

Python cho Khoa học dữ liệu

Bài 13: Kiểu dữ liệu mảng - Python Arrays

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. Arrays trong Python

2. Các methods quan trọng

Giới thiệu về Arrays trong Python

- Python không có kiểu dữ liệu mảng (array) tích hợp sẵn như C++ hoặc Java.
- Thay vào đó, ta thường dùng list, tuple hoặc module array.
- Mảng là một cấu trúc dữ liệu có kích thước cố định, chứa các phần tử cùng kiểu.
- Các phần tử được lưu liên tiếp trong bộ nhớ và có chỉ số từ 0.

Đặc điểm của Array

- Chỉ số bắt đầu từ 0.
- Có thể chứa kiểu số nguyên, số thực, hoặc ký tự Unicode.
- Mỗi phần tử được truy cập bằng chỉ số.
- Các thao tác cơ bản: **Traverse, Insertion, Deletion, Search, Update.**

Tạo Array trong Python

```
import array as arr

# Array số nguyên
a = arr.array('i', [1, 2, 3])
print(type(a), a)
```

```
# Array ký tự Unicode
a = arr.array('u', 'BAT')
print(type(a), a)
```

```
# Array số thực
a = arr.array('d', [1.1, 2.2, 3.3])
print(type(a), a)
```

```
<class 'array.array'> array('i', [1, 2, 3])
<class 'array.array'> array('u', 'BAT')
<class 'array.array'> array('d', [1.1, 2.2, 3.3])
```

Truy cập phần tử trong Array

```
from array import *
array1 = array('i', [10,20,30,40,50])
print(array1[0])
print(array1[2])
```

10

30

Thêm phần tử (Insertion)

```
from array import *
array1 = array('i', [10,20,30,40,50])
array1.insert(1,60)
for x in array1:
    print(x)
```

10
60
20
30
40
50

Xóa phần tử (Deletion)

```
from array import *
array1 = array('i', [10,20,30,40,50])
array1.remove(40)
for x in array1:
    print(x)
```

10
20
30
50

Tìm kiếm phần tử (Search)

```
from array import *
array1 = array('i', [10,20,30,40,50])
print(array1.index(40))
```

3

Cập nhật phần tử (Update)

```
from array import *
array1 = array('i', [10,20,30,40,50])
array1[2] = 80
for x in array1:
    print(x)
```

10
20
80
40
50

Các methods quan trọng của Array

- `append(x)`: Thêm phần tử vào cuối mảng.
- `extend(iterable)`: Thêm nhiều phần tử từ iterable.
- `insert(i, x)`: Thêm phần tử tại vị trí chỉ số i.
- `remove(x)`: Xóa phần tử đầu tiên có giá trị bằng x.
- `pop([i])`: Trả về và xóa phần tử tại vị trí i (mặc định là cuối mảng).
- `index(x)`: Trả về chỉ số của phần tử x đầu tiên.
- `reverse()`: Đảo ngược mảng.
- `buffer_info()`: Trả về thông tin (địa chỉ, số phần tử).

Method append()

```
from array import *
arr1 = array('i', [1, 2, 3])
arr1.append(4)
print(arr1)
```

```
array('i', [1, 2, 3, 4])
```

Method extend()

```
from array import *
arr1 = array('i', [1, 2, 3])
arr1.extend([4, 5, 6])
print(arr1)
```

```
array('i', [1, 2, 3, 4, 5, 6])
```

Method insert()

```
from array import *
arr1 = array('i', [10, 20, 30])
arr1.insert(1, 15)
print(arr1)
```

```
array('i', [10, 15, 20, 30])
```

Method remove()

```
from array import *
arr1 = array('i', [1, 2, 3, 2, 4])
arr1.remove(2)
print(arr1)
```

```
array('i', [1, 3, 2, 4])
```

Method pop()

```
from array import *
arr1 = array('i', [5, 6, 7, 8])
x = arr1.pop(2)
print("Removed:", x)
print(arr1)
```

Removed: 7

array('i', [5, 6, 8])

Method index()

```
from array import *
arr1 = array('i', [10, 20, 30, 40])
print(arr1.index(30))
```

2

Method reverse()

```
from array import *
arr1 = array('i', [1, 2, 3, 4])
arr1.reverse()
print(arr1)
```

```
array('i', [4, 3, 2, 1])
```

Method buffer_info()

```
from array import *
arr1 = array('i', [1, 2, 3, 4])
print(arr1.buffer_info())
```

(140620393045360, 4)