

Simple Online Calculator

Naznin Nahar
ID: 20118

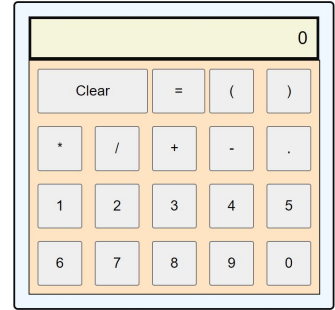


Table of Contents

- Introduction
- Calculator Design
- HTML Structure
- CSS Styling
- Javascript functionality
- Usage of eval() function
- Live Demo
- Conclusion
- References

Introduction

Calculator is a noun which simple means - something used for making mathematical calculations, in particular a small electronic device with a keyboard and a visual display.

In this simple online calculator, HTML has been used for structure, CSS for styling and javascript for functionalities.

You also can check it online - [Simple Calculator](#)

Calculator Design

- **Utilizing Flexbox for Responsive Design:** Flexbox allows for dynamic layout adjustment, ensuring optimal user experience.
- **Significance of Color Scheme in UX Design:** Choosing the right colors can evoke emotions, aid usability, and reinforce branding.
- **Button Arrangement for Intuitive Navigation:** Strategic button placement improves user flow and accelerates task completion.

HTML Structure

- **Semantic HTML Elements:** Utilizing semantic elements like <header>, <nav>, and <footer> enhances accessibility.
- **Importance of Meta Tags:** Including meta tags for SEO optimization improves website visibility and ranking.
- **Responsive Design Practices:** Implementing responsive design using media queries ensures compatibility across devices.

```
1 <html lang="en">
2 <head>
3   <style>
4     .clear-btn {
5     }
6   </style>
7 </head>
8 <body>
9   <div class="calculator">
10     <div id="display">0</div>
11     <table>
12       <tr>
13         <td colspan="2"><button class="clear-btn" onclick="clearDisplay()">Clear</button>
14         <td><button class="clear-btn" onclick="compute()">= </button></td>
15       </tr>
16       <tr>
17         <td><button onclick="appendToDisplay('(')">(</button></td>
18         <td><button onclick="appendToDisplay(')')">)</button></td>
19       </tr>
20       <tr>
21         <td><button onclick="appendToDisplay('*')">*</button></td>
22         <td><button onclick="appendToDisplay('/')">/</button></td>
23         <td><button onclick="appendToDisplay('+')">+</button></td>
24         <td><button onclick="appendToDisplay('-')">-</button></td>
25         <td><button onclick="appendToDisplay('.')">.</button></td>
26       </tr>
27       <tr>
28         <td><button onclick="appendToDisplay('1')">1</button></td>
29         <td><button onclick="appendToDisplay('2')">2</button></td>
30         <td><button onclick="appendToDisplay('3')">3</button></td>
31         <td><button onclick="appendToDisplay('4')">4</button></td>
32         <td><button onclick="appendToDisplay('5')">5</button></td>
33       </tr>
34       <tr>
35         <td><button onclick="appendToDisplay('6')">6</button></td>
36         <td><button onclick="appendToDisplay('7')">7</button></td>
37         <td><button onclick="appendToDisplay('8')">8</button></td>
```

CSS Styling

Styling the calculator for a better user interface.

Optimizing Layout Alignment with Flexbox: Using 'justify-content' and 'align-items' in flexbox ensures centered and organized design

•Enhancing Responsiveness for Diverse Devices:

Flexbox facilitates dynamic layout adjustments for seamless user experience across devices

