

Ex. No: 7-A  
Date 04/02/2023

## List items in Library

### AIM:

To create a list of items, present in a library and do all the operations on it.

### ALGORITHM:

STEP 1: Start

STEP 2: Create a list named Lib\_list that contains the elements  
"Research","102","19/01/1956","William"

STEP 3: Print the lib\_list.

STEP 4: Display length of list by using len() function.

STEP 5: Append "CSE" to the list.

STEP 6: Display Lib\_list.

STEP 7: Extend "Periodicals","Newspapers" to the list.

Step 8: Display Lib\_list.

STEP 9: Insert "Historicals" to the index value 1.

STEP 10: Display Lib\_list.

STEP 11: Assign a list named liblist2 and the values in list are Novel, Science, Project

STEP 12: Assign a value conc and Concatenate the Lists Lib\_list and liblist2 and display.

STEP 13: Remove "Research" from the list.

STEP 14: Display Lib\_list.

STEP 15: Pop the value from the list using pop() function

STEP 16: Display Lib\_list.

STEP 17: Display the slicing of the list using index and slicing function.

STEP 18: Sort the Lib\_list using sort() function

STEP 19: Display Lib\_list.

STEP 20: Calculate liblist2\*2 and display.

STEP 21: Stop

**PROGRAM:**

```
Lib_list= ["Research","102","19/01/1956","William"]
print("List:",Lib_list)
print("Length:",len(Lib_list))
Lib_list.append("CSE")
print("Append:",Lib_list)
Lib_list.extend(["Periodicals,Newspapers"])
print("Extend:",Lib_list)
Lib_list.insert(1,"Historicals")
print("Insert:",Lib_list)
liblist2= ["Novel","Science","Project"]
conc = Lib_list + liblist2
print("Concatenation:",conc)
Lib_list.remove("Research")
print("Remove:",Lib_list)
Lib_list.pop(3)
print("Pop:",Lib_list)
Lib_list[0:4]
print("Slicing:",Lib_list)
Lib_list.sort()
print("Sorting:",Lib_list)
print("Repeatition:",liblist2*2)
```

**OUTPUT:**

List: ['Research', '102', '19/01/1956', 'William']

Length: 4

Append: ['Research', '102', '19/01/1956', 'William', 'CSE']

Extend: ['Research', '102', '19/01/1956', 'William', 'CSE', 'Periodicals,Newspapers']

Insert: ['Research', 'Historicals', '102', '19/01/1956', 'William', 'CSE',  
'Periodicals,Newspapers']

Concatenation: ['Research', 'Historicals', '102', '19/01/1956', 'William', 'CSE',  
'Periodicals,Newspapers', 'Novel', 'Science', 'Project']

Remove: ['Historicals', '102', '19/01/1956', 'William', 'CSE', 'Periodicals,Newspapers']

Pop: ['Historicals', '102', '19/01/1956', 'CSE', 'Periodicals,Newspapers']

Slicing: ['Historicals', '102', '19/01/1956', 'CSE', 'Periodicals,Newspapers']

Sorting: ['102', '19/01/1956', 'CSE', 'Historicals', 'Periodicals,Newspapers']

Repeation: ['Novel', 'Science', 'Project', 'Novel', 'Science', 'Project'] ['Research', '102',  
'19/01/1956', 'William']

**RESULT:**

Thus, the programs for the list, tuple, set and dictionaries are given with the result  
successfully

**AIM:**

To create a tuple for components of a car and show all the operations.

**ALGORITHM:**

STEP 1: Start

STEP 2: Create a tuple named car\_components that contains the elements Hood, Turbo, Doors, Steering, Doors as the elements .

STEP 3: Display length of tuple by using **len()** function.

