



# CodeHS

## Georgia Web Development Syllabus

1 year for High School (135-155 contact hours)

### Course Overview and Goals

The Georgia Web Development Course is intended to teach students the fundamentals of web design and development in a project-based learning environment. Students will learn the basic elements of web development, such as web hosting, creating HTML/CSS files, and incorporating Javascript into HTML files. Over the course of the school year, students will collaboratively and independently design, develop and implement functional and responsive web pages using these foundational skills.

**Learning Environment:** The course utilizes a blended classroom approach. The content is fully web-based, with students writing and running code in the browser. Teachers utilize tools and resources provided by CodeHS to leverage time in the classroom and give focused 1-on-1 attention to students. Each unit of the course is broken down into lessons. Lessons consist of video tutorials, short quizzes, example programs to explore, and written programming exercises, adding up to over 100 hours of hands-on programming practice in total. Several units end with a comprehensive unit test that assesses student's mastery of the material from that unit as well as challenge problems where students can display their understanding of the material.

**Development Environment:** Students write and run HTML, CSS, and JavaScript files in the browser using the CodeHS editor. Due to the fact that different browsers treat HTML and CSS differently, we highly recommend that all student computers use an up-to-date version of the Chrome browser. You can download an up-to-date version of Chrome for free here: <https://www.google.com/chrome/browser/>

**More information:** Browse the content of this course at <https://codehs.com/course/16771>

**Prerequisites:** There are no official prerequisites for the CodeHS Georgia Web Development course. The course is designed for students with some previous background in computer science. It is designed ideally for students who have an introductory knowledge of Javascript. Students who have completed the CodeHS *Intro to Programming in Javascript* or *AP Computer Science Principles* courses are eligible to complete this course.

### Course Breakdown

#### Unit 1: HTML - Structuring Websites (3 weeks / 15 hours)

Students learn about the language behind all websites: HTML. Students learn about several different HTML tags as well as the basic structure of a web page. Students use HTML to develop several of their own creative web pages.

Objectives / Topics Covered	<ul style="list-style-type: none"><li>• How do we build web pages?</li><li>• Markup Languages</li><li>• HTML</li><li>• HTML tags</li><li>• HTML attributes</li></ul>
-----------------------------	--

	<ul style="list-style-type: none"> <li>● HTML elements</li> <li>● The Anatomy of an HTML page</li> <li>● Formatting text</li> <li>● Hyperlinks</li> <li>● Images</li> <li>● Copyright fair use</li> <li>● Lists</li> <li>● Nesting tags</li> <li>● Tables</li> <li>● Styling with HTML</li> <li>● HTML Colors</li> </ul>
Assignments / Labs	<ul style="list-style-type: none"> <li>● Students create several web pages to practice each of the concepts above</li> <li>● Example exercises: <ul style="list-style-type: none"> <li>○ Modify existing web pages using formatting tags to make text more readable</li> <li>○ Use links to create a web page linking to your 5 favorite websites</li> <li>○ Use links and images to create a personal library web page showing your favorite books</li> <li>○ Use lists and images to create a flashy list article</li> <li>○ Use tables to create a personal calendar web page</li> <li>○ Use styling attributes to add style to your web pages</li> </ul> </li> </ul>

### Unit 2: CSS - Styling Websites (2 weeks / 10 hours)

Students learn the language CSS and use it to style their web pages. Students learn about the benefits of styling with CSS and will use CSS to create several styled web pages of their own.

Objectives / Topics Covered	<ul style="list-style-type: none"> <li>● How do we style web pages?</li> <li>● CSS vs HTML</li> <li>● CSS Selectors</li> <li>● Selecting by tag</li> <li>● Selecting by class</li> <li>● Selecting by id</li> <li>● The Cascade (order of selector precedence)</li> </ul>
Assignments / Labs	<ul style="list-style-type: none"> <li>● Students create several web pages to practice each of the concepts above</li> <li>● Example exercises: <ul style="list-style-type: none"> <li>○ Use CSS selectors to style your previous web pages</li> <li>○ Use CSS selectors to style new web pages</li> <li>○ Create a music library web page and use CSS to style each song in your table</li> <li>○ Use CSS styling to make several images fit together properly</li> <li>○ Explain the benefits CSS provides over styling with only HTML</li> <li>○ Identify CSS selectors and rules used on example web pages</li> </ul> </li> </ul>

### Unit 3: Advanced HTML and CSS (4-5 weeks / 20-25 hours)

This module dives deeper into different things we can do with HTML and CSS. Students practice advanced topics in HTML and CSS, including visibility, image filtering, interaction, and animation, to develop more advanced websites.

