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

Solution1: True and False are two values of Boolean data type. The commands:- type(True) and type(False) will give bool as output.

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

Solution2: Three different types of Boolean operators are**:- AND , OR, 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 ).

Slution3:

|  |  |  |  |
| --- | --- | --- | --- |
| values | values | AND operator output | OR operator output |
| True | False | False | True |
| True | True | True | True |
| False | False | False | False |
| False | True | False | True |

|  |  |  |
| --- | --- | --- |
| operator | values | output |
| NOT | True | False |
| NOT | 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)

Solution4:

(5 > 4) and (3 == 5) will give **False**

not (5 > 4) will give **False**

(5 > 4) or (3 == 5) will give **True**

not ((5 > 4) or (3 == 5)) will give **False**

(True and True) and (True == False) will give **False**

5. What are the six comparison operators?

Solution5: A comparison operator compares two values and returns a Boolean value, either True or False. Python has six comparison operators: 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.

Solution6: In assignment operators single sign of equal ( = ) is used whereas in “equal to” , a pair of equal signs ( == ) is used.

Condition: We use “equal to” ( == ) whenever we need to check for a condition. Let us consider a list which consists numbers and string. We need to get a list of squares of the numbers in the list. Then we will use “equal to” ( == ) to check whether an element in the list is a number or not then we square it and get a list of squares.

Code:-

lst=[1,2,3,7,8,"kumar",(1,2,3),{1:34,2:25}]

sq=[]

for i in lst:

if type(i)== int:

sq.append(i\*\*2)

print(sq)

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')

solution7:

block1:

spam = 0

if spam == 10:

print('eggs')

BLOCK2:

if spam > 5:

print('bacon')

else:

print('ham')

BLOCK3:

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.

Solution8:

spam=5

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?

Solution9: “ctrl+c”

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

Solution10: The Python break statement stops the loop in which the statement is placed. A Python continue statement skips a single iteration in a loop. Both break and continue statements can be used in a for or a while loop.The continue statement allows us to skip part of a loop when a condition is met.

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

Solution11: all are same. Syntx: range(start point, endpoint+1, jump)

Hence by default jump is always 1. If starting point is not given then it starts with 0 itself.

Hece all three 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.

Solution12:

For loop code:

for i in range(1,11):

print(i)

while loop code:-

i=0

while i<10:

i=i+1

print(i)

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

Solution13:spam.bacon()