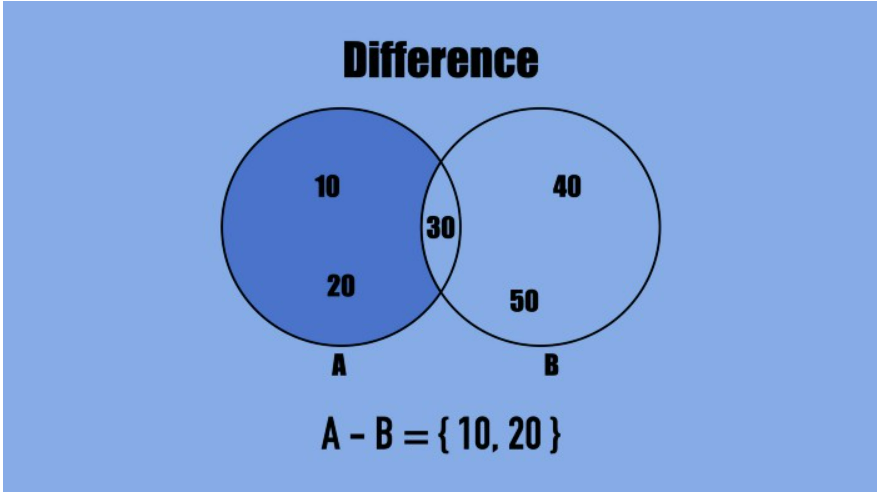
**Frozen Set Methods in Python**

The frozen set in Python supports the basic set operations.

**difference**

You can compute the difference between two frozen sets using the difference() method.

Here is an illustration:



And here is the code:

**A = frozenset({10, 20, 30})**

**B = frozenset({30, 40, 50})**

**print(A.difference(B))**

Output:

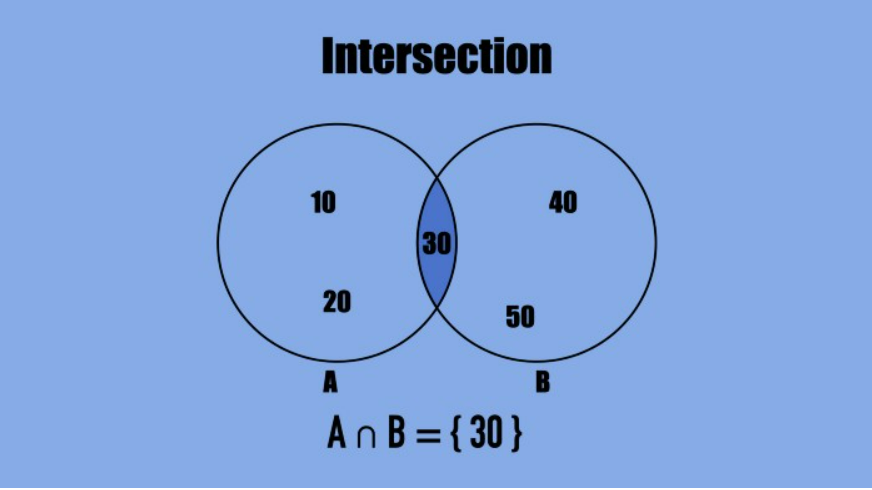
**frozenset({10, 20})**

**A - B is equal to the elements present in A but not in B B - A is equal to the elements present in B but not in A**

### intersection

You can compute the intersection between two frozen sets using the intersection() method.

Here is an illustration:



Here is an example:

**A = frozenset({10, 20, 30})**

**B = frozenset({30, 40, 50})**

**print(A.intersection(B))**

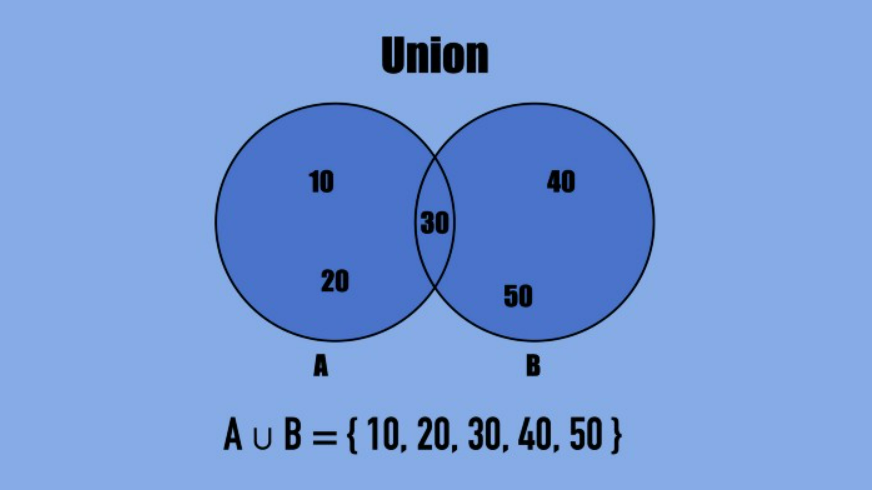
Output:

**frozenset({30})**

### union

You can compute the union between two frozen sets using the union() method.

Here is an illustration:



And here is the code:

**A = frozenset({10, 20, 30})**

**B = frozenset({30, 40, 50})**

**print(A.union(B))**

Output:

**frozenset({50, 20, 40, 10, 30})**

### symmetric\_difference

The symmetric difference between two sets is the opposite of the intersection between them.

In other words, the symmetric difference is the set with values in either set A or B, but not in both.

For example:

**A = frozenset({30, 40, 50})**

**B = frozenset({30, 40, 10})**

**print(A.symmetric\_difference(B))**

Output:

**frozenset({10, 50})**

**isdisjoint**

Two (frozen) sets are disjoint if they have no common elements.

To check if two frozen sets are disjoint, use the isdisjoint() method.

For instance:

**A = frozenset({10, 20, 30})**

**B = frozenset({30, 40, 50})**

**print(A.isdisjoint(B))**

Output:

**False**

These frozen sets are not disjoint as they have one common element—the number 30.

Python set isdisjoint() function check whether the two sets are disjoint or not, if it is disjoint then it returns True otherwise it will return False. Two sets are said to be disjoint when their intersection is null. In simple words, they do not have any common element in between them.

**issubset**

A (frozen) set is a subset of another (frozen) set if all the elements of the set are present in another.

For example:

**A = frozenset({30, 40})**

**B = frozenset({30, 40, 50})**

**print(A.issubset(B))**

Output:

**True**

The result is True because all the elements of set A are found in set B.

Python set issubset() method returns True if all elements of a set A are present in another set B which is passed as an argument and returns false if all elements not present.

**issuperset**

To check if a set has all the elements of another, use the issuperset() method.

Python set issuperset() method returns True if all elements of a set A occupies set B which is passed as an argument and returns false if all elements of B are not present in A. This means if A is a superset of B then it returns true; else False

For example:

**A = frozenset({30, 40, 50})**

**B = frozenset({30, 40})**

**print(A.issuperset(B))**

Output:

**True**

The result is True because all the elements of set B are found in set A.

**copy**

To create a copy of a frozen set, use the copy() method.

For example:

**A = frozenset({30, 40, 50})**

**B = A.copy()**

**print(B)**

Output:

**frozenset({40, 50, 30})**