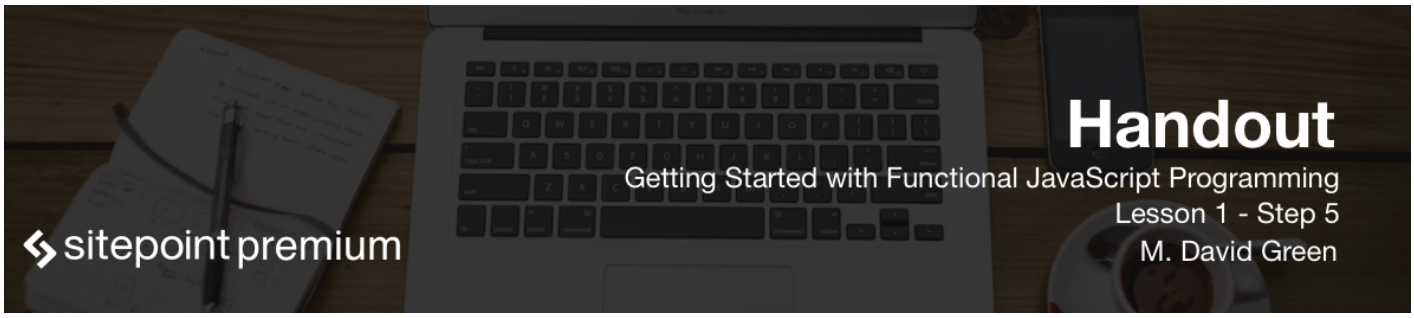

One of the first concepts that it's useful to understand when thinking about functional programming is the concept of **Pure Functions**. And this idea is so useful, that even if you don't do anything else with functional programming, integrating this into the way that you develop your functions is going to change your JavaScript tremendously.

So what are pure functions?

Pure functions don't rely on the state of the code that they're called from. Whatever's happening outside of them as long as you call up your function with the same values, it will always return the same results. Pure functions also don't create side effects that alter variables outside of themselves. So once you're done calling a pure function and its returned a value, nothing outside of that function has been changed, other than what happened inside of the function itself. The value that it returned is the only thing that it alters. There's only one result provided by any set of arguments that you give to a pure function. As long as you give a pure function the exact same set of arguments, it should always return the exact same results. This makes for fantastic ease of testing as well as convenience in conceiving how these functions work.

Once you start thinking about your functions as pure and writing them in this way, you're going to be amazed at the impact that has on the cleanness and the readability of your code.

Pure functions have no side effects. An impure function in contrast makes changes to the values that are outside of the function itself. Finally, a pure function should always return the same result, for the same input. With an impure function, you can't rely on it always returning the same result when you give it the same input, which makes it very difficult to write tests for an impure function.
It doesn't rely on the external state. If you passed in a different external variable, you would get a different result. But now, let's do an external.push, a value of 4. And let's again, paste exactly the same line that we had before. If we clear and run, we'll see that using the exact same code gave us a different result.