ITMD 465/565
Rich Internet Applications

Lecture 1

Fall 2019 – August 21, 2019

Tonight’s Agenda

- Course Introduction & Syllabus Review
- HTML, CSS, JavaScript Introductory Review

Course Introduction

Primarily a JavaScript Programming Class

- We will look at programming using the JavaScript Language, JavaScript frameworks and libraries, and HTML5 JS APIs
- The class will be very heavy on programming with the JavaScript Language and will require you to investigate APIs and documentation on your own, beyond what we talk about in class.
- You are expected to have some level of experience with programming, preferably object-oriented, but it doesn't have to be JavaScript. We will be covering the basics of the language and syntax but you should have experience with the logic and control flow in programming.
- This is not a design class focused on HTML and CSS layout. **We will not be teaching HTML and CSS in detail. You are expected to already be proficient with these topics.**
- This course will probably be somewhat difficult at times. I will try to make it as hard as it needs to be for you to learn skills you need as a front-end JS developer.

Covering

- Topics may include:
  - JavaScript Programming: ES5, ES6+
  - JavaScript Libraries and Frameworks
  - HTML5 JS APIs
  - Web Interactive Graphics (Canvas, 2D, 3D)
  - Web Media (Video, Audio, Images)
  - JavaScript DOM Manipulation
  - ReactJS
  - NodeJS based tools
  - and more

Not Covering

- Server Side Programming
  - .Net
  - PHP
  - JavaEE
  - Python, Ruby, Node JS, etc.
- SQL and Databases
- Graphic Design
  - This is more of a “programming” course than an “art” course
  - Applications still need to be functional and flow well
- Flash or other multimedia plugins

Course Intro

- I will cover some theory and more practical examples.
- I will be strictly following the policies in the syllabus this term
- Grades need to be earned
  - A – You successfully completed everything including the harder requirements and showed a mastering of the material.
  - B – You successfully completed all basic requirements
  - C – Your project works but is missing key requirements
  - D – Your project doesn't work and/or is missing evidence that many features would work
  - E – You didn't demonstrate any mastery of the material to an acceptable level
  - You will not get an A for showing up, doing the basic requirements. You need to show a mastering of the material.
  - See syllabus

Participation/Attendance

- Through Blackboard discussion groups
  - I may occasionally post forums on topics from time to time for discussion
  - They may be graded for frequency and quality
- Attendance
  - In class students are expected to show up for class
    - Attendance will be taken in class, Quizzes will also be a measure of attendance. There are no excused absences as far as attendance is concerned.
  - Online students are expected to watch the video shortly after the live class
  - All lecture content and videos will be posted the next day

Academic Honesty

- This is something I will take seriously and you probably won't like the results.
- The official Policy is in the Syllabus
- No Tolerance for:
  - Sharing Codebase with classmates (you may discuss with others, only use pseudo code, don't give your code.)
  - Stealing Code
  - Using outside content **without attribution** and within a reasonable amount
  - Unauthorized collaboration
- 1st Offense: 0 for the offending assignment, academic honesty violation filed with department
- 2nd Offense: E for the entire course, academic honesty violation filed with the department – you may get dropped from the program or university
- This is the policy regardless of GPA, Visa Status, How well you did on other assignments.

Review Syllabus

- Let’s review the rest of the syllabus

Hardware Requirements

- You will need access to a computer of some kind to develop on.
  - Need to be able to install software. Will be needed when discussing NodeJS
  - Any OS: Windows, Mac, Linux
  - There are computer labs on campus. I don’t know if any has NodeJS installed or what versions. Check OTS Lab software list on their website.
- Your computer will need a developer friendly web browser
  - Install Chrome and/or Firefox
- Your computer will need a plain text editor
  - One made for programming use. Must edit in plain text UTF-8 format.
  - VS Code, Notepad++, Sublime Text, Atom, many others
  - Do not use wordpad, MS Office, or any RTF editor (no bold, italics, etc)
- You will also need to install NodeJS and git scm
- A development web server would be helpful. There is a basic node package (http-server), PHP has one built in, VMs, docker containers, or use a tool like XAMPP, MAMP to provide a complete isolated web stack.

Tips for success

- Largely what will determine what you get out of this course is by your own research.
- I will show examples and talk about some of the theory but will not be able to cover every topic in great detail. You will need to branch out on your own and do some extra research.
- If you have a question you are probably not the only one.
  1. Google the exact problem, probably find stackoverflow
  2. Check the discussion board if there is one
  3. Send TA an email or visit office hours (If we have a TA)
  4. Send me an email
- Do not wait until the last minute to start any projects, start early

Questions?

