

Program	School of Art, Media, and Technology: Communication Design
CRN	2158
Semester	Fall 2020 (Online)
Meeting Day	Fridays
Meeting Time	12:10-14:50, EST
Zoom	zoom.us/meeting/92663455501
Instructor	Lukas Eigler-Harding eigll343@newschool.edu
Class Site	js2020.teachinginter.net
Canvas Url	canvas.newschool.edu/courses/1524310

Course Description

This course will focus on extending students' knowledge of programming and computation using the JavaScript language for web and mobile contexts. Students will learn intermediate and advanced methods in JavaScript, the programming language of the web. Students will be introduced to a variety of dynamic use cases, from front-end frameworks and libraries to the creation of desktop and mobile applications.

Course Outline

■ Part 1 (*Weeks 1 & 2*)

Students are re-introduced to basic HTML + CSS concepts and are asked to produce a simple “homepage” site to catalogue the projects for the rest of the semester. If students feel comfortable, they are given the opportunity to integrate javascript into their site. An emphasis is given on a foundational web knowledge (responsive units, breakpoints, clear code) to provide a framework for more advanced work in the coming weeks.

■ Part 2 (*Weeks 2–5*)

Students are introduced to programming concepts: data storage/structures, loops, and functions and explore their visual potential through the development of generative patterns, animations, and modular abstractions. Emphasis is given towards understanding data types, storing values, and manipulating css (units and classes) with javascript.

■ Part 3 (*Weeks 5–8*)

Students are introduced to eventListeners and interactions through a series of quick interface sketches.

■ Part 4 (*Weeks 8–10*)

Students are introduced to javascript as a method of data management. Students are given demos exploring asynchronous functions, APIs, and localStorage to gain an understanding of fetching and pushing data.

■ Part 5 (*Weeks 10–15*)

After gaining an understanding of how to manipulate css through plain javascript, students are introduced to javascript libraries through a series of small demos. Students are expected to select and become well-acquainted with the documentation of one library through an interface-based project for the remainder of the course.

Learning Outcomes

By the end of the semester, students will be able to:

- Understand the affordances and limitations of using software as a medium
- Read and understand the source code of both their own and their classmate's projects
- Independently research and learn unfamiliar technical topics
- Understand contemporary programming workflows and processes within the JavaScript ecosystem
- Develop the ability to incorporate external data using Application Programming Interfaces (APIs)

Assessment Criteria

10%	Attendance & Professionalism
20%	Ongoing Research: Interfaces + Interface Sketches
10%	Project 1: Index
20%	Project 2: Pattern
15%	Project 3: APIs
25%	Project 4: Library
100%	Total

Attendance, Grading and Work Submission Standards, Program Policies, Making Resources, and University Policies

All CD classes adhere to the same program and university policies.

[The exact policy can be found here.](#)

Assignments

■ Ongoing Research

Throughout the semester students will split their time between two ongoing research tasks.

1 For the first half of the semester, students will contribute to the class are.na channel consisting of *Interface Research Blocks** and post *at least two* blocks to their channel weekly. In addition, each week students will post direct links to their new blocks to the “Interface Research” Canvas Discussion board to streamline weekly reviews.

**Interface Research Blocks: Each block consists of a screenshot of a moment in a digital interface (web, app, fictional). The Block’s description should include a link to the original interface along with a small paragraph of why you found that interface compelling.*

2 For the second half of the semester, students will reflect on the collectively amassed Research Blocks by composing weekly interfaces sketches—small single page javascript experiments exploring interaction, composition, and layout.

■ Project 1: Index (2 Weeks)

Take the Glitch page that you set up in today’s class and convert it into a growing index of the work you will do for this class. As you design your index, consider what information is attached to each item (at minimum, this might include the project title, date, brief, description, or screenshots), and what kind of information you want to include on the periphery (how is the site introduced? How do we know to whom the site belongs to?).

Think about how this index will grow and change in reaction to the projects you will add. Your index should be built responsively (work on desktop and mobile), and emphasize modularity. The goal of this 2-week assignment is to build a dynamic system using only HTML/CSS. The second week of this assignment will be spent introducing Javascript as a method of abstracting and saving data for your site.

■ Project 2: Pattern (5 Weeks)

This project is considered an exploration into the iterative nature of Javascript and consists of two parts.

Part A: you will write a Javascript program which uses loops and if statements to create a modular generative pattern. You will design at least one single pattern unit (a module) (think: div) that changes over the course of its output. Your unit should iterate at least 100 times. With each iteration, your unit should somehow modify itself (content or css attributes: size, color, rotation, shape, texture). Your pattern should have at least 2 variables of change (ex: content, color), is allowed to repeat itself (i.e. be a pattern), and is allowed to be as abstract or literal as you want (it can tell a story, or it can be an exercise in form).

Part B: You will create a second Javascript pattern and a “control panel” to match. Your pattern will initially utilize setTimeout/setInterval methods and CSS classes to animate. You will then add a variety of eventListeners to introduce interactivity.

The goal of this project is to use javascript to dynamically append HTML elements and programmatically alter their attributes. This project will also utilize javascript to explore techniques of animation and interactivity.

