Microsoft Internet Explorer 3.0 Author's Guide and HTML Reference

This author's guide and HTML reference describes the construction of interactive HTML-based documents that can be viewed with Microsoft® Internet Explorer. The guide begins with a chapter that describes the elements necessary to create the basic HTML document. The chapters that follow include topics for lists, tables, frames, objects, ActiveX™ Controls, and so on. The Source Specification section contains source specifications for each HTML element. The HTML Reference section provides a description and an example for each HTML element supported by Internet Explorer.

Category	Description
Release Notes for Internet Explorer 3.01	Describes changes in style sheet implementation.
About HTML Standards	Describes HTML standards and source specifications.
Basics	Reviews construction of a basic HTML document.
Lists	Reviews types of lists and their construction.
Character Formatting	Shows character formatting and font control.
Images and Multimedia	Describes how to include graphics, sound, video, and marquees.
Color	Gives a list of colors that can be referred to by name.
Frames	Lays out the basics of frame design.
Tables	Describes building and formatting tables and table elements.
Forms	Details the construction of forms, and lists the form input controls.
Style Sheets	Illustrates the use of style sheets.
Cascading Style Sheets	Reviews cascading style sheets.
Image Maps	Demonstrates the use of an image with "hot spots."
Scripts	Shows how to include Visual Basic and JScript scripts.
Objects	Gives an overview of inserting OLE objects and ActiveX Controls.
Client Pull	Describes pulling data from the server automatically.
Character Sets	Describes special characters.
Source Specification	Describes source specifications for elements.
HTML Reference	Catalogs HTML 3.2 elements in an A through Z reference.

Release Notes for Internet Explorer 3.01

Internet Explorer version 3.01 includes a change in HTML margin handling when using style sheets. HTML pages that take advantage of this feature will not display in the manner specified when viewed with Internet Explorer 3.0. The following describes how to author HTML pages with margins that are rendered the same size in version 3.0 and 3.01.

In Internet Explorer 3.0

Margin sizes are absolute, not relative. For example, with a 1-inch left margin in the BODY element and a 0.5-inch left margin for a paragraph inside the BODY element, a paragraph would be 0.5 inches, instead of 1.5 inches, from the left side of the window. This includes vertical margins.

In Internet Explorer 3.01

All margins use relative sizes. For example:

```
<BODY STYLE="margin-left: 1in; margin-right: 1in">
<P STYLE="margin-left: 0.5in; margin-right: 1.5in">This is some text.</P>
</BODY>
```

This text will have an effective left margin of 1.5 inches and an effective right margin of 2.5 inches.

Creating a page compatible with Internet Explorer 3.0 and 3.01

By not specifing more than one margin in the hierarchy, your HTML page is compatible with both Internet Explorer 3.0 and 3.01. For example:

```
<BODY>
<P STYLE="margin-left: 1.5in; margin-right: 2.5in">This is some text.</P>
</BODY>
```

About HTML Standards

HTML is the standard language for creating documents for the World Wide Web (WWW). The HTML used by Microsoft Internet Explorer provides a robust implementation of the HTML standards being defined by the main Web organizing bodies and used by the most popular browsers—such as Internet Explorer and Netscape Navigator. HTML for Internet Explorer 3.0 is consistent with published standards, and yet includes new, more powerful elements and controls. This HTML recognizes that authors who write online documents for the WWW are proactive in promoting additions and modifications to HTML. These authors want the most popular of those extensions to be part of the development language and the browser that they use.

The HTML used by Internet Explorer provides key handles for the experienced WWW author to move outside HTML into scripts, such as Visual Basic Scripting Edition (VBScript) and JScript, and object-oriented (OO) programming languages, such as Microsoft Visual C++® and Java™. The author may not write these OO programs, but this HTML gives the author the ability to embed objects, control their input, and influence the layout of these objects within an HTML page. Thus, a spectrum of complexity and power is available to the Web developer with HTML as a foundation. One author can produce a document for the Web using just HTML. Another author who wants to tie a Web page to a database can add VBScript objects. A third author can construct content using a C++ object model. HTML can provide the entry point for each of these authors.

Basics

An HTML (hypertext markup language) document is a text file that contains the elements Internet Explorer uses to display text, multimedia objects, and hyperlinks. Using HTML, an author can format a document for display and add hyperlink jumps to other documents. Text that is formatted as a hyperlink can be selected by a user with the mouse. Once selected, the hyperlink jump will load the referenced document into your browser. A hyperlink and the object to which the link jumps can both be defined using HTML.

An element is the most basic part of HTML. An element consists of a start-tag, an end-tag, and the data characters enclosed by the two tags. A tag starts with a less-than (<) sign and ends with a greater-than (>) sign. An end-tag consists of the tag name immediately preceded by a slash (/). Some tags require that you always provide the matching end-tag; others allow you to omit the end-tag if the result is clear and unambiguous. For example, here is a sentence that will display in bold:

```
<B>This sentence displays in bold.</B>
```

This example is an HTML element. The start-tag is <B>. The end-tag is </B>. The data characters are "This sentence displays in bold." This element, when read by Internet Explorer, will turn on bold formatting, based on the start-tag, and display the data characters in bold. The end-tag switches off the bold formatting. Many elements can be "nested" by placing an entire element inside the tags of another. For example, here is some italicized text placed inside a bold element:

```
<B>This sentence,<I>as written</I>, displays in bold.</B>
```

The <I>as written</I> element displays in both bold and italic because it is nested inside a bold element.

An element can have one or more attributes. An attribute is a parameter associated with an element that extends its meaning. Tags and attribute names are not case-sensitive, but they are typically written in uppercase to distinguish them from the data characters.

This is a very simple HTML document:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">lt;HTML>
<HEAD>
<TITLE>Simple HTML Document</TITLE>
</HEAD>
<BODY>
<P>A very simple HTML document.
</BODY>
</HTML>
```

Every HTML document begins with the !DOCTYPE element. !DOCTYPE specifies to the browser which version of HTML is being used.

The next element in this basic document is HTML, which informs the browser that the content of the file is written in HTML. The matching end-tag (</HTML>) is the last tag in the file.

The HEAD tag marks the beginning of the document header. The document header describes the elements that apply for all sections of the current document and any documents that contain content or are related to this document. Typically, the TITLE element appears in the header. Internet Explorer displays the text of the TITLE element in its title bar. A menu bar or image that is repeated for other documents may appear in the header section.

The BODY element appears at the start of the main content of the document. The BODY element encloses the body text, images, and multimedia objects. The P element inserts a new paragraph with a carriage return and line feed. The end-tag, </P>, is typically omitted.

With HTML, you can create hyperlink jumps between your documents. A hyperlink is any text or image that, when clicked, loads another document or another section of the current document into the Internet Explorer window. The A element, or anchor, associates text or a graphic to another document or to a location within the current document. A hyperlink appears as a clickable "hot spot" (the clickable text or image). To create a hyperlink, you enclose the text or image with the anchor tags and set the HREF=

attribute to the destination address, as in the following:

```
<P>Click <A HREF="//www.microsoft.com/">here</A> to visit the
Microsoft Web site.
```

In this example, the address for the Web site is enclosed in double quotation marks. The double quotation marks are optional unless the attribute value contains spaces. If you enclose a value that contains double quotation marks, use &QUOT; for each occurrence of the mark within the value.

For example, you can create a hyperlink destination (anchor spot) within your HTML document by using the NAME= attribute. Use the A element to relate text or a graphic to a name that you create. Then reference the name with a hyperlink. In the following example, the first line creates the named reference. The second line in the example includes a hyperlink with a jump to that place in the document.

```
<A NAME="using"></A><H2>Using Internet Explorer 3.0</H2>

...

<P>For more information, see <A HREF="#usingie30">Using Internet Explorer
3.0</A>
```

When you click the hyperlink "Using Internet Explorer", you jump to the named reference, usingie30.

Note Although Internet Explorer can display incomplete or improperly tagged files, the result is often not what you may have intended. You should always use the tags carefully, using them only in the context in which they are defined to be used and omitting end-tags only if they are defined as optional.

The typical HTML document consists of one or more text paragraphs organized into sections. You can mark the beginning of the sections in your HTML documents by using the header Hn element, where "n" is a number from 1 to 6 (1 creates the highest level heading and uses the largest font size). These elements create headings by applying changes to the size and style of the text to indicate the section level. The specific heading format can be controlled by using attributes or a style sheet; otherwise, it takes the default formatting. The following example creates a first-level section heading:

```
<H1>Welcome to Internet Explorer!</H1>
```

The Hn element allows for six levels; you specify the level by using the element name that includes that level number (H1, H2, H3, and so on). The end-tag is always required.

By default, section headings are left-aligned. You can override the default alignment and center the heading by using the ALIGN= attribute, as in the following example:

```
<H2 ALIGN=CENTER>How to Use Internet Explorer</H2>
```

In addition to using the P element to create simple paragraphs, you can use elements, such as BLOCKQUOTE, LISTING, PLAINTEXT, PRE, and XMP, to create paragraphs that use a different size and style for the text. For example, you can use the PRE element to display characters in a fixed-width font rather than the variable-width font used for simple paragraphs, or the BLOCKQUOTE element to slightly indent the paragraph text (from both the left and right margins) to make the paragraph stand out.

You can apply a style to a sequence of paragraphs (for example, those tagged with P) by enclosing the paragraphs with an ADDRESS, BLOCKQUOTE, or CENTER element. The following example shows how to center a sequence of simple paragraphs:

```
<CENTER>
<P>This paragraph is centered.

<P>And this paragraph is centered too.
</CENTER>
```

An alternate way to center individual paragraphs is to use the ALIGN= attribute with the P element and set the attribute value to CENTER, as in the following example:

```
<P ALIGN=CENTER>This paragraph is centered.
```

This displays as:

This paragraph is centered.

Lists

You can create a variety of lists in your document by using the UL, OL, MENU, and DIR elements in conjunction with the LI element. You can also create definition lists that give you a simple two-column list for terms and their definitions.

For example, you can create a bulleted list, consisting of individual items preceded by a bullet character, by using UL and LI, as in the following example:

```
<UL>
<LI>Bulleted Lists
<LI>Ordered Lists
<LI>Directory Lists
<LI>Itemized Lists
<LI>Definition Lists
</UL>
```

This displays as:

- Bulleted Lists
- Ordered Lists
- Directory Lists
- Itemized Lists
- Definition Lists

You use the OL and LI elements to create an ordered list. The ordered list consists of individual items that are sequentially numbered or lettered. To set the style of numbering or lettering, you use the TYPE= attribute in OL. Similarly, you use the START= attribute to set the initial number or letter. By default, the style is decimal numbers starting at 1.

```
<OL>
<LI>Step One.
<LI>Step Two.
<LI>Step Three.
</OL>
```

This displays as:

1 Step One.
2 Step Two.
3 Step Three.

The DIR and LI elements create a directory list, consisting of individual items (none containing more than 20 characters) displayed in columns. The MENU and LI elements create an itemized list consisting of individual items.

A definition list formats terms at the left, with their definitions indented below. You use the DL element to create the list, the DT element to refer to the term, and the DD element to contain the term's definition. The following example shows how to use these elements to create a list:

```
<DL>
<DT>Cat
<DD>A small domesticated mammal.
<DT>Lizard
<DD>A reptile generally found in dry areas.
</DL>
```

This displays as:

Cat

  A small domesticated mammal.

Lizard

A reptile generally found in dry areas.

Character Formatting

You can use a variety of elements to set the size and style of the text characters. For example, you can use the B or STRONG element to make text bold, and the I or EM element to make text italic. Similarly, you can use the S or STRIKE element to strike out text, and the U element to underline text. The following examples set words and phrases within the paragraphs to bold and italic:

```
<P>This <B>word</B> is formatted as bold. This <I>word</I> is formatted
as italic.
```

This displays as:

This **word** is formatted as bold. This *word* is formatted as italic.

```
<P><STRONG>Text formatted as strong.</STRONG> <EM>Emphasized text.</EM>.
```

This displays as:

Text formatted as strong. *Emphasized text.*

You can apply character formatting to a sequence of paragraphs (for example, those tagged with P) by enclosing the paragraphs with a character formatting element. For example, to make all the text in a sequence of paragraphs bold, do the following:

```
<B>
<P>This text is bold.

<P>And this text is bold too.
</B>
```

Although you can use this technique with simple paragraphs, some elements "block" the effect of the character formatting elements. For example, you cannot make all the text in a table bold by enclosing it in a B element.

You can create superscripts and subscripts by using the SUP and SUB elements. These elements reduce the size of the text and align it at the top or bottom of the current line of text.

You can change the size of the text by using elements such as BIG and SMALL, or by using the SIZE= attribute with the FONT element. The following example increases the size of the word "LARGE" and reduces the word "TINY":

```
<P>Use the <BIG>LARGE</BIG> machine for business,
the <SMALL>TINY</SMALL> machine for personal items.
```

If you use the FONT element to change text size, you can specify either a fixed or relative size. A fixed size is a number in the range 1 through 7. A relative size is a positive or negative number, preceded by the plus (+) or minus (-) sign, that indicates a size that is relative to the base font size, as set using the BASEFONT element. The following example shows the effect of using relative sizes:

```
<BASEFONT SIZE=3> This sets the base font size to 3.
<FONT SIZE="+4"> Now the font size is 7.
<FONT SIZE="-1"> Now the font size is 2.
```

You can also use the FACE= attribute with the FONT element to set the name of the font used for text. Some of the most-used fonts are "Arial," "Times New Roman," and "Courier New," but you can use the name of any font installed on the computer on which your HTML document is being viewed. The following example sets the Arial font for the text in the section heading:

```
<H1><FONT FACE="ARIAL">Welcome to Internet Explorer!</FONT></H1>
```

If the given font is not available, Internet Explorer uses a default font. To increase the chances that a font of your choice is applied to the text, you can specify more than one font in a FONT element. In this case, Internet Explorer checks for each font (in the order given) before using the default font. In the following example, Internet Explorer checks for "Arial", "Lucida Sans", and "Times Roman" before resorting to the default font:

```
<FONT FACE="Arial,Lucida Sans,Times Roman">
This text will be in either Arial, Lucida Sans, or Times Roman,
depending on which fonts you have installed on your system.
</FONT>
```

You can apply size and font changes to a sequence of paragraphs by enclosing the paragraphs with the FONT element. As with other character formatting elements, some elements do not accept the effect of FONT, so this technique does not work for all paragraphs. For example, table elements accept changes to the font's name but size changes are not accepted.

Color

You can set colors in your HTML document by using the color attributes of the BODY, FONT, HR, MARQUEE, and TABLE elements. For example, you can set background color for your document by using the BGCOLOR= attribute with the BODY element as in the following example:

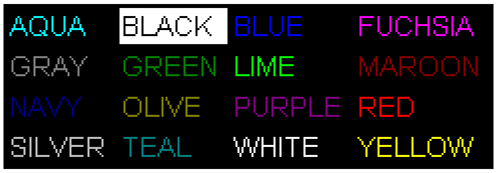
```
<BODY BGCOLOR=WHITE>
<P>This page has a white background.
</BODY>
```

You can specify colors in two ways: by using a color name (as in the preceding example), or by using numbers to denote a red-green-blue color value.

Internet Explorer supports these color names:

AQUA [#00FFFF], BLACK [#000000], BLUE [#0000FF], FUCHSIA [#FF00FF]
GRAY [#808080], GREEN [#008000], LIME [#00FF00], MAROON [#800000]
NAVY [#000080], OLIVE [#808000], PURPLE [#800080], RED [#FF0000]
SILVER [#C0C0C0], TEAL [#008080], WHITE [#FFFFFF], YELLOW [#FFFF00]

These colors display as:



AQUA	BLACK	BLUE	FUCHSIA
GRAY	GREEN	LIME	MAROON
NAVY	OLIVE	PURPLE	RED
SILVER	TEAL	WHITE	YELLOW

Note This feature is not compatible with Netscape v2.0.

A red-green-blue color value consists of three two-digit hexadecimal numbers, with each number specifying the intensity of the corresponding color. For example, the color value #FF0000 is red because the red number is set to its highest value, FF (255 in the decimal, or base 10, system). Green and blue are set to zero. Similarly, #00FF00 is green and #0000FF is blue. The pound sign (#) is optional. In this example, a red horizontal rule is displayed by the following HTML code:

```
<HR COLOR="#00FF0000">
```

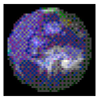
Although red-green-blue color values theoretically allow for many thousands of colors, the actual number of colors available for your HTML document depends on the color capabilities of the devices the document will be viewed on. Many personal computers can display only 16 colors due to the type of video display adapter they use. Always choose colors carefully, and whenever possible test your color choices on a variety of computers.

Images and Multimedia

You can embed images, sounds, and even video clips in your HTML document by using the IMG and BGSOUND elements. And you can apply simple animation to text by using the MARQUEE element.

You use the IMG element to insert images into your document. You specify the image source, typically a .bmp, .gif, or JPEG file, and specify the image attributes, such as the width and height, alignment, and so on. The following example demonstrates how to display a .bmp file, TheEarth.bmp.

