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

A. The Boolean data type is either True or False. In Python Boolean data or Boolean variables are defined by True or False as keywords.

Example:

Graphical user interface, text, application

Description automatically generated

* Above <class ‘bool’> indicates that the variable is of Boolean data type.
* Always the bool variables should start first letter with Upper case i. **True or False**
* If we try to give bool variable as true or false then the following error it generates as:

Graphical user interface, text, application, email

Description automatically generated

* In python integers and floating point numbers can be converted into Boolean data type using function called **bool()**
* When an integer or float set to zero then we get False as return. Similarly when int or float set to 1 then it return True

Graphical user interface, text, application

Description automatically generated Graphical user interface, text, application

Description automatically generated

* We can use Boolean variables in Arithmetic problems such as

Text

Description automatically generated Text

Description automatically generated

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

A. The different Boolean operators in python are:

1. AND Operator (&& or and)
2. NOT Operator (not)
3. OR Operator(|| or “or”)
4. **AND Operator(&& or and):**

Here AND Boolean operator is similar to bitwise AND operator which will analyzes the operands on both side. It is denoted by “and”

True and True = True

True and False = False

False and False = False

False and True = False

Graphical user interface, text, application, chat or text message

Description automatically generated Graphical user interface, text, application, chat or text message

Description automatically generated

1. **NOT Operator(not)**

Here NOT Boolean operator will does the reverse operation to the final result that immediately follows. It is denoted by “not”

not (True) = False

not(False) = True

Graphical user interface, text, application

Description automatically generated Graphical user interface, text, application

Description automatically generated

1. **OR Operator(|| or “or”)**

Here OR Operator is similar to bitwise OR where in bitwise we will look “1” as true and “0” as false. Similarly in OR we look for True or False as final result.

True or False = True

False or True = True

False or False = False

True or False = True

Graphical user interface, text, application

Description automatically generated Graphical user interface, text, application

Description automatically generated

**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. The following are the Boolean operator:

1. AND Operator (&& or and)

2. NOT Operator (not)

OR Operator(|| or “or”)

1. Truth Table for AND Operator

True False True and False False

False True False and True False

False False False and False False

True True True and True True

(True and False) and True False

(False and False) and False False

(True and True) and False False

(True and True) and True True

(False and False) and True False

1. Truth Table for NOT Operator

True not(True) False

False not(False) True

1. Truth Table for OR Operator

True True True or True True

False True False or True True

False False False or False False

True False True or False True

(True or False) or True True

(False or False) or False False

(True or True) or False True

(True or True) or True True

(False or False) or True True

Combination of both OR, NOT AND:

(True or False) and (not(True)) False

(False and True) or(not(False)) True

Not( (True and True) or (False)) False

(Not(False) and True) and (False or not(True) False

**4. What are the values of the following expressions?**

**A.** (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?**

**A.**  The six reference operators are a. Arithmetic Operator

b. Assignment Operator

c. Comparison Operator

d. Logical Operator

e. Identity Operator

a. **Arithmetic Operator**

Arithmetic Operators are used in numeric values to perform operations

+ -> Addition, - -> Subtraction, \* -> Multiplication, / -> division, % -> Modulus, // -> floor

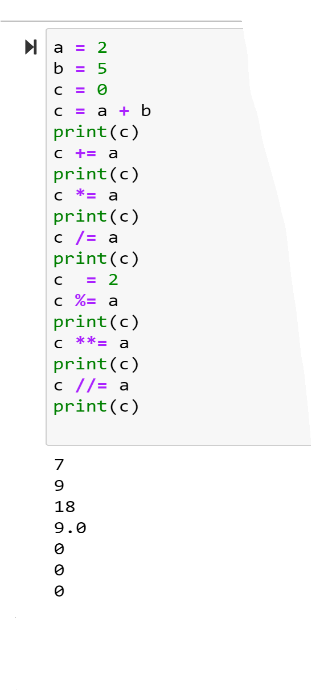
A picture containing graphical user interface

Description automatically generated

b**. Assignment Operator:**

Assignment Opeartor are used to assign values to variables

=, + =, - =, \* =, / =, %=, //= ,\*\*=, ^=, >> =, << = are some assignment operators



c. **Comparison Operator:**

Comparison operator used to compare two values.

