

KỸ THUẬT LẬP TRÌNH

CHƯƠNG 1: HÀM

NỘI DUNG

- 01.** Định nghĩa & cú pháp hàm
- 02.** Tạo và gọi hàm
- 03.** Phạm vi biến trong hàm
- 04.** Đối số của hàm
- 05.** Hàm có giá trị trả về
- 06.** Hàm khuyết danh

1

H

nhiệm


thể c

1. Định nghĩa và chú pháp hàm

1. Định nghĩa và cú pháp hàm

❖ Cú pháp

Func



```
1 | def ten_ham( cac_tham_so ):  
2 |     "function_docstring"  
3 |     function_suite  
4 |     return [bieu_thuc]
```

2. Tạo và gọi hàm

❖ Hàm không có tham số:

```
#Định nghĩa hàm
def print_student_info():
    "Đây là chuỗi tùy ý, không thực thi"
    print("Nguyễn Văn A")
    print("Class: IT01")
    print("Student ID: 29694018")

#Gọi hàm
print_student_info()
```

Output:

```
Nguyễn Văn A
Class: IT01
Student ID: 29694018
```

❖ Hàm có tham số:

```
#Định nghĩa hàm
def printme(str1):
    print(str1)
    print(str1 + "Python")

#Gọi hàm
printme('Hello')
```

Output:

```
Hello
HelloPython
```

3. Phạm vi biến tron

3. Phạm vi biến trong hàm

❖ Biến toàn cục (global):

```
# scoping.level.1.py
def my_function():
    # this is defined in the local scope of the function
    test = 1
    print('my_function:', test)

test = 0 # this is defined in the global scope
my_function()
print('global:', test)
```

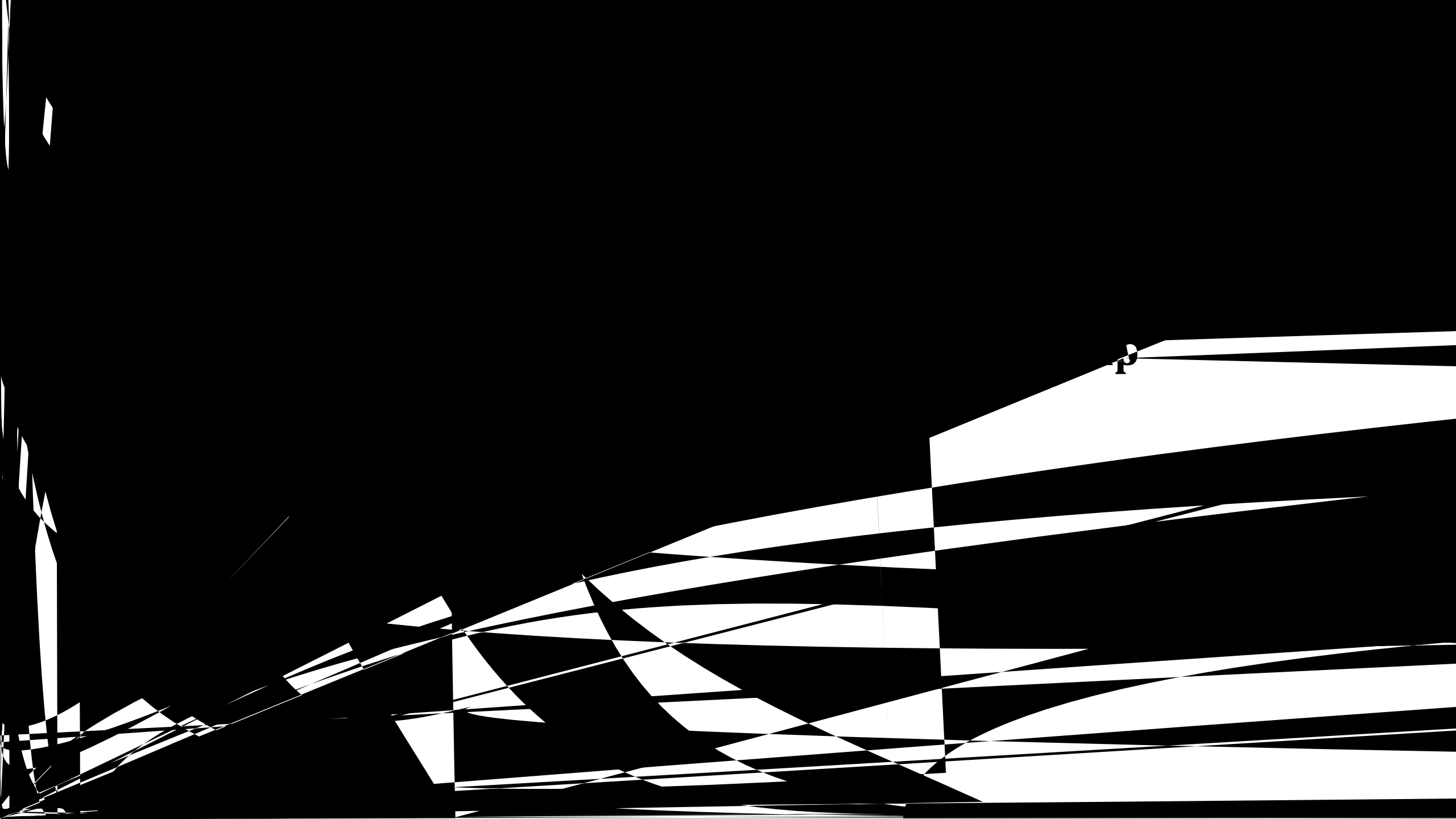
Output:

```
my_function: 1
global: 0
```

