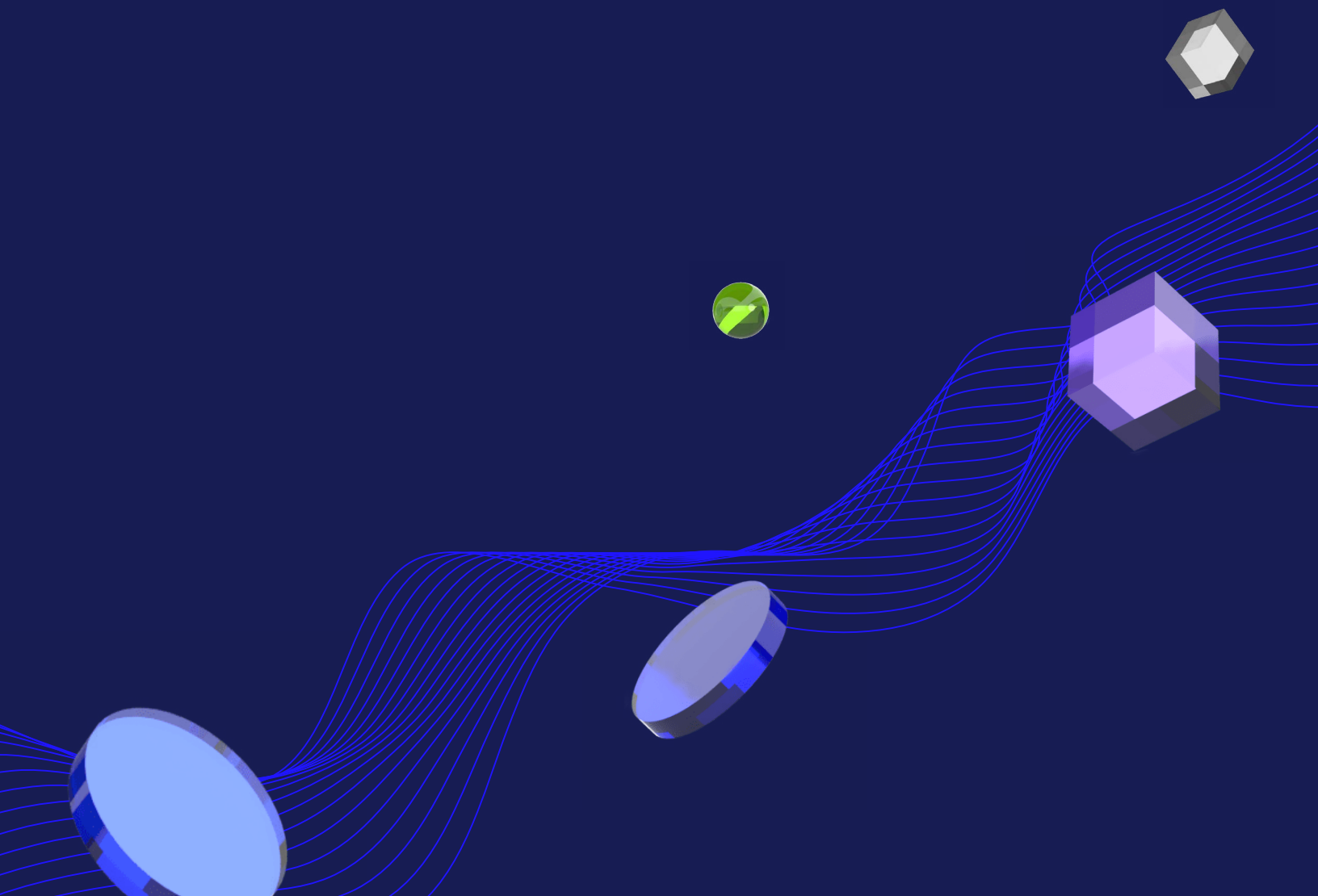




SCHOOL OF PROGRAMMING & DEVELOPMENT

Intermediate JavaScript

Nanodegree Program Syllabus



Overview

The goal of the Intermediate JavaScript Nanodegree program is to prepare learners for roles in web development, server-side application development, and desktop development that require a more advanced set of JavaScript skills. This program will also prepare learners with the skills required to use JavaScript frameworks like React, Angular, and Vue.



