Python Set

Set:-

A Python set is the collection of the unordered items. Each element in the set must be unique, and the sets remove the duplicate elements. Sets are mutable which means we can modify it after its creation.

Unlike other collections in Python, there is no index attached to the elements of the set, i.e., we cannot directly access any element of the set by the index. However, we can print them all together, or we can get the list of elements by looping through the set.

Creating a set:-

The set can be created by enclosing the comma-separated immutable items with the curly braces {}. Python also provides the set() method, which can be used to create the set by the passed sequence.

Example 1: Using curly braces

```
Days = {"Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"}
print(Days)
print(type(Days))
print("looping through the set elements ... ")
for i in Days:
   print(i)
```

Output:

Example 2: Using set() method

```
Days = set(["Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"])
print(Days)
print(type(Days))
print("looping through the set elements ... ")
for i in Days:
   print(i)
```

Output:

- It can contain any type of element such as integer, float, tuple etc. But mutable elements (list,dictionary) can't be a member of set. Consider the following example.

```
# Creating a set which have immutable elements
set1 = {1,2,3, " Python", 20.5, 14}
print(type(set1))
#Creating a set which have mutable element
set2 = {1,2,3,["Python",4]}
print(type(set2))
```

Output:

```
<class 'set'>

TypeError: unhashable type: 'list'
```

In the above code, we have created two sets, the set set1 have immutable elements and set2 have one mutable element as a list. While checking the type of set2, it raised an error, which means set can contain only immutable elements.

- Creating an empty set is a bit different because empty curly {} braces are also used to create a dictionary as well. So Python provides the set() method used without an argument to create an empty set.

```
# Empty curly braces will create dictionary
set3 = {}
print(type(set3))
# Empty set using set() function
set4 = set()
print(type(set4))
```

Output:

```
<class 'dict'>
<class 'set'>
```

- Let's see what happened if we provide the duplicate element to set5 .

```
set5 = {1,2,4,4,5,8,9,9,10}
print("Return set with unique elements:",set5)
```

Output:

```
Return set with unique elements: {1, 2, 4, 5, 8, 9, 10}
```

In the above code, we can see that set5 consisted of multiple duplicate elements when we printed it remove the duplicity from the set.

Access Python Set element:-

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.

Example

Loop through the set, and print the values:

```
thisset = {"PSIT", "CHE", "BCA"}
for x in thisset:
  print(x)
```