CHAPTER 01

Lists in Python

In this Chapter...

- Creating a List
- Accessing Lists
- Traversing a List
- Comparison Operators
- Membership Operators
- List Operations
- Built-in Functions

In Python, list is a type of container in data structures, which is used to store multiple data at the same time. List acts as an array which defined other languages such as C++, Java etc. List contains a sequence of heterogeneous elements which makes it powerful tool in Python. It can store integer, string as well as object in a single list. It is also useful for implementing stacks and queues.

Lists are mutable which means they can be changed after creation. Each element of a list is assigned a number its position or index. The first index is 0, the second index is 1, the third index is 2 and so on.

Each element in the list has its definite place, which allows duplicating of elements in the list with each element having its own distinct place and credibility.

Creating a List

In Python, lists can be created to put the elements in square brackets [ ]. The elements in the list are separated by the comma (,).

For example,

```
a = [34, 76,11, 98]
b = [‘s’, 3, 6, ‘t’]
c = [34, 0. 5, 75]
d = [ ]
```

Creating a List From an Existing Sequence

In Python, list ( ) method is used to create list from an existing sequence.

Syntax

```
new_list_name = list (sequence/string)
```

Here, sequence includes tuples, lists etc.

For example,

```
>>>A = “PYTHON”
>>>A1 = list(A)
>>>A1
[‘P’, ‘Y’, ‘T’, ‘H’, ‘O’, ‘N’]
>>>A = list(“PYTHON”)
>>>A
[‘P’, ‘Y’, ‘T’, ‘H’, ‘O’, ‘N’]
>>>l = (‘P’, ‘Y’, ‘T’, ‘H’, ‘O’, ‘N’)
>>>l1 = list(l)
>>>l1
[‘P’, ‘Y’, ‘T’, ‘H’, ‘O’, ‘N’]
```

Or

list ( ) method is also used to create list of characters and integers through keyboard.

For example,

```
>>> a= list(input (“Enter the elements :”))
Enter the elements : 234576
>>>a
[‘2’, ‘3’, ‘4’, ‘5’, ‘7’, ‘6’]
>>>b = list(input(“Enter string : ”))
Enter the string : ARIHANT
>>>b
[‘A’, ‘R’, ‘I’, ‘H’, ‘A’, ‘N’, ‘T’]
```

We can create different types of list in Python as follows:

(i) Empty List

Empty list can be created in Python using [ ] . Here is the two ways to create empty list as

(a)
```
>>>a = [ ]
>>>print (a)
```
Output
```
[ ]
```

(b)
```
>>>a = list ( )
>>>print (a)
```
Output
```
[ ]
```

(ii) Mixed Data Types List

It can be created to place different data types such as integers, strings, double etc., into one list.

For example,

```
>>>a = ['Neha', 'Sharma', 25, 75, 6, 47]
>>>print(a)
```

Output

['Neha', 'Sharma', 25, 75, 6, 47]

(iii) Nested List

Nested lists are list objects where the elements in the lists can be lists themselves.

For example,

```
>>> A = ['Neha', 4, 1, [5, 23, 4], 98]
>> print (A)
```

Output

['Neha', 4, 1, [5, 23, 4], 98]

List A contains 5 elements while inner list contains 3 elements ([5, 23, 4]). List A is considered [5, 23, 4] as one element.

Accessing Lists

To access the list's elements, index number is used. Use the index operator [ ] to access the elements of a list. The index should be an integer. Index of 0 refers to first element, 1 refers to second element and so on. While the index of **–1** refers to the first last element, **–2** refers to the second last element and so on.

For example,

```
l1 = [5, 7, 3, 4, 5, 6, 9, 0, 8]
```

It is called positive index

0	1	2	3	4	5	6	7	8
5	7	3	4	5	6	9	0	8
– 9	– 8	– 7	– 6	– 5	– 4	– 3	– 2	– 1

↓

It is called negative index

```
>>>l1 = [34, 87, 'Computer', 12, 'Python', 11, 76,
'Option']
>>>l1 [0]
34
>>>l1 [6]
76
>>>l1 [–4]
'Python'
>>>l1 [3]
12
>>>l1 [–3]
11
>>>l1 [9]
```

It will give an error as IndexError: list index out of range. Because it has 9 elements for which indexing are 0 to 8.

Difference between String and List

Strings are immutable which means the values provided to them will not change in the program.

a = "Here is string"

You can extract value using index, find values but cannot modify it.

While lists are mutable which means the values of list can be changed at any point of time.

a = [1, 2, 3]

You have some methods associated with lists like– append, pop, extend etc.

a **[1 : 1]** = **[5, 6]**, then 'a' will be [1, 5, 6, 2, 3].

Traversing a List

Traversing a list is a technique to access an individual element of that list. It is also called **iterate over a list**.

There are multiple ways to iterate over a list in Python.

These are as follows

Using for loop

The most common and easy way to traverse a list is with for loop. for loop is used when you want to traverse each element of a list.

Syntax for variable in list_name:

For example,

```
a = ['P', 'R', 'O', 'G', 'R', 'A', 'M']
for i in a:
    print (i)
```

Output

```
P
R
O
G
R
A
M
```

Using for loop with range ( )

There is another method to traverse a list using for loop with range( ). This is also used len( ) function with range. This method is used when you want to traverse particular characters in a list.

Syntax for variable in range (len(list_name)):

For example,

```
    a = ['P', 'R', 'O', 'G', 'R', 'A', 'M']
    for i in range (len (a)):
        print (a [i])
```

Output

```
P
R
O
G
R
A
M
```

Program to display the elements of list
['P', 'Y', 'T', 'H', 'O', 'N']
in separate line with their index number.

