## Document Purpose

The purpose of this document is to explain why and how programming will be included in the Interaction Design 2: Systems class.

## Why Programming?

There are two goals in including programming in the Systems class:

1. Use the necessarily systematic nature of programming best practices as tangible introduction to systems design principles
2. Provide the students with useful software prototyping skills

## How Will Programming be Included in the Curriculum?

The primary framework for introducing programming into the Systems class will be as a prototyping tool. Front-end web technologies (HTML/CSS/Javascript) will be taught as a system for rapidly creating interactive and testable interface artifacts. Students should come away with the understanding that while web technologies are only one of many interface development environments, essentially any experience can be emulated using these technologies.

## The Plan

### Modeling Screen Experiences

Students will use standard web technologies and open source frameworks to model user interfaces both in terms of layout and interactivity.

#### Environment: Front-End Prototyping

The goal here will be to get students comfortable creating an environment where they can produce web pages with a standards-based separation between structure, style and interactivity. This should include familiarity with

* a simple IDE such as Panic’s Coda (introduce with trial version, students will purchase if they wish or learn to work with a simple text editor or freeware editor)
* basic HTML
* basic CSS (or introduce a CSS framework?)
* basic Javascript, but quickly introduce jQuery

**Lectures/StudiosAssignments (draft - timing, details, relative importance TBD)**

1. How the web works
   1. IP protocol
   2. client-server architecture
2. Browser basics
   1. The DOM tree
   2. How a page is rendered
3. Web Standards
   1. Separation b/n markup, styles, scripts
   2. Semantic markup
4. Static Structures
   1. HTML entities
   2. Programmer-defined page objects
   3. CSS Grid/Framework
5. Dynamics
   1. User Events
   2. Partial page re-draws/AJAX
   3. Page reloads
6. Frameworks
   1. jQuery
   2. 960Grid (Let’s consider using Foundation framwork from zurb for responsive layouts)
   3. Less/SASS (I like the idea but think introducing css and a compiled version is a tough ask)
7. How to turn a wireframe into a web page
   1. structure with HTML
   2. layout and style with CSS
   3. Using jQuery to animate objects, respond to user events
8. Strategies for simulating
   1. faking transactions
   2. different devices
   3. responsive strategies
9. As part of final team assignment, prototype some aspect of system team has designed