

# LOOP

## 1 Loop là gì?

Loop (vòng lặp) = làm đi làm lại một việc nhiều lần.

Ví dụ:

- Con nói: "Con thích toán"
- Cô giáo bắt nói 5 lần.

Thay vì viết lại cụm "Con thích toán" 5 lần thì ta dùng loop để bảo máy tính: 'Làm câu này 5 lần đi'. Đó chính là **vòng lặp**.

Ta cần loop vì máy tính rất thích làm việc lặp lại.

Ví dụ:

- In số từ 1 đến 1000
- Tính tổng  $1 + 2 + 3 + \dots + 10000$
- Kiểm tra từng phần tử trong một mảng
- Duyệt qua từng ký tự trong chuỗi

Nếu không có loop → không thể làm việc hiệu quả.

## 2 Các dạng loop

### 2.1 For loop

Ta dùng For loop khi biết trước số lần lặp. Giống như ví dụ ở trên, ta biết trước số lần cần lặp là 5, do đó For loop phù hợp với yêu cầu này.

```
1  for i in range(1, 6):  
2      print("Con\u00e1thich\u00e1toan")
```

Chương trình sẽ hiểu câu lệnh như sau: bắt đầu từ 1 → kết thúc trước 6.

## 2.2 While loop

While nghĩa là "trong khi", do đó sẽ lặp trong khi điều kiện còn đúng. Ví dụ: Trong khi còn tiền thì còn mua kẹo.

```
1     money = 5
2     while money > 0:
3         print("Mua kẹo")
4         money -= 1
```

Nó sẽ kiểm tra: Nếu điều kiện đúng thì làm, điều kiện sai thì dừng.

## 2.3 Do-While

Khác với while kiểm tra trước rồi mới làm, do-while làm trước rồi mới kiểm tra. Nhưng cần lưu ý, Python không có do-while. Muốn giả lập do-while có thể làm như sau:

```
1     while True:
2         x = int(input("Input:"))
3         if x > 0:
4             break
```

Chương trình sẽ làm trước, nếu đúng thì break.

## 2.4 Infinite loop

Nếu điều kiện luôn đúng thì loop không bao giờ dừng

```
1     while True:
2         print("Running")
```

## 2.5 Nested loop

Loop trong loop. Giống như: Mỗi tuần 7 ngày, ngày ăn 3 bữa.

```
1     for i in range(7):
2         for j in range(3):
3             print("i=", i, "j=", j)
```

Đây là nơi sinh ra độ phức tạp  $O(n^2)$ . Vì nếu loop ngoài chạy n lần, loop trong chạy n lần  $\rightarrow$  tổng là  $n \times n = n^2$

## 3 Loop với mảng

Đối với mảng, loop giúp ta:

- Duyệt mảng
- Tìm max
- Tính tổng
- Sắp xếp

```
1     a = [1, 2, 3, 4, 5]
2     for i in range(len(a)):
3         print(a[i])
```

Hoặc ta cũng có thể viết thế này:

```
1     for x in a:
2         print(x)
```

## 4 Loop với chuỗi

Tương tự như mảng, ta cũng có thể duyệt chuỗi để tìm kí tự, kiểm tra kí tự có tồn tại trong chuỗi,...

```
1     s = "Hello"
2     for ch in s:
3         print(ch)
```

## 5 Khi nào không nên dùng loop?

- Khi có công thức toán học
- Khi có thể dùng đệ quy
- Khi có thư viện tối ưu sẵn

Ví dụ: Tính tổng từ 1 đến n. Ta có thể dùng công thức:  $\frac{n.(n+1)}{2}$  thay vì dùng loop