Experiment-06

Apply HTML, CSS and JavaScript to design a simple calculator to perform the following operations: sum,product, difference, remainder, quotient, power, square-root and square.

Program:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Simple Calculator</title>
   <style>
      body {
         font-family: Arial, sans-serif;
         display: flex;
         justify-content: center;
         align-items: center;
         height: 100vh;
         margin: 0;
```

```
    background-color: #f0f0f0;
}
.calculator {
    background-color: #fff;
    border-radius: 8px;
    box-shadow: 0 0 10px rgba(0,0,0,0.1);
    padding: 20px;
    width: 300px;
}
#display {
    width: 100%;
    height: 40px;
    font-size: 1.5em;
    text-align: right;
    margin-bottom: 10px;
    padding: 5px;
    box-sizing: border-box;
}
.buttons {
```

```
    display: grid;
    grid-template-columns: repeat(4, 1fr);
    gap: 10px;
}
button {
    padding: 10px;
    font-size: 1.2em;
    border: none;
    background-color: #e0e0e0;
    cursor: pointer;
    border-radius: 4px;
}
button:hover {
    background-color: #d0d0d0;
}
.operator {
    background-color: #f0a030;
    color: white;
}
```

```
        .operator:hover {
            background-color: #e09020;
        }
    </style>
</head>
<body>
    <div class="calculator">
        <input type="text" id="display" readonly>
        <div class="buttons">
            <button onclick="appendToDisplay('7')">7</button>
            <button onclick="appendToDisplay('8')">8</button>
            <button onclick="appendToDisplay('9')">9</button>
            <button class="operator"
onclick="setOperation('+')">&plus;</button>
            <button onclick="appendToDisplay('4')">4</button>
            <button onclick="appendToDisplay('5')">5</button>
            <button onclick="appendToDisplay('6')">6</button>
            <button class="operator" onclick="setOperation('-
')">&minus;</button>
            <button onclick="appendToDisplay('1')">1</button>
```

```
        <button onclick="appendToDisplay('2')">2</button>

        <button onclick="appendToDisplay('3')">3</button>

        <button class="operator"
onclick="setOperation('*')">&times;</button>

        <button onclick="appendToDisplay('0')">0</button>

        <button onclick="appendToDisplay('.')">.</button>

        <button class="operator" onclick="calculate()">&equals;</button>

        <button class="operator"
onclick="setOperation('/')">&divide;</button>

        <button class="operator" onclick="setOperation('%')">%</button>

        <button class="operator"
onclick="setOperation('^')">x<sup>y</sup></button>

        <button class="operator" onclick="squareRoot()">√</button>

        <button class="operator"
onclick="square()">x<sup>2</sup></button>

        <button onclick="clearDisplay()">C</button>

    </div>

  </div>


  <script>
```

```
let display = document.getElementById('display');

let currentValue = '';

let operation = '';

let previousValue = '';


function appendToDisplay(value) {

    currentValue += value;

    display.value = currentValue;

}


function clearDisplay() {

    currentValue = '';

    operation = '';

    previousValue = '';

    display.value = '';

}


function setOperation(op) {

    if (currentValue !== '') {
```

```
        if (previousValue !== '') {
            calculate();
        }
        operation = op;
        previousValue = currentValue;
        currentValue = '';
    }
}

function calculate() {
    if (previousValue !== '' && currentValue !== '') {
        let result;
        const prev = parseFloat(previousValue);
        const current = parseFloat(currentValue);
        switch(operation) {
            case '+':
                result = prev + current;
                break;
            case '-':
```

```
            result = prev - current;
            break;
        case '*':
            result = prev * current;
            break;
        case '/':
            result = prev / current;
            break;
        case '%':
            result = prev % current;
            break;
        case '^':
            result = Math.pow(prev, current);
            break;
    }
    display.value = result;
    previousValue = result.toString();
    currentValue = '';
    operation = '';
```

```
    }
  }

  function squareRoot() {
    if (currentValue !== '') {
      const result = Math.sqrt(parseFloat(currentValue));
      display.value = result;
      currentValue = result.toString();
    }
  }

  function square() {
    if (currentValue !== '') {
      const result = Math.pow(parseFloat(currentValue), 2);
      display.value = result;
      currentValue = result.toString();
    }
  }
</script>
```

```
</body>

</html>
```

Explanation

Step 1: Document Structure

The document uses the standard HTML5 structure.

The <head> section contains metadata, title, and embedded CSS styles.

The <body> contains the calculator interface and JavaScript code.

Step 2: Styling (CSS)

The calculator is centered on the page using flexbox.

The calculator has a white background with rounded corners and a shadow.

Buttons are arranged in a grid layout.

Different styles are applied to number buttons and operator buttons.

Step 3: HTML Structure

The calculator is contained in a div with class "calculator".

It has an input field for display and a div for buttons.

Buttons are created for numbers 0-9, decimal point, operators (+, -, *, /), equals, clear, and additional functions (%, ^, √, x^2).

Step 4: JavaScript Functionality

The script defines several functions:

appendToDisplay(): Adds numbers and decimal point to the display.

clearDisplay(): Clears the calculator.

setOperation(): Sets the current operation.

calculate(): Performs the calculation based on the set operation.

squareRoot(): Calculates the square root of the current value.

square(): Calculates the square of the current value.

Step 5: Calculator Logic

The calculator uses three main variables: currentValue, previousValue, and operation.

When a number is clicked, it's appended to the currentValue.

When an operation is clicked, it stores the currentValue as previousValue and sets the operation.

The equals button triggers the calculate() function, which performs the operation and displays the result.

Step 6: Advanced Operations

The calculator includes modulo (%), exponentiation (^), square root (√), and square (x^2) operations.

These operations are handled in the calculate() function or as separate functions (squareRoot() and square()).

Key Features:

Basic arithmetic operations: addition, subtraction, multiplication, division.

Advanced operations: modulo, exponentiation, square root, square.

Clear button to reset the calculator.

Responsive design that works on various screen sizes.

Visual feedback with button hover effects.

This calculator demonstrates several important web development concepts:

DOM manipulation using JavaScript.

Event handling for button clicks.

Use of CSS Grid for layout.

Implementing mathematical operations in JavaScript.

Creating a responsive user interface.