Marwadi University Marwadi Chandarana Group NAAC A+	**Marwadi University** **Faculty of Engineering & Technology** **Department of Information and Communication Technology**	
Subject: Programming With Python (01CT1309)	**Aim:** Write a python program to create, append and remove lists in python.	
Experiment No: 04	**Date:**	**Enrollment No:92510133025**

Aim: Write a python program to create, append and remove lists in python.

IDE:

A collection of items can be managed and stored in an ordered sequence using a Python list, a flexible and robust data structure. Because lists may hold components of several data types—integers, texts, and even other lists—they are incredibly versatile for various computer applications. You can quickly add, remove, and alter elements from Python lists and carry out operations like sorting and slicing.

Example of List in Python

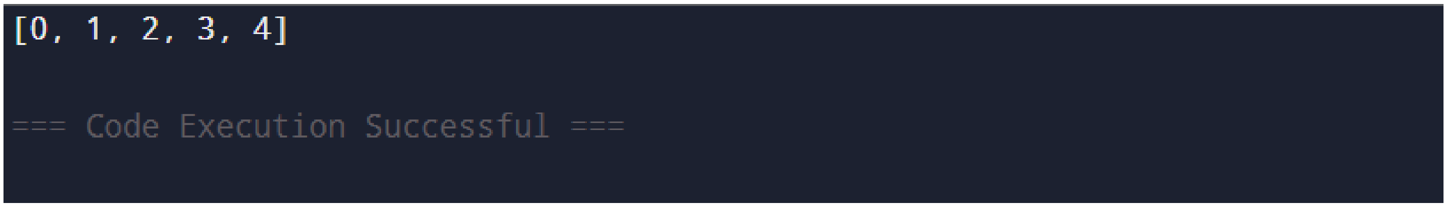
```
ages = [19, 26, 29]
print(ages)
```

Output:

Task:

```
a = list(range(5))
print(a)
```

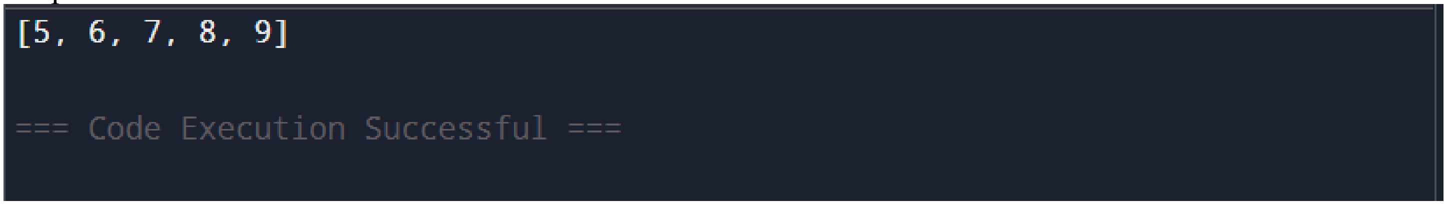
Output:



```
[0, 1, 2, 3, 4]


=== Code Execution Successful ===
```

```
b = list(range(5,10))
print(b)
```

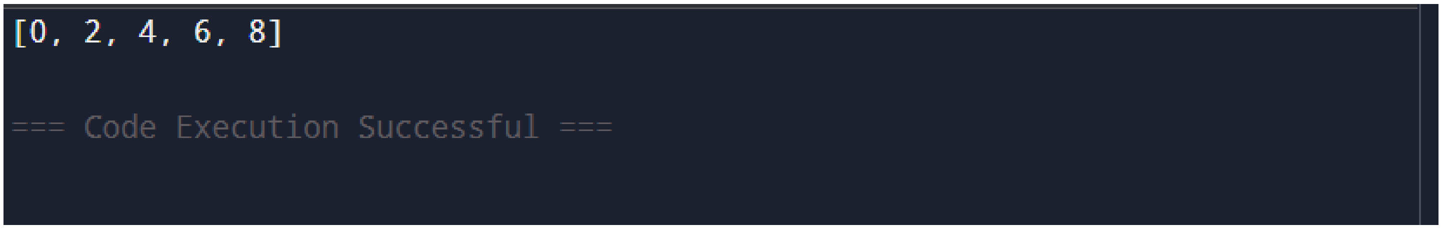
Output:



```
[5, 6, 7, 8, 9]


=== Code Execution Successful ===
```

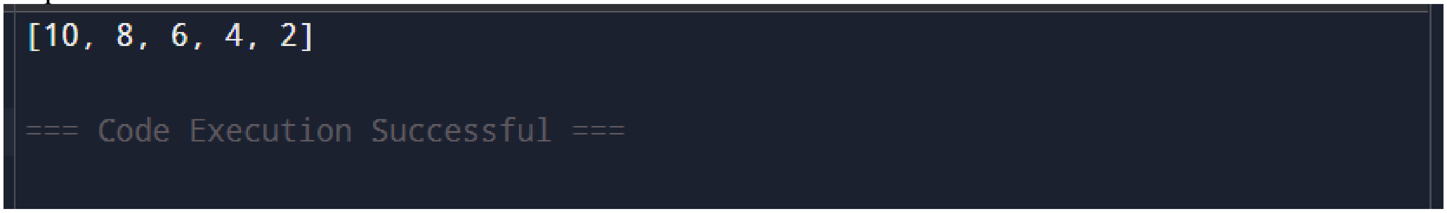
```
c = list(range(0,10,2))
print(c)
```

output:



```
[0, 2, 4, 6, 8]

=== Code Execution Successful ===
```

d = list(range(10,0,-2))
print(d)
output:



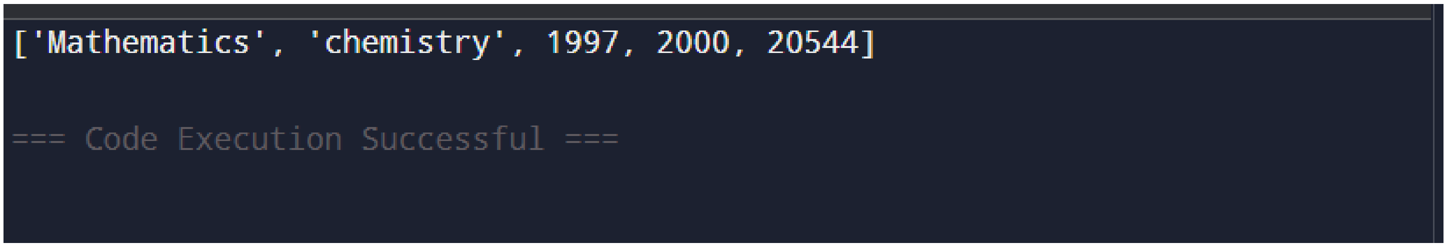
```
[10, 8, 6, 4, 2]

=== Code Execution Successful ===
```

Add Elements to a Python List

1. Python append() Method

Adds element to the end of a list.

