

2. Introduction to HTML & XHTML

HTML stands for Hyper Text Markup Language, which is the most widely used language on Web to develop web pages.

HTML was created by Berners-Lee in late 1991 but "HTML 2.0" was the first standard HTML specification which was published in 1995. HTML 4.01 was a major version of HTML and it was published in late 1999.

Though HTML 4.01 version is widely used but currently we are having HTML-5 version which is an extension to HTML 4.01, and this version was published in 2012.

Originally, HTML was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers.

Now, HTML is being widely used to format web pages with the help of different tags available in HTML.

2.1. Basic HTML Document

In its simplest form, following is an example of an HTML document:

```
<!DOCTYPE html>
<html>
<head>
<title>This is document title</title>
</head>
<body>
<h1>This is a heading</h1>
<p>Document content goes here.....</p>
</body>
</html>
```

2.2. Basic HTML Tags

2.2.1. Heading Tags

Any document starts with a heading. You can use different sizes for your headings. HTML also have six levels of headings, which use the elements <h1>, <h2>, <h3>, <h4>, <h5>, and <h6>. While displaying any heading, browser adds one line before and one line after that

heading.

```
<!DOCTYPE html>
<html>
<head>
<title>Heading Example</title>
</head>
<body>
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
<h4>This is heading 4</h4>
<h5>This is heading 5</h5>
<h6>This is heading 6</h6>
</body>
</html>
```

2.2.2. Paragraph Tag

The <p> tag offers a way to structure your text into different paragraphs. Each paragraph of text should go in between an opening <p> and a closing </p> tag as shown below in the example:

```
<!DOCTYPE html>
<html>
<head>
<title>Paragraph Example</title>
</head>
<body>
<p>Here is a first paragraph of text.</p>
<p>Here is a second paragraph of text.</p>
<p>Here is a third paragraph of text.</p>
</body>
</html>
```

2.2.3. Line Break

Whenever you use the `
` element, anything following it starts from the next line. This tag is an example of an empty element, where you do not need opening and closing tags, as there is nothing to go in between them.

The `
` tag has a space between the characters `br` and the forward slash. If you omit this space, older browsers will have trouble rendering the line break, while if you miss the forward slash character and just use `
` it is not valid in XHTML.

```
<!DOCTYPE html>
<html>
<head>
<title>Line Break Example</title>
</head>
<body>
<p>Hello<br />
You delivered your assignment on time.<br />
Thanks<br />
Mahnaz</p>
</body>
</html>
```

2.2.4. Center Tag

You can use `<center>` tag to put any content in the center of the page or any table cell.

```
<!DOCTYPE html>
<html>
<head>
<title>Centring Content Example</title>
</head>
<body>
<p>This text is not in the center.</p>
<center>
<p>This text is in the center.</p>
</center>
```

```
</body>  
</html>
```

2.2.5. Horizontal Rule

Horizontal lines are used to visually break up sections of a document. The `<hr>` tag creates a line from the current position in the document to the right margin and breaks the line accordingly.

For example, you may want to give a line between two paragraphs as in the given example below:

```
<!DOCTYPE html>  
<html>  
<head>  
<title>Horizontal Line Example</title>  
</head>  
<body>  
<p>This is paragraph one and should be on top</p>  
<hr />  
<p>This is paragraph two and should be at bottom</p>  
</body>  
</html>
```

Again `<hr />` tag is an example of the empty element, where you do not need opening and closing tags, as there is nothing to go in between them.

The `<hr />` element has a space between the characters `hr` and the forward slash. If you omit this space, older browsers will have trouble rendering the horizontal line, while if you miss the forward slash character and just use `<hr>` it is not valid in XHTML.

2.2.6. Pre-formatted Text

Sometimes you want your text to follow the exact format of how it is written in the HTML document. In those cases, you can use the preformatted tag `<pre>`.

Any text between the opening `<pre>` tag and the closing `</pre>` tag will preserve the formatting of the source document.

```
<!DOCTYPE html>  
<html>  
<head>
```

```
<title>Preserve Formatting Example</title>
</head>
<body>
<pre>
function testFunction( strText ){
alert (strText)
}
</pre>
</body>
</html>
```

2.2.7. Non Breaking Space

Suppose you want to use the phrase "12 Angry Men." Here you would not want a browser to split the "12, Angry" and "Men" across two lines:

In cases where you do not want the client browser to break text, you should use a nonbreaking space entity ` ` instead of a normal space. For example, when coding the "12 Angry Men" in a paragraph, you should use something similar to the following code:

```
<!DOCTYPE html>
<html>
<head>
<title>Nonbreaking Spaces Example</title>
</head>
<body>
<p>An example of this technique appears in the movie
"12&nbsp;Angry&nbsp;Men."</p>
</body>
</html>
```

2.2.8. Bold Tag

Anything that appears within `...` element is displayed in bold as shown below:

```
<! DOCTYPE html>
<html>
<head>
```

```
<title>Bold Text Example</title>
</head>
<body>
<p>The following word uses a <b>bold</b> typeface.</p>
</body>
</html>
```

2.2.9. Italics Tag

Anything that appears within <i>...</i> element is displayed in italicized as shown below:

```
<!DOCTYPE html>
<html>
<head>
<title>Italic Text Example</title>
</head>
<body>
<p>The following word uses a <i>italicized</i> typeface.</p>
</body>
```

2.2.10. Underline Tag

Anything that appears within <u>...</u> element is displayed with underline as shown below:

```
<!DOCTYPE html>
<html>
<head>
<title>Underlined Text Example</title>
</head>
<body>
<p>The following word uses a <u>underlined</u> typeface.</p>
</body>
</html>
```

2.2.11. Strikethrough Tag

Anything that appears within `<strike>...</strike>` element is displayed with strikethrough, which is a thin line through the text as shown below:

```
<!DOCTYPE html>
<html>
<head>
<title>Strike Text Example</title>
</head>
<body>
<p>The following word uses a <strike>strikethrough</strike> typeface.</p>
</body>
</html>
```

2.2.12. Superscript Tag

The content of a `^{...}` element is written in superscript; the font size used is the same size as the characters surrounding it but is displayed half a character's height above the other characters.

```
<!DOCTYPE html>
<html>
<head>
<title>Superscript Text Example</title>
</head>
<body>
<p>The following word uses a <sup>superscript</sup> typeface.</p>
</body>
</html>
```

2.2.13. Subscript Tag

The content of a `_{...}` element is written in subscript; the font size used is the same as the characters surrounding it, but is displayed half a character's height beneath the other characters.

```
<!DOCTYPE html>
<html>
```

```
<head>
<title>Subscript Text Example</title>
</head>
<body>
<p>The following word uses a <sub>subscript</sub> typeface.</p>
</body>
</html>
```

2.1.14. Division and Span Tags

The <div> and elements allow you to group together several elements to create sections or subsections of a page.

For example, you might want to put all of the footnotes on a page within a <div> element to indicate that all of the elements within that <div> element relate to the footnotes. You might then attach a style to this <div> element so that they appear using a special set of style rules.

```
<!DOCTYPE html>
<html>
<head>
<title>Div Tag Example</title>
</head>
<body>
<div id="menu" align="middle" >
<a href="/index.htm">HOME</a> |
<a href="/about/contact_us.htm">CONTACT</a> |
<a href="/about/index.htm">ABOUT</a>
</div>
<div id="content" align="left" bgcolor="white">
<h5>Content Articles</h5>
<p>Actual content goes here.....</p>
</div>
</body>
</html>
```

The element, on the other hand, can be used to group inline elements only. So, if

you have a part of a sentence or paragraph which you want to group together, you could use the `` element as follows :

```
<!DOCTYPE html>
<html>
<head>
<title>Span Tag Example</title>
</head>
<body>
<p>This is the example of <span style="color:green">span tag</span> and the <span
style="color:red">div tag</span> along with CSS</p>
</body>
</html>
```

These tags are commonly used with CSS to allow you to attach a style to a section of a page.

2.2.15. Blockquote Tag

When you want to quote a passage from another source, you should put it in between `<blockquote>...</blockquote>` tags.

Text inside a `<blockquote>` element is usually indented from the left and right edges of the surrounding text, and sometimes uses an italicized font.

```
<!DOCTYPE html>
<html>
<head>
<title>Blockquote Example</title>
</head>
<body>
<p>The following description of XHTML is taken from the W3C Web site:</p>
<blockquote>XHTML 1.0 is the W3C's first Recommendation for XHTML, following on
from earlier work on HTML 4.01, HTML 4.0, HTML 3.2 and HTML 2.0.</blockquote>
</body>
</html>
```

2.2.16. Code Tag

Any programming code to appear on a Web page should be placed inside `<code>...</code>` tags. Usually the content of the `<code>` element is presented in a monospaced font, just like the code in most programming books.

```
<!DOCTYPE html>
<html>
<head>
<title>Computer Code Example</title>
</head>
<body>
<p>Regular text. <code>This is code.</code> Regular text.</p>
</body>
</html>
```

2.2.17. Meta Tags

HTML lets you specify metadata - additional important information about a document in a variety of ways. The META elements can be used to include name/value pairs describing properties of the HTML document, such as author, expiry date, a list of keywords, document author etc.

The `<meta>` tag is used to provide such additional information. This tag is an empty element and so does not have a closing tag but it carries information within its attributes.

You can include one or more meta tags in your document based on what information you want to keep in your document but in general, meta tags do not impact physical appearance of the document so from appearance point of view, it does not matter if you include them or not.

Adding Meta Tags to Your Documents

You can add metadata to your web pages by placing `<meta>` tags inside the header of the document which is represented by `<head>` and `</head>` tags.

A meta tag can have following attributes in addition to core attributes:

Attribute Description

- **Name:** Name for the property. Can be anything. Examples include, keywords, description, author, revised, generator etc.
- **content:** Specifies the property's value.
- **scheme:** Specifies a scheme to interpret the property's value (as declared in the content attribute).

- **http-equiv:** Used for http response message headers. For example http-equiv can be used to refresh the page or to set a cookie. Values include content-type, expires, refresh and set-cookie.

Specifying Keywords

You can use <meta> tag to specify important keywords related to the document and later these keywords are used by the search engines while indexing your webpage for searching purpose.

Following is an example where we are adding HTML, Meta Tags, Metadata as important keywords about the document.

```
<!DOCTYPE html>
<html>
<head>
<title>Meta Tags Example</title>
<meta name="keywords" content="HTML, Meta Tags, Metadata" />
</head>
<body>
<p>Hello HTML5!</p>
</body>
</html>
```

You can use <meta> tag to give a short description about the document. This again can be used by various search engines while indexing your webpage for searching purpose.

```
<!DOCTYPE html>
<html>
<head>
<title>Meta Tags Example</title>
<meta name="keywords" content="HTML, Meta Tags, Metadata" />
<meta name="description" content="Learning about Meta Tags." />
</head>
<body>
<p>Hello HTML5!</p>
</body>
</html>
```

Document Revision Date

You can use <meta> tag to give information about when last time the document was updated. This information can be used by various web browsers while refreshing your webpage.

```
<!DOCTYPE html>
<html>
<head>
<title>Meta Tags Example</title>
<meta name="keywords" content="HTML, Meta Tags, Metadata" />
<meta name="description" content="Learning about Meta Tags." />
<meta name="revised" content="google, 3/7/2014" />
</head>
<body>
<p>Hello HTML5!</p>
</body>
</html>
```

Document Refreshing

A <meta> tag can be used to specify a duration after which your web page will keep refreshing automatically.

If you want your page keep refreshing after every 5 seconds then use the following syntax:

```
<!DOCTYPE html>
<html>
<head>
<title>Meta Tags Example</title>
<meta name="keywords" content="HTML, Meta Tags, Metadata" />
<meta name="description" content="Learning about Meta Tags." />
<meta name="revised" content="google, 3/7/2014" />
<meta http-equiv="refresh" content="5" />
</head>
<body>
<p>Hello HTML5!</p>
</body>
</html>
```

Page Redirection

You can use <meta> tag to redirect your page to any other webpage. You can also specify a duration if you want to redirect the page after a certain number of seconds.

Following is an example of redirecting current page to another page after 5 seconds. If you want to redirect page immediately then do not specify content attribute.

```
<!DOCTYPE html>
<html>
<head>
<title>Meta Tags Example</title>
<meta name="keywords" content="HTML, Meta Tags, Metadata" />
<meta name="description" content="Learning about Meta Tags." />
<meta name="revised" content="google, 3/7/2014" />
<meta http-equiv="refresh" content="5" url=http://www.google.com" />
</head>
<body>
<p>Hello HTML5!</p>
</body>
</html>
```

Setting Cookies

Cookies are data, stored in small text files on your computer and it is exchanged between web browser and web server to keep track of various information based on your web application need.

You can use <meta> tag to store cookies on client side and later this information can be used by the Web Server to track a site visitor.

Following is an example of redirecting current page to another page after 5 seconds. If you want to redirect page immediately then do not specify content attribute.

```
<!DOCTYPE html>
<html>
<head>
<title>Meta Tags Example</title>
<meta name="keywords" content="HTML, Meta Tags, Metadata" />
<meta name="description" content="Learning about Meta Tags." />
<meta name="revised" content="google, 3/7/2014" />
```

```
<meta http-equiv="cookie" content="userid=xyz; expires=Wednesday, 08-Aug-15
23:59:59 GMT;"
/>
</head>
<body>
<p>Hello HTML5!</p>
</body>
</html>
```

If you do not include the expiration date and time, the cookie is considered a session cookie and will be deleted when the user exits the browser.

You can set an author name in a web page using meta tag. See an example below:

```
<!DOCTYPE html>
<html>
<head>
<title>Meta Tags Example</title>
<meta name="keywords" content="HTML, Meta Tags, Metadata" />
<meta name="description" content="Learning about Meta Tags." />
<meta name="author" content="Mahnaz Mohtashim" />
</head>
<body>
<p>Hello HTML5!</p>
</body>
</html>
```

Specify Character Set

You can use <meta> tag to specify character set used within the webpage.

By default, Web servers and Web browsers use ISO-8859-1 (Latin1) encoding to process Web pages. Following is an example to set UTF-8 encoding:

```
<!DOCTYPE html>
<html>
<head>
<title>Meta Tags Example</title>
<meta name="keywords" content="HTML, Meta Tags, Metadata" />
```

```
<meta name="description" content="Learning about Meta Tags." />
<meta ame="author" content="Mahnaz Mohtashim" />
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</head>
<body>
<p>Hello HTML5!</p>
</body>
</html>
```

To serve the static page with traditional Chinese characters, the webpage must contain a `<meta>` tag to set Big5 encoding:

```
<!DOCTYPE html>
<html>
<head>
<title>Meta Tags Example</title>
<meta name="keywords" content="HTML, Meta Tags, Metadata" />
<meta name="description" content="Learning about Meta Tags." />
<meta ame="author" content="Mahnaz Mohtashim" />
<meta http-equiv="Content-Type" content="text/html; charset=Big5" />
</head>
<body>
<p>Hello HTML5!</p>
</body>
</html>
```

2.2.18. HTML Comments

Comment is a piece of code which is ignored by any web browser. It is a good practice to add comments into your HTML code, especially in complex documents, to indicate sections of a document, and any other notes to anyone looking at the code. Comments help you and others understand your code and increases code readability.

HTML comments are placed in between `<!-- ... -->` tag. So any content placed within `<!-- ... -->` tag will be treated as comment and will be completely ignored by the browser.

```
<!DOCTYPE html>
<html>
```

```
<head> <!-- Document Header Starts -->
<title>This is document title</title>
</head> <!-- Document Header Ends -->
<body>
<p>Document content goes here.....</p>
</body>
</html>
```

Comments do not nest which means a comment cannot be put inside another comment. Second the double-dash sequence "--" may not appear inside a comment except as part of the closing --> tag. You must also make sure that there are no spaces in the start-of-comment string.

Here given comment is a valid comment and will be wiped off by the browser.

```
<!DOCTYPE html>
<html>
<head>
<title>Valid Comment Example</title>
</head>
<body>
<!--
This is valid comment -->
<p>Document content goes here.....</p>
</body>
</html>
```

But following line is not a valid comment and will be displayed by the browser. This is because there is a space between the left angle bracket and the exclamation mark.

```
<!DOCTYPE html>
<html>
<head>
<title>Invalid Comment Example</title>
</head>
<body>
< !--
```



```
This is not a valid comment -->
<p>Document content goes here.....</p>
</body>
</html>
```

So far we have seen single line comments, but HTML supports multi-line comments as well. You can comment multiple lines by the special beginning tag `<!--` and ending tag `-->` placed before the first line and end of the last line as shown in the given example below.

```
<!DOCTYPE html>
<html>
<head>
<title>Multiline Comments</title>
</head>
<body>
<!--
This is a multiline comment and it can
span through as many as lines you like.
-->
<p>Document content goes here.....</p>
</body>
</html>
```

Conditional comments only work in Internet Explorer (IE) on Windows but they are ignored by other browsers. They are supported from Explorer 5 onwards, and you can use them to give conditional instructions to different versions of IE.

```
<!DOCTYPE html>
<html>
<head>
<title>Conditional Comments</title>
<!--[if IE 6]>
Special instructions for IE 6 here
<![endif]-->
</head>
<body>
```

```
<p>Document content goes here.....</p>
</body>
</html>
```

You will come across a situation where you will need to apply a different style sheet based on different versions of Internet Explorer, in such situation conditional comments will be helpful.

Using Comment Tag

There are few browsers that support <comment> tag to comment a part of HTML code.

```
<!DOCTYPE html>
<html>
<head>
<title>Using Comment Tag</title>
</head>
<body>
<p>This is <comment>not</comment> Internet Explorer.</p>
</body>
</html>
```

2.2.19. Image Tag

Images are very important to beautify as well as to depict many complex concepts in simple way on your web page. This tutorial will take you through simple steps to use images in your web pages.

Insert Image

You can insert any image in your web page by using tag. Following is the simple syntax to use this tag.

```

```

The tag is an empty tag, which means that it can contain only list of attributes and it has no closing tag.

```
<!DOCTYPE html>
<html>
<head>
<title>Using Image in Webpage</title>
</head>
```

```
<body>
<p>Simple Image Insert</p>

</body>
</html>
```

You can use PNG, JPEG or GIF image file based on your comfort but make sure you specify correct image file name in src attribute. Image name is always case sensitive.

The alt attribute is a mandatory attribute which specifies an alternate text for an image, if the image cannot be displayed.

Set Image Location

Usually we keep our all the images in a separate directory. So let's keep HTML file test.htm in our home directory and create a subdirectory images inside the home directory where we will keep our image test.png.

Assuming our image location is "image/test.png", try the following example:

```
<!DOCTYPE html>
<html>
<head>
<title>Using Image in Webpage</title>
</head>
<body>
<p>Simple Image Insert</p>

</body>
</html>
```

You can set image width and height based on your requirement using width and height attributes. You can specify width and height of the image in terms of either pixels or percentage of its actual size.

```
<!DOCTYPE html>
<html>
<head>
<title>Set Image Width and Height</title>
</head>
<body>
```

```
<p>Setting image width and height</p>

</body>
</html>
```

By default image will have a border around it, you can specify border thickness in terms of pixels using border attribute. A thickness of 0 means, no border around the picture.

```
<!DOCTYPE html>
<html>
<head>
<title>Set Image Border</title>
</head>
<body>
<p>Setting image Border</p>

</body>
</html>
```

By default image will align at the left side of the page, but you can use align attribute to set it in the center or right.

```
<!DOCTYPE html>
<html>
<head>
<title>Set Image Alignment</title>
</head>
<body>
<p>Setting image Alignment</p>

</body>
</html>
```

2.2.20. Table Tag

The HTML tables allow web authors to arrange data like text, images, links, other tables, etc. into rows and columns of cells.

The HTML tables are created using the <table> tag in which the <tr> tag is used to create table rows and the <td> tag is used to create data cells.

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Tables</title>
</head>
<body>
```

your top row as table heading as shown below, otherwise you can use <th> element in any row.

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Table Header</title>
</head>
<body>
<table border="1">
<tr>
<th>Name</th>
<th>Salary</th>
</tr>
<tr>
<td>Ramesh Raman</td>
<td>5000</td>
</tr>
<tr>
<td>Shabbir Hussein</td>
<td>7000</td>
</tr>
</table>
</body>
</html>
```

There are two attributes called cellpadding and cellspacing which you will use to adjust the white space in your table cells. The cellspacing attribute defines the width of the border, while cellpadding represents the distance between cell borders and the content within a cell.

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Table Cellpadding</title>
</head>
<body>
64HTML Tutorial
<table border="1" cellpadding="5" cellspacing="5">
<tr>
<th>Name</th>
<th>Salary</th>
</tr>
<tr>
<td>Ramesh Raman</td>
<td>5000</td>
</tr>
<tr>
<td>Shabbir Hussein</td>
<td>7000</td>
</tr>
</table>
</body>
</html>
```

You will use colspan attribute if you want to merge two or more columns into a single column. Similar way you will use rowspan if you want to merge two or more rows.

```
<!DOCTYPE html>
<html>
<head>
```

```

<title>HTML Table Colspan/Rowspan</title>
</head>
<body>
<table border="1">
<tr>
<th>Column 1</th>
<th>Column 2</th>
<th>Column 3</th>
</tr>
<tr><td rowspan="2">Row 1 Cell 1</td><td>Row 1 Cell 2</td><td>Row 1 Cell
3</td></tr>
<tr><td>Row 2 Cell 2</td><td>Row 2 Cell 3</td></tr>
<tr><td colspan="3">Row 3 Cell 1</td></tr>
</table>
</body>
</html>

```

You can set a table width and height using width and height attributes. You can specify table width or height in terms of pixels or in terms of percentage of available screen area.

```

<!DOCTYPE html>
<html>
<head>
<title>HTML Table
Width/Height</title>
</head>
<body>
<table border="1"
width="400" height="150">
<tr>
<td>Row 1, Column
1</td>
<td>Row 1, Column
2</td>

```

```
</tr>
<tr>
<td>Row 2, Column 1</td>
<td>Row 2, Column 2</td>
</tr>
</table>
</body>
</html>
```

The caption tag will serve as a title or explanation for the table and it shows up at the top of the table. This tag is deprecated in newer version of HTML/XHTML.

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Table Caption</title>
</head>
<body>
<table border="1" width="100%">
<caption>This is the caption</caption>
<tr>
<td>row 1, column 1</td><td>row 1, columnn 2</td>
</tr>
<tr>
<td>row 2, column 1</td><td>row 2, columnn 2</td>
</tr>
</table>
</body>
</html>
```

Tables can be divided into three portions: a header, a body, and a footer. The head and foot are rather similar to headers and footers in a word-processed document that remain the same for every page, while the body is the main content holder of the table.

The three elements for separating the head, body, and foot of a table are:

- <thead> - to create a separate table header.
- <tbody> - to indicate the main body of the table.
- <tfoot> - to create a separate table footer.

A table may contain several <tbody> elements to indicate different pages or groups of data. But it is notable that <thead> and <tfoot> tags should appear before <tbody>

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Table</title>
</head>
<body>
<table border="1" width="100%">
<thead>
<tr>
<td colspan="4">This is the head of the table</td>
</tr>
</thead>
<tfoot>
<tr>
<td colspan="4">This is the foot of the table</td>
</tr>
</tfoot>
<tbody>
<tr>
<td>Cell 1</td>
<td>Cell 2</td>
<td>Cell 3</td>
<td>Cell 4</td>
</tr>
```

```
</tbody>
</table>
</body>
</html>
```

You can use one table inside another table. Not only tables you can use almost all the tags inside table data tag <td>.

Following is the example of using another table and other tags inside a table cell.

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Table</title>
</head>
<body>
<table border="1" width="100%">
<tr>
<td>
<table border="1" width="100%">
<tr>
<th>Name</th>
<th>Salary</th>
</tr>
<tr>
<td>Ramesh Raman</td>
<td>5000</td>
</tr>
<tr>
<td>Shabbir Hussein</td>
<td>7000</td>
</tr>
</table>
</td>
```

```
</tr>
</table>
</body>
</html>
```

2.2.21. List Tags

HTML offers web authors three ways for specifying lists of information. All lists must contain one or more list elements. Lists may contain:

- `` - An unordered list. This will list items using plain bullets.
- `` - An ordered list. This will use different schemes of numbers to list your items.
- `<dl>` - A definition list. This arranges your items in the same way as they are arranged in a dictionary.

HTML Unordered Lists

An unordered list is a collection of related items that have no special order or sequence. This list is created by using HTML `` tag. Each item in the list is marked with a bullet.

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Unordered List</title>
</head>
<body>
<ul>
<li>Beetroot</li>
<li>Ginger</li>
<li>Potato</li>
<li>Radish</li>
</ul>
</body>
</html>
```

You can use type attribute for `` tag to specify the type of bullet you like. By default it is a disc. Following are the possible options:

```
<ul type="square">
```

```
<ul type="disc">  
<ul type="circle">
```

Following is an example where we used <ul type="square">

```
<!DOCTYPE html>  
<html>  
<head>  
<title>HTML Unordered List</title>  
</head>  
<body>  
<ul type="square">  
<li>Beetroot</li>  
<li>Ginger</li>  
<li>Potato</li>  
<li>Radish</li>  
</ul>  
</body>  
</html>
```

Following is an example where we used <ul type="disc">

```
<!DOCTYPE html>  
<html>  
<head>  
<title>HTML Unordered List</title>  
</head>  
72HTML Tutorial  
<body>  
<ul type="disc">  
<li>Beetroot</li>  
<li>Ginger</li>  
<li>Potato</li>  
<li>Radish</li>  
</ul>
```

```
</body>  
</html>
```

Following is an example where we used `<ul type="circle">`:

```
<!DOCTYPE html>  
<html>  
<head>  
<title>HTML Unordered List</title>  
</head>  
<body>  
<ul type="circle">  
<li>Beetroot</li>  
<li>Ginger</li>  
<li>Potato</li>  
<li>Radish</li>  
</ul>  
</body>  
</html>
```

If you are required to put your items in a numbered list instead of bulleted list then HTML ordered list will be used. This list is created by using `` tag.

The numbering starts at one and is incremented by one for each successive ordered list element tagged with ``.

```
<!DOCTYPE html>  
<html>  
<head>  
<title>HTML Ordered List</title>  
</head>  
<body>  
<ol>  
<li>Beetroot</li>  
<li>Ginger</li>  
<li>Potato</li>  
<li>Radish</li>
```

```
</ol>
</body>
</html>
```

You can use type attribute for tag to specify the type of numbering you like. By default it is a number. Following are the possible options:

- <ol type="1"> -- Default-Case Numerals.
- <ol type="I"> -- Upper-Case Roman Numerals.
- <ol type="i"> -- Lower-Case Roman Numerals.
- <ol type="a"> -- Lower-Case Letters.
- <ol type="A"> -- Upper-Case Letters.

Following is an example where we used <ol type="1">

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Ordered List</title>
</head>
<body>
<ol type="1">
<li>Beetroot</li>
74HTML Tutorial
<li>Ginger</li>
<li>Potato</li>
<li>Radish</li>
</ol>
</body>
</html>
```

Following is an example where we used <ol type="I">

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Ordered List</title>
```

```
</head>
<body>
<ol type="I">
<li>Beetroot</li>
<li>Ginger</li>
<li>Potato</li>
<li>Radish</li>
</ol>
</body>
</html>
```

Following is an example where we used <ol type="I">

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Ordered List</title>
</head>
75HTML Tutorial
<body>
<ol type="I">
<li>Beetroot</li>
<li>Ginger</li>
<li>Potato</li>
<li>Radish</li>
</ol>
</body>
</html>
```

Following is an example where we used <ol type="A">

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Ordered List</title>
```

```
</head>
<body>
<ol type="A">
<li>Beetroot</li>
<li>Ginger</li>
<li>Potato</li>
<li>Radish</li>
</ol>
</body>
</html>
```

Following is an example where we used `<ol type="a">`

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Ordered List</title>
</head>
<body>
<ol type="a">
<li>Beetroot</li>
<li>Ginger</li>
<li>Potato</li>
<li>Radish</li>
</ol>
</body>
</html>
```

You can use start attribute for `` tag to specify the starting point of numbering you need. Following are the possible options:

- `<ol type="1" start="4">` -- Numerals starts with 4.
- `<ol type="I" start="4">` -- Numerals starts with IV.
- `<ol type="i" start="4">` -- Numerals starts with iv.
- `<ol type="a" start="4">` -- Letters starts with d.

- `<ol type="A" start="4">` -- Letters starts with D.

Following is an example where we used `<ol type="i" start="4">`

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Ordered List</title>
</head>
<body>
<ol type="i" start="4">
<li>Beetroot</li>
<li>Ginger</li>
<li>Potato</li>
<li>Radish</li>
</ol>
</body>
</html>
```

HTML and XHTML support a list style which is called definition lists where entries are listed like in a dictionary or encyclopedia. The definition list is the ideal way to present a glossary, list of terms, or other name/value list.

Definition List makes use of following three tags.

- `<dl>` - Defines the start of the list
- `<dt>` - A term
- `<dd>` - Term definition
- `</dl>` - Defines the end of the list

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Definition List</title>
</head>
<body>
```

```
<dl>
<dt><b>HTML</b></dt>
<dd>This stands for Hyper Text Markup Language</dd>
<dt><b>HTTP</b></dt>
<dd>This stands for Hyper Text Transfer Protocol</dd>
</dl>
</body>
</html>
```

2.2.22. Anchor Tag

A webpage can contain various links that take you directly to other pages and even specific parts of a given page. These links are known as hyperlinks. Hyperlinks allow visitors to navigate between Web sites by clicking on words, phrases, and images. Thus you can create hyperlinks using text or images available on a webpage.

A link is specified using HTML tag <a>. This tag is called anchor tag and anything between the opening <a> tag and the closing tag becomes part of the link and a user can click that part to reach to the linked document.

Following is the simple syntax to use <a> tag.

```
<a href="Document URL" ... attributes-list>Link Text</a>
```

Let's try following example which links <http://www.google.com> at your page:

```
<!DOCTYPE html>
<html>
<head>
<title>Hyperlink Example</title>
</head>
<body>
<p>Click following link</p>
<a href="http://www.google.com" target="_self">Tutorials Point</a>
</body>
</html>
```

We have used target attribute in our previous example.

Use of Base Path

When you link HTML documents related to the same website, it is not required to give a

complete URL for every link. You can get rid of it if you use <base> tag in your HTML document header. This tag is used to give a base path for all the links. So your browser will concatenate given relative path to this base path and will make a complete URL.

Following example makes use of <base> tag to specify base URL and later we can use relative path to all the links instead of giving complete URL for every link.

```
<!DOCTYPE html>
<html>
<head>
<title>Hyperlink Example</title>
<base href="http://www.google.com/">
</head>
<body>
<p>Click following link</p>
<a href="/html/index.htm" target="_blank">HTML Tutorial</a>
</body>
</html>
```

This will produce following result, where you can click on the link generated HTML Tutorial to reach to the HTML tutorial.

Linking to a Page Section

You can create a link to a particular section of a given webpage by using name attribute. This is a two-step process.

First create a link to the place where you want to reach within a webpage and name it using <a...> tag as follows:

```
<h1>HTML Text Links <a name="top"></a></h1>
```

Second step is to create a hyperlink to link the document and place where you want to reach:

```
<a href="/html/html_text_links.htm#top">Go to the Top</a>
```

This will produce following link, where you can click on the link generated Go to the Top to reach to the top of the HTML Text Link tutorial.

You can create text link to make your PDF, or DOC or ZIP files downloadable. This is very simple, you just need to give complete URL of the downloadable file as follows:

```
<!DOCTYPE html>
<html>
<head>
```

```
<title>Hyperlink Example</title>
</head>
<a href="http://www.google.com/page.pdf">Download PDF File</a>
</body>
</html>
```

We have seen how to create hypertext link using text and we also learnt how to use images in our webpages. Now we will learn how to use images to create hyperlinks.

