



**AI VIET NAM**

@aivietnam.edu.vn

# Basic Python for AI

## Memorization Matters!

**Quang-Vinh Dinh**  
**PhD in Computer Science**

# Objectives

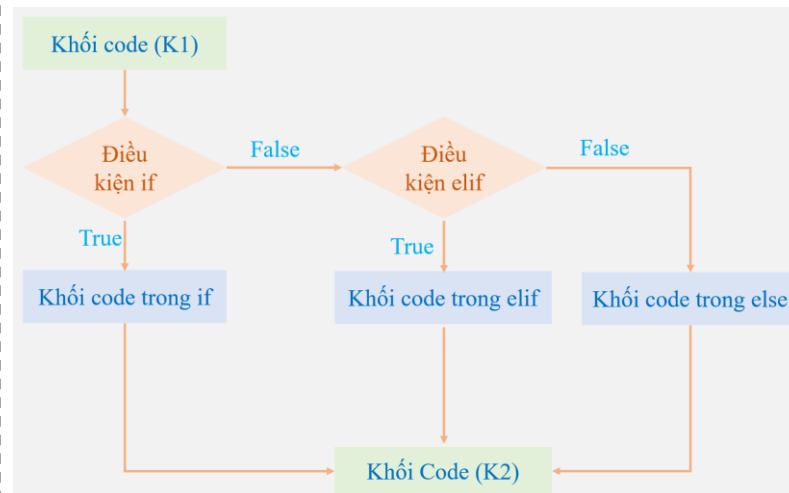
## Data Representation

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



## Conditions

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



## Functions

### Built-in Functions

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

### Function

\_ Define a function:

```
def function_name(parameters):  
    '''  
    docstring  
    '''  
    # your code goes here  
    return result
```

# Outline

## SECTION 1

### Introduction

## SECTION 2

### Data Representation

## SECTION 3

### Functions

## SECTION 4

### Conditions

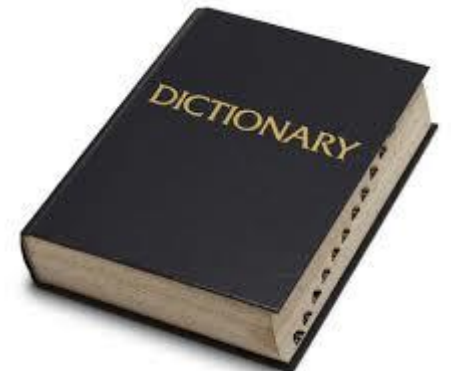
#### Variable

\_ Declare a variable:

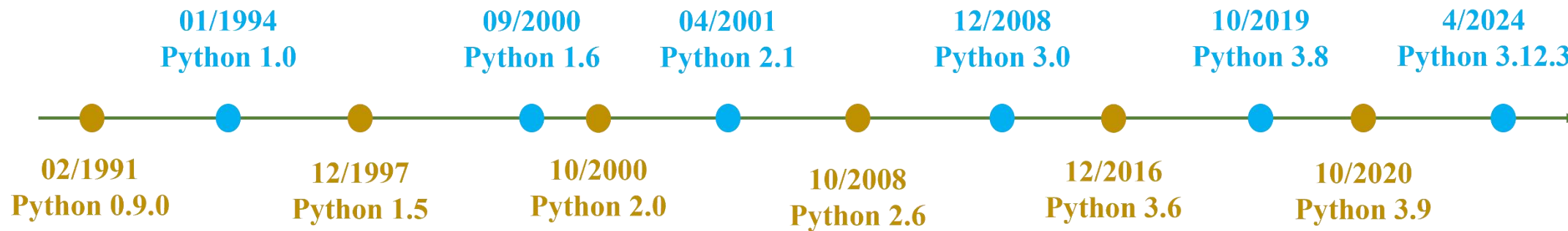
```
variable_name = variable_value
```

\_ Variable types:

Integer	1, 2, 0, -1, -2
Float	1.5, 0.5, -3.21
String	'AI', "VIETNAM"
Boolean	True, False



## ❖ Python history



Ý tưởng từ 1980s



Bắt đầu cài đặt  
từ 12/1989



Được đặt tên theo  
nhóm hài Monty Python



Guido van Rossum

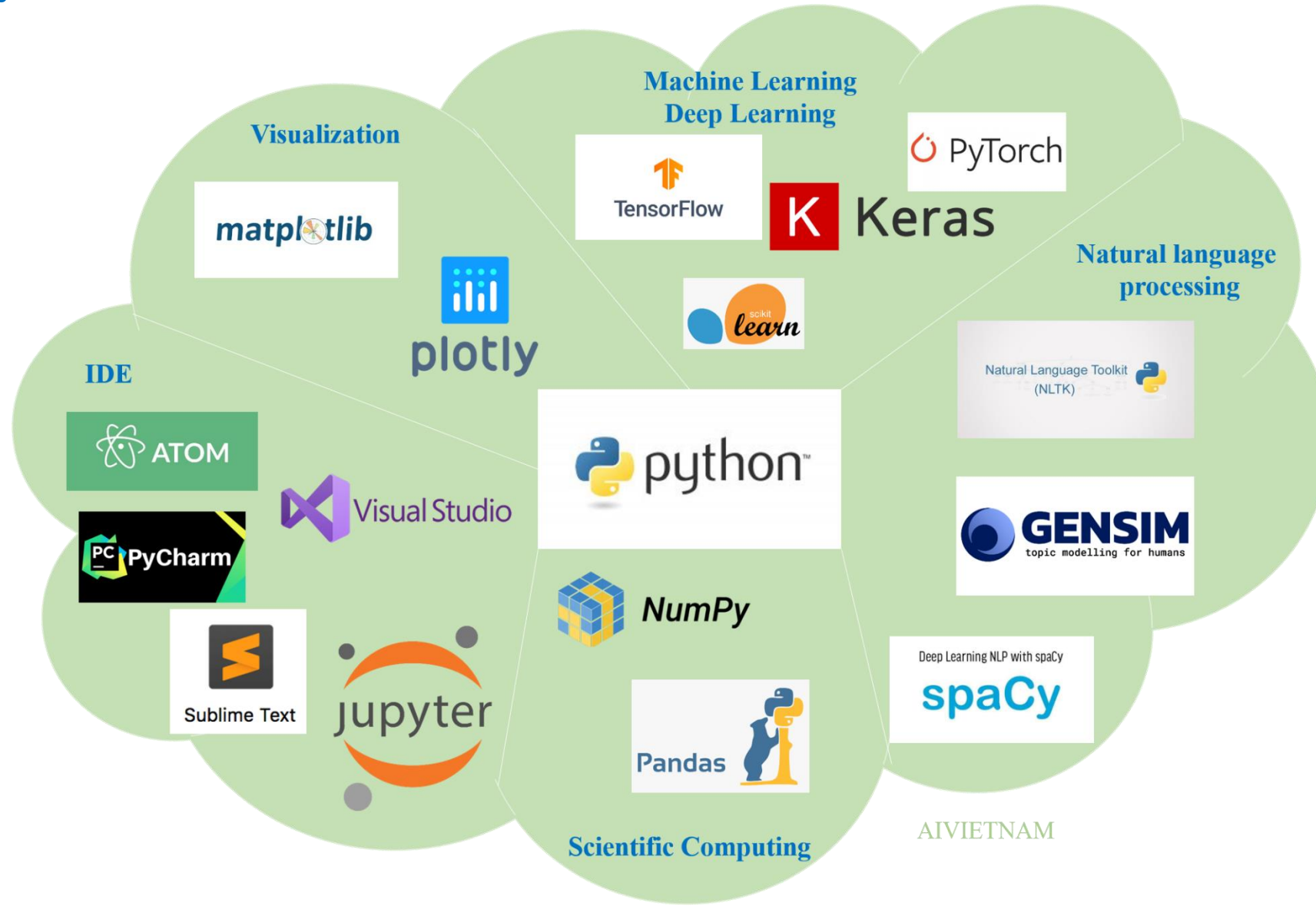


228,855 python packages  
(PyPI)



Hỗ trợ rất mạnh cho  
Data Science và Machine Learning

## ❖ Python ecosystem

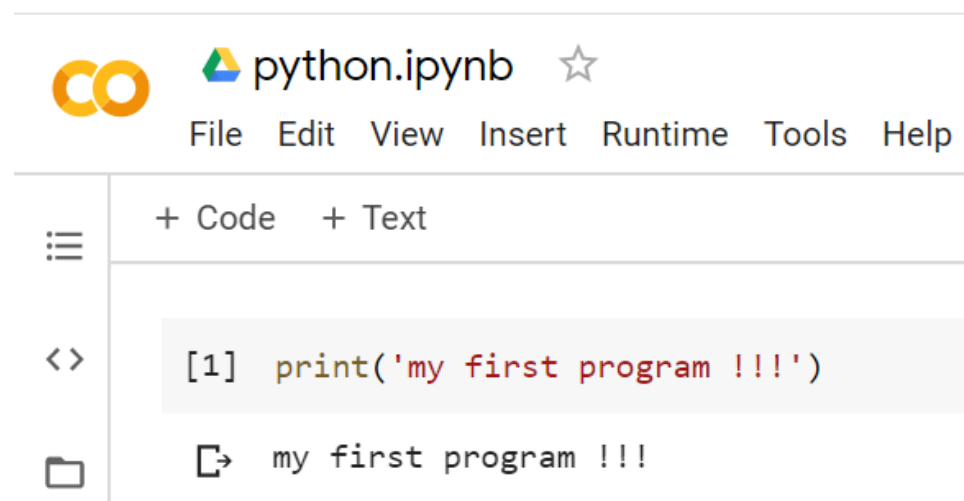




## ❖ First Python program

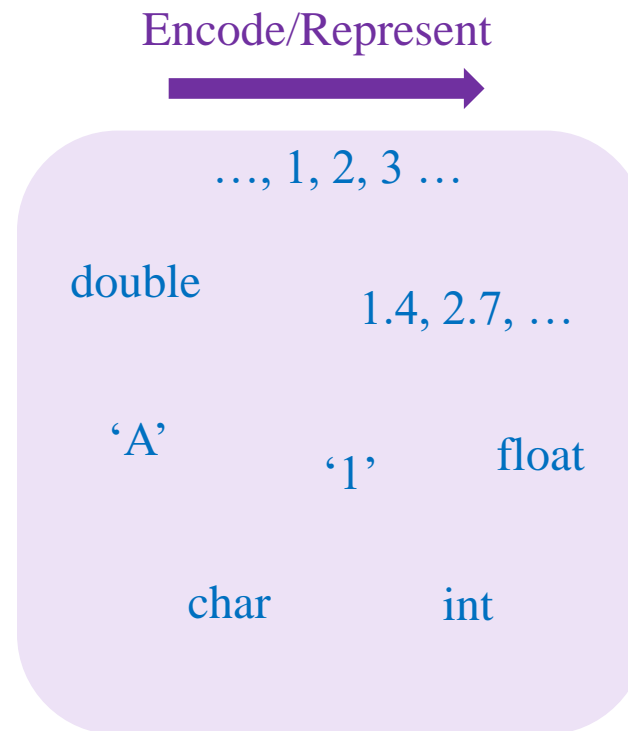
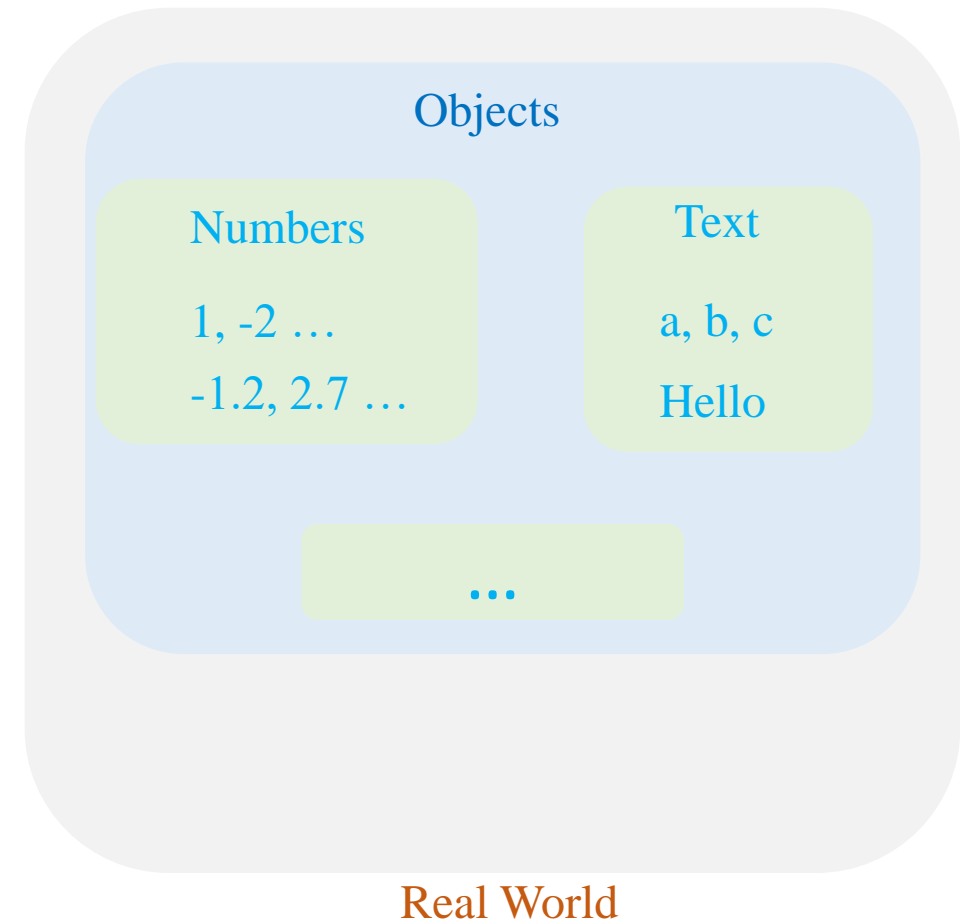
### ❖ Using Google Colab

