Solution Review: Using Multiple Operators

In the following lesson, we will go over the solution of the challenge: Using Multiple Operators.

We'll cover the following

Task

In this challenge, you were provided with a variable that stored a number and you were asked to check if the number lies between **8** and **75**, inclusive of **8**.

In other words, `check` >= **8** and `check` < **75**.

Solution

Let's look at how we should go about this.

- The first thing was to make sure `check` was less than **75**.

```
check < 75
```

- Next, we had to make sure that `check` was also greater than or equal to **8**.

```
check >= 8
```

- The last step is simply combining both conditions using the `&&` operator.

```
check < 75 && check >= 8
```

You can find the complete solution below:

> You were required to write the code on **line 2**.

This code requires the following environment variables to execute:

LANG

```
val check = 33
val compareCheck = check < 75 && check >= 8

// Driver Code
println(compareCheck)
```

`compareCheck` is comparing if the value stored in `check` is less than **75** using the relational operator `<`. It is also comparing if the value stored in `check` is greater than or equal to **8** using another relational operator `>=`.

Finally, if a number lies between **8** and **75** both comparisons should be `true`. For this, we use the logical operator `&&` which only returns `true` if both its left and right operand are true. Here, `check < 75` is the left operand and `check >= 8` is the right operand.

The objective of this exercise was to learn how to use several different types of operators in the same expression.

Let’s wrap up the discussion with operator precedence in the next lesson.