■ Project 3: APIs (3 Weeks)

Design and develop a website which incorporates an external data feed, using one of the APIs discussed in class. You may also use another API or service that you have researched independently, so long as it provides a live data stream to work with. Consider how your specific design logic might change the meaning of said data, and how the user may (or may not) interact with the feed as they engage with your site. This site will be designed with the goal to re-contextualize your data source(s) and generate unique interface moments.

On a technical level, this assignment will allow you to further develop an understanding for javascript-written interfaces and the components they are made of.

■ Project 4: Library (5 Weeks)

This assignment is meant to expand a student’s understanding of customized user experience and interactivity in a mobile context. Each student will select an object from which a user-site relationship will be translated. Using your knowledge of Javascript so far, design and develop a linear relationship between a user’s actions and the visual rendering of a site. Using the localStorage API, consider what information you store about the user and their behaviour (i.e. time-visited, mouse movement, time spent on site), and to what degree/in what form this information is relayed back to the user upon each visit.

Similar to Latour’s key and Suri’s thoughtless acts (see readings below), keep in mind that the site should become increasingly personal, shaping itself to the individual user.

Materials and Supplies

Ware	Description
Hardware	Laptop
Account	Glitch Account
Account	Are.na Account
Software	Sublime Text 3 or VS Code
Software	Chrome Browser
Software	Design Tools: Sketch/InDesign XD, Figma

Schedule

Our weekly schedule *begins on Fridays*.

Asynchronous tasks, Assignments, and Research/Readings are due the following week.

- Assignments are must be posted to Canvas by Fridays, 12:10 EST (class time)
 - Research/Reading responses must be posted to Canvas by Wednesdays, 23:59 EST
 - Research requirements are described above
 - For each Reading assigned, students must submit either 2 non-binary questions *or* a paragraph critique to the Canvas Discussion board “Reading Responses”.
- These responses will be used for in-class discussions

• Asynchronous Tasks are never *due*, but are expected to be completed in order to fully complete assignments

Week	Sync (Fridays, 12:10 EST)	Async	Assignments	Research/Reading
Week 1 Sync: 09.04	<i>All</i> • Introductions • Syllabus Review • Schedule • Software Introduction	• Lecture: HTML + CSS • Set-up Glitch Account	Index, Part 1	• IR Blocks 2x • Readings
Week 2 Sync: 09.11	<i>Groups</i> • Review IR Blocks • Reading Discussion • Critique Index Project	• Lecture: Javascript Variables + DOM manipulation • Lecture: Organizing your Code	Index, Part 2	• IR Blocks 2x
Week 3 Sync: 09.18	<i>All</i> • Review IR Blocks • Introduce Pattern Project • Critique Index Project	• Lecture: JS For loops • Lecture: JS + CSS	Pattern, Part 1	• IR Blocks 2x • Readings
Week 4 Sync: 09.25	<i>Groups</i> • Review IR Blocks • Reading Discussion • Critique + debug Patterns	• Lecture: Javascript setTimeout/setIntervals • Lecture: JS + CSS Classes	Pattern, Part 1, Revisions	• IR Blocks 2x
Week 5 Sync: 10.02	<i>Groups</i> • Review IR Blocks • Introduce Pattern, Part 2	• Lecture: eventListeners	Pattern, Part 1 Pattern, Part 2,	• IR Blocks 2x
Week 6 Sync: 10.09	<i>All</i> • Review IR Blocks • Critique Patterns, Part 1	–	Pattern, Part 2	• IR Blocks 2x
Week 7 Sync: 10.16	<i>Individual</i> • Semester Check-ins	• Lecture: APIs, Part 1	Pattern, Part 2	• IR Blocks 2x
Week 8 Sync: 10.23	<i>Group (½ class)</i> • Review IR Blocks (2 weeks) • Critique Pattern, Part 2 • API Project introduction	• Lecture: APIs, Part 2	API, Working demo + design sketches	• Interface Sketch 1x
Week 9 Sync: 10.30	<i>Group</i> • IS Showcase	• Lecture: JS Libraries	API	• Interface Sketch 1x

	<ul style="list-style-type: none"> • API Review 			
Week 10 Sync: 11.06	<i>Individual</i> <ul style="list-style-type: none"> • API Check-in 	<ul style="list-style-type: none"> • Lecture: JS Libraries 	API, finalize	<ul style="list-style-type: none"> • Interface Sketch 1x
Week 11 Sync: 11.13	<i>All</i> <ul style="list-style-type: none"> • IS Showcase 2x • Critique APIs • Introduce Final Project 	<ul style="list-style-type: none"> • Design Resource Lecture 	Final Project: <ul style="list-style-type: none"> • designs • JS Library sketch 	<ul style="list-style-type: none"> • Interface Sketch 1x • Readings
Week 12 Sync: 11.20	<i>Group</i> <ul style="list-style-type: none"> • IS Showcase • Reading Discussion • Final Project Review 	<ul style="list-style-type: none"> • Design Resource Lecture 	Final Project, design + development	–
Week 13 Sync: 12.04	<i>Individual</i> <ul style="list-style-type: none"> • Check-in 	<ul style="list-style-type: none"> • Design Resource Lecture (based on project) 	Final Project, development	–
Week 14 Sync: 12.11	<i>Individual</i> <ul style="list-style-type: none"> • Check-in 	Submit Semester Documentation	Final Project, finalize	–
Week 15 Sync: 12.18	<i>All</i> <ul style="list-style-type: none"> • Final Critique and Presentation 		–	–