WEB DESIGN - HYPERTEXT MARKUP LANGUAGE (HTML) FOR COMPUTERS

Introduction and Brief History of HTML

More and more of the world's software is being run within a web browser. Web software offers many legitimate benefits: ease of deployment, ubiquity of access to a global audience, and availability of serverside data and services.

HTML stands for the Hypertext Markup Language. It was defined by Tim Berners-Lee in 1990 as the method of marking up pages of information to be looked at by a browser. A markup language is a set of commands that tells a computer how to format your document. HTML tags tell a browser such as Netscape, Mozilla Firefox, or Internet Explorer how to structure your Web page. HTML only provides information about presentation, not meaning. Since 1990s, it has blossomed with the explosive growth of the web.

HTML is not a true programming language because it's just a mark up language that is not based on syntax of programming languages. It just uses tags, no variables, no functions, no control structures, no objects etc which are typical of any programming languages. It must incorporate scripting language in order to develop a function in the case of interactive and dynamic websites.

With HTML you can create your own Web site. The purpose of a web browser (like Internet Explorer or Firefox) is to read HTML documents and display them as web pages. The browser does not display the

HTML tags, but uses the tags to interpret the content of the page. The extension of a HTML document is .html or .htm. For you to practice on the tags in this document, you will need a browser and any text editor (e.g notepad) or any HTML authoring tool. A WWW browser interprets the HTML code and displays it.

HTML Tags

HTML tags consist of a left angle bracket (<), a tag name, and a right angle bracket (>). HTML tags are keywords surrounded by angle brackets e.g <html>. Tags are always enclosed in angle brackets. Here is an example of a tag:

<html>

Tags are usually paired (e.g <H1> and </H1>) to start and end the tag instruction. The end tag looks like the start tag except a slash (/) precedes the text within the brackets.

HTML Elements

An element comprises of three parts: a start tag, content, and an end tag. An element is a fundamental component of the structure of an HTML document. An HTML element starts with a start tag/opening tag and ends with an end tag/closing tag. Some examples of elements are headings, tables, paragraphs, lists etc. Think of it this way: you use HTML tags to mark the elements of a file for your browser. Most HTML elements can have attributes. The element content is everything between the start and the end tag. e.g.

The element: This is my first paragraph.

HTML Attributes

Some elements may include attributes which is additional information that is included inside the start tag.

Attributes provide additional information about an element. Attributes are always specified in the start tag e.g.

<BODY BGCOLOR=”#FFFF00”>. BGCOLOR is an attribute.

NOTE: HTML tags are not case sensitive. <title> is equivalent to <TITLE>. However, file names on the Web follow Unix conventions. This means they are case sensitive. Be sure to save your file using the complete (.html) extension. Not all tags are supported by World Wide Web browsers. If a browser does not support a tag, it will simply ignore the tag and display any text placed between a pair of unknown tags.

HTML Document Structure

An HTML document consists of two main parts: the Head and the Body. The head contains the title, and the body contains the actual text that is made up of headings, paragraphs, lists, tables and other elements.

Browsers expect specific information because they are programmed according to HTML and SGML specifications. Standard Generalized Markup Language (SGML) is a standard for describing markup languages. The basic structure is:

<HTML>

<Head>

<Title> This is the Title </Title>

</Head>

<Body>

This is the Body

</Body>

</HTML>

HTML MARKUP TAGS

HTML

This element tells your browser that the file contains HTML-coded information. The file extension .html OR .htm also indicates this is an HTML document and must be used.

The directive for the HTML tag is <html>

HEAD

The head element identifies the first part of your HTML-coded document that contains the title. The title is shown as part of your browser’s window. The directive for the head tag is <head>

TITLE

The title element contains your document title and identifies its content in a global context. The title is typically displayed in the title bar at the top of the browser window, but not inside the window itself. A title is also used to identify your page for search engines (such as Google or

Yahoo). It's recommended that you choose a title that is descriptive, unique, and relatively short. The title generally goes on the first line of the document. The directive for the title tag is <title>

BODY

The second and largest part of your HTML document is the body, which contains the content of your document (displayed within the text area of your browser window). The tags explained below are used within the body of your HTML document. The directive for the body tag is <body>

HEADINGS

HTML has six levels of headings (numbered 1 through 6), with 1 being the most prominent. Headings are displayed in larger and/or bolder fonts than the normal body text. The first heading in each document should be tagged <H1>.

The syntax of the heading element is: <Hy>Text of heading </Hy> where y is a number between 1 and 6 specifying the level of the heading.

For example:

<H1> this is a level one heading </H1> (Most prominent header)

<H6> this is a level six heading </H6> (Least prominent header)

PARAGRAPHS

Unlike documents in most word processors, carriage returns and white space in HTML files aren't significant. In fact, any amount of whitespace including spaces, carriage returns, and multiple spaces are collapsed into a single space when your HTML document is displayed in a browser. So you don't have to worry about how long your lines of text are.

