Conditions and Logic

Computer programs make decisions based on logic: if some condition applies, do something, otherwise, do something else.

Unit 1 Boolean Operations

Expressions may include Boolean operators to implement logic. If all three operators appear in the same expression, Python will evaluate `not` first, then `and`, and finally `or`. If there are multiple of the same operator, they are evaluated from left to right.

1. Assuming `P` and `Q` each represent a Boolean expression that evaluates to the Boolean value indicated, complete the following table.

`P`	`Q`	`P and Q`	`P or Q`
False	False		
False	True		
True	False		
True	True		

2. Assume that two Boolean expressions are combined using the `and` operator. If the value of the first expression is `False`, is it necessary to determine the value of the second expression? Explain why or why not.

3. Assume that two Boolean expressions are combined using the `or` operator. If the value of the first expression is `True`, is it necessary to determine the value of the second expression? Explain why or why not.

4. Suppose you wanted to print a result only when both `x` and `y` are positive. Determine the appropriate operators, and write a single Boolean expression for the `if`-statement condition.

5. Rewrite the expression from #4 using the `not` operator. Your answer should yield the same result as in #4, not the opposite. Describe in words what the new expression means.

6. Suppose that your team needs to print a result, except for when both `x` and `y` are positive. Write a Boolean expression for this condition. How is this different from the previous question?

Unit 2 Truthy or Falsy

In Python, most values are considered to be "true" when used as a Boolean. However, some values are considered to be "false." **Read questions 7 and 8 before completing the table.**

Python value	Predicted output	Actual output
`123`		
`0`		
`"abc"`		
`4.56`		
`""`		
`"0"`		
`True`		
`"True"`		
`False`		
`"False"`		
`0.0`		
`"0.0"`		

Questions

7. Run the following code snippet, and use the output to complete the first row of the table.

```
value = 123
if value:
    print("Truthy")
else:
    print("Falsy")
```

8. Predict the output (Truthy or Falsy) for the other values in the table. Then test each value using the code snippet from the previous question. Record your results in the table.

9. Using complete sentences, summarize what kinds of values are considered to be `True` in Python (and what kinds of values are considered to be `False`).

10. Predict the value (`True` or `False`) of the following expressions. Then check your answers using a Python Shell.

Python value	**Predicted output**	**Actual output**
`bool(5)`		
`bool("")`		
`not 5`		
`not ""`		

11. Rewrite the following `if` statements without using == or `!=` (or any other operator).

a)
```
if value != 0:
    print("Not zero")
```

b)
```
if happy == True:
    print("You know it!")
```

c)
```
if absent == False:
    print("Good job")
```

d)
```
if answer != "":
    print("Thank you")
```

e)
```
if answer == "":
    print("Try again")
```