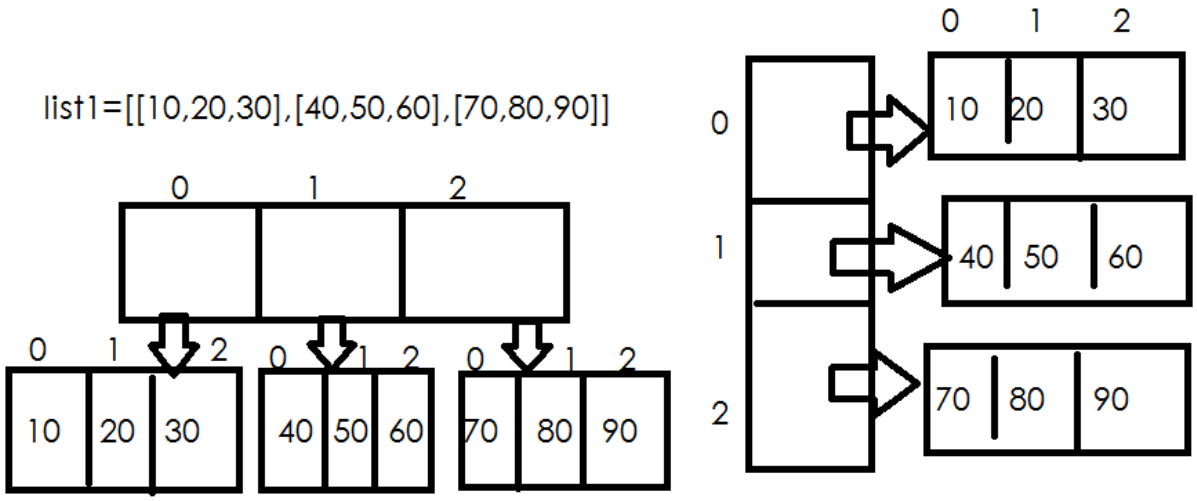
Nested list
List within list is called nested list.
Defining a list as an element inside list is called nested list.
Using nested list we can organize data in rows and columns (OR) using nested we can store more data.



To read and write the elements inside nested list, we have to use two subscripts or indexes
Nested list can be used as matrix.

Example:

```
list1=[[10,20,30],[40,50,60],[70,80,90]]

# using index
print(list1[0])
print(list1[1])
print(list1[2])
print(list1[0][0],list1[0][1],list1[0][2])
print(list1[1][0],list1[1][1],list1[1][2])
print(list1[2][0],list1[2][1],list1[2][2])


#using for loop

for a in list1:
    for b in a:
        print(b,end=' ')
```

```
    print()

#using for loop with index
for i in range(3): # 0 1 2
    for j in range(3): # 0 1 2
        print(list1[i][j],end=' ')
    print()
```

Output:

```
[10, 20, 30]
[40, 50, 60]
[70, 80, 90]
10 20 30
40 50 60
70 80 90
10 20 30
40 50 60
70 80 90
10 20 30
40 50 60
70 80 90
```

Reading elements from nested list using slicing

Slicing is used to read list from a list/sub list

```
list1=[[1,2,3],[4,5,6],[7,8,9]]
list2=list1[::-1]
>>> print(list1)
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
>>> print(list2)
[[7, 8, 9], [4, 5, 6], [1, 2, 3]]
>>> list3=list2[-2:]
>>> print(list3)
[[4, 5, 6], [1, 2, 3]]
>>> list4=list1[-2:]
>>> print(list4)
[[4, 5, 6], [7, 8, 9]]
>>> list5=list1[:2]
>>> print(list5)
[[1, 2, 3], [4, 5, 6]]
```

Example:

```
# write a program read 3x3 matrix and display


matrix=[]

for i in range(3):
    row=[]
    for j in range(3):
        value=int(input("enter any value"))
        row.append(value)
    matrix.append(row)

print(matrix)
for row in matrix:
    for value in row:
        print(value,end=' ')
    print()
```

Output:

```
enter any value1
enter any value2
enter any value3
enter any value4
enter any value5
enter any value6
enter any value7
enter any value8
enter any value9
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
1 2 3
4 5 6
7 8 9
```

Example:

```
# write a program to read 3 students and 3 subject marks
# and display

stud_marks=[]
```

```
for i in range(3):
    stud=[]
    for j in range(3):
        s=int(input("enter marks"))
        stud.append(s)
    stud_marks.append(stud)

for stud in stud_marks:
    print(stud,sum(stud),sum(stud)/3,"pass" if stud[0]>=40 and stud[1]>=40
and stud[2]>=40 else "fail")
```

Output:

```
enter marks70
enter marks80
enter marks90
enter marks30
enter marks50
enter marks60
enter marks60
enter marks70
enter marks80
[70, 80, 90] 240 80.0 pass
[30, 50, 60] 140 46.666666666666664 fail
[60, 70, 80] 210 70.0 pass
```

Displays

For constructing a list, a set or a dictionary Python provides special syntax called “displays”, each of them in two flavors:

1. either the container contents are listed explicitly, or
   [10,20,30,40,50]
2. they are computed via a set of looping and filtering instructions, called a *comprehension*.

Comprehension allows to create list, set and dictionary using for loop and if condition.

List comprehension

Syntax: [expression for variable in iterable if test]