**Calculator Application Documentation**

**Overview**

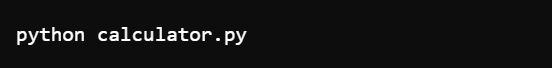
This application is a basic calculator developed using Python’s Tkinter library. It provides essential functionalities such as addition, subtraction, multiplication, and division. The calculator supports both button-based input and keyboard interactions for a seamless user experience. Additional features include a backspace button for corrections and error handling for invalid inputs.

**Requirements**

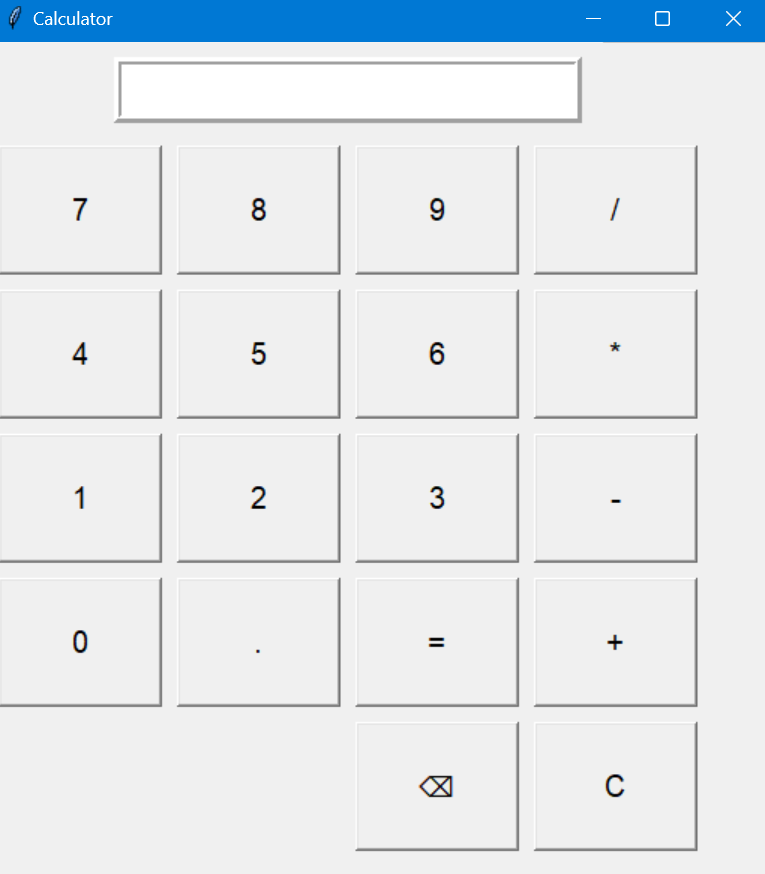
* Python 3.x
* Tkinter (usually included with Python’s standard library)

**Installation**

1. Clone the repository or download the script file.

****

1. Run the script in any Python IDE or via command line
2. **A window with the calculator interface should appear.**

****

**Code:**

import tkinter as tk

root = tk.Tk()

root.title("Calculator")

root.geometry("400x500")

display = tk.Entry(root, font=("Arial", 20), borderwidth=5, relief="ridge", justify="right")

display.grid(row=0, column=0, columnspan=4, padx=10, pady=10)

def update\_display(value):

display.insert(tk.END, value)

def clear\_display():

display.delete(0, tk.END)

def backspace():

display.delete(len(display.get())-1, tk.END)

def calculate():

try:

result = eval(display.get())

clear\_display()

display.insert(tk.END, result)

except ZeroDivisionError:

clear\_display()

display.insert(tk.END, "Cannot divide by zero")

except Exception:

clear\_display()

display.insert(tk.END, "Error")

buttons = [

('7', 1, 0), ('8', 1, 1), ('9', 1, 2), ('/', 1, 3),

('4', 2, 0), ('5', 2, 1), ('6', 2, 2), ('\*', 2, 3),

('1', 3, 0), ('2', 3, 1), ('3', 3, 2), ('-', 3, 3),

('0', 4, 0), ('.', 4, 1), ('+', 4, 3), ('=', 4, 2)

]

for (text, row, col) in buttons:

if text == "=":

button = tk.Button(root, text=text, height=3, width=9, font=("Arial", 15), command=calculate)

else:

button = tk.Button(root, text=text, height=3, width=9, font=("Arial", 15), command=lambda txt=text: update\_display(txt))

button.grid(row=row, column=col, padx=5, pady=5)

clear\_button = tk.Button(root, text="C", height=3, width=9, font=("Arial", 15), command=clear\_display)

clear\_button.grid(row=5, column=3, padx=5, pady=5)

backspace\_button = tk.Button(root, text="⌫", height=3, width=9, font=("Arial", 15), command=backspace)

backspace\_button.grid(row=5, column=2, padx=5, pady=5)

root.bind('<Return>', lambda event: calculate())

root.bind('<BackSpace>', lambda event: backspace())

root.bind('<Escape>', lambda event: clear\_display())

for key in '0123456789/\*-+.':

root.bind(key, lambda event, digit=key: update\_display(digit))

root.mainloop()

