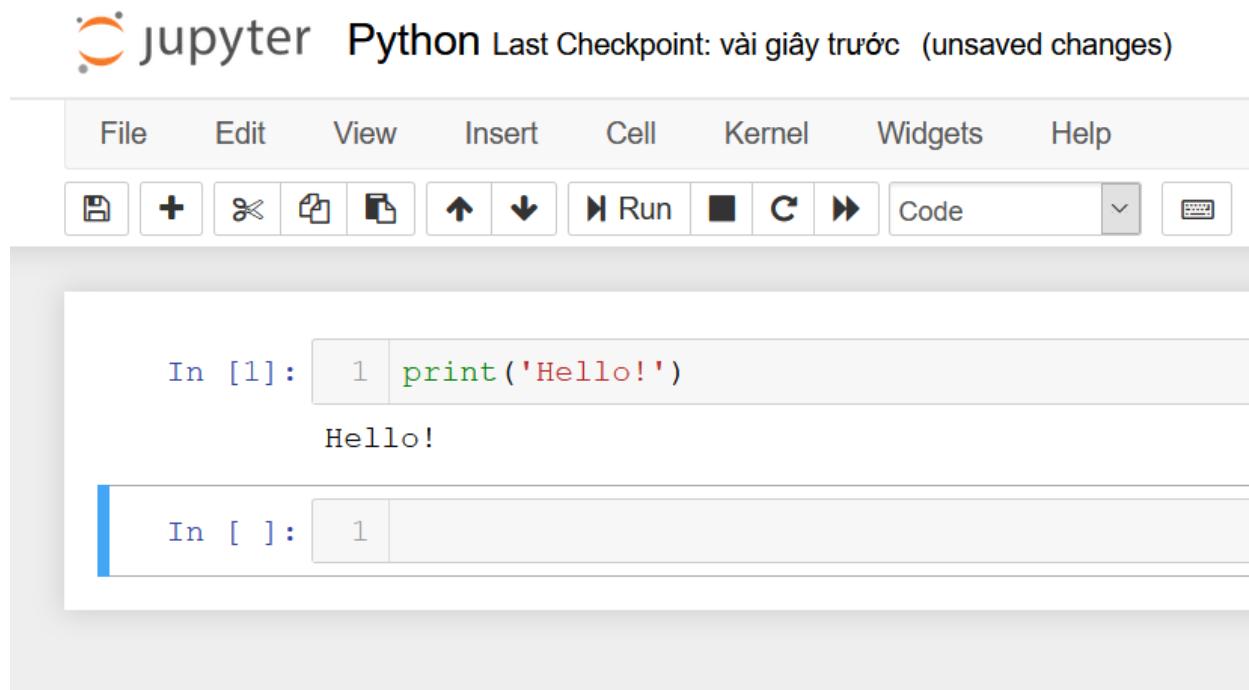


The screenshot shows a Google Colab interface. At the top, there's a logo consisting of orange 'CO' letters, the file name 'python.ipynb', and a star icon. Below the logo is a menu bar with 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. A horizontal line separates the menu from the main workspace. On the left side of the workspace, there's a sidebar with three icons: a grid for '+ Code', a text document for '+ Text', and a folder for files. In the main area, there's a code cell with a grey background. The cell contains the Python command `[1] print('my first program !!!')`. The output of the cell, which is the string 'my first program !!!', is displayed below it in a white box with a black border. The entire interface has a clean, modern look with a white background.

Demo

# Introduction

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



Demo

Setting up environment

```
conda create -n test_env2 python=3.9
```

```
conda activate an_env
```

```
conda deactivate
```

Some common commands

To check packages installed  
pip list

Open the jupyter program  
jupyter notebook

To install a package  
pip install package\_name

# Introduction

## ❖ Install new package

pip install matplotlib

```
1 pip install matplotlib
```

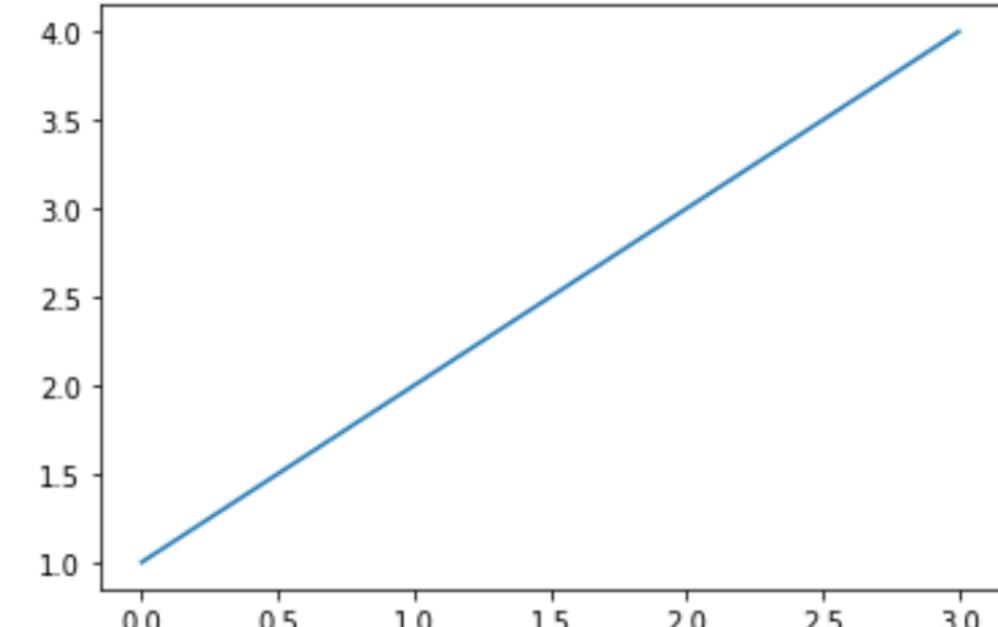
```
Collecting matplotlib
  Using cached matplotlib-3.3.4-cp36-cp36m-manylinux1_x86_64.whl (11.
Requirement already satisfied: python-dateutil>=2.1 in ./lib/python3.
Collecting cycler>=0.10
  Using cached cycler-0.10.0-py2.py3-none-any.whl (6.5 kB)
Collecting kiwisolver>=1.0.1
  Using cached kiwisolver-1.3.1-cp36-cp36m-manylinux1_x86_64.whl (1.1
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.6
4.7)
Collecting numpy>=1.15
  Using cached numpy-1.19.5-cp36-cp36m-manylinux2010_x86_64.whl (14.8
Collecting pillow>=6.2.0
  Downloading Pillow-8.3.1-cp36-cp36m-manylinux_2_5_x86_64.manylinux1
    [██████████] | 3.0 MB 2.6 MB/s eta 0:00:01
Requirement already satisfied: six in ./lib/python3.6/site-packages (
Installing collected packages: pillow, numpy, kiwisolver, cycler, mat
Successfully installed cycler-0.10.0 kiwisolver-1.3.1 matplotlib-3.3.
```

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

```
ModuleNotFoundError Traceba
<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'
```

```
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 values

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

Demo

variable\_name = variable\_value

## 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

# 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

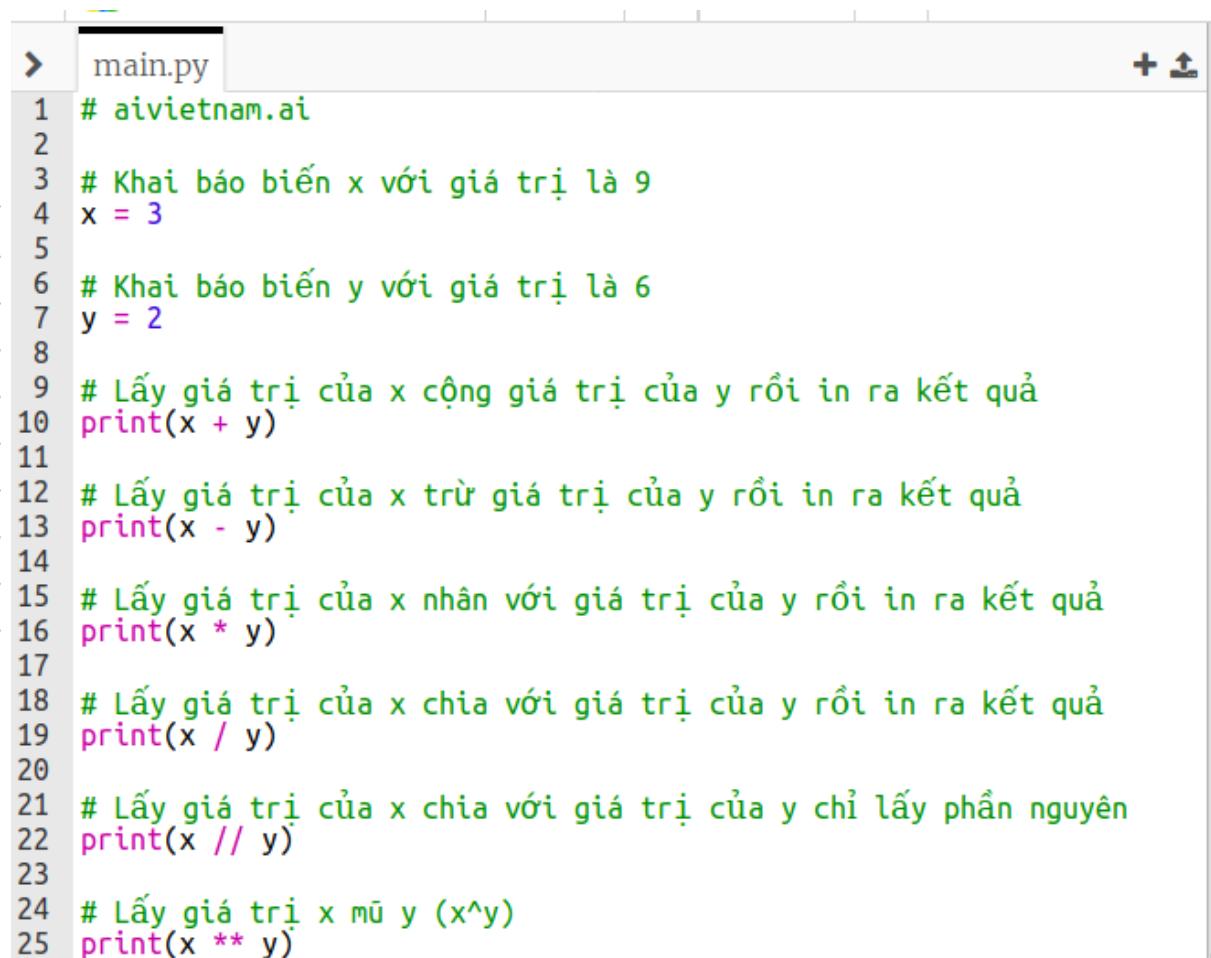
```
1 # generate random integer numbers in [a, b]
2
3 import random
4
5 print(random.randint(1, 9))
6 print(random.randint(1, 9))
7 print(random.randint(1, 9))
8 print(random.randint(1, 9))
```

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

7  
3  
6  
4

# 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:

```
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)
```

Demo

# Example

## Celsius to Fahrenheit conversion

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

## Celsius to Fahrenheit conversion

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

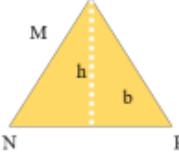
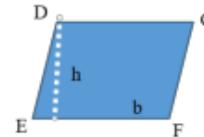
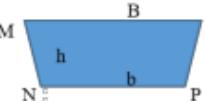
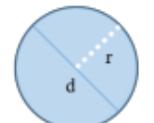
```
1. # input
2. temp_c = float(input('Nhập nhiệt độ theo độ C: '))
3.
4. # process
5. temp_f = ((9 / 5) * temp_c) + 32
6.
7. # output
8. print('Nhiệt độ F là: ', temp_f)
```