```
<IMG SRC="TheEarth.bmp" WIDTH=46 HEIGHT=46 ALT="Picture of the Earth">
```



The SRC= attribute specifies the name of the image file. The file will be sized in a square 46 pixels wide by 46 pixels high. For a text-only browser, the text "Picture of the Earth" will display in place of the image. When you place an IMG element in text, Internet Explorer aligns the surrounding text with the bottom of the image. You can align the text with the top or center of the image by using the ALIGN= attribute to set the alignment to TOP, BOTTOM, or CENTER. In this case, the image keeps its position within the surrounding text.

You can also use the ALIGN= attribute to have the text flow around the image. For example, setting this attribute to LEFT aligns the image with the left margin and wraps all subsequent text around the right side of the image. Similarly, setting it to RIGHT wraps all subsequent text around the left side. When you use the LEFT or RIGHT alignment, typically it is useful to also use the BR element with the CLEAR= attribute to stop wrapping and force all remaining text below the image, as in the following example:

```
<IMG SRC="TheEarth.gif" ALIGN=LEFT> Here's some text to the right of a
picture.
<BR CLEAR=LEFT>Here's some text beneath the picture.
```

You can make an image a hyperlink hot spot "anchoring" the IMG element to a reference with the A element. By default, Internet Explorer draws a border around the image to mark it as a hot spot. To remove the border, set the BORDER= attribute in the IMG element to zero.

You can add background sounds or music to your document by using the BGSOUND element. You specify the address of a sound file, such as a .wav, .au, or MIDI file, and use the LOOP= attribute to set how often the file plays, as in the following example:

```
<BGSOUND SRC="boing.wav" LOOP=5>You will hear a sound played five times
in a row.
```

You can animate a line of text by using the MARQUEE element. The element automatically scrolls the text, to the left or right, whenever a user views your document. To animate the text, you enclose it in the element and set attributes for scroll direction, type, and amount, as in the following example:

```
<MARQUEE DIRECTION=RIGHT BEHAVIOR=SCROLL SCROLLAMOUNT=10
SCROLLDELAY=200>This text is displaying in a right scrolling
marquee.</MARQUEE>
```

In this example, the text "This text is displaying in a right scrolling marquee." scrolls from the left margin to the right. After it disappears beyond the right margin, it starts again at the left. The text moves 10 pixels after each 200 millisecond delay.

You can align marquees to the left or right, like images, and also set the background color, height, width, and extra spacing around the marquee.

Frames

Frames give you a way to organize and structure the content of your HTML documents by letting you create compound documents that the user can view within the main window of Internet Explorer. To use frames, you create a document that uses the FRAMESET and FRAME elements to divide the main window into rectangular frames (like panes in a window). Then, for each frame, you specify an HTML document that contains the content (text and images) to fill the frame. Floating frames enable you to open a browser within a browser. You can insert a floating frame in the same manner in which you can insert an image on an HTML page. You can specify the size of the frame and its border, and you can align it with other text and images on the page. With frames, you can create sophisticated layouts that add and mix sounds, video, animation, and colors. Using two frames in a single page, you can display an index in one frame and the content in another. For example, you can split the main window into two equal frames and fill these with different documents by using the following elements:

```
<HTML>
<HEAD>
<TITLE>Two Equal Frames</TITLE>
</HEAD>
<FRAMESET COLS="50%,*">
<FRAME SRC=x.htm>
<FRAME SRC=y.htm>
</FRAMESET>
</HTML>
```

In this example, the COLS= attribute in the FRAMESET element specifies the width of the frames. The width of the first frame is 50 percent of the main window, and the width of the second, given as an asterisk, is relative to the first (meaning it spans whatever is left of the main window). Note that this document does not contain a BODY element. This is because documents that define frames do not contain content. Instead, the SRC= attribute in each FRAME element specifies a document. In this example, the x.htm and y.htm files are content sources for the frames.

You can divide the main window into rows, as well as columns, by using the ROWS= attribute. Furthermore, you can independently divide individual rows into rows and columns by nesting FRAMESET elements. The following example shows how to divide the main window into two rows in which the last row is divided into two columns:

```
<HTML>
<HEAD>
<TITLE>Nested Frames</TITLE>
</HEAD>
<FRAMESET ROWS="10%,*">
<FRAME SCROLLING=NO SRC=z.htm>
<FRAMESET COLS="50%,*">
<FRAME SRC=x.htm>
<FRAME SRC=y.htm>
</FRAMESET>
</FRAMESET>
</HTML>
```

In this example, the SCROLLING= attribute is used in the first FRAME element to prevent the scroll bar from being displayed. By default, Internet Explorer displays the scroll bar only if the entire content of the frame does not fit within the frame. Setting SCROLLING= to NO always prevents the scroll bar.

The FRAME element has attributes to let you set the width and height of margins within the frame, and whether the frame has a border. The FRAMESET element has attributes to let you set the spacing between frames, and whether the frames in the set have borders.

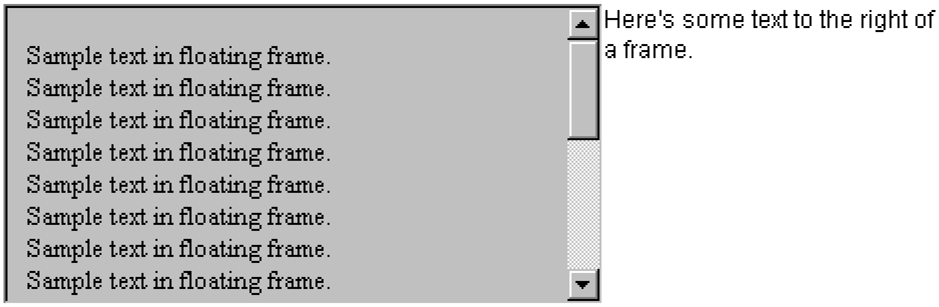
An important feature of the FRAME element is the NAME= attribute. This attribute lets you assign a unique name to the frame; you then use this name when creating hyperlinks to direct documents into the frame. To create such a hyperlink, use the TARGET= attribute in the A element. For example, the following element creates a hyperlink that displays the x.htm file in a frame named CONTENT:

```
<A HREF="x.htm" TARGET="CONTENT">List of Components</A>
```

Internet Explorer provides an alternate way to create compound documents by letting you place frames in your HTML document using the IFRAME element. Called "floating frames," this design technique allows you to insert HTML documents into your document in the same way you insert images using the IMG element. This means you can use the ALIGN= attribute just as you do with IMG to align the frame with the surrounding text. The following example aligns a frame at the left margin and wraps subsequent text around the right side of the frame:

```
<IFRAME SRC="xx.htm" ALIGN=LEFT>
</IFRAME> Here's some text to the right of a frame.
<BR CLEAR=LEFT>Here's some text beneath the frame.
```

This displays as:



Sample text in floating frame.
Sample text in floating frame.
Sample text in floating frame.
Sample text in floating frame.
Sample text in floating frame.
Sample text in floating frame.
Sample text in floating frame.
Sample text in floating frame.

Here's some text to the right of a frame.

The IFRAME syntax might not be compatible with all browsers. In that case, you can use a FRAME element within the IFRAME tags to provide an alternative presentation. For example:

```
<IFRAME SRC="xx.htm" ALIGN=LEFT><FRAME SRC="xx.htm">
<IFRAME>
```

In the previous example, the text of xx.htm will display in a floating frame in either an IFRAME or non-IFRAME-compatible browser. Remember that you set the attributes of IFRAME and FRAME independently. For example, if you want to specify position or size, you include those attributes in both the IFRAME tag and the FRAME tag.

Tables

Use the TABLE element to format a table. The TR element (table row) inserts a row in the table, and the TD element (table detail) inserts a cell within a row. With Internet Explorer 3.0 as the browser, images or text can be placed within the cells. This example shows the HTML elements for a simple table:

```
<TABLE>
<TR>
<TD>Apples<TD>Celery
<TR>
<TD>Oranges<TD>Carrots
</TABLE>
```

This displays as:

Apples	Celery
Oranges	Carrots

By default, Internet Explorer aligns the table to the left. The width of the table, unless specified, is determined by the content of the longest element in each column. The content of each cell is aligned to the center and to the left edge of each cell, but you can override these defaults. For example, you can set the width of the table by using the WIDTH= attribute in the TABLE element. You can align the content of each cell to the top, left, right, or bottom of a cell by using the ALIGN= and VALIGN= attributes in TR or TD. This example creates a table that is the full width of the Internet Explorer window. The contents of the cells are at the top and the left:

```
<TABLE WIDTH="100%">
<TR VALIGN=TOP ALIGN=LEFT>
<TD>Apples<TD>Celery
<TR VALIGN=TOP ALIGN=LEFT>
<TD>Oranges<TD>Carrots
</TABLE>
```

This displays as:

Apples	Celery
Oranges	Carrots

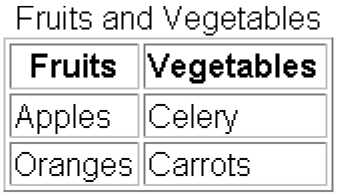
In this example, the table width is given as a percentage of the total Internet Explorer window width. But you can also specify table widths in pixels.

You can add a caption, row and column headings, and a border to a table by using elements and attributes. For example, you can add a caption to a table by using the CAPTION element. By default, the caption is centered above the table, but you can use the ALIGN= attribute to place the caption at the top or bottom and at the left or right edge of the table.

To add headings to the rows and columns of a table, use the TH element. This element is like the TD element in that it creates a cell and can contain text and images, but it automatically emphasizes its text to distinguish it from text in other cells. To draw a border around the table and the individual cells, you use the BORDER= attribute in the TABLE element. Specify the border width in pixels. The following example creates a table with headings, border, and a caption:

```
<TABLE BORDER=1>
<CAPTION>Fruits and Vegetables</CAPTION>
<TR>
<TH>Fruits<TH>Vegetables
<TR>
<TD>Apples<TD>Celery
<TR>
<TD>Oranges<TD>Carrots
</TABLE>
```

This displays as:



Fruits and Vegetables

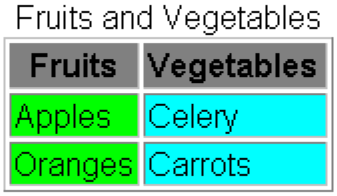
Fruits	Vegetables
Apples	Celery
Oranges	Carrots

By default, Internet Explorer centers headings in the cell, but you can override this by using the ALIGN= and VALIGN= attributes.

You can add color to your tables by using the BGCOLOR= and BORDERCOLOR= attributes. These attributes are available in the TABLE, TR, and TD elements, so you can apply colors to all cells in a table, to cells in selected rows, or to individual cells. The **BGCOLOR=** attribute sets the color used to fill the background of the cell before text and images are drawn. The **BORDERCOLOR=** attribute sets the color of the borders drawn around the table, row, or cell. The following example uses the same background color for the column headings, but different colors for the two columns in the table:

```
<TABLE BORDERCOLOR=NAVY BORDER=1>
<CAPTION>Fruits and Vegetables</CAPTION>
<TR BGCOLOR=GRAY>
<TH>Fruits<TH>Vegetables
<TR>
<TD BGCOLOR=LIME>Apples<TD BGCOLOR=AQUA>Celery
<TR>
<TD BGCOLOR=LIME>Oranges<TD BGCOLOR=AQUA>Carrots
</TABLE>
```

This displays as:



Fruits and Vegetables

Fruits	Vegetables
Apples	Celery
Oranges	Carrots

You can change the character formatting for the text in a table by using elements such as B, I, and FONT. You can change the color and font name for all text in a table by enclosing the table in an appropriate FONT element, but the table elements block the effect of other character formatting elements. To get these effects, you must apply the elements within each cell.

Within a cell, you can use most of the elements that you ordinarily use in the body of the HTML document, including elements for section headings, lists, and even other tables. Using tables in this way can give you additional control over the placement of text and images when your document is displayed, but can also make the management of your document more complex. For example, you can use tables to give your document a two-column layout by nesting a single-column table in each cell of a two-column table. But if you do this, you must take special care to divide the content of your document equally between the two nested tables and be prepared to account for differences in the size of the window through which users view your document. In most cases, documents that use tables in this way are designed to be viewed within a minimum window size at a given screen resolution.

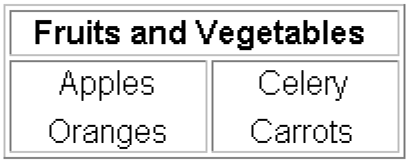
If you use tables in the more traditional way (that is, presenting information in rows and columns), there are some additional elements and attributes that can make that job easier. The THEAD, TBODY, and TFOOT elements let you divide your tables into parts: header, body, and footer. The COLGROUP and COL elements let you group columns within the table and globally apply properties, such as alignment, to the columns without having to specify these properties in each TD element.

The FRAME= and RULES= attributes in the TABLE element let you control how the table border is drawn. For example, you can choose to have no border around the outside of the table, while restricting the border inside the table to just vertical rules separating the columns and horizontal rules separating the table header, body, and footer. The COLSPAN= and ROWSPAN= attributes in the TD and TH elements let you extend the content of a cell into adjoining cells. This is useful, for example, if you need to stretch a column heading across more than one column.

The following example shows how some of these elements and attributes can be used in a table:

```
<TABLE WIDTH="50%" BORDER=1 FRAME=BOX RULES=GROUPS>
<COLGROUP ALIGN=CENTER>
<COLGROUP ALIGN=CENTER>
<THEAD>
<TR>
<TH COLSPAN=2>Fruits and Vegetables
<TBODY>
<TR>
<TD>Apples<TD>Celery
<TR>
<TD>Oranges<TD>Carrots
</TABLE>
```

This displays as:



Fruits and Vegetables	
Apples	Celery
Oranges	Carrots

Internet Explorer supports advanced table functionality, including displaying background images behind table cells, specifying rules or borders just along columns or along rows, and aligning text to the baseline within a table cell.

Style Sheets

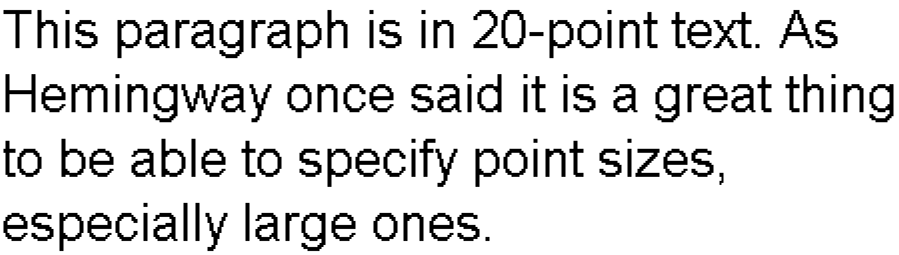
Style sheets give you the ability to attach styles to HTML elements. This allows you to control margins, line spacing, the placement of text and graphics, colors, font faces, and font sizes. Style sheets make it easier to create an index because indexing software has only to read the structural elements rather than the full content of a page. Cascading style sheets, as defined by the W3C CSS1 specifications, are supported by Microsoft Internet Explorer 3.0. For more information on CSS1 and Internet Explorer's support for its properties, see *A User's Guide to Style Sheets*.

Using the STYLE element, you can include style information as part of an HTML document and apply the style to some or all of the text, or you can create a style sheet as a separate document and attach it to one or more pages on your Web site. You can use both methods in a single document—creating a style sheet for all the documents on a Web site, while selectively applying a special style sheet to text within selected documents. When you include multiple sets of style information, there may be some contention as to what style controls the display of an element. CSS1 provides rules of precedence to decide this.

There are two ways to place style information inside a document. The first is to assign a style to an element. For example, here's how to specify a paragraph with a font size of 20 points.