For example,

```
    list1 = ['P', 'Y', 'T', 'H', 'O', 'N']
    L1 = len (list1)
    for i in range (L1) :
        print("Element :", list1 [i], "at index
                                          number", i)
```

Output

```
Element : P at index number 0
Element : Y at index number 1
Element : T at index number 2
Element : H at index number 3
Element : O at index number 4
Element : N at index number 5
```

Comparison Operators

A comparison operator in Python, also called Python relational operator (<, >, = =, ! = ,> = , <=) that compare the values of two operands and returns True or False based on whether the condition is met.

Comparison operators for comparing lists are as follows

Less than (<) operator

It checks if the left value is lesser than that on the right.

For example,

```
    >>>a = [1, 2, 3, 4]
    >>>b = [5, 2, 3, 4]
    >>>a < b
    True
```

It gives True, which get from first element from two lists as 1 < 5.

Greater than (>) operator

It checks whether the left value is greater than that on the right.

For example,

```
    >>>a = [1, 2, 3, 4]
    >>>b = [5, 2, 3, 4]
    >>>a > b
    False
```

It gives False, which get from first element from two lists as 1 > 5.

Less than or Equal to (<=) operator

This operator returns True only, if the value on the left is either less than or equal to that on the right of the operator.

For example,

```
    >>>a = [2, 4, 3, 7]
    >>>b = [3, 4, 5, 7]
    >>>a <= b
    True
```

Greater than or Equal to (>=) operator

This operator returns True only, if the value on the left is greater than or equal to that on the right of the operator.

For example,

```
    >>>a = [4, 3, 6, 8]
    >>>b = [2, 5, 4, 3]
    >>>a >= b
    True
```

Equal to (= =) operator

This operator returns True, if the values on either side of the operator are equal.

For example,

```
    >>>a = [2, 3, 4, 6]
    >>>b = [2, [3, 4], 6]
    >>>c = [2, 3, 4, 6]
    >>>a = = b
    False
    >>>a = = c
    True
```

Not equal (! =) operator

This operator returns True, if the values on either side of the operator are unequal.

For example,

```
    >>>a = [2, 3, 4, 6]
    >>>b = [2, [3, 4], 6]
    >>>a != b
    True
```

Membership Operators

These operators are used to find out whether a value is a member of a sequence such as string, list, tuple, dictionary etc.

There are two types of membership operator as follows

in operator

It evaluates True, if the value in the left operand appears in the sequence found in the right operand.

For example,

```
l1 = [45, 76, [3, 98], 6]
l2 = [60, [23, 43], 65]
for item in l1:
    if item in l2:
        print (“Exist”)
else :
    print (“Not Exist”)
```

Output

Not Exist

not in operator

It evaluates True, if the value in the left operand does not appear in the sequence found in the right operand.

For example,

```
a = 54
b = 87
list1 = [45, 65, 30, 78, 512, 87]
if (a not in list1):
    print (“a is NOT present in given list”)
else :
    print (“a is present in given list”)
if (b in list1):
    print (“b is present in given list”)
else :
    print (“b is NOT present in given list”)
```

Output

a is NOT present in given list

b is present in given list

List Operations

We can perform various operations on list in Python, some of them are describe below

Concatenate Lists

The most conventional method to perform on the list concatenaton, the use of (+) operator can easily add the whole of one list to other list and hence perform the concatenation.

Syntax list = list1 + list2

For example,

```
>>>l1 = [43, 56, 34]
>>>l2 = [22, 34, 98]
>>>l = l1 + l2
>>>l
[43, 56, 34, 22, 34, 98]
```

The (+) operator cannot add list with other type as number or string because this operator is used only with list types.

For example,

```
>>>l1 = [2, 5, 7]
>>>l = l1 + 5
Trackback (most recent call last) :
File “<pyshell#5>”, line 1, in <module>
   l = l1 + 5
TypeError : can only concatenate list (not
                    “int”) to list.
```

For example,

```
>>>l1 = [3, 2, 6]
>>>l = l1 + “Try”
Trackback (most recent call last):
File “<pyshell#1>”, line 1, in <module>
   l = l1 + “Try”
TypeError : can only concatenate list (not
                    “str”) to list.
```

Replicating List

You can repeat the elements of the list using (*) operator. This operator is used to replicate the list.

Syntax list = list1 * digit

For example,

```
>>>l1 = [3, 2, 6]
>>>l = l1 * 2
>>>l
[3, 2, 6, 3, 2, 6]
```

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 colon (:).

Syntax s = list_name [Start : End]

For example,

```
>>>list1 = [4, 3, 7, 6, 4, 9, 5, 0, 3, 2]
>>>s = list1 [2 : 5]
>>>s
[7, 6, 4]
```

To print elements from beginning to a range use [: Index], to print elements from end use [: –Index] and to print elements from specific index till the end use [Index :].

For example,

```
>>>list1 = [4, 3, 7, 6, 4, 9, 5, 0, 3, 2]
>>>s = list1 [: 5]
>>>s
[4, 3, 7, 6, 4]
>>>s = list1 [: –6]
>>>s
[4, 3, 7, 6]
>>>s = list1 [3 :]
```

```
>>>s
[6, 4, 9, 5, 0, 3, 2]
```

You can also print all elements of list in reverse order using [: : –**1**].

For example,

```
>>>s = list1 [: : –1]
>>>s
[2, 3, 0, 5, 9, 4, 6, 7, 3, 4]
```

Lists are also provide slice steps which used to extract elements from list that are not consecutive.

Syntax s = list_name [Start : Stop : Step]

It takes three parameters which are as follows

- **Start** starting integer where the slicing of the object starts.
- **Stop** integer until which the slicing takes place. The slicing stops at index –**1**.
- **Step** integer value which determines the increment between each index for slicing.

