Python - List Comprehension

List comprehension in Python is an easy and compact syntax for creating a list from a string or another list. It is a very concise way to create a new list by performing an operation on each item in the existing list. List comprehension is considerably faster than processing a list using the for loop.

List Comprehension Signature:

```
[x for x in iterable if expression]
```

Suppose we want to separate each letter in a string and put all letters in a list object. We can do it using a for loop, as shown below:

Example: Separate Letters from String

```
chars=[]
for ch in 'TutorialsTeacher':
    chars.append(ch)
print(chars)
```

The chars list object is displayed as follows:

Result:

```
['T', 'u', 't', 'o', 'r', 'i', 'a', 'l', 's', 'T', 'e', 'a', 'c', 'h', 'e',
'r']
```

The same result can be easily achieved using a list comprehension technique.

```
>>> [ch for ch in 'TutorialsTeacher']
['T', 'u', 't', 'o', 'r', 'i', 'a', 'l', 's', 'T', 'e', 'a',
'c', 'h', 'e', 'r']
```

The following example uses a list comprehension to build a list of squares of the numbers between 1 and 10.

```
>>>squares = [x*x for x in range(11)]
>>>squares
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

It is possible to use nested loops in a list comprehension expression. In the following example, all combinations of items from two lists in the form of a tuple are added in a third list object.

```
>>>list1=[1,2,3]
>>>list2=[4,5,6]
>>>CombLst=[(x,y) for x in list1 for y in list2]
>>>CombLst
[(1, 4), (1, 5), (1, 6), (2, 4), (2, 5), (2, 6), (3, 4), (3, 5),
(3, 6)]
```

We can even have the if condition in a list comprehension. The following statement will result in a list of all even numbers between 1 and 20.

```
>>> [x for x in range(21) if x%2==0]
[0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
```

We can use nested if conditions with list comprehension.

```
>>> [x for x in range(21) if x%2==0 if x%5==0]
[0, 10, 20]
```

The following example demonstrates the *if..else* loop with list comprehension.

```
>>> obj=["Even" if i%2==0 else "Odd" for i in range(10)]
>>> obj
['Even', 'Odd', 'Even', 'Odd', 'Even', 'Odd', 'Even', 'Odd',
'Even', 'Odd']
>>> obj=[str(i)+" = Even" if i%2==0 else str(i)+" = Odd" for i
in range(10)]
>>> obj ['0 = Even', '1 = Odd', '2 = Even', '3 = Odd', '4 =
Even', '5 = Odd', '6 = Even', '7 = Odd', '8 = Even', '9 = Odd']
```

One of the applications of list comprehension is to flatten a list comprising of multiple lists into a single list.

```
>>>matrix=[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
>>>flatList=[num for row in matrix for num in row]
>>>flatList
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```