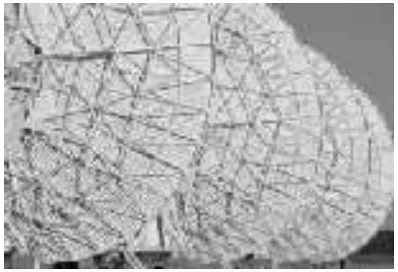
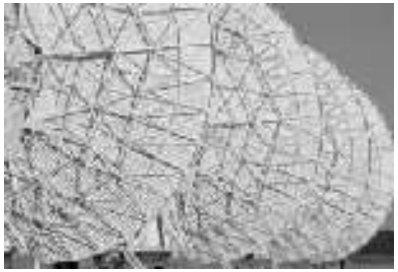


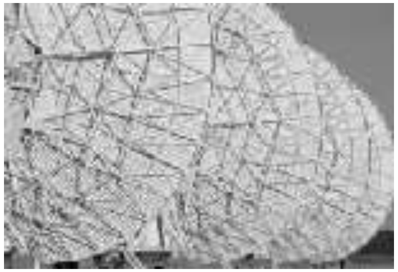
New Perspectives on Creating Web Pages with HTML

Creating Web Page Forms



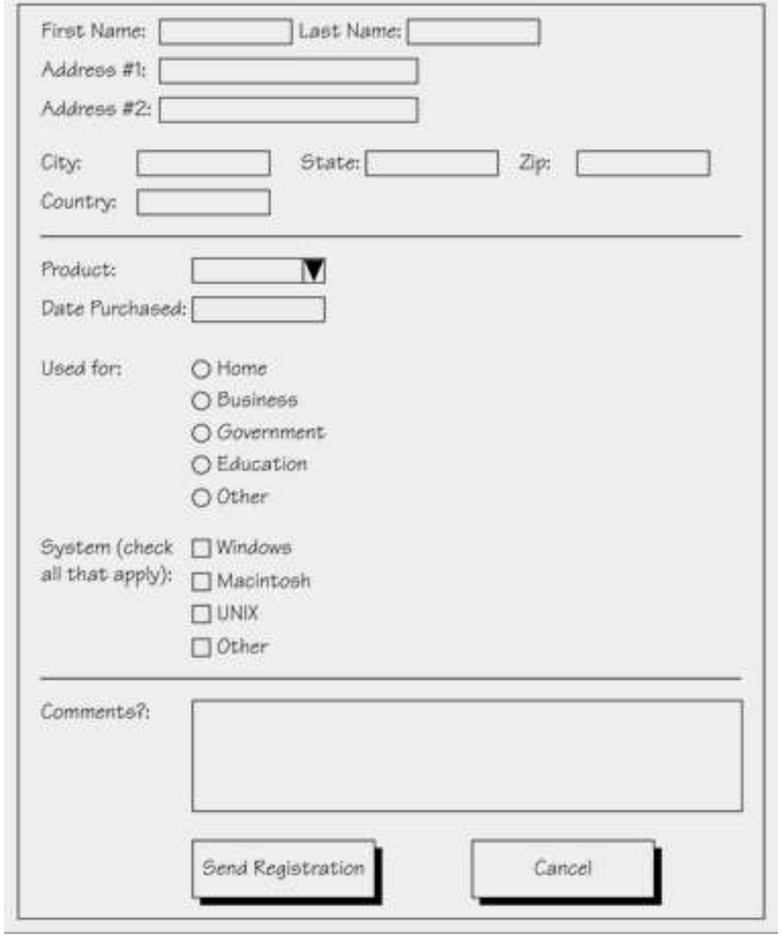
Tutorial Objectives

- Review the various parts of an online form
- Create form elements
- Create a hidden field on a form
- Work with form attributes



An Example of a Form

This figure shows a sketch of a proposed registration form.



First Name: ______ Last Name: ______

Address #1: ______

Address #2: ______

City: ______ State: ______ Zip: ______

Country: ______

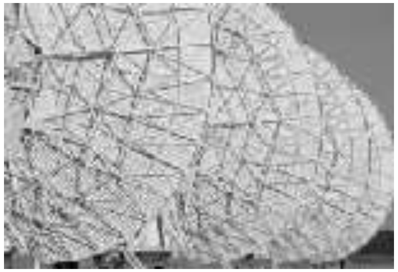
Product: ______ ▼

Date Purchased: ______

Used for:
○ Home
○ Business
○ Government
○ Education
○ Other

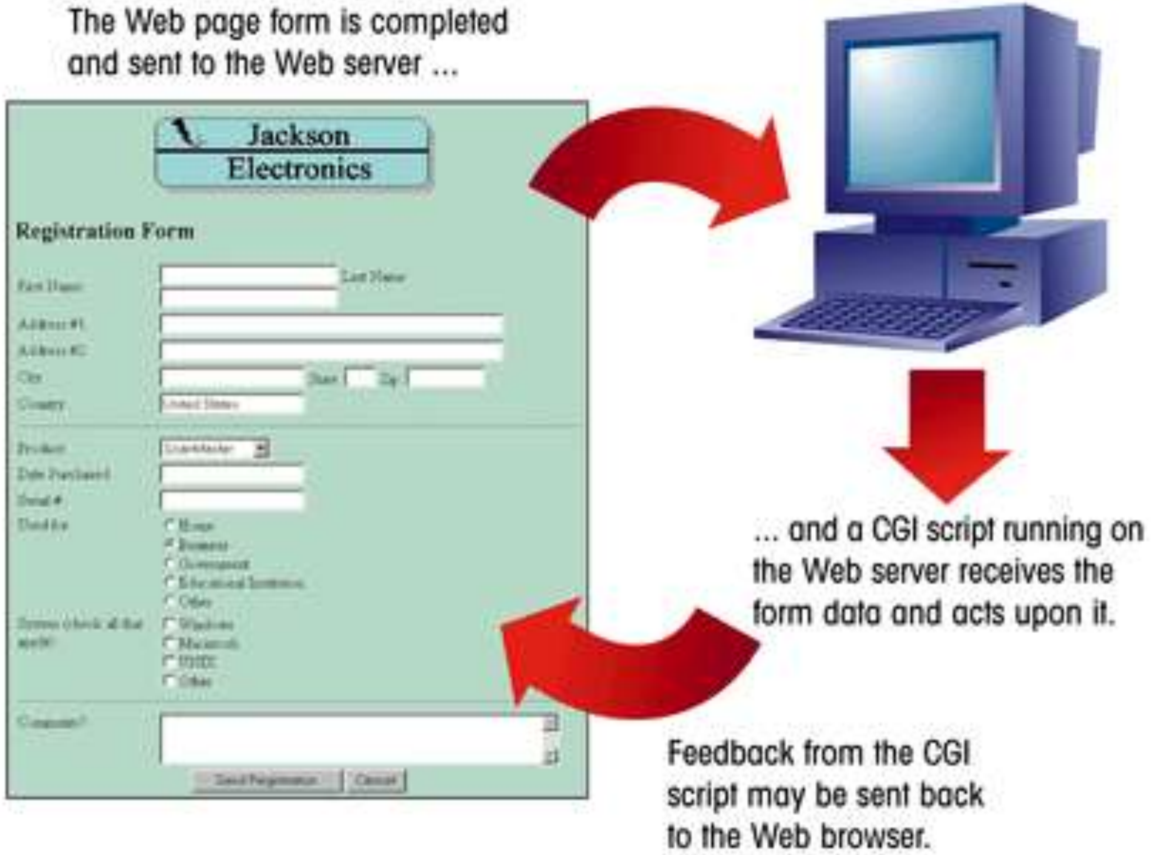
System (check all that apply):
□ Windows
□ Macintosh
□ UNIX
□ Other

Comments?: ______



The Interaction between a Web Page Form and a CGI/PHP Script

This figure shows how a Web page form interacts with a CGI/PHP script.



The Web page form is completed and sent to the Web server ...

Jackson Electronics

Registration Form

First Name: ______ Last Name ______

Address #1: ______

Address #2: ______

City ______ State ___ Zip ______

Country: United States

Product: [illegible]

Date Purchased: ______

Serial #: ______

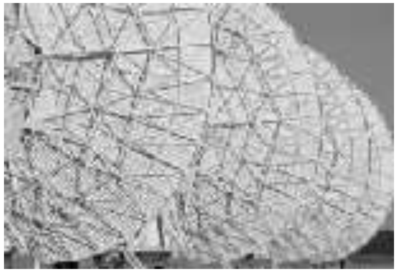
Used for: ○ Home ● Business ○ Government ○ Educational Institution ○ Other

System (check all that apply): □ Windows □ Macintosh □ UNIX □ Other

Comments?

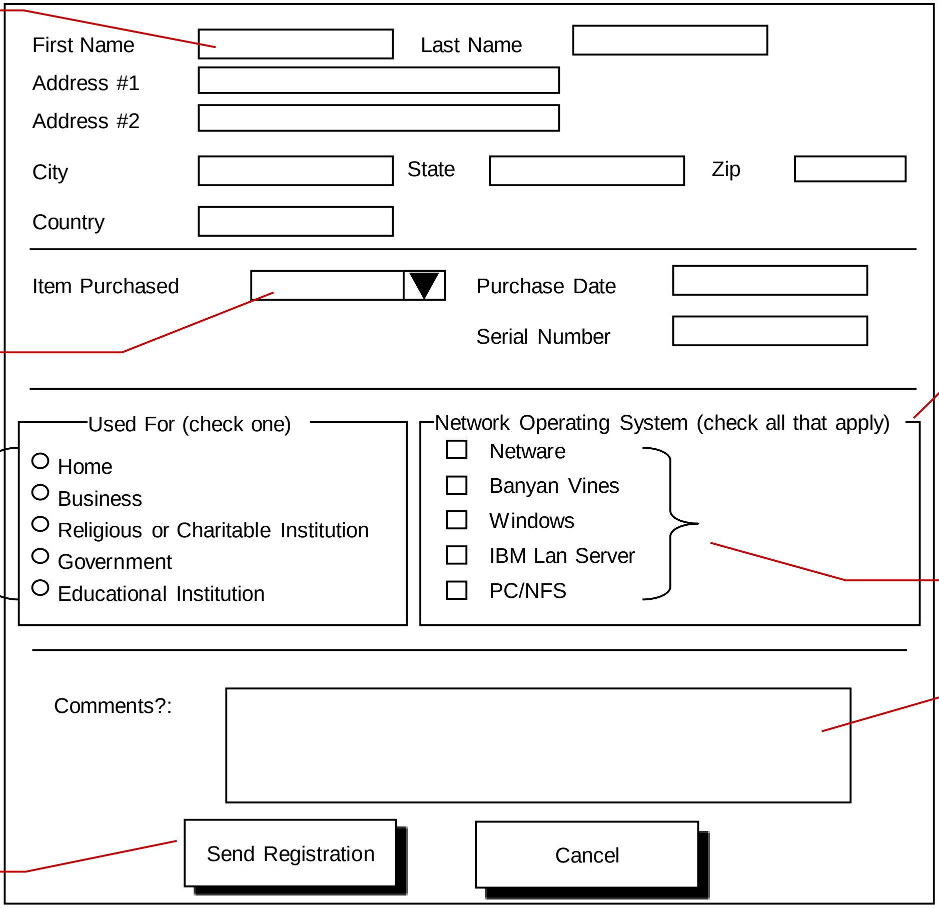
... and a CGI script running on the Web server receives the form data and acts upon it.

Feedback from the CGI script may be sent back to the Web browser.



Form Components and Elements

This figure shows a form that contains various control elements commonly used in Web page forms.



First Name
Last Name
Address #1
Address #2
City
State
Zip
Country

Item Purchased ▼
Purchase Date
Serial Number

Used For (check one)

- ○ Home
- ○ Business
- ○ Religious or Charitable Institution
- ○ Government
- ○ Educational Institution

Network Operating System (check all that apply)

- □ Netware
- □ Banyan Vines
- □ Windows
- □ IBM Lan Server
- □ PC/NFS

Comments?:

Send Registration

Cancel

text box

drop-down list box

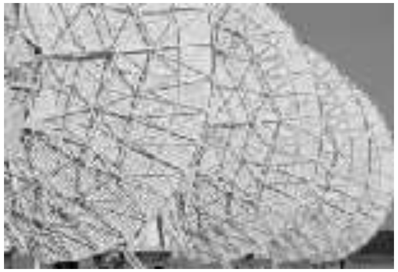
radio buttons

form button

group box

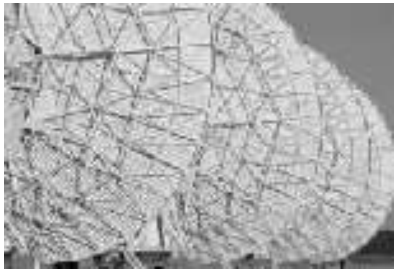
check boxes

text area



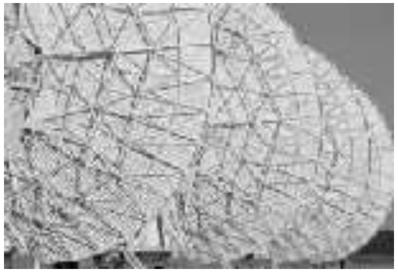
Form Control Elements

- Control elements that are commonly used:
  - **text boxes** for text and numerical entries
  - **selection lists** for long lists of options, usually appearing in a **drop-down list box**
  - **radio buttons**, also called **option buttons**, to select a single option from a predefined list
  - **check boxes** to specify an item as either present or absent
  - **groups boxes** to organize form elements
  - **text areas** for extended entries that can include several lines of text
  - **buttons** that can be clicked to start processing the form



Form Control Elements Continued

- Each control element in which the user can enter information is called a **field**.
- Information entered into a field is called the **field value**, or simply the **value**.
- In some fields, users are free to enter anything they choose.
- Other fields, such as selection lists, limit the user to a predefined list of options.

