Module 4 - Calculator Assignment - Model Answer

```
<!--calculator.html-->

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Calculator</title>
    <link rel="stylesheet" href="./css/calculator.css">
</head>

<body>
    <div class="calculator">
        <input type="text" class="calculator-screen" value="0" disabled>
        <div class="calculator-keys">

            <button type="button" class="operator" value="+">+</button>
            <button type="button" class="operator" value="-">-</button>
            <button type="button" class="operator" value="*">&times;</button>
            <button type="button" class="operator" value="/">&divide;</button>

            <button type="button" value="7">7</button>
            <button type="button" value="8">8</button>
            <button type="button" value="9">9</button>

            <button type="button" value="4">4</button>
            <button type="button" value="5">5</button>
            <button type="button" value="6">6</button>

            <button type="button" value="1">1</button>
            <button type="button" value="2">2</button>
            <button type="button" value="3">3</button>

            <button type="button" value="0">0</button>
            <button type="button" class="decimal function" value=".">.</button>
            <button type="button" class="all-clear function" value="all-clear">AC</button>

            <button type="button" class="equal-sign operator" value="=">=</button>

        </div>
    </div>
    <script src="./js/calculator.js"></script>
</body>

</html>
```

```css
/*calculator.css*/

html {
    font-size: 70%;
    box-sizing: border-box;
}

*,
*::before,
*::after {
    margin: 0;
    padding: 0;
    box-sizing: inherit;
}

.calculator {
    border: 1.2px solid darkslategray;
    border-radius: 7px;
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
    width: 500px;
}

.calculator-screen {
    width: 100%;
    font-size: 5rem;
    height: 80px;
    border: none;
    background-color: black;
    color: white;
    text-align: right;
    padding-right: 20px;
    padding-left: 20px;
    border-radius: 7px 7px 0px 0px;
}

button {
    height: 70px;
    border-radius: 3px;
    border: 1px solid darkslategray;
    background-color: transparent;
    font-size: 3.2rem;
    color: black;
    background-image: linear-gradient(to bottom, transparent, transparent 50%, rgba(0, 0, 0, 0.6));
    text-shadow: 0 1px rgba(255, 255, 255, .4);
    box-shadow: inset 0 0 0 1px rgba(255, 255, 255, .05),
        inset 0 1px 0 0 rgba(255, 255, 255, .05),
        inset 0 -1px 0 0 rgba(255, 255, 255, .05),
        0 1px 0 0 rgba(255, 255, 255, .05);
}

button:hover {
    background-color: #e1e1e1;
}

.operator {
    color: black;
    background-image: linear-gradient(to bottom, transparent, transparent 50%, rgba(0, 0, 0, 0.5));
    background-color: lightgray;
}
```

```css
.all-clear {
    background-color: red;
    border-color: darkslategray;
    color: #ffffff;
}

.equal-sign {
    background-color: blue;
    border-color: darkslategray;
    color: #fff;
    height: 100%;
    grid-area: 2/4/6/5;
}

.equal-sign:hover {
    background-color: darkblue;
}

.calculator-keys {
    display: grid;
    grid-template-columns: repeat(4, 1fr);
    grid-gap: 20px;
    padding: 20px;
}
```

```js
//calculator.js

//Creates an object  to keep track of values
const Calculator = {
    Display_Value: '0',
    First_Operand: null,
    Wait_Second_Operand: false,
    operator: null,
};

//This modifies values each time a button is clicked
function Input_Digit(digit) {
    const { Display_Value, Wait_Second_Operand } = Calculator;
    //Checks if the Wait_Second_Operand variable is true and sets
    //Display_Value to the key that was clicked on
    if (Wait_Second_Operand === true) {
        Calculator.Display_Value = digit;
        Calculator.Wait_Second_Operand = false;
    } else {
        //This overwrites Display_Value if the current value is 0. Otherwise,
        //it concatenates the value
        Calculator.Display_Value = Display_Value === '0' ? digit : Display_Value + digit;
    }
}

//This section handles decimal points
function Input_Decimal(dot) {
    if (Calculator.Wait_Second_Operand === 'true') return;
    if (!Calculator.Display_Value.includes(dot)) {
        Calculator.Display_Value += dot;
    }
}
```

```
//This section handles operators
function Handle_Operator(Next_Operator) {
    const { First_Operand, Display_Value, operator } = Calculator;
    const Value_of_Input = parseFloat(Display_Value);
    //Checks if an operator exists and if Wait_Second_Operand is true
    //Then updates the operator and exits the function
    if (operator && Calculator.Wait_Second_Operand) {
        Calculator.operator = Next_Operator;
        return;
    }
    if (First_Operand == null) {
        Calculator.First_Operand = Value_of_Input;
    } else if (operator) {
        const Value_Now = First_Operand || 0;
        let result = Perform_Calculation[operator](Value_Now, Value_of_Input);
        result = Number(result).toFixed(9);
        //This will remove any trailing 0s
        result = (result * 1).toString();
        Calculator.Display_Value = result;
        Calculator.First_Operand = result;
    }
    Calculator.Wait_Second_Operand = true;
    Calculator.operator = Next_Operator;
}

//Handles the actual calculations
const Perform_Calculation = {
    '/': (First_Operand, Second_Operand) => First_Operand / Second_Operand,
    '*': (First_Operand, Second_Operand) => First_Operand * Second_Operand,
    '+': (First_Operand, Second_Operand) => First_Operand + Second_Operand,
    '-': (First_Operand, Second_Operand) => First_Operand - Second_Operand,
    '=': (First_Operand, Second_Operand) => Second_Operand
};

//Resets the calculator when the AC button is clicked
function Calculator_Reset() {
    Calculator.Display_Value = '0';
    Calculator.First_Operand = null;
    Calculator.Wait_Second_Operand = false;
    Calculator.operator = null;
}
```

```
//This function updates the calculator screen with the contents of Display Value
function Update_Display() {
    const display = document.querySelector('.calculator-screen');
    display.value = Calculator.Display_Value;
}

Update_Display();

//This section monitors button clicks
const keys = document.querySelector('.calculator-keys');
keys.addEventListener('click', (event) => {
    const { target } = event;
    if (!target.matches('button')) {
        return;
    }
    if (target.classList.contains('operator')) {
        Handle_Operator(target.value);
        Update_Display();
        return
    }
    if (target.classList.contains('decimal')) {
        Input_Decimal(target.value);
        Update_Display();
        return;
    }
    if (target.classList.contains('all-clear')) {
        Calculator_Reset();
        Update_Display();
        return;
    }
    Input_Digit(target.value);
    Update_Display();
})
```