43b8ab8f1c

Complete-Python-3-Bootcamp / 01-Python Comparison Operators / **02-Chained Comparison Operators.ipynb**

Go to file ...

Pierian-Data PYTHON 3 UPDATES

Latest commit `6bdd11b` on Feb 13, 2018 **History**

1 contributor

`202 lines (202 sloc) | 4.15 KB`

Raw Blame

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 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]:

```python
1 < 2 < 3
```

Out[1]:

```
True
```

The above statement checks 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]:

```python
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]:

```python
1 < 3 > 2
```

Out[3]:

```
True
```

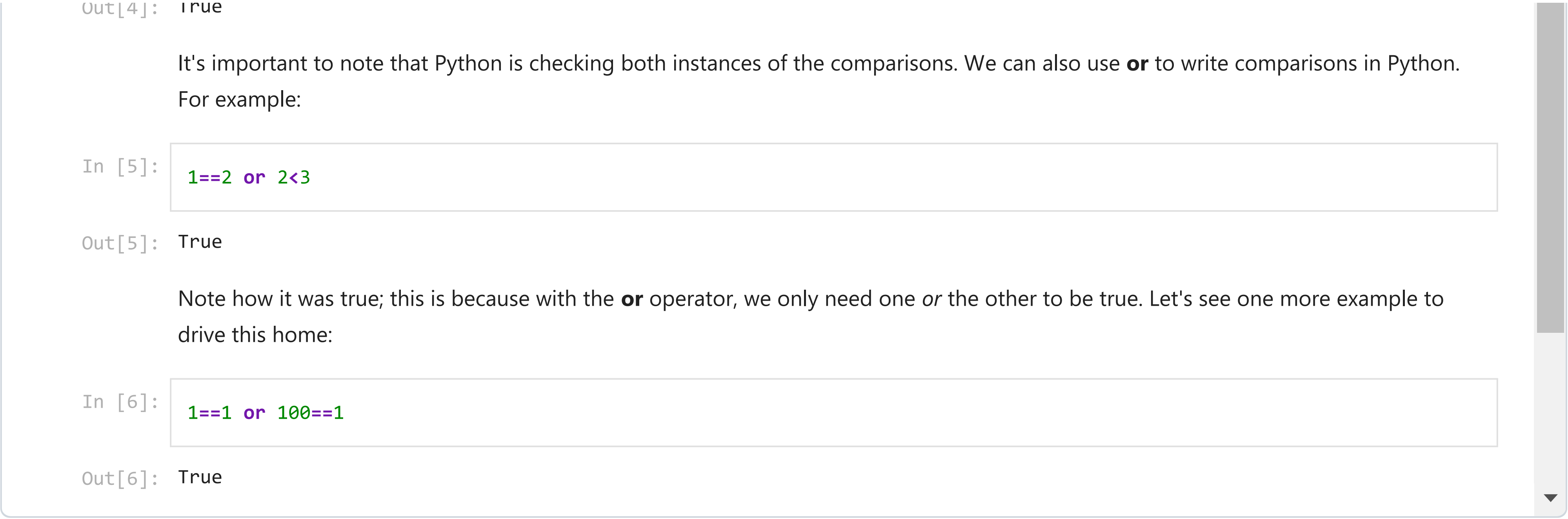
The above checks if 3 is larger than both of the other numbers, so you could use **and** to rewrite it as:

In [4]:

```python
1<3 and 3>2
```

Out[4]:

```
True
```



```
Out[4]: True
```

It's 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 to be true. Let's see one more example to drive this home:

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

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