

## Experiment 3 b)

**Aim: Indexing of Strings, List, Tuple, Set and Dictionary in Python.**

### Theory:

#### Indexing:

- Indexing is a method to search for a given element from the start of the list and returns the index position of the element.
- It is used to directly access the elements of choice and do various operations.
- In Python, objects are “zero-indexed” i.e. the position count of first element (i.e. the leftmost element) starts at zero.
- Index operator is represented by opening and closing square brackets [ ].

#### Index() :

- It is the built-in function that:
  - ✓ finds the first occurrence of the specified element.
  - ✓ raises an exception if the element is not found.
- **Set items cannot be accessed by referring to an index, since sets are unordered and the items has no index.**

#### #Indexing of String

```
txt = "Hello, welcome to my world."  
x = txt.index("e", 5, 10)  
print(x)
```

#-----OR-----

```
txt = "Hello, World!"  
print(txt[0])  
print(txt [5])
```

#### #Indexing of List

```
List = ["A", "B", "C"]  
print("Accessing a element from the list")  
print(List[0])  
print(List[2])
```

#### #Indexing of Tuple

```
thistuple = (1, 3, 7, 8, 7, 5, 4, 6, 8, 5)  
x = thistuple.index(8)  
print(x)
```

#### #Indexing of Dictionary

```
thisdict = {  
    "brand": "Ford",  
    "model": "Mustang",  
    "year": 1964  
}  
x = thisdict["model"]  
print(x)
```

#### Output:

8

H  
,

Accessing a element from the  
list  
A  
C

3

Mustang

### Conclusion:

Hence implemented Indexing operation on Strings, List, Tuple and Dictionary.