

HTML, XHTML, and XML

3rd Edition

Tutorial 6

Working with Web Forms

**Thanks to the author of the textbook for providing these
slides. I made slight changes/additions.**

Turgay Korkmaz





Objectives

- Explore how Web forms interact with Web servers
- Create form elements
- Create field sets and legends
- Create input boxes and form labels
- Creation option buttons
- Create selection lists
- Create check boxes

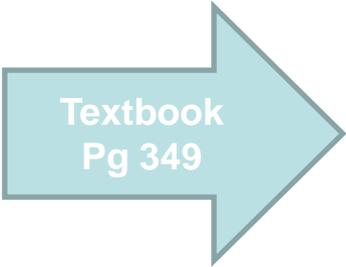


Objectives

- Create text area boxes
- Apply styles to Web forms
- Work with form buttons
- Explore image elements and hidden fields
- Work with form actions and methods



Tutorial_06/tutorial





The Lighthouse 150 Cavates Rd. St. Peters, MO 63376

home
projects
upcoming events
community links
staff
donations
volunteers
contact info

The success of The Lighthouse reflects the dedication and support of members of the community who have helped make our dream a reality. We cannot continue to operate without contributions from people like you.

You can make a tax-deductible donation online using your American Express, Discover, Master, or Visa card. Please fill out the form on this page.

The Lighthouse is always looking for volunteers. We especially need help in the following areas:

- Mechanics
- Carpenters
- Electricians
- Cooks
- Computer technicians
- Babysitters
- Data entry persons

and many others. Please consider donating your time and talents to your community and your neighbors.

Thank you so much for your generosity!

— Terry Ives
Director, The Lighthouse

Donation Form

▪ indicates required information

Contact Information

Address for home business

First Name

Last Name

Phone

Street Address

City State Zip

Donation Information

Donation Amt.

Credit Card

Card Holder Name

Card Number

Expiration Date

Feedback

I'm interested in volunteering at The Lighthouse

Comments

insert the Web form in the page's right column

The Lighthouse • 150 Cavates Rd. • St. Peters, MO 63376 • (636) 555 - 4477



Introducing Web Forms

- Web forms collect information from Web site visitors.
- Web forms include different control elements including:

- Input boxes
- Option buttons or radio buttons
- Selection lists
- Check boxes
- Text areas

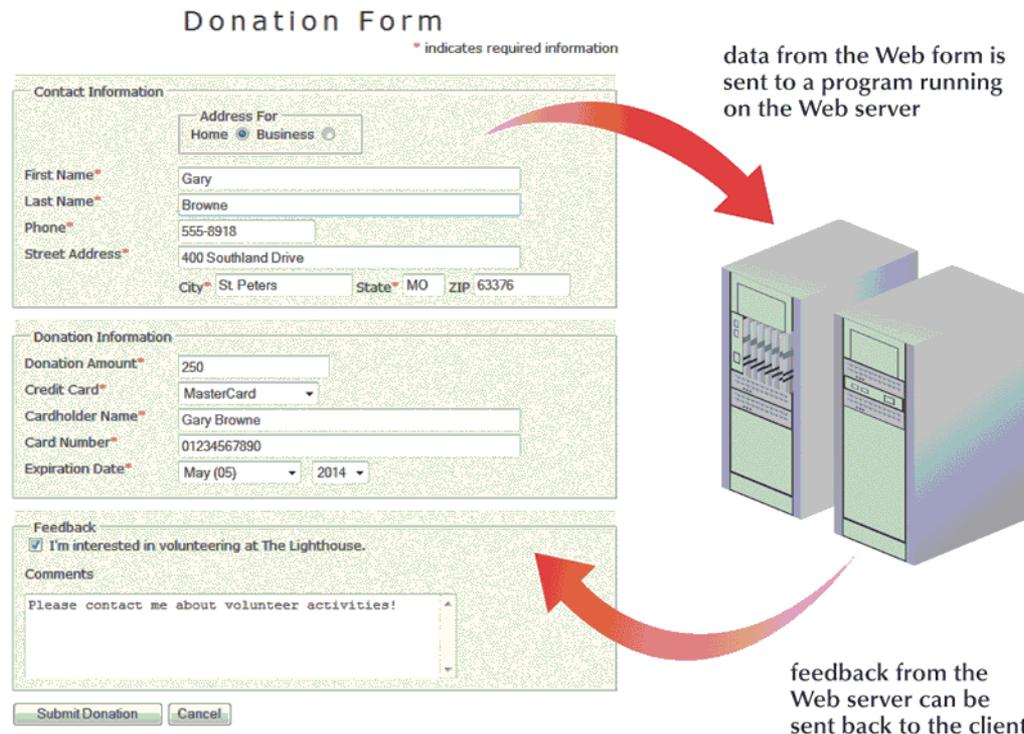
The diagram illustrates a web form with three main sections: Contact Information, Donation Information, and Feedback. Callouts identify various control elements:

- legend**: Points to the 'Contact Information' section header.
- option buttons**: Points to the radio buttons for 'home' and 'business' under 'Address for'.
- input boxes**: Points to the text input fields for 'First Name', 'Last Name', 'Phone', 'Street Address', 'City', 'State', and 'Zip'.
- selection lists**: Points to the dropdown menus for 'Credit Card' (showing 'American Express') and 'Expiration Date' (showing 'January (01)' and '2014').
- check box**: Points to the checkbox for 'I'm interested in volunteering at The Lighthouse'.
- text area box**: Points to the large text input area for 'Comments'.
- form button**: Points to the 'Submit Donation' and 'Cancel' buttons at the bottom.



Forms and Server-Based Programs

- While HTML supports the creation of forms, it does not include tools to process the information.
- The information can be processed through a program running on a Web server.





Forms and Server-Based Programs

- Server-based programs are written in many languages
- The earliest and most commonly used are **Common Gateway Interface (CGI) scripts** that are written in **Perl**.
- Other popular languages include:
 - ASP
 - ColdFusion
 - C/C++
 - PHP
 - VBScript



Creating a Web Form

- Forms are created using the form element, structured as follows:

```
<form attributes>  
elements  
</form>
```

```
<div id="rightcolumn">  
  <h1>Donation Form</h1>  
  <p><span>*</span> indicates required information</p>  
  <form name="donationForm" id="donationForm">  
  </form>  
</div>
```

- Where *attributes* are the attributes that control how the form is processed and *elements* are elements placed within the form.



Creating a Web Form

- Form attributes usually tell the browser the location of the server-based program to be applied to the form's data. (more at the end of these slides)

```
<form action="url" method="type">
```

...

```
</form>
```

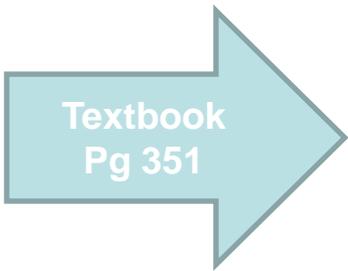
- Always specify an `id` or `name` for the form.
The syntax of the `id` and `name` attributes are as follows:

```
<form name="name" id="id">... </form>
```

Where *name* is the name of the form and *id* is the id of the form.



Creating a Field Set



- HTML and XHTML allow you to organize option buttons into a group of fields called field sets.

```
<fieldset id="id">  
  controls  
</fieldset>
```

where id identifies the field set and controls are the control elements associated with fields within the field set

```
<form name="donationForm" id="do  
  <fieldset id="contact">  
  </fieldset>  
  
  <fieldset id="donation">  
  </fieldset>  
  
  <fieldset id="feedback">  
  </fieldset>  
</form>
```

Contact Information

Address For
Home Business

First Name* Gary
Last Name* Browne
Phone* 555-6918
Street Address* 400 Southland Drive
City* St Peters State* MO ZIP 63376

Donation Information

Donation Amount* 250
Credit Card* MasterCard
Cardholder Name* Gary Browne
Card Number* 01234567890
Expiration Date* May (05) 2014

Feedback

I'm interested in volunteering at The Lighthouse.

Comments

Please contact me about volunteer activities!

Submit Donation Cancel



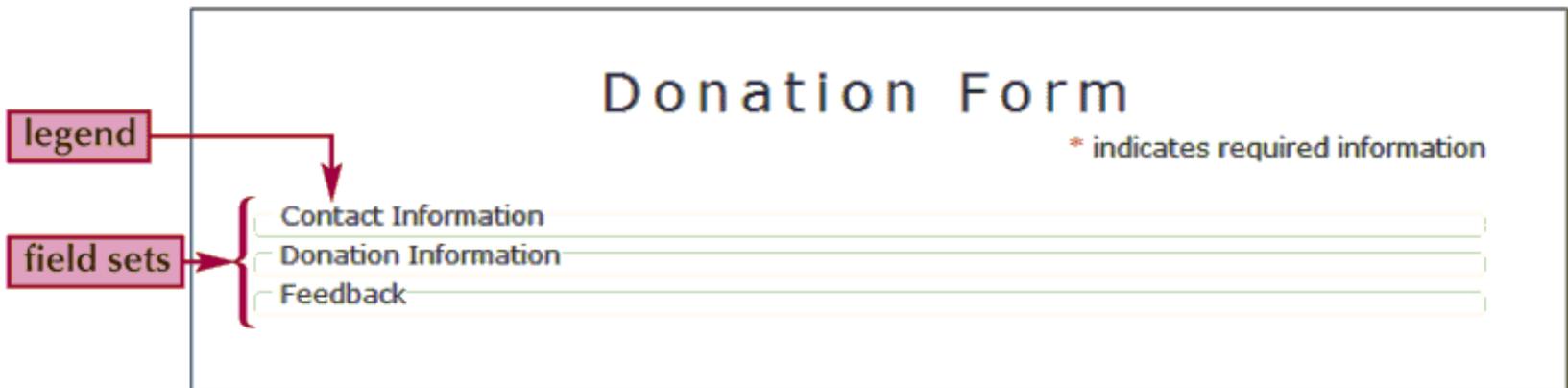
Creating a Field Set

- To add a caption to a field set, add the following tag after the opening `<fieldset>` tag:

```
<legend>  
    text  
</legend>
```

Where *text* is the text of the field set caption.

```
<form name="donationForm" id="donationForm">  
  <fieldset id="contact">  
    <legend>Contact Information</legend>  
  </fieldset>  
  
  <fieldset id="donation">  
    <legend>Donation Information</legend>  
  </fieldset>  
  
  <fieldset id="feedback">  
    <legend>Feedback</legend>  
  </fieldset>  
</form>
```





Creating Input Boxes

- The general syntax of input elements is as follows:
`<input type="type" name="name" id="id" />`
Where **type** specifies the type of input control, and the name and id attributes provide the control's name and id.

```
<form name="donationForm" id="donationForm">
  <fieldset id="contact">
    <legend>Contact Information</legend>
    First Name
    <input type="text" id="firstName" name="firstName" />
    Last Name
    <input type="text" id="lastName" name="lastName" />
  </fieldset>
```

text type indicates an input text box

- Contact Information

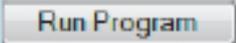
First Name Last Name

- Donation Information

- Feedback



Creating Input Boxes

Type Value	Description	General Appearance
button	Displays a button that can be clicked to perform an action from a script	
checkbox	Displays a check box	<input checked="" type="checkbox"/> <input type="checkbox"/>
file	Displays a Browse button to locate and select a file	donations.htm <input type="button" value="Browse..."/>
hidden	Creates a hidden field, not viewable on the form	
image	Displays an inline image that can be clicked to perform an action from a script	
password	Displays an input box that hides text entered by the user	<input type="password" value="....."/>
radio	Displays an option button	<input checked="" type="radio"/> <input type="radio"/>
reset	Displays a button that resets the form when clicked	<input type="button" value="Cancel Donation"/>
submit	Displays a button that submits the form when clicked	<input type="button" value="Submit Donation"/>
text	Displays an input box that displays text entered by the user	<input type="text" value="Terry Ives"/>



Working with Field Labels

- You can also expressly link a label with an associated text element for scripting purposes.
- The syntax for creating a form label is as follows:

```
<label for="id">label text</label>
```

Where id is the value of the id attribute for a field's control element, and label text is the text of the label.

```
<fieldset id="contact">
  <legend>Contact Information</legend>

  <label>
    First Name
    <input type="text" id="firstName" name="firstName" />
  </label>
  <label>
    Last Name
    <input type="text" id="lastName" name="lastName" />
  </label>
</fieldset>
```



Working with Form Styles

Textbook
Pg 357-8-9

Contact Information

First Name Terry Last Name Ives

Contact Information

First Name

Last Name

```
<label class="blockLabel">  
  First Name  
  <input type="text" id="firstName" name="firstName" />  
</label>  
<label class="blockLabel">  
  Last Name  
  <input type="text" id="lastName" name="lastName" />  
</label>
```

```
label.blockLabel {display: block; position: relative; margin: 12px 0px}
```

```
label.blockLabel input {position: absolute; left: 150px}
```

```

<label class="blockLabel">
  First Name<span>*</span>
  <input type="text" id="firstName" name="firstName" />
</label>

<label class="blockLabel">
  Last Name<span>*</span>
  <input type="text" id="lastName" name="lastName" />
</label>

```

#donationForm span {color: red}

```

<fieldset id="contact">
  <legend>Contact Information</legend>

  <label class="blockLabel">
    First Name<span>*</span>
    <input type="text" id="firstName" name="firstName" />
  </label>

  <label class="blockLabel">
    Last Name<span>*</span>
    <input type="text" id="lastName" name="lastName" />
  </label>

  <label class="blockLabel">
    Phone<span>*</span>
    <input type="text" id="phone" name="phone" />
  </label>

  <label class="blockLabel">
    Street Address<span>*</span>
    <input type="text" id="street" name="street" />
  </label>
</fieldset>

```

The screenshot shows a form titled "Contact Information" with a legend. Below the legend are four input fields, each with a red asterisk indicating a required field:

- First Name*
- Last Name*
- Phone*
- Street Address*



Setting the Width of an Input Box

- To change the width of an input box, use the width attribute, which is displayed as follows:

```
#id {width: value}
```

Where *id* is the id of the control and *value* is the width you want to apply to the input box

```
#firstName, #lastName, #street {width: 25em}  
#phone, #city {width: 10em}  
#state {width: 3em}  
#zip {width: 7em}
```

Contact Information

First Name*	<input type="text"/>
Last Name*	<input type="text"/>
Phone*	<input type="text"/>
Street Address*	<input type="text"/>
City*	<input type="text"/>
State*	<input type="text"/>
ZIP	<input type="text"/>

Setting Other attributes for a Field

- Setting the Maximum Width of an Input Box

```
<label>
  State<span>*</span>
  <input type="text" id="state" name="state" maxlength="2" />
</label>
<label>
  ZIP
  <input type="text" id="zip" name="zip" maxlength="10" />
</label>
```

values of the state field are limited to two characters

values of the zip field are limited to 10 characters

- Setting a Default Value for a Field

```
<label class="indentLabel">
  City<span>*</span>
  <input type="text" id="city" name="city" value="St. Peters" />
</label>
<label>
  State<span>*</span>
  <input type="text" id="state" name="state" maxlength="2" value="MO" />
</label>
```

St. Peters is the default value for the city field

MO is the default value for the state field



Session 6.2

Creating Option Buttons

- **Option buttons**, or **radio buttons** allow users to make selections.
 - Option buttons only allow the user to select one option at a time.

Contact Information

Address For
Home Business

First Name* Gary

Last Name* Browne

Phone* 555-8918

Street Address* 400 Southland Drive

City* St Peters State* MO ZIP 63376



Creating a Group of Option Buttons

- To
fie
<i
<i
<i
to
th
sp
va
• To
<:

```
<fieldset>  
  <legend>Party Affiliation</legend>  
  
  <label for="demoption">Democrat</label>  
  <input type="radio" name="party" id="demoption" value="dem" />  
  
  <label for="gopoption">Republican</label>  
  <input type="radio" name="party" id="gopoption" value="gop" />  
  
  <label for="indoption">Independent</label>  
  <input type="radio" name="party" id="indoption" value="ind" />  
</fieldset>
```

HTML code

Party Affiliation
Democrat Republican Independent

open buttons

ingle

" />
" />
" />

with

d

;

```
<form name="donationForm" id="donationForm">
```

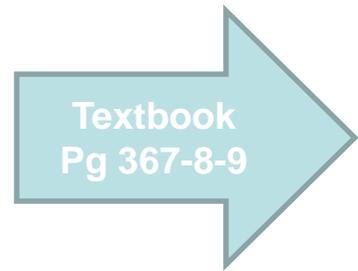
```
<fieldset id="contact">
  <legend>Contact Information</legend>
```

```
<fieldset id="addressoptions">
  <legend>Address For</legend>
```

```
<label for="homeType">Home</label>
<input type="radio" id="homeType" name="addressType" value="home" />
<label for="busType">Business</label>
<input type="radio" id="busType" name="addressType" value="business" />
```

```
</fieldset>
```

```
<label class="blockLabel">
  First Name<span>*</span>
  <input type="text" id="firstName" name="firstName" />
</label>
```



```
#firstName, #lastName, #street {width: 25em}
#phone, #city {width: 10em}
#state {width: 3em}
#zip {width: 7em}

#addressoptions {width: 180px; margin-left: 150px}
```



Creating a Selection List

- A **selection list** is a list box from which a user selects a particular field value or set of field values.
 - Selection lists are useful when there are a fixed set of possible responses from the user.

```
<select id="party" name="party">  
  <option value="dem">Democrat</option>  
  <option value="gop">Republican</option>  
  <option value="ind">Independent</option>  
</select>
```

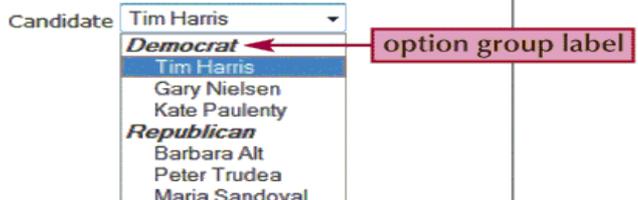
HTML code



rendered selection list

```
<label for="candidate">Candidate</label>  
<select id="candidate" name="candidate">  
  <optgroup label="Democrat">  
    <option>Tim Harris</option>  
    <option>Gary Nielsen</option>  
    <option>Kate Paulenty</option>  
  </optgroup>  
  <optgroup label="Republican">  
    <option>Barbara Alt</option>  
    <option>Peter Trudeau</option>  
    <option>Maria Sandoval</option>  
  </optgroup>  
</select>
```

HTML code



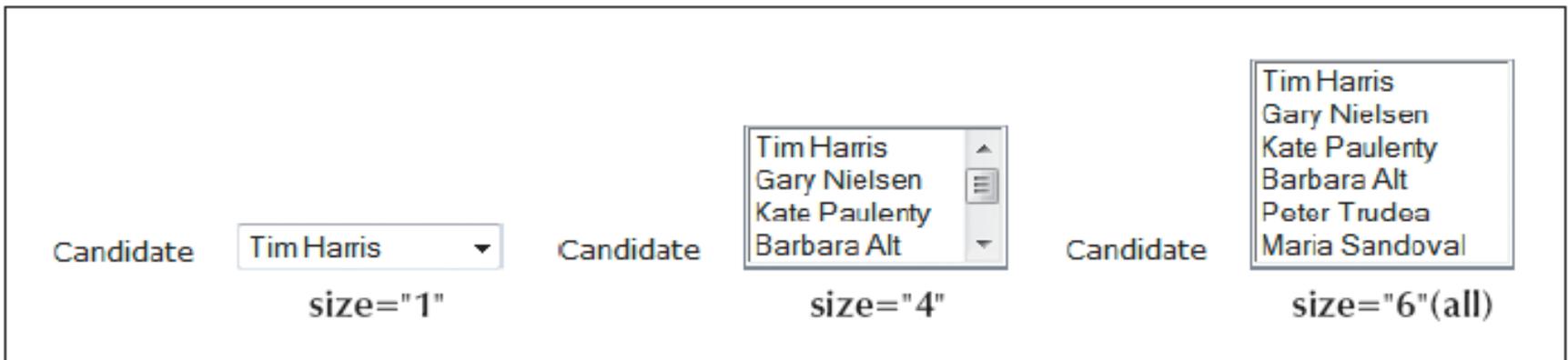
selection list option group

Setting the Selection List Size

- You can change the number of options displayed in the selection list by modifying the size attribute. The syntax is as follows:

```
<select size= "value">... </select>
```

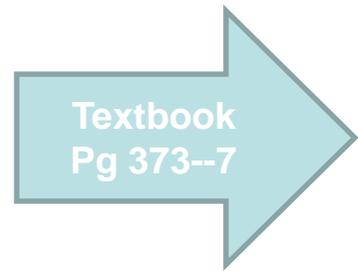
Where *value* is the number of items that the selection list displays in the form.



The image shows three examples of HTML select lists, each labeled "Candidate" and with a corresponding size attribute:

- size="1"**: A single dropdown menu showing "Tim Harris".
- size="4"**: A list box showing four names: "Tim Harris", "Gary Nielsen", "Kate Paulenty", and "Barbara Alt".
- size="6"(all)"**: A list box showing all six names: "Tim Harris", "Gary Nielsen", "Kate Paulenty", "Barbara Alt", "Peter Trudea", and "Maria Sandoval".

Inserting a selection list



```
<fieldset id="donation">
  <legend>Donation Information</legend>

  <label class="blockLabel">
    Donation Amount<span>*</span>
    <input type="text" id="amount" name="amount" />
  </label>

  <label class="blockLabel">
    Credit Card<span>*</span>
    <select id="creditCard" name="creditCard">
      <option value="Amex">American Express</option>
      <option value="Disc">Discover</option>
      <option value="MC">MasterCard</option>
      <option value="visa">Visa</option>
    </select>
  </label>
</fieldset>
```

selection list options

```
#firstName, #lastName, #street {width: 25em}
#phone, #city {width: 10em}
#state {width: 3em}
#zip {width: 7em}

#addressoptions {width: 180px; margin-left: 150px}
#creditCard {position: absolute; left: 150px}
```

```
<label class="blockLabel">
  Credit Card<span>*</span>
  <select id="creditCard" name="creditCard">
    <option value="Amex">American Express</option>
    <option value="Disc">Discover</option>
    <option value="MC">MasterCard</option>
    <option value="Visa">Visa</option>
  </select>
</label>

<label class="blockLabel">
  Cardholder Name<span>*</span>
  <input type="text" id="cardHolder" name="cardHolder" />
</label>

<label class="blockLabel">
  Card Number<span>*</span>
  <input type="text" id="cardNumber" name="cardNumber" />
</label>

</fieldset>
```

```

<label class="blockLabel">
  Card Number<span>*</span>
  <input type="text" id="cardNumber" name="cardNumber" />
</label>

```

```

<label class="blockLabel">
  Expiration Date<span>*</span>
  <select id="expMonth" name="expMonth">
    <option value="01">January (01)</option>
    <option value="02">February (02)</option>
    <option value="03">March (03)</option>
    <option value="04">April (04)</option>
    <option value="05">May (05)</option>
    <option value="06">June (06)</option>
    <option value="07">July (07)</option>
    <option value="08">August (08)</option>
    <option value="09">September (09)</option>
    <option value="10">October (10)</option>
    <option value="11">November (11)</option>
    <option value="12">December (12)</option>
  </select>
  <select id="expYear" name="expYear">
    <option value="2011">2011</option>
    <option value="2012">2012</option>
    <option value="2013">2013</option>
    <option value="2014">2014</option>
    <option value="2015">2015</option>
  </select>
</label>

```

Donation Information

Donation Amount*

Credit Card* American Express ▾

Cardholder Name*

Card Number*

Expiration Date* January (01) ▾ 2011 ▾

```

</fieldset>

```

```

#creditCard          {position: absolute; left: 150px}
#cardholder, #cardNumber {width: 25em}
#expMonth            {position: absolute; left: 150px}
#expYear             {position: absolute; left: 280px}

```



Making Multiple Selections

- Add the multiple attribute to the select element to create multiple selections:

```
<select multiple="multiple">... </select>
```



Working with Check Boxes

- To create a check box, use:

```
<input type="checkbox" name="name" id="id" value="value" />
```

Feedback

I'm interested in volunteering at The Lighthouse.

```
<fieldset id="feedback">
  <legend>Feedback</legend>

  <label>
    <input type="checkbox" id="volunteer" name="volunteer" />
    I'm interested in volunteering at The Lighthouse.
  </label>
</fieldset>
```

- To specify that a check box be selected by default, use the checked attribute as follows:

```
<input type="checkbox" checked="checked" />
```



Working with Text Area Control

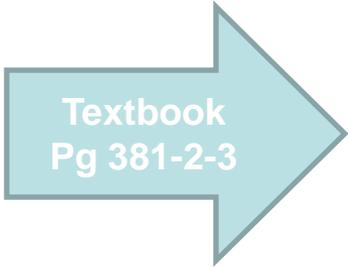
- Text area boxes allow users to enter comments.
- An input box would be too small to accommodate the length of text for this use.
- To create a text area box, use the `textarea` element:

```
<textarea name="name" id="id"  
rows="value" cols="value" wrap="type"> ...  
</textarea>
```

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	Text wraps automatically to the next line when it extends beyond the width of the input 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 area box.
hard	Text wraps automatically to the next line when it extends beyond the width of the input 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 area box.



Working with Text Area Control



```
<fieldset id="feedback">
  <legend>Feedback</legend>
  <label>
    <input type="checkbox" id="volunteer" name="volunteer" />
    I'm interested in volunteering at The Lighthouse.
  </label>
  <label for="comments" class="blockLabel">Comments</label>
  <textarea id="comments" name="comments" rows="5" cols="50"></textarea>
</fieldset>
```

the text area box will have five lines of 50 characters each

Feedback

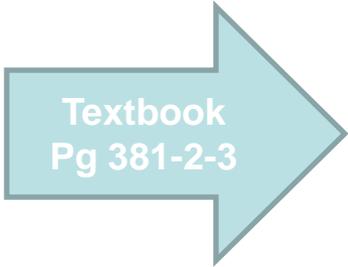
I'm interested in volunteering at The Lighthouse.

Comments

text box area



Background for fieldset and space



Donation Form

* indicates required information

Contact Information

Address For
Home Business

First Name*

Last Name*

Phone*

Street Address*

City* St Peters State* MO ZIP

Donation Information

Donation Amount*

Credit Card* American Express

Cardholder Name*

Card Number*

Expiration Date* January (01) 2011

Feedback

I'm interested in volunteering at The Lighthouse.

Comments



Specifying the Tab Order

- Users typically navigate through a form with the tab key.
- You can specify an alternate tab order by adding the `tabindex` attribute to any control element in your form.
- The syntax is as follows:

```
<input name="fname" id="firstName"  
      tabindex="1" />
```

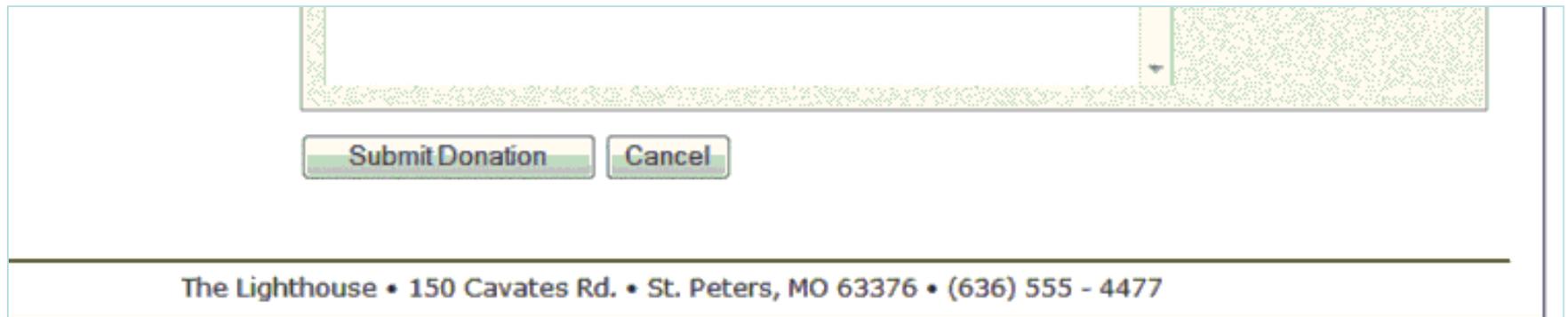
This syntax assigns the tab index number “1” to the `fname` field from the registration form.



Session 6.3

Working with Form Buttons

- Buttons are a type of control element that performs an action.
- Types of buttons:
 - Command button
 - Submit button
 - Reset button





Creating a Command button

- Command buttons are created using the `<input>` tag:

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

(No action by itself. We need a script for example JavaScript as we see later)
- Submit buttons submit forms to the server for processing when clicked. Syntax is as follows:

```
<input type="submit" value="text" />
```
- Reset buttons reset forms to their original (default) values. Syntax is as follows:

```
<input type="reset" value="text" />
```

```
</fieldset>
```

```
<input type="submit" value="Submit Donation" />
<input type="reset" value="Cancel" />
```

```
</form>
```

```
</div>
```

```
<address>
```

```
The Lighthouse &bull;
150 Cavates Rd. &bull;
St. Peters, MO 63376 &bull;
(636) 555 - 4477
```

```
</address>
```

The success of The Lighthouse reflects the dedication and support of members of the community who have helped make our dream a reality. We cannot continue to operate without contributions from people like you.

