List Comprehension

- It is a method to create a list in a simple way from existing list (or) other iterables.
- A list can be generated from a list , tuple , string or a set.
- A list can be empty, it is denoted as []
- append() in list adds an element to a list
- A loop can be used to append values in a list of particular range.

Example 1:

```
>>> l1 = []
>>> for x in range(10):
... l1.append(x)
...
>>> l1
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>>
```

• The above process can be done in a simple way using 'List Comprehension'

Syntax of List Comprehension:

```
L1 = [Expression for item in iterable]
```

Example 2:

```
>>> l1 =[x for x in range (10)]
>>> l1
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> |
```

- We can observe from this example that how using list comprehension we simplified using example 1
- · Some other examples of list comprehension are

Example 3:

here we are using an expression I.e for numbers in range 1 to 6 which give its square value.

Example 4:

```
>>> | 13 = [x for x in (10,5,7,8,12,13) if x%2 == 0] | 13 | [10, 8, 12] | >>> |
```

We are defining a tuple and writing a condition that says for given numbers in the tuple, divide it by 2 give the remainder of those numbers divisible by 2 as result.

Example 5:

here x.lower() is an expression which means take all the string letters in lower case.

- You can also read a list from a keyboard
- Suppose you want to take input from keyboard (which is str type) and want to convert it into list then this can also be done in a simplified way using list comprehension.

Example 5: