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

Ans1: The 2 values of the Boolean data type are True and False. We can write them as – **True** and **False**.

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

Ans2: 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 evaluates).**

Ans3:   
 Considering True = 1 and False = 0

Following will be the truth table for: **and, or, not**

1. True (1) **and** True (1) = True (1)
2. True (1) **and** False (0) = False (0)
3. False (0) **and** True (1) = False (0)
4. False (0) **and** False (0) = False (0)
5. True (1) **or** True (1) = True (1)
6. True (1) **or** False (0) = True (1)
7. False (0) **or** True (1) = True (1)
8. False (0) **or** False (0) = False (0)
9. **not** True (1) = False (0)
10. **not** False (0) = True (1)

**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)**

Ans4:

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**

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

Ans5:

Six comparison operators are as follows:

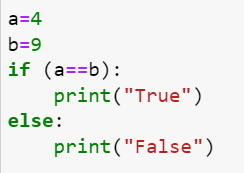
1. == (equals equals)
2. < (less than)
3. > (greater than)
4. <= (less than equals to)
5. >= (greater than equals to)
6. != (not equals to)

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

Ans6:

Equal to (==) operator compares the values and returns True or False.

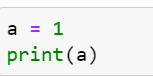
Eg:



Output- False

Whereas assignment operator (=) uses to store data in a variable.

Eg:



Output- 1

**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')**

Ans7:

spam = 0

if spam == 10:

**BLOCK 1**

print('eggs')

if spam > 5:

**BLOCK 2**

print('bacon')

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.**

Ans8:

spam = int(input("Enter number 1 or 2: "))

print(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?**

Ans9:

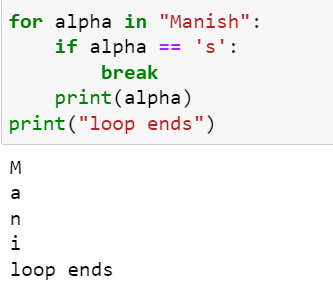
We can stop endless loop by pressing – CNTRL + C

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

Ans10:

**Break** statement will terminate the loop as soon as condition met, whereas **continue** statement will not terminate the loop and proceed further with next iteration.

Eg:



This will break the loop as soon as alpha equals to ‘s’ and comes out of the loop.

A screenshot of a computer code

Description automatically generated

This will skip the iteration when alpha equals to ‘s’ and continues to next iteration or comes back to starting of the loop.

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

Ans11:  
  
range(10) – It shows that the loop will end at 10. [range(stop)] (\*prints number 0 to 9)

range(0,10) – It shows that the loop will start from 0 and stops at 10. [range(start, stop)] (\*prints number 0 to 9)

range(0,10,1) – It shows that the loop will start from 0 and stops at 10 with increment value of 1. [range(start,stop,step)] (\*prints number 0 to 9)

**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.**

Ans12:

Program to print numbers 1 to 10 using ‘for’ loop:

for i in range(1, 11):

print(i)

Program to print numbers 1 to 10 using ‘while’ loop:

i=1

while (i <= 10):

print(i)

i = i+1

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

Ans13: If a function bacon() is inside a module named spam then after importing spam, we can call it by using following syntax:

spam.bacon()