Python Set

**There are four collection data types in the Python programming language:**

* [**List**](https://www.w3schools.com/python/python_lists.asp)**is a collection which is ordered and changeable. Allows duplicate members.**
* [**Tuple**](https://www.w3schools.com/python/python_tuples.asp)**is a collection which is ordered and unchangeable. Allows duplicate members.**
* **Set is a collection which is unordered, unchangeable\*, and unindexed. No duplicate members.**
* [**Dictionary**](https://www.w3schools.com/python/python_dictionaries.asp)**is a collection which is ordered\*\* and changeable. No duplicate members.**

**Q:Duplicate values will be ignored:**

thisset = {"apple", "banana", "cherry", "apple"}  
print(thisset)

**Q:True and 1 is considered the same value:**

thisset = {"apple", "banana", "cherry", True, 1, 2}  
print(thisset)

**Q:False and 0 is considered the same value:**

thisset = {"apple", "banana", "cherry", False, True, 0}  
print(thisset)

**Q:Using the set() constructor to make a set:**

thisset = set(("apple", "banana", "cherry")) # note the double round-brackets  
print(thisset)

**A. Add Set Items**

**Q:Add an item to a set, using the add() method:**

thisset = {"apple", "banana", "cherry"}  
thisset.add("orange")  
print(thisset)

**Q:Add elements from tropical into thisset:**

thisset = {"apple", "banana", "cherry"}  
tropical = {"pineapple", "mango", "papaya"}  
thisset.update(tropical)  
print(thisset)

**Q:Add elements of a list to at set:**

thisset = {"apple", "banana", "cherry"}  
mylist = ["kiwi", "orange"]  
thisset.update(mylist)  
print(thisset)

**B. Remove Item**

**Q:Remove "banana" by using the remove() method:**

thisset = {"apple", "banana", "cherry"}  
thisset.remove("banana")  
print(thisset)

**Q:Remove "banana" by using the discard() method:**

thisset = {"apple", "banana", "cherry"}  
thisset.discard("banana")  
print(thisset)

**Q:Remove a random item by using the pop() method:**

thisset = {"apple", "banana", "cherry"}  
x = thisset.pop()  
print(x)  
print(thisset)

**Q:The clear() method empties the set:**

thisset = {"apple", "banana", "cherry"}  
thisset.clear()  
print(thisset)

**Q:The del keyword will delete the set completely:**

thisset = {"apple", "banana", "cherry"}  
del thisset  
print(thisset)

**C. Join Sets**

**There are several ways to join two or more sets in Python.**

The union() and update() methods joins all items from both sets.

The intersection() method keeps ONLY the duplicates.

The difference() method keeps the items from the first set that are not in the other set(s).

The symmetric\_difference() method keeps all items EXCEPT the duplicates.

**Q:Join set1 and set2 into a new set:**

set1 = {"a", "b", "c"}  
set2 = {1, 2, 3}  
set3 = set1.union(set2)  
print(set3)

**Q:Use | to join two sets:**

set1 = {"a", "b", "c"}  
set2 = {1, 2, 3}  
set3 = set1 | set2  
print(set3)

**Q:Join multiple sets with the union() method:**

set1 = {"a", "b", "c"}  
set2 = {1, 2, 3}  
set3 = {"John", "Elena"}  
set4 = {"apple", "bananas", "cherry"}  
myset = set1.union(set2, set3, set4)  
print(myset)

**Q:Use | to join two sets:**

set1 = {"a", "b", "c"}  
set2 = {1, 2, 3}  
set3 = {"John", "Elena"}  
set4 = {"apple", "bananas", "cherry"}  
myset = set1 | set2 | set3 |set4  
print(myset)

**Q:Join a set with a tuple:**

x = {"a", "b", "c"}  
y = (1, 2, 3)  
z = x.union(y)  
print(z)

**Q:The update() method inserts the items in set2 into set1:**

set1 = {"a", "b" , "c"}  
set2 = {1, 2, 3}  
set1.update(set2)  
print(set1)

**Q:Join set1 and set2, but keep only the duplicates:**

set1 = {"apple", "banana", "cherry"}  
set2 = {"google", "microsoft", "apple"}  
set3 = set1.intersection(set2)  
print(set3)

**Q:Use & to join two sets:**

set1 = {"apple", "banana", "cherry"}  
set2 = {"google", "microsoft", "apple"}  
set3 = set1 & set2  
print(set3)

**Q:Keep the items that exist in both set1, and set2:**

set1 = {"apple", "banana", "cherry"}  
set2 = {"google", "microsoft", "apple"}  
set1.intersection\_update(set2)  
print(set1)

**Q:Join sets that contains the values True, False, 1, and 0, and see what is considered as duplicates:**

set1 = {"apple", 1,  "banana", 0, "cherry"}  
set2 = {False, "google", 1, "apple", 2, True}  
set3 = set1.intersection(set2)  
print(set3)

**Q: Keep all items from set1 that are not in set2:**

set1 = {"apple", "banana", "cherry"}  
set2 = {"google", "microsoft", "apple"}  
set3 = set1.difference(set2)  
print(set3)

**Q:Use - to join two sets:**

set1 = {"apple", "banana", "cherry"}  
set2 = {"google", "microsoft", "apple"}  
set3 = set1 - set2  
print(set3)

**Q:Use the difference\_update() method to keep the items that are not present in both sets:**

set1 = {"apple", "banana", "cherry"}  
set2 = {"google", "microsoft", "apple"}  
set1.difference\_update(set2)  
print(set1)

**Q:Keep the items that are not present in both sets:**

set1 = {"apple", "banana", "cherry"}  
set2 = {"google", "microsoft", "apple"}  
set3 = set1.symmetric\_difference(set2)  
print(set3)

**Q:Use ^ to join two sets:**

set1 = {"apple", "banana", "cherry"}  
set2 = {"google", "microsoft", "apple"}  
set3 = set1 ^ set2  
print(set3)

**Q:Use the symmetric\_difference\_update() method to keep the items that are not present in both sets:**

set1 = {"apple", "banana", "cherry"}  
set2 = {"google", "microsoft", "apple"}  
set1.symmetric\_difference\_update(set2)  
print(set1)