Day9_Range

May 27, 2025

1 Day 9 – Python Basics: `range()` Function

Today I learned about the **`range()` function**, which is used to generate a sequence of numbers in Python. It is commonly used with loops, especially `for` loops.

Key Properties of `range()`:

- `range(start, stop, step)` is the format.
- **Start** is optional and defaults to 0.
- **Stop** is required — it defines where the sequence ends (not included).
- **Step** is optional and defines the increment (or decrement for negative step).

Topics Covered:

- Creating ranges with different start, stop, and step values
- Using `range()` in `for` loops
- Converting ranges to lists for viewing
- Using `range()` to loop backward or skip steps

Examples:

```
# Basic range from 0 to 4
for i in range(5):
    print(i)
# Output: 0, 1, 2, 3, 4

# Range from 2 to 9
for i in range(2, 10):
    print(i)
# Output: 2, 3, 4, 5, 6, 7, 8, 9

# Range with step of 2
for i in range(0, 10, 2):
    print(i)
# Output: 0, 2, 4, 6, 8
```

```
# Reverse range
for i in range(10, 0, -2):
    print(i)
# Output: 10, 8, 6, 4, 2
```

The `range()` function is a simple but powerful tool when working with **loops, indexing, and number sequences** in Python.

2 1. range(stop) – Only One Argument

2.1 Starts from 0, goes up to stop - 1.

[1]:
```
for i in range(5):
    print(i)
```

```
0
1
2
3
4
```

3 2. range(start, stop) – Two Arguments

3.1 Starts from start, goes up to stop - 1.

[2]:
```
for i in range(2, 7):
    print(i)
```

```
2
3
4
5
6
```

4 3. range(start, stop, step) – Three Arguments

4.1 Starts from start, increases by step, goes up to stop - 1.

[3]:
```
# Positive step
for i in range(1, 10, 2):
    print(i)
```

```
1
3
5
```

```
7
9
```

[4]:
```python
# Negative step (counting backwards)
for i in range(10, 0, -2):
    print(i)
```

```
10
8
6
4
2
```

4.2 Bonus: Convert range to list

[5]:
```python
print(list(range(5)))
```

```
[0, 1, 2, 3, 4]
```

4.2.1 Doesn't support floats.

[7]:
```python
range(1.0, 5.0)  #  This will cause an error
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[7], line 1
----> 1 range(1.0, 5.0)

TypeError: 'float' object cannot be interpreted as an integer
```

[ ]:
```python

```