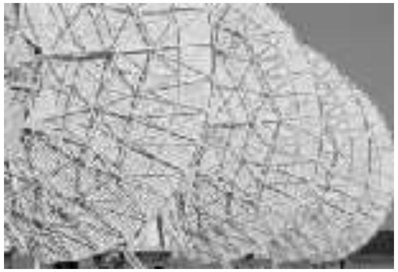


The **`<form>`** Tag

- The **`<form>`** tag identifies the beginning and end of a form.
- A single page can include several different forms, but you cannot nest one form inside another.
- The general syntax of the **`<form>`** tag is:

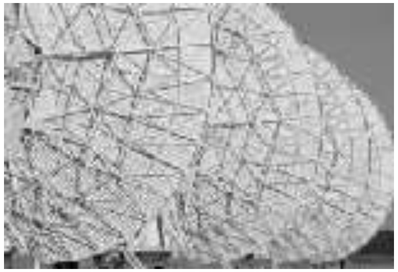
```
<form attributes>
  form elements and layout tags
</form>
```

- Between the **`<form>`** and **`</form>`** tags, place the various tags for each of the fields in the form.
- Use standard HTML tags to specify the form’s appearance.



The `<form>` Tag Continued

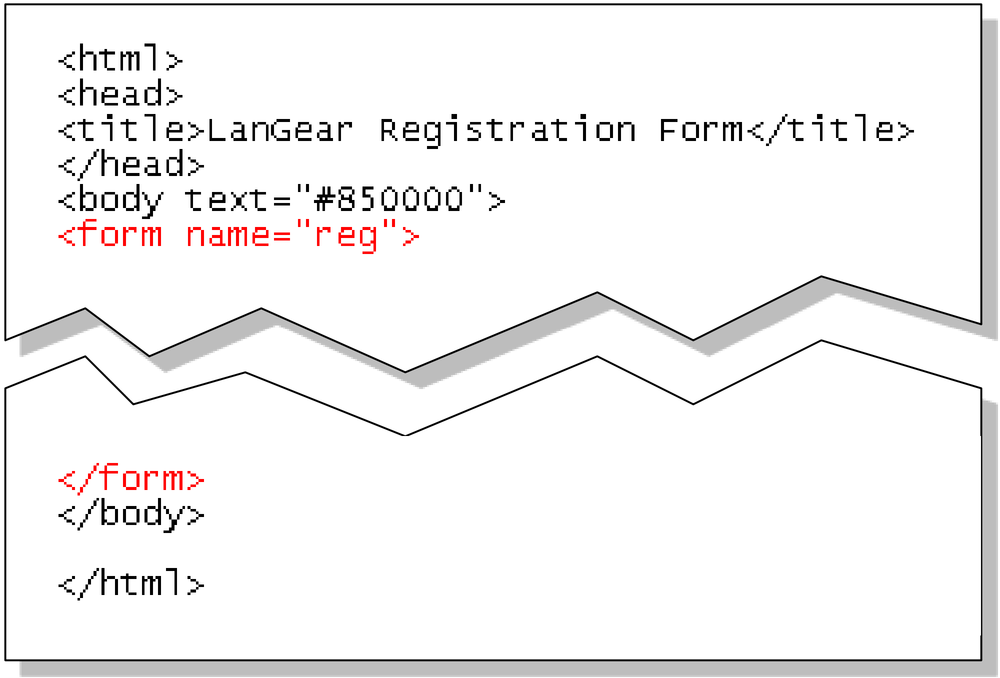
- A single Web page can contain multiple forms, the **`<form>`** tag includes the **name** attribute.
- The name attribute identifies each form on the page.
- The name attribute is also needed for programs that retrieve values from the form.



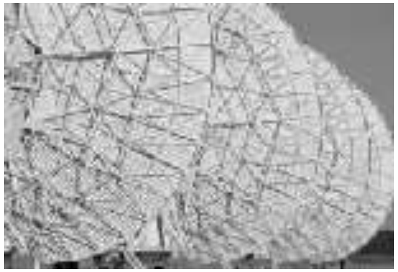
Adding the `<form>` Tag

The `<form>` tag includes attributes that control how the form is processed, including information on what CGI/PHP script to use, how the data is to be transferred to the script, and so forth.

This figure shows the form name “`reg`.”



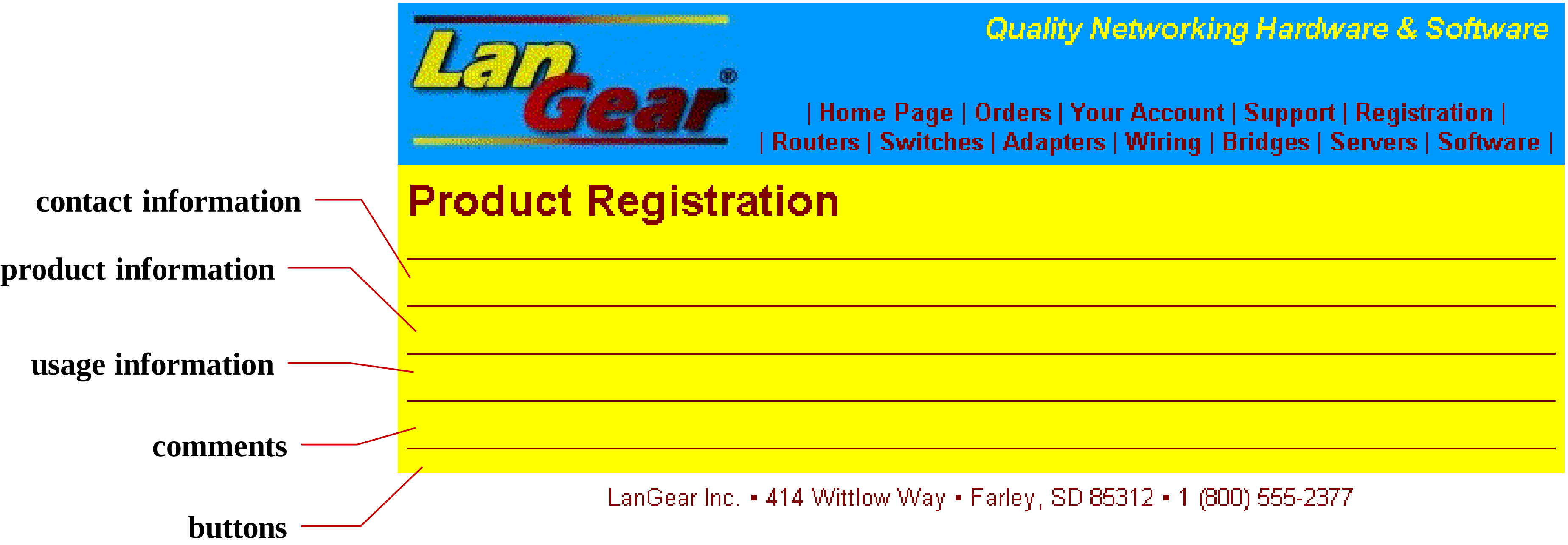
```
<html>
<head>
<title>LanGear Registration Form</title>
</head>
<body text="#850000">
<form name="reg">

</form>
</body>

</html>
```



Layout of a Registration Form

Horizontal lines can provide structure to a form by separating sections.

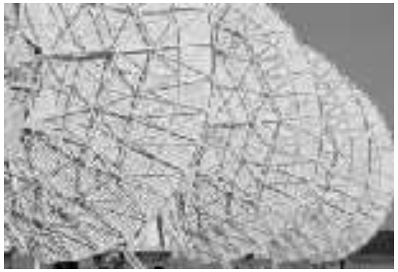
This figure shows sections that are separated from each other with a horizontal line. Structuring a form in this fashion can make it easier to identify the different sections.



LanGear®
Quality Networking Hardware & Software
| Home Page | Orders | Your Account | Support | Registration |
| Routers | Switches | Adapters | Wiring | Bridges | Servers | Software |
contact information
product information
usage information
comments
buttons

Product Registration

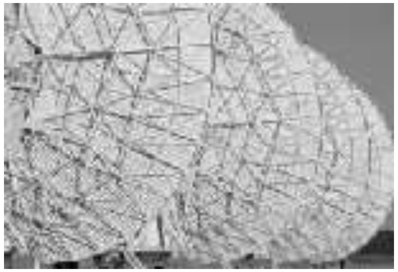
LanGear Inc. ▪ 414 Wittlow Way ▪ Farley, SD 85312 ▪ 1 (800) 555-2377



HTML Form Elements

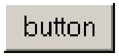
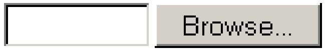
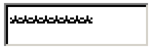
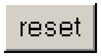
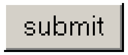
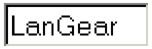
The HTML <form> element can contain one or more of the following form elements:

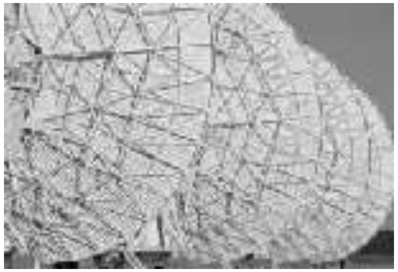
- <input>
- <label>
- <select>
- <textarea>
- <button>
- <fieldset>
- <legend>
- <datalist>
- <output>
- <option>
- <optgroup>



Input Types

This figure shows other supported HTML input types.

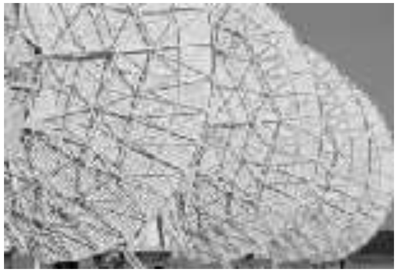
Type	**Description**
type=”button”	Display a button which can be clicked to perform an action from a script 
type=”checkbox”	Display a check box
type=”file”	Display a browse button to locate and select a file 
type=”hidden”	Create a hidden field, not viewable on the form
type=”image”	Display an inline image which can be clicked to perform an action from a script
type=”password”	Display a text box in which hides text entered by the user 
type=”radio”	Display a radio (option) button
type=”reset”	Display a button which resets the form when clicked 
type=”submit”	Display a button which submits the form when clicked 
type=”text”	Display a text box in which displays text entered by the user 



Input type

```
<input type="button">
<input type="checkbox">
<input type="color">
<input type="date">
<input type="datetime-local">
<input type="email">
<input type="file">
<input type="hidden">
<input type="image">
<input type="month">
<input type="number">
```

```
<input type="password">
<input type="radio">
<input type="range">
<input type="reset">
<input type="search">
<input type="submit">
<input type="tel">
<input type="text">
<input type="time">
<input type="url">
<input type="week">
```



Working with Text Boxes

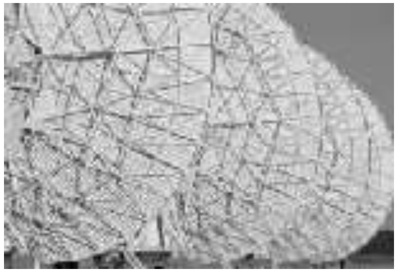
- Text boxes are created using the **`<input>`** tag.
- The general syntax is:

```
<input type=“type” name=“name” id=“id”>
```

  - ***type*** specifies the type of input field
  - ***name*** and ***id*** attributes identifies the input field for the script
- To create a text box, you would enter the tag:

```
<input type=“text”>
```

- If the ***type*** attribute is not included, the Web browser assumes, by default, that you want to create a text box.