List = ['Mathematics', 'chemistry', 1997, 2000]
List.append(20544)
print(List)
output:



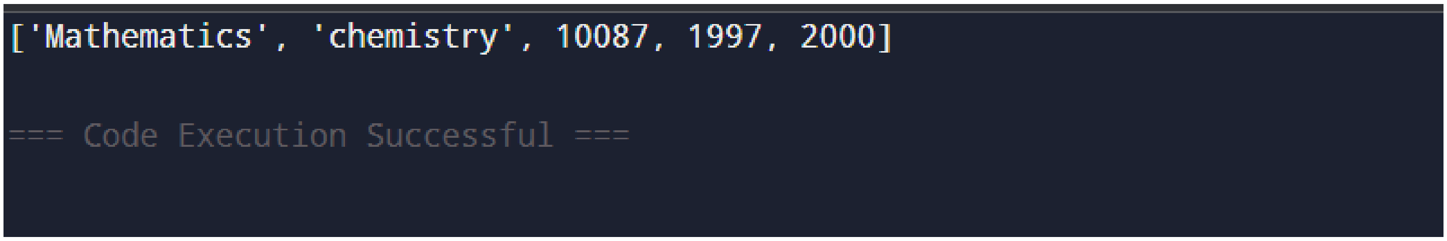
```
['Mathematics', 'chemistry', 1997, 2000, 20544]

=== Code Execution Successful ===
```

2. Python insert() Method

Inserts an element at the specified position.

List = ['Mathematics', 'chemistry', 1997, 2000]
# Insert at index 2 value 10087
List.insert(2, 10087)
print(List)

output:



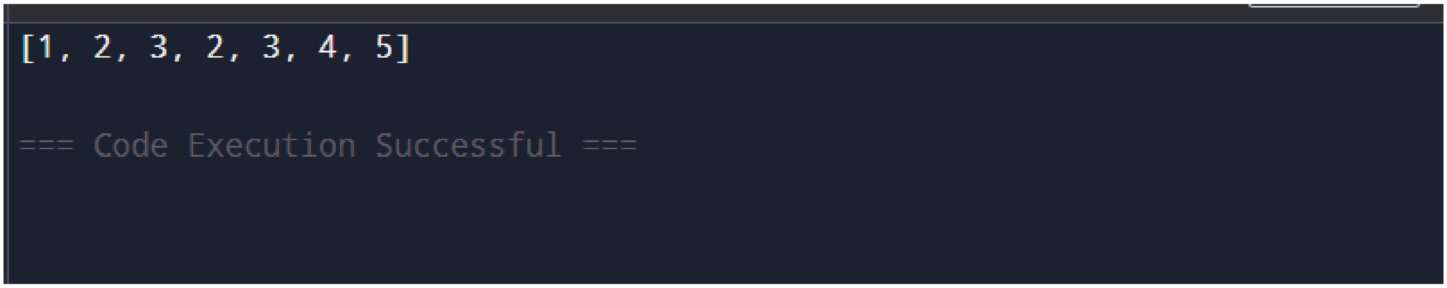
```
['Mathematics', 'chemistry', 10087, 1997, 2000]


=== Code Execution Successful ===
```

3. Python extend() Method

Adds items of an iterable(list.) to the end of a list.

```
List1 = [1, 2, 3]
List2 = [2, 3, 4, 5]
# Add List2 to List1
List1.extend(List2)
print(List1)
```

output:



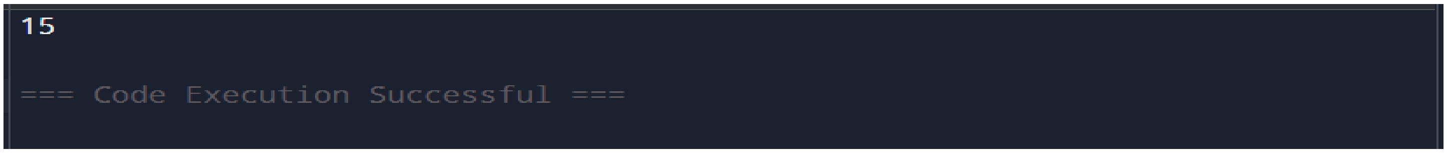
```
[1, 2, 3, 2, 3, 4, 5]


=== Code Execution Successful ===
```

Important Functions of the Python List

1. Python sum() Method
Calculates the sum of all the elements of the List.

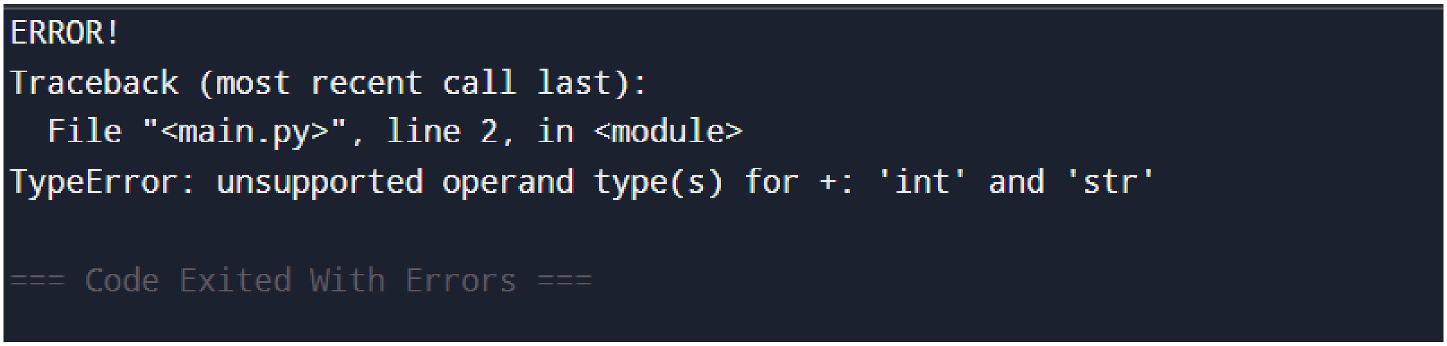
```
List = [1, 2, 3, 4, 5]
print(sum(List))
```

