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

The two values of the Boolean data type are True and False. An example of how we can write them is:

a = True

type(a)

b = False

type(b)

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

Common boolean operations are – or, and, not, == (equivalent), != (not equivalent).

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 ).

**Boolean OR Operator:**

The Boolean or operator returns True if any one of the inputs is True else returns False.

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

**Boolean And Operator:**

The Boolean and operator returns False if any one of the inputs is False else returns True.

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

**Boolean Not Operator:**

The Boolean Not operator only require one argument and returns the negation of the argument i.e. returns the True for False and False for True.

|  |  |
| --- | --- |
| 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?

The six comparison operators are-- less than ( < ), less than or equal to ( <= ), greater than ( > ), greater than or equal to ( >= ), equal to ( == ), and not 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.

= is a assignment operator and == is a comparison/equal to operator.

= operator is used to assign value to a variable and == operator is used to compare two variable or constants. The left side of = operator can not be a constant, while for == operator both sides can be constant.

|  |  |
| --- | --- |
| = | == |
| It is an assignment operator. | It is a relational or equal to or comparison operator. |
| It is used for assigning the value to a variable. | It is used for comparing two values. It returns 1 if both the values are equal otherwise returns 0. |
| Constant term cannot be placed on left hand side. Example: 1=x; is invalid. | Constant term can be placed in the left hand side. Example: 1==1 is valid and returns 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')

Indentation is used to define a block of code in python.

**Code:**

spam = 0

if spam == 10:

print('eggs')

if spam > 5:

print('bacon')

else:

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.

**Code:**

spam = 1

if spam== 1:

print('Hello')

if spam== 2:

print('Howdy')

if spam != 1 and spam !=2:

print('Greetings!')

**Output:**

Hello

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?

A break statement in Python alters the flow of a loop by terminating it once a specified condition is met.

The continue statement in Python is used to skip the remaining code inside a loop for the current iteration only.

**Code:**

for num in range(0,10):

if num == 5:

break

print(f'Iteration: {num}')

**Output:**

Iteration: 0

Iteration: 1

Iteration: 2

Iteration: 3

Iteration: 4

Here, no output is generated after Iteration: 4. We’ve added an if statement to check a condition. When this condition becomes True, the program flow will go inside the if statement and execute the break statement.

Therefore, the for loop executed until the condition num == 5 becomes True. When the condition becomes True, the break statement is executed to terminate the flow of the for loop.

**Code:**

for num in range(0,3):

if num == 2:

continue

print(f'Iteration: {num}')

**Output:**

Iteration: 0

Iteration: 1

Iteration: 3

Here, Iteration:2 is not printed. When the condition num == 2 becomes True, the continue statement gets executed. The remaining code in the loop is skipped only for that iteration.

Therefore, the continue statement works opposite to the break statement. Instead of terminating the loop, it forces it to execute the next iteration of the loop.

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

In a for loop, there is no difference in output for range(10), range(0, 10), and range(0, 10, 1). Here, we are just specifying the arguments for the range() function as follows:

range(end), range(start, end), range(start, end, step)

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.

**Code using for loop:**

for i in range(1, 11):

print(i)

**Code using while loop:**

i = 1

while(i<=10):

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?

spam.bacon()