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

**Answer:**

The two values of Boolean (bool) are **True** and **False.**

Sample declaration, is\_assignment\_completed = True, is\_course\_completed = False

--------------------------------------------------------------------------------------------------------------------------------------

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

**Answer:**

1. **not** – takes only one argument and returns the opposite

For example,

is\_completed = True

print (not is\_completed)

>> False

1. **and** – takes two arguments and returns True only if both the arguments are true else it returns False.

For example,

A = True

B = True

C = False

A and B

>> True

A and C

>> False

1. **or** – takes two arguments and returns False only if both the arguments are false else it returns True.

For example,

A = False

B = False

C = True

A or B

>> False

B or C

>> True

--------------------------------------------------------------------------------------------------------------------------------------

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 ).

**Answer:**

1. NOT Boolean Operator

|  |  |
| --- | --- |
| **A** | **Not A** |
| True | False |
| False | True |

1. AND Boolean Operator

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **A and B** |
| True | True | True |
| True | False | False |
| False | True | False |
| False | False | False |

1. OR Boolean Operator

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **A or B** |
| True | True | True |
| True | False | True |
| False | True | True |
| False | False | False |

--------------------------------------------------------------------------------------------------------------------------------------

4. What are the values of the following expressions?

(5 > 4) and (3 == 5)

not (5 > 4)

(5 > 4) or (3 == 5)

not ((5 > 4) or (3 == 5))

(True and True) and (True == False)

(not False) or (not True)

**Answer:**

(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?

**Answer:**

|  |  |
| --- | --- |
| **Comparison Operator** | **Meaning** |
| < | value1 < value2  Checks if value1 is less than value2 |
| <= | value1 <= value2  Checks if value1 is less than or equal to value2 |
| > | value1 > value2  Checks if value1 is greater than value2 |
| >= | value1 >= value2  Checks if value1 is greater than or equal to value2 |
| == | value1 == value2  Checks if value1 is equal to value2 |
| != | value1 != value2  Checks if value1 is not equal to value2 |

--------------------------------------------------------------------------------------------------------------------------------------

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

**Answer:**

Equal to: **==**

It is used to check whether the expressions on the two sides of the operator is equal or not.

Assignment: =

It is used to assign the result of the expression on the right side of the operator to the variable on the left side.

For example,

Value1 = 7 //assignment operator is used to assign the value 7 to the variable Value1

Value2 = 8 //assignment operator is used to assign the value 8 to the variable Value2

print(Value1 == Value2) //equal to operator is used to check if Value1 and Value2 are equal or not (which will return False)

>>False

--------------------------------------------------------------------------------------------------------------------------------------

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')

**Answer:**

**Indentations are missing in the code after the if and else statements** which will throw “IndentationError”. Here is the corrected code,

spam = 0

if spam == 10:

print('eggs')

if spam > 5:

print('bacon')

else:

print('ham')

print('spam')

print('spam')

//Which will print the following

**ham**

**spam**

**spam**

Also, instead of second if condition, **elif can be used to create a nested-if statement**

--------------------------------------------------------------------------------------------------------------------------------------

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.

**Answer:**

spam = input ("Please enter your input: ")

print (f"User input - {spam}")

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?

**Answer:**

**Ctrl + C** which generates an interrupt from the keyword**. (Ctrl + M I in google colab)**

--------------------------------------------------------------------------------------------------------------------------------------

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

**Answer:**

Break: Terminates the loop immediately and comes out of the current loop body and the execution proceeds to the first line of the next loop body (if any)

Continue: Terminates only the current iteration of the loop and the execution jumps to the top of the loop for the next iteration.

--------------------------------------------------------------------------------------------------------------------------------------

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

**Answer:**

In for loops, “range” is used to loop through a set of code for a specific number of times. The range() function returns a sequence of number starting from 0 (by default) and increment by 1 (by default).

1. range(10): The sequence starts from 0(**by default**), increments by 1 (**by default**) and ends at 10th iteration.

Notation: **range(<iteration\_count>)**

1. range(0,10): The sequence starts from 0(**explicitly mentioned**), increments by 1 (**by default**) and ends at 10th iteration.

Notation: **range(<start range>,<iteration\_count>)**

1. range(0,10,1): The sequence starts from 0(**explicitly mentioned**), increments by 1 (**explicitly mentioned**) and ends at 10th iteration. **[Here 1 in range function is the increment count]**

Notation: **range(<start range>,<iteration\_count>,<increment\_by\_number>)**

Note: All the three range functions mentioned in the question will print the following,

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.

**Answer:**

|  |  |
| --- | --- |
| **For Loop** | **While Loop** |
| for x in range(1,11):  print(x) | i = 1  while i <= 10:  print(i)  i += 1 |

--------------------------------------------------------------------------------------------------------------------------------------

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

**Answer:**

**Approach1:**

**from spam import \*** //imports all the functions from spam.py

**bacon()** //calling bacon function

**Approach2:**

**from spam import bacon** //imports only bacon function from spam.py

**bacon()** //calling bacon function

--------------------------------------------------------------------------------------------------------------------------------------