output:



```
15

=== Code Execution Successful ===
```

Task:
List = ['gfg', 'abc', 3]
print(sum(List))
output:

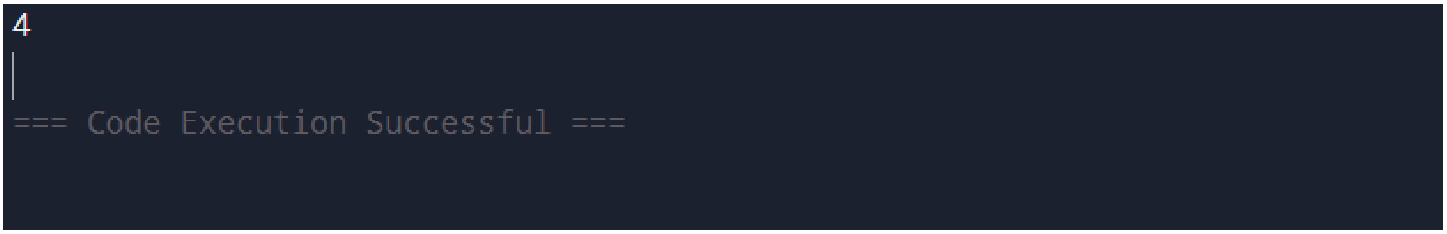


```
ERROR!
Traceback (most recent call last):
  File "<main.py>", line 2, in <module>
TypeError: unsupported operand type(s) for +: 'int' and 'str'

=== Code Exited With Errors ===
```

2. Python count() Method

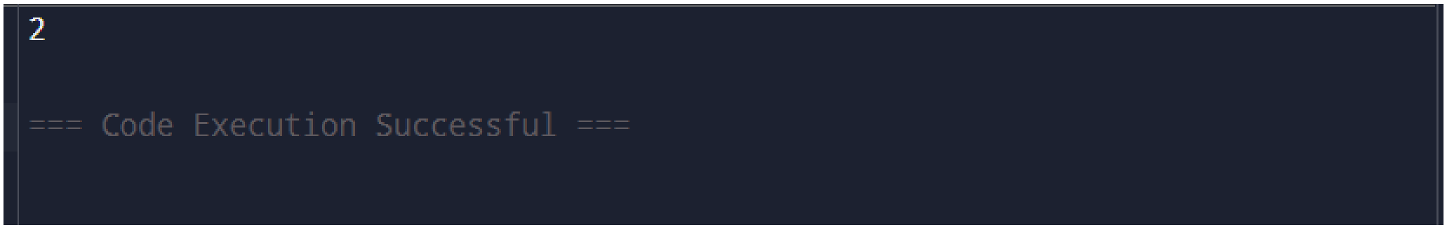
Calculates the total occurrence of a given element of the List.

List = [1, 2, 3, 1, 2, 1, 2, 3, 2, 1]
print(List.count(1))
OUTPUT:



```
4

=== Code Execution Successful ===
```

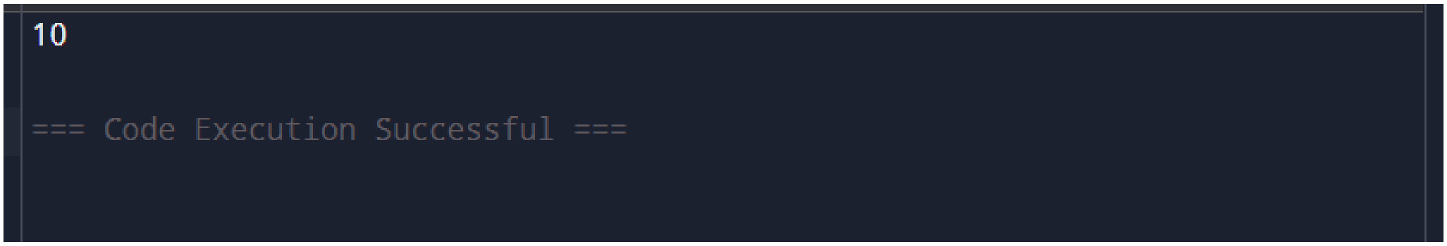
List = [‘a’,’b’,c’,’d’,’a’]
print(List.count(‘a’))
output:



```
2

=== Code Execution Successful ===
```

3. Python len() Method
Calculates the total length of the List.

List = [1, 2, 3, 1, 2, 1, 2, 3, 2, 1]
print(len(List))
output:

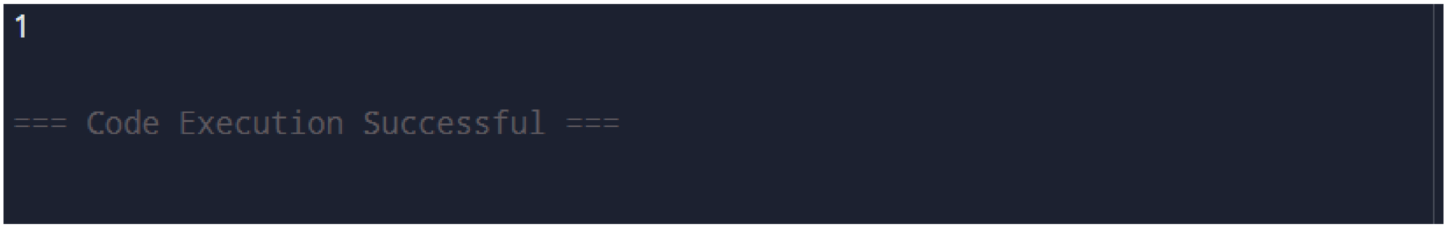


```
10

=== Code Execution Successful ===
```