For example,

```
>>>list1 = [4, 3, 7, 6, 4, 9, 5, 0, 3, 2]
>>>l1 = list1 [1 : 10 : 3]
>>>l1
[3, 4, 0]
>>>l2 = list1 [2 : 12 : 2]
>>>l2
[7, 4, 5, 3]
>>>l3 = list1 [::4]
>>>l3
[4, 4, 3]
>>>l4 = list1 [::8]
>>>l4
[4, 3]
>>>l5 = list1 [3 ::]
>>>l5
[6, 4, 9, 5, 0, 3, 2]
```

For example,

Python program to count the number of elements in a given range using traversal and multiple line code.

```
c = 0
l = 40
r = 80
list1 = [10, 20, 30, 40, 50, 40, 40, 60, 70]
for x  in list1 :
    if x >= l and x <= r:
        c + = 1
        print("List:", list1)
        print("Elements in a list1:", c)
```

Output

```
List : [10, 20, 30, 40, 50, 40, 40, 60, 70]
Elements in a list1 : 6
```

List Modification using Slicing

List can be modified after it created using slicing.

For example,

```
>>>l1 = [2, 4, "Try", 54, "Again"]
>>>l1 [0 : 1] = [34, "Hello"]
>>>l1
[34, 'Hello', 4, 'Try', 54, 'Again']
>>>l1 [4] = ["World"]
>>>l1
[34, 'Hello', 4, 'Try', ['World'], 'Again']
>>>l1 [2] = "Hiiii"
>>>l1
[34, 'Hello', 'Hiiii', 'Try', ['World'],
                              'Again']
```

Built-in Functions

Python has large number of built-in functions and methods that make programming easier.

Some of them are as follows

(i) append ( )

This method is used for appending and adding elements to a list. It is used to add elements to the last position of a list.

Syntax list_name.append (element)

For example,

```
>>>l1 = [34, 65, 23, 98]
>>>l1.append (76)
>>>l1
[34, 65, 23, 98, 76]
```

(ii) insert ( )

This method is used to insert an element at specified position in the list. This method takes two arguments : one for index number and second for element value.

Syntax list_name. insert (index, element)

For example,

```
>>>l1 = [34, 65, 23, 98]
>>>l1.insert (3, 'New')
>>>l1
[34, 65, 23, 'New', 98]
```

(iii) extend ( )

This method is used to add contents of list 2 to the end of list 1.

Syntax listname1. extend (list_name2)

For example,

```
>>>l1 = [43, 'Hello', 56]
>>>l2 = ['World', 'Try', 65,77]
>>>l1.extend(l2)
>>>l1
[43, 'Hello', 56, 'World', 'Try', 65, 77]
```

(iv) sum ( )

This method is used to calculate the sum of all the elements in the list.

Syntax sum (list_name)

For example,

```
>>>l = [45, 23, 87, 5, 9]
>>>sum (l)
169
```

sum ( ) method is used for only numeric values otherwise it gives an error.

```
>>>l = [45, 23, 87, 5, 'Hello']
>>>sum (l)
Trackback (most recent call last) :
File "<pyshell# 17>", line 1, in <module>
    sum(l)
TypeError: unsupported operand type (s) for
                + : 'int' and 'str'
```

(v) count ( )

This method is used to calculate total occurrence of given element of list.

Syntax list_name. count (element)

For example,

```
>>>list1 = [4, 3, 5, 2, 54, 4, 2, 6, 4, 4, 5]
>>>list1. count(4)
4
```

(vi) len ( )

This method is used to calculate the total length of list.

Syntax len (list_name)

For example,

```
>>>list1 = [4, 3, 5, 2, 54, 4, 2, 6, 4, 4, 5]
>>>len (list1)
11
```

(vii) index ()

It returns the index of first occurrence. Start and end index are not necessary parameters.

Syntax list_name.index (element[, start [, end]])

For example,

```
>>>list1 = [3, 'New', 2, 6, 'Hello', 2]
>>>list1.index ('Hello')
4
```

(viii) min ( )

It is used to return the minimum element out of elements of list.

Syntax min (list_name)

For example,

```
>>>l1 = [45, 87, 23, 90, 12]
>>>min (l1)
12
>>>l2 = ['A', 'B', 'c', 'e', 'a']
>>>min (l2)
'A'
```

It will return min value of character using ASCII value.

```
>>>l3 = ['Rahul', 'Shiv', 'Sandhaya', 'Ankit']
>>>min (l3)
'Ankit'
```

(ix) max ( )

It is used to return the maximum element out of elements of list.

Syntax max (list_name)

For example,

```
>>>l1 = [34, 76, 89, 33, 54, 65]
>>>max (l1)
89
>>>l2 = ['t', 'e', 'E', 'U', 'v']
>>>max (l2)
'v'
```

It will return max value of character using ASCII value.

(x) reverse ( )

Using the reverse ( ) method, we can reverse the contents of the list object in-place, i.e. we don't need to create a new list instead we just copy the existing elements to the original list in reverse order.

Syntax list_name. reverse ( )

For example,

```
>>>list1 = [34, 76, 89, 33, 54, 65]
>>>list1. reverse( )
>>>print (list1)
[65, 54, 33, 89, 76, 34]
>>>l = ['Hii', 'Hello', 'Hey', 'Namesty']
>>>l.reverse ( )
>>>print (l)
['Namesty', 'Hey', 'Hello', 'Hii']
```

(xi) pop ( )

This function is used to remove the element and return last value from the list or the given index value.

Syntax list_name.pop (index)

For example,

```
>>>l1 = [34, 65, 22, 90, 87, 61]
>>>l1.pop (3)
90
>>>print (l1)
[34, 65, 22, 87, 61]
```

If you do not give any index value, then it will remove last value from the list.

```
>>>l1. pop ( )
61
>>>print (l1)
[34, 65, 22, 87]
```

(xii) remove ( )

This method searches for the given element in the list and removes the first matching element. It takes a single element as an argument and remove it from the list.

