Testing JavaScript with JsUnit

By now, it should be obvious that to really make the most of Ajax, you’re going to need to write some JavaScript. Frameworks and toolkits can ease some of the burden, but in the end, you’ll probably have more JavaScript than normal. Having written a fair amount of JavaScript ourselves, we know this can be daunting, but in this chapter we’ll put a few more arrows in your quiver.

Specifically, we’ll introduce test-driven development (TDD) and show how you can apply it to JavaScript. While this approach won’t instantly solve all your coding problems, it should at least help you get home in time to eat dinner with your family. We’ll start with a brief overview of TDD and the ubiquitous JUnit. Once we have established the foundation, we’ll discuss JsUnit and show how you can write and run tests.

Rising to the JavaScript Challenge

If you’ve done any amount of work with Web applications, you’ve probably had to write some JavaScript; of course, if you’ve written anything more than the simplest of functions, your opinion of JavaScript might not be too high. Browser incompatibilities, a lack of decent development tools, no code completion, and no debuggers—it’s enough to make most developers long for vi.1

We know your pain. In Chapter 5, we discussed a number of tools to make your life easier. In this chapter, we’ll show how to make developing JavaScript as easy as possible (at least until the tool vendors catch up²). By writing your JavaScript in the test-first manner, you can greatly simplify the entire process.

Introducing the Test-First Approach

Yeah, we can hear you now—“I write tests, just before the product ships.” Some of you are snickering and saying something about the quality assurance department. Still others have project managers who have said something along the lines of, “We can’t waste time writing tests; we need to write real code.” So, what does it mean to practice TDD?

TDD rose out of the agile development movement, specifically extreme programming (XP), where it is a central principle. Rather than write your tests when you are done, often as an afterthought, TDD practitioners write the tests before they write any code. In essence, the

---

1. OK, so for some of you, vi is obvious, but c’mon, even you’ve got to have issues with JavaScript!
2. We fully expect the major tool vendors will solve this problem in the near future.