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

A variable of the primitive data type Boolean can have two values: **True and False (Boolean literals)** or off. Boolean expressions use relational and logical operators. The result of a Boolean expression is either true or false.

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

There are three logical operators that are used to compare values. They evaluate expressions down to Boolean values, returning either True or False. These operators are **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 evaluates).

### == Truth Table

| **x** | **==** | **y** | **Returns** |
| --- | --- | --- | --- |
| True | == | True | True |
| True | == | False | False |
| False | == | True | False |
| False | == | False | True |

### AND Truth Table

| **x** | **and** | **y** | **Returns** |
| --- | --- | --- | --- |
| True | and | True | True |
| True | and | False | False |
| False | and | True | False |
| False | and | False | False |

### OR Truth Table

| **x** | **or** | **y** | **Returns** |
| --- | --- | --- | --- |
| True | or | True | True |
| True | or | False | True |
| False | or | True | True |
| False | or | False | False |

### NOT Truth Table

| **not** | **x** | **Returns** |
| --- | --- | --- |
| not | True | False |
| not | False | True |

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 comparison operators?

| **Operator** | **What it means** |
| --- | --- |
| == | Equal to |
| != | Not equal to |
| < | Less than |
| > | Greater than |
| <= | Less than or equal to |
| >= | Greater 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.

**== is the equal operator** to operator that compares two values and evaluates to a Boolean, while **= is the assignment operator** that stores a value in a variable.

A condition is an expression used in a flow control statement that evaluates to a Boolean value.

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')

**sol:** The three blocks are everything inside the if statement and the lines print('bacon') and print('ham').  
print('eggs')  
if spam > 5:  
print('bacon')  
else:  
print('ham')  
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 programme is stuck in an endless loop, what keys you’ll press?

An infinite loop occurs when a program keeps executing within one loop, never leaving it. To exit out of infinite loops on the command line, **press CTRL + C.**

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

**Break** statement mainly used to terminate the enclosing loop such as while or for statement wherever break is declared.

**Continue** statement mainly skip the rest of loop wherever continue is declared and execute the next iteration.

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

All returns a sequence of numbers starting with 0 and ending at 9.

12. Write a short program that prints the numbers 1 to 10 using for loop. Then write an equivalent program that prints the numbers 1 to 10 using a while loop.

* Using for loop

for i in range (1,11):

print(i)

* Using 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?

This function can be called with **spam**.**bacon ()**