# Calculator Application Documentation

## Introduction

This document provides an overview and explanation of the Calculator application code.   
The application is designed to perform basic arithmetic operations and some unit conversions.   
It is built using HTML, CSS, and JavaScript, offering an interactive user interface.

## HTML Code

The HTML code defines the structure of the calculator application. It includes:  
- A text input field to display the entered values and results.  
- Buttons for arithmetic operations, number inputs, and unit conversions.  
- A script reference for JavaScript functionality and a link to the CSS file for styling.  
  
Key Elements:  
1. `display`: A disabled text field for showing calculations and results.  
2. Buttons with event handlers (`onclick`) to trigger corresponding JavaScript functions for functionality.  
3. Unit conversion buttons for converting values like kilometers to miles, minutes to seconds, and more.

HTML Code:

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <title>Calculator</title>  
 <link rel="stylesheet" href="style.css">  
</head>  
<body>  
 <div class="cont">  
 <div class="display">  
 <input type="text" placeholder="0" name="display" id="display" disabled>  
 </div>  
 <div class="buttonsCont">  
 <input type="button" value="AC" id="clear" onclick="clearDisplay()">  
 <input type="button" value="DEL" id="delete" onclick="deleteLast()">  
 <input type="button" value="/" class="input-button" onclick="appendToDisplay('/')">  
 <input type="button" value="\*" class="input-button" onclick="appendToDisplay('\*')">  
 <input type="button" value="7" class="input-button" onclick="appendToDisplay('7')">  
 ...  
 </div>  
 </div>  
 <script src="script.js"></script>  
</body>  
</html>

## JavaScript Code

The JavaScript file provides the logic for calculator operations. Key functions include:  
1. `appendToDisplay(value)`: Appends the clicked button's value to the display.  
2. `clearDisplay()`: Clears the display content.  
3. `deleteLast()`: Deletes the last entered character.  
4. `calculateResult()`: Evaluates the entered mathematical expression.  
5. Conversion functions (e.g., `convertKilometersToMiles`): Perform specific unit conversions and display results.

JavaScript Code:

function appendToDisplay(value) {  
 display.value += value;  
}  
function clearDisplay() {  
 display.value = "";  
}  
function deleteLast() {  
 display.value = display.value.slice(0, -1);   
}  
function calculateResult() {  
 try {  
 display.value = eval(display.value);  
 } catch(error) {  
 display.value = "error";  
 }   
}  
function convertKilometersToMiles() {  
 let num = parseFloat(display.value);  
 if(!isNaN(num)) {  
 display.value = num \* 0.621371 + " miles";  
 } else {  
 display.value = "error";  
 }  
}  
...

## CSS Code

The CSS file is responsible for styling the calculator, ensuring a clean and responsive design.   
Key features include:  
1. Dark theme with styled buttons and display field.  
2. Grid layout for organizing buttons.  
3. Hover effects for interactive feedback.

CSS Code:

body {  
 background-color: rgba(0, 0, 0, 0.91);  
}  
.cont {  
 width: 400px;  
 height: 610px;  
 background-color: rgb(42, 41, 41);  
 display: flex;  
 flex-direction: column;  
 align-items: center;  
 border-radius: 10px;  
 margin: 30px auto;  
 box-shadow: 4px 4px 15px rgb(32, 32, 32);  
}  
...