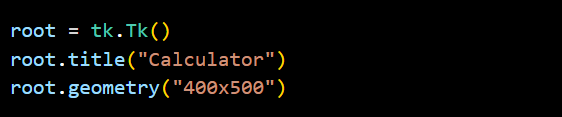
**Code Structure**

**1. Imports**

****

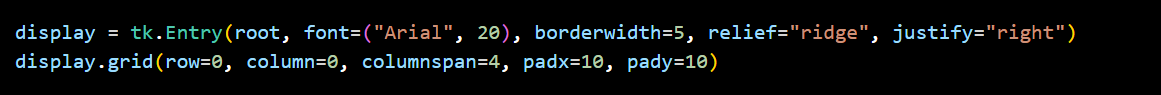
We import Tkinter, which is used to create the graphical interface of the calculator.

**2. Root Window Setup**



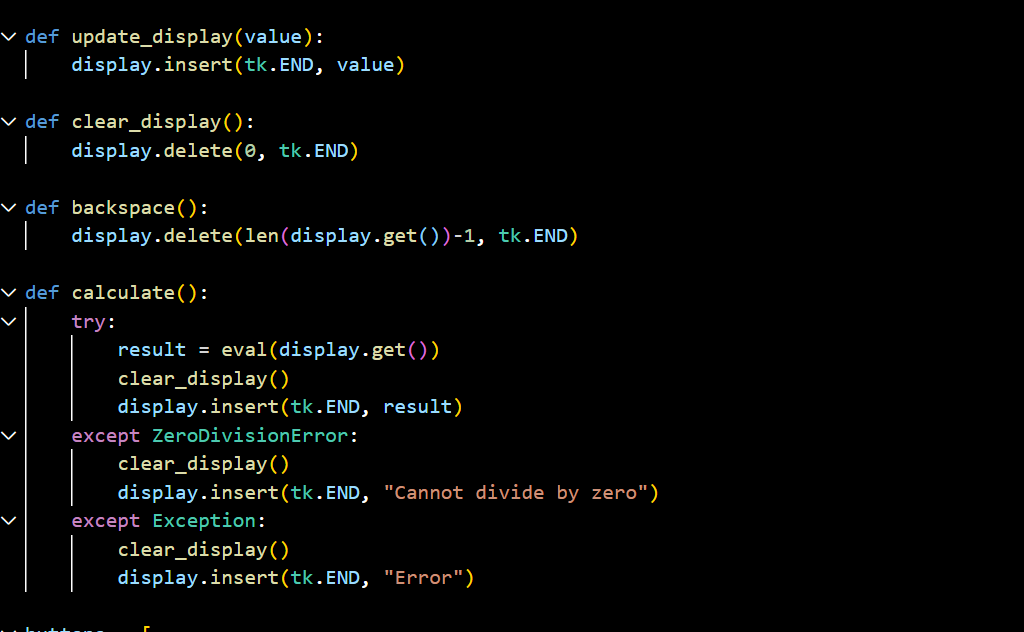
* **root**: Creates the main application window.
* **title**: Sets the title of the application window.
* **geometry**: Sets the initial window size.

**3. Display Setup**



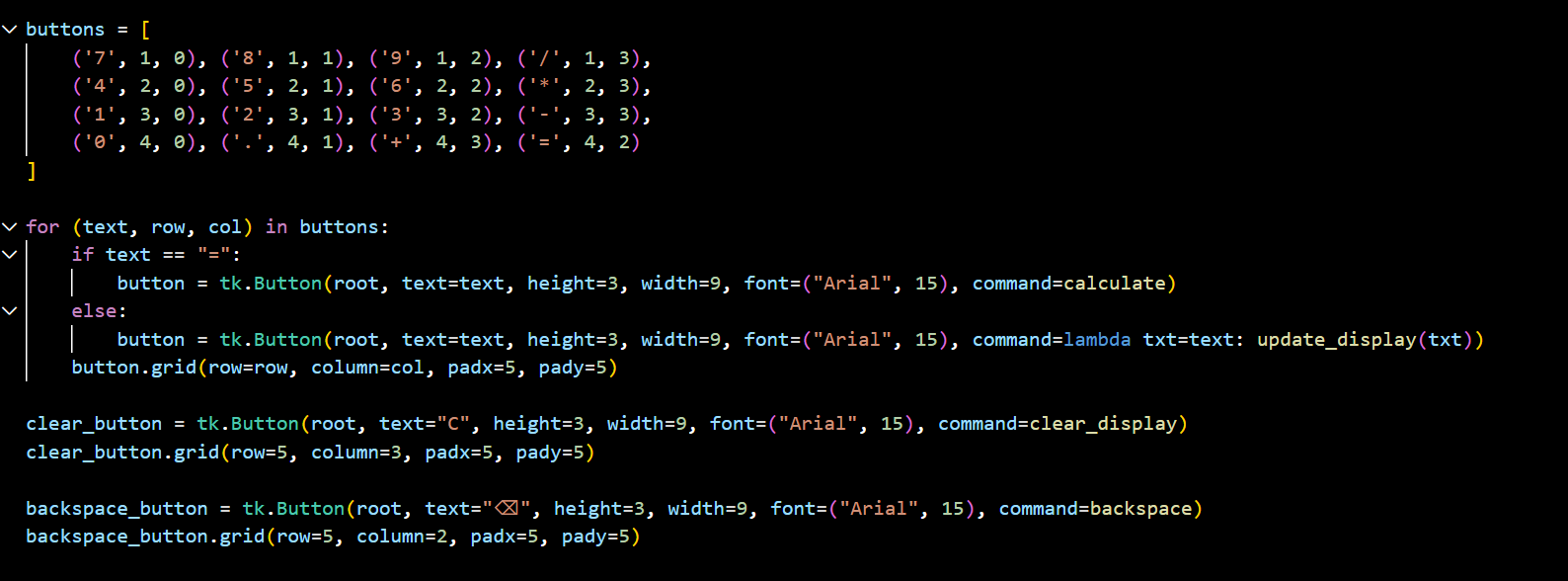
* **display**: A Tkinter Entry widget where the user’s input and results are shown.
* **grid()**: Places the Entry widget in the specified grid location.

