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

**Ans**: True and False are two values of the Boolean data types. We have to use capital T and F and with the rest of the word in lowercase

**Input: -**

a**=True**

b**=False**

print(a,type(a))

print(b,type(b))

**Output: -**

True <class 'bool'>

False <class 'bool'>

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

**Ans**: The three different types of Boolean operators in python are: or and not

**Input: -**

a**=**100

b**=**200

print(a**>**50 **and** b**>**100) *# Example of boolean and*

print(a**>**200 **or** b**>**100) *# Example of boolean or*

print(**not**(a**>**10)) *# Example of boolean not*

**Output: -**

True

True

False

#### 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) ?

**Ans:** The Truth tables for the boolean tables are as follows:

* **Truth Table for and operator**  
  True and True is TrueTrue and False is FalseFalse and True is FalseFalse and False is False
* **Truth Table for or operator**  
  True and True is TrueTrue and False is TrueFalse and True is TrueFalse and False is False
* **Truth Table for not operator**  
  True not is False False not is 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)

**Input: -**

print((5**>**4)**and**(3**==**5)) *# False*

print(**not**(5**>**4)) *# False*

print((5**>**4)**or**(3**==**5)) *# True*

print(**not**((5**>**4)**or**(3**==**5))) *# False*

print((**True** **and** **True**)**and**(**True==False**)) *# False*

print((**not** **False**)**or**(**not** **True**)) *# True*

**Output: -**

False

False

True

False

False

True

**5. What are the six comparison operators?**

**Ans:** The Six comparison operators available in python are:  
== , != , < , > , <= , =>

**6. How do you tell the difference between the equal to and assignment operators? Describe a condition and when you would use one ?**

**Ans:** == is the equal to operator that compares two values and evaluates to a Boolean, while = is that assignment operator that stores a value in a variable.

**Input: -**

a**=**3 *# Assigning operator that stores 3 value in a variable a*

**if** a**==**3:*#comparing values of a variable value and 3*

print(a**==**3)

**Output: -**

True

**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')

**Ans**: In Python, code block refers to a collection of code that is in the same block or indent. This is most commonly found in classes, functions, and loops.

**Input: -**

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')

**Output: -**

ham

spam

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.**

**Input: -**

**def** spamCode(spam):

**if** spam**==**1:

print('Hello')

**elif** spam**==**2:

print('Howdy')

**else**:

print('Greetings')

spamCode(1)

spamCode(2)

spamCode(3)

**Output: -**

Hello

Howdy

Greetings

**9.If your programme is stuck in an endless loop, what keys you’ll press?**

**Ans:** Press Ctrl-c to stop a program stuck in an infinite loop

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

**Ans:** The break statement will move the execution outside the loop if break condtion is satisfied. Whereas the continue statement will move the execution to the start of the loop.

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

**Ans:** The Differences are as follows:

1. The ***range (10)*** call range from 0 to 9 (but not include 10)
2. The ***range (0,10)*** explicitly tells the loop to start at 0
3. The ***range (0,10,1)*** explicitly tells the loop to increase the variable by 1 on each iteration

**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 ?**

**Input: -**

print('Using For Loop')

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

print(i, end**=**" ")

print('\n')

print('Using While Loop')

i**=**1

**while** i**<=**10:

print (i, end**=**" ")

i**+=**1

**Output: -**

Using For Loop

1 2 3 4 5 6 7 8 9 10

Using While Loop

1 2 3 4 5 6 7 8 9 10

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

**Ans:** This function can be called with spam. bacon ()