```
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

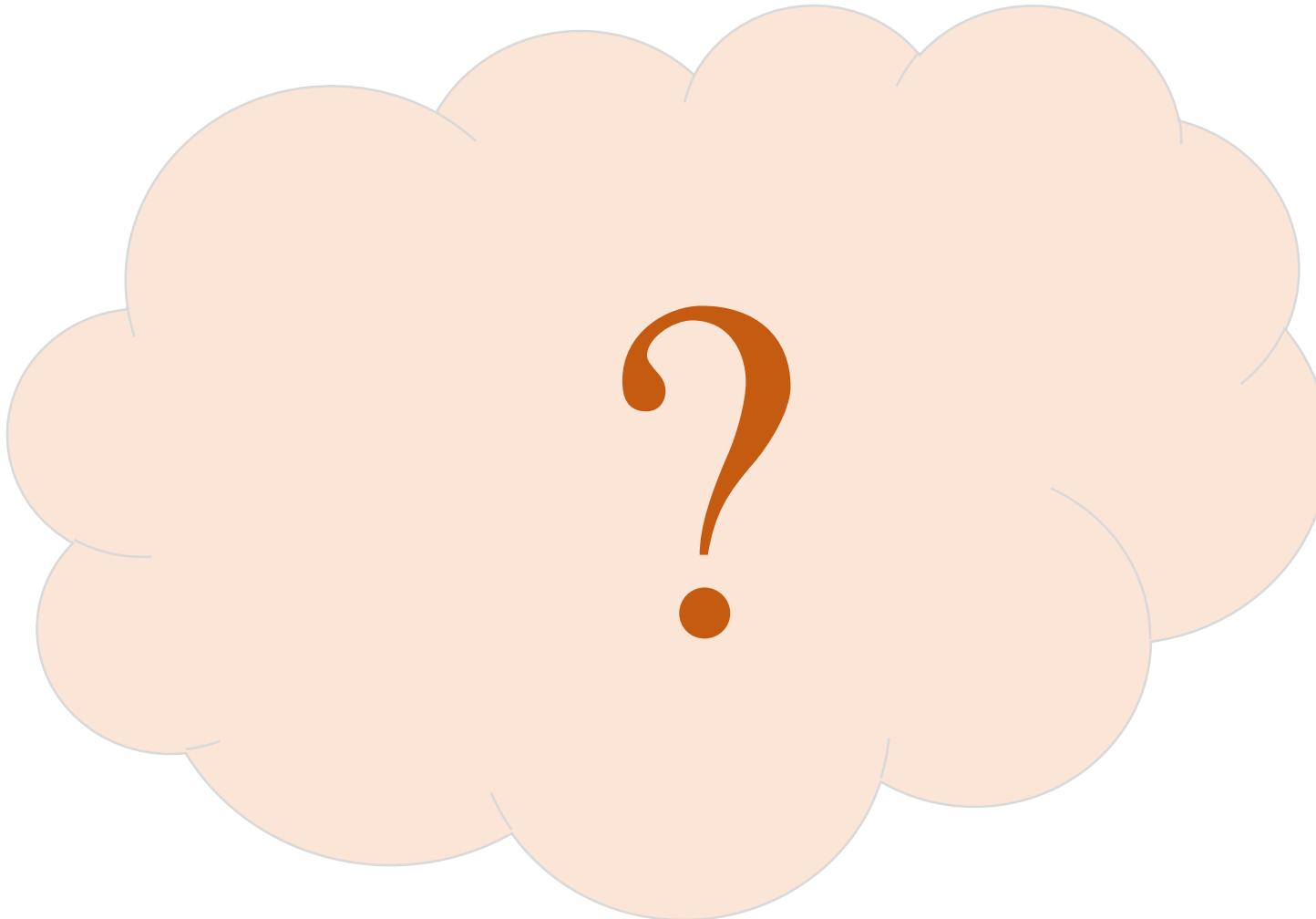
## Circumference and area of common shapes

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

---



# Functions

## Notes for function construction

Define function name

Lowercase with underscores  
and begin with a verb

Indentation

Use 4 spaces for indentation

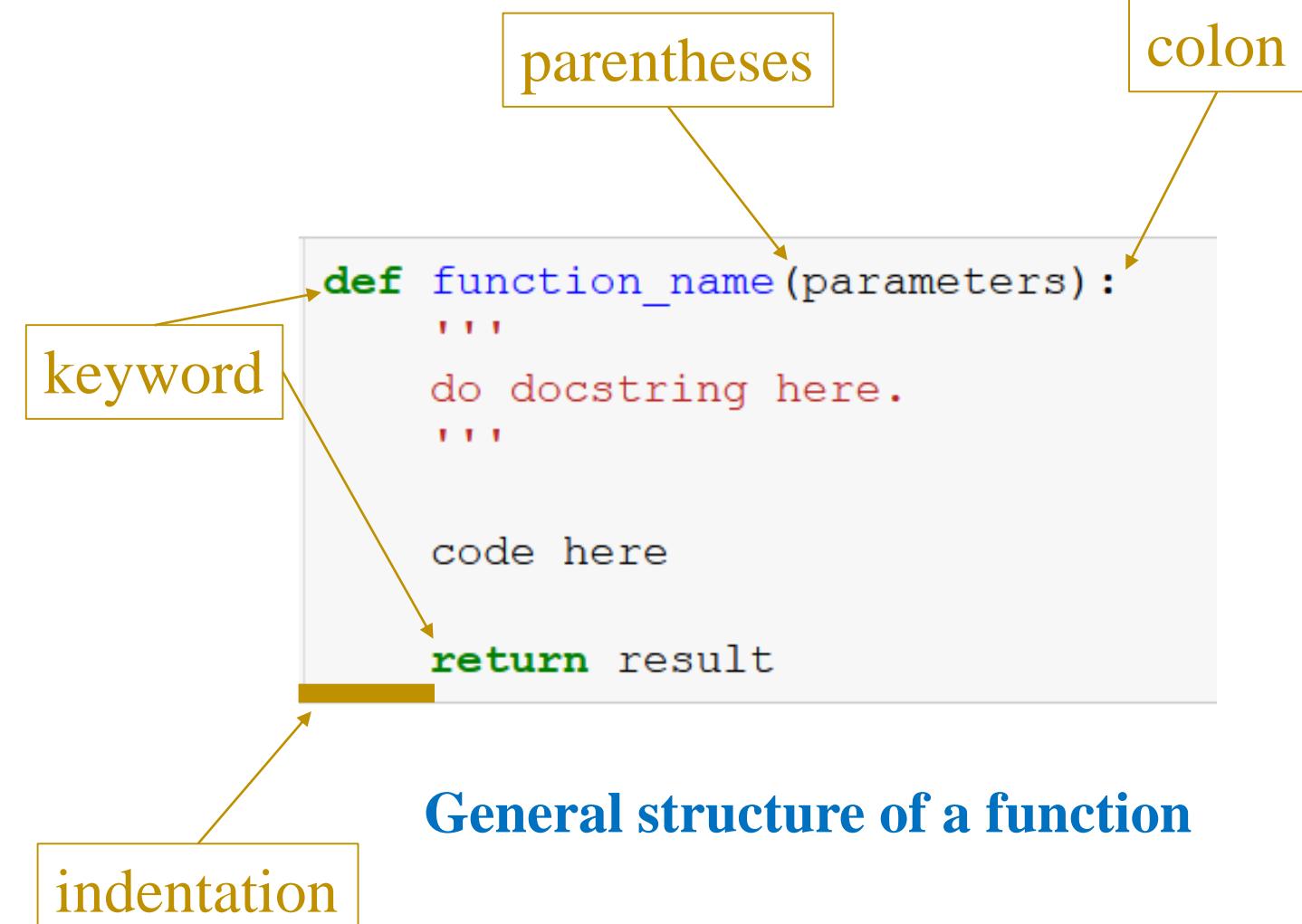
Determine function parameters

Input data help the function done

Do docstring

Explain and describe the function

Output of the function



Demo (rectangle)

# Functions

```
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

```
def compute_rectangle_area(height, width):
    """
    This function aims to compute area for a rectangle.

    height -- the height of the rectangle
    width -- the width of the rectangle

    This function returns the area of the rectangle
    """

    area = height*width

    return area
```

# Functions

## ❖ Default values

### Default values



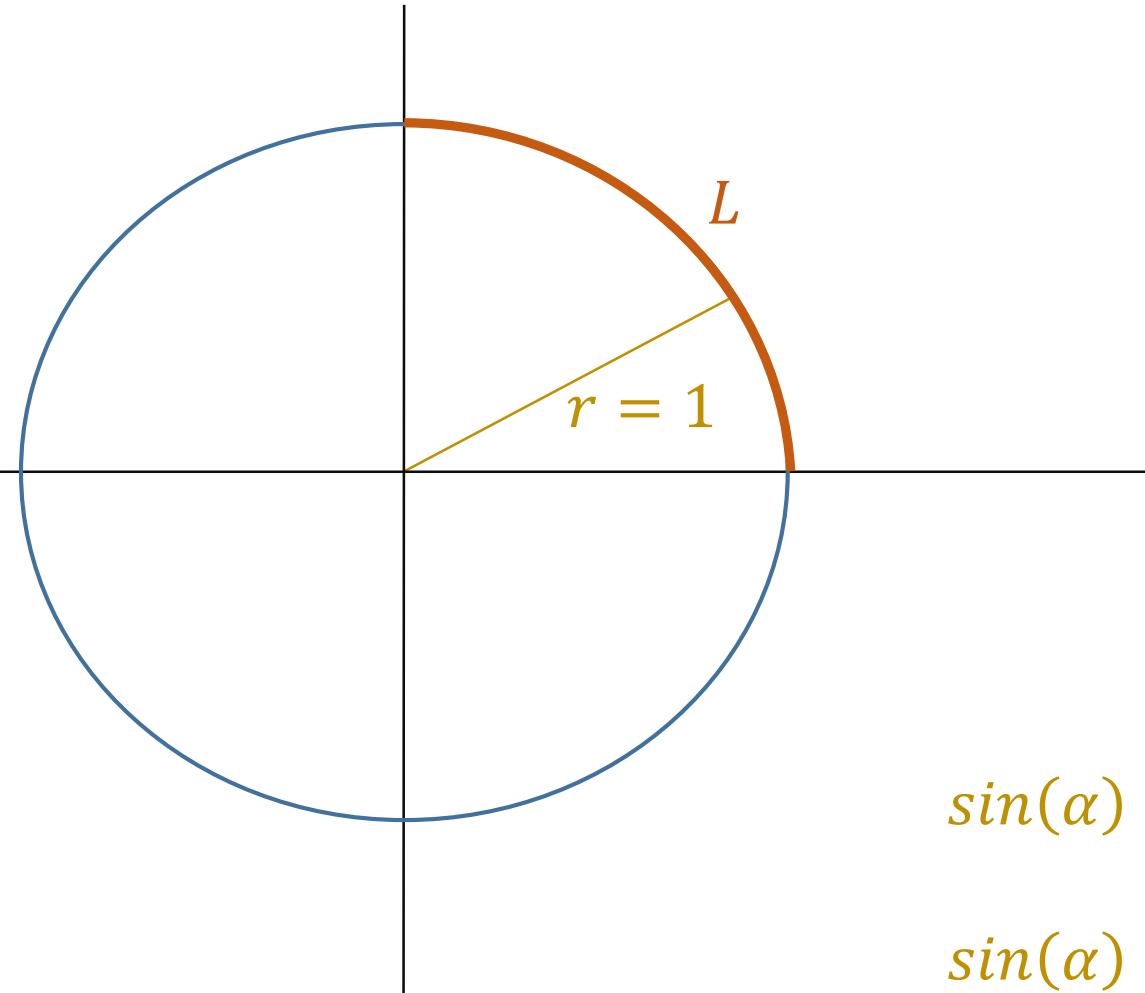
```
1 def compute_rectangle_area(height=0, width=0):
2     """
3         This function aims to compute area for a rectangle.
4
5         height -- the height of the rectangle
6         width -- the width of the rectangle
7
8         This function returns the area of the rectangle
9     """
10
11    area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

