

Python range() Function

Syntax, Examples, Negative Steps and Practice Programs

1. What is the purpose of range()?

The **range()** function in Python is used to generate a sequence of numbers according to the values specified by the programmer.

For example:

```
range(5)
```

represents the sequence:

```
0, 1, 2, 3, 4
```

Notice that **5** is not included.

2. Syntax of range()

Python provides three forms of the range() function:

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

Where:

Argument	Meaning
start	The number from which the sequence starts
stop	The number at which the sequence stops (not included)
step	The difference between consecutive numbers

3. range(stop)

When only one value is given, it is treated as the stop value.

```
range(5)
```

It represents:

```
0, 1, 2, 3, 4
```

Therefore, range(5) generates numbers starting from 0 and ending just before 5.

4. range(start, stop)

Two values can be specified.

```
range(2, 7)
```

Here:

- start = 2
- stop = 7

The sequence is:

```
2, 3, 4, 5, 6
```

Again, the stop value 7 is not included.

5. range(start, stop, step)

Three values can be specified.

```
range(2, 10, 2)
```

Here:

- start = 2
- stop = 10
- step = 2

The sequence is:

```
2, 4, 6, 8
```

The numbers increase by 2 each time.

6. Important Point: stop is always excluded

The stop value in range() is always excluded.

For example:

```
range(1, 6)
```

represents:

```
1, 2, 3, 4, 5
```

not:

```
1, 2, 3, 4, 5, 6
```

7. A Small Experiment with list()

You can use list() to see the numbers represented by a range object.

```
numbers = range(5)  
print(list(numbers))
```

Output: [0, 1, 2, 3, 4]

Here, range(5) creates a range object, and list() converts that sequence into a list so that we can easily see its values.

8. More Examples

```
print(list(range(0, 5)))
```

Output: [0, 1, 2, 3, 4]

```
print(list(range(2, 10)))
```

Output: [2, 3, 4, 5, 6, 7, 8, 9]

```
print(list(range(3, 50, 4)))
```

Output: [3, 7, 11, 15, 19, 23, 27, 31, 35, 39, 43, 47]

9. Practice Programs: Negative step Values

These programs demonstrate descending sequences, negative step values, zero and negative arguments. No for loops are used.

1. Descending sequence

```
print(list(range(5, 0, -1)))
```

Output: [5, 4, 3, 2, 1]

3. Starting from zero

```
print(list(range(0, -5, -1)))
```

Output: [0, -1, -2, -3, -4]

5. Zero step — what happens?

```
print(list(range(1, 6, 0)))
```

Output: ValueError: range() arg 3 must not be zero

2. Negative step of 2

```
print(list(range(10, 1, -2)))
```

Output: [10, 8, 6, 4, 2]

4. Negative start and stop

```
print(list(range(-1, -10, -2)))
```

Output: [-1, -3, -5, -7, -9]

10. Key Points to Remember

- range() generates a sequence of numbers.
- range(stop) starts from 0 by default.
- The stop value is always excluded.
- range(start, stop) generates numbers from start up to, but not including, stop.
- step determines the difference between consecutive numbers.
- A positive step produces an increasing sequence.
- A negative step produces a decreasing sequence.
- The step value cannot be zero; range(..., 0) raises a ValueError.
- list() can be used to display the values represented by a range object.

Note: In this post, we are learning the range() function itself. Its use with for loops can be covered separately.

