

Incorporating Forms in Web Pages

OBJECTIVES

You will be able to:

- understand what an HTML form is.
- understand how form processing occurs.
- understand the relationship between an HTML form and a Common Gateway Interface (CGI) script.
- understand and use the various posting methods.
- create forms using form tags and attributes (text, check box, radio buttons, submit, reset).
- create forms using an option list and a multiple option list.
- submit form contents to an e-mail address.
- understand the use of CGI posting methods.

This lesson introduces you to HTML forms and their uses. You will learn how to construct HTML forms using graphical user interface objects such as text area fields, radio buttons and check boxes, option list menus and multiple option lists, and use a form as an interface to a search engine.

HTML Forms

Up until now, you have learned how to create HTML documents to present information to the user. HTML also offers a mechanism called a "form" that allows the user to send information by interactively entering data into an HTML page. The user-entered data is then submitted to the server where it is stored or processed. In this lesson, you will learn how to create a simple form and have the user data sent to an e-mail address that you specify.

HTML forms may contain several elements, including:

- Text entry fields
- Radio buttons
- Checkboxes
- List boxes and Scrolling lists
- Submit and Reset buttons

These user interface elements allow customized forms to be constructed. You will learn how to incorporate each of these form elements.

Processing a Form

The data input into a form by the user may be processed in one of two ways. The results of the form may simply be delivered to an appropriate e-mail address, or may be sent to a *Common Gateway Interface (CGI)* script residing on the server.

A CGI script can perform any number of tasks with the returned data. The data may be entered into an existing database, used to create a customized Web page (often as a reply to the user), or may be used to search a database and return a result to the user.

Although a CGI script can be a very powerful tool, there are a number of considerations to be examined before selecting this method. First, because this script will run each time a form is completed, it puts an additional processing burden on the server. Second, these CGI scripts are somewhat complex to write, and require at least a basic knowledge of programming languages. An in-depth discussion of CGI scripting goes beyond the scope of this class, but additional information may be found on the Internet.

To simplify the use of form data, the HTML code for the form may be written to have the completed data sent to an e-mail address. This is the method that will be used in

class. One disadvantage to this method is the difficulty that arises for users whose browser has not been properly configured to send a form to an e-mail address. For those users whose browser is not configured for mail, it is suggested that you include a fax number or postal address to which the results may alternately be printed and sent.

When form data is processed, the data will appear as a string, similar to the example below:

FirstName=Jane&LastName=Student&Email=JStudent@abc.com

Although the message may appear somewhat cryptic at first glance, it is quite simple to use a search and replace function to translate the data to a more useable form. The first task for a CGI script is to *parse* this information. The data might be translated to appear as the sample below:

FirstName: Jane

LastName: Student

Email: JStudent@abc.com

If received via e-mail, there are utilities that will parse these strings for you, which you may then import into a database.

Form Tag

The FORM tag is a closed tag used within the body of an HTML document. It has two attributes associated with it.

<i>Attribute</i>	<i>Description</i>
METHOD	Either GET or POST. GET sends the form data as part of the URL in the header of the transmission. POST sends the form data as part of the body of the transmission. Choice is determined by the requirements of the recipient.
ACTION	Specifies where the information is to be sent when the user submits the form.

The ACTION will refer to the URL where the data is to be submitted. This is normally a CGI script, but may be an e-mail address. (If using an e-mail address, the METHOD specified must be POST.)

<FORM METHOD=GET ACTION="/cgi/scriptname">

<FORM METHOD=POST ACTION="/cgi/scriptname">

<FORM METHOD=POST ACTION="mailto:manager@abc.com">

Input Tag

The INPUT tag is an entry field that may be displayed in a variety of ways. The INPUT tag has five attributes associated with it, depending on the type specified:

<i>Attribute</i>	<i>Description</i>
TYPE (required)	Indicates the display of the entry field. Possible values are TEXT, RADIO, CHECKBOX, SUBMIT and RESET.
NAME (required for some types)	The name for the data field. Required for text fields, radio buttons and checkbox elements. If the returned data is to be submitted to a script or entered into a database, it may be necessary to use a name corresponding to that script or database.
SIZE (optional)	Specifies the displayed width of a text field, specified as a number of characters.
MAXLENGTH (optional)	Specifies the maximum number of characters that may be entered into a text field.
VALUE (required for some types)	Indicates the content of the selection submitted by the user. Required for radio buttons and checkboxes, optional for other elements.

