Softech®
Gather to thrive together

Aptech™
COMPUTER EDUCATION
Unleash your potential

BÀI 10

Sets Python
Kiểu dữ liệu tập hợp

Tóm Tắt Nội Dung

Trong bài học này chúng ta sẽ đi tìm hiểu lần lượt các nội dung

1. Tổng quan về Sets
2. Các phương thức xử lí với sets

10.1 Tổng quan về Sets

Khái niệm về Sets

- Trong Python, Set là một kiểu dữ liệu hỗn hợp dùng để lưu trữ tập hợp các phần tử duy nhất (distinct) không theo thứ tự. Set là một dạng của Collection trong Python và giống với tập hợp (set) trong toán học

- Một số điểm quan trọng về Set trong Python:

  1. Set không chứa các phần tử trùng lặp

  2. Set không duy trì thứ tự của các phần tử. Khi bạn in Set, thứ tự của các phần tử có thể thay đổi mỗi lần bạn thực hiện in

  3. Có thể chứa các phần tử không thay đổi như số nguyên, số thực, chuỗi, tuple trong Set. Nhưng Set không thể chứa các phần tử có thay đổi như danh sách (list) hoặc set khác

  4. Set không hỗ trợ truy cập phần tử bằng chỉ số như danh sách, vì vậy bạn không thể sử dụng Set[index]. Bạn phải sử dụng vòng lặp hoặc các phương thức có sẵn để truy cập và xử lý phần tử trong Set.

10.1 Tổng quan về Sets

Cách khai báo một Sets

```
thisset = {"apple", 3.14 , (1, 2), True, False, 30}
print(thisset)
```

- Set được lưu trữ trong một cặp ngoặc nhọn
- Các phần tử cách nhau bởi dấu phẩy ,
- Các phần tử có thể chứa nhiều loại dữ liệu int, float, bool, str, tuple, None

10.2 Các phương thức xử lý với Sets

Truy cập đến các phần tử của Sets

Bạn chỉ có thể duyệt qua các phần tử của sets bằng vòng lặp

```
thisset = {"apple", "banana", "cherry"}
for x in thisset:
  print(x)
```

Sử dụng từ khóa in để kiểm tra một phần tử có tồn tại trong set không

```
print("banana" in thisset)
```

10.2 Các phương thức xử lý với Sets

Thêm mới phần tử vào Sets

Với set bạn không thể thay đổi các phần tử sau khi khởi tạo, tuy nhiên có thể thêm mới

```
thisset = {"apple", "banana", "cherry"}
thisset.add("orange")
print(thisset)
```

Hoặc bạn có thể sử dụng phương thức **update()** để thêm mới

```
tropical = {"pineapple", "mango", "papaya"}
thisset.update(tropical)
print(thisset)
```

Bạn có thể thêm một tuples, lists, dictionaries vào set với update()

10.2 Các phương thức xử lý với Sets

Xóa phần tử ra khỏi Sets

Xóa với phương thức **remove()**

```
thisset = {"apple", "banana", "cherry"}
thisset.remove(" banana ")
print(thisset)
```

Hoặc bạn có thể sử dụng phương thức **discard()**

```
thisset.discard(" banana ")
print(thisset)
```

10.2 Các phương thức xử lý với Sets

Xóa phần tử ra khỏi Sets

Bạn cũng có thể sử dụng phương thức pop() để xóa, tuy nhiên nó xóa một phần tử **ngẫu nhiên** trong set

```
thisset = {"apple", "banana", "cherry"}
x = thisset.pop()
print(x)
print(thisset)
```

Xóa tất cả phần tử trong set, trả lại một set rỗng

```
thisset.clear()
```

Xóa hoàn toàn một set

```
del thisset
```

10.2 Các phương thức xử lý với Sets

Gộp 2 Sets

Bạn có thể sử dụng union() hay update() để gộp 2 set với nhau

```
set1 = {"a", "b" , "c"}
set2 = {1, 2, 3}
set3 = set1.union(set2)
set1.update(set2)
print(set3)
```

union() trả lại một set3 mới từ 2 set đã gộp

update() là gộp thêm set 2 vào set 1

10.2 Các phương thức xử lý với Sets

Một số phương thức khác

Phương thức	Ý nghĩa
intersection()	Trả về một Set mới chứa các phần tử chung của hai Set
difference()	Trả về một Set mới chứa các phần tử chỉ có trong Set đầu tiên nhưng không có trong Set thứ ha
issubset()	Kiểm tra xem Set có là tập con của một Set khác hay không
issuperset()	Kiểm tra xem Set có là tập cha của một Set khác hay không

Set là một công cụ hữu ích trong Python để làm việc với các tập hợp duy nhất của dữ liệu mà không cần quan tâm đến thứ tự của chúng

Tổng kết lại bài

1. Nắm được tổng quan về Sets
2. Nắm được các phương thức xử lí với sets