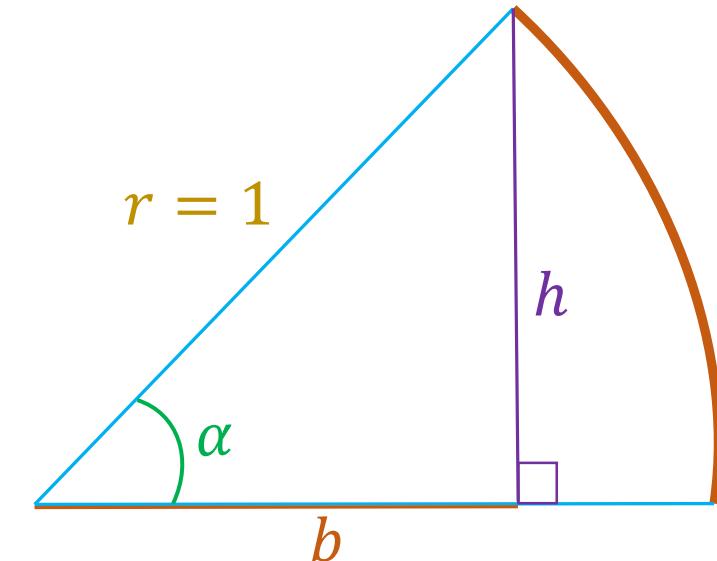
# Example

- ❖ Compute the circumference of a unit circle



$$\sin(\alpha) = \frac{h}{r}$$

$$\sin(\alpha) = h$$

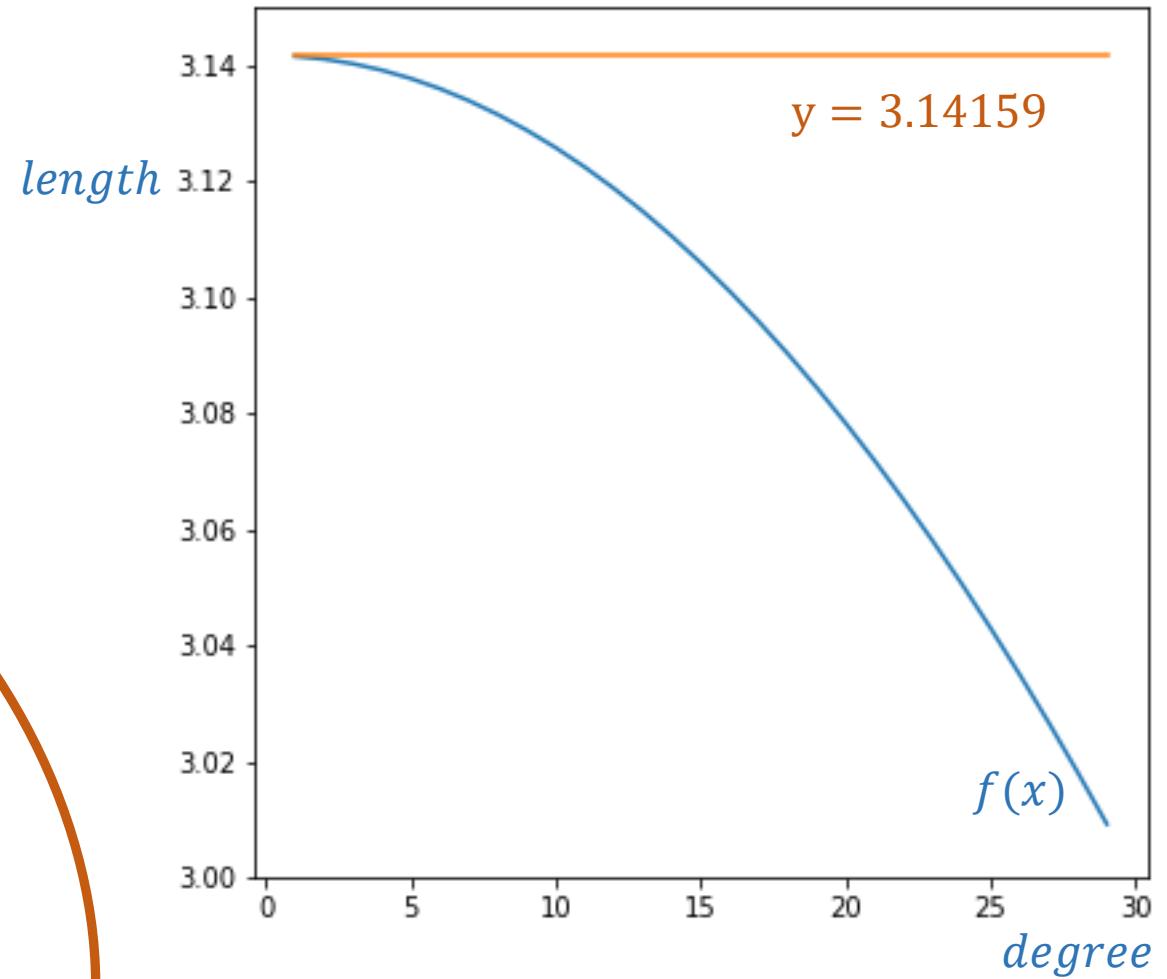
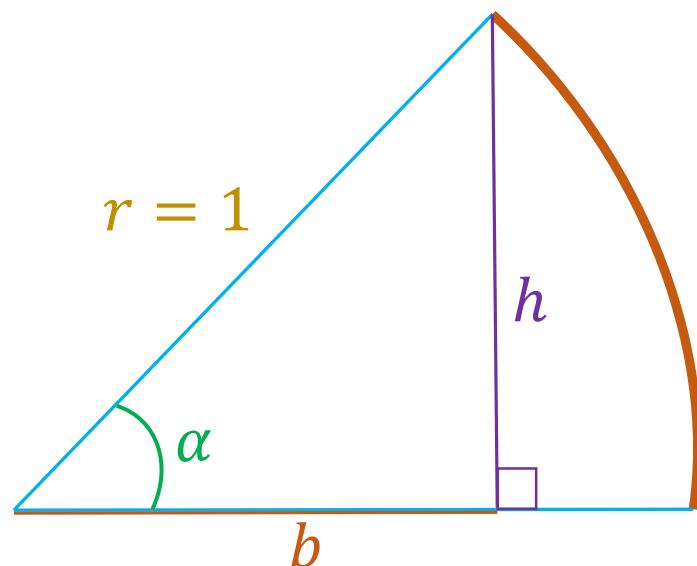


Divide the length into m segments

$$L = m * \sin(\alpha)$$

## ❖ Compute the circumference of a unit circle

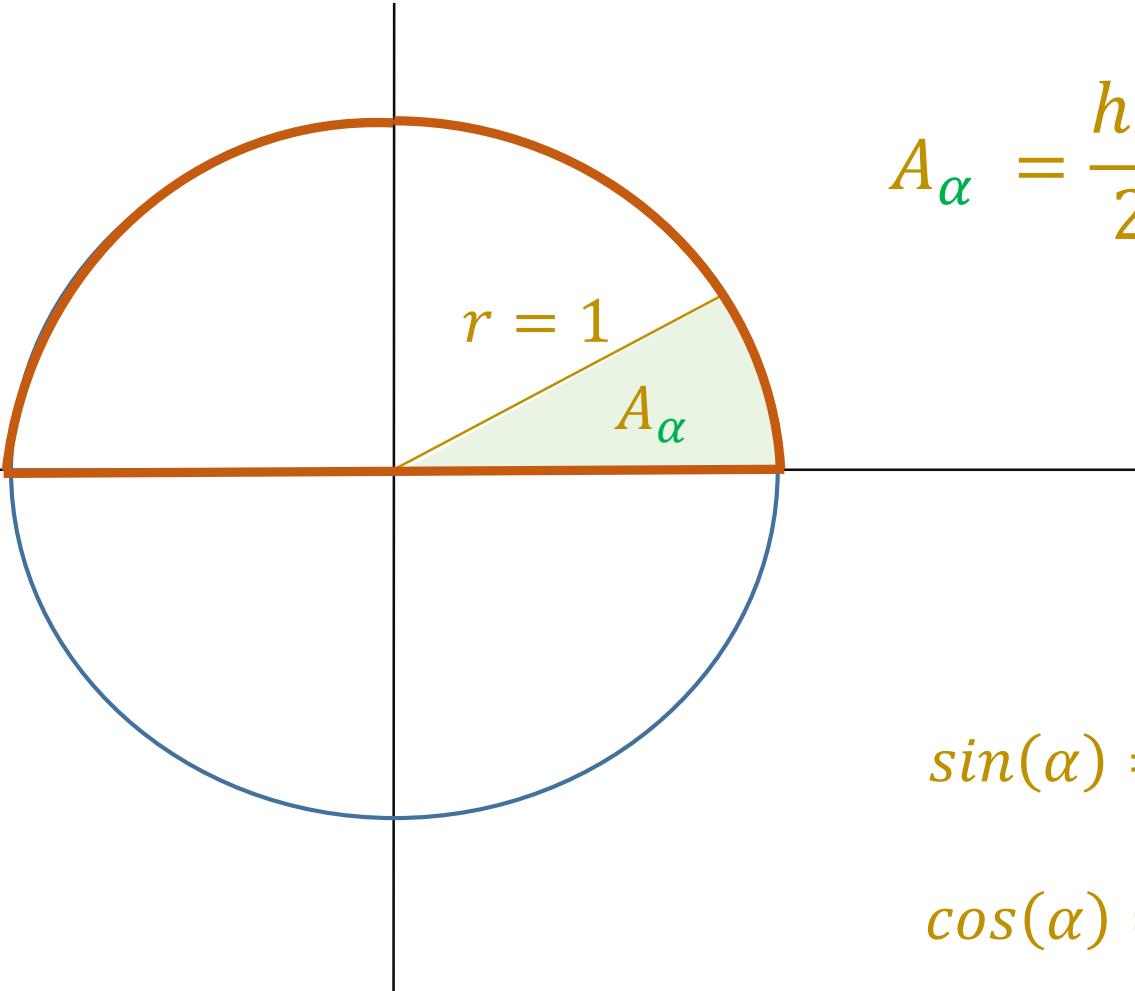
```
1 import math  
2  
3 def compute_length(degree):  
4     sin = math.sin(math.radians(degree))  
5     length = sin*(180/degree)  
6  
7     return length
```



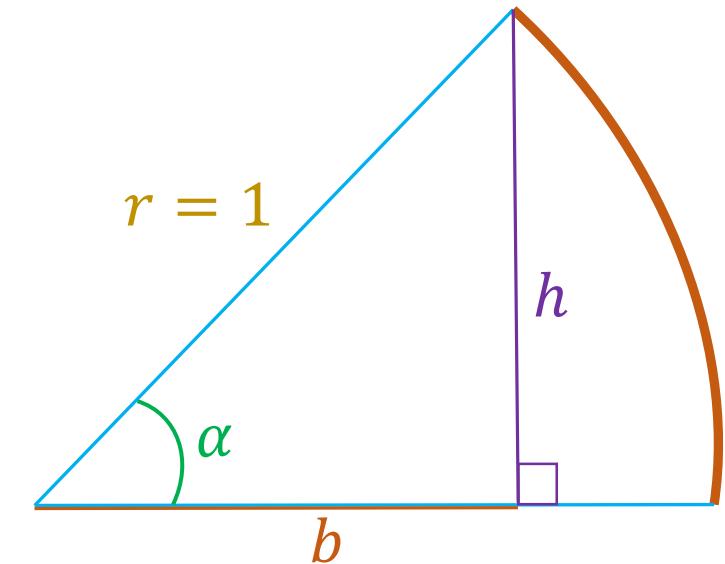
Divide the length into m segments  
 $L = m * \sin(\alpha)$

