



NANODEGREE PROGRAM SYLLABUS

Front End Web Developer



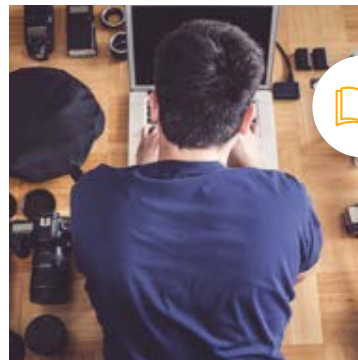
Overview

The goal of the Front End Web Developer Nanodegree program is to equip learners with the unique skills they need to build and develop a variety of websites and applications. A graduate of this Nanodegree program will be able to:

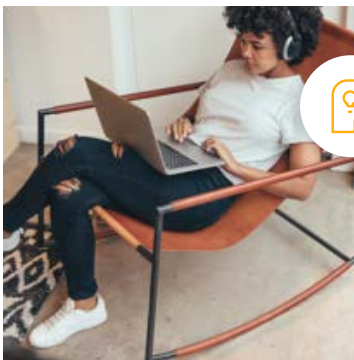
- Construct responsive websites using CSS, Flexbox and CSS Grid
- Develop interactive websites and UI (User Interface) applications using JavaScript and HTML
- Connect a web application to backend server data using JavaScript
- Automate application build and deployment using Webpack
- Improve offline performance of websites using Service Worker



Estimated Time:
4 Months at
5-10hrs/week



Prerequisites:
basic HTML, CSS,
and JavaScript



Flexible Learning:
Self-paced, so
you can learn on
the schedule that
works best for you



Need Help?
udacity.com/advisor
Discuss this program
with an enrollment
advisor.

Course 1: CSS & Website Layout

For your first project, you'll create a multi-page blog 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 for your own blog.

Course Project Stylized Blog Website

In this project you will code a fixed-wing aircraft, and then implement solutions to a significantly more challenging control problem.

LEARNING OUTCOMES

LESSON ONE

Introduction to HTML

- Create a programming project with a code editor
- Construct nested websites with HTML tags and elements
- Troubleshoot and debug HTML errors and bugs

LESSON TWO

Introduction to CSS

- Style website components by ID, class and type
- Connect CSS to a website
- Position and display website elements
- Modify and control website typography
- Troubleshoot and debug issues with stylesheets

LESSON THREE

Introducing 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

LESSON FOUR

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

LESSON FIVE

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

Course 2: JavaScript and the DOM

Use JavaScript to control a webpage. Learn what the Document Object Model (DOM) is, and use JavaScript and the DOM to dictate page content and interactions. Gain experience working with Browser Events and managing website performance by controlling content creation efficiently.

Course Project Dynamic Landing Page for Marketing Content

In this project, you will build a multi-section landing page. Often times, you won't know how much content will be added to a page through a Content Management System (CSM) or an API. To handle this problem, you will dynamically add content to a web page. You'll be building a landing page that combines your skills with JavaScript, HTML, and CSS to update and control the page and create a dynamic user experience.

LEARNING OUTCOMES

LESSON ONE

Syntax

- Declare block-scoped variables using let and const
- Format JavaScript strings using template literals
- Manage arrays and objects using JavaScript destructuring syntax
- Iterate over arrays and objects using JavaScript for...of syntax

LESSON TWO

The Document Object Model

- Describe and explain the Document Object Model for web browsers
- Access page elements by ID, class, and type using JavaScript

LESSON THREE

Creating Content with JavaScript

- Modify HTML content with JavaScript
- Create HTML content and elements with JavaScript
- Remove HTML content with JavaScript
- Style HTML content with JavaScript and CSS

LESSON FOUR

Working with Browser Events

- Describe and explain the phases of browser events
- Create event listeners that handle browser events by writing code that runs when an event is triggered
- Describe and explain the events that are fired as a web page loads

LEARNING OUTCOMES

LESSON FIVE

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



Course 3: Web APIs and Asynchronous Applications

In this project, you'll apply your new skills to combine data from the OpenWeatherMap API and client side (browser) HTML forms to create a web app that records a weather journal for users. This project requires you to create an asynchronous web app that uses Web API and user data to dynamically update the UI for a Weather Journal App

Course Project Weather Journal

In this project, you'll apply your new skills to combine data from the OpenWeatherMap API and client side (browser) HTML forms to create a web app that records a weather journal for users. This project requires you to create an asynchronous web app that uses Web API and user data to dynamically update the UI for a Weather Journal App.

