Web Advanced: JavaScript (Section B) Fall 2016
MFA Design and Technology - Parsons School of Design

PGTE 5505, Section B, CRN 5264

- **Course Dates and Times:** Wednesdays, from 7:00 pm to 9:40 pm
- **Location:** 6 East 16th Street, Room 1208
- **Faculty:** Umi Syam
- **Faculty e-mail:** umi@newschool.edu
- **Teaching Assistant:** Soomi Lee (lees105@newschool.edu)

Course Description

Applications rarely are comprised of a tangled set of static files. They're dynamic. They update. They remember things. The course will introduce the concepts of a database and Web applications as a set of design problems. The class will investigate server-side applications, third-party data sources and APIs and how applications can become dynamic and highly functional. To approach the design and construction of applications that derive from data sources and databases, this course builds upon an essential knowledge of Web Standards, user interface design, and usability to support the creation of functional and responsive projects for the Web across multiple devices and platforms. Students will incorporate interaction design, visual design, user experience and code to rapidly sculpt data into content. By the end of the course, students should be able to design, build and deploy data-driven Web applications. The course employs a pedagogy suitable for designers, artists and technologists who seek an empowerment achieved by being able to build data-driven applications.

We will learn JavaScript properly. Then, we will learn useful design patterns. Then we will pick up useful tools for making cool things better.

Course Outline

Topics of the session are subject to change as the course progresses. See each week's folder on our Github class repo for an expanded view of classes and assignments.

class #	date	topic	assignment
1	8/31/2016	JS Basics Part 1	Choose: Calculator/Quiz App
2	9/7/2016	JS Basics Part 2 + Web App	A simple API-driven web app
3	9/14/2016	Single-Page Application	Add more filters to our exercise

4	9/21/2016	Review + Responsive Web	Turn any of the 1-3 homework you've made, responsive
5	9/28/2016	Node.js Intro + scraping	Web scraping/HTTP Request
6	10/5/2016	Database Intro	Midterm Proposal Due
	10/12/201	No Class - Yom Kippur	
7	10/19/2016	Midterm Presentations	-
8	10/26/2016	Websockets 1	TBA
9	11/2/2016	Websockets 2	TBA
10	11/9/2016	JS Libraries: P5.js	TBA
11	11/16/2016	Guest Lecture: Three.js	TBA
12	11/22/2016 (Tuesday! Because it's Thanksgiving week.)	JS Libraries: D3.js	Final Project proposal due
13	11/30/2016	JS Libraries: D3.js	-
14	12/7/2016	Class Review + Workshop	-
15	12/14/2016	Final Presentations	Final Project

Class 1 - 8/31/2016 - Javascript Basics 1

- Introduction: Welcome to the world of web!
- Javascript Part 1: Back to Basics.
    - JS definitions, Variable and data types, Arrays vs Objects, Functions: declarations and expressions, Iteration, Conditionals statement, Scope and closure, Timing functions
- Preparing your coding environment
- Github setup for assignment submission.
    - Github Desktop overview
    - Create individual repo
    - Git adding submodule to the homework repo

Class 2 - 9/7/2016 - Javascript Basics 2

- Javascript Part 2: A Little More In-Depth.

    - Immediately-Invoked Function Expression, Introduction to Module Pattern, Namespacing your app, Private? Public?, External modules
- Web App
    - Bootstrappers
    - Naming conventions for HTML/CSS and JS
    - Structuring an app: Page load, Event listeners
    - Document Object Model (DOM)
    - Libraries: What is a library, cdnjs, Vanilla JS vs JQuery
    - Fetching data from API
    - Loading and displaying JSON data

Class 3 - 9/14/2016 - Single-Page Application

- Understanding Templating Engine
    - Handlebars intro
    - Binding data to the DOM
- Single-Page Application
    - Separating logics from content
    - Creating the Controller: render different pages based on its content
    - Creating the View: html structure
    - hashchange
    - Implementing filter checkboxes

Class 4 - 9/21/2016 - Responsive Web Design

- Responsive Web Design
    - Device Detection
    - Redirect
    - Viewport
    - Media queries

