



Lập trình Python

Bài 2: Hàm và rẽ nhánh trong python

Tài liệu này phân phối dưới giấy phép Creative Commons Attribution 4.0
(bất kỳ ai cũng đều có quyền tự do sử dụng, chia sẻ, sao chép, phân phối, phân phối lại, áp dụng, trích xuất, tùy biến, mở rộng, thương mại hóa,... miễn là ghi nhận công của các tác giả ban đầu của tài liệu)



Tóm tắt nội dung bài trước

- Hai cách thực thi python: chạy chương trình và dòng lệnh
- Dùng dấu thăng (#) để viết dòng chú thích
- Biến không cần khai báo trước, không cần chỉ kiểu
- Dữ liệu chuỗi nằm trong cặp nháy đơn ('), nháy kép ("), hoặc ba dấu nháy (""" / ''') – nếu viết nhiều dòng
 - Sử dụng chuỗi thoát (\) để khai báo các ký tự đặc biệt
 - Sử dụng chuỗi thô: r"nội dung"
- Hàm print để in dữ liệu, hàm input để nhập dữ liệu
 - Có thể kết hợp với hàm chuyển đổi kiểu
- Kiểu số và phép toán có một số điểm cần chú ý
 - Số nguyên không giới hạn độ lớn
 - Phép chia nguyên và phép chia chính xác



Nội dung

1. Hàm
2. Phép toán “if”
3. Rẽ nhánh
4. Bài tập



Phần 1

Hàm



Khai báo và gọi hàm

- Cú pháp khai báo hàm rất đơn giản

```
def <tên-hàm>(danh-sách-tham-số):  
    <lệnh 1>  
    ...  
    <lệnh n>
```

- Ví dụ: hàm tính tích 2 số

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

- Hàm trả về kết quả bằng lệnh **return**, nếu không trả về thì coi như trả về **None**

- Gọi hàm thông qua tên và các đối số

```
dt = tích(100, 200)  
s = tích(20, 30) + tích(40, 50)
```



Khai báo và gọi hàm

```
Function Name Parameters
↑      ↑      ↑
def add(num1, num2):
    print("Number 1:", num1)
    print("Number 2:", num1)
    addition = num1 + num2
    return addition → Return Value
} Function Body

res = add(2, 4) → Function call
print(res)
```



Hàm với tham số mặc định

- Hàm có thể chỉ ra giá trị mặc định của tham số

nếu không nói gì thì mặc định b = 1

```
def tinh(a, b = 1):  
    return a * b
```

- Như vậy với hàm trên ta có thể gọi thực hiện nó:

```
print(tinh(10, 20))      # 200  
print(tinh(10))          # 10  
print(tinh(a = 5))       # 5  
print(tinh(b = 6, a = 5)) # 30
```

- Chú ý: các tham số có giá trị mặc định phải đứng cuối danh sách tham số

Trả về kết quả từ hàm



Hàm không có kiểu, vì vậy có thể trả về bất kì loại dữ liệu gì, thậm chí có thể trả về nhiều kiểu dữ liệu khác nhau

```
def fuc1():  
    return 1001      # trả về một loại kết quả  
  
def fuc2():  
    print('None')    # không trả về kết quả  
  
def fuc3():  
    return 1001, 'abc', 4.5      # trả về phức hợp nhiều loại  
  
def fuc4(n):  
    if n < 0:  
        return 'số âm'          # trả về chuỗi  
    else:  
        return n + 1            # trả về số
```


Python không cho phép nạp chồng hàm



Python không cho phép hàm trùng tên, nếu cố ý định nghĩa nhiều hàm trùng tên, python sẽ sử dụng phiên bản cuối cùng

```
def abc():  
    return 'abc version 1'
```

```
def abc(a):  
    return 'abc version 2'
```

```
def abc(a, b):  
    return 'abc version 3'
```

```
print(abc())          # lỗi, hàm abc cần 2 tham số a và b
```

```
print(abc(1, 2))      # ok, in ra 'abc version 3'
```



Tham số tùy biến trong python

Python cho phép số lượng tham số tùy ý bằng cách đặt dấu sao (*) vào phía trước tên tham số.

