**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** |

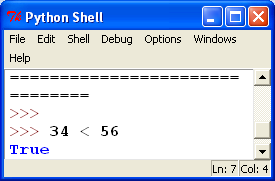
1. Which structure best describes the types of Python programs you have written so far?

Sequence structures

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

Decision structure.

|  |
| --- |
| **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. < Less Than b. > Greater than

c. <= Less than or equal to d. >= Greater than or equal to

e. != not equal too f. == equal too

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 True

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? The same way as ints do. They know if they are the same or not. Word1 is greater because h comes before g

7. What are the two possible answers for each expression in questions 4 and 5? True and 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 conditions are met you move forward |
| **or** | (cost < 20.00) **or** (shipping = = 0.00) | If one or the other is met |
| **not** | **not** (credits> 120) | Needs to meet the criteria |

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==num2

False

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**

**time<maxTime or cost<maxCost**

**True**

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

weight<10 and cost<20

False