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

**Ans:-** There are two Boolean data type True and False and these both are case sensitive in python. You can write the like True and False in which first letter should be uppercase followed by the lowercase letter.

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

**Ans:- Three Boolean operators are as follows:**

1. **and** – and operator require all conditions to be true for the overall expression to be True. The entire expression can be False if any condition is not true.
2. **0r** – or operator requires at least one condition to be true. The entire expression can only be False if all the conditions are not true.
3. **not** – not operator reverses (negates) the value of a Boolean expression. If the value is True, not makes it False or vice-versa.
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 evaluates).

**Ans:- 1.** AND (and) Truth Table

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

**2.** OR (or) Truth Table

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

**3.** NOT (not) Truth Table

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

4. What are the values of the following expressions?

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

not (5 > 4)

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

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

(True and True) and (True == False)

(not False) or (not True)

**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?

**Ans:- Six comparison operator are:**

1. == (Equal to)
2. != (not Equal to)
3. < (less than)
4. < (greater than)
5. <= (less than or equal to)
6. >= (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.

**Ans:- Equal operator ( == )** compares two values to check if they are equal and evaluates to a boolean. For Ex:

if a == 100:

print("a is 100")

**Assign Operator ( = )** assign a value to a variable. For ex: my\_string = "Hello Danish" which means "Hello Danish" is assign to my\_string variable.

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')

**Ans:-**

1. Block 1- inside first if statement

if spam == 10:

print('eggs')

1. Block 2- if else structure

if spam > 5:

print('bacon')

else:

print('ham')

1. Code outside any if

print('spam')

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.

**Ans:-**

**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:- Ctrl + C** use to prevent sucking the programme in an endless loop.

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

**Ans:- break** exit the loop entirely and used to early termination of loop, while **continue** skips the rest of the current loop iteration based on a condition and continue to the next iteration.

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

**Ans:-** There is no difference between above ranges as the all do same thing. In **range(10)** it gives from 0 to 9, in **range(0,10)** means it returns 0 to 9 excluding 10, and in **range(0, 10, 1)** means starting from 0 up to (but not including) 10, with a step of 1.

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.

**Ans:- Using for loop:-**

for i in range(1,11):

print(i)

**Using while loop:-**

i = 1

while i <= 10:

print(i)

i = i + 1

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

**Ans:- spam.bacon()**