



Chapter 1

Introduction to JavaScript

Key Skills & Concepts

- Using Text Editors, WYSIWYG Editors, and Web Browsers
 - Defining JavaScript
 - Differences Between JavaScript and Other Languages
-

Welcome to *JavaScript: A Beginner's Guide, Third Edition*! You're obviously interested in learning JavaScript, but perhaps you're not sure what you need to know to use it. This chapter answers some basic questions about what JavaScript is, discusses its advantages and limitations, explains how you can use it to create more dynamic and inviting Web pages, and provides a brief history of the language.

JavaScript is ubiquitous on the World Wide Web. You can use JavaScript both to make your Web pages more interactive, so that they react to a viewer's actions, and to give your Web pages some special effects (visual or otherwise).

JavaScript often gets thrown in with Hypertext Markup Language (HTML) as one of the recommended languages for beginning Web developers (whether you build Web sites for business or pleasure). Of course, you can build a Web page by using only HTML, but JavaScript allows you to add additional features that a static page of HTML can't provide without some sort of scripting or programming help.

What You Need to Know

Before you begin learning about JavaScript, you should have (or obtain) a basic knowledge of the following:

- HTML and Cascading Style Sheets (CSS)
- Text editors
- Web browsers
- The different versions of JavaScript

If you have this basic knowledge (the different versions of JavaScript will be discussed in this chapter), then you'll do just fine as you work through this book. Knowing another programming/scripting language or having previous experience with JavaScript isn't required. This book is a beginner's guide to JavaScript.

If you think you don't have enough experience in one of the aforementioned areas, a closer look at each one may help you decide what to do.

Basic HTML and CSS Knowledge

While you don't need to be an HTML guru, you do need to know where to place certain elements (like the head and body elements) and how to add your own attributes. This book will reference scripts in the head section (between the <head> and </head> tags) and the body section (between the <body> and </body> tags).

Occasionally, you will also need to add an attribute to a tag for a script to function properly. For example, you may need to name a form element using the id attribute, as shown in the following code:

```
<input type="text" id="thename" />
```

If you know the basics of using tags and attributes, the HTML portion shouldn't pose any problems to learning JavaScript.

If you don't have a basic knowledge of HTML, you can learn it fairly quickly through a number of media. For example, you can buy a book or look for some helpful information on the Web. A good book is *HTML: A Beginner's Guide, Fourth Edition* by Wendy Willard (McGraw-Hill Professional, 2009). To find information about HTML on the Web, check out these sites: www.pageresource.com/html and www.w3schools.com/html/default.asp.

Occasionally, you will need to use CSS to add or change presentation features on a Web page. We will mainly use CSS for the purposes of dynamically changing CSS properties via JavaScript in this book. A good place to learn CSS is www.w3schools.com/css/css_intro.asp.

Basic Text Editor and Web Browser Knowledge

Before jumping in and coding with JavaScript, you must be able to use a text editor or HTML editor, and a Web browser. You'll use these tools to code your scripts.

Text Editors

A number of text editors and HTML editors support JavaScript. If you know HTML, you've probably already used an HTML editor to create your HTML files, so you might not have to change.

However, some HTML editors have problems related to adding JavaScript code (such as changing where the code is placed or altering the code itself when you save the file). You may need to use a simpler editor or look for an HTML editor that handles the addition of your own JavaScript code easily (such as Adobe Dreamweaver). Some examples of text editors are Notepad, TextPad, and Simple Text.

Web Browsers

Again, if you've been coding in HTML, you probably won't need to change your browser. However, some browsers have trouble with the newer versions of JavaScript. The choice of Web browser is ultimately up to you, as long as it's compatible with JavaScript. I recommend one of the following browsers to test your JavaScript code:

- Microsoft Internet Explorer version 6.0 or later
- Mozilla Firefox version 1.0 or later
- Opera version 6.0 or later

New versions of these browsers continue to be produced. At the time of this writing, nonbeta versions of Internet Explorer 8, Firefox 3, and Opera 9 are available.

