1.What are the two values of the Boolean data type? How do you write them?

Boolean is a built-in datatype in python. It has two values: either true or false. The result maybe Boolean in the case of certain expressions or conditions. The bool() function also allows you to evaluate any value and is normally evaluated to True if there’s any content. Null values are evaluated as False.

For eg: print(7>10) returns false

2. What are the three different types of Boolean operators?

And , or , not are the three different Boolean operators. ‘And’ returns True if both are True. ‘Or’ returns True if atleast one is True and ‘Not’ returns True when the statement is false.

3. Make a list of each Boolean operator's truth tables (i.e. every possible combination of Boolean values for the operator and what it evaluate ).

4. What are the values of the following expressions?

(5 > 4) and (3 == 5) False

not (5 > 4) False

(5 > 4) or (3 == 5) True

not ((5 > 4) or (3 == 5)) False

(True and True) and (True == False) False

(not False) or (not True) True

5. What are the six comparison operators?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Operators** | **Title** | **Description** | **Example** | **Output** |
| == | Equal to | It results in True if both operands equal, False if indeed. | Print (9==9) | True |
| != | Not equal to | It results in True if both operands are unequal, and False if equal. | Print (3! =5) | True |
| > | Greater than | It results in True if first operand is bigger than second operand. | Print (5>3) | True |
| < | Less than | Results in True if first operand is smaller than the second. | Print (2<5) | True |
| >= | Greater than or equal to | Results in True if first operand is bigger or same as second operand | Print (9>=4)  Print (4>=4) | True |
| <= | Less than or equal to | Results in True if first operand is smaller or same as second operand. | Print (3<=7)  Print (5<=5) | True |

6. How do you tell the difference between the equal to and assignment operators? Describe a condition and when you would use one.

The equal assignment operator assigns values or expressions to the variable on the left side. It is represented by the ‘=’ symbol. For eg: x = 8, here x is assigned the value of 8.

A comparison or relational operator is ‘==’. It is used to compare values and the results are Boolean. For eg: 8 == 7, the output wll be False here.

7. Identify the three blocks in this code:

spam = 0

if spam == 10:

print('eggs')

if spam > 5:

print('bacon')

else:

print('ham')

print('spam')

print('spam')

The output will be:

ham

spam

spam

First, the value of 0 is assigned to spam. Given the condition that the value of spam is equal to 10, eggs is to be printed. This is False in this case so moving on to the second condition, if spam is greater than 5, bacon is to be printed. This is also False and the final condition is to print ham, spam, spam one after the other if the first two conditions are unsatisfied.

8. Write code that prints Hello if 1 is stored in spam, prints Howdy if 2 is stored in spam, and prints Greetings! if anything else is stored in spam.

spam = 2

if spam == 1:

print(‘Hello’)

if spam == 2:

print(‘Howdy’)

else:

print(‘Greetings!’)

In this case, output will be Howdy as spam is assigned 2 in the first line of code.

9.If your programme is stuck in an endless loop, what keys you’ll press?

When a conditional loop repeats without stop, it’s called an infinite loop. This continuous repeating loop can result in insufficient CPU memory and is caused due to some error in code. This occurs when a condition for termination of loop is not specified. In such cases, Ctrl + C can terminate the loop and prevent CPU exhaustion.

10. How can you tell the difference between break and continue?

Break statement is used with for loop to terminate the loop when certain conditions are met.

Eg: For i in range(6):

If i ==3:

Break

print(i)

This will stop the loop when i is equal to 3.

The continue statement skips the specific iteration mentioned when that specific condition is met. It doesn’t end the loop completely like the break statement.

Eg: for i in range(5):

If i==3:

Continue

Print(i)

In the above example, when iteration is equal to 3, it is skipped and the loop continues from 4.

11. In a for loop, what is the difference between range(10), range(0, 10), and range(0, 10, 1)?

Range(10) = indices 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 (first 10 indices)

Range(0,10) = indices 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 (start from 0, end at 10 – not included)

Range(0,10,1) = indices 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 (start from 0, end at 10 – not included and step of 1)

12. Write a short program that prints the numbers 1 to 10 using a for loop. Then write an equivalent program that prints the numbers 1 to 10 using a while loop.

a)

for i in range(11):

if i == 0:

continue

print(i)

print()

b)

count = 0

while count <=9:

count+=1

print(count)

13. If you had a function named bacon() inside a module named spam, how would you call it after importing spam?

spam.bacon()