Learning Objectives

A graduate of this program will be able to:

- Use object oriented JavaScript to build classes to construct objects that encapsulate data and functionality.
- Create private state with closures and immediately-invoked function expressions.
- Parse, organize, and transform data in JavaScript using common methods for collections, arrays, and objects.
- Use the functional programming paradigm with immutable objects, pure functions, and common functional methods.
- Design functional programming pipelines using higher order wrapping functions and currying.
- Implement asynchronous programming with JavaScript including using named functions, handler functions, and JavaScript Promises.

Program information



Estimated Time

3 months at 10hrs/week*



Skill Level

Intermediate



Prerequisites

A well-prepared learner will be able to explain and utilize JavaScript's primitive types, write conditions and loops, recognize object syntax and structure, declare functions, and navigate and use Chrome DevTools.



Required Hardware/Software

Learners need access to the internet and a 64-bit computer.

*The length of this program is an estimation of total hours the average student may take to complete all required coursework, including lecture and project time. If you spend about 5-10 hours per week working through the program, you should finish within the time provided. Actual hours may vary.

Object Oriented JavaScript

Learn how to use the object oriented programming features of JavaScript to build applications with reusable and maintainable blocks of code. Learners will build classes to construct objects that include both data and functionality. They will also learn how to use prototypal inheritance to maintain DRY code and keep data safe and secure with private properties.



Course Project

Create a User-Generated Infographic

In this project, learners will be presented with a real-world scenario of working with object oriented JavaScript. Learners will build an infographic creating and modifying objects from provided data and user input data. Learners will also pull in information from a form and use it to complete an array of objects that will be appended back to the DOM.

Lesson 1

Objects in Depth

- Create objects and add/remove properties to them.
- Write methods to access and mutate an object's properties.
- Learn how variables are properties on window.

Lesson 2

Functions at Runtime

- Leverage functions as first-class functions.
- Identify when a scope is created.
- Utilize closures and IIFEs to build private state.

Lesson 3

Classes & Objects

- Write a constructor function to instantiate objects.
- Identify and manually set the value.

Lesson 4

Object Oriented Design Patterns

- Use Mixins to copy properties from one object to another object.
- Create objects with private properties using module pattern and revealing module pattern.

Course 2

Functional Programming

Functional programming is a hot new topic in the world of JavaScript, but in this course learners will find out that functional programming has been around for a long time! This course will help learners gain a better understanding of programming paradigms and why the functional paradigm is trending right now. Learners will also discover and practice the modern JavaScript syntax that can benefit their JavaScript programs.



Course Project

Create a Mars Rover Dashboard

The functional JavaScript course project will be to build a dashboard for the past and current NASA Mars rovers, including information about each mission and images from their excursions. Learners will be tasked with using all the ES6 syntax and functional programming concepts from the course to interact with a real NASA API and build the front end logic to create the dashboard.

Lesson 1

Course Introduction: Functional Programming

- Get an introduction to the course and concepts.
 - Learn requirements, prerequisites, and resources.
 - Get introduced to functional programming.
 - Explore a brief history of functional programming.
-

Lesson 2

Introduction to Functional Programming

- Get introduced to programming paradigms.
 - Learn the benefits of understanding programming paradigms.
 - Compare paradigms.
 - Learn the benefits of functional programming.
-

Lesson 3

Functional JS Syntax

- Explore ES6 array methods.
 - Explore ES6 variables.
 - Learn object methods.
-

Lesson 4

Functional Programming in JS

- Write functional programs in JavaScript.
 - Learn about higher order functions.
 - Learn about functional data manipulation.
 - Understand functional DOM manipulations.
-

Lesson 5

Going Further with Functional JS

- Understand persistent data structures.
- Learn about Immutable.js (vanilla JS).
- Learn about RxJS (primarily AngularJS).
- Learn about Redux (primarily ReactJS).

