Project: Calculator

Software Design Document

Author: Paul Williams
Email: advanced_solution_delivery@outlook.com
Date: 2025-06-20
Version: 00.09.00

Overview

A project to build a basic calculator graphical user interface that works in a web browser. Part of the JavaScript Basics section of the Foundation Course for The Odin Project[1].

Detailed Fetures

The course expects development for the Google Chrome browser.

The project will use html for content, css for styling and javascript for functionality.

Specific design details pulled from the assignment[2] text:

numeric input

numeric display

operations:

- add
- subtract
- multiply
- divide
- equals
- clear

round answers with long decimals so that they don't overflow the display

handle divide by zero errors

consecutive operator presses should not run operations, only the last operation pressed should be used for the next operation

clear should wipe existing data not just clear display

> **"***When a result is displayed, pressing a new digit should clear the result and start a new calculation instead of appending the digit to the existing result.***"**

> *“Your calculator should not evaluate more than a single pair of numbers at a time. Example: you enter a number (12), followed by an operator button (+), a second number button (7), and a second operator button (−). Your calculator should then do the following: first, evaluate the initial pair of numbers (12 + 7), then display the result of that calculation (19). Finally, use that result (19) as the first number in a new calculation, along with the next operator (−). An example of the behaviour we’re looking for can be seen in this student’s calculator live preview.”*

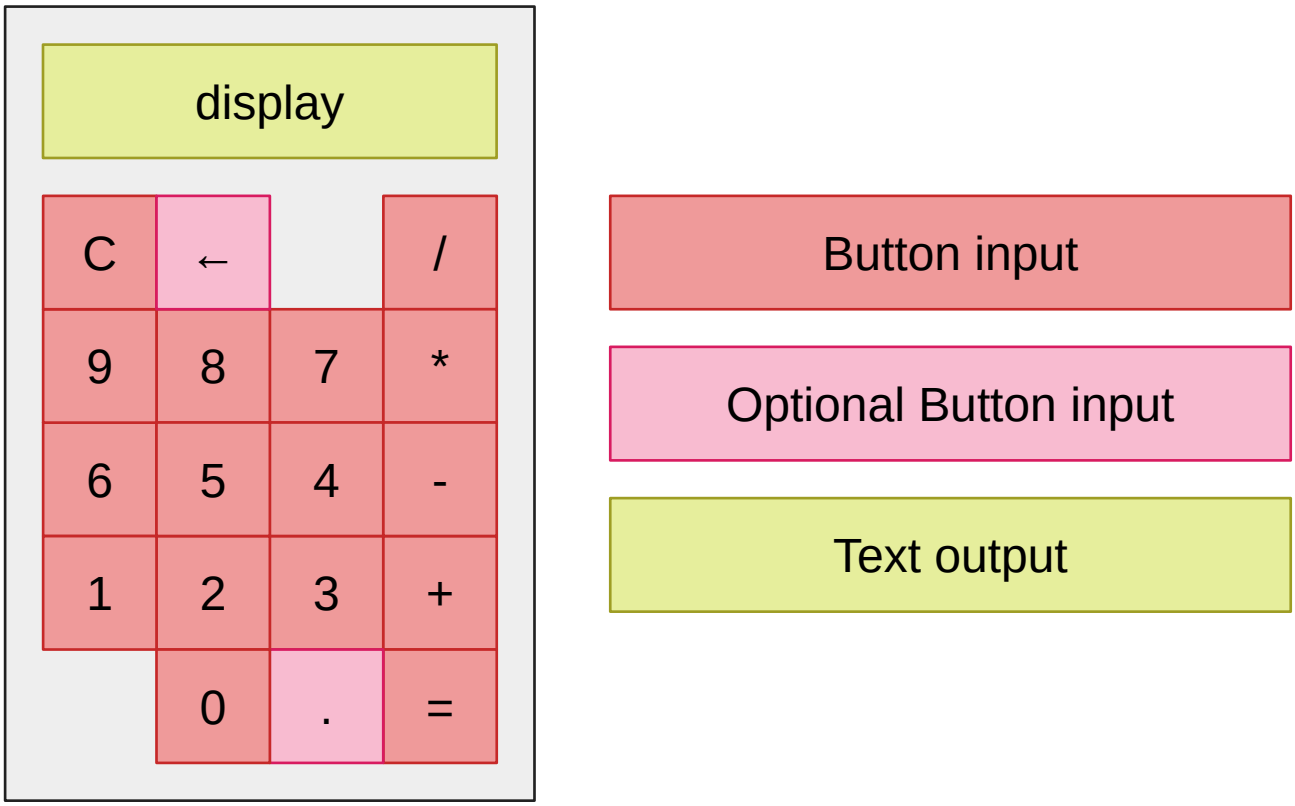
Optional Features

> *“Add a button and let users input decimals! Make sure you don’t let them type more than one though, like: 12.3.56.5. Disable the button if there’s already a decimal separator in the display.”*

> *“Add a “backspace” button, so the user can undo their last input if they click the wrong number.”*

