

14 Creating User Forms

In this chapter, you will learn how to

- ✓ Create a basic form.
 - ✓ Create check boxes and option buttons.
 - ✓ Create lists.
 - ✓ Understand CGI and other advanced tools.
-

Gathering feedback from your Web site's visitors can be a valuable way of assessing your site's success, and it can help you build a customer or subscriber database. For example, a business could collect the addresses of people who are interested in receiving product samples, e-mail newsletters, or notifications of special offers.

To receive feedback, you can set up an e-mail hyperlink, as you did in Chapter 5, "Creating Hyperlinks and Anchors," but an e-mail message is not structured, and respondents are given no guidance as to the type of information you want to collect. When you need specific information, such as complete mailing addresses, it is helpful to provide visitors with a form to complete. You can use HTML to create user input forms that can send their results to you in an e-mail message, or you can store the data in a file on your server.

In this chapter, you will learn how to create several types of user input forms. You'll set them up to deliver their results to you in an e-mail message because that's the simplest method—and the most convenient for a low-traffic Web site. You'll learn how to create forms with text boxes, option buttons, check boxes, and drop-down lists. You'll also learn about some of the new HTML5 form controls, such as date boxes. At the end of this chapter, you'll find some information and Web resources that can help you create even more advanced forms by using Common Gateway Interface (CGI) scripting and third-party utilities and services.

Caution Many Web design professionals strongly recommend against using e-mail to deliver Web form results. Not only is e-mail not secure, but Web forms do not interface very well with some e-mail programs and some older browsers (such as Internet Explorer 3.0). If a visitor's e-mail program or browser does not support Web form submittal, the form won't work, but the visitor won't know until he clicks the Submit button, at which point the form will simply be cleared or a blank message window will open. However, nearly all of the other alternatives to e-mail submission require either programming knowledge or going through an external service. In this chapter, you'll test your forms by using e-mail, but use caution when relying on them for your real-world sites. Be sure to read the section, "Understanding CGI and Other Advanced Tools," at the end of this chapter for alternatives.

See Also Do you need only a quick refresher on the topics in this chapter? See the Key Points at the end of this chapter.

Practice Files Before you can use the practice files provided for this chapter, you need to install them from the book's companion content page to their default locations. See "Using the Practice Files" in the beginning of this book for more information.

Creating a Basic Form

You can place a form anywhere in the body of an HTML document. Some people like to use a table to organize form elements; others create form fields within ordinary paragraphs. A form is enclosed in a two-sided `<form>` tag:

```
<form method="post">
...
</form>
```

The *method* attribute specifies what will happen when the form is submitted. Almost all forms use the *method="post"* attribute, meaning that the data users enter into the form will be collected and either delivered (by e-mail) or sent to the server, where server-side code can retrieve it and perform tasks such as storing it (in a database).

Within the opening `<form>` tag, you specify an *action* attribute. This is typically either an e-mail address to which to send information, or the URL of a script that will run when the user submits the form. For an e-mail delivery, the *action* attribute might look like this:

```
<form action="mailto:edward@contoso.com" enctype="text/plain">
```

Note The *enctype* attribute specifies how the results will be encoded. An encoding type of *text/plain* is required when sending result by using e-mail; otherwise, the results might be unreadable.

To send the form contents to a CGI script, you include the URL for the appropriate CGI script stored on your server, as shown here:

```
<form action="http://www.contoso.com/cgi-bin/feedback.pl">
```

You place the various tags that create form controls between the opening and closing `<form>` tags. Form controls available include text boxes, buttons, check boxes, lists, and/or command buttons. A **command button** is a button that executes a function, such as submitting the form or resetting it.

Creating a Text Box

The most basic type of control is a text box. Users can enter data such as names, addresses, phone numbers, and comments into text boxes. There are two types of text boxes: regular text boxes (single line) and text areas (multi-line). In the following figure, the top four fields are single-line text boxes, while the Comments box is a multi-line text area.

First Name: Last Name:
 City: State:
 Comments:

You create a regular text box using a single-sided `<input>` tag with a `type="text"` attribute, as shown in the following:

```
<input type="text">
```

Note Remember, if you are writing XHTML-compliant code, you must place a space and a forward slash / at the end of single-sided tags, like this: `<input type="text" />`. That's not necessary if you are creating HTML code, so you won't see it in the examples presented in this book.

