LIST COMPREHENSION

Rahul J Reynold

PY MAY 2024 B1 ANSHYA 2PM

**List comprehension**

List comprehension is a concise way to create lists in Python. It allows you to generate a new list by applying an expression to each item in an existing iterable, such as a list, tuple, or string. This is often more readable and concise than using a traditional for loop to populate a list.

Components of List Comprehension

1. Expression: `x\*\*2` is the expression that computes the value to be added to the list.

2. Loop Variable: `x` is the variable that takes the value of each item in the iterable.

3. Iterable: `range(10)` is the iterable providing items for the loop.

The syntax of list comprehension in Python

**List=[expression for item in iterable if condition]**

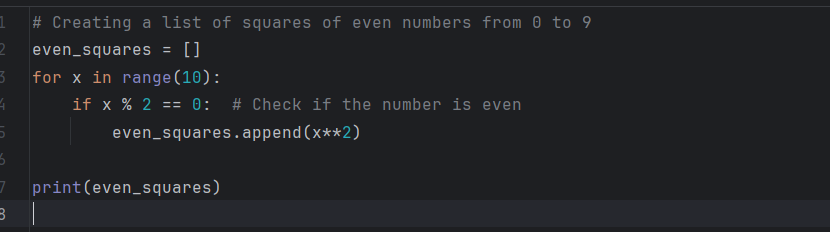
- expression: The operation or value that you want to apply or include in the new list.

- for item in iterable: The loop that iterates over each item in the given iterable (e.g., list, tuple, string, etc.).

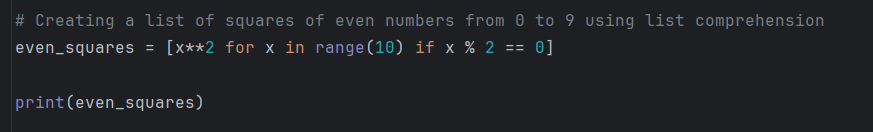
- if condition (optional): A conditional statement that filters items. Only items for which the condition is true will be processed and included in the new list.

**Example:**

**Without List Comprehension**

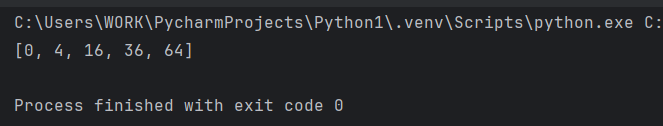


**With List Comprehension**



Output of Both Programs

Both versions of the program will output the same result:



The list comprehension version is more concise and easier to read, especially for simple list transformations like this one.