Class 5 - 9/28/2016 - Server-side

- NodeJS + HTTP Server
    - Simple Hello World
    - Basic Routing
- NodeJS + Express API
    - Node Package Manager (npm)
    - Initializing & Installing packages, gitignore the modules
    - HTTP methods: GET vs POST
    - Router

    - API
- NodeJS + Express Server Basic Setup
    - Static files
    - body-parser module
    - JQuery post
- Scraping the web with NodeJS + X-Ray

Class 6 - 10/5/2016 - Databases

- Relational vs Non-Relational (NoSQL) Databases
    - What, when to use, the data model, the form, etc.
- Install MongoDB
- Using Mongo Shell
- Using mLab to access MongoDB in the cloud
- Performing CRUD operations with:
    - Node.js + MongoDB
    - Node.js + MongoDB + Mongoose
        - Mongoose as a object modelling tool
        - Learn how to make Schema
    - Node.js + MongoDB + Express + EJS view engine

Class 7 - 10/19/2016 - Midterm Presentations

Class 8 - 10/26/2016 - Websockets 1

- Fast client-server communication using WebSockets.
- When to use WebSockets (versus AJAX) and the many applications of it.
- Working session: Simple Chat
    - Application Diagram
    - Adding the socket.io library
    - Messaging: Listeners

Class 9 - 11/2/2016 - Websockets 2

- Continue websockets review
- Create Multiplayer game
    - Application Diagram
    - Updating the server data
    - Updating the client view

Class 10 - 11/9/2016 - p5.js

- P5 Basic Introduction
- Transitioning from Processing to p5.js
- Using node.js to serve a p5 sketch
- Use websockets to enable a synchronous connection between clients

Class 11 - 11/16/2016 - Three.js

- JS Libraries: Three.js basics
- Intro to 3D and WebGL
    - 3D coordinate system
    - Camera & view
    - 3D Primitive shapes
    - Texture, Text
    - Lighting and Materials
- Use node.js and socket.io to serve a Three.js visualization output

Class 12 - 11/22/2016 - d3.js

- D3 Intro & Background
- D3 Data-Driven Building Blocks
- Creating & Placing Visual Elements
- D3 Events, Transitions & Animations
- D3,AJAX, & Servers
- Step By Step Goal Chart Build

Class 13 - 11/30/2016 - d3.js

- D3.js General Update Pattern
- D3.js Reusable Chart Pattern
- D3.js Layouts
- D3.js Behaviors
- D3.js Mapping

Class 14 - 12/7/2016 - Class Review + Workshop

Class 15 - 12/14/2016 - Final Project Presentation

Learning Outcomes

By the successful completion of this course, students will be able to:

- Have a better understanding of how web applications work
- Be comfortable with coding in JavaScript, from server to front-end.
- Develop working prototypes and publish them.
- Select and apply different technologies in order to realize their ideas.

Assessable Tasks

All assignments must be submitted through Github to the class repository.

Weekly Assignments

Please refer to each week's readme file on Github repo for a list of weekly assignments.

Projects

There will be two big projects throughout the semester:

Midterm Project

Build a data-driven web application. Your project must:

- At least read data from an external source — could be from a JSON file, from a database, or whatever you feel more comfortable with.
- Have no less than 2 different "screens." If it's a dynamic html website, have at least 2 pages within the same url, but just different "#"/hash page. If you're building a canvas-based game, maybe you have 2 scenes? If it's a datavis, 2 charts or 2 ways of visualizing the same data and so on.
- Respond to user interaction.

Final Project

Team up with another student and build an interactive web app that meets these requirements:

- Interacts with the server-side using NodeJS
- Includes bi-directional communication using websockets, any of these are valid: client <--> server, hardware <--> server communication. We're looking for true communication, sending and receiving data/messages/actions/inputs back and forth.
- Outputs a visualization on the front-end side using either P5.js, D3.js, or Three.js.

Using database is *optional* in this case.

