Web3 : JavaScript

PSAM 5150 - Section A
Parsons The New School for Design
School of Art, Media and Technology

Teacher: Mani Nilchiani
Email: nilchiam@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 Pre/Co-requisites:

Open to: All School of Art, Media & Technology graduate degree students; upper-level undergraduates by permission.
Pre-requisite(s): PSAM 5116 Web 2, PGTE 5410 Bootcamp, or equivalent.

Overview:

We are creative folks. We build things. The way we look at the tools for making is different than Computer Science majors. Most of us didn't come from a solid programming background and suddenly dove in the deep end, learning to survival and accomplishing goals. It was a hell of an awesome way to kick things off.

But if you are already making things for the web, and want to make them better, or want to start getting serious about the web and its language, then we will nerd out on some core concepts, programming patterns, best practices and utility belts to get you off the ground to make things more maintainable, readable and hopefully smarter.

We will: go back to basics. Take a close look at JS. See the bright and dark sides. Demystify language concepts. Learn programming patterns. Look at example code. Write code. Workshop. Examine the landscape of modern Front-End development. See client and server side JS. Pick our tools, and explore the browser as a bountiful platform for business and artistic expression. We will also have cool guests.

We do have a syllabus but we all know that one syllabus doesn't fit all. Once we hit the ground running, we will gauge things based on the dynamic of the class. We will play it by the ear to set the pace, fine tune the curriculum, and personalize based on interest.

Refer to syllabus for weekly readings. Based on what we discuss in

class, everyone's invited to post good readings, articles and resources.

You DON'T have to read. But it is highly recommended.
What you do have to do, is to submit the weekly small assignments, midterm and final projects.

Outline:

- Week 1 - 01/03/2014
    - Topic: Back to basics. Intro, data types, expressions, statements, functions, objects, arrays, scope, namespace, closures, anonymous functions, callback functions, timing functions, vanilla javascript, github recap.
    - Readings:
        - Javascript: The Good Parts - Chapter 1 and 2
        - Eloquent Javascript: Chapters 1 through 4
        - http://helephant.com/2008/08/23/javascript-anonymous-functions/
        - http://helephant.com/2008/08/19/functions-are-first-class-objects-in-javascript/
    - Assignment[Due 02/06/2014] : Write a simple calculator app capable of performing add, subtract, multiply and divide. The calculator should be namespaced. It should save the result of all operations in an array within the same namespace. For this, you will be needing to reference the namespace using the "this" keyword:

```
var myApp = {
    arr : [],
    store : function(thing) {
       this.arr.push(thing)
     }
  };
```

    We will cover "this" in future sessions. There is no

interface necessary for this assignment. But you will get extra points if you do create an interface for it.

- Week 2 - 02/06/2014
    - Topic: Programming patterns and paradigms, public, private and privileged, inheritance, Immediately Executed Function Expressions, this, Object creation, revealing module pattern. Namespacing your app.Error handling. Date. Learn a thing: Terminal, bash profile, ssh, git - using ssh
- Week 3 - 02/13/2014
    - Topic: Object oriented JS - 2 Inheritance. Prototypes. Public Static methods. Events. Programming patterns and paradigms: Behavioral. Observer, Mediator. Learn a tool: jsLint, debugging JavaScript on the front-end, effective use of Chrome Dev tools
- Week 4 - 02/20/2014
    - Topic: Build your own object oriented front-end framework for an API-driven app: Routers, Request Handlers, Views, Templating engines. Learn a thing: CSS Preprocessors : LESS, SASS, Compilers, rich graphics for the web.
- Week 5 - 02/27/2014
    - Topic: Solidify your own front-end framework. Async Module Definition, CommonJS and how to use them for better Object Oriented JavaScript. Error handling Learn a thing: Front-end dependency management with Bower. Task runners: Grunt - Gulp maybe a little bit of Browserify.
    - Our first guest Bronson Stamp will come in and talk about design. We've had some nerdy few weeks and we need a design refresher. Bronson is the digital art director at TheDailyBeast.
- Week 6 - 03/06/2014
    - Topic: Work on midterm projects. Learn a tool: Underscore.js / lodash.js
- Week 7 - 03/13/2014

