

# Introduction

You'll learn about the topics this chapter contains, which include exploring different functions such as arrow and higher-order functions.

## We'll cover the following



- What does this chapter include?

Several years ago, the [New York Times](#) asked a master furniture builder to review furniture from Ikea, Target, and other discount retailers.

Unsurprisingly, the craftsman was not impressed. He noticed problems with the wood finish, screws that would tear out, and other issues. Astonishingly, he noticed a piece of chipping paint and concluded that the piece was painted before it was cut and that the blade cutting the wood was dull. A single piece of chipped paint told him something about the construction process and the tools used to create it.

A master of his or her craft can see things that others don't.

## What does this chapter include? #

In this chapter, you're going to learn to master functions, and in the process, you'll see ideas in code that you've probably never considered before. Problems disappear and new ideas pop up regularly.

Functions are so *common* that it's easy to ignore them. But functions in JavaScript are different. You may think you know how to use them, but you'll be surprised to find how much more they are capable of. This chapter isn't about the basics. It's about how you can use functions in ways you may not have explored before.

You'll start off by learning how to write testable code. *Testable functions* are more clean and easy to maintain, but learning to write them takes a little effort.

From there, you'll return to *arrow functions* to see how you can take all the ideas from parameters and apply them in single line functions.

Next, you'll dive into **higher-order** functions—*functions that return functions*—and

Next, you'll dive into *higher-order functions* — functions that return functions — and learn two techniques to help you lock in information to make clear, reusable functions.

Finally, you'll return to *arrow functions* to see how you can use them to solve a sneaky context problem that will drive you crazy if you aren't expecting it.

*Functions, currying, higher-order functions.* These are the concepts that turn casual JavaScript writers into fanatics. They're the concepts I love most about the language.

When you're finished, you'll see problems in new ways. You won't be baffled when this returns undefined. You'll see a context problem that you can solve with arrow functions. When a variable is inaccessible, you'll see an opportunity for closures. Masters of a craft see the world differently. When you're finished mastering functions, you'll see all problems in a new way.

---

In the next tip, you'll learn how to write testable code.