Seesaw Communities – Web Development Curriculum

Instructor: Jiwon Woo
Audience: Beginners (no prior coding experience required)
Schedule: 1 hour every weekend (Saturday or Sunday evening)
Delivery Mode: Online via Zoom
Tools:

- Google Drive (materials)
- Slideshows: (https://hack.uclaacm.com/archive : This was a club I was a part of that helps students learn web development),
- Slack (communication)

Cost: Free

Course Purpose:
This course is designed to help individuals—especially those with cognitive impairments—gain job-relevant digital skills. Coding and web development are high-demand, low-social-interaction skills that can promote independent living. This program also serves as professional training for the instructor and a way to give back to the community.

Course Objectives

- Understand how the internet and computers work
- Learn the structure and design of static and dynamic websites
- Build webpages using HTML, CSS, and JavaScript
- Gain job-ready digital skills

- Create and showcase personal or professional websites

Course Structure

- **Duration:** 4 months
- **Format:** 1 class per week (1 hour each)
- **Delivery:** Online via Zoom
- **Support Tools:** Google Drive for lessons, Slack for communication

---

Month 1: Introduction and HTML Basics

Week 1: What is the Internet?

- Internet vs. Web
- How websites work (client-server model)
- Web browsers and how they display code
- Developer tools introduction

Week 2: HTML Document Structure

- Basic structure: doctype, html, head, body

- Writing your first web page
- Tags, elements, attributes

Week 3: Common HTML Tags

- Headings, paragraphs, links
- Lists: ordered and unordered

Week 4: Tables and Forms

- HTML tables: table, tr, th, td
- HTML forms: form, input, select, textarea, button
- How form data is structured

Month 2: HTML Advanced and CSS Basics

Week 5: Media and Attributes in HTML

- Adding images, audio, and video
- Using src, alt, title attributes

Week 6: Introduction to the DOM

- What is the DOM (Document Object Model)?

- Exploring the DOM with browser dev tools
- Basic DOM structure understanding

Week 7: CSS Basics

- Inline, internal, and external CSS
- Selectors: element, class, ID
- Fonts, text color, background

Week 8: CSS Box Model

- Margin, padding, border
- Width and height
- Display types: block, inline, inline-block

At this stage, my goal is for students to feel confident building simple web projects using HTML and CSS. Depending on how comfortably they grasp the material, we'll begin exploring JavaScript next. Learning JavaScript will open the door to creating more interactive and dynamic websites, giving students even more creative and functional tools to work with.

Month 3: CSS Layout and Static Website Project

Week 9: Layout and Positioning

- CSS position: static, relative, absolute, fixed

- Top, left, z-index
- Intro to Flexbox

Week 10: Advanced Styling

- Shadows and effects
- Overflow and white-space control
- Pseudo-classes like :hover and :active

Week 11: Responsive Design and Bootstrap

- Viewport meta tag
- Media queries
- Introduction to Bootstrap framework

Week 12: Static Website Project

- Students create a simple portfolio or menu page
- Static pages that do not need server interaction
- Peer review and feedback

Month 4: Introduction to JavaScript

Week 13: JavaScript Basics

- Including JavaScript in HTML
- console.log
- Variables: var, let, const
- Data types

Week 14: Control Flow and Functions

- if/else statements, switch
- Loops: for and while
- Functions: declaration, calling, return values
- Scope and basic understanding of closures

Week 15: Events and Interactivity

- Using addEventListener
- Click and input events
- Changing the DOM with JavaScript
- Event bubbling basics

Week 16: Final JavaScript Project

- Adding interactivity to existing static site

- Example features: form validator, dark/light mode, quiz
- Student presentations

Optional Advanced Topics (Bonus or Future)

JavaScript Part 2: Data Structures and Algorithms

- Objects, arrays, stacks, queues, maps
- Linked lists
- Sorting basics
- Iterators and generators

JavaScript Part 3: Object-Oriented Programming (OOP)

- Objects and classes
- Constructors and the this keyword
- Prototypes and inheritance
- Encapsulation, abstraction, polymorphism
- Static methods, getters and setters
- Practice quizzes on objects and classes

Tools and Platforms

- Code Editors: Replit (online), Visual Studio Code (optional)
- Practice Platforms: CodePen, JSFiddle
- Documentation: MDN Web Docs, W3Schools
- Lesson Hosting: Google Drive
- Group Communication: Slack