

MÔN HỌC

LẬP TRÌNH CƠ BẢN



Giảng viên: Trần Bảo Trung

PYTHON
HÀM TRONG PYTHON
Part 2



MỤC TIÊU BÀI:



Cách hàm hoạt động trong Python



Cách khởi tạo và gọi hàm



Cơ chế truyền đổi số trong hàm



Cách nhận lại kết quả từ hàm



ĐỐI SỐ CÓ SỐ LƯỢNG KHÔNG XÁC ĐỊNH



ĐỐI SỐ CÓ SỐ LƯỢNG KHÔNG XÁC ĐỊNH



ĐỐI SỐ CÓ SỐ LƯỢNG KHÔNG XÁC ĐỊNH

Trong một số trường hợp khi định nghĩa hàm, chúng ta không xác định được liệu hàm này sẽ nhận vào bao nhiêu đối số



ĐỐI SỐ CÓ SỐ LƯỢNG KHÔNG XÁC ĐỊNH

Trong một số trường hợp khi định nghĩa hàm, chúng ta không xác định được liệu hàm này sẽ nhận vào bao nhiêu đối số

```
1  sum(1,2)
2  sum(1,2,3)
3  sum(1,2,3,4)
4  sum(1,2,3,4,5)
5  sum(1,2,3,4,5,6)
```



ĐỐI SỐ CÓ SỐ LƯỢNG KHÔNG XÁC ĐỊNH

Trong một số trường hợp khi định nghĩa hàm, chúng ta không xác định được liệu hàm này sẽ nhận vào bao nhiêu đối số

```
1  sum(1,2)
2  sum(1,2,3)
3  sum(1,2,3,4)
4  sum(1,2,3,4,5)
5  sum(1,2,3,4,5,6)
```

- Agrument Tuple packing
- Argument Dictionary packing



ĐỐI SỐ CÓ SỐ LƯỢNG KHÔNG XÁC ĐỊNH

Argument Tuple packing – đóng gói bộ tham số



ĐỐI SỐ CÓ SỐ LƯỢNG KHÔNG XÁC ĐỊNH

Argument Tuple packing – đóng gói bộ tham số

Khi tham số trong định nghĩa hàm được đặt sau dấu sao (*), nó biểu thị việc đóng gói bộ tham số.



ĐỐI SỐ CÓ SỐ LƯỢNG KHÔNG XÁC ĐỊNH

Argument Tuple packing – đóng gói bộ tham số

Khi tham số trong định nghĩa hàm được đặt sau dấu sao (*), nó biểu thị việc đóng gói bộ tham số.

Bất kỳ đối số tương ứng nào trong lệnh gọi hàm đều được đóng gói thành một tuple mà hàm có thể tham chiếu đến theo tên tham số đã khởi tạo



ĐỐI SỐ CÓ SỐ LƯỢNG KHÔNG XÁC ĐỊNH

Argument Tuple packing – đóng gói bộ tham số

```
1 def f(*args):  
2     print(args)  
3     print(type(args), len(args))  
4     for x in args:  
5         print(x)  
6  
7 f(1, 2, 3)
```

```
# (1, 2, 3)  
# <class 'tuple'> 3  
# 1  
# 2  
# 3
```



ĐỐI SỐ CÓ SỐ LƯỢNG KHÔNG XÁC ĐỊNH

Argument Dictionary packing – đóng gói từ điển tham số



ĐỐI SỐ CÓ SỐ LƯỢNG KHÔNG XÁC ĐỊNH

Argument Dictionary packing – đóng gói từ điển tham số

Khi sử dụng dấu `**` trước tham số trong khai báo hàm, nó sẽ biểu thị việc tiến hành đóng gói từ điển tham số



ĐỐI SỐ CÓ SỐ LƯỢNG KHÔNG XÁC ĐỊNH

Argument Dictionary packing – đóng gói từ điển tham số

Khi sử dụng dấu `**` trước tham số trong khai báo hàm, nó sẽ biểu thị việc tiến hành đóng gói từ điển tham số

Các đối số sử dụng khi gọi hàm được mong đợi là một cặp khóa – giá trị và sẽ được đóng gói dưới dạng từ điển tương ứng với tham số truyền vào.



ĐỐI SỐ CÓ SỐ LƯỢNG KHÔNG XÁC ĐỊNH

Argument Dictionary packing – đóng gói từ điển tham số

```
1 def f(**kwargs):
2     print(kwargs)
3     print(type(kwargs))
4     for key, val in kwargs.items():
5         print(key, '->', val)
6
7 f(foo=1, bar=2, baz=3)
```

```
# {'foo': 1, 'bar': 2, 'baz': 3}
# <class 'dict'>
# foo -> 1
# bar -> 2
# baz -> 3
```



ĐỐI SỐ CÓ SỐ LƯỢNG KHÔNG XÁC ĐỊNH

Lưu ý:

Khi sử dụng nhiều loại tham số khác nhau trong khai báo hàm, cần tuân thủ theo thứ tự ưu tiên

Tham số vị trí -> tham số mặc định -> đóng gói bộ tham số -> đóng gói từ điển tham số

Các đối số truyền vào cũng cần tuân thủ theo thứ tự này



ĐỐI SỐ CÓ SỐ LƯỢNG KHÔNG XÁC ĐỊNH

```
1 def f(a, b = 1, *args, **kwargs):  
2     print(F'a = {a}')
```

```
3     print(F'b = {b}')
```

```
4     print(F'args = {args}')
```

```
5     print(F'kwargs = {kwargs}')
```

```
6
```

```
7 f(1, 2, 'foo', 'bar', 'baz', 'qux', x=100, y=200, z=300)
```



DOCSTRING



DOCSTRING



DOCSTRING

Khi câu lệnh đầu tiên trong hàm là một chuỗi ký tự, nó được gọi là docstring của hàm. Docstring được sử dụng để cung cấp tài liệu cho một hàm.

Nó có thể chứa mục đích của hàm, các đối số mà hàm sử dụng, thông tin về giá trị trả về hoặc bất kỳ thông tin được cho là hữu ích.



DOCSTRING

```
1  def foo(bar=0, baz=1):
2      """Thực hiện in ra 2 giá trị ra màn hình.\n
3          Từ khóa tham số sử dụng: \n
4          bar -- giá trị thứ nhất (default=0)\n
5          baz -- giá trị thứ 2 (default=1)\n
6      """
7      print(f"bar: {bar}")
8      print(f"baz: {baz}")
9
10 foo(0,1)
```



FUNCTION ANNOTATIONS



FUNCTION ANNOTATIONS



FUNCTION ANNOTATIONS

Từ phiên bản 3.0, Python cung cấp một tính năng bổ sung được gọi là chú thích hàm.

Chú thích cung cấp một cách để đính kèm metadata vào các tham số của hàm và giá trị trả về.



FUNCTION ANNOTATIONS

Từ phiên bản 3.0, Python cung cấp một tính năng bổ sung được gọi là chú thích hàm.

Chú thích cung cấp một cách để đính kèm metadata vào các tham số của hàm và giá trị trả về.



FUNTION ANNOTATIONS

Từ phiên bản 3.0, Python cung cấp một tính năng bổ sung được gọi là chú thích hàm.

Chú thích cung cấp một cách để đính kèm metadata vào các tham số của hàm và giá trị trả về.

```
1  ✓ def f(a: int, b: str) -> float:
2      print(a, b)
3      return(3.5)
4
5  f(1,2)
```



CẢM ƠN ĐÃ LẮNG NGHE



Giảng viên: Trần Bảo Trung