Trong ví dụ dưới *names là một dãy không giới hạn số tham số

tham số tùy biến

```
def sayhello(*names):  
    # duyệt các tham số  
    for name in names:  
        print("Hello", name)
```

gọi hàm với 4 tham số

```
sayhello("Monica", "Luke", "Steve", "John")
```

gọi hàm với 3 tham số

```
sayhello("Aba", "Donald", "Pence")
```



Phần 2

Phép toán “if”

Phép toán “if”



```
if pw == '123':  
    x = 'yes'  
else:  
    x = 'no'
```



```
x = 'yes' if pw == '123' else 'no'
```



Phép toán “if”

```
# X là max của A và B
X = A if A > B else B

# N có phải là số nguyên tố có 1 chữ số hay không
A = "Đúng" if N in [2, 3, 5, 7] else "Sai"

# In ra màn hình “chẵn” nếu n chia hết cho 2,
#   in ra “lẻ” nếu ngược lại
print('chẵn' if (n % 2) == 0 else "lẻ")

# Sinh viên có được thi hay không?
print("được thi" if so_buoi_nghi < 3 else "không được thi")

# Biện luận nghiệm phương trình bậc 2 (if lồng nhau)
KQ = "một nghiệm" if delta == 0 else \
      "vô nghiệm" if delta < 0 else "hai nghiệm"
```



Phép toán “if”

- Cú pháp: `A if <điều-kiện> else B`
- Thực hiện:
 - Phép toán trả về A nếu <điều-kiện> là đúng, ngược lại trả về B
 - A và B có thể là các giá trị, biểu thức tính toán, lời gọi hàm,...
 - Các phép toán if cũng có thể lồng nhau
- Cách sử dụng `if` này khá kỳ lạ, nhưng hợp lý nếu xét về mặt ngôn ngữ và cách đọc điều kiện logic
- Ưu điểm: đây là phép toán, có thể viết trong biểu thức
- Bài tập: Biến X để lưu tình trạng gửi SMS, X=0 tức là chưa gửi được, X=1 tức là đã gửi thành công, X=2 tức là đã gửi và người nhận đã đọc. Viết câu lệnh sử dụng phép toán if để in ra màn hình thông báo tương ứng với giá trị của X.



Phần 3

Rẽ nhánh



Rẽ nhánh

```
if expression:
```

```
    # If-block
```

```
if expression:
```

```
    # If-block
```

```
else:
```

```
    # else-block
```

```
if expression:
```

```
    # If-block
```

```
elif 2-expression:
```

```
    # 2-if-block
```

```
elif 3-expression:
```

```
    # 3-if-block
```

```
...
```

```
elif n-expression:
```

```
    # n-if-block
```

```
if expression:
```

```
    # If-block
```

```
elif 2-expression:
```

```
    # 2-if-block
```

```
...
```

```
elif n-expression:
```

```
    # n-if-block
```

```
else:
```

```
    # else-block
```




Rẽ nhánh

```
# In thông báo nếu được điểm số loại giỏi
if diem >= 8: print("Chúc mừng bạn đã được điểm giỏi")
# In thông báo xem n chẵn hay lẻ
if (N % 2) == 0:
    print("N là số chẵn")
else:
    print("N là số lẻ")
# Biện luận nghiệm của phương trình bậc 2
if delta == 0:
    print("Phương trình có nghiệm kép")
elif delta < 0:
    print("Phương trình vô nghiệm")
else:
    print("Phương trình có hai nghiệm phân biệt")
```



