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

Ans: The two values of Boolean data type are **True and False. And they are written as**

**False** and **True.**

**As python is case sensitive so true and false written in this way will give error**

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

**a. OR**

**b. AND**

**c. 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).

a. **OR Truth Table**

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **A or B** |
| True | False | True |
| False | True | True |
| True | True | True |
| False | False | False |

b. **AND Truth Table**

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **A and B** |
| True | True | True |
| False | True | False |
| True | False | False |
| False | False | False |

c. **NOT Truth Table**

|  |  |
| --- | --- |
| **A** | **not A** |
| True | False |
| False | True |

**4. What are the values of the following expressions?**

(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?**

|  |  |  |
| --- | --- | --- |
| **Sr No** | **Operator** | **Name** |
| 1 | **==** | Equal |
| 2 | **!=** | Not equal |
| 3 | **>** | Greater than |
| 4 | **<** | Less than |
| 5 | **>=** | Greater than or equal to |
| 6 | **<=** | 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.**

**Ans:**

Equal to operator: ==

Assignment Operator: =

1. Equal to (==) operator is used when we need to compare two things.
2. Assignment (=) operator is used to assign a value to variable

Example: price = 500

**If price == 400:**

**print(“ I can buy a pair of pant”)**

Here value 500 is assigned to variable price, but == operator is used to compare the value 400 to the current value of price variable.

**7. Identify the three blocks in this code:**

**Block 1**

spam = 0

if spam == 10:

print('eggs')

**block 2**

if spam > 5:

print('bacon')

**block 3**

else:

print(‘ham’)

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.**

spam = 3

if spam == 1:

print("Hello")

elif spam == 2:

print("Howdy")

else:

print("Greetings!")

**output:**

spam = 1

Hello

spam = 2

Howdy

spam = 3456

Greetings!

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

In PyCharm to end an infinite loop: press **Ctrl + F2**

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

**Ans:**

The **main difference** between **break keyword** and **continue statement** is break terminates the loop at a specified condition (when this condition is True) and moves on to next condition, whereas **continue keyword** skips a particular iteration (if condition associated with continue is True) and continues the loop from next iteration.

This is explained by using below examples:

**break keyword** is a loop control statement. break helps user to terminate a loop for a particular condition, and skip to next condition. It can be used with for and while loop both.

For example:

s="shubham"

for i in s:

print(i)

if i == "b":

break

else:

print("print this if loop completes")

in the above example the loops prints s, h, u, b, but when i == “b” condition gets True the loop gets terminated as break statement is executed.

Suppose if i == “z” then the if condition for string s will never be true and the output of the loop will be as follows:

s

h

u

b

h

a

m

print this if loop completes

**continue keyword** is a loop control statement which forces to execute next iteration while skipping the current iteration.

For example

s="shubham"

for i in s:

if i == "b":

continue

print(i)

else:

print("print this if loop completes")

output:

s

h

u

h

a

m

print this if loop completes

in the above example the loop skips the iteration where i == “b” is skipped, but the loop is continued to execute the rest iterations.

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

**Ans:** range(a , b , c)

here, a is start value, if a is blank then default value 1

b is end value, and is not included.

c is no of steps, by default it is 1 unless specified anything else.

**So range(10) , range(0, 10) and range(0,10,1) are same**

**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.**

**1) using for loop**

for i in range(1,11):

print(i)

**2) using while loop**

i=1

while i<11:

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?

**import spam**

**spam.bacon()**