Grading Criteria

- Weekly assignments are intended to assess your development throughout the course.

Criteria	%
Does it meet the requirements outlined in the assignment?	40%
Does it work?	40%
Was it submitted on time?	20%

- Projects are intended to assess your capacity of creatively applying your skills.

Criteria	Details	%
Does it meet the requirements outlined in the assignment?		20%
Does it work?		20%
Is the purpose of the project clear?	Interests may vary enormously among students. The only way for me to evaluate a project is to understand what you were trying to achieve.	10%
Is it functional?	Unless your intention is to confuse people — in which case you should clearly communicate that —, your project should be usable.	10%
Is it creative?	Even if you're inspired by something that already exists, think about how you can develop the idea further.	10%
Does it show development?	If something is not quite right — conceptually or technically — work to improve it.	10%
Was it submitted on time?		20%

Additional notes:

- Every time you present your project in class, slides are not necessary.
- Your cleanliness and the organization of your code is important, but it won’t be included as my grading criteria. However, you should still follow the recommended

coding practises to help you be more efficient. See Gabriel Gianordoli's coding advice for more.

- I will not check your code line by line looking for bugs. So please submit your assignments as functional as possible. If the page just crashes on loading I will not be able to understand whether or not you have accomplished something. In those cases, comment out the bug and leave a note with details. For instance: "*This part should display the data, but I wasn't able to finish it."*

Some Useful Advice

DT is great, because you have the chance to do (almost) anything you want for 2 years. And assignments are great for keeping up with the skills you're learning. From now on, I'd like you to start thinking of every assignment as something that will help you build up your portfolio. Think of each opportunity as a chance to find out or go deeper into what interests you. Your projects can evolve around:

- A theme — food, gender equality, Trump, Sven Travis, etc.
- An aesthetics — vernacular web, retro-futurism, minimalism, etc.
- An approach/field — games, critical design, net-art, design thinking, etc.
- A technology — physical computing, interactive installations, databases, wearables, etc.

Even if you find out that *you're not* really interested in what you explored, that'll be a good thing to learn. Your interest in approaching each project doesn't need to be consistent, you should explore as much as you can.

Required Tools

- Google Chrome
- Sublime Text
- Terminal (Mac) / Command Prompt (PC)
- A Github account and a way to access it. Feel free to either Terminal or a desktop app — Github for Mac, Source Tree or whatever you feel comfortable with. Send me an email of your github repository so I can add you as a submodule in our Class Homework repo.

Recommended Readings

There are no required readings for this class. However, here are some recommended ones. For a more complete resource please navigate to the class repository on Github and see each week's reading recommendations.

- Interactive Data Visualization for the Web, Chapter 3: Technology Fundamentals
- How to Learn JavaScript Properly
- You Don't Know JS (Book Series)

- Model-View-Controller
- Single-Page Application
- The Past, Present, and Future of Local Storage for Web Applications
- 4 JavaScript Design Patterns You Should Know
- Immediately-Invoked Function Expression (IIFE)
- JavaScript Module Pattern: In-Depth

Final Grade Calculation

Item	%
Participation/Attendance	10%
Mini Assignments(Total)	30%
Project #2 (Midterm)	30%
Project #3 (Final)	30%

Grading Standards

Graduate

- A Work of exceptional quality
- A- Work of high quality
- B+ Very good work
- B Good work; satisfies course requirements

Satisfactory completion of a course is considered to be a grade of B or higher.

- B- Below-average work
- C+ Less than adequate work
- C Well below average work
- C- Poor work; lowest possible passing grade
- F Failure
- GM Grade missing for an individual

Grades of D are not used in graduate level courses.

Grade of W

The grade of W may be issued by the Office of the Registrar to a student who officially withdraws from a course within the applicable deadline. There is no academic penalty, but the grade will appear on the student transcript. A grade of W may also be issued by an instructor to a graduate student (except at Parsons and Mannes) who has not completed course requirements nor arranged for an Incomplete.

