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. They are written as follows:**

* **true**
* **false**

**For example, the following code will print "The number is even" if the number is even, and "The number is odd" if the number is odd:**

**number = 10**

**if number % 2 == 0:**

**print("The number is even")**

**else:**

**print("The number is odd")**

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

**Answer - The three basic Boolean operators are AND, OR, and NOT.**

* **AND is used to combine two or more Boolean expressions and return true only if all of the expressions are true. For example, the expression (x > 0) AND (x < 10) is true only if x is a number greater than 0 and less than 10.**
* **OR is used to combine two or more Boolean expressions and return true if any of the expressions are true. For example, the expression (x > 0) OR (x < 10) is true if x is greater than 0 or less than 10.**
* **NOT is used to reverse the truth value of a Boolean expression. For example, the expression NOT (x > 0) is true if x is not greater than 0.**

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 - Here are the truth tables for each Boolean operator:**

1. **AND Operator (&&):**

| **Operand A** | **Operand B** | **Result** |
| --- | --- | --- |
| false | false | false |
| false | true | false |
| true | false | false |
| true | true | true |

1. **OR Operator (||):**

| **Operand A** | **Operand B** | **Result** |
| --- | --- | --- |
| false | false | false |
| false | true | true |
| true | false | true |
| true | true | true |

1. **NOT Operator (!):**

| **Operand** | **Result** |
| --- | --- |
| false | true |
| true | false |

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 - Here is the explanation for each expression:**

* **(5 > 4) and (3 == 5): The first expression, (5 > 4), is true because 5 is greater than 4. The second expression, (3 == 5), is false because 3 is not equal to 5. Therefore, the overall expression is false.**
* **not (5 > 4): The not operator reverses the truth value of its operand. In this case, the operand is (5 > 4), which is true. Therefore, the overall expression is false.**
* **(5 > 4) or (3 == 5): The first expression, (5 > 4), is true because 5 is greater than 4. The second expression, (3 == 5), is false because 3 is not equal to 5. Therefore, the overall expression is true.**
* **not ((5 > 4) or (3 == 5)): The not operator reverses the truth value of its operand. In this case, the operand is (5 > 4) or (3 == 5), which is true. Therefore, the overall expression is false.**
* **(True and True) and (True == False): The first expression, (True and True), is true because both operands are true. The second expression, (True == False), is false because True is not equal to False. Therefore, the overall expression is false.**
* **(not False) or (not True): The not operator reverses the truth value of its operand. In this case, the operand is False, which is false. Therefore, the overall expression is true.**

5. What are the six comparison operators?

**Answer –**

**The six comparison operators in Python are:**

* **Less than (<)**
* **Less than or equal to (<=)**
* **Greater than (>)**
* **Greater 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 return a Boolean value, either True or False. The assignment operator = is used to assign a value to a variable.**

**The equal to operator is used to check if two values are equal. For example, the expression x == y is true if x is equal to y.**

**The assignment operator is used to assign a value to a variable. For example, the expression x = y assigns the value of y to the variable x.**

**Here is an example of how to use the equal to operator to check if two numbers are equal:**

**x = 10**

**y = 10**

**if x == y:**

**print("x and y are equal")**

**This code will print "x and y are equal" because x and y are both equal to 10.**

**Here is an example of how to use the assignment operator to assign a value to a variable:**

**x = 10**

**y = x**

**print(y)**

**This code will print 10 because the value of x is assigned to the variable y.**

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 three blocks in the code are:**

1. **The first block is the assignment statement spam = 0.**
2. **The second block is the if statement if spam == 10:.**
3. **The third block is the else statement else:**

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("What is stored in spam? "))**

**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, I will press Ctrl+C to stop the program. This will interrupt the loop and return the program to the command line.**

**I can also use the break statement to stop the loop. The break statement will immediately exit the loop, regardless of the condition.**

**Finally, I can use the continue statement to skip the current iteration of the loop and continue with the next iteration.**

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

**Answer - "break" is used to terminate the loop entirely and continue with the next statement after the loop.**

**"continue" is used to skip the remaining code within the current iteration and move 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 - The range() function in Python returns a sequence of numbers, starting from a given start integer to a stop integer, which we can iterate using a for loop.**

**The difference between range(10), range(0, 10), and range(0, 10, 1) is the start and step values.**

**range(10) returns a sequence of numbers starting from 0 to 9.**

**range(0, 10) returns a sequence of numbers starting from 0 to 9.**

**range(0, 10, 1) returns a sequence of numbers starting from 0 to 9, incrementing by 1.**

**In all three cases, the loop iterates over a sequence of numbers from the start value (inclusive) up to the end value (exclusive), incrementing by the step size of 1 (which is the default if not explicitly specified). The main difference lies in the syntax and the way the start, end, and step values are specified in the range() function.**

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 loop**

**for i in range(1, 11):**

**print(i)**

**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 the module named spam which contains a function named bacon(), you can call the bacon() function using the dot notation. Here's how you can do it:**

**import spam**

**spam.bacon()**

**In this example, the spam module is imported using the import statement. Then, you can call the bacon() function by specifying the module name (spam) followed by the function name (bacon()) using the dot notation (spam.bacon()).**

**This way, you can access and call the bacon() function from the spam module in your code.**