Each control within a form must have a unique name, expressed with the `name` attribute. For example, to name a particular text box *firstname*, use the following:

```
<input type="text" name="firstname">
```

You can specify a width for the text box with the *size* attribute. The default width is 20 pixels.

```
<input type="text" name="phone" size="30">
```

You can also specify a maximum length for the text string that users enter into the text box. This is different from the size of the text box. If the specified maximum length is greater than the text box width, the text scrolls horizontally as the user types. When users reach the specified maximum number of characters, the text box does not accept any more input. Use the *maxlength* attribute like this:

```
<input type="text" name="phone" size="30" maxlength="100">
```

In HTML5, you can require users to fill out a field before they will be able to submit the form (applies to HTML5-compliant browsers only). To mark a field as required, add the *required* attribute to its tag, like this:

```
<input type="text" name="phone" size="30" maxlength="100" required>
```

Special Field Types for E-Mail and Web Addresses

Two new *input* field types in HTML5 support e-mail addresses and Web addresses. Use the attribute *type="email"* instead of *type="text"* to define a field designed to collect e-mail addresses. If a browser doesn't support HTML5, the field defaults to a text type, so you don't risk anything by using it.

```
<input type="email" name="email-address">
```

The same goes for Web addresses (also known as uniform resource locators, or URLs). There is a special *type* attribute in HTML5 for them, as shown here:

```
<input type="URL" name="website">
```

In most browsers, you won't notice any difference. One exception is in the Apple iPhone browser, in which a special version of the onscreen keyboard pops up when the user selects an e-mail or Web field. This special keyboard provides dedicated keys for the most common symbols used for typing e-mail addresses and URLs. Other browsers might eventually implement special treatment for these field types, too.

Creating a Text Area

You create a multi-line text area by using a two-sided `<textarea>` tag containing a `rows` attribute that specifies the number of lines of text that the box should accommodate, such as shown in the following example:

```
<textarea name="comments" rows="5"></textarea>
```

You can also include a `columns` attribute that specifies how many characters (each character represents a single column) wide the text area will be. The default is 40 characters.

```
<textarea name="comments" rows="5" cols="60"></textarea>
```

The `columns` attribute affects only the size of the box, not the maximum number of characters that can be entered. You can use the `maxlength` attribute to limit the number of characters a user can enter.

Creating a Submit or Clear Button

You will need to include a Submit button on the form so visitors can send the information to you. *Submit* refers to the button's function, not the wording that appears on the button face. The default button text is *Submit*, but you can use a `value` attribute to display different text on the button. For example, to make the word *Send* appear on the button face, set up the `value` attribute, as shown here:

```
<input type="submit" value="Send">
```

You can also include a Reset button on the form, which allows the user to clear all the fields. Again, use the `value` attribute to change the text on the button.

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

First Name: Last Name:

City: State:

Comments:

Many Web designers find it useful to place form fields in tables to make it easier to align the fields. For example, as shown in the following image, you could place field labels in one column and the actual fields themselves in another. You'll see this type of design in the next exercise.

First Name:

Last Name:

City:

State:

Comments:

Adding Default or Placeholder Text

By default, text boxes and text areas are blank when the form loads. You can optionally place either default or placeholder text in them.

- Default text is regular text that is submitted with the form results as if the user had actually typed it in.
- Placeholder text is “phantom” text that appears as a prompt within the text box but disappears when the user types something else there. If the user chooses to leave that text box blank, nothing is submitted.

Most browsers support the use of default text, even if they do not support HTML5. For a text box, add a *value* attribute to the tag that specifies the default text, as shown here:

```
<input type="text" name="country" value="United States of America">
```

For a text area, you should place default text between the opening and closing *<textarea>* tags, like this:

```
<textarea name="comments" rows="5">Great job! Keep up the good work.</textarea>
```

Placeholder text displays only in HTML5-compliant browsers. To use placeholder text, add the *placeholder* attribute, like this:

```
<input type="text" name="country" placeholder="Enter your country here">
```

In this exercise, you will create a simple form with text boxes and text areas in a table.



SET UP Use the *signup.htm* file in the practice file folder for this topic. This file is located in the Documents\Microsoft Press\HTML5 SBS\14Forms\CreatingForms folder. Open the *signup* file in Microsoft Notepad and Microsoft Internet Explorer.

