



## Bài 6: Function (hàm)

---

TS. Trịnh Tuấn Đạt  
Viện CNTT-TT, ĐHBK Hà Nội



# Nội dung

---

1. Giới thiệu
2. Các loại tham số hàm trong Python
3. Biến cục bộ và biến toàn cục
4. Thay đổi giá trị tham số



# Nội dung

---

1. **Giới thiệu**
2. Các loại tham số hàm trong Python
3. Biến cục bộ và biến toàn cục
4. Thay đổi giá trị tham số



# Khái niệm về hàm

---

- Hàm là một khối mã nguồn (code) thực hiện một công việc nào đó. Hàm giúp tăng tính tái sử dụng.
- Có rất nhiều hàm được xây dựng sẵn trong Python, VD: `print()`, `len()`, `int()`, `float()`
- LTV có thể tự tạo các hàm, gọi là user-defined function (hàm tự định nghĩa)



# Định nghĩa hàm

---

- Từ khóa **def** được sử dụng để bắt đầu phần định nghĩa hàm. Theo sau là tên hàm, dấu đóng mở ngoặc đơn, và dấu hai chấm. Trong dấu ngoặc đơn là các tham số.
- Phần đầu tiên của hàm (docstring) là tùy chọn, dùng để mô tả chức năng của hàm. Tiếp theo là các lệnh trong thân hàm
- Lệnh cuối cùng của hàm là lệnh return, để trả về giá trị nào đó. Lệnh này là tùy chọn
- Cú pháp:

```
def ten_ham( cac_tham_so ):  
    "function_docstring"  
    # Cac lenh trong than ham  
    return [bieu_thuc]
```



# Gọi hàm

---

- Để thực thi một hàm, cần gọi hàm đó. Tên hàm và tham số truyền vào phải phù hợp. Cú pháp

`ten_ham(cac_tham_so)`

- Ví dụ

```
def myPrint(str):  
    "Ham in gia tri truyen vao"  
    print(str + "!!!")  
    return
```

```
myPrint("Hello")  
myPrint("Thank you")  
myPrint("Good bye")
```

```
Hello!!!  
Thank you!!!  
Good bye!!!
```



# Lệnh return

---

- Lệnh return sẽ kết thúc hàm, chuyển điều khiển về nơi gọi hàm. Lệnh return không đi kèm với biểu thức sẽ giống như return None

```
def nhan2So(a, b):  
    return a*b
```

```
n = nhan2So(3, 4)  
print(n)
```

12



# Lệnh return

---

```
def compare(a, b):  
    if a==b:  
        return "Bang nhau"  
    if a<b:  
        return "So 1 nho hon so 2"  
    return "So 2 nho hon so 1"
```

```
n = compare(3, 4)  
print(n)
```

So 1 nho hon so 2





# Lệnh return

---

```
def printAge(age):  
    "Ham in tuoi"  
    print("Ban " + str(age) + " tuoi")  
    if (age%12 == 0):  
        print("Nam nay la nam tuoi cua ban!")  
        return  
    print("Ban da trai qua " + str(age) + " cai Tet")  
  
print("Hello")  
printAge(36);  
print("Good bye!")
```

```
Hello  
Ban 36 tuoi  
Nam nay la nam tuoi cua ban!  
Good bye!
```



# Nội dung

---

1. Giới thiệu
2. **Các loại tham số hàm trong Python**
3. Biến cục bộ và biến toàn cục
4. Thay đổi giá trị tham số



## 2. Các loại tham số hàm trong python

---

Có 4 loại tham số hàm:

- Tham số bắt buộc
- Tham số từ khóa
- Tham số mặc định
- Tham số với số lượng tùy ý



# Tham số bắt buộc

---

- Tham số bắt buộc là các tham số được truyền tới một hàm theo đúng thứ tự.
- Số tham số trong lời gọi hàm phải khớp với phần định nghĩa hàm

```
def sum(a, b):  
    "Hàm có hai tham số"  
    c = a + b  
    print(c)
```

```
sum(10, 20) # a = 10, b = 20  
sum(20) # ERROR
```



# Tham số từ khóa

---

- Trong lời gọi hàm, dùng tên tham số khớp với tên dùng trong phần định nghĩa hàm.
- Vị trí các tham số là tùy ý (không cần đúng thứ tự)

```
def printinfo( name, age ):  
    "Ham in ra cac thong tin truyen vao"  
    print ("Name: ", name)  
    print ("Age ", age)  
    return
```

```
printinfo(age = 50, name = "Alice")  
printinfo(name = "Bob", age = 20)
```

```
Name:  Alice  
Age   50  
Name:  Bob  
Age   20
```



