

The Future of JavaScript

JavaScript added a whole slew of functionality to the language in 2015. These are tools intended to make the language easier, more powerful, and more robust. We'll go over the most useful of these tools and see how they can truly transform the way we write JavaScript.

ES 2015

Changes

JavaScript added a whole slew of functionality to the language in 2015. These are tools intended to make the language easier, more powerful, and more robust. Code that we've been writing up until now can be written faster and cleaner. Problems such as variable hoisting and callback hell are solved.

Job interviews will also feature questions based on ES5 JavaScript. You'll be asked questions about scope and hoisting that you wouldn't be aware of if you'd started by using the new tools we're about to go over.

These new features won't change your world, but they'll make code a little less painful in general. There are little things, like trailing commas in function parameters. Whereas before `function(a, b,) {}` would throw a syntax error, it's now allowed.

There are also big things that will change every piece of code you write, like new ways to declare variables, which we've already discussed.

Nothing in ES2015+ is vital to building websites. For years, people have built sites with just what we've already covered so far. You could stop right here and skip the rest of the course and you'd have everything needed to build websites. This just makes it all a little easier.

Frameworks

While ES2015 isn't necessary for web development, if you'd like to use a modern framework, it's essentially a requirement. React.js and Angular both

rely heavily on ES2015 features and without them, you'll have trouble finding online help such as documentation and answers to questions.

Browser Implementations

Many browsers are slow to implement new JavaScript features. In terms of ES2015, virtually every modern browser has implemented everything. For ES2016 & newer, Chrome, Firefox, and Safari will adopt new functionality at different times.

This problem has given rise to *transpilers*. A transpiler will process bleeding-edge code and transform/compile it down into ES5 code so that it works on any browser, even older versions of Internet Explorer. Several websites are now written with cutting-edge features and the code is then transpiled down into a version compatible with all browsers. The most popular transpiler is called [Babel](#).

If not using a transpiler, it's a good idea to check browser specs to see what has and hasn't been implemented, or else your code may not work.

The Spread

The essential parts of ES2015 are included in this section. There are more lessons involving ES2015 featured interweaved into the next few sections, as well as an entirely separate section for the more advanced ES2015+ topics.

Let's get started in the next lesson.