SNS COLLEGE OF ENGINEERING
Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF CSE (IoT & CYBER SECURITY INCLUDING BLOCKCHAIN TECHNOLOGY)

python

19IT103 – COMPUTATIONAL THINKING AND PYTHON PROGRAMMING

- **A readable, dynamic, pleasant, flexible, fast and powerful language**

Session wise Agenda

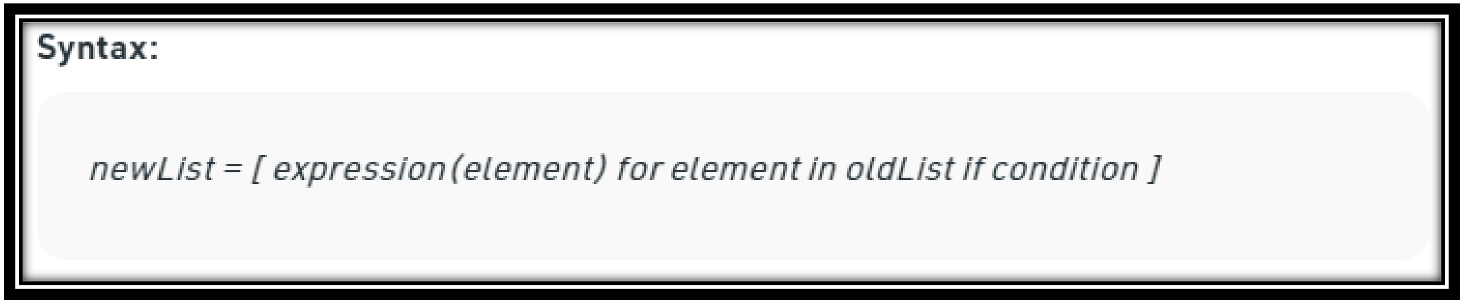
- **session 1 - List (Operations, Slice, Methods)**
- **Session 2 - List (Loop, Mutability)**
- **Session 3 - List (Aliasing, Cloning, Parameters)**
- **Session 4 - Tuples (Assignment, as return value)**
- **Session 5 - Dictionaries (operations and methods)**
- **Session 6 - Advance List processing, List Comprehension**
- Session 7 - Simple Sort, Histogram
- Session 8 - Student Mark Statement
- Session 9 - Retail Bill preparation

Recap

- Dictionary → a non-sequential data type which hold the value in Key:Value pairs
- The key and value can be of any type.
- Dictionary can be traversed with loop
- Dictionary have in-built methods to perform operations on values.

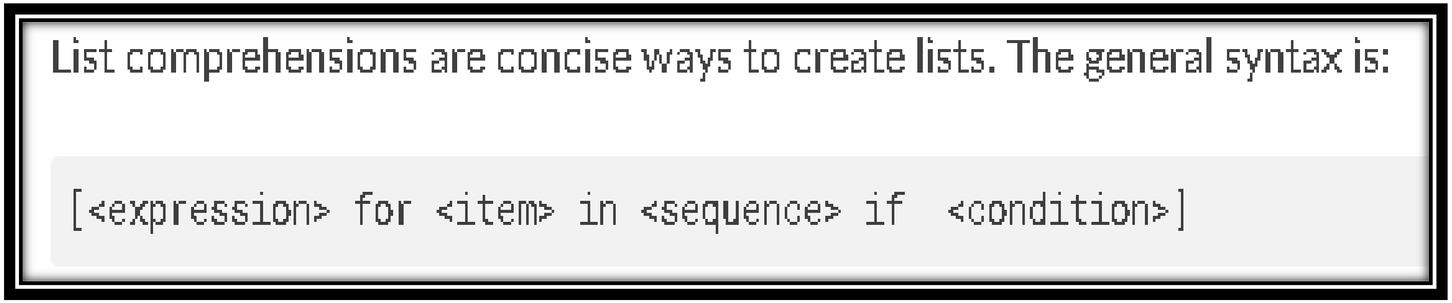
List Comprehension

- List comprehensions are used for creating new lists from other iterables like tuples, strings, arrays, lists, etc.