Grade of WF

The grade of WF is issued by an instructor to a student (all undergraduates and all graduate students) who has not attended or not completed all required work in a course but did not officially withdraw before the withdrawal deadline. It differs from an “F,” which would indicate that the student technically completed requirements but that the level of work did not qualify for a passing grade. The WF is equivalent to an F in calculating the grade point average (zero grade points), and no credit is awarded.

Grades of Incomplete

The grade of I, or temporary incomplete, may be granted to a student under unusual and extenuating circumstances, such as when the student’s academic life is interrupted by a medical or personal emergency. This mark is not given automatically but only upon the student’s request and at the discretion of the instructor. A Request for Incomplete form must be completed and signed by student and instructor. The time allowed for completion of the work and removal of the “I” mark will be set by the instructor with the following limitations:

Graduate students: Work must be completed no later than one year following the end of the class. Grades of “I” not revised in the prescribed time will be recorded as a final grade of “WF” (for Parsons and Mannes graduate students) or “N” (for all other graduate students) by the Office of the Registrar. The grade of “N” does not affect the GPA but does indicate a permanent incomplete.

Divisional, Program and Class Policies

Responsibility

Students are responsible for all assignments, even if they are absent. Late assignments, failure to complete the assignments for class discussion and/or critique, and lack of preparedness for in class discussions, presentations and/or critiques will jeopardize your successful completion of this course.

Participation

Class participation is an essential part of class and includes: keeping up with reading, assignments, projects, contributing meaningfully to class discussions, active participation in group work, and coming to class regularly and on time.

Attendance

Faculty members may fail any student who is absent for a significant portion of class time. A significant portion of class time is defined as three absences for classes that meet once per week and four absences for classes that meet two or more times per week. During intensive summer sessions a significant portion of class time is defined as two absences. Lateness or early departure from class may also translate into one full absence.

Blackboard or Canvas

Use of Blackboard may be an important resource for this class. Students should check it for announcements before coming to class each week.

Delays

In rare instances, I may be delayed arriving to class. If I have not arrived by the time class is scheduled to start, you must wait a minimum of thirty minutes for my arrival. In the event that I will miss class entirely, a sign will be posted at the classroom indicating your assignment for the next class meeting.

Electronic Devices

Use of electronic devices (phones, tablets, laptops) is permitted when the device is being used in relation to the course's work. All other uses are prohibited in the classroom and devices should be turned off before class starts.

Academic Honesty and Integrity

The New School views “academic honesty and integrity” as the duty of every member of an academic community to claim authorship for his or her own work and only for that work, and to recognize the contributions of others accurately and completely. This obligation is fundamental to the integrity of intellectual debate, and creative and academic pursuits. Academic honesty and integrity includes accurate use of quotations, as well as appropriate and explicit citation of sources in instances of paraphrasing and describing ideas, or reporting on research findings or any aspect of the work of others (including that of faculty members and other students). Academic dishonesty results from infractions of this “accurate use”. The standards of academic honesty and integrity, and citation of sources, apply to all forms of academic work, including submissions of drafts of final papers or projects. All members of the University community are expected to conduct themselves in accord with the standards of academic honesty and integrity. Please see the complete policy in the Parsons Catalog. It is the responsibility of students to learn the procedures specific to their discipline for correctly and appropriately differentiating their own work from that of others. Compromising your academic integrity may lead to serious consequences, including (but not limited to) one or more of the following: failure of the assignment, failure of the course, academic warning, disciplinary probation, suspension from the university, or dismissal from the university.

Student Disability Services (SDS)

In keeping with the University’s policy of providing equal access for students with disabilities, any student with a disability who needs academic accommodations is welcome to meet with me privately. All conversations will be kept confidential. Students requesting any accommodations will also need to meet with Jason Luchs in the Office of Student Disability Services, who will conduct an intake, and if appropriate, provide an academic accommodation notification letter to you to bring to me. SDS assists students with disabilities in need of academic and programmatic accommodations as required by the Americans with Disabilities Act of 1990 (ADA) and Section 504 of the Federal Rehabilitation Act of 1973. More at http://www.newschool.edu/studentservices/disability/