

```
<!doctype html>

<html lang="en">

<head>

  <meta charset="utf-8" />

  <meta name="viewport" content="width=device-width,initial-scale=1" />

  <title>Demonstration & Documentation — Demo</title>

  <style>

    :root{

      --bg:#f6f8fa; --card:#fff; --muted:#6b7280; --accent:#2563eb;

    }

    *{box-sizing:border-box;font-family:Inter,ui-sans-serif,system-ui,Helvetica,Arial;}

    body{margin:0;background:var(--bg);color:#0f172a;}

    header{display:flex;align-items:center;justify-content:space-between;padding:18px 22px;background:linear-gradient(90deg,#eef2ff,#ffffff);border-bottom:1px solid #e6eefc}

    h1{font-size:18px;margin:0}

    .wrap{max-width:1100px;margin:20px auto;padding:0 16px;display:grid;grid-template-columns:1fr 420px;gap:18px}

    .card{background:var(--card);border-radius:12px;box-shadow:0 6px 18px rgba(15,23,42,0.06);padding:16px}

    .demo-area{min-height:230px;display:flex;flex-direction:column;gap:12px;align-items:flex-start;justify-content:center}

    .controls{display:flex;gap:8px;align-items:center}

    button{border:0;background:var(--accent);color:white;padding:8px 12px;border-radius:8px;cursor:pointer}

    .muted{color:var(--muted);font-size:13px}

    pre{background:#0b1220;color:#e6eefc;padding:12px;border-radius:8px;overflow:auto;font-family:Consolas,monospace;font-size:13px}

    .doc-title{display:flex;justify-content:space-between;align-items:center}

    .props dt{font-weight:700}

    .props dd{margin:4px 0 10px 0;color:var(--muted)}

    .copy-btn{background:#0f172a;padding:6px 8px;border-radius:8px}

    .small{padding:6px 8px;font-size:13px;border-radius:8px;background:#eef2ff;color:#0b1220}
```

```
.example{background:#fbfdff;padding:10px;border-radius:8px;border:1px dashed #e6f0ff}

footer{padding:10px 22px;text-align:center;color:var(--muted);font-size:13px}

/* small responsiveness */

@media (max-width:960px){.wrap{grid-template-columns:1fr;}}

</style>

</head>

<body>

  <header>

    <h1>Demonstration & Documentation</h1>

    <div style="display:flex;gap:8px;align-items:center">

      <span class="muted">Project: Demo + Docs</span>

      <button id="toggle-theme" class="small">Toggle Theme</button>

    </div>

  </header>

  <main class="wrap">

    <!-- Demo Column -->

    <section class="card" aria-labelledby="demo-heading">

      <div class="doc-title">

        <div>

          <h2 id="demo-heading" style="margin:0;font-size:16px">Live Demo</h2>

          <div class="muted" style="margin-top:6px">Try the component and play with props</div>

        </div>

        <div class="controls">

          <button id="reset">Reset</button>

          <button id="show-code" class="small">Show Code</button>

        </div>

      </div>

      <div class="demo-area" id="demo-area" style="margin-top:14px">

        <!-- Container where the component instance(s) will be mounted -->

        <div id="component-root"></div>

      </div>

    </section>

  </main>

</body>

</html>
```

```
<div style="width:100%;display:flex;gap:8px;flex-wrap:wrap">
```

```
  <label class="small">Start value:
```

```
    <input id="start" type="number" value="0" style="margin-left:6px;width:70px;padding:6px;border-
radius:6px;border:1px solid #e6eefc" />
```

```
  </label>
```

```
  <label class="small">Step:
```

```
    <input id="step" type="number" value="1" style="margin-left:6px;width:70px;padding:6px;border-
radius:6px;border:1px solid #e6eefc" />
```

```
  </label>
```

```
  <label class="small">Max:
```

```
    <input id="max" type="number" value="100" style="margin-left:6px;width:70px;padding:6px;border-
radius:6px;border:1px solid #e6eefc" />
```

```
  </label>
```

```
</div>
```

```
</div>
```

```
</section>
```

```
<!-- Documentation Column -->
```

```
<aside class="card" aria-labelledby="doc-heading">
```

```
  <div class="doc-title">
```

```
    <div>
```

```
      <h2 id="doc-heading" style="margin:0;font-size:16px">Documentation</h2>
```

```
      <div class="muted" style="margin-top:6px">Component: <strong>Counter</strong></div>
```

```
    </div>
```

```
    <button id="copy-doc" class="copy-btn" title="Copy documentation">Copy</button>
```

```
  </div>
```

```
<section style="margin-top:14px">
```

```
  <h3 style="margin:0 0 8px 0;font-size:14px">What it does</h3>
```

```
  <p class="muted" style="margin:0 0 12px 0">
```

A small reusable counter component with increment, decrement, auto-increment and max-limit.

