

Forms

Web Design in a Nutshell, Third Edition

Chapter 15

Understanding How Forms Work

- a. Forms let you build interactive Web pages that collect information from a user and process it on the Web server
 - b. The HTML form is the interface for the user to enter data
 - c. The data is processed by applications that reside on the Web server
-

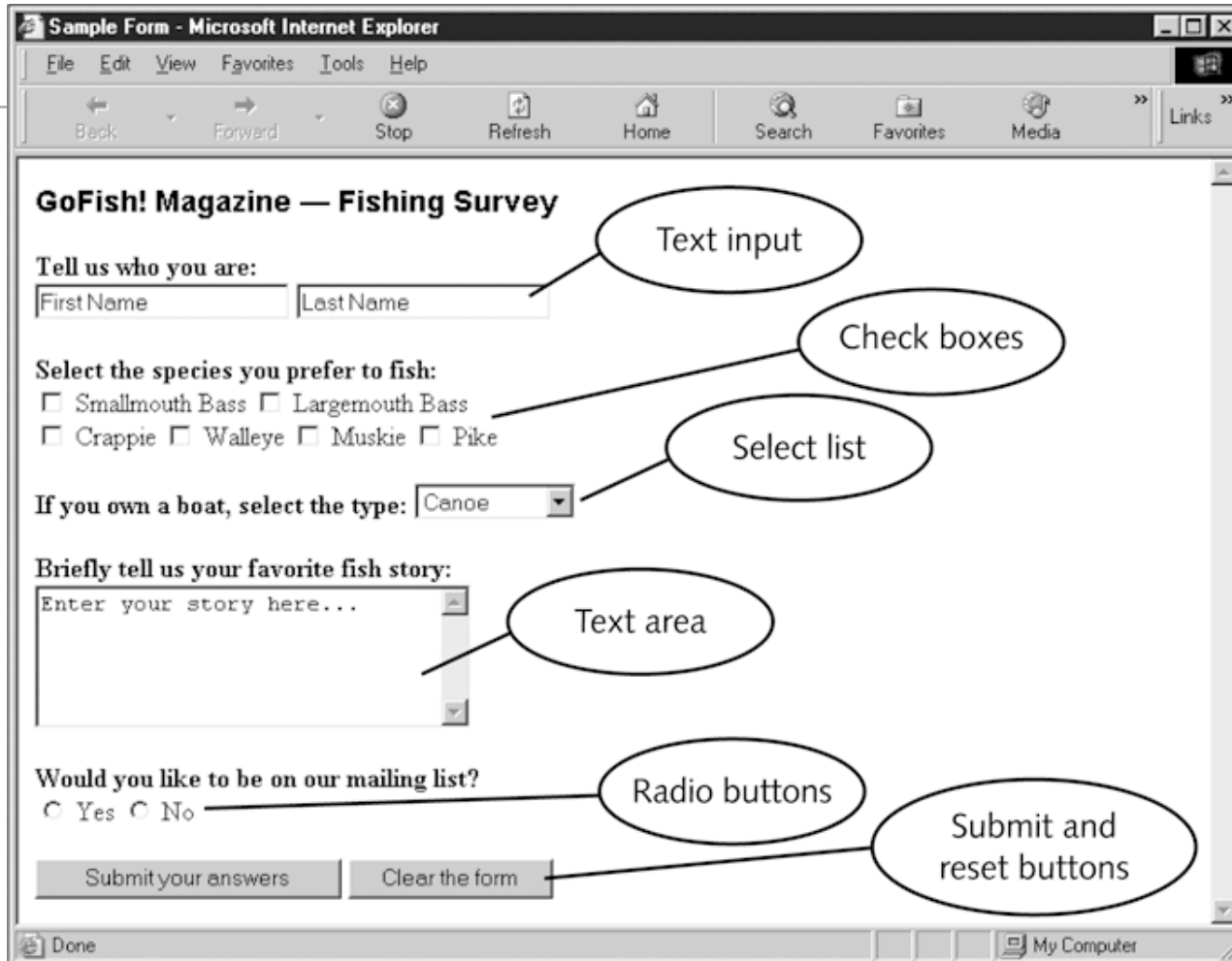


Figure 10-1 A sample HTML form

Using CGI Scripts

- a. The Common Gateway Interface (CGI) is the communications “bridge” between the user's browser and the server

