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

**Answer:** The two values of the Boolean data type are **true** and **false.** A Booleandatatype is declared with the bool keyword and it will take the values true or false. When the value is returned true=1 and false=0.

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

**Answer:** The three different types of Boolean operators are **AND**, **OR** and **NOT.**

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).

**Answer:**

**AND Table OR Table**

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **A OR B** |
| True | True | True |
| True | False | True |
| False | True | True |
| False | False | False |

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **A AND B** |
| True | True | True |
| True | False | False |
| False | True | False |
| False | False | False |

**NOT 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)

**Answer:**

(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?

**Answer:**

|  |  |
| --- | --- |
| **Operators** | **Description** |
| < | It is used to check if the values of left operand is less than the values of right operand, then the condition becomes true. |
| > | It is used to check if the values of left operand is greater than the values of right operand, then the condition becomes true. |
| <= | It is used to check if the values of left operand is less than or equal to the values of right operand, then the condition becomes true. |
| >= | It is used to check if the values of left operand is greater than or equal to the values of right operand, then the condition becomes true. |
| == | It is used to check if the values of two operands are equal, then the condition becomes true |
| != | It is used to check if the values of two operands are not equal, then the condition becomes true |

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

**Answer:**

|  |  |
| --- | --- |
| **Equal to Operator (==)** | **Assignment (=)** |
| The Equal to operator (==) checks whether the two given operands/values are equal or not. (LHS==RHS) | The assignment operator (=) used to assign the value on the right to the variable on the left. |
| **Example:** a=5 (Here I’m assigning the value 5 to the variable ‘a’) | **Example:** a==5 ( Here I’m checking/comparing the value of ‘a’ is equal to 5 or not.) |

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')

**Answer:**

spam = 0

if spam == 10:

print('eggs') # here indentation block we will get

if spam > 5:

print('bacon') # here indentation block we will get

else:

print('ham')

print('spam') # here indentation block we will get

print('spam')

If we will remove the indentation block then will get the output as:

ham

spam

spam

The correct program is:

spam = 0

if spam == 10:

print('eggs')

if spam > 5:

print('bacon')

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.

**Answer:**

spam=int(input("Enter the value of 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?  
  
**Answer:** In this case, we can use the **break** statement to come out of the infinite loop.

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

|  |  |
| --- | --- |
| **Break** | **Continue** |
| for i in range(1, 100):  if i > 50:  break  print(i) | for i in range(1, 20):  if i%2==0:  continue  print(i) |
| In the above example, if i is greater than 50 it will come out of the loop. | In the above example: if the number is even it will continue the loop. Otherwise, the number is an odd number means it will come out of the loop. |
| It is used to stop the execution of the loop at a specific condition. | It is used to skip a particular iteration of the loop. |
| In the break statement, the control exits from the loop. | In the continue statement, the control remains within the loop. |

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

**Answer:**

|  |  |
| --- | --- |
| range(10) | stops at end or at last value of 10(**excluding the index value 10**) |
| range(0,10) | It indicates the range function starts at index value: 0 and ends/stops at last value 10**(excluding the index value 10)** |
| range(0, 10, 1) | It indicates the range function starts at index value: 0 and ends/stops at last value 10**(excluding the index value 10**) with an increment of stepsize 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.

**Answer:**

**For:**

for i in range(1,11):

print(i)

**While:**

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?

**Answer:** This function can be called spam.bacon()