Front-End Fundamentals Bootcamp Syllabus

4 Weeks Part-Time

https://onecodecamp.com/

Module 1: HTML & CSS

1.1) Introduction to HTML

Objectives:

- Understand the purpose and importance of HTML in web development.
- Learn about the basic structure of an HTML document.
- Explore text formatting options in HTML.
- Familiarize with creating lists and linking webpages. Understand the concept of HTML forms and how to create form elements.

1.2) Advance HTML

Objectives:

- Understand the concept of semantic HTML and its importance in web development.
- Learn about advanced form elements and attributes.
- Explore tables and their usage in structuring data.

1.3) Mastering the Terminal

Objectives:

- Navigate directories and file structures efficiently using basic terminal commands.
- Create, rename, and delete directories and files, and manage file permissions.
- Utilize text editors within the terminal to create, edit, and save files.
- Manage processes effectively by listing, killing, and monitoring their activity.
- Search for files and directories, perform bulk renaming, and manipulate text using regular expressions.
- Install, update, and remove software packages using package managers.
- Establish remote connections via SSH, transfer files securely using SCP and SFTP.
- Configure environment variables and modify shell configuration files.
- Automate tasks by writing and executing shell scripts, including handling command-line arguments.

1.4) Git and Using Github

Objectives:

- Understand the fundamentals of version control systems and the benefits of using Git.
- Learn how to initialize a Git repository and configure Git on your local machine.
- Master basic Git commands for tracking changes, committing, and managing versions.
- Gain proficiency in branching and merging strategies to effectively manage project development.
- Learn how to collaborate with remote repositories and GitHub for seamless teamwork.
- Understand the concept of forking and its significance in contributing to open-source projects.

1.5) CSS Introduction

Objectives:

- Understand the role of CSS in web development.
- Learn CSS syntax and rules.
- Gain proficiency in using selectors and understanding specificity.
- Explore different methods of applying CSS: inline, internal, and external.
- Learn how to use CSS comments for documentation and organization.
- Explore CSS selectors and their importance.
- Understand the box model and its components: margin, padding, border, and content.
- Learn about the box-sizing property and its impact on element sizing.
- Explore the display property and its values: block, inline, and inline-block.
- Gain knowledge of different positioning techniques: static, relative, absolute, and fixed. Learn how to use floats and clear floats effectively.
- Understand how to center elements horizontally and vertically.
- Learn about typography and text styling: setting font families and sizes, text formatting, alignment, decoration, line height, letter spacing, and text shadows.
- Explore web-safe fonts and using Google Fonts to enhance typography. Module 5: Colors, Backgrounds, and Images
- Gain proficiency in using color values: named colors, hexadecimal, RGB, and HSL.
- Learn how to set background colors and images, position and repeat backgrounds, and control opacity and transparency.
- Understand how to add images to web pages, resize them, and align them properly.

1.6) CSS Intermediate

Objectives:

- Understand the concept of responsive design and its importance
- Use CSS media queries to target different screen sizes and devices
- Create fluid layouts using percentage and viewport units Implement responsive images and videos for optimal display
- Adopt a mobile-first approach in designing responsive websites
- Learn about CSS Flexbox and its properties for flexible layouts
- Understand the concepts of flex containers and flex items
- Control the flex direction, alignment, and justification of flex items
- Explore CSS Grid layout and its properties for grid-based designs
- Understand grid containers and grid items within the grid layout Define grid templates, columns, and rows for effective grid-based designs

1.7) Intro to Bootstrap

Objectives:

- Understand the fundamentals of Bootstrap
- Learn about the benefits of using Bootstrap in web development
- Set up Bootstrap in a project using CDN and local files
- Understand the Bootstrap grid system and its responsive layout capabilities.
- Create responsive layouts using container, row, and column classes.
- Implement nested columns and offsetting columns for advanced layouts.

- Style text and headings using Bootstrap classes
- Apply text alignment and color classes
- Utilize utility classes for common tasks like margin, padding, and display properties
- Create responsive navigation bars using Bootstrap's Navbar component
- Add navigation links and dropdown menus to the Navbar
- Create forms with Bootstrap styles and validation
- Utilize various form controls like input fields, checkboxes, and radio buttons
- Customize form layouts and styles using Bootstrap classes
- Utilize Bootstrap's pre-built components like cards, alerts, and modals
- Add interactive modals for displaying additional content
- Customize component styles and behaviors using Bootstrap classes

1.8) Extending Bootstrap

Objectives:

- Understand the concept of extending Bootstrap and its benefits.
- Explore the customization options available in Bootstrap.
- Set up the development environment for customizing Bootstrap.
- Learn how to modify Bootstrap's default styles using custom CSS.
- Customize Bootstrap's default color palette to align with your brand. Modify typography and font styles to create a unique look and feel.
- Create and apply custom color schemes to enhance visual appeal.
- Style headings, paragraphs, and other text elements according to your design requirements.
- Understand the structure and classes of Bootstrap components.
- Use custom CSS to modify component styles and override default behavior.
- Create custom styles for buttons, forms, and cards to match your design.
- Extend and customize individual components to add additional functionality or visual enhancements.
- Understand the Bootstrap theme customization process.
- Modify the default Bootstrap theme using custom CSS to create a cohesive brand identity.
- Utilize Bootstrap variables and overrides to create a custom theme.
- Implement custom branding elements like logos and colors to align with your brand image.

