**Course syllabus**

This course is the second of a series that aims to help you learn more about web development.

In this course, you will explore the following:

**Module 1: Introduction to JavaScript**

In this module, you are introduced to JavaScript. You'll learn why JavaScript is so integral to software development. And you'll get an overview of how to write JavaScript code inside the browser. Furthermore, you will learn about the most common operators as well as conditional statements and loops.

After completing this module, you will be able to:

* Explain the importance of JavaScript in software development
* Demonstrate how to write JavaScript code inside the browser
* Demonstrate how to write basic JavaScript code
* List common operators, conditional statements and loops
* Demonstrate how to use variables and output their value in the console

**Module 2: The building blocks of a program**

Here you'll learn how to use objects, arrays and functions. In addition, you will learn about the most common built-in methods, and the difference between undefined, null and empty strings. And you'll explore both error handling and defensive programming.

After completing this module, you will be able to:

* Build and use objects, arrays, and functions
* List some common built-in methods on built-in objects
* Describe handling bugs and errors using try, catch, throw, and defensive programming
* Explain the difference between undefined, null, and empty strings
* Demonstrate how to write basic code using arrays, objects and functions

**Module 3: Programming paradigms**

This module is about functional programming and the object oriented programming paradigm. You will learn what scope is in JavaScript. You'll explore the differences between var, let and const. And you'll learn how to use classes and inheritance in object oriented programming. Additionally, you'll explore how to use write JavaScript using modern features like spread and rest. You will build code that can manipulate the DOM and handle events. And you will use JSON in JavaScript.

After completing this module, you will be able to:

* Outline the tenets of the functional programming and object oriented programming paradigm
* Describe how scope works in JavaScript
* List the differences between var, let, and const
* Use classes and inheritance in OOP in JavaScript
* Write JavaScript code using more modern features like spread, rest, template strings and modules
* Build code that manipulates the DOM and handles events
* Use JSON in JavaScript

**Module 4: Testing and compatibility**

Here you will learn about Node.js and npm. And you will explore how to install npm packages and how to work with package.json. Furthermore, you will learn about testing in JavaScript and you'll code a simple unit test in Jest.

After completing this module, you will be able to:

* Describe Node.js and npm
* Explain how to install npm packages
* Describe how to work with package.json
* Explain the process of testing in JavaScript
* List the three most prevalent kinds of testing
* Demonstrate how to code a simple unit test in Jest

**Module 5: Graded assessment**

In the final module, you'll learn about the graded assessment. After you complete the individual units in this module, you'll synthesize the skills you gained from the course to create code for the "Little lemon receipt maker ".

You'll also have to opportunity to reflect on the course content and the learning path that lies ahead.

# How to Position Yourself for a New Career

You are well on your way to becoming a software developer.

You took the most important first step: you started.

While this specialization on Coursera will make you into a well-rounded junior developer, you are basically just getting started.

Here are some proven tips to make the transition to your new career as smooth as possible.

## **Be persistent**

Succeeding in your career efforts is not easy. Luckily, it's not too hard either. Consider this new endeavor of learning to code a part of your everyday life.

Make it as much of a routine as possible. Hopefully, it will work like this:

* You wake up,
* You brush your teeth,
* You run some errands,
* And then you write and learn to code.

Obviously, there are things like your school obligations, or your day job, or other places you need to be and things you need to do.

However, if you don't code regularly - preferably on a daily basis - your progress will be slower. Try to set aside some time to consistently code and learn every day. Persistence is key.

## **Start building simple apps today**

Don't wait until you "learn enough". There's always more to learn, and it's best to get started with any kind of a simple project right now.

Even just taking the code from this specialization and tinkering with it will do wonders for your confidence and the speed at which you acquire new knowledge.

Also, the more you practice, the better you'll retain what you've learned.

Having your own projects that you can showcase to others - no matter how small or straightforward shows a track record and dedication. This is something that your future employer might be impressed with, so start today.

## **Set up a GitHub account**

Since we're on the topic of personal projects, head on over to [GitHub](https://github.com/) and set up your developer profile right away. It's essential to have an account there since you can keep all your projects in a single location that you can access from any computer.

You can almost think of your GitHub account as an additional brain power. No matter how long ago, whatever you've worked on will remain there, waiting for you to peek into and re-familiarise yourself with.