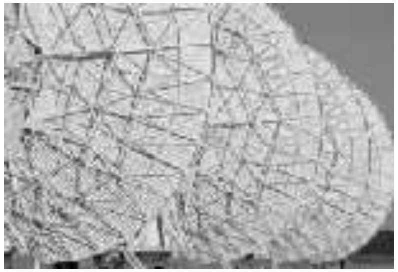


Creating a Text Box

- To create a text box, use the following HTML code:

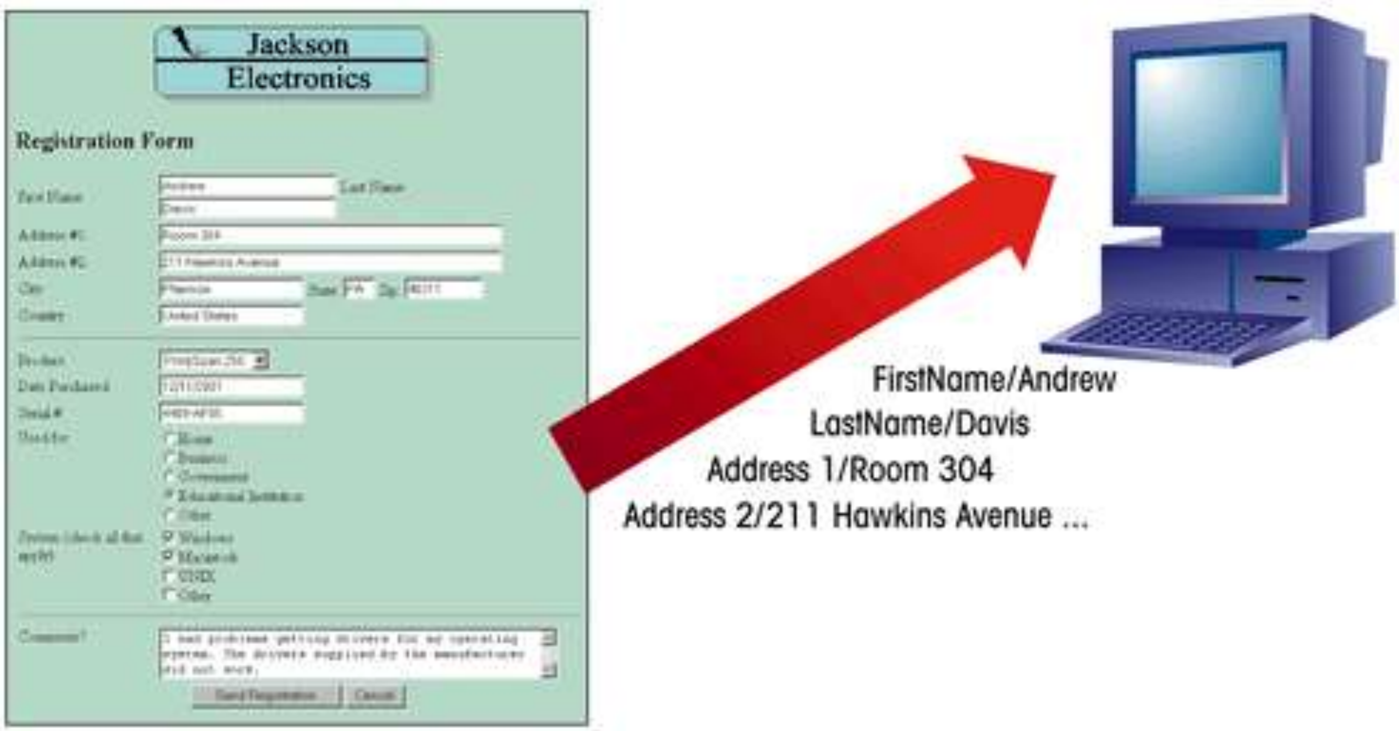
```
<input name=“name” id=“id” value=“value”
  size=“value” maxlength=“value”>
```

  - ***name*** and ***id*** attributes identify the field
  - ***value*** attribute assigns a default value to the text box
  - ***size*** attribute defines the width of the text box in number of characters
  - ***maxlength*** attribute defines the maximum number of characters allowed in the field



Name/Value Pairs sent from the Web Form to the CGI Script

This figure shows when form data is sent to the CGI script, the script receives the name or id of each field in the form paired with whatever value the user entered in the field. The script then processes the data according to each name/value pair.



Jackson Electronics

Registration Form

First Name: Andrew Last Name: Davis

Address #1: Room 304

Address #2: 211 Hawkins Avenue

City: [illegible] State: [illegible] Zip: [illegible]

Country: United States

Product: [illegible]

Date Purchased: [illegible]

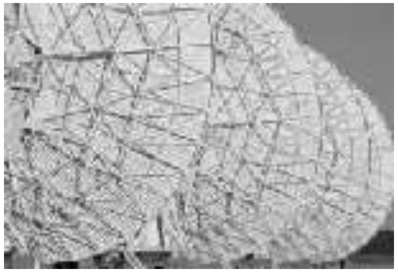
Serial #: [illegible]

Used for:
○ Home
○ Business
○ Government
● Educational Institution
○ Other

System (check all that apply):
☑ Windows
☑ Macintosh
□ UNIX
□ Other

Comments?: I had problems getting drivers for my operating system. The drivers supplied by the manufacturer did not work.

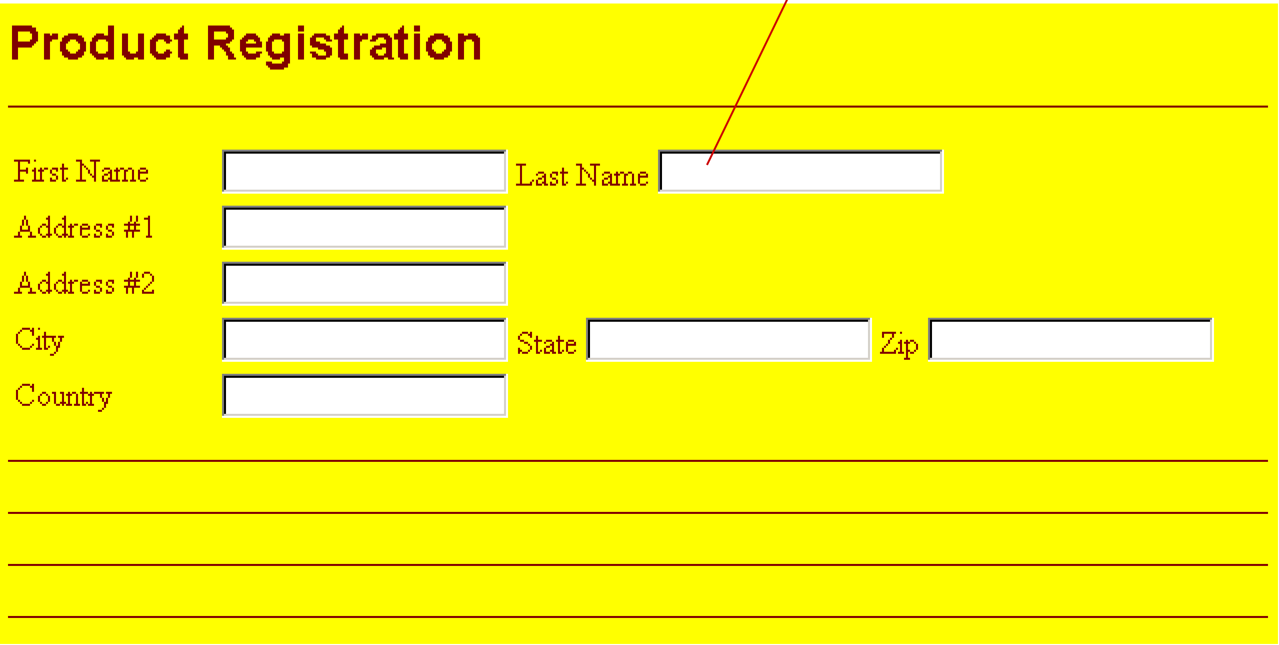
FirstName/Andrew
LastName/Davis
Address 1/Room 304
Address 2/211 Hawkins Avenue ...



Text Boxes on the Form

Text boxes are blank and do not contain any accompanying text, a text description needs to be inserted, such as “Last Name”, adjacent to each box so that the user knows what to enter.

text box



Product Registration

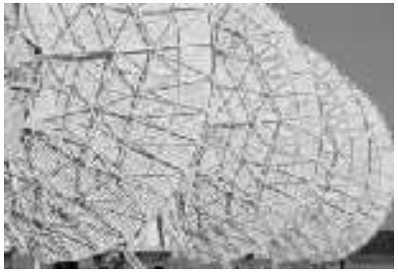
First Name Last Name

Address #1

Address #2

City State Zip

Country

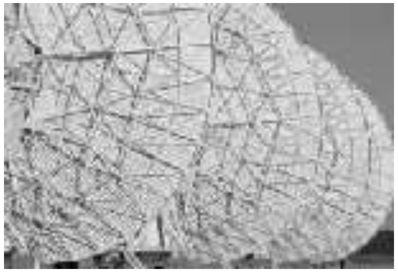


Controlling the Size of a Text Box

- By default, all text boxes are 20 characters wide.
- The syntax for changing the size of a text box is:

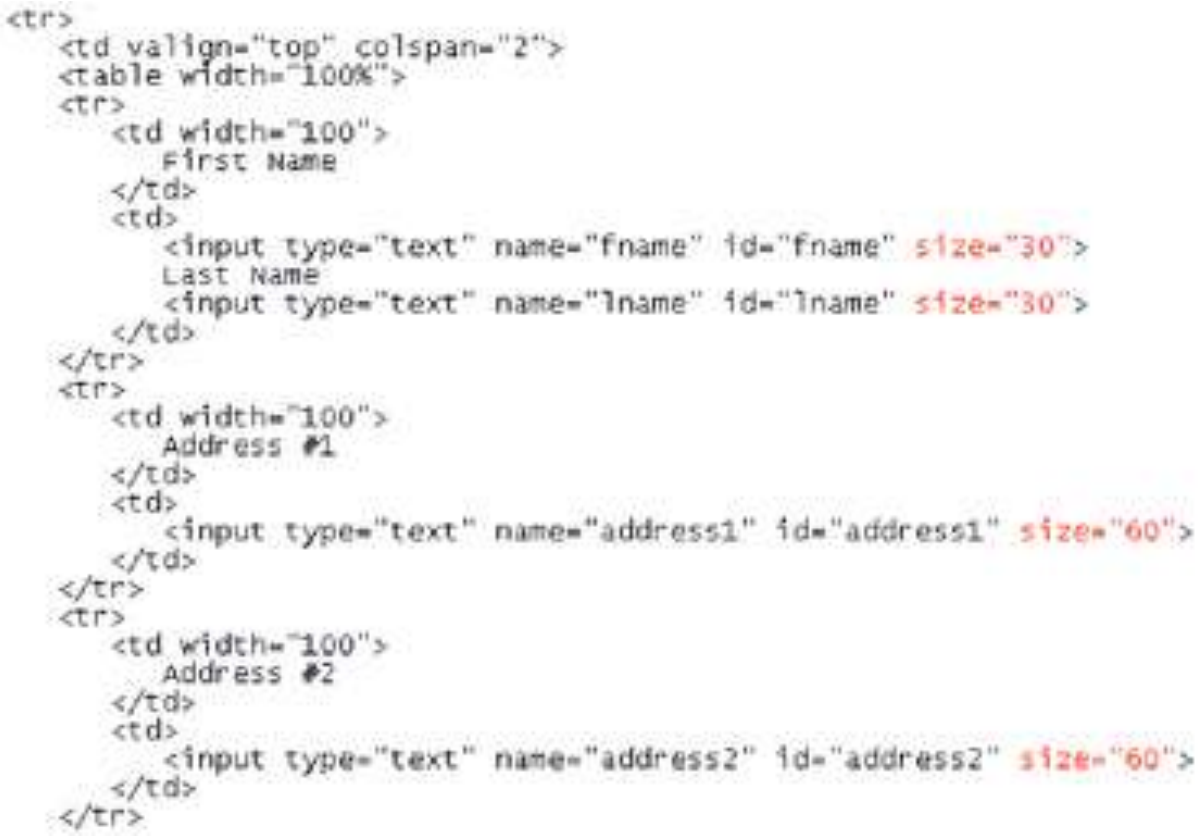
```
<input size=“value”>
```

  - *value* is the size of the text box in characters

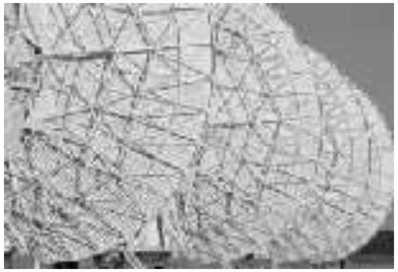


Setting the Width of Text Boxes

This figure shows an example of code for setting the width of text boxes.



```
<tr>
    <td valign="top" colspan="2">
    <table width="100%">
    <tr>
        <td width="100">
            First Name
        </td>
        <td>
            <input type="text" name="fname" id="fname" size="30">
            Last Name
            <input type="text" name="lname" id="lname" size="30">
        </td>
    </tr>
    <tr>
        <td width="100">
            Address #1
        </td>
        <td>
            <input type="text" name="address1" id="address1" size="60">
        </td>
    </tr>
    <tr>
        <td width="100">
            Address #2
        </td>
        <td>
            <input type="text" name="address2" id="address2" size="60">
        </td>
    </tr>
```

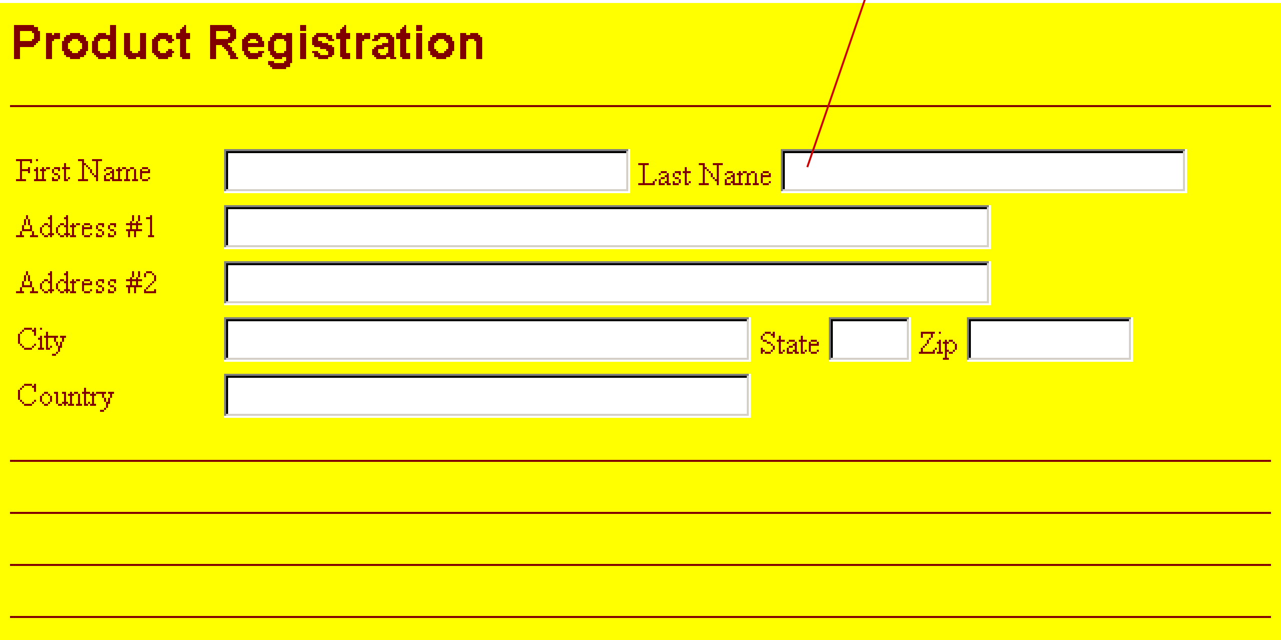


Result of Setting the Width of Text Boxes

Once changes are made, save the changes to a file, and then reload or refresh it in the browser.