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

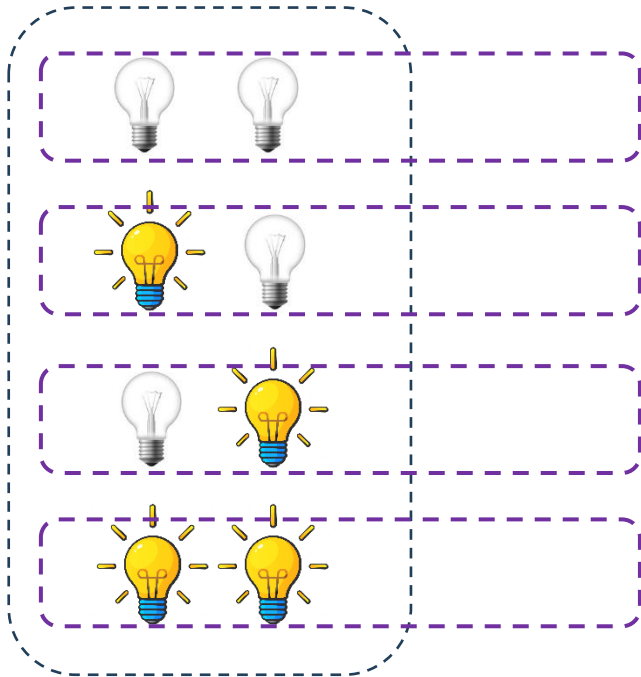


Demo

## ❖ What should a programming language provide?



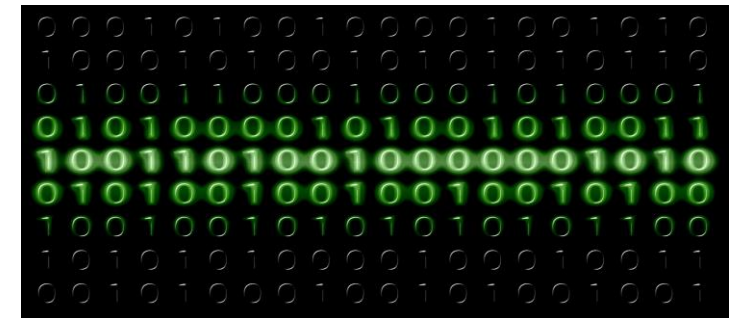
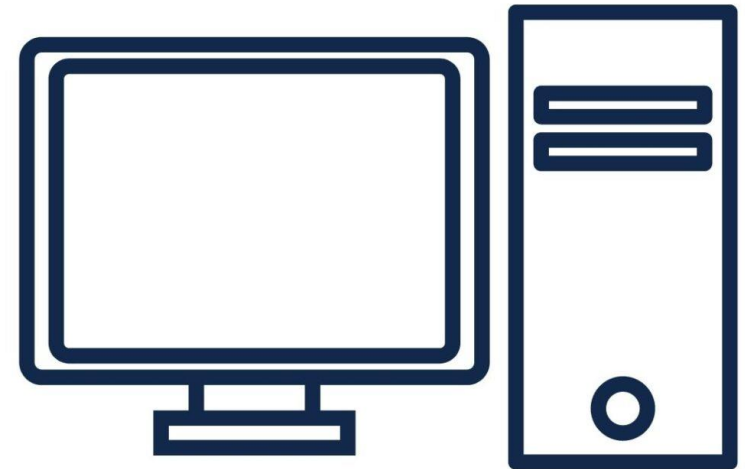
## ❖ What should a programming language provide?



0, 1, 2, 3

100, 101, 102, 103

a, b, c, d







## ❖ Logic, number and text

`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

## ❖ Logic, number and text

`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
```

## ❖ 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





## ❖ String creation

```
1 text1 = 'Tôi yêu AI VIET NAM'  
2 print(text1)
```

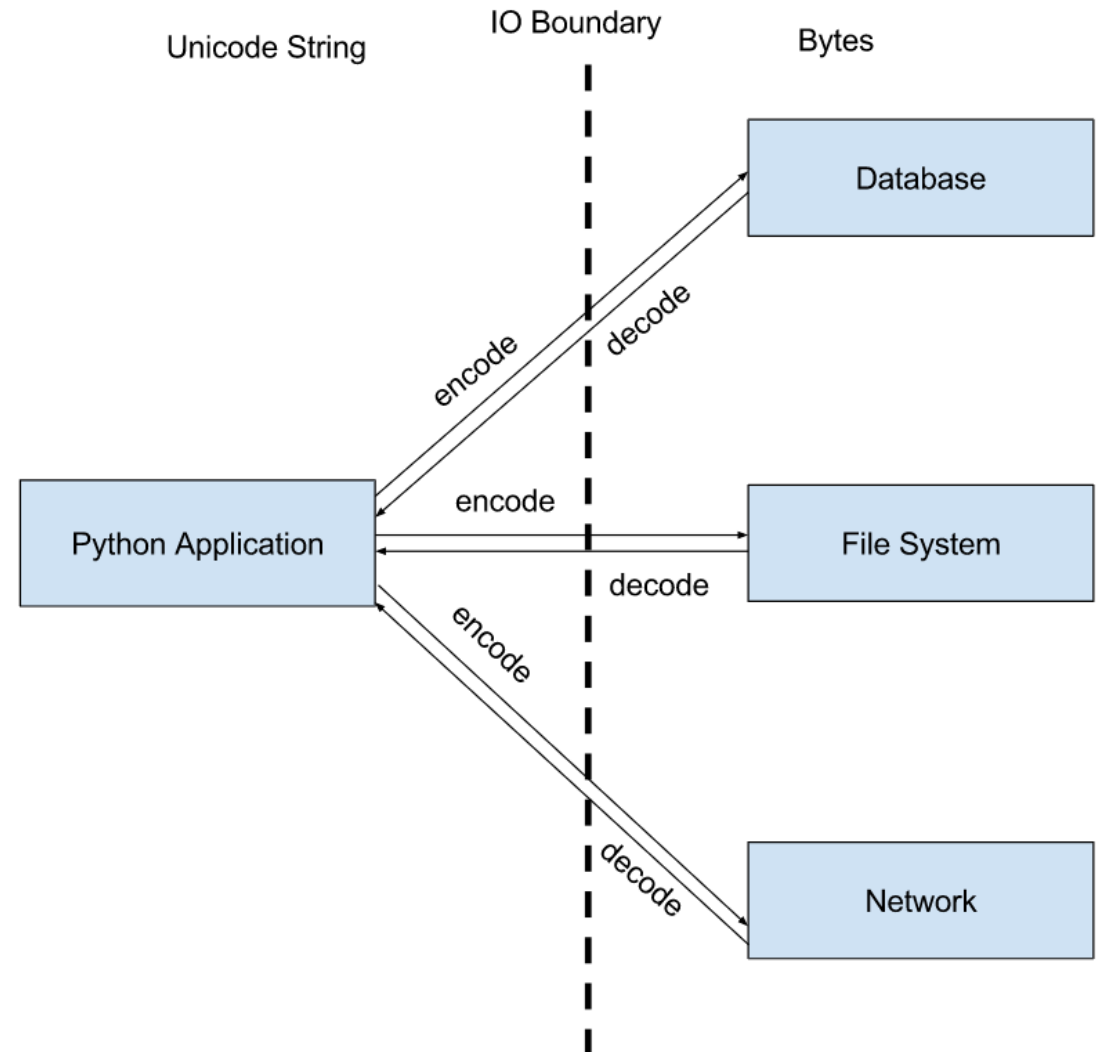
Tôi yêu AI VIET NAM

```
1 text2 = "Tôi yêu AI VIET NAM"  
2 print(text2)
```

Tôi yêu AI VIET NAM

```
1 text3 = '''Tôi yêu AI VIET NAM'''  
2 print(text3)
```

Tôi yêu AI VIET NAM



## ❖ String

### ❖ Logic operators

```
1 s1 = 'a'
2 s2 = 'b'
3
4 print(s1 == s1)
5 print(s1 == s2)
6
7 print(s1 != s2)
8 print(s1 < s2)
9 print(s1 > s2)
```

```
True
False
True
True
False
```

### ❖ + and \* operators

```
1 s1 = 'Tôi thích '
2 s2 = 'AI!'
3
4 s3 = s1 + s2
5 s4 = s3*2
6
7 print(s3)
8 print(s4)
```

```
Tôi thích AI!
Tôi thích AI!Tôi thích AI!
```

### ❖ Insert into a string

```
1 name = "John"
2 age = 21
3 print(f'Hello {name}. Are you {age} years old?')
```

Hello John. Are you 21 years old?

### ❖ replace() Function

```
1 # Thay thế chuỗi
2 mystr = "Đây là bài học của AI VIET NAM"
3 mystr.replace('AI VIET NAM', 'AIVIETNAM')
```

'Đây là bài học của AIVIETNAM'

## ❖ A container that can contain elements

```
list_name = [element-1, ..., element-n]
```



```
// create a list  
data = [6, 5, 7, 1, 9, 2]
```

data =	6	5	7	1	9	2
index	0	1	2	3	4	5

```
data = [4, 5, 6, 7, 8, 9]
```

Forward  
index

0	1	2	3	4	5
4	5	6	7	8	9

data[0]

4

data[3]

7

## ❖ Structure



**clever** (klev' ar) adjective [from Norwegian *klover*, "skillful"]  
skillful, quick thinking, intelligent  
antonym: stupid, slow, dumb

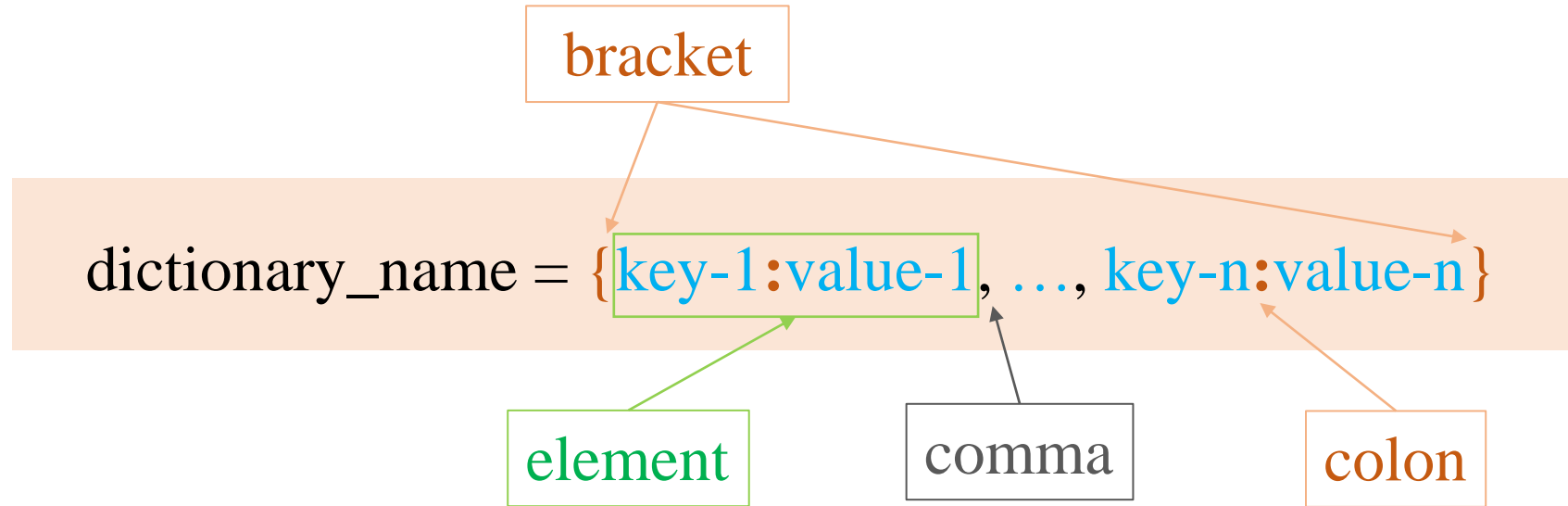
**epidermis** (ep i dur' mis) noun  
[from Greek *epi*, "upon" + *derma*,  
"the skin"] the outermost layer of  
skin; it has no blood vessels and  
covers the next lower skin level,  
the dermis

**exam** (eg zam') noun [shortened  
from *examination* from Latin  
*examinare*, "to weigh"] a test of a  
person's knowledge or skill in a  
certain area, for example, a driving  
exam

**expensive** (ek spen' siv)  
adjective [from Latin *expendere*,  
"to pay"] costly, high-priced  
antonym: inexpensive, cheap

**eye** (ī) noun [from German *auge*,  
"to see"] the sense organ used for  
sight, one of the five senses; the  
other four are hearing, tasting,  
smelling, and touching

**February** (feb' roo wer e) noun  
[from Latin, *februum*,  
"purification"] the second month  
of the year, having 28 days in  
regular years and 29 days every  
fourth year—leap year



## Create a dictionary

```

1 parameters = {'learning_rate': 0.1,
2               'optimizer': 'Adam',
3               'metric': 'Accuracy'}
4
5 print(parameters)
6 print(type(parameters))

```

```

{'learning_rate': 0.1, 'optimizer': 'Adam', 'metric': 'Accuracy'}
<class 'dict'>

```

## ❖ Tabular and image data

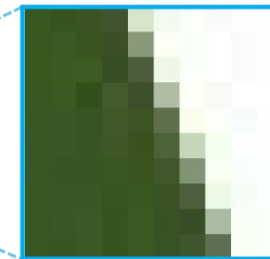
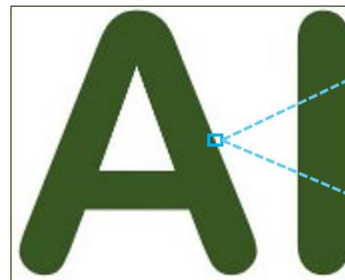
```
import pandas as pd

df = pd.read_csv('advertising_simple.csv')
print(df)
```

	TV	Radio	Newspaper	Sales
0	44	39	45	10
1	17	45	69	12
2	151	41	58	16
3	180	10	58	17
4	8	48	75	7
5	57	32	23	11
6	120	19	11	13
7	8	2	1	4

