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

Answer:

The two values of the Boolean data type are true and false. Here is how we can write them:

-> True for true Boolean data type where T should be in capital letter

-> False for false Boolean data type where F should be in capital letter

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

Answer:

Following are the three different types of Boolean operators:

-> The AND operator (&&; and)

-> The OR operator (||; or)

-> The NOT operator (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 ).

Answer:

Boolean operator’s truth table:

|  |  |  |
| --- | --- | --- |
| -> True and True: True  -> True and False: False  -> False and True: False  -> False and False: False | -> True or True: True  -> True or False: True  -> False or True: True  -> False or False: False | -> not True: False  -> not False: True |

Let’s say, True represented by 1 and False represented by 0:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| AND Operator | | | OR Operator | | | NOT Operator | |
| Input-1 | Input-2 | Output | Input-1 | Input-2 | Output | Input | Output |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 |  |  |
| 0 | 0 | 0 | 0 | 0 | 0 |  |  |

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)

Answer:

-> (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?

Answer:

Following are the six comparison operators:

-> equal to (==)

-> not equal to (!=)

-> greater than (>)

-> less than (<)

-> greater than equal to (>=)

-> less than 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.

Answer:

-> equal to (==) is a comparison operator which compares two values and evaluates to a Boolean value. In the other hand, equal (=) is an assignment operator which stores a value in a variable.

Example:

Let’s say, A = 20, B = 20, and C = 30

Here, the variable A and B was assigned to value 20 and the variable C was assigned to 30.

To check that the variable A and B has same value or not and B and C has different value or not, we can apply equal to operator. The equal to operator will return True Boolean value for first case and will return False for second case.

A == B: True

B == C: False

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')

Answer:

|  |  |  |
| --- | --- | --- |
| Block 1 | Block 2 | Block 3 |
| print('eggs') | print('bacon') | 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.

Answer:

|  |
| --- |
| spam = input("input a number: ")  if spam.isdigit() == True:  if int(spam) == 1:  print(“Hello”)  elif int(spam) == 2:  print(“Howdy”)  else:  print(“Greetings!”)  else:  print(“Greetings!”) |

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

Answer:

We will press ctrl+C to end or stop the infinite loop.

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

Answer:

|  |  |
| --- | --- |
| Break keyword | Continue keyword |
| The break keyword terminates the rest of remaining iterations of the loop. | The continue keyword terminates only the current iteration of the loop. |
| 'break' stops the continuation of loop. | 'continue' do not stops the continuation of loop, it only stops the current iteration. |
| Example:  for i in range(100):  print(i)  if i == 10:  break  Here, for loop will be ended after the iterator of 10. | Example:  for i in range(10):  print(i)  If i == 5:  continue  Here, for loop will continue until the last iterator. |

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

Answer:

-> range(10): this is default declaration of range() function that create sequence of numbers starting from 0 up to 10.

-> range(0, 10): this is a declaration of range() function where sequence start number is 0 and end number is 10.

-> range(0, 10, 1): this is also a declaration of range() function where sequence start number is 0, end number is 10, and jump or sequence incremental step number is 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.

Answer:

|  |  |
| --- | --- |
| prints the numbers 1 to 10 using a for loop | prints the numbers 1 to 10 using a while loop |
| for i in range(1, 11):  print(i) | 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?

Answer:

The bacon() function can be called in this way:

import spam

# call bacon() function

spam.bacon()