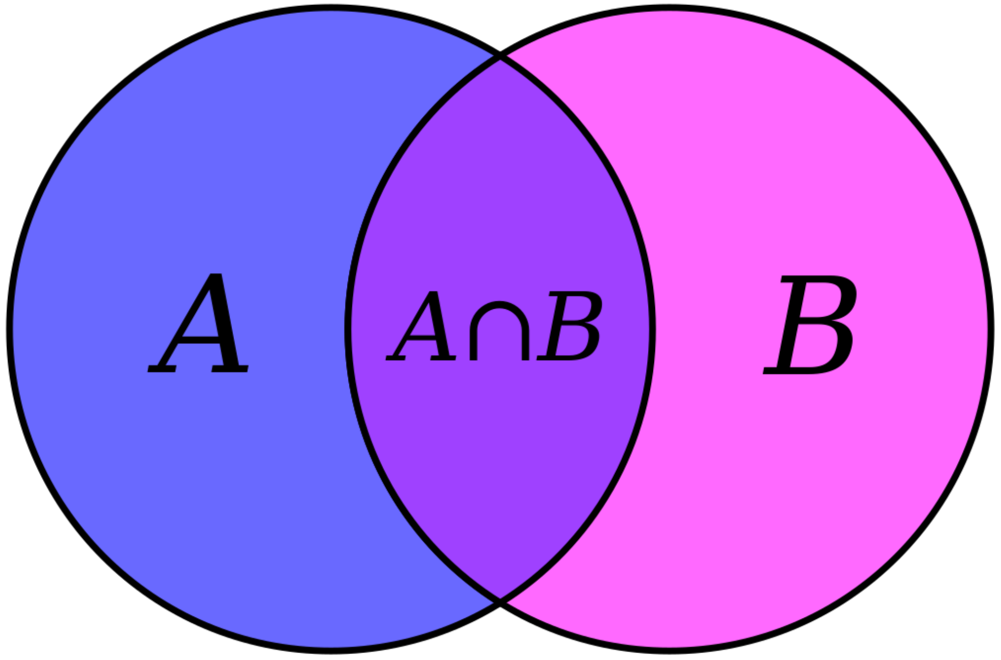
SETS

Built-In Functions



Built-in Functions

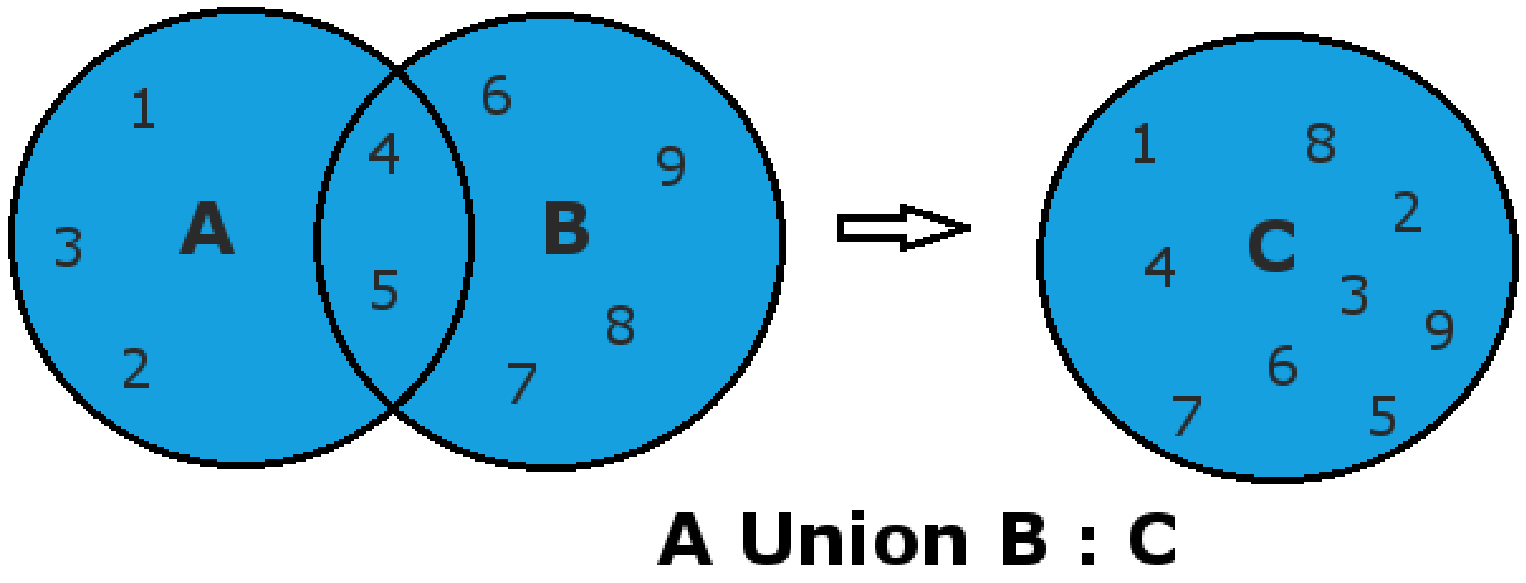
Method	Description
add()	Adds an element to the set
clear()	Removes all the elements from the set
copy()	Returns a copy of the set
difference()	Returns a set containing the difference between two or more sets
difference_update()	Removes the items in this set that are also included in another, specified set
discard()	Remove the specified item
intersection()	Returns a set, that is the intersection of two other sets
intersection_update()	Removes the items in this set that are not present in other, specified set(s)
isdisjoint()	Returns whether two sets have a intersection or not
issubset()	Returns whether another set contains this set or not
issuperset()	Returns whether this set contains another set or not
pop()	Removes an element from the set
remove()	Removes the specified element
symmetric_difference()	Returns a set with the symmetric differences of two sets
symmetric_difference_update()	inserts the symmetric differences from this set and another
union()	Return a set containing the union of sets
update()	Update the set with the union of this set and others

union()

SetA = set([1,2,3,4,5])

SetB = set([4,5,6,7,8,9])

SetA.union(SetB) #the .union() method in sets to update 2 sets but without repetition.

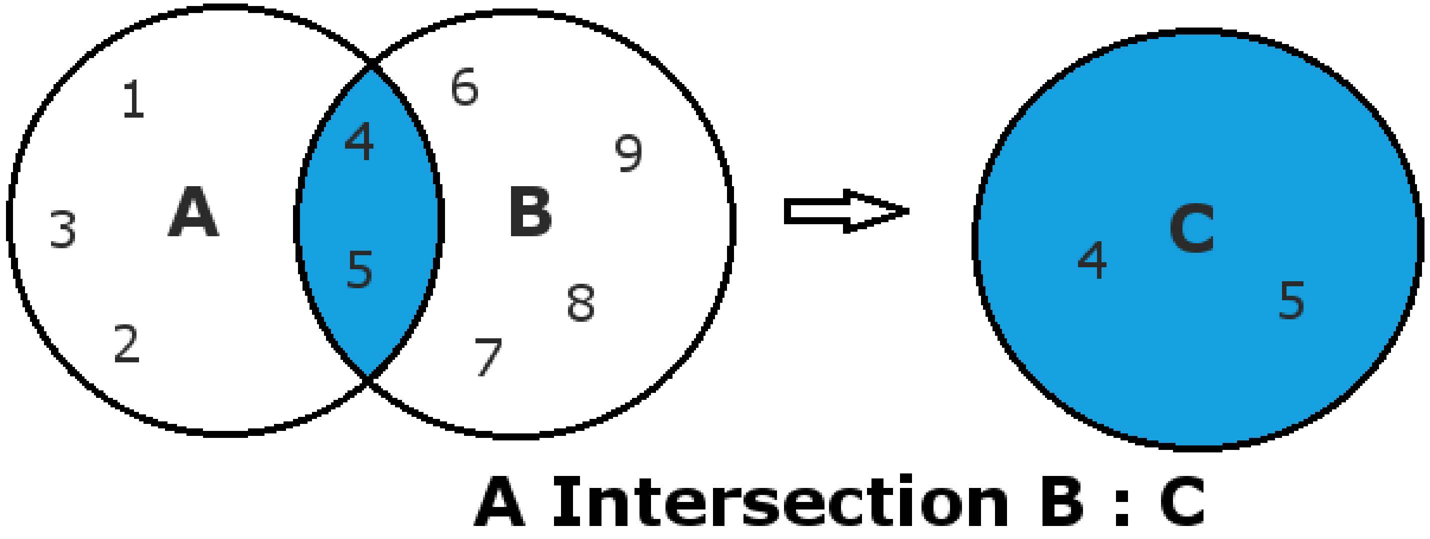


intersection()