1. Immediately following the opening `<table>` tag, create an opening `<form>` tag that sends results to your own e-mail address. Substitute your address for *youremail*.

```
<form method="post" enctype="text/plain" action="mailto:youremail">
```

2. In the empty `<td>` tag following *Name:*, create a single-line text box.

```
<tr>
<colgroup align="right" valign="top">
  <td>Name:</td>
  <td><input type="text" name="name"></td>
</tr>
```

3. In the empty `<td>` tag following *E-mail address:*, create a single-line text box with a type of *email* and a maximum length of 100 characters.

```
<tr>
  <td>E-mail address:</td>
  <td><input type="email" name="email" maxlength="100"></td>
</tr>
```

4. Add a placeholder for the email field of *Enter your e-mail address*.

```
<tr>
  <td>E-mail address:</td>
  <td><input type="email" name="email" size="30" maxlength="100"
  placeholder="Enter your e-mail address"></td>
</tr>
```

5. In the empty `<td>` tag following *Comments:*, create a six-line text area with a width of 50 characters.

```
<tr>
  <td>Comments:</td>
  <td><textarea name="comments" rows="6" cols="50"></textarea></td>
</tr>
```

6. Add a *placeholder* attribute for the *comments* field of *Enter comments here*.

```
<tr>
  <td>Comments:</td>
  <td><textarea name="comments" rows="6" cols="50" placeholder="Enter
  comments here"></textarea></td>
</tr>
```

7. Save the file, and then refresh Internet Explorer to check your work.

8. Add another row at the bottom of the table, immediately before the closing `</colgroup>` tag. Leave the first cell empty, and in the second cell, place **Submit** and **Reset** buttons, separated by a nonbreaking space:

```
<tr>
  <td></td>
  <td><input type="submit" value="Submit">&nbsp;&nbsp;&nbsp;
    <input type="reset" value="Clear"></td>
</tr>
```

9. Save the file, and then refresh Internet Explorer to check your work.

Sign Up for E-Mail Specials

Name:

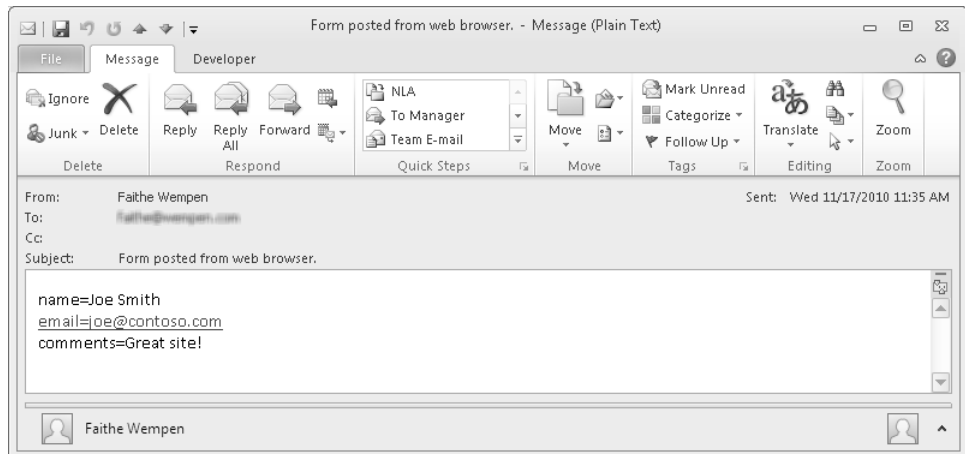
E-mail address:

Comments:

10. In Internet Explorer, enter some text into each field on the form (it doesn't matter what you enter.), and then click the **Submit** button.

Note Depending on your browser and e-mail program settings, you might see a warning message. Respond to these by clicking **Yes** or **OK** to allow the browser to send the message.

11. Check your e-mail inbox for the form results.



Note The speed at which mail servers deliver messages can vary. The results might arrive almost immediately or take an hour or more.

 **CLEAN UP** Close the Notepad and Internet Explorer windows.

Creating Check Boxes and Option Buttons

When the valid responses to a particular prompt will always be one of a few simple options, such as gender or employment status, you will get more consistent and easier-to-evaluate results by using *check boxes* and *option buttons* rather than text boxes.

