Front-End Web Development

GENERAL ASSEMBLY

Overview

Start building responsive websites with HTML, CSS, and JavaScript — a versatile skill set with powerful applications in a variety of design, marketing, and other tech-adjacent roles.

General Assembly's Front-End Web Development course equips complete beginners with the cutting-edge tools they need to build rich, interactive websites. Dive into essential programming languages, then round out your skill set with industry-relevant topics like responsive design, APIs, and version control. You'll apply what you've learned to build a custom website from scratch.

Throughout the course, you'll:

- Discover how the web works.
- Create the structure and style of a website using HTML and CSS.
- Apply interactivity to a site with JavaScript programming fundamentals.
- Host a website on a server.
- Learn the basic technical vocabulary to communicate with front-end web developers and other technical stakeholders.
- Build a website from scratch.

What You'll Learn

Pre-Work

Welcome to Front-End Web Development

Engage in online, self-paced learning that's designed to set you up for success starting day one.

- Gain an introduction to web development.
- Get acquainted with key terminology and tools.

Unit 1

HTML/CSS Basics

Get familiar with the fundamental building blocks of the web, adding and styling content with HTML and CSS..

- Create HTML documents using common element tags.
- Inspect webpages using the browser's developer tools.
- Describe the relationship between HTML, CSS, and JavaScript in websites.
- Choose semantic HTML tags to define and organize content.
- Use CSS to apply style to webpages.
- Learn the basics of CSS syntax, including selectors and style rules.
- Link to files from HTML using relative paths.
- Use margins and padding to create spacing between elements.
- Set the display property of elements to create page layouts.

Unit 2

Responsive Design

Take a developer's approach to problem-solving, coding responsive sites for mobile and the web.

- Use flexbox properties to create responsive layouts.
- Apply normalizing CSS to avoid browser default styling interference.
- Use CSS grid properties to define responsive layouts.
- Compare and contrast flexbox and grid properties.
- Define fractional and percentage-based widths for elements.
- Define CSS rules to apply custom fonts to text.
- Use pseudo-selectors to create more specific CSS rules.
- Define media-query breakpoints to apply separate rules based on screen size.
- Use responsive measurements such as ems to smoothly scale CSS rules to devices.
- Apply a mobile-first methodology to CSS and website design.

Unit 2 Responsive Design (Cont.)

- Use the position property to create responsive layouts.
- Define and choose between absolute, relative, static, and fixed positioning.

Unit 3 Adding Interactivity With JavaScript

Power dynamic websites, incorporating animations, dropdowns, and more.

- Distinguish between code and a program.
- Define basic variables and data types in JavaScript.
- Understand the role of functions in JavaScript.
- Describe the role of JavaScript in front-end web development.
- Access properties of the DOM using JavaScript object syntax.
- Use DOM methods to respond to user actions with event listeners.
- Use JavaScript to trigger CSS animations.
- Design interactive user interfaces using CSS properties.
- Plan application states to reflect user actions.
- Define conditional statements in JavaScript to create logic-driven programs.
- Choose logical operators to enhance conditional statements.
- Use arrays and loops in JavaScript to manage collections of data.
- Invoke array methods to manipulate the array's contents.
- Distinguish between for loops and while loops.

Unit 4 Advanced Concepts

Take your coding skill set to the next level, incorporating functions from external libraries and tools.

- Use HTML forms to collect input from users.
- Evaluate APIs based on documentation.
- Make HTTP requests to external API sources for data.
- Use CSS libraries, such as Bootstrap, to leverage pre-styled components.
- Adapt CSS library components for specific use cases.
- Evaluate CSS frameworks against self-written CSS.
- Use GitHub for version control and code collaboration.
- Execute commands from the command prompt for increased control of file systems.

JavaScript Development

GENERAL ASSEMBLY

Overview

Gain fluency in JavaScript — the world's most popular programming language — and start leveraging its versatile capabilities to build rich, interactive websites and applications.

General Assembly's JavaScript Development course teaches students Intermediate front-end development skills using JavaScript, jQuery, Git and GitHub, and the command line. You'll apply what you've learned to build a modern, single-page web application utilizing industry best practices.

Throughout the course, you'll:

- How to work with JavaScript, jQuery, the browser, and the Document Object Model (DOM).
- The fundamentals of JavaScript in order to more easily learn how to work with associated frameworks and libraries.
- The essentials of object-oriented programming so that you can more easily learn another object-oriented language.
- How to consume data from APIs and persist data using a back-end-as-a-service provider like Firebase.
- How to build a modern, single-page application using common design patterns.

