CÁC LỆNH CƠ BẢN TRONG PYTHON

Nội dung

1. Lệnh rẽ nhánh
2. Lệnh lặp
3. Hàm trong python

Lệnh rẽ nhánh

Cấu trúc rẽ nhánh if-else

```
if expression:
    # If-block
```

```
if expression:
    # If-block
else:
    # else-block
```

Ví dụ:

```
X=10;
if(X==10)
      print(‘’ Ban chon dung”);
else
      print(‘’Ban chon sai”);
```

Lệnh if có thể lồng nhau

Phép toán if

- **Cú pháp:** **A if <điều kiện> else B**
- Nếu **<điều kiện>** trả về giá trị đúng thì phép toán trả về giá trị A,
- Nếu <**điều kiện> trả** về giá trị sai thì phép toán trả về giá trị B.

- **Ví dụ:** Tìm giá trị lớn nhất: A=3 và B=5

```
X = A if A>B else B
print ('Gia tri lon nhat la:', X)
```

Bài tập:

Nhập ba số a,b,c từ bàn phím và tìm số lớn nhất trong ba số trên.

File Edit Format Run Options Window Help

```
a = int(input("Nhap gia tri a: "))
b = int(input("Nhap gia tri b: "))
c = int(input("Nhap gia tri c: "))

max = a
if (max < b):
    max = b

if (max < c):
    max = c

print("==> Gia tri lon nhat la: ", max)
```

Lệnh lặp while

```
while expression:
    # while-block
```

```
while expression:
    #while-block-1
    continue
    #while-block-2
```

```
while expression:
    # while-block
else:
    # else-block
```

- Vòng lặp while trong python giống trong các ngôn ngữ khác
- Trong khối lệnh lặp nói chung có thể dùng **continue** hoặc **break** để về đầu hoặc cuối khối lệnh
- Khối **"else"** sẽ được thực hiện sau khi vòng lặp đã chạy xong
  - Khối này sẽ không chạy nếu vòng lặp bị **"break"**

Ví dụ vòng lặp while

- **while không có else**

```
count = 0
while (count <= 5):
  print (''Các số là:'', count)
  count = count + 1
```

- **while có else**

```
count = 0
while count < 10:
  print (count, " nho hon 10")  count = count + 1
else:
  print (count, " la khong nho hon 10")
```

Bài tập

Xuất các số chẵn từ 1 đến 100 dùng vòng lặp while

Hướng dẫn:

```
i = 0
while (i <= 100):
   print(i)
   i = i + 2
```

Hoặc:

```
i = 0
while (i <= 100):
   if i % 2 == 0:
      print(i)
   i = i + 1
```

Lệnh lặp for

```
for variable_1, variable_2, .. variable_n in sequence:
    # for-block
```

```
for variable_1, variable_2, .. variable_n in sequence:
    # for-block
else:
    # else-block
```

- Vòng lặp **for** sử dụng để duyệt danh sách, khối else làm việc tương tự như ở vòng lặp while
- Dùng **hàm range(a, b)** để tạo danh sách gồm các số từ **a** đến **b-1**.
- Tổng quát: **range(a, b, c)**, với c là bước nhảy

Ví dụ:

```
for i in range (0,10):
   print ('Số thứ tự là:',i)
```

```
for i in range(0,10):
    print ('Số thứ tự là:',i)
        else:
            print (‘Số cuối')
```

Bài tập

Xuất các số từ 1 đến 100 mà chia hết cho 3

Hướng dẫn:

```
for i in range(1, 100):
    if i % 3 == 0:
        print(i)
```

Hoặc:

```
i = 1
while i in range(1, 100):
    if i % 3 == 0:
        print(i)
    i += 1
```

Bài tập

Tìm tất cả các số từ 1 đến 100 sao cho: tích của 2 chữ số bằng 2 lần tổng hai chữ số đó.

