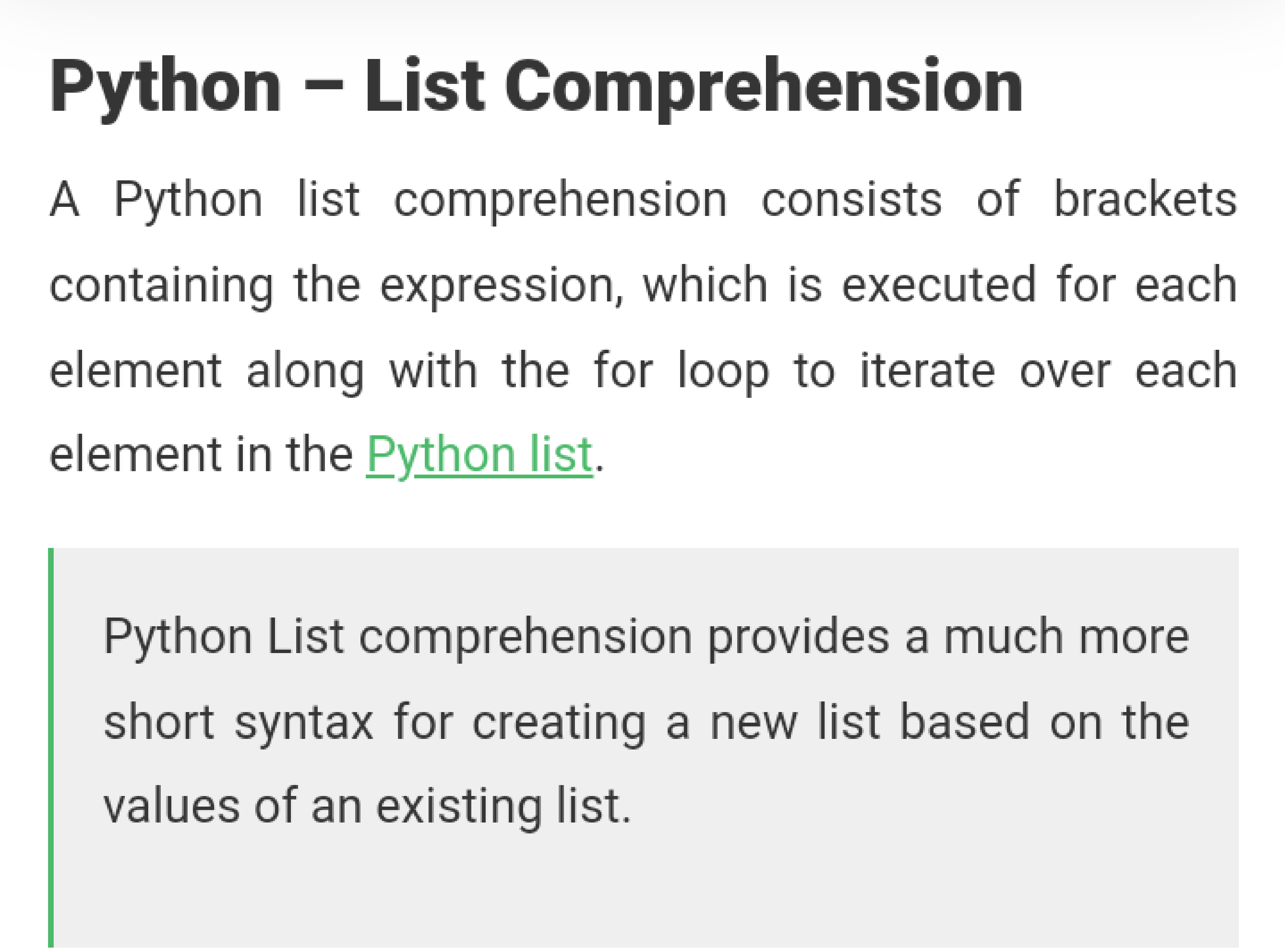
Python

python

List Comprehensions

```
[ x for x in range(10) ]
```



Python – List Comprehension

A Python list comprehension consists of brackets containing the expression, which is executed for each element along with the for loop to iterate over each element in the Python list.