Web Applications

Web Applications vs Site

- Web Site
  - A location (endpoint) on the WWW
  - Typically delivers static and dynamic content over HTTP
- Web Application
  - A tool, utility or other device in that a significant amount of its application is run through HTTP and browser based protocols and languages.
  - Goes beyond just delivering content
- Rich Web Application
  - A tool, utility or other device that has a significant amount of client side functionality
  - Designed to feel more like a desktop application then web site
  - May include offline functionality

History of Internet Applications

Getting to HTML5

HTML

- Created by Tim Berners-Lee while working at CERN
- Wrote the original internet-based hypertext system proposal memo in 1989.
- First description of HTML was a document called “HTML Tags” on 1991.
- Strongly influenced by SGML (standard generalized markup language) based system in use at CERN
- He created HTML and HTTP protocol
- Set of nested “tags” that look like xml
- **HTML** is the **STRUCTURAL** layer of a web page

HTML

- Major Version History
- HTML Tags – Informal CERN document with 18 tags, October 1991
- HTML 1.1 – First draft with a version number, November 1992
- HTML 2.0 – Published as IETF RFC 1866, November 1995
- HTML 3.2 – Published as W3C recommendation, January 1997
- HTML 4.0 – Published as W3C recommendation, December 1997 and reissued in April 1998 with minor edits
- HTML 4.1 – Published as W3C recommendation, December 1999
- HTML 5 – Published as W3C recommendation, October 2014
- XHTML 1.0 – Published as W3C recommendation, January 2000 and revised in April 2002
- XHTML 1.1 – Published as W3C recommendation, May 2001 and based on XHTML 1.0 Strict
- XHTML 2.0 – Was a working draft but abandoned in 2009 in favor of HTML5

CSS

- Used to apply visual styles to a web page and its structure
- **CSS** is the **PRESENTATION** layer of a web page
- CSS 2 – Published as a recommendation May 1998
- CSS 2.1 – Published as a recommendation June 2011
    - Single large specification
- CSS 3 – Current standard being developed
    - Full specification is split up into modules
    - Comprised of many small modules in working drafts
    - Started being worked on when CSS 2 was published
    - Earliest CSS 3 drafts were June 1999, working draft was started in 2001
- https://www.w3.org/TR/2001/WD-css3-roadmap-20010523/
- https://www.w3.org/Style/CSS/specs.en.html

JavaScript

- JavaScript – ECMA compliant Object Oriented language
- http://en.wikipedia.org/wiki/JavaScript
- **JavaScript** or JS is the **BEHAVIOR** layer of your web page.
- JavaScript is a programming language.
- It is capable of the following
- Moving things around the screen, modifying the DOM, updating content
- Manipulating HTML tags and CSS rules
- Implementing browser APIs like location and local storage.
- And much more...

JavaScript

- Originally developed by Brendan Eich at Netscape
- Developed with code name Mocha, then officially called LiveScript in beta Netscape Navigator 2 in Sept 1995.
- Renamed JavaScript before final release of Netscape Navigator 2 (B3). Marketing ploy by Netscape to cash in on the popularity of the new language Java which was also included in Netscape 2.
- November 1996 – Netscape submitted JavaScript to Ecma International to be considered a standard.
- June 1997 – Ecma International published the ECMA-262 Specification in version 1.
- Current widely supported version of the ECMAScript standard is 5.1 and was published in June 2011
- https://en.wikipedia.org/wiki/JavaScript
- ECMAScript 6 published June 2015, ECMAScript 7 published June 2016, ES8 published June 2017
- https://en.wikipedia.org/wiki/ECMAScript

JavaScript

- We are always working towards new versions of JavaScript
- ECMAScript 6th Edition – 2015 Language Specification was the major update
- Adds new syntax and features
- Many new browsers offer direct support but not all
- Can compile ES6 to ES5 with many available tools
- We will discuss some differences later in class
- http://www.ecma-international.org/ecma-262/6.0/

Assignments

Reading/Assignments

- Download and look at books I will place in Blackboard
- If basic JavaScript language features or programming in general are not familiar to you should start reading those books asap. It is expected you have a basic understanding of programming concepts.
- If you don’t know how to use git start doing some tutorials immediately.
- Read to understand JavaScript versions https://benmccormick.org/2015/09/14/es5-es6-es2016-es-next-whats-going-on-with-javascript-versioning/
- Make sure you have your text editor installed and working
- Make sure you have Chrome or Firefox installed and have tried using the developer tools to inspect a web page and know where the JS console is.
- You may want to have a local development server environment ready if needed.
- Get node and git installed and working.