Difference()

The difference() method is used to find elements that exists in one set but not in another. It returns a new set containing elements from the first set that are not present in second set. This operation is similar to the substraction of sets (A-B), where only unique elements from the first set remain.

Example

```
A = {1, 2, 3, 4}
B = {3, 4, 5, 6}
result = A.difference(B)
print(result)
```

Output : {1, 2}

Using operator

```
A = {1, 2, 3, 4}
B = {3, 4, 5, 6}
result = A - B
print(result)
```

Output = {1, 2}

Multiple sets

```
A = {1, 2, 3, 4, 5}
B = {3, 4}
C = {5}
result = A.difference(B, C)
print(result)
```

Output : {1, 2}

Empty set

```
A = {1, 2, 3}
B = set()
result = A.difference(B)
print(result)
```

Output = {1, 2, 3}

Is superset ()

The issuperset() method is used to determine if a set contains all elements of another set. This operation is fundamental in set theory and is particularly useful when you want to verify if one collection encompasses another.

Syntax

set1.issuperset(set2)

Example

1. Basic superset check.

A = {1, 2, 3, 4, 5}
B = {1, 2, 3}
print (A.issuperset (B))

output = True.

Using the >= operator.

A = {1, 2, 3, 4, 5}
B = {1, 2, 3}
print (A >= B)
output = True

Isdisjoint ()

The isdisjoint() method is used to determine if two sets have no elements in common. If they share no elements the method returns 'True', otherwise it returns False.

Syntax

set1.isdisjoint(set2)

Example:-

1) Disjoint set

A = {1, 2, 3}
B = {4, 5, 6}
print (A.isdisjoint (B))

output True

(ii) Non - Disjoint sets

A = {1, 2, 3}

B = {3, 4, 5}

Print (A is disjoint (B))

output = False