    - Topic: Midterm Presentations
- Week 8 - 03/20/2014
    - Topic: Node.JS - Multithreading and asynchronous paradigms. Create your own server and make it work for you. Let's see how far we get with this. Special Note: Your next assignment will be totally free form. You can choose to do absolutely whatever you want. Use JS in any way you want. Here's your chance to pick up whatever you want. Creative coding stuff etc.
- Week 9 - 03/27/2014
    - Spring Break. No Class.
- Week 10 - 04/03/2014
    - Topic: Continue with node JS. Databases. Realtime web. Learn a tool: Yeoman
- Week 11 - 04/10/2014
    - Topic: Intro to MV* and JS implementations. We will pick one between Angular.js and Backbone and discuss that. Learn a thing: ruby gems - how to use them to automate processes.
- Week 12 - 04/17/2014
    - Topic: Continuing with our MV* framework. Pushing it forward. Some leeway as we approach the end of semester.
- Week 13 - 04/24/2014
    - Topic: Bringing it all together. Front-end and backend. MEAN stack. Our guests of honor, Ashley Williams and Michael Keller will share cool stuff with us! Ashley is a web developer and teacher at Flatiron School and together with Michael who's a data journalist at AlJazeera, they run CSV Sound System and have made things like treasury.io
- Week 14 - 05/01/2014
    - Topic: Workshop
- Week 15 - 05/08/2014
    - Topic: Workshop

- Week 16 - 05/15/2014
    - Topic: Final presentations

Learning Outcomes:

- Get comfortable with JavaScript as a powerful programming language
- Be able to develop front-end applications using JavaScript
- Ba able to write modular, object oriented JavaScript
- Be able to create an architecture for an application based on the requirements
- Having learned the concepts of the language, be able to approach new libraries, frameworks and techniques with confidence and use them to their advantage
- Be able to work with Node.js to write JavaScript on the server
- Be able to translate creative ideas from first steps of wireframes and sketches to design, and code.

Assessable Tasks:

- Mini-assignments:
    - Calculator app [due 02/06]:
        - Write a simple calculator application that runs in the console. It should be namespaced, and you should only declare one global variable: An object that contains the properties and functions you need to run your app. The calculator should perform simple math operations between two numbers (passed to the functions) and store all the results in an array, sitting within the same namespace. No html, css or any interface needed. Extra points for a simple interface.
    - Note taking app pt.1 [due 20/06]:
        - We now have an application that allows you to take

notes and store them in an array. Your assignments is to write the implementation of the “like” and “remove” methods of the Note constructor. You are also asked to correctly invoke these methods as callback functions for click event listeners.

- Note taking app pt.2 [due 27/06]:
  - We now have have a note taking app. We can write things that are stored in an array and has “like” and “remove” methods. Your assignment is to add other custom methods and properties for note object. Thin “title”, “body”, “date”, “location” and the like. Look at twitter and notice how each tweet has properties and functionalities attached to it. We are aiming at creating meaningful, useful and reusable objects.
- Student-identified exploratory project
  - This is the assignment that is due after spring break. So you will have some time to take a breather from the note taking app business for a little bit, and think about what you are excited to get javascript to for you. Think why you came to this class. Did you want to make a music player? An installation using javascript to talk to a server? An audiovisual canvas experiment? You are invited to venture out and pick a project you like to work on for two weeks and play around with it. Make something. But keep in mind and follow the patterns and best practices we have learned so far. However you wrote JavaScript before coming to the class stays behind the door. We now know core concepts and some reusable patterns. We can now write more elegantly and more readable and we can write object oriented programs and create simple architectures. Make a

project and submit it.

