for Loop

- A **for** loop is used for iterating over a sequence (that is either a list, a tuple, a dictionary, a set, or a string).

- **Syntax**

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

- **statement(s)** may be a single statement or a block of statements.

- Each item (starting from the first item) in the list is assigned to `iterating_var`, and the statement(s) block is executed until the entire sequence is exhausted.

- In Python, all the statements indented by the same number of character spaces after a programming construct are considered to be part of a single block of code. Python uses indentation as its method of grouping statements.

**Example**

```
# Program-1 to print 1 to 10

for x in range(1, 6, 1):
    print(x)
```
When the above code is executed, it produces the following result –

1
2
3
4
5

# Program-2 to print 1, 3, 5, 7, 9

for x in range(1, 10, 2):
    print(x)

When the above code is executed, it produces the following result –

1
3
5
7
9

# Program-3 to print sum of 1 to 10

sum = 0
for x in range(1, 11, 1):
    sum = sum + x

print('The sum is:', sum)

When the above code is executed, it produces the following result –

The sum is: 55

# Program-4 to factorial of number

count = 1
fact = 1
for x in range(1, 6, 1):
    fact = fact * x

print('The fact is:', fact)

When the above code is executed, it produces the following result –

The fact is: 120
Using **else** Statement with Loops

If the **else** statement is used with a **for** loop, the **else** statement is executed when the loop is finished.

```python
for x in range(1, 6, 1):
    print (x, " is less than 5")
else:
    print (x, " is not less than 5")
```

When the above code is executed, it produces the following result –

- 0 is less than 5
- 1 is less than 5
- 2 is less than 5
- 3 is less than 5
- 4 is less than 5
- 5 is not less than 5

### Range Function

- It is a in-built function.
- The `range()` function returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and ends at a specified number.

**Syntax**

```python
range ( start, stop, step)
```

**Parameter Values**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>start</td>
<td>Optional. An integer number specifying at which position to start. Default is 0</td>
</tr>
<tr>
<td>stop</td>
<td>Required. An integer number specifying at which position to end.</td>
</tr>
<tr>
<td>step</td>
<td>Optional. An integer number specifying the incrementation. Default is 1</td>
</tr>
</tbody>
</table>

**Example**

- `x = range(3, 6)`      →  3, 4, 5
- `x = range(3, 20, 2)`  →  3, 5, 7, 9, 11, 13, 15, 17, 19
- `x = range(10)`       →  1, 2, 3, 4, 5, 6, 7, 8, 9
- `x = range(10, 1, -1)` →  10, 9, 8, 7, 6, 5, 4, 3, 2
- `x = range(-10, -1, 1)` →  -10, -9, -8, -7, -6, -5, -4, -3, -2
- `x = range(10, 1, -2)` →  10, 8, 6, 4, 2
Loop with the tuple items
fruits = ("apple", "banana", "cherry")
for x in fruits:
    print(x)

Output : apple, banana, cherry

Looping Through a String
for x in "banana":
    print(x)

Output : b, a, n, a, n, a

Looping Through a list
Months = ["Jan","Feb","Mar","April","May","June"]
for m in (Months):
    print (m)

Output : Jan, Feb, Mar, April, May, June

Looping with enumerator
Months = ["Jan","Feb","Mar","April","May","June"]
for i, m in enumerate (Months):
    print (i, m)

Output : 0 Jan, 1 Feb, 2 Mar, 3 April, 4 May, 5 June

The break Statement
With the break statement we can stop the loop before it has looped through all the items:

Example
#Exit the loop when x is "banana" :
fruits = ["apple", "banana", "cherry"]
for x in fruits :
    print(x)
    if x == "banana" :
        break

Output : apple, banana
The continue Statement

With the `continue` statement we can stop the current iteration of the loop, and continue with the next.

Example

```python
#Do not print banana:
fruits = ['apple', 'banana', 'cherry']
for x in fruits:
    if x == 'banana':
        continue
    print(x)
```

**Output**: apple, cherry

Assignment (all the from using for loop)

1. Write the program to display the first 10 terms of the following series:
   a. 1, 3, 5, ...........
   b. 2, 4, 6 ...........
   c. 1, 4, 9, 16.......
   d. 1.5, 3.0, 4.5, 6.0 .......
   e. -5, -10, -15, -20 ........
2. Write a program to calculate and display the sum of all odd numbers and even numbers between a range of numbers from m to n where m < n. Input m and n.
3. Write a program to print the 10 multiples of any entered number.
4. Write a program to display the sum of 10 natural numbers.
5. Write a program to calculate and display the factorial of a entered number.
6. Write a program to count the letter in entered string.
7. Write a program to count the vowels in entered string.