PYTHON PROGRAMMING LANGUAGE

OPERATORS

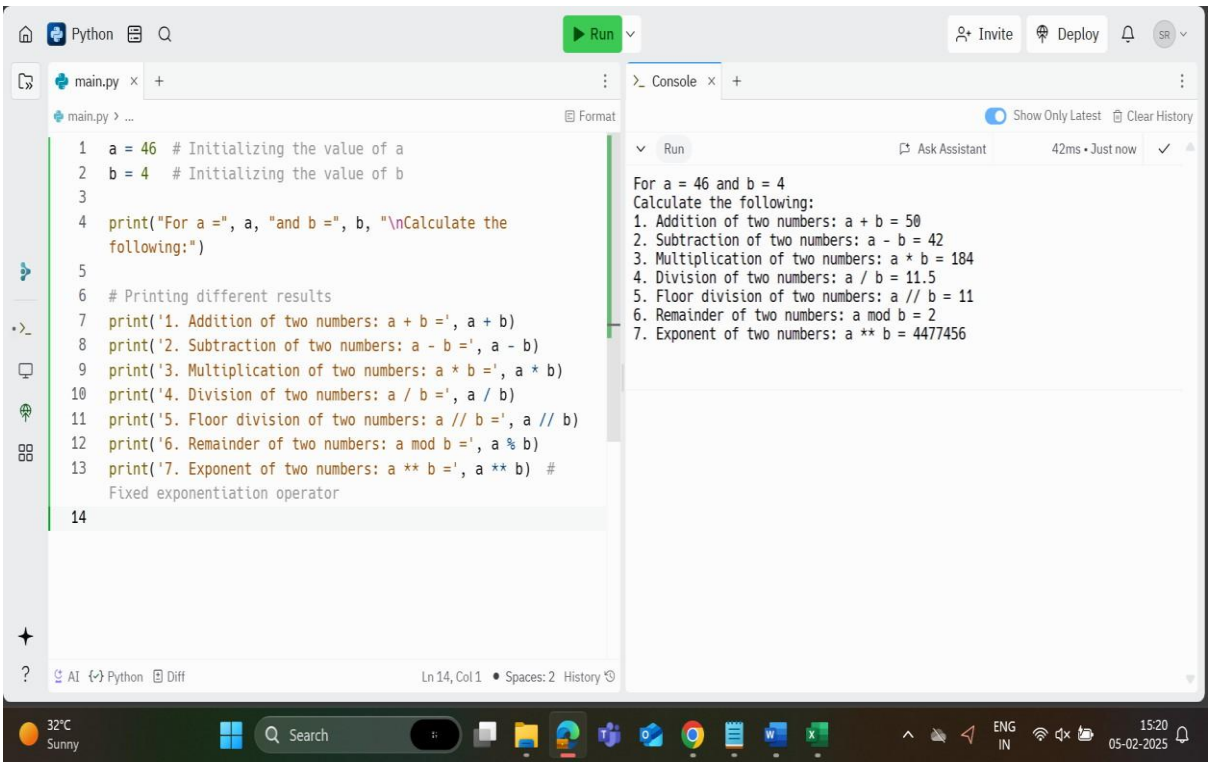
Operators are the symbols used to perform a specific operation on different values and variables. These values and variables are considered as the Operands, on which the operator is applied. Operators serve as the foundation upon which logic is constructed in a program in a particular programming language. In every programming language, some operators perform several tasks.

Arithmetic Operators

- Python Arithmetic Operators are used on two operands to perform basic mathematical operators like addition, subtraction, multiplication, and division.
- There are different types of arithmetic operators available in Python including the '+' operator for addition, '-' operator for subtraction, '*' for multiplication, '/' for division,
- '%' for modulus, '**' for exponent and '//' for floor division.

Example 1: Using all Arithmetic Operators:

Output:

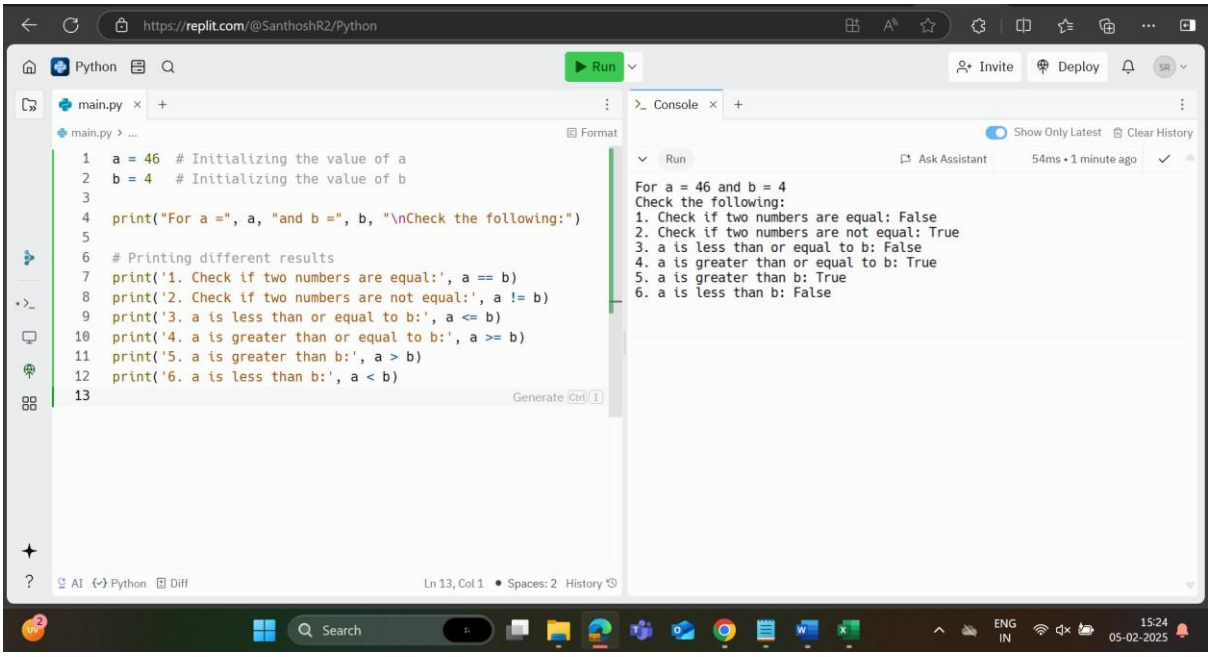


Comparison Operators

Python Comparison operators are mainly used for the purpose of comparing two values or variables (operands) and return a Boolean value as either True or False accordingly. There are various types of comparison operators available in Python including the '==', '!=', '<=', '>=', '<', and '>'.

Example 2: Using all Comparison Operators:

Output:

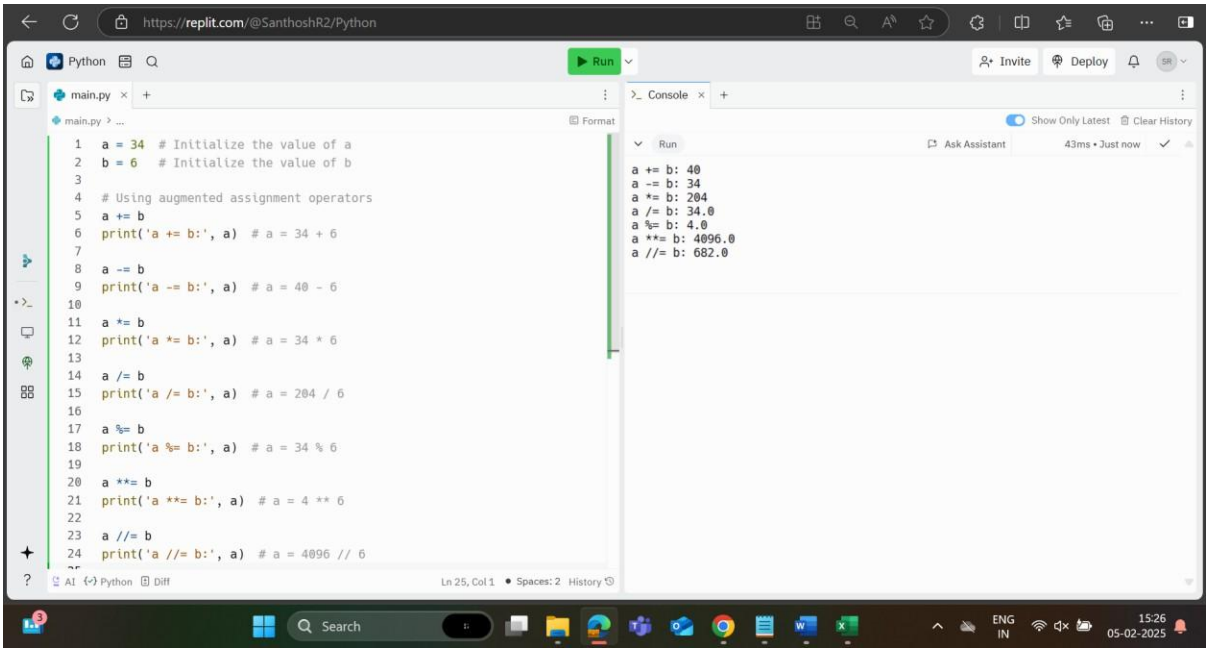


Assignment Operators

Using the assignment operators, the right expression's value is assigned to the left operand. Python offers different assignment operators to assign values to the variable. These assignment operators include '=', '+=', '-=', '*=', '/=', '%=', '//=', '**=', '&=', '|=', '^=', '>>=', and '<<='.

