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

Answer: The two values of Boolean data type are true and false. In python we write it as -
True (for the true value)
False (for the false value)

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

Answer: The three boolean operators in python are -
i. and operator
ii. or operator
iii. not operator

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:
i. and operator

A	B	Y
FALSE	FALSE	FALSE
FALSE	TRUE	FALSE
TRUE	FALSE	FALSE
TRUE	TRUE	TRUE

ii. or operator

A	B	Y
FALSE	FALSE	FALSE
FALSE	TRUE	TRUE
TRUE	FALSE	TRUE
TRUE	TRUE	TRUE

iii. not operator

A	Y
TRUE	FALSE
FALSE	TRUE

4. What are the values of the following expressions?

(5 > 4) and (3 == 5)
Answer: False

not (5 > 4)
Answer: False

(5 > 4) or (3 == 5)
Answer: True

not ((5 > 4) or (3 == 5))
Answer: False

(True and True) and (True == False)
Answer: False

(not False) or (not True)
Answer: True

5. What are the six comparison operators?

Answer: The six comparison operators in python are -
i. == (Equal to)
ii. != (Not equal to)
iii. > (Greater than)
iv. < (Less than)
v. >= (Greater than or equal to)
vi. <= (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: In python equal to operator(==) is used for comparison and checks if two values are equal. If the values values are equal it returns True otherwise returns False. For example,

7 == 7
It will return the boolean value True.

On the other hand, assignment operator(=) is used to assign a value to a variable. It assigns the value on the right side of the operator to the variable on the left side. For example,

a = 9
It will assign the value 9 to the variable a.

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:

1st if block

if spam == 10:
    print(‘eggs')

2nd if block

if spam > 5:
    print(‘bacon')

else block for the 2nd if block

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:

```
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: If my program is stuck in an endless loop then I will press Ctrl + C to interrupt the execution of the program.

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

Answer: break is used to exit the loop completely when a certain condition met. On the other hand continue is used to skip the iteration of the loop and move to the next iteration without executing the remaining code within the loop for that iteration.

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

Answer: A for loop will iterate through 0,1,2,3,4,5,6,7,8,9 in all these different variants of range function. Because,

range(10): Here this form specifies a single argument which represents the stop value. And takes the start value 0 and step size 1 as these are the default value of range function. So it will start from 0 and end at 9 as the range function excludes the stop value.

range(0,10): Here this form specifies two arguments which represents start value and stop value. Start value is 0 and stop value is 10 and step size is 1(default value).

range(0,10,1): Here this form specifies three arguments which represents start value, stop value and step size. Start value is 0 and stop value is 10 and step size is 1.

The only difference between these forms of range function is syntax used to specify the range. range(10), range(0, 10), and range(0, 10, 1) will all iterate over the same range of values, from 0 to 9.

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: After importing spam I will call bacon() using dot notation as dot notation is used to access members (functions, variables, classes, etc.) within a module.

```
import spam
spam.bacon()
```