Syntax:

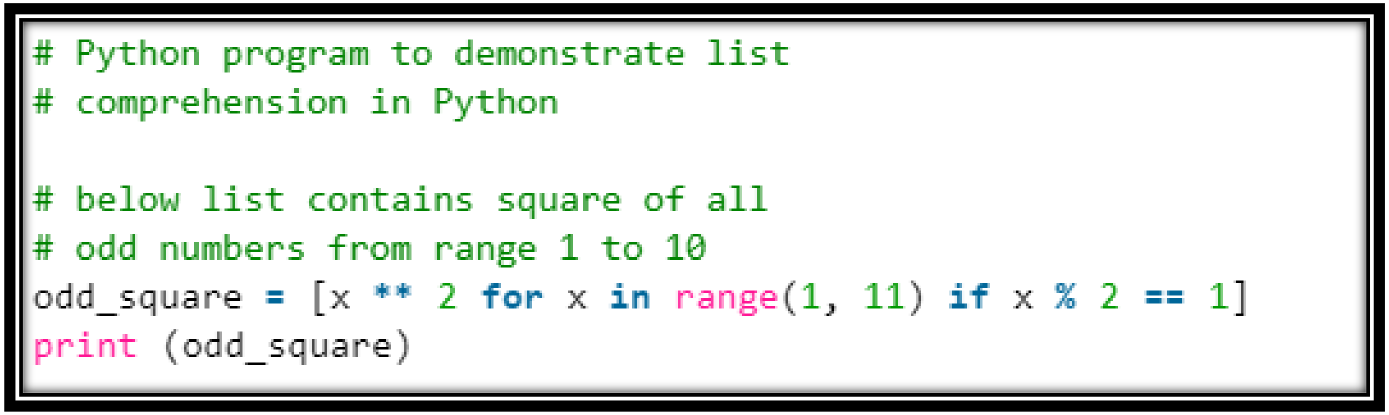
```
newList = [ expression(element) for element in oldList if condition ]
```



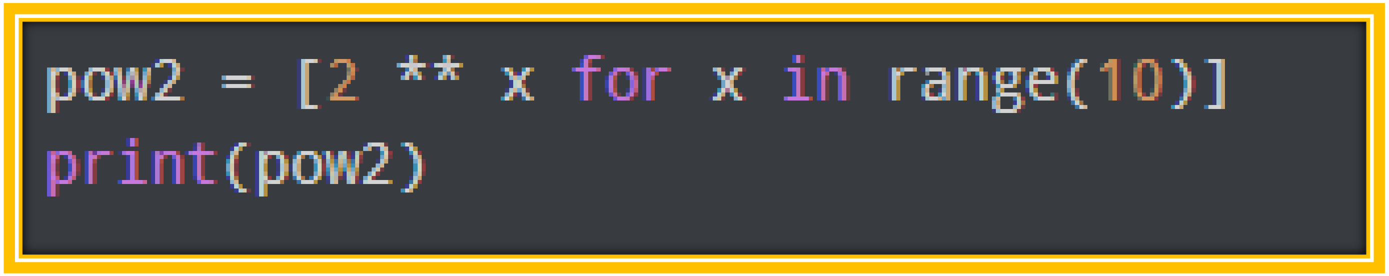
List comprehensions are concise ways to create lists. The general syntax is:

```
[<expression> for <item> in <sequence> if  <condition>]
```

