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

The two values of Boolean data types are true and false where we write them as 1st letter capital of their Spelling which is True and False and not enclosed in quotes

[True and False, using capital T and F]

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

The three different types of Boolean operators are “and” , “or” and “not” used for many logical operations

“and” = it returns true if both operands on its left and right are true and even if a single left or right operand is a false then it will return a false hence for the condition to be true both of the operands shall be true and if any single operands are false then the whole expression will be false

“Or” = as the name suggests or for the whole expression to be true any single operand shall be true and if both the operands are false then the whole expression will be false

‘Not’ = it returns the opposite boolean value of its operand if the value is true then it will return false and if the value is false then it will return true

**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 ).**

AND operator's truth table

|  |  |  |
| --- | --- | --- |
| TRUE | TRUE | =TRUE |
| TRUE | FALSE | =FALSE |
| FALSE | TRUE | =FALSE |
| FALSE | FALSE | =FALSE |

‘’Not” operator's truth table

|  |  |
| --- | --- |
| TRUE | =FALSE |
| FALSE | =TRUE |

‘OR’ operator's truth table

|  |  |  |
| --- | --- | --- |
| 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?**

‘>’ it returns true if the left operand is greater than the right operand or else false

‘<’ it returns true if the left operand is less than Right operand or else false

‘>=’ it returns true if the left operand is greater than or equal to the right operand or else it returns false

‘<=’ it returns true if the left operand is less than or equal to the right operand or else it returns false

‘==’ it returns true if the left operand is exactly equal to the right operand or else it returns false

‘!=’ it returns true if the left operand is not equal to the right operand or else it returns false

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

So here the assignment operator and the equal to are almost similar looking but the functionality is way different the assignment operator is a single equal to sign(‘=’) which is used to assign values to variable it takes right side value of an operator and assigns that value to the left hand side of the operator whereas equal to operator ‘==’ is an comparison operator where it checks if the left hand side operand is exactly equal to the right hand side operand and returns true if it's true or else if it's not equal then it returns false it is mainly used to check equality of two different variables or values if the both are same or not where is just equal/ assignment operator is used to assign values

**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=================**

**1st First block** contains only a single line.

spam = 0

Line two and three are the **2nd block** starting from if spam till printing eggs.

if spam == 10:

print('eggs')

**3rd Block** Line 4 to 7

if spam > 5:

print('bacon')

else:

print('ham')

{print('spam’) , print('spam')}

Above print spam is not part of any blogs hence it will be executed regardless of the condition as there is no indentation required for these two prints statement.

**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.**

x = int(input("Enter the Number - "))

spam = (x)

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?**

I will press Ctrl + C to end the program if stuck in an endless loop

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

The main difference between the break statement and the continue statement is that the break statements end the loop and comes out of it whereas the continue statement is used to skip a specific iteration in a loop. Let me give an example if a loop is printing from a range of 1 to 6 and if we put a break statement that if the iteration comes at 3 the loop will break then the whole loop will be terminated and the code ahead of the loop will be printed whereas if we have used continue statement in the place of break then the loop will be continuing just skipping the third iteration and continue till 6th iteration

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

In range(10) generates a sequence of numbers starting from 0 which is default and goes till 10 which is excluded hence till 9 and the Default step size is 1 (which is the incrementation size)

In range(0,10) The starting sequence has been defined where it goes similarly till 10(where 10 is excluded) and hence till 9 and the Default step size is 1 (which is the incrementation size)

In range(0,10,1) Its where ending , starting and stepsize is mentioned

The main difference between the **range(10), range(0, 10), and range(0, 10, 1) is** how their starting and ending and step size have been mentioned and where not mentioned the default values are automatically taken into consideration

**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.**

**A screenshot of a computer program

Description automatically generatedA screenshot of a computer program

Description automatically generated**

**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()