Web Page Designing: An Introduction

Alka Bansal & **Pushpa Raina**

Abstract

This article is meant for anyone who views a web page and would like to get introduced to the subject of web page designing. It introduces the. subject of web page designing. Steps for developing a web page, website hosting, security policy are described in brief. A brief description of HTML tags used for designing a web page is also given.

1. INTRODUCTION

The type of information on the Internet is as diverse as the people interested in it. People exchange e-mail, participate in discussion groups (known as news groups), socialize (one forum for such activities being called 'chat'), retrieve data files, access all sorts of information in a wide variety of formats, and get engaged in e-commerce.

The question arises that where does all this information come from? The answer is 'everywhere'- because not only can anyone access information on the net, but also publish it. In one sense, Internet can be thought as a vast and growing online library in which anyone can publish anything he wants. .Files on the Internet too need an address for accessing. A file's address is formally known as its Uniform Resource Locator (URL).

Let us take an example of a URL say   http://www.ala.org/events/ev_home.html. An address of a website is commonly enclosed with an arcane list of symbols beginning with the letters 'http://.' Here, 'http' stands for hypertext transfer protocol. It is the content identifier (content-id in short). The content-id tells the browser which protocol or language was used to create the current page. The browsers may support other protocols, such as ftp://, gopher://,   telnet:/ / , etc. The second part of the same address i.e., www.ala.org/events/ev_home.html identifies the location of the particular web page or resource.

To understand the location better, it can be divided into two smaller parts the first part is the domain name or host name. Each computer connected to the Internet has a unique name that makes it easy to identify it from the thousands of other computers connected directly to the web. A PC may not have a domain name but the service provider does because it is connected to the web. (One connect to the web through One's service provider's domain). So the address www.ala.org refers to the web managing portions (WWW) of a computer called ala.org. The org part tells that this computer is basically used for organisation purposes. Other popular extensions include '.com', '.edu' or '.gov', etc., indicating a commercial enterprise, or educational institution, government, etc. The second part of the location is the name of a particular web resource. This name looks very much like a directory path because that's exactly what it is. Here, in our example, the address will connect the viewer to a document called ev_home.html, located in the events directory of the ala computer .

To have an Internet tour, the most important piece of software one needs is called a browser, a program that lets a user visit different sites on the net and display their offerings on the computer.

Most popular browsers are Netscape Navigator and Internet Explorer. One visits a site by supplying the browser with an address or URL. The browser dowloads the document whose URL is given on user's computer screen.

The first page appearing on any site is generally called the home page-which is the primary document of a site giving details about the site and introduces what would follow. From a particular web page one can access other related sites and information by clicking onto links which are conspicuous either by a different colour or type (being underlined, etc.).

2. STEPS FOR DEVELOPING WEB PAGE

To develop a content-rich, easy to read and use web page, one has to follow the following steps:

2.1 Purpose of Web Page

For developing a web page that is easy to read and use, one needs to focus on the purpose of the web page. One has to be very clear about goals and objectives. Maybe the web page is an informational article, or a web page of an organisation where one has to create a menu page so that people can access the available resources quickly. One has to have clear idea regarding who are the target audience of the site.

2.2 Web Page Planning

Planning is a very important part of web page development. One may plan his web pages through two ways:

Internal Planning-regarding the content included in the web page. It includes the following:

- Forming the project team
- What kind of information is to be presented (home page, informational article, etc.)?
- How to present the information (page layout)?
- Size of the page for download purpose
- Does the page engage the visitor?
- Should the page be static or interactive?
- Will the contents be useful enough for visitors to refer again?
- Information about.last updation?
- Future improvements
- Tracking results and responding to visitor feedback
- Revising page.

External planning is regarding customer base and the Internet itself. It includes the following:

- Will the web page be easily navigated?
- How will people visit web page?
- How can we make sure that people stay to explore our site?

- How to popularize the site?

One should note down the resource sites to be incorporated into the web page. The pages offering additional activities like quizzes, puzzles, games, slide shows, and other additional resource links can be added to the web page to engage the visitor.

2.3 **Designing**

2.3.1 **Components of Web Page**

The components that make up a web page are:

- Text
- Pictures
- Animated graphics
- Audio files
- Video files

It is not necessary to include all components on a page. The first priority is to make sure that pages are fast to download easy to read and well-organized. It is not essential to include animated icons to make a page visually interesting; some very nice pages have been created using 'only well-thought-out text and pictures .

2.3.2 **Web Authoring Languages**

- HTML (Hyper text Markup Language)
- DHTML (Dynamic HTML)
- XHTML (eXtensible HTML)
- XML (eXtensible Markup Language)
- VRML (Virtual Reality Modelling Language)
- SMIL (Synchronized Multimedia Integration Language)

2.3.3 **Steps for Creating Web Page**

The general steps one should follow to create a web page are:

1. Use a text editor like word pad or notepad or web page editor/authoring tools like Microsoft's Front Page, Macromedia's Dreamweaver, HoT MetaL, Hot Dog to create the pages for the site and save them as HTML files. These web editors create HTML codes for the text. For example, if one creates a file named 'abc' in word pad, he has to save the file as abc.html. A brief information about use of Front Page in web page publishing is given in later part of the article.

The images can be saved in gif or jpg format. For editing of images any of these tools/programs can be used: Adobe Photoshop, Microsoft's Image Composer, Macromedia's Fireworks, Paint Shop Pro, etc. A clipart can be used, if graphics are not self-created. Programs like Core! Clipart, Front Page Clipart, etc. can be used.

2. Prepare/digitize other individual components, e.g., audio/video/graphics/ani- mation files. Sound or video equipment are needed for creating audio or video files.

- For creating animation, Macromedia's Flash or Macromedia's Shockwave, etc. can be used.
- For including audio/video on web, the following formats can be used:

* Audio file format *.aiff, *.aif,*.au, *.mid, *.wav

* Video file format *.avi, *.mov, *.mpeg, *.mpg

* Streaming format: Allows media to begin playing while it's being downloaded

- Real audio/video

* Quick time
* Microsoft's streaming
* Shockwave

- MP3 format

3. If the page requires more advanced features, such as forms that users can fill in, or a connection to a database of information from which informational pages can be created based on user requests. (A form on a web page is just like a form on paper. Visitors fill out the form to order products, give answers to a survey, or other purposes). Based on the information provided in users' forms, the web page can run CGI scripts, which are the programs that process the information filled in by the visitors.

4. Use a browser to view the page that have been created. Check that the text is spelled correctly, graphics look good and that links among the pages work. Repeat these steps until the page looks presentable enough to publish.

5. Publish the page by putting all of its files (HTML files and graphics files) on a web server.

6. Use a browser, view the web pages as stored on the web server. One should view the page by using at least two web browsers (for example, Netscape Navigator 4.5 and Internet Explorer 5), because different browsers format pages slightly differently.

7. Publicize the page, get feedback, get new ideas and repeat the steps.

2.3.4 **Structure of Web Page**

At the beginning of designing a web page, one should try to organise the information on the site so that it is easy to find and is seen in reasonable context. The first thing one has to know is where people are going to start from. Most sites have a home page similar to front door in a house. Other sites are designed so that people interested in different topics come in via different
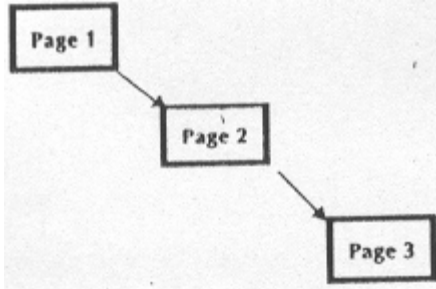
pages, often from links on other pages. The simplest site has just one HTML page.