List Comprehension

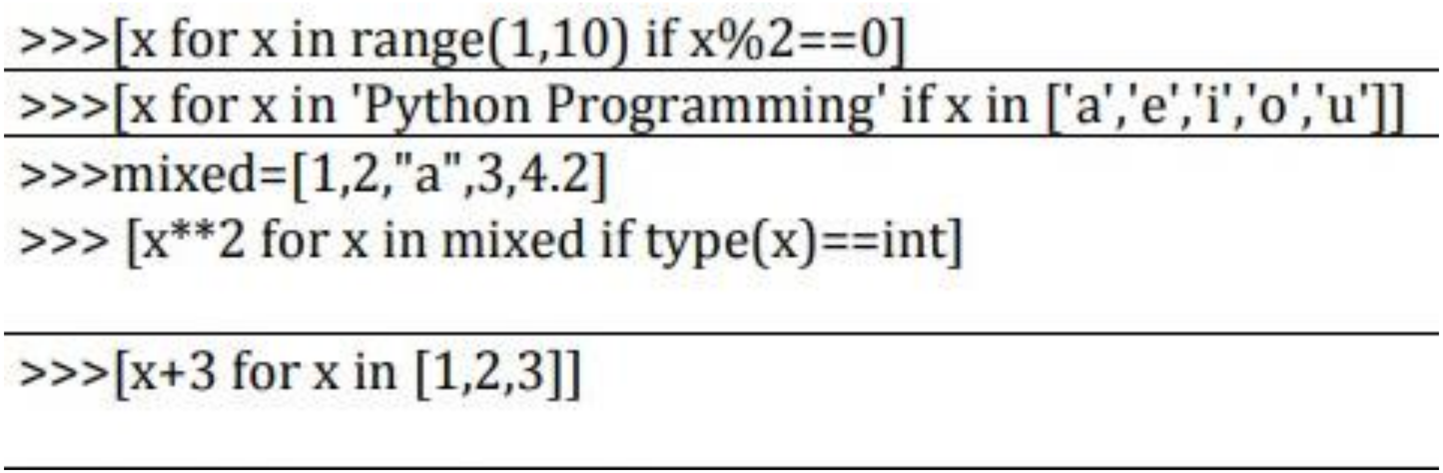


```
# Python program to demonstrate list
# comprehension in Python

# below list contains square of all
# odd numbers from range 1 to 10
odd_square = [x ** 2 for x in range(1, 11) if x % 2 == 1]
print (odd_square)
```

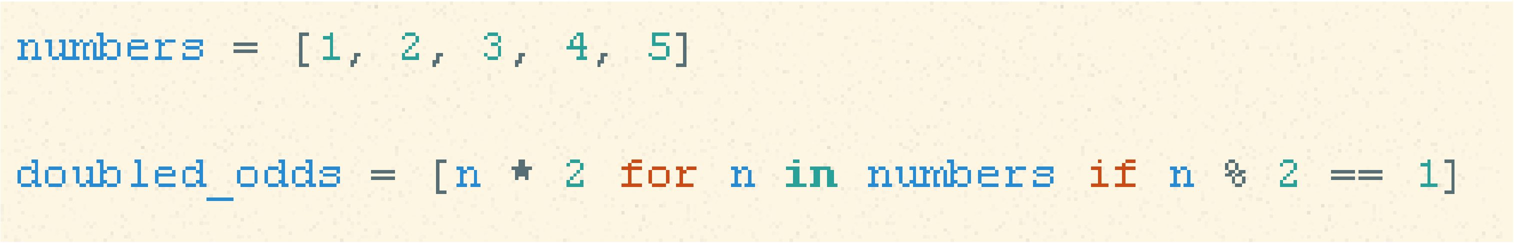


```
pow2 = [2 ** x for x in range(10)]
print(pow2)
```

List Comprehension

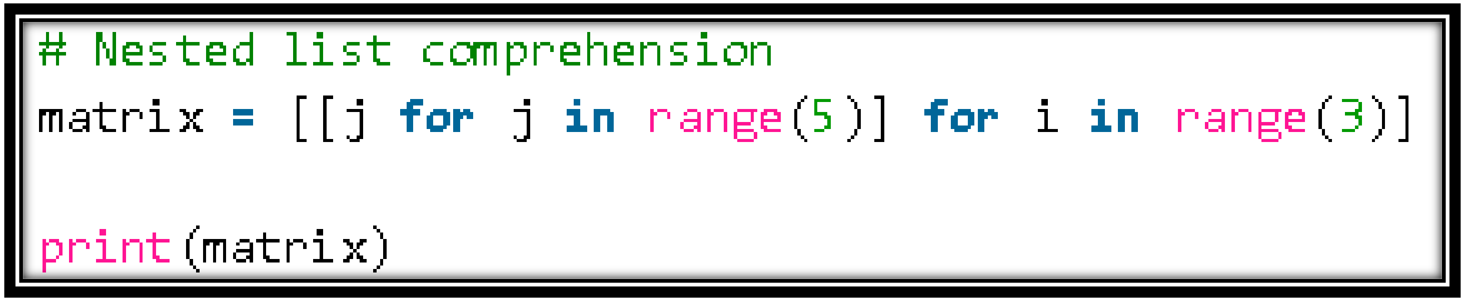


```
>>>[x for x in range(1,10) if x%2==0]
>>>[x for x in 'Python Programming' if x in ['a','e','i','o','u']]
>>>mixed=[1,2,"a",3,4.2]
>>> [x**2 for x in mixed if type(x)==int]

>>>[x+3 for x in [1,2,3]]
```