```
import cv2

# read an image
img = cv2.imread('image.jpg', 1)
```



						33	30	37	47	204	238	250	251	255	254
						87	84	84	84	229	251	255	251	255	254
57	55	58	64	215	243	253	251	255	254	254	254	254	254	254	254
57	60	57	59	138	251	251	255	251	254	254	254	254	254	254	254
57	60	62	56	76	239	251	255	251	254	255	255	255	255	255	255
57	59	51	67	62	176	251	248	254	255	253	253	253	253	253	253
57	59	56	63	61	95	248	251	248	254	253	253	253	253	253	253
57	59	57	62	56	77	198	244	252	254	254	254	254	254	254	254
57	57	57	58	57	65	129	248	251	253	254	254	254	254	254	254
57	58	58	56	58	58	76	237	247	253	255	255	255	255	255	255
57	58	60	54	60	56	56	172	249	251	255	255	255	255	255	255
57	56	60	54	59	56	62	95	249	251	255	255	255	255	255	255



# Outline

## SECTION 1

### Introduction

## SECTION 2

### Data Representation

## SECTION 3

### Functions

## SECTION 4

### Conditions

#### Function

\_ Define a function:

```
def function_name(parameters):  
    '''  
    docstring  
    '''  
    # your code goes here  
  
    return result
```

\_ Default values:

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

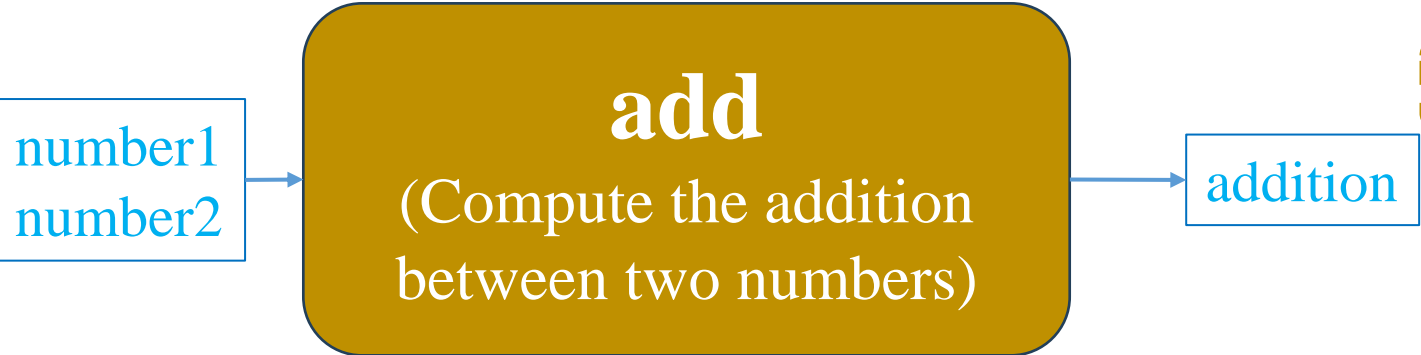
#### Built-in Functions

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

## ❖ Motivation: repeat a task



Input:	what the function needs to do a task
Output:	what comes out of the function



```
# having variables v1 and v2
v1 = 1
v2 = 7

# compute addition between v1 and v2
addition1 = v1 + v2
print(addition1)

# now, having variables v3 and v4
v3 = 4
v4 = 2

# compute addition between v3 and v4
addition2 = v3 + v4
print(addition2)
```

8  
6

## ❖ Motivation: separate a task

```
# give two number
```

```
number1 = 4
```

```
number2 = 6
```

```
# check if the two numbers are positive
```

```
# ...
```

```
# compute percentage
```

```
total = number1 + number2
```

```
per1 = number1/total
```

```
per2 = number2/total
```

```
# print the output
```

```
print(per1)
```

```
print(per2)
```

```
0.4
```

```
0.6
```

```
number1
```

```
number2
```

**compute**

(Compute percentage of  
two numbers)