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

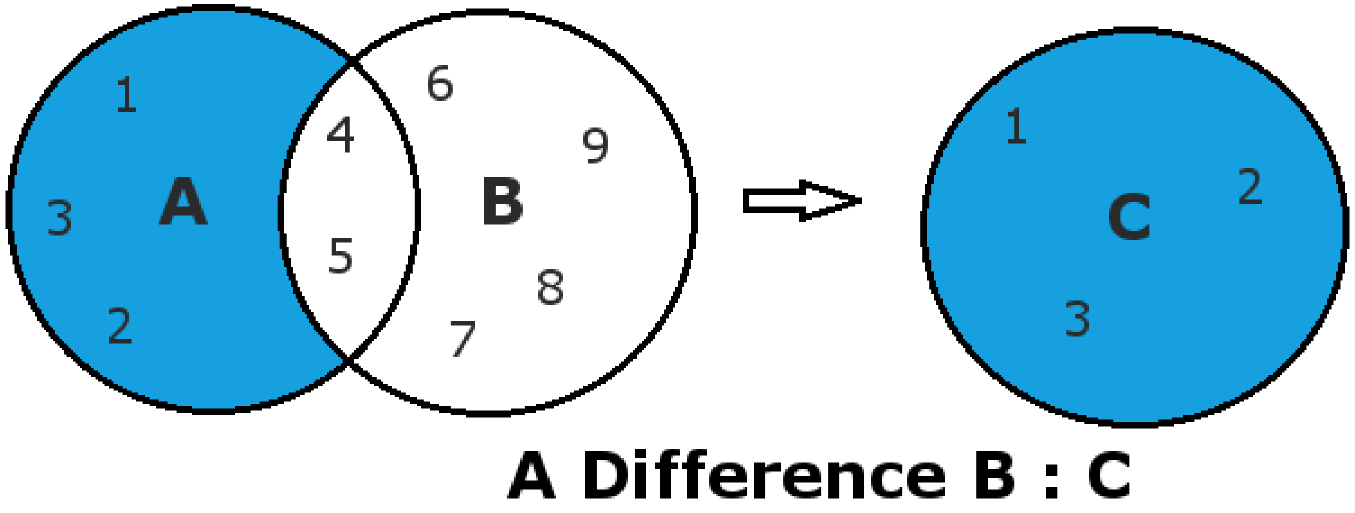
SetB = {4,5,6,7,8,9}

SetA.intersection(SetB) # the .intersection() method in sets to find Common elements in both sets.



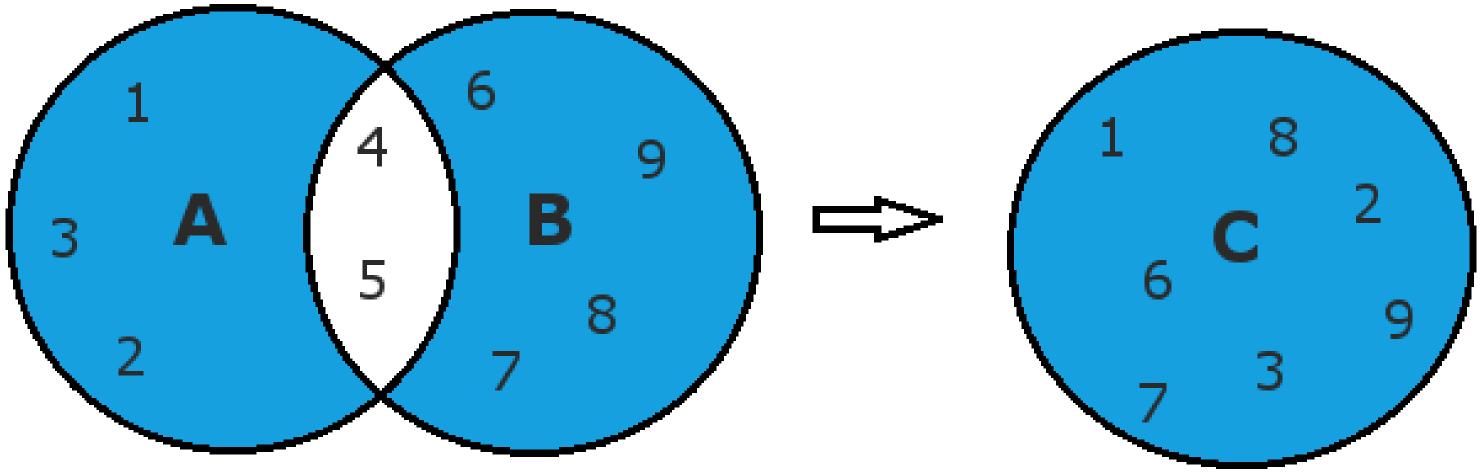
difference()

```
A={1,2,3,4,5}
B={4,5,6,7,8,9}

print(A.difference(B))        #the .difference() method in sets
print()
print(B.difference(A))
```



