

# Co-teaching Web Programming

Sam Auciello

09 December 2012

This fall I co-taught a course in Web programming. I helped by preparing examples of the technologies we were talking about in class, presenting many of the subjects, and answering questions outside of class. This opportunity to teach about software was quite helpful to my plan of concentration in computer science in general. Although I had a lot of experience doing web programming prior to this semester, my understanding of the subject was deepened by my teaching experience.

Looking at web programming in greater detail forced me to discover aspects of the technologies of the web that I had not previously explored. Some of these revelations were relatively trivial. For example I would not have known how powerful and easy to use the Raphael JavaScript library is if a student hadn't used it in a mid-term project. I also was unclear the details of CGI until I had to learn about it to present examples of it to the class. I could have learned about these technologies on my own but the course gave me the excuse. Other discoveries were quite useful. Looking more closely at node.js and CoffeeScript gave me both inspiration and context for my major plan project.

Looking at the subject from the perspective of a beginner forced me to call aspects of my own perspective into question. Ruby has a notion of convention over configuration. As an experienced web developer, I can see the real benefits of this as a design principal. If something is the case ninety percent of the time we should only need to specify when it is different. It wasn't obvious to me at first why rails was not a good choice of framework to teach, given how popular it is. On reflection I can see the value to a novice web programmer to seeing why certain decisions are the case ninety percent of the time before a framework decides things for them.

Working with a co-teacher who had differing perspectives on the subject

matter forced me to re-examine my own ideas and gain a better understanding in the process. Especially in a field where the number of ways to approach a given task is growing so rapidly, it is important to see the merit to methods you might have initially dismissed. Because the majority of my prior experience in web development was self-taught, it was really helpful to build up a better foundation in the ideas behind the technologies particularly working with someone whose perspective so often differs from mine.

In addition to supporting my plan by giving me a better grounding in an increasingly important branch of computer science, this course was directly helpful to my major plan project. Specifically some of the technologies I was teaching about relate directly to my project. Working with and teaching about JavaScript, Node.js, and CoffeeScript, helped to give context to my project. Using CoffeeScript in a class demo gave me much better ideas about how I want Hot Cocoa Lisp to work. It would not have occurred to me that a language that transcompiles to JavaScript might be able to provide useful debugging info in comments until I looked at the output of the CoffeeScript compiler.

More generally looking at a variety technologies and APIs helped to inform the my study of language design. Improving my understanding of an area of CS I was already familiar with ties works well with the aspect of my plan that is studying CS in general.