

Chapter-2: Introduction to HTML

HTML

- HTML stands for **Hyper Text Markup Language**
- HTML defines the structure and layout of a Web document by using a variety of tags and attributes.
- HTML is not a programming language, it is a **markup language**
- A markup language is a set of **markup tags**
- HTML uses **markup tags** to describe web pages

XHTML

- XHTML stands for Extensible Hyper Text Markup Language
- XHTML is almost identical to HTML 4.01
- XHTML is a stricter and cleaner version of HTML
- XHTML is HTML defined as an XML application. So it is a hybrid between HTML and XML
- XHTML is a W3C Recommendation.

Why XHTML??

- **Cleaner, more logical markup:** XHTML brings uniformity to document structure. XHTML help restore the structural integrity of documents.
- **Greater accessibility:** Because they follow strict rules and avoid non-standard markup, well-authored XHTML pages are more accessible than old-school HTML pages
- XML is a markup language where everything must be marked up correctly, which results in "well-formed" documents.

Difference between HTML and XHTML

- HTML is easier to write, whereas XHTML requires a level of discipline or strict rule and syntax.
- Some older browser have problems with XHTML.
- HTML has few syntactic rules than XHTML.
- XHTML elements must be properly nested
- XHTML elements must always be closed
- XHTML elements must be in lowercase
- XHTML documents must have one root element

Origins and Evolution of HTML and XHTML

- This is like the history of HTML and XHTML. I suggest you to read from the book **World Wide Web(Robert w.sebesta)**.

Basic Syntax of HTML

- Syntactic Units of HTML is called *tags*.
- HTML tags are keywords surrounded by **angle brackets** like <html>.
- HTML tags normally **come in pairs** like and .
- The first tag in a pair is the **start tag**, the second tag is the **end tag**.
- Start and end tags are also called **opening tags** and **closing tags**

1. **Heading:** HTML headings are defined with the <h1> to <h6> tags.

```
<h1>This is a heading</h1>
<h2>This is a heading</h2>
<h3>This is a heading</h3>
```

2. **Paragraphs:** <p> element defines a paragraph in the HTML document. Paragraphs are defined with the <p> tag.

```
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
```

3. **Links:** HTML links are defined with the <a> tag.

```
<a href="http://www.aboutus.com"> About Us</a>
```

4. **Images:** HTML images are defined with the tag.

```

```

5. **Line:** The <hr /> tag creates a horizontal line in an HTML page.

```
<p>This is a paragraph</p>
<hr />
<p>This is a paragraph</p>
<hr />
```

6. **Comments:** Comment is written like below. It is hidden in HTML document.

```
<!-- This is a comment -->
```

7. **Line Break:**
 tag is used for line break. It has no closing tag.

```
<p>This is<br />a para<br />graph with line breaks</p>
```

8. **Italic text:** <i> tag is used to make text italic.

```
<i>This text is italic</i>
```

9. **Bold text:** tag is used to make text bold.

```
<b>This text is bold</b>
```

10. **Code:** It makes the text like computer code.

`<code>This is computer output</code>`

11. **Superscript** and **Subscript:**

`<p>This is_{subscript} and ^{superscript}</p>`

12. **Del:** Defines deleted text. **Ins:** Defines inserted text:

`<p>My favorite color is blue <ins>red</ins>!</p>`

13. **Quotation:** For quotation. It makes sentences with inverted comma.

`<p>Here comes a short quotation: <q>This is a short quotation</q></p>`

14. **Abbreviation:** The `<abbr>` tag describes an abbreviated phrase.

The `<abbr title="World Health Organization">WHO</abbr>` was founded in 1948.

15. **Italic:** `<i>` tag is used to make the text italic.

`<i> This is bold text. </i>`

16. **Underline:** `<u>` tag is used to make the text underlined.

`<u> This is bold text. </u>`

Image in HTML:

In HTML, images are defined with the `` tag.

Attributes are:

src: For source of Images.

alt: The alt attribute provides alternative information for an image. While user put pointer over images information appears.

map: Defines an image-map.

area: Defines a clickable area inside an image-map.

height: Defines height of image.

width: Defines width of image.

align: Defines alignment of image.