==, !=, >, <, >=, <= are comparison operators

A picture containing text

Description automatically generated

**d.Logical Operator:**

logical operators are used in conditional statements

AND, OR, NOT are three logical operators

Text

Description automatically generated

e**.Identity Operator:**

identity operator are used to compare objects.

Is , is not are two identity operators.

Text

Description automatically generated

**f. BitWise Operators:**

Bitwise operators are used to compare binary numbers

&, |, ^, ~, <<, >> are operators of bitwise

Calendar

Description automatically generated

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

A. The assignment operator (“=”) is used to assign the value of right to left

Graphical user interface, text, application, chat or text message

Description automatically generated

The “==” operator will checks whether two operands are equal or not. Means it gives true or false

Text

Description automatically generated Graphical user interface, text, application

Description automatically generated

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

A. Condition is statement which helps programmer to evaluate whether following logic or condition is true or false. In python conditional keywords like if, elif and else.

Graphical user interface, application

Description automatically generated

Above is the example for if condition , here we are checking whether the following string fool is exist in the list of elements or not. It returns true when it exists.

Graphical user interface, text, application

Description automatically generated

Elif and else are two more conditions that are used by programmers to evaluate .

**8. Recognize the following three blocks in this code**

**A.** spam = 0

if spam == 10:

print('eggs') ---------------------🡪 Block 1

if spam > 5:

print('bacon') --------------🡪 Block 2

else:

print('ham') -----------------🡪 Block 3

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.**

**A. Text

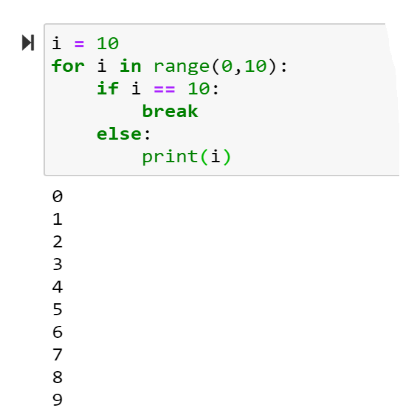
Description automatically generated**

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

**A.**  CTRL + C

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

**A. Break**  statement should be used inside a loop, when it is used it will terminate the loop based on condition . suppose if it is used inside nested loop then it will terminate from the current loop .



Continue statement will terminates or stops to execute current condition and control will go back to next iteration value or first step depends on logic

Graphical user interface, text, application

Description automatically generated

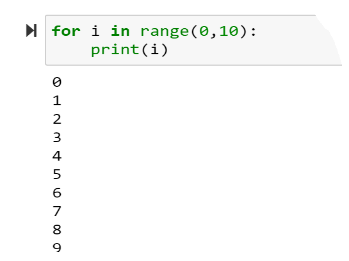
**12. In a for loop, what is the difference between range(10), range(0, 10), and range(0, 10, 1)?**

**A.**  range(10) = it will gives sequence of numbers from 0 to 9

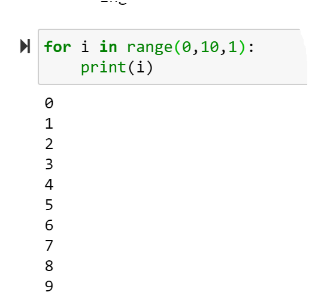
A picture containing text

Description automatically generated

Range(0, 10) = it will generates the numbers from starting index i.e 0 and till end – 1 i.e 9



Range(0,10,1) = it will generates the sequence of numbers from 0 to 9 where start index, end index and step index are arguments . step indicates the increment



**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.**

**A**

Text

Description automatically generated

**A picture containing diagram

Description automatically generated**

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

**A.** spam.bacon()