Week 15: Web forms

251514 WEB PROGRAMMING ESSENTIALS

Objectives

- By the end of this session students should be able to;
  - Understand how forms work
  - Use the <form> element
  - Create input objects
  - Style forms with Cascading Style Sheets (CSS)

Understanding How Forms Work

- Forms let you build interactive web pages that collect information from a user and process it on the web server
- The HTML form is the interface for the user to enter data
- The data is processed by applications that reside on the web server
- The data-processing software can then work with the data and send a response back to the user
- The user enters data via an HTML form

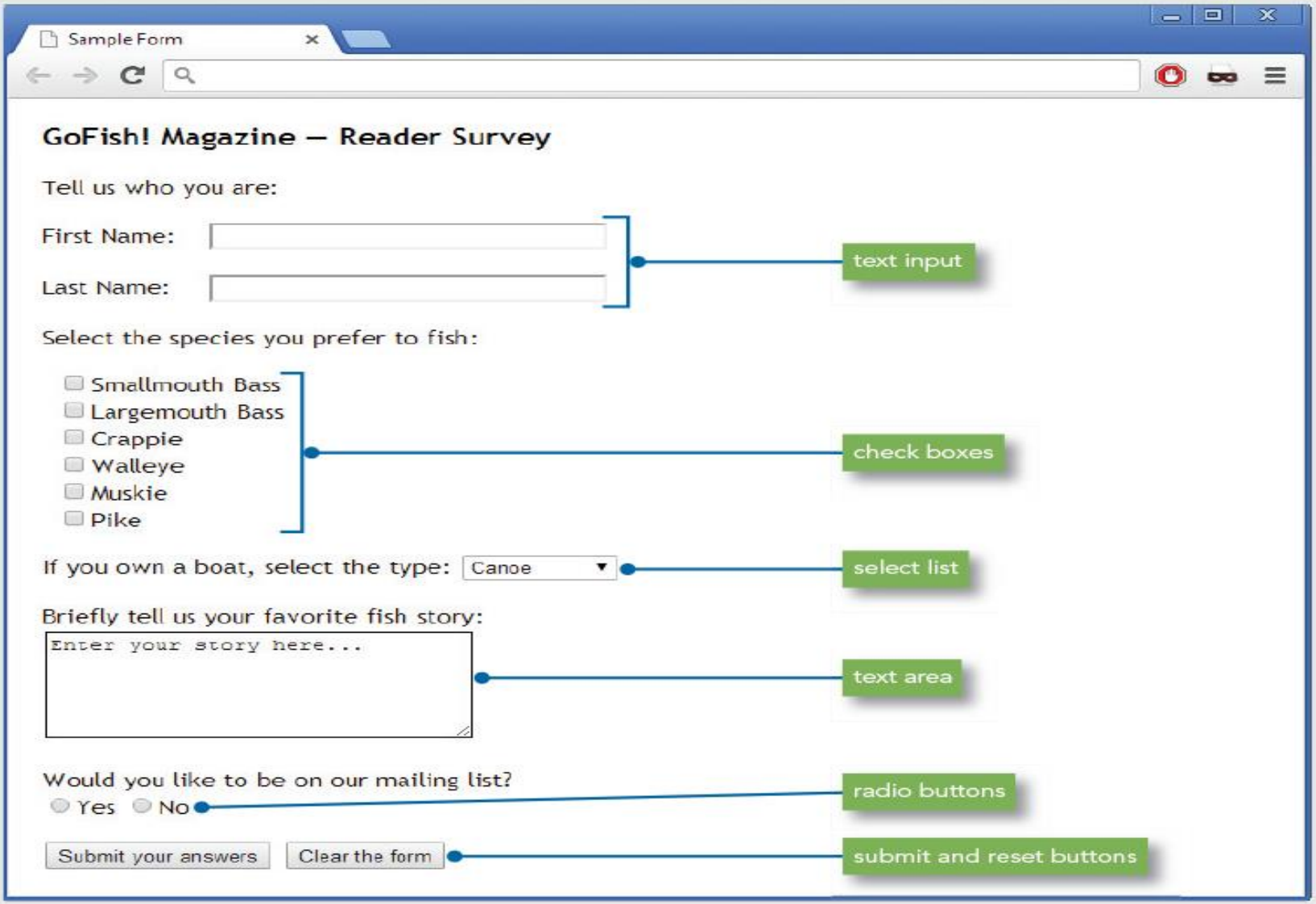


Figure 11-1: Sample HTML form

Using the <form> element (1/2)

- The <form> element is the container for creating a form
- A variety of attributes describe how the form data will be processed

ATTRIBUTE	DESCRIPTION
action	The URL of the application that processes the form data; this URL points to a script file or an email address
enctype	The content type used to submit the form to the server (when the value of the method is *post*); most forms do not need this attribute
method	Specifies the HTTP method used to submit the form data; the default value is *get* get—The form data is appended to the URL specified in the action attribute post—The form data is sent to the server as a separate message
accept	A comma-separated list of content types that a server processing this form can handle correctly; most forms do not need this attribute
accept-charset	A list of allowed character sets for input data that is accepted by the server processing this form; most forms do not need this attribute

Table 11-1: Form Attributes

Using the <form> element (2/2)

- The following code shows a typical <form> element:

```
<form method="post"
 action="https://signup.website.com/register
 .asp">
```

Using get or post

- The difference between *get* and *post* is the way the data is sent to the server
- method="get": this method sends the form information by including it in the URL
- method="post": this method sends the form information securely to the server within the message body

Using the mailto Action

- Lets you collect data from a form and send it to any e-mail address

```
<form action="mailto:joel@joelsklar.com"
method="post" enctype="text/plain">
```

Creating Input Objects (1/2)

- The <input> element defines many of the form input object types
- The type attribute specifies the type of input object

Creating Input Objects (2/2)

INPUT TYPE	DESCRIPTION
radio	Lets a user choose one value from a range of values; when radio buttons are grouped together with the same name, only one choice can be selected
submit	Sends the form data to the server using the transmission method specified in the <form> element; every form needs a submit button
reset	Clears the form of any user-entered data and returns the form to its original state
hidden	Adds a control that is not displayed in the browser; the hidden type is useful for sending additional information with form data that may be needed for processing
image	Adds a graphic button to the form, rather than the default button
button	Creates a button that has no default behavior; the button's function is usually defined by a script; when the user pushes the button, the script function is triggered
file	Lets the user select a file that is submitted with the form

Table 11-2: <input> Element Types

Source: www.w3.org/TR/2011/WD-html5-20110525/the-input-element.html#attr-input-type

INPUT TYPE	DESCRIPTION
range	A range of values. This input type is similar to the number type shown above. The difference is that the range of values is displayed in the browser as a slider control rather than a numeric text entry box. The code looks like this: `<input type="range" min="0" max="10" step="2">`
email	An email address. If the email address is not syntactically correct, some browsers will display an error message to the user. On mobile devices, this input type can change the virtual keyboard to include values that make it easier for the user to enter an e-mail address (for example, including a key for the @ symbol).
url	A URL value; this is an absolute URL, as described in Chapter 3. If the URL address is not syntactically correct, some browsers display an error message to the user. On mobile devices, this input type can change the virtual keyboard to include values that make it easier for the user to enter an URL (for example, including a key for the .com extension).
search	A search term
tel	A telephone number
color	A color name; if supported by the browser, this input type will display a color picker that lets the user choose a color value from a color wheel or color chart

Table 11-3: New HTML5 <input> Element Types

Source: www.w3.org/TR/2011/WD-html5-20110525/the-input-element.html#attr-input-type

Labeling Form Elements (1/2)

- The <label> element lets you create a caption for an input element
- Lets you extend the clickable area of a form element

```
<p>
<label class="username" >First Name:</label>
<input type="text" name="firstname"
size="35" maxlength="35" />
</p>
```

Labeling Form Elements (2/2)

- To make the text clickable, you associate the <label> element with the <input> element by using the *for* and *id* attributes

```
<p>
<label class="username" for="First Name">
First Name:</label>
<input type="text" name="fi rstname" id="First Name"
size="35" maxlength="35" />
</p>
```

Creating Text Boxes

- The text box is the most commonly used form element

```
<input type="text" name="firstname"
size="20" maxlength="35" value="First
Name">
```