**4. Functions**

****

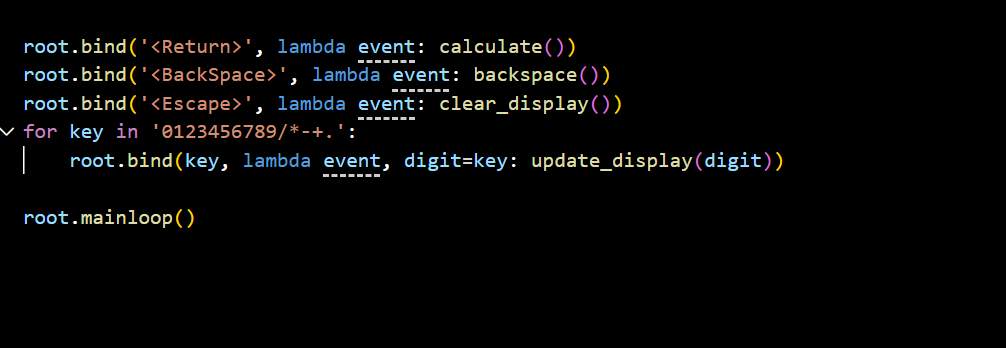
* ***update\_display(value)***
* Inserts a button or key input (like numbers and operators) into the display.
* ***clear\_display()***
* Clears the display content.
* ***backspace()***
* Deletes the last character in the display.
* ***calculate()***
  + Evaluates the expression in the display using eval.
  + Catches errors and displays an appropriate message, such as "Cannot divide by zero" for division errors or "Error" for other issues.

**5. Button Layout**

****

* 1. ***Button List***
     + The buttons list contains tuples representing the text, row, and column for each button in the calculator.
     + Each tuple defines:
       - **Text**: The symbol to display on the button.
       - **Row**: The grid row where the button will be placed.
       - **Column**: The grid column where the button will be placed.
     + This layout covers digits (0-9), operators (/, \*, -, +), the decimal point (.), and the equals button (=).
  2. ***Button creation loop***
* **Loop**: Iterates over each button defined in the buttons list.
  + - If the button text is "=", it assigns the calculate function to the button's command attribute, which will evaluate the expression when clicked.
    - For all other buttons, it assigns the update\_display function with the button’s text *as an argument using lambda txt=text: update\_display(txt).*
  1. ***Button Properties:***
     + **height**: Sets the height of each button.
     + **width**: Sets the width of each button.
     + **font**: Sets the font type and size.
     + **grid()**: Places each button in its respective row and column, with padding (padx and pady) for spacing.
  2. ***Special Button***
* **Clear Button**
* **Creates a clear button labeled "C", which clears the calculator’s display when clicked.**
* **command: Assigned to the clear\_display function, which clears all content in the display.**
* **grid(): Places this button in row 5, column 3, with padding.**
  + **Backspace Button**
  + **Creates a backspace button labeled with a backspace symbol ("⌫"), allowing users to delete the last character in the display.**
  + **command: Assigned to the backspace function, which removes the last character.**
  + **grid(): Places this button in row 5, column 2, with padding.**

**6. Keyboard Binidings**

****

* **bind(): Maps specific keys (like Enter, Backspace, Escape, and digits) to their corresponding functions.**
* **Enables users to perform calculations with keyboard inputs for convenience.**

**Error Handling**

* **ZeroDivisionError: If the user tries to divide by zero, an appropriate message ("Cannot divide by zero") is displayed.**
* **General Errors: For other invalid expressions, a generic "Error" message is shown.**

**Key Features**

* **Keyboard Support: Users can input numbers and operators via keyboard for a faster experience.**
* **Error Feedback: Clear error messages guide the user when incorrect inputs are entered.**
* **Backspace Button: Allows for quick corrections to input mistakes.**