

LẬP TRÌNH PYTHON

CẤU TRÚC ĐIỀU KHIỂN

NGUYỄN HẢI TRIỀU¹

¹Bộ môn Kỹ thuật phần mềm,
Khoa Công nghệ thông tin, Trường ĐH Nha Trang

NhaTrang, February 2022

Cấu trúc điều khiển

- 1 Câu lệnh điều kiện
- 2 Câu lệnh lặp

1 Câu lệnh điều kiện

2 Câu lệnh lặp

Các điều kiện kiểm tra

Các điều kiện logic

Cũng giống như những ngôn ngữ khác, Python có các toán tử so sánh thông dụng như sau

- Bằng: $a == b$
- Không bằng: $a != b$
- Nhỏ hơn: $a < b$
- Nhỏ hơn hoặc bằng: $a \leq b$
- Lớn hơn: $a > b$
- Lớn hơn hoặc bằng: $a \geq b$

Chúng ta có thể kết hợp các biểu thức điều kiện với nhau bằng cách sử dụng các toán tử logic **and**, **or**, **not**...

Câu lệnh if...else

Cú pháp câu lệnh if...else

if <điều kiện>:

câu lệnh khi điều kiện đúng

else:

câu lệnh khi điều kiện sai

Lưu ý

Python dựa vào thụt lề (tiêu chuẩn 4 khoảng trắng) để xác định khối lệnh sau dấu :

```
a=0
if (a==0):
    print(a)
```

⊗ 0.8s

```
Input In [5]
    print(a)
    ^
```

IndentationError: expected an indented block

Câu lệnh if...elif...else

Cú pháp elif

Trong trường hợp giải quyết bài toán có nhiều hơn hai trường hợp, chúng ta có thể sử dụng từ khóa **elif** thay cho cho nhóm lệnh **if** đặt bên trong **else**. Cụ thể:

if <điều kiện 1>:

câu lệnh khi điều kiện 1 đúng

elif <điều kiện 2>:

câu lệnh khi điều kiện 2 đúng

...

elif <điều kiện n>:

câu lệnh khi điều kiện n đúng

else:

câu lệnh khi không có trường hợp nào ở trên là đúng

Câu lệnh if...elif...else

Chú ý

Để ngắn gọn, ta có thể viết được nhiều câu lệnh **if...else** trên một dòng.

```
1 a=0; b=2
2 print("A") if a>b else print("B") if b>a else print("a=b")
3 #->B
```

Câu lệnh if không được để trống, nếu không muốn xuất hiện lỗi thì sử dụng câu lệnh **pass**

```
1 a=2; b=2
2 if(a==b):
3     pass
```

Câu lệnh if...elif...else

Python không có cấu trúc điều khiển **switch...case**

Để mô phỏng cấu trúc **switch...case** trong Python, có thể sử dụng dictionary hoặc class

```
1 def text_of_number(n):  
2     dic_of_number={  
3         1:'Mot', 2:'Hai', 3:'Ba', 4:'Bon', 5:'Nam'  
4     }  
5     return dic_of_number.get(n,'Khong doc duoc')  
6 text_of_number(1)
```


Câu lệnh if...elif...else

Ví dụ 1.1

- ❶ *Viết chương trình xuất ra màn hình trị tuyệt đối của một số nguyên được nhập từ bàn phím (**không được sử dụng hàm `abs`**)*
- ❷ *Viết chương trình nhập số nguyên $0 < n < 11$ từ bàn phím và xuất ra số la mã tương ứng.*
- ❸ *Viết chương trình nhập một kí tự trong bảng chữ cái latin, kiểm tra xem kí tự nhập vào có phải là nguyên âm hay không?*

1 Câu lệnh điều kiện

2 Câu lệnh lặp

Vòng lặp while

Cú pháp 2.1

Sử dụng vòng lặp while với số lần lặp không biết trước.

while <điều_kiện> :

các câu lệnh với điều_kiện==True

Cú pháp 2.2

*Mặc định Python không tồn tại vòng lặp **do...while** như các ngôn ngữ khác. Nhưng chúng ta có thể sử dụng vòng lặp while với biểu thức điều kiện bằng **True** và điều kiện dừng với câu lệnh **break***

while True :

thực thi các câu lệnh

if <điều kiện dừng thỏa mãn> :

break

Vòng lặp while

Ví dụ 2.1

Viết chương trình yêu cầu người dùng nhập vào một số nguyên n thỏa điều kiện $0 \leq n \leq 10$. In ra màn hình thông báo “nhập vào hợp lệ” nếu thỏa yêu cầu đề bài.