Example 3: Using all Assignment Operators:

Output:

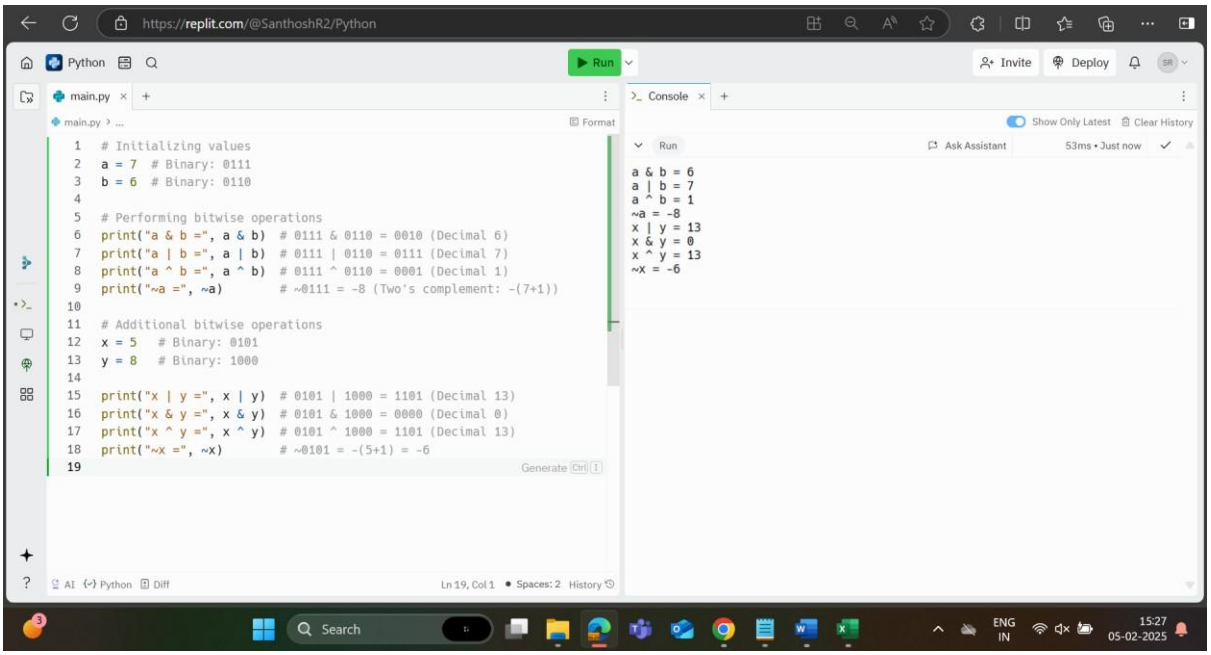


Bitwise Operators

The two operands' values are processed bit by bit by the bitwise operators. There are various Bitwise operators used in Python, such as bitwise OR (|), bitwise AND (&), bitwise XOR (^), negation (~), Left shift (<>). Consider the case below.

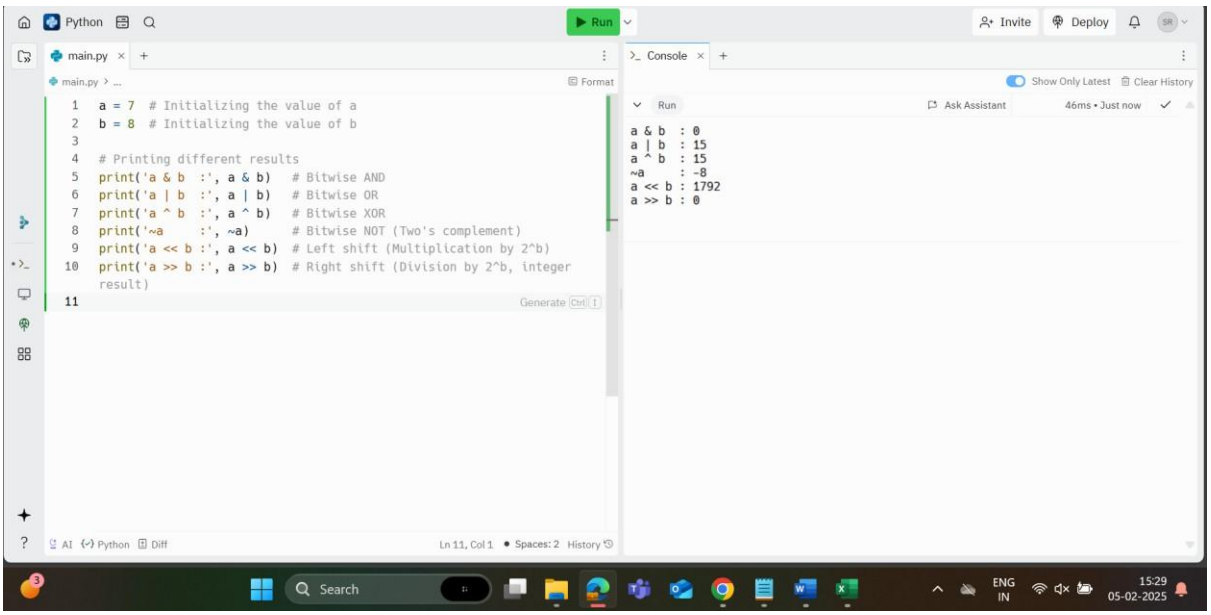
Example 4: Using all Bitwise Operators:

Output:



Example 5: Using all Bitwise Operators:

Output:

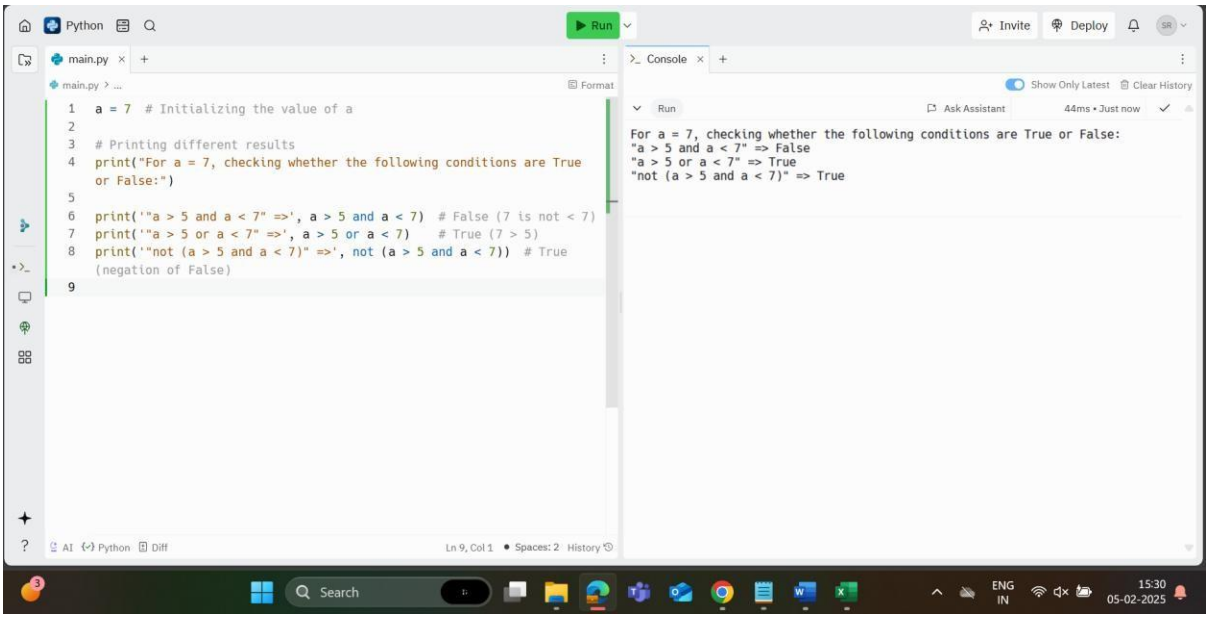


Logical Operators

The assessment of expressions to make decisions typically uses logical operators. Python offers different types of logical operators such as and, or, and not. In the case of the logical AND, if the first one is 0, it does not depend upon the second one. In the case of the logical OR, if the first one is 1, it does not depend on the second one.

