



NANODEGREE PROGRAM SYLLABUS

# Intermediate JavaScript



# Overview

The goal of the Intermediate JavaScript Nanodegree program is to prepare students 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 students with the skills required to use JavaScript frameworks like React, Angular, and Vue.

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.

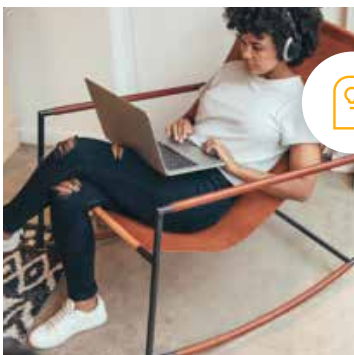
**Prerequisites:** A well-prepared student will be able to Explain and utilize JavaScript's primitive types, write conditions and loops, recognize object syntax and structure, declare functions, and navigate and use Chrome DevTools.



**Estimated Time:**  
3 Months at  
10 hours / week



**Prerequisites:**  
Basic JavaScript  
experience



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



**Need Help?**  
[udacity.com/advisor](https://www.udacity.com/advisor)  
Discuss this program  
with an enrollment  
advisor.

# Course 1: Object Oriented JavaScript

In this course, you'll learn how to use the object oriented programming features of JavaScript to build applications with reusable and maintainable blocks of code. You'll build classes to construct objects that include both data and functionality, learn how to use prototypal inheritance to maintain DRY code, and how to keep data safe and secure with private properties.

## Course Project

Create a User-generated Infographic

In this project, students will be presented with a real-world scenario of working with object-oriented JavaScript. Students will build an infographic creating and modifying objects from provided data and user input data. Students 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.

## LEARNING OUTCOMES

### LESSON ONE

#### 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 TWO

#### Functions at Runtime

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

### LESSON THREE

#### Classes and Objects

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

### LESSON FOUR

#### 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 you'll learn that the idea of Functional Programming has been around for a long time! This course will help you gain a better understanding of programming paradigms and why there is a trend towards the Functional paradigm right now. Beyond a high level understanding of Functional programming, you will also discover and practice the modern JavaScript syntax that will allow you to benefit from these concepts in your 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, information about each mission and images from their excursions. You 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.

## LEARNING OUTCOMES

### LESSON ONE

#### Course Introduction - Functional Programming

- Introduction to the course and concepts
- Requirements, pre-requisites, and resources
- Introduction to Functional Programming
- A brief history of Functional Programming

### LESSON TWO

#### Introduction to Functional Programming

- Introduction to Programming Paradigms
- Benefits of understanding programming paradigms
- Comparing Paradigms
- Benefits of Functional Programming

### LESSON THREE

#### Functional JS Syntax

- ES6 Array methods
- ES6 Variables
- Object Methods

### LESSON FOUR

#### Functional Programming in JS

- Writing Functional Programs in JavaScript
- Higher Order Functions
- Functional data manipulation
- Functional DOM manipulations

### LESSON FIVE

#### Going Further with Functional JS

- Persistent Data Structures
- Immutable.JS (vanilla JS)
- RxJs (primarily AngularJS)
- Redux (primarily ReactJS)

# Course 3: Asynchronous Programming in JavaScript

The focus of this course is to get you 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 your 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, you will complete a partially built single player racing game. Using the skills and concepts you learned in the course, you 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. You will also be able to create a theme for your race, like Nascar, horse racing, or spaceship racing! By the end, you will be able to play through a racing game of your very own creation.

## LEARNING OUTCOMES

### LESSON ONE

#### Stack vs. Heap

- Introduction to the course and concepts
- Requirements, pre-requisites, and resources
- Introduction to asynchronous thinking
- Benefits of learning asynchronous programming

### LESSON TWO

#### Pointers and References

- Threads and single threaded programming
- Blocking and Non-Blocking code
- Helpful terms for single and multi threaded programming
- Callbacks for asynchronous functionality

