



## \* Experiment No. 2 \*

\* Title:- Write a program to demonstrate use of different datatypes & statements in Python.

### \* Objective:-

- To understand various datatypes and it's syntax.
- To understand various types of statement (conditional, loop, control).

### \* Theory :-

- Basic fundamentals of datatypes.

Type	Data Type
1) text type	str
2) numeric type	int, float, complex
3) sequence type	list, tuple, range
4) mapping type	dictionary (dict)
5) set type	set, frozenset
6) boolean type	bool
7) Binary type	bytes, bytearray, memoryview.
8) None type	None type.



- characteristic, operation & syntax of List, Tuple, Dictionary, set.

### (1) List :-

- Characteristics:

A list is a collection of ordered and mutable elements.

- Operations:

- Indexing: accessing individual elements using their position in the list.

- Slicing: accessing a subset of

- slicing: accessing individual elements using + elements using a range of indices.

- Appending: adding new elements to the end of the list.

- Extending: adding multiple elements to the end of the list.

- Inserting: adding a new element at a specific position in the list.

- Removing: deleting an element from the list by value or index.

- Syntax:-

my\_list = [element 1, element 2, element 3]



## (2) Tuple :-

- Characteristics :-

A tuple is a collection of ordered and immutable elements.

- Operations :-

- Indexing: accessing individual elements using their position in the tuple.

- Slicing: accessing a subset of elements using a range of indices.

- Syntax :-

```
my_tuple = (element1, element2, element3)
```

## (3) Dictionary :-

- Characteristics :-

A dictionary is a collection of unordered key - value pairs.

- Operations :-

- Accessing values: accessing the value associated with special key.

- Adding new key-value pairs.

- Updating values: changing the value associated with specific key.

- Removing key-value pairs by key or using pop() method.



- Syntax :-

my\_dict = {key1:value1, key2:value2,  
             key3:value3}

#### (4) Set :-

- Characteristics :-

A set is a collection of unordered and unique elements.

- Operations :-

- Adding elements by value using add() method.
- Removing elements by value or using discard() method.
- Union: combining two sets into one set with no duplicates.
- Intersection: finding common elements between two sets.

- Syntax :-

my\_set = {element1, element2, element3}



\* Program:-

```
# declaring variables of different datatypes
my-list = [1, 2, 3, 4, 5]
my-tuple = ('apple', 'banana', 'cherry')
my-dict = {'name': 'John', 'age': 25,
           'gender': 'male'}
my-set = {1, 2, 3, 4, 5}
```

# printing the values of variables

```
print("List:", my-list)
print("Tuple:", my-tuple)
print("Dictionary:", my-dict)
print("Set:", my-set)
```

# using conditional statement

```
if len(my-list) > len(my-set):
    print("List has more elements than Set")
else:
    print("Set has more elements than List")
```

# using loop statement

```
for fruit in my-tuple:
    print(fruit)
```

# using control statement

```
sum=0
for num in my-set:
    if num % 2 == 0
        continue
```



sum = sum  
print("Sum of odd numbers in set: ", sum)

In this program, we have declared variables of different datatypes - list, tuple, dictionary, and set. We have printed the values of these variables using the print() function.

We have also used a conditional statement (if-else) to check which datatype has more elements. We have used a loop statement (for loop) to print the elements of the tuple. Finally, we have used a control statement (continue) to skip even numbers and calculate the sum of odd numbers in set.

#### ● Difference between List, Tuple, Set, Dictionary:-

List	Tuple	Set	Dictionary
-List can be represented by [ ]	-Tuple can be represented by ()	-Set can be represented by { }	-Dictionary can be represented by {key: value}
-Allows duplicate value	-Allows duplicate value	-Not allow duplicate value	-Not allow duplicate value.
-mutable	-immutable	-mutable	- mutable.
-It is ordered	-It is ordered.	-It is unorder- ed.	- It is ordered .



- create using list()	- create using tuple()	- create using set()	- create using dict()
- Example: [1, 2, 3, 4]	- Example: (1, 2, 3, 4)	- Example: {1, 2, 3, 4}	- Example: {'K1': 1, 'K2': 2}

### \* Conclusion :-

Python is flexible programming language that provides support for various datatypes and statements. The program demonstrated how these datatypes and statement can be used to perform different operations and control program flow. It is important for programmers to understand and practice using these concepts to write effective programs.