Below is a **self-contained** “scientific” calculator that you can drop into any web page (or open as an `index.html` file).

It contains:

Category	Functions
Arithmetic	+ – × ÷ % √ 1/x
Trigonometry	sin, cos, tan, asin, acos, atan (all in degrees)
Logarithms	log10, ln
Exponentiation	x^n, n^y
Misc	factorial (!), absolute value, power of ten (10^x)

The UI is deliberately minimal – a grid of buttons and an input field that also doubles as the display.

All calculations are performed with JavaScript’s `Math` object; for advanced precision you could swap in **Decimal.js** or similar.

1️⃣ HTML + CSS

```html
<!DOCTYPE html>
```

```
<html lang="en">
<head>
<meta charset="UTF-8">
<title>JavaScript Scientific Calculator</title>
<style>
  body{font-family:Arial,Helvetica,sans-serif;background:#f4f4f4;margin:0;padding:20px;display:flex;justify-content:center;}
  .calc{background:#222;color:white;width:350px;border-radius:10px;box-shadow:0 4px 12px rgba(0,0,0,.3);overflow:hidden;}
  #display{width:100%;padding:15px;font-size:1.8rem;background:#111;text-align:right;border:none;color:white;outline:none;}
  .grid{display:grid;grid-template-columns:repeat(5,1fr);gap:4px;padding:10px;}
  button{background:#333;color:white;border:none;font-size:1.2rem;padding:15px;cursor:pointer;transition:.2s;}
  button:hover{background:#555;}
  .btn-op{background:#ff9800;}        /* operators */
  .btn-func{background:#009688;}      /* functions */
  .btn-clear{background:#e53935;}     /* C, AC */
  .btn-eq{background:#3f51b5;grid-column:span 2;}
</style>
</head>
<body>

<div class="calc">
```

```
<input type="text" id="display" placeholder="0" readonly>
<div class="grid">

  <!-- Row 1 -->
  <button class="btn-clear" id="ac">AC</button>
  <button class="btn-clear" id="c">C</button>
  <button class="btn-op" data-value="%">%</button>
  <button class="btn-func" data-value="sqrt(">√</button>
  <button class="btn-op" data-value="**1/2">(x)^½</button>

  <!-- Row 2 -->
  <button data-value="7">7</button>
  <button data-value="8">8</button>
  <button data-value="9">9</button>
  <button class="btn-op" data-value="/">÷</button>
  <button class="btn-func" data-value="fact(">n!</button>

  <!-- Row 3 -->
  <button data-value="4">4</button>
  <button data-value="5">5</button>
```

```
<button data-value="6">6</button>
<button class="btn-op" data-value="*">×</button>
<button class="btn-func" data-value="log10(">log₁₀</button>

<!-- Row 4 -->
<button data-value="1">1</button>
<button data-value="2">2</button>
<button data-value="3">3</button>
<button class="btn-op" data-value="-">−</button>
<button class="btn-func" data-value="ln(">ln</button>

<!-- Row 5 -->
<button data-value="0">0</button>
<button data-value=".">.</button>
<button data-value="(">(</button>
<button data-value=")">)</button>
<button class="btn-op" data-value="+">+</button>

<!-- Row 6 – trigonometry -->
<button class="btn-func" data-value="sin(">sin°</button>
```

```
    <button class="btn-func" data-value="cos(">cos°</button>
    <button class="btn-func" data-value="tan(">tan°</button>
    <button class="btn-func" data-value="asin(">arcsin°</button>
    <button class="btn-func" data-value="acos(">arccos°</button>

    <!-- Row 7 – more -->
    <button class="btn-func" data-value="atan(">arctan°</button>
    <button class="btn-func" data-value="10**">10^x</button>
    <button class="btn-op" data-value="**">xⁿ</button>
    <button class="btn-func" data-value="abs(">|x|</button>

    <!-- Row 8 – equals -->
    <button class="btn-eq" id="equals">=</button>
  </div>
</div>

<script src="calc.js"></script>
</body>
</html>
```

2️⃣ JavaScript (`calc.js`)

```js
// Grab DOM elements
const display = document.getElementById('display');
const buttons = document.querySelectorAll('.grid button');

// ---------- Utility helpers ----------
function degToRad(d) { return d * Math.PI / 180; }
function radToDeg(r) { return r * 180 / Math.PI; }

// Factorial (integer only)
function factorial(n){
  if (n < 0) throw new Error('Negative factorial');
  if (n === 0 || n === 1) return 1;
  let res = 1;
  for(let i=2;i<=n;i++) res *= i;
```

```
  return res;
}

// ---------- Build expression ----------
let expr = '';         // current expression string

// Add a token to the expression
function appendToken(token){
  if (token === 'C') {       // clear last entry
    expr = expr.slice(0, -1);
  } else if (token === 'AC'){
    expr = '';
  } else {
    expr += token;
  }
  display.value = expr || '0';
}

// ---------- Evaluate ----------
function evaluate(){
```

```
try{
 // Replace custom functions with Math equivalents
 let e = expr;

 // 10**x → 10^x
 e = e.replace(/10\*\*/g, 'Math.pow(10,');

 // x**y → Math.pow(x,y)
 e = e.replace(/\*\*([\d\.]+)/g, (m,p)=>`),${p})`);

 // Factorial
 e = e.replace(/fact\(([^)]+)\)/g,(m,v)=>factorial(eval(v)));

 // Trig functions (degrees → radians)
 const trigFuncs = ['sin','cos','tan','asin','acos','atan'];
 trigFuncs.forEach(f=>{
  const re = new RegExp(`${f}\\(([^)]+)\\)`, 'g');
  e = e.replace(re,(m,v)=>{
   let val = eval(v);
   if (['sin','cos','tan'].includes(f)){
```

```
      return `Math.${f}(${degToRad(val)})`;
    }else{
      // inverse trig – output degrees
      const rad = Math[`${f}`](val);
      return `${radToDeg(rad)}`;
    }
  });
});

// log10 → Math.log10 or Math.log(x)/Math.LN10
e = e.replace(/log10\(([^)]+)\)/g,(m,v)=>`Math.log(${v})/Math.LN10`);

// ln → Math.log
e = e.replace(/ln\(([^)]+)\)/g,(m,v)=>`Math.log(${v})`);

// sqrt → Math.sqrt
e = e.replace(/sqrt\(([^)]+)\)/g,(m,v)=>`Math.sqrt(${v})`);

// abs → Math.abs
e = e.replace(/abs\(([^)]+)\)/g,(m,v)=>`Math.abs(${v})`);
```

```
    // Evaluate the final expression
    const result = eval(e);
    display.value = result;
    expr = '' + result;      // allow chaining
  }catch(err){
    display.value = 'Error';
    expr = '';
  }
}

// ---------- Event listeners ----------
buttons.forEach(btn=>{
  btn.addEventListener('click',()=>{
    const val = btn.dataset.value || btn.textContent;

    if (btn.id === 'equals'){
      evaluate();
    }else{
      appendToken(val);
```

```
    }
  });
});
```

How it works

1. **UI** – a grid of buttons, each with either `data-value` or an ID (`ac`, `c`, `equals`).
2. **Expression building** – clicking any button appends its token to the string that's displayed.
3. **Evaluation** – when "=" is pressed, the string is parsed:
   * Custom functions (`sin(`, `cos(` …) are translated into `Math` calls.
   * Trigonometric inputs are treated in **degrees**; inverse trig outputs are also converted back to degrees for user-friendly display.
   * Factorial is implemented manually because JavaScript has no native operator.
4. The result replaces the current expression so you can continue working.

3️⃣ Extending / Customising

| Feature | How to add |

|---------|------------|

| **Keyboard support** | Add `keydown` listener that maps keys → button click events. |

| **History panel** | Store each successful evaluation in an array and display it below the calculator. |

| **Scientific constants** | Add buttons for π (`Math.PI`) or e (`Math.E`). |

| **High-precision** | Replace `eval()` with a safe parser like [math.js](https://mathjs.org/) or use `Decimal.js`. |

Quick test

Save the two files above (`index.html` and `calc.js`) in the same folder, open `index.html` in a browser and try:

```
AC
7 8 9 ×
√
=
sin(30)
cos(60)
tan(45)
```

log10(1000)

ln(e)

fact(5)

10**3

2 ** 8

abs(-42)

```

All should compute correctly, giving you a lightweight but fully-featured scientific calculator written purely in JavaScript. Happy calculating!
```