- The two values of the Boolean data type are `True` and `False`.

- They are written as `True` and `False` in Python, with the first letter capitalized.

2 - The three Boolean operators are:

1. `and`

2. `or`

3. `no

3.`and` Operator Truth Table

| A | B | A and B |

|-------|-------|---------|

| True | True | True |

| True | False | False |

| False | True | False |

| False | False | False |

#### `or` Operator Truth Table

| A | B | A or B |

|-------|-------|---------|

| True | True | True |

| True | False | True |

| False | True | True |

| False | False | False |

#### `not` Operator Truth Table

| A | not A |

|-------|-------|

| True | False |

| False | True |

### 4.

1. `(5 > 4) and (3 == 5)`

- `True and False`

- \*\*Result:\*\* `False`

2. `not (5 > 4)`

- `not True`

- \*\*Result:\*\* `False`

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

- `True or False`

- \*\*Result:\*\* `True`

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

- `not (True or False)`

- `not True`

- \*\*Result:\*\* `False`

5. `(True and True) and (True == False)`

- `True and False`

- \*\*Result:\*\* `False`

6. `(not False) or (not True)`

- `True or False`

- \*\*Result:\*\* `True`

### 5.

1. `==` (equal to)

2. `!=` (not equal to)

3. `<` (less than)

4. `>` (greater than)

5. `<=` (less than or equal to)

6. `>=` (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.

- The `==` operator is a comparison operator used to check if two values are equal. For example, `if x == 5:` checks whether `x` is equal to `5`.

- The `=` operator is an assignment operator used to assign a value to a variable. For example, `x = 5` assigns the value `5` to the variable `x`.

- \*\*Condition Example:\*\* If you want to assign a value to a variable, use `=`. If you want to compare two values, use `==`.

### 7. Identify the three blocks in this code:

```python

spam = 0

if spam == 10:

print('eggs')

if spam > 5:

print('bacon')

else:

print('ham')

print('spam')

print('spam')

```

- \*\*Block 1:\*\* The `if spam == 10:` block:

```python

if spam == 10:

print('eggs')

```

- \*\*Block 2:\*\* The `if spam > 5:` block:

```python

if spam > 5:

print('bacon')

```

- \*\*Block 3:\*\* The `else:` block:

```python

else:

print('ham')

```

### 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`.

```python

spam = int(input("Enter a value for 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 will you press?

- You can usually stop an endless loop by pressing `Ctrl + C` in most programming environments and terminal windows.

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

- `break`: Exits the loop entirely, stopping any further iteration.

- `continue`: Skips the current iteration and moves to the next iteration of the loop.

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

- `range(10)`: Generates numbers from `0` to `9`. (Default start is `0`, step is `1`).

- `range(0, 10)`: Also generates numbers from `0` to `9`. (Specifically sets the start to `0`, end at `10`).

- `range(0, 10, 1)`: Generates numbers from `0` to `9` with a step of `1`. (Explicitly sets start, end, and step).

### 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.

#### Using a `for` loop:

```python

for i in range(1, 11):

print(i)

```

#### Using a `while` loop:

```python

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`?

```python

import spam

spam.bacon()

```

This should cover all your questions! Let me know if you have any further queries.