

## 6th Programm

```
<!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;
    }
  </style>
</head>
<body>
  <div class="calculator">
    <div id="display">
    </div>
    <div class="buttons">
    </div>
  </div>
</body>
</html>
```

```

    }
    .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('+')">+</button>

            <button onclick="appendToDisplay('4')">4</button>
            <button onclick="appendToDisplay('5')">5</button>
            <button onclick="appendToDisplay('6')">6</button>
            <button class="operator" onclick="setOperation('-')">-</button>

            <button onclick="appendToDisplay('1')">1</button>
            <button onclick="appendToDisplay('2')">2</button>
            <button onclick="appendToDisplay('3')">3</button>
            <button class="operator" onclick="setOperation('*')">x</button>

            <button onclick="appendToDisplay('0')">0</button>
            <button onclick="appendToDisplay('.')">.</button>
            <button class="operator" onclick="calculate()">=</button>
            <button class="operator" onclick="setOperation('/')">÷</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 to append values to the display
function appendToDisplay(value) {
    currentValue += value;
    display.value = currentValue;
}

// Function to clear the display
function clearDisplay() {
    currentValue = '';
    operation = '';
    previousValue = '';
    display.value = '';
}

// Function to set the operation (i.e., +, -, *, /, %, ^)
function setOperation(op) {
    if (currentValue !== '') {
        if (previousValue !== '') {
            calculate(); // Calculate the result if there's already a
previous value
        }
        operation = op;
        previousValue = currentValue;
        currentValue = ''; // Clear current value for the next number
    }
}

// Function to perform the calculation based on the selected operation
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 = ''; // Reset current value after calculation
    operation = ''; // Reset operation after calculation
}

// Function to calculate the square root of the current value
function squareRoot() {
    if (currentValue !== '') {
        const result = Math.sqrt(parseFloat(currentValue));
        display.value = result;
        currentValue = result.toString(); // Update current value with
the result
    }
}

// Function to calculate the square of the current value
function square() {
    if (currentValue !== '') {
        const result = Math.pow(parseFloat(currentValue), 2);
        display.value = result;
        currentValue = result.toString(); // Update current value with
the result
    }
}
</script>
</body>
</html>

```

## 7th Programm

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>JSON/CSV Converter and Hash Generator</title>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/crypto-
js/4.1.1/crypto-js.min.js"></script>
  <style>
    body {
      font-family: Arial, sans-serif;
      line-height: 1.6;
      margin: 0;
      padding: 20px;
      background-color: #f4f4f4;
    }
    .container {
      max-width: 800px;
      margin: auto;
      background: white;
      padding: 20px;
      border-radius: 5px;
      box-shadow: 0 0 10px rgba(0,0,0,0.1);
    }
    h1 {
      color: #333;
    }
    textarea {
      width: 100%;
      height: 100px;
      margin-bottom: 10px;
    }
    button {
      background-color: #4CAF50;
      color: white;
      padding: 10px 15px;
      border: none;
      border-radius: 4px;
      cursor: pointer;
      margin-right: 10px;
    }
    button:hover {
      background-color: #45a049;
    }
    #result {
      margin-top: 20px;
      padding: 10px;
```

```

        background-color: #e7e7e7;
        border-radius: 4px;
    }
</style>
</head>
<body>
    <div class="container">
        <h1>JSON/CSV Converter and Hash Generator</h1>

        <h2>a) Convert JSON to JavaScript Object</h2>
        <textarea id="jsonInput" placeholder="Enter JSON here"></textarea>
        <button onclick="convertJsonToObject()">Convert to Object</button>

        <h2>b) Convert JSON to Date</h2>
        <textarea id="jsonDateInput" placeholder="Enter JSON date string
(e.g., {"date": "2023-05-15T12:00:00Z"})"></textarea>
        <button onclick="convertJsonToDate()">Convert to Date</button>

        <h2>c) Convert JSON to CSV and CSV to JSON</h2>
        <textarea id="dataInput" placeholder="Enter JSON or CSV
here"></textarea>
        <button onclick="convertJsonToCsv()">JSON to CSV</button>
        <button onclick="convertCsvToJson()">CSV to JSON</button>

        <h2>d) Create Hash from String</h2>
        <textarea id="hashInput" placeholder="Enter string to
hash"></textarea>
        <button onclick="createHash()">Generate Hash</button>

