Lab02_Conditions_And_Logic

September 8, 2025

Conditions and Logic

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

0.1 Content Learning Objectives

After completing this activity, students should be able to: * Evaluate boolean expressions with comparison operators (<, >, <=, >=, ==, !=) * Explain the syntax and meaning of if/else statements and indented blocks * Evaluate boolean expressions that involve comparisons with and, or, and not.

0.2 Process Skill Goals

During the activity, students should make progress toward: * Evaluating complex logic expressions based on operator precedence. (Critical Thinking)

0.3 Atmospheric Sciences Concepts

Decisions can be used for a variety of things in atmospheric sciences. The most common is using conditions and logic to categorize weather phenomena. These can include: * Hurricanes * Tornadoes * Watch vs Warning * Sky Cover * Scales * Much, much more!!!

1 Part 1 Comparison Operators

In Python, a comparison (e.g., $100 < 200$) will yield a ***Boolean*** value of either **True** or **False**. Most data types (including *int, float, str, list, and tuple*) can be compared using the following operators:

Operator	Meaning
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
==	equal
!=	not equal

Type the following code, one line at a time, into a Python Shell. Record the output for each line (if any) in the second column.

Python Code	Shell Output
type(True)	
type(true)	
type(3 < 4)	
print(3 < 4)	
three = 3	
four = 4	
five = 5	
print(three == four)	
print(three < five and four < five)	
check = three > four	
print(check)	
type(check)	
print(three = four)	
three = four	
print(three == four)	

1.1 Questions

1. What is the name of the data type for Boolean values?

2. Do the words **_True_** and **_False_** need to be capatilized? Explain how you know.

3. What Boolean Operators did you use in the table above?

4. Explain why the same expression **_three == four_** had two different results.

5. What is the difference between the $=$ operator and the $==$ operator?

6. Write a Boolean expression that uses the *!=* operator and evaluates to ***False***.

2 Part 2 if/else Statements

An if statement makes it possible to control what code will be executed in a program, based on a condition. If statements use Boolean Expressions, a statement that evaluates to either True or False, to make a decision. Here are a few examples:

2.1 Example 1

```
[ ]: number = int(input("Enter an interger: "))
     if number < 0:
         print(number, "is negative. This is unacceptable.")
     else:
         print(number, "is a fine number.")
     print("Until next time ...")
```

Python uses *indentation* to define the structure of programs. The line indented under the *if* statement is executed only when **number < 0** is ***True***. Likewise, the line indented under the *else* statement is executed only when **number < 0** is ***False***.

2.2 Example 2

The Saffir-Simpson Scale is used for hurricane classification. It uses wind speeds to estimate potential property damage and storm surge. Here is a larger *if/else* block using the Saffir-Simpson Scale.

```
[ ]: wind_speed = int(input("Enter a wind speed: "))
     if wind_speed < 20:
         print("A wind speed of", wind_speed, "kts is normal")
     elif wind_speed > 20 and wind_speed < 34:
         print("A wind speed of", wind_speed, "kts is a Tropical Depression")
     elif wind_speed > 35 and wind_speed < 63:
         print("A wind speed of", wind_speed, "kts is a Tropical Storm")
     elif wind_speed > 64 and wind_speed < 82:
         print("A wind speed of", wind_speed, "kts is a Category 1 Hurricane")
     elif wind_speed > 83 and wind_speed < 95:
         print("A wind speed of", wind_speed, "kts is a Category 2 Hurricane")
     elif wind_speed > 96 and wind_speed < 113:
         print("A wind speed of", wind_speed, "kts is a Category 3 Hurricane")
     elif wind_speed > 114 and wind_speed < 135:
         print("A wind speed of", wind_speed, "kts is a Category 4 Hurricane")
     else:
         print("A wind speed of", wind_speed, "kts is a Category 5 Hurricane")
```

2.3 Questions

7. What is the Boolean expression in the first *if/else* example?

8. Copy and paste the first example into a Python Shell then run it. What is the output when the user enters the number 5? What is the output when the user enters -5? Why was there a different output?

9. Copy and paste the second example into a Python Shell then run it. What is the output when the user enters the number 71? What is the output when the user enters 167?

10. After an if-condition, what syntax differentiates between (1) statements that are executed if the condition is True and (2) statements that are always executed?

11. Copy and paste the code below into a Python Shell. What Error did you get? What does this tell you about when the indentation in a Python Program is inconsistent?

```
[ ]: pineapple_on_pizza = True
     if pineapple_on_pizza == True:
         print("YUCK!")
       else:
                 print('phew! no pineapple')
```

12. Based on the examples above, what must each line preceding an indented block of code end with?

13. In a Python Shell, write an *if* statement that determines whether a number is even or odd, and then prints the message "*number* is even" or "*number* is odd". Copy and paste the code below once you're done. (Hint: use the % operator)

5

3 Part 3 Boolean Operators

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

Read the questions before filling in the table!!

Python Code	Predicted Output	Actual Output
print(a < b and b < c)		
print(a < b or b < c)		
print(a < b and b > c)		
print(a < b or b > c)		
print(not a < b)		
print(a > b or not a > c and b > c)		

3.1 Questions

15. What is the resulting *data type* when you run *a < b*? What data type is the result of *a < b and b < c*?

16. Set the variables *a = 3, b = 4, c = 5*. Predicted the output of each print statement in the table. Once you've predicted all outputs, run each line in a Python Shell and add the actual outputs.

17. Based on the variables in *#16*, what is the value of *a < b*? What is the value of *b < c*?

18. If two ***True*** Boolean expressions are compared using the ***and*** operator, what is the resulting Boolean value?

19. Using the variables defined in *#16*, write an expression in Python Shell that will compare two **False** Boolean expressions using the *or* operator. Copy and Paste your code below once finished.

20. 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		

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

22. Assume that two Boolean expressions are compared 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.

23. Suppose you want to execute a statement $sum = x + y$ only when both x and y are positive. Determine the appropriate operators, and write a single Boolean expression in a Python Shell for the if-condition. Copy and paste your code below once finished.

24. Precipitation can come in a variety of forms (rain, freezing rain, sleet, snow, etc). The precipitation type is dependent on the temperature at different levels of the atmosphere(cloud temp, temp aloft, and surface temp). Write an *if/else* block that determines the precipitation type based on the table below. Temperatures are in degrees Celsius. We'll work on this one as a class but begin thinking of how you would do this and start testing solutions in a Python Shell.

Precipitation Type	Cloud Temp	Temp Aloft	Surface Temp
Rain	0 to 50	1 to 50	1 to 50
Freezing Rain	-10 to -5	5 to 15	-10 to 0
Sleet	-15 to -10	-5 to 5	-10 to -5
Snow	-25 to -15	-20 to -10	-15 to -5

```
[ ]: cloud_temp = int(input("Enter cloud temp: "))
     temp_aloft = int(input("Enter temp aloft: "))
     surface_temp = int(input("Enter surface temp: "))
```