Basic Text INPUT Field

The most basic element of a form is the text field. Define by the TYPE attribute of the INPUT tag, the text field permits the unlimited entry of text on a single line. If multiple fields are create, the user may tab through the fields according to the order of fields down the page.



Exercise 2-1: Creating a Text Input Field

In this exercise, you will learn how to create a basic form using a text INPUT field.

1. Open the **order.htm** document in your text editor.
2. Enter the FORM tag within the body of the document, as shown. Use your own e-mail address or the address specified by your instructor.

```
<BODY>
<H2><l>If you would like to receive a copy of our catalog,<BR>
please complete the survey below.</l></H2>
```

```
<FORM METHOD=POST ACTION="mailto:YourName@YourCo.com">
```

3. Create the closing tag for the form at the end of the body where indicated.

`<!-- END THE FORM HERE -->`

`</FORM>`

If you have problems sending this form, you can print it, complete it, and mail it or fax it to us at:
`
`

4. In the area indicated, create an INPUT text field for each name and address field.

Full Name:
`
`

`<INPUT TYPE=text NAME="name">`

`
`

Street Address:
`
`

`<INPUT TYPE=text NAME="street">`

`
`

City/State/:
`
`

`<INPUT TYPE=text NAME="city_state">`

5. Save the changes, and switch to your web browser.
6. Open the **order.htm** document in your web browser.

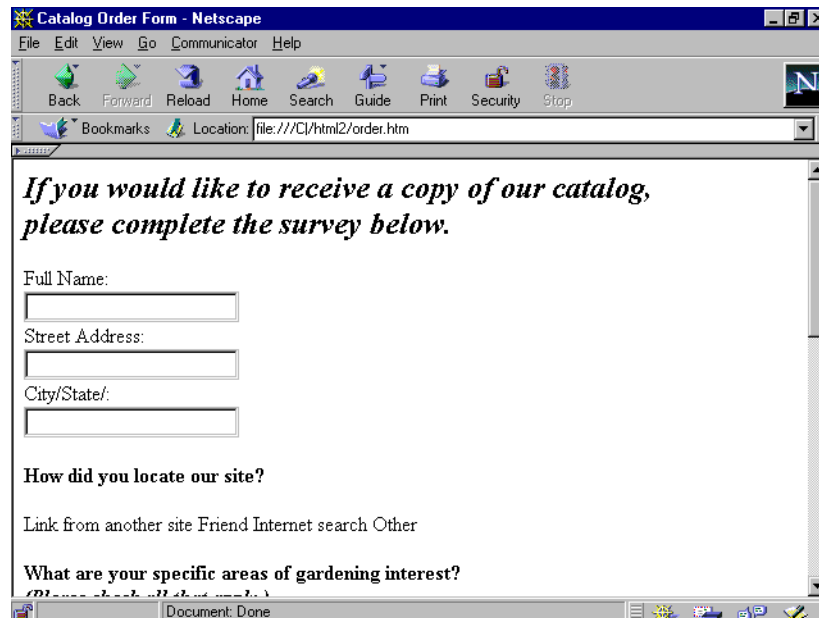


Figure 2-1: Text Input Fields in Forms

It is possible to modify a text field with the SIZE and MAXLENGTH attributes. The SIZE attribute changes the apparent size of the text field, while the MAXLENGTH attribute limits the number of characters that may be entered. Thus, the size of the field can suggest to the user a proper response, while the MAXLENGTH can prevent the user from entering more information than is needed or desired.



Exercise 2-2: Modifying a Text Input Field

In this exercise, you will create a new text field and set the `SIZE` and `MAXLENGTH` of this field.

1. Return to the text editor, and open the **order.htm** document, if necessary.
2. Create a separate field for Zip code. Set a display size at 5 and maximum character count to 5, also.

```
<BR>
```

```
City/State:<BR>
```

```
<INPUT TYPE=text NAME="city_state">
```

```
<BR>
```

```
Zip:<BR>
```

```
<INPUT TYPE=text NAME="zip" SIZE=5 MAXLENGTH=5>
```

3. Save the changes.
4. Reload the **order.htm** document in your web browser.

Try to enter more than five characters for the Zip code. What is the result?