Controlled Site: If one wants visitors to see in a fixed order then a linear site is built with no optional path. For example, one can have a presentation or a story or article.
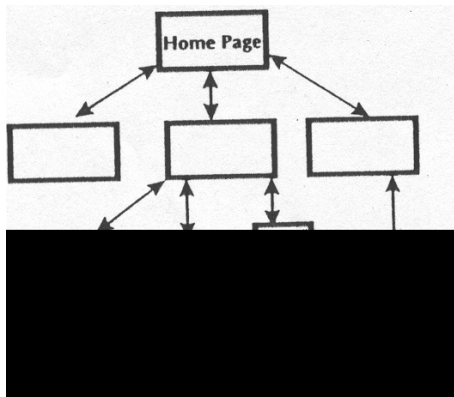
Tree Structure: Here, the home page serves as table of contents. The links on that page take the user to a number of pages, each of which can be a content page or can have a more detailed table of contents with its own page links. Each page should have a link back to the one that it branches from, and possibly a link to the home page.
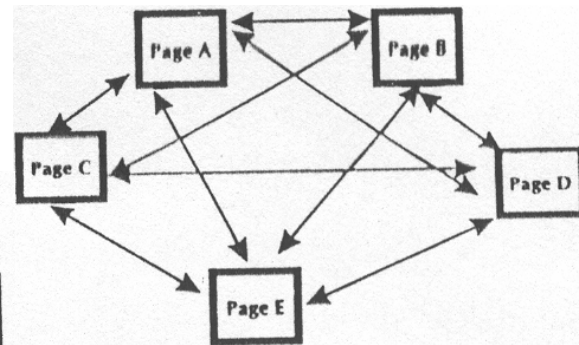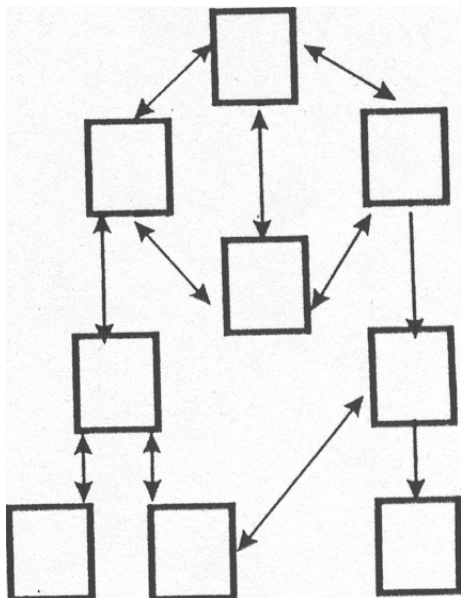
ateral structure, page, usually by 1 (or both) of the quick movement ites the problems or is coming in; l be able to :ype of ike for a site with a large number of pages, the number of links nove a page, one has to change all the pages.

Combining Structures: A site can be formed by combining the above-mentioned structures in a way that makes sense for

the site. The following diagram shows an equilateral set of pages on various topics, some of which serve as the starting page for a tree structure:

Some of these pages contain links that leap to unconnected pages for ease of users. One has to make sure that basic structure must be reasonably clear to the visitor.

2.3.5 Guidelines for Designing

The following guiding principles can be kept in mind while designing a web page:

Content

- Provide useful and valuable information
- Consider the high cost of following each and every link and make sure each page has useful content on it
- Create material targeted specifically at the selected audience
- Add compelling content-something the visitor is not expecting.

Scope

- Start small. It is better to publish a few excellent pages than a large quantity of partially complete pages
- Gradually add material to the page. Do planning for future expansion.

Links

- Do not create a page that only has links
- Be sure to annotate links so that visitors have a clear idea about what's going to follow
- Structure the material so that visitors do not have to follow too many links to get to the desired information
- Include navigational links on every page. At least there should be a link to the home page on every other page.

Appearance Counts

Along with the useful content, it is important for the layout to be attractive. While designing, the following points should be kept in mind:

- *Graphics*
    - If the graphic is larger than a two inch square or a one by three inch rectangle, use a thumbnail image of the picture
    - Use 'thumbnails' (miniature versions of images) if there are many images on a single page. Visitors can click on these images to see the full-sized versior
    - Reduce the colour palette to the lowest possible pallette size to ensure quick downlo~ of the web page
    - Use the same graphics as often as possible. An image that is used repetitively only

needs to be downloaded once. Using a large number of different images will slow the download time of a web page. It may desist the visitors visiting the web page again

- o Break big images from in to parts for simultaneous downloads
- o Don't use more than one animation per page.

Background Colour or Image

- o Add background colour or a background image, but do not use busy backgrounds. Even if the text shows up clearly on the background, it can be difficult for people to read if there is too much text or animation in the background
- o Light backgrounds should have dark text and dark backgrounds should have light text. Be careful in choosing the colours. Not all computers see the same colour hues
- o Make background width 1300-1600 pixels for those with larger screens.

Vary Font Colour & Size

- o Use the font size tags to vary the body text and the header tags to vary the size of the headings and sub-headings and different colours to give good effects.

Credit to Graphic Artist

- o Read the terms artists laid while using an image from their website. Many appreciate a link back to the website while others do not want.

Citing Text Sources

- o Citing Internet sites which have been used to gather information as resource links will lend credibility to work. It also prevents one from misusing other people's work.

Establish Credibility

At the end of each web page, one usually includes the following information:

- o The name of the author of this activity
- o E-mail address of the contact person for this activity
- o When was this web page last updated and links were checked
- o Name of the host or organization.

3. **HTML & WEB PUBLISHING**

The technology of web publishing has evolved fast from HTML to JAVA. HTML is a document description language derived from SGML that consists of text and fixed tags, and is the original and most widely used markup language on the web. Tags describe the attributes of the text and

other content and are used by clients to determine how to display the text or perform other manipulations. In HTML, both the tag set (the choice of tags) and their semantics (what they mean) are fixed. A list of HTMl tags is given as Appendix to the article.

While starting to design a web page, the first thing to realize is that what one creates on the system may not be what is seen by the visitor. The reason may be one of the following or something else:

(a) The user may have a lower resolution screen with fewer colours and poorer graphics;

(b) The user may not be using same browser with its special extensions;

(c) The user may be using a previous version of the browser.

3.1 Designing of Components of Page

3.1.1 *Page text*

Headers can be made using the commands <hl>, <h2>, <h3>, <h4>, <h5>, and <h6>. HTML 2.0 has relatively little control over the fonts in which the document will be displayed, although the text can appear in bold or italics. There are other block formatting and font commands listed below.

3.1.2 *Format blocks in a variety of ways*

Different browsers may handle some of these tags in the same way; i.e., they may not provide as much of a distinction as one wishes. The essential block formatting elements are:

<ADDRESS> ...</ADDRESS>

Format an address section

Force a line break <P> ...</P>

Specify what text constitutes a paragraph

<CENTER> ...</CENTER>

Centers the text inside it

<PRE> ...</PRE>

Use text already formatted

<BLOCKQUOTE> ...</BLOCKQUOTE>

To quote text from another source

3.1.3 *Make most of fonts*

Fonts are a very important part of any web page. A page looks better if it has a good choice of font sizes and styles.

* **How to do it**

The <FONT> tag is very powerful. With its use,. the size or colour of the font can be changed at any time. <FONT SIZE=(+ or - size)> will increase or decrease the size of the current font. Here's one example of making the first letter of each word in title bigger:

<H3><FONT SIZE=+2>T</FONT>IT <FONT SIZE=+2>F</FONT>OR <FONT SIZE=+2>T</FONT>AT</H3>

Here's what it looks like: TIT FOR TAT

The <TT> tag changes the font to Courier (.if the browser settings are not altered).

3.1.4 *Backgrounds & Colours*

The simplest and effective way to make a page attractive is to add to a page a nice background. Colours are specified as a series of three hexadecimal numbers representing the share of red, green, and blue. For monitors that display fewer colours, the colours selected would be approximated. In the Hotdog editor, choose colours using sliding bars for the share of red, green, and blue; it figures out the corresponding hexadecimal values. One can also change the colour of the text and of the links, before and after they have been clicked. These commands all go in the <BODY> tag, which comes at the beginning of the document after any header information.

* **How to do it**

<BODY BACKGROUND="filename">
  Suppose we have made logo of an organisation and want to give it as background image. We will save it as gif file say logo.gif. It will do the job. It's nice to also add an ending </BODY> at the end of the file.

<BODY BGCOLOR="#RRGGBB">
 allows to set the colour of the background to something specific.

<BODY TEXT="#RRGGBB">
 sets the colour of all normal text within the document.

<BODY LINK="#RRGGBB" VLINK="#RRGGBB" ALINK="#RRGGBB">

sets the colour of all other text in the document. LINK is just a regular, never before visited link, VLINK is a previously viewed link, and ALINK is an active link ('The colour when clicked on it'). Any or all of these three options can be omitted.

RR, GG, and BB are two-digit hex numbers (oo-FF) representing the amount of red, green, and blue in a particular colour. Since it would be too difficult to have a good defined list of colour names, hex numbers are used as well, so one can invent own colours. "#FFFFFF" represents white and "#000000" black.

3.1.5 *Creating tables*

The table must start with the command <TABLE>. Each row must start with the command <tr> (for table row) and within the row, each cell must start and end with <TD> and </TD>, respectively. It ends with the command </TABLE>. If one uses these commands, Netscape and other browsers that support tables will take care of the rest and have a generic table. One can add a caption to the table by using the command <CAPTION>, the text you want, and then </CAPTION>. If you want a header row in the table, use the <TH> and </TH> commands for the first row.

* **Tables Definitions**

Define table
  <TABLE></TABLE>

Table border (either on or off)
  <TABLE BORDER></TABLE>

Table border (one can set the value)
  <TABLE BORDER=?></TABLE>

Cell spacing
  <TABLE CELLSPACING=?>

Desired width (in pixels)

  <TABLE WIDTH=?>

3.1.6 *Handling images*

Images are an essential part of web pages as they convey a great deal of information. They can serve as icons on which to click for moving to other places. But many users of the web have slower speed modems. The graphic that looks great to the designer might make visitors click stop and move to another page.

A graphic can be mixed right with the text, in which case it's called an 'in-Line' graphic, or one can have a link to a graphic which will display by itself on the screen. Some browsers can only

handle the latter.

The tag for in-Line images has the form:
   <IMG SRC="filename" ALT="Description"   ALIGN=top | middle | bottom >

For a visitor who cannot view graphics, designer must write a description of the graphic in the 'ALT=' field that he or she might view. ALIGN specifies if the following line of text is aligned with the top, middle, or bottom of the graphic.

A common and exciting way to use a graphic is to make it part of internal or external link. Suppose we have the graphic file apple.gif in. which a picture of an   apple has been drawn. And we want someone to click on this file apple.gif to go to another html file called fruit.htm where the text describing information about apple is given. The syntax for this would be:

<A HREF="fruit.htm"><IMG SRC="apple.gif ALT="An Apple"></A>

Here we start with the A or Anchor command, put in the reference, and then instead of text we put the graphic reference. Finally close the anchor command with </A>.

If a user clicks any web page and chooses 'View Source' from the browser menu, the corresponding HTML code that the user's browser is following to produce the web page will be displayed. A complete HTML document begins with <HTML> and ends with </HTML>. An example of web page tagging using HTML is shown in Figures 1 and 2.

Java is a programming language one can use to create small miniprograms called applets that can be embedded in web pages. Java applets can be played on any type of computer-Windows, Mac, or UNIX.

Applets are small single-purpose applications that cannot run by themselves; one has to use a compatible web browser (such as Netscape Navigator, or Internet Explorer) for them to work.

4. **FRONTPAGE 2000**

FrontPage 2000 is a popular web editing tool, developed by Microsoft for creating, publishing and managing world wide web pages. The main features of the program are

- Creating new web page quickly with templates, themes and wizards
- Web Clip Art-to include cliparts
- Add Drop Down Menus-they enable the user to choose from several possible options
- Bringing existing sites into FrontPage 2000 without losing any of their HTML coding
- Editing web pages exactly as they appear in a browser
- Switching between normal editing and HTML coding
- Verification of links
- Turning features of FrontPage 2000 on and off depending on web browser used by page visitors
- Using FrontPage server extensions to add interactive capabilities like surveys, discussions forums and feedback pages
- Supports inclusion of forms and frames .
- Connecting the site to Microsoft Access database or Microsoft Excel spreadsheets .
- Integrating web with the other programs in office 2000 suite
- Editing, cropping and adding transparencies to images within FrontPage instead of using an image-editing program
- It implements sophisticated web technology like CSS (Cascading Style Sheets), Javascripts, and ASP (Active Server Pages).

4.1 Components of FrontPage 2000

- FrontPage Explorer-tool to create, view and maintain the web
- FrontPage Editor-HTML Editor
- Image Composer-Graphics editing tool

5. MAINTAINING WEB PAGE

Once an activity is created, one needs to constantly check and revise data to keep it current and informative. A web page will wither away if host does not revisit it often to improve its

appearance, content, and organization, or to add up-to-date information to it. To avoid chaos, one needs to establish a clear, simple system for making updates. Every site needs a Webmaster to fix things that go wrong, respond to basic e-mail, keep the server running, and post updates and changes.to the site as they evolve. One must remove outdated material and broken hyperlink, etc. In short, setting up a web page is not likely to be a one time project one can just finish and forget, it is more of an ongoing commitment

6. PUBLISHING TOOLS

Web editors are good for the laborious task of creating web documents, but they do not necessarily help out much with the ultimate project of building a coherent site out of its constituent pages. They might check the links as they exist now, on the computer, but it takes a special program to help remotely control a site on the server. Such programs can update changed pages regularly and move pages around without manually retyping all the links to those pages.

These programs are more than web editors. Since they help out with the site management they are called web publishing programs.

Adobe PageMill: It publishes our locally created pages to the public site. It finds bad links and corrects links when we re-arrange pages.

Microsoft FrontPage: It is like a basic web editor but has sophisticated site management and automatic/dynamic content creation.

NetObjects Fusion: It allows to design pages in its own format which it eventually translates to HTML when all the pieces are put together.

