

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

The two values of Boolean data type are: *True* & *False*

It returns *True* when condition is satisfied and returns *False* if condition is not satisfied.

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

The three different types of Boolean Operators are:

1. and – If all conditions are true
2. or – If either or both conditions are true
3. not – Inverse of True or False

## 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).

Truth Table - 'and' operator		
Input A	Input B	Output
False	False	False
False	True	False
True	False	False
True	True	True

Truth Table - 'or' operator		
Input A	Input B	Output
False	False	False
False	True	True
True	False	True
True	True	True

Truth Table - 'not' operator	
Input	Output
False	True
True	False

## 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)*

```
In [2]: (5 > 4) and (3 == 5)
```

```
Out[2]: False
```

```
In [3]: not (5 > 4)
```

```
Out[3]: False
```

```
In [4]: (5 > 4) or (3 == 5)
```

```
Out[4]: True
```

```
In [5]: not ((5 > 4) or (3 == 5))
```

```
Out[5]: False
```

```
In [6]: (True and True) and (True == False)
```

```
Out[6]: False
```

```
In [7]: (not False) or (not True)
```

```
Out[7]: True
```

## 5. What are the six comparison operators?

The six comparison operators are:

1. ==
2. !=
3. >
4. <
5. <=
6. >=

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

Double Equal to (==) operator is used for checking the validity of a condition and the two possible outcome can be True & False (Boolean values).

```
In [8]: a=17
        b=34
        print(a>b)

False
```

Assignment (=) operator is used to assign value to the variable.

```
In [11]: a = 10
        print(a)
        a = 23
        print(a)

10
23
```

**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')
```

```
In [13]: spam = 0

# Block 1
if spam == 10:
    print('eggs')

# Block 2
if spam > 5:
    print('bacon')
else:
    print('ham')

# Block 3
print('spam')
print('spam')

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.**

```
In [18]: spam = input("Input a value for Spam: ")
        if spam == 1:
            print("Hello")
        elif spam == 2:
            print("Howdy")
        else:
            print("Greetings!")

Input a value for Spam: 3
Greetings!
```

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

If program is stuck in endless loop we will use *Ctrl + c*.

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

A break statement is used to come out of the loop without completion; whereas a continue statement is used to skip the current operation and continue again in the loop with next operation.

```
In [6]: for i in range(5):
        if(i==3):
            print("Break now")
            break
        print(i)
```

```
0
1
2
Break now
```

```
In [9]: for i in range(8):
        if(i==3):
            print("Skip & Continue")
            continue
        print(i)
```

```
0
1
2
Skip & Continue
4
5
6
7
```

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

In the range(10), the start value is not mentioned and hence start value will be taken as zero by default. However, end value is mentioned as 10. So it will continue from 0 to 9 i.e. excluding the end or upper value.

In the range(0, 10), the start value is mentioned as zero and end value is mentioned as 10. So it will continue from 0 to 9 i.e. excluding the end or upper value.

In the range(0, 10, 1), the start value is mentioned as zero and end value is mentioned as 10. The third element i.e. 1 is the no of steps it will move each time. Since, it is 1 it is moving to next value. However, if it is 2 then it will move 2 steps i.e. from 0 it will move to 2, 4, 6 and so on.

```
In [23]: genOutput = []
        for i in range(10):
            genOutput.append(i)
            print(genOutput)
            print("End of First Block")

        genOutput = []
        for i in range(0, 10):
            genOutput.append(i)
            print(genOutput)
            print("End of Second Block")

        genOutput = []
        for i in range(0,10,2):
            genOutput.append(i)
            print(genOutput)
            print("End of Last Block")
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
End of First Block
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
End of Second Block
[0, 2, 4, 6, 8]
End of Last Block
```

```
In [24]: genOutput = []
         for i in range(0,10,1):
             genOutput.append(i)
         print(genOutput)

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

**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.**

```
In [25]: for i in range(1,11): print(i)

1
2
3
4
5
6
7
8
9
10
```

```
In [3]: i =1
        while i <= 10:
            print(i)
            i+=1

1
2
3
4
5
6
7
8
9
10
```

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

Module spam in spam.py:

```
def bacon():
    print('Hi')
```

It can be called in other Module by importing spam module:

```
import spam
spam.bacon()
```