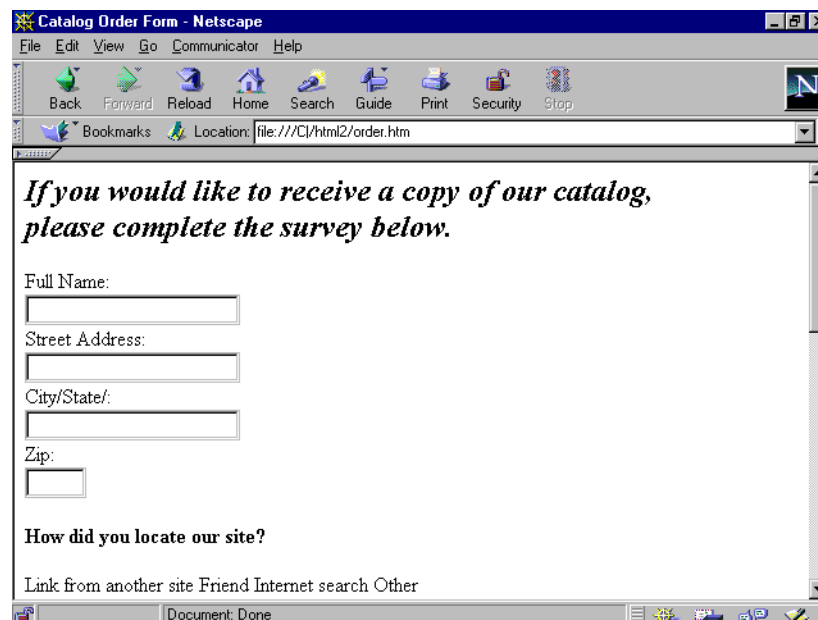


Figure 2-2: Modify Display and Limit Input for Text Fields

Radio Buttons

When creating a form that contains a question with a finite set of answers, you may wish to the user to make only one choice from a list. Radio buttons have become the design convention for indicating to users such a choice. When choosing from a set of radio buttons, the user may *only* select one. Each INPUT tag within a set of choices will have the same name, but a different VALUE attribute so that only one choice is recorded.



Exercise 2-3: Creating Radio Buttons

In this exercise, you will add the radio button INPUT TYPE to a form.

1. Return to the **order.htm** document in your text editor.
2. At the position indicated, add INPUT tags for a set of radio buttons.

```
<INPUT TYPE=radio NAME="locate" VALUE="link"> Link from another  
site  
<INPUT TYPE=radio NAME="locate" VALUE="friend"> Friend  
<INPUT TYPE=radio NAME="locate" VALUE="search"> Internet search  
<INPUT TYPE=radio NAME="locate" VALUE="other"> Other
```

3. Save the changes, and switch to your web browser.
4. Reload the **order.htm** document in your web browser.

Test each of the buttons to be certain that you may only select one.

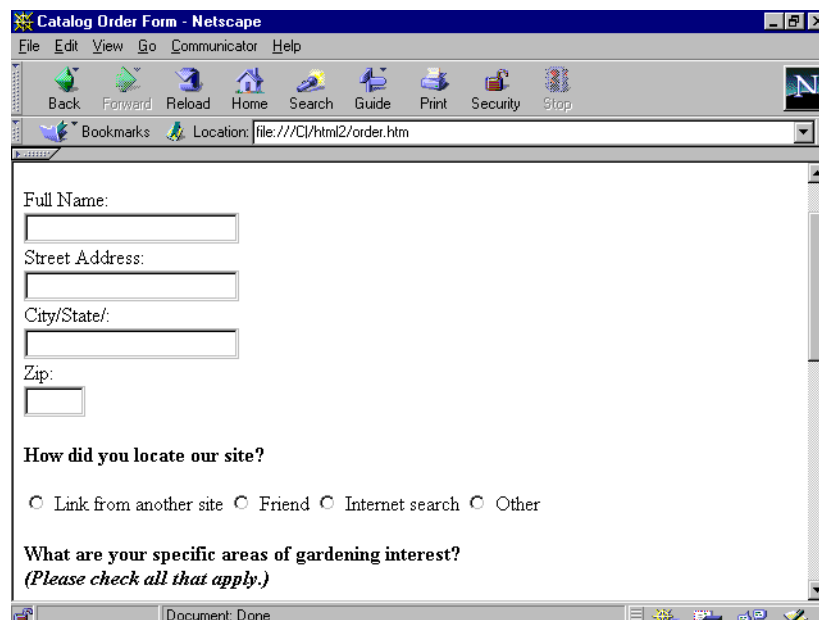


Figure 2-3: Radio Buttons in Forms

Checkboxes

Checkboxes are used to offer a list from which users may choose more than one option. Whereas the radio buttons will only allow one choice, checkboxes will report back multiple selections. As with radio buttons, each INPUT tag within a set of checkboxes will have the same name but a different VALUE attribute.