Syntax list_name. remove (element)

For example,

```
>>>list1 = [34, 76, 11, 98, 26, 20]
>>>list1.remove (11)
>>>list1
[34, 76, 98, 26, 20]
>>>l1 = [‘Maths’, ‘English’, ‘Hindi’, ‘History’, ‘Science’]
>>>l1.remove(‘History’)
>>>l1
[‘Maths’, ‘English’, ‘Hindi’, ‘Science’]
```

If the element (argument) passed to the remove ( ) method does not exist, ValueError exception is thrown.

```
>>>l1. remove ( )
Trackback (most recent call last) :
File “<pyshell# 11>”, line 1, in <module>
   l1.remove ( )
TypeError : remove ( ) takes exactly one argument (0
given)
```

(xiii) clear ( )

This function is used to remove all the items of a list. This method will empty the entire list.

Syntax list_name. clear ( )

For example,

```
>>>l1 = [23, 45, 87, 12, 98]
>>>l1.clear( )
>>>l1
[ ]
```

(xiv) sort ()

This function is used to sort the given list in ascending order.

Syntax list _name. sort ( )

For example,

```
>>>list1 = [23, 65, 77, 23, 90, 99, 12]
>>>list1. sort ( )
>>>print (list1)
[12, 23, 23, 65, 77, 90, 99]
>>>list2 = [‘abc’ ‘gdr’, ‘uyt’, ‘abc’,‘nki’]
>>>list2. sort ( )
>>>print (list2)
[‘abc’, ‘abc’, ‘gdr’, ‘nki’, ‘uyt’]
```

The sort function has an argument called reverse = True. This allows us to sort the list elements in descending order.

Syntax list _ name. sort (reverse = True)

For example,

```
>>>list1 = [23, 65, 77, 23, 90, 99, 12]
>>>list1. sort (reverse = True)
>>>print (list1)
[99, 90, 77, 65, 23, 23, 12]
```

(xv) list ()

It takes sequence types and converts them to lists. This is used to convert a given sequence (tuple/list/string) into list.

Syntax list (seq)

For example,

```
>>>t1 = (‘Hello’, 34, 54, ‘xyz’)
>>>list1 = list(t1)
>>>print (list1)
[‘Hello’, 34, 54, ‘xyz’]
>>>t1 = ( )
>>>list1 = list (t1)
>>>print (list1)
[]
```

Chapter Practice

PART 1

Objective Questions

• Multiple Choice Questions

1. Which value is used to represent the first index of list?

(a) 1 (b) 0

(c) –1 (d) a

Ans. (*b*) To access the list's elements, index number is used. The index number should be an integer. Index of 0 refers to first element, 1 refers to second element and so on.

2. Choose the output of following Python code.

```
l1 = list ()
print (l1)
```

(a) [] (b) ()

(c) [,] (d) Empty

Ans. (*a*) Empty list can be created in Python using []. To create empty list, list () is also used.

3. Suppose list

```
l1 = [10, 20, 30, 40, 50, 60, 70]
print(l1[–3])
```

(a) 30 (b) 50

(c) 40 (d) Error

Ans. (*b*) The index of **–1** refers to the last element, **–2** refers to the second last element and so on. Hence, **–3** refers to third last element, i.e. 50.

4. Choose the output from following code.

```
list1 = ['A', 'R', 'I','H','A','N','T']
print (list1 [7])
```

(a) T (b) N

(c) A (d) Error

Ans. (*d*) In the given code, we are trying to access 8th element from the list which does not exist as we are having total 7 elements for which the last index is 6. So, Python will give an IndexError.

5. Which function is used to insert an element at specified position in the list?

(a) extend () (b) append ()

(c) insert () (d) add ()

Ans. (*c*) insert () function is used to insert an element at specified position in the list. This method takes two arguments : one for index number and second for element value.

Syntax `list_name.insert(index, element)`

6. Choose the correct option for the following.

```
l1 = [2, 5, 7]
l = l1 + 5
print (l)
```

(a) [7, 10, 12]

(b) [2, 5, 7, 5]

(c) [5, 2, 5, 7]

(d) TypeError

Ans. (*d*) + operator cannot add list with other type as number or string because this operator is used only with list types.

So, it will give TypeError as it can only concatenate list (not "int") to list.

7. What is the output of following code?

```
l1 = [3, 2, 6]
l = l1 * 2
print (l)
```

(a) [3, 2, 6, 3, 2, 6]

(b) [6, 4, 12]

(c) [3, 4, 12]

(d) TypeError

Ans. (*a*) * operator can repeat the elements of the list.

Syntax `list = list1 * digit`

8. Which of the following is true regarding lists in Python?

(a) Lists are immutable.

(b) Size of the lists must be specified before its initialisation.

(c) Elements of lists are stored in contiguous memory location.

(d) size(list1) command is used to find the size of lists.

Ans. (*c*) Elements of lists are stored in contiguous memory location, so it is true regarding lists in Python.

9. Suppose list1 is [56, 89, 75, 65, 99], what is the output of list1 [– 2]?

(a) Error (b) 75

(c) 99 (d) 65

Ans. (*d*) **–1** corresponds to the last index in the list, **–2** represents the second last element and so on.

So, the output for list1 [– 2] is 65 because 65 is second last element of list1.

10. Identify the output of following code.

```
List1=[1, 2, 3, 7, 9]
L=List1.pop(9)
print(L)
```

(a) Syntax error
(b) 9
(c) [1, 2, 3, 7]
(d) None of these

Ans. (*a*) In pop(9), parentheses put index number instead of element. In the given list, maximum index number is 4, then 9 is out of index range.

11. Suppose list1 is [2445,133,12454,123], what is the output of max(list1)?

(a) 2445
(b) 133
(c) 12454
(d)123

Ans. (*c*) max() returns the maximum element in the list. From given options, 12454 is the element with maximum value.