# Example

❖ Compute the area of a unit circle



$$A_\alpha = \frac{hb}{2}$$



$$\sin(\alpha) = \frac{h}{r}$$

$$\cos(\alpha) = \frac{b}{r}$$

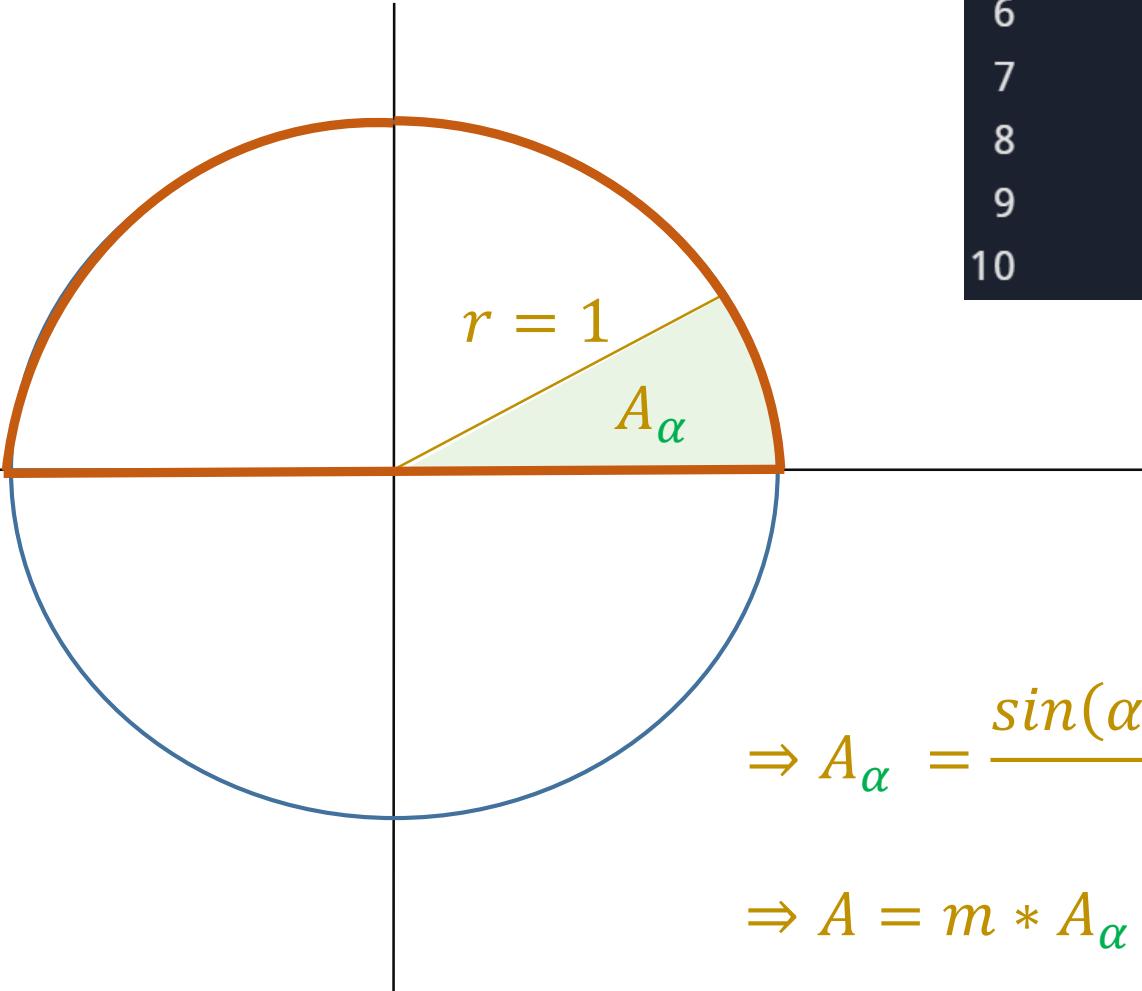
$$\Rightarrow A_\alpha = \frac{\sin(\alpha)\cos(\alpha)}{2}$$

$$\Rightarrow A = m * A_\alpha$$

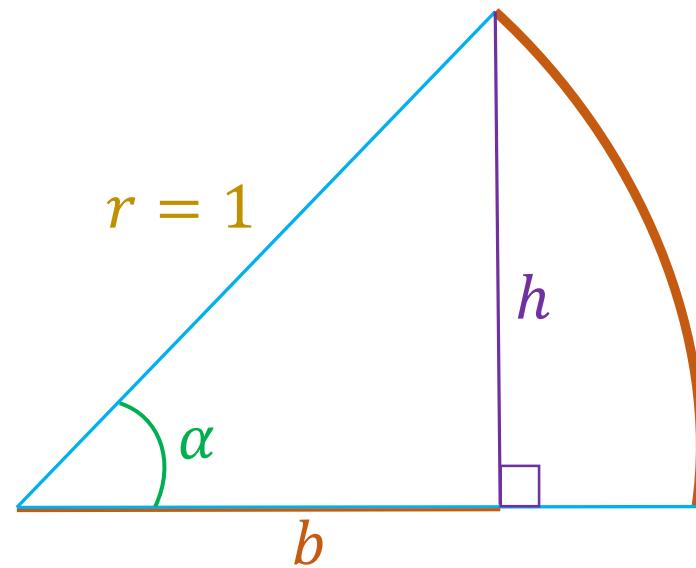
where  $m$  is a number of parts

# Example

## ❖ Compute the area of a unit circle

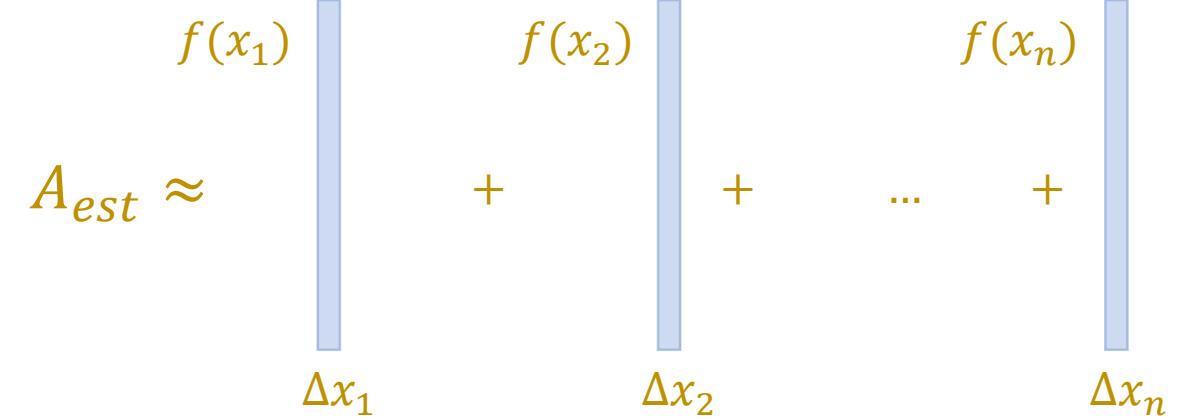
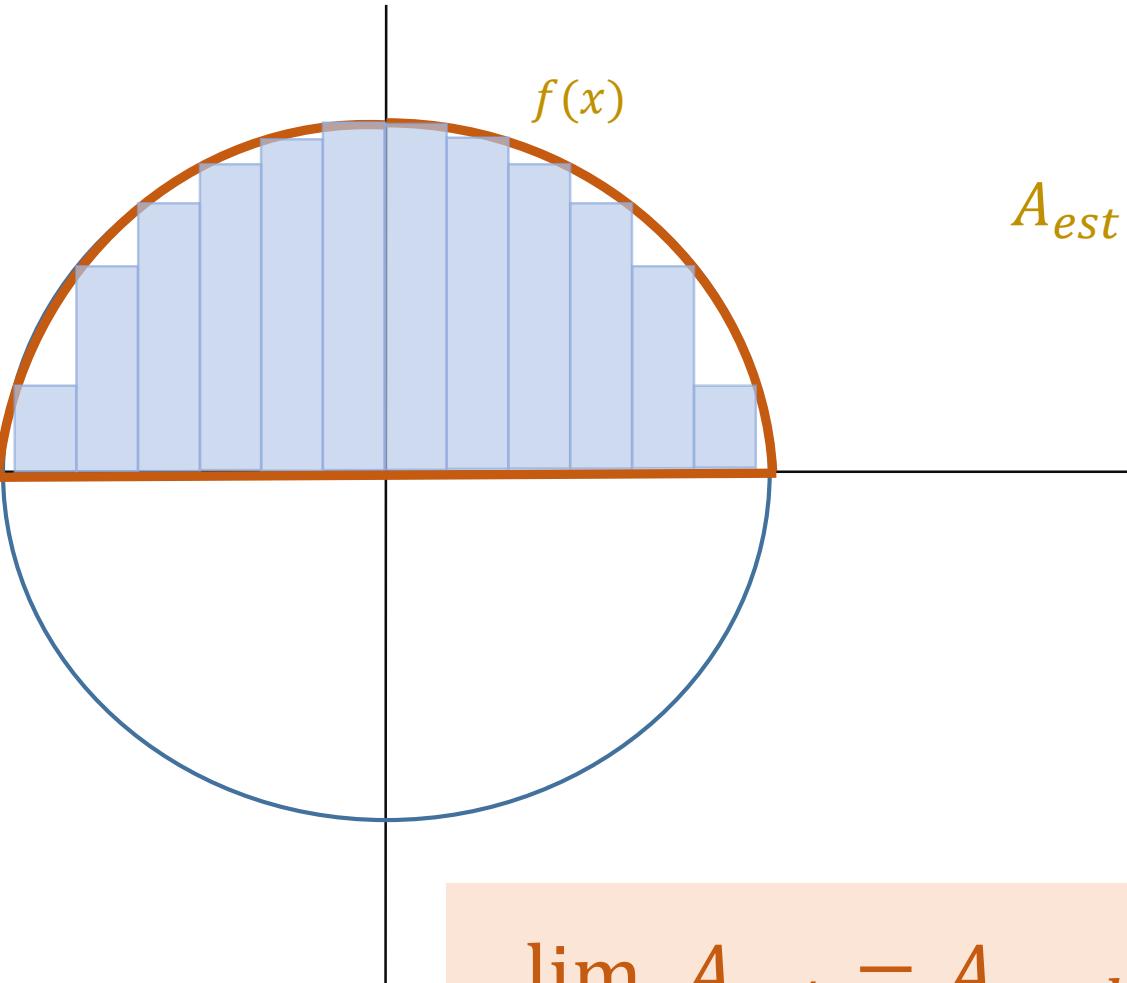


```
1 import math
2
3 def compute_area(degree):
4     sin_degree = math.sin(math.radians(degree))
5     cos_degree = math.cos(math.radians(degree))
6
7     area_degree = sin_degree*cos_degree/2
8     area = area_degree*(360/degree)
9
10    return area
```