LEARNING OUTCOMES

LESSON ONE

Node & Express Environment

- Set up a Node and Express environment to develop a web application on your local machine
- Install JavaScript packages using npm (Node Package Manager)
- Setup and run a local development server
- Manage web application file structure and website assets using Express

LESSON TWO

HTTP Requests & Routes

- Handle requests to an Express with routes
- Describe and explain the differences between GET and POST requests
- Build a web server and use it to serve data and responses to web requests

LESSON THREE

Asynchronous JavaScript

- Manage asynchronous JavaScript control flow with Promises
- Request data from a server using JavaScript Fetch
- Update and modify website elements dynamically using asynchronously retrieved data

Course 4: Build Tools, Webpack, and Service Worker

Develop an understanding of how to use build tools, such as Webpack, for automating build tasks. Create CSS variables with Sass and configure Webpack to use Sass controlled stylesheets. Learn how to cache server data and website functionality using Service Worker.

Course Project Article Analysis Website

In this project, you'll get a taste of some common production environments and tools that you will likely come across in a front end developer role. You will be building a web tool that allows users to run Natural Language Processing (NLP) on articles or blogs found on other websites. Using an exciting new API called Aylien, you can build a simple web interface to interact with their NLP system. This tool will give back pertinent information about the article: whether the content is subjective (opinion) or objective (fact-based) and whether it is positive, neutral, or negative in tone.

For this project, you will use:

- Node
- Express
- Aylien API
- Webpack
- Service Worker

Course Project : Front End Web Developer Nanodegree Program Capstone

In the final project, you'll combine all of the skills you've developed throughout the Nanodegree program to build an online travel app. You'll work with data sources from multiple APIs to create a dynamic travel weather planning application that helps people plan trips by generating weather forecasts for the places they're visiting.

You'll pull together all of the JavaScript, HTML, CSS, and build tool skills and knowledge you've gained to create this application. The design is up to you, and you'll have the flexibility to include and combine other APIs (even your own!) to build this final project.

LEARNING OUTCOMES

LESSON ONE

Intro to Build Tools

- Describe and explain the problems solved by using automated build tools
- Inspect and analyze the activity that occurs when loading a web page in a browser
- Describe and explain what Webpack is and how it can be used

LESSON TWO

Basics of Webpack

- Install Webpack to a computer
- Configure webpack for automating build tasks
- Define an entry point for a webpack bundle
- Install and configure middleware for building an application
- Extend Webpack functionality by installing plugins

LESSON THREE

Sass and Webpack

- Describe and explain the benefits and use cases for Sass
- Create CSS variables with Sass
- Extend and nest CSS sheets and classes with Sass
- Configure Webpack to use Sass controlled stylesheets

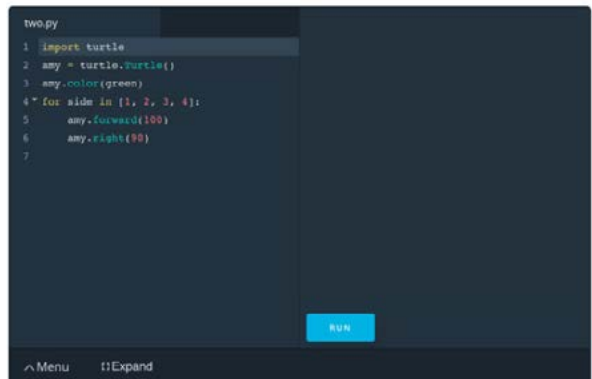
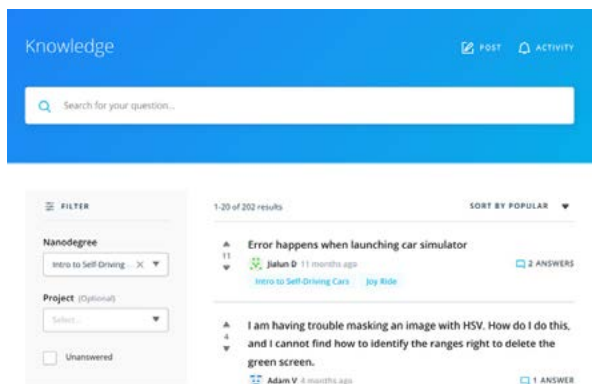
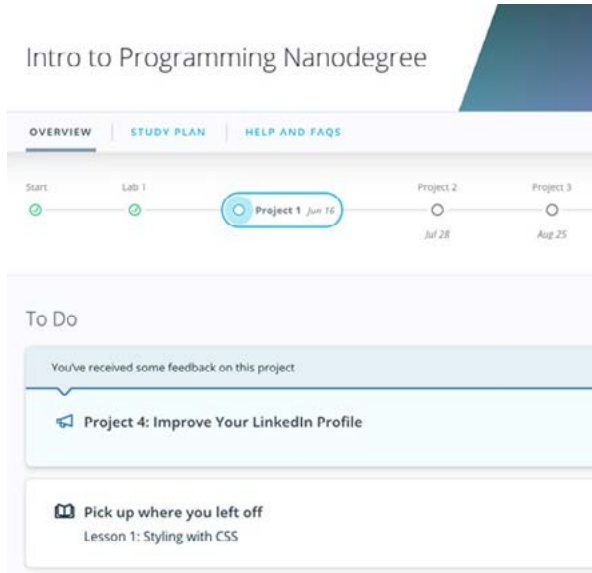
LESSON FOUR

