

## VÒNG LẶP FOR

- Vòng lặp **for** được sử dụng để lặp qua một chuỗi (có thể là list, tuple, dictionary, set hoặc string).
- Với vòng lặp **for**, chúng ta có thể thực thi một tập hợp các câu lệnh, một lần cho mỗi mục trong danh sách, bộ dữ liệu, bộ, v.v.

**Ví dụ:** In ra các phần tử của một đối tượng list:

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

```
apple
banana
cherry
```

*Vòng lặp for không yêu cầu phải đặt biến lặp chỉ mục trước.*

### 1. Lặp qua một chuỗi

Như các bài trước, chuỗi cũng là một đối tượng có thể lặp, nó chứa một chuỗi các ký tự:

**Ví dụ:** Lặp qua các ký tự trong chuỗi "banana":

```
In [ ]: for x in "banana":
        print(x)
```

```
b
a
n
a
n
a
```

### 2. Câu lệnh break

Với câu lệnh **break**, bạn có dừng vòng lặp trước khi nó lặp qua tất cả các phần tử

**Ví dụ:** Thoát khỏi vòng lặp khi x là "banana":

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

```
apple
banana
```

**Ví dụ:** Thoát khỏi vòng lặp khi x là "banana", nhưng câu lệnh **break** đặt trước hàm **print()**

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

```
break
print(x)
```

apple

### 3. Câu lệnh continue

Với câu lệnh *continue*, chúng ta có thể dừng lần lặp hiện tại của vòng lặp và tiếp tục lần lặp kế tiếp

**Ví dụ:** Lặp qua các phần tử của list nhưng không in ra phần tử là "banana"

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

apple

cherry

### 4. Hàm range()

- Để lặp qua một bộ mã một số lần xác định, chúng ta có thể sử dụng hàm **range()**.
- Hàm **range()** trả về một dãy số, bắt đầu từ 0 theo mặc định và tăng dần theo 1 (theo mặc định) và kết thúc tại một số đã chỉ định.

**Ví dụ:** Dùng hàm **range()** để in ra các giá trị từ 0 đến 5

```
In [ ]: for x in range(6):
        print(x)
```

0  
1  
2  
3  
4  
5

Lưu ý: **range(6)** không phải là các giá trị từ 0 đến 6, mà là các giá trị từ 0 đến 5.

Hàm **range()** mặc định là 0 làm giá trị bắt đầu, tuy nhiên có thể chỉ định giá trị bắt đầu bằng cách thêm một tham số: `range(2, 6)`, có nghĩa là các giá trị từ 2 đến 6 (nhưng không bao gồm 6)

**Ví dụ:** Dùng tham số start của hàm **range()** để in ra các giá trị từ 2 đến 6 nhưng không bao gồm 6:

```
In [ ]: for x in range(2, 6):
        print(x)
```

2  
3  
4  
5

Hàm **range()** mặc định tăng chuỗi lên 1, tuy nhiên, có thể chỉ định bước nhảy bằng cách thêm tham số thứ ba: **range(2,30,3)**

```
In [ ]: for x in range(2, 30, 3):  
        print(x)
```

```
2  
5  
8  
11  
14  
17  
20  
23  
26  
29
```

## 5. Dừng câu lệnh else trong vòng lặp

Từ khóa **else** trong một vòng lặp chỉ định một khối mã sẽ được thực thi khi vòng lặp kết thúc

**Ví dụ:** In tất cả các số từ 0 đến 5, và in ra thông báo khi vòng lặp kết thúc:

```
In [ ]: for x in range(6):  
        print(x)  
        else:  
            print("Finally finished!")
```

```
0  
1  
2  
3  
4  
5  
Finally finished!
```

Lưu ý: Khối **else** sẽ **KHÔNG** được thực thi nếu vòng lặp bị dừng bởi một câu lệnh **break**

```
In [ ]: for x in range(6):  
        if x == 3: break  
        print(x)  
        else:  
            print("Finally finished!")
```

```
0  
1  
2
```

## 6. Vòng Lặp lồng Nhau

- Một vòng lặp lồng nhau là một vòng lặp bên trong một vòng lặp.
- "Vòng lặp bên trong" sẽ được thực hiện một lần cho mỗi lần lặp lại của "vòng lặp bên ngoài"

**Ví dụ:** In ra mỗi tính từ đối với từng loại quả:

```
In [ ]: adj = ["red", "big", "tasty"]  
        fruits = ["apple", "banana", "cherry"]  
  
        for x in adj:
```

```
for y in fruits:  
    print("{0} - {1}".format(x, y))
```

```
red - apple  
red - banana  
red - cherry  
big - apple  
big - banana  
big - cherry  
tasty - apple  
tasty - banana  
tasty - cherry
```

## 7. Câu lệnh pass

Vòng lặp *for* không thể để trống, nhưng nếu vì lý do nào đó bạn có một vòng lặp không có nội dung, hãy đặt câu lệnh *pass* vào để tránh gặp lỗi.

### Ví dụ:

```
In [ ]: for x in [0, 1, 2]:  
        pass
```

---

Biên dịch: Nguyễn Ngọc Dương

Facebook: <https://www.facebook.com/danh.kean.353803/>

Email: [ngocduong5642@gmail.com](mailto:ngocduong5642@gmail.com)