List Comprehensions

Agenda

1. List comprehension
2. Compare List comprehension with FOR loop

List comprehension

- Comprehensions allow us to create a collection from some other iterable in a very short way.
- List comprehension is an elegant way to define and create lists based on existing lists.
- List comprehension is generally more compact and faster than normal functions and loops for creating list

Syntax of List Comprehension

```
[expression for item in list]
```

Program (1/2) : compare with For and LC

create natural number less than or equal to 50, x is a perfect square using FOR

```
In FOR loop :

for i in range(1,50):
   if int(i**0.5)==i**0.5:
     print(i)
```

```
Output:
1
4
9
16
25
36
49
```

```
FOR loop pattern:

for (set of values to iterate):
  if (conditional filtering):
    output_expression()
```

Program (2/2)

create natural number less than or equal to 50, x is a perfect square using LC

```
List Comprehension Pattern

[ output_expression() for(set of values to iterate) if(conditional filtering) ]
```

```
In List comprehension:

[print(i) for i in range(1,50) if int(i**0.5)==i**0.5]
```

```
Output
1
4
9
16
25
36
49
```

Program 2

Iterating through a string

```
c_letters = [ letter for letter in 'Wipro' ]
print(c_letters)
```

```
Output:
['W', 'i', 'p', 'r', 'o']
```

Program 2

Create an output list which contains only the even numbers using List comprehensions using conditions

```
number_list = [ x for x in range(20) if x % 2 == 0]
print(number_list)
```

Output:
[0, 2, 4, 6, 8, 10, 12, 14, 16, 18]

Program 3

Nested IF with List Comprehension

```
mylist = [y for y in range(100) if y % 2 == 0 if y % 5 == 0]
print(mylist)
```

```
Output:
[0, 10, 20, 30, 40, 50, 60, 70, 80, 90]
```

Program 3

Dictionary Comprehension

```
def captical_lc(keys, values):
    return { keys[i] : values[i] for i in range(len(keys)) }

country = ['India', 'Nepal', 'Bangladesh']
capital = ['New Delhi', 'Kathmandu', 'Dhaka']

print (str(captical_lc(country, capital)))
```

```
Output:
{'India': 'New Delhi', 'Nepal': 'Kathmandu', 'Bangladesh': 'Dhaka'}
```

Thank you