

Web Development Course Syllabus

Course Overview

Web Development introduces students to the tools and technologies at the core of every modern website. In this course, students will learn to design and build webpages from scratch. They will become familiar with the different roles of the three fundamental languages of the web – HTML, CSS, and JavaScript – and understand the interplay between them. As they practice building websites, students will also learn the right way to use and share online content, contributing to a computing culture based on mutual respect and collaboration.

Prerequisites

This course is designed for students with no prior programming experience. However, a strong foundation in first-year high school algebra is recommended.

Required Resources

- Online resources and labs provided by the instructor
- Online Text: [JuiceMind.com](https://www.juicemind.com)

Course Goals

- Increase accessibility and participation in computer science.
- Demystify the Internet and the web.
- Understand core technologies of the web.
- Encourage the creation of webpages as expressions of creativity.

Course Goals (continued)

- Foster collaborative problem-solving skills.
- Promote responsible and ethical use of others' intellectual property.

Learning Environment

The course is structured as a blended learning experience, combining web-based resources with in-class activities. Students will engage in coding exercises, digital presentations, written reflections, and work collaboratively on projects. Lessons include interactive readings, projects, interactive exercises, and occasional quizzes.

Programming Environment

Students will use a browser-based editor to write and run HTML, CSS, and JavaScript files. This environment is designed to support the development of computational thinking skills from the beginning of the course.

Assessment

Students are regularly assessed for knowledge and skills through completion of programming exercises, multiple-choice questions, and larger programming projects.

- **Formative Assessment:**
 - **Mid-Lesson Review Questions:** Short multiple-choice review questions are integrated into the text of each lesson. These questions serve to ensure students are paying active attention, and reinforce comprehension of essential knowledge. Students are given only one chance to answer each question.

- Formative Assessment (continued):
 - Exercises: All lessons are followed by a short programming exercise to assess mastery of essential skills. Students are given unlimited attempts to run and test their code against the exercise requirements, but only one chance to submit it for grading. Due to the creative nature of web development, many exercises are somewhat open-ended, assessing both mastery and creative application of new skills.
- Summative Assessment:
 - Review Quizzes: All modules except those devoted to summative assessments end with a longer multiple-choice review quiz.
 - Group Projects: Most modules contain at least one hour-long collaborative programming exercise intended to be completed by groups of two to four students. These exercises are designed to be extensible, with project requirements scaling based on the number of students in each group. As with all exercises, students are given unlimited attempts to run and test their code against the exercise requirements, but only one chance to submit it for grading.
- Major Projects: This course contains two major projects to assess key programming and computer science skills. Each is contained within its own module.
 - Portfolio (Module 4): This individual project assesses students' ability to design, create, and style a single webpage from scratch.
 - Travel Guide (Module 7): This collaborative project assesses students' ability to work together to create a high-quality interactive website.

Recommended Grading Scheme

Category	Weighting	Method
Exercises	50%	Correctness (graded by teacher)
Module Review Quizzes	10% (5 total, 2% each)	Correctness (graded automatically)
Group Projects	15% (3 total, 5% each)	Quality of Submission (graded by teacher)
Project - Portfolio	10%	Quality of Submission (graded by teacher)
Group Project - Travel Guide	15%	Quality of Submission (graded by teacher)

Notes:

- Although exercises are formative assessment, we recommend they be graded for correctness instead of completion, because students have unlimited chances to verify that their solution works before submitting it.
- Due to the creative nature of web development, we recommend projects be graded at least partly on the quality of students' submissions, to encourage each student to try their best.

Pacing

Lesson plans are structured to accommodate 45-minute class periods. Lesson folders should take no more than 45 minutes in total, with most taking between 20 and 30 minutes. The instructional content of a lesson should never take more than 20 minutes to teach, and exercises should likewise never take more than 20 minutes for students to complete.

In total, JuiceMind's Web Development is designed to take 30-40 days to complete using only material included within this course. This is shorter than a typical school semester, to allow this course to be integrated into an existing computer science elective.

Course Breakdown

Module 0: Welcome to Web Development (2 periods - 1 hr 15 mins)

Students are introduced to the concept of web design and encouraged to think critically about the websites they've encountered. They will learn to use the JuiceMind IDE and understand the main objectives of this course.