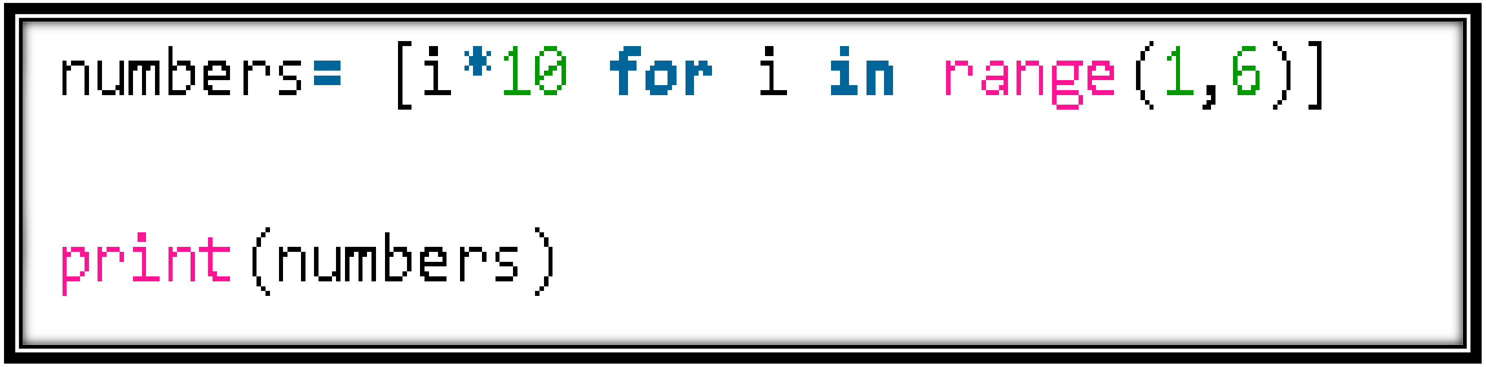


```
numbers = [1, 2, 3, 4, 5]

doubled_odds = [n * 2 for n in numbers if n % 2 == 1]
```

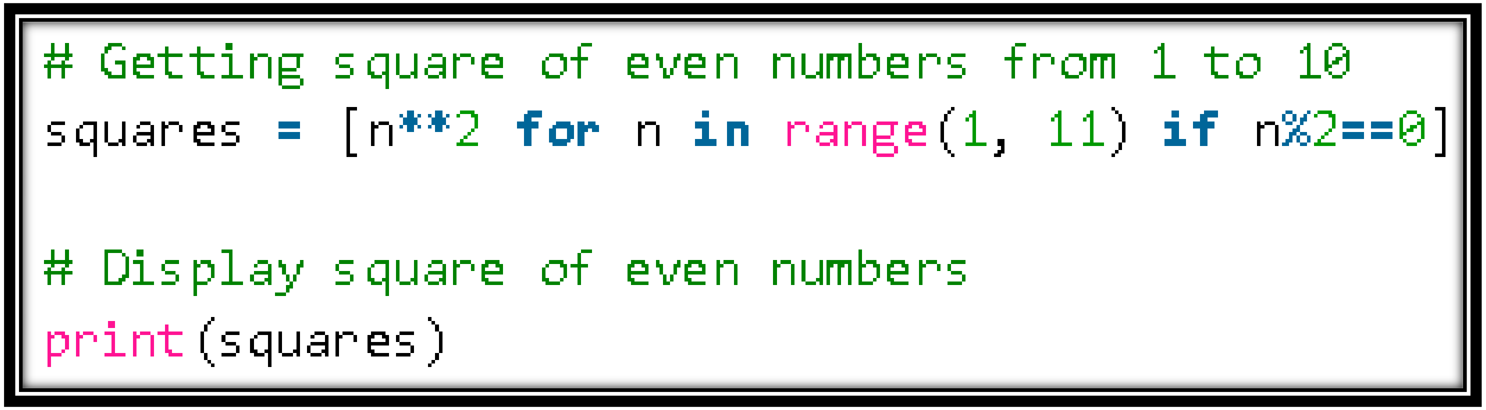
List Comprehension



```
# Nested list comprehension
matrix = [[j for j in range(5)] for i in range(3)]

print(matrix)
```