Notice that in the example shown below, there is a line break between the sentences. A Web browser ignores this line break and starts a new paragraph only when it encounters another <P> tag.

Example 1:

<P> Welcome to the world of HTML.

This is the first paragraph.

While short it is still a paragraph! </P>

Important: You must indicate paragraphs with <P> element and you must end each paragraph with </P>.

The viewer ignores any indentations or blank lines in the source text. Without the <P> tags, the document becomes one large paragraph. HTML relies almost entirely on the tags for formatting instructions. (The exception is text tagged as ``preformatted").

Example 2:

<P> Welcome to the world of HTML. This is the first paragraph.

While short it is still a paragraph! </P>

<P> And this is the second paragraph. </P>

To preserve readability in HTML files, put headings on separate lines, use a blank line or two where it helps identify the start of a new section, and separate paragraphs with blank lines (in addition to the <P> tags).

These extra spaces will help you when you edit your files (but your browser will ignore the extra spaces because it has its own set of rules on spacing that do not depend on the spaces you put in your source file).

NOTE: The </P> closing tag may be omitted. This is because browsers understand that when they encounter a <P>, it means that the previous paragraph has ended. However, since HTML now allows certain attributes to be assigned to the <P> tag, it's generally a good idea to include it.

Paragraph Alignment

Using the <P> and </P> as a paragraph container means that you can center a paragraph by including the:

- ALIGN=CENTER attribute in your source file.

Example: <P ALIGN = CENTER> This is a centered paragraph </P>

It is also possible to align a paragraph to the right instead, by including the ALIGN=RIGHT attribute.

- ALIGN=LEFT is the default alignment; if no ALIGN attribute is included, the paragraph will be left-aligned.

FORCED LINE BREAKS

The
 tag forces a line break with no extra (white) space between the lines. Using <P> for short lines of text such as postal addresses results in unwanted additional white space. The
 tag is not used in pairs.

For example

Moi University Main Campus,

P.O. Box 3900,

ELDORET, KENYA

The output is:

Moi University Main Campus,

P.O. Box 3900,

ELDORET, KENYA

LISTS

HTML supports unordered (bulleted), ordered (numbered) and descriptive lists. For list items, no paragraph separator is required. The tags for the items in the list terminate each list item.

1. Unordered (Bulleted) Lists

A bulleted list (also called an unordered list, from where the abbreviation comes) uses the <ul> directive to start a list.

1. Start with an opening list <ul> tag.

2. Enter the <li> tag followed by the individual item. (Remember that no closing tag is needed.)

3. End with a closing list tag.

Below an example two-item list:

<UL>

<LI> apples

<LI> bananas

</UL>

The output is:

· apples

· bananas

Note that different viewers display an unordered list differently. A viewer might use bullets, filled circles, or dashes to show the items.

2. Ordered (Numbered) Lists

A numbered list (also called an ordered list, from where the abbreviation comes) uses the <ol> directive to

start a list. The items are tagged using the same <li> tag as for a bulleted list. For example:

<OL>

<LI> oranges

<LI> peaches

<LI> grapes

</OL>

The output list looks like this online:

1. oranges
2. peaches
3. grapes

3. Nested Lists

Lists can be nested. A list item can itself contain lists. You can also have a number of paragraphs, each containing nested lists, in a single list item, and so on. When you are coding your web page, keep in mind that HTML elements cannot overlap each other.

Remember that the display of an unordered list varies with the viewer. A browser may not provide successive levels of indentation or modify the bullets used at each level.

Here is an example of a nested list:

<UL>

<LI> A few New England states:

<UL>

<LI> Vermont

<LI> New Hampshire

</UL>

<LI> One Midwestern state:

<UL>

<LI> Michigan

</UL>

</UL>

The nested list is displayed as

· A few New England states:

o Vermont

o New Hampshire

· One Midwestern state:

o Michigan

CHARACTER FORMATTING

Individual words or sentences can be put in special styles.

· Italic

o <I> Italicized text </I> puts text in italics

· Bold

o <B> Bolded text </B> puts text in bold

o <STRONG> Bolded text </STRONG> also emphasizes text like Bold <B>.

The FONT Tag

The <FONT> … </FONT> tag enables you to change the size, face and colour of text.

The SIZE attribute

The SIZE attribute used inside the <FONT> tag will enable you to change the size of text in relation to the size that the person has set their browser to.

For example: <FONT SIZE=2> This text is size 2 </FONT>

The COLOR attribute

The COLOR attribute specifies the color of the text and will override any colour specified within the BODY tag. Try not to overuse the COLOR attribute as it can confuse.

Example: <FONT COLOR=”blue”> This text is blue in color </FONT>

Note: You can also use hexadecimal values.

