For more complex comparisons you can use the following:

Order	Operator	Explanation
Highest. Do this first	()	Do this first, same as with calculations.
	not	True returns False and vice versa.
	and	Both sides need to be True for the whole to be True.
Lowest. Do this last	or	At least one side needs to be True for the whole to be True.

Have a look at these examples:

```
>>> not 'A' == 'B'
True

>>> 'A' == 'A' and 1 == 1
True

>>> 'A' == 'A' and 1 == 2
False

>>> 'A' == 'A' or 1 == 1
True

>>> 'A' == 'A' or 1 == 2
True

>>> 'A' == 'B' or 1 == 2
False
```

And something a bit more complex:

```
>>> 'A' == 'B' and 1 == 1 or 2 == 2
True
>>> ('A' == 'B' and 1 == 1) or 2 == 2
True
>>> 'A' == 'B' and (1 == 1 or 2 == 2)
False
```

The first two examples above are exactly the same. The 'and' operator is handled before the 'or' operator, with or without the brackets. This is because 'and' comes before 'or'. The brackets make it a little easier to read, so you may want to use it anyway.

Get the free Daily Team To-do list here to take your team to the next level!