To give you an idea of what some browsers look like, Figure 1-1 shows a Web page when viewed in Microsoft Internet Explorer, and Figure 1-2 shows the same page when viewed in Mozilla Firefox.

If you have an older browser and you can't upgrade, a number of features (mostly discussed later in the book) may not work in that browser. Even so, the book can still help you learn the JavaScript language itself, so you don't need to give up if you have an older browser. The three browsers mentioned and the versions of JavaScript they support are shown in Table 1-1.

The next section, "Which Version?," explains what the version numbers mean in more detail. Once you've determined that you meet the basic requirements, you're ready to begin learning the language.

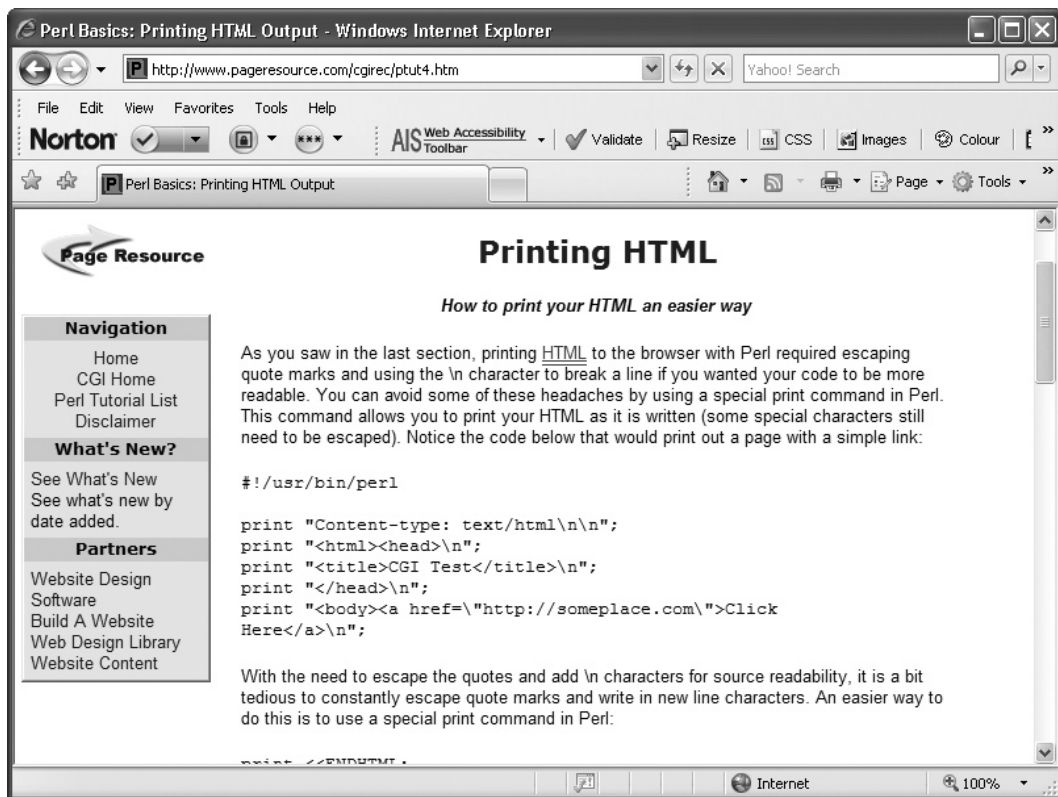


Figure 1-1 A Web page viewed in Microsoft Internet Explorer

Microsoft Internet Explorer Version	Mozilla Firefox Version	Opera Version	JavaScript Version Supported
–	3	–	1.8
–	2	–	1.7
–	1.5	–	1.6
5.5–8	1	6–9	1.5
4	–	–	1.3

Table 1-1 JavaScript Versions Supported by the Three Major Browsers

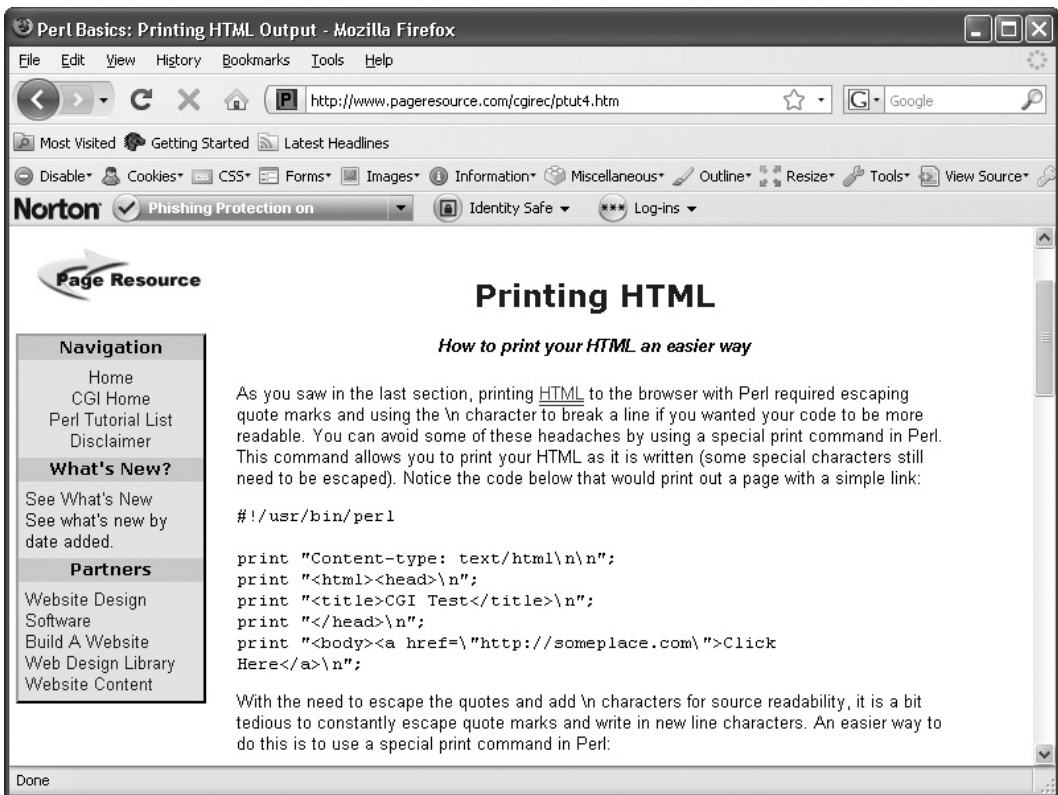


Figure 1-2 A Web page viewed in Mozilla Firefox

Which Version?

At the time of this writing, the browsers recommended earlier in this chapter should support at least JavaScript 1.5. (The newest version of Firefox supports JavaScript 1.8.)

You may also see or hear about JScript or ECMAScript. JScript is the version of JavaScript that Microsoft Internet Explorer uses (which has additional features because it is implemented as a Windows Script engine; it can use server-side languages to perform more complex tasks like updating databases). For more information on JScript, see <http://msdn.microsoft.com/en-us/library/hbxc2t98.aspx>.

ECMAScript is the international standard name and specification for the JavaScript language, so it's not a new language but a standard that is set for JavaScript and JScript. For more on ECMAScript, see www.ecma-international.org/publications/standards/Ecma-262.htm.

Ask the Expert

Q: You mentioned that I could use a text editor or HTML editor of my choice, but I'm not quite sure what that means. What is a text editor and where can I find one?

A: A text editor is a program that you can use to save and edit written text. Text editors range from simple to complex, and a number of choices are available: Notepad, WordPad, Simple Text, and Corel WordPerfect X4, to name a few. You can also purchase and download some from the Web, like NoteTab or TextPad.

