

SETS IN PYTHON

Sets are used to store multiple items in a single variable. A set is a collection which is unordered, unchangeable*, and unindexed. Sets are written with curly brackets.

Set Items

Set items are unordered, unchangeable, and do not allow duplicate values.

Unordered

Unordered means that the items in a set do not have a defined order.

Set items can appear in a different order every time you use them, and cannot be referred to by index or key.

Unchangeable

Set items are unchangeable, meaning that we cannot change the items after the set has been create.

Duplicates Not Allowed

Sets cannot have two items with the same value.

Access Items

You cannot access items in a set by referring to an index or a key.

But you can loop through the set items using a for loop, or ask if a specified value is present in a set, by using the in keyword.

Add Items

To add one item to a set use the add() method.

Add Sets

To add items from another set into the current set, use the update() method.

Add Any Iterable

The object in the update() method does not have to be a set, it can be any iterable object (tuples, lists, dictionaries etc.).

Remove Item

To remove an item in a set, use the `remove()`, or the `discard()` method.

Loop Items

You can loop through the set items by using a for loop:

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.

PRACTICE QUESTIONS:

1. Create a set with the first five prime numbers.

```
[ ] a={2,3,5,7,11}  
    print(a)
```

⇌ {2, 3, 5, 7, 11}

2. Add the number 7 to the set.

```
[ ] a = {1,2,3,4,5,6}  
    a.add(7)  
    print(a)
```

⇌ {1, 2, 3, 4, 5, 6, 7}

3. Remove the number 3 from the set.

```
[ ] a={1,2,3,4,5,6,7}
    a.discard(3)
    print(a)
```

➞ {1, 2, 4, 5, 6, 7}

4. Check if the set is a subset of {2, 3, 5, 7, 11}

```
[ ] a={2,3,5,7,11,13}
    b={2,3,5,7,11}
    print(b.issubset(a))
```

➞ True

5. Find the union of the set with {7, 11, 13}.

```
[ ] a={2,3,5,7,11}
    b={7,11,13}
    print(a.union(b))
```

➞ {2, 3, 5, 7, 11, 13}

6. Create a frozen set from the original set.

```
[ ] prime={2,3,5,7,11}
    frozen= frozenset(prime)
    print(frozen)
```

➞ frozenset({2, 3, 5, 7, 11})

7. Check if the set has any common elements with {1, 4, 9}.

```
[ ] a = {1,2,3,4,5}
    b = {1,4,9}
    print(a.intersection(b))
```

↔ {1, 4}

8. Remove all elements from the set.

```
[ ] a = {1,2,3,4,5}
```

9. Create a set of your favorite fruits and find the intersection with {'apple', 'banana', 'orange'}

```
▶ my_fruit= {"apple", "banana", "cherry"}
  fruit={"apple", "banana", "orange"}
  print(my_fruit.intersection(fruit))
```

↔ {'apple', 'banana'}

10. Use set comprehension to create a set of squares for numbers 1 to 10

```
▶ squares = {x**2 for x in range(1,11)}
  print(squares)
```

↔ {64, 1, 4, 36, 100, 9, 16, 49, 81, 25}