

Interview with John Resig, jQuery Creator

John Resig's role at the Mozilla Corporation is *JavaScript Evangelist*. I challenge you to invent a cooler title!

He also happens to be the creator and lead developer of the jQuery JavaScript library. Ara Pehlivanian caught up with him recently to talk about the future of JavaScript, online privacy, and a handful of his long list of ongoing projects.

SitePoint: Let me start out by congratulating you on the [recent announcements](#) by Microsoft and Nokia that they'll be using jQuery in their platforms. I'm curious, was there any lobbying from the jQuery community for this outcome, or did it come as a surprise?

There has certainly been a number of jQuery developers who also worked on ASP.NET. Reading the blogs, you could see just how happy they were over the announcement.

As far as I know, there wasn't official lobbying by the community – but there was certainly lobbying from within the respective organizations to support jQuery as the library of choice. Both of the announcements were a surprise (in that we were approached by the companies without expecting it) but we talked with them and worked hard to make sure that everything went smoothly for any sort of collaboration.

Both companies were very reasonable about the collaboration. We set out a number of terms surrounding any future work, so that the community would benefit in the end (such as no code forking, sending contributions back to the project, open communication, and such).

SP: How will this affect jQuery's evolution?

It'll absolutely be for the better. By being able to have more resources (because we'll need more of everything: more testing of jQuery, more patches, more input) at our disposal, the entire project will benefit. In the end it'll only force us to become more professional and have a more-refined development process – which is really what we should shoot for, anyway.

SP: As creator and lead developer on jQuery, what's your opinion of competing JavaScript libraries, and how do you think the landscape will evolve over time?

Libraries go through cycles (as do browsers, for that matter). There's a period of competition, followed by cooperation, followed by homogenization. For example, for the

CSS selector engines that all libraries now implement, there was a lot of competition surrounding the performance of their engines; there had to be some differentiation since everyone was, effectively, implementing the same feature over and over again. Libraries then cooperated and came up with a solid set of selectors to support, sharing performance-improvement techniques, as well. Finally, we're now seeing the homogenization of the code base – unifying on a single back end (like Sizzle). I absolutely expect that we'll see this with other features that libraries duplicate.

The biggest benefit of a library is two-fold: it should remove the user from dealing with browser issues, and it should provide the user with an API that will simplify their development. It must do both for the library to succeed. At the same time, though, it's important to realize that asking for a unified library (some sort of amalgamation of different libraries) just doesn't make sense. Much of what makes a library unique is its API and how it chooses to tackle the problems presented by the browser. Turning it into a bland, generic API will only result in a device that users will have to paper over again and again.

SP: What is Sizzle?

[Sizzle](#) is a new pure JavaScript CSS selector engine. Sizzle is the code name for what will, eventually, become the new engine within jQuery. I wanted an engine that was fast (at least competitive with other engines), small (a light file size), easily extensible, and had no dependencies.

The bonus of achieving this is that the engine is likely to become part of a number of major libraries. I've been talking with the [MochiKit](#), [Prototype](#), [Dojo](#), and [TinyMCE](#) teams and they're all interested in using Sizzle for their code base. It'll be great to have a unified approach to tackling JavaScript CSS selectors.

SP: These libraries often fill in the gaps in JavaScript's evolution, but there seems to be some contention regarding the shape it should take moving forward. Recently, for example, Douglas Crockford and Brendan Eich have been at odds over the need to complexify or simplify the language. What's your take on this issue?

I don't think that they're mutually exclusive. The biggest contention appeared to be (and was relatively solved by the [recent Harmony merger](#)) that the new features of the language should be built using smaller building blocks. The Microsoft JScript team wanted the language to be built in this way and I agree: if there's going to be a `class{}` structure, then it needs to be nothing more than syntactic sugar for a task that can already be done using the built-in methods of the language. Starting simple and then building up the complexity is a very efficient way to write code.

SP: With browser support for anything beyond JavaScript 1.5 being a bit sketchy, in your opinion, what should coders do who want to play with more advanced features of the language?

I would worry less about the advanced features of the language and look more into the upcoming features of HTML5. If you want to be practical and be able to use what you learn today, then you'll rarely find solace in the realm of JavaScript language. However, since the DOM is nicely object-detectable (`if (document.querySelectorAll)`), it's quite easy to get up and running right away, to start playing with new features.

