List Comprehension

▼ List Comprehension

It is one of the great features of Python programming as it can create the most complicated lists through only a few lines of code.

▼ Example

```
# firstly create the nor
natural_number=[]
for i in range(1,10,1):
    natural_number.apper
```

```
#now we do this using li
natural_number=[i for i
print(natural_number)
```

The list comprehension method creates a Python list in a single line of code. The core component of a list comprehension is the for loop. To create a list using the list comprehension method, first, write the square brackets []. Then inside the brackets, use the for loop.

▼ Q) Create a Python list containing even numbers from 1 and 20 using a conventional list creation approach

```
for i in range(1,21):
  if i%2==0:
    even_numbers.append(
```

now this in List comprehension

```
even_nums=[i for i in ra
```

Nested List using List comprehension method

The creation of nested lists manually is a tedious job. We have to take care of indentation so that the list is readable in the code form. You may miss out on commas, brackets, spaces, etc. while creating a nested list. Hence, the manual nested list creation process becomes time-consuming.Using list comprehension method, we can create a nested list more neatly and quickly.

▼ Example

Let's assume that we want to join two lists; one containing names of the planets and another containing their corresponding diameters.

planets = ['Mercury', 'Venus', 'Earth', 'Mars', 'Jupiter', 'Saturn',

'Uranus', 'Neptune', 'Pluto']
diameters = [4879, 12104,
12756, 6972, 142984, 120536,
51118, 49528, 2370]

```
planets = ['Mercury', 'V
diameters = [4879, 12104
a=[]
for i in range(len(plane
    b=[]
    b.append(planets[i])
    b.append(diameters[i
    a.append(b)
print(a)
```

```
planets = ['Mercury', 'V
diameters = [4879, 12104
b=[[planets[i],diameters
print(b)
```

▼ Example 2

Now, combine four lists into one nested list. They are:

planets = ['Mercury', 'Venus',
'Earth', 'Mars', 'Jupiter', 'Saturn',
'Uranus', 'Neptune', 'Pluto']
diameters = [4879, 12104,
12756, 6972, 142984, 120536,

51118, 49528, 2370]
densities = [5427, 5243, 5514, 3933, 1326, 687, 1271, 1638, 2095]
gravities = [3.7, 8.9, 9.8, 3.7, 23.1, 9.0, 8.7, 11.0, 0.7]

```
planets = ['Mercury', 'V
diameters = [4879, 12104
densities = [5427, 5243,
gravities = [3.7, 8.9, 9
ans=[]
for i in range(len(plane
    a=[planets[i],diamet
    ans.append(a)
print(ans)
```

Now , using list comprehension method

```
Now, using the same proc
planets = ['Mercury', 'V
diameters = [4879, 12104
densities = [5427, 5243,
gravities = [3.7, 8.9, 9
ans=[[planets[i],diamete
print(ans)
```

▼ Now, let's create a nested list using the list comprehension method which has 4 identical lists

containing 0 as its each item.
Here's the desired output:

[[0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0]]

```
b=[]
for i in range(4):
    a=[]
    for j in range(4):
        a.append(0)
    b.append(a)
print(b)
```

Now using List comprehension method

```
b=[[0 for i in range(4)]
print(b)
```

▼ [[[1, 1], [1, 1], [1, 1], [1, 1]],
[[1, 1], [1, 1], [1, 1], [1, 1]],
[[1, 1], [1, 1], [1, 1], [1, 1]]]

here as you ca see there is a 3D list with 3 roes each conting a 2D list of 2 ones 4 times

```
b=[[[1 for i in range(2)
print(b)
```

▼ [[11, 12],
[21, 22]]

```
b=[[10*j+i for i in rang
print(b)
```

▼ [[[111, 112, 113],[121, 122, 123], [131, 132, 133]],[[211, 212, 213], [221, 222, 223],[231, 232, 233]], [[311, 312, 313],[321, 322, 323], [331, 332, 333]]]

[[[111, 112, 113],

[121, 122, 123],

[131, 132, 133]],

[[211, 212, 213],

[221, 222, 223],

[231, 232, 233]],

[[311, 312, 313],

[321, 322, 323],

[331, 332, 333]]]

```
b=[[[100*k+10*j+i for i
print(b)
```