

# Course Name: Python Programming with Django

Name: Bipul Dutta

Batch ID: 13328

Order No: 4099423

---

Assignment- 04.

Date: 04.04.22

---

**\*\* Make a note on List, Tuple, Set.**

## **What is Python List:**

**List** is using for store multiple data in single variable. A list can contain mixer objects. For example: ['Python', 'JavaScript', 'C++']

## **How to create a List in Python:**

To create python list of item, you need to mention the items, separated by commas, in square brackets. This is the python syntax you need to follow. Then assign it to a variable and declare the data type, because Python is dynamically-typed. Example:

```
My_list = ['red', 'green', 'blue']
```

## **Python List Methods:**

Python has many useful List methods that make it really easy to work with list. Some of the commonly used list method.

| Methods   | Descriptions   |
|-----------|--|
| Append()  | Adds an element to the end of the list                         |
| Insert()  | Insert an item at the defined index                            |
| Remove()  | Removes an item from the list                                  |
| Pop()     | Returns and removes an element at the given index              |
| Clear()   | Removes all items from the list                                |
| Index()   | Returns the index of the first matched item                    |
| Count()   | Returns the count of the number of items passed as an argument |
| Reverse() | Reverse the order of items in the list                         |

## **How to access Python List:**

There are various ways to access the elements of a List. Example given bellow:

```
My_list = ['A', 'B', 'C', 'D', 'E']
```

```
# first item:
```

```
Print(my_list[0]) #output: ['A']
```

```
# last item:
```

```
Print(my_list[4]) #output: ['E']
```

```
# add item in list:
```

```
My_list.append('F') #output: ['A', 'B', 'C', 'D', 'E', 'F']
```

```
# remove item in list:
```

```
My_list.remove('B') #output: ['A', 'C', 'D', 'E']
```

-----0000000000000000-----

## **What is Tuple:**

Python Tuple are like a list. It can hold a sequence of items. The difference is that it is immutable. Tuple is one of four built-in data types in python use to store collections of data, the other three are List, Set and Dictionary.

## **Creating a Tuple:**

A tuple is created by placing all the items(elements) inside parentheses(), separated by commas. There are some examples:

```
my_tuple = ("apple", "banana", "mango")
```

```
Print(my_tuple)
```

## **Python Tuple Functions:**

A lot of functions that work on list and work on tuples too. A function applies on a construct and returns a result. It does not modify the construct.

Example:

```
len() =
```

```
max()
```

min()

sum()

sorted()

### **Basic Tuple Operations:**

The operators like concatenation (+), repetition (\*), Membership (in) work in the same way as they work with the list.

| Operator      | Description  | Example                                     |
|---------------|--|---|
| Repetition    | The repetition operator enables the tuple elements to be repeated multiple times | T1*2 = (1,2,3,4,5,1,2,3,4,5)                |
| Concatenation | It concatenates the tuple mentioned on either side of the operator               | T1+T2 = (1,2,3,4,5,6,7,8,9)                 |
| Membership    | It returns true if a particular item exists in the tuple otherwise false.        | Print (2 in T1) print True                  |
| Iteration     | The for loop is used to iterate over the tuple elements.                         | For I in T1:<br>Print(i) #output: 1 2 3 4 5 |
| Length        | It is used to get the length of the tuple  | Len (T1) = 5                                |

-----000000000000000-----

### **Python Sets:**

A Python set is the collection of the unordered items. Each element in the set must be unique, immutable, 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:

```
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)
```

## **Python Set Operations:**

Set can be performed mathematical operation such as union, intersection, difference, and symmetric difference. Python provides the facility to carry out these operations with operators or methods. We describe these operations as follows.

### **Union of two Sets**

The union of two sets is calculated by using the pipe (|) operator. The union of the two sets contains all the items that are present in both the sets.

Example:

```
Days1 = {"Monday", "Tuesday", "Wednesday", "Thursday", "Sunday"}
```

```
Days2 = {"Friday", "Saturday", "Sunday"}
```

```
print(Days1|Days2) #printing the union of the sets
```

### **Intersection of two sets**

The intersection of two sets can be performed by the and & operator or the intersection() function. The intersection of the two sets is given as the set of the elements that common in both sets.

Example:

```
Days1 = {"Monday", "Tuesday", "Wednesday", "Thursday"}
```

```
Days2 = {"Monday", "Tuesday", "Sunday", "Friday"}
```

```
print(Days1&Days2) #prints the intersection of the two sets
```