HTML Basics and Tags

Fundamentals of Web Engineering

Prepared by

Prof. Muhammad SAAD

Software Engineer (PUCIT)

HTML BASICS

HTML and XHTML are the foundation of all web development. HTML is used as the graphical user interface in client-side programs written in JavaScript. Server-side languages like PHP and Java also receive data from web pages and use HTML as the output mechanism. The emerging Ajax technologies likewise use HTML and XHTML as their visual engine. HTML was once a very loosely-defined language with very little standardization, but as it has become more important, the need for standards has become more apparent. Regardless of whether you choose to write HTML or XHTML, understanding the current standards will help you provide a solid foundation that will simplify all your other web coding. Fortunately HTML and XHTML are actually simpler than they used to be, because much of the functionality has moved to CSS.

Common Elements

Every page (HTML or XHTML shares certain elements in common.) All are essentially plain text files, with the .html extension. HTML files should not be created with a word processor, but in some type of editor that creates plain text. Every page has a large container (HTML or XHTML) and two major subcontainers, the head and the body. The head area contains information useful behind the scenes, such as CSS formatting instructions and JavaScript code. The body contains the part of the page that is visible to the user.

Tags and Attributes

An HTML document is based on the notion of tags. A tag is a piece of text inside angle brackets (<>). Tags typically have a beginning and an end, and usually contain some sort of text inside them. For example, a paragraph is normally denoted like this:

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

Tags and Attributes

The <p> indicates the beginning of a paragraph. Text is then placed inside the tag, and the end of the paragraph is denoted by an end tag, which is similar to the start tag but with a slash (</p>.) It is common to indent content in a multi-line tag, but it is also legal to place tags on the same line:

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

Tags and Attributes

Tags are sometimes enhanced by attributes, which are name value pairs that modify the tag. For example, the <img> tag (used to embed an image into a page) usually includes the following attributes:

```
<img src = "logo.jpg" Alt = "this is logo" />
```

Tags and Attributes

Tags are sometimes enhanced by attributes, which are name value pairs that modify the tag. For example, the <img> tag (used to embed an image into a page) usually includes the following attributes:

```
<img src = "logo.jpg" Alt = "this is a logo" />
```

The src attribute describes where the image file can be found, and the alt attribute describes alternate text that is displayed if the image is unavailable.

Nested tags

Tags can be (and frequently are) nested inside each other. Tags cannot overlap, so <a><b></a></b> is not legal, but <a><b></b></a> is fine.

HTML VS XHTML

HTML has been around for some time. While it has done its job admirably, that job has expanded far more than anybody expected. Early HTML had very limited layout support. Browser manufacturers added many competing standards and web developers came up with clever workarounds, but the result is a lack of standards and frustration for web developers. The latest web standards (XHTML and the emerging HTML 5.0 standard) go back to the original purpose of HTML: to describe the structure of the data only, and leave all formatting to CSS (Please see the DZone CSS Refcard Series). XHTML is nothing more than HTML code conforming to the stricter standards of XML. The same style guidelines are appropriate whether you write in HTML or XHTML (but they tend to be enforced in XHTML):

- Use a doctype to describe the language (described below)
- Write all code in lowercase letters
- Encase all attribute values in double quotes
- Each tag must have an end specified. This is normally done with an ending tag, but a special case allows for non-content tags.

HTML VS XHTML

Most of the requirements of XHTML turn out to be good practice whether you write HTML or XHTML. I recommend using XHTML strict so you can validate your code and know it follows the strictest standards.

XHTML has a number of flavors. The strict type is recommended, as it is the most up-to-date standard which will produce the most predictable results. You can also use a transitional type (which allows deprecated HTML tags) and a frameset type, which allows you to add frames. For most applications, the strict type is preferred.

HTML Template

The following code can be copied and pasted to form the foundation of a basic web page:

```
<html>
    <head>
        <title></title>
    </head>
    <body>
    </body>
</html>
```

PAGE STRUCTURE ELEMENTS

The following elements are part of every web page.

Element	*Description*
<html></html>	Surrounds the entire page
<head></head>	Contains header information (metadata, CSS styles, JavaScript code)
title></title>	Holds the page title normally displayed in the title bar and used in search results
<body></body>	Contains the main body text. All parts of the page normally visible are in the body

KEY STRUCTURAL ELEMENTS

Most pages contain the following key structural elements:

Element	*Name*	Description
<h1> </h1>	Heading 1	Reserved fo strongest emphasis
<h2> </h2>	Heading 2	Secondary level heading. Headings go down to level 6, but <h1> through <h3> are most common
<p> </p>	Paragraph	Most of the body of a page should be enclosed in paragraphs
<div> </div>	Division	Similar to a paragraph, but normally marks a section of a page. Divs usually contain paragraphs

LISTS AND DATA

Web pages frequently incorporate structured data so HTML includes several useful list and table tag

Element	*Name*	*Description*
<ul></ul>	Unordered list	Normally these lists feature bullets (but that can be changed with CSS)
<ol></ol>	Ordered list	These usually are numbered, but this can be changed with CSS
<li></li>	List item	Used to describe a list item in an unordered list or an ordered list
<dl></dl>	Definition list	Used for lists with name-value pairs
<dt></dt>	Definition term	The name in a name-value pair. Used in definition lists
<dd></dd>	Definition description	The value (or definition) of a name, value pair
<table></table>	Table	Defines beginning and end of a table
<tr></tr>	Table row	Defines a table row. A table normally consists of several <tr> pairs (one per row)
<td></td>	Table data	Indicates data in a table cell. <td> tags occur within <tr> (which occur within <table>)
<th></th>	Table heading	Indicates a table cell to be treated as a heading with special formatting

Standard List Types

HTML supports three primary list types. Ordered lists and unordered lists are the primary list types. By default, ordered lists use numeric identifiers, and unordered lists use bullets.

However, you can use the list-style-type CSS attribute to change the list marker to one of several types.

```
<ol>
      <li>one</li>
      <li>two</li>
      <li>three</li>
</ol>
```

Standard List Types

Lists can be nested inside each other

```
<ul>
    <li>English
        <ol>
            <li>One
            <li>Two
            <li>Three
        </ol>
    </li>
    <li>French
        <ol>
            <li>un
            <li>duex
            <li>trios
        </ol>
    </li>
</ul>
```

Definition lists

The special definition list is used for name / value pairs. The definition term (dt) is a word or phrase that is used as the list marker, and the definition data is normally a paragraph:

```
<h2>Types of list</h2>
<dl>
    <dt>Unordered list</dt>
    <dd>Normally used for bulleted lists, where the order of data is
    not important.</dd>
    <dt>Ordered lists</dt>
    <dd>Normally use numbered items, for example a list of
    instructions where the order is significant.</dd>
    <dt>Definition list</dt>
    <dd>Used to describe a term and definition. Often a good
    alternative to a two-column table</dd>
</dl>
```

Use of tables

Tables were used in the past to overcome the page-layout shortcomings of HTML. That use is now deprecated in favor of CSS-based layout. Use tables only as they were intended, to display tabular data.

A table mainly consists of a series of table rows (tr.) Each table row consists of a number of table data (td) elements. The table heading (th) element can be used to indicate a table cell should be marked as a heading.

The rowspan and colspan attributes can be used to make a cell span more than one row or column.

Each row of a table should have the same number of columns, and each column should have the same number of rows. Use of the span attribute may require adjustment to other rows or columns.

Use of tables

```
<table border = "1">
        <tr>
                <th>English</th>
                <th>French</th>
        </tr>
        <tr>
                <td>One</td>
                <td>Un</td>
        </tr>
        <tr>
                <td>Two</td>
                <td>Duex</td>
        </tr>
</table>
```

LINKS AND IMAGES

Links and images are both used to incorporate external resources into a page. Both are reliant on URIs (Universal Resource Indicators), commonly referred to as URLs or addresses.

<a> (anchor)

The anchor tag is used to provide the basic web link:

```
<a href = "http://www.example.com">link to example.com</a>
```

In this example, ***http://www.example.com*** is the site to be visited. The text "link to example.com" will be highlighted as a link.

absolute and relative references

<link>

The link tag is used primarily to pull in external CSS files:

```
<link rel = "stylesheet" type = "text/css" href = "style.css" />
```

absolute and relative references

<img>

The img tag is used in to attach an image. Valid formats are .jpg, .png, and .gif. An image should always be accompanied by an alt attribute describing the contents of the image.

```
<img src=”../images/logo.png” alt = "me before shaving" />
```

Image formatting attributes (height, width, and align) are deprecated in favor of CSS.

SPECIALTY MARKUP

HTML / XHTML includes several specialty tags. These are used to describe special purpose text. They have default styling, but of course the styles can be modified with CSS.

<quote>

The quote tag is intended to display a single line quote:

```
<quote>The weak can never forgive. Forgiveness is the attribute of the strong.</quote>
```

Quote is an inline tag. If you need a block level quote, use <blockquote>.

FORMS

The <form></form> pair describes the form. In XHTML strict, you must indicate the form's action property. This is typically the server-side program that will read the form. If there is no such program, you can set the action to null ("") The method attribute is used to determine whether the data is sent through the get or post mechanism.

Input

The input element is a general purpose inline element. It is meant to be used inside a form, and it is the basis for several types of more specific input. The subtype is indicated by the type attribute. Input elements usually include an id attribute (used for CSS and JavaScript identification) and / or a name attribute (used in server-side programming.) The same element can have both a name and an id.

Text

This element allows a single line of text input:

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

Password

Passwords display just like textboxes, except rather than showing the text as it is typed, an asterisk appears for each letter. Note that the data is not encoded in any meaningful way. Typing text into a password field is still entirely unsecure.

```
<input type = "password" id = "MyPasswd" />
```

Radio Button

Radio buttons are used in a group. Only one element of a radio group can be selected at a time. Give all members of a radio group the same name value to indicate they are part of a group.

```
< type = "radio" name = "radSize"
vainputlue = "small" id = "radSmall" selected = "selected" />
<label for = "radSmall">Small</label>
<input type = "radio" name = "radSize"
value = "large" id = "radLarge" />
<label for = "radLarge">Large</label>
```

Radio Button

Radio buttons are used in a group. Only one element of a radio group can be selected at a time. Give all members of a radio group the same name value to indicate they are part of a group.

```
< type = "radio" name = "radSize"
vainputlue = "small" id = "radSmall" selected = "selected" />
<label for = "radSmall">Small</label>
<input type = "radio" name = "radSize"
value = "large" id = "radLarge" />
<label for = "radLarge">Large</label>
```

Attaching a label to a radio button means the user can activate the button by clicking on the corresponding label. For best results, use the selected attribute to force one radio button to be the default.

Checkbox

Checkboxes are much like radio buttons, but they are independent. Like radio buttons, they can be associated with a label.

```
<input type = "checkbox" id = "chkFries" />
<label for = "chkFries">Would you like fries with that?</label>
```

Hidden

Hidden fields hold data that is not visible to the user (although it is still visible in the code) It is primarily used to preserve state in server-side programs.

```
<input type = "hidden"
      name = "txtHidden"
      value = "recipe for secret sauce" />
```

Note that the data is still not protected in any meaningful way.

Button

Buttons are used to signal user input. Buttons can be created through the input tag:

```
<input type = "button" value = "Submit" onclick = "MyFunction()" />
```

Button

This will create a button with the caption "launch the missiles." When the button is clicked, the page will attempt to run a JavaScript function called "launchMissiles()" Standard buttons are usually used with JavaScript code on the client. The same button can also be created with this alternate format:

```
<button type = "button" Onclick = "MyFunction()">
     Submit
</button>
```

This second form is preferred because buttons often require different CSS styles than other input elements. This second form also allows an <img> tag to be placed inside the button, making the image act as the button.

Reset

The reset button automatically resets all elements in its form to their default values. It doesn't require any other attributes.

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

Select / option

Drop-down lists can be created through the select / option mechanism. The select tag creates the overall structure, which is populated by option elements.

```
<select id = "colors">
      <option value = "#000000">black</option>
      <option value = "#FF0000">red</option>
      <option value = "#FFFFFF">white</option>
</select>
```

The select has an id (for client-side code) or name (for serverside code) identifier. It contains a number of options. Each option has a value which will be returned to the program. The text between <option> and </option> is the value displayed to the user. In some cases (as in this example) the value displayed to the user is not the same as the value used by programs.

Multiple Selections

You can also create a multi-line selection with the select and option tags:

```
<select id = "colors" size = "3" multiple = "multiple">
      <option value = "#000000">black</option>
      <option value = "#FF0000">red</option>
      <option value = "#FFFFFF">white</option>
</select>
```

DEPRECATED FORMATTING TAGS

Certain tags common in older forms of HTML are no longer recommended as CSS provides much better alternatives.

Font

The font tag was used to set font color, family (typeface) and size. Numerous CSS attributes replace this capability with much more flexible alternatives. See the CSS refcard for details.

DEPRECATED FORMATTING TAGS

I (italics)

HTML code should indicate the level of emphasis rather than the particular stylistic implications. Italicizing should be done through CSS. The <em> tag represents emphasized text. It produces italic output unless the style is changed to something else. The <i> tag is no longer necessary and is not recommended. Add font-style: italic to the style of any element that should be italicized.

DEPRECATED FORMATTING TAGS

B (bold)

Like italics, boldfacing is considered a style consideration. Use the <strong> tag to denote any text that should be strongly emphasized. By default, this will result in boldfacing the enclosed text. You can add bold emphasis to any style with the font-weight: bold attribute in CSS.