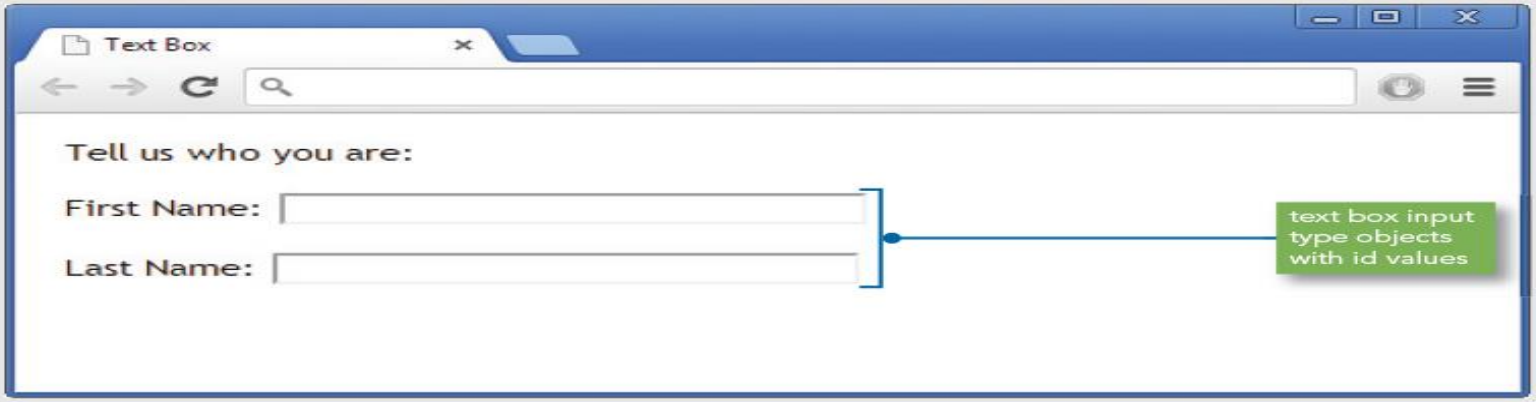


Figure 11-2: Text box input type

Creating Check Boxes

- Check boxes are an on/off toggle that the user can select

```
<input type="checkbox" name="species"
value="smbass"> Smallmouth Bass
```

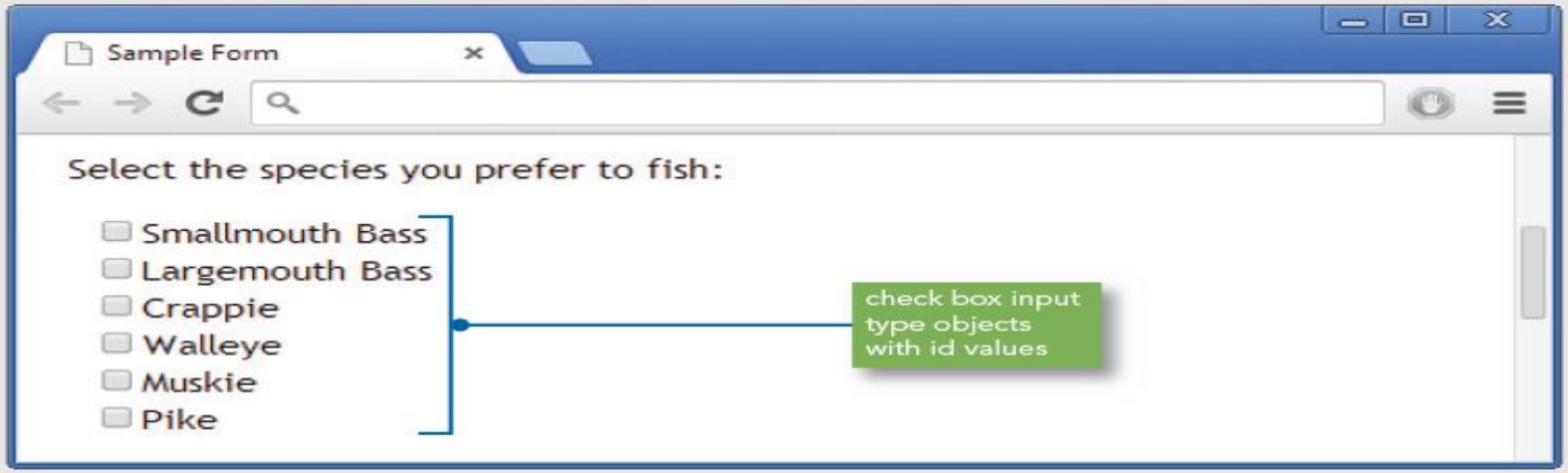


Figure 11-3: Check box input type

Creating Radio Buttons

- Radio buttons are like check boxes, but only one selection is allowed

```
<p>Would you like to be on our mailing list?</p>
<p><input type="radio" name="list" value="yes"
id="Yes" />
<label for="Yes">Yes</label>
<input type="radio" name="list" value="no"
id="No" />
<label for="No">No</label>
</p>
```

Figure 11-4: Radio buttons input type

Creating Submit & Reset Buttons

- The submit and reset buttons let the user choose whether to send the form data or start over

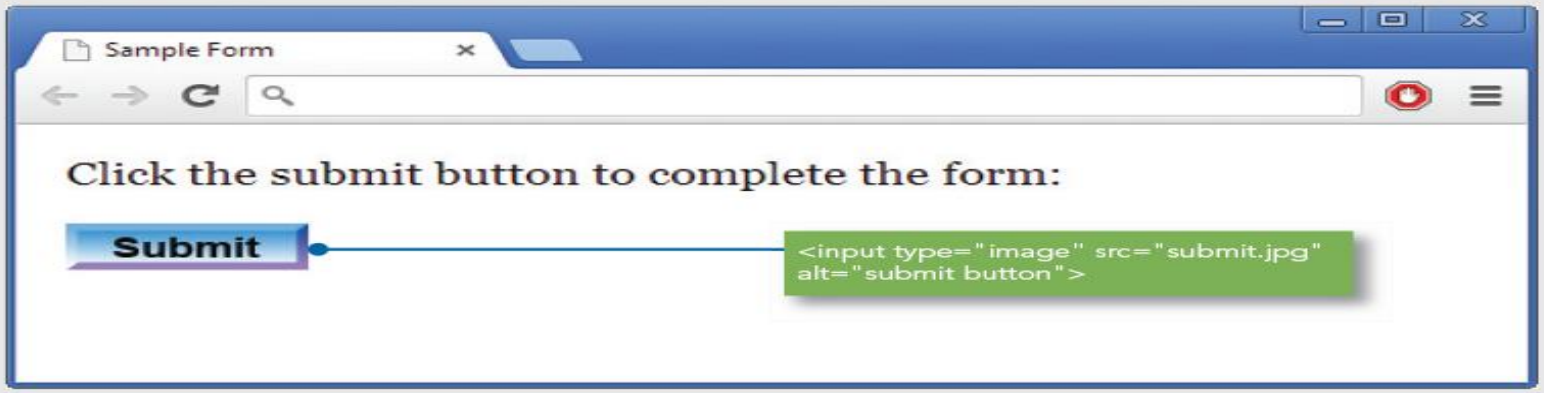
```
<input type="submit" value="Submit your answers">

<input type="reset" value="Clear the form">
```

Figure 11-5: Submit and reset buttons
© 2015 Cengage Learning®

Creating an Image for the Submit Button

- You can choose an image file and use it instead of the default submit button

```
<input type="image" src="submit.gif"
alt="submit button">
```



Letting the User Submit a File

- Users can select a file on their own computer and send it to the server

```
<p>Use the browse button to select your file:</p>

<p><input type="file" size="30“></p>
```

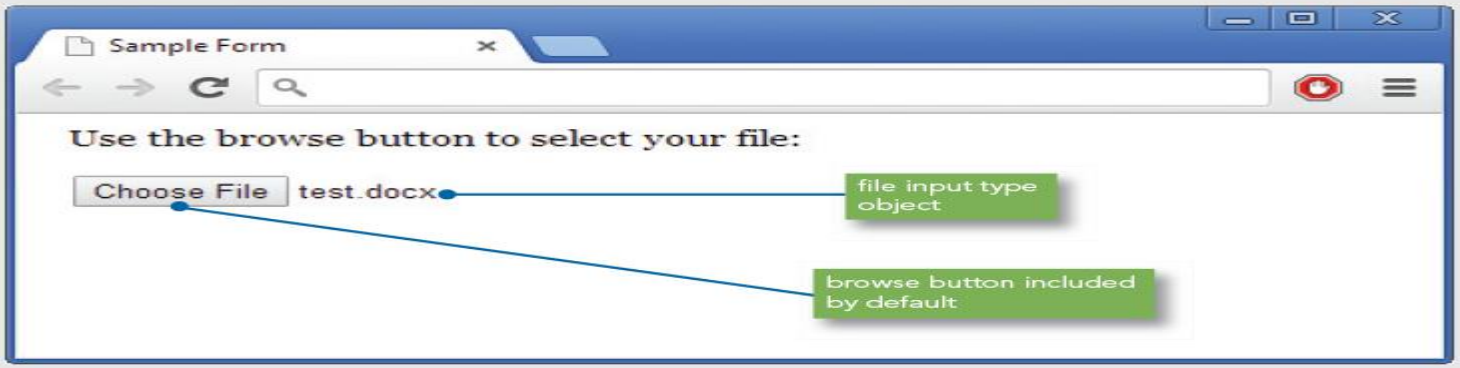


Figure 11-7: File input type lets user select a file

Creating a Password Entry Field

- The password input box works like the text input, but the entered text is hidden by asterisks

```
<p>Enter your user name and password:</p>
<p>
User Name: <input type="text" size="30" />
Password: <input type="password" size="30" />
</p>
```