# Tham số mặc định

- Trong định nghĩa hàm, tạo giá trị mặc định cho các tham số
- Khi gọi hàm, có thể không cần truyền vào các tham số mặc định

```
def printinfo( name, age = 35 ):  
    "Ham in ra cac thong tin truyen vao"  
    print ( "Name: ", name)  
    print ( "Age ", age)  
    return
```

```
printinfo(age = 50, name = "Alice")  
printinfo(name = "Bob")  
printinfo("Peter")
```

```
Name: Alice  
Age  50  
Name: Bob  
Age  35  
Name: Peter  
Age  35
```

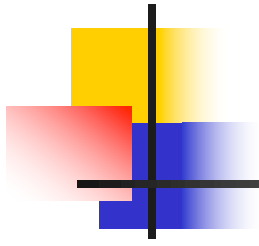


# Số lượng tham số tùy ý

---

- Ví dụ, cần tính tổng của các số nhập vào.
  - tong(1, 2)
  - tong(2, 3, 4)
  - tong(7, 5, 8, 9)
  - ...
- Cú pháp

```
def tenham([tham_so_chinh_thuc,] *tham_so_tuple ):
    "function_docstring"
    # Cac lenh trong tham ham
    return [bieu_thuc]
```



# Số lượng tham số tùy ý

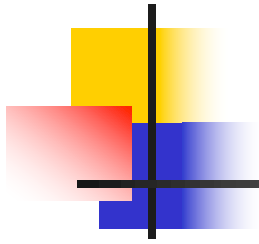
---

## ■ Ví dụ

```
def tong(*n):  
    res = 0  
    for i in n:  
        res += i  
    return res  
  
print(tong(4, 7, 3))  
print(tong(7, 7, 6))
```

14  
20





# Số lượng tham số tùy ý

---

```
def hello(className, *students):  
    print("Hello everyone in " + className)  
    for student in students:  
        print("Hello " + student)  
  
hello("Samsung Class", "Hung", "An", "Minh", "Nam")
```

```
Hello everyone in Samsung Class  
Hello Hung  
Hello An  
Hello Minh  
Hello Nam
```



# Nội dung

---

1. Giới thiệu
2. Các loại tham số hàm trong Python
3. **Biến cục bộ và biến toàn cục**
4. Thay đổi giá trị tham số



### 3. Biến biến cục bộ và biến toàn cục

---

- Các biến trong một chương trình không phải được truy cập tùy ý tại vị trí bất kỳ
- Tùy vào nơi khai báo biến, biến đó sẽ có phạm vi sử dụng khác nhau
- Có hai khái niệm về phạm vi biến:
  - Biến toàn cục
  - Biến cục bộ



# Biến cục bộ

---

- Các biến được khai báo bên trong một thân hàm là biến cục bộ. Các biến cục bộ này chỉ có thể được truy cập ở bên trong hàm đó, không thể được truy cập từ bên ngoài hàm
- Ví dụ

```
def msg():  
    a=10  
    print("Giá trị của a là", a)  
    return
```

```
msg()  
print a # ERROR
```



# Biến toàn cục

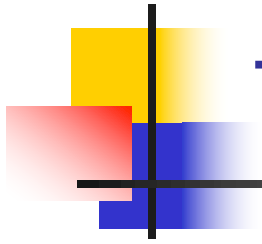
---

- Biến được định nghĩa bên ngoài hàm được gọi là biến toàn cục. Biến toàn cục có thể được truy cập bởi tất cả các hàm ở khắp nơi trong chương trình.
- Ví dụ:

```
b=20
def msg():
    a = 10
    print("Gia tri cua a la", a)
    print("Gia tri cua b la", b)
    return
```

```
msg()
print(b)
```

```
Gia tri cua a la 10
Gia tri cua b la 20
20
```



# Trùng tên Biến toàn cục và Biến cục bộ

- Lưu ý: trong thân hàm, khi dùng lệnh gán giá trị cho biến toàn cục, biến đó thực chất là 1 biến cục bộ trùng tên với biến toàn cục

```
b=20 # biến toàn cục
def msg():
    a = 10
    b = 30      # b là biến cục bộ

    print("Giá trị của a là", a)
    print("Giá trị của b là", b)
    return
```

```
msg()
print(b)
```

```
Gia trị của a là 10
Gia trị của b là 30
20
```



