**Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Partners: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Python Activity 5: Boolean Expressions**

|  |
| --- |
| **Learning Objectives**  Students will be able to:  *Content:*   * Explain the three types of programming structures * Explain how conditional operators and logical operators are used in programming * Use conditional operators with strings and numeric values   *Process:*   * Write correct Boolean expressions and compound expressions   **Prior Knowledge**   * Python concepts from Activities 1-4 |

**Critical Thinking Questions**

|  |  |  |
| --- | --- | --- |
| **Programming Structures** | | |
| **consecutive statementsSequence Structure** | **Decision or Branching Structure**  **tf** | **Looping Structure**  **loop** |

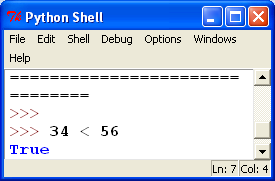
1. Which structure best describes the types of Python programs you have written so far?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. Which structure allows the programmer to create code that decides what code is executed?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| **FYI: Conditional operators**, also known as relational operators, are used to compare the relationship between two operands. Expressions whose result can only be **true or false** are known as **Boolean expressions.** |

3. State the meaning of each of the following **conditional operators.** If you are not sure of the meaning of any symbol, create some example expressions, type them into the Python *interpreter* (See Figure to the right) and examine the results.

a. < \_\_\_\_\_\_True\_\_\_\_ b. > \_\_False\_\_\_\_\_\_\_\_\_\_\_\_\_\_

c. <= \_\_\_\_\_True \_\_\_\_ d. >= \_\_False\_\_\_\_\_\_\_\_\_\_\_\_\_\_

e. != \_\_\_\_\_\_\_True\_\_\_\_ f. == \_\_False\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4. What is the result of each of the following expressions?

Assume: x = 4, y = 5, and z = 4

a. x > y \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_False\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

b. x < y \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_True\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

c. x == y \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_False\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

d. x != y \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_True\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

e. x >= z \_\_\_\_\_\_\_\_\_\_\_\_\_\_True\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

f. x <= z \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_False\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

g. x + y > 2 \* x \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_True\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

h. y \* x – z != 4 % 4 + 16 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_False\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

i. pow(x,2) == abs(-16) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_True\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5. What is the result of the following expressions?

Assume: word1 = “hello” and word2 = “good-bye”

a. word1 = = word2 \_\_\_\_\_\_\_\_\_\_\_\_False\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

b. word1 != word2 \_\_\_\_\_\_\_\_\_\_\_\_True\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

c. word1 < word2 \_\_\_\_\_\_\_\_\_\_\_\_\_False\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

d. word1 >= word2 \_\_\_\_\_\_\_\_\_\_\_\_\_True\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

6. How do the conditional operators work when the operands are strings? \_\_\_\_\_\_\_\_\_Checking the contents of the variable\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

7. What are the two possible answers for each expression in questions 4 and 5? \_\_\_\_\_True or False \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**FYI:** We can use **logical operators** to determine logic between conditions (relational expressions).

8. Sometimes you want to test more than one condition to determine which code segment should be executed. You can use the following **logical operators** to create **compound conditions**. Examine each operator and a sample of its use. Provide an explanation of how each operator works.

|  |  |  |
| --- | --- | --- |
| **Operator** | **Example** | **Explanation** |
| **and** | (age >= 17) **and** (hasLicense = = true) | If both of these are True, then the value is true |
| **or** | (cost < 20.00) **or** (shipping = = 0.00) | If either of these are True, then the value is true |
| **not** | **not** (credits> 120) | If the value does not exceed this condition, it stays True; otherwise it becomes False. |

9. Assume the value of the variable **numBooks** is 40. State the values of each of the Boolean expression.

|  |  |
| --- | --- |
| **Expression** | **Value** |
| (numBooks > 5) and (numBooks < 100) | True |
| (numBooks < 5) or (numBooks > 100) | False |
| not(numBooks \* 10 == 100) | True |

**Application Questions: Use the Python Interpreter to check your work**

1. Assign a value to **num1** and **num2.** Write a Boolean expression that tests if the value stored in the variable **num1** is equal to the value stored in the variable **num2**.

num1= 3

num2= 4

print(num1 == num2)

2. Assign a value to the variables listed in this problem (time, maxTime, cost, and maxCost). Write a Boolean expression that tests if the value stored in the variable **time** is less than the value stored in the variable **maxTime** or if the value stored in the variable **cost** is less than the value stored in the variable **maxCost**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

3. Assign a value to **weight** and **cost.** Write a Boolean expression that tests if the value stored in weight is < 10 and the value store in cost is not greater than 20.00

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_