Exercise 2-4: Creating Checkboxes

In this exercise, you will add the checkbox INPUT TYPE to a form.

1. Return to the file **order.htm** in your text editor.
2. At the position indicated, add INPUT tags for a set of checkboxes.

```
<INPUT TYPE=checkbox NAME="interest" VALUE="indoor"> Indoor
<INPUT TYPE=checkbox NAME="interest" VALUE="vegetable">
Vegetable
<INPUT TYPE=checkbox NAME="interest" VALUE="flower"> Flowers
<INPUT TYPE=checkbox NAME="interest" VALUE="tree_shrub">
Tree/Shrubs
```

3. Save the changes, and switch to your web browser.
4. Reload the **order.htm** document in your web browser.

Test to verify that you may select any number of checkboxes.

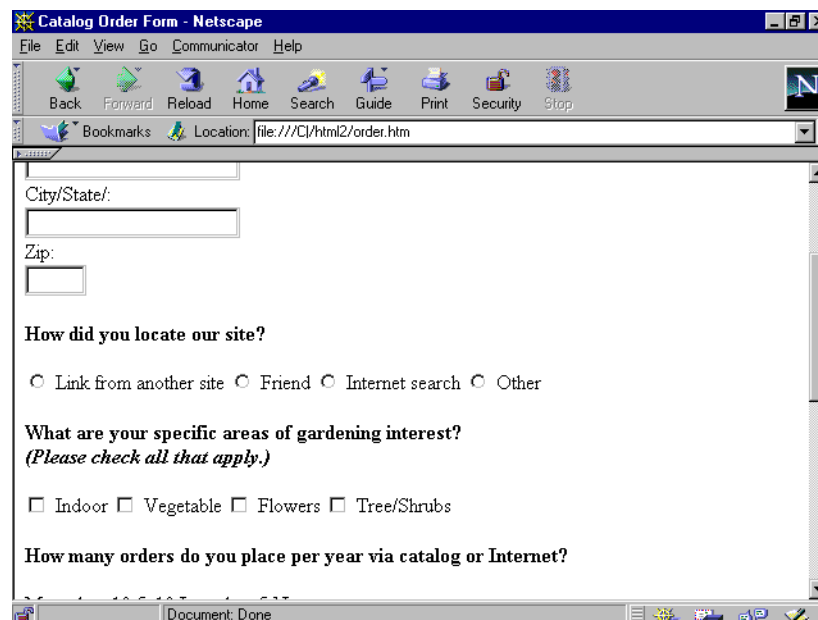


Figure 2-4: Checkboxes in Forms

Selection Lists

Though the INPUT tags are the most common form elements, there are other tags to add functionality to your forms. The SELECT tag is used to create a pop-up menu or a scrollable list. The SELECT tag is different from the INPUT tag because it must be closed. The tag has three attributes associated with it:

<i>Attribute</i>	<i>Description</i>
NAME (required)	The variable name for the returned data. If the returned data is to be submitted to a script or entered into a database, it may be necessary to use a name corresponding to that script or database.
MULTIPLE (optional)	Specifies that multiple choices may be selected.
SIZE (optional)	Specifies the number of items to be displayed within a scrolling list.

The OPTION tag is nested inside the SELECT tag to specify each of the available choices. You may add the SELECTED attribute to the OPTION tag so that a particular item is pre-selected automatically.



Exercise 2-5: Creating A Selection Menu

In this exercise, you will create a pop-up selection menu that allows the user to make only one selection.