For example, suppose you are asking site visitors a yes/no question such as, “Are you interested in receiving our catalog by mail?” If you provide a text box, people might answer the question in any number of ways: *y, n, Y, N, yes, no, Yes, No, YES, NO, maybe, Sure, No Thanks*, and so on. Tabulating such results would be tedious because a human would need to evaluate each one. But by providing a check box, you remove any doubt: a check mark means yes; the lack of a check mark means no.

You use *check boxes* for single binary (yes/no) questions. A form might contain multiple check boxes, but each one is a separate decision for the person filling out the form.

What topics would you like to read about?

Home repair

Gardening

Child care

To create a check box, use the *type="checkbox"* attribute with the *<input>* tag, such as in the following:

```
<input type="checkbox" name="repair">
```

By default, the results of the form will show a value of *On* when the check box has been selected. For the check box just shown, the results would appear like this:

```
repair=on
```

You can change that default by specifying a *value* attribute. For example, you could report the word *Yes* for the check box, as shown here:

```
<input type="checkbox" name="repair" value="Yes">
```

By default, check boxes appear unselected; users must click each check box to select it. In some cases, however, it might be advantageous to have a check box preselected. For example, to encourage people to sign up for your newsletter, you could select the Newsletter check box by default, so that users must click it to clear it. To do this, add the `checked="checked"` attribute, as in the following tag:

```
<input type="checkbox" name="newsletter" checked="checked">
```

Use *option buttons* (also called *radio buttons*) to present a group of mutually-exclusive options. When you select an option button, all the other option buttons in the group are cleared.

When will you be buying a new or used car?

- Immediately
- Within 6 months
- Within 1 year
- Not sure
- Do not plan to purchase

To create a group of option buttons, choose a name for the group. You will specify the same name in the `name` attribute for each individual button. Use the `value` attribute (which will be different for each button in the set) to specify the value that will be reported for the group in the form results.

For example, suppose you want users to choose among three membership categories: Gold, Silver, and Bronze. Because you make the most money on a Gold membership, you want to make it the default choice.

```
<p><input type="radio" name="category" value="gold" checked="checked"> Gold<br>
<input type="radio" name="category" value="silver"> Silver<br>
<input type="radio" name="category" value="bronze"> Bronze</p>
```

Each button is followed by text describing that option (Gold, Silver, Bronze). This is just ordinary text.

- Gold
- Silver
- Bronze

Note The space before the text is inserted by default to prevent the option buttons from running into the text. You don't need to add any space yourself.

When the form results are returned, this button group will report its name and the selected value like this:

```
category=gold
```

In this exercise, you will enhance a form by adding a group of option buttons and a check box.



SET UP Use the *signup.htm* file in the practice file folder for this topic. This file is located in the Documents\Microsoft Press\HTML5 SBS\14Forms\CreatingButtons folder. Open the *signup* file in Notepad and Internet Explorer.

1. In the cell after the one that contains *Level of gardening expertise:*, create a set of option buttons that allow the user to choose among Beginner, Intermediate, Expert, or Professional.

```
<tr>
  <td>Level of gardening expertise:</td>
  <td>
    <input type="radio" name="level" value="Beginner">Beginner<br>
    <input type="radio" name="level" value="Intermediate">Intermediate<br>
    <input type="radio" name="level" value="Expert">Expert<br>
    <input type="radio" name="level" value="Pro">Professional<br>
  </td>
</tr>
```

2. Save the file, and then refresh Internet Explorer to see the results.
3. Insert a check box to the left of the *Yes, I would also like...* text, and set its default value to **checked**.

```
<tr>
  <td></td>
  <td><input type="checkbox" name="partner" value="Yes"
checked="checked">Yes, I would also like to receive coupons and offers from
other gardening-related companies.</td>
</tr>
```

4. Change the *mailto* address in the opening *<form>* tag to your own e-mail address.
5. Save the file, and then refresh Internet Explorer to see the results.

