Exercise 8

Exercise 8

12/12 points (graded)

ESTIMATED TIME TO COMPLETE: 6 minutes
Note that you will have to answer all questions before you can click the Check button.

For each of the following expressions, indicate the value returned, or if the evaluation would lead to an error, write the word 'error' (note this is a word, not a string, no quotes). While you could simply type these expressions into your IDE, we encourage you to answer them directly since this will help reinforce your understanding of basic Python expressions.

Hint: Python boolean types

Remember that in Python words are case-sensitive. The word `True` is a Python keyword (it is the value of the Boolean type) and is not the same as the word `true`. Refer to the Python documentation on Boolean values.

Hint: Priority order of Boolean operations

For these problems, it's important to understand the priority of Boolean operations. The order of operations is as follows:

1. Parentheses. Before operating on anything else, Python must evaluate all parentheticals starting at the innermost level.
2. `not` statements.

3. `and` statements.
4. `or` statements.

What this means is that an expression like

```
not True and False
```

evaluates to `False`, because the `not` is evaluated first (`not True` is `False`), then the `and` is evaluated, yielding `False and False` which is `False`.

However the expression

```
not (True and False)
```

evaluates to `True`, because the expression inside the parentheses must be evaluated first - `True and False` is `False`. Next the `not` can be evaluated, yielding `not False` which is `True`.

Overall, you should always use parenthesis when writing expressions to make it clear what order you wish to have Python evaluate your expression. As we've seen here, `not (True and False)` is different from `(not True) and False` - but it's easy to see how Python will evaluate it when you use parentheses. A statement like `not True and False` can bring confusion!

- `3 > 4`

  False ✔

- `4.0 > 3.999`

  True ✔

- `4 > 4`

  False ✔

- `4 > + 4`

  False ✔

- `2 + 2 == 4`

  True ✔

- `True or False`

  True ✔

- `False or False`

  False ✔

- `not False`

  True ✔

- `3.0 - 1.0 != 5.0 - 3.0`

  False ✔

- `3 > 4 or (2 < 3 and 9 > 10)`

  False ✔

- `4 > 5 or 3 < 4 and 9 > 8`

  True ✔

- `not(4 > 3 and 100 > 6)`

  False ✔