4. Python index() Method

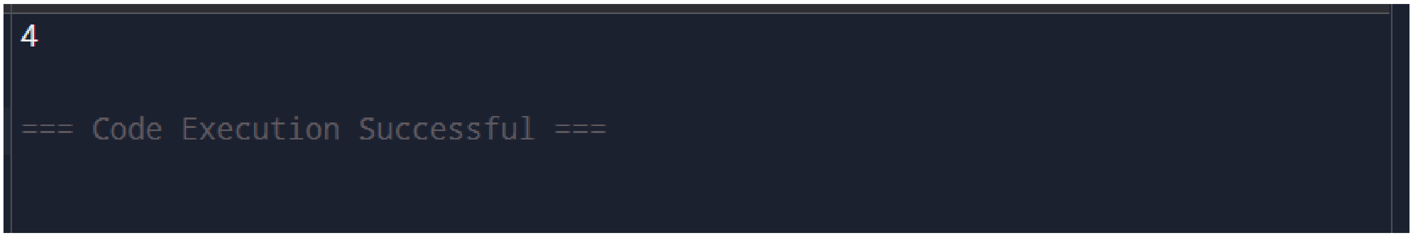
Returns the index of the first occurrence. The start and end indexes are not necessary parameters.

List = [1, 2, 3, 1, 2, 1, 2, 3, 2, 1]
print(List.index(2))
output



```
1

=== Code Execution Successful ===
```

Task:
List = [1, 2, 3, 1, 2, 1, 2, 3, 2, 1]
print(List.index(2, 2))
output



```
4

=== Code Execution Successful ===
```

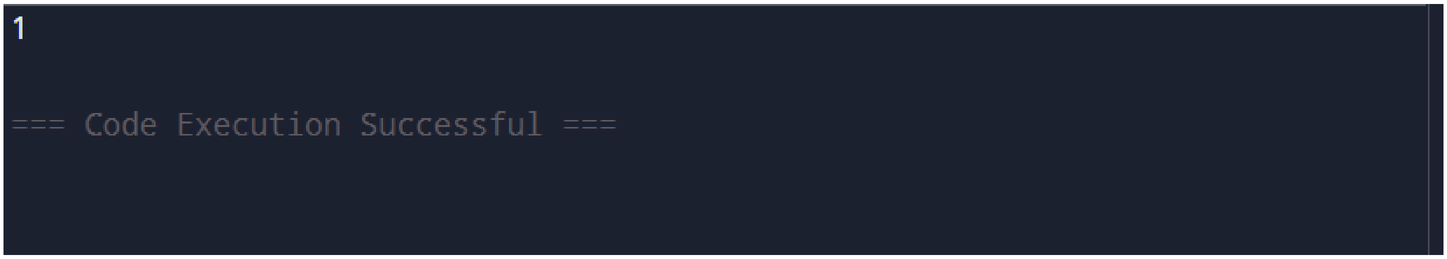
5. Python min() Method

Calculates minimum of all the elements of List.

numbers = [5, 2, 8, 1, 9]

print(min(numbers))

output



```
1

=== Code Execution Successful ===
```

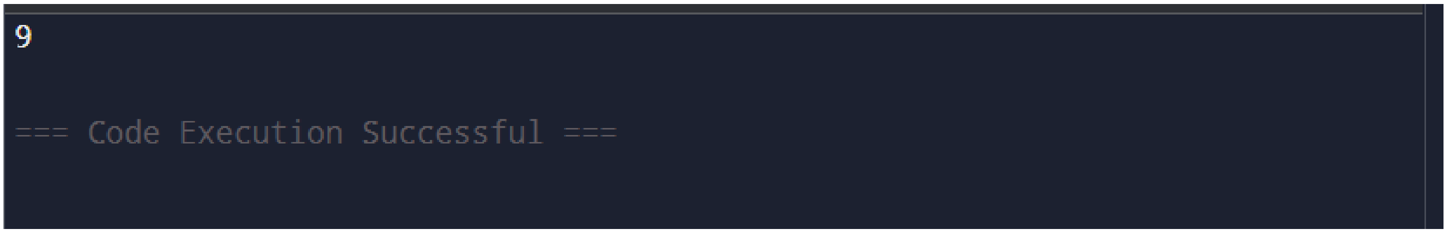
6. Python max() Method

Calculates the maximum of all the elements of the List.

numbers = [5, 2, 8, 1, 9]

print(max(numbers))

output



```
9

=== Code Execution Successful ===
```

7. Python sort() Method

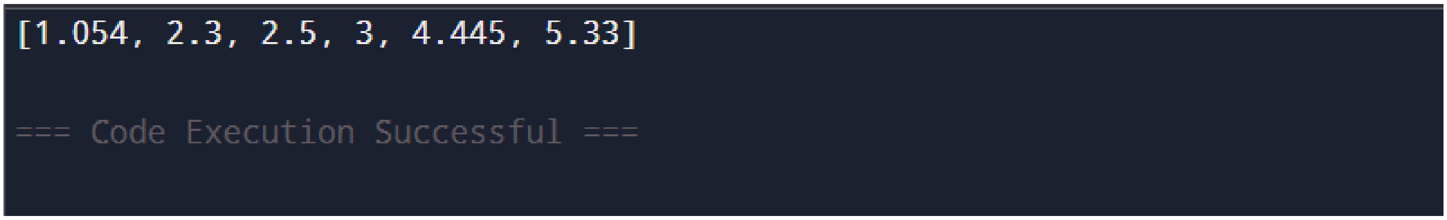
Sort the given data structure (both tuple and list) in ascending order.