For Example: <FONT COLOR=”#0000FF”> This text is blue in color </FONT>

Colours are given as hexadecimal numbers (e.g COLOR=”#0000FF”), or as one of 16 widely understood color names. These colors were originally picked as being the standard 16 colors supported with Windows VGA palette.

Black=”#000000”

Green=”#008000”

Silver=”#C0C0C0”

Lime=”#00FF00”

Gray=”#808080”

Olive=”#808000”

White=”#FFFFFF”

Yellow=”#FFFF00”

Maroon=”#800000”

Navy=”#000080”

Red=”#FF0000”

Blue=”#0000FF”

Purple=”#800080”

Teal=”#008080”

Fuchsia=”#FF00FF”

Aqua=”#00FFFF”

However, there is a much wider variety of colours available to you if you use the hexadecimal colour code rather than words.

The FACE attribute

This attribute specifies the type of font for the text and will override any browser settings. If your computer does not support the type of font specified, it will default to the browser settings.

Example 1: <FONT FACE=”Helvetica”> This font is Helvetica </FONT>

Example 2: <FONT size = 2 face = "verdana, arial, helevitica"> This is a mixed font </FONT>

DOCUMENT BACKGROUND AND FOREGROUND COLOURS

By default, browsers display text in black on a white background. However, you can change both elements

if you want. Always preview changes to make sure your pages are readable (for example, many people find

red text on a black background difficult to read).

Controlling the document background: The BGCOLOR attribute

You can use BGCOLOR attribute to specify the background colour of your Web document. This is done within the BODY tag.

Example:

<BODY BGCOLOR=”#FFFFF00”>

…….

Your content goes here

…….

</BODY>

Where FFFFF00 is hexadecimal number used to specify the colour.

Controlling the document foreground: The TEXT attribute

Once you have control of the background, you will also need to control the foreground colour using the

TEXT attribute. This involves controlling the colour of the text and links. The TEXT attribute specifies the

colour of all text apart from links.

Example:

<BODY TEXT=”#008000”>

…….

Your content goes here

…….

</BODY>

You need to be very careful with the choice of colours so that they do not ‘clash’ with each other.

CENTERING

Can be accomplished with the <CENTER> ..... </CENTER> tags.

Example: <CENTER> This text is centered </CENTER>

LINKING TO OTHER DOCUMENTS

The chief power of HTML comes from its ability to link text and/or an image to another document or section of a document. The tag that creates those links is called the anchor tag (A). It has one commonly used attribute: HREF, which specifies the URL of the target document. HREF stands for Hypertext REFerence and is used to link to:

· Another document (external link); or

· Another part of the same document (internal link)

The hyperlinks are typically highlighted with color and/or underlines by the browser to indicate that they are hypertext links (often shortened to hyperlink or just link).

To include anchors in your document:

1. Start the anchor with <A (include a space after the A).

2. Specify the document that's being pointed to by giving the parameter HREF="filename" followed by a closing angle bracket: (>)

3. Enter the text that will serve as the hypertext link in the current document (i.e., the text that will be in a different color and/or underlined)

4. Enter the ending anchor tag: </A> (no space is needed before the end anchor tag)

Here is a sample hypertext reference: <A HREF="informatics.html"> BSc in Informatics </A>

This entry makes `` BSc in Informatics " the hyperlink to the document informatics.html, which is in the same directory as the first document.

Example 2: <A HREF="applicationform.pdf"> Click here to download the Application form </A>

You can link to documents in other directories by specifying the relative path from the current document to the linked document. For example, a link to a file deptit.html located in the subdirectory sis would be:

<A HREF="sis/deptit.html"> Dept of Information Technology </A>

These are called relative links because you are specifying the path to the linked file relative to the location of the current file.

Anchors to Specific Sections in Other Documents

Anchors can also be used to move to a particular section in a document. Suppose you wish to set a link from document A and a specific section in document B. First you need to set up what is called a named anchor in document B. For example, to add an anchor named ``Jabberwocky" to document B, you would insert

Here's <A NAME="Jabberwocky"> some text in B </A>.

Now when you create the link in document A, you include not only the filename, but also the named anchor, separated by a hash mark (``#"):

This is my <A HREF="documentB.html#Jabberwocky"> link to a section in B </A>.

Now clicking on the word `` link to a section in B " in document A would send the reader directly to the words `` some text in B “ in document B.

Anchors to Specific Sections within the Current Document

The technique is exactly the same as above except the file name is now omitted.

External Links

To link to an external document, you need to know the URL of the document. You also need to decide what text you will use for the link. When creating links try to make the hypertext a natural part of the sentence. In other words, minimize the use of the “click here” syndrome when creating links.

