Experiment 3 d)

Aim: Slicing operation on Strings, List, Tuple, Set and Dictionary in Python.

Theory:

Slicing:

- Slicing is the process of accessing a particular piece of a sequence.
- It allows to access parts of sequences like strings, tuples, and lists. These parts left after after slicing are known as slices.
- Slicing iare non-destructive i.e. accessing a slice doesnotchange the original value of the sequence.
- Slicing can be done using 2 methods:
  - ✓ Using the **slice() function**
  - ✓ Using array slicing [: :]method
- a[start:stop:step] is equivalent to: a[slice(start, stop, step)]

1) Using the slice() function

✓ The slice() creates a slice object usig set of indices specified by slice(start, stop, step).

Syntax:

- slice(stop)
- slice(start, stop, step)
- **start:** Starting index where the slicing of object starts.
- **stop:** Ending index where the slicing of object stops.
- **step:** It is an optional argument that determines the increment between each index for slicing.
- **Return Type:** Returns a sliced object containing elements in the given range only.

2) Using array slicing [: :] method

Indexing using array slicing method have the same mechanism as the slice() i.e. [start : stop : step]

Syntax

```
arr[start:stop]        #all items from start to stop-1

arr[start:]            #all items from start to end of the complete array

arr[:stop]             #all items from the beginning to stop

arr[:]                 # a copy of the whole array

arr[::]                #copy of whole array

arr[start:stop:step]   # start to stop, by step
```

The other feature is that start or stop may be a *negative* number, which means it counts from the end of the array up to the beginning

```
a[-1]    # las t item in the array
a[-2:]   # last two items in the array
a[:-2]   # everything except the last two items
```

Similarly, step may be a negative number:

```
a[::-1]    # all items in the array, reversed
a[1::-1]   # the first two items, reversed
a[:-3:-1]  # the last two items, reversed
a[-3::-1]  # everything except the last two items, reversed
```

.

String Slicing

```
# String slicing
a = 'ASTRING'

# Using slice()
s1 = slice(3)
s2 = slice(1, 5, 2)
s3 = slice(-1, -12, -2)

print(a[s1])
print(a[s2])
print(a[s3])
```

Output

```
AST
SR
GITA
```

```
# String slicing
a = 'PYTHONFORDATASCIENCE'
print(a[:3])
print(a[-1:-12:-2])

# Prints string in reverse
print(a[::-1])
```

Output:

```
PYT
ENISTD
ECNEICSATADROFNOHTYP
```

List Slicing

```
# Initialize list
Lst = [50, 70, 30, 20, 90, 10, 50]

print(Lst[::])
print(Lst[-7::1])
```

Output:

```
[50, 70, 30, 20, 90, 10, 50]

[50, 70, 30, 20, 90, 10, 50]
```

```
# Initialize list
L = [1, 2, 3, 4, 5, 6, 7, 8, 9]

print("\nOriginal List:\n", L)

print("\nSliced Lists: ")
print(L[3:9:2])
print(L[::2])
print(L[::])
```

Output:

```
Original List:

 [1, 2, 3, 4, 5, 6, 7, 8, 9]

Sliced Lists:
[4, 6, 8]
[1, 3, 5, 7, 9]
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
# Show original list
print("\nOriginal List:\n", L)

print("\nSliced Lists: ")
print(L[::-1])
print(L[::-3])
print(L[:1:-2])
```

Output:

```
Original List:
 [1, 2, 3, 4, 5, 6, 7, 8, 9]

Sliced Lists:
[9, 8, 7, 6, 5, 4, 3, 2, 1]
[9, 6, 3]
[9, 7, 5, 3]
```

Tuple Slicing

```
tup = (22, 3, 45, 4, 2.4, 2, 56, 890, 1)

print(tup[1:4])
# prints 2nd to 4th element

print(tup[:4])
# prints 1st to 4th element

print(tup[4:])
# prints 5th to the last element

print(tup[:])
# prints first to the last element

print(tup[::2])
# prints first to the last element with a step of 2 i.e. printing the alternate elements

print(tup[-4])
# prints the fourth element from the last

print(tup[-4:-1])
# prints elements from last fourth to last second

print(tup[-1:-4])
# not applicable
```

Output:

```
(3, 45, 4)
(22, 3, 45, 4)
(2.4, 2, 56, 890, 1)
(22, 3, 45, 4, 2.4, 2, 56, 890, 1)
(22, 45, 2.4, 56, 1)
2
(2, 56, 890)
()
```

Set Slicing and Dictionary slicing

Since both Set and Dictionary are unordered slicing operation cannot be implemented using slice() or array slicing method.

Conclusion:

Hence implemented Slicing operation on Strings, List and Tuple.