Designed for demo and documentation: show usage, props, and source code.

</p>

<h4 style="margin:10px 0 6px 0">Props / Options</h4>

<dl class="props">

<dt>start</dt><dd>Initial numeric value (default: 0)</dd>

<dt>step</dt><dd>Amount to increment/decrement on click (default: 1)</dd>

<dt>max</dt><dd>Maximum allowed value (optional)</dd>

<dt>autoInterval</dt><dd>If provided (ms), component will auto-increment every interval</dd>

</dl>

<h4 style="margin:10px 0 6px 0">Usage (plain JS)</h4>

<div class="example muted">

<div>Mount:</div>

<div style="font-family:monospace;margin-top:6px">const c = Counter({start:0, step:1, max:10});
c.mount(document.querySelector('#root'));</div>

</div>

<h4 style="margin:12px 0 6px 0">Source code</h4>

<pre id="source-code" aria-label="Counter source code" style="height:260px"></pre>

</section>

</aside>

</main>

<footer>

Tip: Edit the inputs on the left and click "Show Code" to view the component's source.

</footer>

<script>

/*****

* Counter component (vanilla JS)

* - returns an object: {mount, unmount, getValue, setValue}

* - lightweight, no framework required

*****/

```
function Counter({ start = 0, step = 1, max = Infinity, autoInterval = null } = {}) {
```

```
  let value = Number(start) || 0;
```

```
  const root = document.createElement('div');
```

```
  // base styles for the component
```

```
  root.style.display = 'flex';
```

```
  root.style.flexDirection = 'column';
```

```
  root.style.gap = '8px';
```

```
  root.style.alignItems = 'flex-start';
```

```
  // display
```

```
  const display = document.createElement('div');
```

```
  display.style.fontSize = '34px';
```

```
  display.style.fontWeight = '700';
```

```
  display.textContent = String(value);
```

```
  // small hint
```

```
  const info = document.createElement('div');
```

```
  info.className = 'muted';
```

```
  info.textContent = `step: ${step} • max: ${isFinite(max) ? max : 'none'}`;
```

```
  // buttons
```

```
  const btnRow = document.createElement('div');
```

```
  btnRow.style.display = 'flex';
```

```
  btnRow.style.gap = '8px';
```

```
  const incBtn = document.createElement('button');
```

```
  incBtn.textContent = 'Increment';
```

```
  const decBtn = document.createElement('button');
```

```
decBtn.textContent = 'Decrement';

const autoBtn = document.createElement('button');

autoBtn.textContent = 'Start Auto';

const resetBtn = document.createElement('button');

resetBtn.textContent = 'Reset';

resetBtn.className = 'small';


// helpers

function render() {

  display.textContent = String(value);

  info.textContent = `step: ${step} • max: ${isFinite(max) ? max : 'none'}`;

  if (!isFinite(max) || value < max) {

    incBtn.disabled = false;

  } else {

    incBtn.disabled = true;

  }

}


function increment() {

  const next = value + Number(step);

  value = isFinite(max) ? Math.min(next, max) : next;

  render();

}


function decrement() {

  value = value - Number(step);

  render();

}


function setValue(v) {

  value = Number(v) || 0;

  if (isFinite(max)) value = Math.min(value, max);

  render();

}
```

```

// auto increment

let autoTimer = null;

function toggleAuto(ms) {

  if (!ms) { // clear

    clearInterval(autoTimer); autoTimer = null; autoBtn.textContent = 'Start Auto';

    return;

  }

  if (autoTimer) { clearInterval(autoTimer); autoTimer = null; autoBtn.textContent = 'Start Auto'; return; }

  autoTimer = setInterval(() => {

    increment();

  }, ms);

  autoBtn.textContent = 'Stop Auto';

}


// wire up events

incBtn.addEventListener('click', increment);

decBtn.addEventListener('click', decrement);

resetBtn.addEventListener('click', () => setValue(start));

autoBtn.addEventListener('click', () => toggleAuto(autoInterval || 1000));


// assemble

btnRow.appendChild(incBtn);

btnRow.appendChild(decBtn);

btnRow.appendChild(autoBtn);

btnRow.appendChild(resetBtn);

root.appendChild(display);

root.appendChild(info);

root.appendChild(btnRow);


// public API

return {

