**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. We write them as True and False (starting with first letter capital).

**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 :- 1) AND :-**

|  |  |  |
| --- | --- | --- |
| **X** | **Y** | **X AND Y** |
| **0** | **0** | **0** |
| **0** | **1** | **0** |
| **1** | **0** | **0** |
| **1** | **1** | **1** |

**2) OR :-**

|  |  |  |
| --- | --- | --- |
| **X** | **Y** | **X OR Y** |
| **0** | **0** | **0** |
| **0** | **1** | **1** |
| **1** | **0** | **1** |
| **1** | **1** | **1** |

**3) NOT :-**

|  |  |
| --- | --- |
| **X** | **NOT X** |
| **0** | **1** |
| **1** | **0** |

**4) XOR :-**

|  |  |  |
| --- | --- | --- |
| **X** | **Y** | **X XOR Y** |
| **0** | **0** | **0** |
| **0** | **1** | **1** |
| **1** | **0** | **1** |
| **1** | **1** | **0** |

**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 :- 1) (5 > 4) and (3 == 5): False** - This expression is False because both conditions must be true for the expression to be true and 3 does not equal 5.

**2) not (5 > 4): False** - This expression is False because not (5 > 4) is the same as saying 5 is not greater than 4, which is not true.

**3) (5 > 4) or (3 == 5): True** - This expression is True because either condition must be true for the expression to be true and 5 is greater than 4.

**4) not ((5 > 4) or (3 == 5)): False** - This expression is False because not ((5 > 4) or (3 == 5)) is the same as saying it is not true that either condition is true, which is false.

**5) (True and True) and (True == False): False** - This expression is False because both conditions must be true for the expression to be true and True does not equal False

**6) (not False) or (not True): True** - This expression is True because either condition must be true for the expression to be true and not False is the same as saying True.

**5. What are the six comparison operators?**

**Answer :-**

1. Equal to (==)
2. Not equal to (!=)
3. Greater than (>)
4. Less than (<)
5. Greater than or equal to (>=)
6. Less 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.**

**Answer :-**

* The equal to operator (==) is used to compare two values for equality. It returns a Boolean value indicating whether the two values are equal.

**For example :-**

x = 5

y = 10

x == y # False

* The assignment operator (=) is used to assign a value to a variable.

**For example :-**

x = 5

y = 10

x = y # x is now equal to 10

**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 : -**

* **The first block is : -**

spam = 0.

* **The second block is :-**

if spam == 10:

print(‘eggs’)

if spam > 5:

print(‘bacon’)

else:

print(‘ham’)

* **The third block is :-**

print(‘spam’)

print(‘spam’)

**OR**

* **The first block is : -**

spam = 0

* **The second block is : -**

if spam == 10:

print('eggs')

* **The third block is : -**

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 an integer: "))**

**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 :-**

To stop an infinite loop, press CTRL+C (on Windows) or CTRL+Z (on Mac) to terminate the programme.

**Or**

While working in IDE we will stop the kernel to terminate the infinite loop.

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

**Answer :-**

* **Break :-** Break is used to exit a loop completely. Once the break statement is executed, the program control flow moves out of the loop.
* **Continue :-**  Continue is used to skip the current iteration of the loop. Once the continue statement is executed, the program control flow moves to 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)?**

**Answer :-**

* **Range(10) :-**  will generate a sequence of numbers from 0 to 9 (length 10).
* **Range(0,10) :-** will generate a sequence of numbers from 0 to 9 (length 10).
* **Range(0,10,1) :-**  will generate a sequence of numbers from 0 to 9 (length 10).

The difference between them is the starting point and step size.

Range(10) has an implicit starting point of 0 and a step size of 1.

Range(0,10) also has a starting point of 0 and a step size of 1.

Range(0,10,1) has an explicit starting point of 0 and a step size 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.**

**Answer :-**

* **Using For Loop :-**

for i in range(1,11):

print(i)

* **Using While Loop :-**

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 :-**

* **We will call bacon() after importing from spam like this:-**

**spam.bacon()**