



BÀI 4

Kiểu dữ liệu range và Vòng lặp While, for

Tóm Tắt Nội Dung

Trong bài học này chúng ta sẽ đi tìm hiểu lần lượt các nội dung

- 1 Tìm hiểu về lệnh input()
- 2 Tìm hiểu về toán tử Logical trong Python
- 3 Tìm hiểu về câu lệnh điều kiện rẽ nhánh if else
- 4 Tìm hiểu về vòng lặp while , for trong Python

3.4 Kiểu dữ liệu range



Câu lệnh range() trong Python

Hàm **range()** trả về một dãy số, bắt đầu từ 0 theo mặc định và tăng dần 1 (theo mặc định) và dừng trước một số đã chỉ định.

range([start], stop, [step])

start: là vị trí bắt đầu (Tùy chọn, mặc định là 0)

stop: vị trí kết thúc (Bắt buộc, và không bao gồm giá trị stop)

Step: bước nhảy (Tùy chọn, mặc định là 1, có thể số số nguyên âm)

```
x = range(3, 6)
for i in x :
    print(i)
#output:
3
4
5
```



Hàm **range(3,6)** cho ra một danh sách các số: 3, 4, 5 (stop -1)

Vòng lặp for, lặp qua danh sách đó, và in từng phần tử của danh sách số.

3.4 Vòng lặp trong Python



Câu lệnh range() trong Python

Một số trường hợp khi sử dụng Hàm **range()** trả về một dãy số:

Cú pháp	Ý nghĩa
range(0,n)	Cho ra dãy số từ 0,1...n-1
range(1,n,2)	Cho ra dãy số từ 0,3,5,7...n-1 (tăng dần 2 đơn vị)
range(n)	Cho ra dãy số từ 0,1...n-1 (lúc này n là stop, start là 0)
range(n,0,-1)	Cho ra dãy số n, n-1...1 (chiều giảm dần)

Chúng ta sẽ tìm hiểu range rõ hơn qua một vài ví dụ về vòng lặp **for** sau đây

3.4 Vòng lặp trong Python



Thế nào là vòng lặp ?

Ví dụ trong đời sống có các phát biểu sau:

- 1 Bạn hãy nhảy lò cò **2 lần**
- 2 Chạy **5 lần** quanh sân bóng
- 3 Nói thật to **10 lần** 2 từ “Xin chào”

Thì có nghĩa bạn đang **lặp đi lặp lại** một hoặc nhiều hành động gì đó.

Thì trong ngôn ngữ lập trình, việc lặp lại đó được viết thành câu lệnh như là **for, while**

Vòng lặp là một khái niệm trong lập trình được sử dụng để lặp lại các hành động nhiều lần, mà không cần phải viết lại các lệnh nhiều lần.

3.4 Vòng lặp trong Python



Thế nào là vòng lặp ?

3

Nói thật to **10 lần** 2 từ “Xin chào”

Chúng ta có thể mô tả phát biểu này trong lập trình Python như sau:

```
print("Xin chào")  
print("Xin chào")  
print("Xin chào")  
print("Xin chào")  
print("Xin chào")  
print("Xin chào")  
print("Xin chào")  
print("Xin chào")  
print("Xin chào")  
print("Xin chào")
```

```
for i in range(1,11) :  
    print("Xin chào", i)
```

Thay vì dùng 10 lần lệnh print() chúng ta có thể sử dụng 1 lệnh print() kết hợp với vòng lặp for như trên

Code vì thế trở nên ngắn gọn hơn

3.4 Vòng lặp trong Python



Vòng lặp for

Vòng lặp "for" được sử dụng để **lặp** qua một tập hợp các phần tử (như danh sách, chuỗi, tuple) và thực thi một khối mã cho mỗi phần tử trong tập hợp đó. Cú pháp của vòng lặp "for" như sau:

for **phần_tử_lặp_qua** **in** **tập_hợp**:
 # Các câu lệnh thực thi cho mỗi phần tử

Phần_tử_lặp_qua: là biến lưu trữ giá trị của phần tử trong mỗi lần lặp

Tập_hợp: là một tập hợp có thể lặp qua, chẳng hạn như danh sách, chuỗi, hoặc tuple

```
x = range(1,6)
for i in x:
    print(i)
```

3.4 Vòng lặp trong Python



Ví dụ về Vòng lặp for

Sử dụng với hàm **range()**

```
for i in range(1, 5):  
    print(i)
```

Lặp qua một string

```
for x in "banana":  
    print(x)
```


3.4 Vòng lặp trong Python



Vòng lặp while

Vòng lặp "while" được sử dụng để lặp lại một khối mã miễn là điều kiện **đúng** (True). Cú pháp của vòng lặp "while" như sau:

while <điều kiện kiểm tra>:
Các câu lệnh thực thi khi điều kiện đúng

"**condition**" là một biểu thức hoặc một điều kiện logic. Vòng lặp sẽ tiếp tục lặp lại cho đến khi điều kiện này trở thành sai (False).

```
count = 0
while count < 5:
    print(count)
    count += 1
```

vòng lặp "while" được sử dụng để lặp lại việc in ra giá trị của biến "count" từ 0 đến 4. Mỗi lần lặp, giá trị của "count" được tăng lên 1 và điều kiện "count < 5" được kiểm tra

3.4 Vòng lặp trong Python



Điều kiện dừng vòng lặp

Trong vòng lặp "while", điều kiện đúng (True) được kiểm tra ở đầu mỗi lần lặp. Nếu điều kiện đúng từ đầu, vòng lặp sẽ không được thực thi. Điều này có thể gây ra vòng lặp vô hạn nếu không có một cơ chế nào đảm bảo điều kiện sẽ trở thành sai (False) tại một thời điểm nào đó

```
count = 5
while count > 0:
    print(count)
```

Vòng lặp sẽ không bao giờ kết thúc vì điều kiện "count > 0" luôn đúng. **Để tránh vòng lặp vô hạn**, chúng ta cần đảm bảo rằng điều kiện cuối cùng sẽ trở thành **sai** tại một điểm nào đó trong vòng lặp

3.4 Vòng lặp trong Python



Vòng lặp với câu lệnh **break** và **continue**

Câu lệnh **break** dừng vòng lặp và bỏ qua câu lệnh phía sau nó

```
fruits = ["apple", "banana", "cherry"]
for x in fruits:
    if x == "banana":
        break
    print(x)
```

*Trong ví dụ này. Nếu vòng lặp duyệt qua đến phần tử x. Nếu x bằng "banana" thì câu lệnh **break** được thực thi, thoát khỏi vòng lặp*

Câu lệnh **continue** tiếp tục vòng lặp và bỏ qua câu lệnh phía sau nó

```
fruits = ["apple", "banana", "cherry"]
for x in fruits:
    if x == "banana":
        continue
    print(x)
```

*Trong ví dụ này. Nếu vòng lặp duyệt qua đến phần tử x. Nếu x bằng "banana" thì câu lệnh **continue** được thực thi, lặp tiếp phần tử tiếp theo.*

3.4 Vòng lặp trong Python



Nguồn tham chiếu

- ◆ Kiểu Dữ liệu Range

https://www.w3schools.com/python/ref_func_range.asp

- ◆ Vòng lặp for

https://www.w3schools.com/python/python_for_loops.asp

- ◆ Vòng lặp While

https://www.w3schools.com/python/python_while_loops.asp

Tổng kết lại bài 3

- 1 Nắm được cách sử dụng range()
- 2 Nắm được cách sử dụng vòng lặp while trong Python
- 3 Nắm được cách sử dụng vòng lặp for trong Python