PYTHON SETS

By
Hiba Al-Senawi

SET

- A set is a collection which is unordered and unindexed. In Python, sets are written with curly brackets.

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

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.

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

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

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

print("banana" in thisset)
```

CHANGE ITEMS

- Once a set is created, you cannot change its items, but you can add new items.

ADD ITEMS

- To add one item to a set use the add() method.
- To add more than one item to a set use the update() method.

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

thisset.add("orange")

print(thisset)
```

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

thisset.update(["orange", "mango", "grapes"])

print(thisset)
```

GET THE LENGTH OF A SET

- To determine how many items a set has, use the len() method.

REMOVE ITEM

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

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

thisset.remove("banana")

print(thisset)
```

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

thisset.discard("banana")

print(thisset)
```

REMOVE ITEM

- You can also use the **pop(),** method to remove an item, but this method will remove the last item. Remember that sets are unordered, so you will not know what item that gets removed.

- The return value of the pop() method is the removed item.

- The clear() method empties the set

- The del keyword will delete the set completely

JOIN TWO SETS

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

- You can use the union() method that returns a new set containing all items from both sets, or the update() method that inserts all the items from one set into another:

```
set1 = {"a", "b" , "c"}
set2 = {1, 2, 3}

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

```
set1 = {"a", "b" , "c"}
set2 = {1, 2, 3}

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

Both union() and update() will exclude any duplicate items.