1. Return to the file **order.htm** in your text editor.
2. At the position indicated, add tags for a selection menu.

Adding the SELECTED attribute to the OPTION tag sets that item as the default choice.

```
<SELECT NAME="orders">  
<OPTION> More than 10  
<OPTION SELECTED> 5-10  
<OPTION> Less than 5  
<OPTION> Never  
</SELECT>
```

3. Save the changes, and return to the web browser.
4. Reload the **order.htm** document in your web browser.

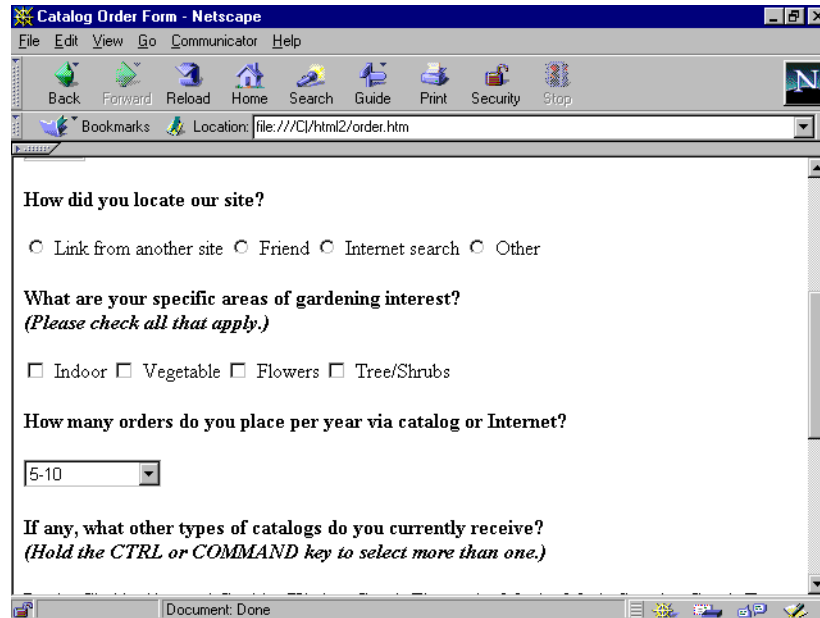


Figure 2-5: Selection Menu in a Form

The MULTIPLE attribute indicates that more than one item may be selected, usually by holding the CTRL or COMMAND key as the selections are made. The SIZE attribute defines the size of the field as determined by the number of choices that will be visible.



Exercise 2-6: Creating A Scrolling Selection List

In this exercise, you will create a selection list that allows the user to scroll through the choices, and select more than one.

1. Return to the file **order.htm** in your text editor.
2. At the position indicated, add tags for a scrolling selection list.

```
<SELECT NAME="catalogs" MULTIPLE SIZE=4>
<OPTION> Books
<OPTION> Clothing/Apparel
<OPTION> Cooking/Kitchen Goods
<OPTION> Electronics
<OPTION> Movies
<OPTION> Music
<OPTION> Sporting Goods
<OPTION> Toys
</SELECT>
```

3. Save the changes, and return to the web browser.
4. Reload the **order.htm** document in your web browser.

Test to verify that you may select more than one item.

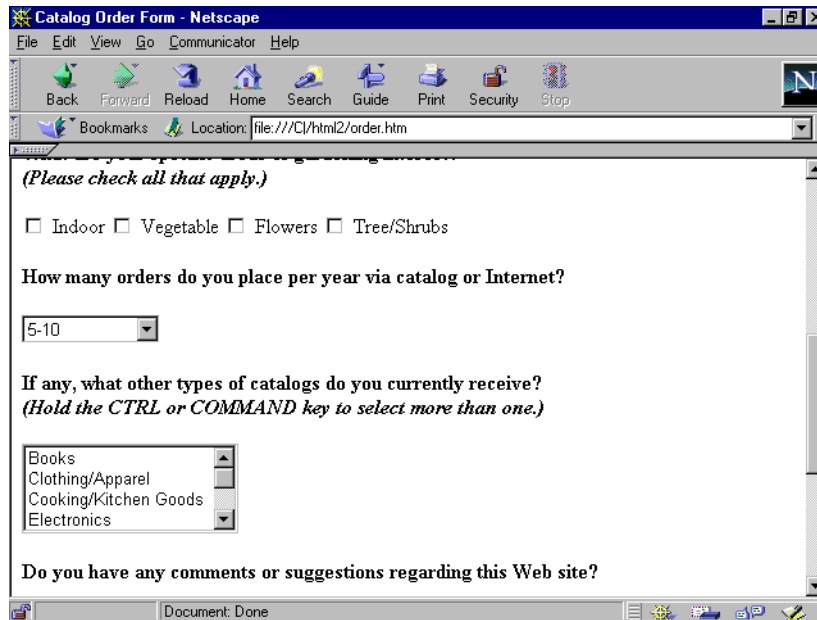


Figure 2-6: Scrolling Selection List in a Form

TEXTAREA Tag

The TEXTAREA tag is an entry field where the size of the field can be defined in rows by number and columns by number of characters. You can create a scrollable box with the TEXTAREA tag. The tag must also be closed. The TEXTAREA tag has three attributes associated with it:

<i>Attribute</i>	<i>Description</i>
NAME (required)	The variable name for the returned data. If the returned data is to be submitted to a script or entered into a database, it may be necessary to use a name corresponding to that script or database.
ROWS (optional)	Specifies the displayed height of the text area, as a number of lines.
COLS (optional)	Specifies the displayed width of the text area, specified as a number of characters.

You may also create some default text to appear in the field. Text appearing between the opening and closing of the TEXTAREA tag will appear in the field.

<TEXTAREA NAME="Opinions" ROWS=4 COLS=60>Please enter your thoughts here...</TEXTAREA>



Exercise 2-7: Creating A TEXTAREA

In this exercise, you will create a field for comments using the TEXTAREA tag.

1. Return to the file **order.htm** in your text editor.
2. At the position indicated, add tags for a text area.

<H4>Do you have any comments or suggestions regarding this Web site?</H4>

<!-- CREATE TEXT AREA FIELD BELOW -->

```
<TEXTAREA NAME="comments" ROWS=4 COLS=60>
</TEXTAREA>
```

3. Save the changes, and return to the web browser.
4. Reload the **order.htm** document in the web browser.

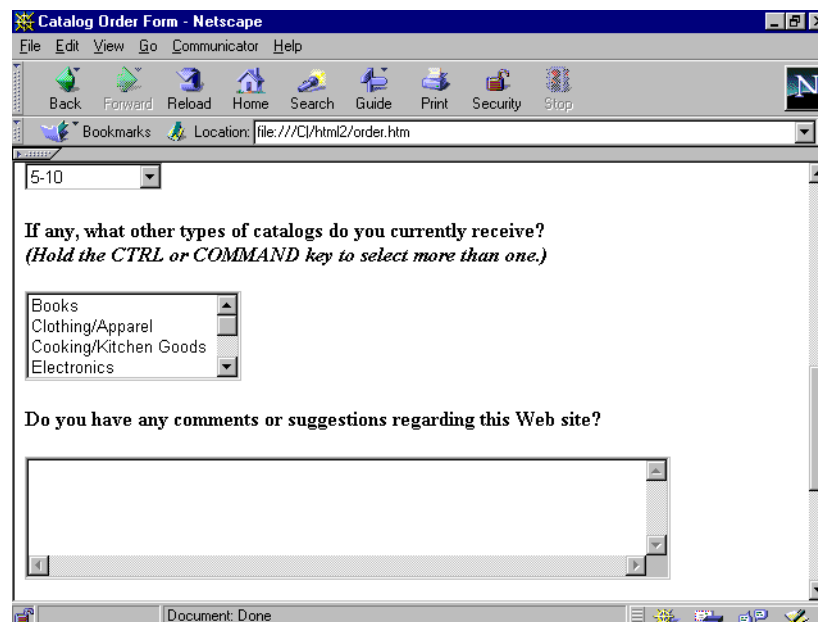


Figure 2-7: Text Area in a Form

Submit and Reset Buttons

The SUBMIT and RESET buttons are created with an INPUT tag. To complete a form, a SUBMIT button is required. Otherwise, the data cannot be processed. The RESET button is optional, but is helpful to users who wish to clear the form and start over. Either tag may include a VALUE attribute to specify the label for the button.



Exercise 2-8: Adding Submit and Reset Buttons

In this exercise, you will add Submit and Reset buttons to finish the form. You may test the form when complete if you have an Internet connection.

1. Return to the file **order.htm** in your text editor.
2. At the position indicated, add INPUT tags to create the SUBMIT and RESET buttons.

Click the SEND button if you're finished, or click RESET to start again.

