

Python range() function

This section introduces the `range()` function and includes examples of using `range()` to specify how many times a `for` loop will be repeated.

- The `range()` function can be used to create a sequence of numbers.
- The structure of the function is **`range(start, stop, step)`**.
 - `start` (optional): starting number of sequence. The default is 0.
 - `stop`: generate sequence up to, but not including the specified number.
 - `step` (optional): number specifies the sequence incrementation. The default is 1.
- The arguments, `start`, `stop`, and `step`, must be integers.

Range Examples

```
range(1, 5)
```

- generates sequence of integers 1, 2, 3, 4. From 1 up to, but not including, 5.

```
range(7)
```

- produces: 0,1,2,3,4,5,6 (default start argument is 0)

```
for i in range(10):  
    print(i)
```

- `range()` with `for` loop
- The number passed to the **`range`** function tells the program how many times to loop.
- The sequence generated is: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- The sequence automatically starts at 0. The stop (10) is never part of the sequence.

```
range(1, 7)
```

- Sequence generated: 1,2,3,4,5,6 (setting the start number to 1)

```
for i in range(5, 10):  
    print(i)
```

- `range()` with `for` loop
- Sequence generated: 5,6,7,8,9 (setting the start number to 5).
- Remember, the end point (10) is never part of the generated sequence.

```
range(0, 7, 2)
```

- Third argument specifies the sequence incrementation (step).
- The sequence generates the sequence in steps of 2. The default is 1.
- Generates sequence: 0, 2, 4, 6

```
print ("counting in fives")
for i in range (0,50,5):
    print(i)
```

- Third argument specifies the sequence incrementation (step).
- The first number is the **start number**
- Second number is the **stop number**
- Sequence generated: 0, 5, 10, 15, 20, 25, 30, 35, 40, 45

```
print("Counting Backwards")
for i in range (10, 0, -1):
    print(i)
```

- Third argument specifies the sequence incrementation (step).
- As well as positive numbers you can also program the computer to loop backwards with a negative number.
- Sequence generated: 10, 9, 8, 7, 6, 5, 4, 3, 2, 1

Self-Check

What happens if you give range only one argument? E.g., range(4)

- A. It will generate a sequence starting at 0, with every number included up to but not including the argument it was passed.
- B. It will generate a sequence starting at 1, with every number up to but not including the argument it was passed.
- C. It will generate a sequence starting at 1, with every number including the argument it was passed.
- D. It will cause an error: range always takes exactly 3 arguments.