Module 2: Javascript

2.1) Introduction to JS

Objectives:

- Understand the role of JavaScript in web development and its significance in creating interactive websites.
- Set up a JavaScript development environment, including installing necessary tools and configuring a code editor.
- Familiarize yourself with the basic syntax of JavaScript, including statements, variables, and data types.
- Gain an understanding of type coercion and how it impacts JavaScript code execution.
- Explore various operators and expressions in JavaScript and their usage.
- Learn about control structures, including conditional statements and loops, and how to use them to control program flow and make decisions.
- Explain the importance of Arrays
- Enumerate the most common array methods in JS
- Create Object Literals
- Explain the importance of Object Literals in JS

2.2) Functions & Scope:

Objectives:

- Understand the concept of functions as reusable blocks of code.
- Recognize the importance of functions in organizing and modularizing code.
- Explain the benefits of using functions for code reusability and maintainability.
- Differentiate between function declaration and function expression syntax.
- Understand how function declarations and function expressions are defined and invoked.
- Explain the differences in hoisting behavior between function declarations and function expressions.
- Understand the purpose and usage of function parameters and arguments.
- Differentiate between formal parameters and actual arguments.
- Explain how return values allow functions to produce results or values.
- Understand the concept of variable scope and how it relates to functions.
- Explain the difference between global scope and local scope.
- Understand lexical scoping and how closures allow functions to retain access to variables from their parent scopes.
- Function hoisting and anonymous functions:
- Understand the concept of hoisting and its impact on function declarations and variable declarations.
- Explain the behavior of hoisting with function declarations.
- Understand the use of anonymous functions and their benefits in certain scenarios.
- Understand the concept of higher-order functions and their ability to accept and/or return other functions.
- Explain the role of callbacks as function arguments that are invoked later.
- Understand how higher-order functions and callbacks facilitate code abstraction and composition.

2.3) DOM Manipulation 1

Objectives:

- Understand the structure and purpose of the Document Object Model (DOM).
- Learn how to access and manipulate DOM elements using JavaScript.
- Modify content, attributes, and styles of DOM elements dynamically.
- Create and remove elements dynamically within the DOM.
- Traverse the DOM tree to access parent and child elements.
- Handle events and utilize event delegation for efficient event management.

2.4) DOM Manipulation 2

Objectives:

- Apply DOM manipulating techniques to build small JS applications
- Explain the process of building a small JS application

2.5) Modern JS Features

Objectives:

- Understand the key features introduced in ES6 (ECMAScript 2015) and their benefits.
- Learn to use arrow functions as concise alternatives to traditional function expressions.
- Explore template literals for more flexible and readable string interpolation.
- Understand destructuring and how it simplifies the extraction of values from arrays and objects.
- Master the spread and rest operators for easily merging arrays, passing multiple function arguments, and working with objects.
- Discover enhanced object literals for concise property and method definitions.
- Learn about default parameters and how they provide fallback values in function declarations.
- Understand the differences between function expressions and arrow functions, and learn to transform one into the other.
- Gain practical experience by implementing code examples that utilize these modern JavaScript features.
- Recognize the advantages of using these features in terms of code readability, maintainability, and developer productivity.

2.6) Asynchronous JS

Objectives:

- Understand and apply asynchronous programming concepts in JavaScript.
- Identify and address the challenges posed by the callback hell problem.
- Utilize Promises and chaining for effective asynchronous programming.
- Implement async/await syntax for modern asynchronous JavaScript.
- Handle errors in asynchronous operations using appropriate techniques.
- Make HTTP requests and handle responses using the Fetch API.

2.7) JS OOP

Objectives:

- Understand OOP principles: Learn the core principles of OOP, including encapsulation, inheritance, and polymorphism.
- Create and use classes: Learn how to define classes to create objects with properties and behaviors.
- Encapsulate data and behavior: Bundle data and related methods together within classes to ensure data integrity and controlled access.
- Apply inheritance: Create child classes that inherit and extend properties and behaviors from parent classes.
- Implement polymorphism: Write code that can work with objects of different types by treating them as instances of a common superclass.
- Model real-world entities: Use OOP to model real-world entities by breaking them down into objects and defining their relationships.
- Design reusable and maintainable code: Design classes and objects that are modular, extensible, and easy to maintain.

Module 3: React.js

3.1) Intro to React

Objectives:

- Understand the fundamentals and benefits of React.
- Learn to create components using JSX syntax and handle props.
- Use ternary operators for conditional rendering, work with lists, and apply basic CSS styling.
- Understand the concept of state and manage it using the useState hook.
- Build and handle forms, capture and validate user input.
- Implement CRUD functionality, create, read, update, and delete tasks in a ToDo list application.
- Understand the component lifecycle and handle side effects using the useEffect hook.
- Fetch data from APIs and update the UI with the retrieved data.
- Navigate between different views using React Router DOM, style components with React Bootstrap 5.