Learning Assignment 5

Set Operations in Python

1. Union (| or union()):-

Combines elements from both sets.

Duplicates are removed.

Example:

set1 = {1, 2, 3}

set2 = {3, 4, 5}

# Using | operator

union\_set = set1 | set2

print(union\_set) # Output: {1, 2, 3, 4, 5}

# Using union() method

union\_set2 = set1.union(set2)

print(union\_set2) # Output: {1, 2, 3, 4, 5}

2. Intersection (& or intersection()):-

Returns only the common elements present in both sets.

Example:

set1 = {1, 2, 3, 4}

set2 = {3, 4, 5, 6}

# Using & operator

intersection\_set = set1 & set2

print(intersection\_set) # Output: {3, 4}

# Using intersection() method

intersection\_set2 = set1.intersection(set2)

print(intersection\_set2) # Output: {3, 4}

3. Difference (- or difference()):-

Returns elements present in the first set but not in the second.

Example:

set1 = {1, 2, 3, 4}

set2 = {3, 4, 5, 6}

# Using - operator

difference\_set = set1 - set2

print(difference\_set) # Output: {1, 2}

# Using difference() method

difference\_set2 = set1.difference(set2)

print(difference\_set2) # Output: {1, 2}

4. Is Disjoint (isdisjoint()):-

Returns True if the sets have no common elements.

Example:

set1 = {1, 2, 3}

set2 = {4, 5, 6}

set3 = {2, 3, 4}

print(set1.isdisjoint(set2)) # Output: True (no common elements)

print(set1.isdisjoint(set3)) # Output: False (2 and 3 are common)

5. Is Superset (issuperset()):-

Returns True if the first set contains all elements of the second set.

Example:

set1 = {1, 2, 3, 4, 5}

set2 = {2, 3}

set3 = {6, 7}

print(set1.issuperset(set2)) # Output: True (set1 contains all elements of set2)

print(set1.issuperset(set3)) # Output: False (set1 does not contain 6 and 7)

6. Symmetric Difference (^ or symmetric\_difference()):-

Returns elements that are in either of the sets, but not both.

Removes the common elements.

Example:

set1 = {1, 2, 3, 4}

set2 = {3, 4, 5, 6}

# Using ^ operator

symmetric\_diff = set1 ^ set2

print(symmetric\_diff) # Output: {1, 2, 5, 6}

# Using symmetric\_difference() method

symmetric\_diff2 = set1.symmetric\_difference(set2)

print(symmetric\_diff2) # Output: {1, 2, 5, 6}

Summary Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Operation** | **Symbol** | **Method Equivalent** | **Description** |
| **Union** | ` | ` | union() |
| **Intersection** | & | intersection() | Returns common elements in both sets |
| **Difference** | - | difference() | Returns elements present in first set but not second |
| **Symmetric Difference** | ^ | symmetric\_difference() | Returns elements present in either set, but not both |
| **Is Disjoint** | N/A | isdisjoint() | Returns True if sets have no common elements |
| **Is Superset** | N/A | issuperset() | Returns True if first set contains all elements of second |