Chapter 4 - Loops

Loops

Nested Loops

Learning Objectives - Nested Loops

- **Explain nested loop syntax (especially the whitespace)**
- **Identify the relationship between the variables in each loop with the output produced**

Nested Loops

Nested loops are loops inside of another loop. Nested loops almost exclusively take the form of a for loop inside another for loop. If you have three nested loops, this adds a lot of complexity. You should refactor your code to reduce this complexity.

Syntax

The code below will draw a rectangle of 100 `#` in a 10 x 10 grid. The first loop controls the row of output, while the second loop prints 10 `#` to the screen.

```
for row in range(10):
    for column in range(10):
        print("#", end="")
    print("") # add new line character
```

▼ **What does `end=""` mean?**

By default, the `print` function adds a new line character to whatever it prints to the console. By adding `end=""` to the print function, Python will not go to the next line. Notice, however, the last line of code is a `print` function without `end=""`. This will force the output to the next line.

Code Visualizer

challenge

What happens if you:

- Change the first `range` statement to `range(5)`:?
- Change the second `range` statement to `range(20)`:?
- Remove `, end=""` from the first `print` statement?

Nested Loop Coding Challenge 1

Using nested loops, write some code that outputs the following:

```
##########
##########
##########
##########
##########
```

Code Visualizer

▼ **Hint**

The output is the same character (#). Make sure that your nested loops have the right numbers in the `range` statements to get the appropriate number of rows and columns.

Nested Loop Coding Challenge 2

Using nested loops, write some code that outputs the following:

```
##########
**********
##########
**********
##########
```

Code Visualizer

▼ **Hint**

The output is a # when the outer loop variable is even (0, 2, 4) and a `*` when the outer loop variable is odd (1, 3).

Nested Loop Coding Challenge 3

Using nested loops, write some code that outputs the following:

```
1
22
333
4444
55555
```

Code Visualizer

▼ **Hint**

First, the outer loop does not start with 0. Second, the inner loop runs the same amount of times as the row number. Third, think back to the unit on operators. There is an operator that can repeat a string.