```

```

    mount(container) { container.appendChild(root); render(); },

    unmount() { if (root.parentNode) root.parentNode.removeChild(root); if (autoTimer)
clearInterval(autoTimer); },

    getValue() { return value; },

    setValue,

    toggleAuto: () => toggleAuto(autoInterval)

};

}

```

```

/*****

```

```

* Demo wiring

```

```

*****/

```

```

const componentRoot = document.getElementById('component-root');

const sourceCodeEl = document.getElementById('source-code');

const startInput = document.getElementById('start');

const stepInput = document.getElementById('step');

const maxInput = document.getElementById('max');

const resetBtn = document.getElementById('reset');

const showCodeBtn = document.getElementById('show-code');

const copyDocBtn = document.getElementById('copy-doc');

const toggleThemeBtn = document.getElementById('toggle-theme');

```

```

// Helper: pretty-print the Counter source (extract function text)

```

```

function getCounterSource() {

    // convert the Counter function to string and trim internal indentation

    return Counter.toString();

}

```

```

let instance = null;

```

```

function mountInstance() {

    // clear previous

    componentRoot.innerHTML = "";

```



```
const start = Number(startInput.value) || 0;

const step = Number(stepInput.value) || 1;

const max = maxInput.value === " ? Infinity : Number(maxInput.value);

instance = Counter({ start, step, max, autoInterval: null });

instance.mount(componentRoot);

}
```

```
// initial mount
```

```
mountInstance();
```

```
sourceCodeEl.textContent = getCounterSource();
```

```
// re-mount when the inputs change (instant demo)
```

```
[startInput, stepInput, maxInput].forEach(inp => {
```

```
  inp.addEventListener('change', () => {
```

```
    if (instance) instance.unmount();
```

```
    mountInstance();
```

```
  });
```

```
});
```

```
resetBtn.addEventListener('click', () => {
```

```
  startInput.value = 0; stepInput.value = 1; maxInput.value = 100;
```

```
  if (instance) instance.unmount();
```

```
  mountInstance();
```

```
});
```

```
showCodeBtn.addEventListener('click', () => {
```

```
  // toggle visibility of source code panel by scrolling into view (simple UX)
```

```
  sourceCodeEl.scrollIntoView({ behavior: 'smooth', block: 'nearest' });
```

```
  // also copy latest function source to the documentation panel
```

```
  sourceCodeEl.textContent = getCounterSource();
```

```
});
```

```

copyDocBtn.addEventListener('click', async () => {

  const docString = [

    'Component: Counter',

    'Description: A small reusable vanilla-JS counter.',

    'Props: start, step, max, autoInterval',

    ",

    'Example:',

    `const c = Counter({start:0, step:1, max:10}); c.mount(document.querySelector('#root'))`,

    ",

    'Source:',

    getCounterSource()

  ].join("\\n\\n");

  try {

    await navigator.clipboard.writeText(docString);

    copyDocBtn.textContent = 'Copied!';

    setTimeout(()=> copyDocBtn.textContent = 'Copy', 1200);

  } catch (e) {

    alert('Could not copy automatically. Select and copy manually.');
```

```

// Theme toggle
```

```

let dark = false;
```

```

toggleThemeBtn.addEventListener('click', () => {

  dark = !dark;

  if (dark) {

    document.documentElement.style.setProperty('--bg', '#0b1220');

    document.documentElement.style.setProperty('--card', '#071024');

    document.documentElement.style.setProperty('--muted', '#94a3b8');

    document.documentElement.style.setProperty('--accent', '#06b6d4');

    document.body.style.color = '#e6eefc';

  } else {
```

```
document.documentElement.style.setProperty('--bg', '#f6f8fa');

document.documentElement.style.setProperty('--card', '#fff');

document.documentElement.style.setProperty('--muted', '#6b7280');

document.documentElement.style.setProperty('--accent', '#2563eb');

document.body.style.color = '#0f172a';

}

});
```

```
// Show source right away in a collapsed block
```

```
sourceCodeEl.textContent = getCounterSource();
```

```
</script>
```

```
</body>
```

```
</html>
```