Biến toàn cục (global variable) có thể gọi bất cứ đâu trong chương trình.

statement):

```
local stat  
er():  
= 1 # scope  
ner()  
ocal st  
2 nearest enclosing sc  
ner:', to  
uter:' test)
```



4. Đối số của hàm

❖ Truyền đối số theo thứ tự (positional)

```
# arguments.positional.py
def func(a, b, c):
    print(a, b, c)

func(1, 2, 3) # prints: 1 2 3
```

Q&A:

Chương trình cho kết quả gì khi gọi hàm:

func (1)

func (1, 2)

func (1, 2, 3, 4)

4. Đối số của hàm

❖ Truyền đối số theo từ khoá (keyword)

```
# arguments.keyword.py
def func(a, b, c):
    print(a, b, c)

func(a=1, c=2, b=3) # prints: 1 3 2
```

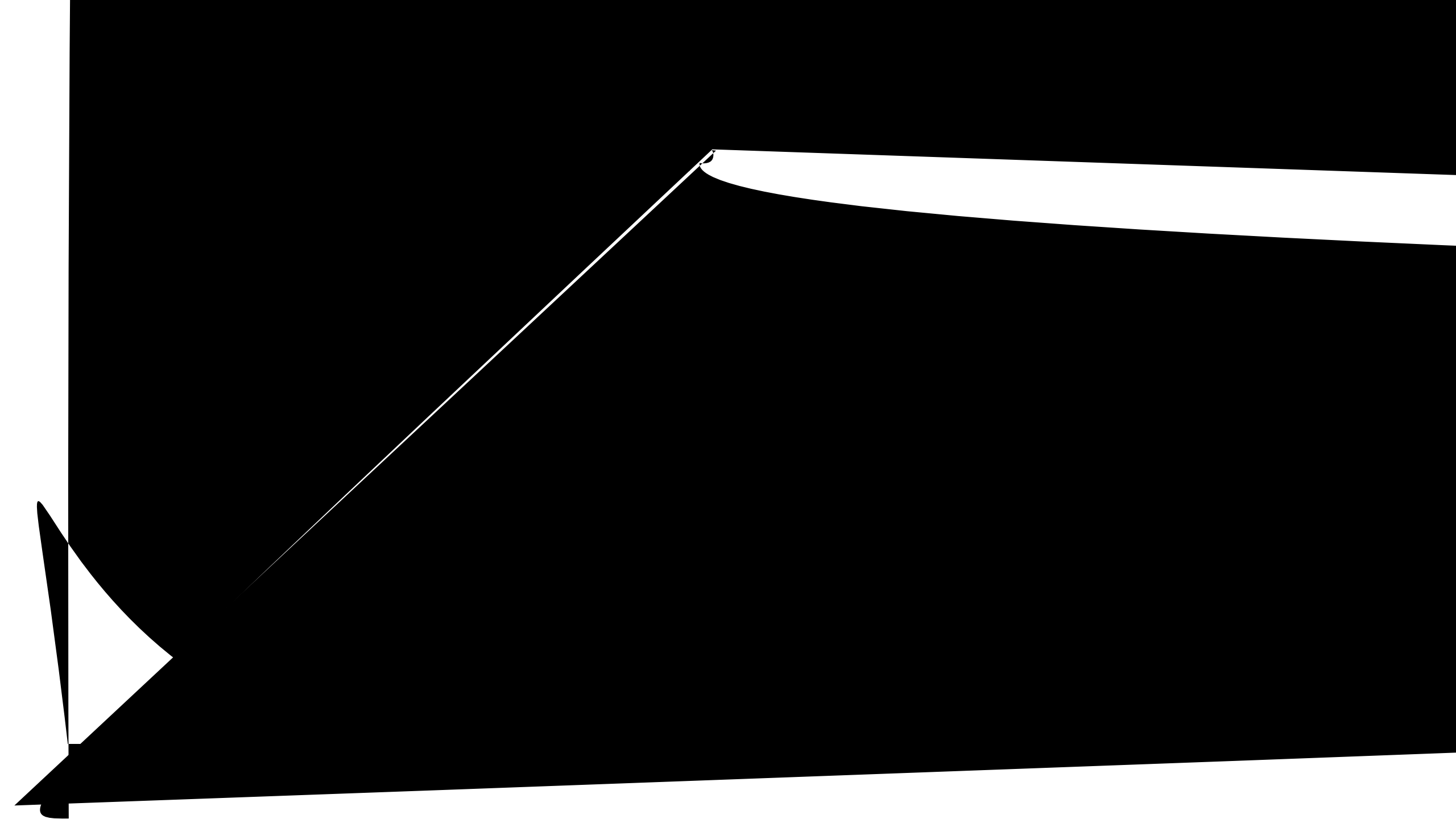
Q&A:

Chương trình cho kết quả gì khi gọi hàm:

func (a=1)

func (a=1, c=2)

func (a=1, c=2, d=3)