List = [2.3,4.445,3,5.33,1.054,2.5]

List.sort()

print(List)

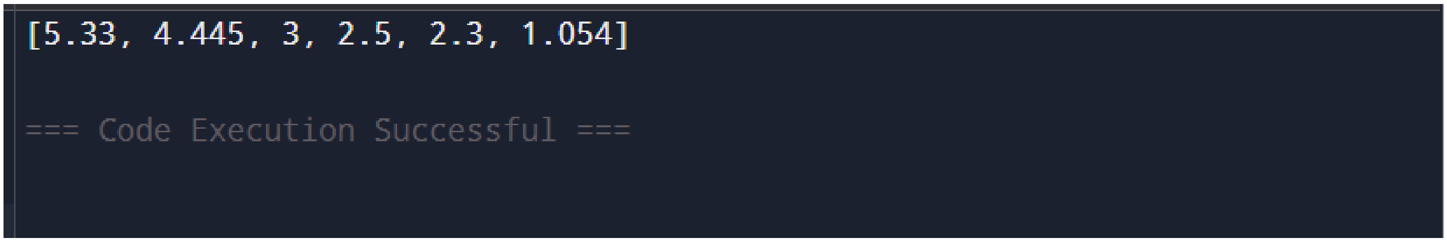
output



```
[1.054, 2.3, 2.5, 3, 4.445, 5.33]

=== Code Execution Successful ===
```

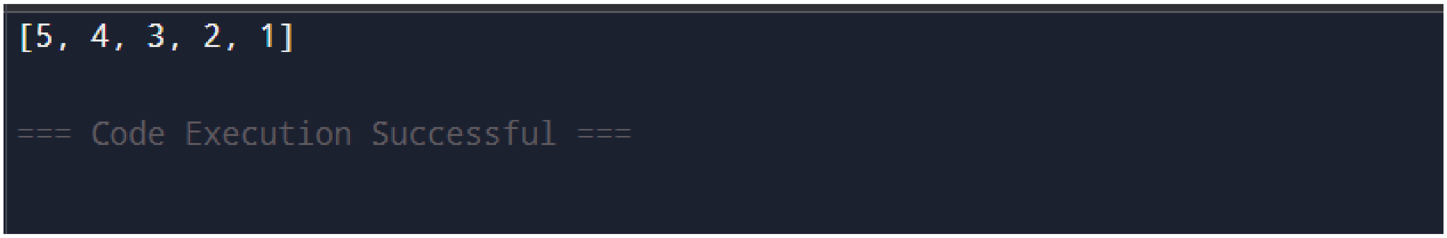
List = [2.3, 4.445, 3, 5.33, 1.054, 2.5]
#Reverse flag is set True
List.sort(reverse=True)
print(List)

output



```
[5.33, 4.445, 3, 2.5, 2.3, 1.054]


=== Code Execution Successful ===
```

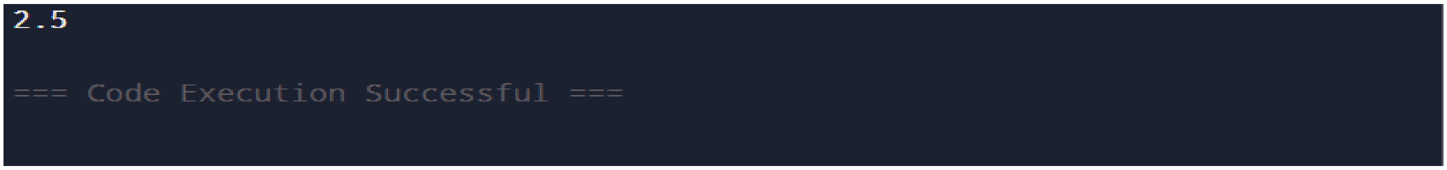
8. Python reverse() Method
reverse() function reverses the order of list.
# creating a list
list = [1,2,3,4,5]
#reversing the list
list.reverse()
#printing the list
print(list)
OUTPUT:



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


=== Code Execution Successful ===
```

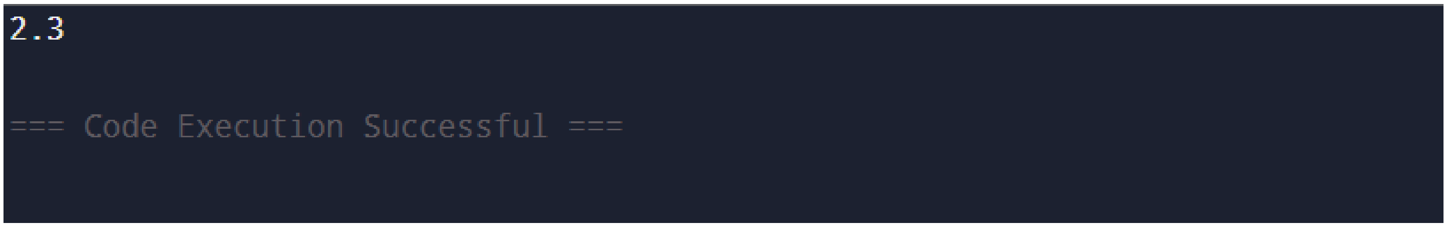
Deletion of List Elements
To Delete one or more elements, i.e. remove an element, many built-in Python list functions can be used, such as pop() and remove() and keywords such as del.

1. Python pop() Method
Removes an item from a specific index in a list.
List = [2.3, 4.445, 3, 5.33, 1.054, 2.5]
print(List.pop())
output:



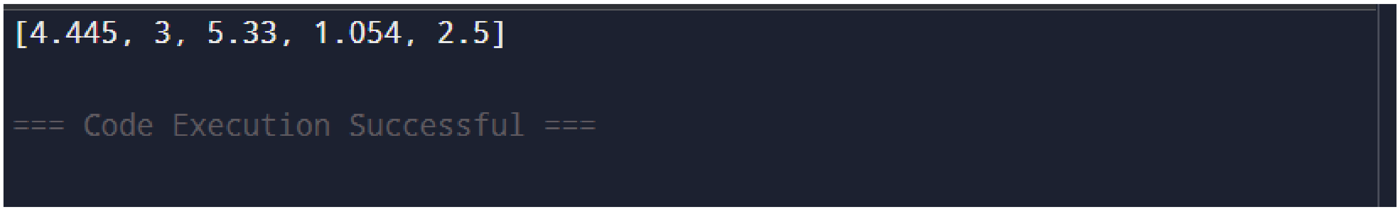
```
2.5

