About this course

Let's go over what this course is about, what you'll learn, and the tips you can make use of.

We'll cover the following

What will you learn in this course?

In this course, you’ll learn to write modern JavaScript from the ground up. But I’m not going to throw a bunch of syntax at you. I want you to learn to think in terms of contemporary JavaScript. You’ll see every piece of syntax with a recommendation for not just how to use it, but when to use it.

Also, I’m not going to run through every minor syntax change. I know your time is valuable, so I’ll only show you syntax that has high impact, and that’s something you’ll use over and over again. You can deal with edge cases when they arise. This course will give you the best features and nothing else.

Throughout the course, you’ll see why the syntax changes aren’t random—they follow a simple set of principles designed to make the language easier to read and write. Modern JavaScript is *simple, predictable, readable, and flexible*. And every new piece of syntax should be evaluated in those terms, including the syntax that isn’t yet part of the spec.

And because modern JavaScript has as many paradigm changes as there are syntax changes, you’ll spend some time reviewing older JavaScript concepts to see how you can use them in a modern context. This is important because JavaScript was previously written primarily using libraries (*jQuery* above all else), and it was easy to plug in just enough code to get something working without understanding the underlying concepts. As more code projects incorporate JavaScript as part of major features, and not just a handful of plugins, you’ll need to understand concepts you may have glossed over before.

For example, you might have safely ignored syntax changes in ES5. You could use *array* methods such as `map()` and `reduce()` for several years, but you might have ignored them because browsers weren't fully compatible and because transpilers weren't yet mature. Array methods are now essential parts of good code. It's worth taking a step back to review them.

Still, I know you have plenty to do every day—that's why this course follows a tips format that allows you to jump in and out at will. You don't need to read this straight through. I'll reference syntax from previous and future tips so that you can jump around as you need to. Keep a copy at your desk and skim it during a break or load it up on your phone so you can get a quick tip-in while waiting at the dentist's office.

When learning new syntax, the most important thing you can do is read code. And the best code is real code. You won't see **foo-bar** examples or lots of *math expressions* in this course. Instead, you'll work with *strings, objects, currency, email addresses*, and the like. In other words, the code samples you see will be close to life. This adds a little complexity to the examples, but it will ultimately make it much easier for you to internalize the ideas so that you can bring them into your code quickly and easily.

Finally, every sample is available in a repo for this course. The code has nearly 100 percent test coverage, so jump in and try ideas out. To get things working, you'll need at least Node.js version 8.5 installed. Be sure to use a modern browser such as Chrome, Edge, or Internet Explorer 11+.

> The setup required to run the codes in this course has already been provided to you on the platform.

Next up, you'll see who the intended audience of this course is.