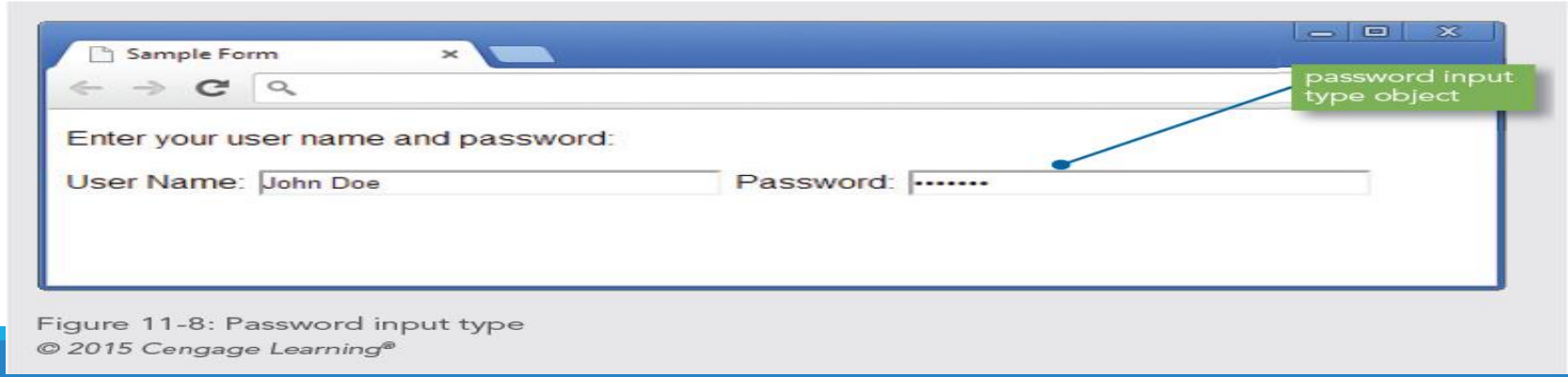


Figure 11-8: Password input type

Creating a Password Entry Field

- The password input box works like the text input, but the entered text is hidden by asterisks

```
<p>Enter your user name and password:</p>
<p>
User Name: <input type="text" size="30" />
Password: <input type="password" size="30" />
</p>
```

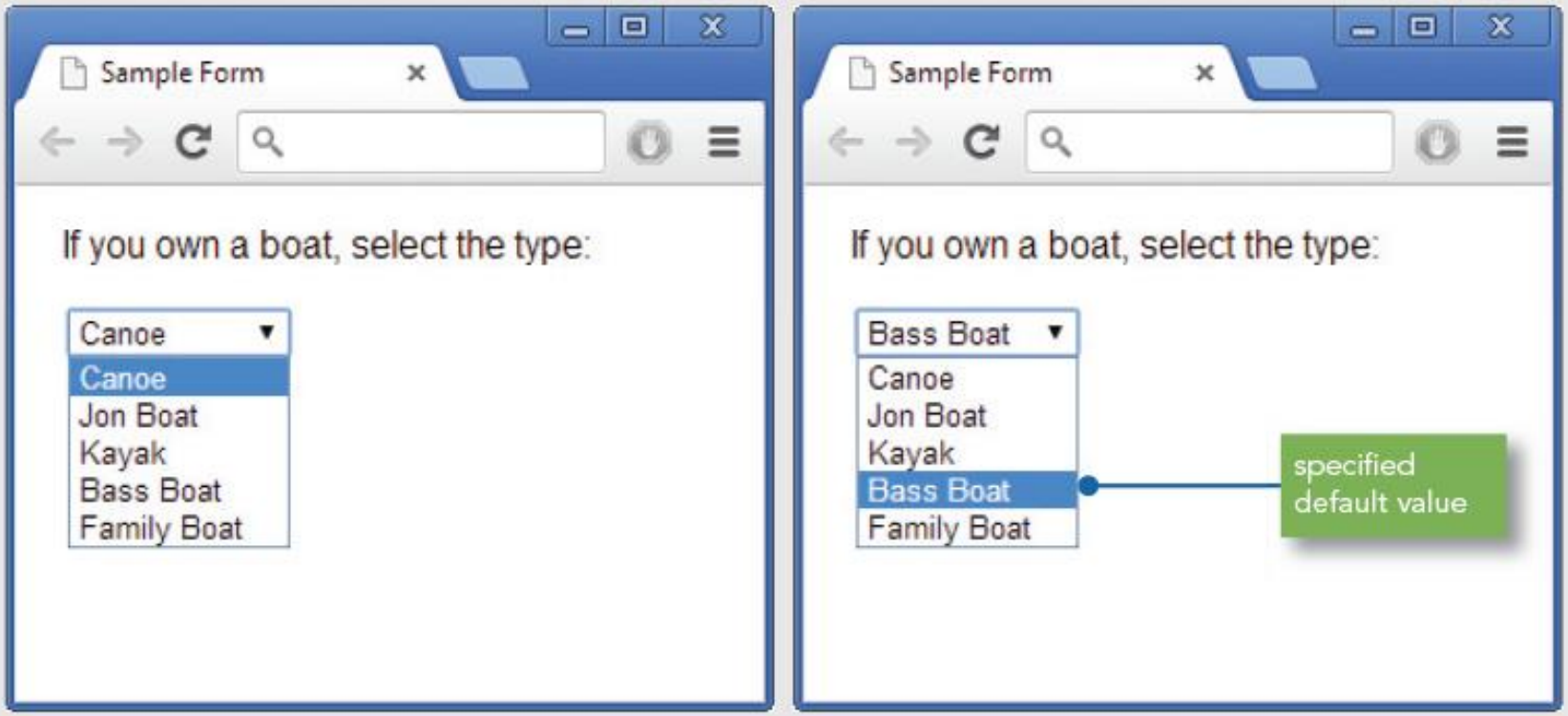


Figure 11-9: Select list element

Using the <select> Element (1/6)

- The <select> element lets you create a list box or scrollable list of selectable options

```
<select name="boats">
   <option>Canoe</option>
   <option>Jon Boat</option>
   <option>Kayak</option>
   <option>Bass Boat</option>
   <option>Family Boat</option>
</select>
```

Using the <select> Element (2/6)