=== Code Execution Successful ===
```

List = [2.3, 4.445, 3, 5.33, 1.054, 2.5]
print(List.pop(0))
output



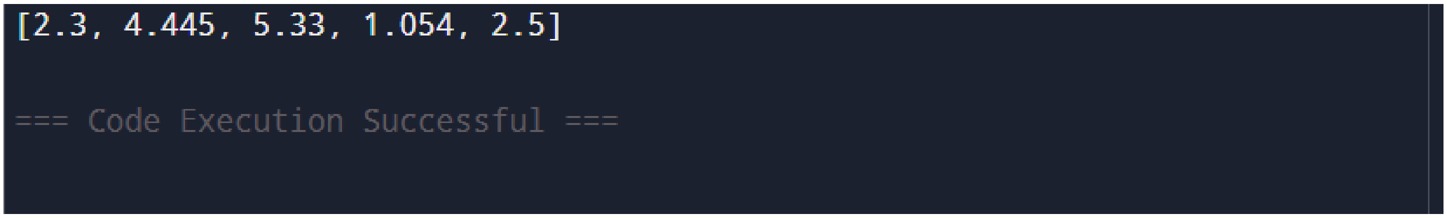
```
2.3

=== Code Execution Successful ===
```

2. Python del() Method
Deletes an element from the list using it’s index.
List = [2.3, 4.445, 3, 5.33, 1.054, 2.5]
del List[0]
print(List)
output:



```
[4.445, 3, 5.33, 1.054, 2.5]

=== Code Execution Successful ===
```

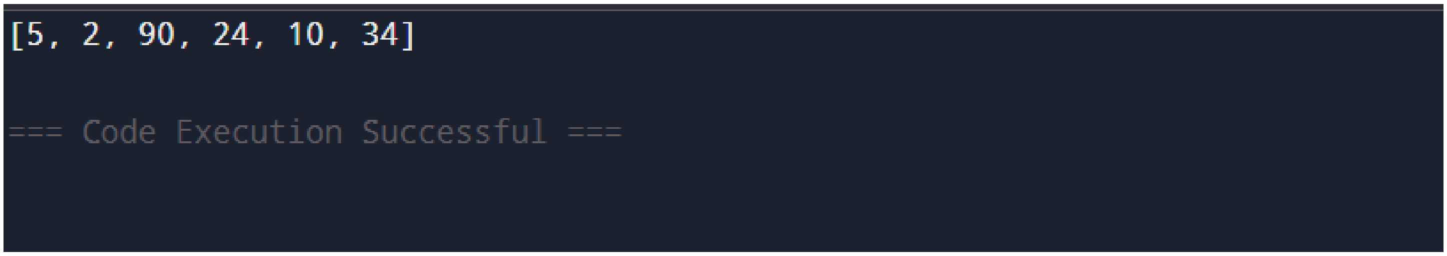
3. Python remove() Method
Removes a specific element using it’s value/name.
List = [2.3, 4.445, 3, 5.33, 1.054, 2.5]
List.remove(3)
print(List)
output:



```
[2.3, 4.445, 5.33, 1.054, 2.5]

=== Code Execution Successful ===
```

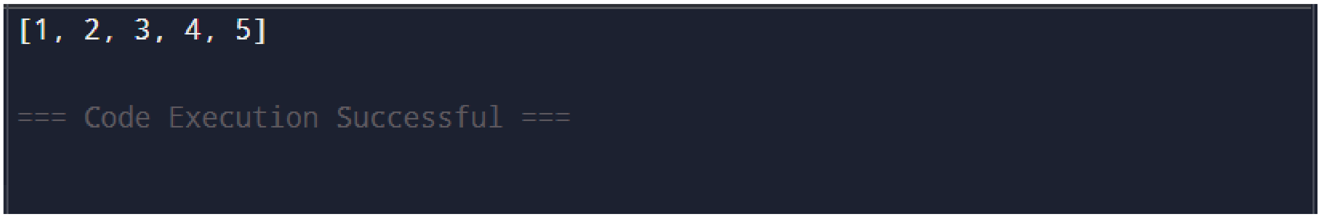
# removing duplicates from a list using dictionaries
my_list_1 = [5, 2, 90, 24, 10, 2, 90, 34]
my_list_2 = ['a', 'a', 'a', 'b', 'c', 'd', 'd', 'e']

# removing duplicates from list 1
my_list_1 = list(dict.fromkeys(my_list_1))
print(my_list_1)
output:



```
[5, 2, 90, 24, 10, 34]

=== Code Execution Successful ===
```

# removing duplicates from list 2
my_list_2 = list(dict.fromkeys(my_list_2))
print(my_list_2)
output:



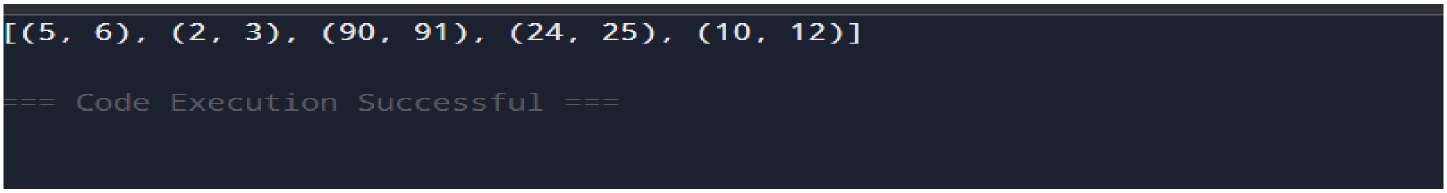
```
[1, 2, 3, 4, 5]

