

# Table of Contents

Return of the Browser Jedi? . . . . .	1
---------------------------------------	---



# Return of the Browser Jedi?

By Danny O'Brien

If you thought the browser wars of old were all washed up, think again. With Apple, Mozilla and Opera all getting together to develop HTML based web applications, argues Danny O'Brien, it looks like Microsoft could be left in the shade

I've always felt there weren't enough Cosmopolitan-style personality tests in Mute. Take a look at the three boxes on the facing page. Pick the one you find most reassuring. After you've done that, I'll draw some specious conclusions about the current battle between Microsoft and the rest of the world, and why that Browser War people endlessly banged on about in the '90s isn't quite over yet.

Was it number three? It wasn't? Then I have no further use for you and you may leave. (Go on, say it was number three.)

The three boxes are, in turn, snippets of the way you code a standard Windows C application, the way Microsoft would like you to code in the future, and the way the web is coded.

If you picked number three, it was probably because you've seen bits of web code, either in popular culture, scrolling past in an OCR font to indicate generic internet activity, or a webpage that you were viewing broke and you saw its code guts spill out in an error message. Alternatively, you're a web designer, or have dabbled a bit with building websites, hitting 'View Source' in your browser and cutting and pasting other people's HTML code together into your own site, uploading and reloading until it looks just fine.

Historically, that's how most people started to code the web – looking at the code for other people's web pages, clanking it together for their own. It has its disadvantages – folk dismissively describe it as 'cargo cult' programming – but it works, and it's how the web was won. Every new webpage learnt from another; anyone, in theory, could use the tools they had in front of them and create a new site.

Clanking it together was also how the first microcomputer code was built. Indeed, before the concept was polished into a high political sheen, that was one of the key attractions of Open Source software: its pedagogical dimension. With Open Source, you could look inside the machine crafted by your predecessors, and learn from their work. Shoulder-surfing giants, as it were.

But it all got a bit more hidden over time, as applications became more complex, and the source grew more closed, and the gulf between computer user and computer programmer widened.

Unless you're a programmer, it's unlikely that you'll have seen anything like Box 1. That's Windows code, the end result of that gulf, and it takes a long time to get the hang of it. There's historical oddities to be learnt, and Microsoft paradigms to grok, and old dead ends to be avoided. It's ugly but it's not impossible, and it's certainly not uncommon to learn it. If you're a developer, you can buy almost every bit of code and advice Microsoft has to offer for \$375 in their independent software vendor programme. And, as you might have noticed, there's a lot of people developing Windows software.

But there are more web pages; and Web pages don't depend on Microsoft, and you can build them using Notepad, which doesn't even cost \$375.

That, in essence, was the pitch that Netscape made to coders all those years ago. Come with us and our browser, and write for the web, they said in 1995, and you'll be free. Netscape promised then that the web would become the platform, just like Windows. That you'd be able to write web pages that worked just like, say, Microsoft Word. Only they'd be \*your\* Web application, and all that lovely money that went to Microsoft whenever anyone bought Word would go to you, whenever anyone used your web page. We will call them 'web applications', and we will all be rich, do you hear me? Rich! And freeeeeeeee!

It didn't quite pan out that way. No one knew how to make web pages sing like applications. Even Netscape didn't know how to do it – yet. And while they worked hard trying to figure out how, they neglected to maintain their browser, and Microsoft came by and fucked them over with Internet Explorer 5, and Netscape died and it was the end. And everyone went back to writing Windows applications, because it may be dirty work, but it was familiar dirty work. At least Microsoft – and programmers – knew how to write one of those.

Bolt upright in bed, covered in sweat, let us return to the present day. We sort of know how to do web applications now: Google's Gmail is a good example. It's a bit of a bitch to do, however, because Internet Explorer is deliberately designed not to permit web pages to do anything like what a desktop application can do. The web has been successfully shackled.

Except: Scared by Explorer's security flaws and general bit-rottenness, a few percent of Windows users are going over to Mozilla, the open source remnants of the Netscape expeditionary force (you should too – it's a fantastic browser, <http://www.mozilla.org/>).

And Apple has Safari, their own browser.  
And Apple has Steve Jobs.  
And Steve Jobs has a plan.

Apple has made it known that they're extending HTML, the code of the web, to do desktop-like applications. Oh, nothing for Microsoft to get worried about: it's just silliness really – international clocks, games of Minesweeper. Mere fripperies. But they're also working together with the Mozilla and Opera people to standardise these extensions.

And if it all works the way it should, those extensions will be the extras that the web needs to do applications.

And what's Microsoft doing while this goes on? Well, they're hard at work producing .Net, which is that middle box above. Which is a beautiful work of art. It gets rid of almost all the fugliness of Box 1. It's the future for programming for Microsoft.

Unfortunately, it's a little unfamiliar to most Windows coders. Faced with throwing away all their old knowledge, where should they go? With Microsoft to the promised land of .Net? Or just add a few extensions to what these coders already know about building web pages?

It's not clear which way most of them will lean. It's not clear that Apple and Mozilla will be able to pull off for a second time what Netscape so royally screwed up before. But there's a chance – and watching contenders with the barest chance take on Microsoft is a pleasure you see only once a decade.

Danny O'Brien <danny AT spesh.com> is co-editor of NTK in exile