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

Sol. The boolean data type is either True or False. In Python, Boolean variables are defined by the True and False keywords.

Eg:=a = True

>>> type(a)

<class 'bool'>

 the keywords True and False must have an Upper Case first letter. Using a lowercase true returns an error.

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

Sol. The three different types of boolean operators are:-

* The AND operator (&& or "and")
* The OR operator (|| or "or")
* The NOT operator (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 ).

Sol. 1. The AND operator truth table:

|  |  |  |
| --- | --- | --- |
| a | B | a and b |
| False(0) | false(0) | false(0) |
| True(1) | false(0) | false(0) |
| false(0) | true(1) | false(0) |
| true(1) | true(1) | true(1) |

2.the OR operator.

|  |  |  |
| --- | --- | --- |
| a | B | a or b |
| False(0) | false(0) | false(0) |
| True(1) | false(0) | true(1) |
| false(0) | true(1) | true(1) |
| true(1) | true(1) | true(1) |

3. the Not operator

|  |  |
| --- | --- |
| a | Not a |
| False(0) | True(1) |
| True(1) | false(0) |

4. What are the values of the following expressions?

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

Sol. False

not (5 > 4)

sol. False

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

Sol. True

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

sol. False

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

Sol. False

(not False) or (not True)

Sol. True

5. What are the six comparison operators?

Sol.

|  |  |  |
| --- | --- | --- |
| **Operator** | **Meaning** | **Example** |
| > | Greater than - True if left operand is greater than the right | x > y |
| < | Less than - True if left operand is less than the right | x < y |
| == | Equal to - True if both operands are equal | x == y |
| != | Not equal to - True if operands are not equal | x != y 10+2 == 10  False |
| >= | Greater than or equal to - True if left operand is greater than or equal to the right | x >= y |
| <= | Less than or equal to - True if left operand is less than or equal to the right | x <= y |

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

Sol. In Python = symbol is defined as **assignment**operator. It requires one variable on its left and an expression on its right. Value of the expression on right is assigned to variable on left. Expression and name of variable are not interchangeable.

Eg:- a=10, b= 2+5

The == symbol is a comparison operator and called **equal to** operator. It returns true if operands on either side are equal, otherwise it returns false.

Eg:- 10+2 == 10

Sol.False

7. Identify the three blocks in this code:

Sol.

1)spam = 0

if spam == 10:

2)print('eggs')

if spam > 5:

3)print('bacon')

else:

3)print('ham')

2)print('spam')

1)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.

Sol:-

print ("input value of spam")

spam = int(input())

if spam == 1:

print('hello')

if spam == 2:

print('howdy')

else:

print('greetings!')

9.If your programme is stuck in an endless loop, what keys you’ll press?

Sol. Ctrl c

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

Sol. Break statement: - It terminates the current working loop and passes the control to the next statement, and if the break statement resides inside the nested loop, it passes control to the outer loop. It can be used with both while and for loops.

Continue statement: - It is the second form of the Loop Control Statement, very similar to the break statement. But when it comes to the working of the continue statement, it works just opposite to the break statement. Instead of terminating certain conditions, it jumps off to the very next condition. But it will continue the execution of the loop statement as per its name

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

Sol. range(10):- defines range upto 10.

range(0,10):-  defines the range as between 0 and 10.  
range(0, 10, 1) defines the range as between 0 and 10 with steps 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.

Sol:-

for i in range(1,11):

print(i)

using while loop:

n = 1

while n <= 10:

print(n)

n = n+1

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

Sol. import spam

Spam.Bacon():