Certificate in HTML5, CSS3 & Responsive Design for Web Development

HTML5 Application Programming Interfaces (APIs) (300) Syllabus

Contact Information
Instructor: David Anderson
Email: dma36@uw.edu

Course Description:
This course covers the major JavaScript APIs in the HTML5 standard, which you can use to create rich multimedia Web sites and applications that shine on a wide range of devices. The course will also introduce you to some popular frameworks and libraries that make CSS and JavaScript easier to use.

Course Learning Objectives:
You will learn

- how to use AJAX to communicate with a server
- how to draw on the page using Canvas and SVG graphics
- how to play and control audio and video
- how to interact with touchscreens, cameras, GPS, and accelerometers
- how to store and retrieve data
- how to use WebSockets for client-server messaging

Course Format:
Classes will combine lectures, discussions, and student presentations. You will explore each topic in weekly homework assignments. You will also give two class presentations.

Course Materials:
These books are not required, but you will find them valuable.

Eric Freeman & Elizabeth Robson, *Head First HTML5 Programming*, O'Reilly, 2011 (ISBN: 978-1-449-39054-9) is a good introduction to JavaScript and several of the topics in this course.

Bruce Lawson & Remy Sharp, *Introducing HTML5*, New Riders, 2011 (978-0-321-68729-6) surveys most of what's new in HTML5, including most of the APIs we will discuss.

David Herman, *Effective JavaScript*, Addison-Wesley, 2012 (978-0-321-81218-6) explains many aspects of the language as it details best practices and patterns.

David Geary, *Core HTML5 Canvas*, Prentice Hall, 2012 (978-0-13-276161-1) covers Canvas graphics and animation in depth.

Lecture notes will be online at https://pure-coast-1257.herokuapp.com and will include links to a more documentation and resources.

This class will also have an online newsgroup where we can share ideas, questions, solutions, and information. I will participate, but I encourage you to help each other and to feel free to ask for help.

Technical Requirements:
This is a hands-on course with in-class programming exercises as well as assignments to be completed outside of class. Students will be required to bring a laptop. Linux or Mac OS X are optimal, although Windows XP or later may also be used. (If you have Windows, consider installing Linux in a VirtualBox or as a dual-boot option.)

Many of the course topics and most modern Web development also pertain to mobile devices, so please bring any smart phones and tablets you can.

Schedule: Topics and Assignments by Date:
April 9: Introductions. Setting up a public Web site
April 16: Client-server communication with AJAX
April 23: Storing user data. Canvas graphics
April 30: Canvas graphics
May 7: SVG graphics. Data-Driven Documents (D3)
May 14: Audio and video. Cameras and microphones
May 21: Geolocation. Device orientation. Touch events
May 28:. Web sockets. Going offline with the Cache manifest
June 4: Web workers
June 11: Student presentations

Student Assessment:
UW Professional & Continuing Education courses require at least 80% attendance, i. e. eight of the ten classes in this course.

Your primary assignment for this class is to develop one or two Web sites of your own devising. The subject matter, content, styling, and behavior will be up to you. The site(s) should display your command of modern, semantic HTML, CSS, and some of the JavaScript coding techniques and APIs you have been learning in this course.

Policies and Values:
Our class is a community. I will spend much more time at the front of the room than you, but your active participation will benefit you, your classmates, and me. Anytime a question comes to mind, ask; someone else is probably wondering the same thing. Your contributions are more than welcome; they are essential. Feel free to disagree with me and with each other. But do so respectfully.

Ultimately your work needs to be your own—otherwise you learn nothing—but it is fine to ask or look for help. If you do borrow some code, from a classmate, contributor to StackOverflow, or elsewhere, be sure you understand what it does and why it solves your problem, and add a comment in your code citing the source. That said, please do share and help each other.