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

**Q1 Ans-** There two Boolean values in python. That are Ture and False.

Ex. A=True, B=False

This indicate that variable A has True (Boolean type 1) value and variable B has False value (Boolean type 0).

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

**Q2 Ans –**

There are three main types of Boolean operators.

1. and – It takes two operands/conditions and return True if both the operands or conditions are true else it will return False.

EX. x=8

y=5.0

z= type(x)==int and type(y)==float

Here z will return True as both the conditions are true.

1. or - It takes two operands/conditions and return True if both the operands or conditions are true or any of them are true else it will return False.

EX. x=8

y=’5’

z= type(x)==int or type(y)==float

Here z will return True as one of the conditions is True.

1. not – not Boolean operator checks for a reverse condition and if it matches then it returns True else it returns False.

Ex. a=100

not(a>1000) – This will return True as a(100) is not greater than 1000.

In short, whatever outcome we get from operands or conditions. not Boolean operator simply reverse it.

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).

**Q3. Ans**

Truth table for **and** Boolean operator Truth Table for or Boolean operator

|  |  |  |
| --- | --- | --- |
| a | b | a and b |
| False | False | False |
| False | True | False |
| True | False | False |
| True | True | True |

Truth Table for **or** Boolean operator

|  |  |  |
| --- | --- | --- |
| a | b | a or b |
| False | False | False |
| False | True | True |
| True | False | True |
| True | True | True |

Truth table for **not** Boolean operator

|  |  |
| --- | --- |
| a | not a |
| True | False |
| False | True |

4. What are the values of the following expressions?

**Q4 Ans**

(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?

**Q5 Ans**

Below are the 6 comparison operators.

1. == 🡪 This operator is used to check for equality between two or more variables or values.
2. > 🡪 This operator is used to check whether the left side value is greater than right value.
3. < 🡪 This operator is used to check whether the left side value is lesser than right value.
4. >= 🡪 This operator is used to check whether the left side value is greater than or equals to right value.
5. <= 🡪 This operator is used to check whether the left side value is lesser than or equals to right value.
6. != 🡪 This operator is used to check if a value is not equal to another value.

Note 🡪 All these operators return either True or False. If the condition is met then it will return True else it will return False.

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

**Q6 Ans –**

Equal to operator is a comparison operator which is basically used to relate the equality between two or more values and it returns True if condition is met else it returns False and ‘==’ is used for equal to operator. However, assignment operator is used to assign some values to a variable or objects in python. ‘=’ is used for assignment operator. Please refer below example.

course=’FSDS’ # Here course is a variable and we assigned FSDS to this variable using assignment operator (=).

language =’Python’

print(language==’python’) # Here we have assigned Python as a value to variable language. Now, in the print statement we ar checking if the variable language and ‘python’ are equals or not. It will print False as output because language and python is not equal here.

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')

**Q7 Ans –**

First block is at - if spam == 10:

2nd block is at -- if spam > 5:

3rd block is at – else:

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.

**Q8 Ans -**

spam= int(input(‘Enter a numeric value for spam varaible’))

‘’’ Here I am taking user input so that it can be tested on all the three conditions. We can also remove the input part and can just assign some values to spam variable at the beginning of code.

‘’’

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?

**Q9 Ans –**

We can press **ctrl+c** and we can stop program execution explicitly.

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

**Q10 Ans –**

The basic difference between break and continue is –

 The break statement terminates the whole iteration of a loop whereas continue skips the current iteration.

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

**Q11 Ans –**

In a for loop basically there is no any difference between all these given ranges. As we know that by default range will start from 0 and will go till end\_value-1 with a step of 1. Here, the same thing is happening in all these 3 range functions. It will generate range values from 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.

**Q12 Ans.**

**Using for loop**

for num in range(1,11):

print(num)

**Using while Loop**

num=1

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?

**Q13 Ans -**

Import spam as sp

sp.bacon()