What You'll Learn

Pre-Work

Welcome to JavaScript Development

Engage in online, self-paced learning that's designed to set you up for success starting day one.

- Gain an introduction to JavaScript and object-oriented programming.
- Get acquainted with fundamental terminology and tools.

Unit 1

Fundamentals of JavaScript HTML/ CSS Basics

Learn the fundamentals of JavaScript and object-oriented programming by working with JavaScript on the command line.

- Explore how the web works and the client-server model.
- Compare JavaScript in the browser versus the command line.
- Run JavaScript code on the command line using Node.js.
- Gain an introduction to working with variables and conditionals.
- Understand data types.
- Work with collections and loops, and iterate over collections.
- Write functions.
- Discover the concept of scope.

Project: Using the provided scaffolding, build a basic Slackbot that responds to user input and run it locally from your machine in the class Slack channel.

Unit 2

The Browser and APIs

Use JavaScript to interact with the browser, the Document Object Model, and APIs.

- Get acquainted with objects and methods.
- Work with JSON-formatted data.
- Explore the jQuery library and its capabilities.
- Understand the Document Object Model (DOM) and manipulate objects in the DOM.
- Handle forms and user input.
- Use events and listeners.
- Gain an introduction to AJAX.
- Make API calls, consuming and incorporating API data.
- Compare and contrast asynchronous and synchronous JavaScript.
- Leverage callbacks.

Project: Build a simple, single-page application that consumes data from a third-party API like Twitter, Facebook, or Instagram.

Unit 3 Persisting Data and Advanced Topics

Use advanced programming topics and persist user data via API calls to a back-end service provider.

- Dive into authentication, working with tokens and API keys.
- Utilize OAuth.
- Get acquainted with prototypal inheritance, prototypes, and constructors.
- Explore the concept of "this."
- Handle anonymous functions.
- Understand CRUD.
- Gain an introduction to Firebase and write user data to it.
- Retrieve and update user data.

In-Class Lab: Begin working on your final project: a single-page application that consumes data from at least one API and persists user data via Firebase.

Unit 4 Building and Deploying Your App

Work on your final project and deploy your app using GitHub Pages or Heroku.

- Get started with advanced JavaScript frameworks.
- Explore app deployment strategies.
- Deploy your app to GitHub Pages or Heroku and use a custom domain name.
- Use Firebase with Heroku.

Final Project: Present your capstone piece: a single-page application that consumes data from at least one API and persists user data via Firebase.

React Development

GENERAL ASSEMBLY

Overview

Facebook created React to build a robust, dynamic platform that could adapt to continually changing data. To date, this JavaScript library fuels countless websites and applications across industries.

Learn to leverage React's power in this hands-on, project-based course. Build your own web application and compile a professional project portfolio to showcase your new skills.

What You'll Learn

Pre-Work

- Dive into React fundamentals with four hours of online preparatory lessons and prepare to hit the ground running on day one of class.
- Get acquainted with ES6.
- Start building a custom blog application.

Unit 1 Key React Concepts (7 hours)

Lab: Add nested components to your blog application.

- Apply React fundamentals to solve common user interface (UI) problems.
- Render components within another component.
- Pass props to a nested component.

Unit 2 React State (7 hours)

Lab: Use state to create editable entries in your blog application.

- Differentiate between props and state.
- Create and change state in a component.
- Describe the flow of methods in a component.
- Identify the triggers for the re-rendering of a component.
- Contrast class components with functional components.
- Define unidirectional flow.
- Diagram data in a component hierarchy.

Unit 3 Underlying Concepts (3 hours)

Lab: Implement functional components in your blog application.

- Rewrite class components into functional components.
- Define the main categories of the component life cycle.
- Identify general methods in each category of the component life cycle.
- Contrast the concepts imperative and declarative programming.

Unit 4 APIs and Heroku (3 hours)

Lab: Use an API to display the current weather on your blog application.

- Describe what an application programming interface (API) is and why we might use one.
- Using fetch() to make an API call and working with API keys.
- Describe Heroku.
- Deploy an app on Heroku.
- Set up a CORS proxy on Heroku.

Unit 5 React Router (4 hours)

Lab: Use React Router to add multiple pages to your blog application.

- Contrast historical and modern browser history mechanics.
- Define routing.
- Describe React Router's main features and history.
- Use React Router to map URLs to components.
- Use React Router to create links to different components.

Unit 6 Applied Practice (16 hours)

Final Lab: Build a working React application of your choice from scratch. Prompts and guidelines are available for students who would like help brainstorming ideas.

- Build a tic-tac-toe game.
- Confidently find and apply features from documentation.
- Create an ATM application.