It's simple to use an image as hyperlink. We just need to use an image inside hyperlink at the place of text as shown below:

```
<!DOCTYPE html>
<html>
<head>
<title>Image Hyperlink Example</title>
</head>
<body>
<p>Click following link</p>
<a href="http://www.google.com" target="_self">

</a>
</body>
</html>
```

This was the simplest way of creating hyperlinks using images. Next we will see how we can create Mouse-Sensitive Image Links.

Mouse-Sensitive Images

The HTML and XHTML standards provide a feature that lets you embed many different links inside a single image. You can create different links on the single image based on different coordinates available on the image. Once different links are attached to different coordinates, we can click different parts of the image to open target documents. Such mouse-sensitive images are known as image maps.

There are two ways to create image maps:

- Server-side image maps - This is enabled by the ismap attribute of the tag and requires access to a server and related image-map processing applications.
- Client-side image maps - This is created with the usemap attribute of the tag, along with corresponding <map> and <area> tags.

Server-Side Image Maps

Here you simply put your image inside a hyperlink and use ismap attribute which makes it special image and when the user clicks some place within the image, the browser passes the coordinates of the mouse pointer along with the URL specified in the <a> tag to the web server. The server uses the mouse- pointer coordinates to determine which document to deliver back to the browser.

When ismap is used, the href attribute of the containing <a> tag must contain the URL of a server application like a cgi or PHP script etc. to process the incoming request based on the passed coordinates.

The coordinates of the mouse position are screen pixels counted from the upper-left corner of the image, beginning with (0,0). The coordinates, preceded by a question mark, are added to the end of the URL.

2.2.23. Frameset and Frame Tags

HTML frames are used to divide your browser window into multiple sections where each section can load a separate HTML document. A collection of frames in the browser window is known as a frameset. The window is divided into frames in a similar way the tables are organized: into rows and columns.

Disadvantages of Frames

There are few drawbacks with using frames, so it's never recommended to use frames in your webpages:

- Some smaller devices cannot cope with frames often because their screen is not big enough to be divided up.
- Sometimes your page will be displayed differently on different computers due to different screen resolution.
- The browser's back button might not work as the user hopes.
- There are still few browsers that do not support frame technology.

Creating Frames

To use frames on a page we use <frameset> tag instead of <body> tag. The <frameset> tag defines how to divide the window into frames. The rows attribute of <frameset> tag defines horizontal frames and cols attribute defines vertical frames. Each frame is indicated by <frame> tag and it defines which HTML document shall open into the frame.

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Frames</title>
</head>
```

```
<noframes>
<body>
Your browser does not support frames.
</body>
</noframes>
</frameset>
</html>
```

Let's put above example as follows, here we replaced rows attribute by cols and changed their width. This will create all the three frames vertically:

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Frames</title>
</head>
<frameset cols="25%,50%,25%">
<frame name="left" src="/html/top_frame.htm" />
<frame name="center" src="/html/main_frame.htm" />
<frame name="right" src="/html/bottom_frame.htm" />
<noframes>
<body>
Your browser does not support frames.
</body>
</noframes>
</frameset>
</html>
```

Following are important attributes of the <frameset> tag:

- **Attribute Description :** cols specifies how many columns are contained in the frameset and the size of each column. You can specify the width of each column in one of four ways:
 - **Absolute values in pixels.** For example to create three vertical frames, use cols="100, 500,100".
 - **A percentage of the browser window.** For example to create three vertical

frames, use cols="10%, 80%,10%".

- **Using a wildcard symbol.** For example to create three vertical frames, use cols="10%, *,10%". In this case wildcard takes remainder of the window.
- **framespacing** This attribute specifies the amount of space between frames in a frameset. This can take any integer value. For example framespacing="10" means there should be 10 pixels spacing between each frames.

The <frame> Tag Attributes

Following are important attributes of <frame> tag:

- **Attribute Description :** src This attribute is used to give the file name that should be loaded in the frame. Its value can be any URL. For example, src="/html/top_frame.htm" will load an HTML file available in html directory.
- **name:** This attribute allows you to give a name to a frame. It is used to indicate which frame a document should be loaded into. This is especially important when you want to create links in one frame that load pages into another frame, in which case the second frame needs a name to identify itself as the target of the link.
- **frameborder:** This attribute specifies whether or not the borders of that frame are shown; it overrides the value given in the frameborder attribute on the <frameset> tag if one is given, and this can take values either 1 (yes) or 0 (no).
- **marginwidth :** This attribute allows you to specify the width of the space between the left and right of the frame's borders and the frame's content. The value is given in pixels. For example marginwidth="10".
- **marginheight:** This attribute allows you to specify the height of the space between the top and bottom of the frame's borders and its contents. The value is given in pixels. For example marginheight="10".
- **noresize:** By default you can resize any frame by clicking and dragging on the borders of a frame. The noresize attribute prevents a user from being able to resize the frame. For example noresize="noresize".
- **scrolling:** This attribute controls the appearance of the scrollbars that appear on the frame. This takes values either "yes", "no" or "auto". For example scrolling="no" means it should not have scroll bars.
- **longdesc:** This attribute allows you to provide a link to another page containing a long description of the contents of the frame. For example longdesc="framedescription.htm"

Frame's name and target attributes

One of the most popular uses of frames is to place navigation bars in one frame and then load main pages into a separate frame.

Let's see following example where a test.htm file has following code:

```

<!DOCTYPE html>
<html>
<head>
<title>HTML Target Frames</title>
</head>
<frameset cols="200, *">
<frame src="/html/menu.htm" name="menu_page" />
<frame src="/html/main.htm" name="main_page" />
<noframes>
<body>
Your browser does not support frames.
</body>
</noframes>
</frameset>
</html>

```

Here we have created two columns to fill with two frames. The first frame is 200 pixels wide and will contain the navigation menubar implemented by menu.htm file. The second column fills in remaining space and will contain the main part of the page and it is implemented by main.htm file. For all the three links available in menubar, we have mentioned target frame as main_page, so whenever you click any of the links in menubar, available link will open in main_page. Following is the content of menu.htm file

```

<!DOCTYPE html>
<html>
<body bgcolor="#4a7d49">
<a href="http://www.google.com" target="main_page">Google</a>
<br /><br />
<a href="http://www.microsoft.com" target="main_page">Microsoft</a>
<br /><br />
<a href="http://news.bbc.co.uk" target="main_page">BBC News</a>
</body>
</html>

```

Following is the content of main.htm file:


```
<!DOCTYPE html>
<html>
<body bgcolor="#b5dcb3">
<h3>This is main page and content from any link will be displayed here.</h3>
<p>So now click any link and see the result.</p>
</body>
</html>
```

Now you can try to click links available in the left panel and see the result. The target attribute can also take one of the following values:

Option Description

_self Loads the page into the current frame.

_blank Loads a page into a new browser window.

_parent Loads the page into the parent window, which in the case of a single frameset, is the main browser window.

_top Loads the page into the browser window, replacing any current frames.

targetframe Loads the page into a named targetframe.

HTML IFrames

You can define an inline frame with HTML tag <iframe>. The <iframe> tag is not somehow related to <frameset> tag, instead, it can appear anywhere in your document. The <iframe> tag defines a rectangular region within the document in which the browser can display a separate document, including scrollbars and borders.

The src attribute is used to specify the URL of the document that occupies the inline frame.

Following is the example to show how to use the <iframe>

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Iframes</title>
</head>
<body>
<p>Document content goes here...</p>
<iframe src="/html/menu.htm" width="555" height="200">
Sorry your browser does not support inline frames.
</iframe>
```

```
<p>Document content also go here...</p>
</body>
</html>
```

Most of the attributes of the <iframe> tag, including name, class, frameborder, id, longdesc, marginheight, marginwidth, name, scrolling, style, and title behave exactly like the corresponding attributes for the <frame> tag.

2.2.24. Block Level and Inline Elements

All the HTML elements can be categorized into two categories:

(a) Block Level Elements (b) Inline Elements

Block Elements

Block elements appear on the screen as if they have a line break before and after them. For example the <p>, <h1>, <h2>, <h3>, <h4>, <h5>, <h6>, , , <dl>, <pre>, <hr />, <blockquote>, and <address> elements are all block level elements. They all start on their own new line, and anything that follows them appears on its own new line.

Inline Elements

Inline elements, on the other hand, can appear within sentences and do not have to appear on a new line of their own. The , <i>, <u>, , , <sup>, <sub>, <big>, <small>, , <ins>, , <code>, <cite>, <dfn>, <kbd>, and <var> elements are all inline elements.

Grouping HTML Elements

There are two important tags which we use very frequently to group various other HTML tags (i) <div> tag and (ii) tag The <div> tag .

This is the very important block level tag which plays a big role in grouping various other HTML tags and applying CSS on group of elements. Even now <div> tag can be used to create webpage layout where we define different parts (Left, Right, Top etc.) of the page. This tag does not provide any visual change on the block but this has more meaning when it is used with CSS.

Following is a simple example of <div> tag. We will learn Cascading Style Sheet (CSS) in a separate chapter but we used it here to show the usage of <div> tag:

```
<!DOCTYPE html>
<html>
<head>
<title>HTML div Tag</title>
100HTML Tutorial
</head>
<body>
```

```
<!-- First group of tags -->
<div style="color:red">
<h4>This is first group</h4>
<p>Following is a list of vegetables</p>
<ul>
<li>Beetroot</li>
<li>Ginger</li>
<li>Potato</li>
<li>Radish</li>
</ul>
</div>
<!-- Second group of tags -->
<div style="color:green">
<h4>This is second group</h4>
<p>Following is a list of fruits</p>
<ul>
<li>Apple</li>
<li>Banana</li>
<li>Mango</li>
<li>Strawberry</li>
</ul>
</div>
</body>
</html>
```

The HTML `` is an inline element and it can be used to group inline- elements in an HTML document. This tag also does not provide any visual change on the block but has more meaning when it is used with CSS.

The difference between the `` tag and the `<div>` tag is that the `` tag is used with inline elements whereas the `<div>` tag is used with block-level elements.

Following is a simple example of `` tag. We will learn Cascading Style Sheet (CSS) in a separate chapter but we used it here to show the usage of `` tag:

```
<!DOCTYPE html>
```

```
<html>
<head>
<title>HTML span Tag</title>
</head>
<body>
<p>This is <span style="color:red">red</span> and this is <span
style="color:green">green</
span></p>
</body>
</html>
```

2.2.25. Forms Tags

HTML Forms are required when you want to collect some data from the site visitor. For example, during user registration you would like to collect information such as name, email address, credit card, etc.

A form will take input from the site visitor and then will post it to a back-end application such as CGI, ASP Script or PHP script etc. The back-end application will perform required processing on the passed data based on defined business logic inside the application.

There are various form elements available like text fields, textarea fields, drop- down menus, radio buttons, checkboxes, etc.

The HTML <form> tag is used to create an HTML form and it has following syntax:

```
<form action="Script URL" method="GET|POST">
form elements like input, textarea etc.
</form>
```

Form Attributes

Apart from common attributes, following is a list of the most frequently used form attributes:

Attribute Description

- **action** Backend script ready to process your passed data.
- **method** Method to be used to upload data. The most frequently used are GET and POST methods.
- **target** Specify the target window or frame where the result of the script will be displayed. It takes values like _blank, _self, _parent etc.

HTML Form Controls

There are different types of form controls that you can use to collect data using HTML form:

- Text Input Controls
- Checkboxes Controls
- Radio Box Controls
- Select Box Controls
- File Select boxes
- Hidden Controls
- Clickable Buttons
- Submit and Reset Button
- Text Input Controls

There are three types of text input used on forms:

- **Single-line text input controls** - This control is used for items that require only one line of user input, such as search boxes or names. They are created using HTML `<input>` tag.
- **Password input controls** - This is also a single-line text input but it masks the character as soon as a user enters it. They are also created using HTML `<input>` tag.
- **Multi-line text input controls** - This is used when the user is required to give details that may be longer than a single sentence.

Multi-line input controls are created using HTML `<textarea>` tag.

Single-line text input controls

This control is used for items that require only one line of user input, such as search boxes or names. They are created using HTML `<input>` tag.

Here is a basic example of a single-line text input used to take first name and last name:

```
<!DOCTYPE html>
<html>
<head>
<title>Text Input Control</title>
</head>
<body>
<form >
First name: <input type="text" name="first_name" />
```

```
<br>
Last name: <input type="text" name="last_name" />
</form>
</body>
</html>
```

Following is the list of attributes for <input> tag for creating text field.

Attribute Description

- **type** Indicates the type of input control and for text input control it will be set to text.
- **name** Used to give a name to the control which is sent to the server to be recognized and get the value.
- **value** This can be used to provide an initial value inside the control.
- **size** Allows to specify the width of the text-input control in terms of characters.
- **maxlength** Allows to specify the maximum number of characters a user can enter into the text box.

Password input controls

This is also a single-line text input but it masks the character as soon as a user enters it. They are also created using HTML <input> tag but type attribute is set to password.

Here is a basic example of a single-line password input used to take user password:

```
<!DOCTYPE html>
<html>
<head>
<title>Password Input Control</title>
</head>
<body>
<form >
User ID : <input type="text" name="user_id" />
<br>
Password: <input type="password" name="password" />
</form>
</body>
</html>
```

Following is the list of attributes for <input> tag for creating password field.

Attribute Description

- **type** Indicates the type of input control and for password input control it will be set to password.
- **name** Used to give a name to the control which is sent to the server to be recognized and get the value.
- **value** This can be used to provide an initial value inside the control.
- **size** Allows to specify the width of the text-input control in terms of characters.
- **maxlength** Allows to specify the maximum number of characters a user can enter into the text box.

Multiple-Line Text Input Controls

This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML <textarea> tag.

Here is a basic example of a multi-line text input used to take item description:

```
<!DOCTYPE html>
<html>
<head>
<title>Multiple-Line Input Control</title>
</head>
<body>
<form>
Description : <br />
<textarea rows="5" cols="50" name="description">
Enter description here...
</textarea>
</form>
</body>
</html>
```

Following is the list of attributes for <textarea> tag.

Attribute Description

- **name** Used to give a name to the control which is sent to the server to be recognized and get the value.

- **rows** Indicates the number of rows of text area box.
- **cols** Indicates the number of columns of text area box

Checkbox Control

Checkboxes are used when more than one option is required to be selected. They are also created using HTML <input> tag but type attribute is set to checkbox.

Here is an example HTML code for a form with two checkboxes:

```
<!DOCTYPE html>
<html>
<head>
<title>Checkbox Control</title>
</head>
<body>
<form>
<input type="checkbox" name="maths" value="on"> Maths
<input type="checkbox" name="physics" value="on"> Physics
</form>
</body>
</html>
```

Following is the list of attributes for checkbox.

Attribute Description

- **type** Indicates the type of input control and for checkbox input control it will be set to checkbox.
- **name** Used to give a name to the control which is sent to the server to be recognized and get the value.
- **value** The value that will be used if the checkbox is selected.
- **checked** Set to checked if you want to select it by default.

Radio Button Control

Radio buttons are used when out of many options, just one option is required to be selected. They are also created using HTML <input> tag but type attribute is set to radio.

Here is example HTML code for a form with two radio buttons:

```
<!DOCTYPE html>
<html>
```



```

<head>
<title>Radio Box Control</title>
</head>
<body>
<form>
<input type="radio" name="subject" value="maths"> Maths
<input type="radio" name="subject" value="physics"> Physics
</form>
</body>
</html>

```

Following is the list of attributes for radio button.

Attribute Description

- **type** Indicates the type of input control and for checkbox input control it will be set to radio.
- **name** Used to give a name to the control which is sent to the server to be recognized and get the value.
- **value** The value that will be used if the radio box is selected.
- **checked** Set to checked if you want to select it by default.

Select Box Control

A select box, also called drop down box which provides option to list down various options in the form of drop down list, from where a user can select one or more options.

Here is example HTML code for a form with one drop down box.

```

<!DOCTYPE html>
<html>
<head>
<title>Select Box Control</title>
</head>
<body>
<form>
<select name="dropdown">
<option value="Maths" selected>Maths</option>
<option value="Physics">Physics</option>

```

```
</select>
</form>
</body>
</html>
```

Following is the list of important attributes of <select> tag:

Attribute Description

- **name** Used to give a name to the control which is sent to the server to be recognized and get the value.
- **size** This can be used to present a scrolling list box.
- **multiple** If set to "multiple" then allows a user to select multiple items from the menu.

Following is the list of important attributes of <option> tag:

Attribute Description

- **value** The value that will be used if an option in the select box box is selected.
- **selected** Specifies that this option should be the initially selected value when the page loads.
- **label** An alternative way of labeling options

File Upload Box

If you want to allow a user to upload a file to your web site, you will need to use a file upload box, also known as a file select box. This is also created using the <input> element but type attribute is set to

- **submit** This creates a button that automatically submits a form.
- **reset** This creates a button that automatically resets form controls to their initial values.
- **button** This creates a button that is used to trigger a client-side script when the user clicks that button.
- **image** This creates a clickable button but we can use an image as background of the button.

Here is example HTML code for a form with three types of buttons:

```
<!DOCTYPE html>
<html>
<head>
<title>File Upload Box</title>
```

```
</head>
<body>
<form>
<input type="submit" name="submit" value="Submit" />
<input type="reset" name="reset" value="Reset" />
<input type="button" name="ok" value="OK" />
<input type="image" name="imagebutton" src="/html/images/logo.png" />
</form>
</body>
</html>
```

Hidden form controls are used to hide data inside the page which later on can be pushed to the server. This control hides inside the code and does not appear on the actual page. For example, following hidden form is being used to keep current page number. When a user will click next page then the value of hidden control will be sent to the web server and there it will decide which page has to be displayed next based on the passed current page.

2.3. XHTML

XHTML stands for e**X**tensible **H**yper **T**ext **M**arkup **L**anguage . It is almost identical to HTML . It is stricter than HTML . It is HTML defined as an XML application. It is supported by all major browsers .

Many pages on the internet contain "bad" HTML. The following HTML code works fine in most browsers (even if it does NOT follow the HTML rules):

```
<html>
<head>
<title>This is bad HTML</title>
<body>
<h1>Bad HTML
<p>This is a paragraph
</body>
```

Today's market consists of different browser technologies. Some browsers run on computers, and some browsers run on mobile phones or other small devices. Smaller devices often lack the resources or power to interpret "bad" markup.

XML is a markup language where documents must be marked up correctly (be "well-formed").

The Most Important Differences from HTML:

Document Structure :

- XHTML DOCTYPE is mandatory
- The xmlns attribute in <html> is mandatory
- <html>, <head>, <title>, and <body> are mandatory

XHTML Elements :

- 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

XHTML Attributes :

- Attribute names must be in lower case
- Attribute values must be quoted
- Attribute minimization is forbidden

How to Convert from HTML to XHTML :

- Add an XHTML <!DOCTYPE> to the first line of every page
- Add an xmlns attribute to the html element of every page
- Change all element names to lowercase
- Close all empty elements
- Change all attribute names to lowercase
- Quote all attribute values