

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

True and False are Boolean data types. Capital T and F, remaining letters in lower case.

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

and, or, not are three Boolean operators.

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).

A	B	A and B	A or B	not A
True	False	False	True	False
False	True	False	True	True
False	False	False	False	True
True	True	True	True	False

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?

== equal

!= not equal

< less than

<= less than or equal to

> greater than

>= greater than or equal to

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

`==` is equal to operator that compares the two values and return True, if both values are same and return False if both values are different.

`=` is an assignment operator which simply assigns the value to a variable.

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

Condition is a statement which allow us to tell the computer to perform alternative set of actions based on the result after the statement execution. If, elif, else statements allow us to use condition.

E.g.

```
a = 78
```

```
b = 56
```

```
if b > a:
```

```
    print("b is greater than a")
```

```
elif a == b:
```

```
    print("a and b are equal")
```

```
else:
```

```
    print("a is greater than b")
```

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')
```

The diagram illustrates the nesting of code blocks. A large blue bracket on the right side of the code, labeled "Block1", encompasses the entire conditional statement starting from `if spam == 10:` and ending at the line `print('spam')` just before the final `print('spam')`. Inside this block, there are two smaller blue brackets. The first, labeled "Block2", is positioned to the right of the `if spam > 5:` line and its corresponding `print('bacon')` line. The second, labeled "Block3", is positioned to the right of the `else:` line and its corresponding `print('ham')` line.

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 = 0

if spam == 1:
    print('Hello')

elif spam == 2:
    print('Howdy')

elif spam == 3:
    print('Salutations')

else:
    print('Good bye!')
```

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

CTRL-C key

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

The Break statement terminate the current loop and resumes the execution from the next statement.

```
for i in range(0,10):
```

```
    print(i)

    if i == 5:
        break;
```

output – 0,1,2,3,4,5

The continue statement works opposite to break. It continues the execution of loop. When the continue statement is executed in the loop, the code inside the loop after the continue statement will be skipped and the next iteration of the loop will begin.

```
for i in range(0,10):
```

```
    if i == 5:
        continue;

    print(i)
```

output – 0,1,2,3,4,6,7,8,9

12. In a for loop, what is the difference between `range(10)`, `range(0, 10)`, and `range(0, 10, 1)`?  
`range(10)`, `range(0, 10)`, and `range(0, 10, 1)` all give the same set of numbers ranging from 0 to 9  
`[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.

Using For loop:

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

Using while loop:

```
print('Numbers are:')  
I = 1  
while I < 11:  
    print(i)  
    i += 1
```

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

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