DAY 3 **[27 - JUNE - 2025]**

LIST IN PYTHON

In Python, a list is a built-in, ordered, and changeable collection data type used to store multiple items in a single variable. Lists are defined by enclosing comma-separated items within square brackets []. Lists are used to store multiple items in a single variable.

- List Items

  List items are ordered, changeable, and allow duplicate values. List items are indexed, the first item has index [0], the second item has index [1] etc.

- Ordered
  When we say that lists are ordered, it means that the items have a defined order, and that order will not change.If you add new items to a list, the new items will be placed at the end of the list.
- Changeable
  The list is changeable, meaning that we can change, add, and remove items in a list after it has been created.
- Allow Duplicates
  Since lists are indexed, lists can have items with the same value:
- List Length
  To determine how many items a list has, use the len() function
- The list() Constructor
  It is also possible to use the list() constructor when creating a new list.
- Access Items
  List items are indexed and you can access them by referring to the index number.
- Negative Indexing
  Negative indexing means start from the end. -1 refers to the last item, -2 refers to the second last item etc
- Check if Item Exists
  To determine if a specified item is present in a list use the in keyword.
- Change Item Value
  To change the value of a specific item, refer to the index number.
- Change a Range of Item Values
  To change the value of items within a specific range, define a list with the new values, and refer to the range of index numbers where you want to insert the new values.
- Insert Items
  To insert a new list item, without replacing any of the existing values, we can use the insert() method. The insert() method inserts an item at the specified index.
- Append Items
  To add an item to the end of the list, use the append() method:

- Insert Items

  To insert a list item at a specified index, use the insert() method. The insert() method inserts an item at the specified index

- Remove Specified Item

  The remove() method removes the specified item

- Remove Specified Index

  The pop() method removes the specified index.

- Clear the List

  The clear() method empties the list. The list still remains, but it has no content.

- Sort List Alphanumerically

  List objects have a sort() method that will sort the list alphanumerically, ascending, by default:

- Sort Descending

  To sort descending, use the keyword argument reverse = True:

- Copy a List

  You cannot copy a list simply by typing list2 = list1, because: list2 will only be a *reference* to list1, and changes made in list1 will automatically also be made in list2.

- Use the copy() method

  You can use the built-in List method copy() to copy a list

- Use the list() method

  Another way to make a copy is to use the built-in method list().

- Use the slice Operator

  You can also make a copy of a list by using the : (slice) operator.

- Join Two Lists

  There are several ways to join, or concatenate, two or more lists in Python. One of the easiest ways are by using the + operator.

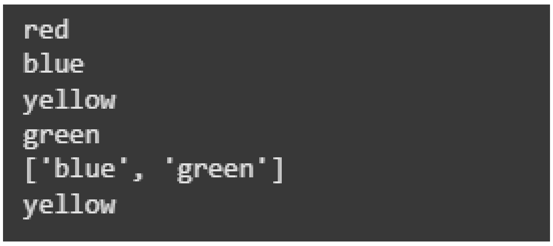
PRACTICE QUESTIONS:

1. colors = ['red', 'blue', 'green', 'yellow'] Using the colors list defined above, print the: First element, Second element, Last element, Second-to-last element, Second and third elements, Element at index 4.

Code:

```
colors = ['red', 'blue', 'green', 'yellow']

print(colors[0])

print(colors[1])

print(colors[-1])

print(colors[-2])

print(colors[1:3])

print(colors[3])
```

Output:



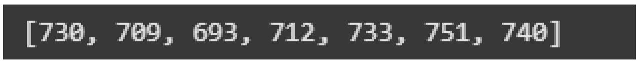
```
red
blue
yellow
green
['blue', 'green']
yellow
```

2. Below is a list with seven integer values representing the daily water level (in cm) in an imaginary lake. However, there is a mistake in the data. The third day’s water level should be 693. Correct the mistake and print the changed list. water_level = [730, 709, 682, 712, 733, 751, 740]

Code:

```
water_level = [730, 709, 682, 712, 733, 751, 740]

water_level[2]=693

print(water_level)
```

Output:



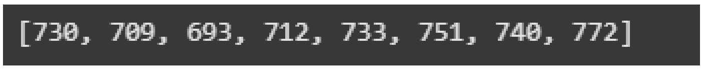
```
[730, 709, 693, 712, 733, 751, 740]
```

3. Add the data for the eighth day to the list from above. The water level was 772 cm on that day. Print the list contents afterwards.

Code:

```
water_level.append(772)

print(water_level)
```

Output:



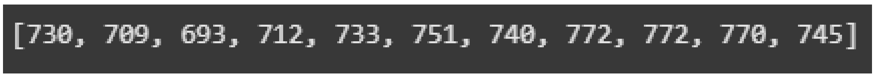
```
[730, 709, 693, 712, 733, 751, 740, 772]
```

4. Still using the same list, add three consecutive days using a single instruction. The water levels on the 9th through 11th days were 772 cm, 770 cm, and 745 cm. Add these values and then print the whole list.

Code:

```
water_level.extend([772,770,745])

print(water_level)
```

Output:



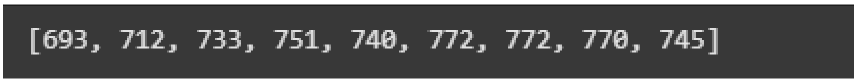
```
[730, 709, 693, 712, 733, 751, 740, 772, 772, 770, 745]
```

5. There are two ways to delete data from a list: by using the index or by using the value. Start with the original water_level list we defined in the second exercise and delete the first element using its index. Then define the list again and delete the first element using its value.

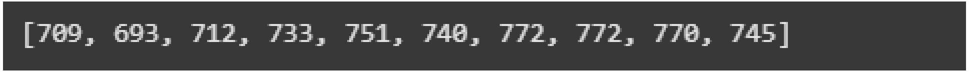
Code:

```
del water_level[0]

print(water_level)

water_level=[730, 709, 693, 712, 733, 751, 740, 772, 772, 770, 745]

water_level.remove(730)

print(water_level)
```

Output:



```
[693, 712, 733, 751, 740, 772, 772, 770, 745]
```



```
[709, 693, 712, 733, 751, 740, 772, 772, 770, 745]
```