# CALCULATOR

## Introduction

This project is a simple basic calculator built using HTML, CSS, and JavaScript. The goal is to create a web-based calculator that performs basic arithmetic operations like addition, subtraction, multiplication, and division. We will break down each section of the project: HTML for structure, CSS for styling, and JavaScript for functionality.

## Key Takeaways

- The HTML provides the structure for the calculator, including the display and buttons.
- The CSS styles the calculator's appearance, making it more visually appealing and interactive.
- JavaScript handles the calculator's functionality, ensuring that inputs are processed and results are displayed.

## Step 1: Structure the HTML

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Basic Calculator</title>

    <link rel="stylesheet" href="styles.css">

</head>

<body>

         <h1><u> CLACULATER</u></h1>

    <div class="calculator">

        <input type="text" id="display" disabled style="background-color: #f4f4f4;">

 <div class="buttons">

            <button class="btn" onclick="appendToDisplay('7')">7</button>

            <button class="btn" onclick="appendToDisplay('8')">8</button>

            <button class="btn" onclick="appendToDisplay('9')">9</button>

            <button class="btn operator" onclick="appendToDisplay('+')">+</button>
           <button class="btn" onclick="appendToDisplay('4')">4</button>
            <button class="btn" onclick="appendToDisplay('5')">5</button>
            <button class="btn" onclick="appendToDisplay('6')">6</button>
            <button class="btn operator" onclick="appendToDisplay('-')">-</button>
           <button class="btn" onclick="appendToDisplay('1')">1</button>
            <button class="btn" onclick="appendToDisplay('2')">2</button>
            <button class="btn" onclick="appendToDisplay('3')">3</button>
            <button class="btn operator" onclick="appendToDisplay('*')">*</button>
           <button class="btn" onclick="appendToDisplay('0')">0</button>
            <button class="btn" onclick="clearDisplay()">C</button>
            <button class="btn" onclick="calculateResult()">=</button>
            <button class="btn operator" onclick="appendToDisplay('/')">/</button>
        </div>
    </div>
</body>
</html>
```

## Step 2: Style the Calculator (CSS)

```
<style>
      /* Basic Styling */
body {
   font-family: Arial, sans-serif;
   display: flex;
   justify-content: center;
   align-items: center;
   height: 100vh;
   background-color: #f4f4f4;
   margin: 0;
}
```

