PLP-Week-2-Python-data-structures-my_list-

This guide includes explanations and code examples for creating, modifying, and working with lists effectively.

Step 1: Creating a List

Python lists are versatile and can store a collection of items. You can create a list using square brackets `[]` or the `list()` constructor.

Code Example

- Here's a Python script that accomplishes all the tasks in your assignment:

1. Create an empty list called my_list.

- my_list = []

2. Append the following elements to my_list: 10, 20, 30, 40.

- my_list.append(10)
- my_list.append(20)
- my_list.append(30)
- my_list.append(40)

3. Insert the value 15 at the second position in the list.

- my_list.insert(1, 15)

4. Extend my_list with another list: [50, 60, 70].

- my_list.extend([50, 60, 70])

5. Remove the last element from my_list.

- my_list.pop()

6. Sort my_list in ascending order.

- my_list.sort()

7. Find and print the index of the value 30 in my_list.

- index_of_30 = my_list.index(30)
- print("Index of 30:", index_of_30)

  # Optional: Print the final state of the list
- print("Final my_list:", my_list)

PLP-Week-2-Python-data-structures-my_list-

Explanation of the Code:

- **Creating an empty list:** Initializes `my_list`.
- **Appending elements:** Adds 10, 20, 30, and 40 to the list.
- **Inserting an element:** Places 15 at index 1 (second position).
- **Extending the list:** Adds the elements from another list `[50, 60, 70]`.
- **Removing the last element:** Uses `pop()` to remove the last item.
- **Sorting the list:** Sorts the list in ascending order.
- **Finding the index:** Uses `index()` to find the position of 30 and prints it.

This script can be run in any Python environment to see the results!