**Modern JavaScript**

This lesson discusses the use of modern JavaScript in this course.

**WE'LL COVER THE FOLLOWING**

* + [About Modern JavaScript](https://www.educative.io/courses/simplifying-javascript-handy-guide/N8XV31KZxJp#about-modern-javascript)

**About Modern JavaScript**[#](https://www.educative.io/courses/simplifying-javascript-handy-guide/N8XV31KZxJp#about-modern-javascript)

Ever get the feeling you’ve woken up and everything’s different? If you’ve looked at modern JavaScript, you may feel like you woke up in a new world where nothing’s the same.

Yesterday, you were tweaking a simple jQuery accordion. Today, you look at some JavaScript code and the ubiquitous jQuery $ operator is gone. In its place are lots of strange dots (...) and strange symbols (=>) (the **spread** operator and **arrow** functions, as you’ll soon learn). Sure, some things look familiar, such as class, but even that seems to be unpredictable (where are the private methods?).

Are you perplexed? Maybe a little excited? Well, I have good news for you: Everything has changed for the better.

When the **ECMAScript 6** spec was released—**ES6** for short—JavaScript code changed dramatically. The changes were so substantial that after reading this course you’ll be able to tell at a glance which code was written with pre-ES6 syntax and which code was written with post-ES6 syntax. JavaScript now is different. Modern JavaScript is any code composed with post-ES6 syntax.

Modern JavaScript is a pleasure to write. I’m a long-time unrepentant Java- Script fan, but I’ll admit that I found the old syntax a little clunky at times, and it was pretty hard to defend. Modern JavaScript is better. But it’s changed significantly, and catching up can be hard.

Next up, let’s see what you’ll learn in this course and how it’ll be useful.

**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?](https://www.educative.io/courses/simplifying-javascript-handy-guide/q2ygQoymRw0#what-will-you-learn-in-this-course)

**What will you learn in this course?**[#](https://www.educative.io/courses/simplifying-javascript-handy-guide/q2ygQoymRw0#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](https://github.com/jsmapr1/simplifying-js) 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.

**Intended Audience & Other Sources**

Let's look at the intended audience and resources for using this course.

**WE'LL COVER THE FOLLOWING**

* + [Who is this course for?](https://www.educative.io/courses/simplifying-javascript-handy-guide/RLRQwjknVYY#who-is-this-course-for)
  + [Online Resources](https://www.educative.io/courses/simplifying-javascript-handy-guide/RLRQwjknVYY#online-resources)

**Who is this course for?**[#](https://www.educative.io/courses/simplifying-javascript-handy-guide/RLRQwjknVYY#who-is-this-course-for)

This course is for anyone who has a little programming experience. It helps to have some JavaScript experience, but that’s not strictly necessary. I’ll assume that you know some basic programming concepts, such as *loops, conditionals, functions, and classes*. In short, if you’ve seen modern JavaScript and you’re excited to learn more, then this course is for you.

You may feel like you woke up in a strange new world. That’s not a bad thing. In fact, this new world is awesome—the coffee’s brewing, and the aroma of fresh-baked danishes makes this new world a place you want to wake up to. JavaScript is better than ever—it’s time to enjoy it.

**Online Resources**[#](https://www.educative.io/courses/simplifying-javascript-handy-guide/RLRQwjknVYY#online-resources)

You can also find the code on [github](https://github.com/jsmapr1/simplifying-js" \t "_blank) or on the book’s [Pragmatic Bookshelf website](https://pragprog.com/book/es6tips/simplifying-javascript). The website also includes a handy community forum if you’d like to reach out for help along the way. Thanks in advance for reporting any issues that you find in the book code or text via the errata form, also conveniently found on the book website. To stay up-to-date on new syntax changes, you can follow me on twitter— [@joesmorgan](https://twitter.com/joesmorgan)—or online at [thejoemorgan.com](http://thejoemorgan.com/).

Let’s jump into the course. The first chapter provides tips on variable assignment.

**Intended Audience & Other Sources**

Let's look at the intended audience and resources for using this course.

