LIST

If we want to represent a group of elements as a single entity where insertion order is preserved and duplicates are allowed, then we should be using lists.

Properties:

1. Duplicates are allowed
2. Insertion order is preserved
3. Heterogeneous objects are allowed
4. Dynamic in nature (can increase or decrease size)

1. To create or represent a list

```
a=[]
print(type(a))
```

Output:

```
<class 'list'>
```

2. To showcase that duplicates are allowed

```
a=[10,20,30,10,10]
print(a)
print(type(a))
```

Output:

```
[10, 20, 30, 10, 10]
<class 'list'>
```

3. To showcase that heterogeneous objects are allowed

```
a=[10,20,30,10,10,'hello',87.6]
print(a)
print(type(a))
```

Output:

```
[10, 20, 30, 10, 10, 'hello', 87.6]
<class 'list'>
```

4. To showcase that lists are dynamic in nature

```
a=[10,20,30,10,10,'hello',87.6]
a.append(2010)
a.remove('hello')
```

```
print(a)
print(type(a))
```

Output:

```
[10, 20, 30, 10, 10, 87.6, 2010]
<class 'list'>
```

- **Split function**

```
l='learning python is good'
mylist=l.split()
print(mylist)
```

Output:

```
['learning', 'python', 'is', 'good']
```

- **Accessing elements of a list**
  1. By using index
  2. By using slice

1. By using index:

- +ve index = left to right

```
l=[10,20,30,40,50,60]
print(l[4])
```

Output:

```
50
```

- -ve index = right to left

```
l=[10,20,30,40,50,60]
print(l[-2])
```

Output:

```
50
```

2. By using slice

```
l=[10,20,30,40,50,60]
print(l[1:5:1])
print(l[1:5:2])
```

Output:

```
[20, 30, 40, 50]
[20, 40]
```

- **Traversing a list**

```
l=[10,20,30,40,50,60]
i=0
while i<len(l):
    print(l[i])
    i=i+1
```

Output:

```
10
20
30
40
50
60
```

- **Important function in list**
  1. len()
  2. count()
  3. index()

1. len(): gives length of list

```
l=[10,20,30,40,50,60]
print(len(l))
```

Output:

```
6
```

2. count(): helps to provide count of an item inside list

```
l=[10,20,30,40,50,21,10,30,10,40]
print(l.count(10))
```

Output:

```
3
```

3. index(): helps to provide the position of an element

```
l=[10,20,30,40,50,21,10,30,10,40]
print(l.index(50))
```

Output:

```
4
```

- **Manipulating elements of list**

1. append()

```
l=[10,20,30,40,50,21,10,30,10,40]
l.append(24)
print(l)
```

Output:

```
[10, 20, 30, 40, 50, 21, 10, 30, 10, 40, 24]
```

2. Insert()

```
l=[10,20,30,40,50,21,10,30,10,40]
l.insert(3,24)
print(l)
```

Output:

```
[10, 20, 30, 24, 40, 50, 21, 10, 30, 10, 40]
```

3. extend()

```
l=[10,20,30,40,50,21]
l2=['hello python',670]
l.extend(l2)
print(l)
```

Output:

```
[10, 20, 30, 40, 50, 21, 'hello python', 670]
```

4. remove() : remove by element

```
l=[10,20,30,40,50,21]
l2=['hello python',670]
l.extend(l2)
l.remove('hello python')
print(l)
```

Output:

```
[10, 20, 30, 40, 50, 21, 670]
```

5. pop(): remove by index

```
l=[10,20,30,40,50,21]
l2=['hello python',670]
l.extend(l2)
l.pop(2)
print(l)
```

Output:

```
[10, 20, 40, 50, 21, 'hello python', 670]
```

- **Ordering elements of list**

1. reverse()

```
l=[10,20,30,40,50,21]
l.reverse()
print(l)
```

Output:

```
[21, 50, 40, 30, 20, 10]
```

2. sort(): by default ascending

```
l=[10,20,30,50,54,98,56,76]
l.sort()
print(l)
```

Output:

```
[10, 20, 30, 50, 54, 56, 76, 98]
```

- **Aliasing**

Example:

```
x=[10,20,30,40]
y=x
print(id(x))
print(id(y))
y[0]=20
print(x)
print(y)
print(id(x))
print(id(y))
```

Output:

```
1826701973952
1826701973952
[20, 20, 30, 40]
[20, 20, 30, 40]
1826701973952
1826701973952
```

- **Cloning:**

- Slice operator:

```
x=[10,20,30,40]
y=x[::]
print(id(x))
print(id(y))
y[0]=403
print(x)
print(y)
print(id(x))
print(id(y))
```

Output:

```
2407283366336
2407283319488
[10, 20, 30, 40]
[403, 20, 30, 40]
2407283366336
2407283319488
```

- Copy operator

```
x=[10,20,30,40]
y=x.copy()
print(id(x))
print(id(y))
y[0]=245
print(x)
print(y)
print(id(x))
print(id(y))
```

Output:

```
2628619868608
2628619821760
[10, 20, 30, 40]
[245, 20, 30, 40]
2628619868608
2628619821760
```

- **Clearing list**

```
x=[10,20,30,40]
x.clear()
print(x)
```

Output:

```
[ ]
```

Example:

1.

```
x=["Dog","Cat","Rat"]
y=["Dog","Cat","Rat"]
z=["DOG","CAT","RAT","Bat"]
print(x==y)
print(x==z)
print(x!=z)
```

Output:

```
True
False
True
```

2.

```
y=[100,200,300,400]
print(100 in y)
print(10 not in y)
```

Output:

```
True
True
```

3.

```
y=[100,2020,3300,4400,[150,620,730,40]]
print(y[2])
print(y[3])
print(y[4][3])
```

Output:

```
3300
4400
40
```

4.

```
x=[[1,2,3],[4,5,6],[7,8,9]]
print(x)
print("Row Wise:")
for elements in x:
    print(elements)
```

Output:

```
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
Row Wise:
[1, 2, 3]
[4, 5, 6]
[7, 8, 9]
```

- **List Comprehension**

The process of creating a list from other sequences.(condition not mandatory)

Example:

```
l=[x for x in range(1,11)]
print(l)
```

Output:

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

Example:

```
words=["bhasker","isha","kabin","nikita","ram","shyam"]
l=[w[0] for w in words]
print(l)
```

Output:

```
['b', 'i', 'k', 'n', 'r', 's']
```

Example:

```
s="The quick brown fox jumps over the lazy dog".split()
p=[[x.upper(),len(x)]for x in s]
print(p)
```

Output:

```
[['THE', 3], ['QUICK', 5], ['BROWN', 5], ['FOX', 3],
['JUMPS', 5], ['OVER', 4], ['THE', 3], ['LAZY', 4], ['DOG',
3]]
```