Python List Comprehension

Anoop S Babu
Faculty Associate
Dept. of Computer Science & Engineering
bsanoop@am.amrita.edu

List Comprehension

- A concise way to create a list
- Use when each element in the list
  - is a result of some operations on another sequence
  - or has to satisfy a specific condition

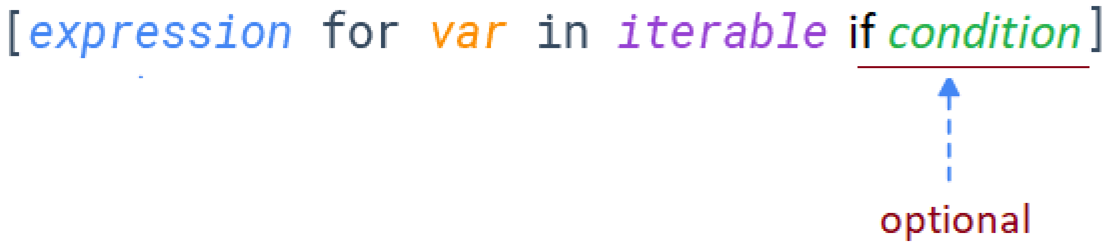
List Comprehension

```
#creating list using for loop
>>> myList = []
>>> for i in range(5):
        myList.append(i+1)
>>> print(myList)
[1, 2, 3, 4, 5]

# list comprehension
>>> myList = [i+1 for i in range(5)]
>>> print(myList)
[1, 2, 3, 4, 5]
```

List Comprehension - Syntax

- Consists of three sections and an optional one



```
[expression for var in iterable if condition]
```

List Comprehension - example

```
# create a list of cubes within a range
>>> cubes = [x**3 for x in range(10)]
>>> print(cubes)
[0, 1, 8, 27, 64, 125, 216, 343, 512, 729]

# create a list of even numbers between 2 limits
>>>evenList = [x for x in range(10,20) if x%2 == 0]
>>> print(evenList)
[10, 12, 14, 16, 18]
```

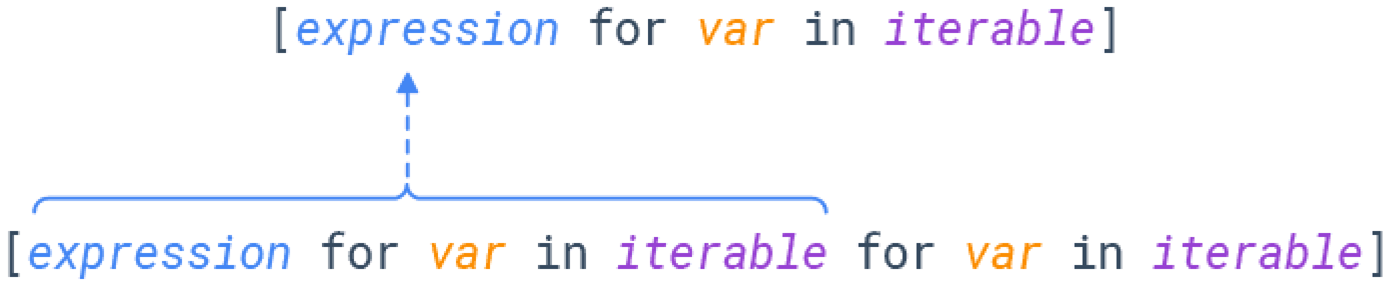
List Comprehension – multiple variables

- Expression can also contain multiple variables

```
#Pythagorean triples
>>> triples=[(x,y,z) for x in range(1,11)
        for y in range(x,11)
        for z in range(y,11)
        if z*z==x*x+y*y]
>>> print(triples)
[(3, 4, 5), (6, 8, 10)]
```

Nested List Comprehension

- Here expression is another list comprehension



```
[expression for var in iterable]

[expression for var in iterable for var in iterable]
```

Nested List Comprehension - example

```
>>> matrix = [[2,4,6],
          [8,9,10]]
>>> transpose = [[row[i] for row in matrix] for i in range(3)]
>>> print(transpose)
[[2, 8], [4, 9], [6, 10]]
```

list comprehension

Nested List Comprehension – example 2

```
#create a multiplication table of numbers 1 to 4 as a matrix
>>> table = [[x*y for y in range(1,11)] for x in range(1,5)]
>>> print(table)
[[1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
        [2, 4, 6, 8, 10, 12, 14, 16, 18, 20],
        [3, 6, 9, 12, 15, 18, 21, 24, 27, 30],
        [4, 8, 12, 16, 20, 24, 28, 32, 36, 40]]
```