You can make a tax-deductible donation online using your American Express, Discover, Master, or Visa card. Please fill out the form on this page.

The Lighthouse is always looking for volunteers. We especially need help in the following areas:

- Mechanics
- Carpenters
- Electricians
- Cooks
- Computer technicians
- Babysitters
- Data entry persons

and many others. Please consider donating your time and talents to your community and your neighbors.

Thank you so much for your generosity!

— Terry Ives
Director, The Lighthouse

Donation Form
* indicates required information

Contact Information

Address For
Home Business

First Name*

Last Name*

Phone*

Street Address*

City* St Peters State* MO ZIP

Donation Information

Donation Amount*

Credit Card* American Express

Cardholder Name*

Card Number*

Expiration Date* January (01) 2011

Feedback

I'm interested in volunteering at The Lighthouse.

Comments

Submit Donation Cancel

The Lighthouse • 150 Cavates Rd. • St. Peters, MO 63376 • (636) 555 - 4477



Designing a Custom Button

- Use the button element for greater artistic control over the appearance of a button.

```
<button name="name" id="id" value="value"  
  type="type">  content  </button>
```

```
<button name="home" id="home" type="button">  
    
  <span style="color: blue; font-weight: bold; font-style: italic">  
    Return to the Home Page  
  </span>  
</button>
```

HTML code



custom button



Creating File Buttons

- **File buttons** are used to select files so that their contents can be submitted for processing to a program.

`<input type="file" id="filename" name="filename" />`

HTML code

1. Click the Browse button

2. Select a file from the Choose file window

3. The file name and location are placed in the input box

The diagram illustrates the process of creating a file button in three steps:

1. Click the Browse button: An empty text input field is shown next to a blue "Browse..." button.
2. Select a file from the Choose file window: A "Choose file" dialog box is shown. The "Name" column lists "donations.pdf", which is selected. The "File name" field at the bottom contains "donations.pdf".
3. The file name and location are placed in the input box: The text input field now contains the full path "C:\donations.pdf" and the "Browse..." button remains.



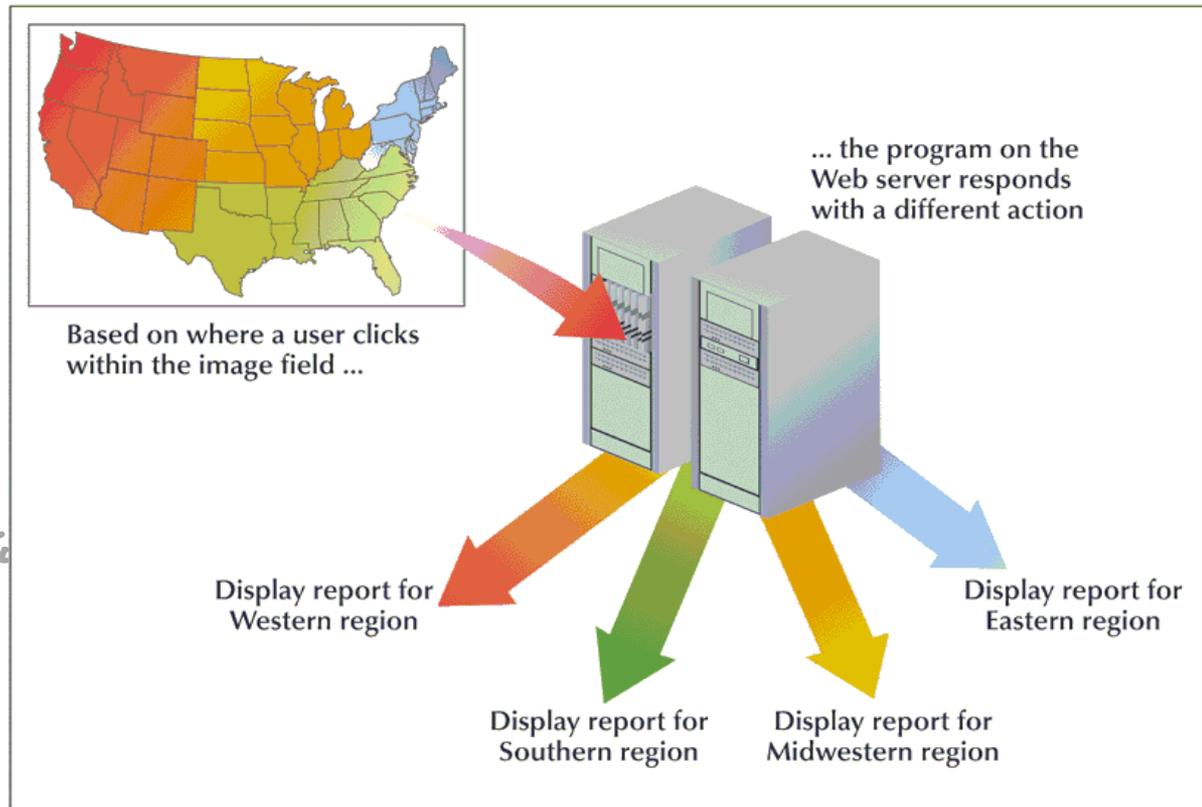
Creating Image Field Buttons

Using an image control field with a server-based program

```
<input  
type="image"  
src="url"  
name="text"  
id="id" />
```

```
name.x=coordinate&  
name.y=coordinate
```

