Sean McQueen
09/12/13

Topic proposal:

The Politics of JavaScript: the “Language of the Web”

Widely cited as the most popular programming language in terms of the sheer number of runtime environments and lines of code out in the world, JavaScript is both the most important programming languages of the modern era and also one of the least well understood.

Some JavaScript knowledge is required to write even basic content for the web, meaning that many people's first experience with programming is in JavaScript and a massive amount of JavaScript code is written each day by complete novices. At the same time, JavaScript experts are pushing the language beyond its home in the browser to new runtime environments on every type of machine imaginable. JavaScript is quickly becoming a general purpose programming language, while retaining its position as the king of the browser. For a language which was written in about ten days in May 1995 (according to its creator), this seems like a very unlikely trajectory.

How did JavaScript get to where it is today? What social and political forces contributed to the language's adoption in all major browsers and its massive popularity (despite many of its technical drawbacks)? While completing a semester-long web development project using JavaScript in both the browser and on the back-end server, I plan to study the political history of JavaScript and answer these research questions. There is increasing interest in where JavaScript is headed. For millions of web developers—as well as for companies like Google, Facebook and Microsoft—the question of what tools will be used for web application development in the coming years could not be more important. Some advocate for replacing JavaScript, some advocate for improving JavaScript, and some do not see a need for either. No matter what, Javascript deserves more study in the academic community. A political study of the language will reveal some interesting things.

Some subtopics to explore:

- The early history and original development of JavaScript at Netscape.
- JavaScript's roll in the 'browser wars' of the 1990s.
- JQuery and other front-end libraries being so popular that in some cases they are effectively part of the language.
- The perception of JavaScript as a 'toy' language: JavaScript's reliance on and connection to HTML.
- JavaScript as a starter language for novices and beginners.
- The development of Dart, CoffeeScript and ClosureScript: languages designed to compile to JavaScript and/or improve it.
- The development of NodeJS and runtime environments on Unix systems; breaking the connection to the browser.
- ECMA and the process of continued development of the language.
- The relationship between ECMA and Mozilla, Microsoft, Apple, Google and other browser makers.
- The relationship between browser makers.