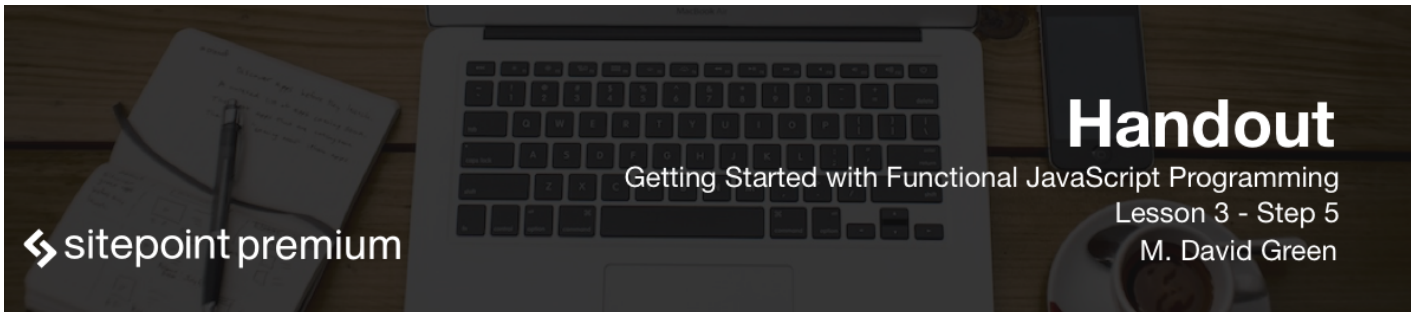

Handout

Getting Started with Functional JavaScript Programming

Lesson 3 - Step 5

M. David Green

So far in this course, we've looked at basic currying, and we've looked at partial application. But what if we wanna take it to the next level? We've created a partial utility, but frankly, it's not very robust. For one thing, it doesn't deal very well with missing parameters.

It doesn't handle changing parameter order. It doesn't deal with optional parameters very well. It doesn't do any error handling for us. And there are other problems with it as well. But instead of trying to build up the robustness of this particular partial utility, I recommend using currying and partial application functions that come from a well-maintained and well-recognized library.

And there are a number of functional programming libraries out there. A few examples are Ramdajs, Functional.js, Lodash, and Fantasyland. And there are others. It's mostly a matter of getting familiar with how they work and watching the argument order issues that might come up. So there are a few things to keep in mind if you want to start using currying and partial application in your own programming.

Plan ahead for currying and partial application. Think about the way that you're designing your functions. Work with your team to decide what syntax works best for you. If you like multiple sets of parentheses, that's fine. If you want to use a currying utility or partial application utility to avoid them, that's also fine.

But choose a good library with functions that everybody can agree to. And stick with it. Make sure that it's well maintained, well documented, and has a good community of support behind it. And when you're designing your functions, always design them with the argument order in mind. Make sure that the arguments that are passed in last are the ones that you're most likely to wanna modify.

And make sure the utility functions in the library that you're using recognize and respect that order appropriately.