```
<P "font-size: 20pt"> This paragraph is in 20-point text.
As Hemingway once said, "It is a great thing to be able to specify
point sizes, especially large ones."
```

This displays as:



This paragraph is in 20-point text. As Hemingway once said it is a great thing to be able to specify point sizes, especially large ones.

To place style information at the top of a page, insert a STYLE block at the top of your document. The block is placed after the HTML element and before the BODY element. One type of style information that can be used within the STYLE element is cascading style sheet (CSS) properties. In the examples for this section, CSS properties appear within the curly braces after the elements for which they provide formatting information. (For a full listing of the CSS properties supported by Microsoft Internet Explorer 3.0, see *A User's Guide to Style Sheets*.)

```
<HTML>
<STYLE>
BODY {background: white; color: black}
H1 {font: 14pt Arial bold}
P {font: 10pt Arial; text-indent: 0.5in}
A {text-decoration: none; color: blue}
</STYLE>
<BODY>
<H1>This is a headline! In 14-point Arial bold!</H1>
</BODY>
</HTML>
```

To assign more than one kind of style information at the same time, separate the styles with semicolons. For example, to set an entire HTML page to 10-point Times font, the colors to black on white, and both left and right margins to one inch, use the following:

```
<STYLE>
BODY {font: 10pt Times; color: black; background: white;
margin-left: 1in; margin-right: 1in}
</STYLE>
```

You can attach an external style sheet to a document or documents on a site. To link a page to this style sheet, use the LINK element, as in the following example (where mystyles.css is the external style sheet):

```
<LINK REL=STYLE TYPE="text/css"
HREF="http://www.mycompany.com/mystyles.css">
```

The best method for accomplishing this, however, is to use the STYLE tag.

Note Notice the use of the TYPE= attribute. This describes the MIME type of the external file that contains the style information.

The text formatting features supported by Internet Explorer 3.0 are described fully in *A User's Guide to Style Sheets*.

Cascading Style Sheets

Using cascading style sheets, more than one style sheet can influence the presentation simultaneously. There are two main reasons for this feature: modularity and author/reader balance.

A style sheet designer can combine several (partial) style sheets to reduce redundancy:

```
@import url(http://www.style.org/stylea);
@import url(http://www.style.org/styleb);

H1 { color: red }     /* override imported sheets */
```

Also, both readers and authors can influence the presentation through style sheets. To do so, they use the same style sheet language, thus reflecting a fundamental feature of the Web: everyone can become a publisher.

Sometimes conflicts will arise between the style sheets that influence the presentation. Conflict resolution is based on each style rule having a weight. By default, the weights of the reader's rules are less than the weights of rules in the author's documents, but style sheet designers can increase the weights of their rules by using the keyword "important". In this example, the H1 and P styles will override all other style sheet formatting:

```
H1 { color: red ! important }
P  { font-size: 12pt ! important }
```

A reader rule labeled important will override an author rule with normal weight. An author rule labeled important will override an important reader rule.

Cascading Order

Conflicting rules are intrinsic to the CSS mechanism. To find the value for an element/property combination, the following algorithm should be followed:

1. Find all declarations that apply to the element/property in question. Declarations apply if the selector matches the element in question. If no declarations apply, the inherited value is used. If there is no inherited value (this is the case on the root element and for properties that do not inherit), the initial value is used.
2. Sort the declarations by explicit weight: declarations marked '!important' carry more weight than unmarked (normal) declarations.
3. Sort by origin: the author's style sheets override the reader's style sheets, which override the default values.
4. Sort by specificity of selector: more specific selectors will override more general ones. To find the specificity, count the number of ID attributes in the selector (a), the number of CLASS attributes in the selector (b), and the number of tag names in the selector (c). Concatenating the three numbers (in a number system with a large base) gives the specificity. Some examples are:

   ```
   LI            {...}  /* a=0 b=0 c=1 -> specificity =   1 */
   UL LI         {...}  /* a=0 b=0 c=2 -> specificity =   2 */
   UL OL LI      {...}  /* a=0 b=0 c=3 -> specificity =   3 */
   LI.red        {...}  /* a=0 b=1 c=1 -> specificity =  11 */
   UL OL LI.red  {...}  /* a=0 b=1 c=3 -> specificity =  13 */
   #x34y         {...}  /* a=1 b=0 c=0 -> specificity = 100 */
   ```

5. Sort by order specified: if two rules have the same weight, the latter specified should live.

The search for the property value can be terminated whenever one rule has a higher weight than the other rules that apply to the same element/property combination.

This strategy gives authors' style sheets considerably higher weight than those of the reader. It is therefore important that the reader has the ability to turn off the influence of a certain style sheet, for example through a pull-down menu.

Image Maps

Image maps allow users to access other documents by clicking different areas in an image. You can implement image maps in two ways: by storing image map information on a server or by including image map information in your document.

If you store image map information on a server, you need a script or other service on the server to process click information. In your document, you mark the image as a "server-side" image map by using the **ISMAP** attribute in the IMG element and enclosing the image in an A element, as in the following example:

```
<A HREF="Jump.map"><IMG SRC="Sample.gif" ISMAP></A>
```

In this example, the image map information is in the file named Jump.map. When the user clicks the picture in Sample.gif, the server receives the coordinates of the click and can pick the appropriate destination for the click by checking the information in Jump.map.

If you include image map information in your document, Internet Explorer processes the click information and picks the appropriate destination for the click. In your document, you mark the image as a "client-side" image map by using the USEMAP= attribute in the IMG element, and you add image map information by using the MAP and AREA elements, as in the following example:

```
<MAP NAME="map1">
<AREA SHAPE="RECT" COORDS="0, 0, 16, 16" HREF="Sample1.htm">
<AREA SHAPE="RECT" COORDS="0, 16, 16, 16" HREF="Sample2.htm">
</MAP>
<IMG BORDER=0 SRC="map1.gif" USEMAP="#map1">
```

In this example, the image map defines two equal rectangular areas. The area on the left is linked to Sample1.htm. The area on the right is linked to Sample2.htm.

The AREA element permits other shapes, such as circles and polygons. If two or more shapes overlap, Internet Explorer uses the first shape defined in the MAP element to determine the destination. Any number of AREA elements can specify the same destination. This is useful if you want to map a complex shape to a single destination. If a portion of the image is not within a given shape, clicking in that portion has no effect.

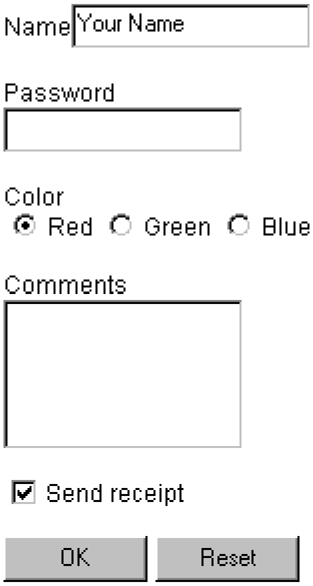
Forms

Forms provide a way to prompt the user for information and to carry out actions based on that input. A form consists of one or more input controls that the user uses to enter text and select choices. Once the user provides the input, the form collects the data and sends it to a destination specified in the form element. To carry out the requested action, the server must have a script or other service that corresponds to the given destination. This script processes the information and can perform actions necessary to process the data.

To create a form, you use the FORM element to enclose one or more INPUT elements. The FORM element specifies the action to take when the user has provided the information. The INPUT elements define the type and function of the input controls in the form. The following example shows how to combine these elements to create a form:

```
<FORM ACTION="http://intranet/survey" METHOD=POST>
<P>Name
<BR><INPUT NAME="CONTROL1" TYPE=TEXTBOX VALUE="Your Name">
<P>Password
<BR><INPUT TYPE="PASSWORD" NAME="CONTROL2">
<P>Color
<BR><INPUT TYPE="RADIO" NAME="CONTROL3" VALUE="0" CHECKED>Red
<INPUT TYPE="RADIO" NAME="CONTROL3" VALUE="1">Green
<INPUT TYPE="RADIO" NAME="CONTROL3" VALUE="2">Blue
<P>Comments
<BR><INPUT TYPE="TEXTAREA" NAME="CONTROL4" SIZE="20,5" MAXLENGTH="250">
<P><INPUT NAME="CONTROL5" TYPE=CHECKBOX CHECKED>Send receipt
<P><INPUT TYPE="SUBMIT" VALUE="OK"><INPUT TYPE="RESET" VALUE="Reset">
</FORM>
```

This displays as:



Name Your Name

Password

Color
● Red ○ Green ○ Blue

Comments

☑ Send receipt

Scripts

Scripts are blocks of code that can run when an HTML page is loaded or when an event happens, such as the click of a button. To learn more about scripts and scripting, read the Object Model for Scripting document included in the ActiveX SDK, as well as Microsoft's pages on VBScript and JavaScript. This section describes how to incorporate a script into an HTML document.

There are three ways to attach and invoke scripts in HTML:

- Use the SCRIPT element.
- Use those attributes of HTML elements that support scripts.
- Use a custom URL type.

Using the SCRIPT Element

Use the SCRIPT element to add scripts to HTML documents. Scripts reside inside the container of a SCRIPT element.

Using SCRIPT, the full source code of a script can be included within the document. The SCRIPT element can be used to point to external scripts as well.

For example, this HTML element describes a page with a SCRIPT element that includes code written in VBScript:

```
<SCRIPT language="VBScript">
Document.write("Hello, Webmaster.")
</SCRIPT>
```

The example in JScript would read:

```
<SCRIPT language="JavaScript">
document.write("Hello, Webmaster.")
</SCRIPT>
```

Evaluation of SCRIPT and its use with objects

The SCRIPT element is evaluated when the document is loaded. All code is executed at load time in the order in which it appears in the document. Therefore, any reference to an object, such as an ActiveX Control, must appear in the text *after* the script element in which the object is defined. These objects can be referenced only in a script block following the script block that defined them. You will be able to refer to and copy references to objects that are the result of a code download at any time after the object has been downloaded to your computer.

Using Scripts as Attributes of HTML Elements

Another way to insert scripts is to use the attributes of HTML elements that support scripts. When these attributes match with events on the elements, the script is executed when the event occurs. This can be done with HTML elements, such as forms, buttons, or links; however, this method does not work for items inserted using the OBJECT tag.

The following example uses this syntax in Button1 to handle the onClick event. To demonstrate the ability to combine scripting languages on the same page, the scriptlet for Button1 is implemented in VBScript, and that for Button2 in JScript.

```
<SCRIPT NAME="Form1">
  <INPUT TYPE="button" NAME="Button1" VALUE="VBScript"     onClick="pressed"
LANGUAGE="VBScript">
        <INPUT TYPE="button" NAME="Button2" VALUE="JScript"
onClick="pressed2()" LANGUAGE="JavaScript">
</FORM>

<SCRIPT LANGUAGE="VBSCRIPT">
  sub pressed
    document.Form1.Button1.value="Pressed"
                alert "Pressed the VBScript button"
  end sub
</SCRIPT>

<SCRIPT LANGUAGE="JavaScript">
  function pressed2()
  {
                document.Form1.Button2.value="Pressed"
    alert("Pressed the JScript button.")
  }
</SCRIPT>
```

Note the use of the language attribute on the input tag to indicate the script's language. If no language is specified, the scriptlet defaults to the language of the most recently encountered script block. If no script block has been encountered, the language defaults to JScript.

The FORM, INPUT, BODY, and A elements support this syntax, but with differing events. See the individual tags referenced later in this document.

An alternative using SCRIPT

This method can be used for any named elements, and for any elements inserted using the OBJECT tag. The following example is similar to the previous script example, but it uses a different syntax:

```
<FORM NAME="Form1">
        <INPUT TYPE="button" NAME="Button1" VALUE="Click">
  <SCRIPT FOR="Button1" EVENT="onClick" LANGUAGE="VBScript">
    alert "Button has been pressed"
                document.Form1.Button1.value="PRESSED"
  </SCRIPT>
</FORM>
```

Using Scripts in URLs

Scripts can be invoked using the A element combined with a custom URL type. This allows a script to be executed when the user clicks a hyperlink. This URL type is valid in any context, but is most useful when used with the A element. For example:

```
<A HREF="javascript:alert('hi there')">Click me to see a message.</A>
```

displays an alert message box that contains the text 'hi there'.

Objects

Internet Explorer supports objects according to the WWW Consortium (W3C) object model extension to HTML 3.2. Objects add functionality to your HTML document by letting you insert images, video, and programs, such as Java applets, and ActiveX controls.

To insert an object, you use the OBJECT element, supplying attribute values that specify the object type, location, initial data, and so on. If the object has configurable properties, you can set these using the PARAM element. The following example shows how to insert the marquee ActiveX Control and fill it with content:

```
<OBJECT
ALIGN=CENTER CLASSID="clsid:1a4da620-6217-11cf-be62-0080c72edd2d"
WIDTH=200 HEIGHT=200 BORDER=1 HSPACE=5
ID=marquee>
<PARAM NAME="ScrollStyleX" VALUE="Circular">
<PARAM NAME="ScrollStyleY" VALUE="Circular">
<PARAM NAME="szURL" VALUE="marqcont.htm">
<PARAM NAME="ScrollDelay" VALUE=60>
<PARAM NAME="LoopsX" VALUE=-1>
<PARAM NAME="LoopsY" VALUE=-1>
<PARAM NAME="ScrollPixelsX" VALUE=0>
<PARAM NAME="ScrollPixelsY" VALUE=-3>
<PARAM NAME="DrawImmediately" VALUE=0>
<PARAM NAME="Whitespace" VALUE=0>
<PARAM NAME="PageFlippingOn" VALUE=0>
<PARAM NAME="Zoom" VALUE=100>
<PARAM NAME="WidthOfPage" VALUE=400>
</OBJECT>
```

In this example, the OBJECT element specifies the class identifier of the control (assumed to be already installed and registered) and the alignment, width, height, and other attributes of the control. The series of PARAM elements sets the values for the individual properties of the control, determining how and when the contents are scrolled. The szURL property determines which HTML document is used as content.

A matching end-tag is required for each OBJECT element. Within these elements, you can place one or more PARAM elements. You can also place any elements and text that you would ordinarily use in the body of the HTML document, but these elements and text are not processed and displayed unless the HTML viewer does not process the OBJECT element. To learn more about the inclusion of objects in HTML, read Microsoft's documentation for the *ActiveX Engineer's Guide and Object Reference*.

EMBED

Microsoft Internet Explorer 3.0 will support the EMBED element syntax for embedding objects on an HTML page. This support is meant for compatibility with other browsers. The accepted WWW Consortium (W3C) HTML standard recommends using the OBJECT element for embedding objects in HTML, and Internet Explorer 3.0 supports this syntax as well.

OBJECT Compatibility

Internet Explorer 3.0 supports the OBJECT element. Browsers that comply with the accepted W3C HTML standard will understand this HTML syntax. However, the OBJECT syntax degrades gracefully in other browsers as well, because browsers that are compatible with the OBJECT element will ignore additional elements placed within the OBJECT element. The following illustrates how to use the OBJECT element but allow content to be viewable by other browsers:

```
< OBJECT
  CLASSID="clsid:05589FA1-C356-11CE-BF01-00AA0055595A" WIDTH=100 HEIGHT=250>
     <PARAM NAME=SRC VALUE="TheEarth.AVI">
     <PARAM NAME=AUTOSTART VALUE=TRUE>
     <PARAM NAME=PLAYBACK VALUE=FALSE>
     <EMBED SRC="TheEarth.AVI" WIDTH=100 HEIGHT=250 AUTOSTART=TRUE
PLAYBACK=FALSE>
</OBJECT>
```

In browsers that support the OBJECT element, the EMBED element will be ignored. Browsers that do not support OBJECT will ignore the OBJECT element and PARAM attribute.

Client Pull

Client pull provides the ability to automatically load a new document in the specified time or reload a document on a regular basis. Internet Explorer supports client pull using the META element. The META element must be inside the HEAD element of the HTML document. For instance, META HTTP-EQUIV=REFRESH CONTENT=2 reloads a document every two seconds. Setting the HTTP-EQUIV= attribute to REFRESH gives the instruction to reload. The CONTENT= attribute specifies the time in seconds that the page refreshes. You can specify any URL in the element. If no URL is specified, the current document is reloaded. The META element also has several other functions, such as specifying keywords for Web search engines to use for indexing. Read more about META in the HTML Reference, and see the following chapter regarding character sets.

Character Sets

The following information is described in this section:

- Character recognition in IE 3.0
- ISO Latin-1 Character Set
- Complete Character Set, including Microsoft Internet Explorer additions

Charset Recognition in IE3

This charset recognition specification defines which charset identifiers Internet Explorer recognizes in the HTTP header of HTTP replies, and which charset IDs it recognizes in the <META ... CHARSET=*charsetID*> tag. It also specifies which built-in charset translation the charset ID maps to. This does not specify what IE should send out as the ACCEPT-CHARSET parameter in the HTTP request.

Table of Base Charsets, Display Names, and Aliases

In the following table, the base charset is the basic translation built into IE3. *Aliases* lists all other charset IDs that are recognized and can be represented without translation, using the "base charset" translation method. This does not, in all cases, mean that alias and base charset represent the same charset; the alias charset can be a subset of the base charset. Base charset is not a recognized name unless repeated in the "aliases" column.

Base Character	Display Name	Aliases
1252	Western	us-ascii, iso8859-1, ascii, iso_8859-1, iso-8859-1, ANSI_X3.4-1968, iso-ir-6, ANSI_X3.4-1986, ISO_646.irv:1991, ISO646-US, us, IBM367, cp367, csASCII, latin1, iso_8859-1:1987, iso-ir-100, ibm819, cp819
28592	Central European (ISO)	iso8859-2, iso-8859-2, iso_8859-2, latin2, iso_8859-2:1987, iso-ir-101, l2, csISOLatin2
1250	Central European (Windows)	windows-1250, x-cp1250
1251	Cyrillic (Windows)	windows-1251, x-cp1251
1253	Greek (Windows)	windows-1253
1254	Turkish (Windows)	windows-1254
932	Shift-JIS	shift_jis, x-sjis, ms_Kanji, csShiftJIS
EUC-JP	EUC	Extended_UNIX_Code_Packed_Format_for_Japanese, csEUCPkdFmtJapanese, x-euc-jp
JIS	JIS	csISO2022JP, iso-2022-jp
1257	windows-1257	
950	Traditional Chinese (BIG5)	big5, csbig5, x-x-big5
936	Simplified Chinese	GB_2312-80, iso-ir-58, chinese, csISO58GB231280, csGB2312, gb2312
20866	Cyrillic (KOI8-R)	csKOI8R, koi8-r

949	Korean	ks_c_5601, ks_c_5601-1987, korean, csKSC56011987

Correct Usage

The correct usage is as specified in RFC 1341. For example:

```
<META HTTP-EQUIV="Content-Type"
  CONTENT="text/html; charset=Windows-1251">
```

This should be in or before HEAD but certainly before BODY.

Priority

The following list shows the priorities of charset declarations that IE will use.

1. Use any charset parameter passed in the HTTP content-type.
2. Use the <META ... CHARSET...> tag.
3. Use the user preference for default document encoding.

A frameset can have differing charsets per frame.

Position of <META .. CHARSET=..> in the Document

The <META .. CHARSET=..> sequence can appear anywhere in the document BEFORE the BODY tag. In any case, it affects the whole document, including TITLEs, appearing before the <META CHARSET> tag.

ISO Latin-1 Character Set

The following table contains the ISO Latin-1 character set. The table describes each character, its decimal code, and its special entity reference for HTML, as well as providing a brief description.

Character	Decimal Code	HTML	Description
À	&#192;	&Agrave;	Capital A, grave accent
à	&#224;	&agrave;	Small a, grave accent
Á	&#193;	&Aacute;	Capital A, acute accent
á	&#225;	&aacute;	Small a, acute accent
Â	&#194;	&Acirc;	Capital A, circumflex
â	&#226;	&acirc;	Small a, circumflex
Ã	&#195;	&Atilde;	Capital A, tilde
ã	&#227;	&atilde;	Small a, tilde
Ä	&#196;	&Auml;	Capital A, diæresis / umlaut
ä	&#228;	&auml;	Small a, diæresis / umlaut
Å	&#197;	&Aring;	Capital A, ring
å	&#229;	&aring;	Small a, ring
Æ	&#198;	&AElig;	Capital AE ligature
æ	&#230;	&aelig;	Small ae ligature
Ç	&#199;	&Ccedil;	Capital C, cedilla
ç	&#231;	&ccedil;	Small c, cedilla
È	&#200;	&Egrave;	Capital E, grave accent
è	&#232;	&egrave;	Small e, grave accent
É	&#201;	&Eacute;	Capital E, acute accent
é	&#233;	&eacute;	Small e, acute accent
Ê	&#202;	&Ecirc;	Capital E, circumflex
ê	&#234;	&ecirc;	Small e, circumflex

Ë	&#203;	&Euml;	Capital E, diæresis / umlaut
ë	&#235;	&euml;	Small e, diæresis / umlaut
Ì	&#204;	&Igrave;	Capital I, grave accent
ì	&#236;	&igrave;	Small i, grave accent
Í	&#205;	&Iacute;	Capital I, acute accent
í	&#237;	&iacute;	Small i, acute accent
Î	&#206;	&Icirc;	Capital I, circumflex
î	&#238;	&icirc;	Small i, circumflex
Ï	&#207;	&Iuml;	Capital I, diæresis / umlaut
ï	&#239;	&iuml;	Small i, diæresis / umlaut
Ð	&#208;	&ETH;	Capital Eth, Icelandic
ð	&#240;	&eth;	Small eth, Icelandic
Ñ	&#209;	&Ntilde;	Capital N, tilde
ñ	&#241;	&ntilde;	Small n, tilde
Ò	&#210;	&Ograve;	Capital O, grave accent
ò	&#242;	&ograve;	Small o, grave accent
Ó	&#211;	&Oacute;	Capital O, acute accent
ó	&#243;	&oacute;	Small o, acute accent
Ô	&#212;	&Ocirc;	Capital O, circumflex
ô	&#244;	&ocirc;	Small o, circumflex
Õ	&#213;	&Otilde;	Capital O, tilde
õ	&#245;	&otilde;	Small o, tilde
Ö	&#214;	&Ouml;	Capital O, diæresis / umlaut
ö	&#246;	&ouml;	Small o, diæresis / umlaut
Ø	&#216;	&Oslash;	Capital O, slash
ø	&#248;	&oslash;	Small o, slash
Ù	&#217;	&Ugrave;	Capital U, grave accent
ù	&#249;	&ugrave;	Small u, grave accent
Ú	&#218;	&Uacute;	Capital U, acute accent
ú	&#250;	&uacute;	Small u, acute accent
Û	&#219;	&Ucirc;	Capital U, circumflex
û	&#251;	&ucirc;	Small u, circumflex
Ü	&#220;	&Uuml;	Capital U, diæresis / umlaut
ü	&#252;	&uuml;	Small u, diæresis / umlaut
Ý	&#221;	&Yacute;	Capital Y, acute accent
ý	&#253;	&yacute;	Small y, acute accent
Þ	&#222;	&THORN;	Capital Thorn, Icelandic
þ	&#254;	&thorn;	Small thorn, Icelandic
ß	&#223;	&szlig;	Small sharp s, German sz
ÿ	&#255;	&yuml;	Small y, diæresis / umlaut

Character Set

The following table describes the complete character set for Internet Explorer 3.0 English (U.S.). The first column shows the character as it appears in Internet Explorer 3.0. The second column shows the decimal number as it is written in an HTML document to produce the characters. Occasionally, special characters have mnemonic names. For example, the registered trademark character can be written in

HTML as &reg;. The third column lists these HTML characters. The last column gives a description of each character where appropriate.

Character	Decimal Code	HTML	Description
	&#00;		Unused
	&#01;		Unused
	&#02;		Unused
	&#03;		Unused
	&#04;		Unused
	&#05;		Unused
	&#06;		Unused
	&#07;		Unused
	&#08;		Unused
	&#09;		Horizontal tab
	&#10;		Line feed
	&#11;		Unused
	&#12;		Unused
	&#13;		Carriage Return
	&#14;		Unused
	&#15;		Unused
	&#16;		Unused
	&#17;		Unused
	&#18;		Unused
	&#19;		Unused
	&#20;		Unused
	&#21;		Unused
	&#22;		Unused
	&#23;		Unused
	&#24;		Unused
	&#25;		Unused
	&#26;		Unused
	&#27;		Unused
	&#28;		Unused
	&#29;		Unused
	&#30;		Unused
	&#31;		Unused
	&#32;		Space
!	&#33;		Exclamation mark
"	&#34;	&quot;	Quotation mark
#	&#35;		Number sign
$	&#36;		Dollar sign
%	&#37;		Percent sign
&	&#38;	&amp;	Ampersand
'	&#39;		Apostrophe
(	&#40;		Left parenthesis

)	&#41;		Right parenthesis
*	&#42;		Asterisk
+	&#43;		Plus sign
,	&#44;		Comma
-	&#45;		Hyphen
.	&#46;		Period (fullstop)
/	&#47;		Solidus (slash)
0	&#48;		Digit 0
1	&#49;		Digit 1
2	&#50;		Digit 2
3	&#51;		Digit 3
4	&#52;		Digit 4
5	&#53;		Digit 5
6	&#54;		Digit 6
7	&#55;		Digit 7
8	&#56;		Digit 8
9	&#57;		Digit 9
:	&#58;		Colon
;	&#59;		Semicolon
<	&#60;	&lt;	Less than
=	&#61;		Equals sign
>	&#62;	&gt;	Greater than
?	&#63;		Question mark
@	&#64;		Commercial at
A	&#65;		Capital A
B	&#66;		Capital B
C	&#67;		Capital C
D	&#68;		Capital D
E	&#69;		Capital E
F	&#70;		Capital F
G	&#71;		Capital G
H	&#72;		Capital H
I	&#73;		Capital I
J	&#74;		Capital J
K	&#75;		Capital K
L	&#76;		Capital L
M	&#77;		Capital M
N	&#78;		Capital N
O	&#79;		Capital O
P	&#80;		Capital P
Q	&#81;		Capital Q
R	&#82;		Capital R
S	&#83;		Capital S
T	&#84;		Capital T

U	&#85;	Capital U
V	&#86;	Capital V
W	&#87;	Capital W
X	&#88;	Capital X
Y	&#89;	Capital Y
Z	&#90;	Capital Z
[	&#91;	Left square bracket
\	&#92;	Reverse solidus (backslash)
]	&#93;	Right square bracket
^	&#94;	Caret
_	&#95;	Horizontal bar (underscore)
`	&#96;	Acute accent
a	&#97;	Small a
b	&#98;	Small b
c	&#99;	Small c
d	&#100;	Small d
e	&#101;	Small e
f	&#102;	Small f
g	&#103;	Small g
h	&#104;	Small h
i	&#105;	Small i
j	&#106;	Small j
k	&#107;	Small k
l	&#108;	Small l
m	&#109;	Small m
n	&#110;	Small n
o	&#111;	Small o
p	&#112;	Small p
q	&#113;	Small q
r	&#114;	Small r
s	&#115;	Small s
t	&#116;	Small t
u	&#117;	Small u
v	&#118;	Small v
w	&#119;	Small w
x	&#120;	Small x
y	&#121;	Small y
z	&#122;	Small z
{	&#123;	Left curly brace
|	&#124;	Vertical bar
}	&#125;	Right curly brace
~	&#126;	Tilde
□	&#127;	Unused
€	&#128;	Unused

	&#160;	&nbsp;	Non-breaking Space
¡	&#161;	&iexcl;	Inverted exclamation
¢	&#162;	&cent;	Cent sign
£	&#163;	&pound;	Pound sterling
¤	&#164;	&curren;	General currency sign
¥	&#165;	&yen;	Yen sign
¦	&#166;	&brvbar; or &brkbar;	Broken vertical bar
§	&#167;	&&sect;	Section sign
¨	&#168;	&&um; or &&die;	Diæresis / Umlaut
©	&#169;	&&copy;	Copyright
ª	&#170;	&&ordf;	Feminine ordinal
«	&#171;	&&laquo;	Left angle quote, guillemot left
¬	&#172;	&&not	Not sign
	&#173;	&shy;	Soft hyphen
®	&#174;	&reg;	Registered trademark
¯	&#175;	&macr; or &hibar;	Macron accent
°	&#176;	&deg;	Degree sign
±	&#177;	&plusmn;	Plus or minus
²	&#178;	&sup2;	Superscript two
³	&#179;	&sup3;	Superscript three
´	&#180;	&acute;	Acute accent
µ	&#181;	&micro;	Micro sign
¶	&#182;	&para;	Paragraph sign
·	&#183;	&middot;	Middle dot
¸	&#184;	&cedil;	Cedilla
¹	&#185;	&sup1;	Superscript one
º	&#186;	&ordm;	Masculine ordinal
»	&#187;	&raquo;	Right angle quote, guillemot right
¼	&#188;	&frac14;	Fraction one-fourth
½	&#189;	&frac12;	Fraction one-half
¾	&#190;	&frac34;	Fraction three-fourths
¿	&#191;	&iquest;	Inverted question mark
À	&#192;	&Agrave;	Capital A, grave accent
Á	&#193;	&Aacute;	Capital A, acute accent
Â	&#194;	&Acirc;	Capital A, circumflex
Ã	&#195;	&Atilde;	Capital A, tilde
Ä	&#196;	&Auml;	Capital A, diæresis / umlaut
Å	&#197;	&Aring;	Capital A, ring
Æ	&#198;	&AElig;	Capital AE ligature
Ç	&#199;	&Ccedil;	Capital C, cedilla
È	&#200;	&Egrave;	Capital E, grave accent

É	&#201;	&Eacute;	Capital E, acute accent
Ê	&#202;	&Ecirc;	Capital E, circumflex
Ë	&#203;	&Euml;	Capital E, diæresis / umlaut
Ì	&#204;	&Igrave;	Capital I, grave accent
Í	&#205;	&Iacute;	Capital I, acute accent
Î	&#206;	&Icirc;	Capital I, circumflex
Ï	&#207;	&Iuml;	Capital I, diæresis / umlaut
Ð	&#208;	&ETH;	Capital Eth, Icelandic
Ñ	&#209;	&Ntilde;	Capital N, tilde
Ò	&#210;	&Ograve;	Capital O, grave accent
Ó	&#211;	&Oacute;	Capital O, acute accent
Ô	&#212;	&Ocirc;	Capital O, circumflex
Õ	&#213;	&Otilde;	Capital O, tilde
Ö	&#214;	&Ouml;	Capital O, diæresis / umlaut
×	&#215;	&times;	Multiply sign
Ø	&#216;	&Oslash;	Capital O, slash
Ù	&#217;	&Ugrave;	Capital U, grave accent
Ú	&#218;	&Uacute;	Capital U, acute accent
Û	&#219;	&Ucirc;	Capital U, circumflex
Ü	&#220;	&Uuml;	Capital U, diæresis / umlaut
Ý	&#221;	&Yacute;	Capital Y, acute accent
Þ	&#222;	&THORN;	Capital Thorn, Icelandic
ß	&#223;	&szlig;	Small sharp s, German sz
à	&#224;	&agrave;	Small a, grave accent
á	&#225;	&aacute;	Small a, acute accent
â	&#226;	&acirc;	Small a, circumflex
ã	&#227;	&atilde;	Small a, tilde
ä	&#228;	&auml;	Small a, diæresis / umlaut
å	&#229;	&aring;	Small a, ring
æ	&#230;	&aelig;	Small ae ligature
ç	&#231;	&ccedil;	Small c, cedilla
è	&#232;	&egrave;	Small e, grave accent
é	&#233;	&eacute;	Small e, acute accent
ê	&#234;	&ecirc;	Small e, circumflex
ë	&#235;	&euml;	Small e, diæresis / umlaut
ì	&#236;	&igrave;	Small i, grave accent
í	&#237;	&iacute;	Small i, acute accent
î	&#238;	&icirc;	Small i, circumflex
ï	&#239;	&iuml;	Small i, diæresis / umlaut
ð	&#240;	&eth;	Small eth, Icelandic
ñ	&#241;	&ntilde;	Small n, tilde
ò	&#242;	&ograve;	Small o, grave accent
ó	&#243;	&oacute;	Small o, acute accent
ô	&#244;	&ocirc;	Small o, circumflex

õ	&#245;	&otilde;	Small o, tilde
ö	&#246;	&ouml;	Small o, diæresis / umlaut
÷	&#247;	&divide;	Division sign
ø	&#248;	&oslash;	Small o, slash
ù	&#249;	&ugrave;	Small u, grave accent
ú	&#250;	&uacute;	Small u, acute accent
û	&#251;	&ucirc;	Small u, circumflex
ü	&#252;	&uuml;	Small u, diæresis / umlaut
ý	&#253;	&yacute;	Small y, acute accent
þ	&#254;	&thorn;	Small thorn, Icelandic
ÿ	&#255;	&yuml;	Small y, diæresis / umlaut

Source Specification

The following table lists the elements and attributes supported by Internet Explorer 3.0. The table identifies the most current HTML specification that supports the element.

Tag	Attribute	Source specification
!--		HTML 1.0
!DOCTYPE		HTML 3.2
A		HTML 1.0
A	CLASS	HTML 3.2
A	HREF	HTML 2
A	ID	HTML 3.2
A	NAME	HTML 2
A	onClick	HTML 3.2 W3C Extensions
A	onMouseOver	HTML 3.2 W3C Extensions
A	REL	HTML 2.0
A	REV	HTML 2.0
A	STYLE	HTML 3.2
A	TARGET	Netscape
A	TITLE	HTML 3.2
ADDRESS		HTML 2
APPLET		HTML 3.2
APPLET	ALIGN	HTML 3.2
APPLET	ALT	HTML 3.2
APPLET	CODE	HTML 3.2
APPLET	CODEBASE	HTML 3.2
APPLET	DOWNLOAD	HTML 3.2
APPLET	HEIGHT	HTML 3.2
APPLET	HSPACE	HTML 3.2
APPLET	NAME	HTML 3.2
APPLET	PARAM NAME	HTML 3.2
APPLET	TITLE	HTML 3.2
APPLET	VSPACE	HTML 3.2
APPLET	WIDTH	HTML 3.2
AREA		HTML 2.0
AREA	ALT	HTML 3.2
AREA	CLASS	HTML 3.2
AREA	COORDS	IE 3.0
AREA	HREF	IE 3.0
AREA	ID	HTML 3.2
AREA	NOHREF	IE 3.0
AREA	SHAPE	IE 3.0
AREA	STYLE	HTML 3.2
AREA	TABINDEX	HTML 3.2
AREA	TARGET	Netscape

AREA	TITLE	HTML 3.2
B		HTML 2.0
BASE		
BASE	HREF	HTML 2.0
BASE	TARGET	Netscape
BASEFONT		
BASEFONT	COLOR	Netscape 2.0
BASEFONT	FACE	Netscape 2.0
BASEFONT	SIZE	Netscape
BGSOUND		
BGSOUND	SRC	IE 3.0
BGSOUND	LOOP	IE 3.0
BIG		HTML 3.2
BLOCKQUOTE		HTML 2.0
BODY		HTML 2.0
BODY	ALINK	HTML 2.0
BODY	BACKGROUND	HTML 3.2
BODY	BGCOLOR	Netscape
BODY	BGPROPERTIES	IE 2.0
BODY	CLASS	HTML 3.2
BODY	ID	HTML 3.2
BODY	LEFTMARGIN	IE 2.0
BODY	LINK	Netscape
BODY	STYLE	HTML 3.2+
BODY	TEXT	Netscape
BODY	TOPMARGIN	IE 2.0
BODY	VLINK	Netscape
BR		HTML 1.0
BR	CLASS	HTML 3.2
BR	CLEAR	HTML 3.2
CAPTION		HTML 3.2
CAPTION	ALIGN	HTML 3.2
CAPTION	VALIGN	HTML 3.2
CENTER		Netscape
CITE		HTML 2.0
CODE		HTML 2.0
COL		HTML 3.2
COL	ALIGN	HTML 3.2
COL	SPAN	HTML 3.2
COLGROUP		HTML 3.2
COLGROUP	HALIGN	HTML 3.2
COLGROUP	SPAN	HTML 3.2
COLGROUP	VALIGN	HTML 3.2
COLGROUP	WIDTH	HTML 3.2

COMMENT		HTML 2.0
DD		HTML 2.0
DD	CLASS	HTML 3.2
DD	ID	HTML 3.2
DD	STYLE	HTML 3.2
DFN		HTML 2.0
DIR		HTML 2.0
DIR	COMPACT	HTML 2.0
DIV		Netscape
DIV	ALIGN	Netscape
DIV	CLASS	HTML 3.2 W3C extension
DIV	CLEAR	HTML 3.2
DIV	ID	HTML 3.2 W3C extension
DIV	LANG	Netscape
DIV	NOWRAP	Netscape
DL		HTML 2.0
DT		HTML 2.0
EM		HTML 2.0
EMBED		HTML 2.0
EMBED	HEIGHT	HTML 2.0
EMBED	NAME	HTML 2.0
EMBED	OPTIONAL PARAM	HTML 2.0
EMBED	PALETTE	HTML 2.0
EMBED	SRC	HTML 2.0
EMBED	WIDTH	HTML 2.0
FONT		HTML 2.0
FONT	COLOR	IE 3.0
FONT	FACE	IE 3.0
FONT	SIZE	Netscape
FORM		HTML 2.0
FORM	ACTION	HTML 2.0
FORM	METHOD	HTML 2.0
FORM	onSubmit	HTML 3.2
FORM	TARGET	Netscape
FRAME		
FRAME	ALIGN	Netscape
FRAME	FRAMEBORDER	IE 3.0
FRAME	MARGINHEIGHT	Netscape
FRAME	MARGINWIDTH	Netscape
FRAME	NAME	Netscape
FRAME	NORESIZE	Netscape
FRAME	SCROLLING	Netscape
FRAME	SRC	Netscape

FRAMESET		
FRAMESET	COLS	Netscape
FRAMESET	FRAMEBORDER	IE 3.0
FRAMESET	FRAMESPACING	IE 3.0
FRAMESET	ROWS	Netscape
H*n*		HTML 2
H*n*	ALIGN	HTML 3.2
HEAD		HTML 2
HR		HTML 2
HR	ALIGN	HTML 3.2
HR	CLASS	HTML 3.2
HR	COLOR	IE 3.0
HR	ID	HTML 3.2
HR	NOSHADE	Netscape
HR	SIZE	Netscape
HR	STYLE	HTML 3.2
HR	WIDTH	Netscape
HTML		HTML 2.0
I		HTML 2.0
IFRAME		IE 3.0
IFRAME	ALIGN	IE 3.0
IFRAME	FRAMEBORDER	IE 3.0
IFRAME	HEIGHT	IE 3.0
IFRAME	MARGINHEIGHT	IE 3.0
IFRAME	MARGINWIDTH	IE 3.0
IFRAME	NAME	IE 3.0
IFRAME	SCROLLING	IE 3.0
IFRAME	SRC	IE 3.0
IFRAME	NORESIZE	IE 3.0
IFRAME	WIDTH	IE 3.0
IMG		
IMG	ALIGN	HTML 3.2
IMG	ALT	HTML 2
IMG	BORDER	HTML 3.2
IMG	CLASS	HTML 3.2
IMG	CONTROLS	IE 3.0
IMG	DYNSRC	IE 3.0
IMG	HEIGHT	HTML 3.2
IMG	HSPACE	HTML 3.2
IMG	ID	HTML 3.2
IMG	ISMAP	HTML 3.2
IMG	LOOP	IE 3.0
IMG	SRC	HTML 2
IMG	START	IE 3.0

IMG	STYLE	HTML 3.2
IMG	TITLE	HTML 2.0
IMG	USEMAP	IE 3.0
IMG	VSPACE	HTML 3.2
IMG	WIDTH	HTML 3.2
INPUT		HTML 2.0
INPUT	ALIGN	HTML 2.0
INPUT	CHECKED	HTML 2.0
INPUT	CLASS	HTML 2.0
INPUT	ID	HTML 2.0
INPUT	MAXLENGTH	HTML 2.0
INPUT	NAME	HTML 2.0
INPUT	NOTAB	HTML 2.0
INPUT	onBlur	HTML 3.2
INPUT	onChange	HTML 3.2
INPUT	onClick	HTML 3.2
INPUT	onFocus	HTML 3.2
INPUT	onSelect	HTML 3.2
INPUT	SIZE	HTML 2.0
INPUT	SRC	HTML 2.0
INPUT	STYLE	HTML 2.0
INPUT	TABINDEX	HTML 2.0
INPUT	TITLE	HTML 2.0
INPUT	TYPE	HTML 2.0
INPUT	VALUE	HTML 2.0
ISINDEX		HTML 2.0
ISINDEX	ACTION	Netscape
ISINDEX	PROMPT	HTML 3.2
KBD		HTML 2.0
LI		HTML 2.0
LI	CLASS	HTML 3.2
LI	ID	HTML 3.2
LI	STYLE	HTML 3.2
LI	TYPE	HTML 3.2
LI	VALUE	HTML 3.2
LINK		HTML 1.0
LINK	HREF	HTML 1.0
LINK	REL	HTML 1.0
LINK	REV	HTML 1.0
LINK	TITLE	HTML 1.0
LINK	TYPE	HTML 1.0
LISTING		HTML 2.0
MAP		HTML 2.0
MAP	NAME	IE 3.0

MARQUEE		IE 3.0
MARQUEE	ALIGN	IE 3.0
MARQUEE	BEHAVIOR	IE 3.0
MARQUEE	BGCOLOR	IE 3.0
MARQUEE	DIRECTION	IE 3.0
MARQUEE	HEIGHT	IE 3.0
MARQUEE	HSPACE	IE 3.0
MARQUEE	LOOP	IE 3.0
MARQUEE	SCROLLAMOUNT	IE 3.0
MARQUEE	SCROLLDELAY	IE 3.0
MARQUEE	VSPACE	IE 3.0
MARQUEE	WIDTH	IE 3.0
MENU		HTML 2.0
MENU	COMPACT	HTML 2.0
META		HTML 2.0
META	CHARSET	HTML 2.0
META	CONTENT	Netscape
META	HTTP-EQUIV	Netscape
META	NAME	HTML 3.2
META	URL	HTML 2.0
NOBR		Netscape
NOFRAMES		Netscape
OBJECT		HTML 3.2
OBJECT	ALIGN	HTML 3.2
OBJECT	BORDER	HTML 3.2
OBJECT	CLASSID	HTML 3.2
OBJECT	CODEBASE	HTML 3.2
OBJECT	CODETYPE	HTML 3.2
OBJECT	DATA	HTML 3.2
OBJECT	DECLARE	HTML 3.2
OBJECT	HEIGHT	HTML 3.2
OBJECT	HSPACE	HTML 3.2
OBJECT	NAME	HTML 3.2
OBJECT	NOTAB	HTML 3.2
OBJECT	SHAPES	HTML 3.2
OBJECT	STANDBY	HTML 3.2
OBJECT	TABINDEX	HTML 3.2
OBJECT	TITLE	HTML 3.2
OBJECT	TYPE	HTML 3.2
OBJECT	USEMAP	HTML 3.2
OBJECT	VSPACE	HTML 3.2
OBJECT	WIDTH	HTML 3.2
OL		HTML 2
OL	CLASS	HTML 3.2

OL	COMPACT	HTML 2
OL	ID	HTML 3.2
OL	START	HTML 3.2
OL	STYLE	HTML 3.2
OL	TYPE	HTML 3.2
OPTION		HTML 2
OPTION	SELECTED	HTML 2
OPTION	VALUE	HTML 2
P		HTML 2
P	ALIGN	HTML 3.2
P	CLASS	HTML 3.2
P	ID	HTML 3.2
P	STYLE	HTML 3.2
PARAM	NAME	HTML 3.2
PARAM	VALUE	HTML 3.2
PARAM	VALUETYPE	HTML 3.2
PARAM	TYPE	HTML 3.2
PLAINTEXT		HTML 2
PRE		HTML 2
S		HTML 2
SAMP		HTML 2
SCRIPT		HTML 3.2
SCRIPT	LANGUAGE	HTML 3.2
SCRIPT	SRC	HTML 3.2
SCRIPT	TYPE	HTML 3.2 W3C Extensions
SELECT		HTML 2
SELECT	NAME	HTML 2
SELECT	SIZE	HTML 2
SMALL		HTML 3.2
SPAN		HTML 2
SPAN	STYLE	HTML 3.2
STRIKE		HTML 2
STRONG		HTML 2
STYLE		IE 3.0
STYLE	TITLE	HTML 3.2
STYLE	TYPE	HTML 3.2
SUB		HTML 3.2
SUP		HTML 3.2
TABLE		HTML 2.0
TABLE	ALIGN	HTML 3.2
TABLE	BACKGROUND	IE 3.0
TABLE	BGCOLOR	IE 3.0
TABLE	BORDER	IE 3.2
TABLE	BORDERCOLOR	IE 3.0

TABLE	BORDERCOLORDARK	IE 3.0
TABLE	BORDERCOLORLIGHT	IE 3.0
TABLE	CELLPADDING	HTML 3.2
TABLE	CELLSPACING	HTML 3.2
TABLE	CLASS	HTML 3.2
TABLE	CLEAR	HTML 3.2
TABLE	COLS	HTML 3.2
TABLE	FRAME	HTML 3.2
TABLE	ID	HTML 3.2
TABLE	NEEDS	HTML 3.2
TABLE	NOWRAP	HTML 3.2
TABLE	RULES	HTML 3.2
TABLE	STYLE	HTML 3.2
TABLE	VALIGN	HTML 3.2
TABLE	WIDTH	HTML 3.2
TBODY		HTML 3.2
TBODY	CLASS	HTML 3.2
TBODY	ID	HTML 3.2
TBODY	STYLE	HTML 3.2
TD		HTML 2.0
TD	ALIGN	HTML 3.2
TD	BACKGROUND	HTML 3.2
TD	BGCOLOR	IE 3.0
TD	BORDERCOLOR	IE 3.0
TD	BORDERCOLORDARK	IE 3.0
TD	BORDERCOLORLIGHT	IE 3.0
TD	CLASS	HTML 3.2
TD	COLSPAN	HTML 3.2
TD	HEIGHT	HTML 3.2
TD	ID	HTML 3.2
TD	NOWRAP	HTML 3.2
TD	ROWSPAN	HTML 3.2
TD	STYLE	HTML 3.2
TD	VALIGN	IE 3.0
TD	WIDTH	HTML 3.2
TEXTAREA		HTML 2.0
TEXTAREA	COLS	HTML 2.0
TEXTAREA	NAME	HTML 2.0
TEXTAREA	ROWS	HTML 2.0
TFOOT		HTML 3.2
TFOOT	CLASS	HTML 3.2
TFOOT	ID	HTML 3.2
TFOOT	STYLE	HTML 3.2

TH		HTML 2.0
TH	ALIGN	HTML 3.2
TH	BACKGROUND	HTML 3.2
TH	BGCOLOR	IE 3.0
TH	BORDERCOLOR	IE 3.0
TH	BORDERCOLORDARK	IE 3.0
TH	BORDERCOLORLIGHT	IE 3.0
TH	CLASS	HTML 3.2
TH	COLSPAN	HTML 3.2
TH	ID	HTML 3.2
TH	NOWRAP	HTML 3.2
TH	ROWSPAN	HTML 2
TH	STYLE	HTML 3.2
TH	VALIGN	IE 3.0
TH	WIDTH	HTML 3.2
THEAD		HTML 3.2
THEAD	ALIGN	HTML 3.2
THEAD	CLASS	HTML 3.2
THEAD	ID	HTML 3.2
THEAD	STYLE	HTML 3.2
THEAD	VALIGN	HTML 3.2
TITLE		HTML 2
TR		HTML 2
TR	ALIGN	HTML 2
TR	BGCOLOR	IE 3.0
TR	BORDERCOLOR	IE 3.0
TR	BORDERCOLORDARK	IE 3.0
TR	BORDERCOLORLIGHT	IE 3.0
TR	CLASS	HTML 3.2
TR	ID	HTML 3.2
TR	NOWRAP	HTML 3.2
TR	STYLE	HTML 3.2
TR	VALIGN	IE 3.0
TT		HTML 2.0
U		HTML 2.0
UL		HTML 2.0
UL	CLASS	HTML 2.0
UL	COMPACT	HTML 2.0
UL	ID	HTML 2.0
UL	STYLE	HTML 2.0
VAR		HTML 2
WBR		Netscape
XMP		HTML 2

HTML Reference

Tag	Description
!--	Comments. Any text between tags will not display in the browser.
!DOCTYPE	Describes the HTML version used in the current document.
A	Stands for *anchor*. Attributes create hyperlinks and named references.
ADDRESS	Specifies a mailing address.
APPLET	Embeds a Java applet. See OBJECT.
AREA	Specifies the shape of a "hot spot" in a client-side image map.
B	Changes text to bold. See STRONG.
BASE	Specifies a document's URL.
BASEFONT	Sets the base font value.
BGSOUND	Adds background sounds that play on initial load.
BIG	Enlarges the font size.
BLOCKQUOTE	Sets apart a quotation in text.
BODY	Specifies the beginning and the end of the document body. See HEAD.
BR	Inserts a line break.
CAPTION	Specifies a caption for a table. Valid only within the TABLE element.
CENTER	Centers text and images.
CITE	Indicates a citation. Used to present a book, paper, or other published source material.
CODE	Presents a code sample.
COL	Sets the properties of a column.
COLGROUP	Sets the properties of one or more columns as a group.
COMMENT	Indicates a comment. Text in a comment element does not display in a browser.
DD	Specifies definition data. Used to format the text for a definition. See DL, DT.
DFN	Specifies a definition. Formats a defined term.
DIR	Denotes a directory list.
DIV	Sets a document division. Groups related elements together within a document.
DL	Denotes a definition list. Used for a list of defined terms. See DT, DD.
DT	Specifies a definition term. Used to format the defined term. See DL, DD.
EM	Emphasizes text, usually by rendering text in italic.
EMBED	Indicates an embedded object. See OBJECT.
FONT	Formats the font style, size, and color.
FORM	Denotes a form with which users enter data. See INPUT for a list of form elements.
FRAME	Defines independent windows, or frames, within a page. See FRAMESET.
FRAMESET	Defines layout for frames within a page. See FRAME.
Hn	Renders text in heading style, usually with a larger font than the body text. The *n* is a value from 1 to 6.
HEAD	Marks the HTML document heading.
HR	Draws a horizontal rule. Used to separate sections.
HTML	Denotes the file is an HTML document.
I	Renders text in italic.

IMG	Inserts a graphic file.
INPUT	Specifies a form control such as a check box or radio button. See FORM.
ISINDEX	Indicates the presence of an index.
KBD	Indicates text to be entered at a keyboard. Appears in fixed-width and bold type.
LI	Denotes an item in a list. Adds special character or number depending on use. See UL, OL.
LINK	Establishes the relationship between documents. Appears only in the HEAD element.
LISTING	Renders text in fixed-width type.
MAP	Specifies a collection of hot spots for a client-side image map.
MARQUEE	Displays text in a scrolling marquee.
MENU	Denotes a list of items.
META	Provides information about the document. Used for client pull, also by some search engines for indexing. Describes the character set to be used.
NOBR	Turns off line breaking.
NOFRAMES	Indicates content viewable only by browsers that do not support frames.
OBJECT	Inserts an OLE Control.
OL	Specifies an ordered list. Each item has a number or letter reference. See UL, LI.
OPTION	Denotes one choice in a list box.
P	Inserts a paragraph break and denotes a new paragraph.
PARAM	Sets object properties.
PLAINTEXT	Renders text in fixed-width type without processing elements.
PRE	Displays text exactly as typed—with all line breaks and spacing.
S	Renders text in strikethrough type.
SAMP	Specifies sample text. See CODE.
SCRIPT	Specifies the inclusion of a script.
SELECT	Denotes a list box or drop-down list.
SMALL	Decreases the font size.
SPAN	Applies style information to the enclosed text.
STRIKE	Renders text in strikethrough type. See S.
STRONG	Emphasizes text, usually with bold. See B.
STYLE	Allows authors to include rendering information using a specified style notation.
SUB	Renders text in subscript.
SUP	Renders text in superscript.
TABLE	Creates a table. See TH, TR, and TD to learn how to define rows and columns.
TBODY	Defines the table body.
TD	Creates a cell in a table.
TEXTAREA	Creates a box in which a user can enter and edit text.
TFOOT	Defines the table footer.
TH	Creates a row or column heading in a table.
THEAD	Defines the table header.
TITLE	Specifies a document title. Appears in the browser title bar.
TR	Creates a row in a table.

TT Denotes teletype. Displays text in fixed-width type.

U Renders text underlined.

UL Formats lines of text as a bulleted list. See LI.

VAR Indicates placeholder text for a variable. Displays text in a small, fixed-width font.

WBR Inserts a soft line break in a block of NOBR text.

XMP Indicates example text. Displays text in fixed-width font.

!--

<!-- .. -->

Specifies that enclosed text is an author's comment. Text may be multi-line. Any text between the tags will not appear in an HTML browser. You can include lines of text between the start-tag and end-tag.

Example

```
<!--This line of text, enclosed in an HTML page, will not display.
  This second line of text will not display either.-->
```

!DOCTYPE

<!DOCTYPE>

Specifies the version of HTML used in the document. !DOCTYPE is the first element in any HTML document. !DOCTYPE is a required element for any HTML 3.2-compliant document.

Example

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">
```

A

<A
CLASS=*type*
HREF=*reference*
ID=*value*
NAME=*name*
onClick=*function*
onMouseOver=*function*
REL=SAME|NEXT|PARENT|PREVIOUS
REV=*value*
STYLE=*style*
TARGET=*window*
TITLE=*title***>**
</A>

Stands for anchor. The end-tag is required.

CLASS=*type*

Indicates the class to which the element belongs.

HREF=*reference*

Specifies either a destination address or a destination file. A destination address must be in URL format. A destination file must name a file and be in the format of the given file system. If no path or domain name is specified, the file is searched for in the same location as the current document.

ID=*value*

Specifies a unique value for the element over the document.

NAME=*name*

Specifies a named reference within an HTML document.

onClick=*function*

Activates a script when the onClick event occurs.

onMouseOver=*function*

Activates a script when the onMouseOver event occurs.

REL=SAME|NEXT|PARENT|PREVIOUS

Defines the relationship of a link. The default is SAME.

SAME	The author of the linked document is the same as the current document.
NEXT	The link is the next page in a sequence.
PARENT	The current page is the parent of a destination document.
PREVIOUS	The link is to the previous document.

REV=*value*

Specifies the reverse link.

STYLE=*style*

Specifies style information.

TARGET=*window*

Specifies to load the link into the targeted window. This attribute can be used with a frameset where a frame has been named in the FRAME element. The *window* can be one of these values:

window	Specifies to load the link into the targeted window. The *window* must begin with an alphanumeric character to be valid, except for the following four target windows:
blank	Load the link into a new blank window. This window is not named.
parent	Load the link into the immediate parent of the document the link is in.
self	Load the link into the same window the link was clicked in.

top	Load the link into the full body of the window.

TITLE=*title*

Specifies the title that appears when the hyperlink is selected.

The properties of elements that can follow A are applied to the data characters or elements in the container. The anchor element is used to link text or other elements using the **HREF=** attribute. The anchor element is used to specify text or graphics as a named reference, to which hyperlinks can link, using the **NAME=** attribute. Anchors cannot be nested.

Example

```
<A HREF="http://www.microsoft.com"> This is a link to Microsoft.</A>

<A HREF="home.htm">This is a link to a file called home.htm in the same
directory as this page.</A>

<A TARGET="viewer" HREF="sample.htm">Click here to load the link into
"viewer" window.</A>
```

ADDRESS

<ADDRESS>
&lt/ADDRESS>

Specifies the mailing address. This element typically is used at the bottom of a document. Text is displayed in italic. The end-tag is required.

Example

```
<ADDRESS>This text will be in italic.</ADDRESS>
```

APPLET

<**APPLET**
ALIGN=LEFT|CENTER|RIGHT|TOP|MIDDLE|BOTTOM
ALT=*alternateText*
CODE=*appletFile*
CODEBASE=*codebaseURL*
DOWNLOAD=*n*
HEIGHT=*pixels*
HSPACE=*pixels*
NAME=*appletInstanceName*
TITLE=*text*
VSPACE=*pixels*
WIDTH=*pixels*>
...
<**PARAM NAME=***AttributeName* **...>**
...
<**/APPLET>**

Embeds a Java applet in an HTML document. The end-tag is required.

ALIGN=LEFT|CENTER|RIGHT|TOP|MIDDLE|BOTTOM

Describes the alignment of an object to text. The default is LEFT.

LEFT	The applet is aligned to the left of surrounding text.
CENTER	The applet is aligned to the center of surrounding text.
RIGHT	The applet is aligned to the right of surrounding text.
TOP	The applet is aligned to the top of surrounding text.
MIDDLE	The applet is aligned to the middle of surrounding text.
BOTTOM	The applet is aligned to the bottom of surrounding text.

ALT=*alternateText*

Specifies alternate text for text-only browsers or browsers that do not support Java.

CODE=*appletFile*

Specifies the name of the Java applet.

CODEBASE=*codebaseURL*

Specifies the base URL of the applet (the directory in which the applet is located).

DOWNLOAD=*n*

Specifies the image download order.

HEIGHT=*pixels*

Specifies the initial height of the applet display area.

HSPACE=*pixels*

Specifies the horizontal space in which the applet displays.

NAME=*appletInstanceName*

Identifies an applet to other applets within the HTML page.

PARAM NAME=*AttributeName*

Passes applet-specific arguments in from an HTML page. There may be no, one, or any number of **PARAM NAME=** attributes present in the **APPLET** start-tag.

TITLE=*text*

Specifies an advisory title string.

VSPACE=*pixels*

Specifies the space in pixels above the applet.

WIDTH=*pixels*

Specifies the initial width of the applet display area.

Applet resources are loaded relative to the document URL. The **CODEBASE=** attribute is used to change this default behavior. If the **CODEBASE=** attribute is defined, it specifies a different location in which to find applet resources. The value can be an absolute or a relative URL. The absolute URL is used as is, without modification, and is not affected by the document's BASE element. When the **CODEBASE=** attribute is relative, it is relative to the document URL (or BASE tag, if defined).

Example

```
<APPLET CODEBASE="applets/javaapps"
  CODE=JavaDemo.class
  WIDTH=100
  HEIGHT=200>
<PARAM NAME=text value="This is a Java applet.">
<IMG SRC=logo.gif ALT="Upgrade to Internet Explorer to view this Java
applet.">
</APPLET>
```

AREA

<AREA
ALT=*text*
CLASS=*type*
COORDS=*coords*
HREF=*url*
ID=*value*
NOHREF
NOTAB
SHAPE=*shape-type*
STYLE=*css1* **properties**
TABINDEX=*n*
TARGET=*window*
TITLE=*text*>

Specifies the shape of a "hot spot" in a client-side image map.

ALT=*text*
Specifies the alternative display for text-only browsers.

CLASS=*type*
Indicates the class to which the element belongs.

COORDS=*coords*
Specifies coordinates that define the hot spot's shape.

HREF=*url*
Specifies the destination of the hot spot.

ID=*value*
Specifies a unique value for the element over the document.

NOHREF
Indicates that clicks in this region should cause no action.

NOTAB
Excludes the element from the tabbing order.

SHAPE=*shape-type*
Denotes the type of shape. The *shape-type* can be one of these values:

RECT	Rectangle. Takes four coordinates: *x1*, *y1*, *x2*, and *y2*. This is the default.
RECTANGLE	Rectangle. Takes four coordinates: *x1*, *y1*, *x2*, and *y2*.
CIRC	Circle. Takes three coordinates: *centerx*, *centery*, and *radius*.
CIRCLE	Circle. Takes three coordinates: *centerx*, *centery*, and *radius*.
POLY	Polygon. Takes three or more pairs of coordinates denoting a polygonal region.
POLYGON	Polygon. Takes three or more pairs of coordinates denoting a polygonal region.

STYLE=*css1 properties*
Specifies style information.

TABINDEX=*n*
Specifies the position in the tabbing order.

TARGET=*window*
Specifies to load the link into the targeted window. The *window* can be one of these values:

window	Specifies to load the link into the targeted window. The *window* must begin with an alphanumeric character to be valid, except for the following four target windows:
_blank	Load the link into a new blank window. This window is not named.

_parent Load the link into the immediate parent of the document the link is in.

_self Load the link into the same window the link was clicked in.

_top Load the link into the full body of the window.

TITLE=*text*

Specifies an advisory title for balloon help.

Examples

```
<AREA SHAPE="RECT" COORDS="50, 25, 150, 125" HREF="http://www.sample.com">

<AREA SHAPE="RECT" COORDS="50, 25, 150, 125" NOHREF>

<AREA TARGET="viewer" HREF="sample.htm" SHAPE="CIRCLE" COORDS="50, 25,
150, 125">
```

B

<B>
</B>

Renders text in bold. The end-tag turns off the bold formatting.

Example

```
<B>Displayed in a bold typeface.</B>
```

BASE

<BASE
HREF=*url*
TARGET=*window***>**

Specifies the document's URL.

HREF=*url*

Specifies the document's full URL in case the document gets read out of context and the reader wants to refer to the original.

TARGET=*window*

Specifies to load all the links on the page into the targeted window. This can be overridden by specifying a different target attribute for a specific link. The *window* can be one of these values:

window	Specifies to load the link into the targeted window. The *window* must begin with an alphanumeric character to be valid, except for the following four target windows:
_blank	Load the link into a new blank window. This window is not named.
_parent	Load the link into the immediate parent of the document the link is in.
_self	Load the link into the same window the link was clicked in.
_top	Load the link into the full body of the window.

Examples

```
<BASE HREF="http:// www.sample.com/hello.htm">
```

```
<BASE HREF="http:// www.sample.com/hello.htm" TARGET="viewer">
```

BASEFONT

<**BASEFONT**
    COLOR=*color*
    NAME=*name*
    SIZE=*n*>

Sets the base font value. This value will be used as a default for any text not formatted with a style sheet or using the FONT element.

COLOR=*color*

Specifies the color of the base font.

NAME=*name*

Specifies the name of the base font.

SIZE=*n*

Specifies the size of the base font. The *n* can be between 1 and 7 inclusive; default is 3; 7 is largest. Throughout the document, relative font size settings (for example, <FONT SIZE=+3>) are set according to this.

Example

```
<BASEFONT SIZE=3> This sets the base font size to 3.
<FONT SIZE=+4> Now the font size is 7.
<FONT SIZE=-1> Now the font size is 2.
```

BGSOUND

<**BGSOUND**
    LOOP=*n*
    SRC=*url*>

Adds background sounds or "soundtracks" to a page. Sounds can either be samples (.wav or .au format) or MIDI format.

LOOP=*n*

Specifies how many times a sound will loop when activated. If *n*=-1, or if **LOOP=**INFINITE is specified, it will loop indefinitely.

SRC=*url*

Specifies the address of a sound to be displayed.

BIG
<BIG>

Makes text one size larger.

Example

```
<BIG>This text is larger.</BIG>
```

BLOCKQUOTE

<BLOCKQUOTE>
</BLOCKQUOTE>

Indents both left and right margins. Used to set apart quotations in text. The end-tag is required.

Example

```
<P>He said,
<BLOCKQUOTE>"Hi there!"</BLOCKQUOTE>
```

BODY

<**BODY**
ALINK=*color*
BACKGROUND=*url*
BGCOLOR=*color*
BGPROPERTIES=FIXED
CLASS=*type*
ID=*value*
LEFTMARGIN=*n*
LINK=*color*
STYLE=*css1* **properties**
TEXT=*color*
TOPMARGIN=*n*
VLINK=*color*>
<**/BODY>**

Specifies the beginning and end of the document body. This element also allows you to set the background image, the background color, the link colors, and the top and left margins of the page. The end-tag is required.

ALINK=*color*
Specifies the color of the active hyperlink.

BACKGROUND=*url*
Specifies a background picture. The picture is tiled behind the text and graphics on the page.

BGCOLOR=*color*
Sets the background color of the page. The *color* can be either a hexadecimal, red-green-blue color value or a predefined color name. See Color.

BGPROPERTIES=FIXED
Specifies a "watermark," which is a background picture that does not scroll. The background image remains fixed behind any scrolling foreground information.

CLASS=*type*
Indicates the class to which the element belongs.

ID=*value*
Specifies a unique value for the element over the document.

LEFTMARGIN=*n*
Specifies the left margin for the entire body of the page and overrides the default margin. If set to zero, the left margin will be exactly on the left edge.

LINK=*color*
Sets the color of hyperlinks that have not yet been visited. The *color* can be either a hexadecimal, red-green-blue color value or a predefined color name. See Color.

STYLE=*css1 properties*
Specifies style information.

TEXT=*color*
Sets the color of text on the page. The *color* can be either a hexadecimal, red-green-blue color value or a predefined color name. See Color.

TOPMARGIN=*n*
Specifies the margin for the top of the page and overrides the default margin. If set to zero, the top margin will be on the precise top edge.

VLINK=*color*
Sets the color of hyperlinks that have already been visited. The *color* can be either a hexadecimal, red-green-blue color value or a predefined color name. See Color.

Examples

The HTML used to insert the background image of this page is:

```
<BODY BACKGROUND="/ie/images/watermrk.gif" BGPROPERTIES=FIXED
BGCOLOR=#FFFFFF TEXT=#000000 LINK=#ff6600 VLINK=#330099>
```

```
<HTML><BODY>Here's a Web page!</BODY></HTML>
```

BR

<**BR**
  CLASS=*type*
  CLEAR=NONE|LEFT|RIGHT|ALL>

Inserts a line break.

CLASS=*type*

Indicates the class to which the element belongs.

CLEAR=NONE|LEFT|RIGHT|ALL

Controls the display of text below or beside an image. The default is NONE.

NONE	The next text in the document appears immediately after the image.
LEFT	The next text in the document is displayed below the image at the next full left margin.
RIGHT	The next text in the document is displayed below the image at the next full right margin.
ALL	The next text in the document is displayed below the image at the next full left and right margins.

CAPTION

<CAPTION
ALIGN=CENTER|LEFT|RIGHT
VALIGN=TOP|BOTTOM>
</CAPTION>

Specifies a caption for a table. This element is valid only within the TABLE element. The end-tag is required.

ALIGN=CENTER|LEFT|RIGHT

Sets the alignment of the caption to the table. The default is CENTER.

CENTER	The caption is centered above the margins of the table.
LEFT	The caption is aligned to the left side of the table.
RIGHT	The caption is aligned to the right side of the table.

VALIGN=TOP|BOTTOM

Sets the vertical alignment of the table caption. The default is TOP.

TOP	The caption is at the top of the table.
BOTTOM	The caption is at the bottom of the table.

Example

```
<TABLE>
<CAPTION VALIGN=BOTTOM>
This caption will appear below the table.
</CAPTION>
<TR>
     ....
</TR>
</TABLE>
```

CENTER

<CENTER>
</CENTER>

Centers text and images. The end-tag returns the alignment to its previous state.

Example

```
<CENTER>This text appears centered on the page.</CENTER>
```

CITE

<CITE>
</CITE>

Indicates a citation. Refers to a book, paper, or other published source material. The end-tag is required.

Example

```
<CITE>Book Title.</CITE>
```

CODE

<CODE>
</CODE>

Specifies a code sample. Renders text in a small font. (If no font face is specified, the font used is fixed-width.) The end-tag is required.

Example

```
<CODE>Here is some text in a small, fixed-width font.</CODE>
```

COL

<COL
    ALIGN=LEFT|CENTER|RIGHT
    SPAN=*n*>

Sets the properties of one or more columns. Use this element in conjunction with a COLGROUP element to set the properties of a column within a group of columns.

ALIGN=CENTER|LEFT|RIGHT

Specifies the text alignment in cells within the column. The default is CENTER.

CENTER	Text is aligned with the center of cells in the column.
LEFT	Text is aligned to the left of cells in the column.
RIGHT	Text is aligned to the right of cells in the column.

SPAN=*n*

Sets the number of consecutive columns for which the properties are set.

This element is valid only within a table. The end-tag is not required and is not recommended.

The properties specified by the COL element always override the properties specified by the preceding COLGROUP element.

Example

```
<TABLE>
<COLGROUP>
    <COL ALIGN=RIGHT>
    <COL ALIGN=LEFT>
<COLGROUP>
    <COL ALIGN=CENTER>
<TBODY>
    <TR>
    <TD>This is the first column in the group and is right-aligned.</TD>
    <TD>This is the second column in the group and is left-aligned.</TD>
    <TD>This column is in a new group and is centered.</TD>
    </TR>
</TABLE>
```

COLGROUP

<COLGROUP
HALIGN=CENTER|LEFT|RIGHT
SPAN=*n*
VALIGN=MIDDLE|TOP|BOTTOM
WIDTH=*n***>**

Sets the properties of one or more columns.

HALIGN=CENTER|LEFT|RIGHT

Specifies the horizontal alignment of text in the cells for the column group specified in the element. The default is CENTER.

CENTER	Text is aligned in the center of cells in the column.
LEFT	Text is aligned to the left of cells in the column.
RIGHT	Text is aligned to the right of cells in the column.

SPAN=*n*

Sets the number of consecutive columns that are in the group and for which the properties are set.

VALIGN=MIDDLE|TOP|BOTTOM

Specifies the vertical alignment of text within the cells specified in the columns. The default is MIDDLE.

MIDDLE	Text is aligned in the middle of cells specified in the column group.
TOP	Text is aligned to the top of cells specified in the column group.
BOTTOM	Text is aligned to the bottom of cells specified in the column group.
WIDTH=*n*	Indicates the width of the columns in the column group.

This element is valid only within a table. The end-tag is not required and is not recommended.

If the columns in a group of columns require varying properties, use **COLGROUP** in conjunction with one or more COL elements to individually set the properties for the columns.

This element affects how rules are drawn within a table when groups are specified with the RULES= attribute in the TABLE element. In this case, vertical rules are drawn between column groups rather than between individual columns.

Example

```
<TABLE>
<COLGROUP ALIGN=RIGHT>
<COLGROUP SPAN=2 ALIGN=LEFT>
<TBODY>
    <TR>
    <TD>This column is in the first group and is right-aligned.</TD>
    <TD>This column is in the second group and is left-aligned.</TD>
    <TD>This column is in the second group and is left-aligned.</TD>
    </TR>
</TABLE>
```

COMMENT

<COMMENT>
</COMMENT>

Indicates a comment. The text between the elements is ignored, unless it contains HTML code.

This element is provided for backward compatibility. See the entry for above.

Example

```
<COMMENT>This won't be printed.</COMMENT>
```

DD

<DD
  CLASS=*type*
  ID=*value*
  STYLE=*css1* **properties>**

Indicates that the following text is a definition of a term, and therefore should be displayed in the right-hand column of a definition list.

CLASS=*type*
  Indicates the class to which the element belongs.

ID=*value*
  Specifies a unique value for the element over the document.

STYLE=*css1 properties*
  Specifies style information.

Example

```
<DL><DT>Cat<DD>A small domesticated mammal.
<DT>Lizard<DD>A reptile generally found in dry areas.</DL>
```

DFN

<DFN>

Specifies a definition. Formats a term for its first appearance in a document.

Example

```
<DFN>HTML stands for hypertext markup language.</DFN>
```

DIR

<DIR

COMPACT>

</DIR>

Specifies that the following text is a directory list that consists of individual items, each beginning with an LI element and none containing more than 20 characters, that should be displayed in columns. The end-tag is required.

COMPACT

Speficies a compact list style. The entries in the list appear closer together.

Example

```
<DIR><LI>Art
<LI>History
<LI>Literature
<LI>Sports
<LI>Entertainment
<LI>Science
</DIR>
```

DIV

<DIV
ALIGN=LEFT|CENTER|RIGHT|JUSTIFY
CLASS=*type*
ID=*value*
LANG=*iso*
NOWRAP>
</DIV>

Represents different kinds of containers, for example, chapter, section, abstract, or appendix. The end-tag is required.

ALIGN=LEFT|CENTER|RIGHT|JUSTIFY

Specifies the default horizontal alignment for the contents of this element. This is needed for compatibility with deployed browsers and can be overridden by style sheets. The default is LEFT.

LEFT	Text is aligned to the left.
CENTER	Text is centered.
RIGHT	Text is aligned to the right.
JUSTIFY	Text is justified to the right and left margins.

CLASS=*type*

Indicates the class to which the element belongs.

ID=*value*

Specifies a unique value for the element over the document.

LANG=*iso*

Indicates the ISO standard language. Enter the standard abbreviation to indicate the language of the element.

NOWRAP

Specifies that the text lines within a document are not automatically wrapped by the browser.