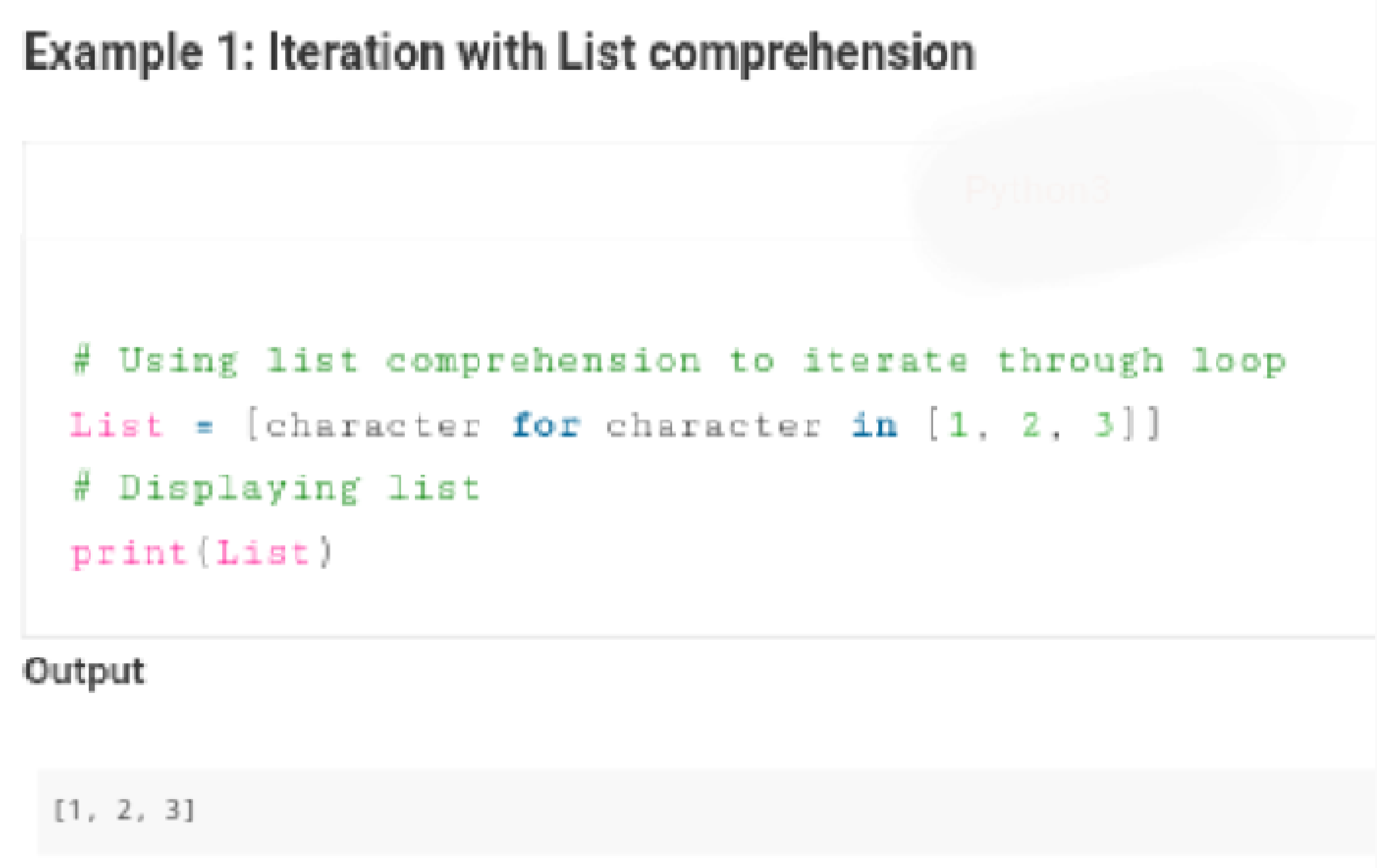
> Python List comprehension provides a much more short syntax for creating a new list based on the values of an existing list.