        <div id="result"></div>
    </div>

    <script>
        function convertJsonToObject() {
            try {
                const jsonInput = document.getElementById('jsonInput').value;
                const jsObject = JSON.parse(jsonInput);
                document.getElementById('result').innerText = 'Converted
Object: ' + JSON.stringify(jsObject, null, 2);
            } catch (error) {
                document.getElementById('result').innerText = 'Error: ' +
error.message;
            }
        }

        function convertJsonToDate() {
            try {

```

```

        const jsonInput =
document.getElementById('jsonDataInput').value;
        const jsObject = JSON.parse(jsonInput);
        const date = new Date(jsObject.date);
        document.getElementById('result').innerText = 'Converted Date:
' + date.toString();
    } catch (error) {
        document.getElementById('result').innerText = 'Error: ' +
error.message;
    }
}

function convertJsonToCsv() {
    try {
        const jsonInput = document.getElementById('dataInput').value;
        const jsObject = JSON.parse(jsonInput);
        if (Array.isArray(jsObject) && jsObject.length > 0) {
            const headers = Object.keys(jsObject[0]);
            const csvRows = [
                headers.join(','),
                ...jsObject.map(row => headers.map(fieldName =>
                    JSON.stringify(row[fieldName])).join(','))
            ];
            const csvString = csvRows.join('\n');
            document.getElementById('result').innerText = 'Converted
CSV:\n' + csvString;
        } else {
            document.getElementById('result').innerText = 'Error:
Input is not a valid array of JSON objects.';
        }
    } catch (error) {
        document.getElementById('result').innerText = 'Error: ' +
error.message;
    }
}

function convertCsvToJson() {
    try {
        const csvInput = document.getElementById('dataInput').value;
        const lines = csvInput.split('\n');
        const headers = lines[0].split(',');
        const jsonArray = lines.slice(1).map(line => {
            const values = line.split(',');
            return headers.reduce((obj, header, index) => {
                obj[header] = values[index];
                return obj;
            }, {});
        });
    }
}

```

```
        document.getElementById('result').innerText = 'Converted
JSON:\n' + JSON.stringify(jsonArray, null, 2);
    } catch (error) {
        document.getElementById('result').innerText = 'Error: ' +
error.message;
    }
}

function createHash() {
    try {
        const input = document.getElementById('hashInput').value;
        const hash =
CryptoJS.SHA256(input).toString(CryptoJS.enc.Base64); // Convert hash to a
readable format
        document.getElementById('result').innerText = 'Generated Hash
(SHA-256): ' + hash;
    } catch (error) {
        document.getElementById('result').innerText = 'Error: ' +
error.message;
    }
}
</script>
</body>
</html>
```



## 8th Programm (track.php)

```
<?php
$counterFile = "counter.txt";
if (!file_exists($counterFile)) {
    file_put_contents($counterFile, "0");
}
$currentCount = file_get_contents($counterFile);
$newCount = $currentCount + 1;
file_put_contents($counterFile, $newCount);
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Visitor Counter | vtucode</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            text-align: center;
            margin: 0;
            padding: 0;
            display: flex;
            flex-direction: column;
            justify-content: center;
            height: 100vh;
            background-color: #f4f4f9;
            color: #333;
        }
        .container {
            background: #fff;
            padding: 20px;
            box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
            border-radius: 8px;
            margin: 0 auto;
            width: 60%;
        }
        h1 {
            font-size: 2.5em;
            margin: 0;
        }
        p {
            font-size: 1.2em;
            color: #555;
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>Welcome to Our Website!</h1>
```

```

    <p>You are visitor number: <strong><?php echo $newCount;
?></strong></p>
  </div>
</body>
</html>

```

## 9 th Programm

```

<!DOCTYPE html>

<head>
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  <title>jQuery Example</title>
  <style>
    body {
      font-family: 'Roboto', sans-serif;
      background-color: #f4f7f6;
      margin: 0;
      padding: 0;
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
    }

    .container {
      text-align: center;
      background: #fff;
      padding: 30px;
      border-radius: 12px;
      box-shadow: 0 4px 20px rgba(0, 0, 0, 0.1);
      transition: transform 0.3s ease-in-out;
    }

    .container:hover {
      transform: scale(1.02);
    }

    #paragraph {
      margin-bottom: 20px;
      color: #333;
      font-size: 18px;
      line-height: 1.5;
    }

    #list {
      margin-bottom: 20px;

```

```
    list-style: none;
    padding: 0;
}

#list li {
    background: #e8f0fe;
    margin: 5px 0;
    padding: 10px;
    border-radius: 8px;
    transition: background 0.3s;
}