symmetric_difference()

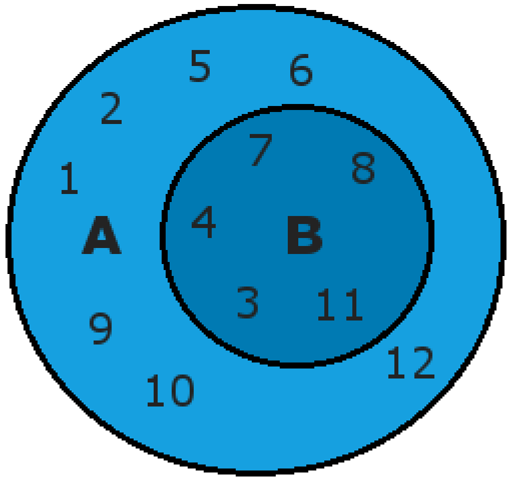
```
A = {1,2,3,4,5}
B = {4,5,6,7,8,9}

print(A.symmetric_difference(B)) #the .symmetric_difference method in sets.
print()
print(B.symmetric_difference(A))
```



A Symmetric Difference B : C

issubset()

```
A={1,2,3,4,5,6,7,8,9,10,11,12}
B={3,4,7,8,11}

print(A.issubset(B)) #returns (Boolean Value/True or False)
print()
print(B.issubset(A))  #returns True or False
```

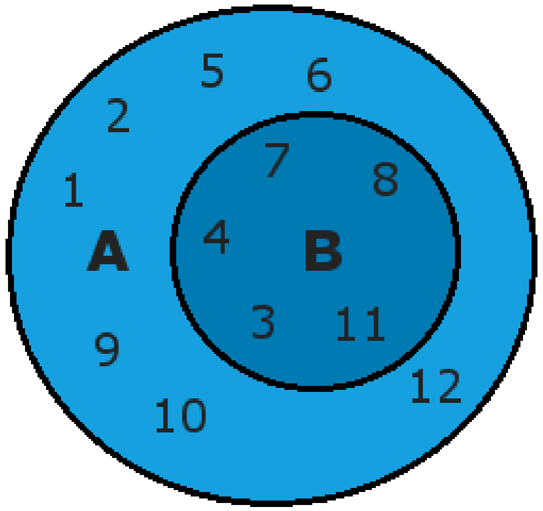


B is SubSet of A

issuperset()

A={1,2,3,4,5,6,7,8,9,10,11,12}
B={3,4,7,8,11}

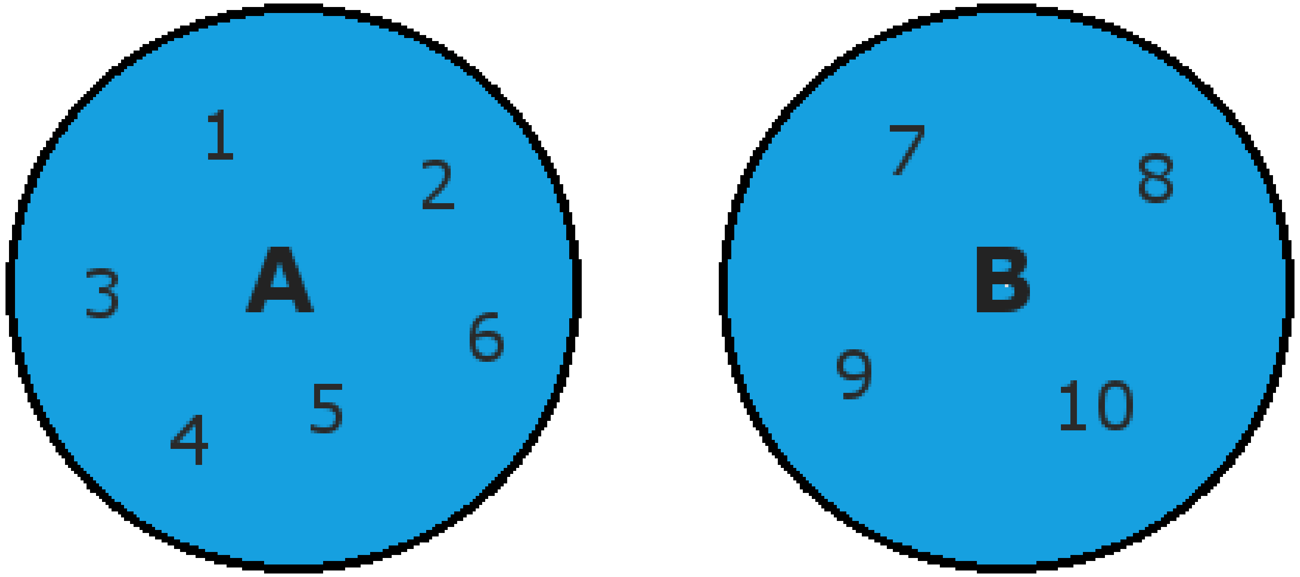
#print(A.issuperset(B))   #will return (Boolean Value/True or False)
print(B.issuperset(A))
returns: True. As A is the superset of B



A is SuperSet of B

isdisjoint()

```
A={1,2,3,4,5,6}
B={7,8,9,10}

print(A.isdisjoint(B))  #will return True or False.  True if there are no common elements in them.
```

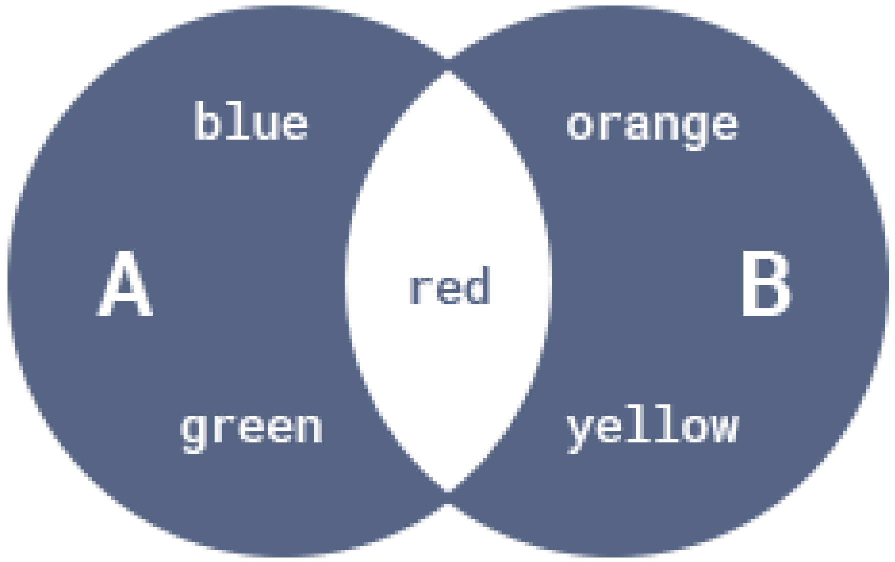


Disjoint Set

symmetric_difference_update

Updates the set by keeping only elements found in either set, but not in both.

```
# Update A by adding items from B, except common items
A = {'red', 'green', 'blue'}
B = {'yellow', 'red', 'orange'}

A.symmetric_difference_update(B)
print(A)
# Prints {'blue', 'orange', 'green', 'yellow'}
```



The End