Advantages of List Comprehension

- More time-efficient and space-efficient than loops.
- Require fewer lines of code.
- Transforms iterative statement into a formula.

Syntax of List Comprehension

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

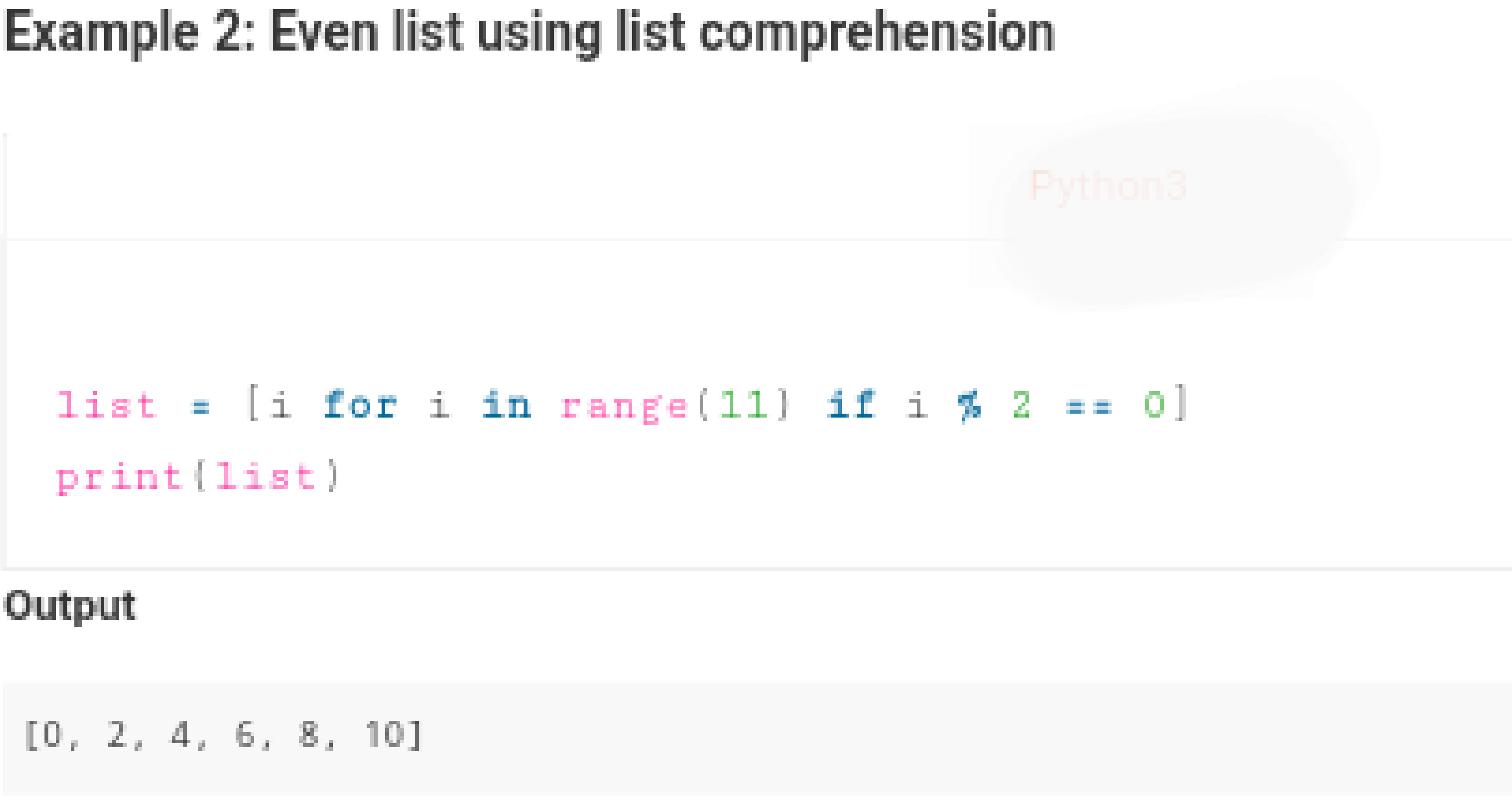


Example 1: Iteration with List comprehension

```python
# Using list comprehension to iterate through loop
List = [character for character in [1, 2, 3]]