- Topics Covered:
 - Web Design
 - The JuiceMind IDE
 - Web Development Course Overview

Module 1: Intro to the Web (3 periods - 1 hr 45 mins)

This module provides students essential context to begin their programming journey. It introduces the Internet, the web, and the various languages used by web developers.

- Topics Covered:
 - The Internet
 - The World Wide Web
 - HTTP
 - The roles of HTML, CSS, and JavaScript

Module 2: HTML (9 periods - 6 hrs 25 mins)

In this module, students learn how to create basic HTML webpages and websites.

- Topics Covered:
 - Markup Languages
 - HTML Document Structure
 - Tags, Elements, and Attributes
 - Hyperlinks and URLs
 - Webpages vs. Websites
 - Images, Copyright, and the Public Domain
 - Lists and Tables
 - Buttons and Embedded Javascript
 - The JavaScript Console and Popup Alerts
 - Inline CSS
 - Decimal and Hexadecimal RGB Color Codes

HTML Elements	CSS Properties	JavaScript Functions
<ul style="list-style-type: none"> ● !DOCTYPE, <html> ● <head>, <body> ● <p> ● <h1>...<h6> ● <title> ● , ● <div> ● <hr>,
 ● <a> ● ● , , ● <table>, <tr>, <td>, <th> ● <button> ● <script> 	<ul style="list-style-type: none"> ● color, background-color ● font-size ● text-align 	<ul style="list-style-type: none"> ● console.log() ● alert()

Module 3: Intro to CSS (5 periods - 3 hrs 5 mins)

In this module, students learn how to create CSS stylesheets and define custom style rules.

- Topics Covered:
 - Inline CSS vs. External Stylesheets
 - CSS Rule Structure
 - Styling Text
 - Element Backgrounds and Spacing
 - Selecting Elements by Tag, Class, and ID
 - Combining Selectors
 - The CSS Cascade

HTML Elements	CSS Properties
<ul style="list-style-type: none"> ● <link> 	<ul style="list-style-type: none"> ● border, border-radius ● width, height ● font-weight, font-style, font-family ● background-image, background-size ● margin, padding

Module 4: Project - Portfolio (2 periods - 1 hr 30 mins)

In this multi-day project, students will design and develop a portfolio webpage to highlight a few samples of their work. They'll get the chance to express themselves creatively, show off their own work, and experience the act of creating a webpage entirely from scratch.

- Topics Covered:
 - Brainstorming and Planning
 - Abstraction and Organization
 - Self-Assessment and Iteration

Module 5: Advanced HTML/CSS (5 periods - 2 hrs 50 mins)

Students explore the conveniences of modern HTML and CSS and learn how to use these concepts to elevate their webpages.

- Topics Covered:
 - Lists
 - Tuples
 - Strings as Data Structures
 - List Operations, and List Methods
 - Iterating Through a List
 - For-Each Loops and Enumeration
 - Models and Simulations

HTML Elements	CSS Properties and Selectors
<ul style="list-style-type: none"> ● <iframe> ● <header>, <nav>, <article>, <figure>, <footer> 	<ul style="list-style-type: none"> ● :hover, :link, :active, :visited ● :: first-line, ::first-letter ● ::before, ::after ● ::selection ● visibility, opacity ● display ● animation, from, to ● transition

Module 6: DOM Manipulation (3 periods - 2 hrs 10 mins)

Students learn about the Document Object Model, how browsers use it to represent an HTML document, and how to manipulate it using JavaScript.

- Topics Covered:
 - The Document Object Model
 - Selecting DOM Elements with JavaScript
 - Modifying a DOM Element’s Content and Styles
 - Adding and Removing DOM Elements
 - Event Listeners and Event Handlers

JavaScript Functions and Fields

- `document.getElementById()`, `document.getElementsByClassName()`, `document.getElementsByTagName()`
- `document.querySelector()`, `document.querySelectorAll()`
- `element.innerText`, `element.innerHTML`, `element.src`
- `element.style`
- `element.classList.add()`, `element.classList.remove()`
- `document.createElement()`, `element.remove()`
- `element.appendChild()`, `element.removeChild()`
- `element.addEventListener()`

Module 7: Project - Travel Guide (4 periods - 3 hrs)

JuiceMind's Web Development course ends with a multi-day group project. In it, students will design and develop a modern, interactive website to serve as a travel guide for a destination of their choice.

- Topics Covered:
 - Brainstorming and Planning
 - Collaboration and Division of Labor
 - Self-Assessment, Peer Assessment, and Feedback