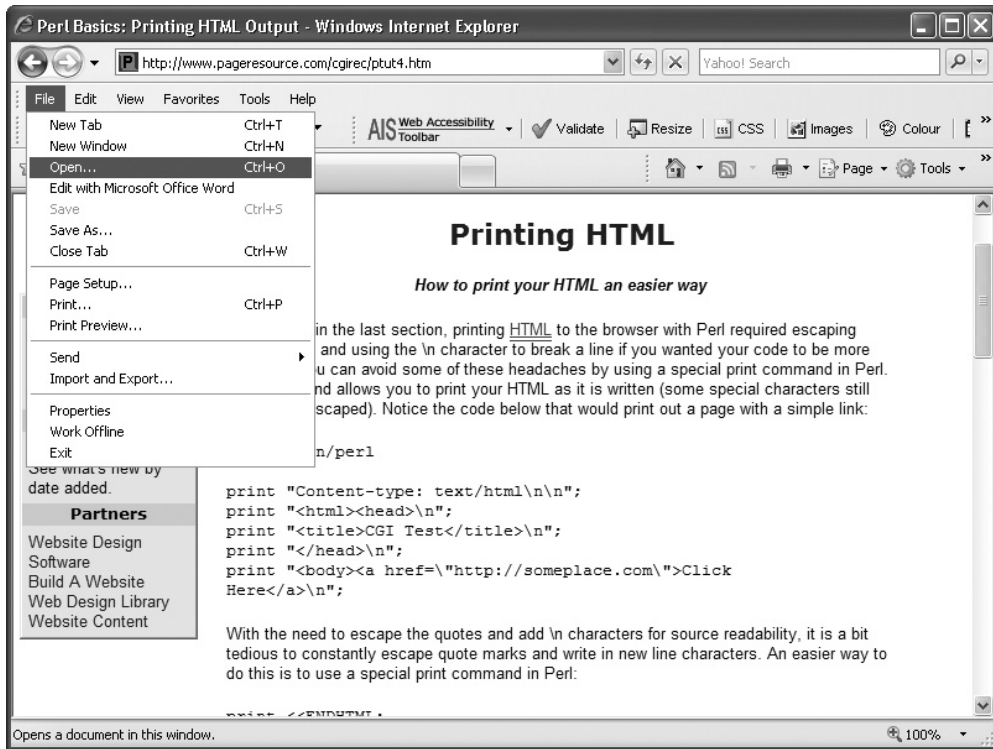
An HTML editor is either a more complex text editor or an editor that allows you to add code by clicking buttons or by other means—often called a What You See Is What You Get (WYSIWYG) editor. I recommend a plain text editor or an HTML editor that doesn't change any code you add to it manually. Some examples of HTML editors are Adobe Dreamweaver and Softpress Freeway.

Q: What exactly do I need to know about using a text editor?

A: Basically, you only need to know how to type plain text into the editor, save the file with an .html or .htm extension, and be able to open it again and edit it if necessary. Special features aren't needed because HTML files are made up of plain text.

Q: What do I need to know about using a browser?

A: All you absolutely need to know is how to open a local HTML file on your computer (or on the Web) and how to reload a page. If you don't know how to open an HTML file from your own computer, open your browser and go to the File menu. Look for an option that says something like Open or Open File, and select it. You should be able to browse for the file you want to open like you would with other programs. The following illustration shows where the option is in Microsoft Internet Explorer:



Q: Where do I get those browsers you mentioned?

A: Here are links for the browsers:

- **Microsoft Internet Explorer** www.microsoft.com/ie
- **Mozilla Firefox** www.mozilla.com/firefox
- **Opera** www.opera.com

Remember, It's Not Java

JavaScript and Java are two different languages. Java is a full programming language that must be compiled (running a program through software that converts the higher-level code to machine language) before a program (often called a Java *applet*) can be executed. Java is more powerful but also more complex. JavaScript doesn't need a compiler and is more lenient in a number of areas, such as syntax.

