Report

String Manipulation

This report provides an overview of a Python program designed for string manipulation. The program takes a user input string and performs various operations to demonstrate fundamental text processing capabilities. These operations include calculating the length of the string, converting it to uppercase and lowercase, checking if it contains only alphabetic characters, replacing a specific substring, and splitting the string into a list of words.

The program begins by soliciting user input. Users are prompted to enter a string of their choice.

- **Operation 1: Length of the String:** The program calculates and prints the length of the input string using the `len()` function. This operation provides a simple but essential metric for understanding the size of the text.
- **Operation 2: Uppercase and Lowercase Conversion:** Next, the program converts the input string to both uppercase and lowercase using the `upper()` and `lower()` string methods, respectively. It then prints both versions. These operations are valuable for standardizing text or performing case-insensitive comparisons.
- **Operation 3: Alphabetic Character Check:** The program checks if the input string contains only alphabetic characters using the `isalpha()` method. It prints the result, indicating whether the string consists solely of letters. This operation is useful for validating input data for specific text-based applications.
- **Operation 4: Substring Replacement:** Users are prompted to enter a specific substring they wish to replace and the replacement substring. The program utilizes the `replace()` method to substitute the old substring with the new one. It then prints the modified string. This operation is beneficial for text editing and manipulation tasks.
- **Operation 5: Splitting into Words:** Finally, the program splits the input string into a list of words using the `split()` method. It defaults to splitting on whitespace but can be customized to use a different delimiter. The list of words is then printed, which is valuable for text analysis, tokenization, and parsing.

In conclusion, the Python program presented in this report offers a practical set of text manipulation operations. It enables users to interactively explore and modify textual data, making it a valuable tool for a wide range of applications in data science, natural language processing, and general text processing tasks. Its ease of use and clear outputs make it accessible to users with varying levels of programming experience.