EcomLabs

STEP-BY-STEP GUIDE

CSS container queries: a product card for any slot

The complete HTML and CSS of a card that goes from one to three columns depending on the width of its container, a Playwright script to check it at three widths, and a cheat sheet for container-type, @container and the cqi units.

FOR	Themes and stores that show the same product card in a collection, a sidebar and a cart
TESTED WITH	Chromium 151.0.7922.34 · Playwright 1.62.0 · Node 22.22.3 · Windows 11
UPDATED	November 12, 2026
DOCS	developer.mozilla.org/en-US/docs/Web/CSS/container-type
ARTICLE	ecomlabs.dev/g/container-en
AUTHOR	Luis Morales · @luismoralesdev

Contents

Before you start	What you will build
Step 01	The HTML
Step 02	The CSS
Step 03	Cheat sheet
Step 04	Check it at three widths
When you finish	Checklist

BEFORE YOU START

What you will build

A product card that picks its layout from the width of the slot it sits in, not from the window: one column in a 240 px slot, two in a 480 px slot and three in a 720 px slot. I tested it on November 12, 2026 with Playwright 1.62.0 and Chromium 151.0.7922.34, in a fixed 1280 px window.

STEP 01

The HTML

The card lives inside a slot, `.card-slot`. The slot is the container, and whoever places the card (the collection, the carousel, the cart) declares it:

```
<div class="card-slot">
  <article class="card">
    <img class="card__img" src="/shirt.jpg" alt="Cotton t-shirt" width="400" height="300">
    <h3 class="card__title">Cotton t-shirt</h3>
    <button class="card__cta" type="button">Add to cart</button>
  </article>
</div>
```

STEP 02

The CSS

This is the CSS I ran, unchanged:

```
.card-slot {
  container: card / inline-size;
}

.card {
  display: grid;
  grid-template-areas: "img" "title" "cta";
  gap: 0.75rem;
  padding: 0.75rem;
  border: 1px solid #bbb;
}

.card__img {
  grid-area: img;
  width: 100%;
  height: auto;
  aspect-ratio: 4 / 3;
  object-fit: cover;
}
```

```
.card__title {
  grid-area: title;
  margin: 0;
  font-size: 1rem;
}

.card__cta {
  grid-area: cta;
}

@container card (min-width: 24rem) {
  .card {
    grid-template-columns: 8rem 1fr;
    grid-template-areas: "img title" "img cta";
    align-items: start;
  }
}

@container card (min-width: 36rem) {
  .card {
    grid-template-columns: 12rem 1fr auto;
    grid-template-areas: "img title cta";
    align-items: center;
  }

  .card__title {
    font-size: clamp(1.125rem, 3cqi, 1.5rem);
  }
}
```

At a 16 px base font, `24rem` is 384 px and `36rem` is 576 px. The base layout, with no queries, is the one-column one.

STEP 03

Cheat sheet

PIECE	WHAT IT DOES
`container-type: inline-size`	Queries the width; the width comes from context, not from the children
`container-type: size`	Queries width and height; collapses without a size of its own
`@container card (min-width: 24rem)`	Named query; styles descendants, not the container (an `outline` on `.card-slot` did not apply)
`cqi`, `cqw`, `cqb`, `cqh`	1% of the container; with no container they fall back to the window (`10cqi` gave 128 px in a 1280 px window)
Support	Baseline "Widely available" since February 2023, per MDN (November 12, 2026)

STEP 04

Check it at three widths

Save this as `check-card.mjs`. It opens a page with one card, changes the width of the slot and counts the columns of the grid:

```
import { chromium } from "@playwright/test";

const browser = await chromium.launch();
const page = await browser.newPage({ viewport: { width: 1280, height: 900 } });
await page.goto("http://localhost:3000/demo-card"); // your page with one .card-slot

for (const width of [240, 480, 720]) {
  await page.$eval(".card-slot", (el, w) => { el.style.width = w + "px"; }, width);
  const columns = await page.$eval(".card", (el) => getComputedStyle(el).gridTemplateColumns.split("
").length);
  console.log(width, "px ->", columns, "column(s)");
}
await browser.close();
```

Against a local page with the card (opened with `file://`) it printed `240 px -> 1 column(s)`, `480 px -> 2 column(s)` and `720 px -> 3 column(s)`. With three copies of the card at once, the title measured 16 px, 16 px and 21.6 px (`3cqi` of 720). With `@media` instead of `@container`, all three copies got three columns.

WHEN YOU FINISH

Checklist

- [ ] The container is the slot, not the card, and has `container: name / inline-size`.
- [ ] The slot has a width that comes from context or from a `width`.
- [ ] Every `@container` carries the container name.
- [ ] The base layout, with no queries, works in one column.
- [ ] The image has `width`, `height` and `aspect-ratio`.
- [ ] Every `cqi` sits inside a container.
- [ ] I tested the card at three slot widths with the window fixed.