# Simple Calculator Application Documentation

## Overview

This Python script creates a graphical user interface (GUI) calculator application using tkinter, the standard Python interface to the Tk GUI toolkit. The calculator includes basic arithmetic operations such as addition, subtraction, multiplication, division, and percentage, as well as the ability to change the sign of the number.

## Requirements

- Python: Version 3.x+  
- tkinter: Should be included with Python. If not, it can be installed or verified according to Python's installation on your system.

## Setup and Configuration

No special setup or external configuration is needed beyond having Python installed with the tkinter library.

## Components of the Application

- Main Window Configuration: Sets up the main window with a title and fixed size.  
- Entry Widget: A text entry for displaying results and input, configured with a large font size for readability.  
- Buttons: A series of buttons corresponding to digits, operations, and functionalities like clear, sign toggle, and percentage.

## Key Functionalities

1. Arithmetic Operations: Allows basic operations like addition (+), subtraction (-), multiplication (×), and division (÷).  
2. Percentage: Converts a number to a percentage.  
3. Toggle Sign: Changes the sign of the current number.  
4. Clear Entry: Clears the current input.  
5. Result Evaluation: Evaluates the expression and handles errors like invalid inputs or operations.

## Running the Application

To run the application:  
- Navigate to the directory containing the script.  
- Run the command `python filename.py` replacing `filename.py` with the name of your script file.

## Error Handling

Errors such as invalid arithmetic operations or non-numeric input for arithmetic are caught and displayed as 'Error' in the entry widget.

## Future Enhancements

- Adding functionality for more complex mathematical operations.  
- Implementing memory functions like M+, M-, MR, and MC.  
- Enhancing the UI with more sophisticated styles and responsive design.

## Security and Limitations

The use of `eval()` for arithmetic operations can pose a security risk if not properly sandboxed, especially in more open or extended implementations. It should be used cautiously and replaced with a safer alternative if the application scope increases.