Lists in Python

* A list in Python is a group of related values (elements) that are arranged in a specific order and can be changed.
* It can store different data types, including integers, floats, strings, and lists themselves.
* Lists are enclosed in square brackets, and their elements can be accessed using index positions, starting from 0.
* You can add, remove, and modify elements in a list, and you can perform many operations on them like sorting, slicing, and searching.
* Lists are a crucial and adaptable data structure in Python that has various applications in programming.

**Function that are commonly used in list data structure.**

1. Create a list in python



List can be Initialized by just creating “[ ]” and any no of values can be inserted into it. In the above example a list of numbers is created and printed.

1. Accessing the elements in the list



Index numbers can be used to access the elements in the list. the 0th index is the first element and -1 index is the last one.

1. Modifying elements in a list



Elements can be modified using the index numbers and re-assigning other values to it.

1. Adding Elements to list



Elements can be added into the list using the “append()” method. The append method adds the values at the end of the list

1. Removing elements from list



“remove( )” method is used to remove an element from the list. The element's number value is passed to the method.

1. Finding the Length of a list



“len( )” method is used to find the length of the list/

1. Sorting the List

 

“ListName.sort( )” can be used to sort a list in python

1. Checking if an element is present in the list



The “ in “ operator in python is used to check whether an element is present in the list or not. It returns True if the element is present and False if not.

NESTED LISTS :

In Python, a nested list is a list that contains one or more sublists as its elements. In other words, a nested list is a list where each element can be another list.

For Example, consider the following list : 

In this Example *“lst”* is a nested list since it contains the sublists “[2, 3]” and “[6, 7, 8]” in it.

Nested lists can be used to represent structured data with multiple levels of hierarchy.

For example, you might use a nested list to represent a table of data where each row is a list and the entire table is a list of rows.

To access elements of a nested list, you can use multiple levels of indexing. For example, to access the second element of the first sublist in *“lst”*, you would write:



In the above example lst is a nested list. Like normal lists, nested list elements can be accessed using the indexes.

COMPREHENSIVE LISTS :

In Python, a comprehensive list is a concise way to create a new list by performing some operation on each element of an existing list, or by filtering the elements of an existing list based on some condition. The resulting list is created using a single line of code.

Example 1 : 

Using the above example, we can create a list of cubes excluding number 5 from 1 to 5. The above “ lst = “ statement uses “ for “ which loops from 1 to 4 performing the cubic operation and adding the result to the lst.

Example 2 : 

This example filters a list of odd numbers from a given list and creates an *“odd\_list ”.*

1. List comprehension begins with the expression x for x in lst, which iterates through each element x in the original list lst to create a new list.
2. In the following if statement, even numbers are filtered.Specifically, it checks if the current value of x is not divisible by 2(i.e., if x%2!= 0). If this condition is true, the current value of x is included in the new list.
3. All odd numbers are included in the new list after the iteration is complete.