Netscape Composer: It has complete integration with the Netscape package, one can browse to a page, notice an error, click the edit button, correct the error, click the publish button and post the page back to the server, after error correction.

7. EXAMPLE

Following example may help one get started in planning a web page of a department. The content of the page may include:

- Information about the department-historicalbackground, general description, including location, message from the Director, organisational philosophy, mission, charter of duties.
- Employees' description and activities-name, Qualification, place of work, photographs, e-mail number, address, job responsibility, etc.
- Projects assigned to the department.
- Products developed by the department.
- Achievements, awards.
- Publications-house bulletin and newsletters, literary magazine, course catalogue, publicity material.
- Additional information-activities of other groups, services rendered by the library, HRD

    activities -conferences and courses conducted.
- For background, the photograph of the organisation building, logo, etc., can be given.
- Towards the end of the page, one should give e-mail and phone numbers for any additional information. A line stating date of last updation of the page should also be mentioned
- To make the page more attractive, scientific puzzles, chat facility and latest news updates can also be included.

FURTHER READING

1. Bartlett, Kynn. Six principles of accessible web design: An introduction to the WAI page author guidelines. (Revision 98/12/07)

2. Fulton, Jennifer & Gertler, Nat. Netscape Navigator 6 in 1. QUE Publishing, Indianapolis, Indiana, USA, 1996.

3. Murarka, P.D.; Murarka, Aniket & Shishodia, Y.S. Internet and web page design. BPB Publications, New Delhi, 2000.

4. http:/ /www.cataluna.com

5. Cadenhead, Rogers. SAMS teach yourself Microsoft FrontPage 2000. Techmedia, New Delhi, 1999.

HTML Tag List

Attributes	Description
< !-text->	Comment (text is ignored)
< !DOCTYPE>	Header information about *file* format
HTML	Indicates an HTML *file*
PUBLIC	Indicates a readable document
"Standard"	Indicates the HTML standard in use
< /tag>	Ends the effect of the indicated tag
<A HREF-"URL ">	Marks the start of a link to a document
METHODS= "method,.."	(Advanced) lists functions document supports
REl= " *value, ...*"	(Advanced) lists relationship of link
REV="value, ..."	(Advanced) Reverses relationship of link
TITlE="text"	Gives a name for the page linked to
URN="URN"	(Advanced)Resource name of document
<A NAME=name>	Names a location, for use as a target
<ADDRESS>	Text format for mailing addresses
<APPLET>	Loads a java applet

AllGN=alignment	Locates applet display within text
ALT= "text"	Text for display by non-java browsers
CODE- .URL	(Required) Indicates the program file
CODEBASE="URL"	Directory the program files are in
HEIGHT=pixels	(Required) Applet display area
HSPACEapixels	Horizontal space from applet to text
NAME= " name"	Names applet for intertask messages
VSPACE=pixels	Vertical space from applet to text
WIDTH=pixels	(Required) Applet display area
<AREA>	Describes one link on a mapped image
COORDS="pixels,.."	Left, top, right, bottom of link area
HREF= 'URL	Location to link to
NOHREF	This area isn't a link
SHAPE='RECT"	Rectangular map area
TARGET="frame"	Links to indicated frame
<B>	Makes text bold
<BASE>	Changes defaults for URLs in document
HREF="*URL"*	The new base for relative URLs
TARGET="*frame"*	Specifies default frame for links
<BASEFONT>	Changes default for fonts in document
SIZE=*number*	Sets default font size (1-7)
<BIG>	Increases text size
<BLINK>	Causes text to blink
<BODY>	Starts the page content .
ALlINK="color"	Sets active link color
BACKGROUND= "url"	Sets an image as page backdrop
BGCOLOR="color"	Sets background color for page
LINK="color"	Sets unvisited link color
TEXT="color"	Sets default text color
VLINK="color"	Sets visited link color
 	Starts a new text line
CLEAR=ALL	Starts next line below any images
CLEAR=LEFT	Starts new line below image on left
CLEAR=RIGHT	Starts new line below image on right
<CAPTION>	Sets a caption for a table
ALIGN=BOTTOM	Puts caption below table
ALIGN=TOP	Puts caption above table (default)
<CENTER>	Centers text and images across pages;
<CITE>	Text format for citations
<CODE>	Text format for program code
<DIR>	A directory list
<DD>	Descriptor in definition list