Example 6: Using all Logical Operators:

Output:

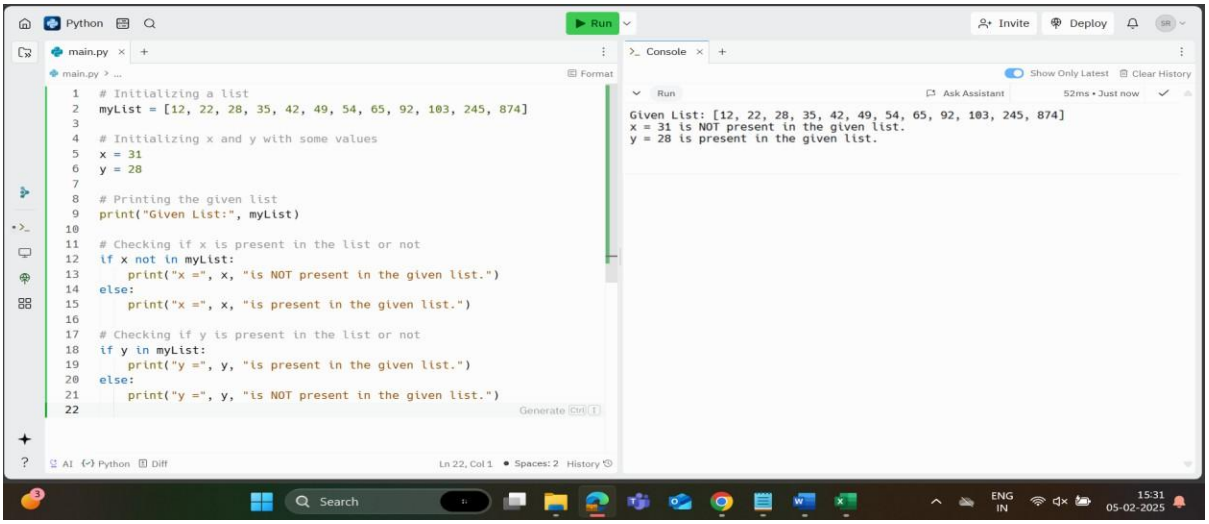


Membership Operators

We can verify the membership of a value inside a Python data structure using the Python membership operators. The result is said to be true if the value or variable is in the sequence (list, tuple, or dictionary); otherwise, it returns false.

Example 7: Using all Membership Operators:

Output:

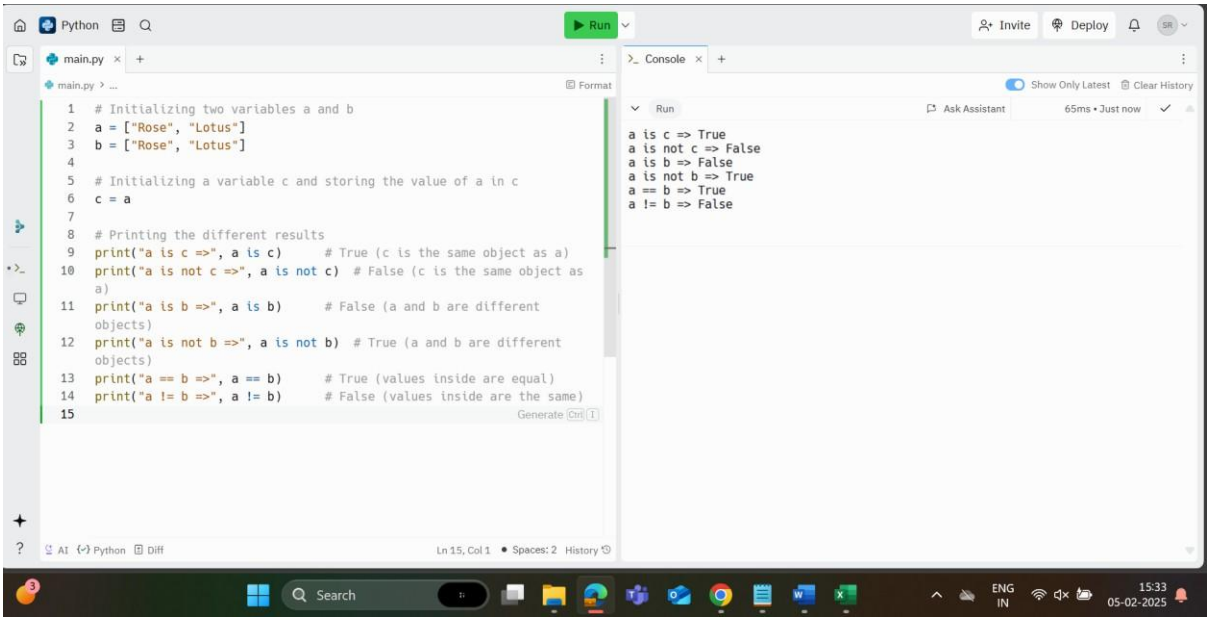


Identity Operators

Python offers two identity operators as is and is not, that are used to check if two values are located on the same part of the memory. Two variables that are equal do not imply that they are identical.

