

Lecture 5 - Exercises

5A - Boolean Expressions and Operators

1. Evaluate each of the following boolean expressions:

- `4 > 6`
- `5 <= 5`
- `3 == 6`
- `4 != 5`
- `True != True`

2. Evaluate each of the following, keeping in mind order of operations.

- `10 == 4 + 6`
- `"abc" == "ab" + "c"`
- `'abc' == "abc"`
- `"Scott" == "scott"`
- `(4+3 > 5) == (1.0 > 4)`
- `int(5.6) != int(5.1)`

3. Evaluate the following expressions, writing out each step of simplification after an operator is evaluated to show the order of evaluation as we did in the Bigger Example from the lecture video.

```
(3 == 5 or (3 != 5 and 5 != 7)) and 3 < 5
```

```
11 == 4 ** 2 or 6 != True or not "a" + "b" == "ab"
```

4. For each expression below, give in its **type** and **value**. The first one is done for you.

- `True or False`

Type: bool

Value: True

- `True and False or True`

Type:

Value:

- `2**3.0`

Type:

Value:

- `not 1 + 5 // 2 == 3 and 4 < 5 or 4 != 5`

Type:

Value:

Problems

1. In the lecture video, you saw truth tables for the `and` and `or` operators. Another boolean operator that doesn't have a special keyword in Python is called "exclusive or", or "XOR" for short. This operator evaluates to true if exactly one of its operands is true, and evaluates to false if both operands are true or if both operands are false. Fill in the truth table for the XOR operator:

<code>a xor b</code>	<code>b is True</code>	<code>b is False</code>
----------------------	------------------------	-------------------------

<code>a xor b</code>	<code>b is True</code>	<code>b is False</code>
<code>a is True</code>		
<code>a is False</code>		

2. Suppose your program contains variables `a` and `b` that have boolean values. Which of the following evaluate to `a XOR b`, according to the definition of XOR in the previous question?

1. `not (a and b) and not(not a and not b)`
2. `a or b`
3. `a and b or not (a or b)`
4. `(a or b) and not (a and b)`
5. `a or b and not a and b`

3. Way back in Lecture 1, recall that we started writing a math quiz program:

Write a **math quiz program** that works as follows: it begins by printing an arithmetic problem of your choosing. Then, it prompts the user to enter an answer. Finally, once they have pressed enter, the program prints a message showing the correct answer.

With boolean comparison operators, we're one step closer to being able to check the user's answer. Write the math quiz program so that it prints `True` if the user's answer is correct and `False` if the user's answer is not correct. A couple sample runs of the program are given below:

```
What is 4 * 6? 24
Your answer is True!
```

```
What is 4 * 6? 2
Your answer is False!
```