

## **UNIT - 2**

### **XHTML – 2**

2.1 Images

2.2 Hypertext Links

2.3 Lists

2.4 Tables

2.5 Forms

2.6 Frames

2.7 Syntactic differences between HTML and XHTML

## UNIT - 2

### XHTML – 2

HTML is defined using the Standard Generalized Markup Language (SGML) which is an ISO standard notation for describing text formatting languages. The Original version of HTML was designed in conjunction with the structure of the web and the first browser. It was designed to specify document structure at a higher and more abstract level. It also defines details of text such as font style, size and color. The use of web became more popular when the first browser MOSAIC was developed and marketed by Netscape. The next browser was Internet explorer by Microsoft.

#### **History of HTML**

HTML was the standard developed by Microsoft for web technologies. First HTML standard was HTML 2.0 was released in 1995. Next HTML standard was HTML 3.2 was released in 1997. The latest version of HTML standard is HTML 4.01 was released and approved by W3C in 1999. The XHTML1.0 standard was approved in early 2000 which is a redefinition of HTML 4.01 using XML IE7 and FIREFOX2 (FX2) support XHTML 1.1

### 2.1 Images

Images can be included to enhance appearance of the document. Most common methods of representing the images are GIF (graphical interchange format and JPEG (joint photographic experts group) format. The former is 8bit color representation whereas the latter is 24 bit color representation. The image tag specifies an image that appears in a document. It has attributes like src which specifies the source of the image.

```
<html xmlns = "http://www.w3.org/1999/xhtml">
```

```
<head><title> Images</title></head>
```

```
<body>
```

```
<h1> Twinkle twinkle
```

```
<h2>little star
```

```
<h3> how I wonder
```

```
<h4>what you are ???
```

```
up above the world so high
```

like a diamond in the sky.

```

</body>
</HTML>
```

XHTML validation

## 2.2 Hypertext Links

A hypertext link in a XHTML document acts as a pointer to some resource. It could be an XHTML document anywhere in the web or another place in the document currently displayed. Links that point to another document specifies the address of the document. It could be a filename, complete url, directory path and a filename. Links are specified in an attribute of an anchor tag `<a>` which is an inline tag. The anchor tag is the source of an link whereas the document is the target of the link. Links facilitate reader to click on links to learn more about a particular subtopic of interest and also return back to the location of the link

If target is in the same document as the link it is specified in the href attribute value by preceding the id with a pound sign(#)

```
<html xmlns = "http://www.w3.org/1999/xhtml">
<head><title> Images</title></head>
<body>
<h1> Twinkle twinkle
<h2>little star
<h3> how I wonder
<h4>what you are ???
up above the world so high like a diamond in the sky.
<p>
<a href = "C:\Documents and Settings\Administrator\My Documents\XHTML
programs\1my.html">The blue hill image document
</a>
<p>
```

```

</body>
</HTML>
<html xmlns = "http://www.w3.org/1999/xhtml">
<head><title> Images</title></head>
<body>
<h1 id = "twinkle" > Twinkle twinkle </h1>
<h2>little star</h2>
<h3> how I wonder </h3>
<h4 >what you are ???
up above the world so high
like a diamond in the sky.</h4>
<a href = "#twinkle">Which poem
</a>
</body>
</HTML>

```

# Twinkle twinkle

## little star

### how I wonder

what you are ??? up above the world so high like a diamond in the sky.

[Which poem](#)

## 2.3 Lists

XHTML provides simple and effective ways to specify both ordered and unordered lists `<ul>` `<ol>` are tags for unordered and ordered lists. Each item in a list is specified with an `<li>` tag. Any tags can appear in a list item including nested lists.

### Definition lists

They are used to specify list of terms and their definitions such as glossaries. They are given by <dl> tag which is a block tag. The definitions are specified as the content of <dd> tag and the definition list is given as the content of a <dt> tag

```
<html xmlns = "http://www.w3.org/1999/xhtml">
```

```
<head><title> ordered lists</title></head>
```

```
<body>
```

```
<h3 > Lists of poems</h1>
```

```
<ul>
```

```
<li> Twinkle twinkle</li>
```

```
<li> Baa Baa black sheep</li>
```

```
<li> pussy cat </li>
```

```
<li> Humpty dumpty</li>
```

```
</ul>
```

```
</body>
```

```
</html>
```

```
<body>
```

```
<h3 > Lists of poems</h1>
```

```
<ol>
```

```
<li> Twinkle twinkle</li>
```

```
<li> Baa Baa black sheep</li>
```

```
<li> pussy cat </li>
```

```
<li> Humpty dumpty</li>
```

```
</ol>
```

```
</body>
```

### **Lists of poems**

1. Twinkle twinkle
2. Baa Baa black sheep
3. pussy cat
4. Humpty dumpty

## Lists of poems

- Twinkle twinkle
- Baa Baa black sheep
- pussy cat
- Humpty dumpty

```
head<title> ordered lists</title></head>
```

```
<body>
```

```
<ol>
```

```
<li > Lists of poems
```

```
<ol>
```

```
<li> Twinkle twinkle</li>
```

```
<li> Baa Baa black sheep</li>
```

```
<li> pussy cat </li>
```

```
<li> Humpty dumpty</li>
```

```
</ol>
```

```
<ol>
```

```
</li>
```

```
<li > Lists of stories
```

```
<ol>
```

```
<li> Thirsty crow</li>
```

```
<li> Lion and the mouse</li>
```

```
<li> pussy cat </li>
```

```
<li> Midas touch</li>
```

```
</ol>
```

```
</li>
```

```
</ol>
```

1. Lists of poems
  1. Twinkle twinkle
  2. Baa Baa black sheep
  3. pussy cat
  4. Humpty dumpty
2. Lists of stories
  1. Thirsty crow
  2. Lion and the mouse
  3. pussy cat
  4. Midas touch

```
<html xmlns = "http://www.w3.org/1999/xhtml">
<head><title> ordered lists</title></head>
<body>
<h3 > Lists of poems </h3>
<dl>
<dt> 111 </dt>
<dd> Twinkle twinkle</dd>
<dt> 222 </dt>
<dd> Pily poly</dd>
<dt> 333 </dt>
<dd> Rudolf the reindeer</dd>
</dl>
</body>
</HTML>
```

### **Lists of poems**

```
111
    Twinkle twinkle
222
    Pily poly
333
    Rudolf the reindeer
```

## 2.4 Tables

A table is a matrix of cells, each possibly having content. The cells can include almost any element some cells have row or column labels and some have data. A table is specified as the content of a `<table>` tag which is a block tag. A border attribute in the `<table>` tag specifies a border between the cells. Rule specifies the lines that separate the cells.

If border is set to `border=""`, the browser's default width border is used. The border attribute can be set to a number, which will be the border width in pixels(0 is no border no rules). Without the border attribute, the table will have no lines!. Tables are given titles with the `<caption>` tag, which can immediately follow `<table>`.

Each row of a table is specified as the content of a `<tr>` tag. The row headings are specified as the content of a `<th>` tag. The contents of a data cell is specified as the content of a `<td>` tag. The empty cell is specified with a table header tag that includes no content `<th> </th>`.

```

<body>
<table border = "border">
<caption> fruit juice drinks </caption>
<tr>
<th> </th>
<th> Apple </th>
<th> Mango </th>
<th> Strawberry </th>
</tr>
<tr> <th> Breakfast </th>
<th> 0</th>
<th> 1 </th>
<th> 0</th>
</tr>
<tr> <th> lunch </th>
<th> 1</th>
<th> 1 </th>
<th>1 </th>
</tr>
<tr> <th> dinner </th>

```

```

<th> 0 </th>
<th> 1 </th>
<th> 0 </th></tr><table></body></HTML>

```

### Colspan Rowspan attributes

A table can have two levels column labels and also row labels. If so, the colspan attribute must be set in the <th> tag to specify that the label must span some number of columns.

```

<tr>
<th colspan = "3"> Fruit Juice Drinks </th>
</tr>
<tr>
<th> Orange </th>
<th> Apple </th>
<th> Screwdriver </th>
</tr>
<caption> fruit juice drinks </caption>
<tr>
<td rowspan="2"></td>
<th colspan="3">Juices chart</th>
</tr>
<tr>
<th> </th>

```

If the rows have labels and there is a spanning column label, the upper left corner must be made larger, using rowspan.

```

<table border = "border">
<tr>
<td rowspan = "2"> </td>
<th colspan = "3"> Fruit Juice Drinks
</th>
</tr>
<tr>
<th> Apple </th>
<th> Orange </th>
<th> Screwdriver </th>

```

```

</tr>
...
</table>

```

fruit juice drinks

	Apple	Mango	Strawberry
Breakfast	0	1	0
lunch	1	1	1
dinner	0	1	0

The align attribute controls the horizontal placement of the contents in a table cell. Values are left, right, and center (default) align is an attribute of <tr>, <th>, and <td> elements. The valign attribute controls the vertical placement of the contents of a table cell. Values are top, bottom, and center (default) valign is an attribute of <th> and <td> elements. The cellspacing attribute of <table> is used to specify the distance between cells in a table. The cellpadding attribute of <table> is used to specify the spacing between the content of a cell and the inner walls of the cell.

```

<head><title> simple table</title></head>
<body>
<table border = "border">
<caption> align</caption>
<tr align ="center">
<th> </th>
<th> Apple </th>
<th> Mango </th>
<th> Strawberry </th>
</tr>
<tr>
<th> align </th>
<td align="left">left</td>
<td align="center">center</td>
<td align="right">right</td>

```

```

</tr>
<tr>
<th> valign<br /> <br /></th>
<td >default</td>
<td valign="top">top</td>
<td valign="bottom">bottom</td>
</tr>
<table>
</body>
</HTML>

```

align

	<b>Apple</b>	<b>Mango</b>	<b>Strawberry</b>
<b>align</b>	left	center	right
<b>valign</b>	default	top	bottom

## 2.5 Forms

A form is the usual way information is gotten from a Web browser to a server. HTML has tags to create a collection of objects that implement this information gathering. The objects are called widgets (e.g., radio buttons and checkboxes). All control tags are inline tags. These controls gather information used from user in the form of either text or button selections. Each control has a value given through the user input. Collectively values of all these controls in a form are called the form data. When the Submit button of a form is clicked, the form's values are sent to the Web server for processing.

### The <form> tag

All of the widgets, or components of a form are defined in the content of a <form> tag which is a block tag. This tag can have many attributes of which the most required attribute is the action. The action attribute specifies the URL of the application on the web server that is to be called when the user clicks the Submit button.

Eg: action = <http://www.cs.ucp.edu/cgi-bin/survey.pl>

If the form has no action, the value of action is the empty string. The method attribute of <form> specifies one of the two techniques, get or post, used to pass the form data to the server. get is the default, so if no method attribute is given in the <form> tag, get will be used. The alternative technique is post. With these techniques the form data is encoded into text string on click of submit button

Widgets or Controls Many commonly used controls are created with the <input> tag which specifies the kind of control. It is used for the text, passwords, checkboxes, radio buttons and the action buttons Reset and Submit.

The type attribute of <input> specifies the kind of widget being created. Except Reset and Submit all other controls have a name attribute other than type attribute.

Text: Creates a horizontal box for text input

Default size is 20; it can be changed with the size Attribute. If more characters are entered than will fit, the box is scrolled (shifted) left. If you don't want to allow the user to type more characters than will fit, set max length, which causes excess input to be ignored

```
<input type = "text" name = "Phone" size = "12" >
```

If the contents of the textbox should not be displayed when user types it than a password control should be used. Labeling a text box can be done by adding label control to the text box. Both the controls can be encapsulated. This has several advantages. The text content of the label will indicate content of text box and when a label is selected the cursor is implicitly moved to the control.

```
<form action = "">
```

```
<p>
```

```
<input type = "text" name = "Phone" size = "12" /></p>
```

```
<p>
```

```
<input type = "password" name = "myPassword" size = "12" maxlength="12" />
```

```
</p>
```

```
<p>
```

```
<label>Phone:<input type = "text" name = "Phone" size = "12" /></label>
```

```
</p>
```

Phone: 

### Checkboxes

Checkboxes collect multiple choice input. Every checkbox requires a value attribute, which is the widget's value in the form data when the checkbox is `checked`. A checkbox that is not `checked` contributes no value to the form data. By default, no checkbox is initially `checked`. To initialize a checkbox to `checked`, the `checked` attribute must be set to `checked`.

### Radio Buttons

Radio buttons are collections of checkboxes in which only one button can be `checked` at a time. Every button in a radio button group **MUST** have the same name. If no button in a radio button group is `pressed`, the browser often `presses` the first one. Checkboxes and radio buttons are both multiple choice input from the user

```
<form action = "">
```

```
<p>
```

```
<input type = "checkbox" name = "groceries" value = "milk" checked = "checked">
```

Milk

```
<input type = "checkbox" name = "groceries" value = "bread">
```

Bread

```
<input type = "checkbox" name = "groceries" value = "eggs">
```

Eggs

```
</p>
```

```
</form>
```

```
<p> <input type = "radio" name = "age" value = "under20" checked = "checked">
```

0-19

```
<input type = "radio" name = "age" value = "20-35">
```

20-35

```
<input type = "radio" name = "age" value = "36-50">
```

36-50

```
<input type = "radio" name = "age" value = "over50">
```

Over 50

Milk  Bread  Eggs

0-19  20-35  36-50  Over 50

## Menus

Each item of a menu is specified with an `<option>` tag, whose pure text content (no tags) is the value of the item. An `<option>` tag can include the `selected` attribute, which when assigned "selected" specifies that the item is preselected. Menus - created with `<select>` tags. There are two kinds of menus, those that behave like checkboxes and those that behave like radio buttons (the default). Menus that behave like checkboxes are specified by including the `multiple` attribute, which must be set to "multiple". The `name` attribute of `<select>` is required. The `size` attribute of `<select>` can be included to specify the number of menu items to be displayed (the default is 1). If `size` is set to  $> 1$  or if `multiple` is specified, the menu is displayed as a pop-up menu

Text areas - created with `<textarea>`. Usually include the `rows` and `cols` attributes to specify the size of the text area. Default text can be included as the content of `<textarea>`. Scrolling is implicit if the area is overfilled.

Reset and Submit buttons both are created with `<input>`.

```
<input type = "reset" value = "Reset Form">
```

```
<input type = "submit" value = "Submit Form">
```

Submit has two actions:

1. Encode the data of the form
2. Request that the server execute the server-resident program specified as the value of the `action` attribute of `<form>`.

A Submit button is required in every form

```
<form action = ""> <p>
```

With `size = 1` (the default)

```
<select name = "groceries">
```

```
<option> milk </option>
```

```
<option> bread </option>
```

```

<option> eggs </option>
<option> cheese </option>
</select>
</p>
<p>
<textarea name = "aspirations" rows = "3"
cols = "40">
(Be brief in expressing your views)
</textarea>
</p>
<p>
<input type = "Submit" value = "Submit Form" />
<input type = "reset" value = "Reset Form" />
</p>
</form>

```

## 2.6 Frames

Frames are rectangular sections of the display window, each of which can display a different document. Because frames are no longer part of XHTML, you cannot validate a document that includes frames. The `<frameset>` tag specifies the number of frames and their layout in the window `<frameset>` takes the place of `<body>`. We Cannot have both! `<frameset>` must have either a `rows` attribute or a `cols` attribute, or both (usually the case). Default is 1. The possible values for `rows` and `cols` are numbers, percentages, and asterisks.

A number value specifies the row height in pixels. A percentage specifies the percentage of total window height for the row. An asterisk after some other specification gives the remainder of the height of the window

Examples:

```

<frameset rows = "150, 200, 300">
<frameset rows = "25%, 50%, 25%">
<frameset rows = "50%, 20%, *" >
<frameset rows = "50%, 25%, 25%"
cols = "40%, *">

```

The <frame> tag specifies the content of a frame. The first <frame> tag in a <frameset> specifies the content of the first frame, etc. An asterisk after some other specification gives the remainder of the height of the window

Examples:

```
<frameset rows = "150, 200, 300">
```

```
<frameset rows = "25%, 50%, 25%">
```

```
<frameset rows = "50%, 20%, *" >
```

```
<frameset rows = "50%, 25%, 25%"
```

```
cols = "40%, *">
```

The <frame> tag specifies the content of a frame. The first <frame> tag in a <frameset> specifies the content of the first frame, etc.

**Row-major order is used. Frame content is specified with the src attribute. Without a src attribute, the frame will be empty (such a frame CANNOT be filled later).** If

<frameset> has fewer <frame> tags than frames, the extra frames are empty Scrollbars are implicitly included if needed (they are needed if the specified document will not fit). If a name attribute is included, the content of the frame can be changed later (by selection of a link in some other frame)

Nested frames - to divide the screen in more interesting ways

```
<head> <title> Nested frames </title> </head>
```

```
<frameset cols = "40%, *">
```

```
<frameset rows = "50%, *" >
```

```
<frame src = "1my.html" />
```

```
<frame src = "12list.html" />
```

```
</frameset>
```

```
<frameset rows = "20%,35%, *" >
```

```
<frame src = "1my.html" />
```

```
<frame src = "12list.html" />
```

```
<frame src = "2twinkle.html" />
```

```
<frame src = "5head.html" />
```

```
</frameset>
```

```
</frameset>
```

```
</html>
```

## 2.7 Syntactic differences between HTML and XHTML

Case sensitivity: In HTML, tag and attribute names are case insensitive meaning that <FORM>, <form>, and <Form> are equivalent. In XHTML, all tag and attribute names must be in lowercase.

Closing tags: In HTML, closing tag may be omitted if the processing agent can infer their presence. For example, in HTML, paragraph elements often do not have closing tags. For example

```
<p> During Spring, flowers are born. ...
```

```
</p>
```

During Fall, flowers die.....

HTML documents are case insensitive whereas XHTML documents have to be in lowercase. Closing tags may be omitted in HTML but not in XHTML where all elements must have closing tags except for content tags where it is not a must

```
<input type= "text" name="address" />
```

Quoted attribute values :all attribute values must be quoted whether it is numeric or character based

Explicit attribute values: In HTML some attributes are implicit

Quoted attribute values: In HTML, attribute values must be quoted only if there are embedded special characters or white space characters.

Explicit attribute values: In HTML, some attribute values are implicit, that is, they need not be explicitly stated. For example, if the border attribute appears in a <table> tag without a value, it specifies a default width border on the table. For example: <table border>. id and name attributes. HTML markup often uses the name attribute for elements. Element nesting: Although HTML has rules against improper nesting of elements, they are not enforced.

Examples of nesting rules are:

1. An anchor element cannot contain another anchor element, and a form element cannot contain another form element.
2. If one element appears inside another element its closing tag should appear before the closing tag of the outer element.
3. Block elements cannot be nested inside inline elements.
4. Text cannot be nested directly in body or form elements.
5. List elements cannot be directly nested in list elements.