 - b. Using programs called scripts, CGI can collect data sent by a user via the Hypertext Transfer Protocol (HTTP) and transfer it to a variety of data processing programs including spreadsheets, databases, or other software running on the server
-

Form Workflow

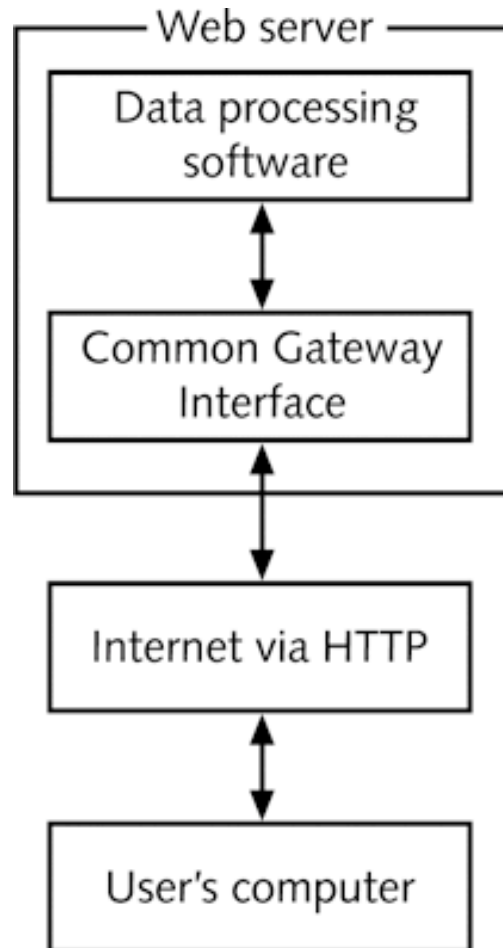


Figure 10-2 Common Gateway Interface architecture

Forms Syntax

a. Five basic form elements are commonly

b. supported by the major browsers:

c. `<form>`

d. `<input>`

e. `<select>`

f. `<option>`

g. `<textarea>`

Using the `<form>` element

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

Form tag

Table 10-1 Form attributes

Attribute	Description
action	The URL of the application that processes the form data; this URL usually points to a CGI script file
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" <ul style="list-style-type: none">• 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

Using the <form> element

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

```
<form action="http://www.website.com/  
cgi_bin/script.cgi" method="post">
```

Creating Input Objects

- a. The `<input>` element defines many of the form input object types

 - b. The `type` attribute specifies the type of input object
-

Table 10-2 <input> element types

Type Attribute Value	Description
text	Creates a text entry field that lets the user enter a single word or a line of text; this is the default object type
password	Creates the same type of text entry field created by the value "text", but the user entry is masked by asterisks
check box	Provides on/off toggles that the user selects. Check boxes are best used with multiple-choice questions. Multiple check boxes can contain the same name, letting you group them together so that users can select multiple values for the same property.
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 it 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 the 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

Creating Text Boxes

a. The text box is the most commonly used form element.