Asynchronous Programming in JavaScript

The focus of this course is to get learners comfortable reading, writing, and thinking in asynchronous patterns. Understanding asynchronous programming is not only a vital JavaScript skill to master, but is also an important step in a learner's progress as a web developer. This course will cover the original and modern ways to handle asynchronous events in JavaScript, from callbacks, to promises, and `async/await`. Alongside these topics will come important conceptual lessons, real life applications, and lots of hands-on practice.



Course Project

Build a UdaciRacer Simulation Game

In this project, learners will complete a partially built single player racing game. Using the skills and concepts they learned in the course, learners will combine callbacks, promises, and `async/await` syntax to complete TODOs marked in the code. The TODOs will range from creating API requests to completing asynchronous logic flows. Learners will also be able to create a theme for their race, like Nascar, horse racing, or spaceship racing! By the end, learners will be able to play through a racing game of their very own creation.

Lesson 1

Stack vs. Heap

- Get introduced to the course and concepts.
- Learn requirements, prerequisites, and resources.
- Get introduced to asynchronous thinking.
- Learn the benefits of asynchronous programming.

Lesson 2

Pointers & References

- Learn about threads and single-threaded programming.
 - Understand blocking and non-blocking code.
 - Learn helpful terms for single-threaded and multi-threaded programming.
 - Learn callbacks for asynchronous functionality.
-

Lesson 3

New, Delete, Memset & Mallo

- Learn basic syntax and promise chaining.
 - Learn how promises relieved the pain points of callbacks.
 - Learn implementations of promises with Fetch.
 - Learn how to handle errors with best practices for promises.
 - Learn advanced promise methods.
-

Lesson 4

Smart Pointers

- Learn synchronous try/catch syntax.
- Learn basic syntax of async/await.
- Explore async/await with promise chains.
- Learn error handling with async/await.
- Learn when and when not to use async/await.

Meet your instructors.



Alyssa Hope

Software Developer at Seamgen

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.



Rachel Manning

Front End Developer at Acquia

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.



Andrew Wong

Developer Advocate at Adyen

Andrew is a course developer who enjoys making the world a better place through code. He first discovered his passion for teaching as an instructor at App Academy, and continues to enjoy empowering students to advance their education.

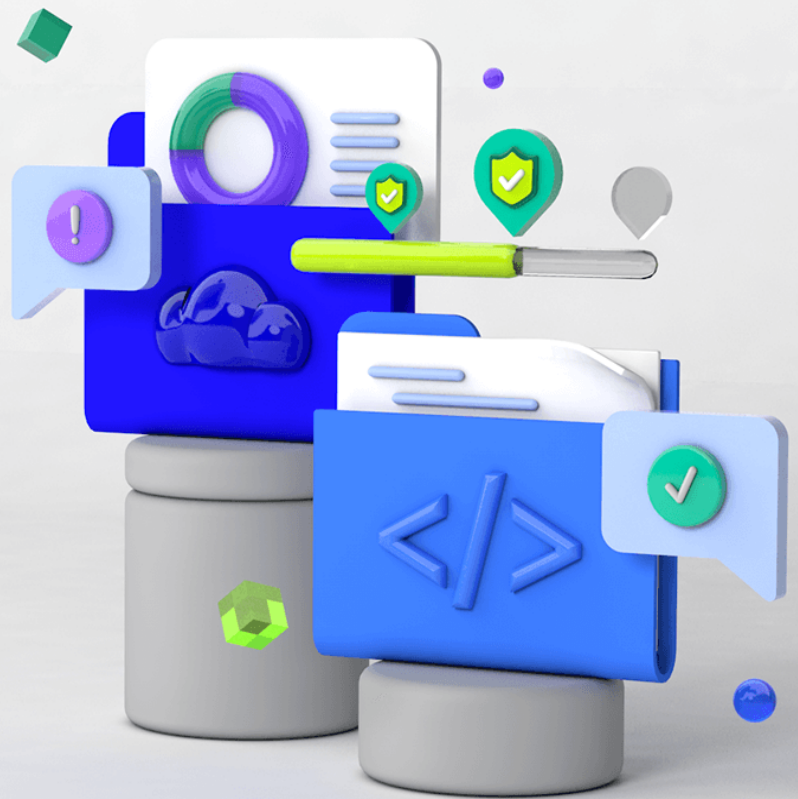


Richard Kalehoff

Software Engineer at Udacity

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.