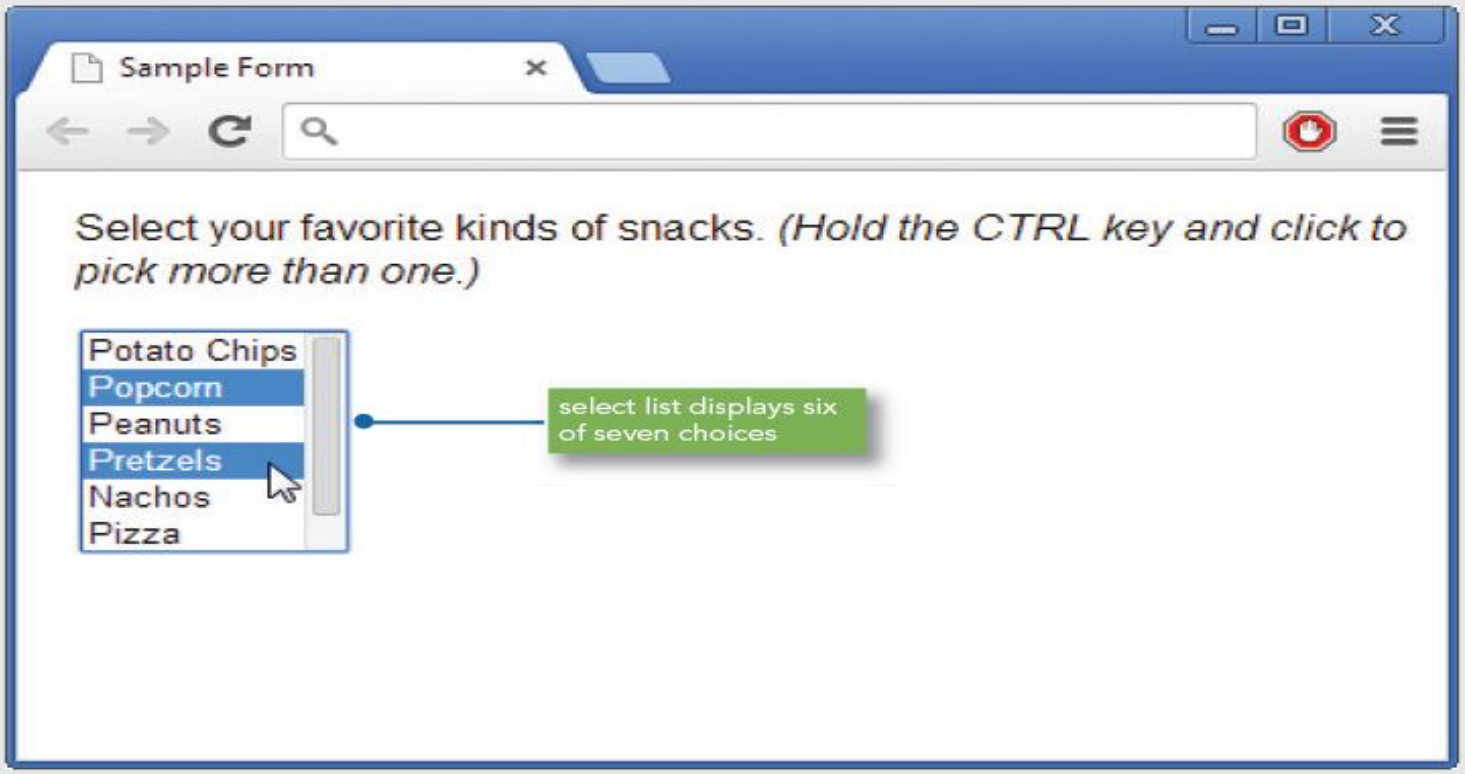


Figure 11-10: Scrollable select list with multiple choices

Using the <select> Element (3/6)

- You can choose to let the user pick multiple values from the list by adding the multiple attribute

```
<select name="snacks" multiple size="6">
   <option>Potato Chips</option>
   <option>Popcorn</option>
   <option>Peanuts</option>
   <option>Pretzels</option>
   <option>Nachos</option>
   <option>Pizza</option>
   <option>Fries</option>
</select>
```

Using the <select> Element (4/6)

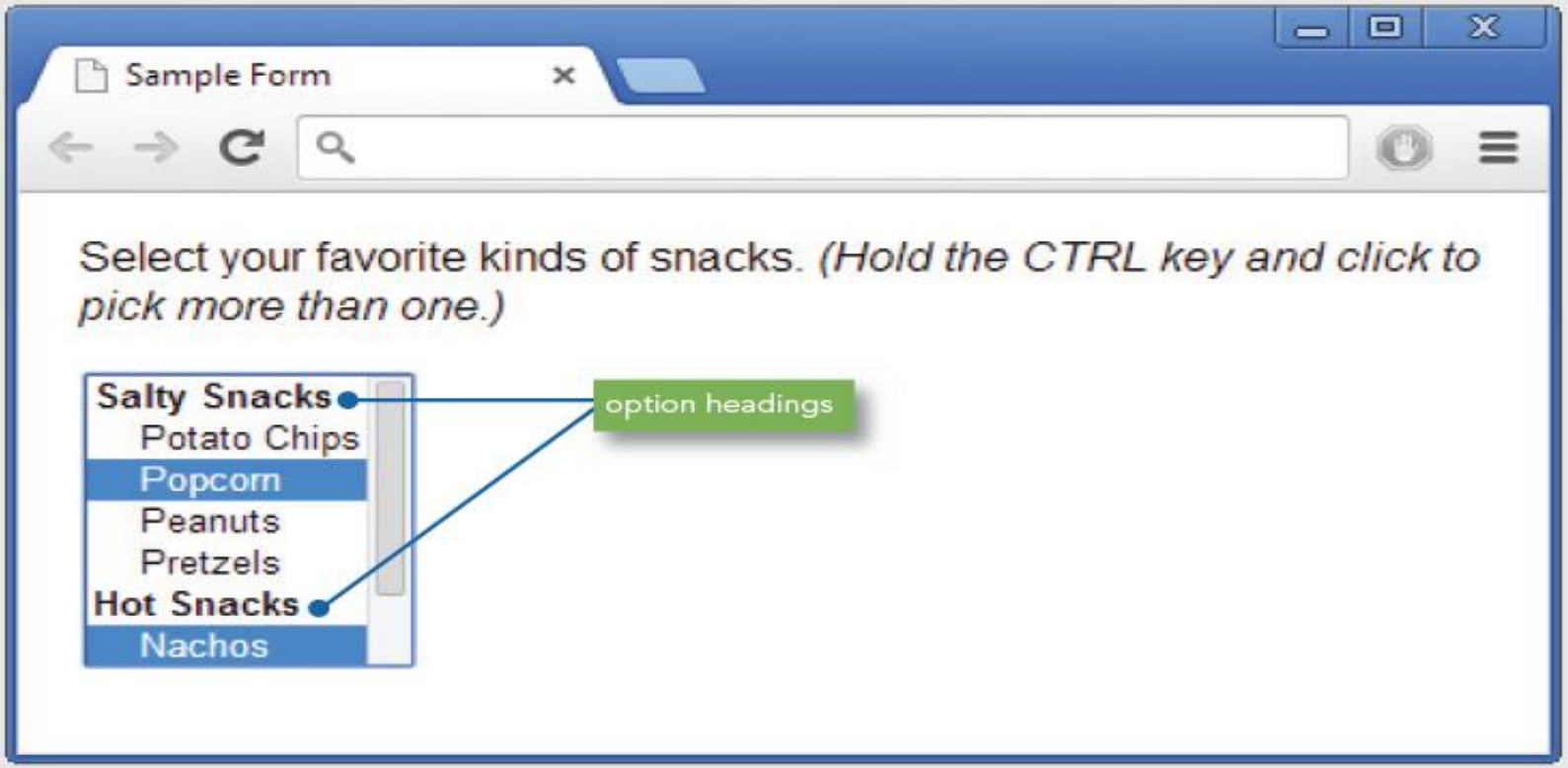


Figure 11-11: Grouping list options

Using the <select> Element (5/6)

- You group and label sets of list options with the <optgroup> element and label attribute

```
<optgroup label="Salty Snacks">
   <option>Potato Chips</option>
   <option>Popcorn</option>
   <option>Peanuts</option>
   <option>Pretzels</option>
</optgroup>
```

Using the <select> Element (6/6)