- Note taking app pt.3
  - Your note taking application is on again. Your assignment is to write a router module for it and create two views for the app: The new note (Compose Note) view and archive view (chronological list of all notes with group selection and actions, such as delete all). the app should now save each note to localstorage and read from it. Use localstorage as database. You are free to use jQuery and underscore libraries.
- Note taking app pt.4
  - You assignment is to use a templating engine to store process your "views". Use any library you need with your project. But keep in mind, use libraries when you really need to. You will now have a note taking app that writes to and reads from the local storage, have one "model" (Note), have the "controller" logic in place and "views" a la templating engine. And a router on top of that to tie parts together. We have successfully created a modular architecture.

- Midterm project
  - This should be an attempt at doing something a bit bigger. You can now write javascript applications. Think of one and build it. You can work alone or in teams of maximum 3 peers. As far as UI / UX / Visual design goes, you are expected to show that you are designers. Cognisant of user's intent, visually coherent and tasteful. As far as development goes, all team members should actively participate. Meaning that the way you devie up the work as a team, should be primarily based on who writes what module or what part of the application. It is up to you how you divide the design tasks. But the goal of the team

work in this project is to get you develop together. If need be, work remotely and collaborate using github as an effective collaborative tool. You can also get together and jam away at the same place and pair-program. There's value in doing both.

- Final project
    - We have come a very long way from the humble beginnings. We can now write front-end applications and create backend services with node.js. Take all you've learned and make something. Everyone should work in teams.

Final Grade Calculation

Participation/Attendance - 20%

Mini-Assignments(Total) - 30%

Mid-term Project - 25%

Final Project - 25%

Required Reading:

none

Recommended Reading:

- Eloquent JavaScript A Modern Introduction to Programming by Marijn Haverbeke
- JavaScript: The good parts by Douglas Crockford
- Learning JavaScript design patterns by Addy Osmani
- JavaScript design patterns: Stoyan Stefanov
- Node beginner by Manuel Kiessling
- Professional Node.js by Pedro Teixeira

Resources:

- Mozilla Developer Network
- Library documentations

Material and supplies:

- Code editor (sublime text, webstorm, coda, textMate, text wrangler, notepad++, etc.)
- Chrome of Firefox with debug tools
- Git

Grading Standards:

For note taking app mini-assignments, you are graded based on how you fulfill the requirement. The interface design should also be tasteful.

For free-form assignment, you are graded based on how you take the concepts you learned and approach building something different in quick turn around. It does not need to be a complete working product. It can be a prototype round for something bigger.

For mid-term and final projects, Architecture, best practices, clean readable code, good team-work workflow, sensible design and presentation.

Undergraduate

A [4.0; 96–100%]
Work of exceptional quality, which often goes beyond the stated goals of the course

A- [3.7; 91 –95%]
Work of very high quality

B+ [3.3; 86–90%]
Work of high quality that indicates substantially higher than average abilities

B [3.0; 81–85%]

Very good work that satisfies the goals of the course

B- [2.7; 76–80%]
Good work

C+ [2.3; 71–75%]
Above-average work

C [2.0; 66–70%]
Average work that indicates an understanding of the course material; passable
Satisfactory completion of a course is considered to be a grade of C or higher.

C- [1.7; 61–65%]
Passing work but below good academic standing

D [1.0; 46–60%]
Below-average work that indicates a student does not fully understand the assignments;
Probation level though passing for credit

F [0.0; 0–45%]
Failure, no credit

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: [You should include one the following standards, depending on the level of your course].

Undergraduate students: Work must be completed no later than the seventh week of the following fall semester for spring or summer term incompletes and no later than the seventh week of the following spring semester for fall term incompletes. Grades of "I" not revised in the prescribed time will be recorded as a final grade of "WF" by the Office of the Registrar.

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 [You should include the following headings with the recommended text. In addition, you should include any other policies you may have.]

- 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.
http://www.newschool.edu/studentservices/disability/.