Split

Rjust,ljust,center

Dict

Trong set{

Sử dụng add() để thêm một phần tử đơn lẻ vào set.

Sử dụng update() để thêm nhiều phần tử vào set. update() có thể nhận vào một danh sách, tuple, hoặc thậm chí là một set khác.

Sử dụng remove() để xóa một phần tử khỏi set. Nếu phần tử không tồn tại, phương thức này sẽ gây ra lỗi KeyError.

Sử dụng discard() để xóa một phần tử khỏi set. Nếu phần tử không tồn tại, phương thức này sẽ không gây ra lỗi.

Sử dụng pop() để xóa và trả về một phần tử ngẫu nhiên từ set.

Phương thức union() trả về một set mới chứa tất cả các phần tử của các set được kết hợp (phép hợp).

python

Copy code

set1 = {1, 2, 3}

set2 = {3, 4, 5}

set3 = {5, 6, 7}

# Sử dụng phương thức union()

result = set1.union(set2, set3)

print(result) # Output: {1, 2, 3, 4, 5, 6, 7}

# Sử dụng toán tử |

result = set1 | set2 | set3

print(result) # Output: {1, 2, 3, 4, 5, 6, 7}

Phương thức intersection() trả về một set mới chỉ chứa các phần tử chung giữa các set (phép giao).

python

Copy code

set1 = {1, 2, 3}

set2 = {3, 4, 5}

set3 = {3, 5, 7}

# Sử dụng phương thức intersection()

result = set1.intersection(set2, set3)

print(result) # Output: {3}

# Sử dụng toán tử &

result = set1 & set2 & set3

print(result) # Output: {3}

Phương thức difference() trả về một set mới chứa các phần tử chỉ có trong set đầu tiên mà không có trong các set còn lại (phép hiệu).

python

Copy code

set1 = {1, 2, 3, 4}

set2 = {3, 4, 5}

set3 = {4, 5, 6}

# Sử dụng phương thức difference()

result = set1.difference(set2, set3)

print(result) # Output: {1, 2}

# Sử dụng toán tử -

result = set1 - set2 - set3

print(result) # Output: {1, 2}

A screenshot of a computer program

Description automatically generated

}