



webmonkey
The Web Developer's Resource

SEARCH:

JUMP TO A TOPIC:

1 8 0 0 NEWS
A Wired News Site

[home](#) / [programming](#) / [javascript](#) /

JavaScript

 [Print](#)

this article for free.

Pages:

- 1 **Simple JavaScript Debugging**
- 2 [The Agony of Debug](#)



Simple JavaScript Debugging

by [Chris Klimas](#) 10 May 2006

[Chris Klimas](#) tries his best to abuse HTML and the English language in equal amounts. He tills the soil at [Gimcrack'd](#) and eats breakfast whenever possible.

Page 1

The first time you try something, you probably won't get it completely right.

Writers edit, architects redraw, and programmers debug. Development environments are usually nice enough to give programmers a bunch of tools to help determine where their program's going wrong. But with JavaScript, your development environment is a web browser, and things aren't quite so hospitable.

All modern browsers will tell you when your code is screwed up, and what it thinks went wrong. However, the browser may tell you misleading or ambiguous information. Even worse, the errors it reports may in fact be triggered by the browser's own bugs and peculiarities.

Most modern browsers will tell you what line an error occurs on. However, the line it points to may not actually be the real source of the problem. It's just where things got irrevocably screwed up. For example, you may be making a call to a framework like [Prototype](#) when your script dies — the problem may not be the framework, but what you're passing to it via a function call. Also, if an error occurs in an [anonymous function](#), browsers will just throw their hands up in confusion.

In short, debugging JavaScript can be a real mess. Fortunately, the most popular Web browsers on the block, Internet Explorer and Firefox, come with full-fledged script debuggers. Less well-known browsers like Safari, Opera, and Konqueror aren't as nice. Though it's tempting to write them off entirely, the truth is that their JavaScript repertoire is almost as good as the major players — it's just a bit harder to figure things out when they don't work.

You know that old saw about an ounce of prevention? Well, it turns out to be true for debugging, too. There are some things you can do as you write your code to make debugging later on easier. First of all, always keep your JavaScript in an external file that you link to, like so:

```
<script type="text/javascript" src="myscript.js"></script>
```

This guarantees that the line numbers your text editor sees are the same your browser does. If your page contains both dynamic content and JavaScript, then the line numbers in browser error messages will be off.

Second, always keep your browser's JavaScript console open as you test your code. Browsers tend to keep quiet about JavaScript errors, just as they keep quiet about quirks in HTML. Firefox and Opera have menu items to open the console, but Internet Explorer and Safari take a more roundabout approach.

In IE, go to Tools > Internet Options and choose the Advanced

tab. Make sure the check box for "Display a notification for every script error" is checked. To turn on the console for Safari, follow [these steps](#).

Finally, try some defensive programming with assertions. An assertion is a line of code that says, "At this point in the program, this must always be true." Here's an example:

```
function divide (x)
{
  assert('x is nonzero', x != 0);
  alert(10 / x);
}
```

The assertion in this function keeps your code from dividing by zero. Assertions aren't a replacement for good error handling — they just note problems right at the source and tell you exactly what's going wrong.

I've pulled the assertion functions from a really nice testing framework called [jsUnit](#) and put them into [an easy download](#), (Right-click and choose "Save As..." to download the source).

The [online documentation](#) explains how to get started with them. jsUnit is available under several licenses, including the [LGPL](#), which means you can use it in your projects without having to make your own source code public. However, if you use jsUnit in production code, you must post its copyright notice on your site.

So all this is great, but what do you do when you actually need to start debugging?

[next page](#) >>

Wired News: [Contact Us](#) | [Advertising](#) | [Subscribe](#)
We are translated daily into [Korean](#) and [Japanese](#)

© [Copyright](#) 2006, Lycos, Inc. Lycos is a registered trademark of Lycos, Inc. All Rights Reserved.
Your use of this website constitutes acceptance of the Lycos [Privacy Policy](#) and [Terms & Conditions](#)

[an error occurred while processing this directive]