**Summary**

During this chapter, you’ve learned what web scripting is and what JavaScript is. You’ve also

learned how to insert a script into an HTML document or refer to an external JavaScript file, what

sorts of things JavaScript can do, and how JavaScript differs from other web languages. You also

wrote a simple JavaScript program and tested it using a web browser. You discovered how to modify

and test scripts and what happens when a JavaScript program runs into an error.

In the process of writing this script, you have used some of JavaScript’s basic features: variables, the

document.write statement, and functions for working with dates and times.

Now that you’ve learned a bit of JavaScript syntax, you’re ready to continue on to learn all manner

and sorts of things about web development before settling in to write interactive websites using

client-side scripting.

**Q&A**

**Q. Do I need to test my JavaScript on more than one browser?**

**A.** In an ideal world, any script you write that follows the standards for JavaScript will work in all

browsers, and 93% of the time (give or take) that’s true in the real world. But browsers do have

their quirks, and you should test your scripts on Internet Explorer and Firefox at a minimum.

**Q. If I plan to learn PHP or some other server-side programming language anyway, will I have**

**any use for JavaScript?**

**A.** Certainly. JavaScript is the ideal language for many parts of a web-based application, such as

form validation. Although PHP and other server-side languages have their uses, they can’t interact

directly with the user on the client-side.

**Q. When I try to run my script, the browser displays the actual script in the browser window**

**instead of executing it. What did I do wrong?**

**A.** This is most likely caused by one of three errors. First, you might be missing the beginning or

ending <script> tags. Check them, and verify that the first reads <script type="text/javascript">. Second, your file

might have been saved with a .txt extension, causing the browser to treat it as a text file. Rename it

to .htm or .html to fix the problem. Third, make sure your browser supports JavaScript and that it

is not disabled in the Preferences dialog.

**Q. Why are the** <strong> **and** <br /> **tags allowed in the statements to print the time? I thought**

**HTML tags weren’t allowed within the** <script> **tags.**

**A.** Because this particular tag is inside quotation marks, it’s considered a valid part of the script.

The script’s output, including any HTML tags, is interpreted and displayed by the browser. You

can use other HTML tags within quotation marks to add formatting, such as the <h1> tags we added

for the large clock display.

**Workshop**

The workshop contains quiz questions and exercises to help you solidify your understanding of the

material covered. Try to answer all questions before looking at the “Answers” section that follows.

**Quiz**

**1.** When a user views a page containing a JavaScript program, which machine actually executes the

script?

**a.** The user’s machine running a web browser

**b.** The web server

**c.** A central machine deep within Netscape’s corporate offices

**2.** What software do you use to create and edit JavaScript programs?

**a.** A browser

**b.** A text editor

**c.** A pencil and a piece of paper

**3.** What are variables used for in JavaScript programs?

**a.** Storing numbers, dates, or other values

**b.** Varying randomly

**c.** Causing high school algebra flashbacks

**4.** What should appear at the very end of a JavaScript script embedded in an HTML file?

**a.** The <script type="text/javascript"> tag

**b.** The </script> tag

**c.** The END statement

**Answers**

**1.** a. JavaScript programs execute on the web browser. (There is actually a server-side version of

JavaScript, but that’s another story.)

**2.** b. Any text editor can be used to create scripts. You can also use a word processor if you’re

careful to save the document as a text file with the .html or .htm extension.

**3.** a. Variables are used to store numbers, dates, or other values.

**4.** b. Your script should end with the </script> tag.

**Exercises**

• Add a millisecond field to the large clock. You can use the getMilliseconds function, which works just

like getSeconds but returns milliseconds.

• Modify the script to display the time, including milliseconds, twice. Notice whether any time

passes between the two time displays when you load the page.