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

Ans:-

* True and False are the two values of the Boolean data type.
* T and F are in uppercase and the remaining words are in lowercase.
* E.g. True, False

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

Ans:- Three different Boolean operators are: AND OR & NOT.

* AND operator:- The Boolean AND operator returns False if any one of the inputs is False else return True.
* OR operator:- The Boolean OR operator returns True if any one of the inputs is True else return False.
* NOT operator:- The Boolean NOT operator requires only one argument and returns the negation of the argument i.e returns True for False and False for True.

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:-

* Boolean AND operator

| A | B | A AND B |
| --- | --- | --- |
| TRUE | TRUE | TRUE |
| TRUE | FALSE | FALSE |
| FALSE | TRUE | FALSE |
| FALSE | FALSE | FALSE |

* Boolean OR operator

| A | B | A OR B |
| --- | --- | --- |
| TRUE | TRUE | TRUE |
| TRUE | FALSE | TRUE |
| FALSE | TRUE | TRUE |
| FALSE | FALSE | FALSE |

* Boolean NOT operator

| A | NOT A |
| --- | --- |
| TRUE | FALSE |
| FALSE | 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)

Ans:-

(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?

Ans:- The six comparison operators are as follows:

1. Greater than (>)
   1. Returns true if the left operand is greater than the right operand.
   2. E.g:-

5 > 2

Output: - True

1. Less than(<)
   1. Returns true if the left operand is less than the right operand.
   2. E.g.:-

33<23

Output:- False

1. Greater than equal to (>=)
   1. Returns true if the left operand is greater than or equal to the right operand.
   2. E.g.:-

5 >= 5

Output:- True

1. Less than equal to (<=)
   1. Returns true if the left operand is less than or equal to the right operand.
   2. E.g.:-

33<=23

Output:- True

1. Equal to(==)
   1. Returns true if the left operand is equal to the right operand.
   2. E.g.:-

5 == 5

Output:- True

1. Not equal to (!=)
   1. Returns true if the left operand is not equal to the right operand.
   2. E.g.:-

3 != 6

Output:- True

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 is used to compare two values and the assignment operator is used to assign the value to the variable.

#Assignment operator

a = 3 # 3 value is assigned to the variable name a

print(f'Value assigned to a :- {a}')

#Equal to operator

if (a==3): # comparing values of variable a with 3

print("A's value is equal to 3")

else:

print("A's value is not equal to 3")

Output:-

Value assigned to a :- 3

A's value is equal to 3

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:-

spam = 0

if spam == 10: # block 1

print('eggs')

if spam > 5: # block 2

print('bacon')

else: # block 3

print('ham')

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.

Program:-

spam = int(input("Give any number: "))

if spam == 1:

print("Hello")

elif spam == 2:

print("Howdy")

else:

print("Greetings!")

Output:-

Give any number: 1

Hello

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

Ans- If programme is stuck in an endless loop, I’ll press Ctrl+C to stop endless loop.

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

Ans:- the difference between break and continue is break will exit a loop immediately and continue will skip the current iteration of a loop, i.e. following code in the loop will not be executed (but continue the next iteration of loop).

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

Ans:- There is no difference between above range() with different parameters because all are doing the same thing printing from 0-10 as it has range() which has 3 parameters as follows - start, stop, and step respectively.

start(optional) - start value of the sequence

stop- end value of the sequence

step(optional)- the difference between any two numbers in the sequence

for i in range(10):

print(i)

for i in range(0,10):

print(i)

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

print(i)

For above 3 codes output is as shown below:-

0

1

2

3

4

5

6

7

8

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.

Program to print the numbers 1 to 10 using a for loop:

for i in range(1,11):

print(i)

Program to print the numbers 1 to 10 using a while loop

i=1

while i<=10:

print(i)

i+=1

Output for both “for” and “while” loop is :

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 spam.bacon().

import spam

spam.bacon()