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

The two values of the Boolean data type are True and False.

They are written as follows:

1. True
2. False

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

The three different types of Boolean operators are

- 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 ).**

- **AND**: This operator returns true only if both input values are true, otherwise it returns false. It is denoted by the symbol ∧ or &. The truth table for AND is:

| **P** | **Q** | **P &Q** |
|---|---|---|
| T | T | T |
| T | F | F |
| F | T | F |
| F | F | F |

- **OR**: This operator returns true if at least one of the input values is true, otherwise it returns false. It is denoted by the symbol ∨ or |. The truth table for OR is:

| **P** | **Q** | **P \|Q** |
|---|---|---|
| T | T | T |
| T | F | T |
| F | T | T |
| F | F | F |

- **NOT**: This operator returns the opposite value of the input value, i.e., it returns false if the input is true and vice versa. It is denoted by the symbol ¬ or ~. The truth table for NOT is:

| **P** | **~P** |
|---|---|
| T | F |
| F | T |

- **XOR**: This operator returns true if the input values are different, otherwise it returns false. It is also known as exclusive or or inequality. It is denoted by the symbol ⊕ or ^. The truth table for XOR is:

| **P** | **Q** | **P ^ Q** |
|---|---|---|
| T | T | F |
| T | F | T |
| F | T | T |

| P | Q | P ^ Q |
|---|---|---|
| F | F | F |

**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)**

The values of the following expressions are as follows:

- (5 > 4) and (3 == 5)  is False.
- not (5 > 4) is True.
- (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?**

The six comparison operators are:

- **==** (equal to)
- **!=** (not equal to)
- **>** (greater than)
- **>=** (greater than or equal to)
- **<** (less than)
- **<=** (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.**

The equal to operator **“==”** compares two values and returns True if they are equal, or False if they are not equal. The assignment operator **“=”** assigns a value to a variable. For example, the following code assigns the value 5 to the variable “x”:

- **x = 5**

The following code compares the values of the variables **“x”** and **“y”** and returns True if they are equal, or False if they are not equal:

- **x == y**

I would use the equal to operator when I want to compare two values and check if they are equal. I would use the assignment operator when I want to assign a value to a variable.

**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')**

The code you provided has three blocks:

- The first block is the `if` statement that checks if `spam` is equal to 10. If it is, the code prints `eggs`.
- The second block is the `if` statement that checks if `spam` is greater than 5. If it is, the code prints `bacon`. Otherwise, the code prints `ham`.
- The third block is the code that prints `spam` twice.

**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.**

**Program :**

```
spam = int(input("Enter a number: "))


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?**

If your program is stuck in an endless loop, you can press the **"Ctrl + C"** keys to stop the program. This will interrupt the program and return you to the command line.

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

The `break` statement terminates the current loop and skips the rest of the loop body. The `continue` statement skips the rest of the current iteration of the loop and continues with the next iteration. For example, the following code prints the numbers from 1 to 10:

- **Break Statement :**

```
for i in range(1, 11):
if i == 5:
   break
print(i)
```

The `break` statement in this code will terminate the loop when the value of `i` is equal to 5. The following code prints the numbers from 1 to 10, but skips the number 5:

- **Continue Statement** :

```
for i in range(1, 11):
    if i == 5:
        continue
    print(i)
```

The `continue` statement in this code will skip the iteration of the loop when the value of `i` is equal to 5. The rest of the loop body will be executed for the remaining values of `i`.

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

The `range()` function in Python is used to generate a sequence of numbers. The first argument to the `range()` function is the number of numbers to generate. The second argument is the starting number, and the third argument is the step size.

- The `range(10)` function will generate a sequence of numbers from 0 to 9.
- The `range(0, 10)` function will generate a sequence of numbers from 0 to 9, including 0 and 9.
- The `range(0, 10, 1)` function will generate a sequence of numbers from 0 to 9, including 0 and 9, with a step size of 1. Here is an example of how to use the `range()` function in a for loop:

- ```
  for i in range(10):
  print(i)
  ```

This code will print the numbers from 0 to 9.

- ```
  for i in range(0, 10):
      print(i)
  ```

This code will also print the numbers from 0 to 9, but it will include 0 and 9.

- for i in range(0, 10, 1):

  print(i)

This code will also print the numbers from 0 to 9, but it will include 0 and 9, and it will use a step 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.**

- **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?**

Once the module has been imported, you can call the `bacon()` function by using the `spam.bacon()` syntax. For example, the following code calls the `bacon()` function:

```
spam.bacon()
```

The `spam.bacon()` syntax tells Python to look in the `spam` module for the `bacon()` function. Once it finds the function, it will execute it.