•Improving User Experience through Design

Consistency: Consistent use of flexbox properties maintains design integrity and enhances usability

```
.html > .html > .body > .div.calculator > .table > .tr > .td
2  <html lang="en">
3  <head>
4  <style>
5    body {
6      height: 100vh;
7      margin: 0;
8      font-family: Arial, sans-serif;
9    }
10   table, tr {
11     background-color: #bisque ;
12     border: 1px solid #rgb(13, 13, 13);
13     padding: 1px;
14   }
15   .calculator {
16     background-color: #aliceblue;
17     width: 325px;
18     margin: 0 auto;
19     border: 2px solid #rgb(13, 13, 13);
20     border-radius: 5px;
21     padding: 15px;
22   }
23   #display {
24     background-color: #beige;
25     border: 3px solid #rgb(13, 13, 13);
26     text-align: right;
27     padding: 10px;
28     font-size: 20px;
29   }
30   button {
31     width: 50px;
32     height: 50px;
33     margin: 5px;
34     font-size: 16px;
35     cursor: pointer;
36   }
37   .clear-btn {
38     width: calc(100% - 0.5px);
39   }
40 </style>
```

Javascript Functionality

- **Event Handling in JavaScript:**
 - Append to display function works for the operators and numbers which we have on our calculator
 - Clear display function will clear the previous numbers on the display
 - Compute function calculates and uses eval() to show the result.
- **Description:** Adding interactivity to the calculator.

```
<script>
  const display = document.getElementById('display');

  function appendToDisplay(value) {
    if (display.innerText === '0') {
      display.innerText = value;
    } else {
      display.innerText += value;
    }
  }

  function clearDisplay() {
    display.innerText = '0';
  }

  function compute() {
    try {
      display.innerText = eval(display.innerText);
    } catch {
      display.innerText = 'Error';
    }
  }
</script>
</body>
</html>
```

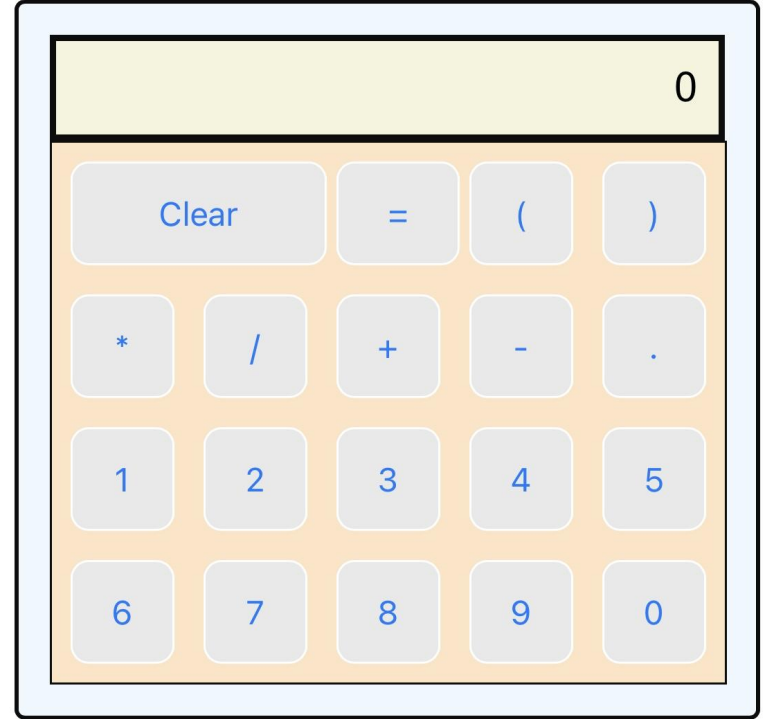
Usage of eval() Function

- **Function of eval() in JavaScript:** eval() evaluates and executes dynamic code strings, offering flexibility in scripting.
- **Pros and Cons of Using eval():** eval() allows for dynamic code execution but can introduce security risks if used improperly.
- **Alternative Approaches to eval():** Consider using safer methods like Function constructor or JSON.parse for dynamic code evaluation.

Live Demo

Live demo of the calculator can be found here-

[Simple calculator](#)



Conclusion

It is a simple calculator which provide addition, subtraction, multiplication, division functionality to user. For future work it has been planned to store data and provide them to user when it is needed as well as adding more functionalities.

References

https://www.w3schools.com/html/html_table_borders.asp

Github Link:

<https://github.com/NazninNaharSr/NazninNaharSr.github.io>