Answer’s / Explanation’s

1. Two values of Boolean type are True, False. We write them as True, False.

2. and, or, not are the tree different type of Boolean operators.

3. Boolean operators are and, or, not.

Operator **and**	Evaluate

True	True	True
True	False	False
False	True	False
False	False	False

Operator **or**	Evaluate

True	True	True
True	False	True
False	True	True
False	False	False

Operator **not**	Evaluate

True	False
False	True

4. False.
False.
True.
False.
False.
True.

5. Six comparison operators are ==, >, <, >=, <=, !=.

6. Equal operator is used represented as == whereas assignment operator is represented as =.

   In case variable assignment we assignment operator. For example,
   A = 1 or B = 17.

   In case of conditional statements to check a value of variable we use equal operator. For example,
   A == 5, if really A is 5 then it will return true else it will return false.

7. Block 1: -
   ```
   if spam == 10:
   print('eggs')
   ```

   Block 2: -
   ```
   if spam > 5:
   print('bacon')
   ```

   Block 3: -
   ```
   else:
   print('ham')
   print('spam')
   print('spam')
   ```

8.
   ```
   If spam == 1:
         print(“Hello”)
   elif spam == 2:
         print(“Howdy”)
   else:
         print(“Greetings!”)
   ```

9. We will press the interruption key in jupyter notebook.

10. The statement ‘break’ will not execute the set of code after break statement and takes control out of the loop.
    Whereas the statement ‘continue’ will not execute the set of code after continue statement and takes control to start of the loop.

11. In range(10), the numbers generated are from 0 – 9, but here default minimum 0 and default step value 1 is used.
    In range(0, 10), the numbers generated are from 0 – 9, but here default step value 1 is used.
    In range(0, 10, 1), the numbers generated are from 0 – 9, here every value is provided.

12. 1st Program using for loop:

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

    2nd program using while loop:

    ```
    i = 1
    while i <= 10:
          print(i)
          i += 1
    ```

13. spam.bacon()