Front-End Web Developer Nanodegree Syllabus

Build Stunning User Experiences

Before You Start

You've taken the first step toward becoming a web developer by choosing the Front End Web Developer Nanodegree program. In order to succeed, we recommend having experience using the web, being able to perform a search on Google, and (most importantly) the determination to keep pushing forward! Prior programming experience is not required, but if you'd like to prepare for this Nanodegree, check out our Intro to HTML & CSS course.

The Front-End Web Developer Nanodegree is composed of 8 projects. With each project, you'll create something to demonstrate your mastery of in demand skills. Projects range in complexity and each builds upon the last. In the end, you will have built a portfolio of projects, including a select set that are resume worthy.

Project: Mockup to Article

In this project, you'll be given a design mockup that you will convert into a website built with HTML. You'll need to carefully examine the mockup to determine the specific tags to use to achieve the correct visual and structural result.

Supporting Lesson Content: HTML Syntax

Lesson Title	Learning Outcomes
HTML Syntax	➔ Identify the parts that make up an HTML tag ➔ Determine when to use specific HTML tags ➔ Correctly structure nested HTML content ➔ Decide between a variety of text editors for writing code

Project: Animal Trading Cards

In this project, you'll be creating a trading card for your favorite animal. You will use your knowledge of HTML to create the structure for your trading card. Then you will use CSS styling to design your trading card.

Supporting Lesson Content: CSS Syntax

Lesson Title	Learning Outcomes
CSS Syntax	➔ Identify the benefit of separating style from content ➔ Use CSS to style a website ➔ Test styles by manipulating CSS properties ➔ Use CSS references to lookup standard CSS properties and values
How to Write Code Faster	➔ Use keyboard shortcuts to write code faster ➔ Use code editor packages and themes to improve workflow and write code more efficiently

Project: Build a Portfolio Site

For this project, you'll be building a portfolio website. You will be provided a design mockup as a PDF-file, and you must replicate that design in HTML and CSS. You will develop a responsive website that will display images, descriptions and links to each of the portfolio projects you will complete through the course of your Nanodegree program on any size of screen.

Supporting Lesson Content: Responsive Web Design, Intro to JavaScript, and ES6

Lesson Title	Learning Outcomes
Why Responsive **Making Sites Responsive**	➔ Create your own responsive web page that works well on any device: phone, tablet, desktop or anything in between. ➔ Explore what makes a site responsive and how some common responsive design patterns work across different devices. ➔ Create your own responsive layout using the `viewport` tag and CSS media queries. ➔ Experiment with major and minor breakpoints ➔ Optimize text for reading.
Starting Small	➔ Build HTML elements for any screen size. ➔ Use the browser viewport to create consistent user experiences.
Building Up	➔ Use media queries and breakpoints to create responsive web page designs ➔ Create flexible HTML elements with an introduction to Flexbox
What Is JavaScript	➔ Gain insight on history of JavaScript. ➔ Begin writing code immediately using the JavaScript console.
Data Types & Variables	➔ Represent real-world data using JavaScript variables. ➔ Recognize distinctions between different data types.
Conditionals	➔ Use conditional statements to add logic and control flow into JavaScript programs.
Loops	➔ Reduce code duplication and automate repetitive tasks by leveraging JavaScript loops.
Functions	➔ Harness the power of functions to streamline and organize your programs.
Arrays	➔ Leverage, arrays to store complex data in JavaScript programs.
Objects	➔ Alongside arrays, use objects to store complex data.
ES6 Syntax	➔ Utilize syntax improvements that have been made to the JavaScript language.

Project: Pixel Art Maker

In this project, you'll build a single-page web application that allows users to draw pixel art on a customizable canvas. You'll be given starter code, including HTML and CSS, to build the application. You'll then use your JavaScript and jQuery skills to manipulate the DOM, allowing users to create a digital masterpiece!