<DL>	A definition list
COMPACT	Reduces list size
<DT>	Defined term of a definition list
<EM>	Emphasizes text (italic)
<EMBED>	Puts area for a plug-in onto page
ALIGN=*align*	Positions area relative to text
BORDER=*pixels*	Sets border color
HEIGHT=*pixels*	Area size
SRC= "*url"*	(Required) Indicated document file
WIDTH=*pixels*	Area size
<FONT>	Changes font attributes
COLOR=color	Changes font color
SIZE=number	Changes font to size number (1-7)
SIZE=+number	Increases font size (up to 6)
SIZE=-number	Decreases font size (down to -6)
<FORM>	Structures a data input form
ACTION= "URL"	Location to send data to
<FRAME>	Sets the attributes for a frame T-pixels
MARGINHEIGH=*pixels*	Space at top and bottom of frame
MARGINWIDTH=*pixels*	Space at side edges of frame
NAME=*frame*	Gives the frame a name
NORESIZE	Prevents frame borders from being moved
SCROLLING=YES	Frame has scroll bars
SCROLLING=NO	Frame doesn't have scroll bars
SCROLLING=AUTO	Frame has scroll bars if needed
SRC="*URL"*	Page to put in frame
<FRAMESET>	Breaks screen into frames
COLS=" framesize,.. "	Sets the width of frame columns
ROWS="framesize,..."	Sets heighl of frame rows
<Hnumber>	Headline Text format level no, (1-6)
ALLIGN=CENTER	Centres the headline
<HEAD>	Indicates the page's header
<HR>	Horizontal rule line
ALIGN=*alignment*	Positions line across page
NOSHADE	Flat line rather than shaded line
SIZE=*pixels*	Thickness of line
WIDTH=*number%*	Line width, as percentage of space
WIDTH=*pixels*	Line Width
<HTML>	Identifies document as being HTML
<I>	Italic font
<IMG>	Inserts an image (graphic)
ALIGN=*alignment*	Positions image relative to text

ALT="*text"*	Text is displayed if graphic can'l be
BORDER=*pixels*	Thickness of border around graphic
HEIGffT=*pixels*	Vertical size of image on page
HSPAC:E=*pixe/ls*	Horizontal space between image & text
ISMAP	This image maps to multiple links
LOWSRC="URL"	Displays this image before SRC image
SRC= "URL".	(Reqd) Image to be displayed
VSPACE=*pixels*	Vertical space between image and text
WIDTH=*pixels*	Horizontal size of image
USEMAP="URL"	File describes links for this image
<INPUT>	A form field
ACCEPT=*"type, ..."*	File types OK in file submission field
ALIGN=*alignment*	Positions image field relativeto text
CHECKED	Check box or option butlon is selected
MAXLENGTH=*number*	Maximum characters user can enter
NAME=*"name"*	Gives field a name
SIZE=number	Size of field in character
SRC=*"URL* "	Image file for button on form
TYPE=CHECKBOX	Check box (yes/no) field
TYPE=File	Field for submission of file
TYPE=HIDDEN	Field not seen by user
TYPE=IMAGE	Form submission button with graphic
TYPE=PASSWORD	Text entry field; text isn't displayed
TYPE=RADIO	An option select field (option button)
TYPE=RESET	Button that clears all fields
TYPE=SUBMIT	Form submission button
TYPE=TEXT	Single-line text field
TYPE=TEXTAREA	Multiple-line text field
VALUE="*text"*	Default value for field
<ISINDEX>	Indicates page is a searchable index
ACTION=*"URL"*	Program to send search request to
PROMPT=*"text"*	Text appears on search form
<KBD>	Text in keyboard formal (monospace)
<LI>	Start of new item on list
TYPE=1	(Default) Arabic numbers (1, 2, 3, etc)
TYPE=a	Lowercase letters (a, b, c, etc.)
TYPE=A	Uppercase letters (A, B, C, etc.)
TYPE=CIRCLE	Use circle as bullet (unordered list)
TYPE=DISC	Use dots as bullet (unordered list)
TYPE=i	Lowercase Roman numerals (i, xiv, etc)
TYPE=I	Uppercase Roman numerals (1, XIV, etc.)
TYPE=SQUARE	Use square as bullet (unordered list)