=== Code Execution Successful ===
```

Combining lists

We can even combine lists with the help of the zip() function which results in a list of tuples. Here each item from list A is combined with corresponding elements from list B in the form of a tuple.

# combing lists with the help of zip() function
my_list_1 = [5, 2, 90, 24, 10]
my_list_2 = [6, 3, 91, 25, 12]

# combined
my_combined_list = list(zip(my_list_1, my_list_2))
print(my_combined_list)
output:



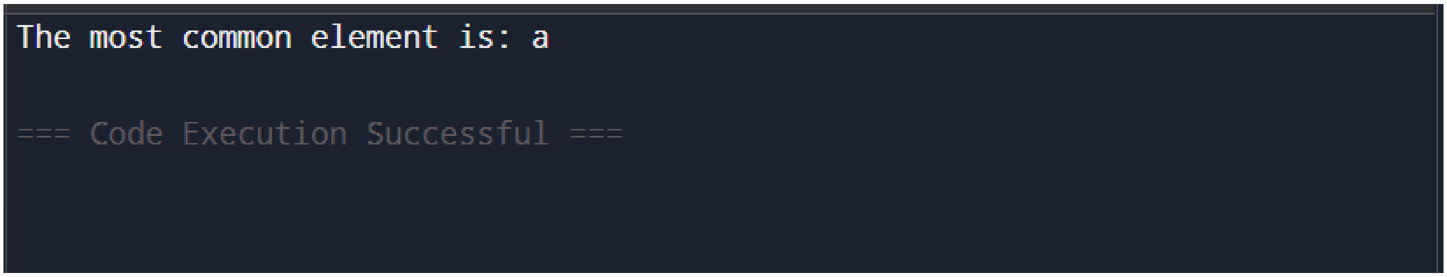
```
[(5, 6), (2, 3), (90, 91), (24, 25), (10, 12)]

=== Code Execution Successful ===
```

Finding the most common item

To find the most frequent element we make use of the set() function. The set() function removes all the duplicates from the list, and the max() function returns the most frequent element (which is found with the help of ‘key’). The key is an optional single argument function.

```
# to find the most frequent element from the list
my_list = ['a', 'a', 'a', 'b', 'c', 'd', 'd', 'e']
most_frequent_value = max(set(my_list), key=my_list.count)
print("The most common element is:", most_frequent_value)
```

output:



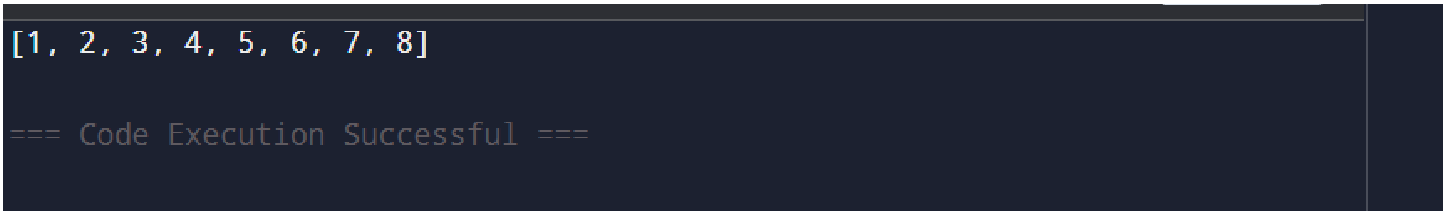
```
The most common element is: a

=== Code Execution Successful ===
```

Flatten a list of lists

Sometimes we encounter a list where each element in itself is a list. To convert a list of lists into a single list, we use list comprehension.

```
# to flatten a list_of_lists by using list comprehension
list_of_lists = [[1, 2],
           [3, 4],
           [5, 6],
           [7, 8]]
# using list comprehension
my_list = [item for List in list_of_lists for item in List]
print(my_list)
```

output:



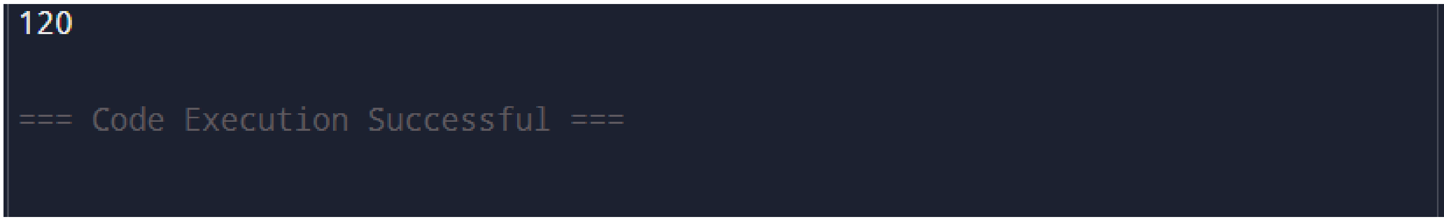
```
[1, 2, 3, 4, 5, 6, 7, 8]

=== Code Execution Successful ===
```

Post Lab Exercise:

a. Write a Python program to multiply all the items in a list.

```
my_list = [2, 3, 4, 5]
result = 1
for x in my_list:
   result *= x
print(result)
```

output:

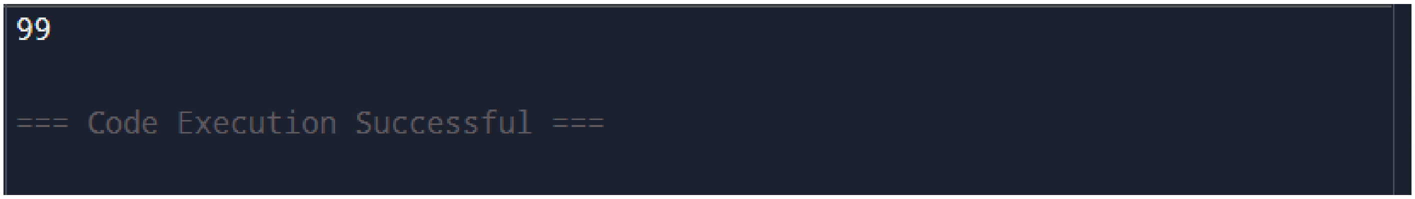


```
120

=== Code Execution Successful ===
```

b. Write a Python program to get the largest number from a list.

