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

TRUE and FALSE are the two values of Boolean data type. We can obtain Boolean values by logical or relational statements.

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

AND, OR and NOT are three different Boolean operators.

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).

In the below tables 0 indicates FALSE and 1 indicates TRUE.

AND Boolean operator:

If both inputs are true then only true else false.

|  |  |  |
| --- | --- | --- |
| input | | output |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

OR Boolean operator:

If both inputs are false then only false else true.

|  |  |  |
| --- | --- | --- |
| input | | output |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

NOT Boolean operator:

If input is false then output is true and vice versa.

|  |  |
| --- | --- |
| input | Output |
| 0 | 1 |
| 1 | 0 |

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?

Six comparison operators are

= equal to

!= not equal to

< less than

> greater than

<= less than or equal to

>= greater 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.

A single equal to (=) is for assignment operation where as double equal to (==) is for comparison operation. For example

Name = “mouli” is an assignment operation where mouli is assigned to variable Name.

001 == 999 is an comparison operation where we are checking whether the two values are same/equal or not.

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')

print('spam')\_\_\_\_\_\_\_\_\_\_\_\_\_block 3

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.

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?

If we are stuck in an infinite loop then we must press CRTL+c to end the loop.

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

Break ends the whole loop, where as continue ends the current iteration.

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

According to the above three statements there will be no difference in the output. The output will be 0,1,2,3,4,5,6,7,8,9. But when coming to the concept of range function, when we change the values in above statements there will be a huge difference.

range(10) simply returns 0,1,2,3,4,5,6,7,8,9.

range(1,10) returns 1,2,3,4,5,6,7,8,9. Comparing the results with above output there will be no 0 value as the output. This clearly tell us that when we give two arguments to range function it includes the first argument and excludes the last argument value.

Lastly, range(0,10,2) returns 0,2,4,6,8. When we provide the third argument in range function it considers as step, which means that every second value is returned as the output. If 3 is provided in the function then every third value will be returned.

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.

Using for loop:

for i in range(1,11):

print(i)

output:

1

2

3

4

5

6

7

8

9

10

Using while loop:

i=1

while i in range(11):

print(i)

i+=1

output:

1

2

3

4

5

6

7

8

9

10

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

We need to use the (.) operator to invoke the function in side a class or module.

spam.bacon() invokes bacon function from spam module.