Shubhangi_Dhikale_35

1. What is a List?

```
1) List is a collection of items.
2) It is mutable data type.
3) It follows the ordered.
4) Enclosed by the sqaure bracket>>[]
5) It is used to store sequence of various data types(int,float,string,list
tuple,dict)
6) It is most useful data type in python.
7) List are comma seperated.
8) Duplicates are allowed.
```

2. What is a Tuple?

```
1) Tuple are used to store the sequence.
2) It is Immutable data type.
3) The values of items stored in tuple can not be changed or modified.
4) It follows the ordered,Duplicates items are allowed.
5) It is defined by the round bracket.
6) Hetergenous data type which we can stored in
tuple(int,float,string,list,dict,tuple,boolean)
7) Tuple is faster than list.
8) It is comma seperated.
9) It cannot perform operations like add,update and delete.
```

3. What is the difference between List and Tuple?

```
List:
1) It is mutable data type.
2) List is a collection of items.
3) Enclosed by the sqaure bracket>>[]
4) It is used to store sequence of various data types(int,float,string,list
tuple,dict)
5) It follows the ordered,Duplicates items are allowed.

Tuple:
1) It is Immutable data type.
2) Tuple are used to store the sequence.
3) It is defined by the round bracket.
4) Hetergenous data type which we can stored in
tuple(int,float,string,list,dict,tuple,boolean)
5) It follows the ordered,Duplicates items are allowed.
```

4. Python Program to find the largest element in the list

In [5]:

```
list1=[10,40,50,60,100]
max1=list1[0]
for i in list1:
    if i>max1:
        max1=i
print("largest number:",max1)
```

```
largest number: 100
```

5. Python program to interchange first and last elements in a list.

In [20]:

```
list1=[10,20,30,40,50,60]
print("original list:",list1)
temp=list1[0]
list1[0]=list1[-1]
list1[-1]=temp
print("updated list:",list1)
```

```
original list: [10, 20, 30, 40, 50, 60]
updated list: [60, 20, 30, 40, 50, 10]
```

6. Python program to swap two elements in a list

In [21]:

```
list1=[30,58,90,45,60,20]
print("original list:",list1)
temp=list1[2]
list1[2]=list1[3]
list1[3]=temp
print("after the swapping:",list1)
```

```
original list: [30, 58, 90, 45, 60, 20]
after the swapping: [30, 58, 45, 90, 60, 20]
```

7. Python program to Reverse a List

In [31]:

```
list1=[]
n=int(input("Enter the list size:"))
for i in range(0,n):
    i=int(input())
    list1.append(i)
print(list1)
a=list1[::-1]
print("reverse list:",a)
```

```
Enter the list size:5
10
30
50
56
20
[10, 30, 50, 56, 20]
reverse list: [20, 56, 50, 30, 10]
```

8. Python program to count occurrences of an element in a list

In [33]:

```
list1=[20,10,50,30,40]
count=0
for i in list1:
    count=count+1
print("count of given list:",count)
```

```
count of given list: 5
```

9. Python program to find the sum of elements in a list

In [36]:

```
n=int(input("Enter the list size:"))
sum=0
for i in range(0,n):
    i=int(input())
    sum=sum+i
print("sum of element in the list:",sum)
```

```
Enter the list size:5
10
20
30
40
50
sum of element in the list: 150
```

10. Python program to Multiply all numbers in the list

In [37]:

```
list1=[2,3,4,5,6,7,8,9]
multiply=list[0]
for i in list1:
    multiply=multiply*i
print("multiply of all number in the list:",multiply)
```

```
multiply of all number in the list: 3628800
```

11. What are the ways to find the length of a list

In [41]:

```
#By using len function

list1=[10,20,50,60,40]
print("length of list:",len(list1))


# To find the length of list without using len function

list2=[12,45,67,89,90,45,67,23]
count=0
for i in list2:
    count=count+1
print("length of list:",count)
```

```
length of list: 5
length of list: 8
```

12. Python program to find the smallest and largest number in a list (Without) min-max function

In [11]:

```
# To find the smallest number in list

list1=[10,30,50,40,25,15]
min1=list1[0]
for i in list1:
    if i<min1:
        min1=i
print("smallest number:",min1)


# To find the largest number in list

list1=[10,30,50,40,25,15]
max2=list1[0]
for i in list1:
    if i> max2:
        max2=i
print("largest number:",max2)
```

```
smallest number: 10
largest number: 50
```

13. Python Program to find the area of a circle

In [15]:

```
pi=3.14
r=int(input("Enter the radius:"))
area=pi*r*r
print("area of circle:%.2f" %(area))
```

```
Enter the radius:5
area of circle:78.50
```

14. Take inputs from the user to make a list. Again take one input from the user and search it in the list and delete that element, if found. Iterate over a list using for loop

