

## Lesson 8-Forms

Please read your textbook pages 399-420.

The XHTML Strict DTD supports a core group of form elements that will transition toward X-Forms. X-Forms is supposed to be THE forms format, but it won't be supported for a year or possibly more. In the meantime:

The XHTML Transitional DTD supports the form elements of HTML 4.0, although some of these elements have been deprecated.

XHTML Modularization supports a Forms Module of form elements broader than the XHTML Strict DTD. XHTML Modularization also acts as a parent set for XHTML Basic Forms, a short, simple set of forms elements to be used for wireless, hand-held, and other portable devices supported by XHTML Basic.

X-Forms is the hope of the commercial form future due to its in-depth capabilities, although it is currently in the working draft stages at the W3C and

This lesson defines and demonstrates XHTML 1.0-Strict DTD, XHTML 1.1 Forms Module and define the XHTML Basic 1.0 Forms Module.

Form creation, from an XHTML 1.0 Strict DTD perspective, looks like:

```
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/TR/xhtml1" xml:lang="en"
lang="en">
<head>
<title>Form with XHTML Strict DTD</title>
</head>
<body>
<h4>Please enter your account number:</h4>
<form action="yourEmail@dns.com"
method="post">
<input type="text" name="accountnumber" id="accountnumber"
/>
<input type="submit" />
</form>
</body>
</html>
```

Please key the above code and open it in your browser.

All XHTML forms begin with the form element that defines the form. Most forms elements are named, for the purpose of identification in transmitting data, and all have several possible attributes for layout customization. In this example:

```
<form action="yourEmail@dns.com" method="post">
```

The form element starts the form and connects it to the resource at the indicated URL.

A form collects data, it does not process data.

The form sends the gathered data to a program on a server, or interacts with a server using a script program. The forms action attribute tells the browser where to send the data, in this case it asks the data to be e-mailed.

The form defines how the data will be packaged for delivery to the server, but the program on the server must understand the details in order to unpackage the information and use it. The form and the program to process the form must be designed together. The form element also contains a method attribute. `method` tells the browser what method to employ to send the information to the server.

Data can be sent in several ways, two examples include an action URL, or data sent as a message. As an example:

```
<input type="text" name="accountnumber" id="accountnumber" />
```

In this example, a textbox is created to allow text to be keyed in. The text box is named account number (you must name `<input>` to transmit to a server).

```
<input type="submit" />
```

The second input element creates a button that submits the information in the form back to the server. The submit button is a special control that makes a browser submit a form when it is activated. The submit button has a default caption "Submit Query" that can be modified.

**Exercise 8-1.** Using the code on page 1, create and view a form with a textbox and a submit button. The purpose of the textbox will be to enter your zip code to search for the nearest Home Appliance Center in your neighborhood.

An example, is:

```
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en"
lang="en">
<!-- zipcode.html by Your Name -->
<head>
<title>Zipcode Form example</title>
</head>
<body>
```

```
<h2>Please enter your Zipcode</h2>
<p>
<br />
</p>
<form action="zip" method="post">
<p><input type="text" size="10" value="enter Zip" /></p>
<p>
<br />
</p>
<p><input type="submit" /></p>
</form>
</body>
</html>
```

## Form Controls

XHTML forms are composed of controls that are components supported by elements and their attributes. Within the form element (and occasionally outside of `<form>` for the purpose of a user interface) form controls can be created such as:

1. textboxes, password boxes, and text-area boxes
2. command buttons, radio buttons, and checkboxes, and buttons with images
3. list boxes
4. dropdown boxes

As an example, to create a checkbox, use the control type:

```
<input type="checkbox" name="prefix" />
```

Developing forms is, for the most part, a 3-step process.

1. The first, and most important step, is to plan the information to be gathered, prioritize the information in the design of the form, and edit out any information that isn't important. You have a short user attention span with forms, make the most of getting the data you need.
2. The second step is to determine which elements and their attributes best allow you to request information from a user.
3. The third step is to plan and code the forms interface and the data transfer to the server and the end client.

The elements for your forms, are defined below:

```
<fieldset>..</fieldset>
```

`fieldset` groups together a set of related form input mechanisms, enabling them to be formatted and labeled as a group.

`fieldset` content includes input elements and labels, the markup that organizes those elements, and a legend element to provide a label for `fieldset`.

The legend should appear just after the `fieldset` start tag.

Supported `fieldset` attributes include: `id`, `class`, `style`, `title`, `lang`, `xml:lang`, and `dir`

`dir` specifies the direction in which the controls text is displayed, left-to-right or right-to-left.

If you want to divide your form into sections, the `fieldset` element, followed directly by the legend element, followed by the elements and attributes for the form section allows for sectioning and labeling a form. As an example:

```
<form> . . </form>
```

`form` holds the content of the fill-in form, as defined by the input elements that can appear inside of a form. Forms do not nest, there can be no form within a form.

Supported forms attributes include: `id`, `class`, `style`, `title`, `lang`, `xml:lang`, `dir`, `action` (required), `method`, `enctype`, `onsubmit`, `onreset`, `accept`, `accept-charset`.

For this class you do not need to memorize the attributes below, however, I would like to define them in order to introduce them to you.

1. `action`, a required attribute, specifies a URL, server-side program or HTTP script that processes the filled-in information.
2. `method` specifies the HTTP method used to send the form to the server (W3C recommends POST).
3. `enctype` specifies the MIME-type encoding for data sent via the POST method.
4. `onsubmit` names a script to run when the user clicks the form's Submit button.
5. `onreset` names a script to run when the user sets the form's Reset button.
6. `accept` is a list of one or more MIME-types that this part of the form and the form-processing server will accept.

7. `accept-charset` lists character sets supported by the server processing the submitted forms.

`<input>..</input>`

`input` specifies editable fields and is only allowed inside a form.

`input` takes a `type` attribute to define the type of input mechanism desired, a `name` attribute, to define the variable name associated with the input data, and other attributes to set size and alignment. Supported input attributes include: `id`, `class`, `style`, `title`, `lang`, `xml:lang`, `dir`, `type`, `name`, `value`, `checked`, `disabled`, `readonly`, `size`, `maxlength`, `src`, `alt`, `usemap`, `tabindex`, `accesskey`, `onfocus`, `onblur`, `onselect`, `onchange`

1. `name` labels the input control (for the most part all input elements are named for easier identification by scripting languages moving data)
2. `value` sets the input control initial value
3. `checked` sets the initial value to on
4. `disabled`, in the future, will disable input
5. `readonly` disallows user changes to input
6. `size` sets maximum size of textbox
7. `maxlength` sets maximum number of characters in a textbox
8. `src` specifies image for button
9. `alt` text only browser image description
10. `usemap` specifies client-side image map displayed in a form
11. `tabindex` defines position of input control for user to "tab" through form
12. `accesskey` assigns shortcut to element for focus
13. `onfocus` names script to run when user moves mouse to point to current element
14. `onblur` names script to run when user moves mouse away from current element

15. `onselect` names script to run when user moves mouse to select text

16. `onchange` names script to run when user moves mouse to element, off element, to another element

Form controls you will use for these exercises include:

`<input type="radio" plus additional attributes for a radio button`

`<input type="text" plus additional attributes for a textbox`

`<input type="checkbox" plus additional attributes for a checkbox`

`<label>..</label>`

`label` defines a label for a specified input element. The `for` attribute of the label defines the relationship between a label and an input. The `for` attribute takes as its value the `id` value of the input element. Supported label attributes include: `id`, `class`, `style`, `title`, `lang`, `xml:lang`, `dir`, `for`, `accesskey`, `onfocus`, `onblur`

An example of a simple form incorporating the four elements above is

```
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
SYSTEM "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
strict.dtd">
<html xmlns="http://www.w3.org/TR/xhtml1 xml:lang="en"
lang="en">
<head>
<title>Fieldset and Legend</title>
</head>
<body>
<form action="yourEmail@dns.com"
method="post">
<fieldset>
<legend>Wholesale Purchase Orders</legend><br />
<label>Dealers: Please key your order number to
obtain current order status
<input type="text" name="w_order_status"
id=" w_order_status " size="30">
</label>
<input type="submit" />
</fieldset>
<br /><br />
<fieldset>
<legend>Retail Purchase Orders</legend><br />
<label>Retail Customers: Please key your account
name to obtain current order status
<input type="text" name="r_order_status "
```

```
id=" r_order_status " size="30">
</label>
<input type="submit" />
</fieldset>
</form>
</body>
</html>
```

**8-2)** Please key the above form

Link 8-1 and 8-2 from your index page, they are the two forms you will hand code.

Next, create the following forms using Dreamweaver. Link them from your index page as well.

**8-3)** The manufacturing division of a start-up recently expanded into a hard-to-find materials facility. Create a fill-in form for their site that will later link to an online map service. For now, the client needs only the format of the form to see if the information will be complete. The form has two parts.

Part 1 includes:

Enter your starting location

Street Address or Intersection

City, State Zip

Country

(the first three will use textbox, country should be a pulldown menu)

Part 2 includes:

Choose one of our destination locations

(a pulldown menu will allow the choices, Corporate Headquarters or Manufacturing facility)

**8-4)** A reseller site needs a form with the following information:

Purch date

Item name

Buyer

Seller

Seller's comments

Condition

Item price

Shipping and handling

Total

Pay by

Shipping address

Received date

This form will be accessed by buyers and sellers, As such, it needs to be simple, clear, and easy to read and maintain.

**8-5)** An Internet Conference needs an online registration form-please incorporate the following information into the form.

Mr. Ms. Dr.

First Name, Last Name, Title, Company, Street, Division, City

State/Province, Zip/Postal Code, Country, Phone, Fax, Email

Select Conference. Package 1 day @525, 2 day @895, 3 day @ 1195, 4 day @1395

Payment Method. Check or Money Order, MasterCard, Visa, American Express

What is your organization's primary business activity at your location?

Web/Online Business(E-Business Sites/Portals/Content/Other)

Web/Software Development

Web Hosting Service/Data Center

Computer Communication/Network Equipment Manufacturer

Internet Service Provider

What is your primary job function?

Corporate Management

IT/Internet Management

Networking/Communications Management

Application Development Management

Marketing/Sales/Product Management

Operations/Financial Management

Web Site Technical Management

Web Content/Design Management

**8-6)** Next, please create a form and incorporate it into your midterm project. Link to the page in your midterm where the form is located.