Pranav Polavarapu - 19BTRCR008

6.Write python program: ¶

a.To elaborate lists with the operations such as creating, navigating, slicing, adding or removing elements, etc.

b.To elaborate tuples with the operations as in above assignment.

c.To elaborate dictionaries with the operations such as given above.

6(a):Lists:

Creating a list

Traversing a list

Slicing

Methods used in lists:

insert(),append(),extend(),pop(),remove(),clear(),del(),index() copy() [check for both aliasing and cloning],sort(),reverse()

In [2]:
```
l = ["a",'b',1,True,'Pranav','Polavarapu']
print('Navigating 2nd Index value - {} and 5th Index value - {}'.format(l[2],l
[5]))


#Traversing
print("----Traversing a list----")
for i in l:
    print(i)
print("----End of Traversing----")


#slicing
print('Slicing from 2nd Index to 6th Index- {}'.format(l[2:6]))


#adding and removing element
l.insert(7,10)
print('Inserted a value 10 in Index 7 - {} '.format(l))
l.remove(1)
print('Removed a value 1 in defined List - {} '.format(l))
s=['c','d']
l.extend(s)
print('Extented list named l with list named s - {} '.format(l))
l.pop()
print('pop() will remove the last element in the List - {}'.format(l))
del(s)  #deletes the list s
print('Index value of Element Pranav - {}'.format(l.index('Pranav')))
n=['me','you',"us"]
print(f'List named n {n}')
print('clear() function empties the List - {}'.format(n.clear()))
v=['a','b']
b=v.copy()#aliasing
print(v,b)
print('list b copies v - {}'.format(b))
print('Id of v {} and Id of b {} '.format(id(v),id(b)))
x=b #cloning
print('Id of x {} and Id of b {} '.format(id(x),id(b)))
u = [3,7,1,5,6,]
u.sort()
print(f'sorting a list - {u}')
u.reverse()
print(f'Reversing a list - {u}')
print(f'Minimum value {min(u)}')
print(f'Minimum value {max(u)}')
```

```
Navigating 2nd Index value - 1 and 5th Index value - Polavarapu
----Traversing a list----
a
b
1
True
Pranav
Polavarapu
----End of Traversing----
Slicing from 2nd Index to 6th Index- [1, True, 'Pranav', 'Polavarapu']
Inserted a value 10 in Index 7 - ['a', 'b', 1, True, 'Pranav', 'Polavarapu',
10]
Removed a value 1 in defined List - ['a', 'b', True, 'Pranav', 'Polavarapu',
10]
Extented list named l with list named s - ['a', 'b', True, 'Pranav', 'Polavar
apu', 10, 'c', 'd']
pop() will remove the last element in the List - ['a', 'b', True, 'Pranav',
'Polavarapu', 10, 'c']
Index value of Element Pranav - 3
List named n ['me', 'you', 'us']
clear() function empties the List - None
['a', 'b'] ['a', 'b']
list b copies v - ['a', 'b']
Id of v 1598431586952 and Id of b 1598449167368
Id of x 1598449167368 and Id of b 1598449167368
sorting a list - [1, 3, 5, 6, 7]
Reversing a list - [7, 6, 5, 3, 1]
Minimum value 1
Minimum value 7
```

6(b):Tuples: Creating tuple (Also check for creating tuple with one item)

