HTML Guide

Appendices

The CTDP HTML Guide Version 0.8.2, Sept 21, 2001

HTML Guide

This HTML Guide is to give the reader the ability to write HTML documents and get a web site up and running. This document should enable the reader to understand HTML quickly and also be useful as reference material. This document is written for the beginner. After learning this document, the reader will be ready for more advanced subjects such as DHTML, XML, SGML, and client and server side script writing.

HTML Versions

You should be aware that there are historically many versions of the HTML specification. You will read most about the HTML 3.0, 3.2, and 4.0 specification. I have tried to display HTML tags and attributes that are supported by the 3.2 and 4.0 specification, but cannot guarantee that all tags and attributes contained herein will work in all versions of HTML. There are several HTML common document types when refering to HTML 4.0 which are:

- Strict - HTML 4.0 without frame or HTML 3.2 depreciated feature support.
- Frame - HTML 4.0/3.2 with frame support.
- Transitional or loose - Supports features of HTML 3.2 that are being depreciated in HTML 4.0. This DTD does not support framesets. Some depreciated tags/elements not supported by strict HTML 4.0 are CENTER, FRAME, FRAMESET, NOFRAMES, IFRAME, ISINDEX, DIR, MENU, APPLET, BASEFONT, FONT, S, STRIKE, and U.

This document covers most commonly used HTML elements and attributes, but not all. For an excellent source of information regarding available HTML 4.0 elements and attributes, refer to the Web Design Group's web page, HTML 4.0 Reference section. Used in combination with information from the Web Design Group and the World Wide Web Consortium web pages (See the links section or "Recommended Reading" section of this document for URL) this document should provide a thorough introduction to HTML.

HTML Support

You should also know that many of the element tags are being depreciated in favor of cascading style

sheets. This means that some of the attributes with regard to text fonts and color are being depreciated and these values are to be controlled using the style sheets. This makes it easier to control the attributes of multiple HTML documents such as the HTML in this document from one cascading file sheet file. You should learn at least minimal control of attributes using style sheets before writing great quanties of documents and thereby being forced to go back and change your files. Some examples in this document demonstrate the use of style sheet and inline style commands. Many of the elements not supported by the strict HTML 4.0 document type definition (DTD) will be noted with the word **"depreciated"** where they are presented.

One other consideration to be aware of is that various client web browsers have various support for various methods of setting attributes and displaying information that is contained in web pages. For instance, not all older web browsers support frames, but most clients should have web browsers today that support frames. Also some style sheet properties may be supported by some browsers but not by others. The best way to determine what will work is to read referenced documentation and experiment.

Getting ready to create a website

You can create your web site off line using a standard text editor, an HTML text editor, or a HTML graphics editor. I recommend using an HTML text editor which lets you see the HTML tags in a different color than the surrounding text which makes them easier to see. The free ware editor, Arachnophillia, is a good choice to begin with. There are also some editors which perform validation dynamically. One of these may be an even better choice if you can obtain one.

Directory Structure

You will want to create your web pages in a separate directory on your computer and as you link deeper into your web pages you will want to create more sub-directories. Normally, I use the following structure:

- mainpage - The main folder for your web page, name it an appropriate name for your web site. Your HTML files for your home page will go here.
- gifs - Where I put all gif files to be used in common with the mainpage and other pages such as buttons used to provide links to other parts of my web page. I usually place all backgrounds here. I do not put pictures for specific sections here. For example, all pictures that are used in "The CTDP Networking Guide" are placed with the HTML files in that folder.
- subject1 - In the case of the CTDP, this may be "Linux". Your html and graphics files for a particular web page subsection will go here.
    - Subject1/sub1 - Sub subjects of subject 1.

    - subect1/sub2
- subject2 - This may be "networking".
- subject2.

Starting with a Template

To begin your first web page, you will want to create a template from which you can start all your pages which has the basic structure of an HTML document as shown in the "HTML document Structure" section of this document. You can copy the example file shown there into your editor's file or make your own. You will want to follow the standard HTML structure of HTML, HEAD, and BODY elements. If you want to set your HTML page with multiple frames, for example with an HTML index on the left side for easier document viewing, you will need to become familiar with the section on FRAMES before you begin. If working with frames is initially too complicated, stick with the standard HEAD, BODY structure until you gain familiarity.

HTML Structure

HTML uses tags that are encased in brackets like the following:

< >

HTML documents consist of **elements** which are constructed with **tags**. For instance, a paragraph is considered to be an html element constructed with the tags <P> and </P>. The <P> tag begins the paragraph element and the </P> tag ends the element. The following is a paragraph element.

```
<p>
This is a sample paragraph element. Any text in a paragraph goes here.
</p>
```

Not all tags have a tag for ending the element such as the line break, <br> tag. The HTML document is begun with the <html> tag and ended with the </html> tag. Elements of an HTML document include the HEAD, BODY, paragraphs, lists, tables, and more. Elements may be embedded within each other. Also some elements have **attributes** embedded in the tag that define characteristics of the element such as the placing of text, size of text, source of an image, and other characteristics depending on the element. These attributes are listed in this document when the element is discussed in detail.

An HTML document is structured with two main elements:

1. HEAD
2. BODY

An Example HTML File

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
   <meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1">
   <meta name="GENERATOR" content="Arachnophilia 3.9">
   <meta name="description" content="Comprehensive Documentation and
information about HTML.">
   <meta name="keywords" content="HTML, tags, commands">
   <title>The CTDP HTML Guide</title>
```

```
<link href="style.css" rel="stylesheet" type="text/css">
<!-- Background white, links blue (unvisited), navy (visited), red
(active) -->
</head>

<body>
<center><h1>HTML Document Structure</h1></center>
<p>
This is a sample HTML file.
</p>
</body>
</html>
```

Comments begin with <! and end with the > bracket. The tags "HTML", "BODY", and all others may be in capital or small letters, however the new XHTML standard requires small letters so small letters are recommended.

In the above file, there is a header and a body. Normally you can copy this file and use it as a template to build your new file while being sure to make some modifications. You can edit HTML using a standard editor, but it is easier to use an HTML editor like Arachnophilia since it displays the tags with different colors than the text is displayed in. Also note the LINK element above which specifies a style sheet to be used. This is a file with a name "style.css". This is a text file that may be created with a text editor but must be saved in plain text format. See the section called "Setting Document Style" for more information about style sheets.

The center tag used above is depreciated. It is best to use style sheets to center headers and other items. This is covered in more detail later.

HTML Header

The Header

The header contains several notable items which include:

1. doctype - This gives a description of the type of HTML document this is.
2. meta name="description" - This gives a description of the page for search engines.
3. meta name="keywords" - This line sets keywords which search engines may use to find your page.
4. title - Defines the name of your document for your browser.

The header in our example document:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
   <meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
   <meta name="GENERATOR" content="Arachnophilia 3.9">
   <meta name="description" content="Comprehensive
Documentation and
 information about HTML.">
   <meta name="keywords" content="HTML, tags, commands">
   <title>The CTDP HTML Guide</title>
<link href="style.css" rel="stylesheet" type="text/css">
</head>
```

The <!DOCTYPE> Line

The <!DOCTYPE> line is used to define the type of HTML document or page. It has no ending tag. The three document types that are recommended by the World Wide Web Consortium (W3C) are:

1. <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
   This implies strict adherence with HTML 4 standards.
2. <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN">
   This supports frameset tags. Some documentation claims this sets strict adherence with HTML 4 standards, however by using HTML validators for testing, I concluded that this is the same as

Transitional with frameset support.

3. <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
   This is used to support depreciated HTML 3.2 features. It does not support frames.

Note: I have noticed that capitalization is important in the document type declaration. Not all validators will recognize lower case when the text "-//W3C//DTD HTML 4.01//EN" is "-//w3c//dtd html 4.01//en". Therefore it is best if you use the upper case values as in the example above.

Elements in the Header

Elements allowed in the HTML 4.0 strict HEAD element are:

- BASE - Defines the base location for resources in the current HTML document. Supports the TARGET attribute in frame and transitional document type definitions.
- LINK - Used to set relationships of other documents with this document.
- META - Used to set specific characteristics of the web page and provide information to readers and search engines.
- SCRIPT - Used to embed script in the header of an HTML document.
- STYLE - Used to embed a style sheet in the HTML document.
- TITLE - Sets the document title.

The additional element allowed by the HTML 4.0 transitional standard is:

- ISINDEX (Depreciated) - Allows a single line of text input. Use the INPUT element rather than ISINDEX.

The <META> Element

The <META> element is used to set specific characteristics of the web page and provide information to readers and search engines. It has no ending tag. **Attributes**

- **http-equiv** - Possible values are:
  - **refresh** - The browser will reload the document after the specified seconds specified with the CONTENT value have el lapsed. Ex: <META HTTP-EQUIV=refresh CONTENT=45>
  - **expires** - Gives the date that content in the document is considered unreliable.
  - **reply-to** - A an email address of the responsible party for the web page. This attribute is

not commonly used. Ex: <META HTTP-EQUIV=reply-to CONTENT="ctdp@tripod.com">

- **name** - Provides non-critical information about the document possibly useful to someone looking at it. Possible values are:
  - **author** - The person who made the page or the HTML editor name . Ex: <META NAME=author CONTENT="Mark Allen">
  - **description** - An explanation of the page or its use, used by search engines at times to provide a page summary. Ex: <META NAME=description CONTENT="The CTDP Home Page">
  - **copyright** - A copyright notice for the page. Ex: <META NAME=copyright CONTENT="Copyright 2000, Mark Allen">
  - **keywords** - A list of keywords which are separated by commas. These keywords are used by search engines. EX: <META name="keywords" content="computer documentation, computers, documentation, computer help">

    This section is very important if you want your web page to be found by search engines. Please note that **keywords are separated by commas, not spaces and that the words "computer documentation" are treated by search engines as one word**. If someone enters the phrase "computer documentation" when doing a search, it gives the web page a much greater chance of being found than just having the separate keywords "computer" and "documentation".

  - **date** - <META name="date" content="2000-05-07T09:10:56+00:00">
- **content** - Specifies a property's value such as the content of this document is text/HTML.
- **scheme** - Names a scheme to be used to interpret the property's value.

The <BASE> Element

This element defines the way relative hyper links are handled in the body of the document. It has no ending tag. It supports the TARGET attribute in frame and transitional document type definitions. Only one BASE element may exist in the header.

If you copy the websites page from the CTDP website at http://ctdp.tripod.com/websites.html to your website at http://yoursite.yourwebserver.com/websites.html, then any relative links on the copied page will try to link to the http://ctdp.tripod.com page rather than the http://yoursite.yourwebserver.com page and you may get the "file not found" error from your browser. Inserting the <base> tag into the copied page will fix the problem.

<base href="http://www.comptechdoc.org/">

Without this tag, if there were a relative link to "linux/index.html" on this page, the user's web browser would look for the file in "http://yoursite.yourwebserver.com/linux/index.html" which would not exist unless it were also copied over and placed at the same relative location on your website.

The <LINK> Element

Used to set relationships of other documents with this document. For example a style sheet that is used to control element style (as in the above example) may be defined with this element. Attributes:

- CHARSET - The character encoding of the link. An example is "ISO-8859-1".
- HREF - The location and name of the resource.
- HREFLANG - Describes the expected language. An example value is "en" for English.
- MEDIA - Describes the type of display the content is designed for. Possible values include all, aural, braille, handheld, projection, print, screen, tty, and tv.
- REL - Relationship to the linked resource.
- REV - Relationship from the linked resource to the document. Possible values of REL and REV include alternate, appendix, stylesheet, bookmark, chapter, contents, copyright, glossary, help, index, next, prev, start, section, and subsection. The value "made" supported by HTML 3.2 is not supported by HTML 4.0.
- TYPE - Describes the expected content of the resource the link points to. Typical values include application/java, text/html, text/css, image/jpeg, and text/javascript.

An additional attribute allowed by the HTML 4.0 transitional standard is:

- TARGET - Specifies the frame target the referenced page will be put into. The target may me a named window or one of the special reserved names below. If the target is a named window that exists the web page the link points to will be loaded into that window. If a window by that name does not exist, a new window with that name will be created.
    - _blank - The web linked page loads in a new window and does not name the window.
    - _parent - The new page will load in the parent frame or window.
    - _self - The new page will load in the same window that the link is in. If the link is in a frame, it will load inside the frame. This is the default target.
    - _top - The new page will load into the top window reguardless of how many framesets deep the link is embedded.

An example:

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

The <TITLE> Element

The text between the <title> and </title> tags is the title of the document.

An example:

<title>The CTDP HTML Guide</title>

The <SCRIPT> Element

Used to embed script in the header of an HTML document. Attributes:

- CHARSET - The character encoding of the script such as "ISO-8859-1".
- DEFER - The script will not be parsed until the document is loaded.
- LANGUAGE - Describes the name of the script language.
- SRC - The external location where the script may be. In this case the script in not includes in between the SCRIPT tags, but an external file is loaded.
- TYPE - Describes the content type of the script language.

The <STYLE> Element

This element is used to embed a style sheet in the HTML document. Attributes:

- MEDIA - Same as described in the LINK element above.
- TITLE - The style sheet title
- TYPE - The content type.

If this element is used an external style sheet is not used as can be done using the LINK element, above. The style sheet information is embedded between the <style> and </style> tags.

HTML Body

The Body

The body will define the rest of the HTML page which is the bulk of your document. It will include headers, paragraphs, lists, tables, and more. These elements are described fully in the rest of this document.

An example body section:

```
<body text="#000000" bgcolor="#FFFFFF" link="#0000FF"
   vlink="#000080" alink="#FF0000">
<h1 style="text-align: center">HTML Document Structure</h1>
<p>
This is a sample HTML file.
</p>
</body>
</html>
```

The above example is similar to the body section in the example in the "HTML Document Structure" section with the exception that this example controls the body background, wallpaper, and link color directly rather than using style sheets. As you can see below many of the attributes to control the BODY are depreciated in favor of style sheet use.

The BODY Element Tags and Attributes

The <body> tag is used to start the BODY element and the </body> tag ends it. It is used to divide a web page within one or more sections. Its tags and attributes are:

- <body> - Designates the start of the body.
  - ONLOAD - Used to specify the name of a script to run when the document is loaded.
  - ONUNLOAD - Used to specify the name of a script to run when the document exits.
  - BACKGROUND="clouds.gif" - **(Depreciated)** Defines the name of a file to use for the background for the page. The background can be specified as in the following line.

    <body text="#000000" bgcolor="#FFFFFF" link="#0000EF" vlink="#51188E" alink="#FF0000" background="clouds.gif">

- BGCOLOR="white" - **(Depreciated)** Designates the page background color.
- TEXT="black" - **(Depreciated)** Designates the color of the page's text.
- LINK="blue" - **(Depreciated)** Designates the color of links that have not been visited.
- ALINK="red" - **(Depreciated)** Designates the color of the link currently being visited.
- VliNK="green" - **(Depreciated)** Designates the color of visited links.

- </body> - Designates the end of the body.

For the depreciated attributes noted above, see the "Setting document style" section for an example of how to set the same attributes using a style sheet. If the colors are defined with "#FFFFFF" as done in the above example, the values indicate the intensity of the colors, red, green, and blue in hexadecimal. The leftmost two "FF" indicate the strength of the color red, the center indicates the strength of the color green, and the rightmost two indicate the strength of the color blue. The values may be a hexadecimal value from 0 through FF which in decimal is 0 through 255. The highest value being the highest strength of the respective color. This format is "RRGGBB". You can also specify the color with one of the following standard colors.

1. black
2. white
3. red
4. green
5. blue
6. yellow
7. magenta or fuschia
8. cyan or aqua
9. purple
10. gray
11. lime
12. maroon
13. navy
14. olive
15. silver
16. teal

Setting a background image on the page

To set up a background on the web page use a BODY statement with the BACKGROUND attribute defined as follows:

```
<body text="#000000" bgcolor="#FFFFFF" link="#0000FF" vlink="#000080"
alink="#FF0000" background="clouds.gif">
```

This method is depreciated and authors are encouraged to use style sheets.

HTML Frames

Frames can be used to split a web page so you can load multiple HTML files or pages into one web page. This will cause the web browser to display multiple pages at one time. A common example of this technique which I like to use is to display an index of a document on the left side of the browser window with descriptions and links so when the reader clicks on the link, that page is displayed on the right side of the browser window. Using frames may be a bit complicated at first for those new to HTML, so you may want to skip this section or skim it, then come back to it when you want to create frames in your HTML documents.

Important Note about Frame use

Should you decide to use frames on your web page, you should be aware of some things.

- Not all browser and webcrawler programs support frames. This is especially important with regard to webcrawlers since, if your pages are not browsed by webcrawlers, search engines may not log your pages for those searching for your subject material. Therefore, you should utilize the NOFRAMES tag described below to support frames. In your file that contains the FRAMESET, you may support this by including a copy of your index body in the NOFRAMES section.
- Also, to support users that can't see frames, you should put a link at the top and bottom of the page that allows them to get back to your main index or home page.
- The Frameset DTD must be specified on the first line as in the below example rather than the strict or transitional DTD in order to support frames.

When I start an html document, I typically call the first page in the document "index.html". This is because it is a standard name used by web servers for the home page, and it is easy to remember that most html documents start with the name "index.html". The "index.html" file and all associated html files it will link to which are a part of the document are contained in the same sub directory on the computer. Therefore, when writing this document I started with the following HTML file as my "index. html" file. Note: Some experts would argue the wisdom of naming your file "index.html", in favor of a more descriptive title that would show up on search engines. What you do should depend on your needs.

```
<!DOCTYPE HTML PUBliC "-//W3C//DTD HTML Frameset 4.01//EN">
<html>
<head>
   <meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1">
   <meta name="GENERATOR" content="Arachnophilia 3.9">
```

```
   <META NAME="FORMATTER" CONTENT="Arachnophilia 3.9">
   <meta name="keywords" content="Documentation, networking, howto,
routing,
information, routers, firewalls, firewall, IP, masquerading,
masquerade, arp, rarp, tcp,
udp, protocol, protocols">
   <title>The CTDP HTML Guide</title>
<head>
<frameset border="1" cols="200,*" frameBorder="0" frameSpacing="4">
<noframes>
<body>
<p>
You should include HTML here to support webcrawlers and browsers that
don't support frames.
You may want to include a second copy of your index, and set your
wallpaper and colors
in the BODY statement above the same as you would in your index file.
</p>
</body>
</noframes>
<frame name="left" src="htmlindex.html">
<frame name="right" src="htmlintroduction.html">
</frameset>
</html>
```

Notice that in this example there are initially no <body> tags. The only <body> tags occur inside the NOFRAMES section which supports browsers that are not frames capable. The <body> tag is replaced by the <frameset> tags. The <frame> tags are used to define each element in the FRAMESET. Notice that there is no </frame> to end the FRAME element. You should also note that rather than setting the document type (in the first line, above) to Transitional, it is set to **Frameset** in order to support frames.

Tags and associated attributes for FRAMES

- <FRAMESET> - Used to divide the browser window into separate frames that can display multiple documents
    - ROWS="10" - Splits the window vertically into a number of rows specified by one of:
        1. A number of rows.
        2. A percentage of the total window height. EX: ROWS="10%".
        3. An asterisk means the frame will occupy all the remaining space. It will divide the space evenly between frames if multiple frames are specified with the asterisk.

- ColS="10" - Splits the window horizontally into a number of columns specified by one of:
  1. A number of columns.
  2. A percentage of the total window width. EX: ColS="10%".
  3. An asterisk means the frame will occupy all the remaining space. It will divide the space evenly between frames if multiple frames are specified with the asterisk.
- ONLOAD - Used to specify the name of a script to run when the document is loaded.
- ONUNLOAD - Used to specify the name of a script to run when the document exits.
- FRAMESPACING="12" - **(Depreciated)** Defines the space between the frames in pixels.
- FRAMEBORDER="NO" - **(Depreciated)** Defines whether a border for the frames should be displayed. The choices are YES or NO.
- BORDER="5" - **(Depreciated)** The size of the frame border in pixels.

- <FRAME> - Specifies a frame to be placed inside a FRAMESET.
  - FRAMEBORDER - Determines if there is a visible border for the frame. The possible values are 0 and 1 with a default value of 1.
  - LONGDESC - A frame long description.
  - NAME=left - Defines the name of the frame. In this case, the name of the frame is "left". This will be used in A (anchor) references later to place various content into the frame.
  - SRC="htmlindex.html" - Specifies the location or the URL (Universal Resource Location) of the document that will be placed in the frame.
  - MARGINHEIGHT="5" - The number of pixels left above and below a document that is in a frame.
  - MARGINWIDTH="10" - The number pixels to left on the right and left side of a document in a frame.
  - SCRolliNG="AUTO" - Specifies the possible existence of scrollbars in a frame. Possible values are YES, NO, and AUTO, with AUTO as the default.
  - NORESIZE - Keeps this frame from being resized with the mouse.
  - SCRolliNG - Determines if the frame can use a scroll bar if it exceeds the size of the available area on the browser. Possible values are yes, no, and auto.
- <NOFRAMES> - Used for web browsers that don't support FRAMESETs. It provides an alternate body in documents that are FRAMESET documents. The <BODY> and </BODY> tags which define the BODY element are normally inside the <NOFRAMES>...</NOFRAMES> tags.
- </NOFRAMES> - Ending tag for the NOFRAMES element..
- </FRAMESET> - Used to end a frameset.

Another example

Another, slightly more complicated example is:

```
<frameset rows="10%, *">
   <frame name="top" src="topselect.html">
```

```
   <frameset cols="10%, 30%, 30%, 30%">
      <frame name="leftbar" src="myindex.html">
      <frame name="mleft" src="mleft.html">
      <frame name="mmiddle" src="mmiddle.html">
      <frame name="mright" src="mright.html">
   <frameset>
</frameset>
```

This example uses two framesets. The first divides a top section and a bottom section. The second frame set divides the rest of the window (90%) into four sections, the first composing the leftmost 10%, and each of the other three sharing 30%.

How to get new information into the frame

To make frames useful, you will need to use anchor links to direct the contents of HTML files to them. This is done using the TARGET attribute with the anchor tag as follows:

```
<a href="htmlstructure.html" target="right">HTML Document Structure</
a>
```

In my first example, I defined a frame with the name "right". When the user clicks on this link with their web browser, the HTML file pointed to by the link will be directed to the target named "right" which is defined on the right side of the browser window. The anchor statements shown above are placed in the file called htmlindex.html which resides on the left side of the browser window. This allows the user to control the contents of the right side of the browser with the index file, full of anchor links, which resides in the left side of the browser window. A sample abbreviated index file is listed below:

```
<head>
   <meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1">
   <meta name="GENERATOR" content="Arachnophilia 3.9">
   <meat name="FORMATTER" CONTENT="Arachnophilia 3.9">
   <title>The CTDP HTML Guide</title>
</head>
<body>
<h3>HTML Index</h3>
<hr>
<ol>
<li><a href="htmlintroduction.html" target="right">Introduction</a>
<h3>Structure</h3>
```

```
<li><a href="htmlstructure.html" target="right">HTML Document
Structure</a>
<li><a href="htmlhead.html" target="right">The Header Section</a>
<li><a href="htmlbody.html" target="right">The Body Section<a>
<li><a href="htmlframes.html" target="right">Using Frames</a>
<h3>Layout Mechanisms<h3>
<li><a href="htmlparagraphs.html" target="right">Headers and
Paragraphs<a>
<li><a href="htmllists.html" target="right">Lists</a>
<li><a href="htmltables.html" target="right">Tables</a>
</ol>
<center>
<a href="../index.html" target="_top" onMouseOver="window.status='To
Computer
 Tech Home Page' ;return true" onMouseOut="window.status='';return
true"><img
 src="../greenhomebutton.gif" alt="Home" vspace=3 border=0 height=33
width=115<a>
</center>
</body>
</html>
```

HTML Guide Contents Page

HTML Element Categories

It is important to be aware, when using HTML and style sheets that HTML elements are categorized into several categories. Some style properties apply to some categories of elements and not to others. The following types of elements exist:

- Block - Include a line break before and after the element.
- Inline - Included with the text as part of a line.
- List Item - Elements that support providing a list of items. List item elements are block level elements.

Block HTML Elements

A block with centered contents is defined

Name	Description	Comment
ADDRESS	Supplies contact information for the document	-
BLOCKQUOTE	Used to quote in block form	-
CENTER	Depreciated	-
DIV	A container allowing specific style to be added to a block of text.	-
DIR	A container allowing specific style to be added to a block of text.	-
FRAMESET	A container allowing multiple frames (HTML documents) to be placed on a web browser.	-
H1, H2, H3, H4, H5, H6	Headings	-
HR	Horizontal rule	-
ISINDEX	An input prompt for a single line of text	Depreciated
NOFRAMES	Alternate content for browsers that do not support frames.	-
NOSCRIPT	Alternate content for browsers that cannot run script programs	-
P	Paragraph - Should not contain other block elements including tables, but may contain list elements	-

PRE	Preformatted text is rendered with spaces and carriage returns as typed.	-
TABLE	Used to present an ordered set of data. Table subelements work as block elements.	Includes table sub elements
FORM	Used to present a form to the client. Form subelements work as block elements.	-

List item elements are also considered to be block elements.

List-item Elements

Name	Description	Comment
DIR	Directory List	Depreciated
DL	Definition List	-
LI	List Item	-
OL	Ordered (numbered) List	-
UL	Unordered List	-

Block/Inline HTML Elements

The elements in this list may be considered block or inline elements, depending on how they are used. They are considered inline elements if used inside a paragraph (P) or another inline element.

-

Name	Description	Purpose	Comment
APPLET	For placing Java applets in the page.	-	(Depreciated)
BUTTON	Used to define a FORM submit, reset, or push button.	-	-
DEL	Used to indicate deleted content	Phrase	Both a block and inline element
IFRAME	Inline frame	-	(Depreciated)
INS	Marks inserted content	Phrase	Both a block and inline element
MAP	Establishes an image map to be used with an IMG or OBJECT element	-	-
OBJECT	Used to include applets, images, videos and sound on the web page.	-	-

SCRIPT	Defines a script to be run on the client computer.	-	-

Inline HTML Elements

Name	Description	Purpose	Comment
A	Anchor	-	-
ABBR	Denotes an abbreviation	Phrase	-
ACRONYM	Denotes an acronym.	-	-
AREA	Define a map region in an image.	Inside the MAP element	-
B	Bold	Font	-
BASEFONT	Allows font changes	-	Depreciated
BDO	Overrides text direction with values of ltr (left to right) or rtl (right to left).	-	-
BIG	Sets size of text to big	Font	-
BR	Break	Treated special by style sheets.	-
CITE	Used to mark titles of articles or other publications.	Phrase	-
CODE	Denotes computer program code.	Phrase	-
DFN	Denotes a definition.	Phrase	-
EM	Denotes emphasis	Phrase	-
FONT	Allows font changes.	-	(Depreciated)
I	Sets text to italics	Font	-
IMG	Allows placement of a graphical image in the page.	-	-
INS	Denotes inserted text.	Phrase	-
KBD	Denotes information typed from the keyboard.	Phrase	-
PARAM	Used to add additional parameters to the object or applet elements.	-	-
Q	Used for short quotations	-	-
S	Strike through text	Font	Depreciated
SAMP	Denotes a sample.	Phrase	-
SMALL	Sets text size to small	Font	-

SPAN	A container used to set special style to specific areas of the page.		-	-
STRIKE	Sets text to have a line struck through it	Font		-
STRONG	Denotes	Phrase		-
SUB	Subscript		-	-
SUP	Superscript		-	-
TEXTAREA	A form for multiline text input.		-	-
TT	Sets text style to monospaced teletype	Font		-
U	Sets text underlined	Font		Depreciated
VAR	Denotes a variablein a program.	Phrase		-

HTML Guide Contents Page

HTML Elements

HTML catagorizes elements into specific categories. The two main types of element categories are:

- Block - Include a line break before and after the element.
- Inline - Included with the text as part of a line.

See the section called "HTMLElement Categories" for a list of block and inline elements with short descriptions.

In HTML there are specific rules with regard to what elements a specific element may contain. For example some elements may only be contained in the HEAD header element. The paragraph element may not contain block elements. The table below lists elements and what they may contain:

Name	Contained Elements
A	inline elements except A
ABBR	inline elements
ACRONYM	inline elements
ADDRESS	inline elements
APPLET	PARAM, inline elements block elements
AREA	none
B	inline elements
BASE	none
BASEFONT	none
BDO	inline elements
BIG	inline elements
BLOCKQUOTE	block elements, SCRIPT
BODY	inline elements, block elements
BR	none
BUTTON	inline elements excluding A, BUTTON, INPUT, LABEL, IFRAME, SELECT, and TEXTAREA
CENTER	inline elements, block elements
CITE	inline elements
CODE	inline elements
DD	inline elements

DEL	inline elements, block elements
DFN	inline elements
DIR	LI excluding block elements
DIV	inline elements, block elements
DL	DD, DT
DT	inline elements, block elements
EM	inline elements
FIELDSET	inline elements, block elements, LEGEND
FONT	inline elements
FORM	block elements excluding FORM
FRAME	none
FRAMESET	FRAMESET, FRAME, NOFRAMES
H1-H6	inline elements
HEAD	BASE, ISINDEX, LINK, META, OBJECT, SCRIPT, STYLE, TITLE
HR	none
I	inline elements
IFRAME	inline elements, block elements
IMG	none
INPUT	inline elements, block elements excluding BUTTON
INS	inline elements, block elements
ISINDEX	none
HTML	HEAD, BODY, FRAMESET
KBD	inline elements
LABEL	inline elements excluding LABEL
LEGEND	inline elements
LI	inline elements, block elements (MENU and DIR list items exclude block elements)
LINK	none
MAP	block elements, AREA
MENU	LI not containing block elements
META	none
NOFRAMES	inline elements, block elements, BODY
NOSCRIPT	inline elements, block elements
OBJECT	PARAM, inline elements block elements
OL	LI
OPTGROUP	OPTION

OPTION	none (text)
P	inline elements
PARAM	none
PRE	inline elements excluding APPLET, BIG, BASEFONT, FONT, IMG, OBJECT, SMALL, SUB, SUP
Q	inline elements
S	inline elements
SAMP	inline elements
SELECT	OPTION, OPTGROUP
SCRIPT	none (script program)
SMALL	inline elements
SPAN	inline elements
STRIKE	inline elements
STRONG	inline elements
STYLE	none (A style sheet)
SUB	inline elements
SUP	inline elements
TBODY	TR
TD	inline elements, block elements
TFOOT	TR
TH	inline elements, block elements
THEAD	TR
TR	TH,TD
TEXTAREA	none (text)
TITLE	none
TT	inline elements
U	inline elements
UL	LI
VAR	inline elements

HTML Headers and Paragraphs

HTML Headings

There are 6 levels (H1 through H6). H1 is the largest. Headings are always displayed as bold text (Unless modified using the STYLE attribute or style sheets). Headings with numbers above the <h3> tag such as <h4><h5> and <h6> are not normally used since they are so small.

<h1> - Start heading of level 1.
</h1> - End level 1 heading.

Heading attributes are the same as the paragraph attributes, below. No attributes are supported by strict HTML 4.0 other than common attributes such as STYLE which are used to set color and position.

HTML Paragraph

The paragraph element begins with the <P> tag and ends with the </P> tag. It should not contain tables and other block elements. A sample is shown below.

```
<p>
This is a sample paragraph element. Any text in a paragraph goes here.
</p>
```

Paragraph Attributes

- ALIGN="CENTER" - **(Depreciated)** Sets the alignment of the paragraph with respect to the page size. Values are LEFT, RIGHT, and CENTER.
- STYLE="font: 16pt courier" - Sets the style or color of the text. This statement starts a paragraph with color, blue: <p style="color: green">. The STYLE attribute is common to most HTML elements (See the "Common Attributes" section).

Paragraph Examples

This is the HTML code for a paragraph demonstrating use of the STYLE attribute along with the SPAN element.

The color of the sky is blue, but if you look at the trees, they are green. This effect is produced using the &#60span&#62 element with the style ="color: 'green'" attribute set.

This is how it looks:

The color of the sky is blue, but if you look at the trees, they are green. This effect is produced using the <SPAN> element with the STYLE ="color: 'green'" attribute set.

An example HTML document using Headers and Paragraphs

This example is from "The CTDP Linux User's Guide":

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<meta name="GENERATOR" content="Arachnophilia 3.9">
<meta name="description" content="Comprehensive Documentation and information about Linux.">
<title>The CTDP Linux User's Guide</title>
</head>
<body>
<center><h1>Logging in, Logging Out, and Shutting down</h1></center>
<h3>Logging in</h3>
<p>
Once you have completed your system install and booted your system, you should see a login prompt on your monitor. When you did your Linux install you should have set a root password. You may have also created a user with a password. Therefore to log in, you will want to type the name of a user or "root" for the login name and enter the appropriate password. If you logged in as a normal user and know the root password and want to use administration commands, you may use the command "su" to become a "super user". Some systems also support the "sudo" command, which allows administrative privileges on a command by command basis.
</p>
```

```
<h3>Logging out</h3>
<p>
Use the command "logout" to exit a given session. If you have logged in, then typed "su" to become a super user or another user, you may need to type "exit" until your SHLVL environment value is 1. Then you can type "logout" to exit your session. The "exit" command will take you back to previous shell levels.
</p>
<h3>Shutting Down</h3>
<p>
The system is intended to be shutdown by the system administrator using the shutdown command in one of the forms shown below. Many systems are set up to capture the &#60CTRL&#62&#60ALT&#62&#60DEL&#62 keystroke combination to issue the shutdown command through the init program. This will work on most systems if the root user is logged in. Examples of using the shutdown command are shown below.
<blockquote>
shutdown -h now<br>
shutdown -r +10 "Rebooting in 10 minutes"<BR>
shutdown -r 13:00<BR>
</blockquote>
The first command will shutdown and halt the system immediately. The second will reboot the system in 10 minutes and send the message to all users. The third command will shut the system down and do a reboot at 1:00 in the afternoon.
</p>
</body>
</html>
```

Although this document is about paragraphs and headers, there are a few things worth noting here.

1. In the last paragraph, there is a <blockquote>...</blockquote> element embedded. Although this works, **the use of BLOCKQUOTE in this instance is improper**. Not only should it be done using style sheets, as follows, the use of BLOCKQUOTE inside the paragraph element, P, is also not proper. The text where the blockquote is should be changed to:

   ```
   <p class="indent">
   shutdown -h now<br>
   shutdown -r +10 "Rebooting in 10 minutes"<br>
   shutdown -r 13:00<br>
   </p>
   ```

   Of course the previous paragraph would need to be terminated with the tag and the following paragraph started with the tag. The style sheet should contain:

."indent" { margin-right: '5%'; margin-left: '5%'; color: '#0000ff' }

This defines a class called ""indent"" with specific margin characteristics and color which, when defined in the starting <P> tag, causes the contents of the element to have the attributes declared for that class.

2. In the last paragraph, the following line:

   Many systems are set up to capture the &#60CTRL&#62&#60ALT&#62&#60DEL&#62 keystroke combination to issue the shutdown command through the init program.

   appears as follows when displayed on the web browser:

   Many systems are set up to capture the <CTRL><ALT><DEL> keystroke combination to issue the shutdown command through the init program.

   This is because there are characters with special meaning in HTML and since HTML uses the < and > characters to set up tags, there must be a special character combination to display them so they are not used as part of a tag. These special characters are listed in the section on text formatting.
3. The and tags are used to center the header, <h1>...</h1>, at the top of the page. This can also be done with the following code using the STYLE attribute which conforms to HTML 4.01 by not using depreciated attributes.

<h1 style="text-align: 'center'">Logging in, Logging Out, and Shutting down</h1>

HTML Lists

HTML 4.0 supports three types of lists which are:

1. OL - Ordered list
2. UL - Unordered list
3. DL - Definition List

Directory and menu lists are depreciated.

Ordered Lists

Tags and Attributes

- <OL> - Designates the start of an ordered list
  - TYPE="1" - **(Depreciated)** Designates the numerical label type for use in the list. Values are A for capital letters, a for small letters, I for large Roman numerals, i for small Roman numerals, 1 for numerical integers. Default is "1".
  - START="3" - **(Depreciated)** Specifies value the list starts with. This value must be a number such as 1,2,3,4, etc and causes the first or current item of the list to begin with that value. If the type is set to "A" and start is set to "3", the list will start with the letter "C", the third letter of the alphabet. Default is "1".
  - COMPACT - **(Depreciated)** The list is displayed with less room between elements.
- <LI> - Designates an item in the list.
  - TYPE="A" - Designates the numerical label type for use in the list. Values are A for capital letters, a for small letters, I for large Roman numerals, i for small Roman numerals, 1 for numerical integers.
  - VALUE="5" - Designates a new value for the list item which affects this value and those listed below it. This value must be a numerical value. If this value is set to "5" and the type is "I" the listed value will be the Roman number 5, "V".
- </OL> - Designates the end of an ordered list.

Ordered List Example

This is the HTML Code using depreciated attributes:

```
<ol type="I" start="5">
<li>This is the Roman number for 5.
<li>This is the Roman number for 6.
<li>This is the Roman number for 7.
<li tyle="A" value="10">This is the letter J, the tenth letter
in the alphabet.
<li type="1">This is the number 11.
<li>This is the Roman number for 12.
</ol>
```

This is the HTML Code using style attributes (CSS has no substitute for the START or VALUE attributes):

```
<ol style="list-style-type='upper-roman'" start="5">
<li>This is the Roman number for 5.
<li>This is the Roman number for 6.
<li>This is the Roman number for 7.
<li style="list-style-type='upper-alpha'" value="10">This is
the letter J, the tenth letter in the alphabet.
<li style="list-style-type='decimal'">This is the number 11.
<li>This is the Roman number for 12.
</ol>
```

And this is how it looks:

5. This is the Roman number for 5.
6. This is the Roman number for 6.
7. This is the Roman number for 7.
10. This is the letter J, the tenth letter in the alphabet.
11. This is the number 11.
12. This is the Roman number for 12.

Unordered Lists

Tags and Attributes

- <UL> - Designates the start of an unordered list
  - TYPE=DISC - Designates the shape used for the bullet. Values are CIRCLE, DISC, and SQUARE. The default of the first embedded list is CIRCLE.

    - COMPACT - **(Depreciated)** The list is displayed with less room between elements.
- <LI> - Designates an item in the list.
    - TYPE=SQUARE - Designates the shape of the bullet for this item.
- </UL> - Designates the end of an ordered list.

Unordered List Example

This is the HTML Code using depreciated attributes:

```
<ul type="square">
<li>Item 1.
<li>Item 2.
<li>Item 3.
<li type="disc">Item 4.
<li>Item 5.
<li>Item 6.
</ul>
```

This is the HTML Code using style attributes:

```
<ul style="list-style-type='square'">
<li>Item 1.
<li>Item 2.
<li>Item 3.
<li style="list-style-type='disc'">Item 4.
<li>Item 5.
<li>Item 6.
</ul>
```

And this is how it looks:

- Item 1.
- Item 2.
- Item 3.
- Item 4.
- Item 5.
- Item 6.

If these items show up as the same on your browser, it is probably because your browser is too old to support these type shapes which were added with HTML 3.2.

Definition Lists

Tags and Attributes

- <DL> - Designates the start of a definition list.
  - COMPACT - **(Depreciated)** The list is displayed with less room between elements.
- <DT> - Designates a definition term.
- <DD> - Designates a definition to go with the definition term.
- </DL> - Designates the end of a definition list.

Definition list example

This is the HTML code:

```
<dl><lh>Team Members
 <dt><b>Team Leader</b>
   <dd>Sarah Smith
 <dt><B>Programmer</b>
   <dd>Jim Jenkins
 <dt><B>Engineer</b>
   <dd>Jeff Jones
 <dt><B>Technician</b>
   <dd>Bill Brown
</dl>
```

And this is how it looks:

Team Members
Team Leader
Sarah Smith
Programmer
Jim Jenkins
Engineer
Jeff Jones
Technician
Bill Brown

HTML Tables

HTML Tables are arranged up and down by rows and left and right by columns which are filled by data cells. A cell may be specify to occupy one or more table rows or columns.

Tags and Attributes

- <TABLE> - Designates the start of a table.
  - ALIGN="left" - **Depreciated** Sets the table alignment. Possible values are left, center, or right.
  - BORDER="5" - Sets the width of the table border in pixels. The default is 0.
  - CELLSPACING="2" - Sets the amount of space in pixels between the cells that are in the table.
  - CELLPADDING="3" - Sets the space in pixels between the edges of each cell and the contents of the cell.
  - DATAPAGESIZE="20" - Can be used to set the size of a table page when the length of the table may exceed content that can fit on one page.
  - WIDTH="50%" - Sets the width of the table on the page in pixels or a percent of the page from 1% to 100%.
  - RULES="ROWS" - Turns on lines around elements of tables. Possible values are NONE, ROWS, COLS, GROUPS (borders between rows), or ALL. The default is NONE.
  - BGCOLOR="#0000FF" - **Depreciated** Sets the background color of all cells unless otherwise specified in a table row or table data cell definition.
  - BACKGROUND="flowers.gif" - Sets a background image like a wallpaper to be used in the table. This works on some web browsers.
  - SUMMARY="To compare values" - The reason for the table.
  - FRAME - Determines which ot the table outer borders are visible. Possible values are above, below, border, box, hsides (left and right), lhs (left only), rhs, vsides (top and bottom), or void.
- <CAPTION> - Must be first in the table before other elements. Must have the </CAPTION> closing tag.
- <COLGROUP> - Sets the table up in column groups essentially dividing one or more columns into several groups. Use after CAPTION and before THEAD. Does not have a closing tag. Attributes:
  - ALIGN - Possible values are center, left, right, char (Aligns the cell contents on the character specified by the CHAR attribute), and justify.
  - CHAR - Alignment character for cells. The character the cell is to be aligned to.
  - CHAROFF - Offset alignment character. Default is a period.
  - SPAN - Defines how many columns are in the group.

- VALIGN - Values are baseline, bottom, middle, and top.
- WIDTH - The column width in pixels or a percent of the table width.

- <THEAD>
  - ALIGN - Possible values are center, left, right, char (Aligns the cell contents on the character specified by the CHAR attribute), and justify.
  - CHAR - Alignment character for cells. The character the cell is to be aligned to.
  - CHAROFF - Offset alignment character. Default is a period.
  - VALIGN - Values are baseline, bottom, middle, and top.
- <TBODY>
  - ALIGN - Possible values are center, left, right, char (Aligns the cell contents on the character specified by the CHAR attribute), and justify.
  - CHAR - Alignment character for cells. The character the cell is to be aligned to.
  - CHAROFF - Offset alignment character. Default is a period.
  - VALIGN - Values are baseline, bottom, middle, and top.
- <TFOOT>
  - ALIGN - Possible values are center, left, right, char (Aligns the cell contents on the character specified by the CHAR attribute), and justify.
  - CHAR - Alignment character for cells. The character the cell is to be aligned to.
  - CHAROFF - Offset alignment character. Default is a period.
  - VALIGN - Values are baseline, bottom, middle, and top.
- <TH> - Designates the start of a table header. Table headers use the same attributes as data cells.
  - ABBR="SHORT" - Text is entered for this attribute that contains a short version of the cell content for when space is limited.
  - AXIS - A categorization of a table cell. This is a list of categories separated by commas.
  - ALIGN="RIGHT" - Sets horizontal alignment of the contents of the cell as LEFT, RIGHT, or CENTER. The default is LEFT.
  - VALIGN="TOP" - Sets the vertical alignment of the contents of the cells in the row as TOP, MIDDLE, or BOTTOM. The default is MIDDLE.
  - CHAR - Alignment character for cells. The character the cell is to be aligned to.
  - CHAROFF - Offset alignment character. Default is a period.
  - ROWSPAN="3" - Sets the number of rows this cell will occupy.
  - COLSPAN="2" - Sets the number of columns this cell will occupy.
  - HEADERS - A list of header cells separated by spaces that define the headers that apply to the TH element.
  - SCOPE - Values are column, row, colgroup, and rowgroup.
  - NOWRAP="NOWRAP" - **Depreciated** Sets the data cell so the contents will not wrap to another line.
  - HEIGHT **Depreciated** - Cell height in pixels.
  - WIDTH="10%" - **Depreciated** Sets the width of this column of cells expressed as a percentage of the table width or as a number of pixels.
  - BGCOLOR="white" - **Depreciated** Sets the background color of this cell.
- </TH> - Designates the end of a table header.
- <TR> - Designates the start of a table row.

- ALIGN="CENTER" - Sets horizontal alignment of the contents of the cells in the row as LEFT, RIGHT, or CENTER. The default is LEFT.
- CHAR - Alignment character for cells. The character the cell is to be aligned to.
- CHAROFF - Offset alignment character. Default is a period.
- VALIGN="TOP" - Sets the vertical alignment of the contents of the cells in the row as TOP, MIDDLE, or BOTTOM. The default is MIDDLE.
- BGCOLOR="blue" - **Depreciated** Sets the background color of cells that are contained in the row.
- BACKGROUND="ice.gif"- Sets a background image like a wallpaper to be used in the row. This works on some web browsers.

- </TR> - Designates the end of a table row.
- <TD> - Designates the start of a table data cell.
  - ABBR="SHORT" - Text is entered for this attribute that contains a short version of the cell content for when space is limited.
  - AXIS - A categorization of a table cell. This is a list of categories separated by commas.
  - ALIGN="RIGHT" - Sets horizontal alignment of the contents of the cell as LEFT, RIGHT, or CENTER. The default is LEFT.
  - VALIGN="TOP" - Sets the vertical alignment of the contents of the cells in the row as TOP, MIDDLE, or BOTTOM. The default is MIDDLE.
  - CHAR - Alignment character for cells. The character the cell is to be aligned to.
  - CHAROFF - Offset alignment character. Default is a period.
  - ROWSPAN="3" - Sets the number of rows this cell will occupy.
  - COLSPAN="2" - Sets the number of columns this cell will occupy.
  - HEADERS - A list of header cells separated by spaces that define the headers that apply to the TH element.
  - SCOPE - Values are column, row, colgroup, and rowgroup.
  - NOWRAP="NOWRAP" - **Depreciated** Sets the data cell so the contents will not wrap to another line.
  - HEIGHT **Depreciated** - Cell height in pixels.
  - WIDTH="10%" - **Depreciated** Sets the width of this column of cells expressed as a percentage of the table width or as a number of pixels.
  - BGCOLOR="white" - **Depreciated** Sets the background color of this cell.
  - BACKGROUND="ice.gif" - Sets a background image like a wallpaper to be used in the cell. This works on some web browsers.
- </TD> - Designates the end of a table data cell.
- </TABLE> - Designates the end of a table.

An example table

This is the HTML code using depreciated attributes:

```
<table cellspacing="5">
<tr align="center" bgcolor="#B0B0FF"><td>ROW 1, COL 1</td><td>ROW 1, COL 2</
td><td>ROW 1, COL 3</td></tr>
<tr align="center" bgcolor="#B0B0FF"><td colspan="3">ROW 2, COL 1-3</td></tr>
<tr align="center" bgcolor="#B0B0FF"><td colspan="2">ROW 3, COL 1-2</td><td>ROW
3, COL 3</td></tr>
<tr align="center" bgcolor="#B0B0FF"><td rowspan="3">ROW 4-6, COL 1</
tdD><td>ROW 4, COL 2</td><td>ROW 4, COL 3</td></tr>
<tr align="center" bgcolor="#B0B0FF"><td>ROW 5, COL 2</td><td>ROW 5, COL 3</
td></tr>
<tr align="center" bgcolor="#B0B0FF"><td>ROW 6, COL 2</td><td>ROW 6, COL 3</
td></tr>
</table>
```

This is the HTML code using style attributes:

```
<table cellspacing="5">
<tr style="text-align='center'; background-color='#B0B0FF'"><td>ROW 1, COL 1</
td><td>ROW 1, COL 2</td><td>ROW 1, COL 3</td></tr>
<tr style="text-align='center'; background-color='#B0B0FF'"><td colspan="3">ROW 2, COL
1-3</td></tr>
<tr style="text-align='center'; background-color='#B0B0FF'"><td colspan="2">ROW 3, COL
1-2</td><td>ROW 3, COL 3</td></tr>
<tr style="text-align='center'; background-color='#B0B0FF'"><td rowspan="3">ROW 4-6,
COL 1</tdD><td>ROW 4, COL 2</td><td>ROW 4, COL 3</td></tr>
<tr style="text-align='center'; background-color='#B0B0FF'"><td>ROW 5, COL 2</
td><td>ROW 5, COL 3</td></tr>
<tr style="text-align='center'; background-color='#B0B0FF'"><td>ROW 6, COL 2</
td><td>ROW 6, COL 3</td></tr>
</table>
```

This is how it looks:

ROW 1, COL 1	ROW 1, COL 2	ROW 1, COL 3
ROW 2, COL 1-3		
ROW 3, COL 1-2		ROW 3, COL 3
	ROW 4, COL 2	ROW 4, COL 3
ROW 4-6, COL 1	ROW 5, COL 2	ROW 5, COL 3
	ROW 6, COL 2	ROW 6, COL 3

This table sets the row colors independently for effect rather than setting the background color for the entire table.

Another example table

This is the HTML code:

```
<tr><th colspan="3"> This is a list of example Linux commands</th></tr>
<table rules="rows">
<tr><td nowrap="nowrap">find /home -user mark</td><td width="20"></td><td>Will find every file under the directory /home owned by the user mark.</td></tr>
<tr><td nowrap="nowrap">find /usr -name *spec</td><td width="20"><td><td>Will find every file under the directory /usr ending in ".spec".</td></tr>
<tr><td nowrap="nowrap">find /var/spool -mtime +40</td><td width="20"></td><td>Will find every file under the directory /var/spool that has data older than 40 days.<td><tr>
</table>
```

The HTML attributes, NOWRAP, and WIDTH are depreciated but I can't get the style attributes of WHITESPACE and BORDER-WIDTH to work here as replacements.

This is how it looks:

This is a list of example Linux commands	
find /home -user mark	Will find every file under the directory /home owned by the user mark.
find /usr -name *spec	Will find every file under the directory /usr ending in ".spec".
find /var/spool -mtime +40	Will find every file under the directory /var/spool that has data older than 40 days.

This table uses the nowrap="nowrap" attribute to keep the first cell in each row from wrapping. It also used a width attribute with value of 20 and no text in the cell to allocate some blank space between column 1 and column 3.

TBODY Tag

Technically the TBODY tag is to be used one or more times inside each occurance of the TABLE element. However, in many cases it may not be used. The THEAD and TFOOT tags may be optionally used one time inside a TABLE element.

HTML Attributes

Common attributes are applicable to many or most HTML elements. These attributes allow additional setting of style or other characteristics to the element. The attributes outlined on this page are common in many HTML elements.

ID attribute

Used as a unique identifier for elements for style sheets. In HTML page:

```
<hr id="firsthr">
```

In style sheet:

```
hr#firsthr { color: #80b0ff; height: 15; width: 100% }
```

CLASS attribute

Used to specify similar attributes for dissimilar elements by putting them in the same class. In HTML page:

```
<p class="indent">
```

In style sheet:

```
.indent { margin-right: 5%; margin-left: 5%; color: #0000ff }
```

STYLE attribute

Used to set various element attributes to specific values. Example:

```
<h1 style="text-align: center">Common Attributes</h1>
```

TITLE attribute

A tool tip on visual browsers.

LANG attribute

An international attribute which is used to set the international language type.

lang="en"

DIR attribute

An international attribute which is used to set whether text is read left to right or right to left.

dir="ltr"

Reserved Attributes

There are also reserved attributes used with the following elements. Reserved attributes may be used to bind HTML documents to XML documents. Elements that use reserved attributes:

- BUTTON
- DIV
- INPUT
- OBJECT
- SELECT
- SPAN
- TABLE
- TEXTAREA

The reserved attributes are:

- datasrc - Specifies the source of data using a URI.
- datafld - Specifies a column or property name.
- dataformatas - Specifies how the data is to be formatted with possible values being "html" or "text".

HTML Formatting

As you may have seen already, there are many ways to format text in HTML. Besides paragraphs, here are two main commands for managing text in block form.

- pre - Used to direct the browser to display a section of text exactly as it is typed with spaces included.
- blockquote - Used to format a block of text as a long quote indented more than the surrounding text on the left and the right.

The PRE element

PRE element attributes

- WIDTH="20" - **(Depreciated).** The WIDTH attribute specifies the number of characters across the screen to display. This is very rarely used.

PRE element example

The PRE element is handy for showing program code or part of file content as in this example. This is the HTML code:

```
Edit the file "/etc/auto.misc" adding lines like:
<pre>
      cd        -fstype=iso9660,ro      :/dev/cdrom
      fl        -fstype=auto            :/dev/fd0
</pre>
```

This is how it looks:

Edit the file "/etc/auto.misc" adding lines like:

```
      cd        -fstype=iso9660,ro      :/dev/cdrom
      fl        -fstype=auto            :/dev/fd0
```

The BLOCKQUOTE element

This element may contain other block elements such as headers , paragraphs, preformat blocks and tables. It should not be contained inside a paragraph element. It should not be used to produce indentation. Style sheets should be used possibly with the CLASS attribute to create indentation. Attribute:

- cite="uri" - Specifies the URI of the source of the quote.

BLOCKQUOTE element example

This is the HTML code:

```
<p>
The preamble of the Constitution says:<br>
</p>
<blockquote>
We the People of the United States, in Order to form a more perfect Union, establish Justice, insure domestic Tranquility, provide for the common defence, promote the general Welfare, and secure the Blessings of Liberty to ourselves and our Posterity, do ordain and establish this Constitution for the United States of America.
</blockquote>
```

This is how it looks:

The preamble of the Constitution says:

> We the People of the United States, in Order to form a more perfect Union, establish Justice, insure domestic Tranquility, provide for the common defence, promote the general Welfare, and secure the Blessings of Liberty to ourselves and our Posterity, do ordain and establish this Constitution for the United States of America.

The following code will produce the same effect using style sheets. In the HTML file:

```
<p>
The preamble of the Constitution says:<BR>
</p>
```

```
<p class="indent">
We the People of the United States, in Order to form a more perfect Union, establish Justice,
insure domestic Tranquility, provide for the common defence, promote the general Welfare,
and secure the Blessings of Liberty to ourselves and our Posterity, do ordain and establish this
Constitution for the United States of America.
</p>
```

In the style sheet:

```
.indent { margin-right: 5%; margin-left: 5%; color: #0000ff }
```

HTML Line Formatting

There are several methods of managing lines which are:

- BR - Line break. Start a new line.
- HR - Used to draw lines across the page

The BR element

An example of use of the BR element

The BR element has no ending tag. Here are two lines as they may be written without using a <br> tag.

This is line 1.
This is line 2.

This is what you would see:

This is line 1. This is line 2.

Here are the same two lines as they are written with a <br> tag.

This is line 1.
This is line 2.

This is what you would see:

This is line 1.
This is line 2.

The HR element

HR Element Attributes

The HR element has no ending tag.

- SIZE=7 - **(Depreciated).** Sets the thickness or size of the line in pixels.
- ALIGN="RIGHT" - **(Depreciated).** Sets the alignment of the line on the page to LEFT, RIGHT, or CENTER. The default is CENTER. The alignment is without purpose if the line width is 100%.
- WIDTH="50%" - **(Depreciated).** Sets the width of the line across the page as a % or in pixels. The default is 100%.
- COLOR="green"; - **(Depreciated).** Sets the color of the line. It may be expressed as one of the words, red, blue, green, etcetera, but may also be expressed in the #RRGGBB color format.
- NOSHADE - **(Depreciated).** A boolean value to set the line to a solid line.

An example using the HR element

Here is how the <hr> tag appears as HTML code:

```
Above the line.
<hr>
Below the line.
```

Here is how it looks:

Above the line.

Below the line.

The line, above, is set according to the defaults or the style sheet associated with this HTML document. To change its characteristics, I only need to change the style sheet. **Note:** Attempting to change the line characteristics using depreciated attributes such as ALIGN, WIDTH, or COLOR will not override values set by the style sheet. Only the use of the STYLE attribute will override those values.

Here's a colored line in HTML code (The top coding example uses HTML 3.2 without embedded style commands and the bottom coding uses HTML 4.01 with embedded style commands:

```
<hr size="15" align="left" width="50%" color="blue">
<hr style="color: 'blue'; height: '15'; text-align: 'left'; width: '50%'">
```

Here's how it looks:

Here's another colored line in HTML code (The second line uses style commands to set attributes rather than directly using some of the older attributes that are being depreciated):

```
<hr size="15" align="left" width="100%" color="#FF0000">
<hr style="height: '15'; text-align: 'left'; color: '#FF0000'; width: '100%'">
```

Here's how it looks:

Here's a line in white:

Shucks! The background.

HTML Text Formatting

There are many tags used to format text. They are listed below:

- .. - Sets bold text.
- <big>..</big> - Sets larger than normal text.
- .. - Sets text in italics and denotes emphasis.
- .. - Sets text in italics.
- <small>..</small> - Makes text smaller than normal.
- <strike>..</strike> - Draws a line through the text as a "strikeout".
- .. - Same as bold but denotes strong emphasis.
- <super>..</super> -Superscript.
- <tt>..</tt> - Monospaced typewriter font.
- <u>..</u> - Underlined text.
- <var>..</var> - Mark a variable.

Examples

Here's the HTML code:

```
This is an example of the <b>&#60;b&#62; tag </b>.<br>
This is an example of the <big>&#60;big&#62; tag</big>.<br>
This is an example of the <em>&#60;em&#62; tag</em><br>
This is an example of the <i>&#60;i&#62; tag</i>.<br>
This is an example of the <small>&#60;small&#62; tag</small>.<br>
This is an example of the <strike>&#60;strike&#62; tag</strike>.<br>
This is an example of the <strong>&#60;strong&#62; tag<strong>.<br>
This is an example of the <sup>&#60;sup&#62; tag<sup>.<br>
This is an example of the <sub>&#60;sub&#62; tag<sub>.<br>
This is an example of the <tt>&#60;tt&#62; tag<tt>.<br>
This is an example of the <u>&#60;u&#62; tag</u><br>
This is an example of the <var>&#60;var&#62; tag</var><br>
```

Here's how it looks:

This is an example of the ** tag**.
This is an example of the <big> tag.
This is an example of the * tag*.

This is an example of the *<i> tag*.
This is an example of the <small> tag.
This is an example of the <strike> tag.
This is an example of the ** tag**.
This is an example of the <sup> tag.
This is an example of the <sub> tag.
This is an example of the `<TT> tag`.
This is an example of the <u> tag.
This is an example of the *<var> tag*.

Other tags associated with text are:

- ABBREV - Denotes an abbreviation
- ACRONYM
- AU - Author
- CODE - Denotes program code and should be set in the same format as the PRE tag, but does not work either with all browsers or all HTML versions.
- DEL - Denotes deleted text.
- DFN - Denotes the definition of a term
- INS - Denotes inserted text
- KBD - Text to be typed at the keyboard, such as a computer command. The test is displayed in a mono spaced format.
- PERSON
- Q - Quotation.
- SAMP - Denotes sample text.

Each of these tags require an ending tag.

The FONT tag

The font element is being **depreciated and replaced by style sheet attributes**. The <FONT> element is used to set a section of text with a specific font. It uses the <FONT> tag to begin and the </FONT> for the end tag.

FONT Attributes and Tags

- <FONT> - Begins the FONT element.
  - SIZE="4" - Sets the font size using a value, between 1 and 7. The default value is 3. It can be specified with a "+n" value to set the size relative to the current size.

    - COLOR="blue" - Sets the text color.
    - FACE="roman" - The font name to be used. If there is more than one font name separated by commas the first font that can be found is used.
- </FONT> - Ends the FONT element.
- <BASEFONT> - Used to set the default font size on the current page.
    - SIZE="2" - Specifies default font size with a value between 1 and 7.

HTML SPAN and DIV Elements

There are a few elements that support additional formatting capabilities both for inline elements and block elements. These include the DIV and SPAN element. The SPAN element is used in line with text to modify the characteristics of a small section of text. The DIV element is very useful for adding certain block element style characteristics to inline or non-block elements.

The DIV Element

This element is useful for adding style to elements that cannot have specific characteristics set otherwise without using depreciated attributes. For example, the CENTER element is being depreciated in HTML 4.0, and there is no inline element attribute in style sheets that allows the inline elements such as anchors and images to be centered. Therefore the anchor code that follows is not centered.

Here is the code:

```
<a href="index.html" target="_top">HTML Guide Contents Page</a>
```

Here is the effect:

HTML Guide Contents Page

The following code centers the anchor:

```
<div class="center"><a href="index.html" target="_top">HTML Guide Contents Page</a></
div>
```

Here's the effect:

HTML Guide Contents Page

What is shown here is a simple, short example. Of course the came effect can be obtained by placing the anchor inside a paragraph element and setting the "text-align" value to "center", however the DIV element can be used to apply attributes to larger sections of the HTML page. Several paragraphs, headers, and lists may be contained inside the DIV element which is not possible using the paragraph elements.

The SPAN Element

The SPAN element does not have any unique attributes, only the common attributes. This is HTML code for an unordered list showing how to make the text and bullets a different color using the SPAN element.

```
<ul style="color: 'red'">
<li><span style="color: 'green'">Line 1</span>
<li><span style="color: 'green'">Line 2</span>
<li><span style="color: 'green'">Line 3</span>
</ul>
```

This is how it looks:

- Line 1
- Line 2
- Line 3

This is the HTML code for a paragraph demonstrating use of the STYLE attribute along with the SPAN element.

```
<p style="text-align: 'center'; font: '16pt courier'; color: 'blue'">
The color of the sky is blue, but if you look at the <span style="color: 'green'">trees, they are green</span>. This effect is produced using the &#60span&#62 element with the style ="color: 'green'" attribute set.
</p>
```

Here is how it looks:

The color of the sky is blue, but if you look at the trees, they are green. This effect is produced using the element with the style ="color: 'green'" attribute set.

HTML Characters

There are some characters in HTML that must be displayed using a string of characters. The string may be a name string or numeric string as in the table below.

Character	Description	Name string	Numeric string
&	Ampersand	&amp;	&#38;
©	Copyright	&copy;	&#168;
"	Quotation	&quot;	&#34;
>	Greater than	&gt;	&#62;
<	Less than	&lt;	&#60;
¢	Cent	&cent;	&#162;
£	Sterling pound	&pound;	&#163;
°	Degree sign	&deg;	&#176;
®	Trademark sign	&reg;	&#174;
±	Plus/minus signs	&plusmn;	&#177;
×	Multiply sign		&#215;
÷	Divide sign		&#247;
¦	Broken vertical bar	&brvbar;	&#166;
§	Section sign	&sect;	&#167;
¹	Superscript 1	&sup1;	&#185;
²	Superscript 2	&sup2;	&#178;
³	Superscript 3	&sup3;	&#179;
½	1/2 fraction	&frac12;	&#189;
¼	1/4 fraction	&frac14;	&#188;
¾	3/4 fraction	&frac34;	&#183;
Æ	Capital AE	&AElig;	&#198;
æ	Small AE	&aelig;	&#230;
É	Capital E accented	&Eacute;	&#201;
é	Small e accented	&eacute;	&#233;

Although these special characters may displayed without the semicolon at the end, it is the proper standard to use the semicolon.

Style Sheets

Purpose of Style Sheets

The purpose of style sheets is to separate the content of the HTML documents from their style. This makes control over the style much easier and group efforts easier since content of an entire set of HTML pages can be easily controlled using one or more style sheets.

STYLE sheets or the inline STYLE element will allow you to set custom margin sizes, and the text font, color and sizes of various elements that are used on your web page. Size of margins can be set in inches (in), pixels(px), percentages (%), centimeters (cm) or the (em). The unit of em is the em is the height of the font. Two em units is twice the height of the font. When the em units are used, everything is scaled relative to the font so the scaling will remain the same regardless of the font used.

Methods of Including Style

Methods of including style content in an HTML document:

- Embedding - Use the STYLE element in the HEAD element to create a style sheet that is embedded in the header. Example:

```
<style type="text/css" MEDIA=screen>
<!--
   body {background-color: "#ffffff"; color: "#000000";
background: url('../../../../gifs/flowers.gif') }
   a:link { color: "#0000ff" }
   a:visited { color: "#7f007f" }
   a:active { color: "#ff0000" }
   h3 { color: "#600060" }
-->
</style>
```

- Linking - Use the link element in the HTML header to link an external text file style sheet to the HTML document. Example:

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

- Importing - One style sheet may be imported from another by using the "@import" command in

the style sheet that is performing the import.

- Inline - Style information may be included with each individual element using the STYLE attribute which is common to all elements. See the "Common Attributes" section. Many examples in this document show style being controlled using the STYLE attribute. Although this is acceptable for specific elements whose characteristics must be different such as the color of a horizontal line, it is wiser to control most element style using the external style sheets.

The STYLE Element

Attributes:

- Type - Defines the content type such as "text/css".
- Media - Defines the intended media the page will be displayed on. Values include all, aural, braille, handheld, print, projection, screen, tty, and tv.
- Title - Gives the title sheet an optional title.

Example Embedded Style Sheet

When setting document style, place the STYLE element with the <STYLE> beginning tag and </STYLE> ending tag between the HEAD and the BODY elements. Therefore placement would be as follows:

```
<html>
<head>
<title>Example Style Settings</title>
</head>
<style type="text/css">
<!--
body {background: #FFFFFF; color: #000000; margin-top: 6%;
margin-left: 3%; margin-right: 3%}
h1 {font: 14pt ariel; color: #0000FF}
h2, h3, h4 {margin-left: -3%}
p {font: 12pt roman; text-indent: 0.5in}
a {color: #00FF00; text-decoration: none}
-->
</style>
```

Other settable parameters include:

- font-style: italic
- font-weight: bold
- font-size: 200% - Indicates twice the size of normal text.
- font-family: roman, serif, "Times New Roman"
- text-transform: uppercase
- background-color: blue

The section "CSS Properties" contains a more complete list of parameters and the elements they work with.

Setting an HTML Page with a Style Sheet

To set an HTML page up to use a style sheet, a link must be provided in the header section as follows:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<meta name="GENERATOR" content="Arachnophilia 3.9">
<meta name="description" content="The Computer Technology Documentation HTML
Guide - Setting Document Style">
<meta name="keywords" content="Documentation, HTML, html document style, style,
setting document style, setting html document style">
<title>The CTDP HTML Guide - Setting Document Style</title>
<link href="style.css" rel="stylesheet" type="text/css">
</head>
<body>
.
.
.
</body>
```

The Style Sheet File (style.css)

```
body {background-color: #ffffff; color: #000000; background: url('../../../../gifs/flowers.gif') }
a:link { color: #0000ff }
a:visited { color: #7f007f }
```

```
a:active { color: #ff0000 }
h1, H2 { text-align: center }
h3 { color: #600060 }
blockquote { color: #0000ff }
pre { color: #0000ff }
.indent { margin-right: 5%; margin-left: 5%; color: #0000ff }
.center { text-align: center }
.firstindent { text-indent: 2em}
div.outline { border-top: solid medium navy }
hr { color: #80b0ff; height: 5; width: 50%; text-align: center }
hr#firsthr { color: #80b0ff; height: 15; width: 100% }
```

This style sheet is used to set the style for various HTML elements on every page of this document. To change the style, only the style sheet needs to be changed. Note that the style sheet sets the wallpaper for the HTML page along with the background color, foreground color, active link color, visited link color, and unvisited link color(A:link). It also sets colors for the BLOCKQUOTE and PRE elements along with defining some CLASS characteristics and one ID characteristic.

Using the CLASS Attribute

The CLASS and ID attributes allow very convenient and precise control of the style of specific elements in HTML documents. The CLASS attribute allows attributes to be assigned to almost any element based on a class name. Class style information is specified in the style sheet with the period, ".", followed by the class name as in the following example:

```
.indent { margin-right: 5%; margin-left: 5%; color: #0000ff }
```

Note that this same line is used in the example style sheet above. To use this in the HTML file, when the beginning tag of the element is declared, the class is specified as in the following statement:

```
<p class="indent">
```

This causes the paragraph element to have the style characteristics assigned to the class "indent" so long as those style characteristics are appropriate to the element they are being applied to. In this case the paragraph element is indented much like the blockquote element and its color is also set to blue. Classes may also be assigned to specific elements so they can only be used with that elements. Assingments in the style sheet to specific elements similar to "PRE.center { text-align: center }".

Using the ID Attribute

The ID attribute is useful since it allows you to specify exactly which element will have the desired style. This is done with a line like the following as shown in the style sheet example above.

```
hr#firsthr { color: #80b0ff; height: 15; width: 100% }
```

When the first horizontal line is declared on the HTML page it is declared as follows:

```
<hr id="firsthr">
```

The line for the code in this example is at the top of this page. All other lines on the page appear as defined by the following entry from the above style sheet:

```
hr { color: #80b0ff; height: 5; width: 50%; text-align: center }
```

The line appears below and in several other places on this page. Please note that ID selectors are not required to be associated with a specific element in the style sheet although I think it makes more sense to do so.

Context Selection

Allows elements containing inline elements to have special assigned values as in the following example:

```
table b ( color: red }
```

This example sets the color of any bold text in a table to red.

Grouping Elements for Style Characteristics

The example above contains the following line:

```
h1, h2 { text-align: center }
```

This line allows style characteristics to be assigned to both elements H1 and H2 at the same time. The elements are separated by commas.

Pseudo Classes

Some elements have specific automatially assigned classes. For example the anchor (A) element has the classes link, visited, and active. The following lines in the style sheet assign color style characteristics to these classes.

```
a:link { color: #0000ff }
a:visited { color: #7f007f }
a:active { color: #ff0000 }
```

Note that the element is separated from the pseudo name using a colon. Also a "first-line" and a "first-letter" pseudo class exists for each block element.

CSS Properties

This section only provides a quick reference for the CSS properties, for complete details refer to the Web Design Group's web page CSS section.

Application of these properties are based on the element type as described earlier in the "Element Categories" section.

Box Properties

Name	Description	Possible Values	Element Types	Example	inherited
border	Sets border width, style and color.	See border width, style, and color	All	{border: medium solid green}	No
border-bottom	Set the bottom border width style, and color.	See border width, style, and color	All	{border-bottom: medium solid green}	No
border-bottom-width	Set the bottom border width	thin, medium, thick, or length value	All	{border-bottom-width: thin}	No
border-color	Set the border color	A color value, color, #RRGGBB	All	{border: #0000ff}	No
border-left	Set the left border width style, and color.	See border width, style, and color	All	{border-left: medium solid green}	No
border-left-width	Set the left border width	thin, medium, thick, or length value	All	{border-left-width: thin}	No
border-right	Set the right border width style, and color.	See border width, style, and color	All	{border-right: medium solid green}	No
border-right-width	Set the right border width	thin, medium, thick, or length value	All	{border-right-width: thin}	No

border-style	Sets border style	none, dotted, dashed, solid, double, groove, ridge, inset, outset	All	{border: dashed}	No
border-top	Set the top border width style, and color.	See border width, style, and color	All	{border-top: medium solid green}	No
border-top-width	Set the top border width	thin, medium, thick, or length value	All	{border-top-width: thin}	No
border-width	Sets Border width.	thin, medium, thick, or length value	All	{border-width: 8}	No
clear	Determines where floating elements are allowed.	none, left, right, both	All	{clear: left}	No
float	Specifies how text is wrapped and where it is aligned.	none, left, right	All	{float: left}	No
height	Height of element	auto or a height value	Block elements and IMG, INPUT, TEXTAREA, SELECT, and OBJECT	{height: 200}	No
margin	Set element margin width. A single value sets all margins, two values set top and bottom, four values set top, right, bottom, and left margins.	auto, length value, or percent value	All	{margin: 4em 2em 4em 2em}	No
margin-bottom	Set element bottom margin	length value, or percent value	All	{margin-bottom: 4em}	No
margin-left	Set element left margin	length value, or percent value	All	{margin-left: 4em}	No
margin-right	Set element right margin	length value, or percent value	All	{margin-right: 4em}	No

margin-top	Set element top margin	length value, or percent value	All	{margin-top: 4em}	No
padding	Space between border and content. A single value sets all sides, two values set top/bottom and left/right, three values set top, right/left, and bottom, and four values set top, right, bottom, and left.	length value, or percent value	All	{padding: 4em}	No
padding-bottom	Space between bottom border and content.	length value, or percent value	All	{padding-bottom: 4em}	No
padding-left	Space between left border and content.	length value, or percent value	All	{padding-left: 4em}	No
padding-right	Space between right border and content.	length value, or percent value	All	{padding-right: 4em}	No
padding-top	Space between top border and content.	length value, or percent value	All	{padding-top: 4em}	No
width	Width of element	auto or a height value in length or percentage	Block elements and IMG, INPUT, TEXTAREA, SELECT, and OBJECT	{width: 40em}	No

---

Background and Color Properties

Name	Description	Possible Values	Element Types	Example	inherited

background	Set background color, repeat, image, attachment, or position.	See background-color, background-image, background-attachment, background-repeat, background-position	All	{background: #8080ff}	No
background-attachment	Determines if the background image is fixed or scroll.	scroll, fixed	All	{background-attachment: scroll}	No
background-color	Sets the background color.	Named or value color	All	{background-color: #8080ff}	No
background-image	Sets the background image.	url	All	{background-image: url('../../../../gifs/flowers.gif')}	No
background-position	Sets the background image initial position.	top, center, bottom, left, center, right, or percent values	Block and IMG, INPUT, TEXTAREA, SELECT, and OBJECT	{background-position: left top}	No
background-repeat	Sets how the background image is repeated.	repeat, repeat-x, repeat-y, no-repeat	All	{background-repeat: no-repeat}	No
color	Sets element color.	Named or value color	All	{color: green}	Yes

---

Classification Properties

Name	Description	Possible Values	Element Types	Example	inherited
display	Sets the type of element.	block, inline, list-item, none	All	{display: list-item}	No

list-style	Sets list style type and/or position.	See list-style-type and list-style-position	List-item	{list-style: circle}	Yes
list-style-image	Sets image to be used as the list item marker.	url	List-item	{list-style-image: url (this.gif)}	Yes
list-style-type	Sets list style type.	circle, disc, decimal, lower-alpha, lower-roman, none, square, upper-alpha, upper-roman	List-item	{list-style-type: square}	Yes
list-style-position	Sets where the marker is place relative to the text and its wrapping position.	inside, outside	List-item	{list-style: circle}	Yes
whitespace	Sets treatment of spaces inside the element.	normal, pre, nowrap	Block	{whitespace: pre}	Yes

---

Font Properties

Name	Description	Possible Values	Element Types	Example	inherited
font	Used to define font properties	See font-family, font-size, font-style, font-variant, and font-weight.	All	{font: 20pt}	Yes
font-family	Used to define font family to use	family name	All	{font-family: ariel roman}	Yes
font-size	Used to define font size to use	xx-small, x-small, small, medium, large, x-large, xx-large, larger, smaller, length value, or percent value	All	{font-size: 18pt}	Yes

font-style	Used to define font style to use	normal, italic, oblique	All	{font-style: italic}	Yes
font-variant	Used to determine whether to use normal or small caps	normal, small-caps	All	{font-variant: small-caps}	Yes
font-weight	Sets font weight.	normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800, 900	All	{font-weight: 600}	Yes

---

Text Properties

Name	Description	Possible Values	Element Types	Example	inherited
letter-spacing	Sets the space between characters.	normal or length value	All	{letter-spacing: 0.2em}	Yes
line-height	Sets the height of lines.	normal, a number, a percent of the element font size,	All	{line-height: 2}	Yes
text-align	Sets the alignment of text.	left, right, center, justify	Block	{text-align: center}	Yes
text-decoration	Sets the special decoration attributes of text.	none, overline, underline, line-through, blink	All	{text-decoration: blink}	No
text-indent	Sets the element's first line amount of indentation.	length or percentage value	Block	{text-indent: 5%}	Yes
text-transform	Transforms text to one of the set values.	none, capitalize, uppercase, lowercase	All	{text-transform: uppercase}	Yes
vertical-align	Sets vertical position.	baseline, sub. super, top, middle, bottom, text-top, text-bottom, or percent value	Inline	{vertical-align: sub}	No
word-spacing	Sets extra space between words.	normal or length value	All	{word-spacing: 0.2em}	Yes

HTML Graphics

The <IMG> tag is used to embed graphics in HTML pages. They may be embedded inside other elements such as anchors. When embedded inside an anchor, then the user left clicks on the image, they will go to the designated link (rather, their browser will load a file from the designated link). The <IMG> element has no ending tag.

IMG Attributes

- SRC="../greenhomebutton.gif" - The path and filename of the image which specifies its location.
- ALT="Home" - This is a message displayed if the image could not be found.
- ALIGN="TOP" - **(Depreciated).** Sets the image alignment. The image is aligned to the left or right column if the value is LEFT or RIGHT,The values, TOP, MIDDLE, BOTTOM, ABSMIDDLE, and ABSBOTTOM, set the vertical alignment with items in the same line.
- VSPACE=3 - **(Depreciated).** The space between the image and the text above and below it in pixels.
- HSPACE=5 - **(Depreciated).** The space between the image and the text to the left and right of it in pixels.
- BORDER=10 - **(Depreciated).** Sets a border of the specified width in pixels to be drawn around the image.
- HEIGHT=33 - The height of the image. If this is not specified, the image will be the size as determined by the gif file. This can be set in pixels or a percentage of the browser window height.
- WIDTH=115 - The width of the image. If this is not specified, the image will be the size as determined by the gif file. This can be set in pixels or a percentage of the browser window width.
- ISMAP - Identifies the image as an image map. The image map lets the user point and click different parts of the image to load other web pages.
- USEMAP - Specifies the client side image map file to be used.

The line break element, <BR> and the horizontal rule element <HR> may be placed inside the <IMG> element.

Example 1

The following code will display a gif file:

```
<img src="MyPicture.gif" alt="Outdoor Scene">
```

This code sets an alternate text of "Outdoor Sceen" in the event the "MyPicture.gif" file is not

displayable. This usually occur if it cannot be found or if the user has their browser controls set not to display pictures.

Example 2

In this example the additional feature is added that sets the size of the picture relative to the size of the browser window. The size of the picture is set to 20% of the height of the browser window and 40% of the width.

```
<img src="MyPicture.gif" alt="Outdoor Scene" height="20%" width="40%">
```

Example 3

If you are interested in displaying pictures that are of significant size which may be accessed for possible download for sharing with friends or family, this example gives some valuable advice. If you are interested in making several graphic files accessable from one page such as files scanned from pictures, it would be worthwhile to make a file of about one fifth the height and width of the original file, then make it a link to the original file for viewing or download. This is because the download time of your web page with many large graphic files may be excessive. In this example two smaller files are placed on a page next to each other. These files are links to the larger files to be viewed or downloaded.

```
<div style="text-align: center"
<a href="Picture1.gif"><img src="SmallPicture.gif" ALT="Picture1" height="20%"
width="40%"></a>
<a href="Picture2.gif"><img srcC="SmallPicture2l.gif" ALT="Picture2" height="20%"
width="40%"></a>
</div>
```

In this example the style attribute "text-align: center" is set. See the section on style and style sheet attributes for more information about this attribute.

Example 4

In this example some JavaScript code is added so the name or description of the picture is displayed when the viewer places their mouse cursor on the link.

```
<div style="text-align: center">
<a href="Picture1.gif" onMouseOver="window.status='Picture1' ;return true"
onMouseOut="window.status=';return true"><img src="SmallPicture.gif" alt="Picture1"
height="20%" width="40%"></a>
<a href="Picture2.gif" onMouseOver="window.status='Picture2' ;return true"
```

```
onMouseOut="window.status=';return true"><img src="SmallPicture2l.gif" alt="Picture2"
height="20%" width="40%"></a>
</div>
```

Example 5

The following code will display a linked image that will return you to the HTML start section:

```
<div style="text-align: center">
<a href="htmlintroduction.html" target="right" onMouseOver="window.status='To The
HTML Start (intro) Page' ;return true" onMouseOut="window.status=';return true"><img
SRC="bluestartbutton.gif" alt="Home" vspace="3" border="3" height="35" width="120"></
a>
</div>
```

This is the result:

Image areas and Maps

To use a web browser to click on various areas of an image file and go to a specific web page depending on the area clicked upon requires the use of the <MAP> and <AREA> elements. The <MAP> element has an ending tag, </MAP> while the <AREA> element does not.

MAP and IMAGE Tags and Attributes

- <MAP> - Starts a client-side image map. This must be referenced in an image <IMG> tag. This element will include <AREA> tags.
- </MAP> - Ends the image map element.
- <AREA> - Defines areas of an image map that the user may click on to establish a link.
    - SHAPE=RECT - Defines the shape of the click able area. Valid values are POLY, RECT, and CIRCLE.
    - COORDS="2,50,102,101" - Defines, in pixels, the top-left and bottom-right coordinates of the clickable linked region in the image.
    - HREF="leftwing.htm" - Defines the location of the web page that is loaded when the area in the image is clicked.

Example Code

```
<img src="airplane.gif" ismap usemap="#airplane.map">
</a> <map name="airplane.map">
<area shape="rect coordsS="2,50,102,101" HREF="leftwing.html">
<area shape="rect" coords="110,10,160,300" HREF="fusalage.html">
<area shape="rect" coords="162,50,262,101" HREF="rightwing.html"> </map>
```

HTML Sound

The BGSOUND element is used to play sounds on a web page. There is no end tag for the <bgsound> element. To play sounds all that is required is to put a line like the following on your web page:

```
<bgsound src="1234usmc.wav" loop="-1">
```

This line is used on this page to play sound.

BGSOUND element attributes

- SRC="1234usmc.wav" - The location or the URL of the WAV, MIDI, or AU file to play.
- LOOP="-1" - The number of times the file will will loop. A value of INFINATE or -1 will loop indefinitely.

HTML Links

HTML Anchors and Links

Anchors are used to specify link locations or to set anchors for locations to link to. This enables the person using the browser to click on a link (anchor) and their browser will go to the location pointed to by the anchor. Anchors begin with the <A> tag and end with the </A> tag.

Anchor Attributes

- NAME - Sets an anchor at a named location in the document for later use by an anchor tag.
- HREF="hw.html" - A hypertext reference which defines the name of the document or the named location in the document for the anchor link to send the web browser to.
- TARGET - Specifies the frame target the referenced page will be put into. The target may me a named window or one of the special reserved names below. If the target is a named window that exists the web page the link points to will be loaded into that window. If a window by that name does not exist, a new window with that name will be created.
    - _blank - The web linked page loads in a new window and does not name the window.
    - _parent - The new page will load in the parent frame or window.
    - _self - The new page will load in the same window that the link is in. If the link is in a frame, it will load inside the frame. This is the default target.
    - _top - The new page will load into the top window reguardless of how many framesets deep the link is embedded.

Examples

Example 1 - Named Anchor

At the top of this page is a named anchor which is used for other examples on this page. Its text is:

```
<a name="top"></a>
```

Example 2 - Link to another page

This is an example of HTML source code using an anchor link:

```
<a href="htmlintroduction.html" target="right">Introduction</a>
```

Here is how it looks:

Introduction

It will take you to the introduction page if you click on it.

Example 3 - A Mail Link

This example shows how to set up a mail link that invokes the installed mail program when it is clicked on. Here's the source code:

```
<a href="mailto: ctdp@operamail.com">ctdp@operamail.com</a>
```

Below is the implemented example. If you click on it it should invoke your mail program such as Microsoft Outlook so you can send an e-mail. Feel free to click on it to see the effect. You can send an e-mail telling us that the example worked or exit the e-mail program.

ctdp@operamail.com

Example 4 - Image Link

This is an example of HTML source code using an anchor link with a button image embedded (It also includes some JavaScript):

```
<a href="htmlgraphics.html" target="right" onMouseOver="window.status='To graphics
page' ;return true" onMouseOut="window.status=';return true"<>img src="bluestartbutton.
gif" alt="Graphics" vspace="3" border="0" height="35" width="120"></a>
```

Here's how it looks and works:

It will take you to the graphics page of this document. The JavaScript part includes the "onMouseOver" and onMouseOut" definitions. They are event handlers for the MouseOver and MouseOut events. The "window.status" definition is an attribute of the window object. You can read about JavaScript in the JavaScript Manual.

Example 5 - Local Page Link

Here's an example of HTML source code that will take you to the top of the page:

```
<a href="#top">Take me to the top of this page.</a>
```

Here's how it looks and works:

Take me to the top of this page.

Example 6 - Non-Underlined Link

Here's an example of HTML source code that will take you to the top of the page, but it is not underlined:

```
<a href="#top" style="text-decoration: none" >Take me to the top of this page.</a>
```

Here's how it looks and works:

Take me to the top of this page.

Example 7 - Hidden Link

Here's an example of HTML source code that will take you to the top of the page, but it is not underlined, nor is the color changed for a normal link color. It is also embedded with other text to make it harder to find. This link is essentially hidden and can only be found by placing the mouse directly over

the link.

Take me to the top of this page.

Here's how it looks and works:

Take me to the top of this page.

Example 8 - Target set to top

Take me to the main HTML page.

Here's how it looks and works:

Take me to the main HTML page.

Example 9 - Target set to self

Take me to the main HTML page.

Here's how it looks and works (Use the BACK button on your browser to return):

Take me to the main HTMLpage.

Example 10 - Target set to blank

Take me to the main HTML page.

Here's how it looks and works:

Take me to the main HTML page.

If you continue clicking on this link, it will continue making new pages.

HTML Forms

Forms allow those who come to your website to provide input to your webserver. Some reasons to use forms are to allow visitors to partake in polls or sign up as a member. Also orders for merchandise may be placed using forms. Usually the forms are supported by a client side script program which will be sure the form is properly filled out before it is sent to the server. Also the form will invoke a script program which runs on the server. This script program will receive the data and store it on the server, then respond with a message to the user telling them that the form was submitted.

This document is not intended to teach script writing, so it will concentrate on the HTML side of form writing.

Form tags and attributes

- <FORM> - Starts an input form. </FORM> - Ends an input form.
  - ACTION="..." - The address of the script to process this form input.
  - METHOD="POST" - The method used to send the data from the form to the server. The options are POST and GET. Normally POST is used.
- <BUTTON> - Used to define a FORM submit, reset, or push button. </BUTTON> - Ends the button. Attributes:
  - ACCESSKEY - A shortkey used to give the focus to the label.
  - DISABLED - A boolean value that makes the button unavailable for use.
  - NAME - Sets the name sent to the server as part of the name/value pair when the form is submitted.
  - ONBLUR - Sets a script program to run when the element loses focus.
  - ONFOCUS - Sets a script program to run when the element gets focus.
  - TABINDEX - Sets the tabbing order of the element.
  - TYPE - Sets the button type to one of submit, reset, or button.
  - VALUE - Sets the value of the button to be sent to the webserver when the form is submitted.
- <FIELDSET> - Allows grouping of form controls. </FIELDSET> ends the fieldset.
- <INPUT> - An input element for a form.
  - ACCEPT - The acceptable media types for file upload.
  - ACCESSKEY - A shortcut character.
  - ALIGN="CENTER" - **(Depreciated)**. Sets the way the text and image will align.
  - ALT - Alternate text for the image.
  - CHECKED - This indicates that the item is checked is it is a check box or radio button.
  - DISABLED - A boolean value that makes the input object unavailable for use.
  - MAXLENGTH="7" - The most characters that may be put in a text region.
  - NAME="RegForm" - The name of the form which is passed to the script running on the server.

- ONBLUR - Sets a script program to run when the element loses focus.
- ONCHANGE - Script to run when the element is changed.
- ONFOCUS - Sets a script program to run when the element gets focus.
- ONSELECT - Script to run when the element text is selected.
- SIZE="3" - The number of characters an input region should be filled with.
- SRC="bluebutton.gif" - The name of the source file for an image to be displayed on the form.
- TABINDEX - Sets the tabbing order of the element.
- TYPE="TEXT" - The type of input to be done. The options are BUTTON, CHECKBOX, FILE, HIDDEN, IMAGE, RADIO, RESET, SUBMIT, PASSWORD, or TEXT.
- VALUE="1" - Specifies the default value for a text item. It specifies the label of a reset or submit button or the value to be returned when a check box or radio button is selected.

- <LABEL> - Allows an ability to click on a text label to select a form control. <.LABEL> ends a label. Attributes:
  - ACCESSKEY - A shortkey used to give the focus to the label.
  - FOR - Sets the control associated with the label which ust match the control name.
  - ONBLUR - Sets a script program to run when the element loses focus.
  - ONFOCUS - Sets a script program to run when the element gets focus.
- <TEXTAREA> - Indicates an element that has multiple text lines. </TEXTAREA> - Ends a Text entry form element. Attributes:
  - ACCESSKEY - A shortkey used to give the focus to the element.
  - COLS="3" - The number of columns in the text area.
  - NAME="AddressLines" - The name of the textarea which is passed to the script running on the server.
  - ONBLUR - Sets a script program to run when the element loses focus.
  - ONCHANGE - Script to run when the element is changed.
  - ONFOCUS - Sets a script program to run when the element gets focus.
  - ONSELECT - Script to run when the element text is selected.
  - ROWS="3" - The number of rows in the text area.
  - TABINDEX - Sets the tabbing order of the element.
- <SELECT> - Creates a list of items that can be selected. </SELECT> ends the select area. Attributes:
  - DISABLED - A boolean value that makes the select object unavailable for use.
  - MULTIPLE - Allows more than one selection from the scrolling list.
  - NAME="state" - The name of the selection item that is passed to the script running on the server.
  - ONBLUR - Sets a script program to run when the element loses focus.
  - ONCHANGE - Script to run when the element is changed.
  - ONFOCUS - Sets a script program to run when the element gets focus.
  - ONSELECT - Script to run when the element text is selected.
  - SIZE="1" - The number of selections that will be displayed. If a value of SIZE is set, the selection will be a scrolling list. If no SIZE value is set the selection will be a pop-up menu.

    - TABINDEX - Sets the tabbing order of the element.
- <OPTGROUP> - Sets up choices inside a SELECT menu. </OPTGROUP> ends the group. Attributes:
    - DISABLED - A boolean value that makes the object unavailable for use.
    - LABEL - Describes available choices.
- <OPTION> - Used to set items in a list of selectable items. </OPTION> ends the OPTION element. This is found within the SELECT element.
    - DISABLED - A boolean value that makes the object unavailable for use.
    - LABEL
    - SELECTED - This will be the default value if this attribute is included in the OPTION element
    - VALUE="10" - This is the value to be submitted to the CGI script program if this option is selected when the user submits the form.

An example Form

```
<h2 style="text-align: center">Computer Technology Documentation
Project</h2>
<h2 style="text-align: center">Member Registration</h2>
<hr style="height: 5" WIDTH=90%>

<div style="text-align: center">
<table>
<td>
<pre>
<form name="RegistrationForm" method="post" action="/cgi-bin/response.
pl">
<b>First Name:   </b><input type="text" name="firstname">
<b>Last Name:   </b><input type="text" name="lastname">
<b>Address1:   </b><input type="text" name="addr1">
<b>Address2:   </b><input type="text" name="addr2">
<b>City:   </b><input type="text" name="city">
<b>State:   </b><select name="state" size="1">
        <option value="1">AL
        <option value="2">AK
        <option value="3">AR
        <option value="4">AS
        <option value="5">AZ
        <option value="6">CA
        <option value="7">CO
        <option value="8">CT
```

```
<option value="9">DC
<option value="10">DE
<option value="11">FL
<option value="12">FM
<option value="13">GA
<option value="14">GU
<option value="15">HI
<option value="16">ID
<option value="17">IA
<option value="18">IL
<option value="19">IN
<option value="20">KS
<option value="21">KY
<option value="22">LA
<option value="23">MA
<option value="24">MD
<option value="25">ME
<option value="26">MH
<option value="26">MI
<option value="28">MN
<option value="29">MO
<option value="30">MP
<option value="31">MS
<option value="32">MT
<option value="33">NC
<option value="34">ND
<option value="35">NE
<option value="36">NH
<option value="37">NJ
<option value="38">NM
<option value="39">NV
<option value="40">NY
<option value="41">OH
<option value="42">OK
<option value="43">OR
<option value="44">PA
<option value="45">PR
<option value="46">PW
<option value="47">RI
<option value="48">SC
<option value="49">SD
<option value="50">TN
<option value="51">TX
```

```
        <option value="52">UT
        <option value="53">VA
        <option value="54">VI
        <option value="55">VT
        <option value="56">WA
        <option value="57">WI
        <option value="58">WV
        <option value="59">WY
</select>
<b>Zip:         </b><input type="text" name="zip" maxlength="10"
size="10">
<b>Area Code/Phone:</b><input type="text"
name="code" maxlength="3" size="3">
<input type="text" name="phone" maxlength="7" size="7">

<div style="text-align: center">
<input type="button" name="cmdSubmit" value="Submit"> <input
type="reset"
 value="clear">
</div>
</pre>

</form>
</td>
</table>
</div>
</body>
```

How the Form Looks:

Computer Technology Documentation Project

Member Registration

First Name:

Last Name:

Address1:

Address2:

City:

State: --

Zip:

Area Code/Phone:

Submit Clear

HTML Script Embedding

This document is not intended to teach script writing, but will only define some of the related tags and give an example script so the user can get an idea how to embed script in HTML. This document also does not explain how to get information entered in a script form to the server. That is a subject that will be described when the hyper text transfer protol (HTTP) and server side script writing is discussed.

The script below will check inputs from the form, below, to be sure they have been entered correctly before sending the data to the server. If you attempt to press submit in the form below without filling the appropriate fields in, you will see that the script is active.

SCRIPT Element Attributes

The script element is usually embedded in the document header but may be placed in the head or the body of the HTML document so long as it occurs before code that invokes it.

- CHARSET - (Depreciated). The encoding of any external script specified such as "iso-8859-1".
- DEFER - (Depreciated). The script is not parsed until the HTML document is rendered.
- EVENT - Specifies the event used to trigger the script.
- FOR - Specifies an object (combined with an action) that is used to trigger the script event
- LANGUAGE - (Depreciated). Tells the name of the language the script is written in. Required for most browsers. Possible values are JavaScript, VBScript, and JavaScript1.1
- SRC - (Depreciated). Specifies an external script which may include a URL path to another webserver.
- TYPE - The new method to specify the script language using MIME specification rather than the LANGUAGE attribute. This attribute is not widely supported yet.

The NOSCRIPT element is used to provide alternate content when a browser does not support script.

Scripting Events

- ondblclick
- onclick
- onkeydown
- onkeypress
- onkeyup

- onmousedown
- onmousemove
- onmouseout
- onmouseover
- onmouseup

An example Script

How the script looks:

```
<!doctype html public "-//w3c//dtd html 4.0 transitional//en">
<head>
<title>New CTDP Member Registration</title>
<scrypt type="text/javascript" for="cmdSubmit" event="onclick"
language=JavaScript>
<!--

var MemSubForm;
MemSubForm = document.RegistrationForm;

if (MemSubForm.firstname.value.length == 0) {
   MemSubForm.firstname.value = prompt("Please enter your first
name.");
        if (MemSubForm.firstname.value.length != 0)
        MemSubForm.lastname.focus();
}
if (MemSubForm.lastname.value.length == 0) {
        MemSubForm.lastname.value = prompt("Please enter your last
name.");
        if (MemSubForm.lastname.value.length != 0)
        MemSubForm.addr1.focus();
}
if (MemSubForm.addr1.value.length == 0) {
        MemSubForm.addr1.value = prompt("Please enter your address.");
        if (MemSubForm.addr1.value.length != 0)
        MemSubForm.addr1.focus();
}
if (MemSubForm.city.value.length == 0) {
        MemSubForm.city.value = prompt("Please enter your city of
residence.");
```

```
        if (MemSubForm.city.value.length != 0)
        MemSubForm.city.focus();
}
if (MemSubForm.state.value == 0) {
        alert("Please select a 2-letter state abbreviation.");
        MemSubForm.state.focus();
}
else if (MemSubForm.code.value.length != 3 ||
        isNaN(MemSubForm.code.value)) {
        alert("Please enter a valid telephone area code.");
        MemSubForm.code.focus();
        MemSubForm.code.select( );
}
else if (MemSubForm.phone.value.length != 7 ||
        isNaN(MemSubForm.phone.value)) {
        alert("Please enter a valid phone number.");
        MemSubForm.phone.focus();
        MemSubForm.phone.select();
}
else if (MemSubForm.zip.value.length < 5) {
        MemSubForm.zip.value = prompt("Please enter a valid zip
code.");
        if (MemSubForm.zip.value.length != 0)
        MemSubForm.zip.focus();
}
// -->
</script>
</head>
<body>
```

The Form HTML Code

```
<h2 style="text-align: center">Computer Technology Documentation
Project</h2>
<h2 style="text-align: center">Member Registration</h2>
<hr size="5" width="90%">

<div style="text-align: center">
<table>
<td>
<pre>
```

```
<form name="RegistrationForm">
<b>First Name:   </b><input type="text" name="firstname">
<b>Last Name:   </b><input type="text" name="lastname">
<b>Address1:   </b><input type="text" name="addr1">
<b>Address2:   </b><input type="text" name="addr2">
<b>City:   </b><input type="text" name="city">
<b>State:   </b><select name="state" size="1">
        <option value="1">AL
        <option value="2">AK
        <option value="3">AR
        <option value="4">AS
        <option value="5">AZ
        <option value="6">CA
        <option value="7">CO
        <option value="8">CT
        <option value="9">DC
        <option value="10">DE
        <option value="11">FL
        <option value="12">FM
        <option value="13">GA
        <option value="14">GU
        <option value="15">HI
        <option value="16">ID
        <option value="17">IA
        <option value="18">IL
        <option value="19">IN
        <option value="20">KS
        <option value="21">KY
        <option value="22">LA
        <option value="23">MA
        <option value="24">MD
        <option value="25">ME
        <option value="26">MH
        <option value="26">MI
        <option value="28">MN
        <option value="29">MO
        <option value="30">MP
        <option value="31">MS
        <option value="32">MT
        <option value="33">NC
        <option value="34">ND
        <option value="35">NE
        <option value="36">NH
```

```
        <option value="37">NJ
        <option value="38">NM
        <option value="39">NV
        <option value="40">NY
        <option value="41">OH
        <option value="42">OK
        <option value="43">OR
        <option value="44">PA
        <option value="45">PR
        <option value="46">PW
        <option value="47">RI
        <option value="48">SC
        <option value="49">SD
        <option value="50">TN
        <option value="51">TX
        <option value="52">UT
        <option value="53">VA
        <option value="54">VI
        <option value="55">VT
        <option value="56">WA
        <option value="57">WI
        <option value="58">WV
        <option value="59">WY
</select>
<b>Zip:         </b><input type="text" name="zip" maxlength="10"
size="10">
<b>Area Code/Phone:</b><input type="text"
name="code" maxlength="3" size="3">
<input type="text" name="phone" maxlength="7" size="7">

<div style="text-align: center">
<input type="button" name="cmdSubmit" value="Submit"> <input
type="reset"
 value="clear">
</div>
</pre>

</form>
</td>
</table>
</div>
</body>
```

How the Form Appears and Works:

Computer Technology Documentation Project

Member Registration

First Name:	
Last Name:	
Address1:	
Address2:	
City:	
State:	--
Zip:	
Area Code/Phone:	

Submit Clear

Using Applets

Putting an Applet on a Web Page

```
<html>
<head>
<title>An Applet Example</title>
</head>
<body>
<p>The Applet example is: <br>
<applet code="MyApplet.class" width="200" height="50">
<param name="font" value="TimesRoman">
<param name="size" value="16">
You see this if your browser does not include Java support.
</applet>
</p>
</body>
</html>
```

If the OBJECT tag is used rather than the APPLET tag, use CLASSID in place of CODE to specify the file name. A JAR (JAVA Archive) file is a package of the applet (or application) along with all required supporting class files. The "jar" command is used to create these JAR files from class and gif files.

Applet Attributes

The APPLET element is depreciated in favor of the OBJECT element for those browsers that support it. The OBJECT element is described below.

- ALT - Alternate text if the applet is not loadable.
- CLASS="..." - The name of the applet.
- CODE - The name of the applet main class file.
- CODEBASE - Specifies a different folder for the browser to look for the applet file.
- WIDTH - The width of the applet output area in pixels.
- HEIGHT - The height of the applet output area in pixels.
- ALIGN - ALIGN="TOP "- Indicates how the applet should be aligned with any text that follows it. Values:
    - LEFT
    - RIGHT

    - TOP - Align with the topmost item in the line.
    - BOTTOM
    - MIDDLE
    - ABSMIDDLE
    - ABSBOTTOM
    - BASELINE
    - TEXTTOP
- PARAM - Used to pass parameters to applets. Parameters expected but not passed are passed as a NULL. Your program needs to test for NULL parameters and set a default for any expected parameters. The "getParameter" function is used to get the passed parameters in the applet.
- SRC="..." - The URL of the directory where the compiled applet can be found (should end in a slash / as in "http://mysite/myapplets/"). Do not include the actual applet name, which is specified with the CLASS attribute.
- HSPACE - Space in pixels between the applet and the text around it.
- VSPACE - Vertical space between the applet and other text.

The Object Element

Example

```
<html>
<head>
<title>An Applet Example</title>
</head>
<body>
<p>The Applet example is: <br>
<object classid="java:MyApplet.class" codetype="application/
java" width="200" height="50">
<param name="font" value="TimesRoman">
<param name="size" value="16">
You see this if your browser does not include Java support.
</applet>
</p>
</body>
</html>
```

Object Attributes

- ALIGN - (Depreciated) - Sets alignment to middle, top, bottom, left, or right.
- ARCHIVE - Lets the browser download several archive files at once.
- BORDER - (Depreciated) - The width of the border.
- CLASSID - Determines how the object is implemented.
- CODEBASE - Sets relative URI locations.
- CODETYPE - Specifies whether it is an application and program language.
- DATA - The embedded object Universal Resource Indicator (URI).
- DECLARE - The object is not instantiated until a link is later clicked on or an object uses it.
- HEIGHT
- NAME
- STANDBY - Text to be displayed while the object is loading.
- TABINDEX - A numerical value that sets the tabbing order of the object.
- TYPE - Sets the media type such as screen, printer, TV.
- USEMAP

See the "Recommended Reading" section for sources of a more complete list.

The PARAM Element

Used to set up parameters for the OBJECT and APPLET elements. (Occurs inside <applet> or <object> tags.) There is no ending tag for this element. Attributes include:

- NAME="..." - The type of information being given to the applet or ActiveX control.
- TYPE - The MIME content type of the resource.
- VALUE="..." - The actual information to be given to the applet or ActiveX control.
- VALUETYPE - Sets the type for the VALUE attribute. Possible values include data, ref, and object.

XHTML

XHTML is the latest World Wide Web Consortium standard as of this writing. It stands for XML Hypertext Markup Language (XHTML). It replaces HTML but is much like HTML with some additional rules to add better structure to the language. It, like HTML, has a strict, transitional, and frameset DTD.

XHTML Rules

1. The XHTML document must be well formed. This means:
    - There must be one and only one top level element.
    - All elements must have a starting and an ending tag with matching starting and ending names. For instance documents that normally have no closing tag are normally written:

      ```
      <br>
      <hr>
      <li>
      ```

      Now must be written:

      ```
      <br />
      <hr />
      <li />
      ```

    - Element names are case sensitive.
    - Elements must be nested properly.
2. Tags must be in lower case letters.
3. Values of attributes must be in quotes. Formally the following would be OK:

   ```
   <A href=index.html target=_top>
   ```

   Now use the following:

   ```
   <a href="index.html" target="_top">
   ```

4. Attributes may not be minimized.

   ```
   <table>
   <tr><td nowrap<find /var/spool -mtime +40</td></tr>
   ```

```
</table>
```

Becomes:

```
<table>
<tr><td nowrap="nowrap"<find /var/spool -mtime +40</td></tr>
</table>
```

5. A DTD Declaration with head and body elements must be present in the document. The DTD can be Strict, Transitional, or Frameset (for Frames). An example DTD declaration is shown below:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

   Of course after the DTD declaration an <html> element should exist which includes both a <head> and <body> element as is contained in a normal HTML document.

6. The name attribute is replaced by the id attribute.

```
<input type="text" name="firstname" />
```

   Becomes:

```
<input type="text" id="firstname" />
```

HTML Tags

Start tag	End tag	Description
<!DOCTYPE>	-	Document type.
<ABBREV>	</ABBREV>	Denotes an abbreviation.
<A>	</A>	Anchor
<APPLET>	</APPLET>	Anchor
<AREA>	-	Defines areas of an image map that the user may click on to establish a link.
<AU>	</AU>	Author.
<B>	</B>	Bold text
<BASE>	-	Defines the base point from which anchor references are made.
<BASEFONT>	-	Used to set the default font size on the current page.
<BGSOUND>	-	Used to play sounds on a web page.
<BIG>	</BIG>	Set text size to larger than normal.
<BLOCKQUOTE>	</BLOCKQUOTE>	Indents the text as though it were a quote.
 	-	Break, starts a new line, but doesn't put a blank line between text.
<BODY>	</BODY>	Defines the body of the HTML page.
<BUTTON>	</BUTTON>	Used to define a FORM submit, reset, or push button.
<CODE>	</CODE>	Denotes program code and should be set in the same format as the PRE tag, but does not work either with all browsers or all HTML versions.
<DIR>	</DIR>	directory
<DD>	</DD>	Designated definition.
<DEL>	</DEL>	Denotes deleted text.
<DIV>	</DIV>	Used to offset parts of the HTML document to perform assignment of special attributes or controls to that section.
<DFN>	</DFN>	Denotes the definition of a term.
<DL>	</DL>	Definition list.
<DOCTYPE>	-	Document type.

<DT>	</DT>	Definition term.
<EM>	</EM>	Sets text in italics and denotes emphasis.
<FIELDSET>	</FIELDSET>	Allows grouping of form controls.
<FONT>	</FONT>	Sets a section of text with a specific font.
<FORM>	</FORM>	Displays a form for user information.
<FRAME>	-	Specifies a frame to be placed inside a FRAMESET.
<FRAMESET>	</FRAMESET>	Used to divide the browser window into separate frames that can display multiple documents.
<H1>	</H1>	Defines a level 1 header.
<H2>	</H2>	Defines a level 2 header.
<H3>	</H3>	Defines a level 3 header.
<H4>	</H4>	Defines a level 4 header.
<H5>	</H5>	Defines a level 5 header.
<H6>	</H6>	Defines a level 6 header.
<HEAD>	</HEAD>	Defines the head of document.
<HR>	-	Horizontal rule, starts a new line with horizontal graphic line between sections.
<HTML>	</HTML>	Defines an HTML document or page.
<IMG>	-	Used to embed graphics in HTML pages.
<INPUT>	</INPUT>	An input element for a form.
<INS>	</INS>	Denotes inserted text.
<I>	</I>	Italics
<KBD>	</KBD>	Text to be typed at the keyboard, such as a computer command. The test is displayed in a mono spaced format.
<LABEL>	<.LABEL>	Allows an ability to click on a text label to select a form control
<LI>	-	List item in a list.
<MAP>	</MAP>	Starts a client-side image map.
<MENU>	</MENU>	A menu list (Depreciated).
<META>	-	Meta element defining special characteristics of the document.
<NOFRAMES>	</NOFRAMES>	Used for web browsers that don't support FRAMESETs.
<NOSCRIPT>	</NOSCRIPT>	Used for web browsers that don't support script code.

<OBJECT>	</OBJECT>	Object, the replacement for Applets and more.
<OL>	</OL>	Numbered or ordered list.
<OPTGROUP>	</OPTGROUP>	Sets up choices inside a SELECT menu.
<OPTION>	</OPTION>	Used to set items in a list of selectable items in a form.
<P>	</P>	Paragraph.
<PARAM>	-	Used to set up parameters for the OBJECT and APPLET elements..
<PRE>	</PRE>	Preformatted text. Text presented as typed with tabs, spaces and other whitespace.
<Q>	</Q>	Quotation.
<SAMP>	</SAMP>	Denotes sample text.
<SCRIPT>	</SCRIPT>	Used to embed client side script in HTML code, usually in the header.
<SELECT>	</SELECT>	Creates a list of items that can be selected.
<SMALL>	</SMALL>	Makes text smaller than normal.
<SPAN>	</SPAN>	Allows text attributes and color to be modified at any location.
<STRIKE>	</STRIKE>	Draws a line through the text as a "strikeout".
<STRONG>	</STRONG>	Same as bold but denotes strong emphasis.
<STYLE>	</STYLE>	Used to set the style of the web page.
_		Subscript.
[	]	Superscript.
<TABLE>	</TABLE>	Table.
<TD>	</TD>	Designates a table data cell.
<TEXTAREA>	</TEXTAREA>	Indicates an element that has multiple text lines
<TH>	</TH>	Designates a table header.
<TR>	</TR>	Designates a table row.
<TITLE>	</TITLE>	Document or page title.
<TT>	</TT>	Monospaced typewriter font.
<U>	</U>	Underlined text.
<UL>	</UL>	Unnumbered list.

HTML Terms

- Angle brackets - The characters, < and >, set HTML tags off from the rest of the text on an HTML page. These two symbols enclose all HTML tags.
- Attributes - Defined words used in an HTML tag to modify the tag properties. They can be used to add or change color or change a size in some element.
- BODY - The main part of an HTML document.
- Element - An HTML designator that defines special objects such as paragraphs, lists, and lines in HTML. It usually consists of a beginning and ending tag, but may have just a beginning tag.
- FTP - File Transfer Protocol. A method used to send or receive files between two computers on the network or internet.
- Graphics drawing program - A program used to draw graphics images that may be used on a web page. Normally these images are stored as gif files, but several formats may be used.
- Header - The beginning part of an HTML document which defines various characteristics such as the title.
- Home page - The main page of an organization or company which is the first page seen when the organization's URL is visited.
- HTML - Hyper-Text Markup Language is the basic language web pages are written in.
- HTML editor - An editor that makes web page creation easier than using a normal text editor. Although you can write HTML code using a standard text editor, it is strongly recommended that you use some type of HTML editor even for learning. There are two categories of text editor.
    1. WYSIWYG (What you see is what you get) or graphical HTML editors which allow the user to see the page as the web browser would see it as they edit the page. You will not see the HTML elements or tag sets using this type of editor, so for learning HTML it is not recommended.
    2. A text based HTML editor lets the user see and edit HTML code directly. Usually the HTML tags are displayed in a different color than the surrounding text which makes them easier to see and work with. This web page was written using the Arachnophilia v3.9 HTML editor.
- HTTP - Hypertext transfer protocol is the internet protocol used to transport information between the client browser and the web page server.
- Hyperlinks or links - HTML coded locations of other material on the web. They are usually underlined and consist of a different text color than the surrounding text. When you click on them they will usually cause your browser to load the page it is pointing to and you will see the new page displayed.
- Tags - Tags are used to surround text which has special meaning in HTML. Tags tell the browser what to do. The tag set <P> </P> is used to tell the browser that text between the two tags is to be set apart as a separate paragraph in HTML.
- URI - Uniform Resource Identifier which is used as an identifier for a resource. The URI may include a complete path to the resource or may only be relative without a complete path.
- URL - Uniform Resource Locator. It is used to specify file locations of html or other files. The

URL "http://ctdp.tripod.com" is the URL for the CTDP home page. The first section "http:" designates the type of transfer which in this case is Hyper-text transfer protocol. Other cases include FTP, GOPHER, and FILE.

- Web browser - Software used to retrieve and display web pages on the web. It is considered to be a client program which makes requests to web servers for web page files. Browsers can all read basic HTML but may be different in other areas such as being able to display or run script code, video and graphics.
- Web server - The computer the web pages are stored on. The web server will transmit the web pages across the network/internet to the client computer which is running a web browser.

Recommended Reading

The following sources should be helpful in learning about HTML and the use of style sheets. They give additional details not listed here.

- Web Design Group's web page, HTML 4.0 Reference section. It is an excellent reference and is also available in an offline viewable format. This is the best source on the web that I have found describing HTML.
- The World Wide Web Consortium Hypertext Markup Language Home Page. Contains excellent references to multiple sets of HTML documentation including XHTML, HTML 4.01 and older versions. Also links to information about style sheets can be found here along with links to HTML document type definitions (DTDs).
- The World Wide Web Consortium HTML 4.01 Specification.
- The World Wide Web Consortium HTML 4.01 Strict DTD.
- The World Wide Web Consortium HTML 4.01 Transitional DTD.
- Web Design Group's web page, Cascading Style Sheets Guide. It includes a tutorial, information about CSS structure, CSS rules, and how to link style sheets to HTML along with several other references.
- The World Wide Web Consortium CSS Level 1 Recomendation.
- The World Wide Web Consortium CSS Level 1 Specification.

Credits

This document was produced for the Computer Technology Documentation Project and the latest version is available at http://www.comptechdoc.org/independent/html/guide/.

Document:

The CTDP HTML Guide Version 0.8.2

Authors:

Mark Allen
Kathy Deka

Those who contributed by submitting comments:

- **Thore Harald Hoye**