Image Map:

The <map> tag is used to define a client-side image-map. An image-map is an image with clickable areas.

eg.

```


<map name="planetmap">
  <area shape="rect" coords="0,0,82,126" href="sun.htm" alt="Sun" />
  <area shape="circle" coords="90,58,3" href="mercur.htm" alt="Mercury"
/>
  <area shape="circle" coords="124,58,8" href="venus.htm" alt="Venus" />
</map>
```

Tables in HTML:

- A table is a matrix of rows and columns in which each intersection of rows and column is called cells. Tables are defined with the <table> tag.
- A table is divided into rows (with the <tr> tag), and each row is divided into data cells (with the <td> tag).

eg.

```
<table border="1">

  <tr>
    <th>Heading1</th>
    <th> Heading2</th>
  </tr>
  <tr>
    <td>row 2, cell 1</td>
    <td>row 2, cell 2</td>
  </tr>
  <tr>
    <td>row 3, cell 1</td>
    <td>row 3, cell 2</td>
  </tr>

</table>
```

Output will be:

Headinnng1	Heading2
row 2, cell 1	row 2, cell 2
row 3, cell 1	row 3, cell 2

Attributes Inside Table are:

- border: Defines the border thickness.
- colspan: Spans two or multiple columns in tables.
- rowspan: Spans two or multiple rows in tables.
- width: Defines width of Table.
- height: Defines height of Table.
- cell spacing: Increase the distance between the cells.
- cell padding: Create more white space between the cell content and its borders.

Colspan and Rowspan attributes: (Example:)

```
<table border="1">
<tr>
  <th>Name</th>
  <th colspan="2">Telephone</th>
</tr>
<tr>
  <td>Bill Gates</td>
  <td>555 77 854</td>
  <td>555 77 855</td>
</tr>
</table>

<table border="1">
<tr>
  <th>First Name:</th>
  <td>Bill Gates</td>
</tr>
<tr>
  <th rowspan="2">Telephone:</th>
  <td>555 77 854</td>
</tr>
<tr>
  <td>555 77 855</td>
</tr>
</table>
```

Output will

Colspan Example

Name	Telephone	
Bill Gates	555 77 854	555 77 855

Rowspan Example

First Name:	Bill Gates
Telephone:	555 77 854
	555 77 855

Cell spacing and Cell padding attributes: (Example:)

```
<h4>Without cellpadding:</h4>
<table border="1">
<tr>
  <td>First</td>
  <td>Row</td>
</tr>
<tr>
  <td>Second</td>
  <td>Row</td>
</tr>
</table>
```

```
<h4>With cellpadding:</h4>
<table border="1"
cellpadding="10">
<tr >
  <td>First</td>
  <td>Row</td>
</tr>
<tr>
  <td>Second</td>
  <td>Row</td>
</tr>
</table>
```

OUTPUT

Without cellpadding:

First	Row
Second	Row

With cellpadding:

First	Row
Second	Row

align and valign attributes:

The placement of the content within a table cell can be specified with align and valign attributes. Align attributes has possible values left, right and center. Valign means: vertical alignment. It has top, middle, bottom values.

```
<table border="1" height="150" align="center">
<tr valign="top">
```

```

        <td>First</td>
        <td>Row</td>
    </tr>
    <tr valign="bottom">
        <td>Second</td>
        <td>Row</td>
    </tr>
</table>

```

OUTPUT

First	Row
Second	Row
First	Row
Second	Row

Table section:

Table naturally occur in two and sometimes three parts: header, body and footer. These 3 parts can be denoted in XHTML with the `thead`, `tbody` and `tfoot` elements.

```

<html>
<head>
<style type="text/css">
thead {color:green}
tbody {color:blue;height:50px}
tfoot {color:red}
</style>
</head>
<body>

<table border="1">
  <thead>
    <tr>
      <th>Month</th>
      <th>Savings</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Sum</td>
      <td>$180</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td>Sum</td>
      <td>$180</td>
    </tr>
  </tfoot>
</table>

```

```

        </tr>
    </tfoot>
    <tbody>
        <tr>
            <td>January</td>
            <td>$100</td>
        </tr>
        <tr>
            <td>February</td>
            <td>$80</td>
        </tr>
    </tbody>
</table>

</body>
</html>

```

Month	Savings
January	\$100
February	\$80
Sum	\$180

Frames in HTML:

- With frames, you can display more than one HTML document in the same browser window.
- Each HTML document is called a frame, and each frame is independent of the others.
- The disadvantages of using frames are:
 - The web developer must keep track of more HTML documents
 - It is difficult to print the entire page

Frameset element:

- The frameset element holds two or more frame elements. Each frame element holds a separate document.
- The frameset element states only HOW MANY columns or rows there will be in the frameset. Attributes are:
 - **COLS**: how many cols in the frameset
 - **ROWS**: how many rows in the frameset
 - **FRAMEBORDER**: if the frames should have borders
 - **FRAMESPACING**: space between the frames
 - **BORDER**: space between frames
 - **BORDERCOLOR**: color of frame borders

Frame element:

- The <frame> tag defines one particular window (frame) within a frameset.


```
<frameset cols="25%,75%">
  <frame src="frame_a.htm" />
  <frame src="frame_b.htm" />
</frameset>
```

- The frameset column size can also be set in pixels (cols="200,500"), and one of the columns can be set to use the remaining space, with an asterisk (cols="25%,*").

```
<html>

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

  <frame src="frame_a.htm" />
  <frameset cols="25%,75%">
    <frame src="frame_b.htm" />
    <frame src="frame_c.htm" />
  </frameset>

</frameset>

</html>
```

- **Iframe:** Allow including other windows in single window. Some browser may not support iframe.

```
<iframe src="http://www.facebook.com"></iframe>
```

Attributes of frame are:

Src: Source file of the frame.

