About this course:

Learn the basic web technologies that control content (HTML), appearance (CSS), and behavior (JavaScript), as well as what is needed to create an interactive website using HTML forms, server-side scripting (using PHP) and basic database operations (using MySQL). Beyond the "how to," learn about good design practices for content preparation, navigation design and site management, plus strategies for supporting visitors connecting with everything from smart phones to desktops (responsive web design). Practice your developing skills by executing a series of projects that gradually build a personal website on the UW web servers. Cap it off with a team project where you can mix in your own ideas about online data or media services, social networks and design, and see what you can create.

Prerequisites:

Students registering for the course should be computer literate. That is, they should have an understanding of basic word-processing and text editing, file transfer, use of email, use of a web-browser, and basic use of image editing tools (e.g. Photoshop).

Goals for the quarter:

• To understand the fundamental technologies that underpin the “wild wild web.”
• To understand current web capabilities, technologies and limitations.
• To develop hands-on skill and judgement in design/construction of simple web sites.
• To become confident and capable of creating or maintaining content using simple tools.
• To create a web site which demonstrates what you have learned.

Grading:

Grading is based on individual assignments (35%), a team capstone project (30%), a final exam (25%), and an assessment of in-class and on-line participation (10%).

Inspection v. Imitation:

Aside from graphics, most web content is text-based. Browsers all include some sort of “View Source” option that lets you inspect that text, making it relatively easy to learn by looking at (inspecting) the work of others. Of course, that means it is also fairly easy to copy and paste that work (imitating). In this course you are expected to use the web as a resource (of course!) but you are also expected to be the author of your own pages (knowing what is there and what it does). So, do look at the work of others. Do ask...
questions about it. **Do** learn from it. But, **do not** copy and paste (plagiarize) the work of others! Feel free to study it, break it, and learn to replicate it on your own, but make sure you understand it. For one thing, some of the “stuff” out there was written to prior standards (including some of my course web pages!), so you may find yourself using or imitating “bad” practices rather than good ones. If there is doubt about authorship, you may be asked to explicate your code. **If you are discovered to be presenting ‘found’ content as your own you could receive a course grade of 0.0. Really.**

**Mystery, Magic, and Mastery**

Web technologies have grown quite complicated over the past two decades, while retaining a high level of “backwards compatibility.” The result is that a simple technological foundation has produced a very complex environment and remains something of a mystery and barrier to entry. Using pre-digested packages of templates and software, such as WordPress, visually and behaviorally complex pages are now relatively routine, but they only work so long as the author colors “inside the lines”. We will resist the urge to adopt magical black boxes of pre-existing code where we can (you can always do that later), and strive for mastery of the fundamentals first, in the belief that such mastery will serve you better in the long run.

**Browsers, Standards & Testing**

The web is not a *product* in the same way that Microsoft Word and Adobe Photoshop are products. No single commercial enterprise owns or controls all the parts that make it work. This causes some confusion—different browsers don’t always render pages in the same way, and there is significant temptation to introduce “features” that are unique to a single browser in the hopes of gaining more market share, etc.

Unfortunately, one of the most egregious offenders in terms of implementing standards correctly and without special features, has been Microsoft. The old MS Internet Explorer browser has been the subject of lawsuits because of the business model through which it was distributed, but it has also been the subject of much criticism in professional web-design circles because of the non-standard features and ‘alternative’ interpretations of standards. The new Microsoft Edge browser included in Windows 10 is reputed to do much better in terms of standards, but remains a “work in progress.”

Good web-design practice suggests that you should strive to adhere to established standards and test your web designs in an array of browsers that matches your visitors’, including new revisions to browsers as they become available. This can make testing and updating pages into a big part of your life as the web evolves. However, if you avoid idiosyncratic code, the need for exhaustive testing and modification declines.

Browsers are just part of the variability of the web. Other factors include screen size (phone v. desktop), connection speed (on campus vs. off), font availability, and text
encoding. Don’t assume everyone can see as well as you do, or browses with a maximized window, or has a super-fast connection. These are central design sensitivities that need to be part of your approach to web design.

For our purposes, the Chrome browser will be the browser to target, but you should test important projects against as many browsers as possible. Because the history of browsers has been fairly turbulent, and HTML has gone through some significant changes (we’ll discuss some of these), browsers are VERY tolerant of “aberrant” code, which is unfortunate, as it means you will often see poor code “in the wild” when you examine someone else’s HTML, and the browser will still produce OK results even when the HTML is poor. For this reason we will aim to satisfy an online page validation tool available from the W3C:

http://validator.w3.org

Using Character Encoding “utf-8” and Document Type “HTML 5”. We’ll learn how to make a ‘validation link’ part of our standard ‘blank’ web page.

Need Help? Don’t Panic! (of Problems, Charrettes & Getting Help)

I am well aware that learning to do new things with computer software can be a very frustrating experience and that “Murphy’s Law” (“whatever can go wrong, will”) seems to operate best in the middle of the night. Part of what you will learn is how to find your way through complex problems by breaking them down into parts you can solve independently.

Still, you will bump into problems requiring help. Some of that you can find on the web; some will come from classmates; and some will come from consulting with me. All that consultation takes time. If you’re working on the project the night before it is due you may not have much time, even when the problem turns out to be simple. So—big piece of advice—do some work on this class every day, focusing on the new stuff. That’ll give you time to address problems without panic, and it shouldn’t take all that long.

Class Mailing List: arch482a_wi18@uw.edu
Brian’s Email: brj@uw.edu
Brian’s Office: Arc Hall G55, phone: 206-543-2132

Office Hours

I don’t set rigid office hours, but I am often in my office, so feel free to stop in. I am also generally available to students via email and through the course Bulletin Board. If you have questions or are struggling with a concept, feel free to come see me (repeatedly if necessary!), or email me. I do often respond to email at night, but there are also other classes I need to support. I appreciate your patience under stress.

Course website: http://quicksilver.be.washington.edu/courses/arch482