The Artist as Programmer
AHD-2241-B | VCD-2241-B

Location: Online
Synchronous: Thursday, 1:00pm-2:30pm

Justin Elm
jelm@sva.edu

OBJECTIVES & DESCRIPTION

In the post-studio interdisciplinary art world, technology plays a critical role in an artist's practice. The ubiquity of the internet and computers demands a new kind of literacy. By examining contemporary artists working on the periphery of traditional media, we'll explore the implications for art and artists. Readings and lectures will be supplemented by in-class exercises that introduce fundamental programming principles with HTML, CSS, and JavaScript. To emulate the interdisciplinary art world mentioned, this course is a hybrid art history course with studio practice.

DEFINITIONS

Synchronous: scheduled online meeting via Zoom
Asynchronous: course content that must be completed on your own time, outside of class
IDE: Integrated development environment
Text Editor: A plain text editor where we'll write most of our code
Libraries: Open source code written by others that we'll make use of

TECHNOLOGY REQUIREMENTS

Under normal circumstances this course takes place in a lab and the majority of technology requirements would be fulfilled by the classroom. Unfortunately, due to the semester taking place online, most of these technology requirements will now fall on you as student. I wish this was not the case, but here we are.

To participate in class you will need some form of working computer. I emphasize computer, as an tablet (Surface Pro, iPad, or Chromebook) will not be sufficient and you will not be able to participate in class. The operating system doesn't matter, but I will be using macOS so if you plan on using a Windows machine you'll have to do some translating. We'll be making use of all open source languages and software, so outside of a computer, there are no associated costs. A few of the technologies we'll be using below.

Operating System: macOS
Browser: Primarily Chrome, sometimes Firefox

Software: Atom (text-editor), possibly Terminal (bash or zsh)
Languages: HTML, CSS, JavaScript
Libraries: jQuery, Node.js
Accounts: Repl.it, a web-based IDE useful for JavaScript demos with instant feedback

COUSRE REQUIREMENTS

Students are expected to attend synchronous course sessions via Zoom and to participate in class demonstrations and discussions. Asynchronous content, including slides, videos, external content, and programming tasks, must be completed before the week's synchronous session and turned in via Canvas. Each week's demonstration is based off the prior week's asynchronous content, so there will be an expectation of basic familiarity with definitions and concepts before meeting via Zoom. The course will be project based, proposed by the student and built over the course of the semester, to be presented and turned in at the end of the semester.

ATTENDANCE

Attendance will be counted via your presence in the Zoom session of the synchronous class sessions. There are no excused absences, and more than two absences in a semester will negatively affect your final grade. If there are any conflicts in terms of disparate time zones, internet availability, VPN usage, or technology limitations, please bring these to my attention as soon as possible.

If you have extenuating health-related circumstances that may necessitate excessive absences, please contact the *Office of Disability Resources* or *Student Health and Counseling*, respectively. Both offices provide support and may be able to grant accommodations for excessive absences. However, only in extreme circumstances would an accommodation be granted for absences that exceed the institutions policy.

ACADEMIC INTEGRITY

Academic dishonesty, including plagiarism, will not be tolerated. Students found to have committed an act of academic dishonesty will fail the assignment for which an infraction is suspected and substantiated. More serious violations will be handled through the process enumerated in the SVA Handbook (p. 8).

STUDENTS WITH DISABILITIES

In order to receive academic accommodations due to disability, a student must first register with the Office of Disability Services (ODS). Students approved for accommodations will be given an *ODS Accommodation Letter* to submit to their instructors. If a student does not provide an *ODS Accommodation Letter* to their instructor, they will not be eligible to receive accommodations in that course. All instructors are required to adhere to SVA's policies regarding accommodations for student’s disabilities. Students who have a need for academic accommodations, or suspect they may have a disability should contact the ODS.

Disability Resources
340 East 24th Street, New York, NY 10010
212.292.2282
disabilityservices@sva.edu
sva.edu/students/disability-resources

Student Health & Counseling Services
340 Easy 24th Street, New York, NY 10010
212.592.2246
health@sva.edu
sva.edu/students/health-counseling

REFERENCE LINKS
https://html.com/
https://github.com/
https://jquery.com/
https://codewars.com/
https://stackoverflow.com/
https://www.javascript.com/
https://www.hackerrank.com/
https://www.w3schools.com/css/
https://developer.mozilla.org/en-US/

INTERESTING LINKS
http://formandcode.com/
http://hello.processing.org/
http://devart.withgoogle.com/
http://chromeexperiments.com/
http://threejsplaygnd.brangerbriz.net/

OUTLINE

**Each topic subject to change*

WEEK 1 – January 14
Introduction to the semester

SWEEK 2 – January 21
HTML & CSS, Positioning, Assets, Structure

Reading:
Ullman, Ellen, *Life In Code*, 'The Dumbing Down of Programming'
(p. 39-55)

WEEK 3 – January 28
HTML & CSS continued, Responsiveness, Media Queries

Spotlight: Casey Reas
Reading:
Reas, Casey, & McWilliams, Chandler, *Form + Code*, 'Introduction & What is Code'
(p. 8-26)

WEEK 4 – February 4
Introduction to JavaScript, Data Types, Variables
Spotlight: Daniel Shiffman

Reading:
Reas, Casey, & McWilliams, Chandler, *Form + Code*, 'Repeat'
(p. 42-65)

WEEK 5 – February 11
Conditionals, Built-in Methods

Spotlight: Bret Victor
Reading:
Victor, Bret, "Learnable Programming"

WEEK 6 – February 18
Functions, Scope

Reading:
Levin, Golan, "Dialtones Report"

WEEK 7 – February 25
Arrays, Loops, Iterators

Spotlight: Rafaël Rozendaal

Reading:
Rozendaal, Rafaël, "Notes on Abstract Browsing"

Spring Break; No Class – March 1-5

WEEK 8 – March 11
The Document Object Model (DOM), External Libraries (jQuery); Part 1

Spotlight: Jon Rafman

Reading:
Zhexi Zhang, Gary, "Infinite Lives: The online anthropology of Jon Rafman"

WEEK 9 – March 18
The Document Object Model (DOM), External Libraries (jQuery); Part 2

Spotlight: Raven Kwok

WEEK 10 – March 25
Objects and Classes

WEEK 11 – April 1
Useful & Practical Demos

WEEK 12 – April 8
Game Stuff

WEEK 13 – April 15
Particle Nonsense (maybe)

WEEK 14 – April 22
Review, Requests, Missing Pieces

WEEK 15 – April 29
Final Projects