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

Ans: The two values of the Boolean data type in Python are **True** and **False**. They are written exactly as shown.

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

The three types of Boolean operators are **And, Or, Not.**

1. And - The operator returns True if the both conditions are true else it will return false.
2. Or – The Operator returns True if at least one condition is True, else it will return False.
3. Not – If the condition is True this operator will return False. Basically, it returns the opposite Boolean value.

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:

True and True = True

True and False = False

False and True = False

False and False = False

2. Or Operator:

True and True = True

True and False = True

False and True = True

False and False = False

3.Not Operator:

not True = False

not False = True

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?

The six comparison operators in Python are:

1. **==** : Checks if two values are equal.

2. **!=** : Checks if two values are not equal.

3. **>** : Checks if the left value is greater than the right value.

4. **<** : Checks if the left value is less than the right value.

5. **>=** : Checks if the left value is greater than or equal to the right value.

6. **<=** : Checks if the left value is less than or equal to the right value.

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) This is a comparison operator and it checks if two values/operand are equal.

2. = (Assignment operator) This is a assignment operator and it assigns a value to a variable. It helps in storing value.

7. Identify the three blocks in this code:

**Block 1:**

spam = 0

if spam == 10:

print('eggs')

**Block 2:**

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?

Ctrl + C

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

**Break** immediately breaks the loop, when break encountered inside a loop programme continues with the next line of code after loop.

**Continue** statement doesn’t exits from the loop immediately rather it skips the rest of the code inside the current iteration and continue with the next iteration of loop.

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

Above three ranges produce the same sequence of numbers from 0 to 9 but there is slightly difference in how they have writer.

range(10) = By default it produces the sequence of numbers from 0 to 10, but not including 10, it is equivalent to 0 to 9.

range(0,10) = In this range it is specified to start from the 0 to 10, it also produces the sequence of numbers from 0 to 10, but not including 10, it is equivalent to 0 to 9.

range(0,10,1) = In this range it is specified to start from the 0 and to but not including 10, with step of 1 with sequence, it is also equivalent to 0 to 9.

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 Loop:**

for num in range(10):

print(num)

**While Loop:**

num = 0

while num < 10:

print(num)

num += 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()