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

ANSWER:

The two values: True , False

They are written by capitalizing the first character True False

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

ANSWER:

and , or , 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:

True and True	True
True and False	False
False and True	False
False and False	False

True or True	True
True or False	True
False or True	True
False or False	False

not True	False
not False	True

4. What are the values of the following expressions?

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:

Equal to (==)

Not equal to (!= or <>)

Greater than (>)

Less than (<)

Greater than or equal to (>=)

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 and assignment operators in Python are distinguished by the number of equal signs used:

Assignment Operator (=): The assignment operator is a single equal sign (=). It is used to assign the value on the right side to the variable on the left side. For example, in the statement x = 5, = is the assignment operator that assigns the value 5 to the variable x.

Equal to Operator (==): The equal to operator is a double equal sign (==). It is used to compare two values to check if they are equal. If the values are equal, the result of the operation is True, otherwise, it's False. For example, in the condition if x == 5, == is the equal to operator that checks if the value of x is equal to 5.

You would use the assignment operator (=) when you want to set a variable to a certain value. This is often at the beginning of a script or function to initialize variables, or throughout a program to update or change the value of a variable.

However, You would use the equal to operator (==) when you want to compare two values. This is often in the condition of an if, elif, or while statement to control the flow of the program based on whether two values are equal.

7. Identify the three blocks in this code:

ANSWER:

```
spam = 0

if spam == 10:

        print('eggs')    # block 1

if spam > 5:

        print('bacon')  # block 2

else:

        print('ham')      # block 3

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 = 1  # this value to test different scenarios

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:

this is done by pressing Ctrl + C on the keyboard.

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

ANSWER:

break: is used to completely exit the current loop in which it is present. After a break statement is encountered, Python's interpreter immediately stops the loop execution and moves on to the next block of code after the loop. It's commonly used when some condition is met, and you want to stop looping entirely.

continue: On the other hand, the continue statement skips the rest of the current iteration and moves on to the next iteration of the loop. It's often used when you want to stop the current iteration prematurely, but you don't want to exit the loop completely.

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

ANSWER:

in all three cases, the resulting sequence of numbers is the same. The different forms just provide more flexibility in defining the sequence.

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

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:

```
import spam  # importing

spam.bacon() # calling
```