

Coding Academy Curriculum Overview

	Monday	Tuesday	Wednesday	Thursday	Friday
Week 1	HTML	Git, CSS	CSS	CSS	JS
Week 2	JS	JS	JS, Git	Mini Proj.	JS
Week 3	Projects	Projects	Projects	Projects	Projects

Week 1: We initially introduce students to the mechanics of the web with a description of what happens after they enter a URL into a browser. Where does that information go? How does the browser know what to display after a user makes a request? What information is transmitted to the browser? We reveal what's under the hood and give the students the ability to analyze the HTML, CSS, and JavaScript contained in any website.

Next, students learn how to build their own web page using HTML. They add text, links, lists, and graphic elements on a page. Then, they incorporate CSS rules to change the styling of their page elements so that users can understand and use the information on their page more easily. Students learn modern CSS display tools such as flexbox and grid to make the site usable on a wide range of screen sizes — from large-screen monitors to smartphones (responsive design).

Specific Topics:

The Web

- The browser
- Web servers
- Hypertext Transfer Protocol (HTTP)
- Requests
- Introduction to HTML, CSS and JS
- The browser inspector
- Element Inspector (HTML and CSS)
- Source Inspector (JavaScript)
- VS Code Introduction

HTML – Hypertext Markup Language

- HTML Elements
- HTML document structure
- Document header
- Document body
- Body Elements
- Paragraphs
- Divs
- Headings

- Unordered and ordered lists
- Links – (Introduction to Attributes)
- Horizontal rules
- Sub and superscripts
- Inputs
- Header Elements
- Title
- Link
- Incorporating Graphic Elements – local and remote
- Scalable Vector Graphics (SVG)

Git

- Command Line
- Git Installation
- Github
- Working with Git in Teams
- Merge Conflicts

CSS – Cascading Style Sheets

- Intro to Cascading Style Sheets
- Element style attributes
- CSS syntax
- Property/value style attribute pairs
- Style tags
- External style sheets
- CSS properties
- Text color
- Color Systems
- Named colors
- Hexadecimal
- RGB
- Background Color
- Height
- Width
- Fonts – family, size, style, and weight
- The Box Model
- Borders
- Padding
- Margin
- Border radius
- Backgrounds
- Divs and Spans
- Classes
- Ids
- Responsive Design
- Float Property
- CSS Display Properties

- Block
- Inline
- Inline-block
- Flex
- Grid
- Media Queries
- CSS Positioning Properties
 - Static
 - Relative
 - Absolute
 - Fixed
- CSS Animation
- Using a CSS Framework (Bootstrap)

Week 2: In the second week, we work with JavaScript, a scripting language specifically designed for the web. Students learn variable types and the syntax of flow-control constructs such as conditional statements, program looping, and function calls. We explain how to put these code structures together to build clean, scalable code. Students learn to use JavaScript's Document Object Model to manipulate a web page's HTML and CSS to make dynamic, easy-to-use sites.

Specific Topics:

- Intro to JavaScript
- Script Tags
- External Scripts
- Document Object Model Elements and DOM methods
- Events and Event Handlers
- Mouse Events
- Key Events
- Input Events
- this (self-referencing)
- Variable Types
 - Numbers and arithmetic operators
 - Booleans and Boolean operators
 - Strings and string methods
 - Arrays and array methods
- Defining and calling functions
- Passing Parameters into functions
- Returning values from functions
- Calling functions using setInterval and setTimeout
- SVG Animation
- Flow Control
 - If statements
 - If else statements
 - If else if statements

- For loops
- For of loops
- While loops
- Objects
 - properties
 - methods
- Classes, Constructors
- Canvas
- Canvas Animation

Week 3: In the final week of class, students flex their new coding muscles by assembling teams and creating web applications of their choosing. They gain practice with Git commands and GitHub repositories to facilitate collaborative code development under version control. They experience the give-and-take required for building an application with other programmers and how to break down and assign tasks efficiently. On the last day of class, each team presents its final working project.