```
per1
```

```
per2
```

```
# give two number
```

```
number1 = 4
```

```
number2 = 6
```

```
# check if the two numbers are positive
```

```
# ...
```

```
# compute percentage
```

```
per1, per2 = compute(number1, number2)
```

```
# print the output
```

```
print(per1)
```

```
print(per2)
```

# 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'>
```

```
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

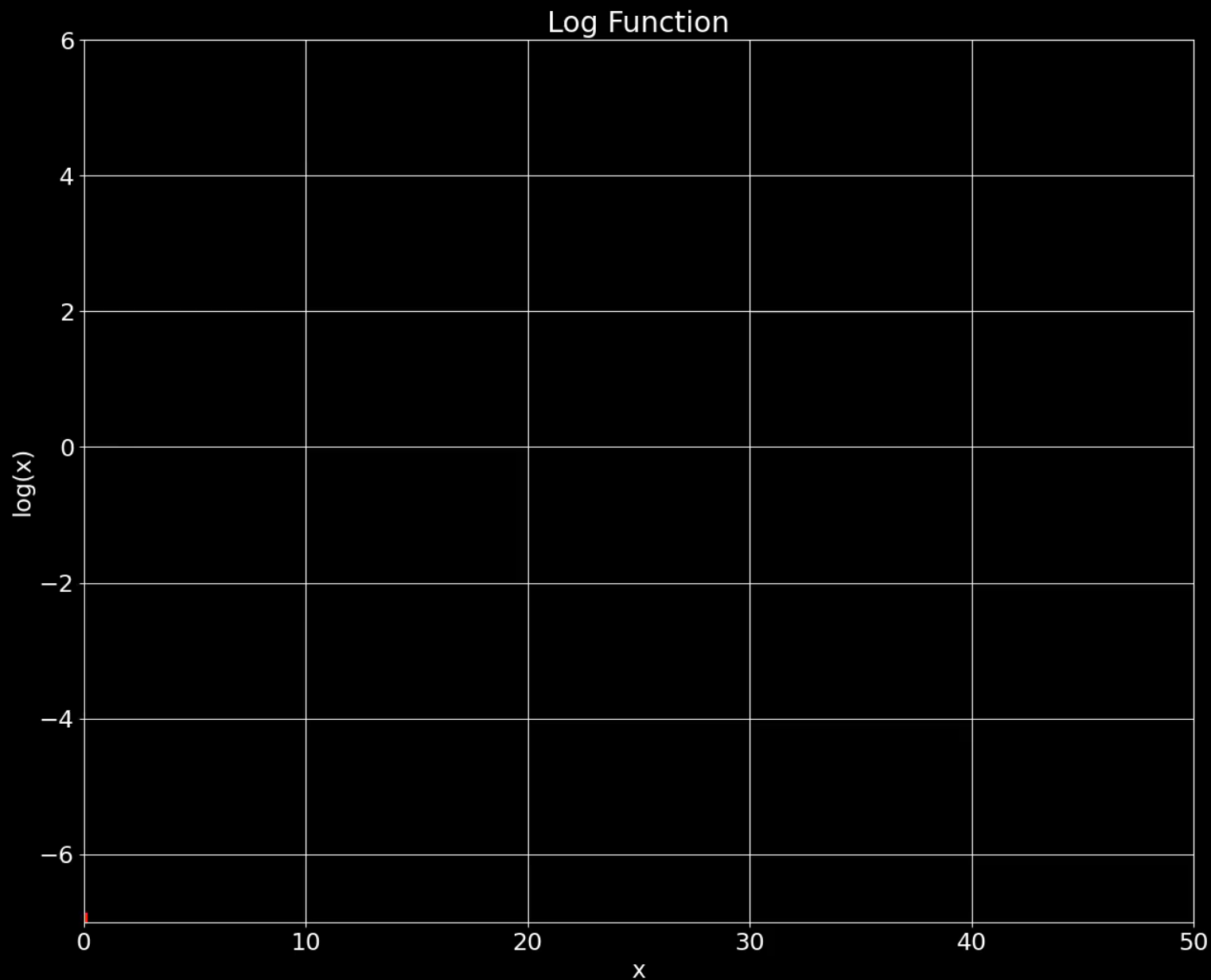
Demo

# Random Module

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

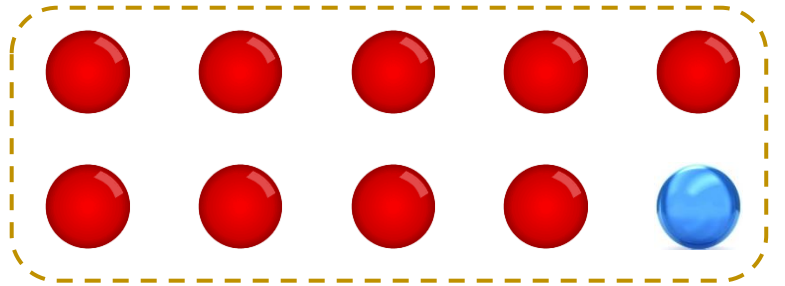
1.3862943611198906

1.0



# Example

## ❖ Design a loss function



A: Get a red ball

B: Get a blue ball

$$p(A) = \frac{9}{10} = 0.9$$

$$p(B) = \frac{1}{10} = 0.1$$

## E: Pick a ball from the basket

Experiment 1

Got a red ball



Experiment 2

Got a blue ball



Which experiment makes you more surprised?

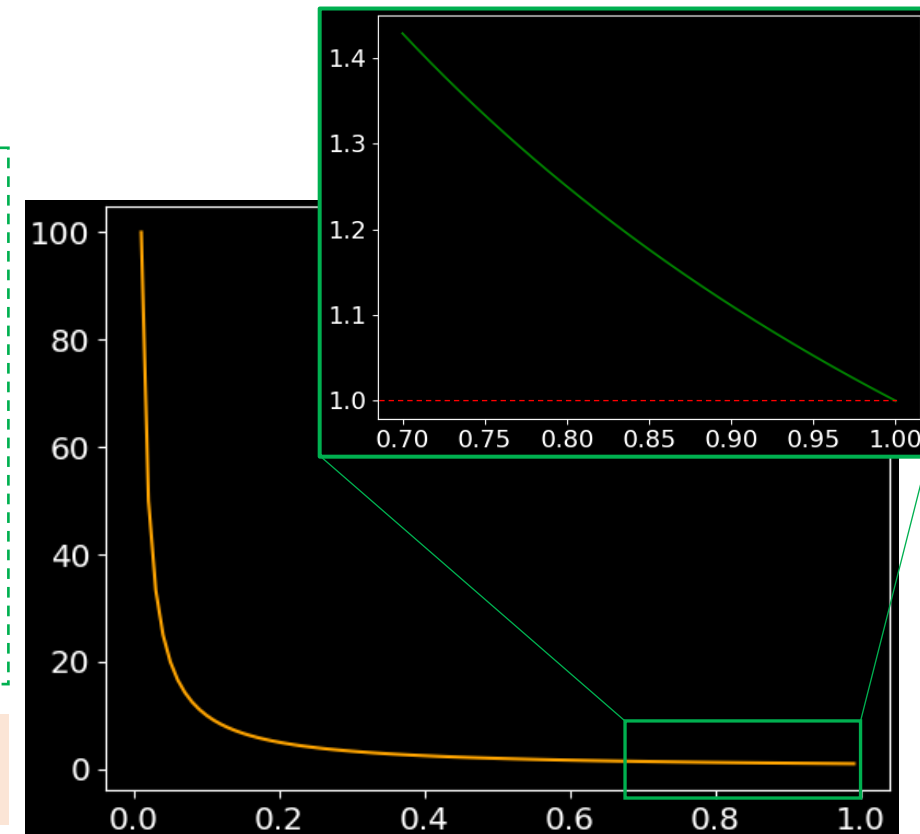
How to measure  
the surprises?

Observation

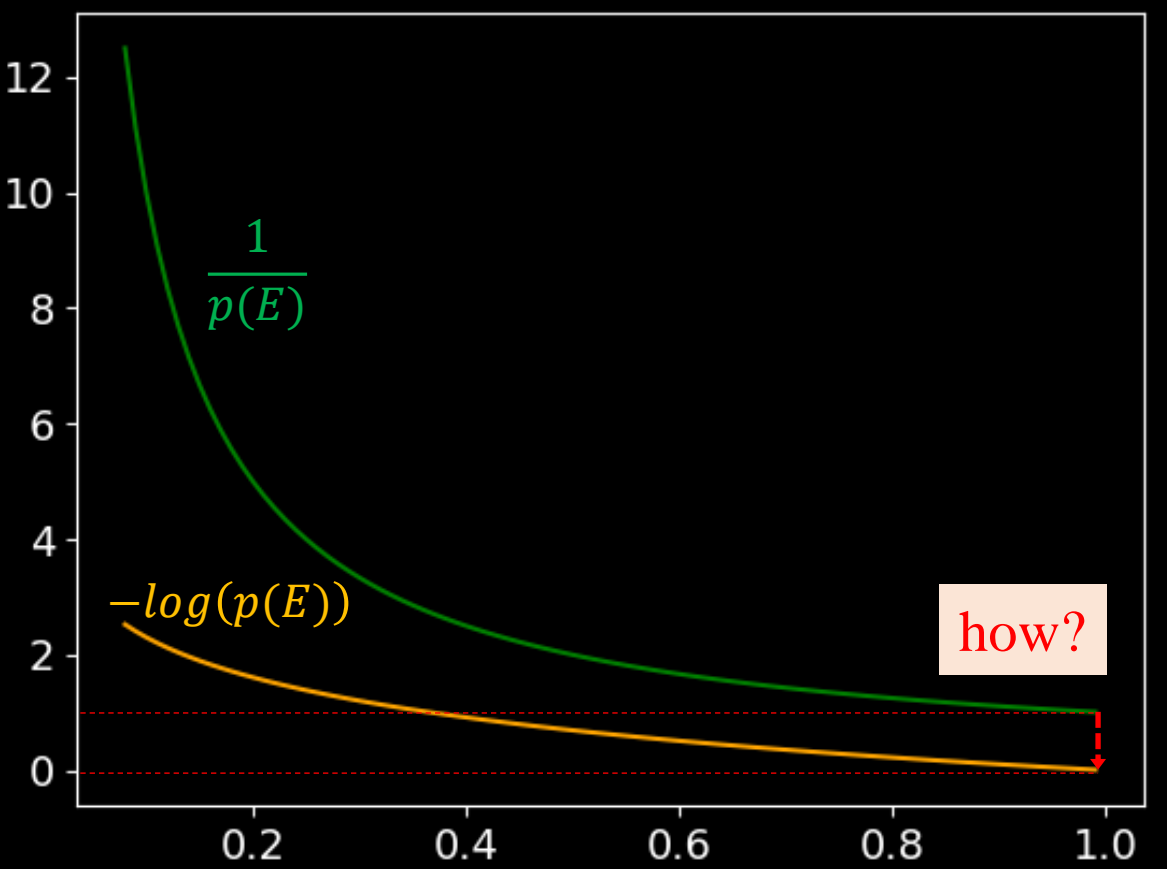
$$\text{Surprise}(E) \updownarrow p(E)$$

$$\rightarrow \text{Surprise}(E) = \frac{1}{p(E)}$$

Problem?

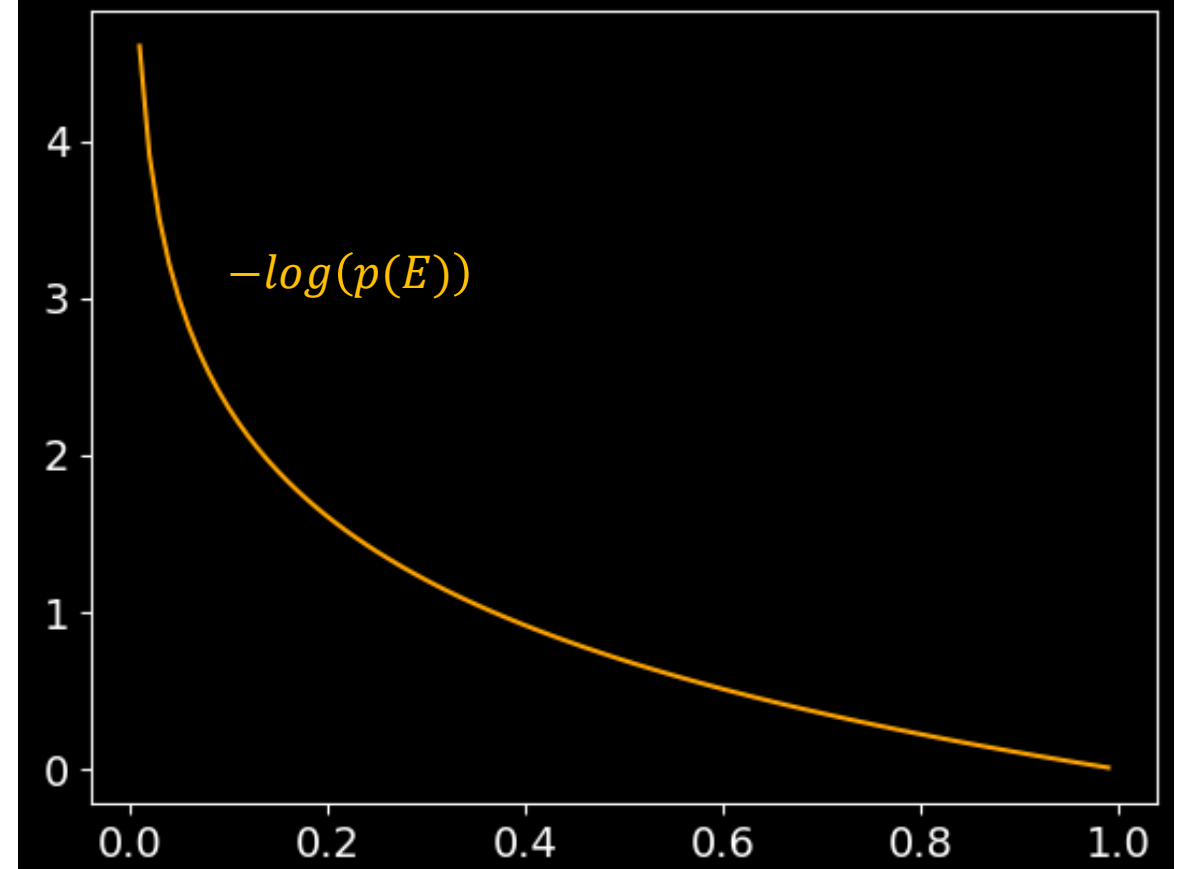






Monotonic decrease of the function surprise(E)

$$\begin{aligned} \log(\text{Surprise}(E)) &= \log\left(\frac{1}{p(E)}\right) \\ &= -\log(p(E)) \end{aligned}$$



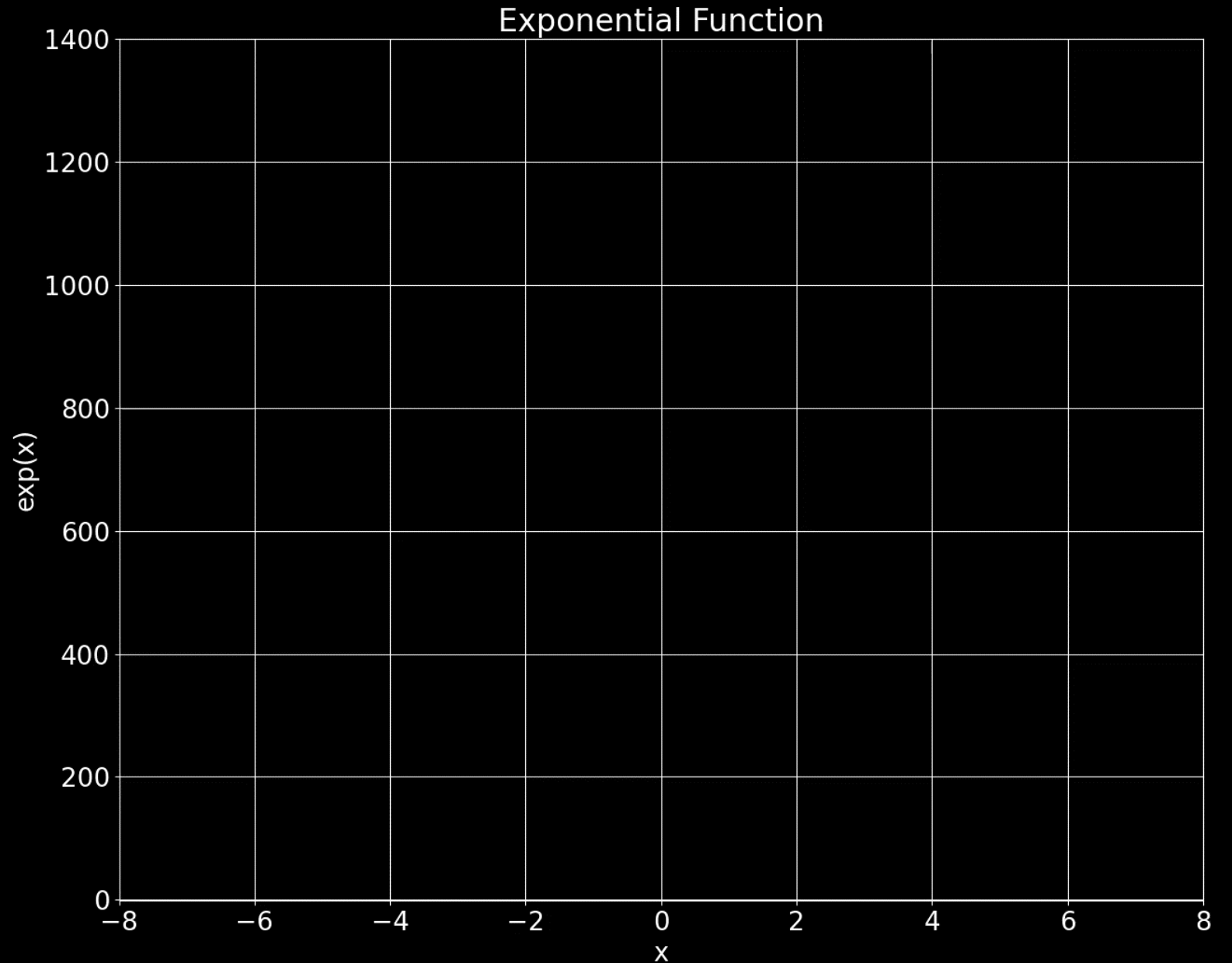
In information theory

$$\text{Information}(x) = -\log(p(x))$$

# Random Module

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

7.38905609893065



## ❖ Compute percentage

**Data**

$$X = \{X_1, \dots, X_N\}$$

**Given the data**

$$X = \{2, 5, 3\}$$

$$S = 10$$

**Formula**

$$S = \sum_{i=1}^N X_i \quad P_i = \frac{X_i}{S}$$

$$X = \{2, 5, 3\}$$

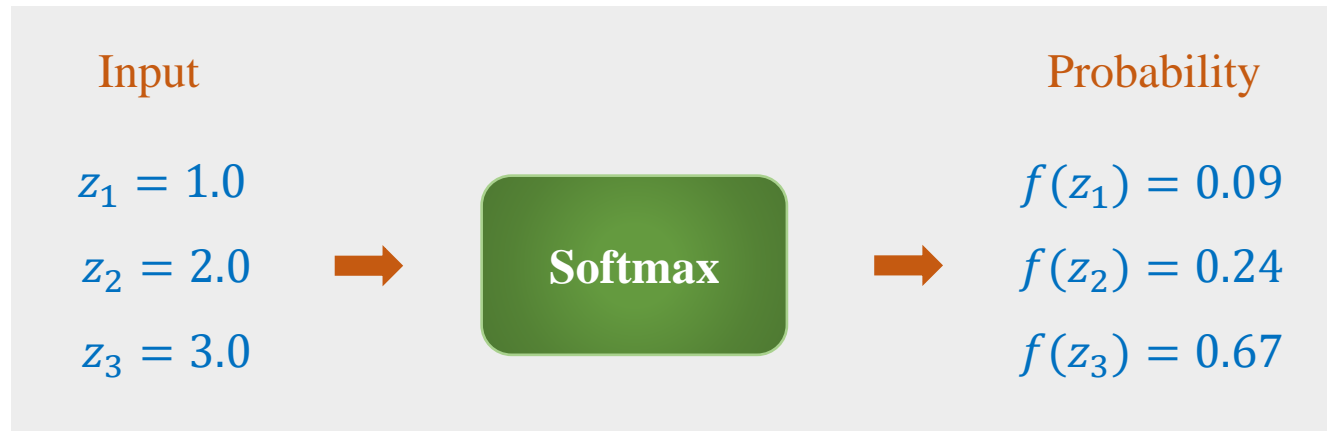
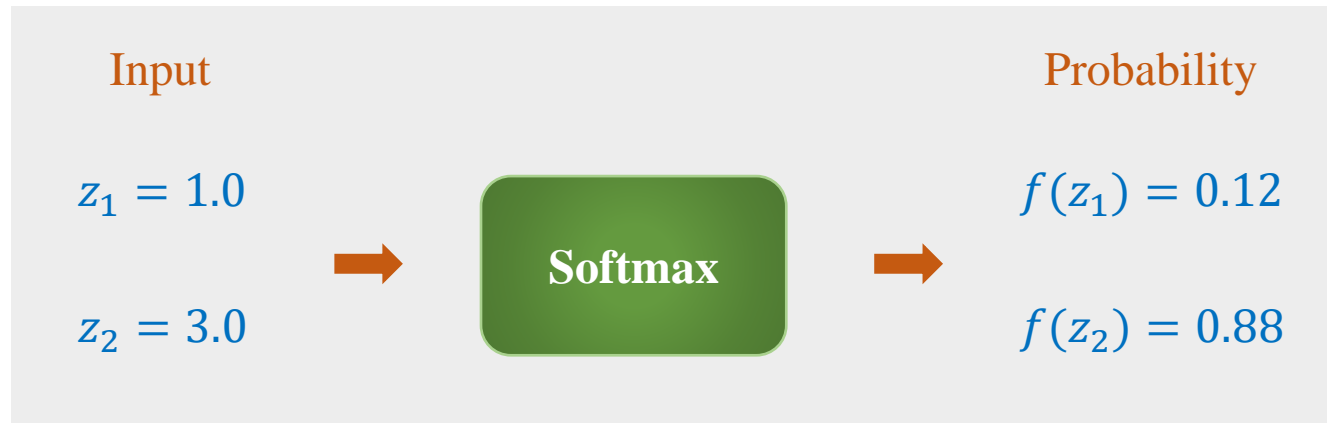
$$\Rightarrow P = \{20\%, 50\%, 30\%\}$$

## ❖ Softmax function

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

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

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



## Implementation (straightforward)

# 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

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

OverflowError: math range error
```

Traceback (most recent call last)

## Implementation (straightforward)

# 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

$$f(x_1) = 0.09$$

$$f(x_2) = 0.24$$

$$f(x_3) = 0.67$$

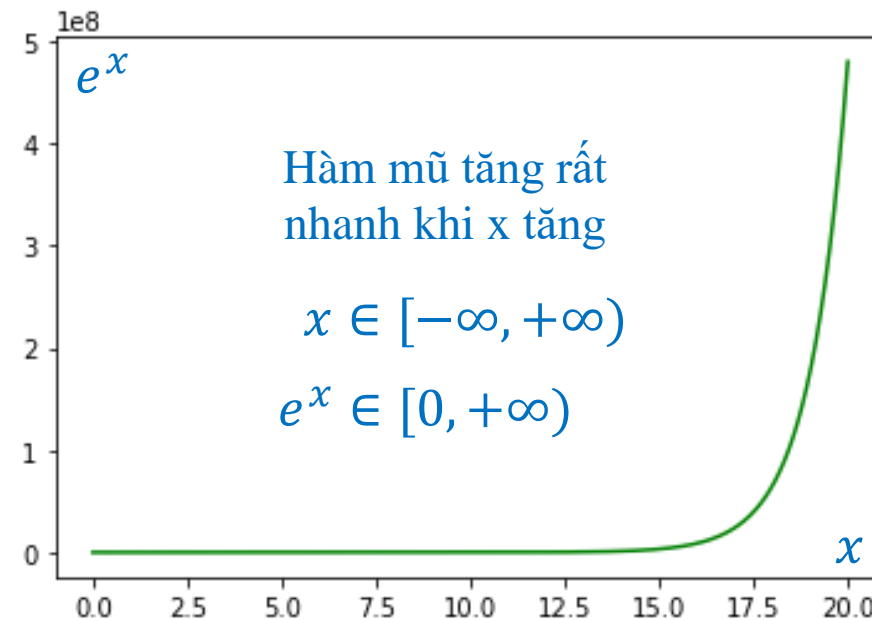
Giá trị nan vì  $e^x$  vượt giới hạn lưu trữ của biến

```
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(v2) + math.exp(v3)
10
11 s1 = math.exp(v1)/total

OverflowError: math range error
```