### LESSON THREE

#### New, Delete, Memset and Mallo

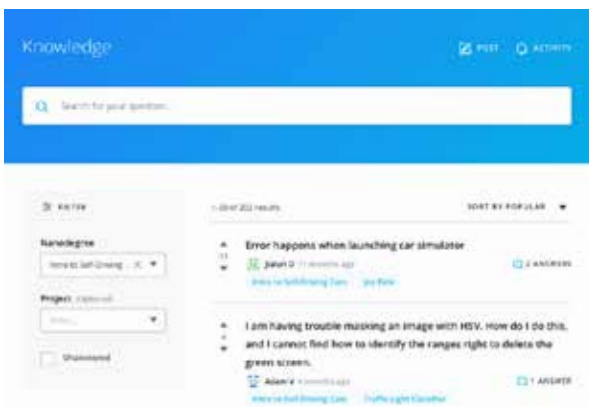
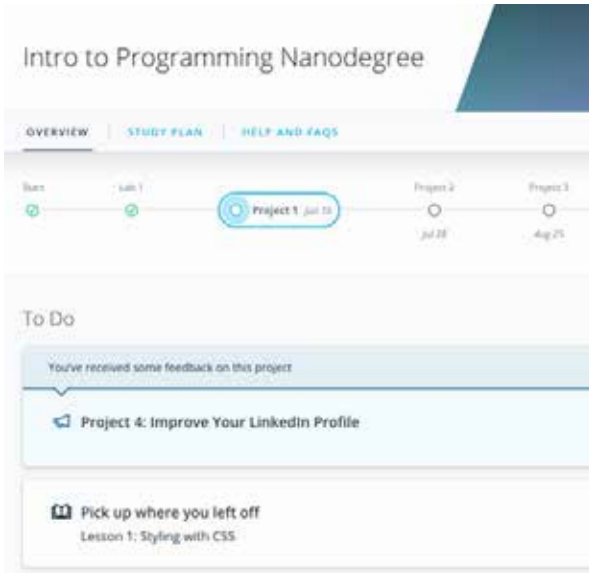
- Basic syntax and Promise chaining
- How Promises relieved the pain points of callbacks
- Implementations of Promises with Fetch
- Error handling and best practices for Promises
- Advanced Promise methods

### LESSON FOUR

#### Smart Pointers

- Synchronous Try/catch syntax
- Basic syntax of Async/await
- Async/await with Promise chains
- Error handling with Async/await
- When and When not to use Async/await

# 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.

## STUDENT HUB

Leverage the power of community through a simple, yet powerful chat interface built within the classroom. Use Student Hub to connect with fellow students in your program as you support and learn from each other.

## 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

Preschedule your study times and save them to your personal calendar to create a custom study plan. Program regular reminders to keep track of your progress toward your goals and completion of your program.

## PROGRESS TRACKER

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

## Learn with the Best



### 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.

# 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?

Every web application and page utilizes JavaScript, making JavaScript the most popular programming language in the world. Utilized in mobile applications, websites, web servers, website applications, server applications, games, server applications, and much more. In fact, JavaScript is the foundation for popular apps like Netflix, Uber, Facebook, and LinkedIn! Most software engineers and developers need to know JavaScript, roles which command salaries averaging \$150k.

### WHAT JOBS WILL THIS PROGRAM PREPARE ME FOR?

The addition of JavaScript skills to your developer toolkit is an excellent move for any developer seeking a critical career advantage.

The goal of the Intermediate JavaScript Nanodegree program is to prepare students 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 students with the skills required to use JavaScript frameworks like React, Angular, and Vue.

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

If you are interested in building out the infrastructure that powers and supports the many web, desktop, mobile, and integrated applications in the business world, this program is a great fit for you.

It is designed for people with an basic background in JavaScript programming who are looking to build a strong foundation in JavaScript to either advance within their current field or position themselves to learn more advanced skills for a career transition.

## ENROLLMENT AND ADMISSION

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

There is no application. This Nanodegree program accepts everyone, regardless of experience and specific background.

### WHAT ARE THE PREREQUISITES FOR ENROLLMENT?

A well-prepared student should be able to:

- Explain and utilize JavaScript's primitive types
- Write conditions and loops
- Recognize object syntax and structure
- Declare functions
- Navigate and use Chrome DevTools



## FAQs Continued

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

Check out the [Front End Web Developer Nanodegree program](#).

### **TUITION AND TERM OF PROGRAM**

#### **HOW IS THIS NANODEGREE PROGRAM STRUCTURED?**

The Intermediate JavaScript Nanodegree program is comprised of content and curriculum to support 3 (three) projects. We estimate that students can complete the program in three (3) months, working 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 on the Nanodegree program overview page. If you do not graduate within that time period, you will continue learning with month to month payments. See the [Terms of Use](#) for other policies around the terms of access to our Nanodegree programs.

### **SOFTWARE AND HARDWARE**

#### **WHAT SOFTWARE AND VERSIONS WILL I NEED IN THIS PROGRAM?**

There are no specific hardware or software requirements for this program, other than access to the internet and a 64-bit computer.