Noresize: Allow to resize the windows.

Scrolling: Allow to scroll the windows.

eg:

```
<frame src="menu.html" scrolling="auto" noresize>
```

```
----- o -----
```

Forms in HTML:

HTML forms are used to pass data to a server.

A form can contain input elements like text fields, checkboxes, radio-buttons, submit buttons and more.

A form can also contain select lists, textarea, fieldset, legend, and label elements.

The <form> tag is used to create an HTML form:

Syntax will be:

```
<form>
.
  input elements
.
</form>
```

Attributes of Form:

- **method** - We will only be using the post functionality of method, which sends the data without displaying any of the information to the visitor.
- **action** - Specifies the URL to send the data to. We will be sending our information to a fake email address.

Some of the HTML form elements are:

Input Element:

- The most important form element is the input element.
- The input element is used to select user information.
- An input element can vary in many ways, depending on the type attribute.
- An input element can be of type text field, checkbox, password, radio button, submit button, and more.

Text Field: `<input type="text" />` defines a one-line input field that a user can enter text. Default width of a text field is 20 characters.

- **type** - Determines what kind of input field it will be. Possible choices are text, submit, and password.
- **name** - Assigns a name to the given field so that you may reference it later.
- **size** - Sets the horizontal width of the field. The unit of measurement is in blank spaces.
- **maxlength** - Dictates the maximum number of characters that can be entered.

```
<form>
First name:
<input type="text" name="firstname" size="10" maxlength="40"/><br />
Last name:
<input type="text" name="lastname" size="10" maxlength="40"/>
</form>
```

Textarea: Defines a multi-line text input control.

```
<form method="post" action="a.html">
<textarea rows="5" cols="20" wrap="physical" name="comments">
Enter Comments Here
</textarea>
<input type="submit" value="Email Yourself">
</form>
```

Attributes of textarea:

- **wrap**
 - "off"
 - "virtual"
 - "physical"
- **rows**
- **name**

Virtual means that the viewer will see the words wrapping as they type their comments, but when the page is submitted to you, the web host, the document sent will not have wrapping words.

Physical means that the text will appear both to you, the web host, and the viewer including any

page breaks and additional spaces that may be inputted. The words come as they are.
Off of course, turns off word wrapping within the text area. One ongoing line.

Password Field: `<input type="password" />` defines a password field. The characters in a password field are masked (shown as asterisks or circles).

```
<input type="password" size="10" maxlength="10" name="password">
```

Radio buttons: `<input type="radio" />` defines a radio button. Radio buttons let a user select ONLY ONE one of a limited number of choices.

- *value* - specifies what will be sent if the user chooses this radio button. Only one value will be sent for a given group of radio buttons (see *name* for more information).
- *name* - defines which set of radio buttons that it is a part of. Below we have 2 groups: shade and size.

```
<form>
<input type="radio" name="sex" value="male" /> Male<br />
<input type="radio" name="sex" value="female" /> Female
</form>
```

Check boxes: `<input type="checkbox" />` defines a checkbox. Checkboxes let a user select ONE or MORE options of a limited number of choices. The check box's *name* and *value* attributes behave the same as a radio button.

```
<form>
<input type="checkbox" name="vehicle" value="Bike" /> I have a
bike<br />
<input type="checkbox" name="vehicle" value="Car" /> I have a car
</form>
```

Dropdown list (<select> tag): Drop down menus are created with the `<select>` and `<option>` tags. `<select>` is the list itself and each `<option>` is an available choice for the user.

```
<form method="post" action="a.html">
College Degree?
<select name="degree">
<option>Choose One</option>
<option>Some High School</option>
<option>High School Degree</option>
<option>Some College</option>
<option>Bachelor's Degree</option>
<option>Doctorate</option>
<input type="submit" value="Email Yourself">
</select>
</form>
```

Submit Button: `<input type="submit" />` defines a submit button. A submit button is used to send form data to a server. The data is sent to the page specified in the form's action attribute. The file defined in the action attribute usually does something with the received input:

```
<form name="input" action="html_form_action.asp" method="get">
Username: <input type="text" name="user" />
```

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

Reset Button: `<input type="submit" />` defines a submit button. It is used to reset all the form data.

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

Upload form: An upload form consists of three basic parts. The first being a hidden field. This hidden field does nothing more than limit the allowed file size of our uploaded file. The second part is the input field itself. In this field, the user has the option to type in the full local URL of the file or he/she may click the *browse* button to thumb through directory after directory. HTML codes this automatically when we place the *type="file"* attribute within the input tag.

```
<input type="hidden" name="MAX_FILE_SIZE" value="100" />
<input name="file" type="file" />
```