Netscape users may have to close and open the file for the changes to the Web form to take effect.

text box width set to 30 characters



Product Registration

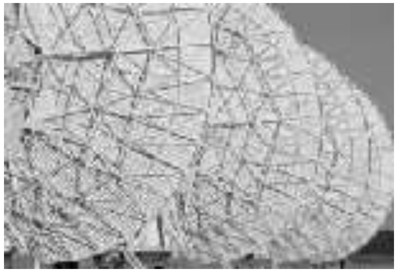
First Name Last Name

Address #1

Address #2

City State Zip

Country

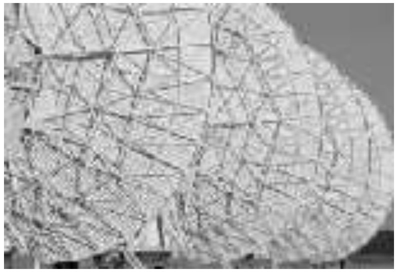


Setting the Maximum Length for Text Input

- Setting the width of a text box does not limit the number of characters the box can hold.
  - if a user enters text longer than the box's width, the text scrolls to the left
  - the user cannot see the entire text, but all of it is sent to the CGI script for processing
- The syntax for setting the maximum length for field input is:

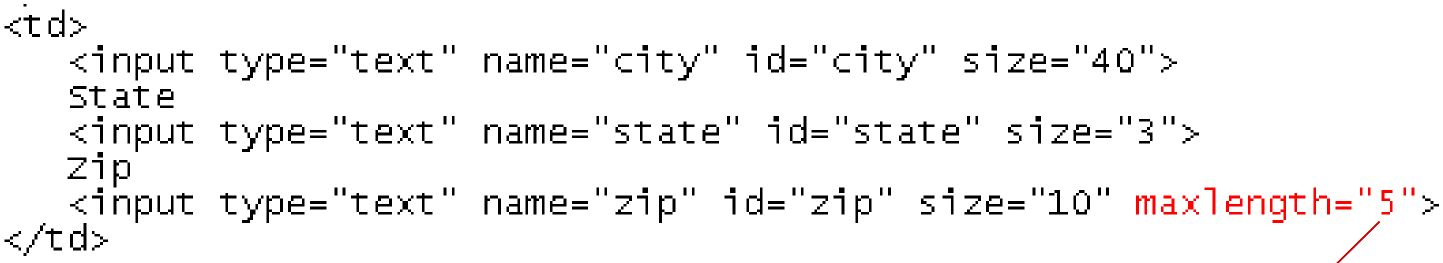
```
<input maxlength="value">
```

  - ***value*** is the maximum number of characters that can be stored in the field



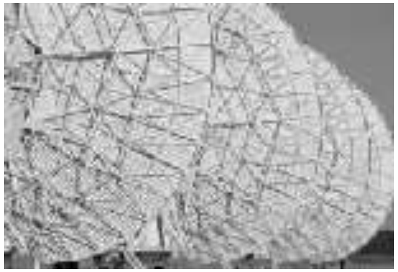
Specify the Maximum Length for a Field

This figure shows an example of limiting the width of the zip code field to five characters.



```
<td>
   <input type="text" name="city" id="city" size="40">
   State
   <input type="text" name="state" id="state" size="3">
   Zip
   <input type="text" name="zip" id="zip" size="10" maxlength="5">
</td>
```

no more than 5 characters are allowed in this text box

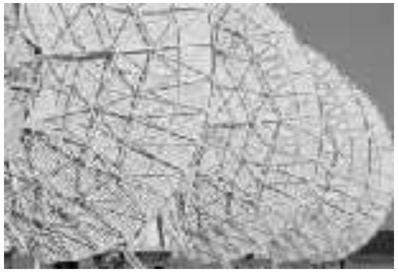


Setting a Default Value for a Field

- When the same value is entered into a field, it may make sense to define a default value for a field.
- Default values can save time and increase accuracy for users of a Web site.
- To define a default value, use the following syntax:

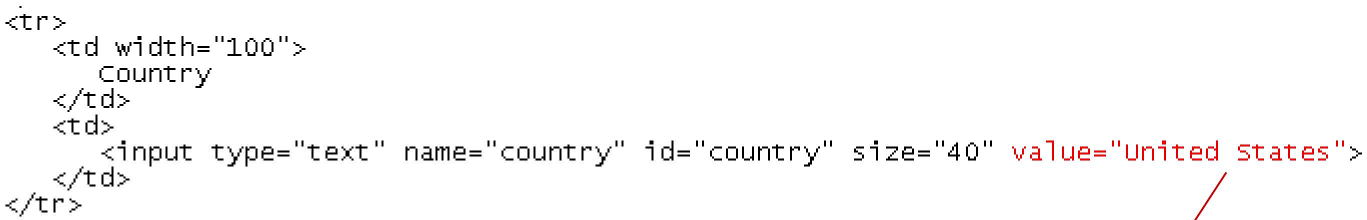
  `<input value=“value”>`

  – ***value*** is the default text or number that is displayed in the field



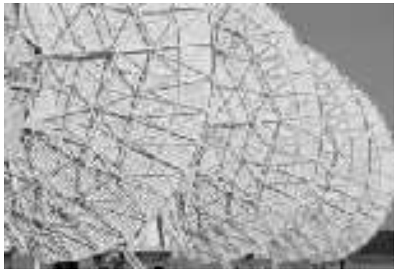
Defining a Default Value for a Field

If customers from countries other than the United States use this Web form, they can remove the default value by selecting the text and pressing the Delete key.



```
<tr>
   <td width="100">
      Country
   </td>
   <td>
      <input type="text" name="country" id="country" size="40" value="United States">
   </td>
</tr>
```

default value

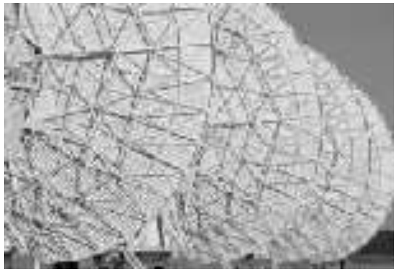


Creating a Password Field

- A **password field** is a text box in which the characters typed by the user are displayed as **bullets** or **asterisks** i.e. ****.
- The syntax for creating a Password field is:

```
<input type="password">
```

- Using a password field should not be confused with having a secure connection.
- The password itself is not encrypted.
- The password field only acts as a mask for a field entry as it is entered.

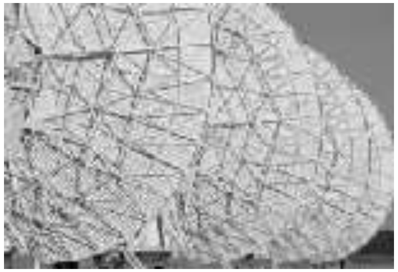


Working with Form Labels

- HTML allows you to formally link a label with an associated text element for scripting purposes.
- The syntax for creating a form label is:

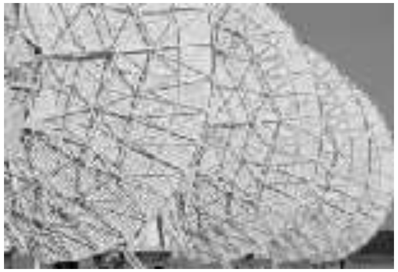
```
<label for=“id”>label text</label>
```

  - *id* is the value of the id attribute for a field on the form
  - ***label*** text is the text of the label
  - you must bind the label to the id attribute of the field and not the name attribute



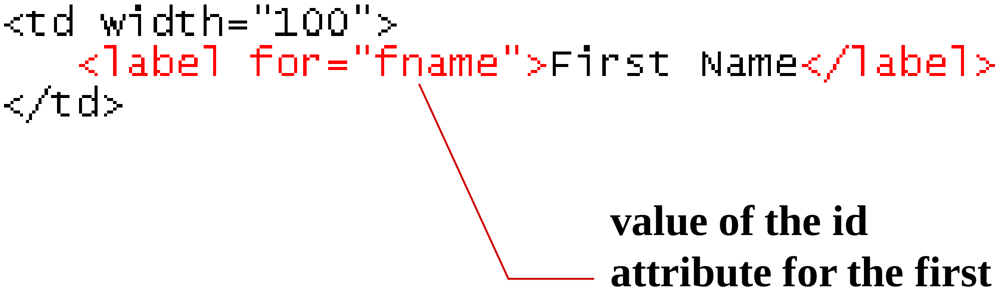
Working with Form Labels

- Labels can simplify the data entry process by allowing a user to click on either the control element or the element’s label to enter data.
- Labels allow users to write scripts to modify their content for interactive forms.
- The `<label>` tag is part of the HTML 4.0 specifications, however, it is not currently supported by **Netscape**.
- The **Netscape** browser, and versions of the **Internet Explorer** browser prior to 4.0, ignore the `<label>` tag, but still display the label text.



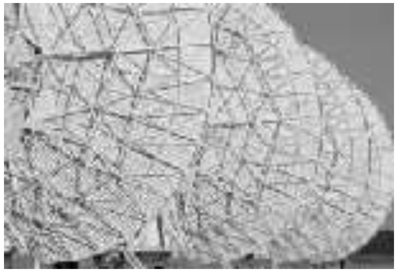
Creating a Label for the `fname` Field

This figure shows a label for the `fname` field.



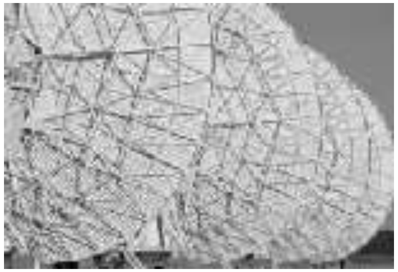
```
<td width="100">
   <label for="fname">First Name</label>
</td>
```

value of the id attribute for the first name field



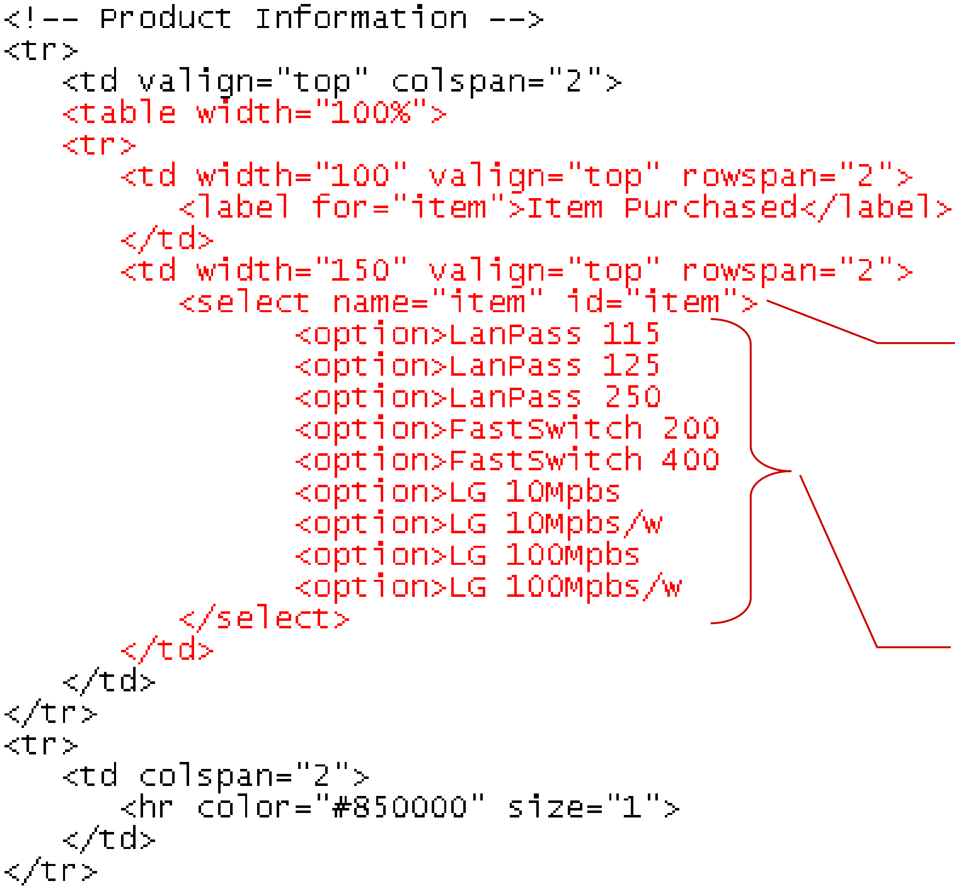
Creating a Selection List

- A **selection list** is a list box from which a user selects a particular value or set of values.
- Selection lists are good to use when there is a fixed set of possible responses.
- Selection lists help prevent spelling mistakes and erroneous entries.
- A selection list is created using the **`<select>`** tag.
- The **`<option>`** tag is used to specify individual selection items.



