

AI VIET NAM – AI COURSE 2026

# Vòng lặp for trong Python

Nguyễn Đăng Nhã

Hồ Quang Hiển

## I. Lý thuyết

### Vòng lặp for trong Python

Vòng lặp **for** cho phép duyệt qua các phần tử của một *sequence* (chuỗi, danh sách, ...) hoặc lặp với số lần xác định qua **range()**.

- For x in string:** Lặp qua từng ký tự trong một chuỗi.
- For với range:** Sử dụng hàm **range()** để tạo dãy số và lặp qua chúng (bắt đầu, kết thúc, bước nhảy)

#### Ví dụ

```

1 # For qua chuỗi
2 text = "Python"
3 for char in text:
4     print(char) # P, y, t, h, o, n
5
6 # For với range
7 for i in range(3):          # 0, 1, 2
8     print(i)
9
10 for i in range(1, 6, 2):   # 1, 3, 5
11     print(i)

```

### Số e (Euler) & Chuỗi Taylor cho e và $e^x$

- Số e:** Là hằng số toán học xấp xỉ 2.71828, được định nghĩa qua chuỗi Taylor:

$$e = \sum_{n=0}^{\infty} \frac{1}{n!} = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots$$

Trong đó,  $n!$  là giai thừa của  $n$  (ví dụ:  $4! = 4 \times 3 \times 2 \times 1 = 24$ ).

- **Hàm mũ**  $e^x$  có khai triển Taylor:

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$$

Trong đó:

- $x$ : Giá trị thực bất kỳ (ví dụ:  $x = 2$  thì tính  $e^2$ ).
- Chuỗi này hội tụ đến giá trị thực của  $e^x$  khi số hạng tăng lên.
- **Thực hành Python:** Dùng `for` để tính tổng các số hạng; cần giai thừa  $n!$  và lũy thừa  $x^n$ .

### Giai thừa trong Python

- Giai thừa  $n!$  là tích các số nguyên dương từ 1 đến  $n$ .
- Có thể tự cài đặt bằng `for` hoặc dùng `math.factorial(n)`.

## II. Bài tập

### Bài 1: Tính giai thừa bằng `for`

#### Yêu cầu bài tập

Viết hàm `factorial(n)` bằng `for` để tính giai thừa của một số nguyên dương  $n$ ; in kết quả với  $n = 6$ .

#### Hàm factorial

```

1 def factorial(n):
2     # TODO: implement
3     return result
4
5 # Test case
6 n = 6
7 print(f"Giai thừa của {n} là: {factorial(n)}") # Kết quả mong muốn: 720

```

## Bài 2: Tính số $e$ bằng chuỗi Taylor

### ☰ Yêu cầu bài tập

Viết hàm `calculate_e(n)` bằng `for` để xấp xỉ  $e$  với  $n$  số hạng đầu tiên của chuỗi Taylor.

- Tính  $e$  với  $n = 8$ .
- So sánh với `math.e`.

### Hàm calculate\_e

```

1 import math
2
3 def calculate_e(n):
4     # TODO: implement
5     return e_approx
6
7 # Test case
8 n = 8
9 e_approx = calculate_e(n)
10 print(f"Số e gần đúng với {n} số hạng: {e_approx}")
11 print(f"Số e thực tế: {math.e}")
12 print(f"Độ lệch: {abs(math.e - e_approx)}")

```

## Bài 3: Tính $e^x$ bằng chuỗi Taylor

### ☰ Yêu cầu bài tập

Viết hàm `calculate_exp(x, n)` sử dụng vòng lặp `for` để tính gần đúng  $e^x$  với  $n$  số hạng đầu tiên của chuỗi Taylor.

- Tính  $e^2$  với  $n = 10$ .
- So sánh với giá trị thực từ `math.exp(2)`.

### Hàm calculate\_exp

```

1 import math
2
3 def calculate_exp(x, n):
4     # TODO: implement
5     return exp_approx
6
7 # Test case
8 x = 2
9 n = 10
10 exp_approx = calculate_exp(x, n)
11 print(f"e^{x} gần đúng với {n} số hạng: {exp_approx}")

```

```
12 print(f"e^{x} thực tế: {math.exp(x)}")  
13 print(f"Độ lệch: {abs(math.exp(x) - exp_approx)}")
```

 **Gợi ý mở rộng**

- Thay đổi  $n$  trong Bài 2 và Bài 3 để quan sát độ chính xác.
- Thêm kiểm tra đầu vào (ví dụ:  $n > 0$ ) để tránh lỗi.