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

- The two values of the Boolean data type are True and False.
- They are written as capital letters, with the first letter capitalized.

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

- The three different types of Boolean operators are:
    - AND: Denoted by the operator "and".
    - OR: Denoted by the operator "or".
    - NOT: Denoted by the operator "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 evaluates).

- Truth table for the AND operator:
    - True and True = True
    - True and False = False
    - False and True = False
    - False and False = False
- Truth table for the OR operator:
    - True or True = True
    - True or False = True
    - False or True = True
    - False or False = False
- Truth table for the NOT operator:
    - not True = False
    - not False = True

4) What are the values of the following expressions?

- (5 > 4) and (3 == 5) evaluates to False.
- not (5 > 4) evaluates to False.
- (5 > 4) or (3 == 5) evaluates to True.
- not ((5 > 4) or (3 == 5)) evaluates to False.
- (True and True) and (True == False) evaluates to False.
- (not False) or (not True) evaluates to True.

5) What are the six comparison operators?

- The six comparison operators are:
    - Greater than: >
    - Less than: <
    - Equal to: ==
    - Not equal to: !=
    - Greater than or equal to: >=
    - Less than or equal to: <=

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

- The equal to operator is "==", used for comparison to check if two values are equal.
- The assignment operator is "=", used for assigning a value to a variable.
- For example, if you want to check if a variable "x" is equal to 5, you would use the equal to operator: if x == 5.
- On the other hand, if you want to assign the value 5 to a variable "x", you would use the assignment operator: x = 5.

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')
```

Block 1:

```
spam = 0
```

Block 2:

```
if spam == 10:

print('eggs')
```

Block 3:

```
      if spam > 5:
          print('bacon')
      else:
          print('ham')
          print('spam')
          print('spam')
```

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.

```
if spam == 1:
    print("Hello")
elif spam == 2:
    print("Howdy")
else:
    print("Greetings!")
```

9) If your program is stuck in an endless loop, what keys would you press?

Press "Ctrl + C" on your keyboard to interrupt the program and stop its execution.

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

The "break" statement is used to exit the current loop and continue with the next statement outside the loop.

The "continue" statement is used to skip the current iteration of the loop and move to the next iteration.

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

- `range(10)` will generate a sequence of numbers from 0 to 9 (exclusive). It starts from 0 by default and increments by 1.
- `range(0, 10)` will also generate a sequence of numbers from 0 to 9 (exclusive). The starting value (0) is explicitly mentioned, and it increments by 1.
- `range(0, 10, 1)` will generate a sequence of numbers from 0 to 9 (exclusive). Both the starting value (0) and the increment value (1) are explicitly mentioned.

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.

Program using a for loop:

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

Program using a while loop:

```
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?

After importing the "spam" module, you can call the "bacon()" function using the following syntax.

```
import spam
spam.bacon()
```