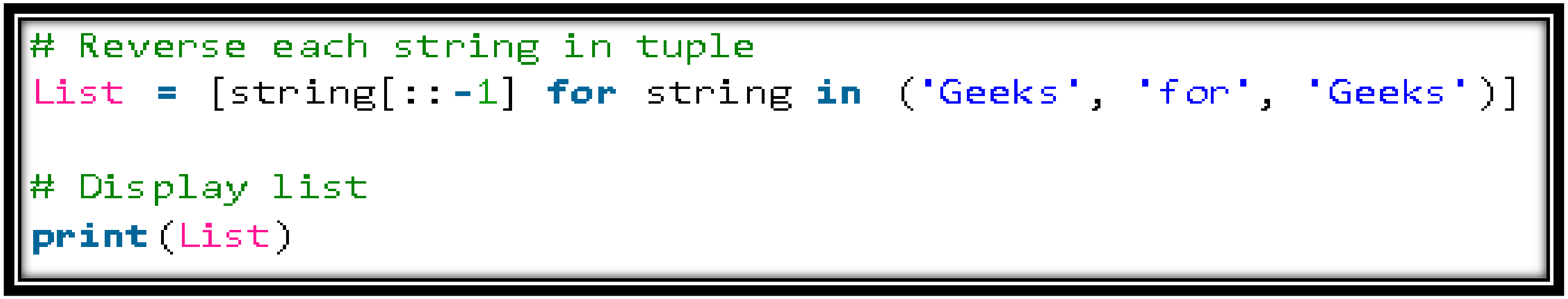


```
numbers= [i*10 for i in range(1,6)]

print(numbers)
```

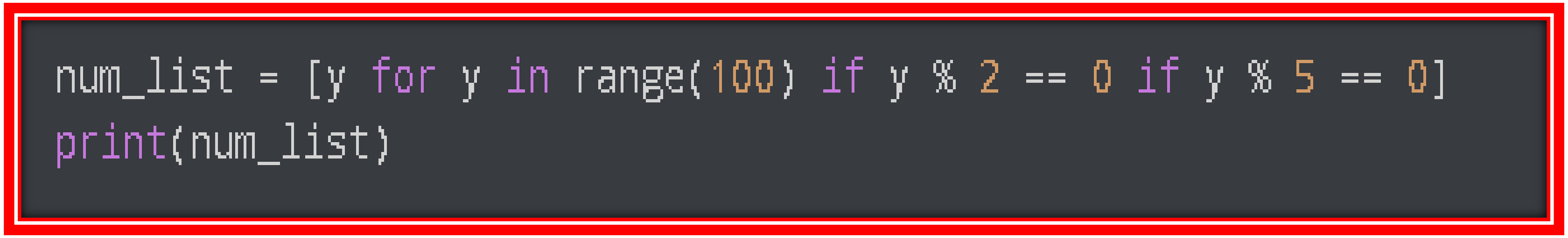


```
# Getting square of even numbers from 1 to 10
squares = [n**2 for n in range(1, 11) if n%2==0]

# Display square of even numbers
print(squares)
```