> *“Add keyboard support!”*

Basic GUI Layout



Functionality

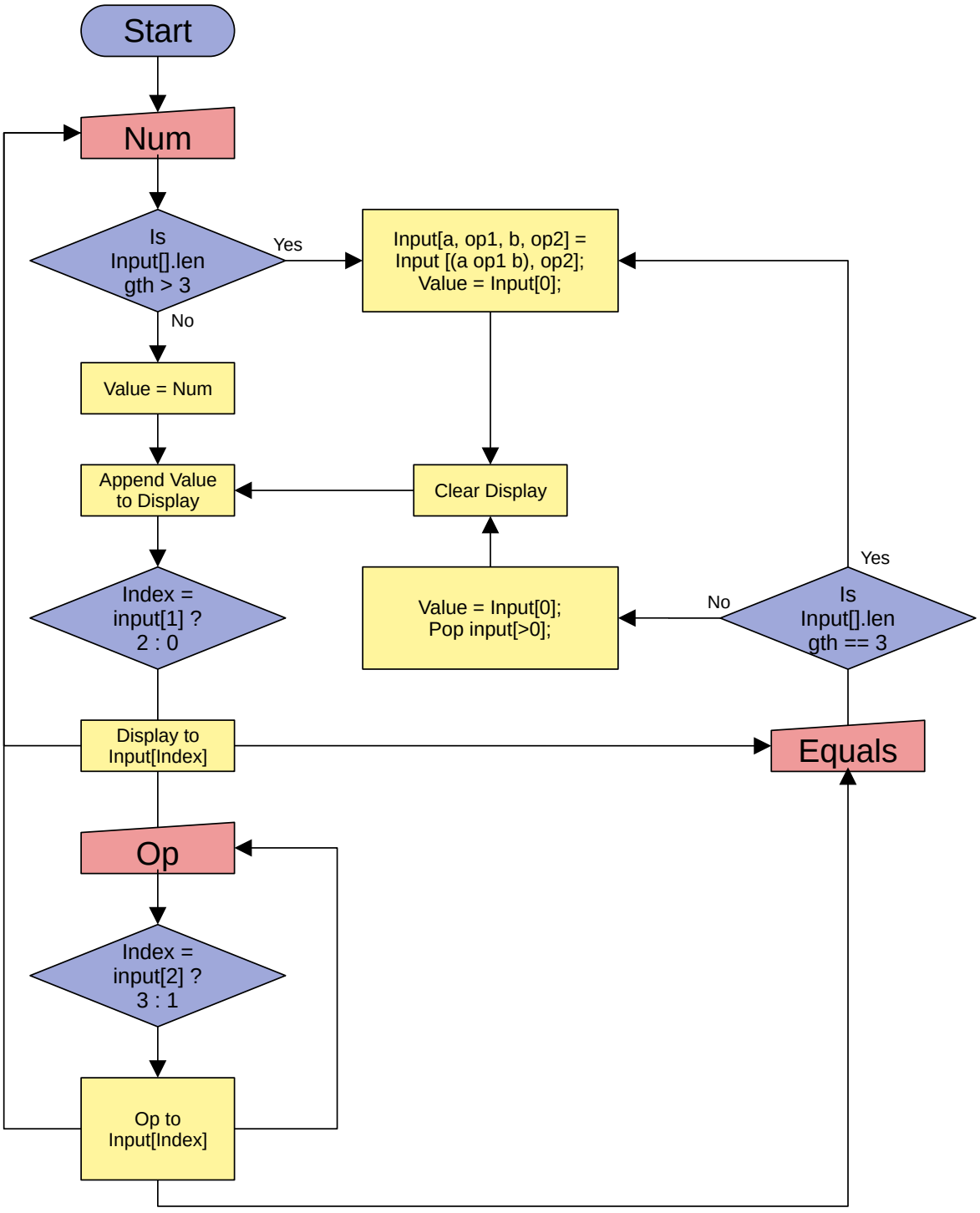
- numeric button inputs:
  - **IF** `<result>`:
    - clear display element text
  - append appropriate digit to display element text
- clear button input:
  - clear display element text
  - clears `<pendingOperation>`, `<operand1>`, `<operand2>` and `<result>` flag
- operation button inputs:
  - **IF** `<pendingOperation>`:
    - as equals button input
  - store display element text as a number in `<operand1>`
  - store operation relevant to button in `<pendingOperation>`

- equals button input:
  - **IF** `<pendingOperation>`:
    - store display element text as a number in `<operand2>`
    - **IF** `<pendingOperation>` is divide & `<operand2>` is 0
      - as clear button input
      - set display element text to “Err”
    - **ELSE**
      - perform `<pendingOperation>` on `<operand1>` and `<operand2>`
      - store result of operation in `<operand1>`
      - set element text to value in `<operand1>`
      - set `<result>` flag true
      - clear value in `<pendigOperation>`

Optional Functionality

- decimal input:
  - **IF** display element text does not contain point already:
    - append point to display element text
- delete input:
  - remove last character from display element text
- keyboard input:
  - setup new event listeners

Main Operational Flow



References

1 https://www.theodinproject.com/

2 https://www.theodinproject.com/lessons/foundations-calculator