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

Text Box Input

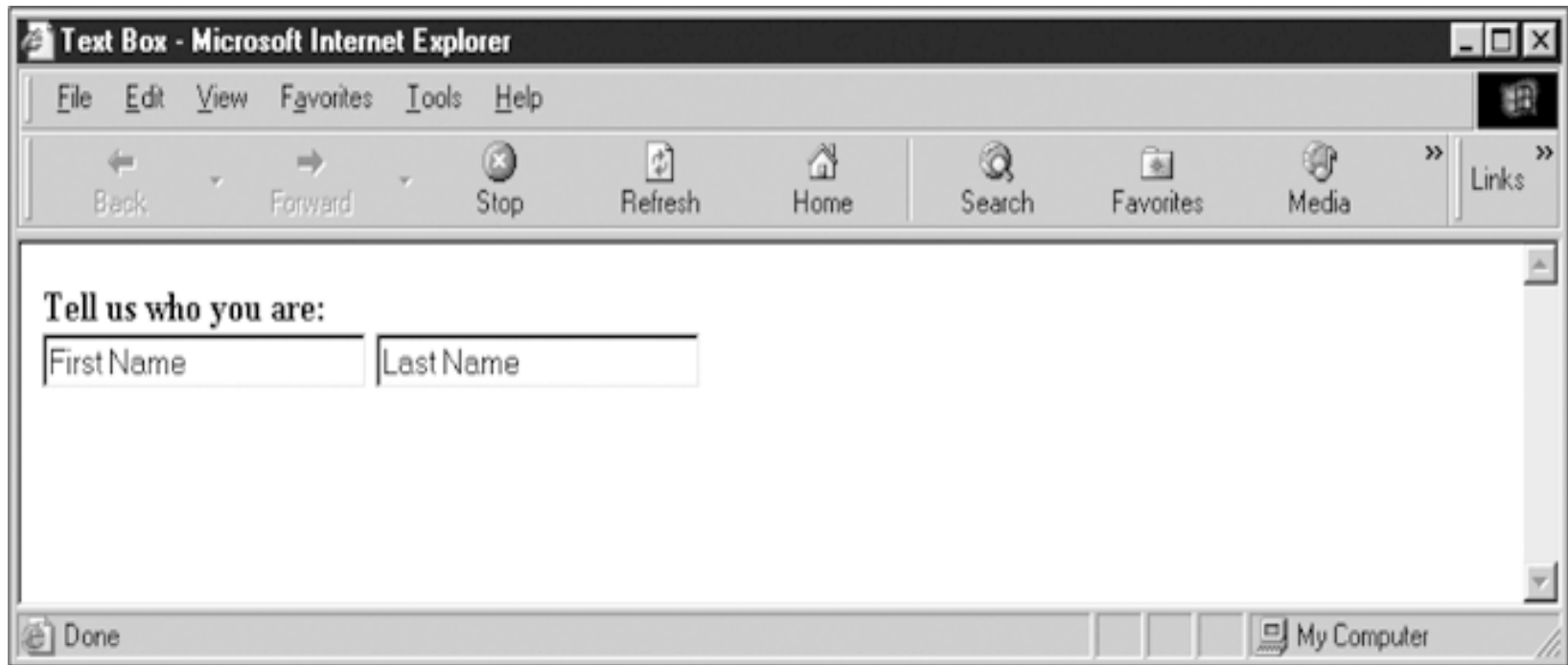


Figure 10-3 Text box inputs

Creating Check Boxes

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

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

Check Box Inputs

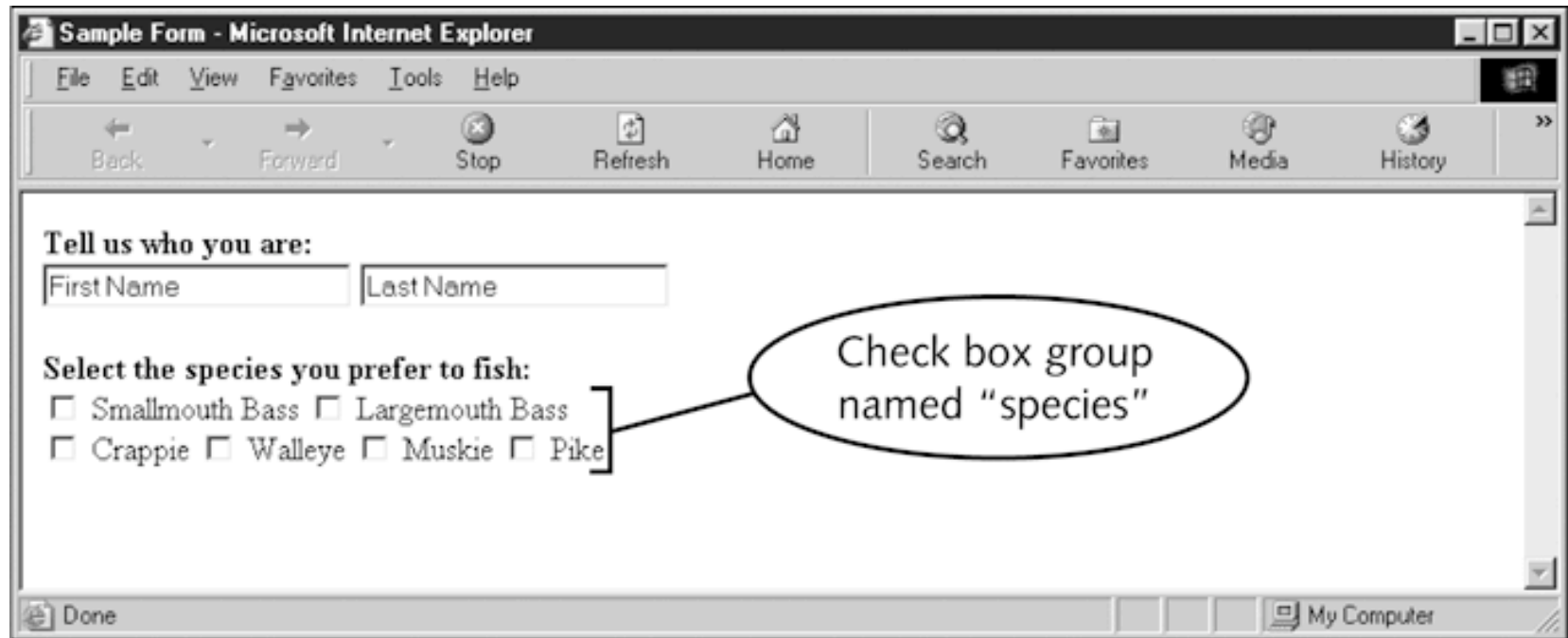


Figure 10-4 Check box inputs

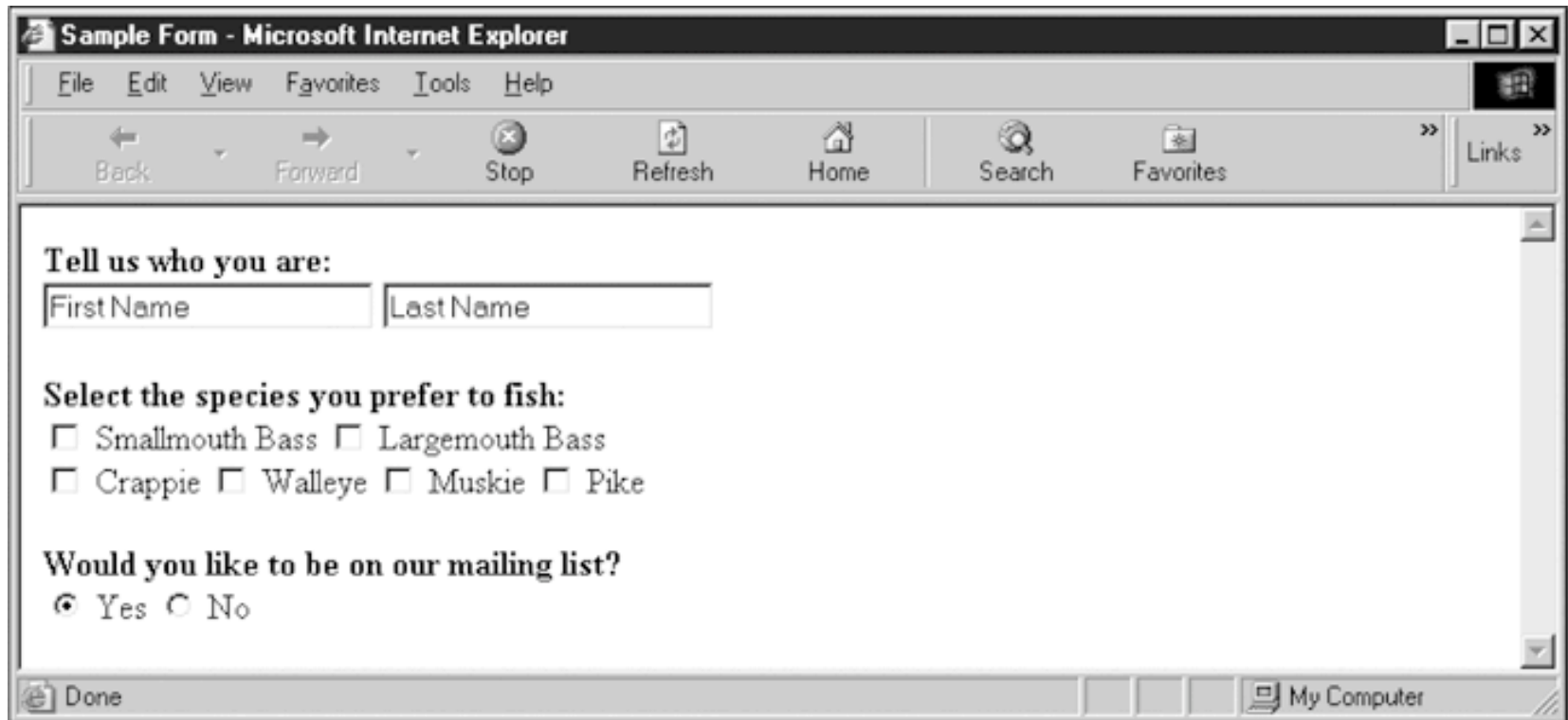
Creating Radio Buttons

a. Radio buttons are like checkboxes, but only one selection is allowed.

```
<input type="radio" name="list" value="yes" checked>
```

Yes

Radio Button Inputs



The image shows a screenshot of a Microsoft Internet Explorer browser window titled "Sample Form - Microsoft Internet Explorer". The browser's menu bar includes "File", "Edit", "View", "Favorites", "Tools", and "Help". The toolbar contains buttons for "Back", "Forward", "Stop", "Refresh", "Home", "Search", "Favorites", and "Links". The main content area displays a form with the following sections:

Tell us who you are:
First Name Last Name

Select the species you prefer to fish:
 Smallmouth Bass Largemouth Bass
 Crappie Walleye Muskie Pike

Would you like to be on our mailing list?
 Yes No

The status bar at the bottom shows "Done" and "My Computer".