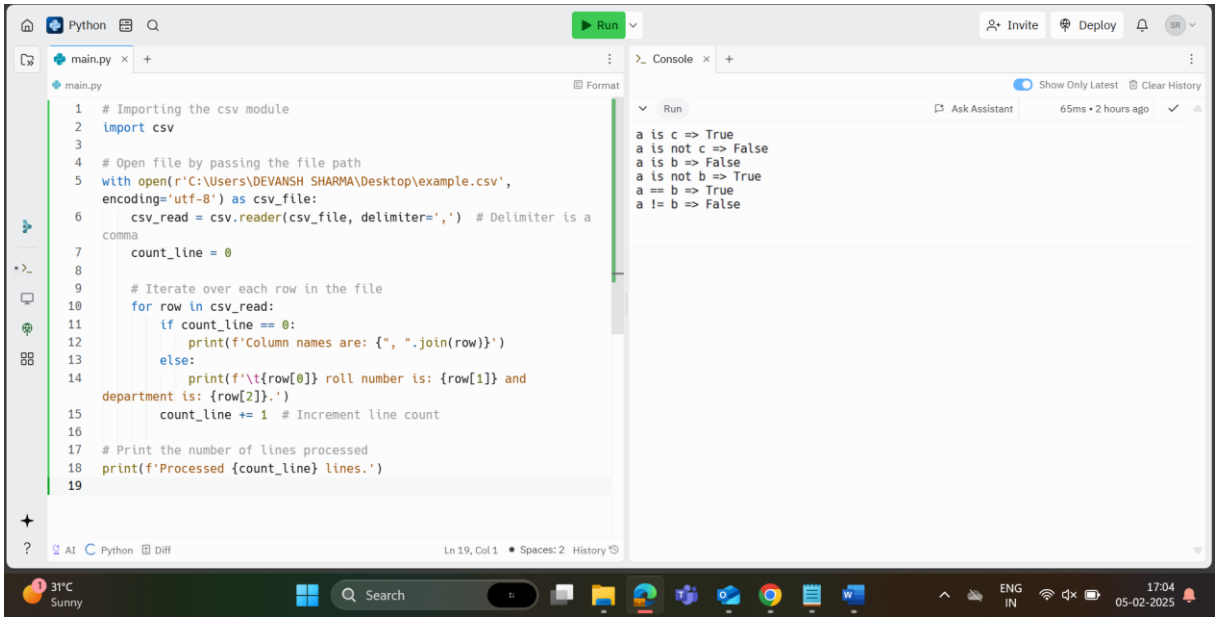
Example 8: Using all Identity Operators:

Output:



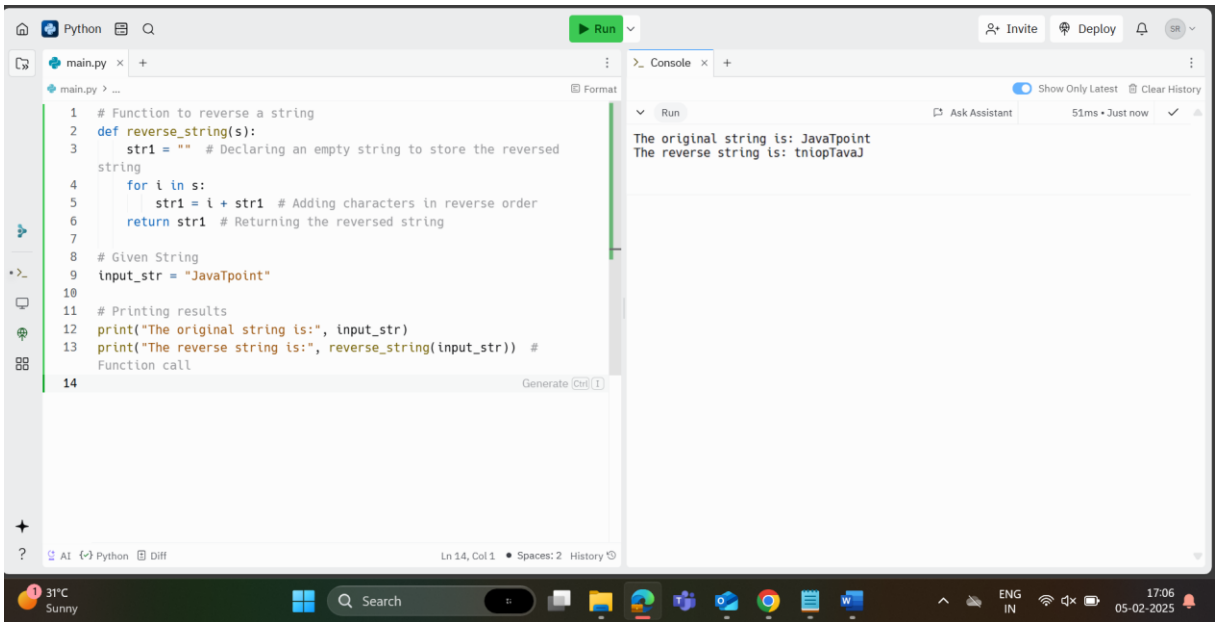
How to read CSV file in Python

The CSV file stands for a comma-separated values file. It is a type of plain text file where the information is organized in the tabular form. It can contain only the actual text data. The textual data don't need to be separated by the commas (,). There are also many separator characters such as tab (\t), colon(:), and semi-colon(;), which can be used as a separator. Let's understand the following example.