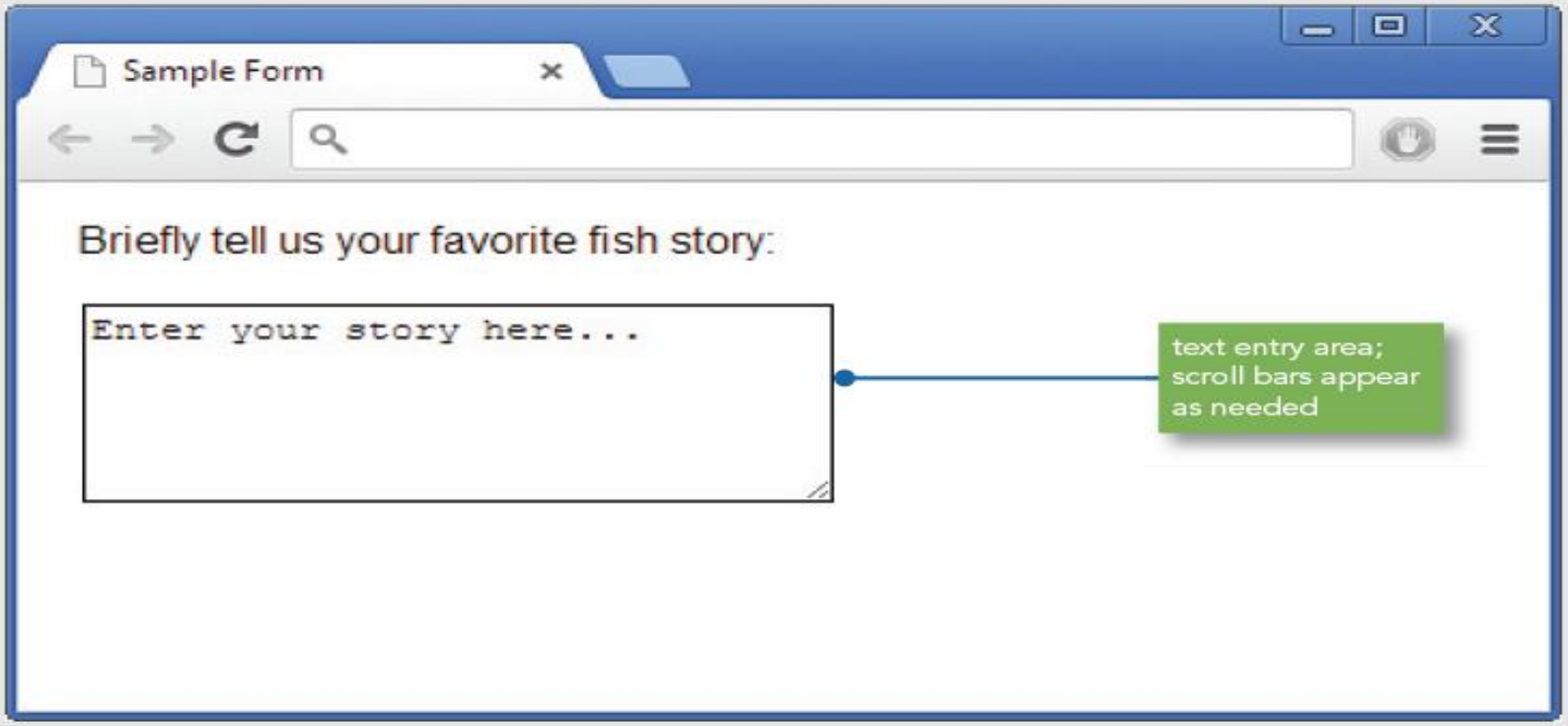


Figure 11-12: Using the <textarea> element

Using the <textarea> Element

- The <textarea> element lets you create a larger text area for user input

```
<p><b>Briefly tell us your favorite fish
story:</b><br>
<textarea name="fishstory" rows="5"
cols="30">
Enter your story here...
</textarea>
</p>
```

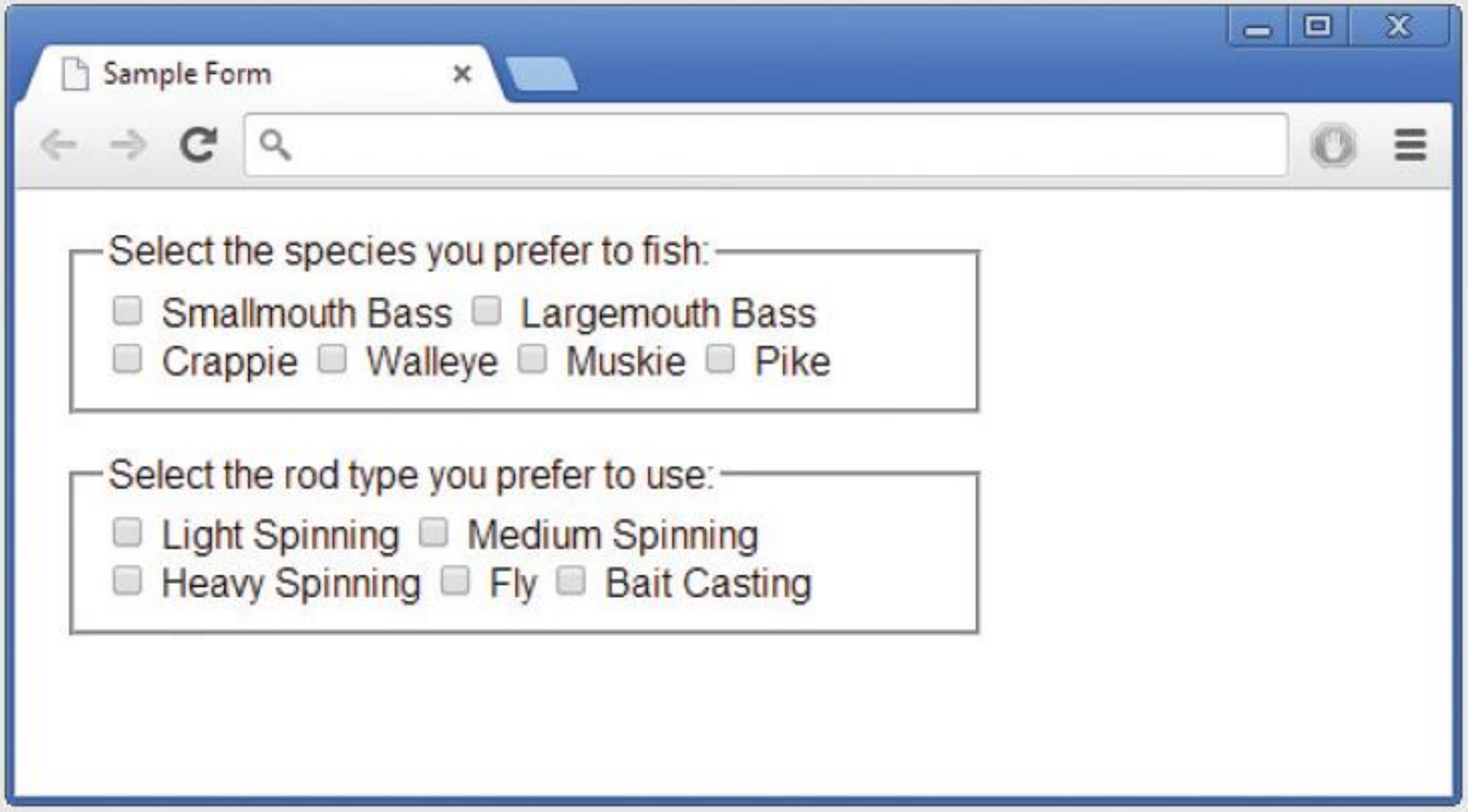


Figure 11-13: Grouping and labeling <input> elements

Creating Input Groupings (1/2)

- You can use the <fieldset> and <legend> elements to create groupings of different types of input elements

```
<fieldset>
<legend><b>Select the species you prefer to
fish:</b></legend>
<input type="checkbox" name="species" value="smbass">
Smallmouth Bass
<input type="checkbox" name="species" value="lgbass">
Largemouth Bass <br>
<input type="checkbox" name="species" value="pike">
Pike
</fieldset>
```

Creating Input Groupings (2/2)

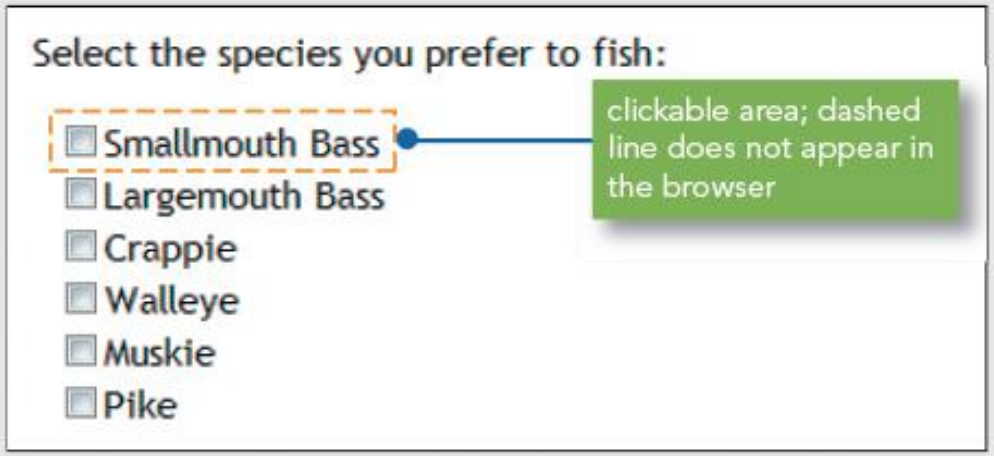


Figure 11-14: The <label> element increases the clickable area

Styling Forms with CSS

USING ADOBE DREAMWEAVER EDIT THE FORMS CSS FOR EACH ELEMENT

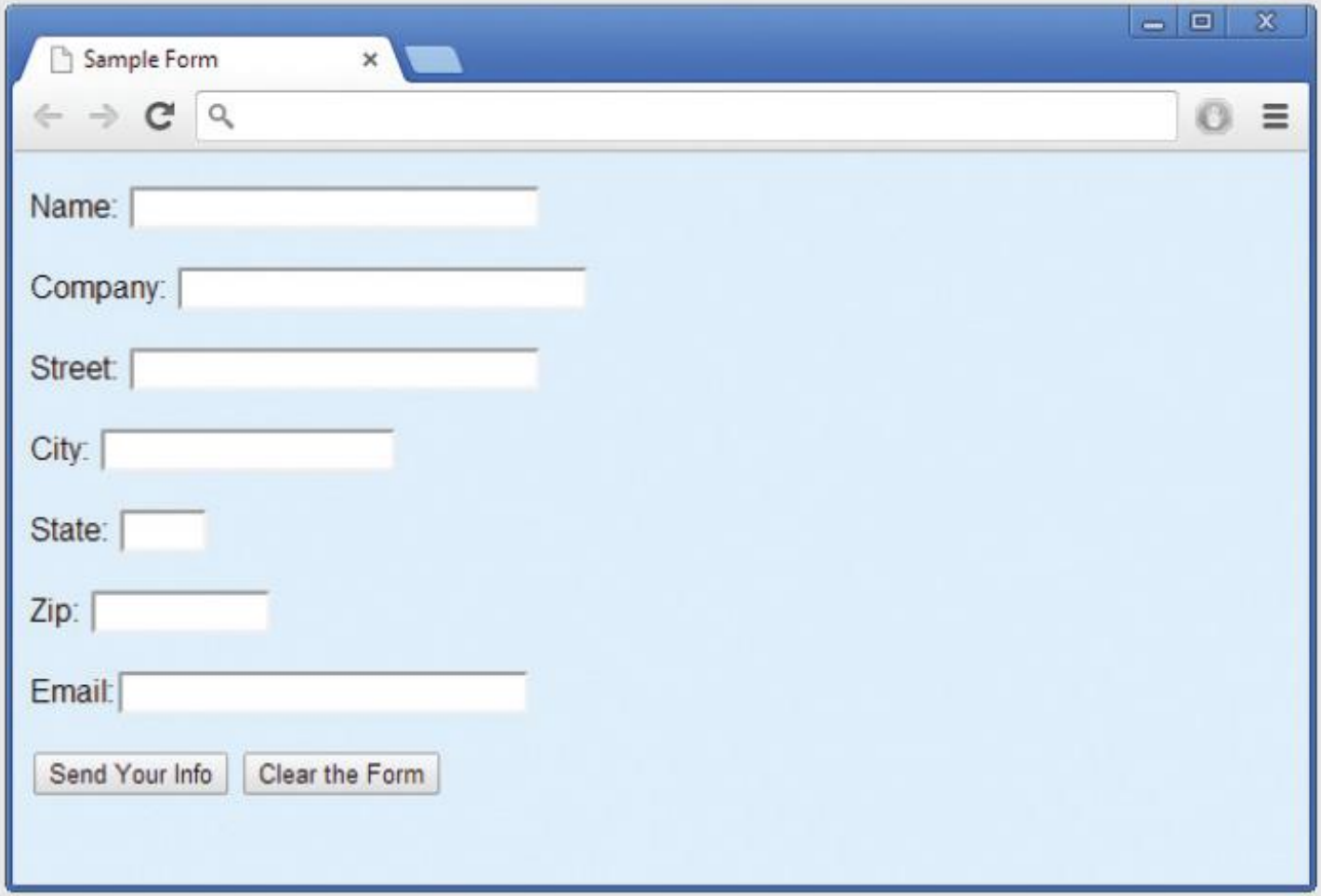


Figure 11-15: Typical form layout

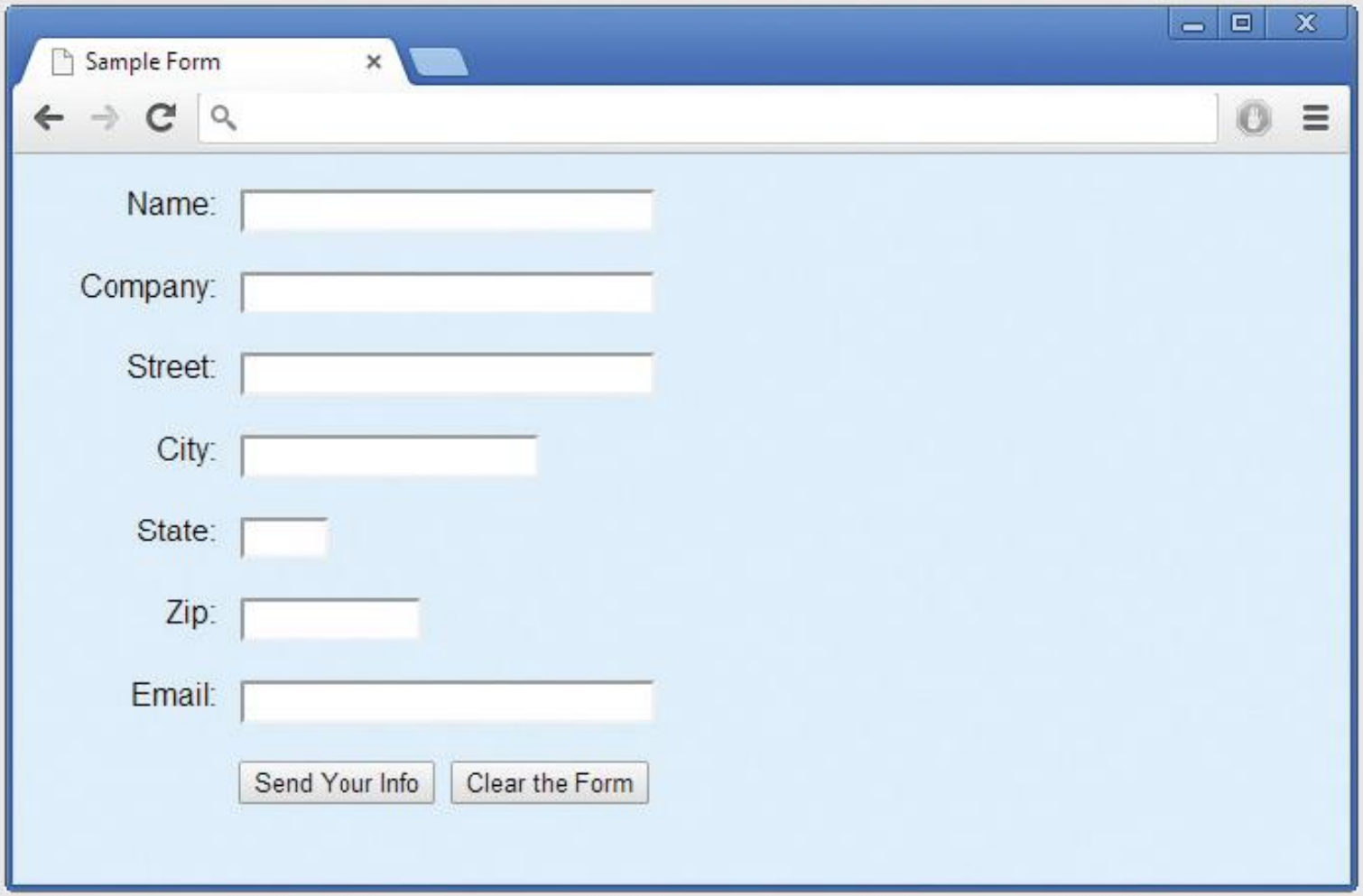


Figure 11-16: Form layout enhanced with CSS

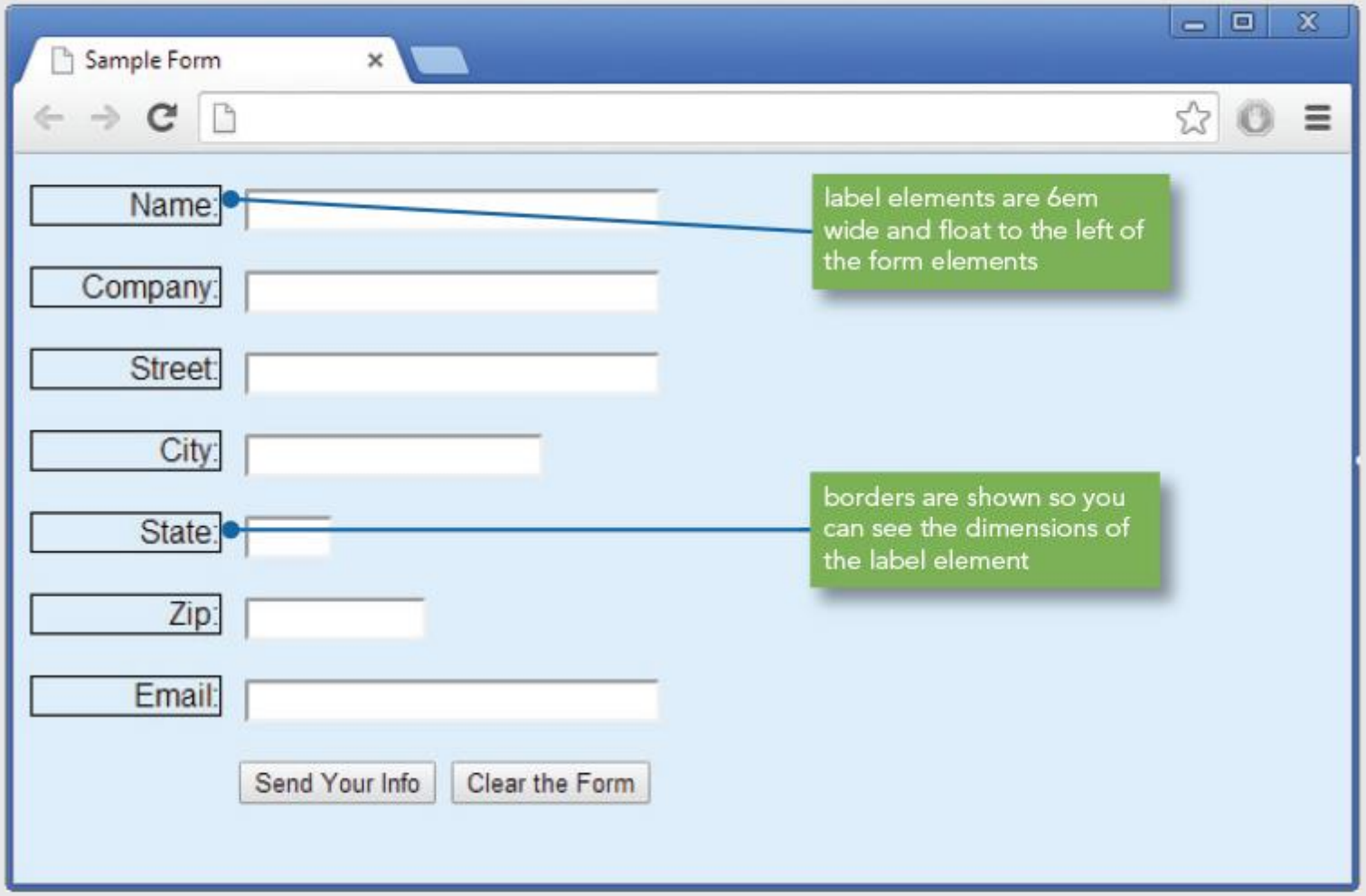


