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

Ans:

The two values of the Boolean data type are true and false. They are written as "True" and "False" or represented by "1" and "0" in Python.

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

Ans:

The three different types of Boolean operators are:

1. AND operator: It combines two Boolean expressions and evaluates to true only if both expressions are true. It is represented by "&&" or "AND".
2. OR operator: It combines two Boolean expressions and evaluates to true if at least one of the expressions is true. It is represented by "||" or "OR".
3. NOT operator: It negates a Boolean expression, converting true to false and false to true. It is represented by "!" or "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 evaluate ).

Ans:

1. AND Operator (&&):

|  |  |  |
| --- | --- | --- |
| Operand 1 | Operand 2 | Result |
| TRUE | TRUE | TRUE |
| TRUE | FALSE | FALSE |
| FALSE | TRUE | FALSE |
| FALSE | FALSE | FALSE |

1. OR Operator (||):

|  |  |  |
| --- | --- | --- |
| Operand 1 | Operand 2 | Result |
| TRUE | TRUE | TRUE |
| TRUE | FALSE | TRUE |
| FALSE | TRUE | TRUE |
| FALSE | FALSE | FALSE |

1. NOT Operator (!):

|  |  |
| --- | --- |
| 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)

Ans:

1. (5 > 4) and (3 == 5) = false
2. not (5 > 4) = false
3. (5 > 4) or (3 == 5) = true
4. not ((5 > 4) or (3 == 5)) = false
5. (True and True) and (True == False) = false
6. (not False) or (not True) = true

**5.** What are the six comparison operators?

Ans:

1. Equal to (==)
2. Not equal to (!=)
3. Greater than (>)
4. Less than (<)
5. Greater than or equal to (>=)
6. 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.

Ans:

The equal to (==) operator is used for comparison, checking if two values are equal. The assignment (=) operator is used to assign a value to a variable.

Eg :

x = 5 (This is assignment operator)

y = 7

if x == y: (This is equal to operator)

print("x and y are equal")

else:

print("x and y are not equal")

**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')

Ans:

1. Block 1:
   1. if spam == 10:
      1. print('eggs')
2. Block 2:
   1. if spam > 5:
      1. print('bacon')
3. Block 3:
   1. 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.

Ans:

spam = input("Enter any number")

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?

Ans:

Ctrl + C

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

Ans:

* The "break" statement is used to immediately terminate a loop and continue with the code that follows the loop.
* The "continue" statement is used to skip the remaining code within a loop iteration and move on to the next iteration.

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

Ans:

* range(10) generates a sequence of numbers from 0 to 9, incrementing by 1
* range(0, 10) also generates a sequence of numbers from 0 to 9, incrementing by 1
* range(0, 10, 1) generates a sequence of numbers from 0 to 9, incrementing by 1 (step value given = 1)

**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.

Ans:

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?

Ans: spam.bacon()