In [84]:

```
list1=[]
n=int(input("Enter the list size:"))
for i in range(0,n):
    i=int(input())
    list1.append(i)
print(list1)
val=int(input("element to be deleted:"))
for i in list1:
    if val in list1:
        list1.remove(val)
print(list1)
```

```
Enter the list size:5
1
2
3
4
5
[1, 2, 3, 4, 5]
element to be deleted:4
[1, 2, 3, 5]
```

15. You are given a list of integer elements. Make a new list that will store a square of elements of the previous list. (With and without list comprehension)

i. Input_list = [2,5,6,12] ii. Output_list = [4,25,36,144]

In [33]:

```
# With list comprehension
input_list=[2,5,6,12]
new_list=[i**2 for i in input_list]
print(new_list)


# Without list comprehension

input_list=[2,5,6,12]
new_list=[]
for i in input_list:
    new_list.append(i**2)
print(new_list)
```

```
[4, 25, 36, 144]
[4, 25, 36, 144]
```

16. WAP to create two lists, one containing all even numbers and the other containing all odd numbers between 0 to 151

In [32]:

```
list1=[]
for i in range(0,152):
    list1.append(i)
print(list1)
even_number_list=[]
odd_number_list=[]
for i in list1:
    if i%2==0:
        even_number_list.append(i)
    else:
        odd_number_list.append(i)
print('even number list is:',even_number_list)
print('odd number list is:',odd_number_list)
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21,
22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41,
42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61,
62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81,
82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 10
1, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 1
17, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132,
133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148,
149, 150, 151]
even number list is: [0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 3
0, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68,
70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96, 98, 100, 102, 104, 106,
108, 110, 112, 114, 116, 118, 120, 122, 124, 126, 128, 130, 132, 134, 136, 138,
140, 142, 144, 146, 148, 150]
odd number list is: [1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31,
33, 35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 69, 71,
73, 75, 77, 79, 81, 83, 85, 87, 89, 91, 93, 95, 97, 99, 101, 103, 105, 107, 10
9, 111, 113, 115, 117, 119, 121, 123, 125, 127, 129, 131, 133, 135, 137, 139, 1
41, 143, 145, 147, 149, 151]
```

17. Python program to Count Even and Odd numbers in a List

In [37]:

```
list1=[]
n=int(input("enter the list size:"))
for i in range(0,n):
    i=int(input())
    list1.append(i)

even=0
odd=0
for i in list1:
    if i%2==0:
        even+=1
    else:
        odd+=1
print("even number in list:",even)
print("odd number in list:",odd)
```

```
enter the list size:10
20
67
89
24
56
67
13
56
67
12
even number in list: 5
odd number in list: 5
```

18. WAP to make new lists, containing only numbers which are divisible by 4, 6,8, 10, 3, 5, 7, and 9 in separate lists for range(0,151)

In [30]:
```
list1=[]
for i in range(0,152):
    list1.append(i)
list2=[4,6,8,10,3,5,7,9]
list3=[]
for i1 in list1:
    for i2 in list2:
        if i1%i2==0:
            list3.append(i1)
            break
print('list containing numbers which are divisible by 4,6,8,10,3,5,7, and 9:' ,li
```

```
list containing numbers which are divisible by 4,6,8,10,3,5,7, and 9: [0, 3, 4,
5, 6, 7, 8, 9, 10, 12, 14, 15, 16, 18, 20, 21, 24, 25, 27, 28, 30, 32, 33, 35,
36, 39, 40, 42, 44, 45, 48, 49, 50, 51, 52, 54, 55, 56, 57, 60, 63, 64, 65, 66,
68, 69, 70, 72, 75, 76, 77, 78, 80, 81, 84, 85, 87, 88, 90, 91, 92, 93, 95, 96,
98, 99, 100, 102, 104, 105, 108, 110, 111, 112, 114, 115, 116, 117, 119, 120, 1
23, 124, 125, 126, 128, 129, 130, 132, 133, 135, 136, 138, 140, 141, 144, 145,
147, 148, 150]
```

19. From a list containing ints, strings, and floats, make three lists to store them separately

In [58]:
```
list1=["data",34,67,"python",34.45,20.20,56,78,"machine"]
int_list=[]
float_list=[]
string_list=[]
for i in list1:
    if type(i)==int:
        int_list.append(i)
    elif type(i)==float:
        float_list.append(i)
        continue
    else:
        string_list.append(i)
        continue
print()
print("list of integer:",int_list)
print("list of float:",float_list)
print("list of string:",string_list)
```

```

list of integer: [34, 67, 56, 78]
list of float: [34.45, 20.2]
list of string: ['data', 'python', 'machine']
```

20. What’s The Difference Between The Python append() and extend() Methods?

```
# append()
append() adds a single element to the end of the list.
```

```
# extend()
extend() can add the multiple individual elements to the end of the list.
```

21.Write a Python program to append a list to the second list.

