**Python Set Exercise**

Set in Python is an unordered collection of items. Every item is unique in it. i.e., the set doesn’t allow duplicates.

### **Exercise 1: Add a list of elements to a set**

Given a [Python list](https://pynative.com/python-lists/), Write a program to add all its elements into a given set.

**Given**:

sample\_set = {"Yellow", "Orange", "Black"}

sample\_list = ["Blue", "Green", "Red"]

**Expected output:**

**Note**: Set is unordered.

{'Green', 'Yellow', 'Black', 'Orange', 'Red', 'Blue'}

sample\_set = {"Yellow", "Orange", "Black"}  
sample\_list = ["Blue", "Green", "Red"]  
  
sample\_set.update(sample\_list)  
print(sample\_set)

**Exercise 2: Return a new set of identical items from two sets**

**Given**:

set1 = {10, 20, 30, 40, 50}

set2 = {30, 40, 50, 60, 70}

**Expected output:**

{40, 50, 30}

set1 = {10, 20, 30, 40, 50}  
set2 = {30, 40, 50, 60, 70}  
  
print(set1.intersection(set2))

### **Exercise 3: Get Only unique items from two sets**

Write a Python program to return a new set with unique items from both sets by removing duplicates.

**Given**:

set1 = {10, 20, 30, 40, 50}

set2 = {30, 40, 50, 60, 70}

**Expected output:**

{70, 40, 10, 50, 20, 60, 30}

**Note**: set is unordered, so not necessary this will be the order of the item.

set1 = {10, 20, 30, 40, 50}  
set2 = {30, 40, 50, 60, 70}  
  
print(set1.union(set2))

### **Exercise 4: Update the first set with items that don’t exist in the second set**

Given two Python sets, write a Python program to update the first set with items that exist only in the first set and not in the second set.

**Given**:

set1 = {10, 20, 30}

set2 = {20, 40, 50}

**Expected output:**

set1 {10, 30}

set1 = {10, 20, 30}  
set2 = {20, 40, 50}  
  
set1.difference\_update(set2)  
print(set1)

### **Exercise 5: Remove items from the set at once**

Write a Python program to remove items 10, 20, 30 from the following set at once.

**Given**:

set1 = {10, 20, 30, 40, 50}

**Expected output:**

{40, 50}

set1 = {10, 20, 30, 40, 50}  
set1.difference\_update({10, 20, 30})  
print(set1)

### **Exercise 6: Return a set of elements present in Set A or B, but not both**

**Given**:

set1 = {10, 20, 30, 40, 50}

set2 = {30, 40, 50, 60, 70}

**Expected output:**

{20, 70, 10, 60}

set1 = {10, 20, 30, 40, 50}  
set2 = {30, 40, 50, 60, 70}  
  
print(set1.symmetric\_difference(set2))

### **Exercise 7: Check if two sets have any elements in common. If yes, display the common elements**

**Given**:

set1 = {10, 20, 30, 40, 50}

set2 = {60, 70, 80, 90, 10}

**Expected output:**

Two sets have items in common

{10}

set1 = {10, 20, 30, 40, 50}  
set2 = {60, 70, 80, 90, 10}  
  
if set1.isdisjoint(set2):  
 print("Two sets have no items in common")  
else:  
 print("Two sets have items in common")  
 print(set1.intersection(set2))

### **Exercise 8: Update set1 by adding items from set2, except common items**

**Given**:

set1 = {10, 20, 30, 40, 50}

set2 = {30, 40, 50, 60, 70}

**Expected output:**

{70, 10, 20, 60}

set1 = {10, 20, 30, 40, 50}  
set2 = {30, 40, 50, 60, 70}  
  
set1.symmetric\_difference\_update(set2)  
print(set1)

### **Exercise 9: Remove items from set1 that are not common to both set1 and set2**

**Given**:

set1 = {10, 20, 30, 40, 50}

set2 = {30, 40, 50, 60, 70}

**Expected output:**

{40, 50, 30}

set1 = {10, 20, 30, 40, 50}  
set2 = {30, 40, 50, 60, 70}  
  
set1.intersection\_update(set2)  
print(set1)

# Python Set Quiz

1. What is the output of the following set operation.

set1 = {"Yellow", "Orange", "Black"}

set2 = {"Orange", "Blue", "Pink"}

set1.difference\_update(set2)

**print**(set1)

{‘Black’, ‘Yellow’}

2. What is the output of the following union operation

set1 = {10, 20, 30, 40}

set2 = {50, 20, "10", 60}

set3 = set1.union(set2)

**print**(set3)

{40, 10, ’10’, 50, 20, 60, 30}

3. What is the output of the following

sampleSet = {"Yellow", "Orange", "Black"}

sampleSet.add("Blue")

sampleSet.add("Orange")

**print**(sampleSet)

{‘Blue’, ‘Orange’, ‘Yellow’, ‘Black’}

4. Select all the correct options to remove “Orange” from the set.

sampleSet = {"Yellow", "Orange", "Black"}

sampleSet.discard("Orange")

5. What is the output of the following

set1 = {10, 20, 30, 40, 50}

set2 = {60, 70, 10, 30, 40, 80, 20, 50}

**print**(set1.issubset(set2))

**print**(set2.issuperset(set1))

True

True

6. What is the output of the following set operation

sampleSet = {"Yellow", "Orange", "Black"}

sampleSet.update(["Blue", "Green", "Red"])

**print**(sampleSet)

{‘Yellow’, ‘Orange’, ‘Red’, ‘Black’, ‘Green’, ‘Blue’}

7. What is the output of the following code

sampleSet = {"Yellow", "Orange", "Black"}

**print**(sampleSet[1])

Syntax Error

8. What is the output of the following code

aSet = {1, 'PYnative', ('abc', 'xyz'), True}

**print**(aSet)

{‘PYnative’, 1, (‘abc’, ‘xyz’)}

9. Select all the correct ways to copy two sets

* set2 = set1.copy()
* set2 = set(set1)
* set2.update(set1)

10. The isdisjoint() method returns True if none of the items are present in both sets, otherwise, it returns False.

True

11. What is the output of the following code

aSet = {1, 'PYnative', ['abc', 'xyz'], True}

**print**(aSet)

{1, ‘PYnative’, [‘abc’, ‘xyz’]}

12. The symmetric\_difference() method returns a set that contains all items from both sets, but not the items that are present in both sets.

 False

 True

13. The union() method returns a new set with all items from both sets by removing duplicates

 True

 False

14. What is the output of the following

sampleSet = {"Yellow", "Orange", "Black"}

sampleSet.discard("Blue")

**print**(sampleSet)

 {‘Yellow’, ‘Orange’, ‘Black’}

 KeyError: ‘Blue’

15. Select all which is true for Python set



* Sets are unordered
* set doesn’t allow duplicate
* sets are written with curly brackets {}



* set object does support indexing
* set is mutable

16. What is the output of the following set operation

set1 = {"Yellow", "Orange", "Black"}

set2 = {"Orange", "Blue", "Pink"}

set3 = set2.difference(set1)

**print**(set3)

* {‘Yellow’, ”Black’, ‘Pink’, ‘Blue’}
* **{‘Pink’, ‘Blue’}**
* {‘Yellow’, ”Black’}