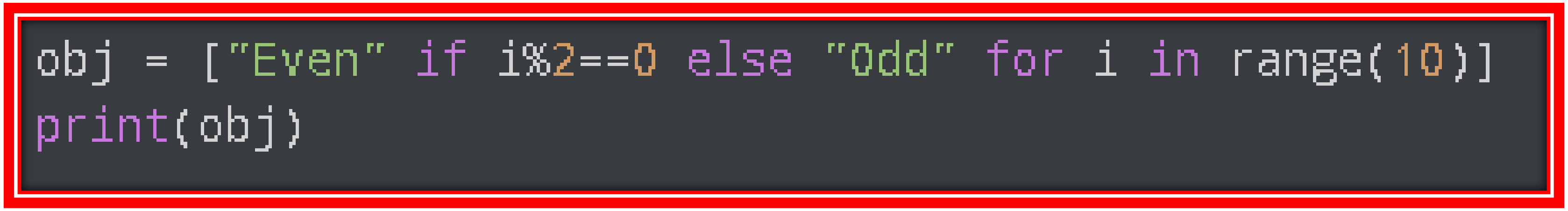
List Comprehension



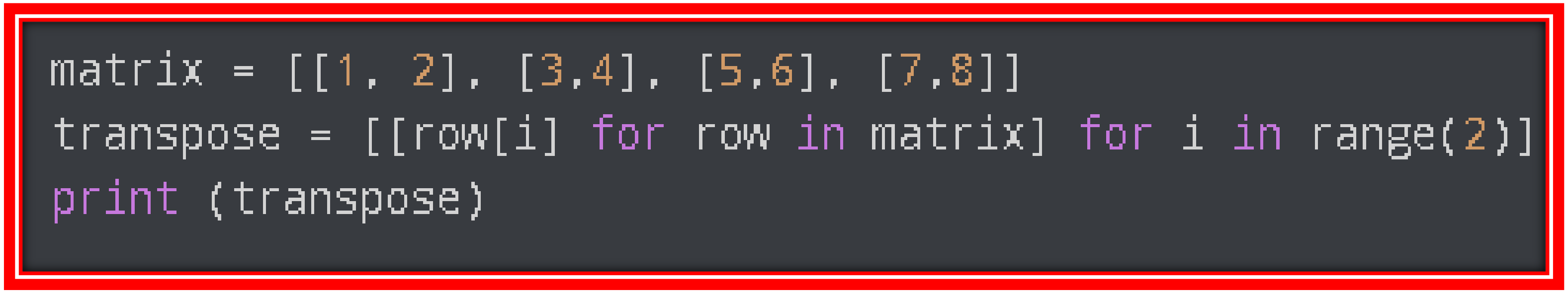
```
# Reverse each string in tuple
List = [string[::-1] for string in ('Geeks', 'for', 'Geeks')]

# Display list
print(List)
```



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

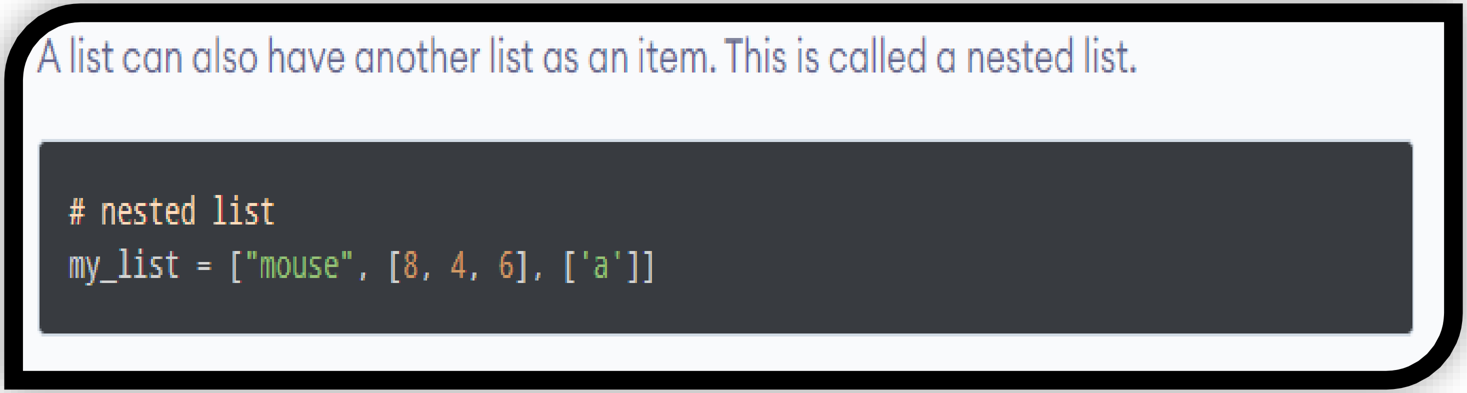


```
obj = ["Even" if i%2==0 else "Odd" for i in range(10)]
print(obj)
```



