* List Operations :→

There are various list operations which are capable of changing the overall structure of series and are called as list operations.

(i) Alteration / Updation / Modification :→

Elements in list can be easily altered or updated using a simple technique.

Eg list1 = [10, 20, 30, 40, 50, 60]

list1[2] = 100

print(list1)

Output : [10, 20, 100, 40, 50, 60]

In this way, we can easily update / alter the value.

(ii) Concatenation :→

Concatenation in list means merging or joining two list or more list using concatenation operator '+'.

Page No.....
Date.....

Eg. list1 = [1, 2, 3, 4, 5]
list2 = ['Ajay', 'Aman']

list3 = list1 + list2
print(list3)

Output: [1, 2, 3, 4, 5, 'Ajay', 'Aman']

- While concatenation, both operands should be list type only.
- If we try to concatenate list with other datatype, TypeError occurs.

(iii) Repetition :→

Python allows replication of lists using '*' operator.

Eg: list1 = [1, 2, 3, 4]
print(list1 * 4)

Output: [1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4]

(iv) Membership :→

We can check presence of a value/element in a list using in and not in operators.

```
list1 = ['Ajay', 'Aman', 'Raj']
print ('Aman' in list1)
print ('Yash' not in list1)
```

Output :→ True
True

(v) Slicing :→

Slicing allows us to fetch/obtain new subset list from an existing list.

Eg. list1 = [1, 2, 3, 4, 5, 6, 7, 8]
indexes: 0 1 2 3 4 5 6 7
negative indexes: -8 -7 -6 -5 -4 -3 -2 -1

- `print (list1[0:6])` # print element from 0 to 5 of list1

Output :→ [1, 2, 3, 4, 5, 6]

- `print (list1[0:100])` # print whole list as second index of of range.

Output :→ [1, 2, 3, 4, 5, 6, 7, 8]

- `print (list1[0:7:2])` # print elements by skipping one.

Output :→ [1, 3, 5, 7]

- `print (list1[-7:-3])` # print element from -7 to -4.

Output :→ [2, 3, 4, 5]

- `print (list1[::-1])` # print list is reverse order.

Output :→ [8, 7, 6, 5, 4, 3, 2, 1]