**Name: Abhas**

**Email: abhas2495@gmail.com**

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

**Answers:**

**True:** Represents the logical value "true" or "yes".

In Python, it is written as the keyword True (with an uppercase T).

Example: x = True

**False:** Represents the logical value "false" or "no".

In Python, it is written as the keyword False (with an uppercase F).

Example: y = False

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

The three different types of Boolean operators are:

**AND Operator (represented by and):**

The AND operator returns True if both operands are True; otherwise, it returns False.

It evaluates to True only if all the conditions connected by the AND operator are True.

Example: True and False evaluates to False.

**OR Operator (represented by or):**

The OR operator returns True if at least one of the operands is True; otherwise, it returns False.

It evaluates to True if any of the conditions connected by the OR operator is True.

Example: True or False evaluates to True.

**NOT Operator (represented by not):**

The NOT operator is a unary operator that negates the value of its operand.

It returns True if the operand is False, and vice versa.

Example: not True evaluates to False.

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 evaluate ).

AND Operator (and):

| **Operand 1** | **Operand 2** | **Result** |
| --- | --- | --- |
| True | True | True |
| True | False | False |
| False | True | False |
| False | False | False |

OR Operator (or):

| **Operand 1** | **Operand 2** | **Result** |
| --- | --- | --- |
| True | True | True |
| True | False | True |
| False | True | True |
| False | False | False |

NOT Operator (not):

| **Operand** | **Result** |
| --- | --- |
| True | False |
| False | True |

4. What are the values of the following expressions?

(5 > 4) and (3 == 5)

not (5 > 4)

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

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

(True and True) and (True == False)

(not False) or (not True)

Answers:

**(5 > 4) and (3 == 5):**

(5 > 4) evaluates to True.

(3 == 5) evaluates to False.

True and False evaluates to False.

**(5 > 4) or (3 == 5):**

(5 > 4) evaluates to True.

(3 == 5) evaluates to False.

True or False evaluates to True.

**not (5 > 4):**

(5 > 4) evaluates to True.

Applying the not operator, True becomes False.

**not ((5 > 4) or (3 == 5)):**

(5 > 4) evaluates to True.

(3 == 5) evaluates to False.

True or False evaluates to True.

Applying the not operator, True becomes False.

**(True and True) and (True == False):**

True and True evaluates to True.

(True == False) evaluates to False.

True and False evaluates to False.

**(not False) or (not True):**

not False evaluates to True.

not True evaluates to False.

True or False evaluates to True.

5. What are the six comparison operators?

**1. Equal to (==):**

Checks if two values are equal.

Example: 5 == 5 evaluates to True.

**2. Not equal to (!=):**

Checks if two values are not equal.

Example: 5 != 3 evaluates to True.

**3. Greater than (>):**

Checks if the left operand is greater than the right operand.

Example: 5 > 3 evaluates to True.

**4. Less than (<):**

Checks if the left operand is less than the right operand.

Example: 3 < 5 evaluates to True.

**5. Greater than or equal to (>=):**

Checks if the left operand is greater than or equal to the right operand.

Example: 5 >= 5 evaluates to True.

**6. Less than or equal to (<=):**

Checks if the left operand is less than or equal to the right operand.

Example: 3 <= 5 evaluates to True.

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

**Equal To Operator (==):**

1. The equal to operator is used to compare two values to check if they are equal.
2. It returns a boolean value of True if the values are equal and False if they are not.

Example: x == y compares the values of variables x and y to determine if they are equal.

**Assignment Operator (=):**

1. The assignment operator is used to assign a value to a variable.
2. It assigns the value on the right-hand side to the variable on the left-hand side.

Example: x = 10 assigns the value 10 to the variable x

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')

**Answers:**

**Block1 :**

if spam == 10:

print('eggs')

**Block 2:**

if spam > 5:

print('bacon')

**Block 3:**

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.

**Answers:**

spam = int(input("Enter the value for spam: ")) # Assuming user provides the value for 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?

**Answers**

1. Ctrl + C:

2. Ctrl + \ (backslash)

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

1. **Break:** The break statement is used to exit or terminate the innermost loop (such as for or while) entirely.

2. **Continue:** The break statement is used to exit or terminate the innermost loop (such as for or while) entirely.

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

**Answers:**

**range(10):**

This form of range() specifies only the stop value.

It generates a sequence of numbers starting from 0 (default start value) up to, but not including, the specified stop value (10 in this case).

The step value is assumed to be 1 by default.

**range(0, 10):**

This form of range() specifies both the start and stop values.

It generates a sequence of numbers starting from the specified start value (0 in this case) up to, but not including, the specified stop value (10 in this case).

The step value is assumed to be 1 by default.

**range(0, 10, 1):**

This form of range() specifies both the start, stop, and step values.

It generates a sequence of numbers starting from the specified start value (0 in this case) up to, but not including, the specified stop value (10 in this case), incrementing by the specified step value (1 in this case).

The step value determines the increment between each number in the sequence.

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**

**for i in range(1, 11):**

**print(i)**

**# 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?

**Answers:**

import spam

spam.bacon()