```
h1{
    position: relative;
    bottom: 250px;
    left: 250px;
}
/* Calculator Container */
.calculator {
    background-color: #3d0101;
    border-radius: 10px;
    box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1);
    overflow: hidden;
    width: 280px;
    padding: 20px;
    padding-right: 30px;
}
/* Display Styling */
#display {
    width: 100%;
    height: 50px;
    text-align: right;
    font-size: 2em;
    padding: 10px;
    border: 1px solid #ddd;
    border-radius: 5px;
    margin-bottom: 20px;
}
/* Buttons Grid */
.buttons {
    display: grid;
    grid-template-columns: repeat(4, 1fr);
    gap: 10px;
}
/* Button Styling */
button {
    font-size: 1.5em;
    padding: 20px;
    background-color: #f1f1f1;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    transition: background-color 0.3s ease;
}
button:hover {
    background-color: #e0e0e0;
}
.operator {
    background-color: rgb(230, 134, 134);
}
.operator:hover {
    background-color: #f66;
}
button:active {
    background-color: #ddd;
}
    </style>
```

## Step 3: Add JavaScript Functionality

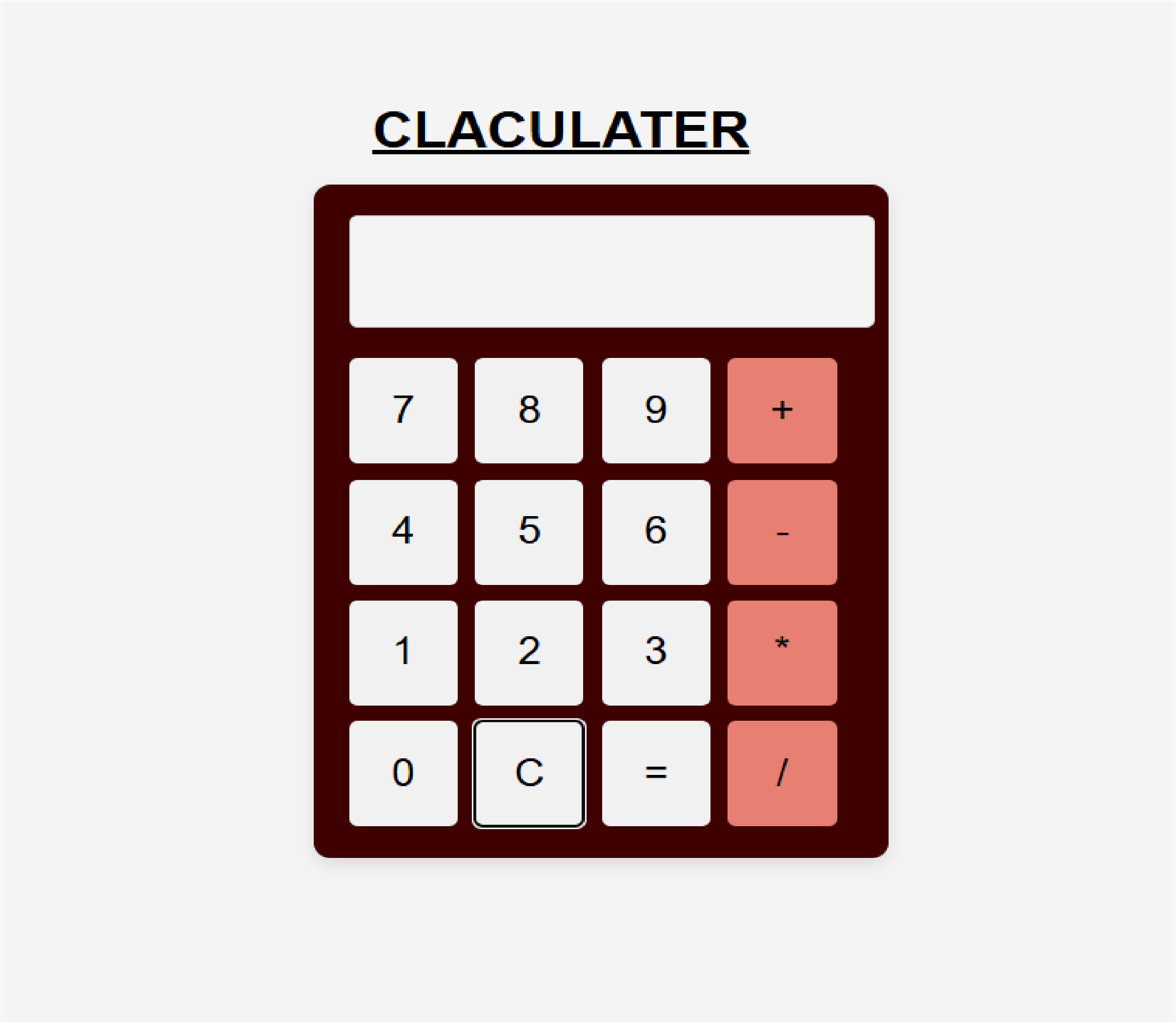
```
<script>
        // Function to append a number or operator to the display
```

```
function appendToDisplay(value) {
    document.getElementById('display').value += value;
}
// Function to clear the display
function clearDisplay() {
    document.getElementById('display').value = '';
}
// Function to calculate and display the result
function calculateResult() {
    const display = document.getElementById('display');
    try {
        // Evaluate the mathematical expression and display the result
        display.value = eval(display.value);
    }

    catch (error) {
        display.value = 'Error'; // If there's an error (like division by zero), show "Error"
    }
}

    </script>
```

## Execution Output



## Conclusion

This basic calculator project demonstrates the integration of HTML, CSS, and JavaScript. By using HTML for structure, CSS for style, and JavaScript for interactivity, we have created a simple yet functional calculator. The project teaches essential web development skills and can be extended further, such as by adding more complex operations (e.g., square roots, exponents) or improving error handling.