Supporting Lesson Content: Shell Workshop, Version Control with Git & GitHub, and jQuery

Lesson Title	Learning Outcomes
Shell Workshop	➔ The Unix shell is a powerful tool for developers of all sorts. You'll get a quick introduction to the very basics of using it on your own computer.
What is Version Control	➔ You'll learn about the benefits of version control and install the version control tool Git!
Create A Git Repo	➔ Create a new repository from scratch ➔ Cloning an existing repository.
Review A Repo's History	➔ Review an existing Git repository's history of commits. ➔ Change how Git Log displays information. ➔ View files that have been modified. ➔ View changes that have been made.
Add Commits To A Repo	➔ Make commits that are saved to the repository. ➔ Write descriptive commit messages. ➔ Verify the changes you're about to save to the repository.
Tagging, Branching, and Merging	➔ Add special markers called tags to commits. ➔ Work on isolated development tracks by making use of Git's branches. ➔ Combine branches together.
Undoing Changes	➔ Modify or undo changes that have been saved to a repository.
Working With Remotes	➔ Create remote repositories on GitHub. ➔ Get and send changes to a remote repository.
Working On Another Developer's Repository	➔ Create copies of a project by forking another developer's repository. ➔ Collaborate with other developers by contributing to a public project.
Staying In Sync With A Remote Repository	➔ Leverage pull requests to send suggested changes to another developer. ➔ Move or combine commits with `git rebase`.

jQuery Basics: the DOM, $, and Selectors	➔ Explain the purpose of jQuery ➔ Use jQuery to select elements from the page ➔ Use jQuery methods to filter the list of selected items
The Tricks: DOM Manipulation	➔ Modify the classes on an element ➔ Change an element's attributes ➔ Add/remove DOM elements ➔ Run a block of code over each item in a set
Event Listeners with jQuery	➔ Listen for browser events and run code in response ➔ Identify different event types ➔ Monitor types of events ➔ Use event delegation to minimize number of event listeners

Project: Memory Game

In this project, you'll demonstrate your mastery of HTML, CSS, and JavaScript by building a complete browser-based card matching game (also known as Concentration). From building a grid of cards, adding functionality to handle user input, and implementing gameplay logic -- you'll combine all your web development skills to create a fully interactive experience for your users.

Supporting Lesson Content: Object Oriented JavaScript

Lesson Title	Learning Outcomes
Scope	➔ Explain the importance of scope in JavaScript ➔ Identify the execution context of functions ➔ Create functions that run outside of their lexical scope
Closure	➔ Explain how functions and function scope create closure ➔ Trace identifier lookup during function execution
The 'this' Keyword	➔ Identify how the `this` keyword is bound ➔ Determine the `this` value in a function ➔ Discover pitfalls when the `this` value loses its context
Prototype Chains	➔ Describe what a prototype chain is ➔ Explain how a prototype method is accessed by an instance ➔ Use a "constructor" function to create similar objects
Object Decorator Pattern	➔ Use the Decorator Pattern to add data and functionality to an object ➔ Remove duplication by using inheritance
Functional Classes	➔ Identify the purpose of a Constructor ➔ Use the `this` value to alter an object in a Constructor ➔ Add data to a constructed object ➔ Look up data on a constructed object
Prototypal Classes	➔ Identify the purpose of a Constructor's .prototype property ➔ Ddd methods to a Constructor's .prototype property
Pseudoclassical Patterns	➔ Identify the pseudoclassical pattern
Superclass and Subclasses	➔ Remove duplication by using Superclasses and Subclasses
Pseudoclassical Subclasses	➔ Use the Pseudoclassical pattern to create instances of objects ➔ Explain how .call works with functions ➔ Use .call to link functions together ➔ Use Object.create() to create objects that delegate functionality

Project: Classic Arcade Game Clone

In this project, you'll recreate the classic arcade game Frogger. You will be provided visual assets and a game loop engine; using these tools you must add a number of entities to the game including the player characters and enemies.