Traceback (most recent call last)



## 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**

$$x_1 = -2.0$$

$$x_2 = -1.0$$

$$x_3 = 0$$

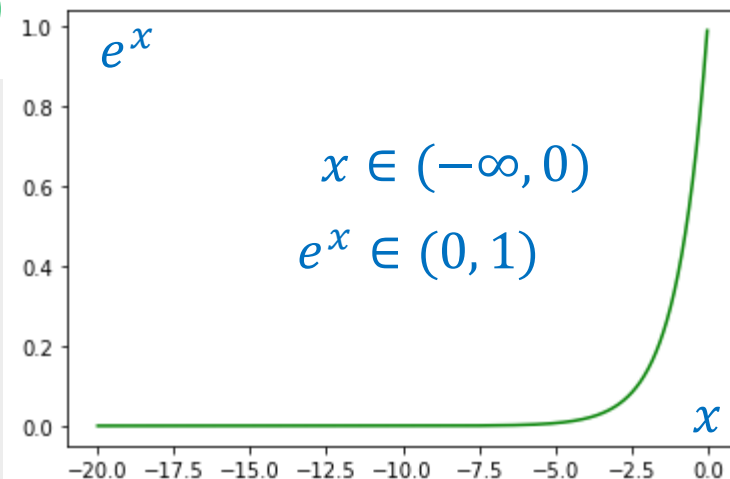
Softmax

**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

# User-defined Functions



## ❖ Syntax

### Notes for function construction

Define function name

Lowercase with underscores  
and begin with a verb

Indentation

Use 4 spaces for indentation

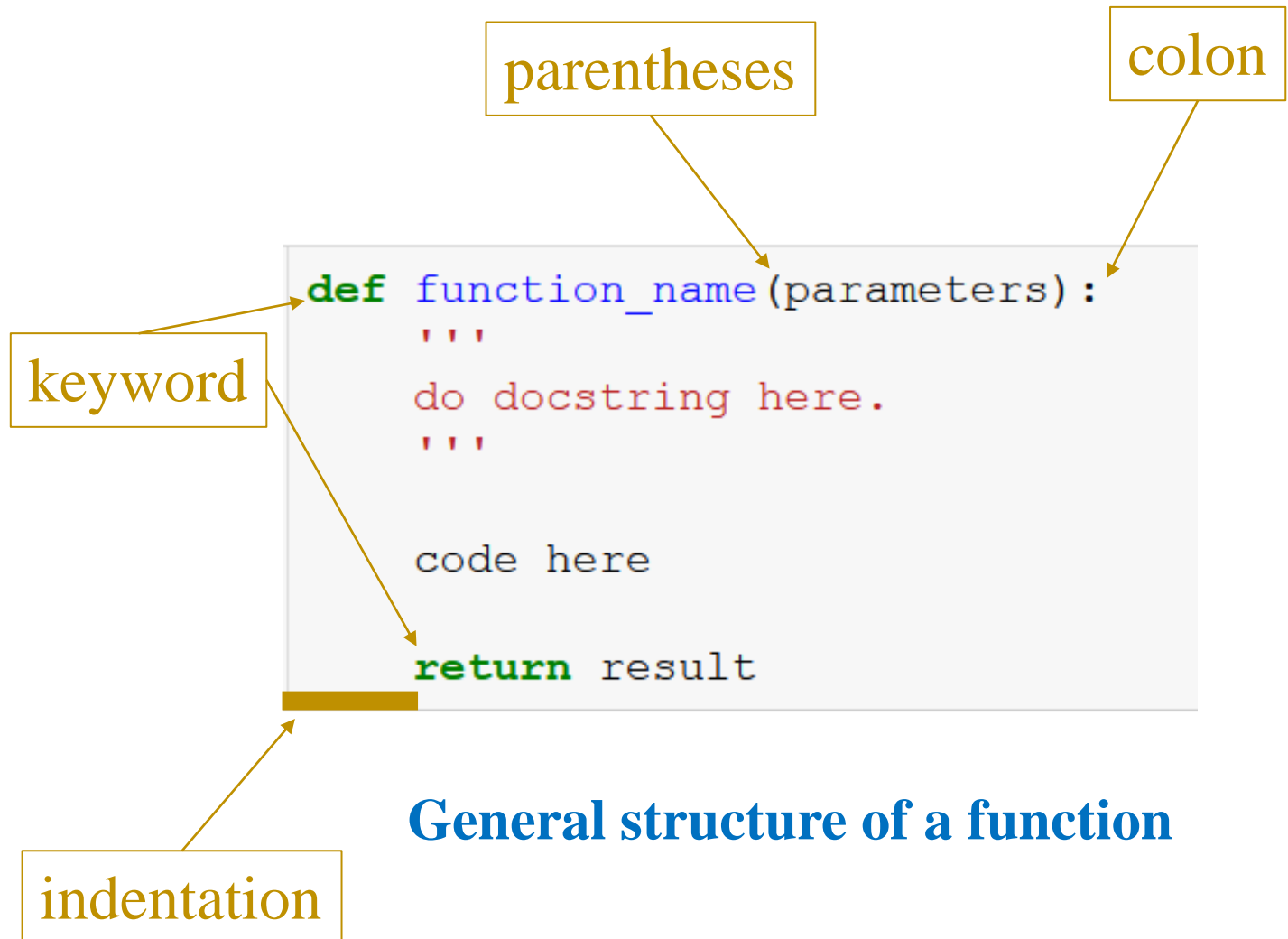
Determine function parameters

Input data help the function done

Do docstring

Explain and describe the function

Output of the function



```
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
```



## ❖ 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 area1 = compute_rectangle_area(5, 6)  
2 print('area 1: ', area1)  
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
```



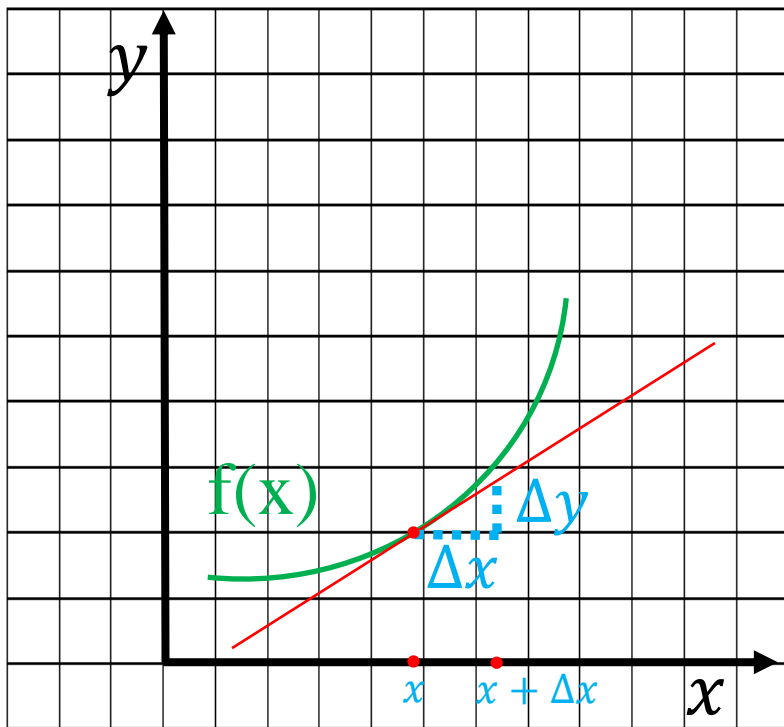
# Example: Derivative

## ❖ Đạo hàm cho hàm liên tục



# Example: Derivative

## ❖ Đạo hàm cho hàm liên tục



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

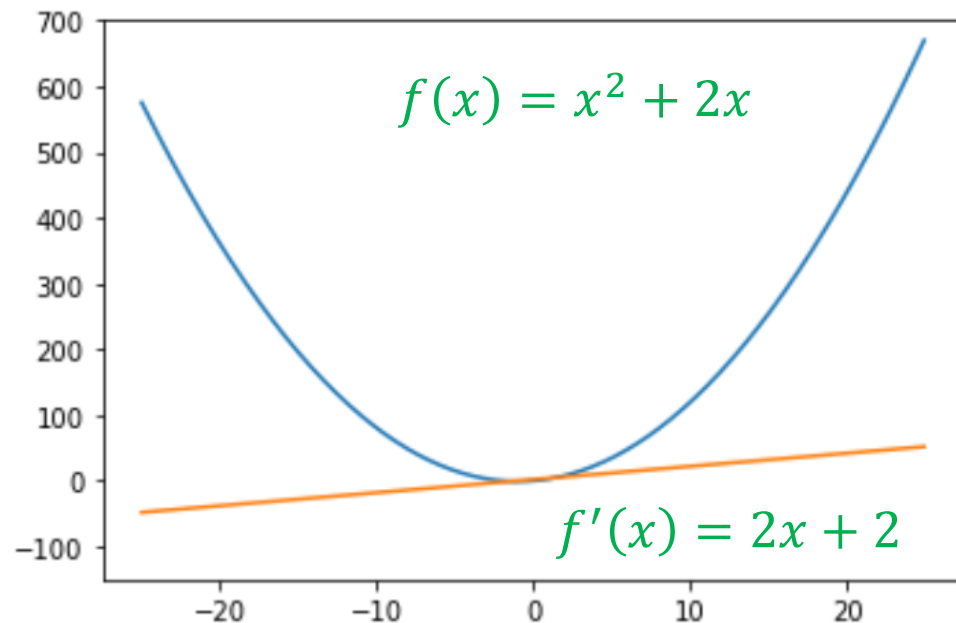
$$\text{Đạo hàm} = \frac{\text{Thay đổi theo } y}{\text{Thay đổi theo } x} = \frac{\Delta y}{\Delta x}$$

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

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

# Example: Derivative

## ❖ 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
```

```
1 d_value = func_derivative(2.0)
2 print('f\'(x=2) is', d_value)
```

f'(x=2) is 6.0

# Example: Derivative

## ❖ Implementation

Cho hàm số  $f(x)$

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

Phương trình đạo hàm

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

Công thức đạo

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

```
def f_x(x):  
    return x*x + 2*x  
  
def derivetive(x, eps):  
    x0 = x  
    f0 = f_x(x0)  
  
    x1 = x0 + eps  
    f1 = f_x(x1)  
  
    derivative = (f1 - f0) / eps  
    return derivative  
  
x = 2  
df_dx = derivetive(x, 0.0001)  
print(df_dx)
```

6.000100000012054

# Outline

## SECTION 1

### Introduction

## SECTION 2

### Data Representation

## SECTION 3

### Functions

## SECTION 4

### Conditions

#### Condition

\_ Comparision Operators:

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

\_ Conditional sentence:

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



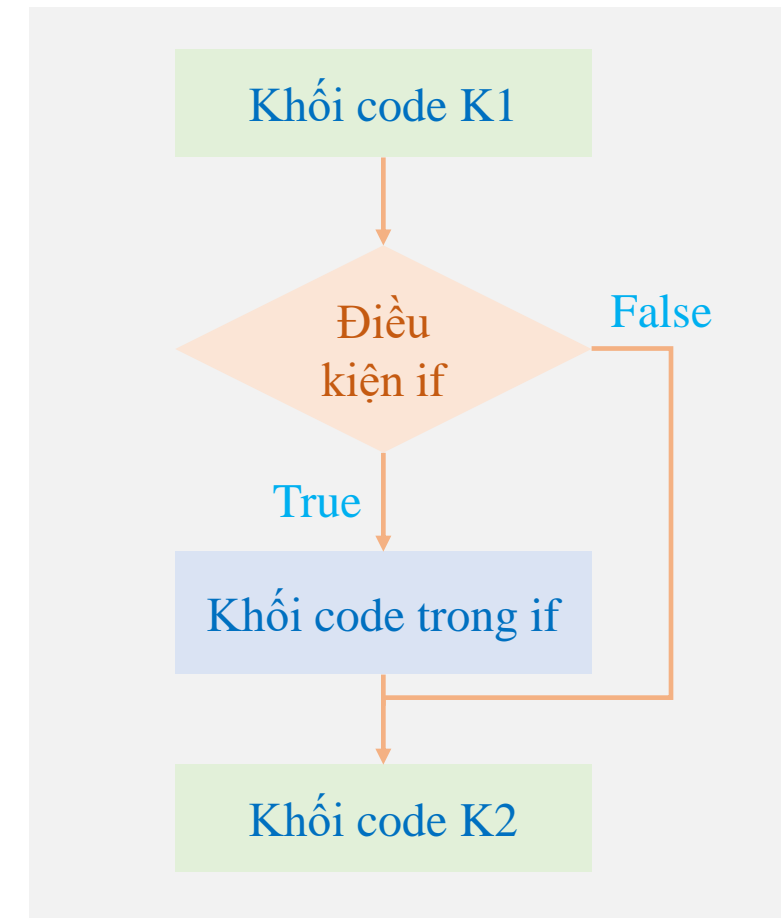
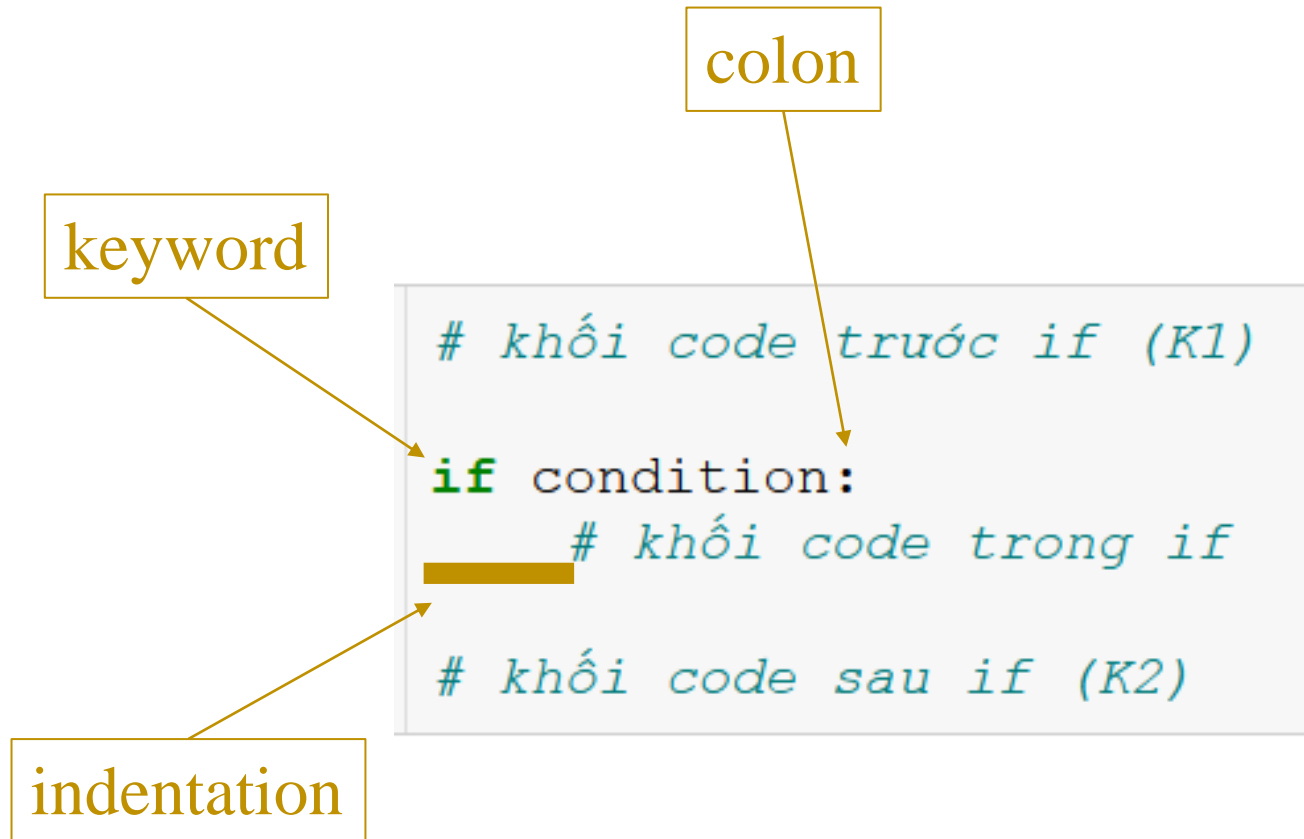
## ❖ Comparison operators

