

Lambda

Hoàng-Nguyên Vũ

1 Lý Thuyết: Hàm lambda trong Python

1.1 1. Khái niệm

- lambda trong Python là cách tạo hàm nhỏ gọn, nhanh chóng, còn gọi là *anonymous function* (hàm ẩn danh).
- Không cần đặt tên cho hàm như khi dùng `def`.

Cú pháp:

```
1 lambda arguments: expression
```

- **arguments:** các biến đầu vào
- **expression:** biểu thức trả về (chỉ 1 dòng, không nhiều lệnh)

1.2 2. Ví dụ đơn giản

```
1 add = lambda x, y: x + y
2 print(add(3, 5)) # Output: 8
```

1.3 3. Khi nào nên dùng lambda?

- Khi cần viết hàm ngắn gọn, chỉ 1 dòng.
- Khi truyền hàm làm đối số cho các hàm khác như `map()`, `filter()`, `sorted()`, v.v.

Ví dụ:

```
1 numbers = [1, 2, 3, 4]
2 squared = list(map(lambda x: x**2, numbers))
3 print(squared) # Output: [1, 4, 9, 16]
```

1.4 4. So sánh lambda và def

Tiêu chí	lambda	def
Độ dài	Ngắn gọn	Dài hơn
Tên hàm	Không cần tên	Cần có tên
Nội dung	1 dòng expression	Nhiều dòng lệnh
Sử dụng	Hàm nhỏ, nhanh gọn	Hàm lớn, phức tạp

2 Bài Tập Thực Hành

Bài 1

Tạo một hàm `lambda` tính bình phương một số.

Input: 4 **Output:** 16

Bài 2

Tạo hàm `lambda` kiểm tra xem một số có lớn hơn 10 không.

Input: 15 **Output:** True

Bài 3

Dùng `map` và `lambda` để nhân đôi từng phần tử trong list `[1, 3, 5, 7]`.

Output: [2, 6, 10, 14]

Bài 4

Dùng `filter` và `lambda` để lọc các số chẵn từ list `[1, 2, 3, 4, 5, 6]`.

Output: [2, 4, 6]

Bài 5

Sắp xếp danh sách các tuple theo phần tử thứ hai dùng `lambda`.

```
1 students = [('John', 80), ('Jane', 95), ('Dave', 90)]
```

Kết quả mong muốn:

```
1 [('John', 80), ('Dave', 90), ('Jane', 95)]
```

3 Bonus: Bài Tập Nâng Cao

Bài 6

Viết hàm `calculator` nhận vào một phép toán ('+', '-', '*', '/') và trả về một hàm `lambda` thực hiện phép toán đó.

Ví dụ:

```
1 add = calculator('+')
2 print(add(3, 4))    # Output: 7
```

Gợi ý:

```
1 def calculator(op):
2     if op == '+':
3         return lambda x, y: x + y
4     elif op == '-':
```

```
5         return lambda x, y: x - y
6     elif op == '*':
7         return lambda x, y: x * y
8     elif op == '/':
9         return lambda x, y: x / y
```