

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

2 values are True and False for boolean data type and are written as:

`v = 10`

`bool(v) -> output is True`

`a = 0`

`bool(a) -> output is False`

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

They 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 evaluates to).

Truth Table for AND, OR, NOT are as below.

Input X	Input Y	X AND Y	X OR Y	NOT X	NOT Y
0	0	0	0	1	1
0	1	0	1	1	0
1	0	0	1	0	1
1	1	1	1	0	0

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 different types of reference operators?

These are as follows:

Arithmetic operators: + (addition), *(multiplication), /(division), %(modulus) , - (subtraction)

Assignment operators : =(equals to), +=(x+= 10 means x = x+10), -=(x-= 10 means x = x-10),
=(x= 10 means x = x*10), /=(x/= 10 means x = x/10)

Comparison operators: >=, <=, !=, ==

Logical operators: AND, OR ,NOT

Identity operators: is (x is y checks if x and y are same), is not

Membership operators: in(let x = 'abc' then 'a' in x will fetch True while 'z' in a will fetch False), not in

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

Equals to has double equal operator(==) while assignment has single equal operator(=)

eg: a= 10, b= 10 (assigns 10 value to a and b)

so, a ==b (returns true because both are same)

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

A student is 'Pass' if percent >33 otherwise he is 'Fail'

Code:

```
percent = 40
```

```
if percent> 33:
```

```
    print('Pass')
```

```
else:
```

```
    print('Fail')
```

8. Recognize the following three blocks in this code:

```
spam = 0
if spam == 10: -----
    print('eggs')                |
    if spam > 5: -----|        |
        print('bacon') -----| ----> if Block | If block
    else: -----|              |
        print('ham')-----| ----> Else block |
    print('spam') -----
print('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.

```
spam = input()
if spam ==1:
    print('Hello')
elif spam ==2:
    print('Howdy')
elif spam ==3:
    print('Salutations')
else:
    print('Something else')
```

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

CTRL+C

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

Break will break that particular loop and terminate all further iterations and will not execute any further statement in that block.

eg: for i in range(4):

```
    print(i)
    if i == 2:
        break
```

output :

0

1

While continue will skip that particular iteration from executing any further statement and go ahead with next iteration

eg: for i in range(3):

```
    print(i)
    if i == 1:
        continue
    print('Hi')
```

output:

0

Hi

1

Hi

2

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

range(10) will consider int 0-9 by default

range(0,10) will consider int starting from 0(as specified) to 9

range(0,10,1) will consider int starting from 0(as specified) to 9 by iterating only by 1 next int(as specified)

There will be no difference in the outputs for all 3 of them.

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.

Code 1:

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

Code 2:

```
num = 0  
while (num <=10):  
    print(num)  
    num+= 1
```

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

```
from spam import bacon  
bacon()
```