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

The two values of boolean data type are **True** and **False.**

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

The three different types of boolean operators are **AND, OR, NOT**. They return True or False values based on the given condition. AND, OR work on two operands to evaluate the logical expression result, which is True or False. The NOT operator works with single operand and returns the value as True if the boolean value is False and vice-versa.

Example: p AND q 🡪 returns true if both p and q are true

p OR q 🡪 returns true if either p or q are true

not p 🡪 returns true if p 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 evaluates).

1. **AND operator**

|  |  |  |
| --- | --- | --- |
| **P** | **Q** | **P AND Q** |
| 1 | 0 | False |
| 0 | 1 | False |
| 1 | 1 | True |
| 0 | 0 | False |

**2. OR operator**

|  |  |  |
| --- | --- | --- |
| **P** | **Q** | **P OR Q** |
| 1 | 0 | True |
| 0 | 1 | True |
| 1 | 1 | True |
| 0 | 0 | False |

**3. NOT operator**

|  |  |
| --- | --- |
| **P** | **NOT P** |
| 1 | False |
| 0 | True |

4. What are the values of the following expressions?

(5 > 4) and (3 == 5) 🡪 True AND False = False

not (5 > 4) 🡪 NOT(True) = False

(5 > 4) or (3 == 5) 🡪 True OR False = True

not ((5 > 4) or (3 == 5)) 🡪 NOT(True) = False

(True and True) and (True == False) 🡪 True AND False = False

(not False) or (not True) 🡪 True OR False = True

5. What are the six comparison operators?

The six relational operators are (**<, >, ==, >=, <=, !=** ). They will return True or False value after evaluating the expression on both the sides of the operator. Examples of comparison expressions are (a + b) >= (c - d), (a\*b)!= (c – d).

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

**Equal to (==) is a comparison operator** is used to compare the expression on the left hand side with that on the right hand side of the operator and evaluate the result to True or False.

Example: a+b == c+d 🡪 this expression evaluates to true if a+b is equal to c+d or false otherwise.

**Assignment operator** is used to assign the value on the right hand operand to the variable on the left. Example: a = 10, here the variable a is assigned the value of 10.

7. Identify the three blocks in this code:

spam = 0

if spam == 10:

print('eggs') 🡪 **Block 1**

if spam > 5:

print('bacon') 🡪 **Block 2**

else:

print('ham')

print('spam')

**Block 3**

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 the number"))

if spam == 1:

print("Hello")

elif spam == 2:

print("Howdy")

else:

print("Greetings")

**Output:**

Enter the number 10

Greetings

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

**CTRL + C** keys are used to come out of the infinite loop.

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

The **break** is used to terminate or stop the execution of the loop. When a break statement is encountered in a loop, the loop terminates and the control will move to the next statement present after the loop in the program. A break is used to stop a loop when a condition is met at a particular point in its execution.

The **continue** statement is used to terminate the execution of the current iteration of the loop. The control of the program will resume from the next iteration of that loop. Unlike break statement it does not stop the execution of the loop, it just stops the current iteration and moves to the next.

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

The **range(start, stop, step-size)** function returns a range object which is a sequence of characters. It is used in a for loop to iterate over sequences. Floating point numbers cannot be used in range function. If start parameter is not provided, default value is zero. The default value of step-size is 1.

**range(10)** 🡪 Returns a range object and on converting this object to a list, we get a sequence of first 10 numbers starting from 0 and up to 9 (10 is excluded).

**range(0,10)** 🡪 This is similar to range(10) function. On converting the return object to a list, we get a sequence of first 10 numbers starting from 0 and up to 9 (10 is excluded).

**range(0,10,1)** 🡪 On converting the return object to a list, we get a sequence of first 10 numbers starting from 0 and up to 9 (10 is excluded) with a step-size of 1.

All the three functions above give the **same output.**

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.

Example using for loop:

**for i in range(0,11):**

**print(i)**

Example using while loop:

**i = 0**

**while i < 11:**

**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()