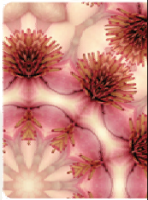


HTML and XHTML  
2nd Edition

## Tutorial 6

### Working with Web Forms



New Perspectives on HTML and XHTML, Comprehensive

## Objectives

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

New Perspectives on HTML and XHTML, Comprehensive

## 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

New Perspectives on HTML and XHTML, Comprehensive

## 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
  - Drop-down lists boxes
  - Check boxes
  - Text areas

New Perspectives on HTML and XHTML, Comprehensive

## 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.

New Perspectives on HTML and XHTML, Comprehensive

## 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

New Perspectives on HTML and XHTML, Comprehensive

## Creating a Web Form

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

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

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

New Perspectives on HTML and XHTML, Comprehensive

7

## 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.
- Always specify an id or name for the form.
- Two attributes are available to identify the form: id and name.

New Perspectives on HTML and XHTML, Comprehensive

8

## Creating a Web 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.

New Perspectives on HTML and XHTML, Comprehensive

9

## 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

New Perspectives on HTML and XHTML, Comprehensive

10

## 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.

New Perspectives on HTML and XHTML, Comprehensive

11

## 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.

New Perspectives on HTML and XHTML, Comprehensive

12

## Creating Input Boxes

Type Value	Description	General Appearance
button	Displays a button that can be clicked to perform an action from a script	<input type="button" value="Run Program"/>
checkbox	Displays a check box	<input type="checkbox"/>
file	Displays a Browse button to locate and select a file	<input type="file" value="donation.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	<input alt="User icon" type="image"/>
password	Displays an input box that hides text entered by the user	<input type="password" value="*****"/>
radio	Displays an option button	<input type="radio"/>
reset	Displays a button that resets the form when clicked	<input type="reset" value="Cancel Donation"/>
submit	Displays a button that submits the form when clicked	<input type="submit" value="Submit Donation"/>
text	Displays an input box that displays text entered by the user	<input type="text" value="Terry Ives"/>

New Perspectives on HTML and XHTML, Comprehensive

13

## 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.

New Perspectives on HTML and XHTML, Comprehensive

14

## 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

New Perspectives on HTML and XHTML, Comprehensive

15

## Setting the Width of an Input Box

```
#donationForm span {color: red}
label.indentLabel {margin-left: 150px}
#firstName, #lastName, #street {width: 25em}
#phone, #city {width: 10em}
#state {width: 3em}
#zip {width: 7em}
```

New Perspectives on HTML and XHTML, Comprehensive

16

## Creating Option Buttons

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

```
<fieldset>
  <legend>Party Affiliations/legends
  <label For="democrat">Democrat</label>
  <input type="radio" name="party" id="democrat" value="dem" />
  <label For="republican">Republican</label>
  <input type="radio" name="party" id="republican" value="rep" />
  <label For="independent">Independent</label>
  <input type="radio" name="party" id="independent" value="ind" />
</fieldset>
```

HTML code

Party Affiliations

Democrat  Republican  Independent

option buttons

New Perspectives on HTML and XHTML, Comprehensive

17

## Creating a Group of Option Buttons

- To create a group of option buttons associated with a single field, add the elements:
 

```
<input type="radio" name="name" id="id1" value="value1" />
<input type="radio" name="name" id="id2" value="value2" />
<input type="radio" name="name" id="id3" value="value3" />
```

 to the Web form, where *name* identifies the field associated with the collection of option buttons; *id1*, *id2*, *id3*, etc. identify the specific options; and *value1*, *value2*, *value3*, etc. are the field values associated with each option.
- To specify the default option, add the following attribute to the `<input>` tag:
 

```
checked="checked"
```

New Perspectives on HTML and XHTML, Comprehensive

18

## 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.
- You can create a selection list using the `<select>` element.
- You can specify each individual selection item using the `<option>` element.

New Perspectives on HTML and XHTML, Comprehensive

19

## 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.

New Perspectives on HTML and XHTML, Comprehensive

20

## Setting the Selection List Size

The image shows three examples of selection lists. The first is a standard dropdown menu with the text 'Candidate' and 'Tim Harris' and the attribute 'size="1"'. The second is a list box showing four names: 'Tim Harris', 'Gary Nielsen', 'Kate Paulerty', and 'Barbara Alt' with the attribute 'size="4"'. The third is a list box showing all six names: 'Tim Harris', 'Gary Nielsen', 'Kate Paulerty', 'Barbara Alt', 'Peter Tuedoa', and 'Maria Sandoval' with the attribute 'size="6"(all)'.

New Perspectives on HTML and XHTML, Comprehensive

21

## Making Multiple Selections

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

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

New Perspectives on HTML and XHTML, Comprehensive

22

## Working with Check Boxes

- To create a check box, use:  
`<input type="checkbox" name="name" id="id" value="value" />`
- Where the name and id attributes identify the check box controls and the value attribute specifies the value sent to the server if the check box is selected.
- To specify that a check box be selected by default, use the checked attribute as follows:  
`<input type="checkbox" checked="checked" />`

New Perspectives on HTML and XHTML, Comprehensive

23

## 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.

New Perspectives on HTML and XHTML, Comprehensive

24

## 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.

## Working with Text Area Control

- To create a text area box, use the `textarea` element:

```
<textarea rows="value" cols="value">
... </textarea>
```

Where the `rows` and `cols` attributes define the dimensions of the input box and the `rows` attribute indicates the number of lines in the input box.

## Working with Text Area Control

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>
<table>
<tr>
<td type="checkbox" id="volunteer" name="volunteer" />
<td>I'm interested in volunteering at The Lighthouse.
</td>
</tr>
<tr>
<td colspan="2">
<table border="1" class="lockable">
<tr>
<td colspan="2">
<td colspan="2">
<td colspan="2">
</tr>
</table>
</td>
</tr>
</table>
```

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



## 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" />
```
- 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" />
```

## Completed Form



## 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>
```

Where the *name* and *value* attributes specify the name of the button and the value sent to a server-based program, the *id* attribute specifies the button's id, the *type* attribute specifies the button type, and the *content* is page content displayed within the button.

## Creating File Buttons

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



## 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

- 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.

## Working with Form Attributes

- 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."

### 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.

New Perspectives on HTML and XHTML, Comprehensive

37

### 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.

New Perspectives on HTML and XHTML, Comprehensive

38

### 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.

New Perspectives on HTML and XHTML, Comprehensive

39