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

**Ans.1.**The two values of the Boolean data the following:-

* 1. **True**
  2. **False**

They are written exactly as shown because python is a case sensitive program.

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

**Ans.2** The 3 different types of Boolean operators are the following:-

1. AND operator (“not”)
2. OR operator (“or”)
3. NOT operator (“not”)
4. 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 ).

**AND operator:**

| A | B | A and B |
| --- | --- | --- |
| True | True | True |
| True | False | False |
| False | True | False |
| False | False | False |

**OR operator:**

| **A** | **B** | **A or B** |
| --- | --- | --- |
| True | True | True |
| True | False | True |
| False | True | True |
| False | False | False |

**NOT operator:**

| **A** | **not A** |
| --- | --- |
| True | False |
| False | True |

4. What are the values of the following expressions?

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

🡪 The expression should be false, because the **AND** operator requires both expressions to be True for the result to be True.

(B) not (5 > 4)

🡪 The **NOT** operator negates the value, so the result of the expression is False.

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

🡪 True because the **OR** operator requires only one expression to be True for the result to be True.

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

🡪 The **NOT** operator negates the value, so the result of the expression is False.

**(E)** (True and True) and (True == False)

🡪 False because the **AND** operator requires both expressions to be True for the result to be True.

(**F**) (not False) or (not True)

🡪 True because the **OR** operator requires only one expression to be True for the result to be True.

5. What are the six comparison operators?

**ANS.5** The six comparison operators are:

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

ANS.6 When comparing two numbers for equality, the equal to operator (==) is employed. It determines whether the values on both sides are the same. For instance, x == y would be True if x equaled y.

(B) In contrast, the assignment operator (=) is used to assign a value to a variable. For instance, x = 5 assigns the value 5 to the variable x.

7. Identify the three blocks in this code:

(A) spam = 0

🡪 This block assigns the value 0 to the variable spam.

(B) if spam == 10:

print('eggs')

🡪 This block checks if the value of **spam** is equal to 10. If it is, it will execute the **print('eggs')** statement.Otherwise it will show error.

(C) if spam > 5:

print('bacon')

else:

print('ham')

print('spam')

print('spam')

🡪 This block determines whether the spam value is larger than 5. If it is, the print('bacon') command will be executed. If the condition is not met, the code inside the else block, which includes the print('ham') statement, is executed. Furthermore, the final two print statements,'spam' and'spam,' are not indented and hence run regardless of the criteria.

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 = 1

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?

Ans. 9 If your programme is stuck in an endless loop, you can interrupt it by pressing Ctrl+C (or Command+C on a Mac).

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

The distinction between break and continue statements is as follows:

• break is used to exit the current loop completely. It ends the loop and resumes execution at the next sentence following the loop.

• continue is used to skip the remaining code in the current loop iteration and proceed to the next iteration. It does not end the loop; rather, it immediately begins the following iteration.

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

**Ans.11**  In a **for** loop, the expressions **range(10)**, **range(0, 10)**, and **range(0, 10, 1)** have the same effect:

• range(10) iterates through the values 0 to 9 (excluding of 10) with a step of 1 (the default step value).

• range(0, 10) specifies the start value as 0 (inclusive) and the end value as 10, with a step of 1.

• range(0, 10, 1) specifies the start value as 0, the end value as 10, and the step value as 1.

1. 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) Print numbers 1 to 10 using a **for** loop

for num in range(1, 11):

print(num)

(B) Printing numbers 1 to 10 using a **while** loop

num = 1

while num <= 10:

print(num)

num += 1

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

**Ans.13** import spam

spam.bacon()