



webmonkey
The Web Developer's Resource

SEARCH: webmonkey the web
GO

JUMP TO A TOPIC:

Choose Topic

1 8 0 0 NEWS
A Wired News Site
GO

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

JavaScript



Debugging with Venkman

by [Chris Klimas](#) 17 May 2006

Print

this article for free.

Pages:

- 1 [Debugging with Venkman](#)
- 2 [Stepping up](#)

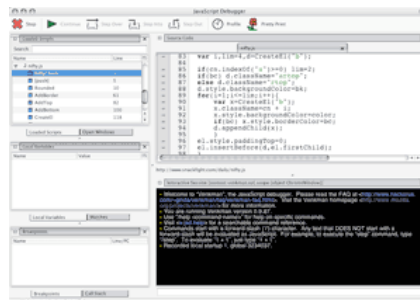
Page 1

In the [first part](#) of our JavaScript debugging tutorial, we talked about debugging methods that work in just about any web browser. These techniques are useful when you're trying to get your script to work in a particularly troublesome browser. More often, however, you're faced with some code that just plain doesn't work, and you're not sure why.

With a script debugger, you can pop the hood and study exactly how things work — the catch being that the only really robust debuggers exist only for Internet Explorer and the Mozilla family of browsers. But that's OK. You don't really care about browser compatibility yet; you just want the darn thing to work.

In today's lesson, I'm going to show you how to use [Venkman](#), Mozilla's script debugger. It's not really any better or worse than Microsoft's [Script Debugger](#), but it has the advantage on running on Windows, Linux, Unix, and Mac OS X. You can download it from [Mozilla's site](#). As with all extensions, you'll need to restart your browser after installation. Once you restart, you should see a menu item named JavaScript Debugger under the Tools menu. Make sure to click on this menu item *after* you've loaded the page you want to debug. It also is a good idea to close any extraneous tabs before you start debugging.

The first time you open Venkman, you'll be treated to a classic open-source interface that offers up eight panes in a single window. Don't freak out just yet. You can rearrange the layout to your liking and get rid of the panes and elements you don't care about. You can drag a pane around the window to rearrange things, and remove a pane entirely by clicking the X in its upper-right corner. To dock two panes together, drag one to the box on the left side of the other's title bar and let go of the mouse button. The two panes will be joined, and you can toggle between them using tabs.



Venkman's interface (Take a [closer look](#))

The Loaded Scripts pane shows the script files currently loaded. By clicking the disclosure triangle beside a script file, you can browse to specific functions in the file. The Open Windows pane

gives you a similar view, but further separates things by browser windows.

The Local Variables pane lets you peek at variable values as your code executes. More about this in a minute. The Watches and Breakpoints panes let you specify when to start debugging. And the Call Stack pane shows where you are in the code — that is, what function is executing and who called it.

That leaves two panes: Source Code, which shows your code as you debug, and Interactive Session, which gives you a command line to work with. You don't have to use this pane at all, but if you're interested in it, type `/help` to get a command reference.

Alright, now let's do some 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]