# Displaying list
print(List)
```

Output

```
[1, 2, 3]
```



Example 2: Even list using list comprehension

Python3

```
list = [i for i in range(11) if i % 2 == 0]
print(list)
```

Output

```
[0, 2, 4, 6, 8, 10]
```

Example 3: Matrix using List comprehension

Python3

```
matrix = [[j for j in range(3)] for i in range(3)]

print(matrix)
```

Output

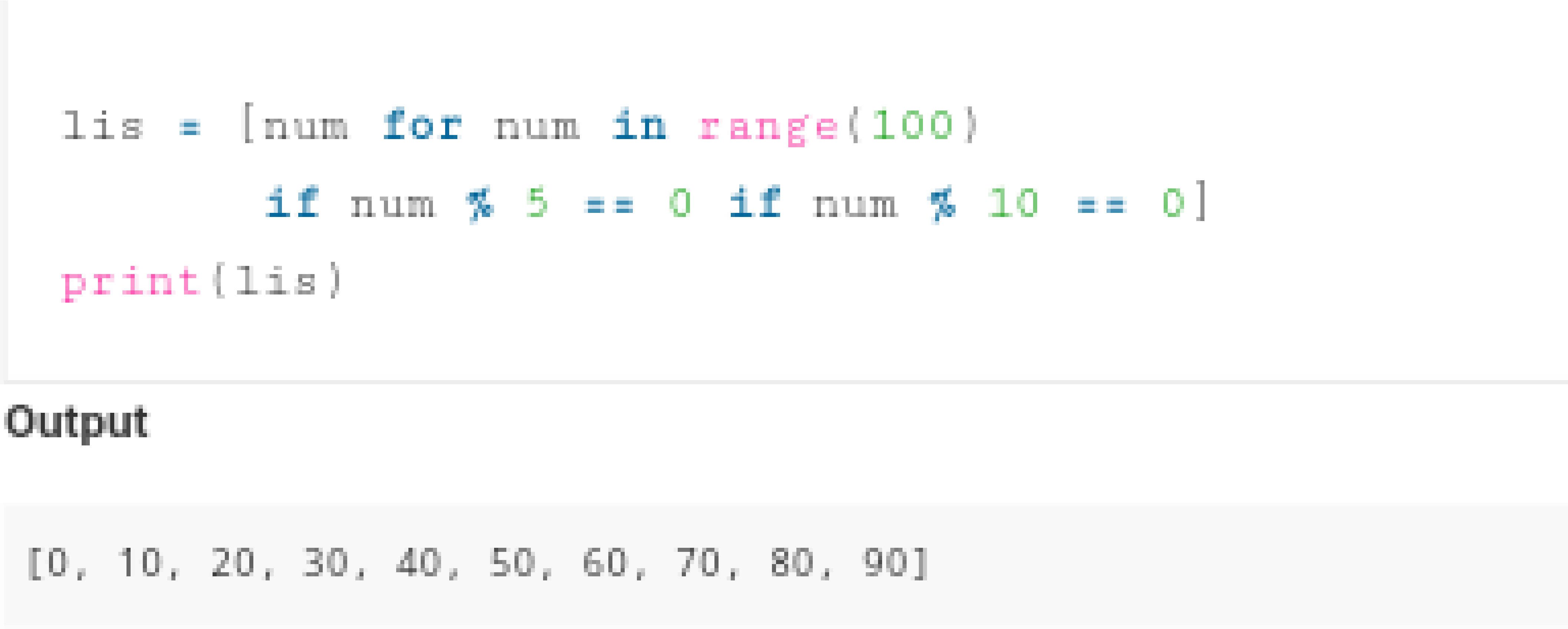
```
[[0, 1, 2], [0, 1, 2], [0, 1, 2]]
```