Figure 10-5 Radio button inputs

Creating Submit & Reset Buttons

a. 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">
```

Submit and Reset Buttons

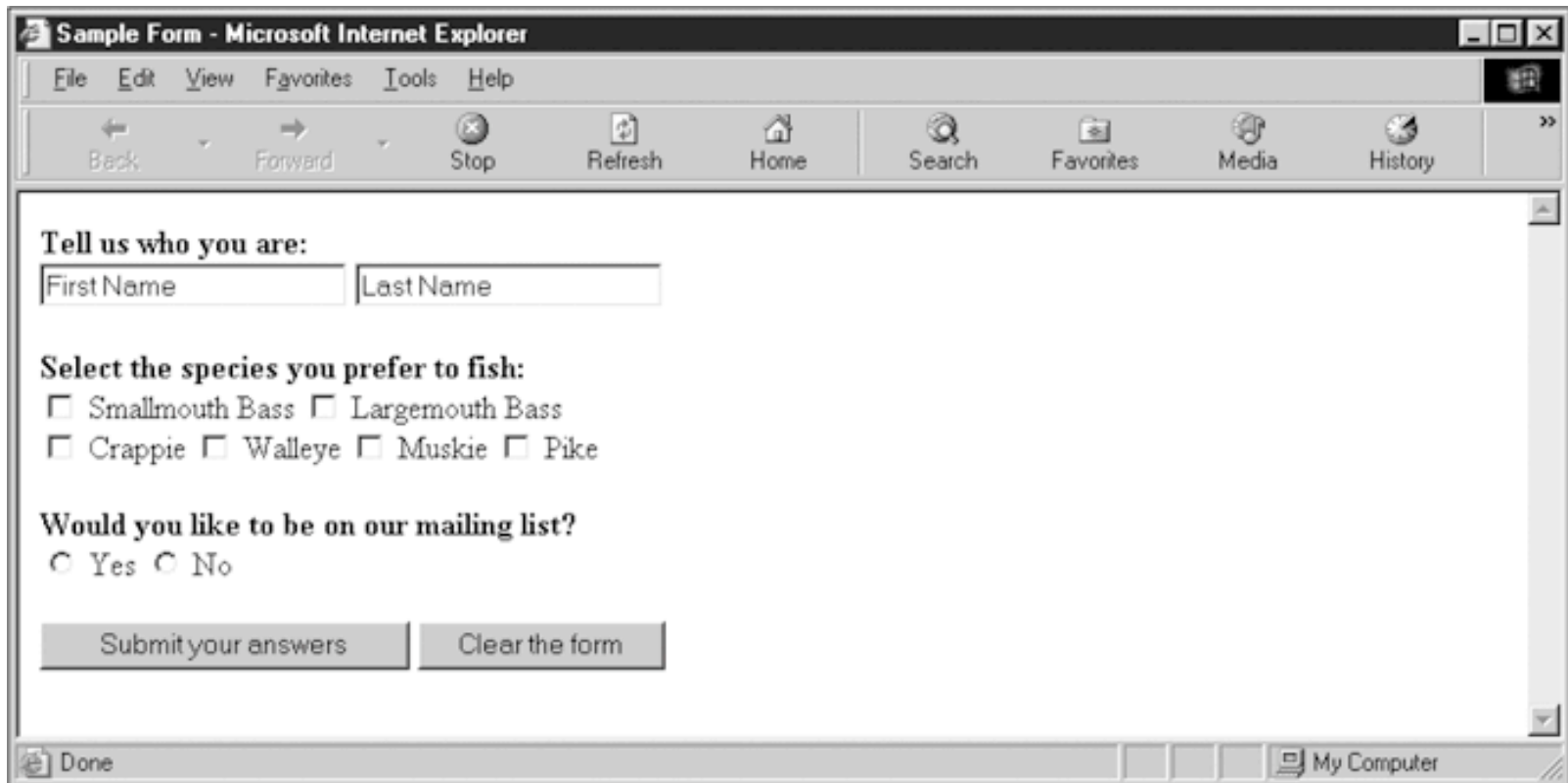


Figure 10-6 Submit and reset input buttons

Creating an Image for the Submit Button

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

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

Submit Button with an Image



Figure 10-8 Using an image for the submit button

Creating a Password Entry Field

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

password: `<input type="password" size="30">`

Password Input



Figure 10-10 Password type input

Using the <select> Element

a. 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>
```

Select List Box



Figure 10-11 A select list box

Using the <select> Element

a. 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>
```

Scrollable Select List



Figure 10-12 A scrollable select list

Using the <textarea> Element

a. 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>
```

Textarea



Figure 10-14 <textarea> element

Creating Input Groupings

```
<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>
```