The screenshot shows a web browser window with the address bar displaying 'C:\Users\Faithelaptop\Documents\...' and the page title 'The Garden Company'. The page features a header with the company name and tagline 'Helping you help your gardens grow since 1975'. Below the header are navigation buttons for Home, Tip & Tricks, Fix Problems, Products, About Us, and Contact Us. The main content area is titled 'Sign Up for E-Mail Specials' and contains a form with the following fields:

- Name:
- E-mail address:
- Level of gardening expertise:
 - Beginner
 - Intermediate
 - Expert
 - Professional
- Yes, I would also like to receive coupons and offers from other gardening-related companies.
- Comments:

At the bottom of the form are 'Submit' and 'Clear' buttons. Below the form is a copyright notice: 'Copyright © 2012 The Garden Company™. No material may be reproduced without written permission. [Contact the Webmaster](#)'.

6. Fill out the form (use any text you like, and select any of the option buttons), and then click **Submit** to send it to yourself.

✕ CLEAN UP Close the Notepad and Internet Explorer windows.

Creating Lists

Check boxes are good for yes/no questions, and option buttons are appropriate when there are a few options to choose from, but what if you have a dozen or more choices? Option buttons for that many choices can take up a lot of space onscreen and can overwhelm a Web visitor.

For situations involving many options, consider a *list*, also called a *menu*. A list can contain as many options as needed, yet it takes up very little space on the form.



To create a list, start with a two-sided `<select>` tag. Within it, place each option in its own `<option>` tag. Place the text that you want to appear on the list between the opening and closing `<option>` tags. For example, to create the list just shown, do the following:

```
<p>Color: <select name="colors" size="1">
<option>Red</option>
<option>Blue</option>
<option>Green</option>
<option>Yellow</option>
<option>Pink</option>
<option>Brown</option>
<option>Black</option>
<option>Teal</option>
<option>Beige</option>
</select></p>
```

Note By default, the form results will report the text of the selected option. If you want to make the form report something different, include it in a `value` attribute in the option's opening tag.

A list can be any height you like. In the preceding code, the `size` attribute is set to 1, which creates a drop-down list. If you set the `size` attribute to a larger value, the element renders as a list box instead. If there are more items in the list than will fit in the viewing space, a scroll bar appears automatically at the right side of the box. For example, you might change the opening `<select>` tag in the preceding code to this:

```
<p>Color: <select name="colors" size="5">
```

The result would be a list like this.



If the list's choices fall into categories, you might want to break them up into sections.

Select your printer model:

Inkjet

SuperJet 1400

SuperJet 1405

SuperJet 1405 Plus

Laser

SuperLaser Value Edition

SuperLaser Pro

SuperLaser Plus

To format a list as shown in the previous example, surround the groups of options with an `<optgroup>` tag. Include a `label` attribute that specifies the heading text for each option group. Here's the code for the preceding example:

```
<p>Select your printer model:</p>
<select name="printers" size="1">
  <optgroup label="Inkjet">
    <option>SuperJet 1400</option>
    <option>SuperJet 1405</option>
    <option>SuperJet 1405 Plus</option>
  </optgroup>
  <optgroup label="Laser">
    <option>SuperLaser Value Edition</option>
    <option>SuperLaser Pro</option>
    <option>SuperLaser Plus</option>
  </optgroup>
</select></p>
```

In this exercise, you will add a drop-down list to a form.



SET UP Use the *signup.htm* and *states.txt* files in the practice file folder for this topic. These files are located in the Documents\Microsoft Press\HTML5 SBS\14Forms\CreatingLists folder. Open the *signup* file in Notepad and Internet Explorer.

