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

The two values of the Boolean data type are true and false. They can be written in a variety of ways, including:

* True: true, 1, on, yes
* False: false, 0, off, no

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

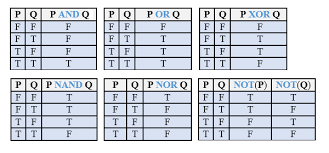
There are three types of Boolean operators: AND, OR, and NOT.

AND: The AND operator returns true if both operands are true. For example, the expression (1 > 0) AND (2 > 1) is true because both operands are true.

OR: The OR operator returns true if either operand is true. For example, the expression (1 > 0) OR (2 < 1) is true because one of the operands is true.

NOT: The NOT operator returns the opposite of the operand. For example, the expression NOT (1 > 0) is false because the operand is true.

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 evaluates).



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 > 4) and (3 == 5) is False because 5 is greater than 4, but 3 is not equal to 5.
* not (5 > 4) is True because the opposite of 5 being greater than 4 is 5 being less than or equal to 4.
* (5 > 4) or (3 == 5) is True because one of the expressions is True (5 is greater than 4).
* not ((5 > 4) or (3 == 5)) is False because the opposite of one of the expressions being True is False.
* (True and True) and (True == False) is False because True and True is True, but True is not equal to False.
* (not False) or (not True) is True because the opposite of False is True, and True or True is True.

5. What are the six comparison operators?

* Less than (<)
* Less than or equal to (<=)
* Greater than (>)
* Greater than or equal to (>=)
* Equal to (==)
* Not equal to (!=)

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 to (==) and assignment (=) operators are two different operators in programming. The equal to operator is used to compare two values and see if they are equal, while the assignment operator is used to assign a value to a variable.

Example of using the equal to operator

if (x == y):

print("x and y are equal")

Example of using the assignment operator

x = y

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')

There are three blocks of code in the code you provided:

The first block is the assignment statement spam = 0. This block assigns the value 0 to the variable spam.

The second block is the if statement if spam == 10:. This block will only be executed if the value of the variable spam is equal to 10.

The third block is the else statement else:. This block will only be executed if the if statement is not executed.

The output of the code will be ham because the value of the variable spam is not equal to 10.

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!")

This code will first prompt the user to enter a number. The user's input will be stored in the variable spam. The if statement will then check if the value of spam is equal to 1. If it is, the code will print "Hello". If the value of spam is equal to 2, the code will print "Howdy". Otherwise, the code will print "Greetings!".

Enter a number: 1

Hello

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

If my program is stuck in an endless loop, I will press the following keys to stop it:

Ctrl+C on Windows, Linux, and macOS

Command+C on macOS

Ctrl+Break on some Windows systems

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

Use **break** if you want to exit a loop when a certain condition is met. For example, you might use break to exit a loop when a user enters a valid number.

Use **continue** if you want to skip the remainder of the current iteration of a loop when a certain condition is met. For example, you might use continue to skip the remainder of the current iteration of a loop when a number is negative.

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

The difference between range(10), range(0, 10), and range(0, 10, 1) is the range of numbers that are generated.

* range(10) generates the numbers from 0 to 9, inclusive.
* range(0, 10) generates the numbers from 0 to 9, exclusive.
* range(0, 10, 1) generates the numbers from 0 to 9, inclusive, with a step size 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.

Loop:

for i in range(1, 11):

print(i)

While loop:

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?

from spam import bacon

bacon()