# Example 2

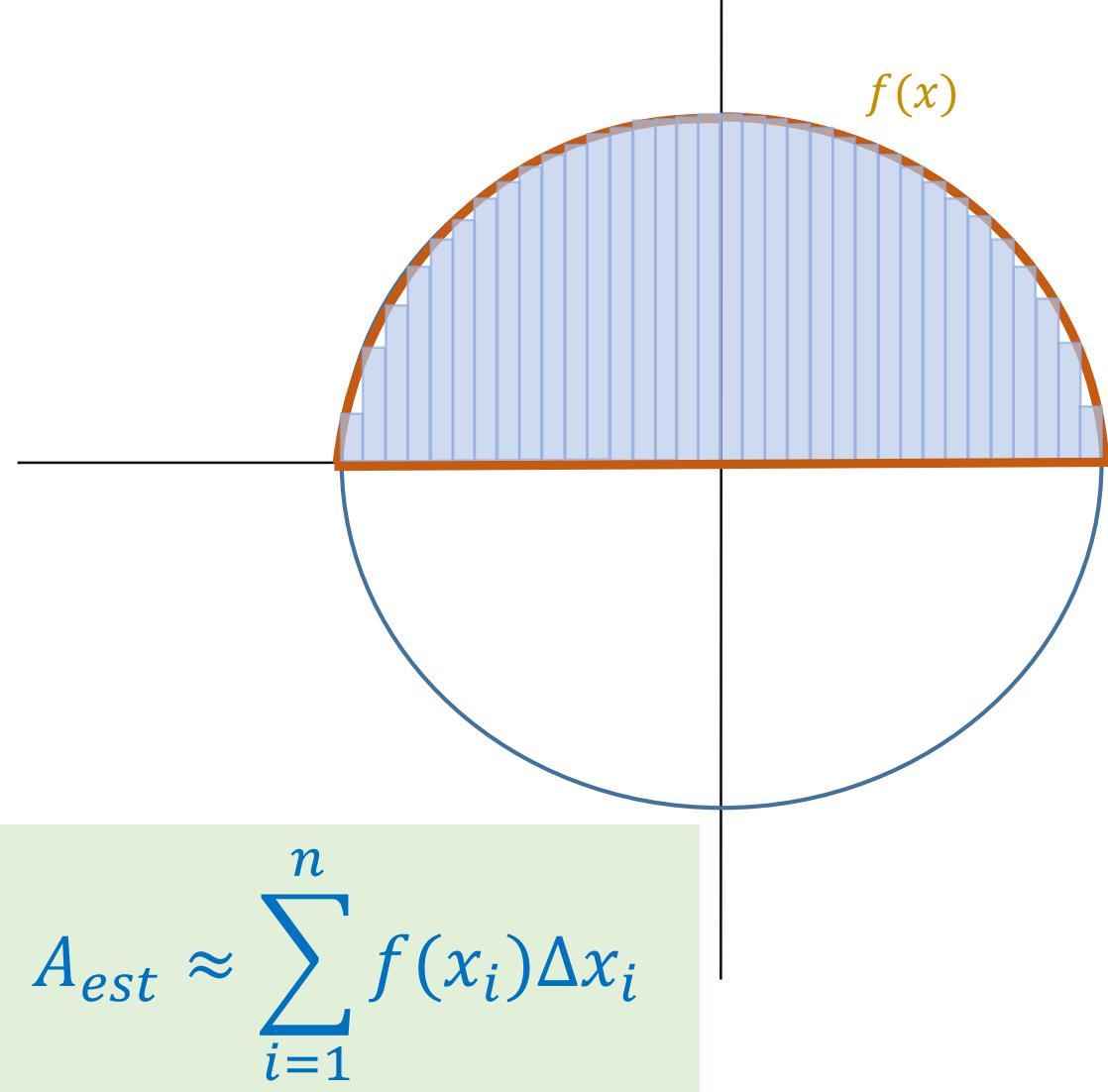
❖ Compute the area of a unit circle



$$A_{est} \approx \sum_{i=1}^n f(x_i) \Delta x_i$$

# Example 2

## ❖ Compute the area of a unit circle



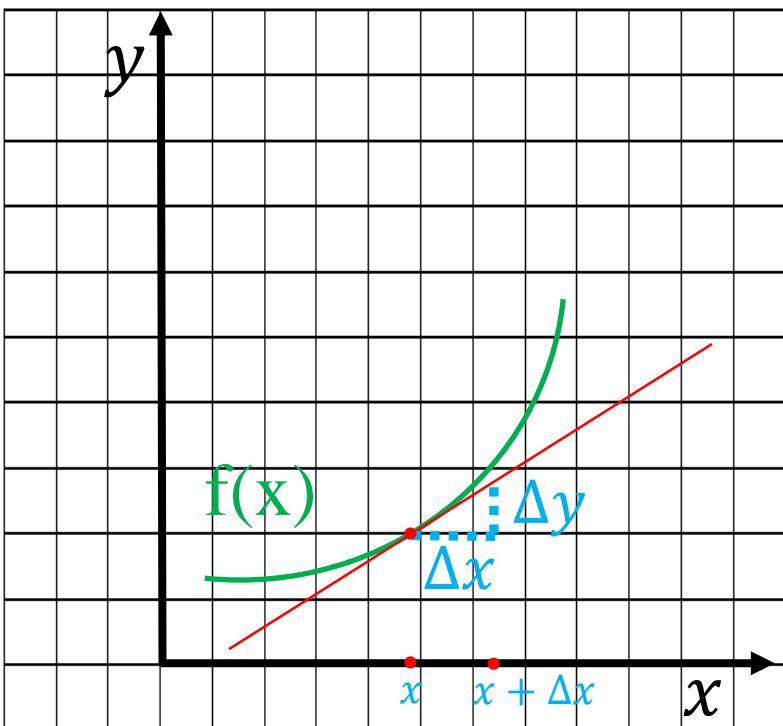
```

1 import numpy as np
2
3 def get_y(x):
4     y = np.sqrt(1 - x*x)
5     return y
6
7 # set delta_x
8 delta_x = 0.0001
9
10 # get the values of x
11 segments = np.arange(-1, 1+delta_x, delta_x)
12
13 # compute rectangle widths
14 begin = segments[0:-1]
15 end = segments[1:]
16 width = end - begin
17
18 # compute heights
19 y = get_y(begin)
20
21 # compute area
22 area = np.sum(width*y)
23 print(area**2) #--> 3.141591

```

# Example 3

## Đạo hàm cho hàm liên tục



$$\frac{d}{dx} f(x), \frac{dy}{dx}, y', f'(x)$$

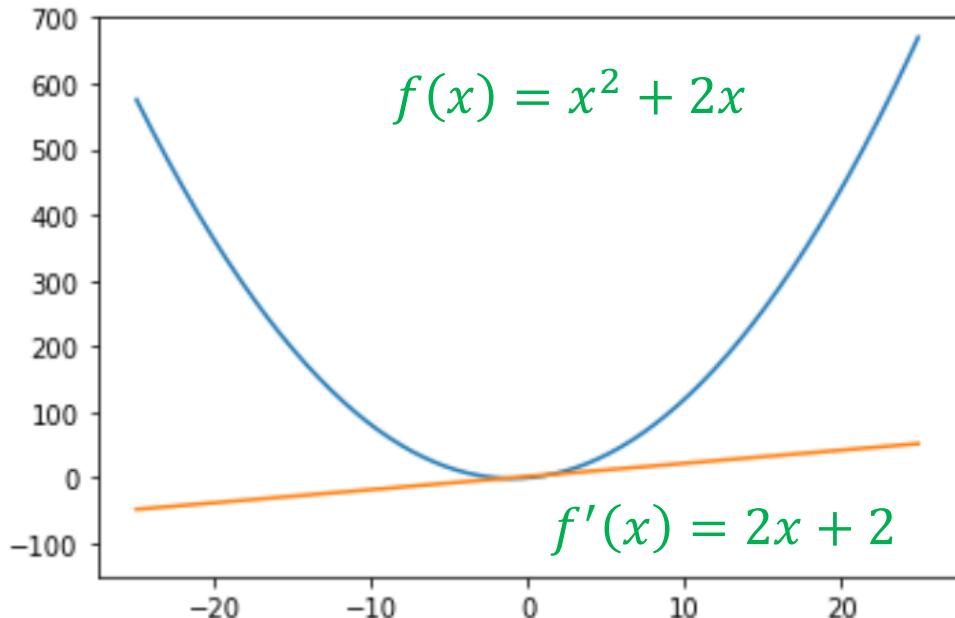
Đạo hàm =  $\frac{\text{Thay đổi theo } y}{\text{Thay đổi theo } x} = \frac{\Delta y}{\Delta x}$

$$\frac{d}{dx} f(x) = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

Δx cần tiến về 0 để  
đường tiếp tuyến tiến  
về hàm f(x) trong vùng  
lân cận tại x

# Example 3

## ❖ Implementation



```

1 # python code
2
3 def func(x):
4     return x**2 + 2*x
5
6 def func_derivative(x):
7     return 2*x + 2
8
9 d_value = func_derivative(2.0)
10 print('f'(x=2) is', d_value)

```

$f'(x=2)$  is 6.0

$$\begin{aligned}
 \frac{d}{dx} f(x) &= \lim_{\Delta x \rightarrow 0} \frac{(x + \Delta x/2)^2 + 2(x + \Delta x/2) - ((x - \Delta x/2)^2 + 2(x - \Delta x/2))}{\Delta x} \\
 &= \lim_{\Delta x \rightarrow 0} \frac{2x\Delta x + 2\Delta x}{\Delta x} \\
 &= \lim_{\Delta x \rightarrow 0} 2x + 2 = 2x + 2
 \end{aligned}$$

# Example 3

Cho hàm số  $f(x)$

$$f(x) = x^2 + 2x$$

Công thức đạo hàm

$$f'(x) = 2x + 2$$

Công thức đạo hàm một bên

$$f'(x) = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

```
1 # python code:  
2  
3 def func(x):  
4     return x**2 + 2*x  
5  
6 def func_derivative(x):  
7     return 2*x + 2
```

```
1 print(f'f'(x=2) is {func_derivative(2.0)})  
f'(x=2) is 6.0
```

Theo lý thuyết đạo hàm, epsilon càng nhỏ thì giá trị đạo hàm tại một điểm càng chính xác!

# Example 3

## ❖ Implementation

Cho hàm số  $f(x)$

$$f(x) = x^2 + 2x$$

Công thức đạo hàm

$$f'(x) = 2x + 2$$

Công thức đạo hàm một bên

$$f'(x) = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

```
1 # đạo hàm một bên
2 def gradient(f, x, epsilon):
3     return (f(x + epsilon) - f(x)) / epsilon
4
5 def func(x):
6     return x**2 + 2*x
7
8 print(f'(e=1.0e2 and x=2) : {gradient(f=func, x=2.0, epsilon=1.0e2)}')
9 print(f'(e=1.0e1 and x=2) : {gradient(f=func, x=2.0, epsilon=1.0e1)}')
10 print(f'(e=1.0e0 and x=2) : {gradient(f=func, x=2.0, epsilon=1.0e0)}')
11 print(f'(e=1.0e-1 and x=2) : {gradient(f=func, x=2.0, epsilon=1.0e-1)}')
12 print(f'(e=1.0e-2 and x=2) : {gradient(f=func, x=2.0, epsilon=1.0e-2)}')
13 print(f'(e=1.0e-3 and x=2) : {gradient(f=func, x=2.0, epsilon=1.0e-3)}')
14 print(f'(e=1.0e-4 and x=2) : {gradient(f=func, x=2.0, epsilon=1.0e-4)}')

(e=1.0e2 and x=2) : 106.0
(e=1.0e1 and x=2) : 16.0
(e=1.0e0 and x=2) : 7.0
(e=1.0e-1 and x=2) : 6.099999999999994
(e=1.0e-2 and x=2) : 6.00999999999849
(e=1.0e-3 and x=2) : 6.000999999999479
(e=1.0e-4 and x=2) : 6.000100000012054
```

# Outline

- Programming Languages
- Python Environment
- Basic Data Types
- Functions
- Conditions
- Case Studies

# 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

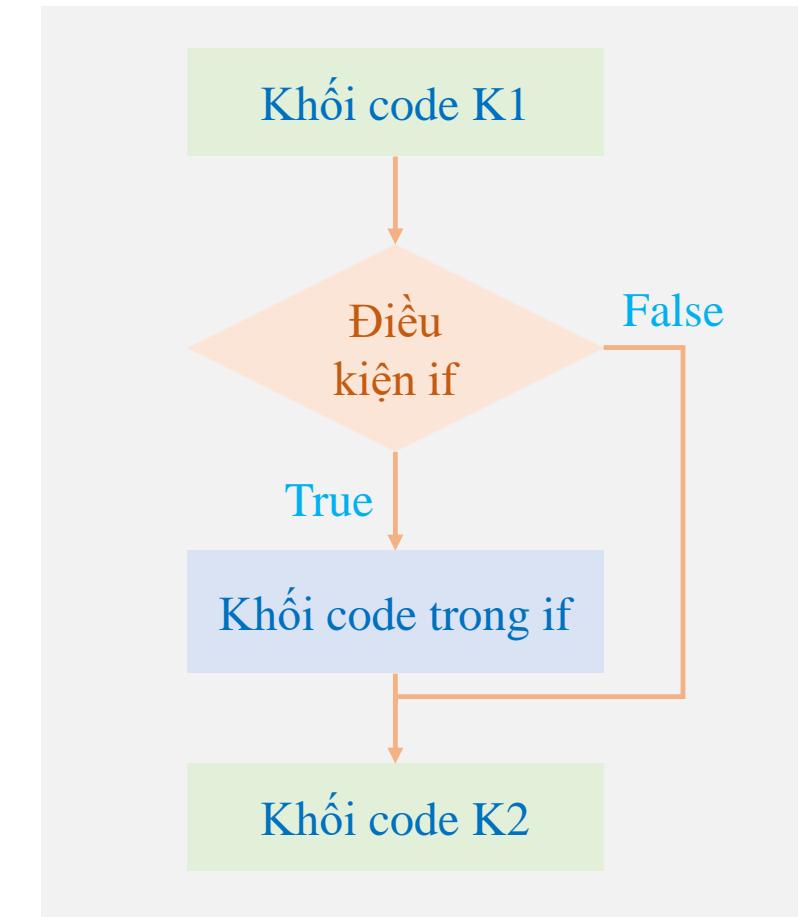
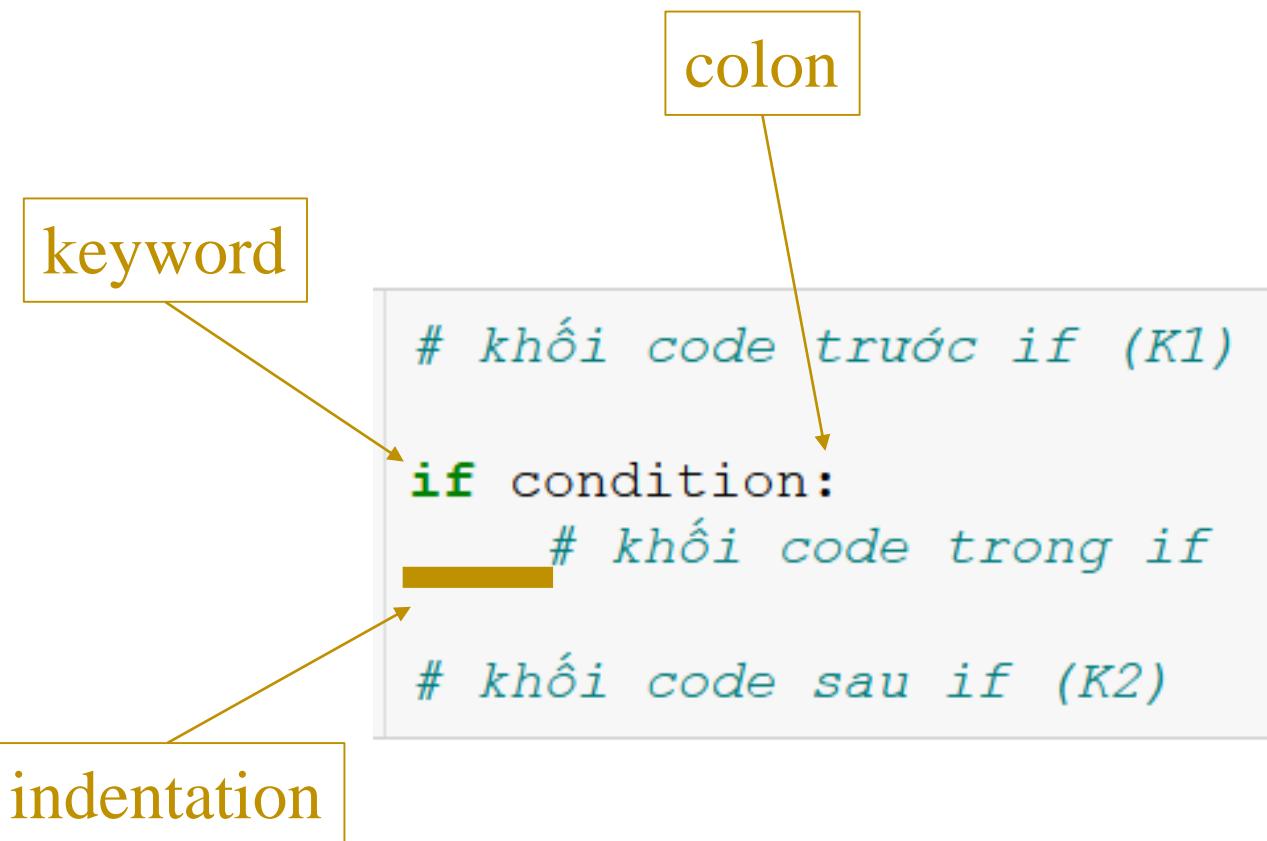
False

