Name:- Divyang Bagla

ERP ID:- 1032180739

Batch A2

Roll No. :- PC33

Lab Assignment No. 3

Aim:-

Write a Python Program to implement following concepts

A. List: 1. List Creation 2. Length 3. Append and Extend 4. Remove 5. Delete 6. Reverse 7. Sort 8. Indexing 9. Slicing

B. Tuple: 1.Tuple Creation 2. Length 3. Delete 4. Count 5. Delete 6. Membership 7. Sort

Theory: -

List

Definition: Lists are used to store multiple items in a single variable.

Lists are one of 4 built-in data types in Python used to store collections of data, the other 3 are Tuple, Set, and Dictionary, all with different qualities and usage.

Lists are created using square brackets []

List Features:

1. Ordered
2. Mutable
3. Duplicates Allowed

Tuples

Definition: Tuples are used to store multiple items in a single variable.

Tuple is one of 4 built-in data types in Python used to store collections of data, the other 3 are List, Set, and Dictionary, all with different qualities and usage.

Tuples are created using parentheses ()

Tuple Features:

1. Ordered
2. Immutable
3. Duplicates Allowed

Lab Assignment 3 Code

Lab Assignment 3

List

In [1]:
```
#List Creation
my_list = []

my_list = [1, 2, 3] # list of integers

my_list1 = [1, "Hello", 3.4] # list with mixed data types

my_list2 = ["mouse", [8, 4, 6], ['a']] # nested list
```

In [2]:
```
#List indexing
my_list = [1,2,3,4,5,6,7,8,9,10]

print(my_list[0])

print(my_list[2])

print(my_list[7])

print(my_list[-1])

#Nested List Indexing
n_list = ["Happy", [2, 0, 1, 5]]

print(n_list[0])

print(n_list[0][1])

print(n_list[1][-2])

#print(my_list[4.0]) #Erroneous code as only integer index is allowed
```

```
1
3
8
10
Happy
a
1
```

In [3]:
```
#List editing
#Correcting mistake values in a list
my_list = [1, 4, 6, 8]

print(my_list)

# change the 1st item
my_list[0] = 7

print(my_list)

# change 2nd to 4th items
my_list[1:4] = [3, 5, 7]

print(my_list)
```

```
[1, 4, 6, 8]
```

```
[7, 4, 6, 8]
[7, 3, 5, 7]
```

In [4]:

```
#List Append, Extend and Concatenate
my_list = [1, 2, 5]

my_list.append(7) #Append

print(my_list)

my_list.append([1,2,3,4,5])

my_list.extend([9, 11, 13]) #Extend

print(my_list)

print(my_list + [9, 7, 5]) #Concatenate using +

print(['mylist'] * 3)
```

```
[1, 2, 5, 7]
[1, 2, 5, 7, [1, 2, 3, 4, 5], 9, 11, 13]
[1, 2, 5, 7, [1, 2, 3, 4, 5], 9, 11, 13, 9, 7, 5]
['mylist', 'mylist', 'mylist']
```

In [5]:

```
#insert() in list
odd = [1,3,4,5,6]
odd.insert(1,2)
print(odd)

odd[6:7] = [7, 8]
print(odd)
```

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

In [6]:

```
#Delete, Remove, Pop, Clear
my_list = [1,2,3,4,5,6,7,8,9,10]

del my_list[2] #Delete one list item

print(my_list)

del my_list[1:5] #Delete multiple list items

print(my_list)

#del my_list #Delete the entire list
#print(my_list) #Erroneous code as the list is deleted

print('------------')

my_list = ['d','i','v','y','a','n','g']

my_list.remove('i')

print(my_list)

print(my_list.pop(1))

print(my_list)

print(my_list.pop())
```

```
print(my_list)

my_list.clear()

print(my_list)
```

```
[1, 2, 4, 5, 6, 7, 8, 9, 10]
[1, 7, 8, 9, 10]
------------
['d', 'v', 'y', 'a', 'n', 'g']
v
['d', 'y', 'a', 'n', 'g']
g
['d', 'y', 'a', 'n']
[]
```

In [7]:

```
#Sort, Count, Reverse
my_list = [2, 7, 5, 8, 1, 6, 0, 8, 4]

print(my_list.index(8))

print(my_list.count(8))

my_list.sort()

print(my_list)

my_list.sort(reverse=True)

print(my_list)

my_list.reverse()

print(my_list)
```

```
3
2
[0, 1, 2, 4, 5, 6, 7, 8, 8]
[8, 8, 7, 6, 5, 4, 2, 1, 0]
[0, 1, 2, 4, 5, 6, 7, 8, 8]
```

In [8]:

```
#List slicing in Python

my_list = ['p','y','t','h','o','n']

# elements 3rd to 5th
print(my_list[2:5])

# elements beginning to 4th
print(my_list[:-5])

# elements 6th to end
print(my_list[5:])

# elements beginning to end
print(my_list[:])
```

```
['t', 'h', 'o']
['p']
['n']
['p', 'y', 't', 'h', 'o', 'n']
```

Tuples

In [9]:
```
my_tuple = ('Divyang','Bagla', 'Python', 1, 2, 3, 4, 5);

print(my_tuple)
```

```
('Divyang', 'Bagla', 'Python', 1, 2, 3, 4, 5)
```

In [10]:
```
#To write a tuple containing a single value you have to include a comma, even though
tup1 = (50,);
print(tup1)
```

```
(50,)
```

In [11]:
```
#Accesing Values in a Tuple
print(my_tuple[0])
print(my_tuple[1][2])
```

```
Divyang
g
```

In [12]:
```
#Update in Tuple
my_tuple = ('Divyang','Bagla', 'Python', 1, 2, 3, 4, 5);
tup1 = (1, 2, 3, 'abc', 'xyz');

#tup1[0] = 100; #Erroneous code since updation is not valid for tuples

# So let's create a new tuple as follows
tup3 = tup1 + my_tuple
print(tup3)
```

```
(1, 2, 3, 'abc', 'xyz', 'Divyang', 'Bagla', 'Python', 1, 2, 3, 4, 5)
```

In [14]:
```
#Delete and Remove
my_tuple = ('Divyang', 'Python', 1, 2, 3, 4, 5)
print(my_tuple)
del my_tuple
#print('After deleting my_tuple:')
#print(my_tuple) #Erroneous code since tuple is deleted

my_tuple1= (1,2,3,4,5,6,7)
#my_tuple1.remove(2) #Erroneous code since tuple is immutable
```

```
('Divyang', 'Python', 1, 2, 3, 4, 5)
```

In [15]:
```
#Length of Tuple
my_tuple = ('Divyang','Bagla', 'Python', 1, 2, 3, 4, 5)
len(my_tuple)
```

Out[15]:
```
8
```

In [16]:
```
#Sort and Count
my_tuple = (1, 5, 7, 1, 9, 10, 2, 6)
#my_tuple.sort() #erroneous since sort() method doesnt work with immutable data type
print(sorted(my_tuple)) #will return sorted list not a tuple
print(my_tuple.count(1)) #will count number of 1's in the tuple
```

```
[1, 1, 2, 5, 6, 7, 9, 10]
2
```

In [18]:
```
#Membership test in tuple
my_tuple = ('d','i','v','y','a','n','g')
```

```
# In operation
print('a' in my_tuple)
print('b' in my_tuple)

# Not in operation
print('g' not in my_tuple)
```

```
True
False
False
```

In [ ]: