Front-End Web Developer Program - Online

BEGINNER **INTERMEDIATE** ADVANCED

The Front-End Web Developer program is right for learners who want to design and build next-generation front-end interfaces for the web.

PREREQUISITE REQUIREMENTS

- Beginner exposure to HTML, CSS and/or Javascript is highly recommended, but no programming experience is required to take this program
- Students are expected to have high-school level mathematics (e.g. up through Algebra 2) as minimum prerequisite

The coursework and projects will allow students to:

- Build HTML, CSS, and JavaScript components and pages
- Interact closely with back-end developers to help specify Restful API to various Data Sources
- Implement sites with style/brand guidelines and standards
- Develop styles with CSS pre-processors
- Utilize Restful API's from a variety of sources
- Mastery of Javascript and client-side MVC
- Mastery of CSS and familiarity with CSS frameworks

Front-End Web Developer Course Descriptions - Online

The following classes are part of the Front-End Web Developer program:

Intro to Web Development - Online, 33.5 hrs
Learn to both convert other people's designs into existing web pages AND prototype your own designs in HTML/CSS. Students will learn the basic principles of DOM to developing your own portfolio site with a responsive CSS framework. This course introduces the way students need to shift their thinking as they go from desktop first design, to responsive design. Lessons will cover the important theoretical concepts of responsive design and include plenty of hands-on exercises implementing what students learn. Students will learn how to work with images on the modern web, so that their images look great and load quickly on any device. Along the way, students will pick up a range of skills and techniques to smoothly integrate responsive images into their development workflow. By then end of the course, students will be developing with images that adapt and respond to different viewport sizes and usage scenarios.

JavaScript Basics - Online, 11.5 hrs
Students get an abbreviated exploration of the JavaScript programming language. We'll teach JavaScript and programming fundamentals needed to build their own interactive résumé. We'll give students the template styles and code to create a modern and mobile friendly résumé that they can modify and customize using the skills learned in this course.

Object-Oriented JavaScript - Online, 18.5 hrs
This course is designed to teach web developers how to utilize the various object-oriented programming features within JavaScript and, more importantly, how to write reusable and maintainable libraries that will make your life easier. After completing this course, students will have a strong understanding of JavaScript's object-oriented features and how these features affect their code's execution and memory model. They'll gain
an appreciation for the various inheritance models JavaScript provides, use these features to write memory-efficient code, and focus on simplicity and modularity in their own code.

Website Performance Optimization - Online, 10 hrs

Students learn about the Critical Rendering Path, or the set of steps browsers must take to convert HTML, CSS and JavaScript into living, breathing websites. From there, they explore and experiment with tools to measure performance, and learn simple strategies to deliver the first pixels to the screen as early as possible. Finally, students dive into recommendations from PageSpeed Insights and the Timeline view of Google Chrome.

Intro to AJAX - Online, 6 hrs

Students learn how to make asynchronous requests with JavaScript (using jQuery's AJAX functionality), and gain a better understanding of what's actually happening when they do so. They'll also learn how to use data APIs, so you can take advantage of freely accessible data in their applications, including photo results, news articles and up-to-date data about the world around us.

JavaScript Testing - Online, 6.5 hrs

This course covers test-driven development cycle (red/green/refactor). Students will progress from identifying expectations within small code samples and reframing those expectations

as test cases, all the way to writing a comprehensive suite of test cases against an application specification, as well as the application code that would result in a passed test.

Intro to JQuery - Online, 8 hrs

Students learn how to use jQuery to select and navigate to DOM elements within a page, how to manipulate DOM elements by altering attributes, how to dynamically change content and how to add/remove DOM elements. Most importantly, students practice making sense of jQuery's documentation so that they can go beyond what they learn in the class and take advantage of jQuery's full suite of features.

HTML5 Canvas - Online 5 hrs

Students will learn how the Canvas 2D API works and how to use it to create interesting applications. They will get an overview of what the Canvas is, how it affects graphics in the browser and what makes it truly awesome. Students will make compositions with text and images (e.g. memes) as they learn the API. They'll also learn about how images are stored by the Canvas2D context, how to modify them on a pixel level by applying various effects and filters, and how to create animations.

Browser Rendering Optimization - Online, 27.5

Demystifying the browser's rendering pipeline will make it easy for students to build high-performance web apps. Students

will start by getting introduced to the individual steps of the rendering pipeline, beginning with parsing HTML and ending with painting pixels on the screen. Then, they will quickly dive into tooling with ample opportunities to practice profiling and debugging apps with Chrome Developer Tools.

Javascript Design - Online, 24.5 hrs

This course covers methods for organizing code, both conceptually and literally. Students learn the importance of separating concerns when writing JavaScript, gaining hands-on experience along the way. Separating concerns can be done with or without an organizational library or framework. Students will learn how to separate concerns without one and explore an organizational library with guidance. They'll also learn strategies for exploring other libraries and frameworks on their own. By the end of this course, students will understand (from experience) the importance of code organization, and how to implement it with either vanilla JavaScript or an organizational library or framework. Their applications will start looking clean and professional—not just to their users, but also to anyone who looks at the code driving your applications.

Front-End Web Developer Final Project Portfolio

By the end of this program, students will have a portfolio that includes the following projects:

Build A Portfolio Site
Students can accurately replicate a provided design in HTML/CSS, using appropriate semantic HTML tags. CSS should take advantage of the "cascading" functionality and include appropriate vendor prefixes. Students will be provided a design mockup (PDF, PNG, etc.) and must replicate that design in HTML/CSS. Students will develop a responsive single page to display an image, description and link to each of the portfolio projects they complete throughout the course of the Front-End program.

Interactive Resume
Students will develop an interactive resume application that reads their resume content from a JSON file and displays the content within the template provided. Students will use objects, functions, conditionals and control structures to compose the content that will display within the resume.

Classic Arcade Game Clone (Frogger Game)
Students will be provided visual assets and a game loop engine; using these tools they will add a number of entities to the game including their character (frog) and various types of vehicles. Students are required to implement these entities as classes and/or subclasses.

Mobile Website Optimization
Students are provided a website and must identify and optimize a variety of performance issues within, achieving a target PageSpeed score and running at 60 frames per second.

Project Rubric

Each project will be evaluated in key areas by a Udacity project reviewer. All criteria must "meet specifications" in order to pass. For example, to complete the project, ***Classic Arcade Game Clone****, students must meet specifications for the following criteria:*

1. **Game Functions:** The game functions correctly and runs error-free.

2. **Object-oriented Code:** Game objects (player and vehicles) are implemented using JavaScript's object-oriented programming features.

3. **Code Quality:** Code is formatted with consistent, logical and easy to read formatting as described in the Udacity JavaScript Style Guide.

4. **Comments:** Comments are present and effectively explain longer-code procedures.

5. **Documentation:** A README file is included detailing all the steps required to successfully run and play the application.