JsUnit contains another feature that you won’t find in JUnit: the setUpPage() function. Some in the JUnit community think its lack of “one-time setup” and “one-time teardown” is a flaw in its design. Some have gone so far as to extend JUnit or create new test frameworks that have this feature baked in; in addition, the JUnit frequently asked questions (FAQ) site even describes one way to approximate this behavior. Most books about JUnit also discuss approaches to this issue.

JsUnit, however, does contain one-time setup; the setUpPage() function is called exactly once per test page and before any of your test functions are called. By now, you’ve probably figured out that this is a great place to stash preprocessing, especially if you need to load any data to your page before running the tests. Unlike the setUp() and tearDown() functions, there is a little more to using setUpPage() than just putting your processing in the function. If you do choose to use this feature, make sure you set the setUpPageStatus variable to complete when your function is done—this tells JsUnit it can move on and fire the tests on your test page.

How about an example? Let’s return to the simpleJS.js file and add three functions to round out the math features. Include subtract, multiply, and divide functions, as shown in Listing 6-6.

Listing 6-6. simpleJS2.js

```javascript
function addTwoNumbers(value1, value2) {
    return parseInt(value1) + parseInt(value2);
}

function subtractTwoNumbers(value1, value2) {
    return parseInt(value1) - parseInt(value2);
}

function multiplyTwoNumbers(value1, value2) {
    return parseInt(value1) * parseInt(value2);
}

function divideTwoNumbers(value1, value2) {
    return parseInt(value1) / parseInt(value2);
}
```

Now use the setUpPage() function to set up some simple test data, as shown in Listing 6-7. Please note the last line of the function—you have to tell JsUnit that you’re done setting up your page.

---

4. junit.sourceforge.net/doc/faq/faq.htm#organize_3