True

True

# Condition

## ❖ if condition



# Condition

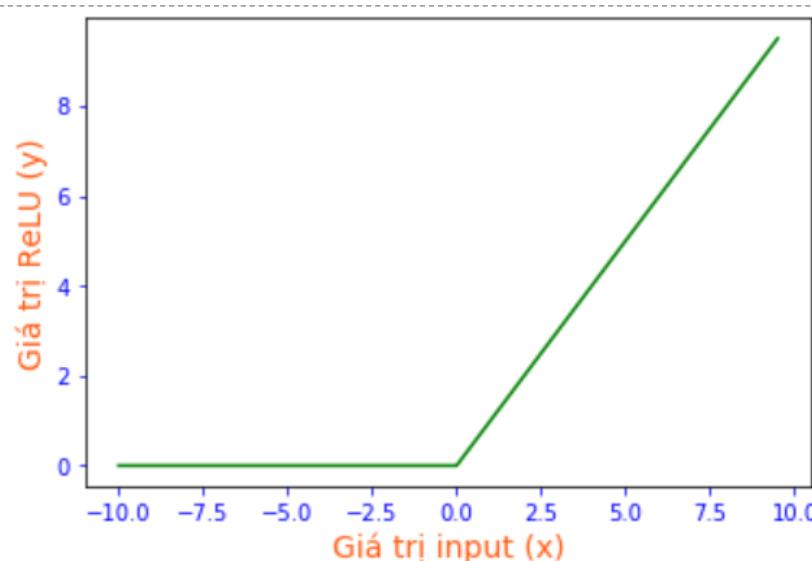
if  
condition

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

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

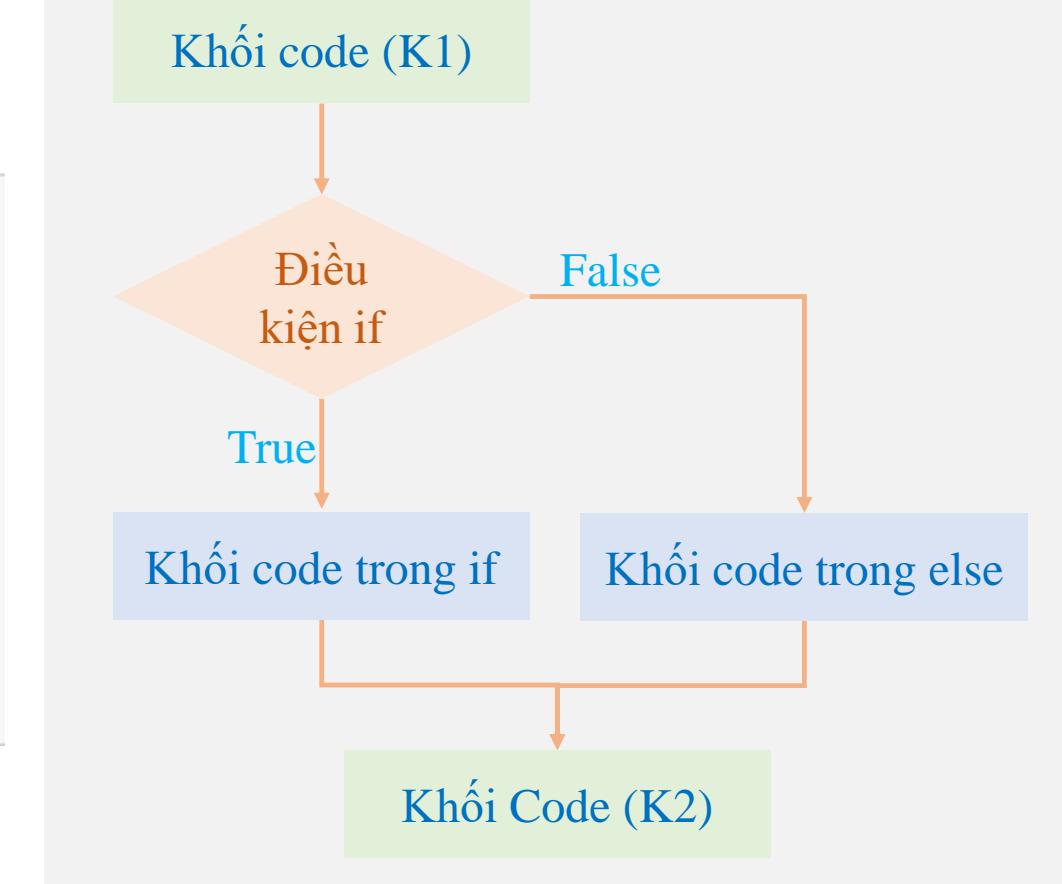
## ❖ if-else condition

```
# 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

indentation

colon



# Condition

## ❖ if-else condition

Input:  $a$  and  $b$

if  $a > 0$  then

$$y = b^2$$

if  $a \leq 0$  then

$$y = \sqrt{b}$$