VALUE=number	Sets entry counter for an ordered list
<LINK>	Shows relationship to another document
METHODs="method.."	.(Advanced) Lists functions documents Supports
REL=*"value,..."*	(Advanced) Lists relationship of links
REV=*"value,..."*	(Advanced) Reverses relationship of links
TITLE="text"	Gives a name for the page link to
URN="URN"	(Advanced) Resource name of document
<LISTING>	Text format with a fixed spacing
<MAP>	Describes what areas of image are links
NAME= *"name"*	(Required) Names the map
<MENU>	A menu list
TYPE=CIRCLE	Use open circle bullets on list
TYPE=DISC	Use dot bullets on list
TYPE=SQUARE	Use square bullets on list
<META>	Holds information to identify page (Reqd)
CONTENT="text"	The information being held
HTTP-EQUIV="text"	Relates info with HTTP response field
NAME= "text" .	Name for the information
<NEXTID>	Machine-picked identifier for document
N= "text"	Identifier for document
<NOBR>	Insert no line breaks into text
<NOFRAMES>	Browsers with frames skip this section
<OL>	Ordered (numbered orettered) list
START=number	First value on list
TYPE=a	Lowercase letters (a, b, c, etc.)
TYPE=A	Uppercase letters (A, B, C, etc.)
TYPE=i	Lowercase Roman numerals (i, xiv, etc.)
TYPE=I	Uppercase Roman numerals (I, XIV, etc.)
TYPE=1	(Default) Arabic numbers (1,2,3, etc.)
<OPTION>	A choice on a form mpnu
DISABLED	This choice cannot be picked
SELECTED	This choice appears as the default
VALUE= *"text"*	Text sent to host if option is chosen
<P>	A text paragraph
ALIGN=*alignment*	Positions text across page
<PARAM>	Passes parameters to an applet
NAME=name	(Reqd) Name of attribute being set
VALUE=value	(Reqd) Value attribute is set to
<PLAINTEXT>	Treat rest of document as text
<PRE>	Preformatted text
<SAMP>	Text format for text samples

<SCRIPT>	A java script
LANGUAGE= "JAVASCRIPT"	lndicates script language
SRC="URL"	Script program file
<SELECT>	A menu field on a form
MULTIPLE	Allows more than one selection
NAME="text"	Name for the field
SIZE=*number*	Number of items visible at a time
<SMALL>	Use a smaller font
<STRIKE>	Display text with a line through it
<STRONG>	Highlighted text (usually bold)
<SUB>	Subscript text
<SUP>	Superscript text
<TABLE>	Create a grid
ALIGN=alignment	Position text within the cell
BORDER	Display a border on the table
BORDER=pixels	Display a border of a certain thickness
CELLPADDING=pixels	Distance between cell frame and content
CELLSPACING=pixels	Distance between cells
HEIGHT=number%	Table height as percentage of space
HEIGHT=pixels	Table height
WIDTH=number%	Table width as percentage of space
WIDTH=pixels	Table width
<TD>	Table cell contents
ALIGN=*alignment*	Horizontal position of text in cell
COLSPAN=*number*	Number of columns this cell covers
HEIGHT=*number%*	Cell height as percentage of table
HEIGHT=*pixels*	Cell height
NOWRAP	No line breaks in cell
ROWSPAN=number	No. of table rows this cell covers
VALIGN=alignment	Vertical position of text in cell
WIDTH=*number%*	Cell width as percentage of table
WIDTH=*pixels*	Cell width
<TEXTAREA>	A multiline text field in a form
COLS=number	(Reqd) Field width, in characters
NAME= "name"	(Reqd) Names the field
ROWS=number	(Reqd) Field height, in characters
WRAP=OFF	(Default) No word wrap
WRAP=PHYSICAL	Word wrap affects display and data
WRAP=VIRTUAL	Affects display but not sent data
<TH>	Table header cell (cell with bold text)
ALIGN=*alignment*	Horizontal position of text in cell
COLSPAN=*number*	Number of columns this cell covers

HEIGHT=number%	Cell height as percentage of table
HEIGHT=*pixels*	Cell height
NOWRAP	No line breaks in cell
ROWSPAN=number	No. of table rows this cell covers
VALIGN=*alignment*	Vertical position of text in cell
WIDTH=*number%*	Cell width as percentage of table
WIDTH=*pixels*	Cell width
<TITLE>	Sets page title, displayed in title bar
<TR>	Table row
ALIGN=*alignment*	Horizontal position of text in cells
VALIGN=*alignment*	Vertical position of text in cells
<TT>	Teletype format (fixed width font)
<UL>	Unnumbered list
TYPE=CIRCLE	Use open circle bullets on list
TYPE=DISC	Use dot bullets on list
TYPE=SQUARE	Use square bullets on list
<VAR>	Text format for program variables
<WBR>	Allows a break even in area
<NOBR>	Area

*Reqd means Required