How to Use the Browser Dev Console to Debug a Web Page

The tool that ships with every browser, that most people never open

How to Open It

Browser	Keyboard Shortcut	Menu Path
Chrome / Edge	F12 or Ctrl+Shift+I	Menu → More Tools → Developer Tools
Firefox	F12 or Ctrl+Shift+I	Menu → More Tools → Web Developer Tools
Safari	Cmd+Option+I	Develop menu (must enable in Preferences first)

The Four Tabs You Actually Need

Tab	What It Shows	When to Use It
Console	JavaScript errors, log output	First stop. Always. If something broke, the error is here.
Network	Every HTTP request the page makes	API calls failing, slow loads, 404s, missing files
Elements	Live HTML and CSS of the page	Layout broken, CSS not applying, wrong content showing
Application	Cookies, localStorage, sessionStorage	Login issues, cached data problems

The Console Tab: Start Here

When a page breaks or a button doesn't work, open the console first. Red text is a JavaScript error. It tells you what broke and on which line. You can also type JavaScript directly into the console and run it.

```
// Type these directly in the console to test things:
document.title // page title
document.querySelectorAll('a') // all links on the page
fetch('/api/test').then(r => r.json()).then(console.log) // test an API call
```

Educational only. Not legal advice. No attorney-client relationship. ©2026 Chariton Media.

The Network Tab: When API Calls Fail

Click Network, then trigger the action that's failing (click the button, submit the form). You'll see every request the page made. Look for red rows; those are failures. Click any row to see the full request and response, including the exact error the server returned.

Status codes to know:

Status	Meaning
200	OK; worked
401	Unauthorized; missing or bad credentials
403	Forbidden; you don't have permission
404	Not Found; URL doesn't exist
500	Server Error; something crashed on the backend

The Elements Tab: Live CSS Editing

Right-click any element on the page and choose Inspect. The Elements tab shows you the exact HTML and which CSS rules apply to it. You can edit values live; changes show immediately on the page. Nothing is saved, but it's the fastest way to test a CSS fix before writing it into your stylesheet.

> **Using AI to help with this.** You don't have to fully understand this to use it. Here are prompts that work: "I have a JavaScript error in my browser console: [paste the full error]. What does it mean and where should I look to fix it?" "My API call is returning a 403 in the Network tab. Here is the request and response: [paste]. What is the likely cause?" "The CSS on my page is not doing what I expect. Here is the element HTML and applied styles from the Elements tab: [paste]. What is wrong?"

Educational only. Not legal advice. No attorney-client relationship. ©2026 Chariton Media.