

Experiment 2 #A

Frason Francis : 201903020 : 25

Aim: To study advanced Data types and functions in Python.

Theory:

Small anonymous functions can be created with the `lambda` keyword. This function returns the sum of its two arguments: `lambda a, b: a+b`. Lambda functions can be used wherever function objects are required. They are syntactically restricted to a single expression. Semantically, they are just syntactic sugar for a normal function definition. Like nested function definitions, lambda functions can reference variables from the containing scope.

The `del` statement can be used to delete an item at a given index. Also, it can be used to remove slices from a list.

The `sum()` function adds the items of an iterable and returns the sum.

The `extend()` method adds all the elements of an iterable (list, tuple, string etc.) to the end of the list.

Algorithm:

1. Create a list with added numbers.
2. By using lambda function filter out the even elements from the list
3. Print the even elements
4. By using the lambda functions filter out the odd element from the list
5. Print the odd elements
6. By using lambda functions filter out all the numbers from the list that is divisible by 3.
7. Print out the elements that is divisible by 3.
8. By using the extend method add the elements at the end of the list.
9. By using extended slices (`::`) and `del` method, delete the all the odd element from the list.
10. Print the list

Code:

```
lst = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
print("Original list: ",lst)
print("-"*50)
even_lst = list(filter(lambda x : x % 2 == 0, lst))
print("even numbers :",even_lst)      #number that is even
print("sum of even numbers: ",sum(even_lst))      #sum of even numbers
odd_lst = list(filter(lambda x : x % 2 != 0, lst))
print("-"*50)
print("odd numbers: ",odd_lst)        #number that is odd
print("sum of odd numbers: ",sum(odd_lst))
#count the number of element which is divisible by 3
div_three = list(filter(lambda x: x % 3 == 0, lst))
print("Count of number divisible by three: ",len(div_three)) #giving the no. % by 3 and the count of
the no.
print("-"*50)
lst.extend([11,12,13,14,15])      #inserting element in a list
print("Extended list: ",lst)
print("-"*50)
del(lst[::2])      #deleting all the odd numbers from the list
print("list after elements deleted: ",lst)
```

Output:

```
In [52]: runfile('C:/Users/jkfra/.spyder-py3/temp.py', wdir='C:/Users/jkfra/.spyder-py3')
Original list:  [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
-----
even numbers :  [2, 4, 6, 8, 10]
sum of even numbers:  30
-----
odd numbers:  [1, 3, 5, 7, 9]
sum of odd numbers:  25
Count of number divisible by three:  3
-----
Extended list:  [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
-----
list after elements deleted:  [2, 4, 6, 8, 10, 12, 14]

In [53]:
```

Conclusion:

In this experiment we have successfully compiled and used advanced data types to execute and implement our programs.