Python Sets

Python Sets

Python Set is a collection of items.

Python Set is Unordered

There is no order in which the items are stored in a Set. We cannot access items in a Set based on index.

Python Set has Unique Items

Python Set can store only unique items. We cannot add an item that is already present in the Set.

Python Set is Mutable

We can modify a Set. In other words, we can add items to a Set or remove items from a Set.

Create Python Set

To create a Set in Python, use curly braces `{}` as shown in the following example.

```
set_1 = {item_1, item_2, item_3}
```

Place as many items as required inside curly braces and assign it to a variable.

We can also use set() builtin function to create a Python Set. Pass an iterable to the set() function, and the function returns a Set created from the items in that iterable.

```
set_1 = set(iterable)
```

Example

In the following program, we create two sets: set_1 and set_2 using curly braces and set() function respectively.

Python Program

```
#create set using curly braces
set_1 = {2, 4, 6}
print(set_1)

#create set using set() builtin function
set_2 = set({'a', 'b', 'c'})
print(set_2)
```

Output

```
{2, 4, 6}
{'b', 'c', 'a'}
```

Access Items of Python Set

Python Set is an iterable. Therefore, we can access items in a Set using for loop or while loop.

Example

In the following program, we create a set with three items, and print items one at a time by accessing them using for loop.

Python Program

```
set_1 = {2, 4, 6}

for item in set_1:
    print(item)
```

Output

```
2
4
6
```

Add Items to Python Set

To add an item to a Python Set, call `add()` method on the set, and pass the item.

Example

In the following program, we create an empty set, and add three items to it using `add()` method.

Python Program

```
set_1 = set()
set_1.add(2)
set_1.add(4)
set_1.add(6)
print(set_1)
```

Output

```
{2, 4, 6}
```

Remove Items from Python Set

To remove an item from a Python Set, call `remove()` method on the set, and pass the item.

Example

In the following program, we create a set with three items, and delete two of them using `remove()` method.

Python Program

```
set_1 = set({2, 4, 6})
print(f'Set before removing: {set_1}')

set_1.remove(2)
set_1.remove(6)
print(f'Set after  removing: {set_1}')
```

Output

```
Set before removing: {2, 4, 6}
Set after  removing: {4}
```

Conclusion

In this Python Tutorial, we learned what Python Sets are, how to create them, how to add elements to a set, how to remove elements from a set, and methods of Python Set.