12. To add a new element to a list, which command will we use?

(a) list1.add(8)
(b) list1.append(8)
(c) list1.addLast(8)
(d) list1.addEnd(8)

Ans. (*b*) We use the function append() to add an element to the list.

13. What will be the output of the following Python code?

```
list1=[9, 5, 3, 5, 4]
list1[1:2]=[7,8]
print(list1)
```

(a) [9,5, 3, 7, 8]
(b) [9, 7, 8, 3, 5, 4]
(c) [9,[7, 8], 3, 5,4]
(d) Error

Ans. (*b*) In the piece of code, slice assignment has been implemented. The sliced list is replaced by the assigned elements in the list.

14. Consider the declaration a=[2, 3, 'Hello', 23.0]. Which of the following represents the data type of 'a'?

(a) String
(b) Tuple
(c) Dictionary
(d) List

Ans. (*d*) List contains a sequence of heterogeneous elements which store integer, string as well as object. It can created to put elements separated by comma (,) in square brackets [].

15. Identify the output of the following Python statement.

```
x = [[1, 2, 3, 4], [5, 6, 7, 8]]
y = x [0] [2]
print(y)
```

(a) 3
(b) 4
(c) 6
(d) 7

Ans. (*a*) x is a list, which has two sub-lists in it. Elements of first list will be represented by [0] [i] and elements of second list will be represented by [1] [i].

16. Which method will empty the entire list?

(a) pop()
(b) clear()
(c) sort()
(d) remove()

Ans. (*b*) clear() method is used to remove all the items of a list. This method will empty the entire list.

Syntax

```
list_name.clear()
```

17. Which of the following allows us to sort the list elements in descending order?

(a) reverse = True
(b) reverse = False
(c) sort (descending)
(d) sort. descending

Ans. (*a*) sort() is used to sort the given list in ascending order. The sort() has an argument called reverse = True. This allows us to sort the list elements in descending order.

18. Identify the correct output.

```
>>>l1=[34, 65, 23, 98]
>>>l1. insert(2, 55)
>>>l1
```

(a) [34, 65, 23, 55]
(b) [34, 55, 65, 23, 98]
(c) [34, 65, 55, 98]
(d) [34, 65, 55, 23, 98]

Ans. (*d*) insert() is used to insert an element at specified position in the list. This method takes two arguments : one for index number and second for element value.

Syntax

```
list_name.insert(index, element)
```

19. Find the output from the following code.

```
list1=[2, 5, 4, 7, 7, 7, 8, 90]
del list1[2 : 4]
print(list1)
```

(a) [2, 5, 7, 7, 8, 90]
(b) [5, 7, 7, 7, 8, 90]
(c) [2, 5, 4, 8, 90]
(d) Error

Ans. (*a*) del keyword is used to delete the elements from the list.

20. Slice operation is performed on lists with the use of

(a) semicolon
(b) comma
(c) colon
(d) hash

Ans. (*c*) 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 colon (:).

• Case Based MCQs

21. Suppose that list L1

```
["Hello", ["am", "an"], ["that", "the", "this"], "you",
"we", "those", "these"]
```

Based on the above information, answer the following questions.

(i) Find the output of len (L1).

(a) 10
(b) 7
(c) 6
(d) Error

(ii) Find the output of L1[3 : 5].

(a) ["that", "the", "this"] (b) ["we", "those"]

(c) ["you", "we"] (d) [you, we]

(iii) What will be the output of L1[5:] +L1[2]?

(a) ['those', 'these', 'that', 'the', 'this']

(b) ['those', 'these']

(c) ['that', 'the', 'this']

(d) Error

(iv) Choose the correct output of

```
print (L1[6:])
```

(a)"those" (b) "these"

(c) "those", "these" (d) None of these

(v) Give the correct statement for

['Hello', ['am', 'an'], ['that', 'the', 'this'], 'you', 'we', 'those', 'these']

(a) L1[] (b) L1[:]

(c) all.L1() (d) L1(all)

Ans. (i) (*b*) len() is used to calculate total occurrence of given element of list. L1 is a nested list which contains two sub-lists and a sub-list is considered as a single element.

(ii) (*c*) To print a specific range of elements from the list, we use slice operation. Slice operation is performed on lists with the use of colon (:). In L1[3 : 5], slicing start from index number 3, i.e. 'you' to index number **(5 − 1)** ⇒ **4** i.e. 'we'.

(iii) (*a*) To print elements from specific index till the end, use [Index:], so L1[5:] will print from index number 5 till end. i.e. 'those', 'these'. To print the element at specified index number use [index_number], so L1[2] will print the element at index number 2 which is a sub-list and considered as single element i.e. 'that', 'the', 'this'.

+ operator is used to concatenate the element of list.

(iv) (*b*) To print elements from specific index till the end, use [Index :], so L1[6:] will print from index number 6 till end i.e. "these".

(v) (*b*) To print the whole list, use [:], so L1[:] will print the entire list.

PART 2

Subjective Questions

• Short Answer Type Questions

1. What is nested list ? Explain with an example.

Ans. Nested lists are list objects where the elements in the lists can be lists themselves.

For example,

```
list1 = [45, 43, 12, 'Math', 'Eng', [75, 34,
                    'A'], 5]
```

Here, list1 contains 7 elements, while inner list contains 3 elements. list1 is considered [75, 34, 'A'] as one element.

2. list1 = [45, 77, 87, 'Next', 'Try', 33, 43]

Observe the given list and find the answer of questions that follows.

(i) list1[−2] (ii) list1[2]

Ans. (i) 33 (ii) 87

3. Distinguish between string and list.

Ans. Strings are immutable, which means the values provided to them will not change in the program. While lists are mutable which means the values of list can be changed at any point of time in program.

