



So, what's next for you with functional programming in JavaScript? That's entirely up to you, but I'd like you to start trying to apply these ideas today. You don't have to use functional techniques exclusively. It's possible to build an entire JavaScript application using nothing but functional code. And while your code will probably be very clean, it might not be the most efficient, it might not be the most maintainable, and it might not be the most understandable.

Using a combination of techniques is appropriate, depending on the type of code that you're writing and the context you're writing it in. Functional code, however, should fit nicely with your existing projects. There's nothing that I showed you in this course that you can't start using today, even in the middle of existing code.

Thinking functionally is going to start improving your code right away. You're going to start looking at the functions that you create and thinking "is this a pure function?". I recommend that you actually try starting with functional solutions first and then refactor them out if you need to. And one of the reasons that you might need to currently is because of performance, although I don't think it's very likely that you're going to run into this. As I mentioned during the course, some of the functional methods that were added in ECMAScript 5, and probably some of the ones that come in with the libraries that you're going to be using, might not perform as efficiently as, for example, a for loop versus map.

But writing the code in the cleanest way possible first, and then refactoring if it turns out that you really need to in production is a better approach. After all, premature optimization is the devil. You should test the effects of what you're doing, and see what the impact is of your functional code in real world situations.

And, in this case, I'm not just talking about performance, but I'm also talking about how clean and maintainable your code is, how easy it is to explain to people, how easy it is to share and understand with the rest of your team. Ultimately, I think you're going to benefit just by having these concepts in your back pocket and applying them when you see an opportunity.

Maybe you should go back and try refactoring one of your existing applications to incorporate some of these functional techniques. See if your code doesn't clean up, see if you don't get some positive benefits from it. I'm going to be curious to hear what you think, so leave me messages in the comments.

Let me know what your experiences are, and what questions you're left with. This has been an overview of functional programming techniques. I intentionally avoided giving you copy and paste code that you can use in your own applications, because I want you to build these things yourself. So go out there, do some functional JavaScript programming, and let me know how it works for you.

This is M. David Green, and this has been Functional Programming in JavaScript!