```
1 import math
2
3 def a_function(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(a_function(2, 4))
14 print(a_function(-2, 4))
```

16  
2.0  
>

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

16  
2.0  
>

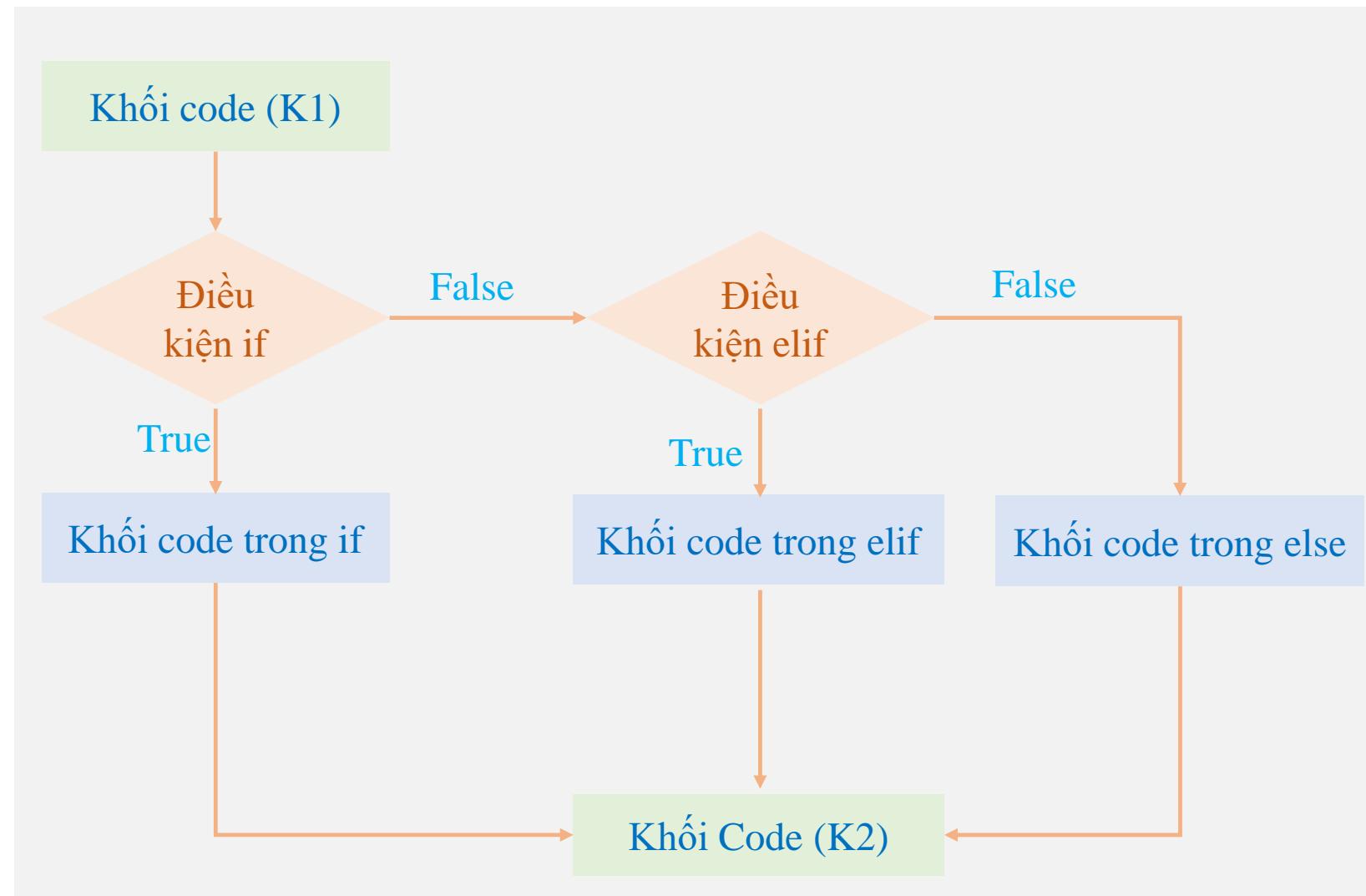
# Condition

## ❖ 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

Input:  $a$  and  $b$

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

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

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

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

16  
2.0  
4

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

16  
2.0  
4

# Outline

- Programming Languages
- Python Environment
- Basic Data Types
- Functions
- Conditions
- Case Studies

# 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

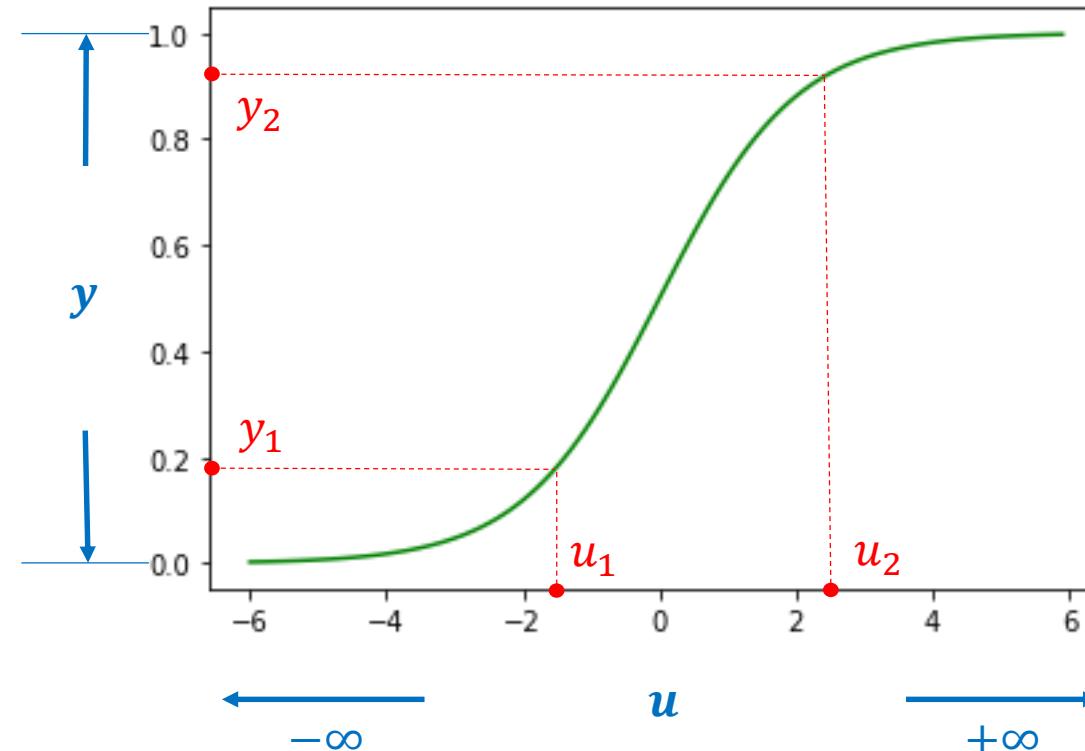
$$y = \sigma(u) = \frac{1}{1 + e^{-u}}$$

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

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

## Property

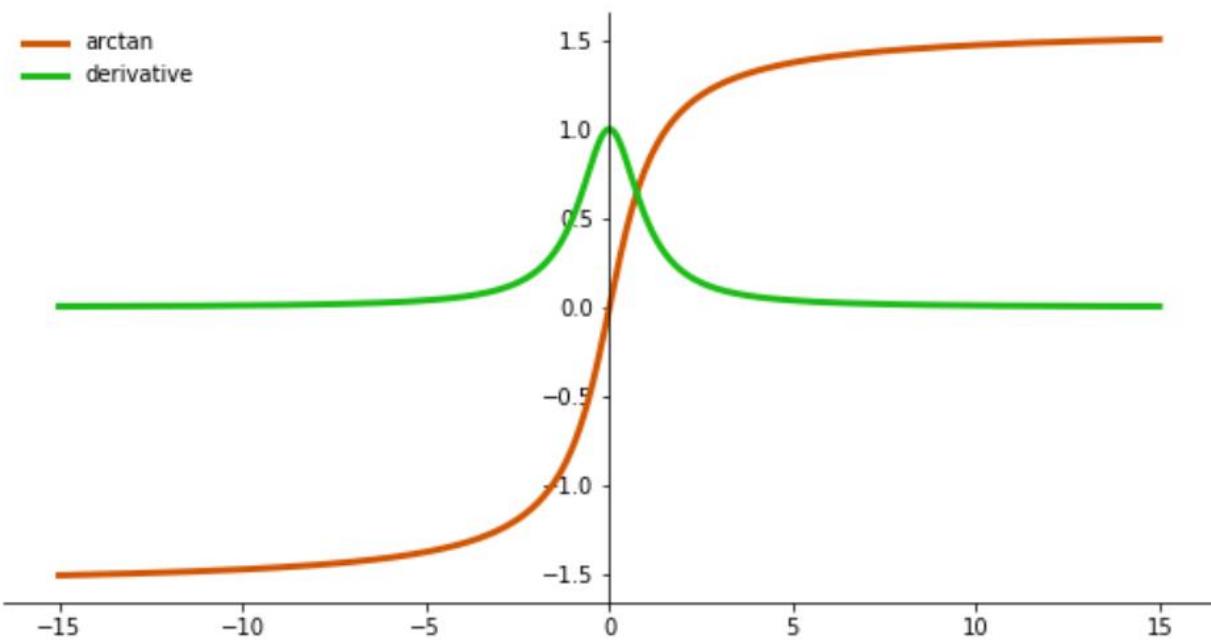
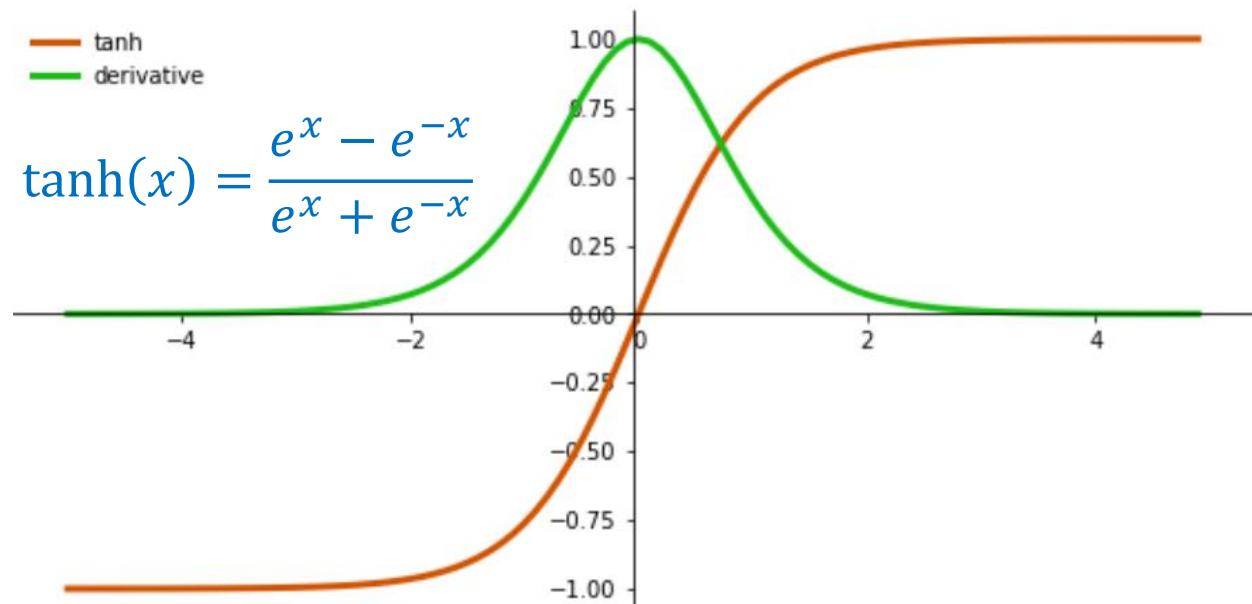
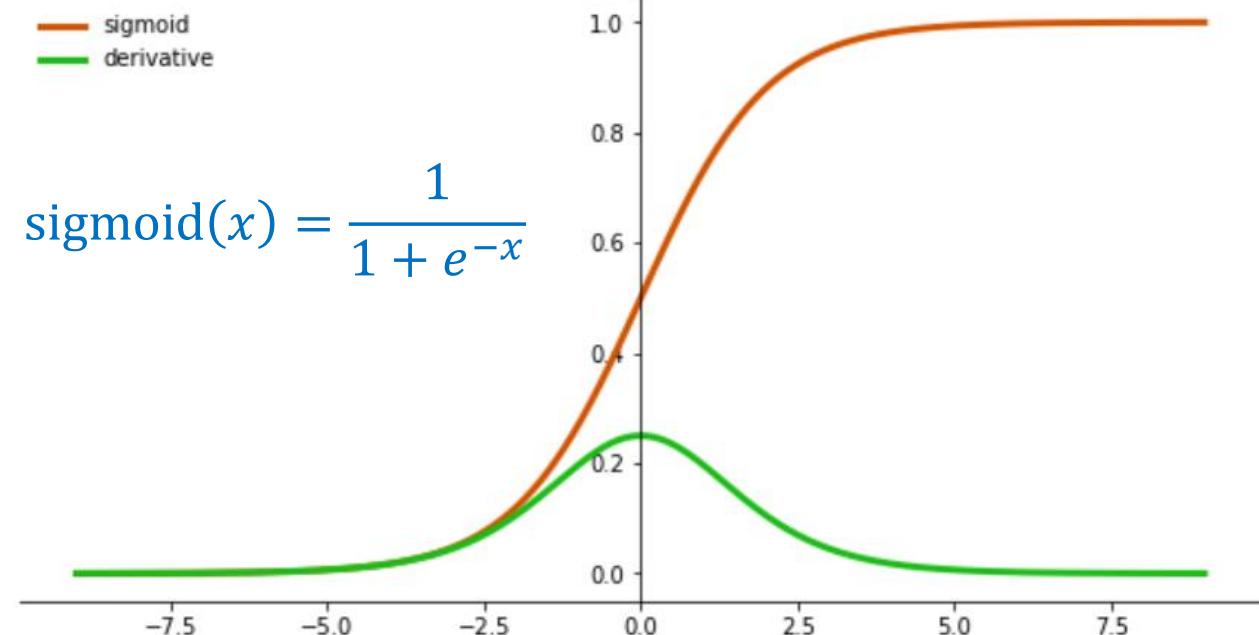
$$\forall u_1, u_2 \in [a, b] \text{ và } u_1 \leq u_2 \rightarrow \sigma(u_1) \leq \sigma(u_2)$$



# Activation Functions

S-shaped curves

Map x values into small ranges



# 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

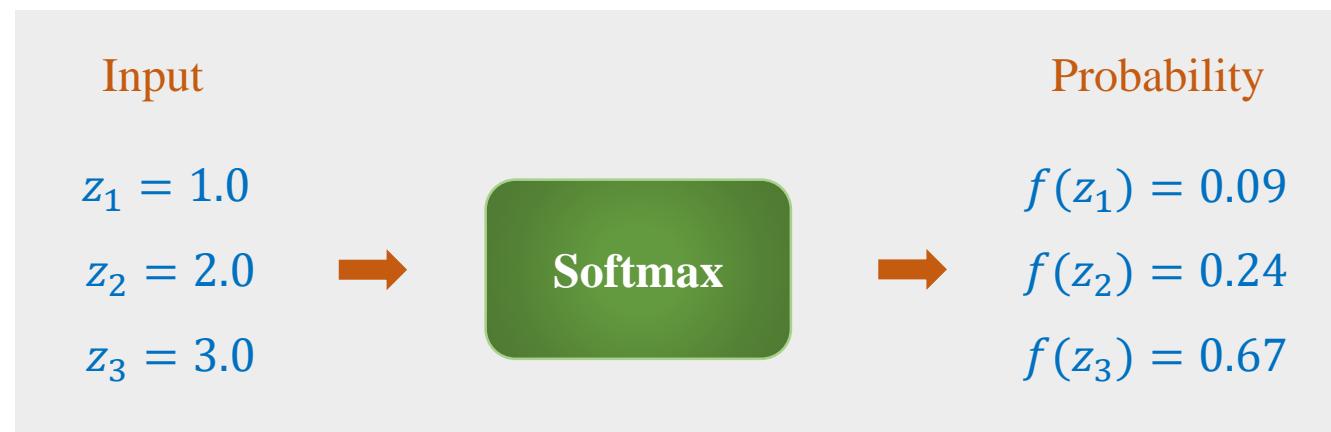
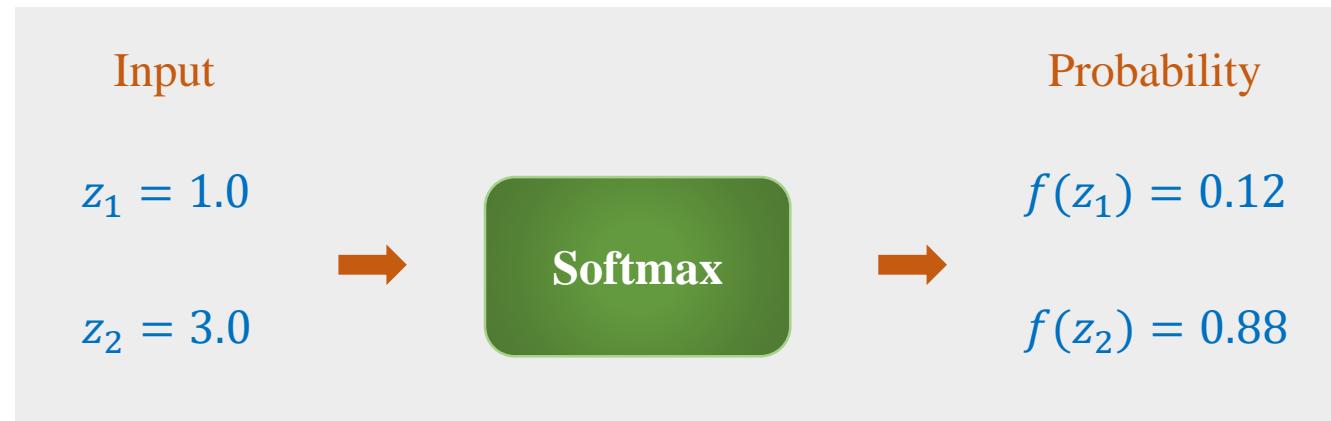
# Overflow and Underflow

## 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

