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". In programming, these values are typically written in lowercase letters as "true" and "false" or as numeric values of 1 and 0, where 1 represents "true" and 0 represents "false".

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

**Answer:** Python has three Boolean operators, or logical operators: 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 operator:

The AND operator returns true if and only if both of its operands are true, otherwise it returns false. The truth table for the AND operator is as follows:

Operand 1 Operand 2 Result

true true true

true false false

false true false

false false false

OR operator:

The OR operator returns true if at least one of its operands is true, otherwise it returns false. The truth table for the OR operator is as follows:

Operand 1 Operand 2 Result

true true true

true false true

false true true

false false false

NOT operator:

The NOT operator returns the opposite of its operand. If the operand is true, NOT returns false. If the operand is false, NOT returns true. The truth table for the NOT operator is as follows:

Operand Result

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) is **False**

not (5 > 4) is **False**

(5 > 4) or (3 == 5) is **True**

not ((5 > 4) or (3 == 5)) is **False**

(True and True) and (True == False) is **False**

(not False) or (not True) is **True**

5. What are the six comparison operators?

**Answer:**

> : greater than

< : less than

>= : greater than or equal to

<= : less than or equal to

== : equal to

!= : not 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 and returns a Boolean value of True if they are equal, and False if they are not. On the other hand, the assignment operator = is used to assign a value to a variable.  
  
x = 5 # using the assignment operator to assign the value 5 to variable x

if x == 5: # using the equal to operator to compare the value of x with 5

print ("x is 5")

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:**

1. The first block consists of a single line of code:

spam = 0

This line assigns the value 0 to the variable spam.

2. The second block consists of three lines of code:

if spam == 10:

print('eggs')

if spam > 5:

print('bacon')

else:

This block contains two if statements and an else statement. However, the first if statement does not have any code nested inside it, which is a syntax error. The second if statement checks if spam is greater than 5, and if it is, it prints the string "bacon". Otherwise, control flow jumps to the else block.

3. The third block consists of two lines of code:

print('ham')

print('spam')

print('spam')

This block contains three print statements that output the strings "ham", "spam", and "spam" respectively. These print statements are not indented, so they are not part of the if-else statements above.

Note that the second block is not well-formed due to the missing indentation and the empty first if 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.  
  
**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 your program is stuck in an endless loop, you can press Ctrl + C (Control and C keys together) on your keyboard to interrupt the program and terminate the loop.

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

**Answer:** ‘break’ is used to terminate a loop prematurely, regardless of whether the loop condition has been satisfied or not. When a break statement is encountered in a loop, the loop is immediately terminated, and the program control is transferred to the statement immediately following the loop. This is useful when you need to exit a loop early based on a certain condition.  
  
‘continue‘ is used to skip the current iteration of a loop and continue with the next iteration. When a continue statement is encountered in a loop, the current iteration of the loop is immediately terminated, and the program control is transferred to the next iteration of the loop. This is useful when you want to skip certain iterations of a loop based on a certain condition.

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

Here's the difference between range (10), range (0, 10), and range (0, 10, 1) in a for loop:

- range (10) generates a sequence of numbers starting from 0 up to (but not including) 10, with a default step size of 1.  
  
This code will output the numbers from 0 to 9.

- range (0, 10) generates a sequence of numbers starting from 0 up to (but not including) 10, with a step size of 1.  
  
This code will also output the numbers from 0 to 9.

- range (0, 10, 1) generates a sequence of numbers starting from 0 up to (but not including) 10, with a step size of 1.

This code will also output the numbers from 0 to 9.

- In summary, range (10) and range (0, 10) generate the same sequence of numbers, while range (0, 10, 1) specifies an explicit step size of 1, which is the default step size if not specified.

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 i in range(1, 11):

print(i)  
  
And here is an equivalent program that prints the numbers 1 to 10 using a while loop:

i = 1

while i <= 10:

print(i)

i += 1  
  
Both of these programs will produce the same output, which is the numbers 1 to 10 printed on separate lines.

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

**Answer:**# Import the module

import spam

# Call the function

spam.bacon()