For this Lab, you will add on to the calculator by adding back-end functionality through JavaScript.

You will need to start by creating a new JavaScript file named Calculator.js. You will then need to do the following inside of that source file:

First, you will need to get all of the keys from the document:

// Get all the keys from document

var keys = document.querySelectorAll('#calculator span');

var operators = ['+', '-', 'x', '÷'];

var decimalAdded = false;

Then, you will need to add the following code to add functionality to all of the buttons:

// Add onclick event to all the keys and perform operations

for(var i = 0; i < keys.length; i++)

{

keys[i].onclick = function(e)

{

// Get the input and button values

var input = document.querySelector('.screen');

var inputVal = input.innerHTML;

var btnVal = this.innerHTML;

// Now, just append the key values (btnValue) to the input string and finally use javascript's eval function to get the result

// If clear key is pressed, erase everything

if(btnVal == 'C') {

input.innerHTML = '';

decimalAdded = false;

}

// If eval key is pressed, calculate and display the result

else if(btnVal == '=')

{

var equation = inputVal;

var lastChar = equation[equation.length - 1];

// Replace all instances of x and ÷ with \* and / respectively. This can be done easily using regex and the 'g' tag which will replace all instances of the matched character/substring

equation = equation.replace(/x/g, '\*').replace(/÷/g, '/');

// Final thing left to do is checking the last character of the equation. If it's an operator or a decimal, remove it

if(operators.indexOf(lastChar) > -1 || lastChar == '.')

equation = equation.replace(/.$/, '');

if(equation)

input.innerHTML = eval(equation);

decimalAdded = false;

}

// Basic functionality of the calculator is complete. But there are some problems like

// 1. No two operators should be added consecutively.

// 2. The equation shouldn't start from an operator except minus

// 3. not more than 1 decimal should be there in a number

// We'll fix these issues using some simple checks

// indexOf works only in IE9+

else if(operators.indexOf(btnVal) > -1)

{

// Operator is clicked

// Get the last character from the equation

var lastChar = inputVal[inputVal.length - 1];

// Only add operator if input is not empty and there is no operator at the last

if(inputVal != '' && operators.indexOf(lastChar) == -1)

input.innerHTML += btnVal;

// Allow minus if the string is empty

else if(inputVal == '' && btnVal == '-')

input.innerHTML += btnVal;

// Replace the last operator (if exists) with the newly pressed operator

if(operators.indexOf(lastChar) > -1 && inputVal.length > 1)

{

// Here, '.' matches any character while $ denotes the end of string, so anything (will be an operator in this case) at the end of string will get replaced by new operator

input.innerHTML = inputVal.replace(/.$/, btnVal);

}

decimalAdded =false;

}

// Now only the decimal problem is left. We can solve it easily using a flag 'decimalAdded' which we'll set once the decimal is added and prevent more decimals to be added once it's set. It will be reset when an operator, eval or clear key is pressed.

else if(btnVal == '.')

{

if(!decimalAdded)

{

input.innerHTML += btnVal;

decimalAdded = true;

}

}

// if any other key is pressed, just append it

else {

input.innerHTML += btnVal;

}

// prevent page jumps

e.preventDefault();

} // end function

} // end for loop

Once you have saved your JavaScript file, then add the following line to the HTML file you created for the Calculator before the last closing <body> tag:

<script src="Calculator.js" type="text/javascript" type="text/javascript"></script>

Be sure that you also update the reference to the external .css file in your HTML file.

The final product of the Calculator in the website should look like this:

 