4. What is the output of below questions?

```
l2=[75, 43, 40, 36, 28, 82]
```

(i) l2.sort ( )

(ii) l2.sort (reverse = True)

Ans. (i) [28, 36, 40, 43, 75, 82]

(ii) [82, 75, 43, 40, 36, 28]

5. Find the errors.

```
L1 = [2, 4, 5, 9]
L2 = L1 * 3
L3 = L1  + 3
L4 = L1. pop (5)
```

Ans. **Error 1** L3 = L1 + 3 because + operator cannot add list with other type as number or string.

Error 2 L1.pop (5) parentheses puts index value instead of element. In the given list, maximum index value is 3 and 5 is out of index range.

6. What will be the output of the following Python code?

```
a=[13,6,77]
a.append([87])
a.extend([45,67])
print(a)
```

Ans. [13,6,77, [87], 45, 67]

7. What will be the output of the following Python code?

```
a=[18,23,69,[73]]
b=list(a)
a[3][0]=110
a[1]=34
print(b)
```

Ans. [18, 23, 69, [110]]

8. Predict the output.

```
L2 = [4, 5, 3, 1]
L3 = [3, 4, 11, 2]
```

(i) print (L2 + L3)

(ii) print (L2. count (0))

(iii) print (L3 [4])

(iv) L3. remove (11)
print (L3)

Ans. (i) [4, 5, 3, 1, 3, 4, 11, 2] (ii) 0

(iii) IndexError (iv) [3, 4, 2]

9. Find the output.

```
L = [3, 4, 53, 4, 0, 2, 4, 7, 29]
```

(i) L [2 : 5] (ii) L [: 7]

(iii) L [4 :] (iv) [: : –1]

Ans. (i) [53, 4, 0] (ii) [3, 4, 53, 4, 0, 2, 4]

(iii) [0, 2, 4, 7, 29] (iv) [29, 7, 4, 2, 0, 4, 53, 4, 3]

10. Find the output of the given code.

```
list2 = ['A','R','I','H','A','N','T']
for i in range (len (list2)):
        print (list2 [i])
```

Ans.

```
A
R
I
H
A
N
T
```

11. Define the replicating lists with an example.

Ans. In Python, you can repeat the elements of the list using (*) operator. This operator is used to replicate the list.

For example,

```
>>>l1 = [4, 5, 7, 1]
>>>l2 = l1 * 2
>>>l2
[4, 5, 7, 1, 4, 5, 7, 1]
```

12. Consider the following list and answer the below questions.

```
l1 = [89, 45, "Taj", "Qutub", 93, 42, "Minar", "Delhi", "Agra"]
```

(i) l1 [5 :] (ii) "Qutab" not in l1

(iii) l1 [–3] (iv) l1 [9]

(v) l1[2 : 5] (vi) l1[–2 : 5]

Ans. (i) [42, 'Minar', 'Delhi', 'Agra']

(ii) False

(iii) 'Minar'

(iv) It gives IndexError because there is index value 0 to 8.

(v) ['Taj', 'Qutub', 93]

(vi) [ ]

13. Predict the output.

```
L1 = [3, 4, 5, 9, 11, 2, 27]
print (L1. index (3))
print (max (L1))
print (len (L1))
```

Ans. **Output**

```
0
27
7
```

14. Write the suitable method's name for the below conditions.

(i) Adds an element in the end of list.

(ii) Returns the index of first occurrence.

(iii) Adds contents of list2 to the end of list1.

Ans. (i) append ( ) (ii) index ( )

(iii) extend ( )

15. Consider the following list myList. What will be the elements of myList after the following two operations:

```
myList = [10,20,30,40]
```

(i) myList.append([50,60])

(ii) myList.extend([80,90]) **[NCERT]**

Ans. (i) [10, 20, 30, 40, [50, 60]]

(ii) [10, 20, 30, 40, 80, 90]

16. What will be the output of the following code segment?

```
myList = [1,2,3,4,5,6,7,8,9,10]
for i in range(0,len(myList)):
    if i%2 = = 0:
        print(myList[i])
```

[NCERT]

Ans.

```
1
3
5
7
9
```

17. What will be the output of the following code segment?

(i)
```
myList = [1,2,3,4,5,6,7,8,9,10]
del myList[3:]
print(myList)
```

(ii)
```
myList = [1,2,3,4,5,6,7,8,9,10]
del myList[:5]
print(myList)
```

(iii)
```
myList = [1,2,3,4,5,6,7,8,9,10]
del myList[::2]
print(myList)
```

[NCERT]

Ans. (i) [1, 2, 3]

(ii) [6, 7, 8, 9, 10]

(iii) [2, 4, 6, 8, 10]

18. Differentiate between append() and extend() functions of list. **[NCERT]**

Ans. Differences between append() and extend() functions of list are

append()	**extend()**
append() method adds an element to a list.	extend() method concatenates the first list with another list (or another iterable).
When append() method adds its argument as a single element to the end of a list, the length of the list itself will increase by one.	extend() method iterates over its argument adding each element to the list, extending the list.

19. Consider a list:

list1 = [6,7,8,9]

What is the difference between the following operations on list1?

(i) list1 * 2

(ii) list1 *= 2

(iii) list1 = list1 * 2 **[NCERT]**

Ans. (i) The statement will print the elements of the list twice, i.e. [6, 7, 8, 9, 6, 7, 8, 9]. However, list1 will not be altered.

(ii) This statement will change the list1 and assign the list with repeated elements, i.e. [6, 7, 8, 9, 6, 7, 8, 9] to list1.

(iii) This statement will also have same result as the statement 'list1 *= 2'. The list with repeated elements, i.e. [6, 7, 8, 9, 6, 7, 8, 9] will be assigned to list1.

