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

True and False this is the two Boolean values.

We are writing this in python like

e.g print(10 > 5)

print(bool("Hello"))

print(bool(15))

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

‘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 evaluate ).**

|  |  |  |
| --- | --- | --- |
| X | Y | NOT X (~X) |
| F | F | T |
| F | T | T |
| T | F | F |
| T | T | F |

|  |  |  |
| --- | --- | --- |
| X | Y | X AND Y(X&&Y) |
| F | F | F |
| F | T | F |
| T | F | F |
| T | T | T |

|  |  |  |
| --- | --- | --- |
| X | Y | X OR Y(X||Y) |
| F | F | F |
| F | T | T |
| T | F | T |
| T | T | T |

**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?**

1) Assignment Operator

2) Arithmetic Operator

3) Comparison Operator

4) Bitwise Operator

5) Logical Operator

6) Precedence Operator

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

Assignment (=) :- The “=” is an assignment operator is used to assign the value on the right to the variable on the left.

Equal to (==) :- The “==” operator checks whether the two given operands are equal or not. If so, it return True. Otherwise it return False.

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

The Boolean expression in a conditional statement that determines which branch is executed conditional statement. A statement that controls the flow of execution depending on some condition. In python the keyword if , elif, and else are used for conditional statement.

Conditional statements are used to decide the flow of execution based on different condition. If a condition is true, you can perform one action and if the condition is false, you can perform another action.

**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')**

**Output:-** 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.**

num = int(input("Please enter any number 1 to 3 = "))

if num==1:

print("Hello")

elif num==2:

print("Howdy;")

elif num==3:

print("Salutations! ")

else:

print("You are crossed your limit.....!")

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

Press ctrl + c key.

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

Break :- the break statement takes care of terminating the loop in which it is used. If the break statement is used inside nested loops, the current loop is terminated, and the flow will continue with the code followed that comes after the loop.

Continue :- The continue statement skips the code that comes after it and the control is passed back to the start for the next iteration.

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

**range(10) :-** when user call range() with one argument, user will get a series of number that start with 0 and includes every whole number up to, but not including, the number that user have provided as the stop .

Syntax:- range(stop)

**range(0,10) :-** when user call the range() with two arguments user get to decided not only where the series of numbers stops but also where it starts, so user don’t have to start at 0 all the time. User can use range() to generate a series of numbers from X to Y using a range(X,Y).

Syntax:- range(start, stop)

**range(0,10,1):-** when user call range() with three argument, user can choose not only where the series of numbers will start and stop but also how big the difference will be between one number and next. Id user don’t provide a step, then range() will automatically behave as if the step is 1.

Syntax:- range(start, stop, step)

**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:-

for i in range(1,11):

print(i)

Using while loop:-

i = 1

while i <=10:

print(i)

i+=1

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

spam.bacon()