# Python Lists

**Lists**are just like **dynamic** sized arrays, declared in other languages (vector in C++ and ArrayList in Java). Lists need not be homogeneous always which makes it a most powerful tool in Python. A single list may contain DataTypes like Integers, Strings, as well as Objects. Lists are mutable, and hence, they can be altered even after their creation.

List in Python are ordered and have a definite count. The elements in a list are indexed according to a definite sequence and the indexing of a list is done with 0 being the first index. Each element in the list has its definite place in the list, which allows duplicating of elements in the list, with each element having its own distinct place and credibility.

**Note-** Lists are a useful tool for preserving a sequence of data and further iterating over it.

# Creating a List

Lists in Python can be created by just placing the sequence inside the square brackets [ ]. Unlike Sets, list doesn’t need a built-in function for creation of list. See the code below for implementation of a list.

|  |
| --- |
| # Python program to demonstrate  # Creation of List    # Creating a List  List = []  print("Blank List: ")  print(List)    # Creating a List of numbers  List = [10, 20,22, 63, 65, 14]  print("\nList of numbers: ")  print(List)    # Creating a List of strings and accessing  # using index  List = ["Python", "C#", "Java"]  print("\nList Items: ")  print(List[0])  print(List[2])    # Creating a Multi-Dimensional List  # (By Nesting a list inside a List)  List = [["Python", "C#", "Java"] ]  print("\nMulti-Dimensional List: ")  print(List) |

### **Creating a list with multiple distinct or duplicate elements**

A list may contain duplicate values with their distinct positions and hence, multiple distinct or duplicate values can be passed as a sequence at the time of list creation.

*# Creating a List with the use of numbers having duplicate values*)

List = [1, 2, 2, 5, 4, 4, 3, 3, 3, 6, 5]

print("\nList with the use of Numbers: ")

print(List)

By using sort () method, the result will be a sorted list. Reverse() method is used to reverse the order of the list. Please take note that if the list was not in a sorted order, reverse does not produce the list in descending order. So what do you do.?

*# Creating a List with  mixed type of values having numbers and strings*)

List = [1, 2, 'Python', 4, 'C#', 6, 'Java']

print("\nList with the use of Mixed Values: ")

print(List)

You can check the length of the list by using len function. For example

print(Len(List))

The output of the above code will be 7

# Adding Elements to a List

### **Using append () method**

Elements can be added to the List by using built-in **append()** function. Only one element at a time can be added to the list by using append() method, for addition of multiple elements with the append() method, loops are used. Tuples can also be added to the List with the use of append method because tuples are immutable. Unlike Sets, Lists can also be added to the existing list with the use of append() method.

List1=[2,3,2,3,6,9,8,7]  
print(len(List1))  
List1.append(55)

print(List1)

You can also use remove () method to remove an element from a list.

### **Using insert() method**

append () method only works for addition of elements at the end of the List, for addition of element at the desired position, insert() method is used. Unlike append() which takes only one argument, insert() method requires two arguments(position, value).

List1=[2,3,2,3,6,9,8,7]  
List1.append(55)  
print(**"The initial list "** )  
List1.insert(3, 77)  
print(List1)

The output of the above code will be as shown below.



### **Using extend() method**

Other than append() and insert() methods, there’s one more method for Addition of elements, **extend()**, this method is used to add multiple elements at the same time at the end of the list. Take note of the [ ] bracket used to hold the added elements. **Note –**append() and extend() methods can only add elements at the end. See code below.

List1 = [2, 3, 2, 3, 6, 9, 8, 7]  
List1.append(55)  
print(**"The initial list "**)  
List1.insert(3, 77)  
print(List1)  
List1.extend([99, **"xyz"**])  
print(List1)

In order to access the list items refer to the index number.Use the index operator [ ] to access an item in a list.The index must be an integer.Nested list are accessed using nested indexing.

print(List1[3])

In Python, negative sequence indexes represent positions from the end of the array. Instead of having to compute the offset as in List[len(List)-3], it is enough to just write List[-3]. Negative indexing means beginning from the end, -1 refers to the last item, -2 refers to the second-last item, etc.

print(List1[-1])

Output of the above code will be xyz

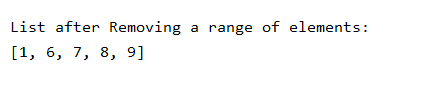
### **Using remove() method**

Elements can be removed from the List by using built-in [**remove()**](https://www.geeksforgeeks.org/python-list-remove/) function but an Error arises if element doesn’t exist in the set. [Remove()](https://www.geeksforgeeks.org/python-list-remove/) method only removes one element at a time, to remove range of elements, iterator is used. The remove() method removes the specified item.

**Note –**Remove method in List will only remove the first occurrence of the searched element.

List1 = [1,2,3,4,5,6,7,8,9]  
**for** i **in** range(2, 6):  
 List1.remove(i)  
print(**"\nList after Removing a range of elements: "**)  
print(List1)

Output

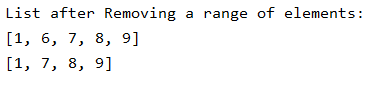


### **Using pop() method**

[Pop()](https://www.geeksforgeeks.org/python-list-pop/) function can also be used to remove and return an element from the set, but by default it removes only the last element of the set, to remove element from a specific position of the List, index of the element is passed as an argument to the pop() method. See the following code:

List1 = [1,2,3,4,5,6,7,8,9]  
**for** i **in** range(2, 6):  
 List1.remove(i)  
print(**"\nList after Removing a range of elements: "**)  
print(List1)  
List1.pop(1) *#removing element at index 1*print(List1)

Output



### **Slicing of a List**

In Python List, there are multiple ways to print the whole List with all the elements, but to print a specific range of elements from the list, we use Slice operation. Slice operation is performed on Lists with the use of a colon(:). To print elements from beginning to a range use [: Index], to print elements from end-use [:-Index], to print elements from specific Index till the end use [Index:], to print elements within a range, use [Start Index:End Index] and to print the whole List with the use of slicing operation, use [:]. Further, to print the whole List in reverse order, use [::-1]. **Note –**To print elements of List from rear end, use Negative Indexes eg

List1 = [1,2,3,4,5, 3, 6,7,8,9]  
print(List1[-1:-]) *#printing items from index 2 to position 6*

Output

[9]

List1 = [1,2,3,4,5, 3, 6,7,8,9]  
print(List1[2:6]) *#printing items from index 2 to position 6*

Output

[3, 4, 5, 3]

### **Exercise 1**

Consider the following values stored in a list called ages

ages= [2, 12, 12, 14, 15, 16, 16, 17, 18, 14, 20, 20]

Write a Python code to get the highest number from a list.

Add Python statements to remove duplicates from a list.

### **Exercise 2- Comparing lists**

Consider the following two lists

ages=[2,12,12,14,15,16,16, 17,18, 14, 20, 20]  
ages1=[2,4,12,14,15,16,16, 17,18, 14, 20, 20]

Write a Python **function** that takes two lists and returns True if they have at least one common member.