



Nested List as Matrix:

In Python we can represent matrix by using nested lists.

```
1) n=[[10,20,30],[40,50,60],[70,80,90]]
2) print(n)
3) print("Elements by Row wise:")
4) for r in n:
5)     print(r)
6) print("Elements by Matrix style:")
7) for i in range(len(n)):
8)     for j in range(len(n[i])):
9)         print(n[i][j],end=' ')
10) print()
11)
12) Output
13) D:\Python_classes>py test.py
14) [[10, 20, 30], [40, 50, 60], [70, 80, 90]]
15) Elements by Row wise:
16) [10, 20, 30]
17) [40, 50, 60]
18) [70, 80, 90]
19) Elements by Matrix style:
20) 10 20 30
21) 40 50 60
22) 70 80 90
```

List Comprehensions:

It is very easy and compact way of creating list objects from any iterable objects(like list,tuple,dictionary,range etc) based on some condition.

Syntax:

```
list=[expression for item in list if condition]
```

Eg:

```
1) s=[ x*x for x in range(1,11)]
2) print(s)
3) v=[2**x for x in range(1,6)]
4) print(v)
5) m=[x for x in s if x%2==0]
6) print(m)
7)
8) Output
9) D:\Python_classes>py test.py
10) [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```