Udacity's learning experience



Hands-on Projects

Open-ended, experiential projects are designed to reflect actual workplace challenges. They aren't just multiple choice questions or step-by-step guides, but instead require critical thinking.



Knowledge

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



Workspaces

See your code in action. Check the output and quality of your code by running it on interactive workspaces that are integrated into the platform.



Quizzes

Auto-graded quizzes strengthen comprehension. Learners can return to lessons at any time during the course to refresh concepts.



Custom Study Plans

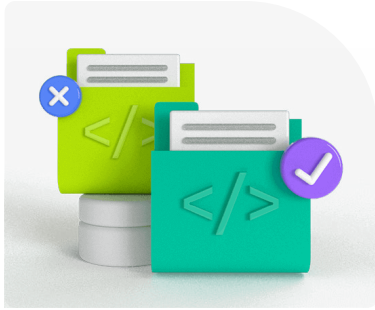
Create a personalized study plan that fits your individual needs. Utilize this plan to keep track of movement toward your overall goal.



Progress Tracker

Take advantage of milestone reminders to stay on schedule and complete your program.

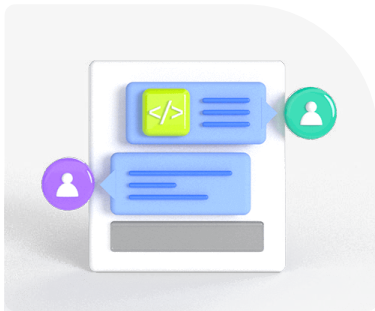
Our proven approach for building job-ready digital skills.



Experienced Project Reviewers

Verify skills mastery.

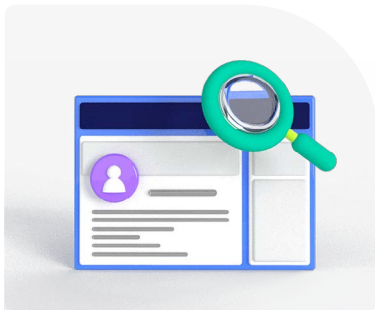
- Personalized project feedback and critique includes line-by-line code review from skilled practitioners with an average turnaround time of 1.1 hours.
- Project review cycle creates a feedback loop with multiple opportunities for improvement—until the concept is mastered.
- Project reviewers leverage industry best practices and provide pro tips.



Technical Mentor Support

24/7 support unblocks learning.

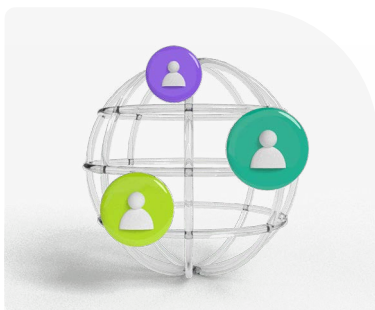
- Learning accelerates as skilled mentors identify areas of achievement and potential for growth.
- Unlimited access to mentors means help arrives when it's needed most.
- 2 hr or less average question response time assures that skills development stays on track.



Personal Career Services

Empower job-readiness.

- Access to a Github portfolio review that can give you an edge by highlighting your strengths, and demonstrating your value to employers.*
- Get help optimizing your LinkedIn and establishing your personal brand so your profile ranks higher in searches by recruiters and hiring managers.



Mentor Network

Highly vetted for effectiveness.

- Mentors must complete a 5-step hiring process to join Udacity's selective network.
- After passing an objective and situational assessment, mentors must demonstrate communication and behavioral fit for a mentorship role.
- Mentors work across more than 30 different industries and often complete a Nanodegree program themselves.

*Applies to select Nanodegree programs only.



Learn more at

www.udacity.com/online-learning-for-individuals →