**WE'LL COVER THE FOLLOWING**

* + [Who is this course for?](https://www.educative.io/courses/simplifying-javascript-handy-guide/RLRQwjknVYY#who-is-this-course-for)
  + [Online Resources](https://www.educative.io/courses/simplifying-javascript-handy-guide/RLRQwjknVYY#online-resources)

**Who is this course for?**[#](https://www.educative.io/courses/simplifying-javascript-handy-guide/RLRQwjknVYY#who-is-this-course-for)

This course is for anyone who has a little programming experience. It helps to have some JavaScript experience, but that’s not strictly necessary. I’ll assume that you know some basic programming concepts, such as *loops, conditionals, functions, and classes*. In short, if you’ve seen modern JavaScript and you’re excited to learn more, then this course is for you.

You may feel like you woke up in a strange new world. That’s not a bad thing. In fact, this new world is awesome—the coffee’s brewing, and the aroma of fresh-baked danishes makes this new world a place you want to wake up to. JavaScript is better than ever—it’s time to enjoy it.

**Online Resources**[#](https://www.educative.io/courses/simplifying-javascript-handy-guide/RLRQwjknVYY#online-resources)

You can also find the code on [github](https://github.com/jsmapr1/simplifying-js" \t "_blank) or on the book’s [Pragmatic Bookshelf website](https://pragprog.com/book/es6tips/simplifying-javascript). The website also includes a handy community forum if you’d like to reach out for help along the way. Thanks in advance for reporting any issues that you find in the book code or text via the errata form, also conveniently found on the book website. To stay up-to-date on new syntax changes, you can follow me on twitter— [@joesmorgan](https://twitter.com/joesmorgan)—or online at [thejoemorgan.com](http://thejoemorgan.com/).

Let’s jump into the course. The first chapter provides tips on variable assignment.

**MARK AS COMPLETED**

[**←    Back**](https://www.educative.io/courses/simplifying-javascript-handy-guide/q2ygQoymRw0)

You’ll learn about the topics this chapter contains, which include variable declaration using let and const.

**WE'LL COVER THE FOLLOWING**

* + [What does this chapter include?](https://www.educative.io/courses/simplifying-javascript-handy-guide/gk4xAzOVwyZ#what-does-this-chapter-include)

Before we begin, I have a question for you. How many variables did you declare in your code yesterday? It doesn’t matter what language you were writing. Was it *ten*? A *hundred*? How about over the last week? Last month? Probably a lot.

Now think about how many variables you read yesterday. Maybe you read your own code, or maybe you were skimming someone else’s. Did you see a hundred variables? a thousand? Chances are, you don’t have a clue.

Now if I asked you how many curried functions you saw yesterday, I bet you’d know the answer. I can tell you that I saw exactly one curried function yesterday. I know that because even though there’s been lots of ink spilled about curried functions in JavaScript (and I’ll be spilling some myself in [Tip 34, Maintain Single Responsibility Parameters with Partially Applied Functions](https://www.educative.io/collection/page/10370001/6728671147589632/5471495531790336)), it’s not nearly as common as a simple variable declaration. In fact, if you’ve never heard of a curried function, that’s even more proof that they aren’t nearly as important as a simple variable declaration. We spend so much time thinking and teaching complex concepts, but something as simple as *variable declaration* will affect your life and the lives of other developers in a much more significant way.

**What does this chapter include?**[#](https://www.educative.io/courses/simplifying-javascript-handy-guide/gk4xAzOVwyZ#what-does-this-chapter-include)

You’re about to rethink JavaScript code from the ground up. And that means you need to start at the most basic level: assigning information to variables. So that’s the theme for our first chapter.

Modern JavaScript has several new ways to declare variables. Whenever you start to write a variable, you just need to ask yourself if this will make the code more *readable and predictable* for the next developer. You’ll find that it actually changes how you write quite a bit.

You’re going to look at *two* new variable declaration types. The first, const, doesn’t allow you to reassign the variable (which you’ll see is a good thing).

The second, let, will allow reassignment, but it’s *block-scoped* and will protect you from potential scope conflicts. Finally, you’ll learn how to use *template literals* to create *new* strings from your variables.

The tips in this chapter will help you understand how your decisions will affect the rest of the code, and also how your decisions will affect anyone else who might eventually pick up and read your code.

I hope that as you read this chapter, you begin to critically examine the JavaScript that you write every day. The bonus is that with just a handful of tips, you’ll be well on your way to writing JavaScript code that’s more simple and expressive. And don’t be surprised if the mindset you learn when assessing variable declarations flows out into the rest of your code. After all, it’s the most common decision you’ll make while you write—a decision you’ll make 10, 20, 100 times tomorrow, and next week, and next month.

Ready? Good. Let’s begin.