Assignment 2(python basic)

Answer 1.

The two values of Boolean data types are True and False. To write them we have to use key words true and false in the code .

Answer 2.

1. Logical and (&&) : this operators returns true if both operands are true , otherwise it returns false.
2. Logical or(//); this operator returns true if at least one is true
3. Logical not: this operates negates the value of operand .

Answer 3.

Logical and

| **Operand 1** | **Operand 2** | **Result** |
| --- | --- | --- |
| true | true | true |
| true | false | false |
| false | true | false |
| false | false | false |

Logical or

| **Operand 1** | **Operand 2** | **Result** |
| --- | --- | --- |
| true | true | true |
| true | false | true |
| false | true | true |
| false | false | false |

Logical not

| **Operand** | **Result** |
| --- | --- |
| true | false |
| false | true |

Answer 4

1. Value =False
2. Value=False
3. Value=True
4. Value =False
5. Value=False
6. Value= True

Answer 5.

1.= = equals to

2. != not equals to

3. >

4. <

5.<=

6.>=

Answer 6.

The == . it is used foe comparison of two variables and returns the value true and false

The = works for assigning the value on right side to the left side .

Answer 7

Block 1:

pythonCopy code

if spam == 10: print('eggs')

This block is an if statement that checks if the value of **spam** is equal to 10. However, there is an indentation error in your code, as the **print('eggs')** statement should be indented to be part of the if block.

Block 2:

pythonCopy code

if spam > 5: print('bacon')

This block is another if statement that checks if the value of **spam** is greater than 5. If it is, it will execute the **print('bacon')** statement.

B

Answer 8

The code is

Spam = #assign the value

If spam ==1:

Print(“hello)

Elif spam == 2:

Print(“howady”)

Else:

Print(“greetings”)

Answer 9

Ctrl key is needed to be pressed

Answer 10

1. "break" statement:
   * When encountered within a loop (such as a "for" or "while" loop), the "break" statement immediately terminates the loop and continues executing the code after the loop.
   * It exits the loop prematurely, regardless of whether the loop's condition is still true or not.
   * After a "break" statement is executed, the program continues with the next statement outside the loop.
   * "break" is commonly used to exit a loop early based on certain conditions.
2. "continue" statement:
   * When encountered within a loop, the "continue" statement skips the remaining code within the current iteration of the loop and proceeds to the next iteration.
   * It essentially bypasses the remaining statements within the loop body for the current iteration and jumps to the next iteration.
   * The loop's condition is re-evaluated, and if it is still true, the loop continues with the next iteration. If the condition is false, the loop terminates.
   * "continue" is commonly used to skip specific iterations or to filter out certain elements in a loop.

Answer 11

1. **range(10)**: This form of **range()** has a single argument, **stop**, which specifies the number of iterations or the upper limit of the sequence. In this case, **range(10)** will generate a sequence of numbers from 0 to 9.
2. **range(0, 10)**: This form of **range()** has two arguments, **start** and **stop**, where **start** indicates the starting value (inclusive) and **stop** indicates the stopping value (exclusive). In this case, **range(0, 10)** will generate a sequence of numbers from 0 to 9.
3. **range(0, 10, 1)**: This form of **range()** has three arguments, **start**, **stop**, and **step**, where **step** indicates the increment between consecutive numbers. In this case, **range(0, 10, 1)** will generate a sequence of numbers from 0 to 9 with a step size of 1. Since the default step is 1, this form is equivalent to the previous two examples.

Answer 12

1.for num in range(1, 11):

Print(num)

2

num = 1

while num <= 10:

print(num)

num += 1

Answer13

import spam

spam.bacon()