# Nội dung

---

1. Giới thiệu
2. Các loại tham số hàm trong Python
3. Biến cục bộ và biến toàn cục
4. **Thay đổi giá trị tham số**



## 4. Thay đổi giá trị tham số

---

### ■ Ví dụ

```
def changeValue(a):  
    a = 10
```

```
n = 5  
print(n)  
changeValue(n)  
print(n)
```

5  
5

```
def changeList(a):  
    a.append(10)
```

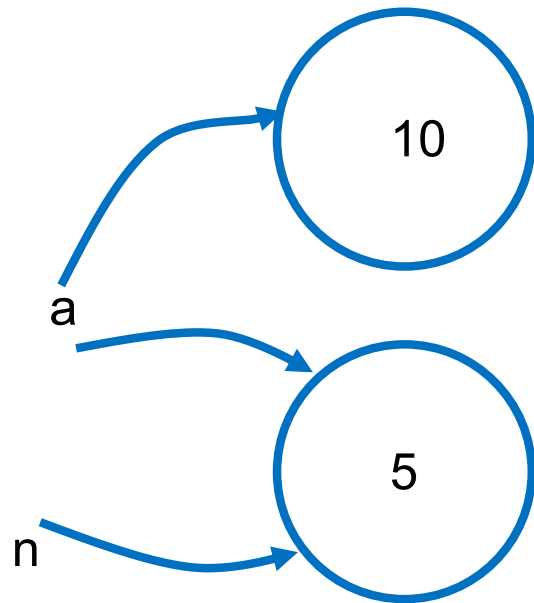
```
n = [2, 4, 6, 8]  
print(n)  
changeList(n)  
print(n)
```

[2, 4, 6, 8]  
[2, 4, 6, 8, 10]



## 4. Thay đổi giá trị tham số

- Python: tất cả là đối tượng
- Python: truyền bản sao tham chiếu đối tượng



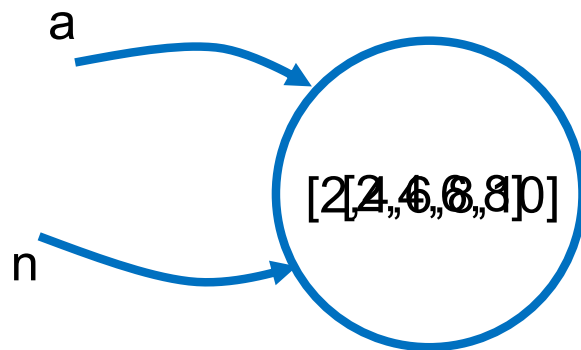
```
def changeValue(a):  
    a = 10
```

```
n = 5  
print(n)  
changeValue(n)  
print(n)
```

```
5  
5
```

## 4. Thay đổi giá trị tham số

- Python: tất cả là đối tượng
- Python: truyền bản sao tham chiếu đối tượng



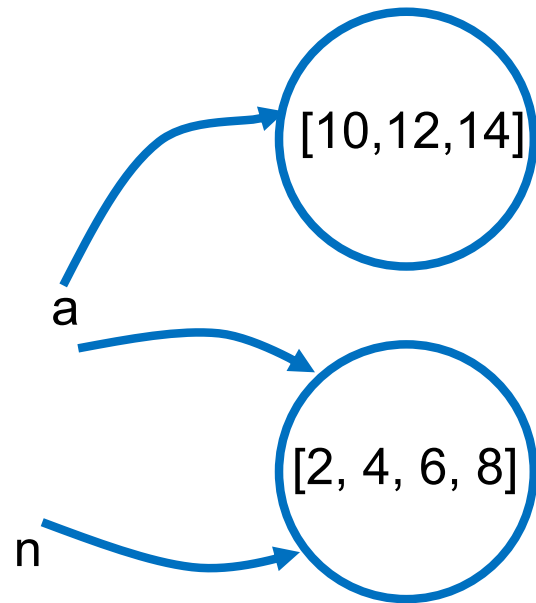
```
def changeList(a):  
    a.append(10)
```

```
n = [2, 4, 6, 8]  
print(n)  
changeList(n)  
print(n)
```

```
[2, 4, 6, 8]  
[2, 4, 6, 8, 10]
```

## 4. Thay đổi giá trị tham số

### ■ Ví dụ



```
def changeList(a):  
    a = [10, 12, 14]
```

```
n = [2, 4, 6, 8]  
print(n)  
changeList(n)  
print(n)
```

```
[2, 4, 6, 8]  
[2, 4, 6, 8]
```