**Python List Comprehension Using If-Else**

Last Updated : 05 Dec, 2024

[List comprehension](https://www.geeksforgeeks.org/python-list-comprehension/) with if-else in [Python](https://www.geeksforgeeks.org/python-programming-language-tutorial/)is a concise way to apply conditional logic while creating a new list. It allows users to add elements based on specific conditions and even modify them before adding.

**Using if-else Inside List Comprehension**

This is the simplest and most efficient way to apply conditional logic directly. This applies the if-else condition directly in the list comprehension. Each number is checked for divisibility by 2, and either "Even" or "Odd" is added to the list.



1

a = [1, 2, 3, 4, 5]

2

​

3

# Add 'Even' for even numbers, otherwise 'Odd'

4

result = ['Even' if n % 2 == 0 else 'Odd' for n in a]

5

print(result)

**Output**

['Odd', 'Even', 'Odd', 'Even', 'Odd']

## ****Using if Condition Only (Without else)****

This method is used when elements are added only if the condition is met. Here, the if condition filters out numbers that do not meet the condition. Only even numbers are added to the new list.



1

a = [1, 2, 3, 4, 5] ​

3

# Include only even numbers in the list

4

result = [n for n in a if n % 2 == 0] 5

print(result)

**Output:**[2, 4]

## ****Nested if-else in List Comprehension****

Handle multiple conditions with nested logic. Using [Nested if-else](https://www.geeksforgeeks.org/nested-if-statement-in-python/) in List allows chaining of if-else conditions. It categorizes each number based on whether it is divisible by 2, 3, or neither.



1

a = [1, 2, 3, 4, 5]

2

​

3

# Categorize numbers based on divisibility

4

result = ['Divisible by 2' if n % 2 == 0 else 'Divisible by 3' if n % 3 == 0 else 'Other' for n in a]

5

print(result)

**Output**

['Other', 'Divisible by 2', 'Divisible by 3', 'Divisible by 2', 'Other']

## List comprehension in Python

[List comprehension](https://www.geeksforgeeks.org/python-list-comprehension-and-slicing/) is used to create new lists from the existing lists or any other iterable in Python. The elements in one iterable are iterated through, checked for some condition if required and if it satisfies the specified criteria it will be added to the new iterable.

**Syntax:**

new\_list=[variable\_name <for loop> <if-condition>]