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

-> Boolean data type has two values **‘True’** and **‘False’**.

Python is case sensitive, so it needs to be written as initial caps ‘True’ and ‘False’ or else the interpreter will not recognise them.

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

-> The three different types of Boolean operators are:

* **Not:** The Boolean ‘not’ operator needs one argument and returns the opposite result.
* **And:** The Boolean ‘and’ operator returns True if both the inputs are true, else returns False.
* **Or:** The Boolean ‘or’ operator returns True if any of the inputs are true, else returns 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 evaluate ).

-> **Boolean Not operator:**

|  |  |
| --- | --- |
| A | Not A |
| True | False |
| False | True |

**Boolean And operator:**

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

**Boolean Or operator:**

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

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** **( != )**
* **Greater than** **( > )**
* **Greater than equal to** **( >****= )**
* **Less than** **( < )**
* **Less than 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.

-> Assignment operator is a **single equal sign** ‘ = ‘ which is used to assign values to variables, but equal to operator is **double equal sign** ‘ == ‘ which is used to check if two variables are equal or not.

**For example:**

A = 10 # assigning value 10 to A

B = 5 #assigning value 5 to B

print( A == B ) #checking if both the variables are equal or not

This returns: **False**

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')

-> **Line 3:** print(‘eggs’) is the 1st Block

**Line 5:** print(‘bacon’) is the 2nd Block

**Line 7,8,9:** print(‘ham’), print(‘spam’), print(‘spam’) is the 3rd Block

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 = 0

if spam == 1:

print('Hello')

elif spam == 2:

print('Howdy')

else:

print('Greetings!')

The output is: **Greetings!**

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

-> **ctrl + C** should be pressed to stop an endless loop.

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

-> The **break** statement will terminate the loop containing it and controls the program flow to the statement after the body of the loop. Its basic use is to exit out of the loop and stop the flow of execution.

The **continue** statement is used to skip the code’s inside loop for the current iteration. The loops do not terminate but continuously goes on with the next iteration.

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

-> **range(10)** means a range containing values 0 to 9. So, the for loop will print numbers 0 to 9.

**range(0, 10)** means we are predefining a starting index as 0 and it will print the numbers 0 to 9.

**range(0, 10, 1)** means starting index is defined as 0, it will run 10 times and steps defined as 1. So, it will print the numbers 0 to 9 here also.

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 x in range(1,11):

print(x)

**While loop:**

x = 1

while x <= 10:

print(x)

x = x + 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()**

Where spam is the module name and bacon() is the function.