**Aim:** Write a python program to create, append and remove lists in python.

**IDE:**

A collection of items can be managed and stored in an ordered sequence using a Python list, a flexible and robust data structure. Because lists may hold components of several data types—integers, texts, and even other lists—they are incredibly versatile for various computer applications. You can quickly add, remove, and alter elements from Python lists and carry out operations like sorting and slicing.

Example of List in Python

ages = [19, 26, 29]

print(ages)

Output:

Task:

a = list(range(5))

print(a)

Output:

b = list(range(5,10))

print(b)

Output:

c = list(range(0,10,2))

print(c)

output:

d = list(range(10,0,-2))

print(d)

output:

Add Elements to a Python List

**1. Python append() Method**

Adds element to the end of a list.

List = ['Mathematics', 'chemistry', 1997, 2000]

List.append(20544)

print(List)

output

**2. Python insert() Method**

Inserts an element at the specified position.

List = ['Mathematics', 'chemistry', 1997, 2000]

# Insert at index 2 value 10087

List.insert(2, 10087)

print(List)

output

**3. Python extend() Method**

Adds items of an iterable(list.) to the end of a list.

List1 = [1, 2, 3]

List2 = [2, 3, 4, 5]

# Add List2 to List1

List1.extend(List2)

print(List1)

output:

Important Functions of the Python List

1. Python sum() Method

Calculates the sum of all the elements of the List.

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

print(sum(List))

output

Task:

List = ['gfg', 'abc', 3]

print(sum(List))

output

2. Python count() Method

Calculates the total occurrence of a given element of the List.

List = [1, 2, 3, 1, 2, 1, 2, 3, 2, 1]

print(List.count(1))

List = [‘a’,’b’,c’,’d’,’a’]

print(List.count(‘a’))

output:

3. Python len() Method

Calculates the total length of the List.

List = [1, 2, 3, 1, 2, 1, 2, 3, 2, 1]

print(len(List))

output

4. Python index() Method

Returns the index of the first occurrence. The start and end indexes are not necessary parameters.

List = [1, 2, 3, 1, 2, 1, 2, 3, 2, 1]

print(List.index(2))

output

Task:

List = [1, 2, 3, 1, 2, 1, 2, 3, 2, 1]

print(List.index(2, 2))

output

5. Python min() Method

Calculates minimum of all the elements of List.

numbers = [5, 2, 8, 1, 9]

print(min(numbers))

output

6. Python max() Method

Calculates the maximum of all the elements of the List.

numbers = [5, 2, 8, 1, 9]

print(max(numbers))

output

7. Python sort() Method

Sort the given data structure (both tuple and list) in ascending order.

List = [2.3,4.445,3,5.33,1.054,2.5]

List.sort()

print(List)

output

List = [2.3, 4.445, 3, 5.33, 1.054, 2.5]

#Reverse flag is set True

List.sort(reverse=True)

print(List)

output

8. Python reverse() Method

reverse() function reverses the order of list.

# creating a list

list = [1,2,3,4,5]

#reversing the list

list.reverse()

#printing the list

print(list)

**Deletion of List Elements**

To Delete one or more elements, i.e. remove an element, many built-in Python list functions can be used, such as pop() and remove() and keywords such as del.

1. Python pop() Method

Removes an item from a specific index in a list.

List = [2.3, 4.445, 3, 5.33, 1.054, 2.5]

print(List.pop())

output

List = [2.3, 4.445, 3, 5.33, 1.054, 2.5]

print(List.pop(0))

output

2. Python del() Method

Deletes an element from the list using it’s index.

List = [2.3, 4.445, 3, 5.33, 1.054, 2.5]

del List[0]

print(List)

output

3. Python remove() Method

Removes a specific element using it’s value/name.

List = [2.3, 4.445, 3, 5.33, 1.054, 2.5]

List.remove(3)

print(List)

output

# removing duplicates from a list using dictionaries

my\_list\_1 = [5, 2, 90, 24, 10, 2, 90, 34]

my\_list\_2 = ['a', 'a', 'a', 'b', 'c', 'd', 'd', 'e']

# removing duplicates from list 1

my\_list\_1 = list(dict.fromkeys(my\_list\_1))

print(my\_list\_1)

output

# removing duplicates from list 2

my\_list\_2 = list(dict.fromkeys(my\_list\_2))

print(my\_list\_2)

output

Combining lists

We can even combine lists with the help of the zip() function which results in a list of tuples. Here each item from list A is combined with corresponding elements from list B in the form of a tuple.

# combing lists with the help of zip() function

my\_list\_1 = [5, 2, 90, 24, 10]

my\_list\_2 = [6, 3, 91, 25, 12]

# combined

my\_combined\_list = list(zip(my\_list\_1, my\_list\_2))

print(my\_combined\_list)

output

Finding the most common item

To find the most frequent element we make use of the set() function. The set() function removes all the duplicates from the list, and the max() function returns the most frequent element (which is found with the help of ‘key’). The key is an optional single argument function.

# to find the most frequent element from the list

my\_list = ['a', 'a', 'a', 'b', 'c', 'd', 'd', 'e']

most\_frequent\_value = max(set(my\_list), key=my\_list.count)

print("The most common element is:", most\_frequent\_value)

output

Flatten a list of lists

Sometimes we encounter a list where each element in itself is a list. To convert a list of lists into a single list, we use list comprehension.

# to flatten a list\_of\_lists by using list comprehension

list\_of\_lists = [[1, 2],

[3, 4],

[5, 6],

[7, 8]]

# using list comprehension

my\_list = [item for List in list\_of\_lists for item in List]

print(my\_list)

output

**Post Lab Exercise:**

1. Write a Python program to multiply all the items in a list.
2. Write a Python program to get the largest number from a list.
3. Write a Python program to remove duplicates from a list.
4. Write a Python program to get the frequency of elements in a list.
5. Find common items from two lists
6. Convert a list of multiple integers into a single integer