Q.1. Create two int type variables, apply addition, subtraction, division and multiplications

and store the results in variables. Then print the data in the following format by calling the

variables:

First variable is \_\_ & second variable is \_\_.

Addition: \_\_ + \_\_ = \_\_

Subtraction: \_\_ - \_\_ = \_\_

Multiplication: \_\_ \* \_\_ = \_\_

Division: \_\_ / \_\_ = \_\_

**Answer:**

a = 42

b = 12

addition = a + b

subtraction = a - b

multiplication = a \* b

division = a / b

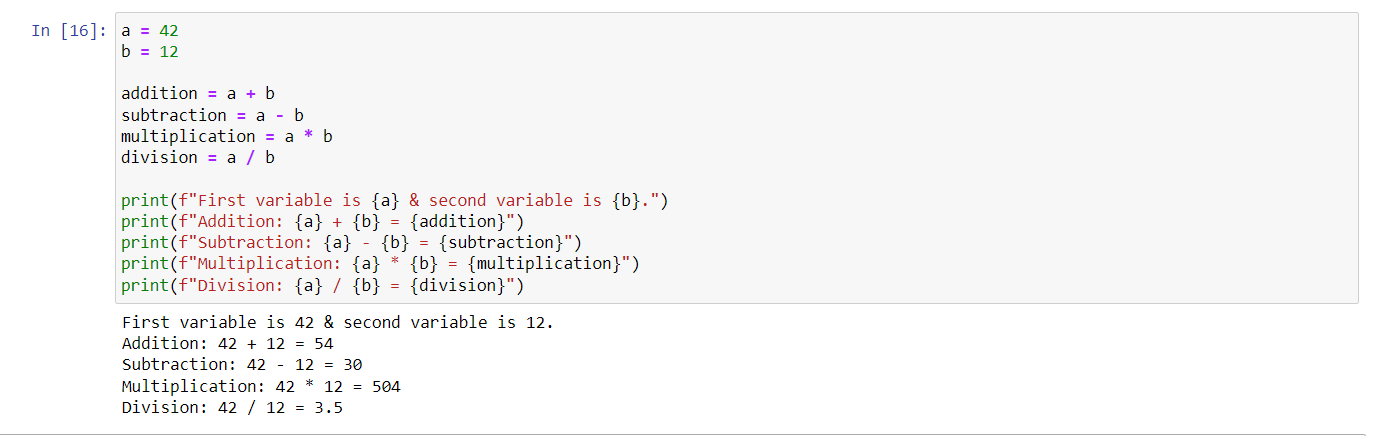
print(f"First variable is {a} & second variable is {b}.")

print(f"Addition: {a} + {b} = {addition}")

print(f"Subtraction: {a} - {b} = {subtraction}")

print(f"Multiplication: {a} \* {b} = {multiplication}")

print(f"Division: {a} / {b} = {division}")



**==================================================================**

Q.2. What is the difference between the following operators:

(i) ‘/’ & ‘//’

(ii) ‘\*\*’ & ‘^’

**Answer:**

**(i) ‘/’ & ‘//’**

🡪

**/** This is the division operator in Python. It performs normal division, which results in a floating-point number as the quotient. It considers decimal points in the result.

Ex: 3/2 = 1.5

**//** This is the floor division operator in Python. It performs division and returns the integer part of the quotient. It disregards the decimal points and returns the closest integer that is less than or equal to the actual quotient.

Ex: 3/2 =1

**(ii) ‘\*\*’ & ‘^’**

**\*\***: This is the exponentiation operator in Python. It raises the base number to the power of the exponent.

Ex: 2\*\*4 = 16

**^** (Caret): This is not the bitwise operator in Python. In Python, the ^ operator is the bitwise XOR operator. It performs bitwise XOR operation on the binary representations of the numbers.

**==================================================================**

Q.3. List the logical operators.

**Answer:**

We have 3 logical operators in the python.

**and**: The logical AND operator returns True if both operands are True, otherwise, it returns False

**or**: The logical OR operator returns True if at least one of the operands is True, otherwise, it returns False

**not**: The logical NOT operator is a unary operator that negates the truth value of its operand. It returns True if the operand is False, and vice versa

**==================================================================**

**Q.4. Explain right shift operator and left shift operator with examples.**

**Answer:**

**Right Shift (>>)**

The right shift operator shifts the bits of a number to the right by a specified number of positions. It moves each bit to the right by the given number of positions, and fills the leftmost positions with zeros.



**Left Shift (<<)**

The left shift operator shifts the bits of a number to the left by a specified number of positions. It moves each bit to the left by the given number of positions, and fills the rightmost positions with zeros.



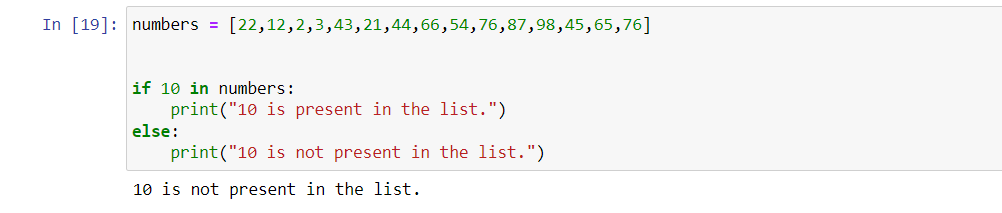
**==================================================================**

Q.5. Create a list containing int type data of length 15. Then write a code to check if 10 is

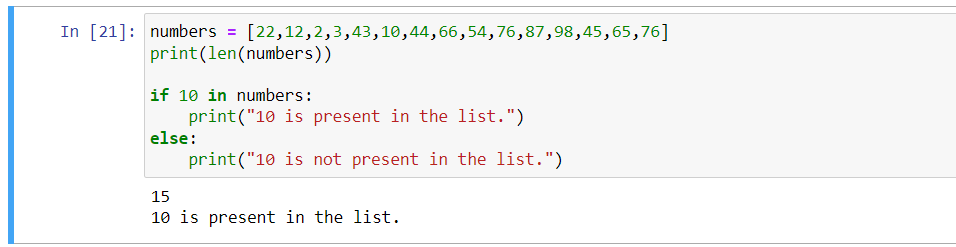
present in the list or not.

**Answer:**

When 10 in not present:



When 10 is present:



**==================================================================**