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

The two values of the Boolean data type in Python are:

True

False

Writing Boolean values:

True: Represents a logical true value.

False: Represents a logical false value.

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

and: Returns True if both conditions are true.

or: Returns True if at least one condition is true.

not: Reverses the Boolean value of the condition.

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 ).

1. and Operator (Logical AND):

The and operator returns True only if both operands are True.

A B A and B

True True True

True False False

False True False

False False False

2. or Operator (Logical OR):

The or operator returns True if at least one operand is True.

A B A or B

True True True

True False True

False True True

False False False

3. not Operator (Logical NOT):

The not operator inverts the Boolean value of the operand.

A not A

True False

False True

4. What are the values of the following expressions?

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

not (5 > 4)

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

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

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

(not False) or (not True)

5. What are the six comparison operators?

Here are the six comparison operators:

== (Equal to):

Returns True if the values on both sides are equal.

Example: x == y returns True if x is equal to y.

!= (Not equal to):

Returns True if the values on both sides are not equal.

Example: x != y returns True if x is not equal to y.

> (Greater than):

Returns True if the value on the left is greater than the value on the right.

Example: x > y returns True if x is greater than y.

< (Less than):

Returns True if the value on the left is less than the value on the right.

Example: x < y returns True if x is less than y.

>= (Greater than or equal to):

Returns True if the value on the left is greater than or equal to the value on the right.

Example: x >= y returns True if x is greater than or equal to y.

<= (Less than or equal to):

Returns True if the value on the left is less than or equal to the value on the right.

Example: x <= y returns True if x is less than or equal to y.

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

1. Equal to (==) Operator:

The == operator is used to compare two values or expressions to check if they are equal.

It returns True if the values are equal, and False if they are not.

2. Assignment (=) Operator:

The = operator is used to assign a value to a variable.

It does not compare values; instead, it assigns the value on the right-hand side to the variable on the left-hand side.

Difference in Context:

When to Use == (Equal to):Use the == operator when you need to compare two values and check if they are equal.

When to Use = (Assignment):Use the = operator when you need to assign a value to a variable.

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')

spam = 0

if spam == 10:

print('eggs')

if spam > 5:

print('bacon')

else:

print('ham')

print('spam')

print('spam')

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 = int(input("Enter a number: "))

if spam == 1:

print("Hello")

elif spam == 2:

print("Howdy")

else:

print("Greetings!")

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

If your program is stuck in an endless loop and you're running it in a terminal or an integrated development environment (IDE), you can use the following key combinations to stop the program:

1. In most terminal environments (Linux, macOS, or Windows):Ctrl + C: This is the most common way to interrupt and stop a running program. It sends a signal to terminate the process and stops the execution immediately.

2. In Jupyter Notebooks:Click the "Stop" button (a square) in the toolbar, or press Ctrl + C in the terminal (if you're running the notebook server from the terminal).

3. In some IDEs (like PyCharm or VS Code):Click the Stop button (red square) in the terminal or the "Run" window to stop the execution of the program.

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

break: Exits the loop entirely. The loop stops, and execution continues with the code after the loop.

continue: Skips the current iteration and proceeds to the next iteration of the loop without exiting the loop.

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

range(10): Starts from 0, ends at 10 (exclusive), and steps by 1. It's a shorter form of range(0, 10).

range(0, 10): Explicitly starts from 0, ends at 10 (exclusive), and steps by 1. This is more explicit but functionally equivalent to range(10).

range(0, 10, 1): Starts from 0, ends at 10 (exclusive), and steps by 1. It explicitly includes the step, but it behaves the same as the previous two in this case.

In all three cases, the sequence of numbers generated is the same: from 0 to 9. The difference lies in the syntax and how explicit the parameters are.

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.

for i in range(1, 11):

print(i)

i = 1

while i <= 10:

print(i)

i += 1

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

import spam

spam.bacon()

from spam import bacon

bacon()