UDACITY

OneTen Scholarship

Frontend Web Development Fundamentals

Frontend Web Development Fundamentals

The goal of the Frontend Web Development Fundamentals program is to equip learners with the unique skills they need to build and develop a variety of websites and applications. Graduates of this Nanodegree program will be able to construct responsive websites using CSS, Flexbox and CSS Grid, develop interactive websites and UI (User Interface) applications using JavaScript and HTML.

Part 1 - Intro to Web Development

In this part, you'll learn how to make basic web pages using HyperText Markup Language (HTML) and how to add style to your pages with Cascading Style Sheets (CSS). You'll begin by learning some basics about how the Web works, then build a very basic web page using only HTML, and finally explore how to add styles to your page with CSS. At the end of the course, you'll demonstrate your new skills by completing a project in which you create a web page that replicates a given design.

Project: Animal Trading Cards

For this project, you'll use HTML and CSS to make Animal Trading Cards. You will apply your knowledge of HTML Document Structure to your html file and then create custom CSS styling based on your preferences. This project will demonstrate your understanding of linking CSS files in HTML files, implementing CSS classes to avoid repetition, as well create semantically organized HTML code.

Supporting Lesson Content

Lesson Title	Learning Outcomes
The Web and HTML	➔ Describe the fundamentals of how the web works ➔ Edit web pages using a text editor and test work in the browser ➔ Create HTML files that use elements and tags to provide the structure of a web page ➔ Write fully qualified URL pathways by identifying each part of file path structures
Lab: Basic HTML Page	➔ Demonstrate your understanding of HTML basics by creating a simple web page
Styling with CSS	➔ Use CSS to change basic style properties, like the font, color, and border of a given element ➔ Use CSS type and class selectors to apply style to specific subsets of HTML elements ➔ Separate the style of a web page from its structure and semantics ➔ Apply style in multiple ways, including via a separate, linked stylesheet ➔ Recognize tree structures in HTML and CSS code ➔ Modify the layout and resizing behavior of a web page using containers and the flexible box model (flexbox) ➔ Use Developer Tools to inspect the elements of a web page

Part 2 - Intermediate CSS: Flexbox and Grid

In this part, you'll learn how to create a multi-page website, using best practices for content and page styling with HTML and CSS. You'll practice using responsive layouts, Flexbox, and CSS Grid to create the structure and design your own website.

Project: Personal Blog Website

For this project, you'll use HTML and CSS Flexbox to create a responsive and good looking personal blog website that fits all kinds of screens.

Supporting Lesson Content

Lesson Title	Learning Outcomes
CSS Flexbox	➔ Control web elements orientation and layout with Flexbox ➔ Control ordering of web elements with Flexbox ➔ Align and justify web elements with Flexbox ➔ Transform and resize web elements with Flexbox
CSS Grid	➔ Compare and contrast the use cases for CSS Grid and Flexbox ➔ Structure the layout of a web page using grid columns and rows
Creating Responsive Layouts	➔ Define custom styles for different screen sizes using media queries ➔ Observe and create breakpoints in a website to change layout and styling as a page is resized

Part 3 - Intro to JavaScript

In this part, you will learn the history of JavaScript and how it compares to Python programming. You will understand how the DOM is formed, what nodes and elements are, and how to select items from the DOM. By the end, you'll write JavaScript code that allows the user to create a grid of squares representing their design, and apply colors to those squares to create a digital masterpiece

Project: Pixel Art Maker

For this project, you'll build a single-page web app that allows users to draw pixel art on a customizable canvas!

Supporting Lesson Content

Lesson Title	Learning Outcomes
What is JavaScript	➔ Understand the history of JavaScript and start writing your code immediately using the JavaScript console

Data Types & Variables	➔ Learn to represent real-world data using JavaScript variables and distinguish between the different data types in the language
Conditionals	➔ Learn how to add logic to your JavaScript programs using conditional statements
Loops	➔ Harness the power of JavaScript loops to reduce code duplication and automate repetitive tasks
Functions	➔ Dive into the world of JavaScript functions. Learn to harness their power to streamline and organize your programs
Arrays	➔ Learn how to use arrays to store complex data in your JavaScript programs
Objects	➔ Meet the next JavaScript data structure: the object. Learn to use it to store complex data alongside arrays

Part 4 - JavaScript & The DOM

In this part, you will use JavaScript to control a webpage. You will learn what the Document Object Model (DOM) is, and use JavaScript and the DOM to dictate page content and interactions. You will gain experience working with Browser Events and managing website performance by controlling content creation efficiently.

Project: Landing Page

For this project, you'll build a single-page web app that allows users to draw pixel art on a customizable canvas!

Supporting Lesson Content

Lesson Title	Learning Outcomes
Syntax	➔ Understand the history of JavaScript and start writing your code immediately using the JavaScript console
The Document Object Model	➔ Understand how the DOM is formed, what nodes and elements are, and how to select items from the DOM
Creating Content with JavaScript	➔ Use JavaScript and DOM methods to create new page content, update existing content and delete content
Working with Browser Events	➔ Learn what an event is, how to listen for an event and respond to it, what data is included with an event, and the phases of an event
Performance	➔ Manage website performance by controlling content creation efficiently. • Describe what happens when a webpage has to be redrawn • Describe and explain

	the JavaScript call stack ➔ Describe and explain the JavaScript event loop ➔ Write efficient code by analyzing the call stack and event loop ➔ Delay code execution with setTimeout