- Python chỉ có một cấu trúc rẽ nhánh duy nhất, sử dụng để lựa chọn làm một trong số nhiều công việc
 - Nhiều ngôn ngữ lập trình khác sử dụng **if** cho trường hợp 2 lỗi rẽ nhánh và **switch** cho trường hợp nhiều lỗi rẽ nhánh
- Nguyên tắc với rẽ nhánh **if-elif-else**:
 - Biểu thức điều kiện của **if** và **elif** phải có kết quả logic
 - Hệ thống sẽ lần lượt tính giá trị từng biểu thức điều kiện từ trên xuống dưới, bắt đầu từ phát biểu **if**
 - Nếu biểu thức điều kiện nào đúng thì khối lệnh tương ứng được thực hiện và bỏ qua các khối lệnh khác
 - Trường hợp mọi biểu thức điều kiện đều sai, khối lệnh ứng với **else** sẽ được thực hiện
 - Khối **else** là tùy chọn, không nhất thiết phải xuất hiện



Phân tích ví dụ

```
a = int(input("A = "))
```

```
if 0 == a % 2:
```

```
    print("A là số chẵn")
```

```
elif 0 == a % 5:
```

```
    print("A chia hết cho 5")
```

```
else:
```

```
    print("A không có gì đặc biệt")
```

a = 11: in ra “A không có gì đặc biệt”

a = 12: in ra “A là số chẵn”

a = 15: in ra “A chia hết cho 5”

a = 10: in ra “A là số chẵn” – chú ý – A vừa là số chẵn vừa chia hết cho 5, nhưng lệnh dừng ngay khi xét điều kiện A chẵn.



Lệnh if lồng nhau, chú ý thụt lề

```
age = int(input("Bạn bao nhiêu tuổi? "))
print("Ồ bạn đã", age, "tuổi rồi!")
if age >= 18:
    print("Đủ tuổi đi bầu cử")
    if age > 100:
        print("Có vẻ sai sai!")
else:                                     # else thuộc về if ở dòng 3
    print("Nhỏ quá")
```

Như vậy trong python thì dấu cách cũng có vai trò lập trình của nó, không chỉ đơn giản là viết cho đẹp



Phần 4

Bài tập



Bài tập

1. Viết hàm `average` nhận 5 tham số và trả về giá trị trung bình cộng của chúng
2. Viết hàm `area` trả về diện tích tam giác khi biết độ dài ba cạnh của chúng
3. Viết hàm `area2` trả về diện tích tam giác biết tọa độ (x, y) của ba đỉnh tam giác
4. Viết hàm `total` nhận số nguyên N làm tham số, hàm trả về tổng các chữ số của số N (chẳng hạn N = 132 thì hàm `total` trả về 6)
5. Viết hàm `isFibo` nhận số nguyên N làm tham số, hàm kiểm tra xem N có phải số fibonacci hay không? Trả về `True` nếu đúng, ngược lại trả về `False`



6. Viết chương trình nhập số nguyên dương N và tính giá trị hàm $F(N)$ dưới đây

$$F(N) = 1! + 2! + \dots + N!$$

7. Viết chương trình nhập điểm trung bình học tập của một sinh viên, sau đó dựa trên quy tắc dưới đây in ra đánh giá kết quả học tập của sinh viên đó.

- Điểm dưới 3.5: xếp loại yếu
- Điểm từ 3.5 đến dưới 5: xếp loại kém
- Điểm từ 5 đến dưới 6.5: xếp loại trung bình
- Điểm từ 6.5 đến dưới 8: xếp loại khá
- Điểm từ 8 đến dưới 9: xếp loại giỏi
- Điểm từ 9 trở lên: xếp loại xuất sắc



8. Nhập vào từ bàn phím ba số a , b và c . Thực hiện các công việc dưới đây:

- In ra màn hình giá trị lớn nhất trong ba số
- Nếu có ít nhất hai số cùng nhận giá trị lớn nhất, in ra giá trị thứ ba còn lại

9. Nhập vào từ bàn phím thông tin về ngày X , bằng cách nhập ba số nguyên dương d , m và y lần lượt là giá trị ngày (d) tháng (m) năm (y) của X . Tính và in ra ngày tiếp sau của X

- Ví dụ: người dùng nhập $d = 31$, $m = 1$, $y = 2019$ (tức X là ngày 31 tháng 1 năm 2019) thì chương trình cần in ra 1/2/2019