Operator	Meaning
<code>==</code>	Equal
<code>!=</code>	Not equal
<code>&gt;</code>	Greater than
<code>&lt;</code>	Less than
<code>&gt;=</code>	Greater than or equal
<code>&lt;=</code>	Less than or equal

```
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
```

## ❖ if condition



# Branching in Python

if  
condition

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

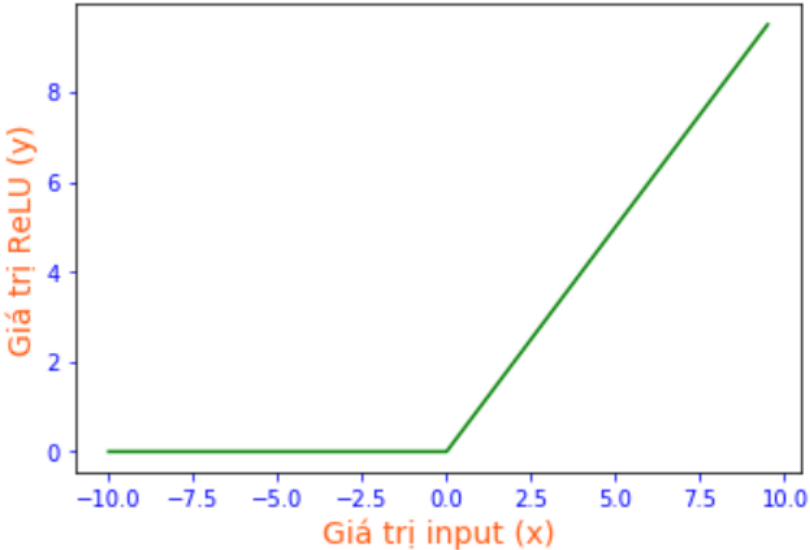
data = 

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

data\_a = ReLU(data)

data\_a = 

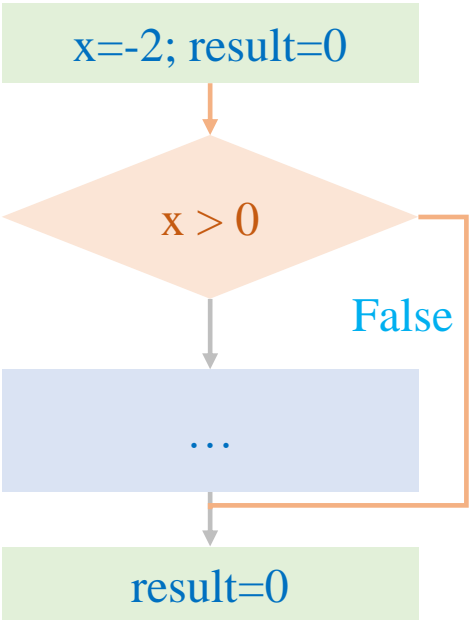
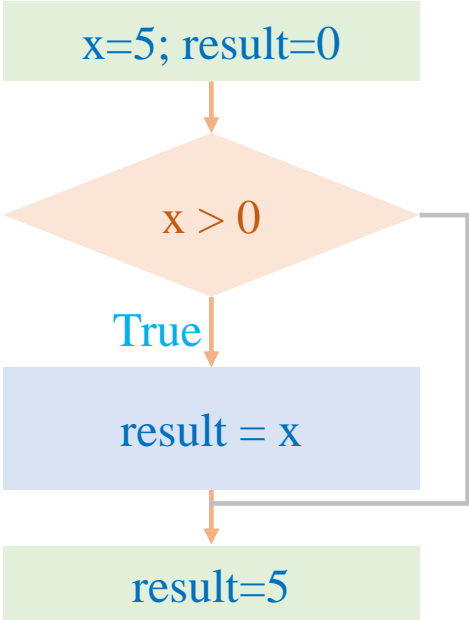
1	5	0	3	0
---	---	---	---	---



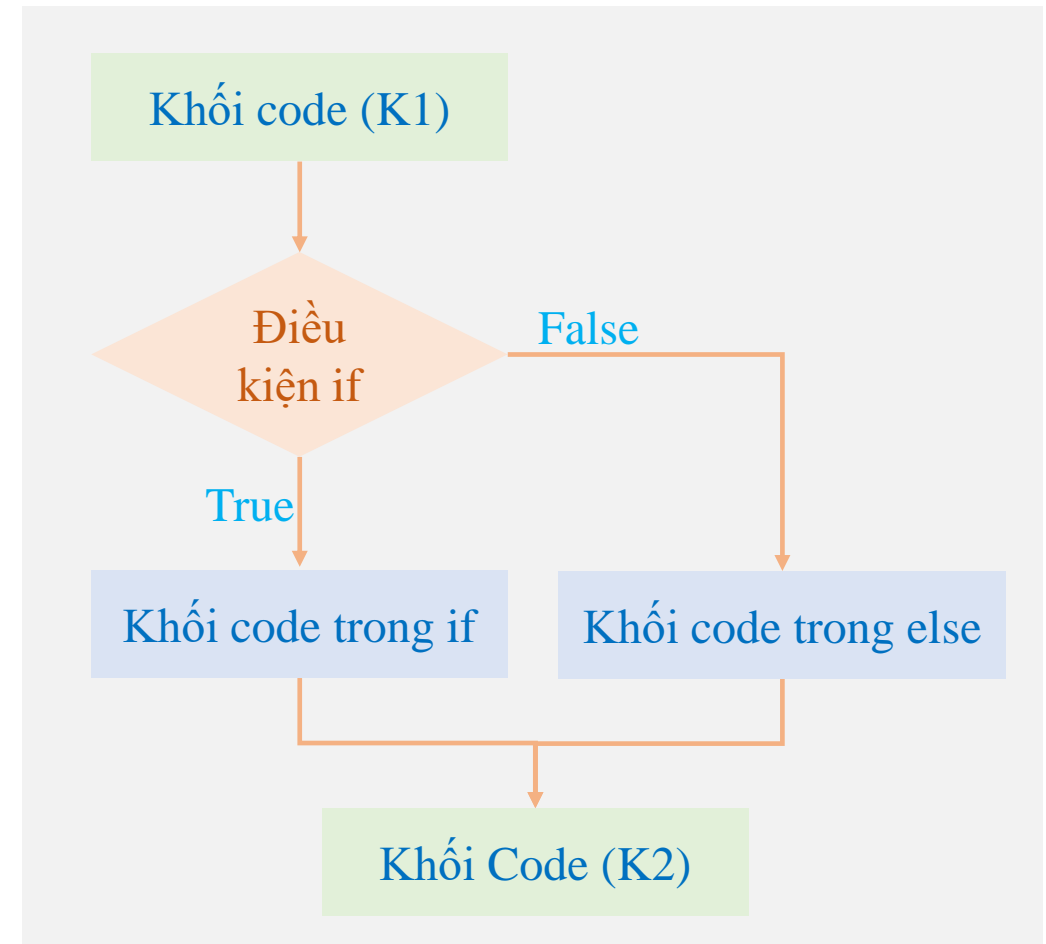
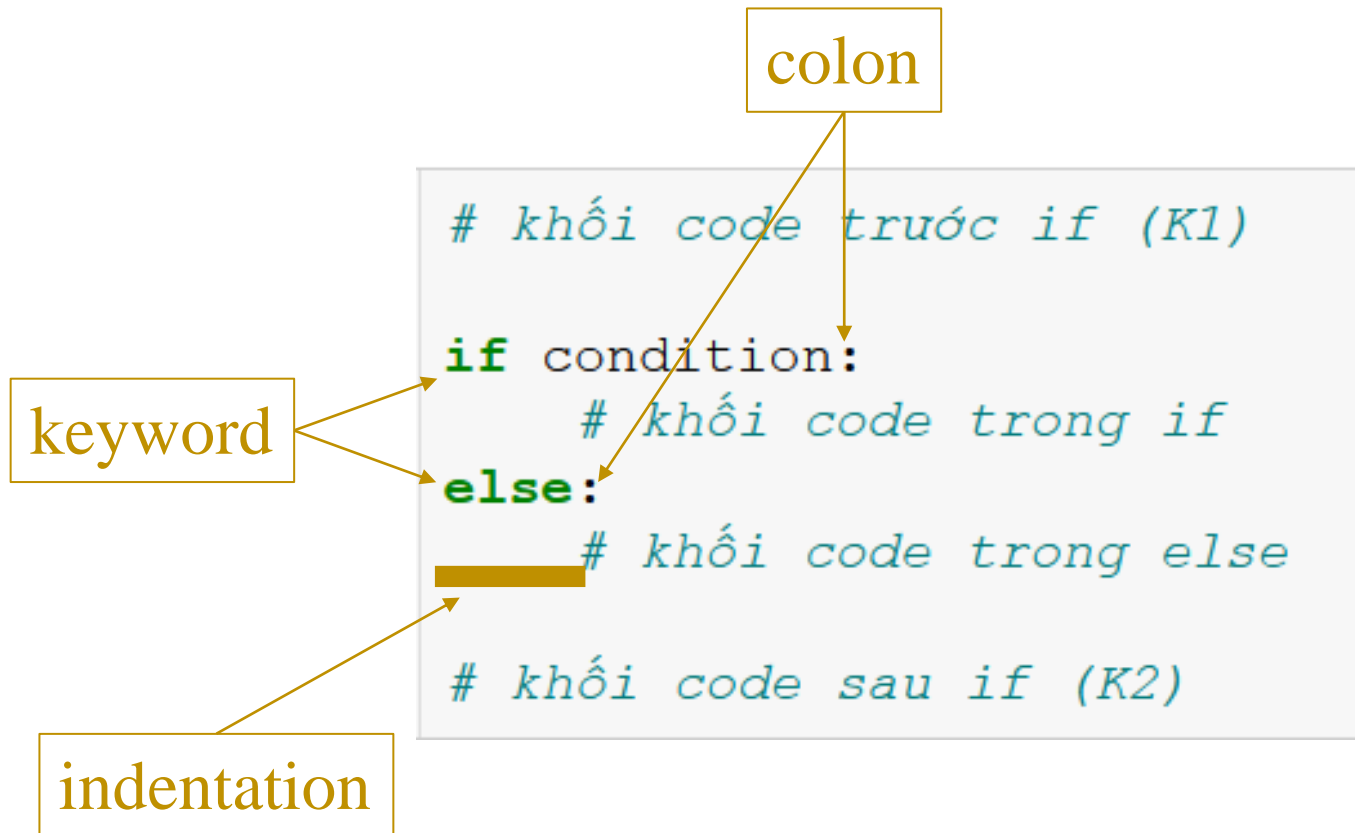
```
1 def ReLU(x):
2     '''
3     This function aims to compute ReLU for a value x.
4
5     x -- an input value
6
7     This function returns the
8     '''
9
10    result = 0
11
12    if x > 0:
13        result = x
14
15    return result
```

