

Python cho Khoa học dữ liệu

Bài 9: Kiểu dữ liệu danh sách - Python list

Hà Minh Tuấn
hmtuan@hcmus.edu.vn

Khoa Toán - Tin học
Trường Đại học Khoa học Tự nhiên, ĐHQG-HCM

Ngày 3 tháng 9 năm 2025

Nội dung bài học

1. Giới thiệu chung với list

2. Một số phương thức thông dụng trong list

Giới thiệu về List trong Python

- List là kiểu dữ liệu tích hợp sẵn trong Python.
- Được khai báo bằng dấu ngoặc vuông [], các phần tử cách nhau bởi dấu phẩy.
- Có thể chứa nhiều kiểu dữ liệu khác nhau (int, float, string, bool, object).
- List là **mutable**: có thể thay đổi phần tử, thêm, xóa trong quá trình chạy.
- Tương tự mảng trong C/C++/Java, nhưng linh hoạt hơn vì cho phép phần tử khác kiểu.

Ví dụ về Python List

```
list1 = ["Rohan", "Physics", 21, 69.75]
list2 = [1, 2, 3, 4, 5]
list3 = ["a", "b", "c", "d"]
list4 = [25.50, True, -55, 1+2j]

print(list1)
print(list4)
```

```
['Rohan', 'Physics', 21, 69.75]
[25.5, True, -55, (1+2j)]
```

Truy cập phần tử trong List

```
list1 = ['physics', 'chemistry', 1997, 2000]
list2 = [1, 2, 3, 4, 5, 6, 7]

print("list1[0]:", list1[0])
print("list2[1:5]:", list2[1:5])
```

list1[0]: physics

list2[1:5]: [2, 3, 4, 5]

Cập nhật phần tử trong List

```
list = ['physics', 'chemistry', 1997, 2000]

print("Giá trị ban đầu:", list[2])
list[2] = 2001
print("Giá trị mới:", list[2])
```

Giá trị ban đầu: 1997

Giá trị mới: 2001

Xóa phần tử trong List

```
list1 = ['physics', 'chemistry', 1997, 2000]  
  
print(list1)  
del list1[2]  
print("Sau khi xóa:", list1)
```

```
['physics', 'chemistry', 1997, 2000]  
Sau khi xóa: ['physics', 'chemistry', 2000]
```

Các phép toán với List

```
print([1, 2, 3] + [4, 5, 6])      # Nối list
print(['Hi!'] * 4)                  # Lặp list
print(3 in [1, 2, 3])              # Kiểm tra membership
```

```
[1, 2, 3, 4, 5, 6]
['Hi!', 'Hi!', 'Hi!', 'Hi!']
True
```

Indexing và Slicing

```
L = ['spam', 'Spam', 'SPAM!']

print(L[2])      # phần tử thứ 3
print(L[-2])    # phần tử thứ 2 từ cuối
print(L[1:])    # cắt từ vị trí 1
```

SPAM!

Spam

['Spam', 'SPAM!']

Duyệt List bằng vòng lặp

```
fruits = ["apple", "banana", "cherry"]

for f in fruits:
    print(f)
```

apple
banana
cherry

List Comprehension

```
numbers = [x**2 for x in range(5)]  
print(numbers)
```

[0, 1, 4, 9, 16]

Thêm phần tử vào List

```
fruits = ["apple", "banana"]
fruits.append("cherry")
print(fruits)
```

```
['apple', 'banana', 'cherry']
```

Sắp xếp List

```
nums = [4, 1, 3, 9, 2]
nums.sort()
print(nums)
```

[1, 2, 3, 4, 9]

Copy và Nối List

```
list1 = [1, 2, 3]
list2 = list1.copy()
list3 = list1 + list2
print(list2)
print(list3)
```

[1, 2, 3]

[1, 2, 3, 1, 2, 3]

Một số phương thức thường dùng của List

- `append(obj)` – thêm phần tử vào cuối.
- `insert(i, obj)` – chèn tại vị trí i.
- `remove(obj)` – xóa phần tử có giá trị obj.
- `pop(i)` – xóa và trả về phần tử tại i.
- `sort()` – sắp xếp list.
- `reverse()` – đảo ngược list.
- `copy()` – sao chép list.
- `clear()` – xóa toàn bộ phần tử.

Một số hàm built-in với List

- `len(list)` – độ dài list.
- `max(list)` – giá trị lớn nhất.
- `min(list)` – giá trị nhỏ nhất.
- `sum(list)` – tính tổng.
- `list(seq)` – chuyển đổi từ tuple/iterable thành list.

Phương thức append()

```
# Thêm phần tử vào cuối list
fruits = ["apple", "banana", "cherry"]
fruits.append("orange")
print(fruits)
```

```
['apple', 'banana', 'cherry', 'orange']
```

Phương thức pop()

```
# Xóa và trả về phần tử tại index (mặc định là cuối)
numbers = [1, 2, 3, 4, 5]
removed = numbers.pop(2)
print("Removed:", removed)
print("Còn lại:", numbers)
```

Removed: 3

Còn lại: [1, 2, 4, 5]

Phương thức remove()

```
# Xóa phần tử đầu tiên có giá trị bằng đối số
fruits = ["apple", "banana", "cherry", "banana"]
fruits.remove("banana")
print(fruits)
```

```
['apple', 'cherry', 'banana']
```

Phương thức insert()

```
# Chèn phần tử tại vị trí chỉ định
fruits = ["apple", "cherry"]
fruits.insert(1, "banana")
print(fruits)
```

```
['apple', 'banana', 'cherry']
```

Phương thức sort()

```
# Sắp xếp list theo thứ tự tăng dần
numbers = [5, 2, 9, 1, 7]
numbers.sort()
print(numbers)

# Sắp xếp giảm dần
numbers.sort(reverse=True)
print(numbers)
```

```
[1, 2, 5, 7, 9]
[9, 7, 5, 2, 1]
```

Phương thức reverse()

```
# Đảo ngược thứ tự các phần tử trong list
letters = ["a", "b", "c", "d"]
letters.reverse()
print(letters)
```

```
['d', 'c', 'b', 'a']
```

Phương thức clear()

```
# Xóa toàn bộ các phần tử trong list
fruits = ["apple", "banana", "cherry"]
fruits.clear()
print(fruits)
```

[]

Phương thức copy()

```
# Tao một bản sao của list
fruits = ["apple", "banana", "cherry"]
new_fruits = fruits.copy()
print("Bản gốc:", fruits)
print("Bản sao:", new_fruits)
```

Bản gốc: ['apple', 'banana', 'cherry']
Bản sao: ['apple', 'banana', 'cherry']