Unit 14 XHTML – 2

Structure:

14.1 Introduction

In previous unit, you have learnt benefits of XHTML and its tags. By now you have understood the names of elements in XHTML tend to refer to the type of markup they contains. In this unit you are going to learn basic table tags, which will help you to create rows and columns of table, and some features of table like cell padding, cell spacing, captions, and headings. You will also learn how to interlink the pages of your website, link to specific parts of a page, link to other sites. This unit also covers the forms and the role of form and how they perform in JavaScript validations.

Objectives:

After reading this unit, you should be able to:

- create a table
- create email address links
- describe Form
- explain form method and action attribute
- define different types of form controls
- validate a form using JavaScript

14.1.1 Tables

Tables are made up of rows of cells. The number of cells in each row determines the table's shape. To define a table, use the table element <table> and the corresponding </table> tags. To define rows within the table, use the tr (table row) element. To define cells within a row, use the td (table data) element.

The following is an example of a very basic table:

```
<table
<tr>
<td>First Row , First Column </td>
<td>First Row , Second Column </td>
</tr>
<tr>
<td>Second Row , First Column </td>
<td>Second Row , Second Column </td>
</tr>
</table>
```

The <table> element can carry the following attributes: Align, bgcolor, border, cell padding cell spacing, frame rules, summary, width, height, and dir.

- **The <td> and <th> elements**

In every table, cells are represented by using <td> and <th> elements.

<td> : this tag is used to represent table data. By default data is aligned to left.

<th> : this tag is used to represent headings. By default heading displayedin bold**.**

The <td> and <th> elements can both carry the same set of attributes, each of which applies just to that cell. In addition to the universal attributes and the basic event attributes, the <td> and <th> elements can also carry the following attributes: **width, height, rowspan, scope, valign, Abbr, align, axis, bgcolor, char, charoff, colspan, headers**.

- **The cellpadding Attribute**

The cellpadding attribute is used to create a gap between the edges of a cell and its contents. The value for this attribute can either be the amount of space or padding you want inside each wall of the cell in pixels or apercentage value (as a percentage of the width of the table).

Example: cellpadding= “5” or cellpadding= “2%”

- **The cellspacing Attribute**

The cellspacing attribute is used to create a space between the borders of each cell. The value for this attribute can be either the amount of space you want to create between the cells in pixels or a percentage value (as a percentage of the width of the table).

Example: cellspacing= “6” or cellspacing= “2%”

- **The dir Attribute**

The dir attribute is supposed to indicate the direction of text that is used in the table. Possible values are ltr for left to right text and rtl for right to left.

Example: dir= “rtl”

If you use the dir attribute with a value of rtl on the <table> element, then the cells appear from the right first and each consecutive cell is placed to the left of that one.

- **The rowspan Attribute**

The rowspan attribute specifies the number of rows of the table a cell will span across, the value of the attribute being the number of rows the cell stretches across.

Example: rowspan= “2”

- **The summary Attribute**

The summary attribute is supposed to provide a summary of the table’s purpose and structure for nonvisual browsers such as speech browsers or Braille browsers. The value of this attribute is not rendered in IE or Firefox, but you should include it in your pages for accessibility purposes.

Example:

<table border = “1” cell padding = “1” cellspacing = “2” summary = “This table specifies available fruit juices in canteen “>

<caption> Fruit juices </caption>

```
<tr>
<th> name </th>
<th>availability </th>
</tr>
<td> apple </td>
<td> yes </td>
<td> mango <td>
<td> no </td>
<td> orange </td>
<td> yes </td>
</tr>
 </table>
```

14.1.2 XHTML LINKS

Links are most important part of web, because they are the means by which a user can navigate from page to page, or from site to site.

Hyperlinks allow visitors to navigate between web sites by clicking on words, phrases, and images. The web site is a group of pages users navigate between using hypertext links. These pages often include links to other web sites as well as to other pages in the same site.

A link is specified using the **<a>** element. Anything between the opening <a> tag and the closing </a> tag becomes part of the link a user can click in a browser.

14.1.3 Linking to Other Documents

To link to another document, the opening <a> tag must carry an attribute called href; the value of the href attribute is the page you are linking to.

As an example, here is the <body> of a document called main.html. This page contains a link to a second page called index.html:

Example:

```
<body>
Return to the <a href="index.html">index page</a>
</body>
```

As long as index.html is in the same folder as main.html, when you click the words “index page,” the index.html page will be loaded into the same window, replacing the current main.html page.

If you want to link to a different site, you can use the following syntax, where you specify a full URL (*Uniform Resource Locator*) for the page you want to link to rather than just the filename main.html

Example:

<body>

<a href=“http://www.wrox.com/”> next page Web site</a>?

</body>

This link points to the Wrox web site. As you can see, the value of the href attribute is the same as you would type into a browser if you wanted to visit the Wrox web site. This is known as a qualified URL because it contains the domain name for the web site.

When you are linking to pages that make up part of the same web site, you can use a shorthand form called *relative URLs.*

14.1.4 Linking to E-mail Addresses

You’ve probably seen links on many sites that show an e-mail address, and you have probably noticed that clicking one of these links will open a new e-mail in your default e-mail program, ready for you to send an e-mail to that address.

To create a link to an e-mail address, you need to use the following syntax with the <a> element:

<a href=“mailto:name@example.com”>name@example.com</a>

Here is the value of the href attribute starts with the keyword mailto, followed by a colon, and then the e-mail address you want the mail sent to. As with any other link, the content of the <a> element is the visible part of the link shown in the browser, so you might choose to use the following:

<a href=“mailto:name@example.com”>E-mail us</a>.

Or, if you want users to see the e-mail address before clicking it, you can use the following:

For sales enquiries e-mail

<a href=”mailto:name@example.com”>sales@example .com</a>.

There is one drawback to using this technique, however: Some less scrupulous inhabitants of the Web use little programs to automatically search web sites for e-mail addresses. After they have found e-mail

addresses on web sites, they will start sending spam (junk mail) to those addresses.

There are a few main alternatives to creating a link to an e-mail address:

Use an e-mail form instead so that visitors fill in a form on your web site to send you an e-mail.

Once you have received the mail, you can then reply as normal because automated programs do not use contact forms to collect e-mail addresses. Use of an e-mail form requires either a CGI script or a server-side scripting language such as ASP.net, JSP, PHP, Cold Fusion, or Ruby.

14.1.5 Linking to a specific part of a page

When you are developing long web page, the web page content is not fitting in single browser window and the user using scrollbar to read from top to bottom. Instead of this the convenient way is splitting page into specific parts and making the proper links between them.

The *destination anchor* allows the page author to mark specific points in a page that a source link can point to.

Common examples of linking to a specific part of a page that you might have seen used on web pages include:

- “Back to top” links at the bottom of long pages
- List of contents for a page that takes the user to the relevant section
- Links to footnotes or definitions

You create a destination anchor using the <a> element again, but when it acts as a destination anchor it must carry an id attribute because the id attribute was only introduced in XHTML.

The whole page does not fit on the screen at once, forcing the user to scroll, so you want to add links to each of the main headings at the start of the document.

Before you can create links to each section of the page (using the source anchors), you have to add the destination anchors. Here you can see the subheadings of the page; each containing an <a> element with the idattribute whose value uniquely identifies that section:

Example:

```
<h1>Linking and Navigation</h1>
<h2><a id= “URL”>URLs</a></h2>
<h2><a id= “SourceAnchors”>Source Anchors</a></h2>
<h2><a id= “DestinationAnchors”>Destination Anchors</a></h2>
<h2><a id= “Examples”>Examples</a></h2>
```

With destination anchors in place, it’s now possible to add source anchors to link to these sections, like so:

Example:

```
<p>This page covers the following topics:
<ul>
<li><a href= “#URL”>URLs</a></li>
<li><a href= “#SourceAnchors”>Source Anchors</a></li>
<li><a href= “#DestinationAnchors”>Destination Anchors</a></li>
<li><a href= “#Examples”>Examples</a></li>
</ul>
</p>
```

The value of the href attribute in the source anchors is the value of the id attribute preceded by a pound or hash sign (#).

Self Assessment Questions

1. The_________attribute is supposed to indicate the direction of text that is used in the table.
2. Which of the following attribute is used to create a space between the borders of each cell?
   a) Cellspacing attribute b) cell padding c) rowspan d) dir
3. The___________allows the page author to mark specific points in a page that a source link can point to.
4. Anything between the opening <a> tag and the closing </a> tag

becomes part of the link a user can click in a browser. (True/False)

14.2 Forms

Forms are created by using <form> tag. There are number of form controls available like text box, button, checkbox, submit button, etc. If you are using form controls you need to represent them between opening <form> and closing </form>.

Once users have entered information into a form, they usually have to click what is known as a *submit button* (although the actual text on the button may say something different such as Search, Send, or Proceed and often pressing the return key on the keyboard has the same effect as clicking this button).This indicates that the user has filled out the form, and this usually sends the form data to a web server.

Once the data that you have entered arrives at the server, a script or other program usually processes the data and sends a new web page back to you. The returned page will usually respond to a request you have made or acknowledge an action you have taken.

Providing you keep your <form> elements separate from each other (and no one <form> element contains another <form> element), your page may contain as many forms as you like. For example, you might have a login form, a search form, and a form to subscribe to a newsletter all on the same page. If you do have more than one form on a page, users will be able to send the data from only one form at a time to the server.

Every <form> element should carry at least two attributes: action and method.

A <form> element may also carry all of the universal attributes, the UI event attributes, and the following attributes: ectype, accept, onsubmit, onreset.

The action Attribute

The action attribute indicates what happens to the data when the form is submitted. Usually the value of the action attribute is a page or program on a web server that will receive the information from this form when a user presses the submit button.

For example, if you had a login form consisting of a username and password, the details the user enters may get passed to the web server.
<form action= “http://www.example.org/membership/login.aspx”>

Most browsers will accept only a URL beginning with http:// as the value of the action attribute.

The method Attribute

You can send Form data to server in two ways: First way is by using Get method. Get method sends the data as part of URL. The second way is using Post method. Post method hides the data in the HTTP headers**.**

14.2.1 Form Controls

This section covers the different types of form controls that you can use to collect data from a visitor to your site.

- Text input
- Buttons
- Checkboxes and radio buttons
- Select boxes (sometimes referred to as drop-down menus) and list boxes
- Hidden controls

Text Inputs

You absolutely have come across text input boxes on many web pages. Possibly the most famous text input box is the one right in the middle of the Google home page that allows you to enter what you are searching for.

Password Input Controls

Password button is used to collect sensitive data like credit card information. When you are typing password field in screen it mask the characters by replacing them with asterisk.

Password input field is similar to text box, except that <input> element type attribute is specified a value of password.

Buttons

Mostly buttons are used to submit a form, in some case you can use to clear or reset a form. Buttons also supports, to trigger client side scripts.
Example: Consider Loan calculation application form, a button can be used to trigger the client side script that related to repayment.

Button can be created in three ways:

- using an <input> element with a type attribute whose value is submit, reset, or button

- using an <input> element with a type attribute whose value is image
- using a <button> element

With each different method, the button will appear slightly different.

Using Images for Buttons

You can use an image for a button rather than using the standard button that a browser renders for you.

Creating an image button is very similar to creating any other button, but the type attribute has a value of image:

<input type= “image” src= “submit.jpg” alt= “Submit” name= “ btnImageMap” />

Checkboxes

Checkboxes are just like the little boxes that you have to check on paper forms. As with light switches, they can be either on or off. When they are checked they are on and the user can simply toggle between on and off positions by clicking the checkbox.

Checkboxes can appear individually, with each having its own name, or they can appear as a group of checkboxes that share a control name and allow users to select several values for the same property.

Checkboxes are ideal form controls when you need to allow a user to:

- Provide a simple yes or no response with one control (such as accepting terms and conditions or subscribing to an e-mail list)
- Select several items from a list of possible options (such as when you want a user to indicate all of the skills they have from a given list)
- A checkbox is created using the <input> element whose type attribute has a value of checkbox.

Radio buttons

Radio buttons are similar to checkboxes in that they can be either on or off, but there are two key differences:

- When you have a group of radio buttons that share the same name, only one of them can be selected. Once one radio button has been selected, if the user clicks another option, the new option is selected and the old one deselected.

- You should not use radio buttons for a single form control where the control indicates on or off because once a lone radio button has been selected it cannot be deselected again (without writing a script to do that).

  Therefore, radio buttons are ideal if you want to provide users with a number of options from which they can pick only one. In such situations, an alternative is to use a drop-down select box that allows users to select only one option from several. Your decision between whether to use a select box or a group of radio buttons depends on three things:
- **Users expectations:** If your form models a paper form where users would be presented with several checkboxes, from which they can pick only one, then you should use a group of radio buttons.
- **Seeing all the options:** If users would benefit from having all theoptions in front of them before they pick one, you should use a group of radio buttons.
- **Space:** If you are concerned

Select Boxes

A drop-down select box allows users to select one item from a drop-down menu. Drop-down select boxes can take up far less space than a group of radio buttons.

Drop-down select boxes can also provide an alternative to single-line text input controls where you want to limit the options that a user can enter. For example, you can use a select box to allow users to indicate which country or state they live in (the advantage being that all users from the USA would have the same value, rather than potentially having people write U.S.A., U.S., United States, America, or North America and then having to deal with different answers for the same country).

A drop-down select box is contained by a <select> element, while each individual option within that list is contained within an <option> element.

For example, the following form creates a drop-down select box for the user to select a color

```
<select name= “selColor”>
<option selected= “selected” value=“>Select color</option>
```

<option value= “red”>Red</option>

<option value= “green”>Green</option>

<option value= “blue”>Blue</option>

</select>

As you can see here, the text between the opening <option> element and the closing </option> tags is used to display options to the user, while the value that would be sent to the server if that option is selected is given in thevalue attribute. You can also see that the first <option> element does not have a value and that its content is Select color; this is to indicate to the user that he or she must pick one of the color choices. Finally, notice again the use of the letters selColor at the start of the name of a select box.

Hidden Controls

Sometimes you will want to pass information between pages without the user seeing it;

<input type=”hidden” name= “hidPageSentFrom” value= “home page” />

For a name and value can still be sent to the server for a hidden form control, the hidden control must carry name and value attributes. Sending Form Data to the Server.

14.2.2 Validating Form Using JavaScript

When you create a form, providing form validation is useful to ensure that your user entered valid and complete data. For example, you may want to ensure that someone inserts a valid email address into textbox, or perhaps you want to ensure that someone fills in certain fields.

Data from forms can be checked on the client computer using JavaScript before being sent to the web server. Form validation often serves two purposes.

- The form must first be validated to ensure that all of the required fields are filled in. Just looping through every field in the form and checking for data would be sufficient. The user-submitted form has to be validated because it may contain values that are incorrect. Hence, user authentication requires validation.

  JavaScript offers the ability to do form validation on the client side, which speeds up data processing compared to server-side validation. JavaScript form validation is preferred by the majority of web developers.We can validate name, password, email, date, cell numbers, and other fields using JavaScript.

- Second, the entered data must be examined for accuracy of form and content data format validation. To test that the data is correct, your code must have the necessary logic.
  Before sending the completed form's data to the web server, we can validate it. The sample that follows demonstrates email address validation. At the very least, an email address needs to have a @ symbol and a dot (.). Moreover, the final dot must appear at least one character after the '@' symbol and cannot be the initial character of the email address.

In JavaScript it creates a set of "valid descriptors" associated with each element in a form. In order to validate a field, you just associate a set of validation descriptors for each input field in the form. Each field in the form can have zero one or more validations. For example, you have an input field that should not be empty, should be less than 25 chars and should be alpha-numeric.

Using the form validation script

1. Include sample.js in your html file before closing the head tag
2. After defining your form, create a validator() object passing the name of the form

3. Now, add the validations required
   Frmvalidator.addvalidation (“firstname”, “req”, “please enter your firstname”);
4. Finally add conditions for addvaliadtion() function
   Frmvalidator.addvalidation (Fieldname, validation descriptor, Err String)

   You can add any number of validations to a field
   Frmvalidator.addValidation (“First Name”, “req”, “please enter your first Name”);
   Frmvalidator.addValidation (“FirstName”, “maxlen=40”, “maxlength for firstname is 40”);

Example: Here is a complete example for checking username, email and phone number.

```
<form action= “ ” id= “myform” >
<p>
<label for=’Username’> FirstName: </label>
<input type= “text” id= “first Name” name= “Username” />
</P>
<p>
< label for= ‘Email’> Email: </label>
<input type= “text” id= “Email” name= “Email” />
</P>
<p>

<label for= ‘phone number’> phone number: </label>
<input type= “text” id= “phone number” name=” phone number” />
</p>
<p>
<input type= “submit” name= “submit” value = “submit”>
</p>
</form>
<script type= “text/JavaScript “>
```

```
Varfrmvalidator= new validator ("myform");
Frmvalidator. addValidation (("Username", "req", "please enter
                                        Your Name");
Frmvalidator.addValidation ("username", "maxlen=40", "maxlength for
username is 40");
Frmvalidator. addValidation (("Email" , maxlen=50);
Frmvalidator. addValidation (("Email" , "req");
Frmvalidator. addValidation (("Email" ,email);
Frmvalidator. addValidation (("phone number" , Maxlen=50);
Frmvalidator. addValidation (("phone number" , "numeric");
</script>
```

Here some more additional steps to use form validations:

- The form validators should be created only after defining the <form> tag
- Your form should be a notable name. If there are more than one form in the same page, you can add validations for each of them. The names of the forms and the validators should not same.
- You can't use the JavaScript onsubmit event of the form if you are using this validator script. It is because the validator script automatically overrides the onsubmit event.

JavaScript email validation

A crucial aspect of validating an HTML form is email validation. A string of ASCII letters, or a subset of them, called an email is divided into two halves by the @ sign. The first portion is made up of personal data, while the second part is the domain name where the email is registered.

The following ASCII characters are permitted in the personal information field:

- letters in capital and lowercase (A-Z and a-z)
- character numbers (0-9)
- Special characters include:! # $% &'* + - / =? _'| '

- A period, dot, or full stop (.), with the exception that they cannot be the email's first or last letter or appear consecutively.

The domain name includes: Letters, Digits, Hyphens and Dots

Regular expressions are one of the most often used techniques for email validation in JavaScript. Regular expressions are used by JavaScript to describe a character pattern.

Password validation

JavaScript's password validation procedure helps to make sure that a password satisfies certain requirements before being approved. This criteria may include standards for length, intricacy, and originality. Password validation can be done by either the user or the server, and it is a crucial component of any web application's security.

By checking passwords, we can prevent brute force assaults from compromising our users' accounts. Password validation can also aid in preventing users from selecting weak passwords that are simple to decipher. Password validation will be even more crucial in ensuring the security of our data as we all become more dependent on online accounts.

What Is the Need for Password Validation?

The process of detecting whether a password entered by a user is legitimate or not is known as password validation in JavaScript. There are several ways to accomplish this, but the most popular one is to see if the user-entered password satisfies the standards established by the system.

The following are some of the most crucial justifications for the significance of password validation:

- To make sure that the system is only accessible to those who are authorized.
- To guard against unwanted access to private information.
- To increase safety.
- To abide by rules.
- To cut back on support expenses.
- To make things more usable.

Self Assessment Questions

5. _____________buttons are similar to checkboxes.
6. In order to validate a field, you just associate a set of validation descriptors for each input field in the form. (True/False)

14.3 Summary

- Tables are made up of rows of cells.
- A link is specified using the <a> element. Anything between the opening <a> tag and the closing </a> tag becomes part of the link a user can click in a browser.
- The number of cells in each row determines the table's shape. You have learned how a form lives inside a <form> element and that inside a form there are one or more form controls. You have seen how the <input> element can be used to create several kinds of form controls, namely

single-line text input controls, checkboxes, radio buttons, buttons, and hidden form controls.

- Once you have created a form with its form controls, you need to ensure that each element is labeled properly so that users know what information they should enter or which selection they will be making.
- Finally, you learned when validations using JavaScript form.

14.4 Terminal Questions

1. Design a table with the following elements:
   a) Border b) cell padding c) cell spacing d) Summary
2. Describe following concepts
   a) Linking to e-mail address b) Linking to specific part of page
3. Explain different types of form controls?
4. Explain form validation using JavaScript?

14.5 Answers

Self Assessment Questions

1. dir
2. a) Cellspacing attribute
3. Destination anchor
4. True
5. Radio
6. True

Terminal Questions

1. Table> tag is used to create a table. To insert a border for table use <border> tag, for inserting summary use <summary> tag. For moredetails refer to section 14.1.1.
2. When you are developing long web page, the web page content is not fitting in single browser window and the user using scrollbar to read from top to bottom. Instead of this the convenient way is splitting page into specific parts and making the proper links between them. For more details refer section 14.1.2
3. There are number of forms available like button, checkbox, text box, submit button, radio button. Form controls are conscious between opening form<form> and closing <form>. For more detail refer section 14.2

4. In JavaScript creates a set of “valid descriptors” associated with each element in a form. For more details refer section 14.2.2

14.6 References

- Gosselin Don (2010) *principles of html xhtml and dhtml*. Eleventh edition.
- Hart Davis (2009). *HTML, XHTML and CSS quick step*s. McGraw-Hill Education.
- Gary Rebholz (2003). *How to use HTML and XHTML.* Preston Gralla Que Publishing, Eighth edition.