In [74]:

```
list1=[10,30,40]
list2=[100,200,300,400]

list1.append(list2)
list1
```

Out[74]:

```
[10, 30, 40, [100, 200, 300, 400]]
```

22. Write a Python program to find the third-largest number in a list

In [93]:

```
list1=[12,10,24,20,30]
largest_num=list1[0]
second_largest=list1[0]
third_largest=list1[0]

for i in list1:
    if i>largest_num:
        third_largest=second_largest
        second_largest=largest_num
        largest_num=i
    elif i>second_largest:
        third_largest=second_largest
        second_largest=i
        continue
    else:
        third_largest=i
print("third largest in list1:",third_largest)
```

```
third largest in list1: 20
```

23. Write a Python program to get the frequency of the elements in a list.

In [123]:

```
#list1=["20","20","10","10","10","40","30","30","40"]
list1=[20,20,10,10,10,40,30,30,40]
frequency={}
for i in list1:
    if i in frequency:
        frequency[i]+=1
    else:
        frequency[i]=1
print(frequency)
```

```
{20: 2, 10: 3, 40: 2, 30: 2}
```

24. Write a Python program to check whether a list contains a sublist

In [4]:

```
list1=[34,67,56,78,[23,67]]
sublist=int(input("enter a number to check:"))
if sublist in list1:
    print(f"yes,list1 contains the {sublist}")
else:
    print(f"No,list1 does not contain the {sublist}")
```

```
enter a number to check:56
yes,list1 contains the 56
```

25. Write a Python program to generate all sublists of a list

In [2]:

```
list1=[2,4,6,8,9,1,5]
for i in list1:
    print(i)
```

```
2
4
6
8
9
1
5
```

26. Write a Python program to find common items from two lists

In [1]:

```
list1=[10,20,30,50,60,80]
list2=[10,20,80,70,100]
list3=[]
for i in list1:
    for j in list2:
        if i==j:
            list3.append(i)
            break
print(list3)
```

```
[10, 20, 80]
```

27. How to flatten a list in python?

In [5]:

```
list1=[[5,6],[1,10],[2,4]]
list2=[]
for i in list1:
    for j in i:
        list2.append(j)
print(list2)
```

```
[5, 6, 1, 10, 2, 4]
```

28. How to sort a list in ascending and descending order without using the sort function?

In [34]:

```
number=[1,5,6,9,0]
for i in range(len(number)):
    for j in range(i+1,len(number)):
        if number[i]<number[j]:
            number[i],number[j]=number[j],number[i]
print(number)
```

```
[9, 6, 5, 1, 0]
```

In [35]:

```
number=[1,5,6,9,0]
for i in range(len(number)):
    for j in range(i+1,len(number)):
        if number[j]<number[i]:
            number[j],number[i]=number[i],number[j]
print(number)
```

```
[0, 1, 5, 6, 9]
```

29. How to sort a tuple?

In [10]:

```
tuple1=(4,8,9,1,3)
a=list(tuple1)
print(a)
a.sort()
print("tuple after the sort:",a)
```

```
[4, 8, 9, 1, 3]
tuple after the sort: [1, 3, 4, 8, 9]
```

In [12]:

```
tuple1=(4,8,9,1,3)
a=list(tuple1)
print(a)
sorted_list=sorted(a)
print("tuple after the sort:",sorted_list)
```

```
[4, 8, 9, 1, 3]
tuple after the sort: [1, 3, 4, 8, 9]
```

30. Write a Python program to convert a list of multiple integers into a single integer

single integer

#a. [11, 33, 50] >>> 113350

In [13]:

```
list1=[11,33,50]
for i in list1:
    print(i,end='')
```

```
113350
```

31. Difference between del and clear?

```
## del >> 1) it will del whole list. or if we give particular indexing at del[]
it will del char of that particular list.
2) The del is a Python reserve word that is used to delete the complete object
or variable from the program.
3) We can use the del keyword to delete the individual list element or the
complete list itself.
4) syntax: del list_name >>delete the complete list
           del list_name[index] >> delete specific element

## clear>>1) It will remove all chars of the list.
2) list_name clear() we can use the list_name.
3) clear() method that will delete all the elements from the list.
4) syntax: list_name.clear()
5) Example: my_list = [1,1,2,2,3,3] my_list.clear() my_list >> []
```

32. Difference between remove and pop

```
# remove >> syntax: list_name.remove()
whatever input we give inside remove function it will remove that char from
existig list.

# pop >> syntax: list_name.pop() >> default pop function wili remove last char
of list.
list_name.pop(index number) >>this function will remove char of that particular
index of the list.
```

