	SHRI SANT GAJANAN MAHARAJ COLLEGE OF ENGG.	**LABORATORY MANUAL**
	PRACTICAL EXPERIMENT INSTRUCTION SHEET	
	EXPERIMENT TITLE : Write a program to create, append, and remove lists in python.	
EXPERIMENT NO. : **SSGMCE/WI/IT/01/3IT09/01**	ISSUE NO. : 00	ISSUE DATE : 30.07.2023
REV. DATE :	REV. NO. :	DEPTT. : INFORMATION TECHNOLOGY
LABORATORY : 3IT09 COMPUTER SKILL LAB – I	SEMESTER : III	PAGE: 1 OF 2

1.0) AIM: Write a program to create, append, and remove lists in python.

2.0) SCOPE: This lab exercise focuses on introducing students to the essential concepts of lists in Python. The scope encompasses creating, appending, and removing elements from lists, providing a foundation for understanding list manipulation and management.

3.0) FACILITIES/ APPARATUS:

1. Python development environment (e.g., IDLE)
2. Input mechanism (keyboard)

4.0) THEORY:

Lists are fundamental data structures in Python that store collections of items. The following comprehensive explanation details the execution of the program, incorporating relevant functions, data types, syntax, examples, and logical flow:

Program Explanation:

1. Introduction to Lists: Python lists are versatile containers capable of holding multiple data items, enclosed within square brackets [ ] and separated by commas. Lists provide an ordered and mutable collection.

2. Creating an Empty List: The program begins by creating an empty list named my_list.

3. Appending Elements: The append() method adds elements to the end of the list. For instance, my_list.append(10) adds the integer 10 to the list. This step is performed multiple times to add elements 20 and 30.

4. Displaying Initial List: The print() function displays the initial content of the list, which includes the appended elements.

5. Removing Elements: The program employs the remove() method to eliminate a specific element from the list. In this example, the element 20 is removed using an if condition to check its presence in the list.

6. Displaying Updated List: The print() function is used again to showcase the modified list after the removal.

Creating and Appending to Lists

Step 1: Introduction to Lists

In Python, a list is a versatile data structure that can hold multiple elements of different types. Lists are enclosed in square brackets [ ] and elements are separated by commas.

Step 2: Creating an Empty List

1. Open your Python development environment.
2. Create a new Python file and save it with an appropriate name (e.g., list_operations.py).
3. Start by creating an empty list named my_list:

```
my_list = []
```

Step 3: Appending Elements

Use the append() method to add elements to the list. This method allows you to add elements to the end of the list. For example:

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

Display the initial list using the **print()** function:

```
print("Initial List:", my_list)
```

Removing Elements from Lists

1. Step 1: Removing Elements
2. Continuing from the previous code, let's remove an element from the list. For example, let's remove the element 20:

```
element_to_remove = 20
if element_to_remove in my_list:
    my_list.remove(element_to_remove)
```

Display the updated list using the **print()** function:

```
print("Updated List:", my_list)
```

Functions/Data Types Used - Explanation with Example:

3. append() Method: This method adds elements to the end of a list. Syntax: list_name.append(element). Example: my_list.append(10) adds 10 to my_list.
4. remove() Method: This method removes the first occurrence of a specific value from the list. Syntax: list_name.remove(element). Example: my_list.remove(20) removes the element 20.

	SHRI SANT GAJANAN MAHARAJ COLLEGE OF ENGG.	**LABORATORY MANUAL**
	PRACTICAL EXPERIMENT INSTRUCTION SHEET	
	EXPERIMENT TITLE : Write a program to create, append, and remove lists in python.	
EXPERIMENT NO. : **SSGMCE/WI/IT/01/3IT09/01**	ISSUE NO. : 00	ISSUE DATE : 30.07.2023
REV. DATE :	REV. NO. :	DEPTT. : INFORMATION TECHNOLOGY
LABORATORY : 3IT09 COMPUTER SKILL LAB – I	SEMESTER : III	PAGE: 4 OF 2

```
my_list = []  # Creating an empty list
my_list.append(10)  # Appending element 10
my_list.append(20)  # Appending element 20
my_list.append(30)  # Appending element 30
print("Initial List:", my_list)  # Displaying the initial list
element_to_remove = 20  # Element to be removed
if element_to_remove in my_list:
    my_list.remove(element_to_remove)  # Removing element 20
print("Updated List:", my_list)  # Displaying the updated list
```

Program

```
# Create an empty list
my_list = []

# Append elements to the list
my_list.append(10)
my_list.append(20)
my_list.append(30)

# Display the initial list
print("Initial List:", my_list)

# Remove an element from the list
element_to_remove = 20
if element_to_remove in my_list:
    my_list.remove(element_to_remove)
```

4.2) Explanation of Execution:

1. An empty list is created.
2. Elements **10**, **20**, and **30** are added using the **append()** method.
3. The initial list with all appended elements is displayed.
4. Element **20** is removed.
5. The modified list without the removed element is displayed.

5.0) Conclusion:

In this lab, you gained hands-on experience with fundamental list operations in Python. You learned to create an empty list, add elements using the **append()** method, and remove elements using the **remove()** method. By exploring additional exercises, you further solidified your understanding of lists and their versatility in programming.