4. Đối số của hàm

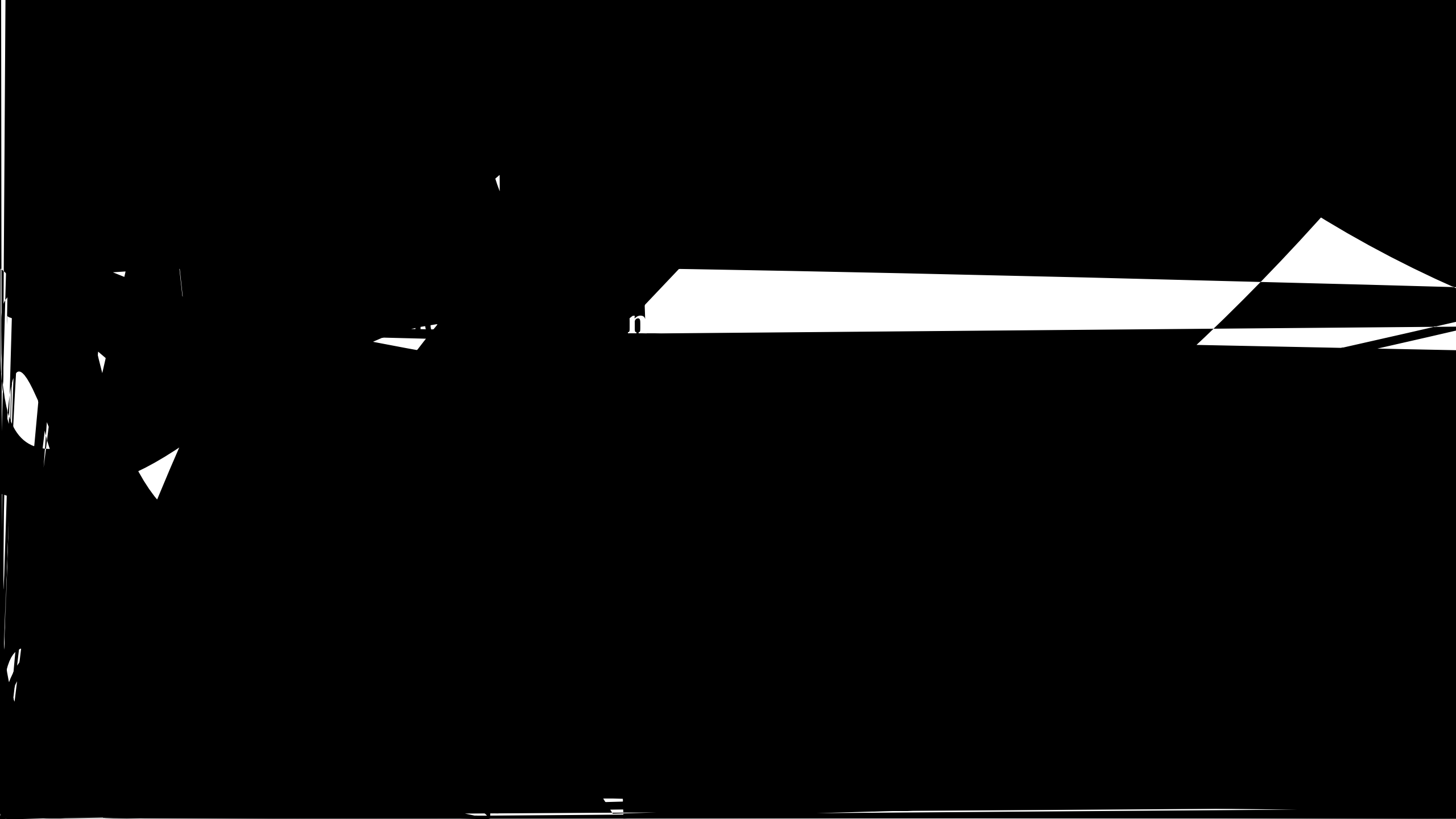
❖ Truyền đối số theo dạng từ điển (dictionary unpacking)

Lưu ý:

- Cú pháp: ****dictionary_name**
- Số lượng đối số và item của dictionary phải bằng nhau

4. Đối số của hàm

❖ Truyền đối số dạng



4. Đối số của hàm

❖ Truy

4. Đối số của hàm

❖ Truyền đối số dạng “hỗn hợp” (combining types)

❖ Thảo luận ví dụ sau:

```
# parameters.default.py  
def func(a, b=4, c=88):  
    print(a, b, c)
```

```
func(1)                # prints: 1 4 88  
func(b=5, a=7, c=9)    # prints: 7 5 9  
func(42, c=9)          # prints: 42 4 9  
func(42, 43, 44)       # prints: 42, 43, 44
```

5. Hàm có giá trị trả về (return)

❖ Return trong hàm của Python :

❖ Là 1 tùy chọn (optional)

✓ Hàm trả về None nếu không có return hoặc bare return

```
def moddiv(a, b):  
    return  
print(moddiv(20, 7))
```

Output:

None

5. Hàm có giá trị trả về (return)

❖ Return trong hàm của Python :

❖ Có thể trả về 1 hay nhiều object (số lượng/kiểu)

```
def moddiv(a, b):  
    return a//b, a%b  
print(moddiv (20,7))
```

Output:

(2, 6)

```
def moddiv(a, b):  
    return a/b, a+b  
print(moddiv (20,7))
```

Output:

(2.857142857142857, 27)

6. Hàm vô danh (lambdas function)

- ❖ Không cần đặt tên hàm, chỉ sử dụng 1 lần
- ❖ Nhanh, đơn giản, '1 dòng' (one-liner)
- ❖ Tuy nhiên khó debug và bảo trì
- ❖ Cú pháp

6. Hàm vô danh (lambdas function)

❖ Khác nhau giữa hàm “chính danh”