Final Touches

- Control variable and function scope with JavaScript IIFEs (Immediately Invoked Function Expressions)
- Optimize an application build pipeline with Webpack
- Cache server data and websites functionality using Service Worker



Our Classroom Experience



REAL-WORLD PROJECTS

Build your skills through industry-relevant projects. Get personalized feedback from our network of 900+ project reviewers. Our simple interface makes it easy to submit your projects as often as you need and receive unlimited feedback on your work.

KNOWLEDGE

Find answers to your questions with Knowledge, our proprietary wiki. Search questions asked by other students, connect with technical mentors, and discover in real-time how to solve the challenges that you encounter.

WORKSPACES

See your code in action. Check the output and quality of your code by running them on workspaces that are a part of our classroom.

QUIZZES

Check your understanding of concepts learned in the program by answering simple and auto-graded quizzes. Easily go back to the lessons to brush up on concepts anytime you get an answer wrong.

CUSTOM STUDY PLANS

Create a custom study plan to suit your personal needs and use this plan to keep track of your progress toward your goal.

PROGRESS TRACKER

Stay on track to complete your Nanodegree program with useful milestone reminders.

Learn with the Best



Daniel Silber-Baker

INSTRUCTOR

Daniel Silber-Baker is a programmer, poet, and educational design expert. He has a master's degree from NYU's Interactive Telecommunications Program, and his professional career has stretched across the non-profit, corporate, and academic spheres.



Rachel Manning

INSTRUCTOR

Rachel is a front end web developer at Acquia and spent 3 years as the curriculum developer for a Silicon Beach bootcamp. An advocate for continued learning, she is passionate about mentoring women and students in technology.



Alyssa Hope

INSTRUCTOR

Alyssa is a full stack developer who was previously the lead instructor at a coding bootcamp. With a degree in International Communications, her passion is to express thoughts well, whether in code or writing.



Richard Kalehoff

INSTRUCTOR

Richard is a Course Developer with a passion for teaching. He has a degree in computer science, and first worked for a nonprofit doing everything from front end web development, to backend programming, to database and server management.

All Our Nanodegree Programs Include:



EXPERIENCED PROJECT REVIEWERS

REVIEWER SERVICES

- Personalized feedback & line by line code reviews
- 1600+ Reviewers with a 4.85/5 average rating
- 3 hour average project review turnaround time
- Unlimited submissions and feedback loops
- Practical tips and industry best practices
- Additional suggested resources to improve



TECHNICAL MENTOR SUPPORT

MENTORSHIP SERVICES

- Questions answered quickly by our team of technical mentors
- 1000+ Mentors with a 4.7/5 average rating
- Support for all your technical questions



PERSONAL CAREER SERVICES

CAREER SUPPORT

- Resume support
- Github portfolio review
- LinkedIn profile optimization



Frequently Asked Questions

PROGRAM OVERVIEW

WHY SHOULD I ENROLL?

