Python Logical and Relational Operators

With Python logical and relational operators, you can evaluate several conditional expressions to determine True or False.

AND, OR, and NOT  
The output of the AND is 1 when all inputs are 1s and 0 otherwise  
The output of the OR is 0 when all inputs are 0s and 1 otherwise  
The output of the NOT is 0 when the input is a 1   
The output of the NOT is 1 when the input is a 0

With Python logical and relational operators, you can have situations, such as these:  
Relational Logical Relational  
Relational Logical Relational Logical Relational  
and many other combinations.

**Two Relational and One Logical**  
Since relational operators are at a higher precedence that logical operators, operator precedence must be considered when you evaluate expressions that include both relational and logical operators

In examples 1 and 2 consider these subexpressions, x = 3 and y = 12  
  
**Example1:**  
The expression,   
x == 3 or y > 14  
should not be evaluated left to right, because doing so will give an incorrect result. From the Python operator precedence, relational are higher than logical, which means you must evaluate the subexpressions first followed by the logical operation; you can evaluate either first:  
x == 3 is True  
y > 14 is False

Combined with logical or:  
True or False is True

**Example2:**  
However, if you had x == 3 and y > 14  
x == 3 is True  
y > 14 is False

Combined with logical and:  
True and False is False

**Three Relational and Two Logical**  
In examples 3, 4 and 5 consider these subexpressions, x = 3, y = 12, and z = 4

**Example3:**  
To determine the result in these expressions,   
x == 3 or y > 14 or z != y  
Python, evaluates the subexpressions first followed by the logical operation:  
x == 3 is True  
y > 14 is False  
z != y is True

Since the expression contain all logical or, the logical operation maybe done in any order:  
True or False or True is True

**Example4:**  
This expression,   
x == 3 and y > 14 and z != y  
evaluates like this,  
x == 3 is True  
y > 14 is False  
z != y is True

True and False and True is False

**Example5:**  
Since the expression,  
x == 3 or y > 14 and z != y  
contains logical or and logical and, the logical operation cannot be done in any order, after the relational  
x == 3 is True  
y > 14 is False  
z != y is True

Therefore, after consulting the operator precedence, you will see that AND has a higher precedence than OR:  
False AND True is False, which is compared with the True from x ==3,  
True OR False is True

**Note:**When typing your Python code, the operators must be lowercase, and, or, not

**Assignment:**  
Evaluate each of the following exercises; then execute each from the primary prompt to verify your answers

**Values for 1, 2, 3, 4, 5:**  
x = 3  
y = 11

1.  
Evaluate, execute, and display,   
a = x < 5 **and** y < 10

2.Evaluate, execute, and display,   
b = x < 5 **or** y < 4

4.Evaluate, execute, and display,   
c = **not** (x < 5 **and** y < 10)

5.You can also **group sub-expressions**

Evaluate, execute, and display,   
d = (x < 5 and y < 10) **and** (x < 5 or y < 4)

6.Evaluate, execute, and display,   
e = (x != 5 or y < 10) **and** (x > 5 or y = 11)