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

Boolean data types take up the values as ‘True’ and ‘False’.

a=10

b=2

p=a>b

if p==True:

print("a greater than b")

elif p==False:

print("b greater than a")

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

They are AND, OR and 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 ).**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | A | B | A AND B | |  |
|  | TRUE | FALSE | FALSE |  | |
|  | FALSE | TRUE | FALSE |  | |
|  | TRUE | TRUE | TRUE |  | |
|  | FALSE | FALSE | FALSE |  | |
|  |  |  |  |  | |
| 2 | A | B | A OR B |  | |
|  | TRUE | FALSE | TRUE |  | |
|  | FALSE | TRUE | TRUE |  | |
|  | TRUE | TRUE | TRUE |  | |
|  | FALSE | FALSE | FALSE |  | |
|  |  |  |  |  | |
| 3 | 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?**

> : greater than, i.e a> b, a is greater than b

< : lesser than, ie a<b , a is lesser than b

==: equal to, ie a==b, a and b are equal

!= : not equal to, ie a!=b, a not equal to b

>=: greater than equal to, a>=b, a is greater or equal to b

<=: lesser than equal to, a<=b, a is lesser or equal to b

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

Equal to is a comparison operator, it compares a value to other value, on successful execution of the expression it precedes further for next logic or execution. Ex: a==b, a>=b

Whereas an assignment operator assigns a value to variables. Ex---x=5, x+=3 or x=x+3, x>>=3 or x=x>>3

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

spam = 0

if spam == 10: #Block 01

print('eggs')

if spam > 5: #Block 02

print('bacon')

else: #Block 03

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 = 1

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?**

CTRL+C

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

The main difference between break and continue is that the break statement alters the flow of a loop by terminating it once a specified condition is met. On the other hand, the continue statement begins the next iteration of the while, enclosing for, or do loop.

Example:

**Break::**

for num in range(0,10):

if num == 5:

break

print(num)

Output :

0

1

2

3

4

**Continue::**

for num in range(0,10):

if num == 5:

continue

print(num)

Output:

0

1

2

3

4

6

7

8

9

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

range(10): In for loop using range(10) will provide us the values starting from 1st position to n-1 position ie 0 to 9. Range takes default value of starting as 0.

Range(0,10): We here defining the range of loop. The output will be 0 to 9. As we provide the initial position is 0 and the last position value will be 10-1=9.

Range(0,10,1): Here we are providing the loop with an incremental sequence. The loop will add 1 with the next value. If we provide any other value like 2 or 3 it will provide a sequence with that incremented value. Output: 0 1 2 3 4 5 6 7 8 9 (0 0+1 1+1 2+1 3+1 4+1 5+1 6+1 7+1 8+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.**

While loop:

i=1

while i<11:

print(i)

if i == 10:

break

i+=1

For loop:

for i in range(1,11):

print(i)

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

from spam import bacon