SP: Let's talk about some of the things you're working on right now. The projects page on your web site lists 76 projects. Is there one or two in particular that you're particularly excited about?

Oh, gosh – there's a bunch. My two favorites that show promise are:

[port the browser environment over to pure JavaScript \(running in Rhino\)](#). It allows you to run web applications, including libraries like jQuery and Prototype, completely offline. There are a bunch of people working on it now and using it for a number of applications.

[processing.js](#) – This is a [more recent project of mine](#), which was a port of the processing visualization language to JavaScript (running on top of Canvas). This has seen a lot of uptake with people using it to build everything from games, real-time visualizations, charts – to art, all made with JavaScript.

One of my longest-running projects is [a schedule maker for RIT](#) (Rochester Institute of Technology, the college that I went to) where you could plug in a number of classes and it would compute all the possible schedule permutations. Using this, you could find the ideal schedule for your classes. It's been running since early 2003 and [there are still a bunch of people using it](#).

SP: I've heard you mention Fuel a few times in the past, what is it?

This was a project that I undertook when I first joined Mozilla back in 2007. It was an attempt to simplify the complex Mozilla APIs by putting a simple JavaScript layer in front of them. There are a number of extensions making good use of it now (since it ships with Firefox 3), so I'm glad to see it starting to help make developers' lives simpler.

SP: Apart from your programming projects, you're also an author. You post regularly to [your blog](#) and have recently started writing [articles for Ars Technica](#). With everything else on your plate, what prompted you to do this?

I wanted to do some writing that was *divorced* from my blog – to write from a more neutral perspective. Ars Technica has a great reputation for writing technical pieces that are relatively unbiased but still very informative. Being able to write for them is a great challenge and should certainly improve my writing ability.

SP: You're also writing a new book called [Secrets of the JavaScript Ninja](#). How's that coming along?

It's coming slowly. I've fallen behind these past couple of months, so I need to catch up again. It's looking likely that it'll be a North American spring release, at the earliest.

SP: Who is the target audience? Can a newbie learn from it or is it geared more toward experienced coders looking to become Ninjas?

It's definitely more towards experienced coders. There are plenty of books for people new to JavaScript, but I felt as if the market for people craving more was quite under-nourished at the moment.

SP: What can we expect from this book that's unique?

It actually dares to cover cross-browser development. Almost all JavaScript books stay in the safe realm of the JavaScript language where there are, comparatively, few bugs to contend with. Once you move into the DOM and tackle hard problems – the kind of problems that libraries handle – that's where the most work comes into play.

SP: I've got a couple of non-work related questions for you. You used to list your cell phone number on your About Me page as a sign of openness. It's no longer there. How come?

Honestly, it was rarely ever used for what it was intended. I'd roughly say that about 70% of the calls were from recruiters – who I had to send elsewhere; 25% were jQuery questions that were better handled by the mailing list; and 5% were other questions that required (at least) a follow-up email just so I could better understand what the problem was.

I still reply to questions that I get by email or IM (handling them on my own timetable is nice); it just turned out that having the phone number displayed was too intrusive without much benefit.

SP: So, you're writing a book, and regularly posting articles on different sites (including your own); you're speaking at conferences, and coding on several different projects. You're obviously a very busy person. Can we get a glimpse of the other side of John Resig? What do you do for fun? What are your favorite bands/movies/games/books/magazines/web sites?

I have a wonderful girlfriend, Claire, who I spend a lot of my free time with. I enjoy going out to shows (recently saw The Dodos, The Presets, and Cut Copy), visiting museums, and generally hanging out with friends. I'm also an enthusiast of good television: currently watching True Blood, Dexter, Mad Men, and It's Always Sunny in Philadelphia.

I'm working my way through some good graphic novels at the moment: have just finished *Watchmen* and working on *Batman: The Dark Knight Returns*. I'm also hopelessly addicted to [reddit](#).

I like to keep a pretty clear separation between my work and my personal life. When I'm having fun with friends, I'm not muddying it up with talk about work or computers. It's very relaxing – in fact, the majority of my real life friends know very little about computers at all. Of course, living in Boston it's hard to ignore sports so I like to catch a game or two, when I can.

Thanks for your time, John!