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

5  
0



## ❖ if-else condition

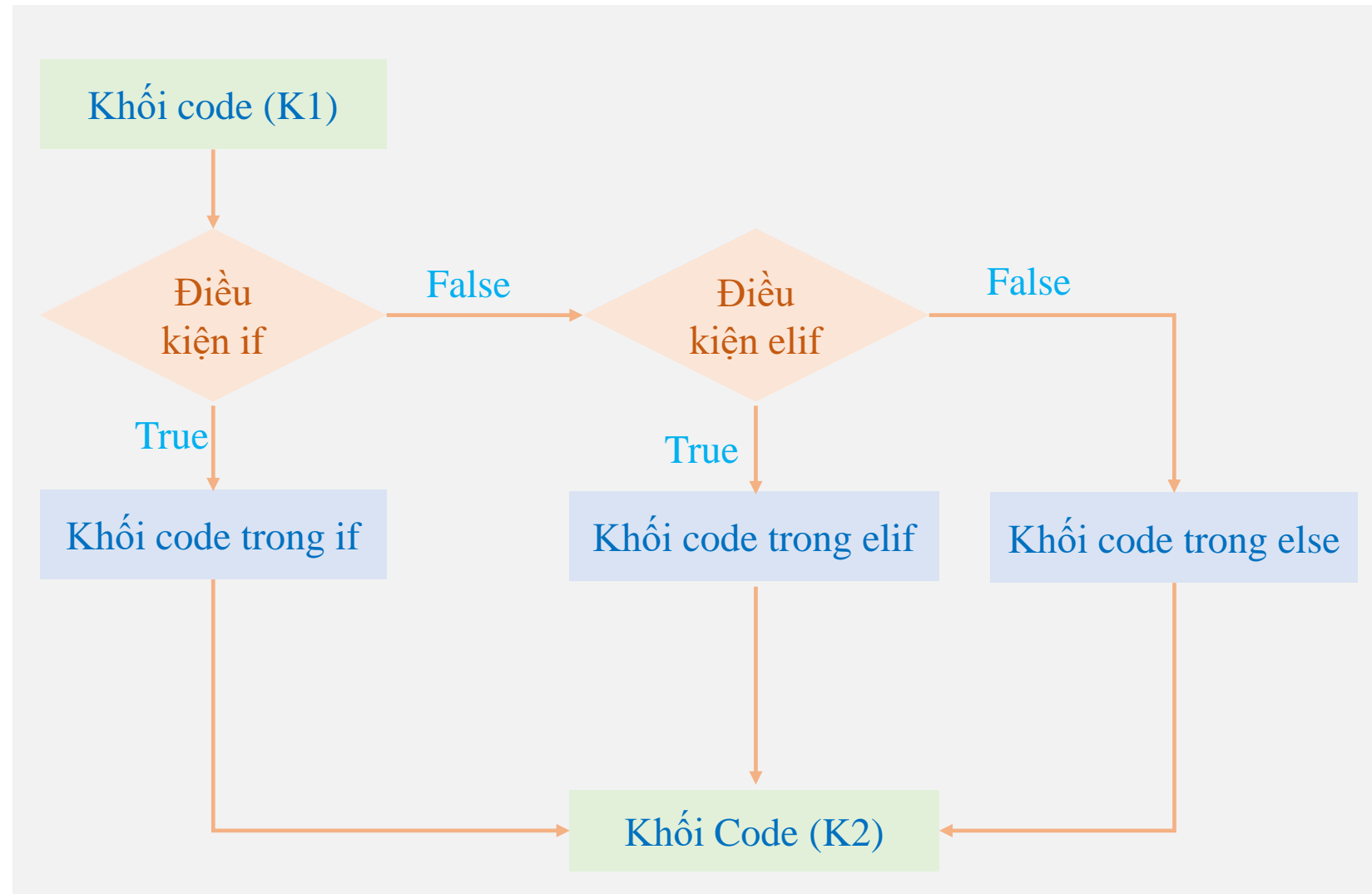


## ❖ if-elif-else condition

colon

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

indentation

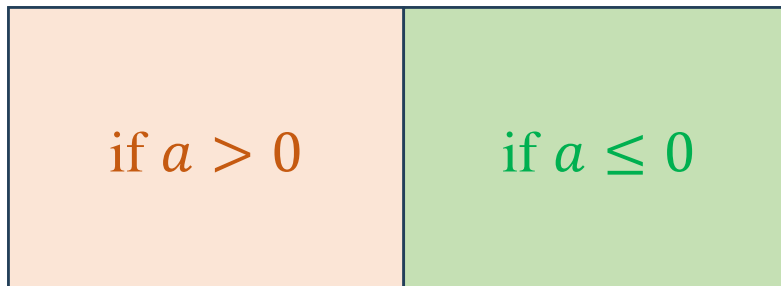


## ❖ if-else condition

Input:  $a$  and  $b$   
if  $a > 0$  then  
 $y = b^2$   
if  $a \leq 0$  then  
 $y = \sqrt{b}$

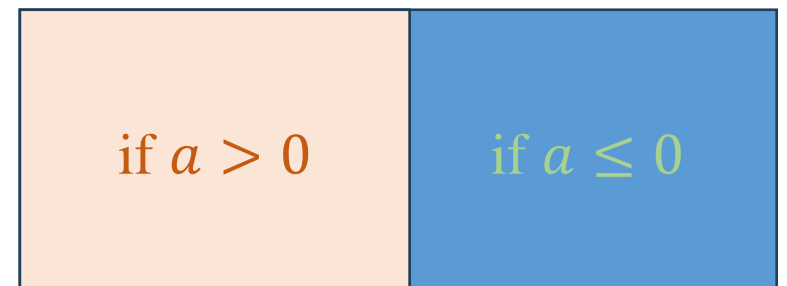
```
1 import math
2
3 def a_function(a, b):
4     result = 0
5
6     if a>0:
7         result = b*b
8     elif a<=0:
9         result = math.sqrt(b)
10
11     return result
12
13 print(a_function(2, 4))
14 print(a_function(-2, 4))
```

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  
> |



## ❖ 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  $y = b$

if  $a == 0$

if  $a == 1$

if  $a == 2$

if  $a == 0$

if  $a == 1$

if  $a == 2$

```
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

QUIZ TIME



# Alternative to If-Else (1)

Input:  $a$  and  $b$

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

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



Input:  $a$  and  $b$

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

```
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
```

# Alternative to If-Else (2)

Input:  $a$  and  $b$

if  $a > 0$  then

$$y = b^2$$

if  $a < 0$  then

$$y = \sqrt{b}$$



Input:  $a$  and  $b$

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

```
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
```

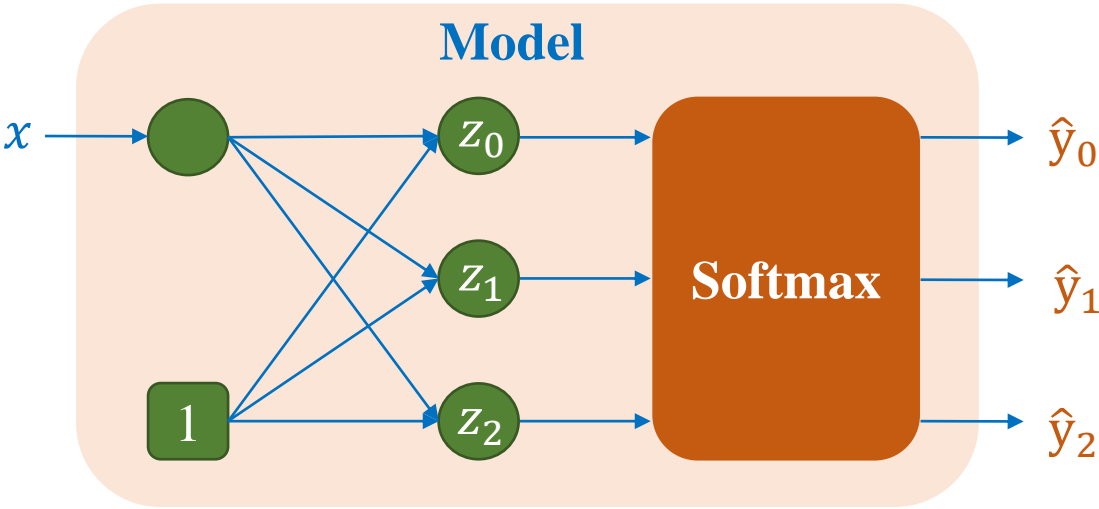
## ❖ One more example

Feature	Label	
Petal_Length	Label	
1.4	0	Yellow
1.3	0	
1.5	0	
4.5	1	Grey
4.1	1	
4.6	1	
5.2	2	Purple
5.6	2	
5.9	2	

#features = 1

#classes = 3

$y \in \{0,1,2\}$



$$y = 0 \rightarrow L(\theta) = -\log(\hat{y}_0)$$

$$y = 1 \rightarrow L(\theta) = -\log(\hat{y}_1)$$

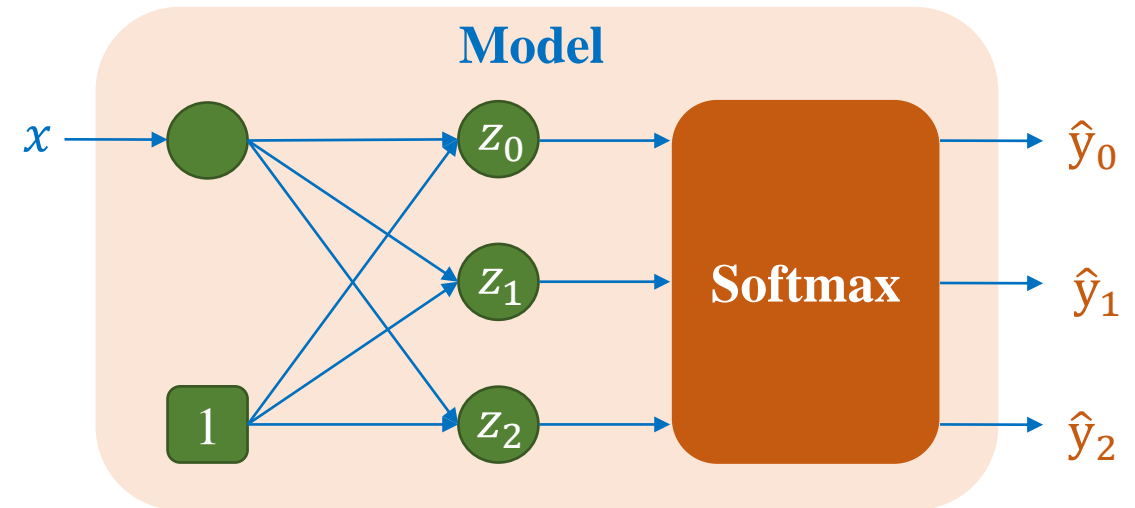
$$y = 2 \rightarrow L(\theta) = -\log(\hat{y}_2)$$

How to convert into a single function?

Feature	Label
Petal_Length	Label
1.4	0
1.3	0
1.5	0
4.5	1
4.1	1
4.6	1
5.2	2
5.6	2
5.9	2

#features = 1

#classes = 3

 $y \in \{0,1,2\}$ 

$$L(\theta) = - \underbrace{\frac{y(1-y)}{-2} \log(\hat{y}_2)}_{y=2} - \underbrace{y(2-y) \log(\hat{y}_1)}_{y=1} - \underbrace{(1-y) \left( \frac{2-y}{2} \right) \log(\hat{y}_0)}_{y=0}$$

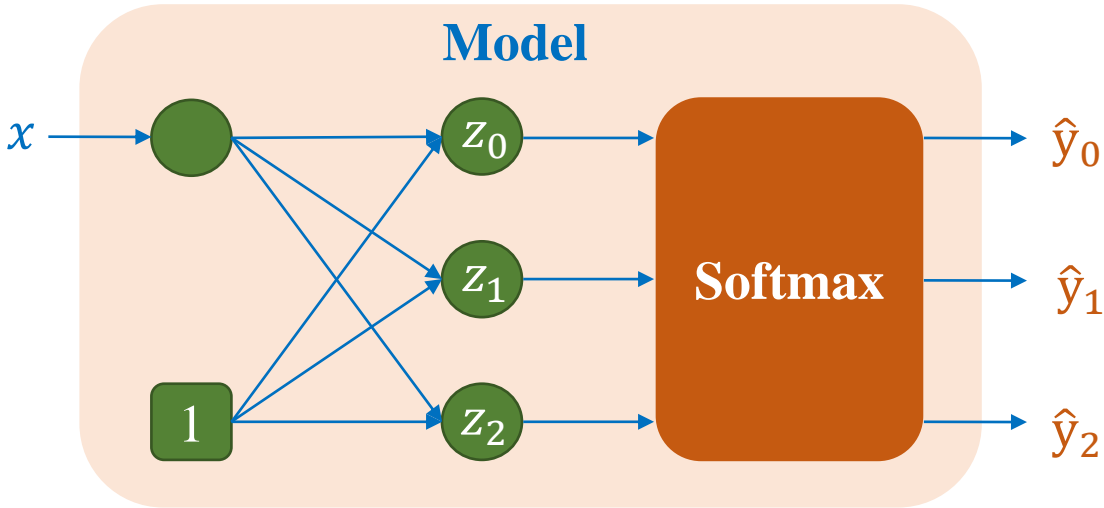
Ok! but awkward!!! ... and how to improve?

Feature	Label	
Petal_Length	Label	
1.4	0	Yellow
1.3	0	
1.5	0	
4.5	1	Grey
4.1	1	
4.6	1	
5.2	2	Purple
5.6	2	
5.9	2	

#features = 1

#classes = 3

$y \in \{0,1,2\}$



One-hot encoding for label

$$\mathbf{y} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \end{bmatrix}$$

$$y_i \in \{0,1\}$$

$$\sum_i y_i = 1$$

$$y = 0 \rightarrow \mathbf{y} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$y = 1 \rightarrow \mathbf{y} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$y = 2 \rightarrow \mathbf{y} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Loss function

$$L(\mathbf{y}, \hat{\mathbf{y}}) = -y_2\log(\hat{y}_2)-y_1\log(\hat{y}_1)-y_0\log(\hat{y}_0)$$

$$= -\sum_i y_i\log(\hat{y}_i)$$

# Alternative to If-Else (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  $v$

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

$$y = v \cdot u$$

$$v = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