Similarities to Other Languages

JavaScript does have similarities to other programming and scripting languages. If you have experience with Java, C++, or C, you'll notice some similarities in the syntax, which may help you to learn more quickly. Because it's a scripting language, JavaScript also has similarities to languages like Perl—it, too, can be run through an interpreter rather than being compiled.

If you have programming or scripting experience in any language, it will make learning JavaScript easier—but it isn't required.

Beginning with JavaScript

JavaScript came about as a joint effort between Netscape Communications Corporation and Sun Microsystems, Inc. The news release of the new language came on December 4, 1995, back when Netscape Navigator 2.0 was still in its beta version. JavaScript version 1.0 became available with the new browser. (Before its release as JavaScript, it was called LiveScript.)

JavaScript is an object-based, client-side scripting language that you can use to make Web pages more dynamic. To make sense of such a definition, let's look at its important parts one by one.

Object Based

Object based means that JavaScript can use items called *objects*. However, the objects are not *class based* (meaning no distinction is made between a class and an instance); instead, they are just general objects. You'll learn how to work with JavaScript objects in Chapter 8. You don't need to understand them in any detail until you know a few other features of the language.

Client Side

Client side means that JavaScript runs in the *client* (software) that the viewer is using, rather than on the Web server of the site serving the page. In this case, the client would be a Web browser. To make more sense of this, let's take a look at how a server-side language works and how a client-side language works.

Server-Side Languages

A server-side language needs to get information from the Web page or the Web browser, send it to a program that is run on the host's server, and then send the information back to the browser. Therefore, an intermediate step must send and retrieve information from the server before the results of the program are seen in the browser.

A server-side language often gives the programmer options that a client-side language doesn't have, such as saving information on the Web server for later use, or using the new information to update a Web page and save the updates.

However, a server-side language is likely to be limited in its ability to deal with special features of the browser window that can be accessed with a client-side language (like the content in a particular location on a Web page or the contents of a form before it's submitted to the server).

Client-Side Languages

A client-side language is run directly through the client being used by the viewer. In the case of JavaScript, the client is a Web browser. Therefore, JavaScript is run directly in the Web browser and doesn't need to go through the extra step of sending and retrieving information from the Web server.

With a client-side language, the browser reads and interprets the code, and the results can be given to the viewer without getting information from the server first. This process can make certain tasks run more quickly.

A client-side language can also access special features of a browser window that may not be accessible with a server-side language. However, a client-side language lacks the ability to save files or updates to files on a Web server like a server-side language can.

NOTE

Using the XMLHttpRequest object allows JavaScript to request data from the server. This will be covered briefly in Chapter 16.

A client-side language is useful for tasks that deal with parts of the browser or that allow information to be validated before it is sent to a server-side program or script. For instance, JavaScript can open a new window with specific dimensions, specific features (such as a location bar or status bar), and a specific point of placement on the screen.

JavaScript can also be used to check the information entered into a form before the form is sent to a server-side program to be processed. This information check can prevent strain on the Web server by preventing submissions with inaccurate or incomplete information. Rather than running the program on the server until the information is correct, that data can be sent to the server just once with correct information.

Scripting Language

A scripting language doesn't require a program to be compiled before it is run. All the interpretation is done on-the-fly by the client.

With a regular programming language, before you can run a program you have written, you must compile it using a special compiler to be sure there are no syntax errors. With a scripting language, the code is interpreted as it is loaded in the client. Thus, you can test the results of your code more quickly. However, errors won't be caught before the script is run and could cause problems with the client if it can't handle the errors well. In the case of JavaScript, the error handling is up to the browser being used by the viewer.

Putting It All Together

With all this in mind, you might wonder how JavaScript is run in a browser. You might wonder where to write your JavaScript code and what tells the browser it is different from anything else on a Web page. The answers are general for now, but the next chapter provides more details.

JavaScript runs in the browser by being added into an existing HTML document (either directly or by referring to an external script file). You can add special tags and commands to

the HTML code that will tell the browser that it needs to run a script. Once the browser sees these special tags, it interprets the JavaScript commands and will do what you have directed it to do with your code. Thus, by simply editing an HTML document, you can begin using JavaScript on your Web pages and see the results.