Creating a Selection List

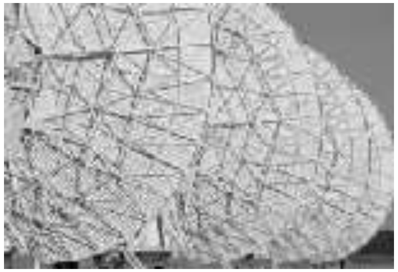
This figure shows HTML code for creating a selection list.



```
<!-- Product Information -->
<tr>
   <td valign="top" colspan="2">
   <table width="100%">
   <tr>
      <td width="100" valign="top" rowspan="2">
         <label for="item">Item Purchased</label>
      </td>
      <td width="150" valign="top" rowspan="2">
         <select name="item" id="item">
               <option>LanPass 115
               <option>LanPass 125
               <option>LanPass 250
               <option>FastSwitch 200
               <option>FastSwitch 400
               <option>LG 10Mpbs
               <option>LG 10Mpbs/w
               <option>LG 100Mpbs
               <option>LG 100Mpbs/w
         </select>
      </td>
   </td>
</tr>
<tr>
   <td colspan="2">
      <hr color="#850000" size="1">
   </td>
</tr>
```

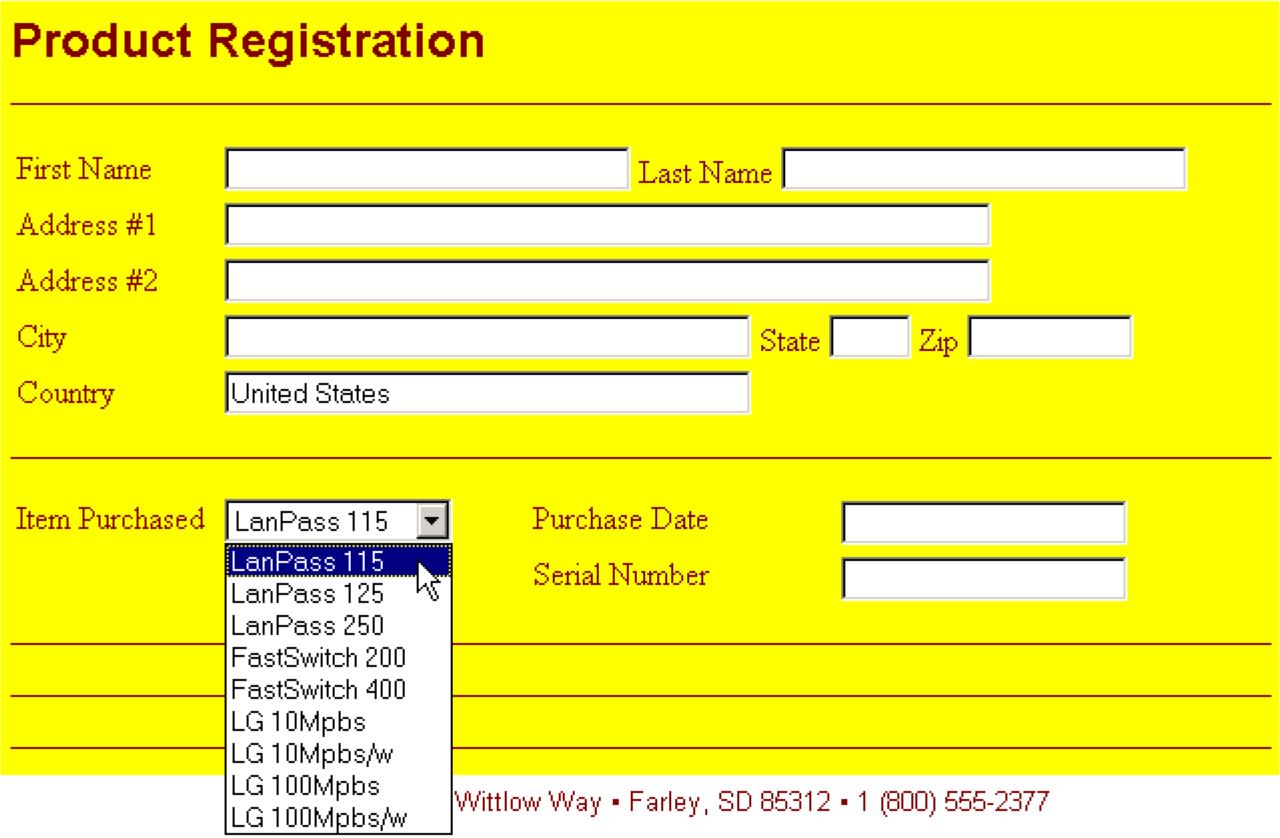
selection list field name

items in the selection list



Using a Selection List

Your selection list might look slightly different depending on the browser and browser version.



Product Registration

First Name Last Name

Address #1

Address #2

City State Zip

Country United States

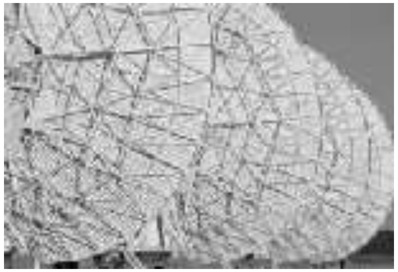
Item Purchased LanPass 115

- LanPass 115
- LanPass 125
- LanPass 250
- FastSwitch 200
- FastSwitch 400
- LG 10Mpbs
- LG 10Mpbs/w
- LG 100Mpbs
- LG 100Mpbs/w

Purchase Date

Serial Number

Wittlow Way • Farley, SD 85312 • 1 (800) 555-2377

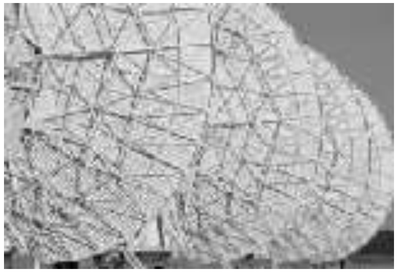


Using the `<select>` and `<option>` Tags

- The general syntax for the **`<select>`** and **`<option>`** tags is:

```
<select name=“name” id=“id”>
<option> item1
  <option> item2
  .
  .
  .
</select>
```

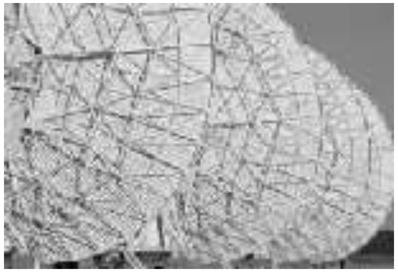
  - ***name*** and ***id*** attribute identify the selection field
  - each **`<option>`** tag represents an individual item in the selection list
  - the text in the selection list is indicated by the text in ***item1***, ***item2***, and so forth
- The **`<option>`** tag is a one-sided tag.



Modifying the Appearance of a Selection List

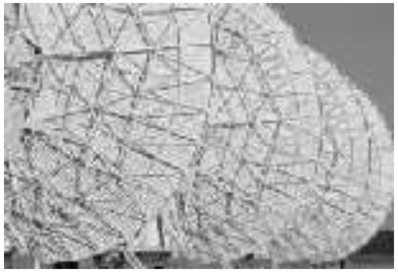
- HTML provides several attributes to modify the appearance and behavior of selection lists and options.
- By default, the **`<select>`** tag displays one option from the selection list, along with a list arrow to view additional selection options.
- The number of options displayed can be modified with the **size** attribute.
- The syntax of the **size** attribute is:

```
<select size=“value”>
```



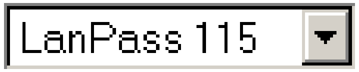
Modifying the Appearance of a Selection List Continued

- – ***value*** is the number of items that the selection list displays in the form
- – by specifying a value greater than 1, the selection list changes from a drop-down list box to a list box
- – if the ***size*** attribute is equal to the number of options in the selection list, the scroll bar is either not displayed or is dimmed



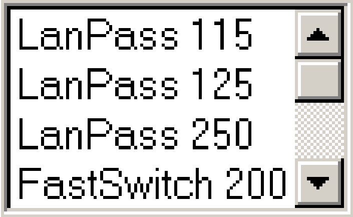
Selection Lists with Different Size Values

This figure shows selection lists with different size values.



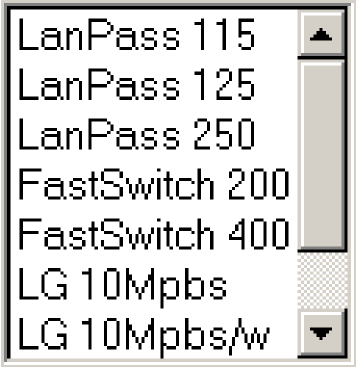
LanPass 115

size = "1"



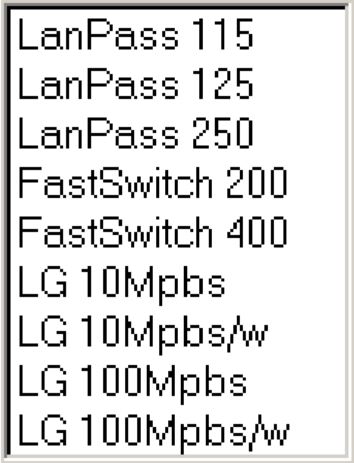
LanPass 115
LanPass 125
LanPass 250
FastSwitch 200

size = "4"



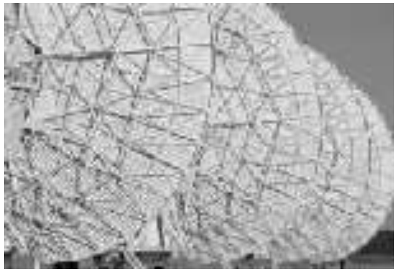
LanPass 115
LanPass 125
LanPass 250
FastSwitch 200
FastSwitch 400
LG 10Mpbs
LG 10Mpbs/w

size = "7"



LanPass 115
LanPass 125
LanPass 250
FastSwitch 200
FastSwitch 400
LG 10Mpbs
LG 10Mpbs/w
LG 100Mpbs
LG 100Mpbs/w

size = "9"

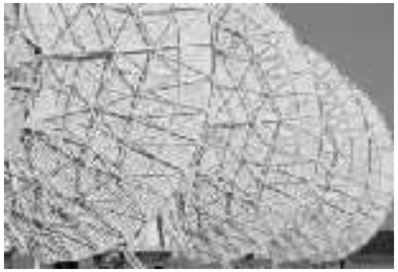


Making Multiple Selections

- When using multiple selections, users are not limited to a single selection from a selection list.
- Adding the **multiple** attribute to the **`<select>`** tag allows multiple selections from a list.
- The syntax for this attribute is:

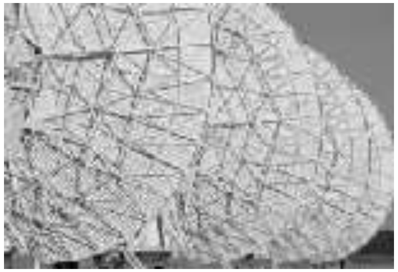
```
<select multiple>
```

- A common method to make multiple selections from a selection list is to hold down a specific key while making selections.



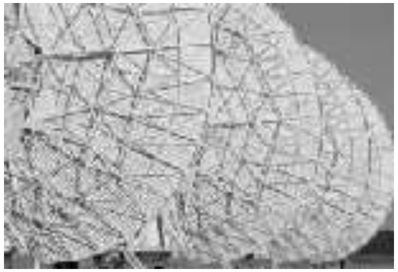
Making Multiple Selections Continued

- The Windows operating system, multiple selections can be made as follows:
  - for **noncontiguous selections**, press and hold the Ctrl key while you make your selections
  - for a **contiguous selection**, select the first item, press and hold the Shift key, and then select the last item in the range
- Check and verify that the CGI scripts used are designed to handle multiple selection lists.



Working with Option Values

- By default, a form sends the values that are displayed in the selection list to the CGI script.
- Instead of sending an entire text string, an abbreviation or code can be sent to the CGI script.
- Specify the value that is sent to the CGI script with the **value** attribute.
- Use the **selected** attribute to specify which item in the selection is selected, or highlighted, when the form is displayed.

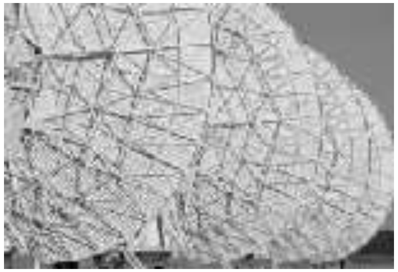


Working with Option Groups

- The most recent releases of HTML allows you to organize selection lists into distinct groups called **option groups**.
- The syntax for creating an option group is:

```
<optgroup label=“label”>
```

  - ***label*** is the label assigned to the option group
  - the text for the label appears in the selection list above each group of items but is not a selectable item from the list



Option Groups

Internet Explorer and Netscape versions prior to 6.0 display the selection list without the group labels.