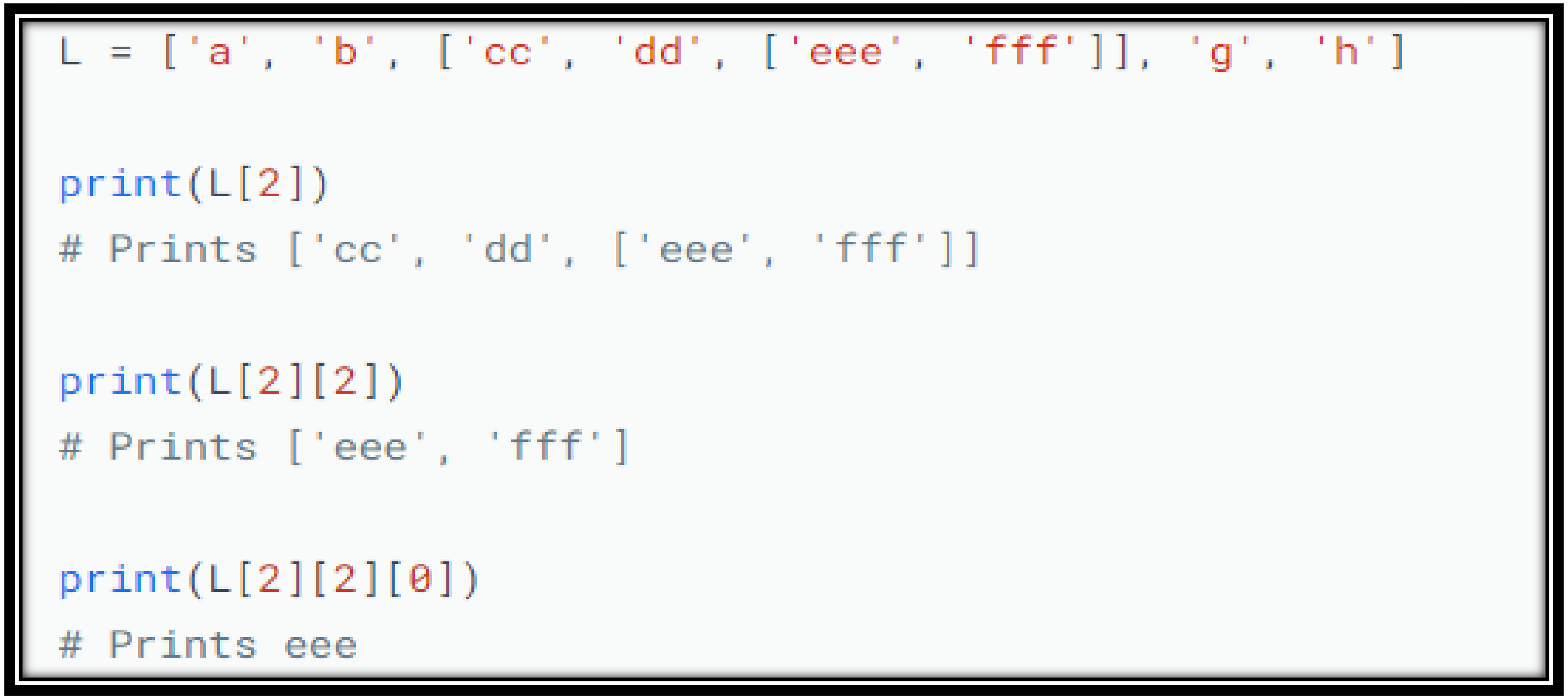
```
matrix = [[1, 2], [3,4], [5,6], [7,8]]
transpose = [[row[i] for row in matrix] for i in range(2)]
print (transpose)
```

