GYATA TABIRI

Inspired. Precise. Purposeful.

Debugging CSS by testing one explanation at a time

By Gyata Tabiri · Revised November 2, 2026

When a layout is wrong, another override is tempting because it produces an immediate visible change. I want the change to explain the problem as well. Otherwise the next adjustment begins on top of a fix nobody quite understands.

Make a prediction

Suppose three cards stack at desktop width when they should sit together. Before changing widths, I would inspect the row's computed display. If a utility class has changed it from flex to block, altering the children cannot resolve the underlying layout rule. The explanation should predict what a small change will do.

Inspect the rule that actually wins

A declaration in the stylesheet may be overridden, inherited differently, or limited by a media query. Developer tools can show the computed result and its source. That is where I would look before increasing specificity. A stronger selector can make the symptom disappear while making the cascade harder to reason about.

Follow the surrounding relationships

An overlapping element may belong to a different stacking context. A child that overflows may be constrained by its parent or by content that cannot shrink. These relationships matter more than a large z-index or another percentage guessed in isolation.

Check where the explanation stops working

A correction needs checking at the original width, nearby widths, and other uses of the affected pattern. If it only works with the current title length, the explanation is incomplete. I would also remove temporary diagnostic changes before leaving the file.

The useful result is a bounded reason: this condition caused the failure, and this change addresses it. That gives the next person something firmer than "it looked right after this rule was added."

Read alongside: A component's boundaries · Changing an existing stylesheet carefully.

Reference: Understanding stacking contexts.