

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

**Solution 1:** The two values of boolean data type: True and False where both starts with upper case and the rest in lower case.

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

**Solution 2:** 3 types of boolean operators: **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 ).

**Solution 3: for and:**

- True and True = True
- True and False = False
- False and False = False
- False and True = False

**For or:**

- True or True = True
- True or False = True
- False or False = False
- False or True = True

**Not:**

- not true = False
- not False = True

4. What are the values of the following expressions?

(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?

**Solution 5:** The 6 comparison operators: ==, !=, <, >, <=, and >=.

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

**Solution 6:** assignment operator is represented by '='. It is used to assign a value to a variable e.g. x = 6 here, x holds the value 6. The equal operator is represented by '==' and is used to compare 2 values and then evaluating the result to a boolean.

E.g. if x == 5:

```
print("FIVE")
```

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')

```

**Solution 7:** 3 Blocks are inside the 2 if and one else statement:

**1<sup>st</sup> block-**

```

if spam == 10:
    print('eggs')

```

**2<sup>nd</sup> block-**

```

if spam > 5:
    print('bacon')

```

**3<sup>rd</sup> block-**

```

else:
    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.

**Solution 8:**

```

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?

**Solution 9:** To come out of an endless loop:

- we can interrupt kernel in Jupyter notebook by clicking square like button adjacent to run icon.
- We can also press **CTRL+C** to stop the program.

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

**Solution 10:** break and continue are used to skip the iteration of a loop. The break will terminate the loop (in which 'break' is inside) and will execute the next statement which is after the body of the

loop. However, the continue statement is used to skip the rest of the code inside a loop for only the current iteration. i.e. the loop would not terminate and continue with the next iteration.

Continue e.g.:

```
for x in "string":  
    if x == "i":  
        continue  
    print(x)  
print("Bye")
```

o/p:

s  
t  
r  
i  
n  
g  
Bye

Break example:

```
for x in "string":  
    if x == "i":  
        break  
    print(x)  
print("Bye")
```

o/p:

s  
t  
r  
Bye

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

**Solution 11:** The functioning for each loop is same.

- range(10): calls ranges starting from 0 to 10(except 10)
- range(0,10): indicates that the loop will start from 0 and will go on till 10(except).
- range(0,10,1): indicates that the loop will increase by 1 in each iteration.

The output generated in each case is same and will be displayed as:

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.

**Solution 12: Using for loop:**

```
for i in range(1,11):  
    print(i)
```

**Using while loop:**

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

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

**Solution 13: import spam**

```
x = spam.bacon()
```