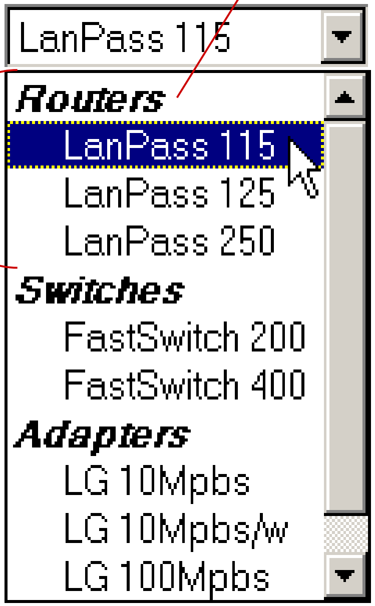
```
<select name="item" id="item">
   <optgroup label="Routers">
      <option>LanPass 115
      <option>LanPass 125
      <option>LanPass 250
   </optgroup>
   <optgroup label="Switches">
      <option>FastSwitch 200
      <option>FastSwitch 400
   </optgroup>
   <optgroup label="Adapters">
      <option>LG 10Mpbs
      <option>LG 10Mpbs/w
      <option>LG 100Mpbs
      <option>LG 100Mpbs/w
   </optgroup>
</select>
```

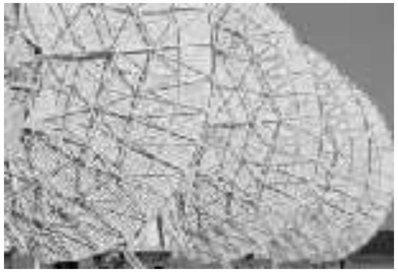
a single option group

option group label

a single option group

option group label



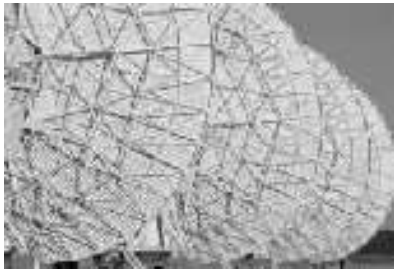


Working with Radio Buttons

- **Radio buttons** display a list of choices from which a user makes a selection.
- Only one radio button can be selected at a time.
- The syntax to create a radio button is:

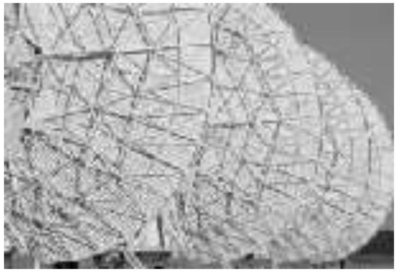
```
<input type=“radio” name=“name” id=“id”
  value=“value”>
```

  - ***name*** identifies the field containing the radio button
  - ***id*** attribute identifies the specific option. Only required if you intend to use a field label with the radio button
  - ***value*** attribute indicates the value sent to the CGI script, if that radio button is selected by the user



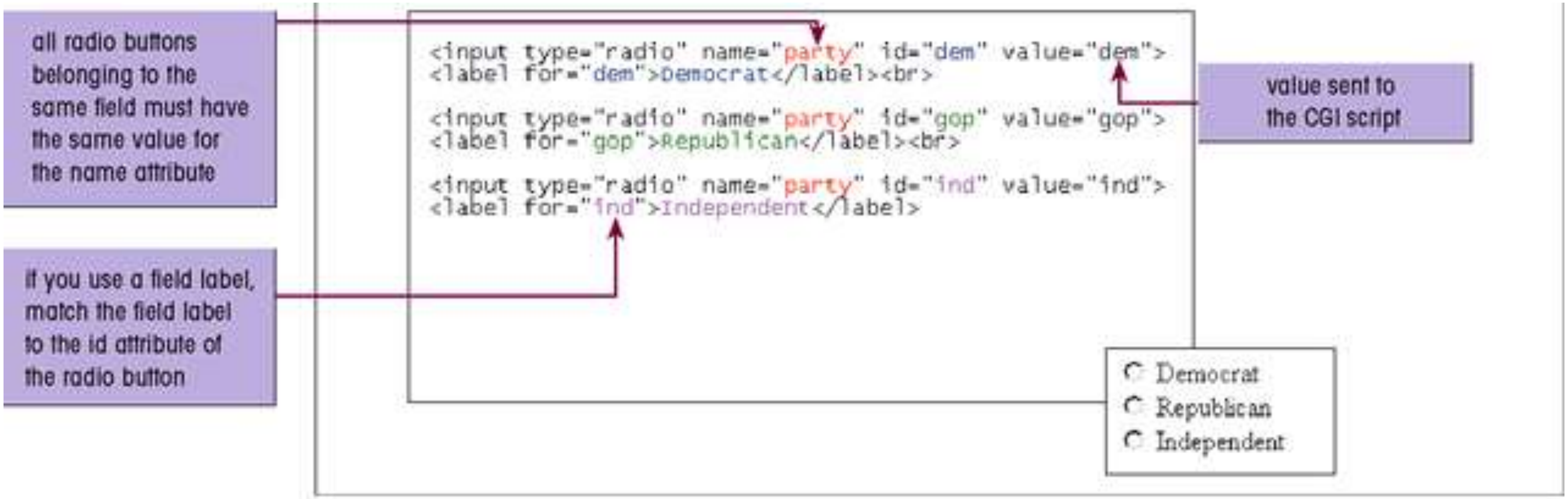
Working with Radio Buttons Continued

- The **`name`** attribute must be included, because it groups distinct radio buttons together.
  - selecting one radio button in the group automatically deselects all of the other radio buttons in that group
- Insert descriptive text next to the button.
- Enclose text within a label tag to allow the user to select the radio button or label.



Creating Radio Buttons

This figure shows an example of HTML code that creates radio buttons for party affiliations.



all radio buttons belonging to the same field must have the same value for the name attribute

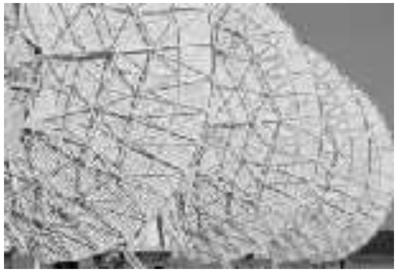
```
<input type="radio" name="party" id="dem" value="dem">
<label for="dem">Democrat</label><br>

<input type="radio" name="party" id="gop" value="gop">
<label for="gop">Republican</label><br>

<input type="radio" name="party" id="ind" value="ind">
<label for="ind">Independent</label>
```

value sent to the CGI script

if you use a field label, match the field label to the id attribute of the radio button

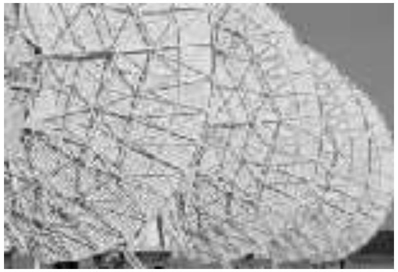
○ Democrat
○ Republican
○ Independent

In this sample code, the value sent to the CGI script does not match the field label.
If the user selects the Republican radio button, the value “gop”
is sent to the CGI script paired with the field name “party.”



Selection Lists vs. Radio Buttons

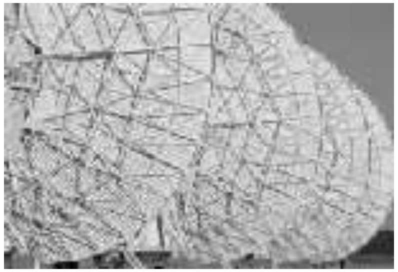
- If you have a long list of options, use a selection list.
- If you want to allow users to select more than one option, use a selection list with the multiple attribute.
- If you have a short list of options, and only one option is allowed at a time, use radio buttons.



Creating a Group Box

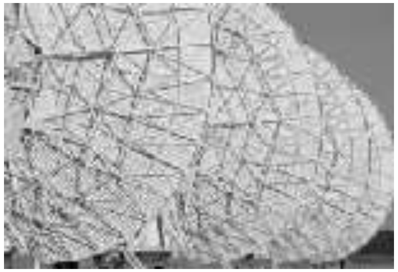
- A **group box** labels an entire collection of radio buttons.
- A group box is a box placed around a set of fields that indicates that they belong to a common group.
- The syntax for creating a group box is:

```
<fieldset>
  <legend align=“align”>legend
  text</legend>
  collection of fields
</fieldset>
```



Creating a Group Box Continued

- the **`<legend>`** tag is used to display a legend on the group box
- ***legend text*** specifies the text for that legend
- the ***align*** attribute specifies where the legend is placed in the box
  - align values are “**top**” (the default), “**bottom**”, “**left**”, and “**right**”
  - browsers only support “**top**” and “**right**” options at this time

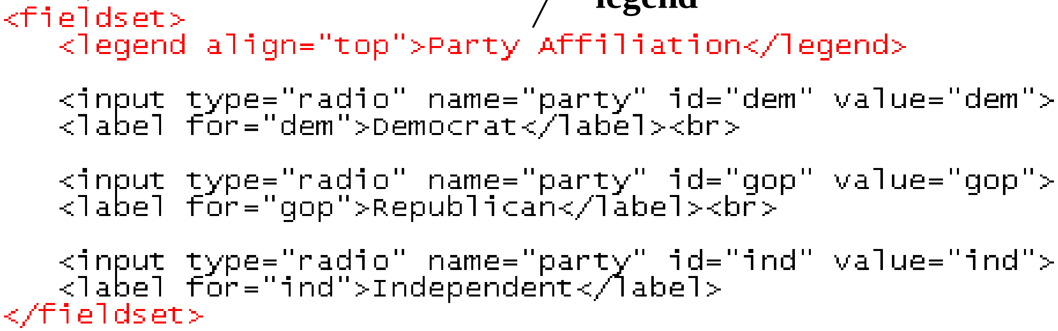


Creating a Group Box and Legend

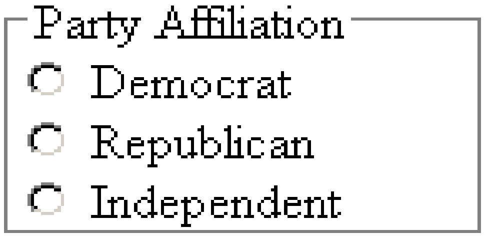
This figure shows an example of a group box applied to a set of radio buttons.

start of group box

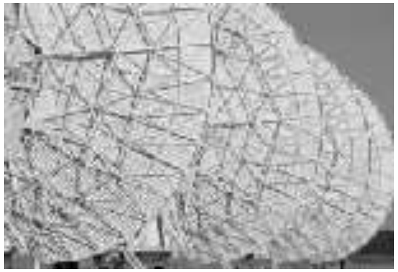
group box legend



```
<fieldset>
   <legend align="top">Party Affiliation</legend>

   <input type="radio" name="party" id="dem" value="dem">
   <label for="dem">Democrat</label><br>

   <input type="radio" name="party" id="gop" value="gop">
   <label for="gop">Republican</label><br>

   <input type="radio" name="party" id="ind" value="ind">
   <label for="ind">Independent</label>
</fieldset>
```

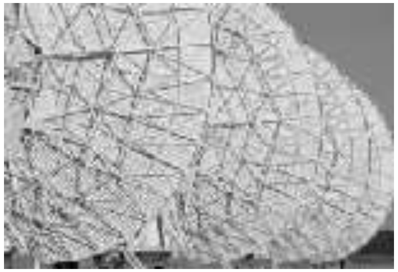


resulting radio buttons and group box



Group Box Size

- There is no attribute to control the size of a group box.
- The box’s height will be large enough to accommodate the fields and labels in the field set.
- The width is the width of whatever space remains on the Web page.
- To set the width to a specific value use a table cell to place the group box and set the width of the cell.
  - group boxes cannot extend across table cells; all of the fields in the field set must be placed within a single cell



Working with Check Boxes

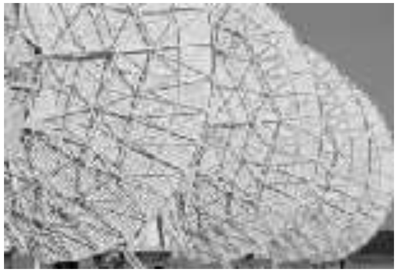
- A **check box** is either selected or not, there is only one check box per field.
- Check boxes are created using the following syntax:

  ```
  <input type=“checkbox” name=“name” id=“id”
    value=“value”>
  ```

  – ***name*** and ***id*** attribute identify the check box

  – the ***value*** attribute specifies the value that is sent to the CGI script when the check box is selected
- The **`<input>`** tag for a check box does not display any text.
- Check boxes are not selected by default.