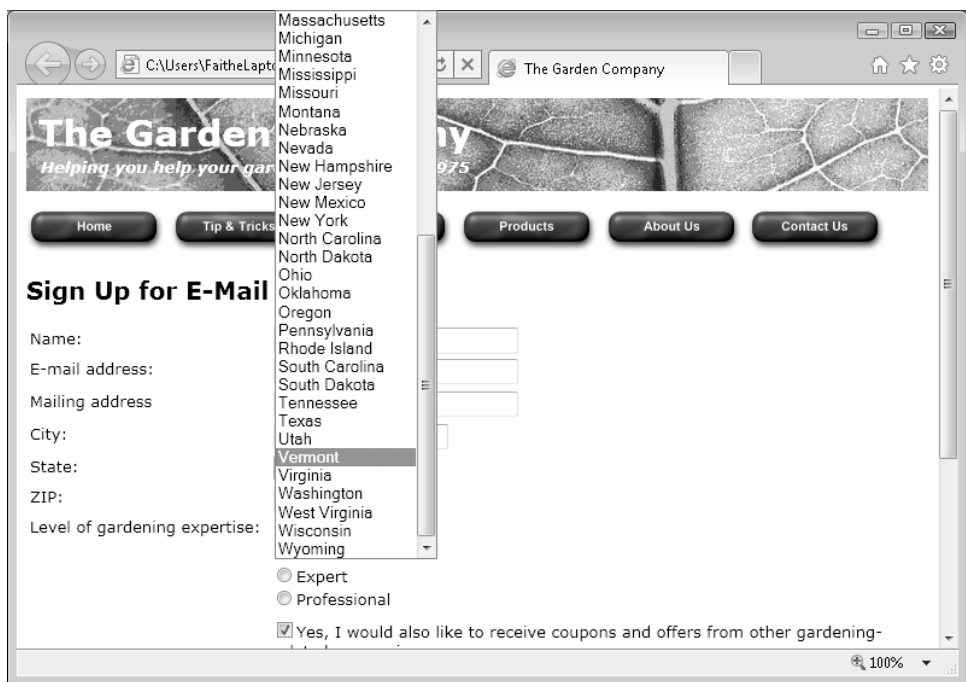
1. In Notepad, replace the `<input>` tag for the text box that follows *State:* with an empty `<select>` tag.

```
<tr>
  <td>State:</td>
  <td>
    <select name="state">

    </select>
  </td>
</tr>
<tr>
```

Note Because it would be time-consuming to type `<option>` tags for all 50 states, I have created them for you.

- In a separate Notepad window, open the `states` text file.
- Press **Ctrl+A** to select the entire content of the file, and then press **Ctrl+C** to copy it to the Clipboard.
- In the `signup` file, click between the opening and closing `<select>` tags. Press **Ctrl+V** to paste the options for each state into the file.
- Save the file, and then refresh Internet Explorer. Click the down arrow to the right of the **State** box to ensure the drop-down list appears.



- Enclose the 50 states in an `<optgroup>` tag with a label of `States`.

```
<optgroup label="States">
  <option>Alabama</option>
  ...
  <option>Wyoming</option>
</optgroup>
```

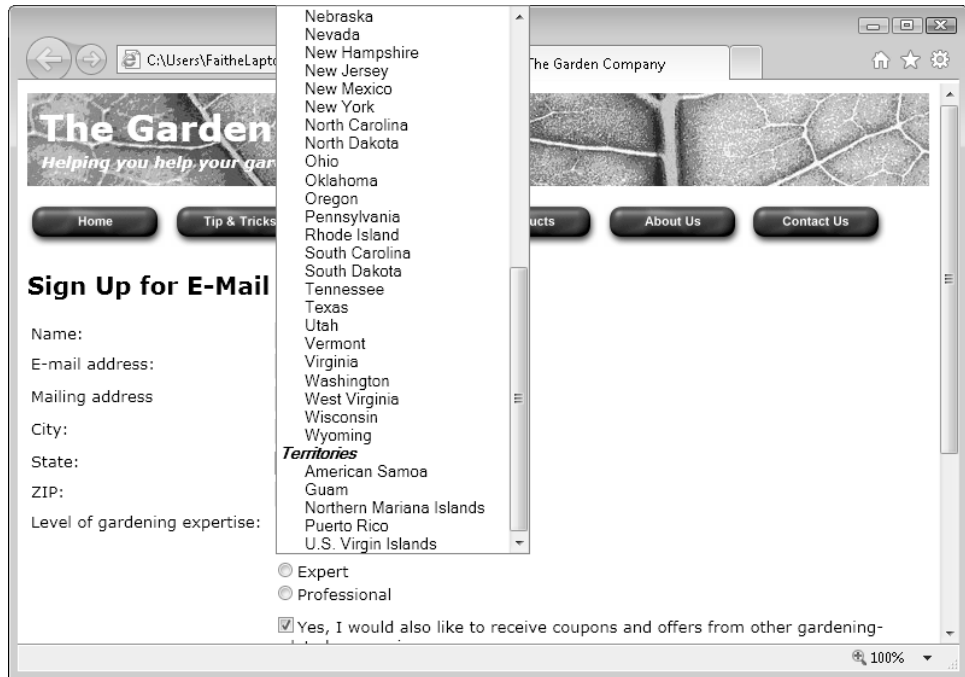
7. After the closing tag of the States option group, add a Territories option group that contains entries for American Samoa, Guam, Northern Mariana Islands, Puerto Rico, and the U.S. Virgin Islands.

```

...
    <option>Wyoming</option>
</optgroup>
<optgroup label="Territories">
    <option>American Samoa</option>
    <option>Guam</option>
    <option>Northern Mariana Islands</option>
    <option>Puerto Rico</option>
    <option>U.S. Virgin Islands</option>
</optgroup>
</select>

```

8. Save the file, and then refresh Internet Explorer. Scroll to the bottom of the **State** drop-down list to see the changes.