20. What possible output(s) are expected to be displayed on screen at the time of execution of the program from the following code?

```
List = [“Poem”, “Book”, “Pencil”, “Pen”]
List1 = List [2 : 3]
List[2] = “Scale”
print (List1)
```

Ans. [“Pencil”]

• Long Answer Type Questions

21. Write the best suited method's name for the following conditions.

(i) Insert an element at specified position.

(ii) Add contents from one list to other.

(iii) Calculate the total length of list.

(iv) Reverse the contents of the list object.

Ans. (i) insert ( )

(ii) extend ( )

(iii) len ( )

(iv) reverse ( )

22. Write a program to move all zeroes to the end of the list.

```
list1 = [4, 7, 0, 9, 8, 0, 2, 0, 4, 9, 0]
n = len (list1)
count = 0
for i in range (n):
    if list1[i] ! = 0:
        list1 [count] = list1[i]
        count + = 1
while count < n :
    list1 [count] = 0
    count + = 1
print (“List after pushing all zeroes to end of list:”)
print (list1)
```

Output

```
List after pushing all zeroes to end of list :
[4, 7, 9, 8, 2, 4, 9, 0, 0, 0, 0]
```

23. Observe the following list and answer the questions that follows.

l1 = [45, 65,“The”, [65, “She”, “He”], 90, 12, “This”, 21]

(i) l1 [3 : 4] = [“That”, 54]

(ii) [l1 [4]]

(iii) l1 [:7]

(iv) l1 [1 : 2] + l1 [3 : 4]

Ans. (i) [45, 65, 'The', 'That', 54, 90, 12, 'This', 21]

(ii) [54]

(iii) [45, 65, 'The', 'That', 54, 90, 12]

(iv) [65, 'That']

24. Write a Python program to find the third largest number in a entered list.

```
y = [ ]
num = int(input(“Enter number of elements:”))
for i in range (1, num + 1):
    x = int (input (“Enter element:”))
    y.append(x)
y. sort ()
print(“Third largest element is :”, y[num-3])
```

Output

```
Enter number of elements : 5
Enter element : 76
Enter element : 45
Enter element : 98
Enter element : 90
Enter element : 23
Third largest element is : 76
```

25. Write Python program to display all the common elements of two lists.

```
list1 = [ ]
num = int (input (“Enter number of elements in List1 :”))
for i in range (1, num +1):
    x = int (input(“Enter element:”))
    list1. append (x)
list2 = [ ]
num1 = int (input (“Enter number of elements in List2 :”))
for j in range (1, num1 +1):
    y = int (input (“Enter element:”))
    list2. append (y)
a_set = set (list1)
b_set = set (list2)
if (a_set & b_set):
    print (“Common elements are:”, a_set & b_set)
```

```
else :
    print (“No common elements”)
```

Output

```
Enter number of elements in List1:5
Enter element : 23
Enter element : –45
Enter element : 84
Enter element : –12
Enter element : 33
Enter number of elements in List2:5
Enter element : 12
Enter element : –45
Enter element : –65
Enter element : 41
Enter element : 32
Common elements are : {–45}
```

26. What will be the output of the following statements?

(i)
```
list1 = [12,32,65,26,80,10]
list1.sort()
print(list1)
```

(ii)
```
list1 = [12,32,65,26,80,10]
sorted(list1)
print(list1)
```

(iii)
```
list1 = [1,2,3,4,5,6,7,8,9,10]
list1[::–2]
list1[:3] + list1[3:]
```

(iv)
```
list1 = [1,2,3,4,5]
list1[len(list1)–1]
```

[NCERT]

Ans. (i) [10, 12, 26, 32, 65, 80]
(ii) [12, 32, 65, 26, 80, 10]
(iii) [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
(iv) 5

27. The record of a student (Name, Roll No., Marks in five subjects and percentage of marks) is stored in the following list:

```
stRecord = [‘Raman’,‘A-36’,[56,98,99,72,69], 78.8]
```

Write Python statements to retrieve the following information from the list stRecord.

(i) Percentage of the student
(ii) Marks in the fifth subject
(iii) Maximum marks of the student
(iv) Roll No. of the student
(v) Change the name of the student from ‘Raman’ to ‘Raghav’ **[NCERT]**

Ans.
(i) `print(stRecord[3])`
(ii) `print(stRecord[2][4])`
(iii) `print(max(stRecord[2]))`
(iv) `print(stRecord[1])`
(v) `stRecord[0]=‘Raghav’`

28. Write a program to read a list of n integers (positive as well as negative). Create two new lists, one having all positive numbers and the other having all negative numbers from the given list. Print all three lists. **[NCERT]**

Ans.
```
List = []
Pos = []
Neg = []
Num = int(input(“Enter the Total Number of
List Elements : ”))
for i in range(1, Num + 1):
     value = int(input(“Enter the Value of
                                %d Element : ” %i))
     List.append(value)
for j in range(Num):
  if(List[j] > = 0):
    Pos.append(List[j])
  else:
    Neg.append(List[j])
print(“Element in Positive List is : ”,
                          Pos)
print(“Element in Negative List is : ”,
                          Neg)
```

29. Write a program to read a list of n integers and find their median.

Note The median value of a list of values is the middle one when they are arranged in order. If there are two middle values then take their average. **[NCERT]**

Hint You can use a built-in function to sort the list.

