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

The two values of the Boolean data type are True or False. In case we want to initialise with True or False.

result=True

is\_pass=False

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

1. Comparison Operators:  Following six operators evaluate the expression to a Boolean value.

|  |  |
| --- | --- |
| Operator | Meaning |
| == | Equal to |
| < | Less than |
| > | Greater than |
| != | Not equal to |
| <= | Less than or equal to |
| >= | Greater than or equal to |

2. Binary Boolean Operators: These operators are the ones that operate on two values which are both Boolean.

There are the following subtypes of Boolean operators:

1. And operator
2. Or operator

Code:

a = True

b = False

a and b

False

a or b

True

#### Not Operator: The ‘not’ operator is the logical Boolean Operator, which compliments the variable’s current Boolean value. That is, if the value is ‘true’, then the not operator will modify it to ‘false’ and vice versa.

1. 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 ).

|  |  |
| --- | --- |
| Operator | Meaning |
| == | Equal to |
| < | Less than |
| > | Greater than |
| != | Not equal to |
| <= | Less than or equal to |
| >= | Greater than or equal to |

>> The truth table for the ‘and’ operator.

|  |  |
| --- | --- |
| Expression | Evaluations |
| True and True | True |
| True and False | False |
| False and True | False |
| False and False | False |

>> The truth table for the “or” operator.

|  |  |
| --- | --- |
| Expression | Evaluations |
| True and True | True |
| True and False | True |
| False and True | True |
| False and False | False |

>> The truth table for the “Not” operator.

| **not** | **x** | **Returns** |
| --- | --- | --- |
| not | True | False |
| not | False | True |

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

1. What are the six comparison operators?

The following six operators evaluate the expression to a Boolean value.

|  |  |
| --- | --- |
| Operator | Meaning |
| == | Equal to |
| < | Less than |
| > | Greater than |
| != | Not equal to |
| <= | Less than or equal to |
| >= | Greater than or equal to |

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

The “Equal to” operator is used to check if the operands mentioned are equal or not. On the other side, the assignment operator is used to assign the value to the variable.

In case we want to assign the value of variable ‘a’ with 10 then we will use the assignment operator.

Code

a=10

In case if I want to check if the value “a” variable is equal to 10 (or any other value) or not then I will use the equal to operator.

Code:

a=10

result= (a==20)

print(result)

o/p🡪 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')

The first block consists of the if block as below:

if spam == 10:

print('eggs')

The second block consists of the second if block:

if spam > 5:

print('bacon')

The third block consists of the else block as below:

else:

print('ham')

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.

**spam=3**

**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?

CTRL+C

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

Breaks leave the loop entirely however continue will jump for the next iteration of the loop.

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

Syntax of range() is as below:

*range(start, stop, step)*

the first option is range(10) will print all the series of values from 0 till 9 but not 10 so 10 will act as a stop argument.

the second option is range(0, 10). In this 0 will act as the start value and 10 will act as the stop value.

The third option is range(0, 10, 1), In this 0 will act as the start value and 10 will act as the stop value and 1 will act as a step value which signifies that it will increment by 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.

**for i in range(1,11):**

**print(i)**

**i=1**

**while i <=10:**

**print(i)**

**i+=1**

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

This function can be called with spam.bacon().