CHAPTER 1

# Getting Started with JavaScript

This book is about a scripting language called **JavaScript** and how to use it in a practical manner. After you read it, you'll be able to

- Understand JavaScript syntax and structures.
- Create scripts that are easy to understand and maintain.
- Write scripts that do not interfere with other JavaScripts.
- Write scripts that make web sites easier to use without blocking out non-JavaScript users.
- Write scripts that are independent of the browser or user agent trying to understand them—which means that in some years they will still be usable and won't rely on obsolete technology.
- Enhance a web site with JavaScript and allow developers without any scripting knowledge to change the look and feel.
- Enhance a web document with JavaScript and allow HTML developers to use your functionality by simply adding a CSS class to an element.
- Use progressive enhancement to make a web document nicer only when and if the user agent allows for it.
- Use Ajax to bridge the gap between back end and client side, thus creating sites that are easier to maintain and appear much slicker to the user.
- Use JavaScript as part of a web methodology that enables you to maintain it independently without interfering with the other development streams.

What you will not find here are

- Instructions on how to create effects that look flashy but do nothing of value for the visitor
- JavaScript applications that are browser specific
- JavaScripts that are only there to prove that they can be used and do not enhance the visitor's experience
- JavaScripts that promote unwanted content, such as pop-up windows or other flashy techniques like tickers or animation for animation's sake

It is my credo that JavaScript has a place in modern web development, but we cannot take it for granted that the visitor will be able to use or even experience all the effects and functionality we can achieve with JavaScript. JavaScript allows us to completely change the web page by adding and removing or showing and hiding elements. We can offer users richer interfaces like drag-and-drop applications or multilevel drop-down menus. However, some visitors cannot use a drag-and-drop interface because they can only use a keyboard or rely on voice recognition to use our sites. Other visitors might be dependent on *hearing* our sites rather than seeing them (via screen readers) and will not necessarily be notified of changes achieved via JavaScript. Last but not least, there are users who just cannot have JavaScript enabled, for example, in high-security environments like banks. Therefore, it is necessary to back up a lot of the things we do in JavaScript with solutions on the server side.

Sadly, JavaScript also has a history of being used as a way to force information onto the visitor that was not requested (pop-up windows are a good example). This practice is frowned on by me, as well as many professional web designers. It is my hope that you will not use the knowledge gained from this book to such an end.

■**Note** Web design has matured over the years—we stopped using `FONT` tags and deprecated visual attributes like `bgcolor` and started moving all the formatting and presentational attributes to a CSS file. The same cleaning process has to happen to JavaScript should it remain a part of web development. We separated content, structure, and presentation, and now it is time to separate the behavior of web sites from the other layers. Web development now is for business and for helping the user rather than for the sake of putting something out there and hoping it works in most environments.

It is high time we see JavaScript as a part of an overall development methodology, which means that we develop it not to interfere with other technologies like HTML or CSS, but to interact with them or complement them. To that end, we see the emergence of a new technology (or at least a new way of using existing technologies) called **Ajax**, which we will discuss in Chapter 8.

Web development has come quite a way since the 1990s, and there is not much sense in creating web sites that are static and fixed in their size. Any modern web design should allow for growth as needed. It should also be accessible to everyone (which does not mean that everybody gets the same appearance—a nice multicolumn layout, for example, might make sense on a high-resolution monitor but is hard to use on a mobile phone or a PDA)—and ready for internationalization. We cannot afford any longer to build something and think it'll last forever. Since the Web is about content and change, it'll become obsolete if we don't upgrade our web products constantly and allow other data sources to feed into it or get information from it.

Enough introductions—you got this book to learn about JavaScript, so let's start by talking quickly about JavaScript's history and assets before diving right into it.