```

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

# If-Else and Dictionary

## ❖ Mixed conditions

Input:  $a$

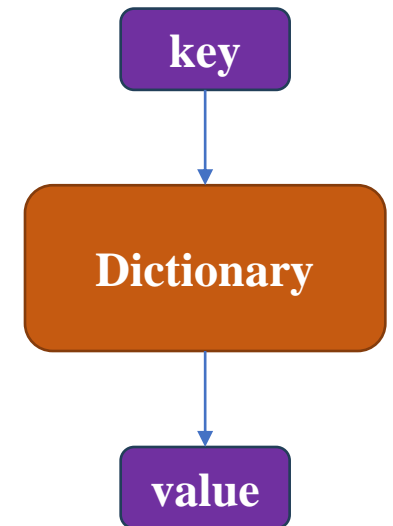
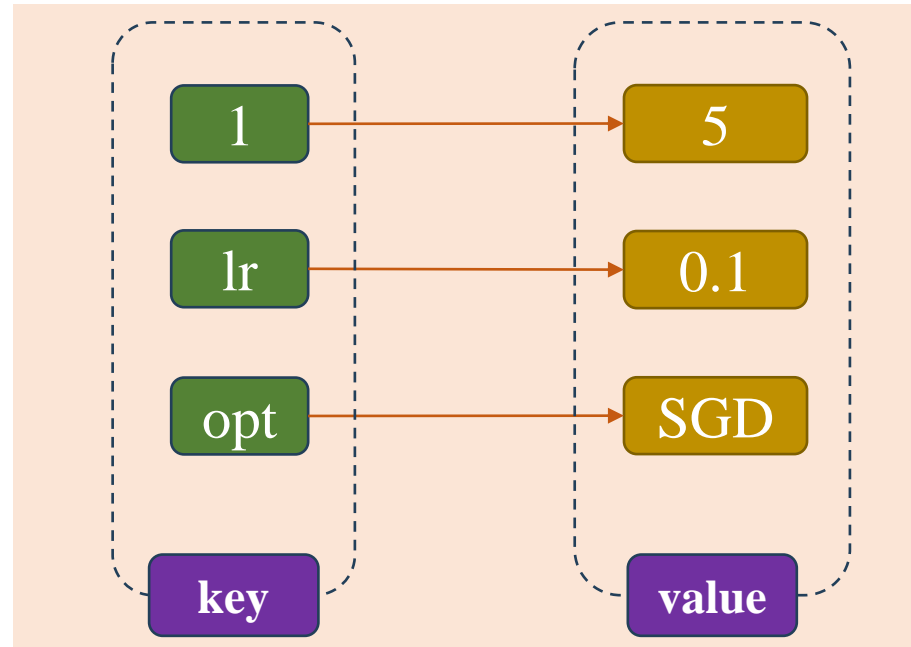
if  $a = 1$  then  $y = 5$

if  $a = \text{'lr'}$  then  $y = 0.1$

if  $a = \text{'opt'}$  then  $y = \text{'SGD'}$

```
1 def get_y(a):
2     if a == 1:
3         y = 5
4     elif a == 'learning_rate':
5         y = 0.1
6     elif a == 'optimizer':
7         y = 'SGD'
8
9     return y
10
11 y = get_y('optimizer')
12 print(y)
```

SGD



```
1 options = {1: 5, 'learning_rate': 0.1, 'optimizer': 'SGD'}
2
3 print(options[1])
4 print(options['optimizer'])
```

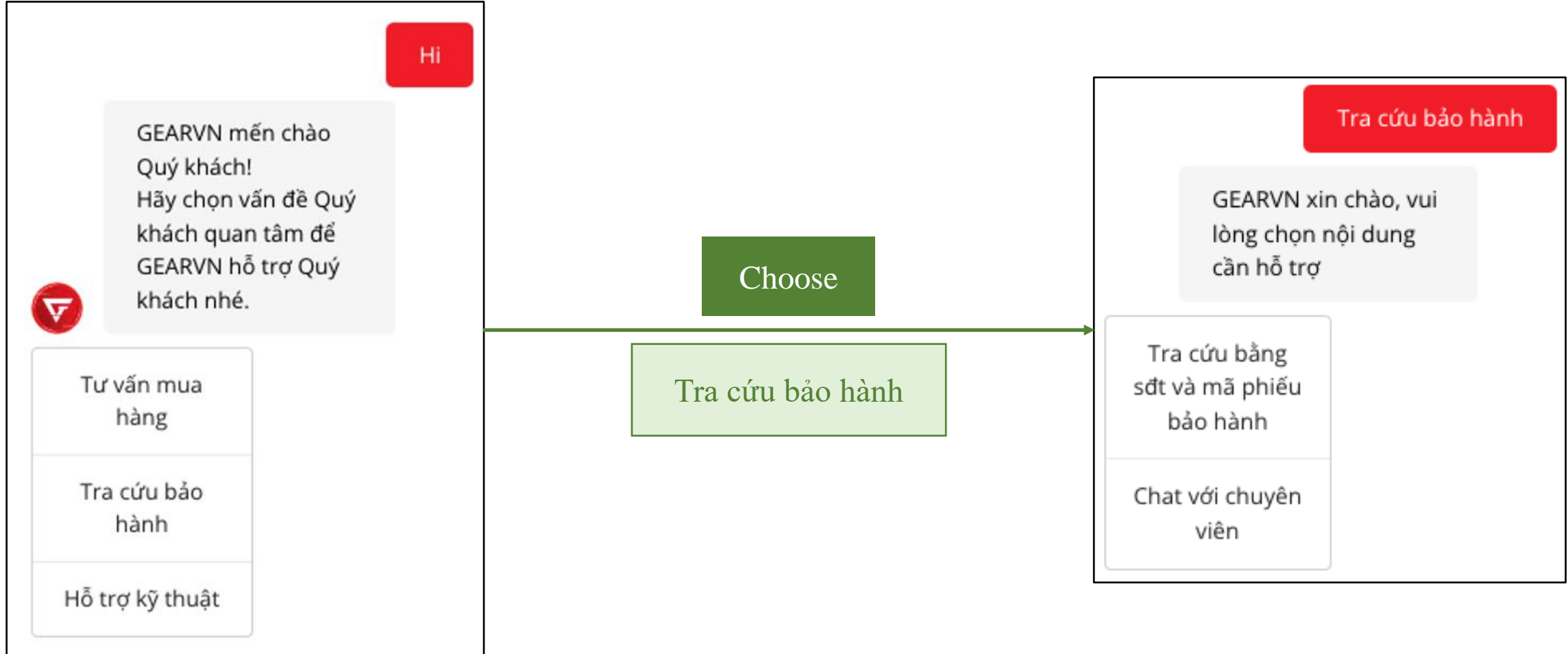
5  
SGD

# Rule-based Chatbot

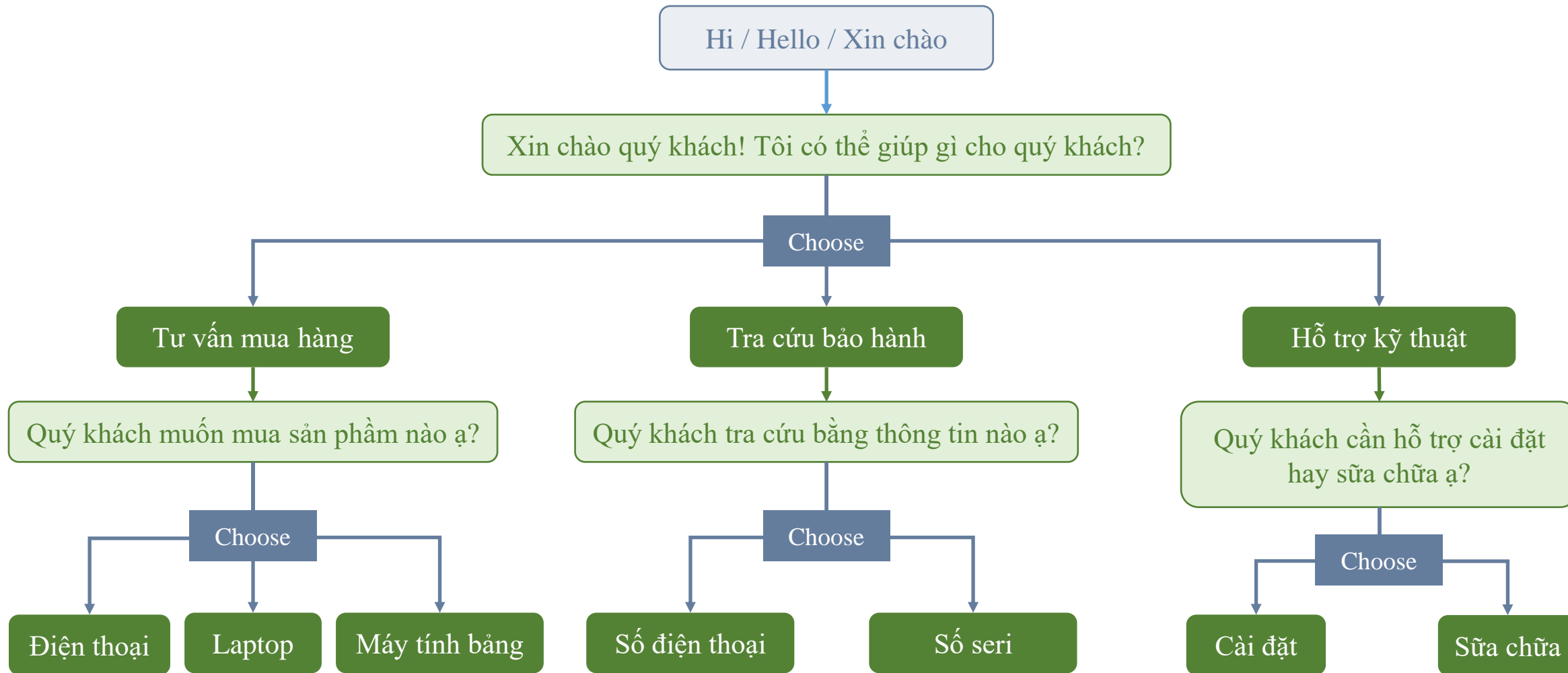




## ❖ Getting Started

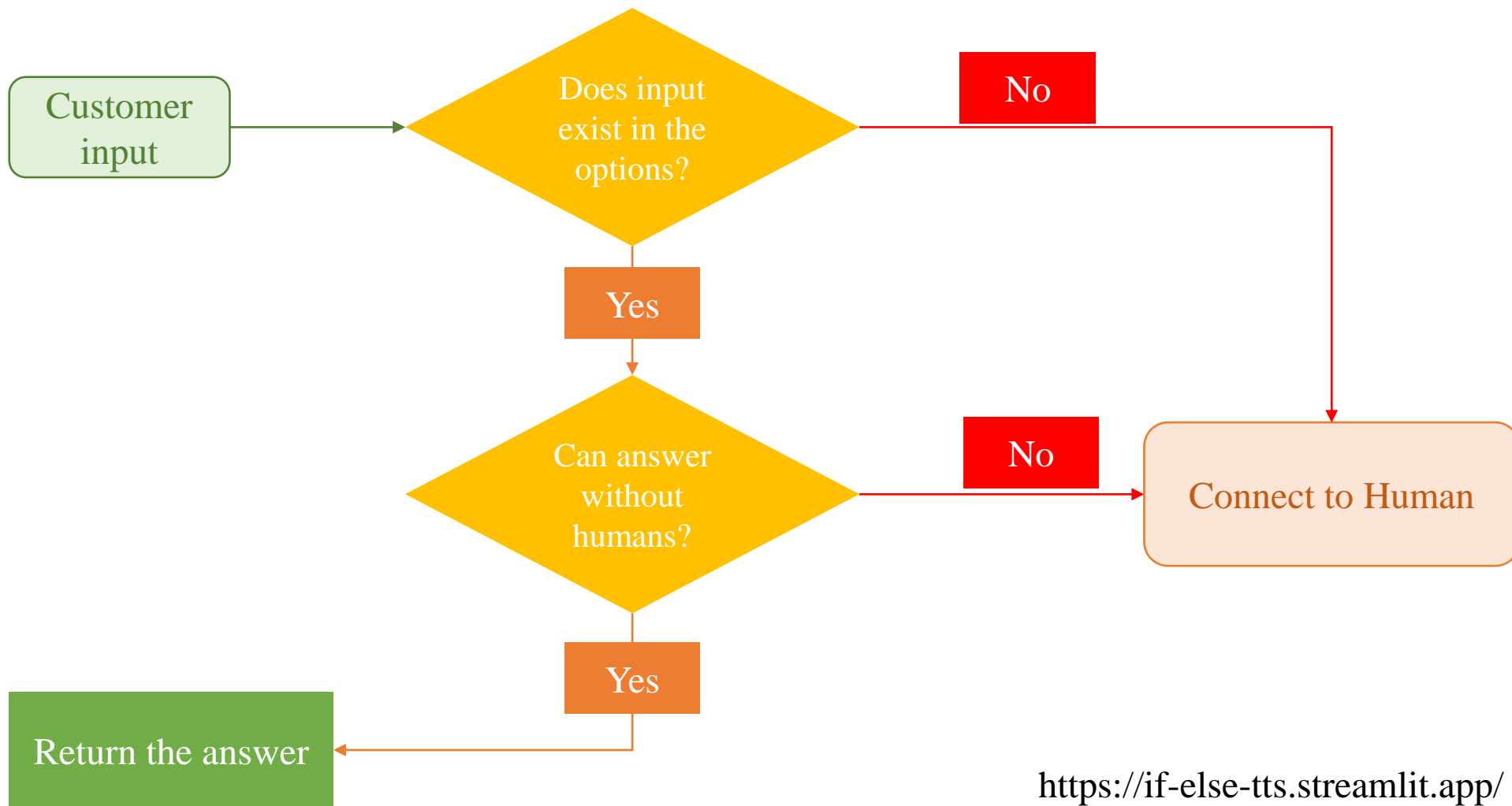


## ❖ Chat tree





## ❖ Flowchart



<https://if-else-tts.streamlit.app/>



# 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-env`
- \_ 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(parameters):  
    '''  
    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

<code>print(params)</code>	Print value onto the screen
<code>type(params)</code>	Return class type of variable
<code>input(prompt)</code>	Ask user to input a string
<code>int()</code> , <code>float()</code>	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:

<code>break</code>	Exit the loop
<code>continue</code>	Move to next iteration