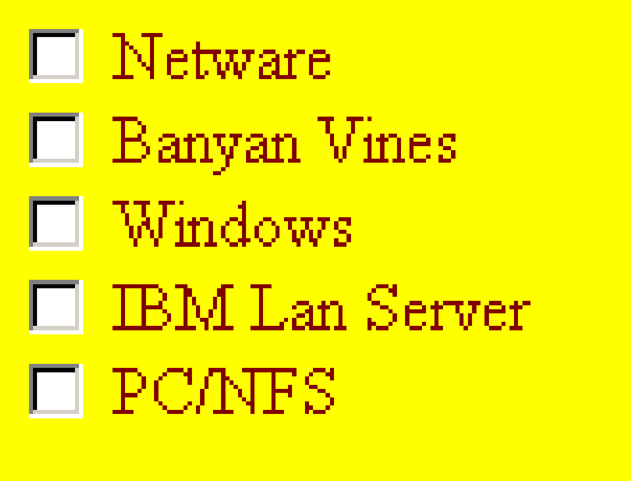
  – to do this, add the **checked** attribute to the **`<input>`** tag



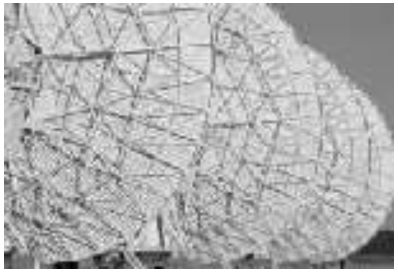
Adding Check Boxes

This figure shows different check boxes. The user can click either the check box or the label associated with the check box to select and deselect.

```
      <input type="radio" name="use" id="edu" value="edu">
      <label for="edu">Educational Institution</label>
      </fieldset>
   </td>
   <td valign="top">
      <input type="checkbox" name="nw" id="nw" value="yes">
      <label for="nw">Netware</label><br>

      <input type="checkbox" name="bv" id="bv" value="yes">
      <label for="bv">Banyan Vines</label><br>

      <input type="checkbox" name="win" id="win" value="yes">
      <label for="win">Windows</label><br>

      <input type="checkbox" name="ibm" id="ibm" value="yes">
      <label for="ibm">IBM Lan Server</label><br>

      <input type="checkbox" name="pcnfs" id="pcnfs" value="yes">
      <label for="pcnfs">PC/NFS</label>
   </td>
</tr>
</table>
```

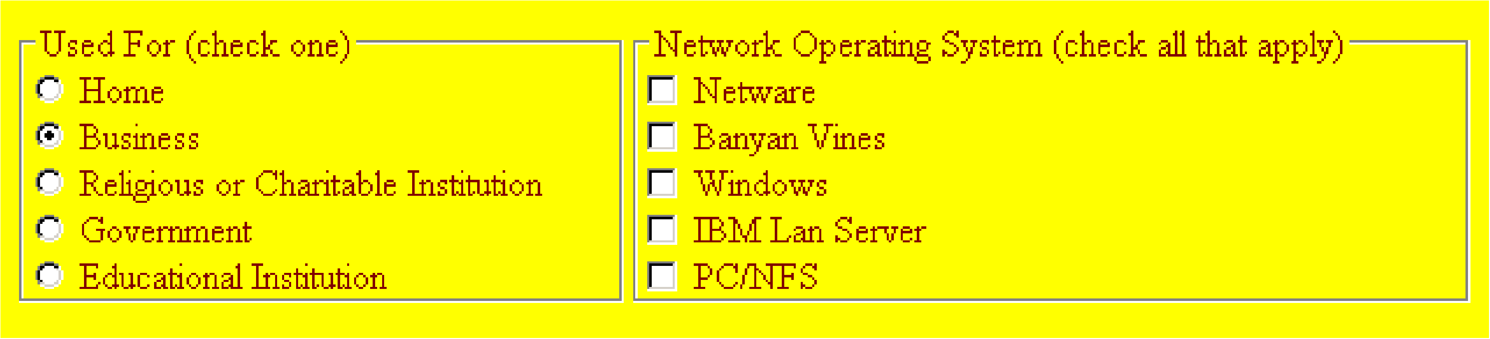


□ Netware
□ Banyan Vines
□ Windows
□ IBM Lan Server
□ PC/NFS



Group Boxes for Radio Buttons and Check Boxes

```
<fieldset>
<legend align="top">Network Operating System (check all that apply)</legend>
<input type="checkbox" name="nw" id="nw" value="yes">
<label for="nw">Netware</label><br>

<input type="checkbox" name="bv" id="bv" value="yes">
<label for="bv">Banyan Vines</label><br>

<input type="checkbox" name="win" id="win" value="yes">
<label for="win">Windows</label><br>

<input type="checkbox" name="ibm" id="ibm" value="yes">
<label for="ibm">IBM Lan Server</label><br>

<input type="checkbox" name="pcnfs" id="pcnfs" value="yes">
<label for="pcnfs">PC/NFS</label>
</fieldset>
```

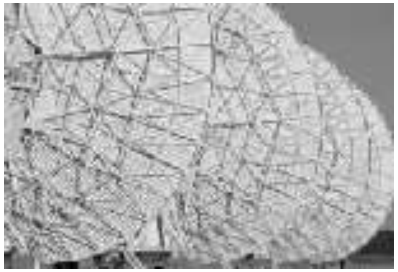


Used For (check one)

○ Home
● Business
○ Religious or Charitable Institution
○ Government
○ Educational Institution

Network Operating System (check all that apply)

□ Netware
□ Banyan Vines
□ Windows
□ IBM Lan Server
□ PC/NFS



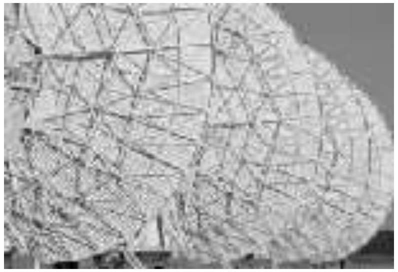
Creating a Text Area

- To create a larger **text area** for a text box, use the tag:

```
<textarea name=“name” id=“id” rows=“value”
  cols=“value”> default text </textarea>
```

  - ***rows*** and ***cols*** attributes define the dimensions of the text box
  - the ***rows*** attribute indicates the number of lines in the text box

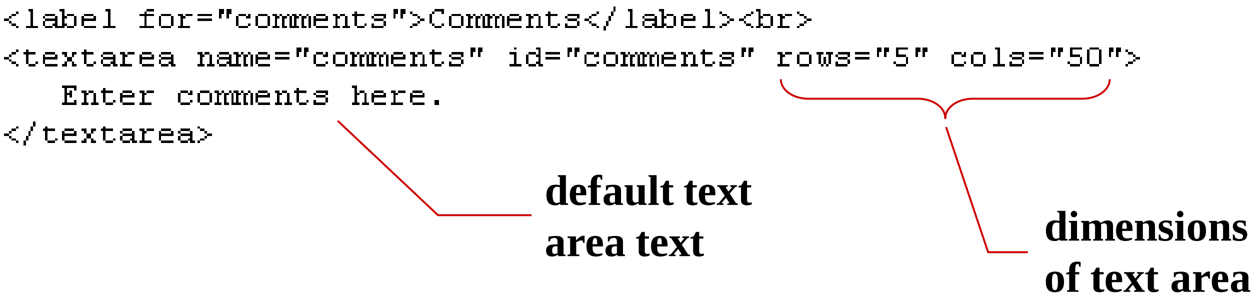
- Default text can be specified in the text box when the form is initially displayed.



Creating a Text Area

This figure shows an example of a text area with default text.

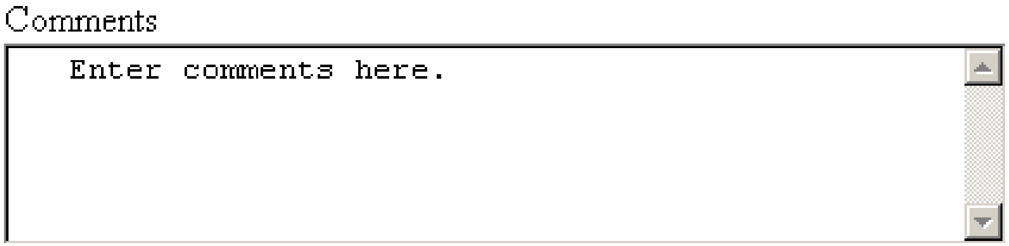
The `<textarea>` tag is a two-sided tag, which means that it has an opening tag `<textarea>`, and a closing tag, `</textarea>`.



```
<label for="comments">Comments</label><br>
<textarea name="comments" id="comments" rows="5" cols="50">
   Enter comments here.
</textarea>
```

default text area text

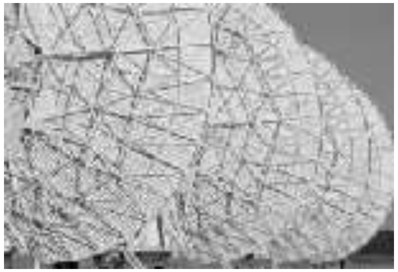
dimensions of text area



Comments

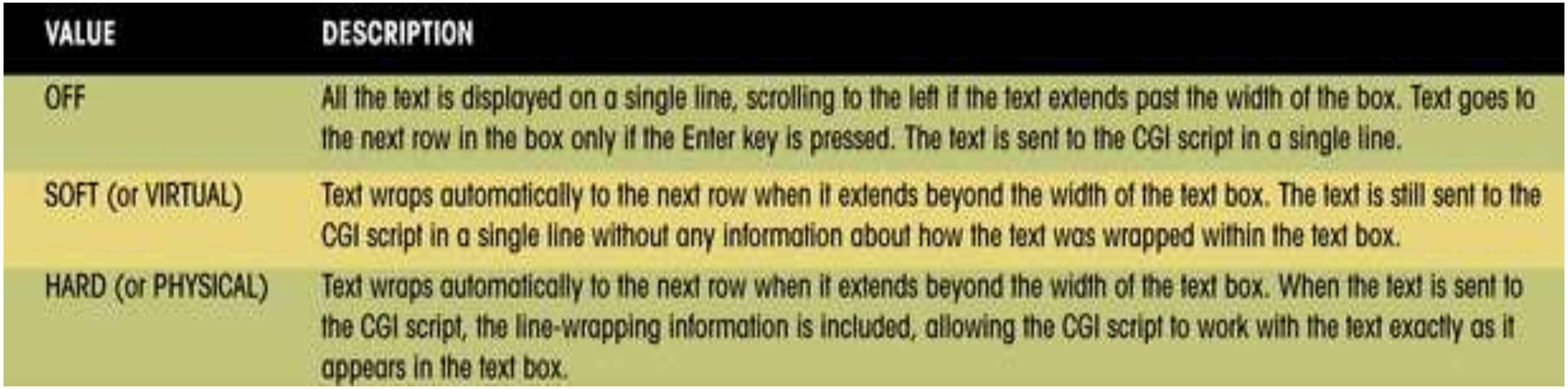
```
   Enter comments here.
```

resulting text area

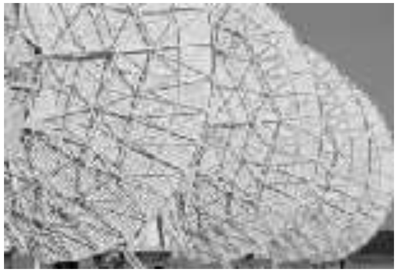


Wrap Attribute Values

The text entered in a text area wraps to the next line when it exceeds the width of the box. To control how a browser wraps text to a new line use the wrap attribute. This figure shows the three possible wrap options.

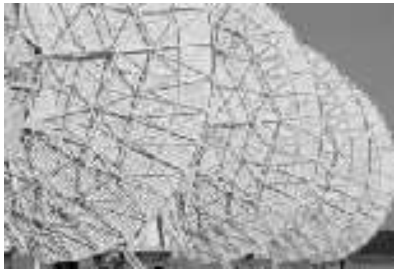


VALUE	DESCRIPTION
OFF	All the text is displayed on a single line, scrolling to the left if the text extends past the width of the box. Text goes to the next row in the box only if the Enter key is pressed. The text is sent to the CGI script in a single line.
SOFT (or VIRTUAL)	Text wraps automatically to the next row when it extends beyond the width of the text box. The text is still sent to the CGI script in a single line without any information about how the text was wrapped within the text box.
HARD (or PHYSICAL)	Text wraps automatically to the next row when it extends beyond the width of the text box. When the text is sent to the CGI script, the line-wrapping information is included, allowing the CGI script to work with the text exactly as it appears in the text box.