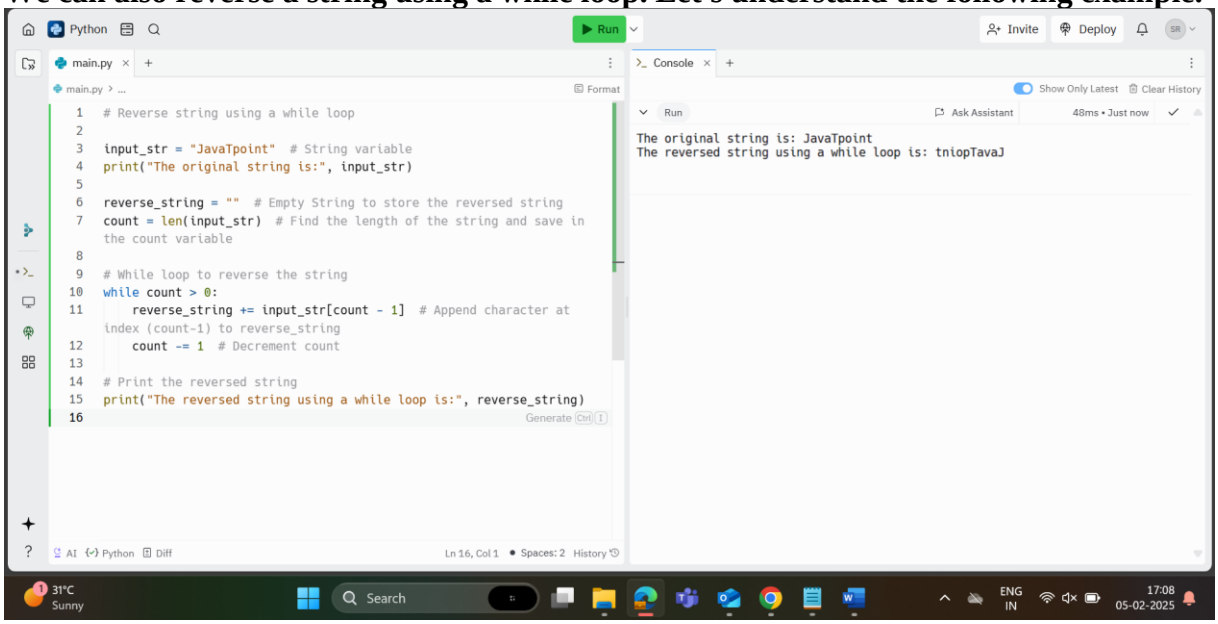
How to reverse a string in Python

The collection of Unicode characters is Python String. Python has various capabilities for string control, yet Python string library doesn't uphold the in-constructed "switch()" capability. However, there are numerous methods for reversing the string. The following reverse Python String method is being defined.



