

1. What are the Boolean data type's two values? How do you go about writing them?

Answer : Two values of Boolean Datatypes are 'True' and 'False'. 'True' value represents value as 1,i.e. when used in an expression/condition it outputs 1 if condition/expression is satisfied. 'False' value represents value as 0,i.e. when used in an expression/condition it outputs 0 if condition/expression is satisfied.

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

Answer : Types of Boolean Operator are :

1. 'and' operator :
'and' operator will return True , if both the operands are True. Example, for expression " A and B" result will be True if A and B both are True at the same time.
2. 'or' operator :
'or' operator will return True , if any one of the operands or both operands are True. Example, for expression " A or B" result will be True if A = True or B=True or both are True at the same time.
3. 'not' operator :
'not' operator takes only one operand and output its opposite result. Example, if A is True, then 'not A' will result False and vice versa.

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 evaluates to).

Answer :

Truth Table for 'and' operator :

A	B	A and B
True	True	True
True	False	False
False	True	False
False	False	False

Truth Table for 'or' operator:

A	B	A or B
True	True	True
True	False	True
False	True	True
False	False	False

Truth Table for '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)

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 different types of reference operators?

Answer :

1. equal to (==) :
Returns result if two operands are equal or not. Example, 8==8 will result True.
2. not equal to (!=):
Returns result if two operands are not equal. Example, 8!=8 will result False.
3. greater than (>):
Returns True if first operand is greater than second operand. Example, 6>5 will result True.
4. greater than equal to (>=):
Checks if first operand is greater or equal to second operand. Example, 6>=5 will result True.
5. less than(<):
Returns True if first operand is less than second operand. Example, 6<5 will result False.
6. less than equal to (<=):
Checks if first operand is less or equal to second operand. Example, 6<=5 will result False.

6. How do you tell the difference between the equal to and assignment operators?

Answer : '=' sign when used once is an assignment operator, it assigns a value to a variable thus allocating it to a memory location. Ex, a=96

'==' sign when used twice is a comparison operator called 'equal to', it checks if two operands are equal results a Boolean output. Ex, 8==8 results in True output.

7. Describe a condition and when you would use one.

Answer : In the following code variable 'a' is assigned value integer 50, and is compared using comparison operator if value stored in variable is 50, output of if condition will be True and print statement in it will be printed.

Code :

```
a = 50 #assignment operator
if a==50: #comparison operator.
    print("Value of a is 50.")
```

Output :

Value of a is 50.

8. Recognize the following 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 :

As value stored in spam=0, so first if statement itself will not satisfy so, execution of statements inside if condition will not take place. Result in execution of final print statement. So output of code will be : spam.

9. Create a programme that prints. If 1 is stored in spam, prints Hello; if 2 is stored in spam, prints Howdy; and if 3 is stored in spam, prints Salutations! if there's something else in spam.

Answer :

```
spam = int(input("Input your choice from 1,2 or 3"))
```

```
if spam==1:
```

```
    print("Hello")
```

```
elif spam==2 :
```

```
    print("Howdy")
```

```
elif spam==3 :
```

```
    print("Good Morning")
```

```
else :
```

```
    print(" Not an input i expected")
```

10.If your programme is stuck in an endless loop, what keys can you press?

Answer : CTRL+C must be pressed.

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

Answer : break and continue statement controls the flow of execution in a loop.

When continue statement is used in a loop, execution for a particular iteration stops and control is shifted to next iteration within a loop itself.

Whereas break statement terminates looping if a particular condition before it is satisfied , causing execution to come out of loop and resume code after loop.

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

Answer : range() function in python has three parameters start, stop and step.

i.e range(start, stop, step).

If start and step values are not defined while using range() function, then default values will be 0 and 1 respectively.

So, range(10), range(0, 10), and range(0, 10, 1) are same and will produce same output i.e. 0,1,2,3,4,5,6,7,8,9 .

13. Using a for loop, write a short programme that prints the numbers 1 to 10 Then, using a while loop, create an identical programme that prints the numbers 1 to 10.

Answer :

Using a for loop, write a short programme that prints the numbers 1 to 10 :

```
for i in range(10):  
    print(i)
```

Using a for while, write a short programme that prints the numbers 1 to 10 :

```
a= 10  
while(a<10):  
    print(a)  
    a+=1
```

14. If you had a bacon() function within a spam module, how would you call it after importing spam?

Answer :

```
import spam #importing spam  
spam.bacon() #calling bacon() function
```