is sent to the server





Working with Hidden Fields

- **Hidden fields** are added to a form, but not displayed in the Web page. The syntax is as follows:

```
<input type="hidden" name="name" id="id" value="value" />
```

```
<form name="donationForm" id="donationForm">
  <input type="hidden" name="email" id="email" value="donations@thelighthouse.org" />
  <fieldset id="contact">
    <legend>Contact Information</legend>
```



Working with Form Attributes to send data to the server

`<form action=http://www.rebol.com/cgi-bin/test-cgi.cgi>`

- After adding the elements to your form, you'll need to specify where to send the form data and how to send it. Use the following attributes:

```
<form action="url" method="type"  
enctype="type">    ...    </form>
```

Where *url* specifies the filename and location of the program that processes the form and the *method* attribute specifies how your Web browser sends data to the server. The *enctype* attribute specifies the format of the data stored in the form's field.

Values of the enctype attribute

Value	Description
application/x-www-form-urlencoded	The default format. In this format, form data is transferred as a long text string in which spaces are replaced with the + character and nontext characters (such as tabs and line breaks) are replaced with their hexadecimal code values. Field names are separated from their field values with a = symbol.
multipart/form-data	Used when sending files to a server. In this format, spaces and nontext characters are preserved, and data elements are separated using delimiter lines. The action type of the form element must be set to post for this format.
text/plain	Form data is transferred as plain text with no encoding of spaces or nontext characters. This format is most often used when the action type of the form element is set to mailto.



Working with Form Attributes

Textbook
Pg 392

- The method attribute can have one of two values:
 - Post
 - Get
- The get method is the default; get appends the form data to the end of the URL specified in the action attribute.
- The post method sends form data in a separate data stream, allowing the Web server to receive the data through “standard input.” *safer and can deal with long data*

```
<form name="donationForm" id="donationForm"  
  action=http://www.rebol.com/cgi-bin/test-cgi.cgi  
  method="post">
```



Using the mailto Action

- The mailto action accesses the user's own e-mail program and uses it to mail form information to a specified e-mail address.
 - Bypasses the need for server-based programs.
- The syntax is as follows:

```
<form action="mailto:e-mail" method="post"
enctype="text/plain"> ... </form>
```
- Where *e-mail_address* is the e-mail address of the recipient in the form.



Tables for Form layout

```
<form action="http://www.rebol.com/cgi-bin/test-cgi.cgi" method="get">
```

```
  Name: <input type="text" name="Field1" size="25" /> <br />
```

```
  Lastname: <input type="text" name="Field2" size="25" /> <br />
```

```
  Address: <input type="text" name="Field3" size="25" /> <br />
```

```
  e-mail: <input type="text" name="Field3" size="25" /> <br />
```

```
  comments: <input type="text" name="field5" /> <br />
```

```
  <input type="submit" name="Submit" value="Submit" />
```

```
</form>
```

Name:

Lastname:

Address:

e-mail:

comments:

Name:

Lastname:

Address:

e-mail:

comments:



Tables for Form layout

```
<form action="http://www.rebol.com/cgi-bin/test-cgi.cgi" method="post">
<table>
<tr> <th>Name: </th>
      <td> <input type="text" name="Field1" size="25" /> </td> </tr>
<tr> <th>Lastname: </th>
      <td> <input type="text" name="Field2" size="25" /> </td> </tr>
<tr> <th>Address: </th>
      <td> <input type="text" name="Field3" size="25" /> </td> </tr>
<tr> <th>e-mail: </th>
      <td> <input type="text" name="Field4" size="25" /> </td> </tr>
<tr> <th>comments: </th>
      <td> <input type="text" name="field5" /> </td> </tr>
<tr> <th> </th>
      <td> <input type="submit" name="Submit" value="Submit" /></td> </tr>
</table>
</form>
```



Tips for Creating Effective Forms

- Mark fields that are required, but also limit the number of unrequired fields. Don't overwhelm your users with requests for information that is not really essential. Keep your forms short and to the point.
- If you need to collect a lot of information, break the form into manageable sections spread out over several pages. Allow users to easily move backward and forward through the forms without losing data.



Tips for Creating Effective Forms

- Provide detailed instructions about what users are expected to do. Don't assume that your form is self-explanatory.
- If you ask for personal data and financial information, provide clear assurances that the data will be secure. If possible, provide a link to a Web page describing your security practices.
- Clearly indicate what users will receive once the form is submitted, and provide feedback on the Web site and through e-mail that tells them when their data has been successfully submitted.