Vòng lặp while

Ví dụ 2.1

Viết chương trình yêu cầu người dùng nhập vào một số nguyên n thỏa điều kiện $0 \leq n \leq 10$. In ra màn hình thông báo “nhập vào hợp lệ” nếu thỏa yêu cầu đề bài.

```
1 while True:
2     n=int(input('nhập số nguyên n: '))
3     if 0<=n<=10:
4         print('nhập vào hợp lệ')
5         break
```

Ví dụ 2.2

Viết chương trình tìm số nguyên n nhỏ nhất sao cho $\frac{1}{n}$ nhỏ hơn số thực a do người dùng nhập vào.

Vòng lặp while

Câu lệnh continue

Với câu lệnh **continue**, chúng ta có thể dừng lần lặp hiện tại và tiếp tục với vòng lặp tiếp theo.

```
1 i = 0
2 while i < 4:
3     i += 1
4     if i == 2:
5         continue
6     print(i) #->1 3 4
```

Vòng lặp while

Câu lệnh **while...else**

Không những đi kèm với câu lệnh **if** mà câu lệnh **else** còn có thể dùng chung với câu lệnh **while**.

```
1 i = 0
2 while i < 4:
3     print(i)
4     i+=1
5 else:
6     print('lệnh while đã kết thúc bởi điều kiện dừng')
```

Các câu lệnh sau **else** chỉ được thực thi sau khi lệnh **while** bị kết thúc bởi điều kiện dừng.

Vòng lặp for

Cú pháp 2.3

Với số lần lặp biết trước, ta sử dụng câu lệnh lặp **for** để lặp qua một dãy cho trước.

for <biến chạy> **in** <dãy> :

<các câu lệnh>

Trong đó, <dãy> có thể là list, tuple, dictionary, set, string ...

<biến chạy> sẽ lần lượt nhận các giá trị của thành phần có trong dãy.

```
1 fruits = ["apple", "banana", "cherry"]
2 for x in fruits:
3     print(x) #-> apple banana cherry
```


Vòng lặp for với chuỗi kí tự

<biến chạy> sẽ lần lượt nhận các giá trị là các kí tự trong chuỗi kí tự.

```
1 for x in "cntt":  
2     print(x) #->c n t t
```

Vòng lặp for với danh sách

<biến chạy> sẽ lần lượt nhận các giá trị là các phần tử trong danh sách.

```
1 l=[2,3,1,4]  
2 s=0  
3 for i in l:  
4     s+=i  
5 #->s=10
```

Vòng lặp for với từ điển

- một <biến chạy> sẽ lần lượt nhận các cặp giá trị của từ điển.

```
1 d={1:'mot', 2:'hai', 3:'ba'}
2 for item in d.items():
3     print(item)
```

- hai <biến chạy> tương ứng với các key và value riêng của các cặp trong từ điển

```
1 d={1:'mot', 2:'hai', 3:'ba'}
2 for key, value in d.items():
3     print(key, value)
```

Hàm range() trong vòng lặp for

Với hàm range(), chúng ta có thể lặp trong một vùng được xác định trước.

- hàm range(n) với một tham số truyền vào sẽ trả về một dãy số bắt đầu từ 0 và tăng lên 1 (theo mặc định), sau đó kết thúc tại $n - 1$.

```
1 for x in range(4):  
2     print(x) #-> 0 1 2 3
```

- hàm range(start, end, step) với tham số bắt đầu start, kết thúc end và bước nhảy step (để trống thì mặc định là 1)





```
1 for x in range(2, 6):  
2     print(x) #-> 2 3 4 5  
3 for x in range(2, 8, 2):  
4     print(x) #-> 2 4 6
```

Lệnh else trong vòng lặp for

Giống như vòng lặp while, từ khóa **else** trong vòng lặp for chỉ định một khối mã sẽ được thực thi khi vòng lặp for kết thúc.

```
1 for x in range(2):
2     print(x)
3 else:
4     print("Thuc thi lenh sau khi ket thuc vong lap!")
5     """
6     0
7     1
8     Thuc thi lenh sau khi ket thuc vong lap!
9     """
```

Tài liệu tham khảo

-  V.H. Quân, C.X. Nam, H.T. Hiếu, N.H. Triều, V.C. Tài
Tự học lập trình Python căn bản. *NXB DHQG Tp.HCM, 2019.*
-  Mark Lutz
Learning Python (5th Edition). *O'Reilly Media, Inc, 2013.*
-  Luciano Ramalho
Fluent Python (2nd Edition). *O'Reilly Media, Inc, 2021.*
-  Python Software Foundation
<https://docs.python.org/3/tutorial/>