Comparison Operators

In this lecture we will be learning about Comparison Operators in Python. These operators will allow us to compare variables and output a Boolean value (True or False).

If you have any sort of background in Math, these operators should be very straight forward.

First we'll present a table of the comparison operators and then work through some examples:

Table of Comparison Operators

Operator	Description	Example
==	If the values of two operands are equal, then the condition becomes true.	(a == b) is not true.
!=	If values of two operands are not equal, then condition becomes true.	(a != b) is true
<>	If values of two operands are not equal, then condition becomes true.	(a <> b) is true. This is similar to != operator.
>	If the value of left operand is greater than the value of right operand, then condition becomes true.	(a > b) is not true.
<	If the value of left operand is less than the value of right operand, then condition becomes true.	(a < b) is true.
>=	If the value of left operand is greater than or equal to the value of right operand, then condition becomes true.	(a >= b) is not true.
<=	If the value of left operand is less than or equal to the value of right operand, then condition becomes true.	(a <= b) is true.

Let's now work through quick examples of each of these.

Equal

```
In [3]: 2 == 2
```

```
Out[3]: True
```

In [4]:
```
1 == 0
```

Out[4]: False

Not Equal

In [5]:
```
2 != 1
```

Out[5]: True

In [6]:
```
2 != 2
```

Out[6]: False

In [7]:
```
2 <> 1
```

Out[7]: True

In [8]:
```
2 <> 2
```

Out[8]: False

Greater Than

In [9]:
```
2 > 1
```

Out[9]: True

In [10]:
```
2 > 4
```

Out[10]: False

Less Than

In [11]:
```
2 < 4
```

Out[11]: True

In [12]:
```
2 < 1
```

Out[12]: False

Greater Than or Equal to

In [13]:
```
2 >= 2
```

Out[13]: True

In [14]:
```
2 >= 1
```

Out[14]: True

Less than or Equal to

In [15]:
```
2 <= 2
```

Out[15]: True

In [16]:
```
2 <= 4
```

Out[16]: True

Great! Go over each comparison operator to make sure you understand what each one is saying. But hopefully this was straightforward for you

Next we will cover chained comparison operators