Demand for front end web developers is widespread across every industry, and continues to rise. By mastering the valuable skills taught in this program, you will be prepared for roles at a wide array of companies — from startups to global organizations. The projects you'll build, and the portfolio you'll develop, will provide ample evidence of your expertise.

In the Front End Web Developer Nanodegree program, you will:

- Explore different JavaScript design patterns, and become skilled with common developer tools, testing suites, and frameworks.
- Demonstrate and refine your skills with five (5) in-class projects
- Receive expert reviews on your project submissions, along with actionable feedback to help you successfully advance through the program.
- Communicate with your mentor, who will provide guidance, lend support, answer questions, and direct you to valuable resources.

WHAT JOBS WILL THIS PROGRAM PREPARE ME FOR?

Graduates of this program will be valuable additions to any team working in the domain of web development, app development, software development, digital marketing, and e-commerce. Opportunities exist in companies ranging from Fortune 500 companies to startups.

Specific roles include:

- Front End Web Developer/Engineer
- UI/UX Developer
- Front End Designer

HOW DO I KNOW IF THIS PROGRAM IS RIGHT FOR ME?

If you're interested in building and developing a variety of websites and applications and creating stunning user experiences, Front End Web Developer is a great program for you!

Before beginning, a well-prepared student should be able to:

- Layout a simple webpage using HTML
- Style a website element using CSS
- Write and test software with JavaScript
- Inspect websites using Developer Tools on a modern web browser (Chrome, Firefox, or Edge)
- Debug and troubleshoot errors and failures in JavaScript programs

Whether you're looking to begin a new career as a front end web developer, strengthen your current skill set, or just want to further your knowledge, this is the perfect way to get started!



FAQs Continued

WHAT IS THE DIFFERENCE BETWEEN THE FRONT END WEB DEVELOPER PROGRAM AND FULL STACK WEB DEVELOPER PROGRAM?

Web development generally fits into distinct concentrations, such as front end web development and full stack web development.

As a front end web developer, you'll build responsive, dynamic user interfaces on the web. You'll leverage your HTML, CSS, and JavaScript skills to manage all client-side scripting.

As a full stack web developer, you'll have an active hand in implementing relationship databases, configure and deploy your applications to the cloud, and build dynamic software application backend systems using the Python programming language.

Whichever path you choose, you'll be building involved, engaging experiences on the web for your users!

ENROLLMENT AND ADMISSION

DO I NEED TO APPLY? WHAT ARE THE ADMISSION CRITERIA?

No. This Nanodegree program accepts all applicants regardless of experience and specific background.

WHAT ARE THE PREREQUISITES FOR ENROLLMENT?

A well-prepared learner is able to:

- Layout a simple webpage using HTMLs
- Style a website element using CSS
- Write and test software with JavaScript
- Inspect websites using Developer Tools on a modern web browser (Chrome, Firefox, or Edge)
- Debug and troubleshoot errors and failures in JavaScript programs
- Use Git for version control. If you do not have this experience, check out our [Version Control with Git course](#).
- Communicate fluently and professionally in written and spoken English.

IF I DO NOT MEET THE REQUIREMENTS TO ENROLL, WHAT SHOULD I DO?

We have a number of short free courses that can help you prepare, including:

- [Intro to Programming](#)
- [Version Control with Git](#)
- [Intro to HTML and CSS](#)



FAQs Continued

TUITION AND TERM OF PROGRAM

HOW IS THIS NANODEGREE PROGRAM STRUCTURED?

The Front End Web Developer Nanodegree program is comprised of content and curriculum to support 4 (four) projects. We estimate that students can complete the program in four (4) months working 5-10 hours per week.

Each project will be reviewed by the Udacity reviewer network. Feedback will be provided and if you do not pass the project, you will be asked to resubmit the project until it passes.

HOW LONG IS THIS NANODEGREE PROGRAM?

Access to this Nanodegree program runs for the length of time specified in the payment card above. If you do not graduate within that time period, you will continue learning with month to month payments. See the [Terms of Use](#) and [FAQs](#) for other policies regarding the terms of access to our Nanodegree programs.

SOFTWARE AND HARDWARE

WHAT SOFTWARE AND VERSIONS WILL I NEED IN THIS PROGRAM?

For this Nanodegree program, you will need access to a computer with a broadband connection, on which you will install a professional code/text editor (e.g., Visual Studio Code, Atom, etc.).