33. Difference between indexing and Slicing?

```
# Indexing: means referring to an element by its position.
+ve indexing start with 0 and ends at length-1.
-ve indexing start with -1 and ends at -length.
List indexing in python allows you to access individual characters from the
list directly by using the index.
syntax: list_name[index number]

# Slicing: in Python it is a feature that allows for accessing specific part of
the sequence.
In slicing a string(or any other sequence), we create a substring, which is
essentially a string that exists within another string.
We use slicing when we require a part of the string and not the complete
string.
```

```
Syntax : string[start_index : end_index : step_size]
```

34. Difference between sort and sorted?

In [14]:

```
# sort()
# syntax: list_name.sort(reverse=True/False)
# >>it will arrange the char of the list in either ascending or descending order.
# numerical data will get arrange numerically
# and alphabets by alphabetically

list1 = [7, 1, 3, 4, 6, 10, 2]
print(list1)
# Sorting list by default in ascending order
list1.sort()
print(list1)
# sorting using reverse keyword for descending order
list1.sort(reverse=True)
print(list1)
print()


# sorted()
# syntax: updated_list.sorted(list1)
# >> the list which we input in the sorted fun , it will sort out data.
# and new updated list will form.
list1 = [7, 1, 3, 4, 6, 10, 2]
list2=sorted(list1)
list3=sorted(list1,reverse=True)
print(list1)
print(list2)
print(list3)
```

```
[7, 1, 3, 4, 6, 10, 2]
[1, 2, 3, 4, 6, 7, 10]
[10, 7, 6, 4, 3, 2, 1]

[7, 1, 3, 4, 6, 10, 2]
[1, 2, 3, 4, 6, 7, 10]
[10, 7, 6, 4, 3, 2, 1]
```

35. Difference between reverse and reversed

In [15]:

```
# #reverse() :
# reverse() actually reverses the elements in the list.
# synatx : list_name.reverse()
# Original Sequence is changed.

original_list = [1,2,3,4,5]
print(original_list)
original_list.reverse()
print(original_list)
print()

# reversed() does not actully reverse the list, it provides a iterator
# which is in reversed order.
# syntax: x for x in reversed(list)
# Original sequence remain same

original_list = [1,2,3,4,5]
print(original_list)
reverse_list = [i for i in reversed(original_list)]
print( reverse_list)
```

```
[1, 2, 3, 4, 5]
[5, 4, 3, 2, 1]

[1, 2, 3, 4, 5]
[5, 4, 3, 2, 1]
```

36. Difference between copy and deep copy?

In [17]:

```python
list1 = [1,100,24,26]
print(list1)
## shallow copy list1
list2 = list1
print(list2)
print()
## Change value in list
## Due to shallow copy, list1 items will also be changed

list2[0] = 1000
print("List1: ",list1)
print("List2: ", list2)
print()



# deep copy

list1 = [2,3,4]
print(list1)

## shallow copy list1

list2 = list1.copy()
print(list2)
print()

## Change value in list
## Due to deep copy, list1 items will not be changed

list2[0] = 1000
print("List1: ",list1)
print("List2: ", list2)
```

```
[1, 100, 24, 26]
[1, 100, 24, 26]

List1:  [1000, 100, 24, 26]
List2:  [1000, 100, 24, 26]

[2, 3, 4]
[2, 3, 4]

List1:  [2, 3, 4]
List2:  [1000, 3, 4]
```

37. How to check whether the list is empty or not?

In [18]:

```
#1
list1 = []
if len(list1) == 0:
    print('List is empty')
else:
     print('List not empty')


#2
if list1 == []:
    print('List is empty')
else:
    print('List is not empty')


#3
if len(list1):
    print('list is not empty')
else:
    print('list is empty')
```

```
List is empty
List is empty
list is empty
```

38. How to concatenate two lists?

In [20]:

```
list1=[1,2,3,4]
list2=[5,6,7,8]
list1.extend(list2)
print(list1)


list1=[1,2,3,4]
list2=[5,6,7,8]
list3=list1+list2
print(list3)
```

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

39. How to find the occurrences of an element in the python list?

In [29]:

```
list1=[100,30,20,40,10]
for i in list1:
    count=list1.count(i)
    print(f" {i} is repeated {count} times")
```

```
 100 is repeated 1 times
 30 is repeated 1 times
 20 is repeated 1 times
 40 is repeated 1 times
 10 is repeated 1 times
```

40. How to flatten a list in python

In [31]:

```
list1=[[5,6],[1,10],[2,4]]
list2=[]
for i in list1:
    for j in i:
        list2.append(j)
print(list2)
```

```
[5, 6, 1, 10, 2, 4]
```

In [ ]: