Chained Comparison Operators

An interesting feature of Python is the ability to *chain* multiple comparisons to perform a more complex test. You can use these chained comparisons as a shorthand for larger Boolean Expressions.

In this lecture we will learn how to chain comparison operators and we will also introduce two other important statements in python: **and** and **or**.

Let's look at a few examples of using chains:

```
In [1]: 1 < 2 < 3
Out[1]: True
```

The above statement check if 1 was less than 2 **and** if 2 was less than 3. We could have written this using an **and** statement in Python:

```
In [2]: 1<2 and 2<3
Out[2]: True
```

The **and** is used to make sure two checks have to be true in order for the total check to be true. Let's see another example:

```
In [3]: 1 < 3 > 2
Out[3]: True
```

The above checks if 3 is larger than both the other numbers, so you could use **and** to rewrite it as:

```
In [4]: 1<3 and 3>2
Out[4]: True
```

Its important to note that Python is checking both instances of the comparisons. We can also use **or** to write comparisons in Python. For example:

```
In [5]: 1==2 or 2<3
Out[5]: True
```

Note how it was true, this is because with the **or** operator, we only need one *or* the other two be true. Let's see one more example to drive this home:

```
In [6]: 1==1 or 100==1
```

```
Out[6]: True
```

Great! For an overview of this quick lesson: You should have a comfortable understanding of using **and** and **or** statements as well as reading chained comparison code.

Go ahead and go to the quiz for this section to check your understanding!