

BÀI GIẢNG: CÁC THUỘC TÍNH VÀ PHƯƠNG THỨC TRONG CÁC LOẠI CONTAINER PYTHON

Tác giả: Đặng Kim Thi

1. Giới thiệu

Trong Python, các loại container (`list`, `tuple`, `set`, `dict`) có các thuộc tính và phương thức giúp thao tác với dữ liệu dễ dàng hơn. Bài giảng này sẽ đi sâu vào chi tiết từng loại container.

2. Danh sách (List)

2.1. Thuộc tính của List

- **Có thứ tự:** Các phần tử được lưu theo thứ tự chèn vào.
- **Thay đổi được:** Có thể thêm, sửa, xóa phần tử.
- **Cho phép trùng lặp.**

2.2. Các phương thức quan trọng của List

Phương thức	Mô tả
<code>append(x)</code>	Thêm <code>x</code> vào cuối danh sách.
<code>extend(iterable)</code>	Nối danh sách với một iterable khác.
<code>insert(i, x)</code>	Chèn <code>x</code> vào vị trí <code>i</code> .
<code>remove(x)</code>	Xóa phần tử đầu tiên có giá trị <code>x</code> .
<code>pop([i])</code>	Xóa và trả về phần tử tại vị trí <code>i</code> (mặc định là phần tử cuối).
<code>index(x)</code>	Trả về vị trí xuất hiện đầu tiên của <code>x</code> .
<code>count(x)</code>	Đếm số lần xuất hiện của <code>x</code> .
<code>sort()</code>	Sắp xếp danh sách.
<code>reverse()</code>	Đảo ngược danh sách.
<code>copy()</code>	Tạo bản sao của danh sách.

Ví dụ:

```
my_list = [1, 2, 3]
my_list.append(4)
print(my_list) # Output: [1, 2, 3, 4]
```

3. Bộ giá trị (Tuple)

3.1. Thuộc tính của Tuple

- Có thứ tự.
- Không thay đổi được (immutable).
- Cho phép trùng lặp.

3.2. Các phương thức của Tuple

Phương thức	Mô tả
<code>count(x)</code>	Đếm số lần xuất hiện của <code>x</code> .
<code>index(x)</code>	Trả về vị trí đầu tiên của <code>x</code> .

Ví dụ:

```
my_tuple = (1, 2, 3, 2)
print(my_tuple.count(2)) # Output: 2
```

4. Tập hợp (Set)

4.1. Thuộc tính của Set

- Không có thứ tự.
- Không chứa phần tử trùng lặp.
- Có thể thay đổi được (mutable).

4.2. Các phương thức quan trọng của Set

Phương thức	Mô tả
<code>add(x)</code>	Thêm <code>x</code> vào tập hợp.
<code>remove(x)</code>	Xóa <code>x</code> khỏi tập hợp (lỗi nếu <code>x</code> không tồn tại).
<code>discard(x)</code>	Xóa <code>x</code> khỏi tập hợp (không lỗi nếu <code>x</code> không tồn tại).
<code>pop()</code>	Xóa và trả về một phần tử ngẫu nhiên.
<code>clear()</code>	Xóa tất cả phần tử.
<code>union(set2)</code>	Trả về hợp của hai tập hợp.
<code>intersection(set2)</code>	Trả về giao của hai tập hợp.
<code>difference(set2)</code>	Trả về phần tử có trong tập <code>set1</code> nhưng không có trong <code>set2</code> .
<code>symmetric_difference(set2)</code>	Trả về các phần tử chỉ có ở một trong hai tập hợp.

Ví dụ:

```
set1 = {1, 2, 3}
set2 = {3, 4, 5}
print(set1.union(set2)) # Output: {1, 2, 3, 4, 5}
```

5. Từ điển (Dictionary)

5.1. Thuộc tính của Dictionary

- Lưu trữ dữ liệu theo cặp khóa-giá trị.
- Không có thứ tự (trước Python 3.7), có thứ tự (từ Python 3.7+).
- Khóa phải là duy nhất và không thay đổi được.

5.2. Các phương thức quan trọng của Dictionary

Phương thức	Mô tả
<code>get(key, default)</code>	Lấy giá trị của <code>key</code> , trả về <code>default</code> nếu <code>key</code> không tồn tại.
<code>keys()</code>	Trả về danh sách các khóa.
<code>values()</code>	Trả về danh sách các giá trị.
<code>items()</code>	Trả về danh sách các cặp (<code>key</code> , <code>value</code>).
<code>update(dict2)</code>	Cập nhật dictionary với <code>dict2</code> .
<code>pop(key)</code>	Xóa <code>key</code> và trả về giá trị tương ứng.
<code>popitem()</code>	Xóa và trả về cặp (<code>key</code> , <code>value</code>) cuối cùng.
<code>clear()</code>	Xóa tất cả phần tử.

Ví dụ:

```
dict1 = {"name": "Alice", "age": 25}
print(dict1.get("name")) # Output: Alice
```

6. So sánh các phương thức của các loại container

Phương thức	List	Tuple	Set	Dictionary
Thêm phần tử	<code>append()</code>	✗	<code>add()</code>	<code>update()</code>
Xóa phần tử	<code>remove()</code> , <code>pop()</code>	✗	<code>remove()</code> , <code>discard()</code>	<code>pop()</code>

Phương thức	List	Tuple	Set	Dictionary
Kiểm tra phần tử	<code>in</code>	<code>in</code>	<code>in</code>	<code>in</code>
Đếm phần tử	<code>count()</code>	<code>count()</code>	✗	✗
Truy cập phần tử	<code>index</code>	<code>index</code>	✗	<code>get()</code>
Duyệt qua phần tử	<code>for x in list</code>	<code>for x in tuple</code>	<code>for x in set</code>	<code>for key, value in dict.items()</code>

7. Tổng kết

- `list`: Phù hợp cho danh sách có thể thay đổi.
- `tuple`: Dùng khi cần danh sách cố định.
- `set`: Dùng khi cần tập hợp không trùng lặp.
- `dict`: Dùng khi cần lưu trữ dữ liệu theo cặp khóa-giá trị.

Lựa chọn container phù hợp sẽ giúp tối ưu chương trình của bạn! 🚀