#list li:hover {
    background: #d0e2fe;
}

.box {
    padding: 0 10px;
    width: 100px;
    height: 100px;
    background-color: #007bff;
    margin: 20px auto;
    line-height: 100px;
    color: white;
    text-align: center;
    border-radius: 8px;
    transition: all 0.3s ease;
}

button {
    padding: 12px 24px;
    margin: 10px;
    cursor: pointer;
    border: none;
    border-radius: 6px;
    font-size: 16px;
    background: #007bff;
    color: white;
    transition: box-shadow 0.3s, transform 0.2s;
    box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1);
}

button:hover {
    box-shadow: 0 0 0 2px #fff, 0 0 0 4px #007bff;
}

button:focus {
    box-shadow: 0 0 0 2px #fff, 0 0 0 4px #007bff;
}
```

```

    }

button:active {
    background: #004494;
    transform: translateY(0);
}
</style>
</head>

<body>
<div class="container">
<p id="paragraph">This is an existing paragraph.</p>
<ul id="list">
    <li>List item 1</li>
    <li>List item 2</li>
</ul>
<div class="box" id="box">Animate me!</div>
<button id="appendButton">Append Content</button>
<button id="animateButton">Animate Box</button>
</div>

<script>
$(document).ready(function () {

    $("#appendButton").click(function () {
        $("#paragraph").append(" Appended text.");
        $("#list").append("<li>New appended list item</li>");
    });

    $("#animateButton").click(function () {

        $("#box").stop(true, true).css({
            width: "100px",
            height: "100px",
            opacity: 1,
            backgroundColor: "blue"
        }).animate({
            width: "200px",
            height: "200px",
            opacity: 0.5
        }, 1000, function () {

            $(this).css("background-color", "green");
        });
    });
});

</script></body>

```

## 10 th Programm

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Ajax Demo Program</title>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.0/jquery.min.js"></scri
pt>
  <style>
    body {
      font-family: Arial, sans-serif;
      line-height: 1.6;
      margin: 0;
      padding: 20px;
      background-color: #f4f4f4;
    }
    .container {
      max-width: 800px;
      margin: auto;
      background: white;
      padding: 20px;
      border-radius: 5px;
      box-shadow: 0 0 10px rgba(0,0,0,0.1);
    }
    h1 {
      color: #333;
    }
    h2 {
      color: #666;
    }
    button {
      background-color: #4CAF50;
      border: none;
      color: white;
      padding: 10px 20px;
      text-align: center;
      text-decoration: none;
      display: inline-block;
      font-size: 16px;
```

```

        margin: 4px 2px;
        cursor: pointer;
        border-radius: 4px;
    }
    pre {
        background-color: #f8f8f8;
        border: 1px solid #ddd;
        border-radius: 4px;
        padding: 10px;
        white-space: pre-wrap;
        word-wrap: break-word;
    }
</style>
</head>
<body>
    <div class="container">
        <h1>Ajax Demo Program</h1>
        <h2>a. Ajax-like operation without jQuery</h2>
        <button onclick="operationWithoutjQuery()">Perform Operation (without
jQuery)</button>
        <pre id="result-a"></pre>
        <h2>b. Ajax-like operation with jQuery</h2>
        <button onclick="operationWithjQuery()">Perform Operation (with
jQuery)</button>
        <pre id="result-b"></pre>
        <h2>c. jQuery-like getJSON() method</h2>
        <button onclick="getJSONOperation()">Get JSON</button>
        <pre id="result-c"></pre>
        <h2>d. jQuery parseJSON() method</h2>
        <button onclick="parseJSONExample()">Parse JSON</button>
        <pre id="result-d"></pre>
    </div>

    <script>
        // Simulated data
        const simulatedData = {
            text: "This is a sample text from a simulated server response.",
            json: {
                name: "John Doe",
                age: 30,
                city: "New York"
            }
        };

        // a. Ajax-like operation without jQuery
        function operationWithoutjQuery() {
            setTimeout(function() {

```

```

        document.getElementById("result-a").textContent =
simulatedData.text;
    }, 500);
}

// b. Ajax-like operation with jQuery
function operationWithjQuery() {
    $.Deferred(function(deferred) {
        setTimeout(function() {
            deferred.resolve(simulatedData.text);
        }, 500);
    }).done(function(result) {
        $("#result-b").text(result);
    });
}

// c. jQuery-like getJSON() method
function getJSONOperation() {
    $.Deferred(function(deferred) {
        setTimeout(function() {
            deferred.resolve(simulatedData.json);
        }, 500);
    }).done(function(result) {
        $("#result-c").text(JSON.stringify(result, null, 2));
    });
}

// d. jQuery parseJSON() method (replaced with JSON.parse())
function parseJSONExample() {
    var jsonString = JSON.stringify(simulatedData.json);
    var jsonObject = JSON.parse(jsonString); // Use JSON.parse instead
of $.parseJSON
    $("#result-d").text(JSON.stringify(jsonObject, null, 2));
}
</script>
</body>
</html>

```