

# Kỹ thuật hay và gọn gàng khi thao tác với danh sách (list)

## 1. **List Comprehension** - Viết gọn vòng lặp trong danh sách

Cơ bản:

```
numbers = [1, 2, 3, 4, 5]
squared = [x * x for x in numbers]
print(squared) # [1, 4, 9, 16, 25]
```

Có điều kiện:

```
evens = [x for x in numbers if x % 2 == 0]
print(evens) # [2, 4]
```

## 2. **enumerate()** - Lặp qua danh sách có chỉ số

```
fruits = ['apple', 'banana', 'cherry']
for index, fruit in enumerate(fruits):
    print(index, fruit)
```

```
0 apple
1 banana
2 cherry
```

## 3. **zip()** - Lặp qua nhiều danh sách cùng lúc

```
names = ['Alice', 'Bob', 'Charlie']
scores = [85, 90, 95]

for name, score in zip(names, scores):
```

```
print(f"{name} got {score}")
```

```
Alice got 85  
Bob got 90  
Charlie got 95
```

#### 4. Lặp ngược - `reversed()`

```
nums = [1, 2, 3, 4]  
for x in reversed(nums):  
    print(x)
```

#### 5. Lặp theo chỉ số cách quãng - `range(start, stop, step)`

```
for i in range(0, 10, 2): # 0 2 4 6 8  
    print(i)
```

#### 6. Nested loop (vòng lặp lồng nhau)

```
for i in range(3):  
    for j in range(2):  
        print(f"i={i}, j={j}")
```

#### 7. List comprehension nâng cao (nested)

```
matrix = [[1, 2], [3, 4], [5, 6]]  
flattened = [num for row in matrix for num in row]  
print(flattened) # [1, 2, 3, 4, 5, 6]
```

#### 8. Sử dụng `map()` và `filter()` - Tính hàm học functional

```
nums = [1, 2, 3, 4]  
squared = list(map(lambda x: x * x, nums))  
evens = list(filter(lambda x: x % 2 == 0, nums))  
  
print(squared) # [1, 4, 9, 16]  
print(evens) # [2, 4]
```

Tác giả: **Đỗ Ngọc Tú**  
Công Ty Phần Mềm **VHTSoft**

Phiên bản #1

Được tạo 18 tháng 4 2025 15:02:21 bởi Đỗ Ngọc Tú

Được cập nhật 18 tháng 4 2025 15:09:29 bởi Đỗ Ngọc Tú