Ans.
```
num = int(input(“Enter the number of
                elements : ”))
list1 = list()
for i in range(num):
  x = int(input(“Enter the integer: ”))
  list1.append(x)
print(“Original list:”,list1)
list1.sort()
print(“Sorted list: ”,list1)
c = len(list1)
if c%2 != 0:
  med = c//2
  print(“Median: ”,list1[med])
else:
  a = list1[c//2]
  b = list1[(c//2) – 1]
  s = a + b
  med = s/2
  print(“Median: ”, med)
```

30. Write a program to read a list of elements. Modify this list, so that it does not contain any duplicate elements, i.e. all elements occurring multiple times in the list should appear only once. **[NCERT]**

Ans.

```
list1=[]
num=int(input(“Enter the number of
                  elements: ”))
for i in range(num):
  x=int(input(“Enter the element: ”))
  list1.append(x)
print(“New list: ”)
print(list(set(list1)))
```

31. Write a program to read a list of elements. Input an element from the user that has to be inserted in the list. Also, input the position at which it is to be inserted. Write a user defined function to insert the element at the desired position in the list. **[NCERT]**

Ans.

```
num = int(input(“Enter the number of elements: ”))
list1 = list()
for i in range(num):
  x = int(input(“Enter the element: ”))
  list1.append(x)
print(“Original list: ”,list1)
print()
pos = int(input(“Enter the index
                   position: ”))
ele = int(input(“Enter the new element: ”))
list1.insert(pos, ele)
print()
print(“New list: ”,list1)
```

32. Write a program to read elements of a list.

(i) The program should ask for the position of the element to be deleted from the list. Write a function to delete the element at the desired position in the list.

(ii) The program should ask for the value of the element to be deleted from the list. Write a function to delete the element of this value from the list. **[NCERT]**

Ans. (i)

```
num = int(input(“Enter the number of
                                  elements: ”))
  list1 = list()
  for i in range(num):
        x = int(input(“Enter the element:
                                  ”))
        list1.append(x)
     print(“Original list: ”,list1)
     print()
     pos = int(input(“Enter the position of
     the element you want to delete: ”))
     del list1 [pos]
     print(“List after deletion:”,list1)
```

(ii)

```
num = int(input(“Enter the number of elements: ”))
list1 = list()
for i in range(num):
      x = int(input(“Enter the element:
                                  ”))
      list1.append(x)
print(“Original list: ”,list1)
x = int(input(“Enter the element you
        want to delete: ”))
list1.remove(x)
print(“List after deletion: ”,list1)
```

33. In Python, list is a type of container in data structures, which is used to store multiple data at the same time. It can store integer, string as well as object in a single list. Lists are mutable which means they can be changed after creation. Each element of a list is assigned a number its position or index. The first index is 0, the second index is 1, the third index is 2 and so on.

Based on the above information, answer the following questions.

(i) List is defined by which type of bracket?

(ii) List contains a sequence of what type of elements?

(iii) Lists are mutable. What is it mean?

(iv) Which method is also used to create list of characters and integers through keyboard?

(v) How to represent the first index of list?

Ans. (i) []

(ii) Heterogeneous

(iii) Lists are mutable, which means they can be changed after creation

(iv) list ()

(v) 0

Chapter Test

Multiple Choice Questions

1. What is the output of following code?

```
list1=[4, 3, 7, 6, 4, 9, 5, 0, 3, 2]
l1=list1[1:10:3]
print(l1)
```

(a) [3, 7, 6] (b) [3, 4, 0]
(c) [3, 6, 9] (d) [7, 9,2]

2. Identify the output of following Python statement.

```
a =[[0, 1, 2], [3, 4, 5, 6]]
b =a [1] [2]
print (b)
```

(a) 2 (b) 1 (c) 4 (d) 5

3. Identify the output of following code.

```
list1 =[2, 3, 9, 12, 4]
list1.insert(4, 17)
list1.insert(2, 23)
print(list1 [–4])
```

(a) 4 (b) 9 (c) 12 (d) 23

4. What will be the output of the following Python code?

```
books = ['Hindi', 'English', 'Computer']
if 'put' in books:
  print(True)
else:
  print(False)
```

(a) True (b) False (c) None (d) Error

5. Identify the output of the following Python statement.

```
list1=[4,3,7,9,15,0,3,2]
s=list1[2:5]
print(s)
```

(a) [7,9,15,0] (b) [3,7,9,15]
(c) [7,9,15] (d) [7,9,15,0,3]

6. What will be the output of following code?

```
list1=[2, 5, 4, [9, 6], 3]
list1[3][2] =10
print(list1)
```

(a) [2, 5, 4, [9, 10], 3] (b) [2, 5, 4, 10, [9, 6], 3]
(c) Index out of range (d) None of these

Short Answer Type Questions

7. What will be the output of the following Python code?

```
list1=[11, 12, 13, 14, 15]
for i in range (1, 5) :
      list1[i–1] = list1[i]
for i in range (0, 5) :
      print(list1[i],end=" ")
```

8. What will be the output of following code?

```
l=[]
for i in range (20, 40) :
      if(i % 7 == 0) and (i % 5 != 0) :
           l . append (str(i))
print ('.'. join(l))
```

9. What are the output of below questions?

```
L=[45, 89, 74, 12, 9, 83]
```

(i) L.remove ()
(ii) L.remove (12)

10. Predict the output.

```
L1=[3, 2, 1]
L2=[5, 9, 8]
```

(i) L1 * L2
(ii) L2.sort ( )
(iii) L1.reverse ( )

11. Write a program to input a list and print it in reverse order.

12. Give the output.

(i)
```
str1 = 'aeiou'
list1 = list(str1)
print(list1)
```

(ii)
```
list1 = [2, 3, 4, 5]
list1. append(1)
print(list1)
```

Long Answer Type Questions

13. Write the best suited method's name for the following conditions.

(i) Remove the value from the list.
(ii) Sort the elements in descending order.
(iii) Calculate the sum of all the elements of list.
(iv) Return the minimum element out of elements.

14. Write program to find the minimum and maximum elements from the entered list.

15. Write program to calculate the sum and mean of the elements which are entered by user.

16. Write program to count the frequency of elements in a list entered by user.

17. Write program to search for an element with its respective index number.

18. Write program to enter the elements of a list and reverse these elements.

Answers

Multiple Choice Questions

1. (b) *2. (d)* *3. (b)* *4. (b)* *5. (c)* *6. (c)*