Objectives / Topics	<ul style="list-style-type: none"> <li>● Splitting your site into separate files</li> </ul>
---------------------	---

Covered	<ul style="list-style-type: none"> <li>● Iframes and embedding</li> <li>● &lt;div&gt;</li> <li>● &lt;span&gt;</li> <li>● Combining CSS selectors</li> <li>● Special CSS selectors</li> <li>● Reading documentation</li> <li>● Avoiding repeated code</li> <li>● Visibility</li> <li>● Positioning</li> <li>● Image manipulation</li> <li>● Animation</li> <li>● Interaction</li> <li>● CSS Frameworks</li> </ul>
Assignments / Labs	<ul style="list-style-type: none"> <li>● Example exercises: <ul style="list-style-type: none"> <li>○ Use iframes to embed a video into your web page</li> <li>○ Use iframes to embed a previous exercise into your web page</li> <li>○ Use divs and spans to style groups of HTML elements</li> <li>○ Add CSS styling to elements that the user clicks on</li> <li>○ Add CSS styling to elements that the user hovers over</li> <li>○ Use CSS to add a black and white filter to a group of images</li> <li>○ Use CSS to add a blur filter to a group of images</li> <li>○ Use CSS animations to make your site slowly fade in over time</li> </ul> </li> </ul>

#### Unit 4: Project - Create Your Homepage (2 weeks / 10 hours)

Students build their own websites about themselves. This site will be accessible on their own custom domain and will be continually improved by the student as they continue on in the course. It will serve as a running portfolio of each creative project they create in the course.

Objectives / Topics Covered	<ul style="list-style-type: none"> <li>● Software Development Life Cycle</li> <li>● Combination of the concepts learned thus far</li> <li>● Allow students to think creatively about applications of the concepts they have learned</li> <li>● Designing a web page from scratch</li> </ul>
Assignments / Labs	<ul style="list-style-type: none"> <li>● Students will build their own website about themselves. This site will be accessible on their own custom url on the CodeHS site, and will be continually improved by the student as they continue on in the course. It will serve as a running portfolio of each creative project they create in the course.</li> </ul>

#### Unit 5: Introduction to JavaScript in HTML (3 weeks / 15 hours)

Students learn about the script tag, and how it can be used to write JavaScript code in their HTML files. Students are also introduced to useful JavaScript methods that can be used to alter the state of the CSS and HTML of a webpage, as well as how the Document Object Model supports the ability to make such changes.

Objectives / Topics Covered	<ul style="list-style-type: none"> <li>● The Script Tag</li> <li>● Using the DOM</li> <li>● Creating Elements Using the DOM</li> <li>● Styling Elements Using JavaScript</li> </ul>
-----------------------------	---

	<ul style="list-style-type: none"> <li>● Functions in HTML</li> <li>● This Keyword</li> <li>● Keyboard Interactions</li> <li>● JavaScript Animations</li> <li>● Positioning and Animations</li> </ul>
Example Assignments / Labs	<ul style="list-style-type: none"> <li>● Make a Quilt <ul style="list-style-type: none"> <li>○ Students practice iteration and appending elements to a web page. They also develop a randomizer set to initiate whenever the mouse hovers over an element.</li> </ul> </li> <li>● Challenge: Make a Keyboard <ul style="list-style-type: none"> <li>○ Students create a clickable keyboard using only JavaScript. This is a three part project, where students develop the keyboard, make it clickable, and finally get text to show up on a webpage.</li> </ul> </li> <li>● Add Div Animation <ul style="list-style-type: none"> <li>○ Students practice developing animations by creating a program designed to create and move divs across a webpage. Students will need to be able to access the height and width of the page, as well as the total distance that the elements have to move before stopping.</li> </ul> </li> </ul>

### Unit 6: Using JavaScript Libraries (2 weeks / 10 hours)

Students are introduced to jQuery, a JavaScript library that makes webpage interaction easier. Students will learn the basic syntax of jQuery, how to incorporate it in their webpages, and useful methods that help animate and change the responsiveness of their websites.

Objectives / Topics Covered	<ul style="list-style-type: none"> <li>● Introduction to jQuery</li> <li>● Iterating with jQuery</li> <li>● Animations with jQuery</li> <li>● Callback Functions and Synchronization</li> <li>● Using Multiple Files in JavaScript</li> </ul>
Assignments / Labs	<ul style="list-style-type: none"> <li>● Smart jQuery Table <ul style="list-style-type: none"> <li>○ Students learn how to create a smart table that changes colors based on the content of the table. If the values in the table are too low, then the table flags them as important.</li> </ul> </li> <li>● The Wave <ul style="list-style-type: none"> <li>○ Students learn how to create animations that rely on asynchronous execution. This assignment has them practice timing animations using callback functions.</li> </ul> </li> <li>● FrankenDiv <ul style="list-style-type: none"> <li>○ This assignment has students practice manipulating elements using the jQuery animations function. Students will create actions to change the size, color, and proportions of a div.</li> </ul> </li> </ul>

### Unit 7: Project: Build an Interactive Resume (2-3 weeks / 10-15 hours)

Students will create a single page, interactive resume. Students will be expected to add several animations using jQuery to prove their ability to modify HTML and CSS. This unit also examines web design theories that can help students improve the aesthetics of their resumes, and to evaluate the quality of a website based on its layout.