```

```

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  
biên

```

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

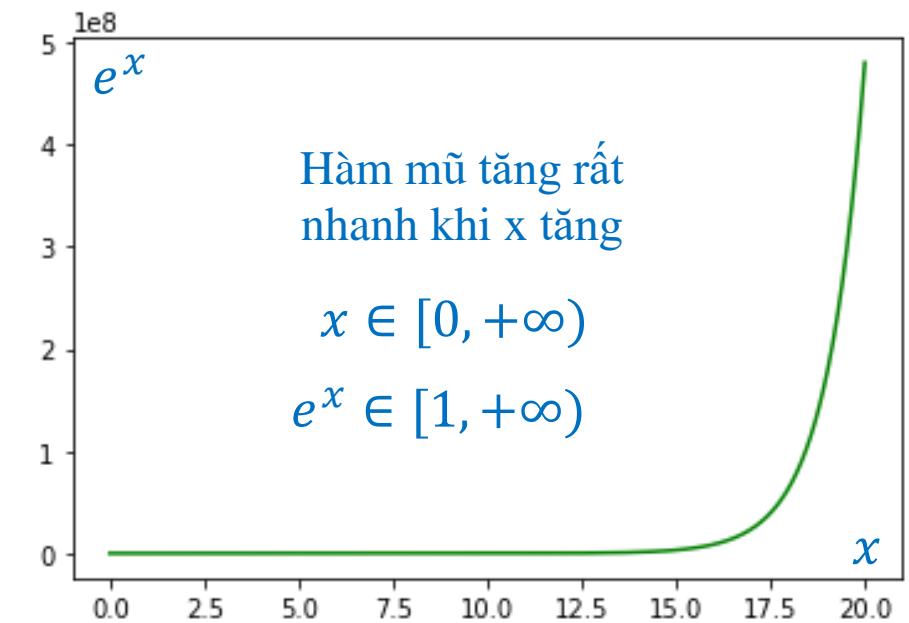
```

```

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

$$x_1 = -2.0$$

$$x_2 = -1.0$$

$$x_3 = 0$$

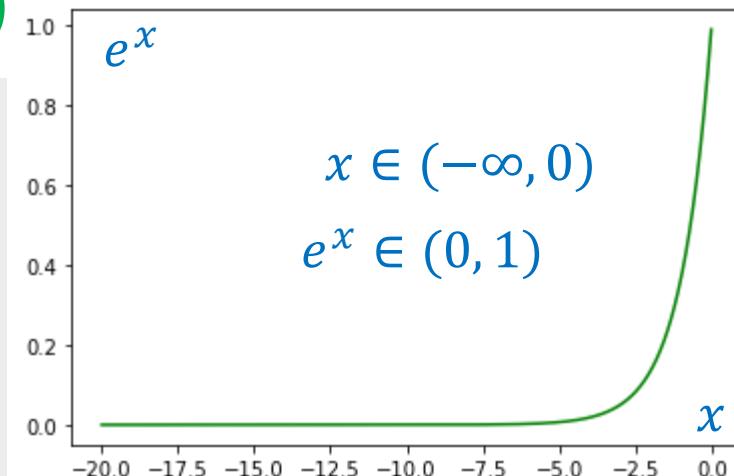
Softmax

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

# Exponential Functions

## ❖ Example

Suppose a person invests  $P$  dollars in a savings account with an annual interest rate  $r$ , compounded annually.

The amount of money after 1 year is

$$A_1 = A + rA = A(1 + r)$$

The amount of money after 2 years is

$$A_2 = A_1 + rA_1 = A(1 + r) + rA(1 + r) = A(1 + r)^2$$

More generally, the amount after  $t$  years is

$$A_t = A(1 + r)^t$$

If the money is compounded 2 times per year, the amount of money after half a year is

$$A_{1/2} = A + \frac{r}{2}A = A\left(1 + \frac{r}{2}\right)$$

The amount of money after 1 year is

$$A_1 = A\left(1 + \frac{r}{2}\right) + \frac{r}{2}A\left(1 + \frac{r}{2}\right) = A\left(1 + \frac{r}{2}\right)^2$$

After  $t$  years, the amount of money in the account is

$$A_t = A\left(1 + \frac{r}{2}\right)^{2t}$$

# Exponential Functions

## ❖ Example

This example is from the reference book

After  $t$  years, the amount of money in the account is

$$A_t = A \left(1 + \frac{r}{2}\right)^{2t}$$

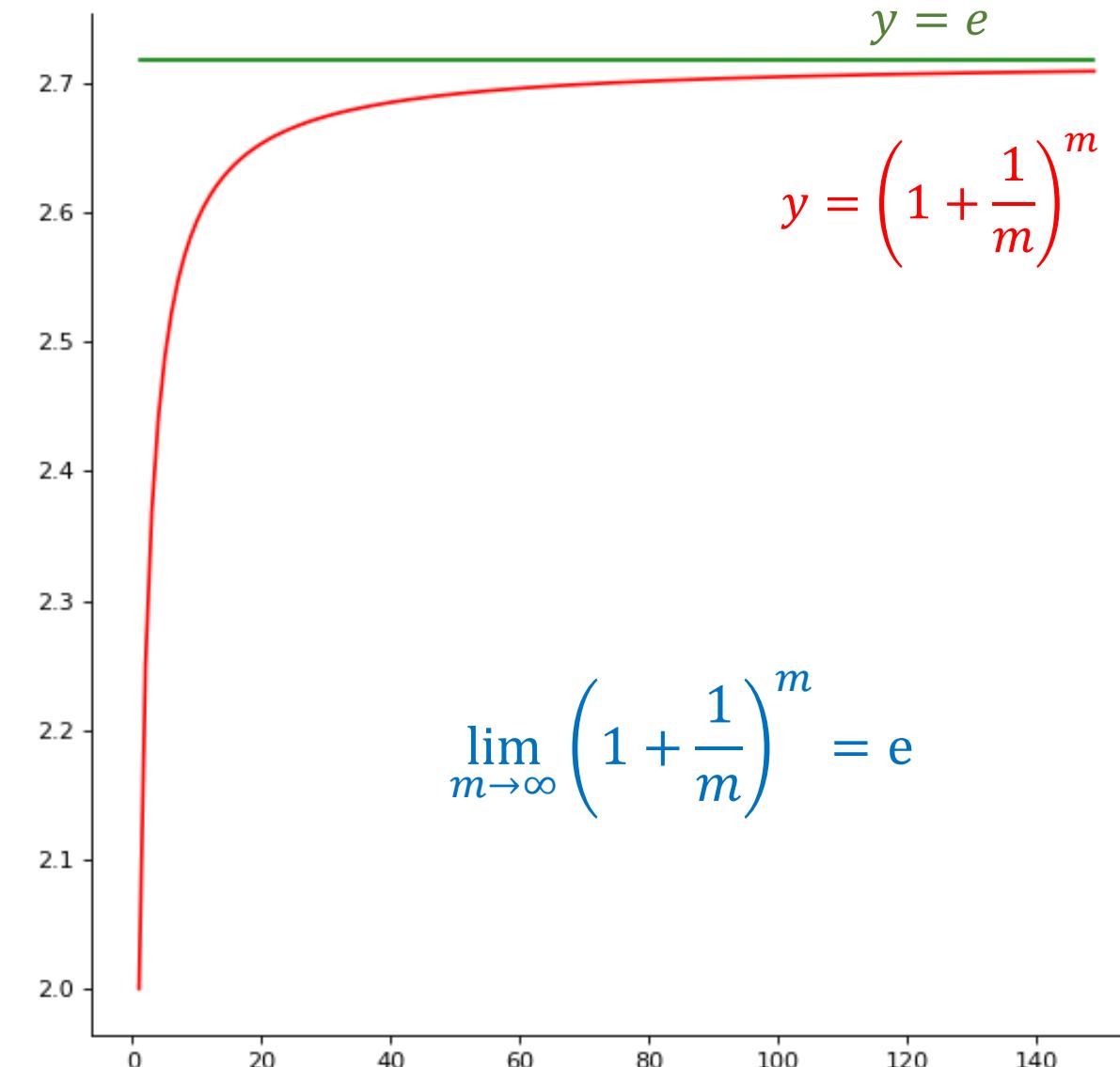
More generally, if the money is compounded  $n$  times per year, the amount of money in the account after  $t$  years is given by the function

$$A_t = A \left(1 + \frac{r}{n}\right)^{nt}$$

$$\text{set } m = \frac{n}{r}$$

$$A_t = A \left(1 + \frac{1}{m}\right)^{mrt}$$

$$A_t = Ae^{rt}$$

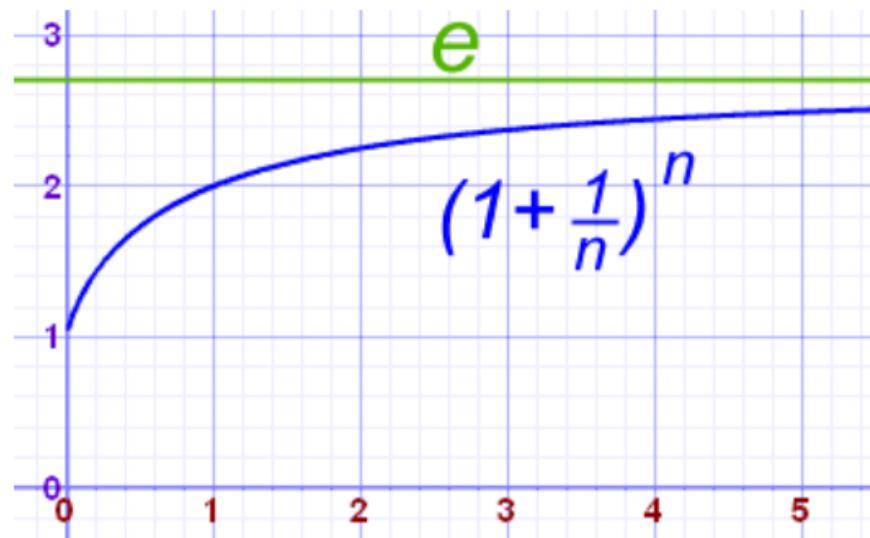


# Example

## ❖ 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

# Cheat Sheet

## Run a Python program

- \_ Python files have “.py” at the end of the filename
- \_ To run a Python file:  
`python file.py`

## Virtual Environment

- \_ Install Python Virtual Environment (in Linux):  
`sudo apt-get install -y python3-venv`
- \_ Create a virtual environment:  
`python3 -m venv a_name`
- \_ Activate a virtual environment:  
`source a_name/bin/activate`

## Install new package

- \_ e.g: to install *matplotlib*:  
`pip install matplotlib`

## Variable

- \_ Declare a variable:  
`variable_name = variable_value`

## Variable types:

Integer	1, 2, 0, -1, -2
Float	1.5, 0.5, -3.21
String	‘AI’, “VIETNAM”
Boolean	True, False

## Basic Operators

Operators	Meaning
+	Summation
-	Subtraction
*	Multiplication
/	Division
%	Modulo
//	Floor Division
**	Power

## Function

### Define a function:

```
def function_name(paramerters):
    """
    docstring
    """
    # your code goes here

    return result
```

### Default values:

```
def function_name(p1=0, p2=0):
    # your code goes here
```

## Condition

### Comparision Operators:

==	Equal
!=	Not equal
>	Greater than
<	Less than
>=	Greater or equal than
<=	Less or equal than

### Conditional sentence:

```
if condition1:
    # your code
elif condition2:
    # your code
else:
    # your code
```

## Built-in Functions

print(params)	Print value onto the screen
type(params)	Return class type of variable
input(prompt)	Ask user to input a string
int(), float()	Type conversion

## Overflow/Underflow

### Underflow:

```
result = 1e-100
print(result) # 1e-100
result = 1e-1000
print(result) # 0.0
```

### Overflow:

```
result = 1e100
print(result) # 1e+100
result = 1e1000
print(result) # inf
```

## For Loop

- \_ Create a loop using for:  
`for element in iterable:  
 # code inside your for`
- \_ Some iterables:

String	“aivietnam”
Tuple	(1, 2, 3)
List	[1, 2, 3]
Dictionary	{‘key1’: 1}
range()	range(0, 5, 1)

### Special keywords:

break	Exit the loop
continue	Move to next iteration