STEP 4: Display the index value of Doors using the index() function

STEP 5: Display the count of Doors using the count() function

STEP 6: Display the minimum value of car\_components using the min() function

STEP 7: Display the maximum value of car\_components using the max() function

STEP 8: Calculate car\_components\*3 and display

STEP 9: Assign a tuple named more\_car\_components and the values in list are Spoilers, Rims, Alloy Wheels

STEP 10: Concatenate the Tuple Car\_components and more\_car\_components and display

STEP 11: Stop

**PROGRAM:**

```
car_components = ("Hood","Turbo","Doors","Steering","Doors")
print("Length of tuple: ", len(car_components))
print("Index of Doors: ", car_components.index("Doors"))
print("Count of 2: ", car_components.count("Doors"))
print("Minimum value: ", min(car_components))
print("Maximum value: ", max(car_components))
print("Tuple repetition: ", car_components * 3)
more_car_components = ("Spoilers","Rims","Alloy Wheels")
```

```
print("Tuple concatenation: ", car_components + more_car_components)
```

**OUTPUT:**

Length of tuple: 5

Index of Doors: 2

Count of 2: 2

Minimum value: Doors

Maximum value: Turbo

Tuple repetition: ('Hood', 'Turbo', 'Doors', 'Steering', 'Doors', 'Hood', 'Turbo', 'Doors', 'Steering',  
'Doors', 'Hood', 'Turbo', 'Doors', 'Steering', 'Doors')

Tuple concatenation: ('Hood', 'Turbo', 'Doors', 'Steering', 'Doors', 'Spoilers', 'Rims', 'Alloy Wheels')

**RESULT:**

Thus, the programs for the list, tuple, set and dictionaries are given with the result successfully

**AIM:**

To Create a set to accept more values and print the elements after removing the duplicate content.

**ALGORITHM:**

STEP 1: Start

STEP 2: Create a empty list

STEP 3: Check for i in range 0 to 5 if true goto step 3.3 else goto step 4

STEP 3.1: Append the value of i

STEP 3.2: Append the value 4

STEP 3.3: Append the value 5

STEP 4: print list

STEP 5: Assign the list value to x as set

STEP 6: Print x

STRP 7: Stop

**PROGRAM:**

```
list=[]
```

```
for i in range(0,5):
```

```
    list.append(i)
```

```
    list.append(4)
```

```
    list.append(5)
```

```
print("Created list which contains duplicate elements : ",list)
```

```
x=set(list)
```

```
print("After creating set removes duplicate elements : ",x)
```

**OUTPUT:**

Created list which contains duplicate elements : [0, 4, 5, 1, 4, 5, 2, 4, 5, 3, 4, 5, 4, 4, 5]

After creating set removes duplicate elements : {0, 1, 2, 3, 4, 5}

**RESULT:**

Thus, the programs for the list, tuple, set and dictionaries are given with the result successfully

Ex. No: 7-D  
Date 04/02/2023

## Laptop specification using Dictionary

### AIM:

To write a program to print the specification of the laptop using dictionary with its operation.

### ALGORITHM:

STEP 1: Start

STEP 2: Create a Dictionary and store in a value laptop.

STEP 3: Assign the key values for Brand:'Asus' , Model:'Vivobook 15' , Processor:'intel core i5' ,  
RAM: 8 , Storage:'1TB' , Graphics:' NVIDIA GeForce RTX 3080' , Screen\_size:15.6

STEP 4: Display laptop specification

STEP 5: Display the value in the key Brand

STEP 6: Display the value in the key Model

STEP 7: Display the value in the key Processor

STEP 8: Display the value in the key RAM

STEP 9: Display the value in the key Storage

STEP 10: Display the value in the key Graphics

STEP 11: Display the value in the key Screen\_size

STEP 12 : Stop

**PROGRAM:**

```
laptop={ "Brand":"Asus",  
         "Model":"Vivobook 15",  
         "Processor":"Intel Core i5",  
         "RAM":8,  
         "Storage":"1TB",  
         "Graphics":"NVIDIA GeForce RTX 3080",  
         "Screen_size":15.6 }  
  
print("Laptop Specification : ")  
print("Brand : ",laptop["Brand"])  
print("Model : ",laptop["Model"])  
print("Processor : ",laptop["Processor"])  
print("RAM : ",laptop["RAM"])  
print("Storage : ",laptop["Storage"])  
print("Graphics : ",laptop["Graphics"])  
print("Screen Size : ",laptop["Screen_size"])
```

**OUTPUT:**

```
Laptop Specification :  
Brand : Asus  
Model : Vivobook 15  
Processor : Intel Core i5  
RAM : 8  
Storage : 1TB  
Graphics : NVIDIA GeForce RTX 3080  
Screen Size : 15.6
```

**RESULT:**

Thus, the programs for the list, tuple, set and dictionaries are given with the result successfully