1. Values of Boolean data type:

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

- They are written as `True` and `False` in Python.

2. Types of Boolean operators:

- The three different types of Boolean operators are `and`, `or`, and `not`.

3. Truth tables for Boolean operators:

AND Operator (`and`)

| A | B | A and B |

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

| True | True | True |

| True | False | False |

| False | True | False |

| False | False | False |

OR Operator (`or`)

| A | B | A or B |

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

| True | True | True |

| True | False | True |

| False | True | True |

| False | False | False |

NOT Operator (`not`)

| A | not A |

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

| True | False |

| False | True |

4. Values of the 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. Comparison operators:

- `==` (equal to)

- `!=` (not equal to)

- `>` (greater than)

- `<` (less than)

- `>=` (greater than or equal to)

- `<=` (less than or equal to)

6. Equal to vs. assignment operators:

- `==` is the equal to operator, which is used to compare two values.

- `=` is the assignment operator, which is used to assign a value to a variable.

- Example:

```python

x = 5 Assignment: x is assigned the value 5

if x == 5: Comparison: checking if x is equal to 5

print("x is 5")

```

7. Identify the blocks in the code:

```python

spam = 0

if spam == 10:

print('eggs') First block

if spam > 5:

print('bacon') Second block

else:

print('ham') Third block

print('spam') This line is outside the blocks

print('spam') This line is also outside the blocks

```

8. Code to print based on the value of `spam`:

```python

spam = 1 You can change this value to test

if spam == 1:

print("Hello")

elif spam == 2:

print("Howdy")

else:

print("Greetings!")

```

9. Keys to press if a program is stuck in an endless loop:

- Press `Ctrl + C` to interrupt the loop and stop the program.

10. Difference between `break` and `continue`:

- `break` exits the nearest enclosing loop entirely.

- `continue` skips the rest of the code inside the current loop iteration and proceeds to the next iteration.

11. Difference between `range(10)`, `range(0, 10)`, and `range(0, 10, 1)`:

- `range(10)` generates numbers from 0 to 9.

- `range(0, 10)` also generates numbers from 0 to 9, explicitly starting from 0.

- `range(0, 10, 1)` generates numbers from 0 to 9, explicitly starting from 0 and incrementing by 1.

12. Programs to print numbers 1 to 10:

```python

Using a for loop

for i in range(1, 11):

print(i)

Using a while loop

i = 1

while i <= 10:

print(i)

i += 1

```

13. Calling a function inside a module:

```python

import spam

spam.bacon()

```