Objectives / Topics Covered	<ul style="list-style-type: none"> <li>● Creating a Resume</li> <li>● Layout and Web Design Theory</li> <li>● Interactive Resume Development</li> </ul>
Assignments / Labs	<ul style="list-style-type: none"> <li>● Online Interactive Resume <ul style="list-style-type: none"> <li>○ Students will develop an online interactive resume, and evaluate the quality of their peers resumes using rubrics to initiate feedback. Students will also determine the quality of the layout, using web design layout theory to establish their reasoning.</li> </ul> </li> </ul>

### Unit 8: Storing and Collecting Data (3-4 weeks / 15-20 hours)

This unit will explore the role that data plays in creating websites. Students will learn about the various ways that data is taken from web pages, as well as ways to secure themselves from unwanted data collection. Students will also learn how to incorporate data collection into their own websites, and collect simple information from users.

Objectives / Topics Covered	<ul style="list-style-type: none"> <li>● Collecting and Storing Data</li> <li>● Storing Data with Local Storage</li> <li>● Storing Data Structures with Local Storage</li> <li>● JavaScript Objects</li> <li>● Using Input Fields</li> <li>● Server-side Languages</li> </ul>
Example Assignments / Labs	<ul style="list-style-type: none"> <li>● Saving Active Divs <ul style="list-style-type: none"> <li>○ Students will learn how to save which elements on their page are active at a given time, and reproduce the same active state the next time a user loads the page.</li> </ul> </li> <li>● Contact List <ul style="list-style-type: none"> <li>○ Students will create a phonebook that stores and collects information about users' names and numbers, as well as retrieve information about users in their phonebook.</li> </ul> </li> <li>● Birthday Tracker <ul style="list-style-type: none"> <li>○ Students will create a Birthday Tracker to log birthdays. If a user enters a date, it will generate a list of all users who have that birthday.</li> </ul> </li> </ul>

### Unit 9: How to Build and Maintain a Website (2 weeks / 10 hours)

While students can run functional and responsive websites on CodeHS, this unit explores how students can create and run websites without the help of CodeHS. This unit will explore how to store web files, secure a domain name, and maintain a website. The majority of the lessons will be explanatory - students will not be expected to host and maintain a webpage off CodeHS, but will be given the tools needed to do so if they desire.

Objectives / Topics Covered	<ul style="list-style-type: none"> <li>● Choosing a Domain Name</li> <li>● Personal Servers vs Web Servers</li> <li>● Web Optimization</li> <li>● Choosing a CMS</li> <li>● Web Accessibility</li> <li>● Web Security</li> </ul>
-----------------------------	--

Assignments / Labs	<ul style="list-style-type: none"> <li>● Comparing CMS Systems <ul style="list-style-type: none"> <li>○ Students compare popular CMS systems and make decisions about which CMS is best suited for their own personal needs.</li> </ul> </li> <li>● Yoast Analyzer <ul style="list-style-type: none"> <li>○ Students use a Yoast analyzer to determine if their website, or websites that they frequent are optimized for search engines. Students learn how to optimize their websites by organizing their file structure and using metadata for search engine crawlers.</li> </ul> </li> </ul>
--------------------	--

**Unit 10: Project - Iterate Your Web Development (2-3 weeks / 10-15 hours)**

Students will continue the creation of their website and improve upon their design using feedback. The website will have to follow specific criteria - certain number of pages, responsiveness, and use of APIs. Students will go through a feedback process, and learn about making their websites more accessible to a wide array of users.

Objectives / Topics Covered	<ul style="list-style-type: none"> <li>● Providing Feedback</li> <li>● Presenting Your Innovation</li> </ul>
Assignments / Labs	<ul style="list-style-type: none"> <li>● Version Control <ul style="list-style-type: none"> <li>○ Students create a running version document that tracks the changes they make to their website. Students learn how to catalog each version, and are asked to document how the website changes over time.</li> </ul> </li> <li>● Present your Innovation <ul style="list-style-type: none"> <li>○ Students are required to make a presentation highlighting the innovative website that they created, and how that site addresses a particular problem in their community. Students highlight how their website changed over the course of development, and as a result of feedback provided by user testing.</li> </ul> </li> </ul>

**Unit 11: Computer Science Careers (2 weeks / 10 hours)**

Students take some time to explore and discover different computer science careers and professional student organizations. They will also examine inclusive coding and how to avoid bias in computer programming.

Objectives / Topics Covered	<ul style="list-style-type: none"> <li>● Computer Science Careers</li> <li>● Inclusive Coding</li> <li>● Career &amp; Technical Student Organizations (CTSO)</li> </ul>
Assignments / Labs	<ul style="list-style-type: none"> <li>● Career Research <ul style="list-style-type: none"> <li>○ Students research a computer science career of their choice, such as cybersecurity, web development, game development, etc.</li> </ul> </li> <li>● Research Georgia's Career &amp; Technical Student Organizations (CTSO) <ul style="list-style-type: none"> <li>○ Students learn about CTSOs in the State of Georgia. They research a CTSO of their choosing and explore options for joining or participating in a professional student organization. They also reflect on the benefits of involvement in student organizations to their career path.</li> </ul> </li> </ul>