



Calculator in Python

```
In [ ]: 1 # Simple Calculator in Python
2
3 This Python notebook demonstrates the implementation of a basic calculator using Python and the tkinter library.
4 The calculator provides a graphical user interface (GUI) where users can perform simple arithmetic operations such
5 addition, subtraction, multiplication, and division.
6
7 The notebook includes the code for creating the calculator GUI, handling user input, and displaying the result.
8 It utilizes the tkinter library for creating the window, buttons, and labels.
9
10 To use the calculator, run the code cells in the notebook. The calculator GUI will open, allowing you to click the
11 buttons to input numbers and perform operations. The result will be displayed on the screen.
12
13 Feel free to modify the code and experiment with different functionalities or design elements of the calculator.
14 It serves as a starting point for building more complex calculators or incorporating additional features.
15
```

```
In [ ]: 1 # importing tkinter library to create a window and to create buttons for numbers and all basic methametical operat
2 import tkinter as tk
3
4 # Append the clicked buttons values to the display label
5 def add_to_display(value):
6     current_value = display.get()
7     new_value = current_value + value
8     display.set(new_value)
9
10 # Evaluate the methametical expression in the display and update the result
11 def evaluate():
12     try:
13         result = eval(display.get())
14         display.set(result)
15     except Exception as e:
16         display.set("Error")
17
18 # Clear the display label
19 def clear_display():
20     display.set("")
21
22 # Create the main window
23 window = tk.Tk()
24 window.title("Calculator")
25
26 # Create the display
27 display = tk.StringVar()
28 display.set("")
29
30 display_label = tk.Label(window, textvariable=display, font=("Arial", 20), bd=5, relief="sunken")
31 display_label.grid(row=0, column=0, columnspan=4, padx=10, pady=10, sticky="w")
32
33 # Create the number buttons
34 for i in range(10):
35     button = tk.Button(window, text=str(i), font=("Arial", 16), width=5, command=lambda num=i: add_to_display(str(
36     button.grid(row=1 + (i // 3), column=i % 3, padx=5, pady=5)
37
38 # Create the operation buttons
39 operations = ["+", "-", "*", "/"]
40 for i, op in enumerate(operations):
41     button = tk.Button(window, text=op, font=("Arial", 16), width=5, command=lambda op=op: add_to_display(op))
42     button.grid(row=i + 2, column=3, padx=5, pady=5)
43
44 # Create the special buttons
45 clear_button = tk.Button(window, text="C", font=("Arial", 16), width=5, command=clear_display)
46 clear_button.grid(row=5, column=0, padx=5, pady=5)
47
48 equal_button = tk.Button(window, text="=", font=("Arial", 16), width=5, command=evaluate)
49 equal_button.grid(row=5, column=1, padx=5, pady=5)
50
51 # Start the GUI event loop
52 window.mainloop()
53
```