Conditionals in List Comprehension

We can also add conditional statements to the list comprehension. We can create a list using range(), operators, etc. and cal also apply some conditions to the list using the if statement.

Key Points

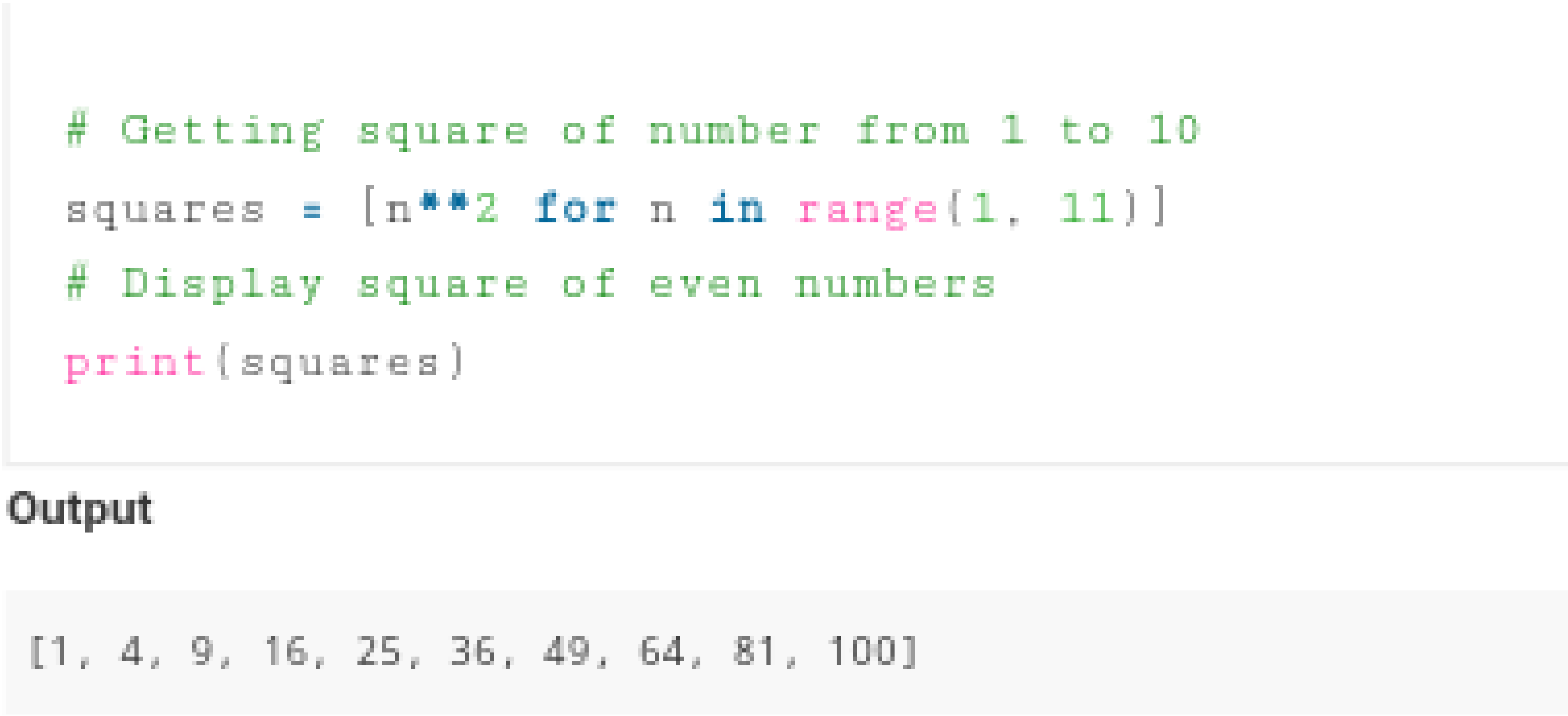
- Comprehension of the list is an effective means of describing and constructing lists based on current lists.
- Generally, list comprehension is more lightweight and simpler than standard list formation functions and loops.
- We should not write long codes for list comprehensions in order to ensure user-friendly code.
- Every comprehension of the list can be rewritten in for loop, but in the context of list interpretation, every for loop can not be rewritten.



```
lis = [num for num in range(100)
       if num % 5 == 0 if num % 10 == 0]
print(lis)
```

