SETS

12-2-2024

By Soofi Shafiya

SET

```
myset = {"apple", "banana", "cherry"}
```

A set is a collection which is *unordered*, *unchangeable**, and *unindexed*.

Note:

- Set items are unchangeable, but you can remove items and add new items.
- Sets are unordered, so you cannot be sure in which order the items will appear.

Set items

Unordered

- means that the items in a set do not have a defined order. They can appear in a different order every time 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 created.

Duplicates NOT allowed

Note: the values TRUE/FALSE and 1/0 and considered same and hence duplicates.

- Length of SET: len()
- Set items can be of any data type.
- A set can contain different data types.
- Type of SET: type()
- SET(): constructor.

Access Set Items

- We cannot access items in a set by referring to an index or a key.
- But we 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.

```
thisset = {"apple", "banana", "cherry"}

for x in thisset:
  print(x)
```

Change items

- Once a set is created, you can NOT change its items, but you can add new items. (Add Set Items)
- To add one item to a set, use the add() method.

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

Add Sets

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

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

print(thisset)
```

```
thisset = {"apple", "banana", "cherry"}
mylist = ["kiwi", "orange"]

thisset.update(mylist)

print(thisset)
```

Remove Item

- Remove() or discard() methods are used to remove an item from a set.
- If the item to remove does not exist remove() will raise an error and discard will NOT raise an error.
- We can also use the pop() method to remove an item, but this method will remove a random item, so you cannot be sure what item that gets removed.
- Clear() method empties a set.
- Del keyword will delete the set completely

Loop Sets

```
for x in thisset: #for loop
  print(x)
```

Join Sets

Union(): returns a net set

```
set3 = set1.union(set2)
```

Update():

```
set1.update(set2)
```

Both will exclude any duplicate items.

Intersection_update(): Keep ONLY the duplicates.

```
x.intersection_update(y) #Saved in x
```

Intersection(): It will return a new set .

```
z = x.intersection(y)
```

Keep All, But NOT the Duplicates

Symmetric_difference_update(): will keep only the elements that are NOT present in both sets.

```
x.symmetric_difference_update(y)
```

Symmetric_difference(): return a new set

```
z = x.symmetric_difference(y)
```

SET methods

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