```
my_list = [10, 25, 5, 99, 45]
print(max(my_list))
```

output:

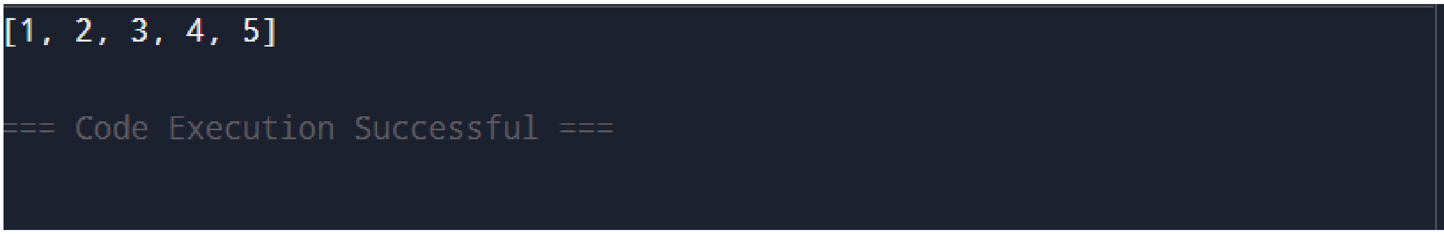


```
99

=== Code Execution Successful ===
```

c. Write a Python program to remove duplicates from a list.

```
my_list = [1, 2, 2, 3, 4, 4, 5]
my_list = list(dict.fromkeys(my_list))
print(my_list)
```

output:



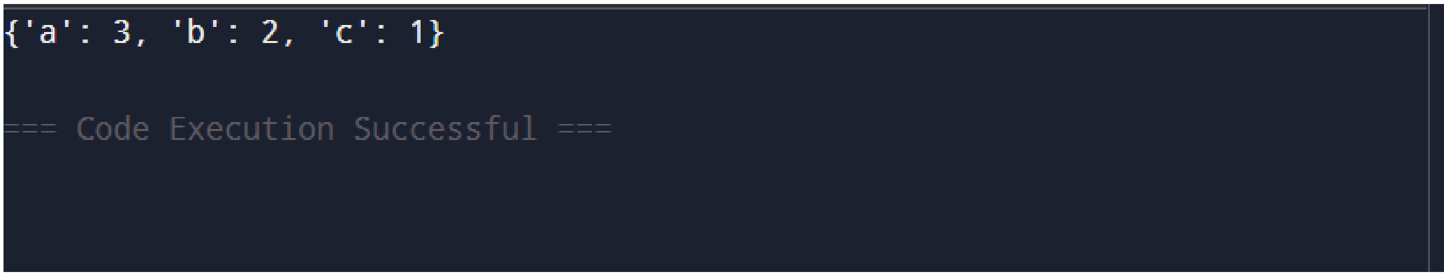
```
[1, 2, 3, 4, 5]

=== Code Execution Successful ===
```

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

```
my_list = ['a', 'b', 'a', 'c', 'b', 'a']
```

```
freq = {}
for item in my_list:
   freq[item] = my_list.count(item)
print(freq)
```

output:

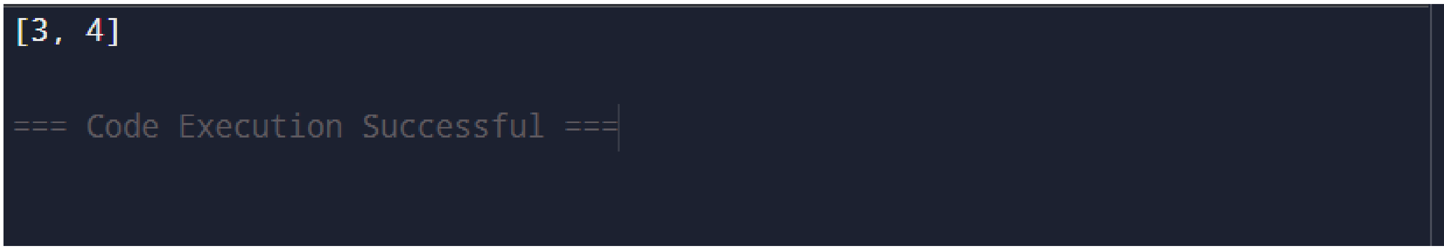


```
{'a': 3, 'b': 2, 'c': 1}

=== Code Execution Successful ===
```

e. Find common items from two lists

```
list1 = [1, 2, 3, 4]
list2 = [3, 4, 5, 6]
common = list(set(list1) & set(list2))
print(common)
```

output:

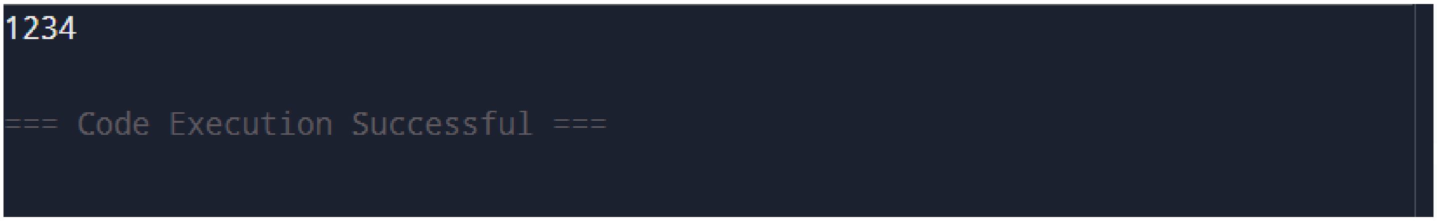


```
[3, 4]

=== Code Execution Successful ===
```

f. Convert a list of multiple integers into a single integer

```
my_list = [1, 2, 3, 4]
result = int("".join(str(i) for i in my_list))
print(result)
```

output:



```
1234

=== Code Execution Successful ===
```

	Marwadi University **Faculty of Engineering & Technology** **Department of Information and Communication Technology**	
Subject: Programming With Python (01CT1309)	**Aim:** Write a python program to create, append and remove lists in python.	
Experiment No: 04	**Date:**	**Enrollment No:92510133025**

Github link: https://github.com/JenishDesai5115/PWP_postlabs