Supporting Lesson Content: READMEs and ES6

Lesson Title	Learning Outcomes
Writing READMEs	➔ Identify Markdown syntax ➔ Explain importance of documentation ➔ Write Markdown to document project instructions and information
ES6 Functions	➔ With ES6, functions are getting some much-needed improvements. Learn a number of new things including arrow functions and classes.
ES6 Built-ins	➔ The JavaScript environment provides you with a number of features by default. You'll learn about Sets, Maps, Proxies, Generators, how iteration works, and more!
ES6 Professional Developer-fu	➔ With this massive improvement, not all browsers are able to support this new version of JavaScript. You'll learn about using polyfills and transpiling your ES6 JavaScript code to ES5.

Project: Feed Reader Testing

In this project, you'll be learning about testing with Javascript. Testing is an important part of the development process and many organizations practice a standard known as "test-driven development" or TDD. This is when developers write tests first, before they ever start developing their application. Whether you work in an organization that writes tests extensively to inform product development or one that uses tests to encourage iteration, testing has become an essential skill in modern web development!

Supporting Lesson Content: JavaScript Testing

Lesson Title	Learning Outcomes
Rethinking Testing	➔ Explain the benefits of Test-Driven Development ➔ Use tests to identify expectations of code functionality
Writing Test Suites	➔ Use the Jasmine testing framework ➔ Identify the key functions that make up the Jasmine framework ➔ Explain the Red-Green-Refactor life cycle of testing ➔ Write Jasmine tests to validate asynchronous code

Project: Neighborhood Map

In this project, you will develop a single-page application featuring a map of your neighborhood or a neighborhood you would like to visit. You will then add additional functionality to this application, including: map markers to identify popular locations or places you'd like to visit, a search function to easily discover these locations, and a listview to support simple browsing of all locations. You will then research and implement third-party APIs that provide additional information about each of these locations (such as StreetView images, Wikipedia articles, Yelp reviews, etc).

Supporting Lesson Content: JavaScript Promises

Lesson Title	Learning Outcomes
Creating Promises	➔ Learn what a promise is, how it makes writing asynchronous JavaScript simpler and how to handle errors.
Chaining Promises	➔ Create sequences of asynchronous work by chaining Promises together and dive into more advanced error handling.

Supporting Lesson Content: Asynchronous JavaScript

Lesson Title	Learning Outcomes
Ajax with XHR	➔ Connect to external web APIs to power asynchronous browser updates
Ajax with jQuery	➔ Use the jQuery Javascript library to build Ajax requests and handle API responses ➔ Handle error responses with Ajax
Ajax with Fetch	➔ Use the new Fetch API to make asynchronous requests and handle the returned data

Supporting Lesson Content: Javascript Design Patterns

Lesson Title	Learning Outcomes
Changing Expectations	➔ React to changing product specifications and developer expectations ➔ Explore the Model-View-Controller design pattern ➔ Analyze an existing application for MVC structure
Refactoring With Separation Of Concerns	➔ Write code with discrete areas of responsibility in an MVC application ➔ Refactor an existing application to make use of modern code design practices

Using An Organizational Library	➔ Build a reactive front end application using an organizational library, knockout.js ➔ Implement knockout models and observable elements in an application
Learning A New Codebase	➔ Use proven strategies to adapt to a new and unfamiliar codebase

Supporting Lesson Content: Google Maps API

Lesson Title	Learning Outcomes
Getting Started with the APIs	➔ Set up your developer credentials and get started with the Google Maps APIs.
Understanding API Services	➔ Explore the location services available in the Google Maps APIs, including the Geocoding, Elevation, and Directions APIs.
Using the APIs in Practice	➔ Learn the practical details you need to know to use the Google Maps APIs in the real world.

Supporting Lesson Content: Front-end Frameworks

Lesson Title	Learning Outcomes
Features of Single Page Apps	➔ Learn about the features of a single page web application.
Examine a Framework's Source	➔ Dig around in the Backbone framework to discover how many of its most popular features work.
Angular	➔ Learn how to build a single page application in the Angular framework.
Ember	➔ Learn how to build a single page application in the Ember framework.