✘ CLEAN UP Close the Notepad and Internet Explorer windows.

Additional Input Types in HTML5

HTML5 provides several other field types that can add that extra bit of polish to your forms. If the user's browser doesn't support them, it renders and treats them as text fields, so you can use them freely without worrying about backward compatibility.

Spin boxes are used to increment numeric values.

Copies:

The preceding spin box was created by using the following code:

```
<input type="number" name="copies" min="0"max="100" step="1" value="1">
```

The *min* and *max* attributes control the minimum and maximum values, respectively. The *step* attribute specifies how much the value increments or decrements when you click the up or down arrow buttons. The *value* attribute specifies the default value.

A **slider** is a sliding bar that you can drag from side to side. Its type is *range*, and its attributes are nearly identical to those for a spin box.

Copies:

```
<input type="range" name="copies" min="1"max="4" step="1" value="1">
```

A **date picker** pops up a calendar on which the user can click and select a date. Use the *date* type to get a date picker in HTML5-compliant browsers, as follows:

```
<input type="date">
```

For a standard date, use *type="date"*. This enables the user to select a specific date from a calendar. You can also use any of the following types for other date and time-related selections:

- *Type="month"* selects an entire month
- *Type="week"* selects an entire week
- *Type="time"* selects a time only (no date)
- *Type="datetime"* select both a date and a time
- *Type="datetime-local"* selects both a date and time using the user's local time zone

Understanding CGI and Other Advanced Tools

As you have seen, directing form results to an e-mail address is a quick, no-hassle way of collecting information, provided the visitor's Web browser and e-mail program support it. As the volume of messages increases, however, organizing all the information you receive can become a challenge. You need to copy the information from the form results into a database, or at least print out a copy of the e-mail messages. When you start receiving hundreds of form submissions a day, that responsibility can become overwhelming.

As an alternative, you can rely on a server-based script or application to handle the form results. A Common Gateway Interface (CGI) script written in a programming language such as Perl is one common, low-cost possibility. You reference the script in your `<form>` tag's *action* attribute. (The server on which you host your site must allow CGI scripts (some don't, for security reasons.)

Important One drawback of using CGI scripts is lack of security. Unless you put security measures in place, the collected data resides in a file on the server, which is a potential security risk. For this reason, you shouldn't use a CGI script to collect sensitive information such as credit card or Social Security numbers unless you also implement security measures that prevent the data from being compromised. Most commercial sites use a secure server for that; you can partner with a company that offers secure form processing, including credit card processing, for a monthly fee.

There are hundreds of Web sites that offer free CGI scripts that you can modify in a text editor (such as Notepad) to meet your needs. To do this, you must know a little something about programming, which is beyond the scope of this book. However, if you're interested in learning about Perl and CGI scripting, or you are looking for a service that will host your CGI script, see one of these Web sites:

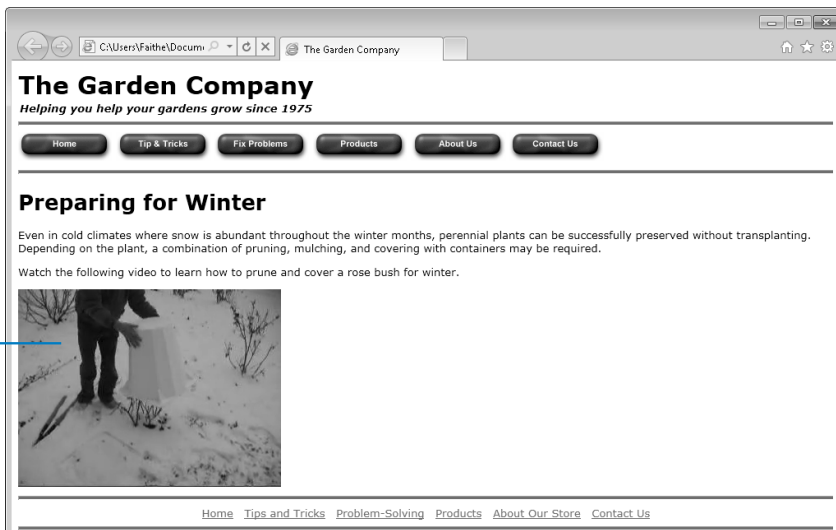
- Comprehensive Perl Archive Network: www.cpan.org
- Matt's Script Archive: www.scriptarchive.com
- The CGI Resource Index: cgi.resourceindex.com
- BigNoseBird.Com: www.bignosebird.com