Output

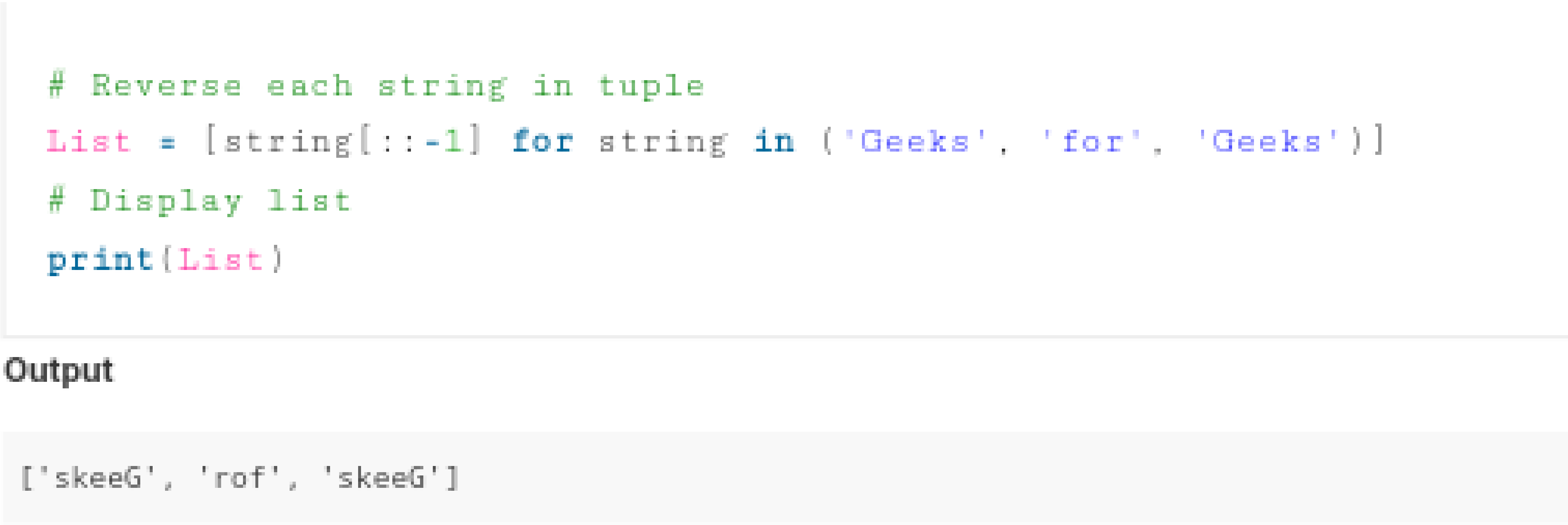
```
[0, 10, 20, 30, 40, 50, 60, 70, 80, 90]
```



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

Output

```
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```



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

Output

```
['skeeG', 'rof', 'skeeG']
```