Website Programming syllabus

csci-215-websiteprogramming-2018.github.io/syllabus/

If you are a teacher or interested in the design of the course, see the meta document.

- **Course:** CSCI-215-WebsiteProgramming
- **Instructor:** Keith Briggs
- **Need help?**
  - Slack

    This is a communications medium for students to interact with one another and myself. I encourage students to interact with one another as a resource as well as provide instructor answers.
  - Office Hours: Tue & Thur- 4:30 to 5:30 (By request)
  - Email for 1-on-1 help, or to set up a time to meet
- Syllabus as PDF

Course Description

A course teaching the design and development of interactive Web pages and client-side based Web programs using Rich Internet Applications (RIAs). Students will learn how to manage elements of a Web page using the Document Object Model (DOM), create and validate forms and communicate with Web servers using current Web technologies. Lectures three hours per week.

Prerequisites

- CSCI 115-Web Design
- Understanding of HTML and CSS
- Working knowledge of a text editor or IDE such as Sublime Text or WebStorm
- Working knowledge of Git and Github

Beginner Materials.

Course Overview

This course builds on your Web Design knowledge by adding a dynamic element to your websites. This is achieved with extensive use of the Javascript programming language and Javascript libraries such as jQuery. Modern tools such as Webstorm (or equivilent editor), Git, GitHub, and the Debugger will be heavily used. Slack will be used so that students and instructor may have a collaborative Question and Answer environment. I expect maximum participation with students answering other students' questions if they know the answer. All projects will be administered through GitHub, where students are expected to initiate the

assignment and complete under Version Control (accomplished with Git and pushed to GitHub). Completing the assignment as assigned will not be enough for an A+. Students must add at least one “WOW” feature to the project and must be successfully demo’d to the Instructor. Topics covered include:

- Write rich web-based programs that execute in a standard web browser
- Write web-programs that use JavaScript and HTML
- Using JavaScript and jQuery to interact with the document object model (DOM)
- Interacting with the user through events
- Communicating with a server
- Using the browser’s Developer Tools to assist in debugging
- Introduction to AJAX
- and More…

Course Outcomes

- Develop client-side web programs using variables, basic data structures (arrays, multidimensional arrays, and associative arrays), conditional logic and branching, loops, etc.
- Create and use functions.
- Manipulate and traverse DOM.
- Create interactive web-programs (create and handle user or machine generated events).
- Create dynamic web-based programs (programmatically add, remove, and update nodes in the DOM).
- Write client-side form verification scripts.

Homework/Projects

All homework assignments are listed within the Course Outline.

Most Thursdays there will be an in-class lab assignment. You will be given 50 minutes to complete and upload the assignment to GitHub. Edits can be made until the next Sunday, but there must be a commit/push before the end of class the day of the lab. In general, the lab assignment will be a coding exercise (JavaScript, HTML, or both), however a short quiz may also be given. The quiz may include multiplechoice, fill in the blank, or short answer questions. For example, 15 minutes of the lab time is a quiz, and the remaining 35 minutes is a coding problem.

Workflow

1. Navigate to the proper project in CSCI-215-WebsiteProgramming GitHub Organization account.
2. Fork the project into your personal GitHub account.
3. Clone the project onto your local workstation.

4. Read the instructions provided by the README file.
5. Modify the files to complete your assignment.
6. Test your solution with the Firefox browser.
7. Edit the README file to communicate anything you need to me.
8. Make sure ALL of your code is committed.
9. Push/sync up to GitHub (be sure to check GitHub to ensure all files were pushed correctly).

All assignments are due at the start of class on the specified date.

- You can continue to push fixes and improvements until the date (with no penalty) or for one week after (with letter grade penalty) – just add a comment in the commit message to let me know it's been updated.
- I will leave appropriate feedback as an Issue inside of your repository.

Requirements

- All code must execute and render properly in the Firefox browser.
- Assignments must be completed by the due date. Late assignments will only be accepted one week late and will be marked down a whole letter grade.

Grading

The final grade for the course is based on 6 grades as follows:

- Exam 1……15%
- Exam 2……15%
- Final Exam……..15%
- Homework…….15%
- Labs……..15%
- Final Project…..15%
- Attendance…….. 5%
- Participation……5%

Grading Scale

- A 90 - 100
- B+ 85 - 89
- B 80 - 84
- C+ 75 - 79
- C 70 - 74
- D+ 65 - 69
- D 60- 64
- F 0 - 59

Course Schedule (tentative)

Topics Covered:

- Aug 21 - 23 Introduction to course, Semester tools, Review of HTML, GIT demo
- Aug 28 - 30 Review HTML and CSS, Lab 01
- Sep 04 - 06 Responsive web design, Responsive frameworks,Document Object Model, Lab 02
- Sep 11 - 13 Introduction to Javascript programming, Javascript data types, Javascript data structures, Lab 03
- Sep 18 - 20 Javascript data types, Javascript data structures, Javascript control flow, Lab 04
- Sep 25 - 27 Javascript control flow, Exam review
- Oct 02 **EXAM #1**
- Oct 04 Manipulating the DOM, Lab 05
- Oct 09 - 11 Event Handling, Form validaton, Lab 06 (part 1 and 2)
- Oct 16 - 18 Event Object, Animation, Lab 07
- Oct 23 - 25 Lab 07 review, Event Object, JSON intro
- Oct 30 Exam review
- Nov 01 **EXAM #2**
- Nov 06 **FALL Break**
- Nov 08 Intro to jQuery Lab 08 (jQuery lab)
- Nov 13 - 15 jQuery events, jQuery animation, group time
- Nov 20 Server side topics, introduction to AJAX
- Nov 22 **Thanksgiving Break**
- Nov 27 - 29 Student presentations
- Dec 04

Resources

Required Reading

- Beginning JavaScript, 5th Edition by Jeremy McPeak
- Javascript and jQuery, The Missing Manual, 3rd Edition by David Sawyer MacFarland

Recommended Reading

Beginner Materials

This class assumes you are confident with this material, but in case you need a brush-up…

- Udacity Courses
- Developer's Blog

Tools

GitHub

Reference

Statements on Plagiarism

SCPS

> The College of Charleston takes plagiarism very seriously and regards it as a form of fraud. The definition of plagiarism that has been adopted by the School of Continuing and Professional Studies is as follows: “Plagiarism is presenting someone else’s work as though it were one’s own. More specifically, plagiarism is to present as one’s own words quoted without quotation marks from another writer; a paraphrased passage from another writer’s work; or facts or ideas gathered, organized, and reported by someone else, orally and/or in writing. Since plagiarism is a matter of fact, not of the student’s intention, it is crucial that acknowledgement of the sources be accurate and complete. Even where there is not a conscious intention to deceive, the failure to make appropriate acknowledgement constitutes plagiarism. Penalties for plagiarism range from failure for a paper or course to dismissal from the University.

Instructor

Reuse and building upon ideas or code are major parts of modern software development. As a professional programmer you will never write anything from scratch. This class is structured such that all solutions are public. You are encouraged to learn from the work of your peers. I won’t hunt down people who are simply copying-and-pasting solutions, because without challenging themselves, they are simply wasting their time and money taking this class.

Please respect the terms of use and/or license of any code you find, and if you reimplement or duplicate an algorithm or code from elsewhere, credit the original source with an inline comment.

Pairing Tips

- **Groups of two, unless there is an odd number of students**
- **Agree on an editor and environment that you’re both comfortable with**
- **Use Github for collaboration, branch/merge model is recommended**
- **Meet regularly. Use Google Hangouts when you cannot meet in person (Hint: Sharing your screen is very useful)**