**
**

<!-- ADD THE SUBMIT AND RESET BUTTONS BELOW -->

<INPUT TYPE=submit VALUE="Send it!">

<INPUT TYPE=reset>

3. Save the changes, and return to the web browser.
4. Reload the **order.htm** document in the web browser.

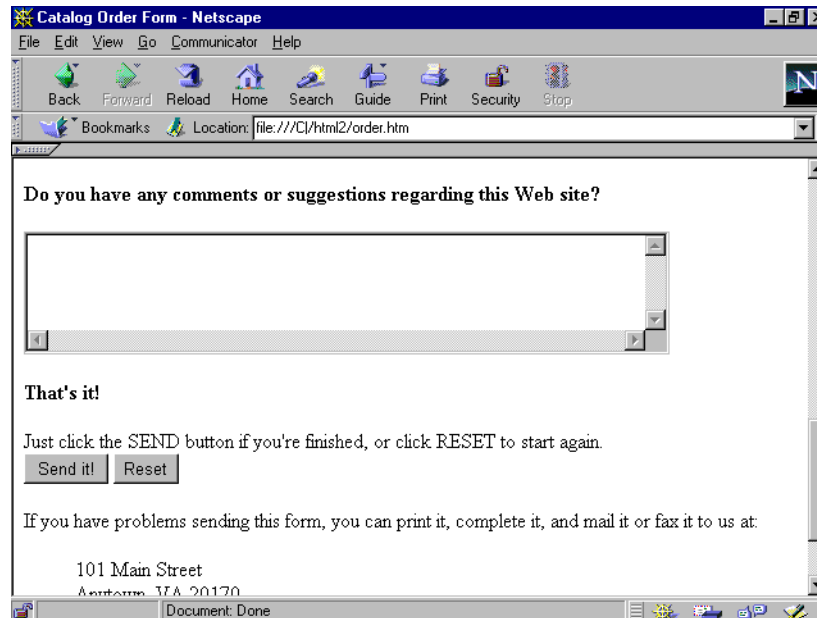


Figure 2-8: Submit and Reset Buttons

Interface to a Search Engine

Most forms submit data to be processed by a CGI script. You identify the script through the URL in the ACTION attribute of the FORM tag. In the following exercise, data will be submitted to a CGI script called "search," that resides on the Yahoo server.



Exercise 2-9: Accessing a CGI Script

In this exercise, you will complete a form that will submit the data to a script. You may test this form if you have a live Internet connection.

1. Open the file **search.htm** in your text editor.
2. At the position indicated, add the FORM tag, specifying the METHOD and ACTION *exactly* as shown.

```
<!-- INSERT FORM TAG HERE -->
```

```
<FORM METHOD=get ACTION="http://search.yahoo.com/bin/search">
```

3. Create a text field INPUT tag. The NAME *must* be specified as shown, in order to access the script correctly.

```
<!-- INSERT TEXT INPUT TAG BELOW -->
```

```
<INPUT TYPE=text NAME="p" SIZE=30>
```

4. Save the changes, and return to the web browser.
5. Open the **search.htm** document in the web browser.

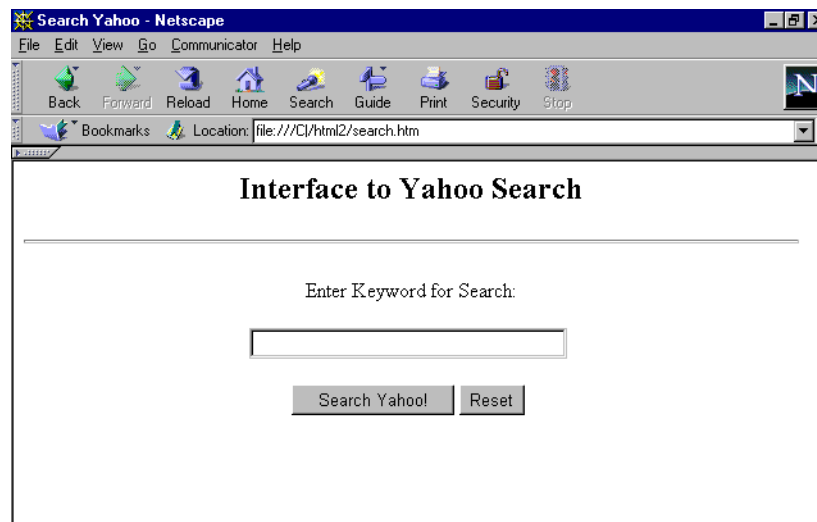


Figure 2-9: Interface to Yahoo Search

Review

In this lesson, you have been introduced to HTML forms and their uses. You have learned how to construct HTML forms using graphical user interface objects such as text area fields, radio buttons and check boxes, option list menus and multiple option lists. You have learned how to submit form data to a script and how to have it posted to you as an e-mail message.