## **Pair program**

Try to find someone at your level or perhaps slightly more knowledgeable than you and ask them to set up a recurring pair programming session.

This works nicely because having a pair programming partner can speed up your learning. You also have someone to be accountable to.

## **Start a coding blog**

Technical communication is important for developers, and just like anything else, you get better with practice.

Starting a coding blog will work the same as having a GitHub account, with some the additional benefits:

* It shows even more dedication - and this increases your chances of getting hired
* It helps you experiment with different technologies
* Setting up your own website is practical learning in its own right and one more project to add to your CV

## **Collaborate on open source projects**

Even if you are just starting out and are really struggling to get into this field, you can still be a valuable contributor to open-source projects.

There are so many open-source projects that are in demand for all kinds of contributors.

Even contributing to a project by fixing some typos in documentation files is a nice start to getting more involved and putting yourself out there.

## **Get a certificate**

Getting certified is always a good thing. The fact that you're reading this lesson right now confirms that you're on your way to receiving a certificate of completion from Coursera!

## **Keep a positive attitude**

As with anything worth doing, you might sometimes get tired, not understanding how something works, and perhaps even feel like giving up.

Remember to stay consistent.

There are always ups and downs in life, but sometimes it's worth it to think of all the things you've achieved so far and use that as motivation to keep at it.

## **Never stop learning**

There's always more to learn in IT, and that's probably the best thing about it. It's the very thing that makes it fun and provides an opportunity for each developer to get ahead in their career.

# How to uncover job opportunities

Learning how to program in JavaScript helps you prepare for a wide range of job opportunities. This is in part because it expands the possibilities of what you can build as a developer.

JavaScript is one of the most in-demand programming languages, as it is used in nearly all active websites. Its versatility enables developers to use popular libraries, plugins, and frameworks such as React, which improves efficiency and productivity.

With all flexibility that programming in JavaScript brings, it is no surprise that there are different careers that you might want to follow. However, no matter what career path you choose, you will always want to learn other technologies like HTML, CSS, React, Node.js, or Python, so you are more marketable.

Let’s cover a few of the most common roles you can get if you know how to program in JavaScript.

**Mobile Developer** There has been a constant increase in the demand for mobile developers as mobile devices’ use for accessing the internet has been on the rise. Mobile developers specialize in building apps for platforms like Google’s Android and Apple’s iOS. Many developers choose to use React Native, which allows them to build one application using JavaScript that works on both Android and iOS devices. Mobile developers work with UX and UI designers and use React’s UI capabilities to implement functionalities that customers will use. Mobile developers also make sure that the front-end and the back-end of the applications work seamlessly. Other skills that mobile developers may have include HTML, CSS, Java, Kotlin, Objective-C, C++, C#, among others.

**Front-End Developer** As the name implies, front-end developers build the user-facing parts of websites and apps. They work closely with designers to implement visual and interactive elements through coding, using HTML, CSS, and of course, JavaScript. They also use libraries and frameworks like React to save time and make their work more efficient. Front-end developers may also be responsible for making sure the final user has a good user experience and that websites and apps behave as expected and are free of errors and bugs.

**Back-End Developer** Back-end developers work on the back-end of websites and applications. They can use JavaScript with Node.js to develop back-end functionalities. Among these functionalities are streaming and chat-based applications, as well as JSON APIs and serverless functions. Back-end developers possess additional skills, including professional working knowledge of Python, APIs, cloud infrastructure, and database.

**Full-Stack Developer** As you might guess, a full-stack developer works with both the front- and back-end of building websites and apps. So, these professionals combine the skills in these two areas. They can use frameworks like React to work on the front-end and Node.js to work on the back-end. They also apply skills in addition to JavaScript to build websites and apps.

**Your professional journey ahead**

As you embark on the exciting career as a professional developer, you will realize that you will progressively expand your skills to include a wide range of technologies and programming languages besides JavaScript.

If you want to have an idea of the opportunities available in these areas, you might want to check your favorite job search website or app and look for jobs related to JavaScript. You will learn that there is no shortage of opportunities. And as you start reading more about these roles, you will find that JavaScript is often only one of the competencies that employers are looking for.

But don’t worry, as you advance in your studies and in your career, you will further develop your skills to focus on the professional path you want to follow.

Good luck on your professional journey!