

# JavaScript

JavaScript is a high-level, interpreted programming language that is most commonly used to develop interactive and dynamic web pages. JavaScript runs directly in the browser in which it was loaded and creates a more inviting experience for users. JavaScript, along with HTML and CSS, is considered one of the three mainstays of web development.

## The role of JavaScript

In the world of development today, JavaScript has a multitude of roles. JavaScript handles the interactivity of items such as forms with validation, dropdowns, sliders, animations, etc. It also enables communication with servers using APIs and AJAX without needing to reload the page. And JavaScript can extend beyond the browser now to browser extensions (Chrome, Firefox, etc.), backend development with Node.js, mobile apps (React Native), and even desktop apps (ElectronJS).

```
index.html JavaScript.html X
Example > JavaScript.html > html > body > script > <function> > window.addEventListener(keydown) callback
1. <doctype html>
2. <html lang="en">
3. <head>
4. <meta charset="utf-8" />
5. <meta name="viewport" content="width=device-width, initial-scale=1" />
6. <title>JS Calculator</title>
7. <style>
8. root { --bg: #f0f2f2; --panel: #f1f3f4; --btn: #e1e3e5; --txt: #546e7a; --muted: #8c9eaf; --accent: #808080; }
9. * { box-sizing: border-box; }
10. body { margin: 0; min-height: 100vh; display: grid; place-items: center; background: var(--bg); color: var(--txt); font-family: system-ui, Segoe UI, Roboto, Arial; }
11. calc { width: 100px; height: 30px; background: var(--panel); border: 1px solid #f5f5f5; border-radius: 10px; padding: 10px; box-shadow: 8px 8px 0px #ccc; }
12. screen { height: 100px; background: #f5f5f5; border: 1px solid #f5f5f5; border-radius: 10px; padding: 10px; display: flex; flex-direction: column; justify-content: center; overflow: hidden; }
13. mini { color: var(--muted); font-size: 12px; height: 10px; white-space: nowrap; overflow: hidden; text-overflow: ellipsis; }
14. main { font-size: 16px; text-align: right; line-height: 1.2; word-break: break-all; }
15. .keys { margin-top: 10px; display: grid; grid-template-columns: repeat(4, 1fr); gap: 10px; }
16. button { background: var(--btn); color: var(--txt); border: 0; border-radius: 10px; padding: 10px; font-size: 16px; cursor: pointer; transition: transform .1s ease, opacity .2s; }
17. button:active { transform: scale(.95); }
18. .op { background: #808080; }
19. .side { background: var(--accent); color: #808080; font-weight: bold; }
20. .side { grid-column: span 2; }
21. </style>
22. </head>
23. <body>
24. <div class="calc" role="application" aria-label="Calculator">
25.   <div class="screen" aria-label="display">
26.     <div id="history" class="mini"></div>
27.     <div id="display" class="main">0</div>
28.   </div>
29.   <div class="keys">
30.     <button data-act="ac" title="All Clear"/>AC</button>
31.     <button data-act="del" title="Delete"/>DEL</button>
32.     <button data-act="sign" title="Plus/Minus"/>+/-</button>
33.     <button data-op="+" class="op"/>+</button>
34.     <button data-op="-" class="op"/>-</button>
35.     <button data-op="*" class="op"/>*</button>
36.     <button data-op="/" class="op"/>/</button>
37.     <button data-num="0">0</button>
38.     <button data-num="1">1</button>
39.     <button data-num="2">2</button>
40.     <button data-num="3">3</button>
41.     <button data-num="4">4</button>
42.     <button data-num="5">5</button>
43.     <button data-num="6">6</button>
44.     <button data-op="%" class="op"/>%</button>
45.     <button data-num="7">7</button>
46.     <button data-num="8">8</button>
47.     <button data-num="9">9</button>
48.     <button data-op="*" class="op"/>*</button>
49.     <button data-act="percent"/>%</button>
50.     <button data-num="0" class="op"/>.</button>
51.     <button data-act="dec" title="Decimal"/>.</button>
52.     <button data-act="eq" class="op"/>=</button>
53.   </div>
54. </div>
55. </body>
56. </html>
57.
58. <script>
59. (function() {
60.   const display = document.getElementById('display');
61.   const history = document.getElementById('history');
62.
63.   let a = null; // first operand
64.   let b = null; // second operand (being typed)
65.   let op = null; // current operator: +, -, *, /
66.   let justEvaluated = false;
67.
68.   const fn = (a, b, op) => {
69.     if (!isFinite(a) || !isFinite(b)) return 'Error';
70.     const res = (a, b, op) => {
71.       // Basic arithmetic: sum, difference
```

```
index.html JavaScript.html X
Example > JavaScript.html > html > body > script > <function> > window.addEventListener(keydown) callback
<html lang=en>
<body>
<script>
() => {
  const fct = a => {
    // (a) is already square, return a
    const s = (a).toString();
    // limit length from precision
    let out = (a).toFixed(12);
    out = (out).toString(); // strip trailing zeros
    return out.length > 14 ? (a).toFixed(10) : out;
  };

  const update = () => {
    const top = [a !== null ? fct(a) : '', op || '', (b !== null && b !== '') ? b : ''].join(' ');
    history.testContent = top;
    display.textContent = (b !== null && b !== '') ? b : (a !== null ? fct(a) : '0');
  };

  const inputNum = d => {
    if (!justEvaluated && op) { a = null; }
    justEvaluated = false;
    if (op === null) { // typing first operand
      a = (a !== null || a === 0 && d !== '-') ? '' + (d === '-' ? b : d) : '' + (a + '' + d);
      if (d === '-' && (a + '').indexOf('.') !== -1) { a = (a + '').replace(/^-$/, '-'); }
      b = null;
    } else { // typing second operand
      b = (b !== null || b === 0 && d !== '-') ? '' + (d === '-' ? b : d) : '' + (b + '' + d);
      if (d === '-' && (b + '').indexOf('.') !== -1) { b = (b + '').replace(/^-$/, '-'); }
    }
    update();
  };

  const setOp = o => {
    if (a === null) a = 0;
    if (b !== null) { equal(); } // chain operations
    op = o;
    justEvaluated = false;
    update();
  };

  const clearAll = () => { a = null; b = null; op = null; justEvaluated = false; update(); };

  const del = () => {
    if (b !== null && b !== '') { b = b.slice(0, -1) || null; }
    else if (op) { op = null; }
    else if (a !== null) { a = (a + '').slice(0, -1); if (a === '' || a === '-') || a === null) a = null; }
    update();
  };

  const sign = () => {
    if (b !== null && b !== '') b = (b.startsWith('-') ? b.slice(1) : '-' + b);
    else if (a !== null) a = -a;
    update();
  };

  const percent = () => {
    if (b !== null && b !== '') b = fct(parseFloat(b) / 100);
    else if (a !== null) a = parseFloat(fct(a / 100));
    update();
  };

  const operate = (x, o, y) => {
    x = x; y = y;
    switch (o) {
      case '+': return x + y;
      case '-': return x - y;
      case '*': return x * y;
      case '/': return y === 0 ? NaN : x / y;
      default: return y;
    }
  };

  const equal = () => {
    // ...
  }
}

```

Output:

