

Instructor training for web development immersion courses

We will cover the topics below as deeply as time permits, but we'll focus on *how* to learn them more than on the material itself. Web developers must learn to teach themselves so that they can stay abreast of the lightning fast of technical change; so it is that teachers of web developers must learn how to teach themselves so that they may help others to do the same.

You will be expected to continue your self-education on these topics *and more* after the course ends. This is a *life long learning process*. We provide here only the basics.

We will cover this material in the order that most makes sense to the class. This type of teaching is coöperative: we work together as a team to learn the material, with each individual proceeding as quickly as his or her aptitude allows. Because we are learning to be teachers, we will teach each other throughout the course.

What you get out of this course will be directly proportional to what you put into it. Motivated, self-directed, conscientious, and socially-skilled individuals wanted!

Developer training

- The Agile software development process and how it applies to learning to develop web applications
 - Pair programming/XP
 - Scrums, standups, etc.
 - Project management with PivotalTracker, Trello, etc.
 - Version control with Git and GitHub (branching, merging, rebasing, pull requests, etc.)
 - Front-to-back modular web application development
- Web site & web application fundamentals
 - Elements of user experience (surface, skeleton, structure, scope, strategy)
 - User needs and product objectives
 - Functional specifications and content requirements
 - Interaction design and information architecture
 - Information design, interface design, and navigation design
 - Sensory design
 - Separation of concerns (content, structure/semantics, presentation, behavior/interaction)
 - HTTP, HTML, and REST
 - CSS and CSS pre-processors
 - CoffeeScript and JavaScript libraries (e.g., jQuery)
 - Usability and user experience (UX)
 - Accessibility/universal design
 - Wireframing, prototyping
 - User testing
- Programming fundamentals

- Procedural vs. object-oriented vs. functional programming
- Elements of a program (variables, conditionals, loops, functions, objects, classes, types, etc.)
- SOLID principles
- SQL and NoSQL databases
- Data modeling
- Immutability, referential transparency, recursion, pattern matching
- Fundamentals of Ruby, CoffeeScript/JavaScript, and, if time permits, an FP language such as Scala or Elixir
- Standards compliance, best practices, linking
- Compiled vs. interpreted languages
- Web framework fundamentals
 - Multiple-page applications with Ruby on Rails & Sinatra
 - JSON APIs to jsonapi.org standards
 - Single-page web applications with AngularJS and node.js
 - Reactive web applications with Meteor (and Scala, if time permits)
 - RDBMS/ActiveRecord with SQLite and PostgreSQL
 - Document databases with MongoDB and RethinkDB
 - Key-value stores with Riak and Redis
 - Standards compliance, best practices, and code validation
- Testing
 - Behavior-driven development with RSpec, Capybara, Laika, Mocha, PhantomJS, etc.
 - Feature, integration, request, controller, model, etc. specifications
 - BDD vs. TDD
 - Continuous integration

Teacher training

We will learn web development using the exact techniques that the students will later use to teach other students. We will focus on *skills vs. knowledge*. Knowledge is easily self-obtained; skills are better learned through mentorship and teamwork.

We will also use the **Just In Time** training method: we will learn only what we need to know, *exactly* when we need to know it, and we'll apply it immediately and repeatedly. The entire process will be iterative and project-based, increasing depth of knowledge and skill on each successive iteration.

Here are some of the many teaching techniques and skills we will cover throughout the program:

- Active learning
- Adult learning strategies
- Asynchronous learning
- Authentic questions/tasks
- Blended learning

- Coaching
- Cooperative learning
- Curricular integration
- Curriculum development
- Differentiated instruction
- Experiential education
- Guided practice
- Mentoring
- Outcomes-based learning
- Project-based learning
- Question answering/generation
- Reciprocal teaching
- Service learning
- Supplemental instruction
- Universal design for learning

Come with an open mind, an open heart, and ready to work hard.