

<https://github.com/ssardarabdullah>
**ABBOTTABAD UNIVERSITY OF SCIENCE AND
TECHNOLOGY**

Lab Task 3

Name	Abdullah niaz
Submitted to	sir jamal abdulahad
Roll no	F24-1312
Section	C

Lab task 1

```
15  
Creating a list with 5  
elements  
my_list = [10, 20, 30,  
40, 50]  
  
Print the second element  
(index 1) and the last  
element (index -1)  
print("Second element:",  
my_list[1])  
print("Last element:",  
my_list[-1])  
  
Modify the third element  
(index 2)  
my_list[2] = 100  
  
Print the updated list  
print("Updated list:",  
my_list)
```

Output:

Second element: 20

Last element: 50

Updated list: [10, 20, 100, 40, 50]

Create two lists

```
list1 = [1, 2, 3, 4, 5]  
list2 = [6, 7, 8, 9, 10]
```

Merge the two lists

```
merged_list = list1 +  
list2
```

Print the merged list

```
print("Merged List:",  
merged_list)
```

Check if a specific
element exists in the
merged list

```
element = 5  
if element in  
merged_list:  
    print(f"{element}  
exists in the merged  
list.")  
else:  
    print(f"{element}  
does not exist in the  
merged list.")
```

Output:

```
Merged List: [1, 2, 3,  
4, 5, 6, 7, 8, 9, 10]  
5 exists in the merged  
list.
```




```
python
```

```
Create a list
```

```
my_list = [10, 20, 30,  
40, 50]
```

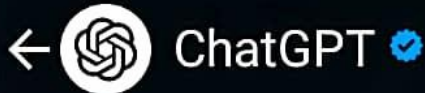
1. Append - Add a single element to the end of the list

```
my_list.append(60)  
print("After append:",  
my_list)
```

2. Extend - Add multiple elements (from another list) to the end of the current list

```
my_list.extend([70, 80,  
90])  
print("After extend:",  
my_list)
```

3. Insert - Insert an element at a specific



4. Remove - Remove the first occurrence of a specified element

```
my_list.remove(40)  
# Remove the first occurrence of 40  
print("After remove:",  
my_list)
```

5. Pop - Remove and return the element at a specific index

```
popped_item =  
my_list.pop(3) # Remove the item at index 3 and store it  
print("After pop:",  
my_list)  
print("Popped item:",  
popped_item)
```

6. Sort - Sort the list in ascending order

```
my_list.sort() # This will sort the list in ascending order  
print("After sort:",  
my_list)
```

Message



7. Reverse - Reverse the order of elements in the list

```
my_list.reverse() #
```

This will reverse the list

```
print("After reverse:",  
my_list)
```


After append: [10, 20, 30, 40, 50, 60]

After extend: [10, 20, 30, 40, 50, 60, 70, 80, 90]

After insert: [10, 20, 15, 30, 40, 50, 60, 70, 80, 90]

After remove: [10, 20, 15, 30, 50, 60, 70, 80, 90]

After pop: [10, 20, 15, 30, 50, 60, 80, 90]

Popped item: 70

After sort: [10, 15, 20, 30, 50, 60, 80, 90]

After reverse: [90, 80, 60, 50, 30, 20, 15, 10]

python

Create a list

```
my_list = ['apple',  
'banana', 'cherry',  
'date', 'elderberry']
```

Use enumerate to print
each element with its
index

```
for index, element in  
enumerate(my_list):  
    print(f"Index  
{index}: {element}")
```

Example Output:

Index 0: apple

Index 1: banana

Index 2: cherry

Index 3: date

Index 4: elderberry

python

Create a list of numbers
from 1 to 20 containing
only multiples of 3

```
multiples_of_3 = [x for  
x in range(1, 21) if x %  
3 == 0]
```

Print the list

```
print(multiples_of_3)
```


Example Output:

[3, 6, 9, 12, 15, 18]

Code:

python

Function to find the sum
of all elements in a
nested list

```
def sum_nested_list(nest  
ed_list):
```

```
    total = 0
```

```
    for element in  
nested_list:
```

```
        if
```

```
isinstance(element,  
list):
```

```
                # If the  
element is a list,  
recursively find the sum
```

```
                total +=
```

```
sum_nested_list(element)
```

```
total = 0
def sum_nested_list(element):
    if isinstance(element, list):
        for i in element:
            total += sum_nested_list(i)
    else:
        # If the
        element is a number, add
        it to the total
        total +=
        element
    return total
```

Example nested list

```
nested_list = [1, [2,
3], [4, [5, 6]], 7]
```

Call the function and
print the result

```
result = sum_nested_list
(nested_list)
print("Sum of all
elements:", result)
```

Example Output:

Sum of all elements: 28


```
python
from collections import
deque
```

Create an empty deque

```
queue = deque()
```

Add elements to the queue (enqueue)

```
queue.append(10)
queue.append(20)
queue.append(30)
```

Print the current state of the queue

```
print("Queue after enqueueing elements:", queue)
```

Remove elements from the queue (dequeue)

```
removed_element = queue.popleft() # Removes the element from the front (10)
print(f"Removed element: {removed_element}")
```

```
Removes the element from  
the front (10)  
print(f"Removed element:  
{removed_element}")
```

```
Print the updated queue  
print("Queue after  
dequeuing an element:",  
queue)
```

```
Adding more elements to  
the queue  
queue.append(40)  
queue.append(50)
```

```
Final state of the queue  
print("Queue after  
adding more elements:",  
queue)
```

Example Output:

Queue after enqueueing
elements: deque([10, 20,
30])

Removed element: 10

Queue after dequeuing
an element: deque([20,
30])

Queue after adding more
elements: deque([20, 30,
40, 50])

<https://github.com/ssardarabdullah>