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

Answer-1:The two values of Boolean data type are True and False.We write them with their first letter capital.Otherwise,Python will throw an error.

Question-2: What are the three different types of Boolean operators?

Answer-2:The three different types of Boolean operators 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 ).

Answer-3:As we know that True is equivalent to 1 and False is equivalent to 0 in Python.

Truth Table of **and** operator

**and** operator will give True only if both the operands on either side of it are True otherwise False.

|  |  |  |
| --- | --- | --- |
| 1 | 0 | 0 |
| 0 | 0 | 0 |
| 1 | 1 | 1 |
| 0 | 1 | 0 |

Truth Table of **or** operator

or operator will give True if one of the operand on its either side is logically True or both of them are True,otherwise Talse.

|  |  |  |
| --- | --- | --- |
| 0 | 1 | 1 |
| 1 | 1 | 1 |
| 1 | 0 | 1 |
| 0 | 0 | 0 |

Truth Table of **not** operator

This operator reverses the logical state of any operand like not True is nothing but False.

|  |  |
| --- | --- |
| 1 | 0 |
| 0 | 1 |

Question-4: What are the values of the following expressions?

1. (5 > 4) and (3 == 5)
2. not (5 > 4)
3. (5 > 4) or (3 == 5)
4. not ((5 > 4) or (3 == 5))
5. (True and True) and (True == False)
6. (not False) or (not True)

Answer 4-(i) (5>4) is logically True.It will evaluate to be True.But (3==5) is logically False so it will evaluate to be False.Moreover,in the context of the logical operators,True and False will evaluate to be False.Hence,the value of the expression (5 > 4) and (3 == 5) is False.

4-(ii) The expression (5>4) is logically correct.It will evaluate to be True.But not operator reverses

Its logical state.Hence not (5 > 4) will be False.

4-(iii) The expression (5>4) is logically True while (3==5) is logically False.And when we perform

Or operation between them,it will evaluate to be True.

4-(iv) The expression (5>4) is logically True while (3==5) is logically False.And when we perform

Or operation between them,it will evaluate to be True,that is,((5>4) or (3==5)) is True.And not

operator reverses its logical state,so the final answer will be False.

4-(v) Since,in True and True,both the operands are True.Also True cannot be equal to

False,hence,(True==False) is False.Finally,True and False will surely give False.

4-(vi) not False is True and not True is False.Moreover,True or False is True.

Question-5: What are the six comparison operators?

Answer-5:The six comparision operators are as follows.

* Less than ( < )
* Less than or equal to ( <= )
* Greater than ( > )
* Greater than or equal to ( >= )
* Equal to ( == )
* Not equal to ( != )

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

Answe-6:When we use single “=” sign,it will be considered as assignment operator in Python.When we use “==” operator it will be considered as an equalty operator in Python.

If we want to assign the value 6 to a variable a,then we will use “=” sign once,i.e.,a=6.

If we want to check whether two objects,suppose two numbers 7 and 9,are equal or not,then we will use “=” twice,e.g.,7==9

Question-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-7:The first if and its corresponding statements after it is the first block,the second if and the statements associated with it is the second block and finally else and the statements associated with it is the third block.

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=input(“Enter either 1 or 2”)

If spam==”1”:

Print(“Hello”)

elif:

print(“Howdy”)

else:

print(“Greetings!”)

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

Answer-9:If we stuck in an endless loop,we can press ctrl+C to exit from it.

Question-10:How can you tell the difference between break and continue?

Answer-10:The break statement,when encountered,gets the control out of a loop in which it is used and the statements just after the loop start to execute while continue statement skips the current iteration and goes for the next one.

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

Answer-11:The range() is a built in function Syntax of the range() function is range([start],stop,[step]).It is used to create a list containing a sequence of numbers from the given start value upto stop value(excluding the stop value),with a difference of given step value.

The start and stop parameters are optional.If start value is not specified,by default the list starts from 0.If step value is not specified,by default the value increases by one in each iteration.

range(10) will generate a list of numbers containing number from 0 to 9 with a step value of 1.

range(0,10) range(0,10,1) will give the same output as range(10).

Question-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-12:

#Programme to find print the numbers from 1 to 10 using a for loop.

For a in range(11):

print(a)

#Programme to print the numbers from 1 to 10 using while loop.

number=1

while number<=10:

print(number)

number+=1

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

Answer-13:The function can be called by the following statement.

spam.bacon()