SDEV 1001

Programming Fundamentals

More Loops and Exceptions - 1

A LEADING POLYTECHNIC COMMITTED TO YOUR SUCCESS

Expectations - What I expect from you

- No Late Assignments

- No Cheating

- Be a good classmate

- Don’t waste your time

- Show up to class

Agenda

On the right is what we will cover today.

- For Loops with Range in Python
- Looping Over a Range
- Calculating Cubes with a For Loop
- Custom Start and Stop Values
- Looping Backwards
- Creating a List from a Range
- Summary

For Loops with Range in Python

- Sometimes you need to repeat an action a specific number of times, not just for each item in a list.
- The `range()` function is perfect for this!

Why use this?

- It allows you to generate a sequence of numbers, sometimes you won't be iterating over a list but just need to perform an action a set number of times.

Looping Over a Range

Below we will loop through numbers from 0 to 4:

- `range(start, stop)` generates numbers from `start` up to but not including `stop`.

Example:

```
print("Counting from 1 to 5:")
for i in range(1, 6):
    print(f"Number: {i}")
```

Here's what the output looks like:

```
Counting from 1 to 5:
Number: 1
Number: 2
Number: 3
Number: 4
Number: 5
```

Calculating Cubes with a For Loop

Below we have an example that calculates the cube of numbers from 0 to 4:

```
print("Cubes from 0 to 4:")
for n in range(5):
    print(f"{n} cubed is {n ** 3}")
```

Here's what the output looks like:

```
Cubes from 0 to 4:
0 cubed is 0
1 cubed is 1
2 cubed is 8
3 cubed is 27
4 cubed is 64
```

Custom Start and Stop Values

You can can also specify a custom "increment" or "step" value in the `range()` function.

- The third argument in `range(start, stop, step)` sets the increment.

Example:

```
print("Even numbers from 4 to 10:")
for num in range(4, 11, 2):
    print(num)
```

Here's what the output looks like:

```
Even numbers from 4 to 10:
4
6
8
10
```

Looping Backwards

You can also loop backwards using a negative step in the `range()` function.

Example:

```
print("Countdown:")
for n in range(5, 0, -1):
    print(n)
print("Blast off!")
```

Here's what the output looks like:

```
Countdown:
5
4
3
2
1
Blast off!
```

Creating a List from a Range

If you want to create a list of numbers, you can use `list()` with `range()`:

```
numbers = list(range(3, 8))
print("Generated list:", numbers)
```

Here `list(range(3, 8))` creates a list of numbers from 3 to 7 like so: `[3, 4, 5, 6, 7]`.

Summary

- Use `range()` to loop over numbers, not just lists.
- You can specify start, stop, and step values.
- Great for counting, generating sequences, and performing repeated calculations.

Example

Let’s go run a few examples together