

HTML5 capable browsers will validate the information before it is sent to the `validate_input` function. Thus, if the browser is completely HTML5, all the information will also pass the validation from the JavaScript method. You could check the browser version to determine if the browser is 100% HTML5. However, that would take more JavaScript coding and is unnecessary since the application accomplishes its task without more code.

It is important that all validations are consistent throughout the process of accepting information from the user and passing that information to the application. HTML and JavaScript code are easily viewable by the user (you can “View Source” from within a browser to see the code). Validation of any highly secure format should be done from within a program language that is compiled and secured on a server.

*Security and performance* —You may be wondering why you should even bother to validate on the user side. Why not just pass all the information to the program on the server and let that program tell you if you need to fix anything? Some programmers actually do this. However, the goal is to have an efficient program. By attempting to validate the information in the user’s browser, you reduce the number of calls to/from the server. This improves the application and web server performance and efficiency. In addition, as you will see, by validating in the browser, the contents in the text boxes will still be available to the user to adjust. If validation is done on the web server, information in the HTML form will be lost because the web page will be reloaded each time you send and receive information from the web server.

The goal of validation in the browser is to make sure that the user provides information that meets the requirements of the program on the server. You may initially cringe that this example displays the format of required information to the user. However, the goal is not to secure the data; the goal is to make sure the data is valid. It is not a security breach to inform the user of the data format requested.

```
<form method="post" action="lab.php" onSubmit="return validate_input(this)">
```

The form uses the parameter `onSubmit` to pass all information that has been entered to the JavaScript method (`validate_input`). The `this` keyword indicates that all information from this object (which is the form itself and all the text boxes) is passed to the method. The `return` keyword indicates that the method will return a `TRUE` or `FALSE` status. The HTML code will submit the form to the PHP program (`lab.php`) on the server if a `TRUE` status has been returned from the `validate_input` method.

## Do It

1. Adjust Example 4-1 to include gender information from the Do It in Chapter 3. Use a text box to receive the information from the user. Try to use HTML5 to restrict the type of information the user can enter in the text box. Test your code.