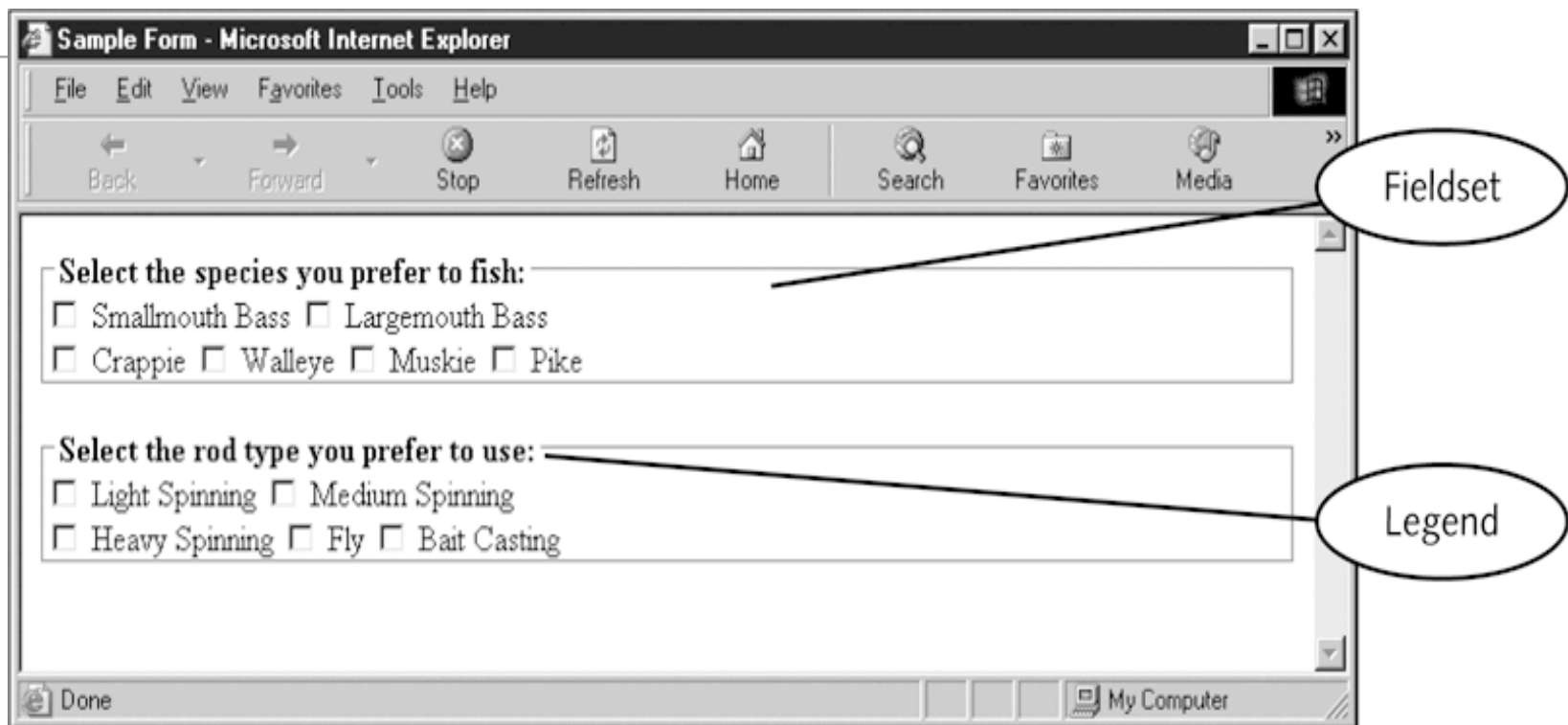


Figure 10-15 Grouping and labeling `<input>` elements

Summary

- a. You will need to work with some type of server-based software program to process the data from your form

 - b. You have a variety of form elements to choose from when building a form. Use the correct type of form element for the type of data you are gathering. For example, use checkboxes for multiple-choice questions. For a long list of choices, use a select list.

 - c. The `<fieldset>` and `<legend>` elements let you create more visually appealing forms that have logical groupings of input elements with a title
-