Accessing, Traversing, Updating

Concatenation, repetition, membership operations

index() and count() methods max(),min(),len() functions deleting tuple

In [3]:
```
t = ('a','b',5,True)
m = ('r')
print(t,m)
print("Accessing the a element from tuple 't' - {}".format(t[1]))
#traversing
print("----Traversing a tuple----")
for i in t:
    print(i)
print("----End of Traversing----")
#Updating --Tuples are immutable which means you cannot update or change the v
alues of tuple elements.
a=(2,3,4)
i= t+a #concatination
print('New tuple - {}'.format(i))
del a #deletes tuple named a
#membership operations
print("----Membership Operation----")
list1=[1,2,3,4,5]
list2=[6,7,8,9]
for item in list1:
    if item in list2:
        print("overlapping")
else:
    print("not overlapping")
print("----End of Membership Operation----")
#repetition
a=(2,3,4)
print(f'Tuple "a" - {a}')
print('Repetiton of tuple - {}'.format(a*3))
#index
j=("a",'b',1,True,'Pranav','Python')
print('tuple j - {}'.format(j))
print(f'Index of value "1"- {j.index(1)}')
#count
print('Length of tuple - {}'.format(len(j)))
# min max
k =(5,8,4,9)
print('Minimum value - {}'.format(min(k)))
print('Maximum value - {}'.format(max(k)))
#count
l=(1,1,1,4,5,7,9,1,5,7)
print('counting a value "5" in a tuple "l" - {}'.format(l.count(5)))
```

```
('a', 'b', 5, True) r
Accessing the a element from tuple 't' - b
----Traversing a tuple----
a
b
5
True
----End of Traversing----
New tuple - ('a', 'b', 5, True, 2, 3, 4)
----Membership Operation----
not overlapping
----End of Membership Operation----
Tuple "a" - (2, 3, 4)
Repetiton of tuple - (2, 3, 4, 2, 3, 4, 2, 3, 4)
tuple j - ('a', 'b', 1, True, 'Pranav', 'Python')
Index of value "1"- 2
Length of tuple - 6
Minimum value - 4
Maximum value - 9
counting a value "5" in a tuple "l" - 2
```

6c:Dictionaries:

Creating dictionaries,

Accessing elements

keys(),values(),items(),update(),copy(),sort(),pop(),popitem(),clear() deleting dictionary

In [4]:
```
#Creating a dictionary
Students={ 1:"Pranav", 2:'Penny', 3:"Rick", 4:"Morty", 5:'Elon Musk'}
print(Students)

#Accessing a dictionary
a = Students[5]
print(a)

#keys(): This method returns the keys of the dictionary, as a list.
x = Students.keys()
print(x)

#values(): This method returns the values of the dictionary, as a list.
x = Students.values()
print(x)

#items(): This method returns the key-value pairs of the dictionary, as tuples
in a list.
x = Students.items()
print(x)

#update(): This method inserts the specified items to the dictionary.
new_Students = {6:"Harvey Specter",7:"Mike Ross"}
print(new_Students)
Students.update(new_Students)
print(Students)

#copy(): This method returns a copy of the specified dictionary.
ok = Students.copy()
print(ok)

#sorted(): method returns a sorted sequence of the keys in the dictionary as a
list
dict1={3:"Three",4:"Four",1:"One",2:"Two"}
dict2=sorted(dict1)
print(dict2)

#pop(): This method is used to remove and display an item from the dictionary.
x = Students.pop(4)
print(x)

#popitem(): This method removes the last item in the dictionary
dict2={3:"Three",4:"Four",1:"One",2:"Two", 5:"Five"}
x=dict2.popitem()
print(x)

#clear(): This method method removes all the elements from a dictionary.
Students.clear()
print(Students)
```

```
{1: 'Pranav', 2: 'Penny', 3: 'Rick', 4: 'Morty', 5: 'Elon Musk'}
Elon Musk
dict_keys([1, 2, 3, 4, 5])
dict_values(['Pranav', 'Penny', 'Rick', 'Morty', 'Elon Musk'])
dict_items([(1, 'Pranav'), (2, 'Penny'), (3, 'Rick'), (4, 'Morty'), (5, 'Elon
Musk')])
{6: 'Harvey Specter', 7: 'Mike Ross'}
{1: 'Pranav', 2: 'Penny', 3: 'Rick', 4: 'Morty', 5: 'Elon Musk', 6: 'Harvey S
pecter', 7: 'Mike Ross'}
{1: 'Pranav', 2: 'Penny', 3: 'Rick', 4: 'Morty', 5: 'Elon Musk', 6: 'Harvey S
pecter', 7: 'Mike Ross'}
[1, 2, 3, 4]
Morty
(5, 'Five')
{}
```