The code for creating a hypertext link to an external document is:- <A HREF="URL"> Text for link </A>

For Example: <A HREF="http://www.google.co.ke"> Google </A> will link to Google search engine.

The LINK, VLINK, and ALINK attributes

These attributes allow you to control the colour of the hyperlinks. The default colouring that you are used to for these links are: LINK=blue, VLINK=purple, and ALINK=red

- LINK – sets the colour for a link before it is visited
- VLINK – sets the colour for a visited link
- ALINK – sets the color for the link when you click on it (active link)

Example

<BODY LINK=”#0000FF” VLINK=”#008000” ALINK=”FF0000#”>

…….

Your content goes here

…….

</BODY>

INLINE IMAGES

Images have made a profound difference in the way the web looks. Probably, there would not have been the incredible explosion of interest in it if inline images had not been added. Most Web browsers can display inline images (that is, images next to text) that are in X Bitmap, GIF, or JPEG format. Other image formats are also being incorporated into Web browsers (e.g the Portable Network Graphic – PNG format).

Each image takes additional time to download and slows down the initial display of a document. Carefully select your images and the number of images in a document.

To include an inline image in your document, enter: <IMG SRC="ImageName">

Where ImageName is the URL of the image file. Make certain the image is in GIF, TIFF, JPEG, RGB, or HDFformat.

Example: <IMG SRC="logo.gif">

You can include two other attributes on <IMG> tags to tell your browser the size of the images it is downloading with the text. The HEIGHT and WIDTH attributes lets your browser set aside the appropriate space (in pixels) for the images as it downloads the rest of the file.

Example: <IMG SRC=SelfPortrait.gif HEIGHT=100 WIDTH= 65>

NOTE: Some browsers use the HEIGHT and WIDTH attributes to stretch or shrink an image to fit into the allotted space when the image does not exactly match the attribute numbers.

You have some flexibility when displaying images. You can have images separated from text and aligned to the left or right or centered. You can align images to the top or center of a paragraph using the

ALIGN=attributes TOP and CENTER.

To display an image without any associated text (e.g your organization's logo), make it a separate paragraph. Use the paragraph ALIGN attribute to center the image or adjust it to the right side of the window as shown below:

<P ALIGN=CENTER>

<IMG SRC = "mulogo.gif" ALT="[Logo]">

</P>

This image (mulogo.gif) is centered.

The ALT attribute lets you specify text to be displayed instead of an image.

For example: <IMG SRC = "mulogo.gif" ALT="[Logo]">

Images as Hyperlinks

Inline images can be used as hyperlinks just like plain text.

For example: <A HREF = "muhome.html"> <IMG SRC = "mulogo.gif" ALT="[Logo]"> </A>

The blue border that surrounds the image indicates that it's a clickable hyperlink. You may not always want this border to be displayed, though. In this case, you can use the BORDER attribute of the IMG tag to make the image appear as normal. Adding the BORDER attribute and setting it to zero.

<A HREF = "muhome.html"> <IMG SRC = "mulogo.gif" BORDER=0 ALT="[Logo]"> </A>

TABLES

Tables are very useful for presentation of tabular information as well as a boon to creative HTML authors who use the table tags to present their regular Web pages. The HTML table model allows authors to arrange data (text, images, links, forms, other tables, etc) into rows and columns of cells. A table has rows and column cells for each item.

HTML Table Tags

Tag Description

<table> Defines a table

<th> Defines a table header

<tr> Defines a table row

<td> Defines a table cell

General Table Format

<TABLE>

<TR>

<TD> content of the cell </TD>

</TR>

</TABLE>

The <TABLE> and </TABLE> tags must surround the entire table definition. You can have a number of rows defined by <TR> and </TR>. Within a row, you can have any number of cells defined by the <TD> …. </TD> tags. Every row of a table is essentially, formatted independently of the rows above and below it.

The BORDER attribute is used to add the border to the Table. With a border=0, you can easily create columns that are separated by empty space.

Example 1:

<TABLE BORDER=2>

<TR>

<TD> content of the cell </TD>

</TR>

</TABLE>

Example: A basic 3 X 2 Table

<TABLE BORDER=2>

<TR>

<TD> first row, first cell contents </TD>

<TD> first row, second cell contents </TD>

<TD> first row, third cell contents </TD>

</TR>

<TR>

<TD> second row, first cell contents </TD>

<TD> second row, second cell contents </TD>

<TD> second row, third cell contents </TD>

</TR>

</TABLE>

Commenting Your Files

You might want to include comments in your HTML files. Comments in HTML are like comments in a computer program i.e. the text you enter is not used by the browser in any formatting and is not directly viewable by the reader just as a computer program comments are not used. The comments are accessible if a reader views the source file, however. Comments such as the name of the person updating a file, the software and version used in creating a file, or the date a minor edit was made are the norm.

To include a comment, use this directive: <!- your comments here ->