Ví dụ: số 36 có: 3*6 = 2*(3+6)

Hướng dẫn:

```
for i in range(1, 9):
    for j in range(0, 9):
        if (i * j) == (2 * (i + j)):
            m = i * 10 + j
            print(m)
```

Hàm

- Cú pháp khai báo hàm:

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

- Ví dụ: hàm tính tích 2 số

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

- Hàm trả về kết quả bằng lệnh **return**, ngược lại trả về **None**

Bài tập:

Nhập hai số a, b từ bàn phím và tính tổng (dùng hàm)

Hướng dẫn:

```
def nhap(n):
        return n;
def tong(a,b):
    return a+b;
print("Nhập giá trị a: ");
        a=int(nhap(input()))
print("Nhập giá trị b: ");
        b=int(nhap(input()))
print("Tong ",a,"+ ",b,"=",tong(a,b));
```

Các hàm thông dụng

Hàm	Mô tả
abs()	Trả về giá trị tuyệt đối của một số
all()	Trả về True khi tất cả các phần tử trong iterable là đúng
any()	Kiểm tra bất kỳ phần tử nào của iterable là True
ascii()	Tả về string chứa đại diện (representation) có thể in
bin()	Chuyển đổi số nguyên sang chuỗi nhị phân
bool()	Chuyển một giá trị sang Boolean
bytearray()	Trả về mảng kích thước byte được cấp
bytes()	Trả về đối tượng byte không đổi
callable()	Kiểm tra xem đối tượng có thể gọi hay không
chr()	Trả về một ký tự (một chuỗi) từ Integer
classmethod()	Trả về một class method cho hàm
compile()	Trả về đối tượng code Python
complex()	Tạo một số phức
delattr()	Xóa thuộc tính khỏi đối tượng
dict()	Tạo Dictionary
dir()	Trả lại thuộc tính của đối tượng
divmod()	Trả về một Tuple của Quotient và Remainder

Hàm	Mô tả
enumerate()	Trả về đối tượng kê khai
eval()	Chạy code Python trong chương trình
exec()	Thực thi chương trình được tạo động
filter()	Xây dựng iterator từ các phần tử True
float()	Trả về số thập phân từ số, chuỗi
format()	Trả về representation được định dạng của giá trị
frozenset()	Trả về đối tượng frozenset không thay đổi
getattr()	Trả về giá trị thuộc tính được đặt tên của đối tượng
globals()	Trả về dictionary của bảng sumbol toàn cục hiện tại
hasattr()	Trả về đối tượng dù có thuộc tính được đặt tên hay không
hash()	Trả về giá trị hash của đối tượng
help()	Gọi Help System được tích hợp sẵn
hex()	Chuyển Integer thành Hexadecimal
id()	Trả về định danh của đối tượng
input()	Đọc và trả về chuỗi trong một dòng
int()	Trả về số nguyên từ số hoặc chuỗi

Hàm	Mô tả
isinstance()	Kiểm tra xem đối tượng có là Instance của Class không
issubclass()	Kiểm tra xem đối tượng có là Subclass của Class không
iter()	Trả về iterator cho đối tượng
len()	Trả về độ dài của đối tượng
list()	Tạo list trong Python
locals()	Trả về dictionary của bảng sumbol cục bộ hiện tại
map()	Áp dụng hàm và trả về một list
max()	Trả về phần tử lớn nhất
memoryview()	Trả về chế độ xem bộ nhớ của đối số
min()	Trả về phần tử nhỏ nhất
next()	Trích xuất phần tử tiếp theo từ Iterator
object()	Tạo một đối tượng không có tính năng (Featureless Object)
oct()	Chuyển số nguyên sang bát phân
open()	Trả về đối tượng File

Hàm	Mô tả
ord()	Trả về mã Unicode code cho ký tự Unicode
pow()	Trả về x mũ y
print()	In đối tượng được cung cấp
property()	Trả về thuộc tính property
range()	Trả về chuỗi số nguyên từ số bắt đầu đến số kết thúc
repr()	Trả về representation có thể in của đối tượng
reversed()	Trả về iterator đảo ngược của một dãy
round()	Làm tròn số thập phân
set()	Tạo một set các phần tử mới
setattr()	Đặt giá trị cho một thuộc tính của đối tượng
slice()	Cắt đối tượng được chỉ định bằng range()
sorted()	Trả về list được sắp xếp
staticmethod()	Tạo static method từ một hàm
str()	Chuyển đối tượng đã cho thành chuỗi
sum()	Thêm một mục vào Iterable
super()	Cho phép tham chiếu đến Parent Class bằng super

Hàm	Mô tả
tuple()	Tạo một Tuple
type()	Trả về kiểu đối tượng
vars()	Trả về thuộc tính __dict__ của class
zip()	Trả về Iterator của Tuple
import()	Hàm nâng cao, được gọi bằng import