As your Web site becomes more sophisticated, you also might want to include a public bulletin board area where people can post and read comments, or a guest book where people can leave public comments. You can't create one of those by using only HTML, but you can integrate add-on components into your site that will do the job. There are many free and low-cost sources of programming code for a message board, both in CGI (mentioned previously) and other languages. For example, check out the phpBB open-source bulletin board package at www.phpbb.com (your server must support PHP). There are also many services that will host your bulletin board on their server, such as ProBoards (www.proboards.com). You place a link to the message board hosting site on your Web page; to your visitors, it seems like the message board is part of your site.

Key Points

- To create a form, use a two-sided `<form>` tag. Within it, place one or more `<input>` tags that define the form fields.
- In the opening `<form>` tag, place a `method="post"` attribute and an `action` attribute that describes how the form should be processed. The most common attribute is `action="mailto:address"` where `address` is a valid e-mail address. If you are collecting form results by using e-mail, include an `enctype="text/plain"` attribute.
- To create a text box, use `<input type="text" name="fieldname">`, where `fieldname` is the unique name you assign to the text box. Optional additional attributes include `size` and `maxlength`.
- For Web and e-mail collection, you can optionally use the `URL` and `email` input types, respectively. These work only in HTML5-compliant browsers.
- To create a multi-line text box (a text area), use a two-sided `<textarea>` tag with a `name` attribute and a number of rows and columns. For example, `<textarea name="comments" rows="5" columns="40"></textarea>`.
- To create a Submit button, use an `<input>` tag with a `type="submit"` attribute. If you want to change the button text, use the `value` attribute. For example, `<input type="submit" value="Send">`. Use `type="reset"` to create a Reset button that clears the form.
- A check box is a one-sided, standalone element. Use an `<input>` tag with a `type="checkbox"` attribute.
- An option button operates in a group with other option buttons; only one in a group can be selected at a time. Use a one-sided `<input>` tag with a `type="radio"` attribute. For each option, use a common `name` attribute and a unique attribute.
- To create a list, use a two-sided `<select>` tag, and within it, include two-sided `<option>` tags for each list item.
- Use a `size="1"` attribute with the `<select>` tag to create a drop-down list, or specify a higher number to create a list box with a scroll bar.
- To create category headings on a list, use a two-sided `<optgroup>` tag with a label for the text that should appear. For example, `<optgroup label="Inkjet">`.
- HTML5 offers several other input types for special cases, such as spin boxes (`type="number"`), sliders (`type="range"`), and date pickers (`type="date"`).
- To process form input on a server, use a Common Gateway Interface (CGI) script or a third-party program.

Chapter at a Glance



The screenshot shows a web browser window with the address bar displaying "C:\Users\Faith\Docum... The Garden Company". The website header features the logo "The Garden Company" and the tagline "Helping you help your gardens grow since 1975". Below the header is a navigation menu with buttons for "Home", "Tip & Tricks", "Fix Problems", "Products", "About Us", and "Contact Us". The main content area is titled "Preparing for Winter" and contains the following text:

Even in cold climates where snow is abundant throughout the winter months, perennial plants can be successfully preserved without transplanting. Depending on the plant, a combination of pruning, mulching, and covering with containers may be required.

Watch the following video to learn how to prune and cover a rose bush for winter.

Below the text is a video player showing a person in a snowy environment working with a large white container, likely a mulch or protective covering for a plant. The video player is currently paused.

At the bottom of the page, there is a footer with links for "Home", "Tips and Tricks", "Problem-Solving", "Products", "About Our Store", and "Contact Us".

Play a video,
page 279