For example, the following code adds some JavaScript to an HTML file that writes some text onto the Web page. Notice the addition of `<script>` and `</script>` tags. The code within them is JavaScript.

```
<html>
<body>
<script type="text/javascript">
document.write("This writes text to the page");
</script>
</body>
</html>
```

This tag tells the browser that JavaScript follows

This line writes the text inside the quote marks on the page

This line tells the browser that this is the end of the script

The next chapter looks at how to add JavaScript in an HTML file by using the `<script>` and `</script>` HTML tags. This will be your first step on the road to becoming a JavaScript coder!

Online Resources

To find additional information online to help you with JavaScript, here are some useful resources:

- A place to find tutorials with working examples of the results: www.pageresource.com/jscript
- An excellent tutorial site that includes cut-and-paste scripts: www.javascriptkit.com
- A place where you can address questions about JavaScript to fellow coders: www.webxpertz.net/forums

Try This 1-1 Use JavaScript to Write Text

pr1_1.html

This project shows you JavaScript in action by loading an HTML document in your browser. The script writes a line of text in the browser using JavaScript.

Step by Step

1. Copy and paste the code shown here into your text editor:

```
<html>
<body>
<script type="text/javascript">
document.write("This text was written with JavaScript!");
</script>
```

```
</body>  
</html>
```

2. Save the file as **pr1_1.html** and open it in your Web browser. You should see a single line of text that was written with JavaScript.

Try This Summary

In this project, you copied and pasted a section of code into a text editor and saved the file. When you opened the saved file in your Web browser, a line of text was displayed in the browser. This text was written in the browser window using JavaScript. You will see more about how this type of script works in Chapter 2.



Chapter 1 Self Test

1. You must know which of the following to be able to use JavaScript?
 - A Perl
 - B C++
 - C HTML
 - D SGML
2. Which of the following is something you should have to use JavaScript?
 - A A Web browser
 - B A C++ compiler
 - C A 50GB hard drive
 - D A CD-RW drive
3. The choice of a Web browser is up to you, as long it's compatible with _____.
 - A Flash MX
 - B VBScript
 - C JavaScript
 - D Windows XP

4. JavaScript and Java are the same language.
 - A True
 - B False
5. JavaScript is more _____ than Java in a number of areas, such as syntax.
 - A complex
 - B powerful
 - C compiled
 - D lenient
6. JavaScript has similarities to other programming and scripting languages.
 - A True
 - B False
7. Before its release as JavaScript, JavaScript was called _____.
 - A Java
 - B JavaCup
 - C LiveScript
 - D EasyScript
8. JavaScript is _____.
 - A object based
 - B object oriented
 - C object deficient
 - D not a language that uses objects
9. A client-side language is run directly through the _____ being used by the viewer.
 - A server
 - B client
 - C monitor
 - D lawyer

10. How can a client-side language help when using forms on a Web page?
- A It can save the information on the server.
 - B It can validate the information before it is sent to the server.
 - C It can update a file and save the file with the new information.
 - D It can't help at all.
11. A _____ language doesn't require a program to be compiled before it is run.
- A programming
 - B server-side
 - C scripting
 - D computer
12. With a scripting language, the code is interpreted as it is loaded in the client.
- A True
 - B False
13. In JavaScript, what handles errors in a script?
- A The Web server
 - B A compiler
 - C A program on the Web server
 - D The Web browser
14. How is JavaScript added to a Web page?
- A It is written into a special editor in the browser.
 - B It is taken from a compiled program on the server.
 - C You place the code in a file by itself and open that file.
 - D It is added to an HTML document.
15. What is added to a Web page to insert JavaScript code?
- A `<script>` and `</script>` HTML tags
 - B The JavaScript code word
 - C `<javascript>` and `</javascript>` HTML tags
 - D `<java>` and `</java>` HTML tags