Nested List



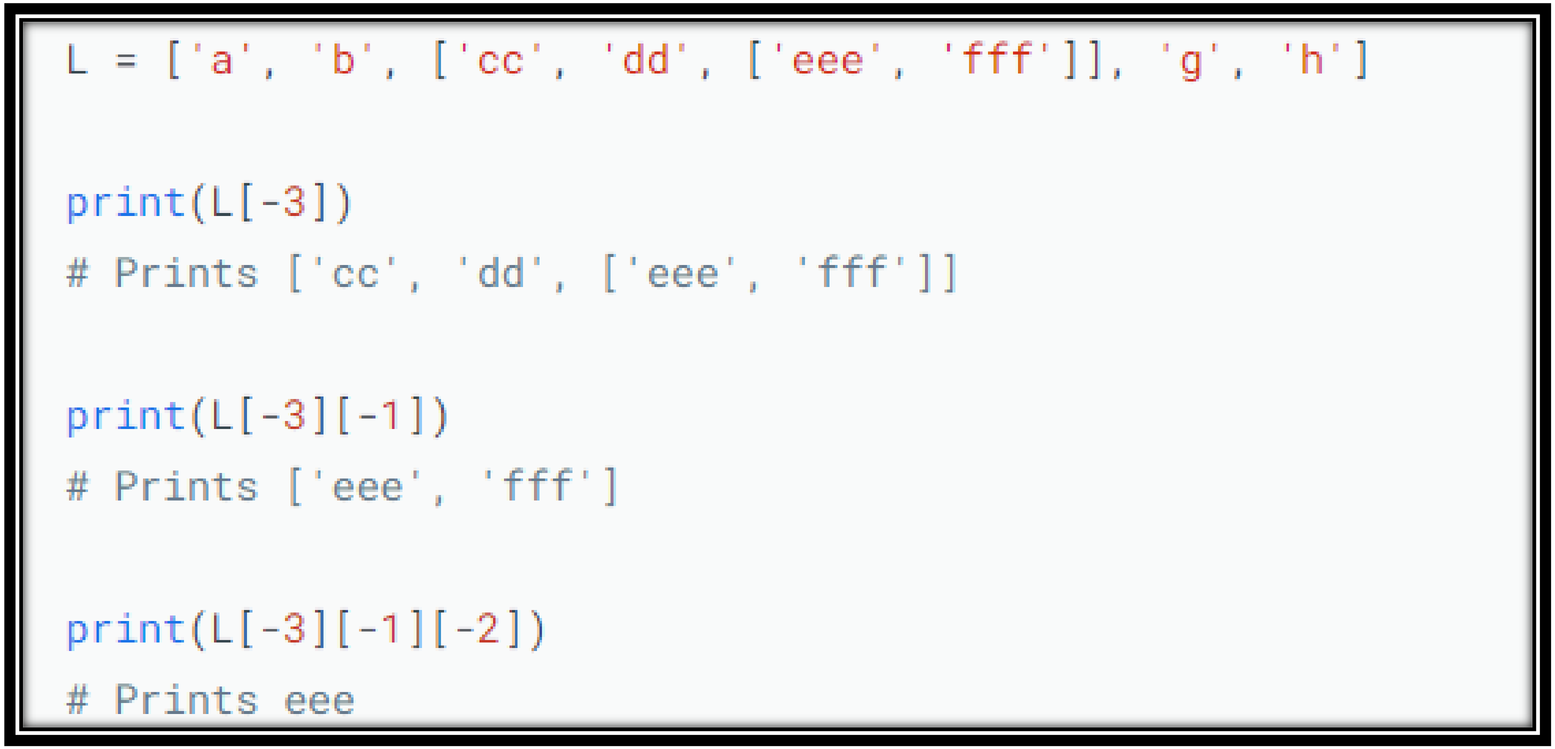
A list can also have another list as an item. This is called a nested list.

```
# nested list
my_list = ["mouse", [8, 4, 6], ['a']]
```



```
L = ['a', 'b', ['cc', 'dd', ['eee', 'fff']], 'g', 'h']

print(L[2])
# Prints ['cc', 'dd', ['eee', 'fff']]

print(L[2][2])
# Prints ['eee', 'fff']

print(L[2][2][0])
# Prints eee
```

Nested List



```
L = ['a', 'b', ['cc', 'dd', ['eee', 'fff']], 'g', 'h']

print(L[-3])
# Prints ['cc', 'dd', ['eee', 'fff']]

print(L[-3][-1])
# Prints ['eee', 'fff']

print(L[-3][-1][-2])
# Prints eee
```

Summary

- List Comprehension → creating a list from other iterable.
- Nested List → to create a list within the list.

THANK YOU