

Kiểu dữ liệu Set trong python

TS. Nguyễn Tuấn Anh

Trường Đại học Công nghệ Thông tin & Truyền thông

Khoa Công nghệ Thông tin

Tháng 2 năm 2023



Kiểu dữ liệu Set

Set là một trong 4 kiểu dữ liệu dạng tập hợp trong python (set,, List, Tuple, và Dictionary)

Set là kiểu dữ liệu không có thứ tự, không thay đổi được và không được lập chỉ mục

Tạo tập hợp thisset

```
thisset = {"apple", "banana", "cherry"}
```

Hàm len() trả về số phần tử của set:

```
print(len(thisset))
```



Kiểu dữ liệu Set

Kiểu dữ liệu set có thể chứa bất kỳ kiểu dữ liệu nào

```
set1 = {"apple", "banana", "cherry"}  
set2 = {1, 5, 7, 9, 3}  
set3 = {True, False, False}  
set4 = {"abc", 34, True, 40, "male"}
```

Hàm khởi tạo set(): Sử dụng hàm khởi tạo để tạo set:

```
thisset = set(("apple", "banana", "cherry")) # note the double round-brackets
```



Truy cập các phần tử trong set

Sử dụng vòng lặp for và toán tử in:

```
thisset = {"apple", "banana", "cherry"}
```

```
for x in thisset:  
    print(x)
```

Hàm khởi tạo set(): Sử dụng hàm khởi tạo để tạo set:

```
thisset = set(("apple", "banana", "cherry")) # note the double round-brackets
```

Ghi chú: Không thể thay đổi giá trị của các phần tử trong set



Thêm phần tử vào set

Phương thức add()

```
thisset = {"apple", "banana", "cherry"}
```

```
thisset.add("orange")
```

Phương thức update() cập nhật thêm các phần tử vào set từ một set khác.

Thêm các phần tử của set tropical vào set thisset.

```
thisset = {"apple", "banana", "cherry"}
```

```
tropical = {"pineapple", "mango", "papaya"}
```

```
thisset.update(tropical)
```



Xóa phần tử trong set

Để xóa phần tử trong set sử dụng phương thức `remove()`, hoặc `discard()`

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

Sử dụng phương thức: `discard()`

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

Xóa ngẫu nhiên một phần tử trong set sử dụng phương thức `pop()`:

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



Xóa phần tử trong set

Xóa toàn bộ set sử dụng phương thức clear()

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

Sử dụng từ khóa del xóa set

```
thisset = {"apple", "banana", "cherry"}  
  
del thisset
```



Duyệt các phần tử trong set

Duyệt toàn bộ các phần tử trong set

```
thisset = {"apple", "banana", "cherry"}
```

```
for x in thisset:  
    print(x)
```



Gộp hai set

Có thể sử dụng phương thức `union()` để trả về một tập hợp mới chứa tất cả các mục từ cả hai tập hợp hoặc phương thức `update()` để chèn tất cả các phần tử từ tập hợp này sang tập hợp khác:

```
set1 = {"a", "b" , "c"}  
set2 = {1, 2, 3}
```

```
set3 = set1.union(set2)
```

Sử dụng phương thức `update()` chèn các phần tử của `set2` vào `set1`

```
set1 = {"a", "b" , "c"}  
set2 = {1, 2, 3}
```

```
set1.update(set2)
```



Gộp hai set

Phương thức `intersection_update()` giữ lại các phần tử có trong cả hai set

```
x = {"apple", "banana", "cherry"}  
y = {"google", "microsoft", "apple"}
```

```
x.intersection_update(y)
```

Phương thức `intersection()` trả về set mới bao gồm các phần tử có trong cả hai set

```
x = {"apple", "banana", "cherry"}  
y = {"google", "microsoft", "apple"}
```

```
z = x.intersection(y)
```



Gộp hai set

Phương thức `symmetric_difference_update()` giữ lại các phần tử không có trong đồng thời cả hai set.

```
x = {"apple", "banana", "cherry"}  
y = {"google", "microsoft", "apple"}
```

```
x.symmetric_difference_update(y)
```

Phương thức `symmetric_difference()` trả về set mới gồm các phần tử không có trong đồng thời cả hai set.

```
x = {"apple", "banana", "cherry"}  
y = {"google", "microsoft", "apple"}
```

```
z = x.symmetric_difference(y)
```