Figure 11-17: Label elements with width, float, and border styles

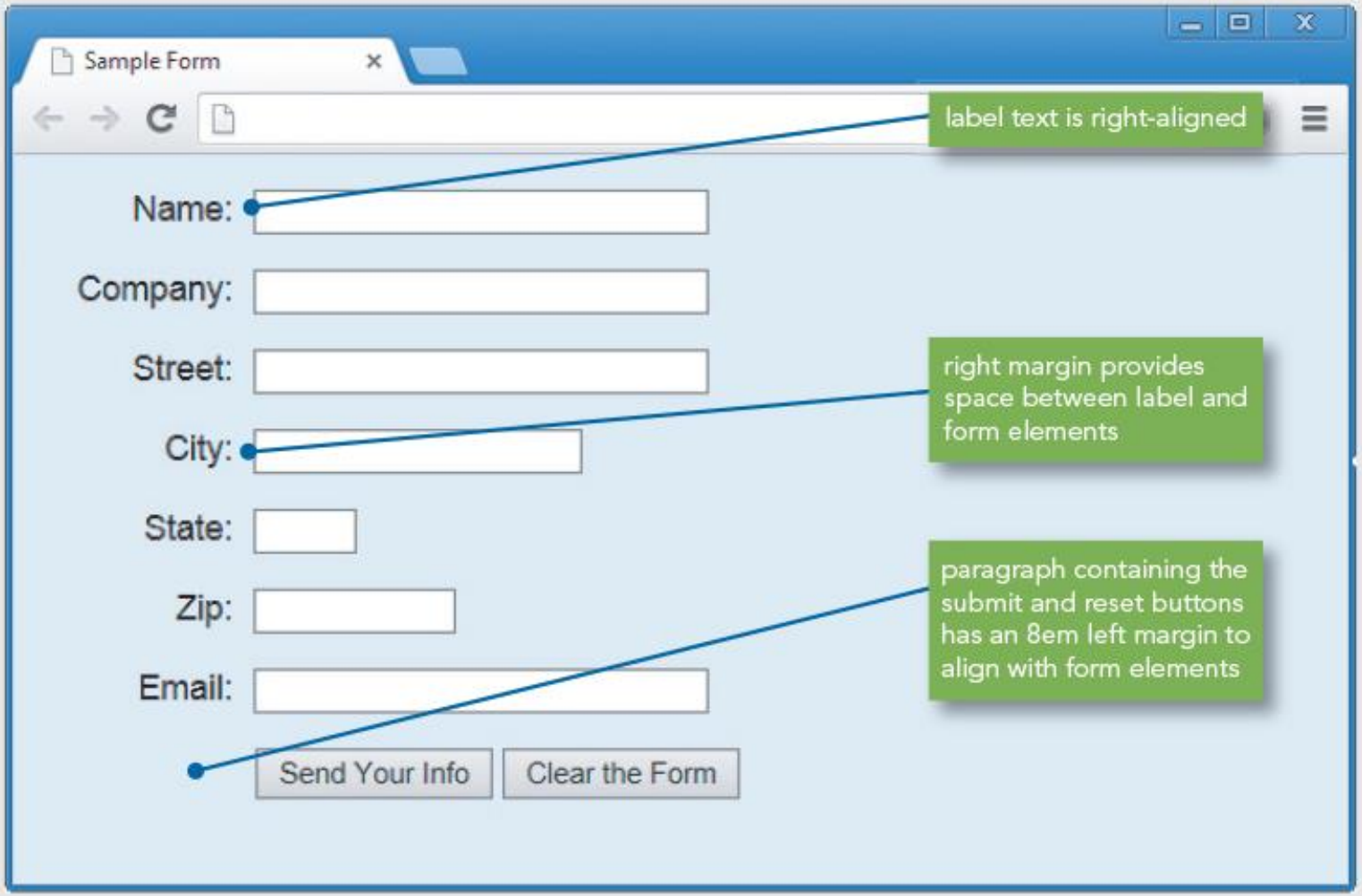


Figure 11-18: Form layout enhanced with CSS

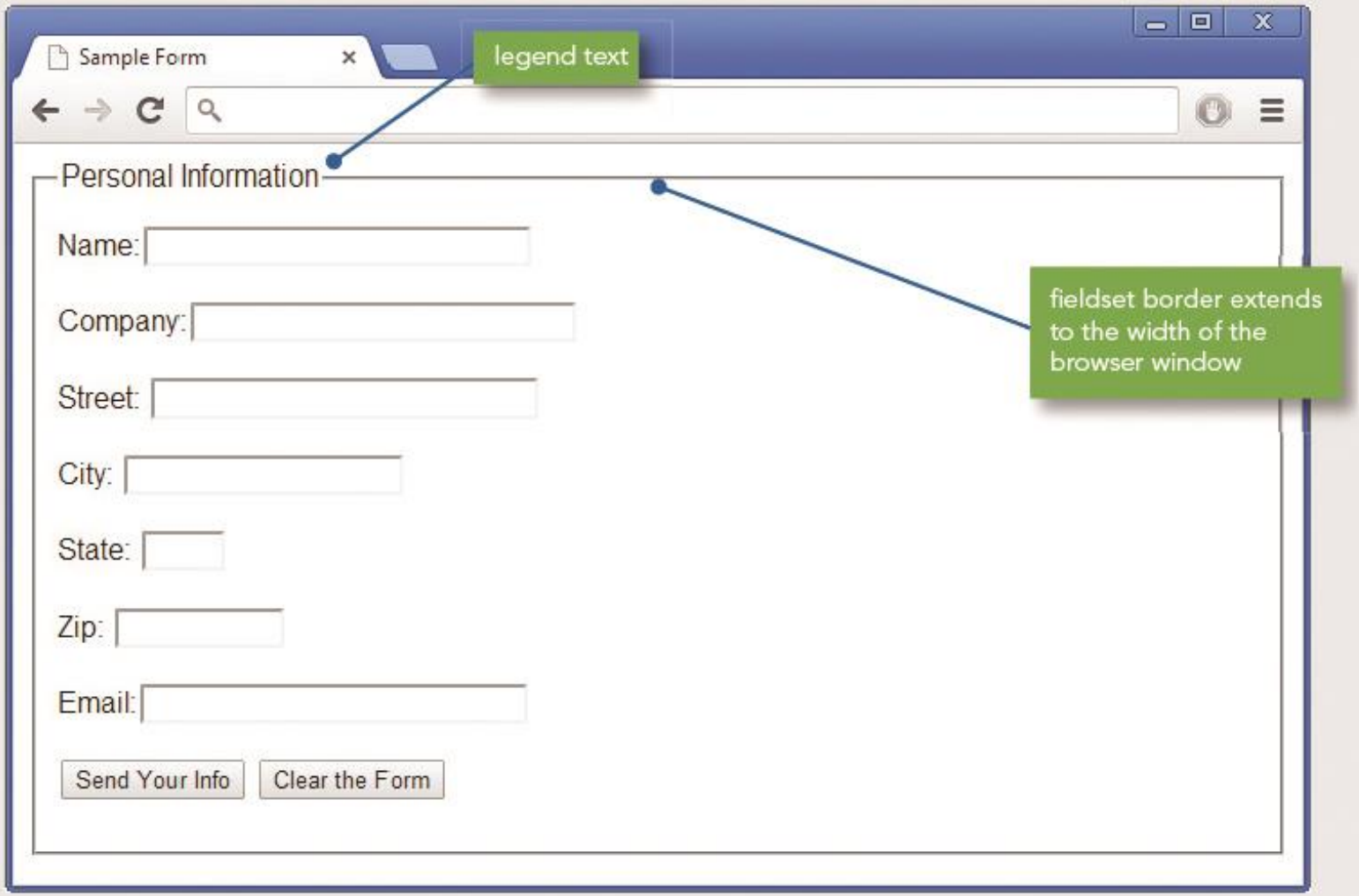


Figure 11-19: Default appearance of <fieldset> and <legend> elements

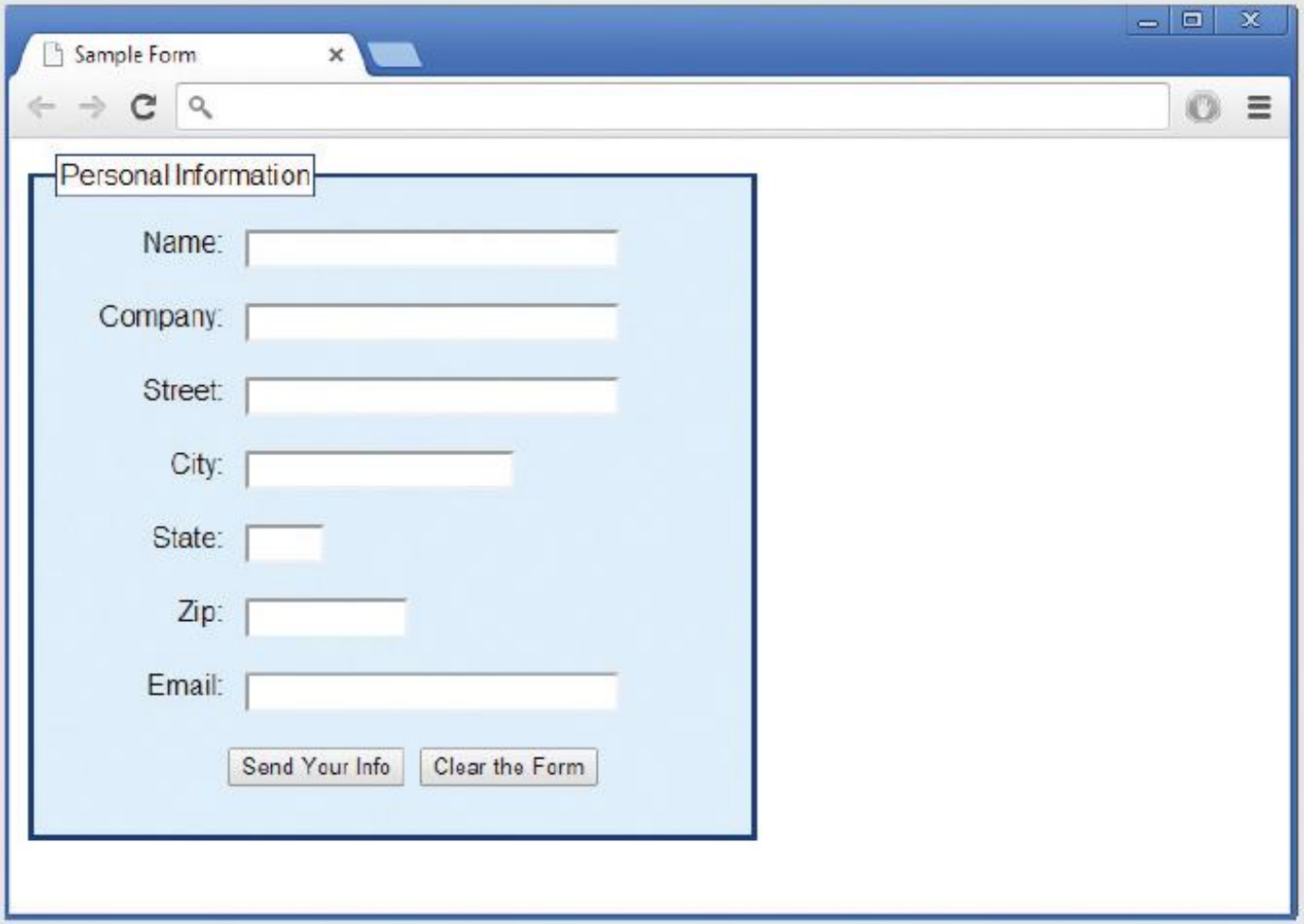


Figure 11-20: Using fieldset and legend style rules

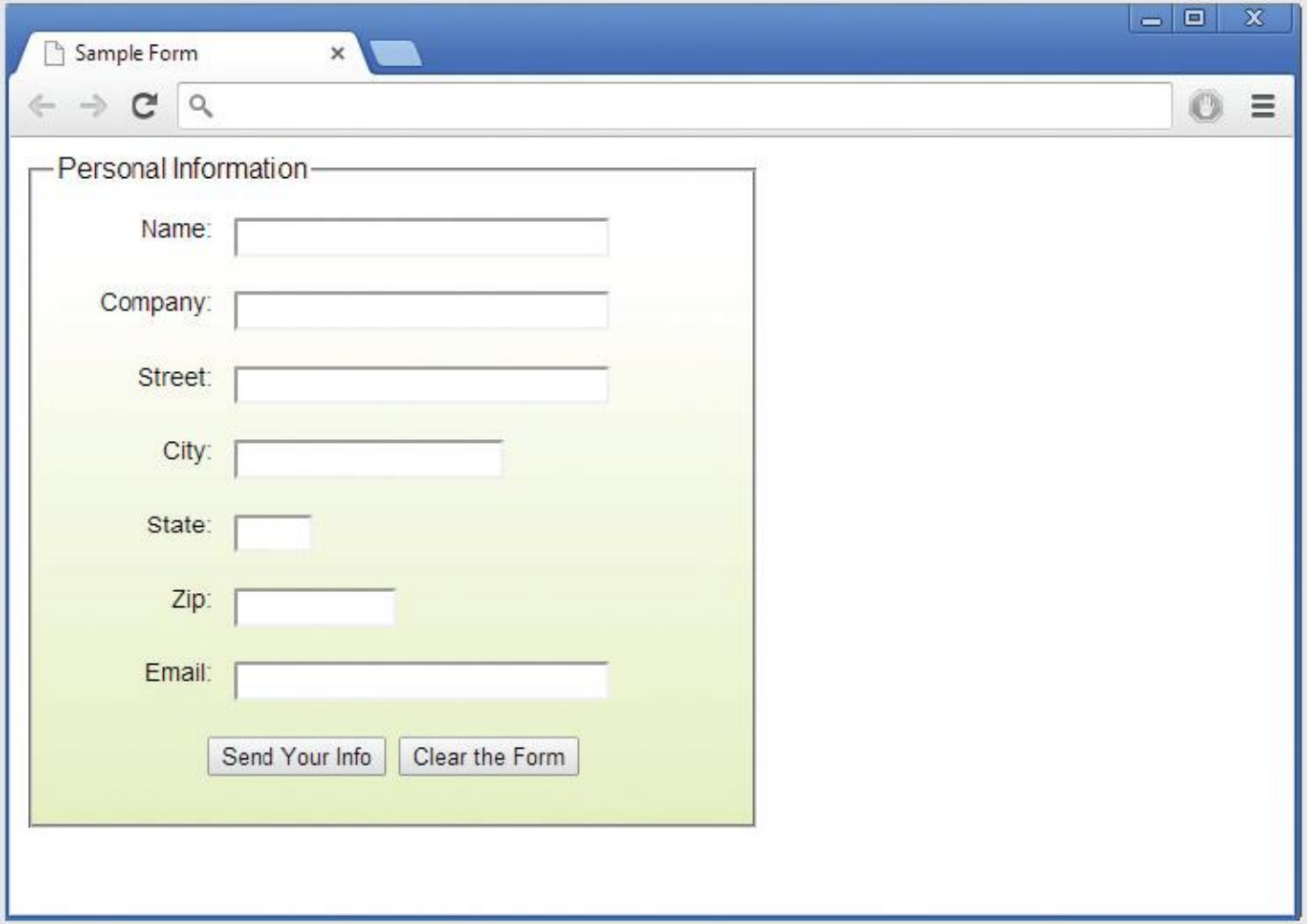


Figure 11-21: Gradient background in a form

Summary

- Choose the right form elements based on the data you want to collect
- A form element has attributes that describe how the form data is processed
- You need a server application to process your form data
- The <fieldset> and <legend> elements let you create more visually appealing forms
- Forms should be formatted to improve their legibility

Thank you