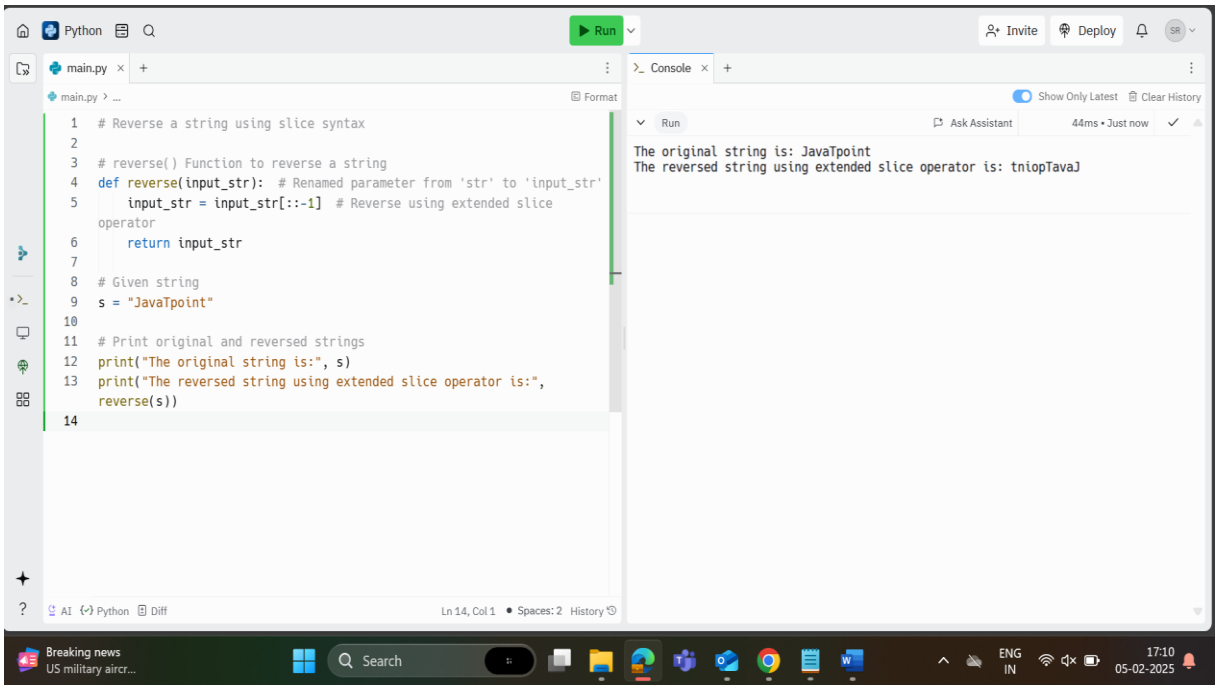
Using while loop

We can also reverse a string using a while loop. Let's understand the following example.



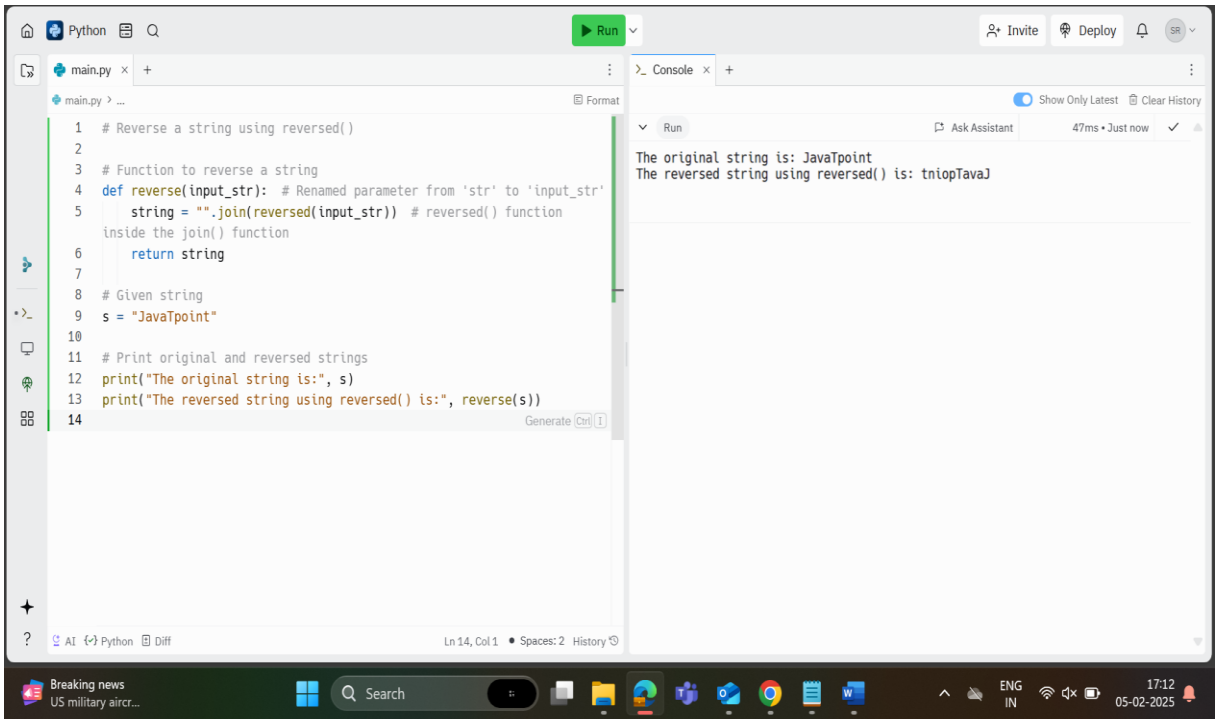
Using the slice ([]) operator

We can also reverse the given string using the extended slice operator. Let's understand the following example.



Using reverse function with join

Python provides the reversed() function to reverse the string. Let's understand the following example.



Using recursion()

The recursion can also be used to turn the string around. Recursion is a cycle where capability calls itself. Look at the following example.