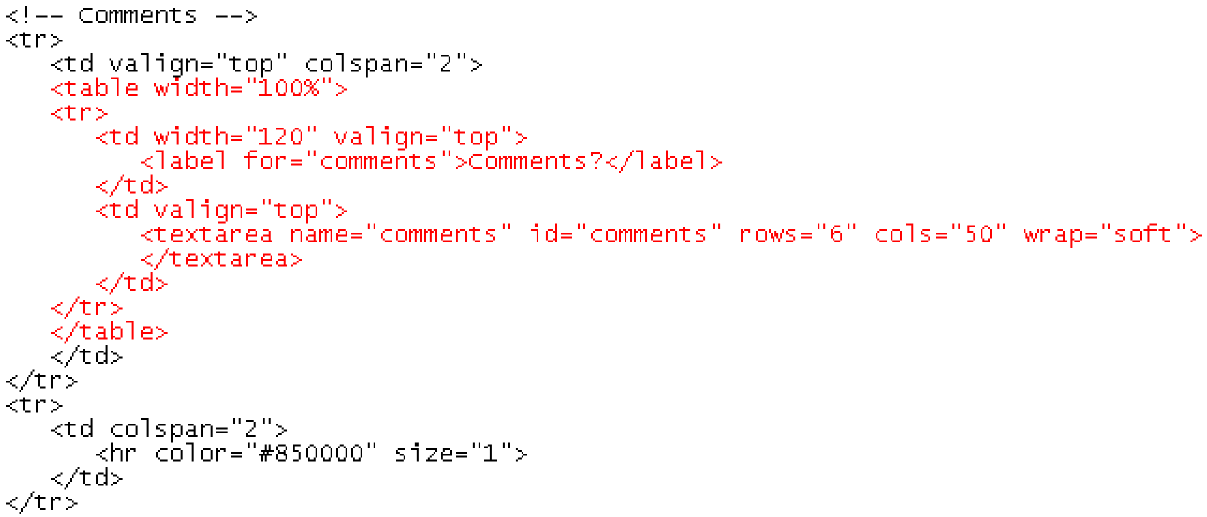
The `wrap` Attribute for Text

- Set the value of the **`wrap`** attribute to either **“soft”** or **“hard”** to allow text to wrap within the text box.
  - the “**hard**” setting preserves any line wrapping that takes place in the text box and the “soft” setting does not.
- If no value for the wrap attribute is specified, a value of “**soft**” is used.
- For comment fields, use the **`<textarea>`** tag with the wrap attribute set to “soft” so that the user’s comments wrap to the next line in the box.

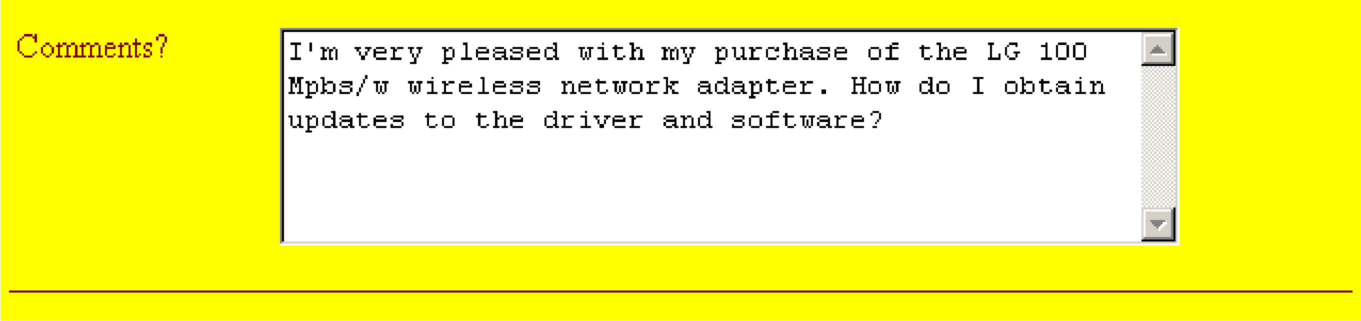


Comment Text Area

In this figure the text box includes a vertical scroll bar, so that a user can scroll to see the hidden text, if needed.

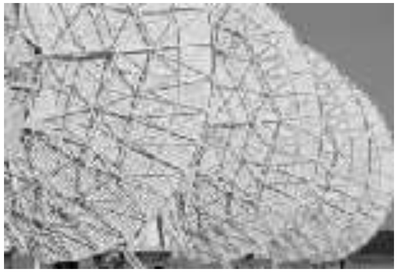


```
<!-- Comments -->
<tr>
   <td valign="top" colspan="2">
   <table width="100%">
   <tr>
      <td width="120" valign="top">
         <label for="comments">Comments?</label>
      </td>
      <td valign="top">
         <textarea name="comments" id="comments" rows="6" cols="50" wrap="soft">
         </textarea>
      </td>
   </tr>
   </table>
   </td>
</tr>
<tr>
   <td colspan="2">
      <hr color="#850000" size="1">
   </td>
</tr>
```



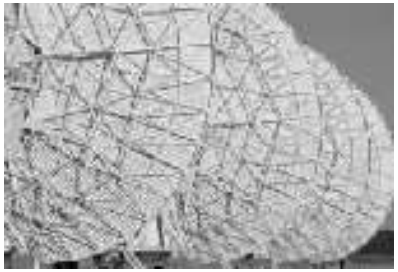
Comments?

I'm very pleased with my purchase of the LG 100 Mpbs/w wireless network adapter. How do I obtain updates to the driver and software?



Creating Form Buttons

- Another type of control element is one that performs an action. In forms, this is usually done with a **button**.
- Buttons can be clicked to:
  - run programs
  - submit forms
  - reset the form to its original state

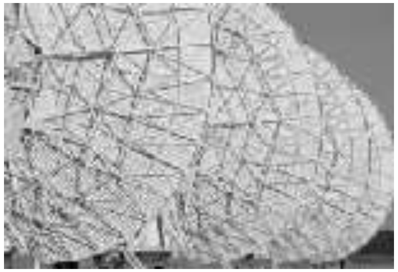


Creating a Push Button

- One type of button, called a **push button**, is created using the **`<input>`** tag as follows:

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

  - ***text*** is the text that appears on the button
- By themselves, push buttons perform no actions in the Web page.
- To create an action, write a script or program that runs automatically when the button is clicked.

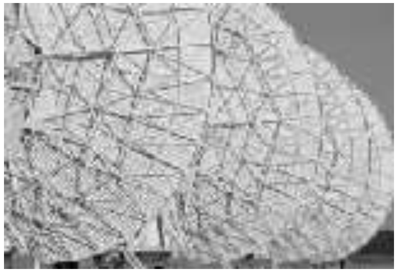


Creating Submit and Reset Buttons

- A **submit button** is a button that submits the form to the CGI script for processing.
- A **reset button** resets the form to its original (default) values.
- The syntax for creating these two buttons is:

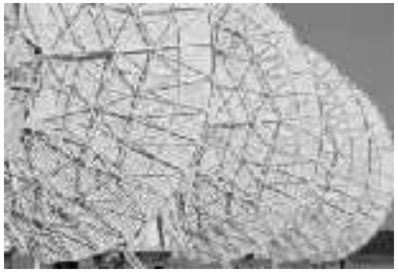
```
<input type=“submit” value=“text”>
<input type=“reset” value=“text”>
```

  - ***value*** attribute defines the text that appears on the button



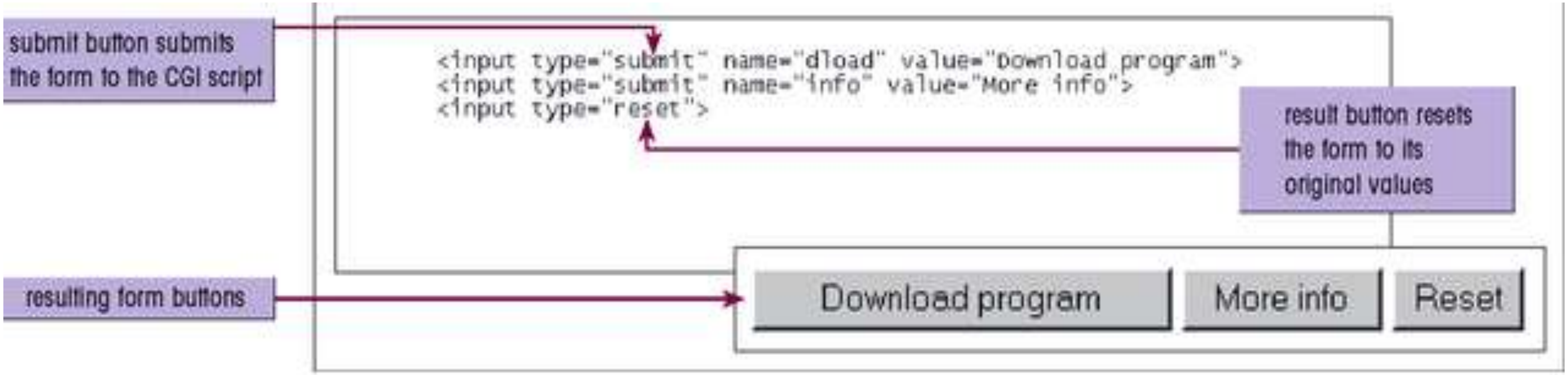
The `name` and `value` Attributes

- The **`name`** and **`value`** attributes can be used for **push**, **submit**, and **reset** buttons.
- Use these attributes when the form contains multiple buttons and a program that processes the form needs to distinguish one button from another.
  - for example, a Web page advertising a shareware program might include three buttons:
    - one used to download the program
    - another used to retrieve information
    - the third to cancel the form



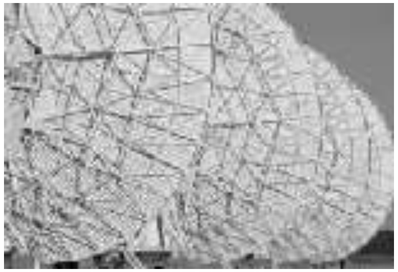
Creating a Form Button

The figure shows HTML tags for buttons that download a program, retrieves information, and resets the form to its original values.



```
<input type="submit" name="dload" value="Download program">
<input type="submit" name="info" value="More info">
<input type="reset">
```

submit button submits the form to the CGI script
result button resets the form to its original values
resulting form buttons
Download program
More info
Reset

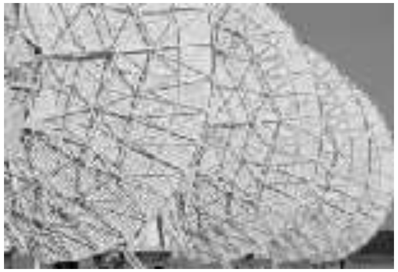


Creating Buttons with the `<button>` Tag

- Buttons created with the **`<input>`** tag do not allow the Web page designer to control the appearance of the button.
- For greater artistic control over the appearance of the button, use the **`<button>`** tag.
- The syntax of the **`<button>`** tag is:

```
<button name=“name” value=“value”
  type=“option”>
  button text and HTML tags
</button>
```

  - ***name*** attribute specifies the name of the button
  - ***value*** attribute sends to a CGI script
  - ***type*** attribute specifies the button type (submit, reset, or button)

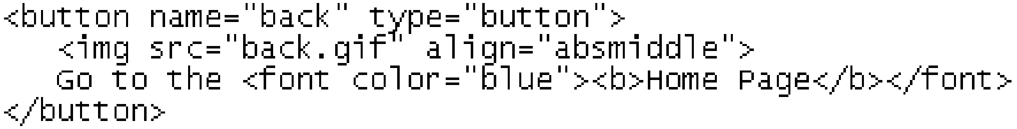


Using the `<button>` Tag

The figure shows how to create a button that contains formatted text and an inline image.

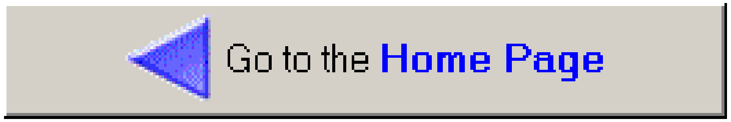
The default value for the type attribute is "button". Within the `<button>` tags you can place whatever HTML tags you wish to format the button's appearance. This includes inline images.

the button type is a simple push button

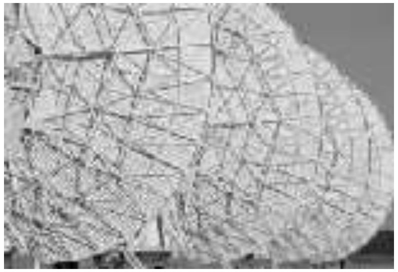


```
<button name="back" type="button">
   <img src="back.gif" align="absmiddle">
   Go to the <font color="blue"><b>Home Page</b></font>
</button>
```

contents of the button



button image

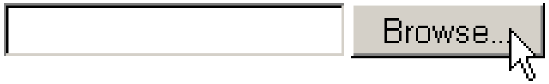


Using a File Button

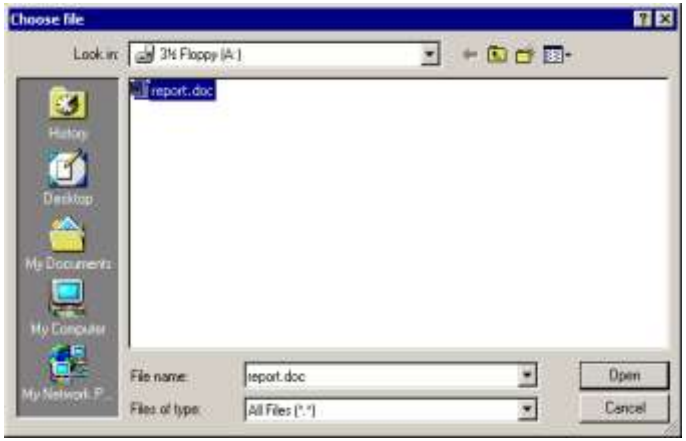
The figure shows an example of using the file button to return the location of a file named “report.doc.”

```
<input type="file" name="file_name">
```

1. User clicks the Browse button



2. Selects a file from the Choose File dialog box


Choose file

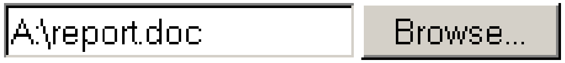
Look in: 3½ Floppy (A:)

report.doc

File name: report.doc

Files of type: All Files (*.*)

3. The filename and location are automatically placed in the text box


A:\report.doc