Looping Statements

- While Loop:
  - A **while** loop statement in Python programming language repeatedly executes a target statement as long as a **given condition is true**.
- The syntax of a **while** loop in Python programming language is –

```
while expression:
        statement(s)
```

While Loop

- <statement(s)> represents the block to be repeatedly executed, often referred to as the body of the loop. This is denoted with indentation.
- The controlling expression, <expr>, typically involves one or more variables that are initialized prior to starting the loop and then modified somewhere in the loop body.

While Loop

- When a while loop is encountered, <expr> is first evaluated in Boolean context.
- If it is true, the loop body is executed. Then <expr> is checked again, and if still true, the body is executed again. This continues until <expr> becomes false, at which point program execution proceeds to the first statement beyond the loop body.

While Loop

```
n = 5
 while n > 0:
        n -= 1
        print(n)
```

While Loop

- n is initially 5. The expression in the while statement header on line 2 is n > 0, which is true, so the loop body executes. Inside the loop body on line 3, n is decremented by 1 to 4, and then printed.
- When the body of the loop has finished, program execution returns to the top of the loop at line 2, and the expression is evaluated again. It is still true, so the body executes again, and 3 is printed.
- This continues until n becomes 0. At that point, when the expression is tested, it is false, and the loop terminates. Execution would resume at the first statement following the loop body, but there isn't one in this case.

For Loop in Python

- The for loop in Python is used to iterate over a sequence (list, tuple, string) or other iterable objects.
- For loop should be used when you need to do something for some predefined number of steps.

For Loop in Python

- Syntax:

```
for iterating_var in sequence:
    statements(s)
```

- Here, iterating_val is the variable that takes the value of the item inside the sequence on each iteration.
- Loop continues until we reach the last item in the sequence.

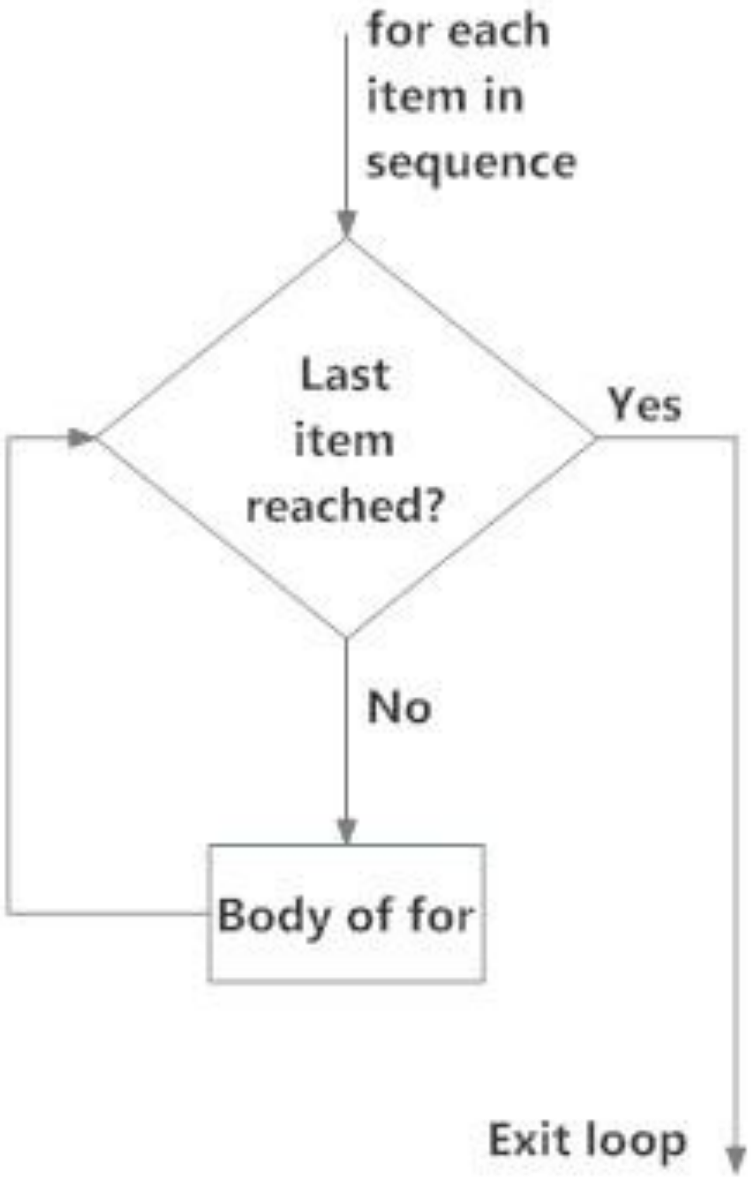

Fig: operation of for loop

Example of for loop

```
languages = ["C", "C++", "Perl", "Python"]
for x in languages:
   print(x)
```