Example

```
<DIV>
This text represents a section.
</DIV>

<DIV ALIGN=CENTER>
This text represents another section, and its text is centered.
</DIV>
```

DL

<DL
CLASS=*type*
COMPACT
ID=*value*
STYLE=*css1* **properties>**
</DL>

Specifies that the following block is a definition list, that is, an automatically formatted list with terms on the left and their definitions indented below. The end-tag is required.

See DT (directory term) and DD (directory definition) for a description of elements that appear within a directory list.

CLASS=*type*

Indicates the class to which the element belongs.

COMPACT

Specifies that the list should be organized in a compact style. Each entry appears closer together.

ID=*value*

Specifies a unique value for the element over the document.

STYLE=*css1 properties*

Specifies style information.

Example

```
<DL>
<DT>Cat
<DD> A small domesticated mammal.
<DT>Lizard
<DD>A reptile generally found in dry areas.
</DL>
```

DT

<DT>

Specifies a term in a definition list. Indicates that the text is a term to be defined, and should therefore be displayed in the left-hand column of a definition list.

Example

```
<DL>
<DT>Cat
<DD> A small domesticated mammal.
<DT>Lizard
<DD>A reptile generally found in dry areas.
</DL>
```

EM

<EM>
</EM>

Emphasizes text, usually by rendering it in italic. The end-tag is required.

Example

```
<EM>This text will be in italics.</EM>
```

EMBED

<EMBED
    HEIGHT=*n*
    NAME=*text*
    [**optional parameter=***&quot;value&quot;*]
    PALETTE=*#rgb***|#rgb**
    SRC=*url*
    UNITS=PIXELS|EN
    WIDTH=*n***>**
    </EMBED>

Indicates an embedded object. OBJECT is the preferred element for inserting objects, but EMBED is included for backward compatibility with earlier HTML documents. See OBJECT.

The end-tag is required.

HEIGHT=*n*

Specifies the height, in pixels, of the object on the page.

NAME=*text*

Indicates the name used by other objects or elements to refer to this object.

optional parameter=*"value"*

Specifies any parameters that are specific to the object. In the example that follows, AUTOSTART and PLAYBACK are optional parameters.

PALETTE=*#rgb|#rgb*

Sets the color palette to the foreground or background color. The first rgb value specifies the foreground color. The second rgb value specifies the background color. For example, **PALETTE=#aa0000|#001100** sets the foreground color to red and the background color to green.

SRC=*url*

Specifies the name of any source data input to the object.

UNITS=PIXELS|EN

Specifies the measurement unit used by the **HEIGHT=** and **WIDTH=** attributes.

PIXELS	The default size.
EN	Half the point size.

WIDTH=*n*

Specifies the width of the object, in pixels, on the page.

Example

```
<EMBED SRC=&quot;MyMovie.AVI&quot; WIDTH=100 HEIGHT=250 AUTOSTART="TRUE"
PLAYBACK="FALSE">
```

FONT

<**FONT**
COLOR=*color*
FACE=*name*
SIZE=*n*>

Sets the size, font, and color of text.

COLOR=*color*

Sets font color. The *color* can be either a hexadecimal, red-green-blue color value or a predefined color name. See Color.

FACE="*name* [,*name2*[,*name3*]]"

Sets the font. A list of font names can be specified. If the first font is available on the system, it will be used; otherwise, the second will be tried, and so on. If none are available, a default font will be used.

SIZE=*n*

Specifies font size between 1 and 7 (7 is largest). A plus or minus before the number indicates a size relative to the current BASEFONT setting. Relative font sizes are not cumulative, so putting two <FONT SIZE="+1"> elements in a row does not result in the font size being increased by 2.

FORM

<FORM
ACTION=*url*
METHOD=GET|POST
onSubmit=*event*
TARGET=*window***>**
</FORM>

Denotes a form. The end-tag is required.

ACTION=*url*

Specifies the address to be used to carry out the action of the form. If none is specified, the base URL of the document is used.

METHOD=GET|POST

Indicates how the form data should be sent to the server. The default is GET.

GET	Appends the arguments to the action URL and opens it as if it were an anchor.
POST	Sends the data via an HTTP post transaction.

onSubmit=*event*

References an event, possibly a script statement such as a VBScript functional call, when the form is sent to the server.

TARGET=*window*

Specifies to load the results of the form submission into the targeted window. The *window* can be one of these values:

window	Specifies to load the link into the targeted window. The *window* must begin with an alphanumeric character to be valid, except for the following four target windows:
blank	Load the link into a new blank window. This window is not named.
parent	Load the link into the immediate parent of the document the link is in.
self	Load the link into the same window the link was clicked in.
top	Load the link into the full body of the window.

Example

```
<FORM TARGET="viewer" ACTION="http://www.sample.com/bin/search">
    ...
</FORM>
```

FRAME

<FRAME
ALIGN=LEFT|CENTER|RIGHT|TOP|BOTTOM
FRAMEBORDER=1|0
MARGINHEIGHT=*height*
MARGINWIDTH=*width*
NAME=*name*
NORESIZE
SCROLLING=yes|no
SRC=*address>*

Defines a single frame in a frameset. There is no matching end-tag.

ALIGN=LEFT|CENTER|RIGHT|TOP|BOTTOM

Sets the alignment of the frame or of the surrounding text. The default is LEFT.

LEFT	The frame is drawn as a left-flush "floating frame," and text flows around it.
CENTER	Surrounding text is aligned with the center of the frame.
RIGHT	The frame is drawn as a right-flush "floating frame," and text flows around it.
TOP	Surrounding text is aligned with the top of the frame.
BOTTOM	Surrounding text is aligned with the bottom of the frame.

FRAMEBORDER=1|0

Renders a 3-D edge border around the frame. 1 (default) inserts a border. 0 displays no border.

MARGINHEIGHT=*height*

Controls the margin height for the frame, in pixels.

MARGINWIDTH=*width*

Controls the margin width for the frame, in pixels.

NAME=*name*

Provides a target name for the frame.

NORESIZE

Prevents the user from resizing the frame.

SCROLLING=yes|no

Creates a scrolling frame.

SRC=*address*

Displays the source file for the frame.

Example

```
<FRAME FRAMEBORDER=0 SCROLLING=NO SRC="sample.htm">
```

FRAMESET

<FRAMESET
COLS=*col-widths*
FRAMEBORDER=1|0
FRAMESPACING=*spacing*
ROWS=*row-heights*>
&lt/FRAMESET>

Hosts the FRAME, **FRAMESET**, and NOFRAMES elements. FRAMESETs can be nested within each other to have layouts within a frame. Each frameset can exist at the same level as a frame.

The end-tag is required.

COLS=*col-widths*
Creates a frame document with columns. You can specify the column dimensions by percentage (%), pixels, or a relative size (*).

FRAMEBORDER=1|0
Provides the option to display or not display a 3-D border for a frame. 1 (default) sets a frame border. 0 displays no border.

FRAMESPACING=*spacing*
Creates additional space between frames, in pixels.

ROWS=*row-heights*
Creates a frame document with rows. You can specify the row dimensions by percentage (%), pixels, or a relative size (*).

The **FRAMEBORDER=** and **FRAMESPACING=** attributes are inherited from any containing FRAMESET element, which means you need only set the attribute on the single, outermost FRAMESET tag to affect all FRAME tags on that page.

Example

```
<FRAMESET SCROLLING=YES COLS="25%, 50%, *">
    <FRAME SRC="contents.htm">
    <FRAME SRC="info.htm">
    <FRAME SCROLLING=NO SRC="graphic.htm">
</FRAMESET>
```

Hn

<H*n*
    ALIGN=LEFT|CENTER|RIGHT>
</H*n***>**

Renders text in heading style. Use H1 through H6 to specify different sizes and styles of headings. The end-tag (required) restores the formatting to normal.

n
    Sets the heading level. This is an integer from 1 to 6.

ALIGN=LEFT|CENTER|RIGHT
    Sets the alignment of heading text. The default is LEFT.

LEFT	Text is aligned to the left.
CENTER	Text is aligned to the center.
RIGHT	Text is aligned to the right.

Example

```
<H1>Welcome to Internet Explorer!</H1>
```

HEAD

<HEAD>
        </HEAD>

Marks the HTML document heading.

The end-tag is required to donote closure for the document heading. Typically the BODY begin-tag appears after the HEAD end-tag.

Example

```
<HEAD>
<TITLE>A Simple Document</TITLE>
</HEAD>
```

HR

<HR
ALIGN=LEFT|CENTER|RIGHT
CLASS=*type*
COLOR=*color*
ID=*value*
NOSHADE
SIZE=*n*
STYLE=*css1* **properties**
WIDTH=*n***>**

Draws a horizontal rule.

ALIGN=CENTER|LEFT|RIGHT

Sets the alignment of the rule. The default is CENTER.

CENTER	Rule is centered on the page.
LEFT	Rule is aligned to the left of the page.
RIGHT	Rule is aligned to the right of the page.

CLASS=*type*

Indicates the class to which the element belongs.

COLOR=*color*

Sets the color of the rule. The *color* can be either a hexadecimal, red-green-blue color value or a predefined color name. See Color.

ID=*value*

Specifies a unique value for the element over the document.

NOSHADE

Draws the rule without 3-D shading.

SIZE=*n*

Sets the height of the rule, in pixels.

STYLE=*css1 properties*

Specifies style information.

WIDTH=*n*

Sets the width of the rule, either in pixels or as a percentage of window width. To specify a percentage, the *n* must end with the percent (%) sign.

Example

```
<HR SIZE=5 WIDTH=80% NOSHADE>
```

HTML

<HTML>
</HTML>

Denotes the file as an HTML document. The begin-tag typically appears after the !DOCTYPE element. The end-tag comes after all HTML elements in the document.

This element has no attributes.

Example

```
<HTML>
<BODY>
<P>This is an HTML document.
</BODY>
</HTML>
```

I

<I>
</I>

Renders text in italic. The end-tag turns off the italic formatting.

Example

```
<I>This text will be in italic.</I>
```

IFRAME

<IFRAME
ALIGN=LEFT|CENTER|RIGHT|TOP|BOTTOM
FRAMEBORDER=1|0
HEIGHT=*height*
MARGINHEIGHT=*height*
MARGINWIDTH=*width*
NAME=*name*
SCROLLING=yes|no
SRC=*address*
WIDTH=*width*>
</IFRAME>

Defines a floating frame. The end-tag is required.

ALIGN=LEFT|CENTER|RIGHT|TOP|BOTTOM

Sets the alignment of the frame or of the surrounding text. The default is LEFT.

LEFT	The frame is drawn as a left-flush "floating frame," and text flows around it.
CENTER	Surrounding text is aligned with the center of the frame.
RIGHT	The frame is drawn as a right-flush "floating frame," and text flows around it.
TOP	Surrounding text is aligned with the top of the frame.
BOTTOM	Surrounding text is aligned with the bottom of the frame.

FRAMEBORDER=1|0

Renders a 3-D edge border around the frame. 1 (default) inserts a border. 0 displays no border.

HEIGHT=*height*

Controls the height (in pixels) of the floating frame.

MARGINHEIGHT=*height*

Controls the margin height for the frame, in pixels.

MARGINWIDTH=*width*

Controls the margin width for the frame, in pixels.

NAME=*name*

Provides a target name for the frame.

SCROLLING=yes|no

Creates a scrolling frame.

SRC=*address*

Displays the source file for the frame.

WIDTH=*width*

Controls the width of the floating frame, in pixels.

Example

```
<IFRAME FRAMEBORDER=0 SCROLLING=NO SRC="sample.htm"></IFRAME>
```

IMG

<IMG
ALIGN=TOP|MIDDLE|CENTER|BOTTOM|LEFT|RIGHT
ALT=*text*
BORDER=*n*
CLASS=*type*
CONTROLS
DYNSRC=*url*
HEIGHT=*n*
HSPACE=*n*
ID=*value*
ISMAP=*image*
LOOP=*n*
SRC=*url*
START=*start-event*
STYLE=*css1* **properties**
TITLE=*text*
USEMAP=*url*
VSPACE=*n*
WIDTH=*n***>**

Inserts an image.

ALIGN=TOP|MIDDLE|CENTER|BOTTOM|LEFT|RIGHT

Sets the alignment of the image or of the surrounding text. The default is TOP.

TOP	Surrounding text is aligned with the top of the image.
MIDDLE	Surrounding text is aligned with the middle of the image.
CENTER	Surrounding text is aligned with the center of the image.
BOTTOM	Surrounding text is aligned with the bottom of the image.
LEFT	The picture is drawn as a left-flush "floating image," and text flows around it.
RIGHT	The picture is drawn as a right-flush "floating image," and text flows around it.

ALT=*text*

Specifies text that will be displayed in place of the picture if Show Pictures is turned off.

BORDER=*n*

Specifies the size of a border to be drawn around the image. If the image is a hyperlink, the border is drawn in the appropriate hyperlink color. If the image is not a hyperlink, the border is invisible.

CLASS=*type*

Indicates the class to which the element belongs.

CONTROLS

If a video clip is present, displays a set of controls under the clip.

DYNSRC=*url*

Specifies the address of a video clip or VRML world to be displayed in the window. Stands for Dynamic Source.

HEIGHT=*n*

Along with **WIDTH=**, specifies the size at which the picture is drawn. If the picture's actual dimensions differ from those specified, the picture is stretched to match what is specified. Internet Explorer also uses this to draw a placeholder of appropriate size for the picture before it is loaded.

HSPACE=*n*

Along with **VSPACE=**, specifies margins for the image. Similar to **BORDER=**, except the margins are not painted with color when the image is a hyperlink.

ID=*value*

Specifies a unique value for the element over the document.

ISMAP=*image*

Identifies the picture as a server-side image map. Clicking the picture transmits the coordinates of the click back to the server, triggering a jump to another page.

LOOP=*n*

Specifies how many times a video clip will loop when activated. If *n*=-1, or if **LOOP=**INFINITE is specified, it will loop indefinitely.

SRC=*url*

Specifies the address of the picture to insert.

START=*start-event*

Specifies when the file specified by the **DYNSRC=** attribute should start playing. The *start-event* can be one of these values:

FILEOPEN	Start playing as soon as the file is done opening. This is the default.
MOUSEOVER	Start playing when the user moves the mouse pointer over the animation.

Both values can be set but must be separated with a comma.

STYLE=*css1 properties*

Specifies style information.

USEMAP=*url*

Identifies the picture as a client-side image map and specifies a MAP to use for acting on the user's clicks.

VSPACE=*n*

Along with **HSPACE=**, specifies margins for the image. Similar to **BORDER=**, except the margins are not painted with color when the image is a hyperlink.

WIDTH=*n*

Along with **HEIGHT=**, specifies the size at which the picture is drawn. If the picture's actual dimensions differ from those specified, the picture is stretched to match what is specified. Internet Explorer also uses this to draw a placeholder of appropriate size for the picture before it is loaded.

Example

```
<IMG SRC=mygraphic.bmp>
```

INPUT

<INPUT
ALIGN=TOP|MIDDLE|BOTTOM|LEFT|RIGHT
CHECKED
CLASS=*type*
ID=*value*
MAXLENGTH=*length*
NAME=*name*
NOTAB
onBlur=*event*
onChange=*event*
onClick=*event*
onFocus=*event*
onSelect=*event*
SIZE=*size*
SRC=*url*
STYLE=*css1* **properties**
TABINDEX=*n*
TITLE=*text*
TYPE=TEXT|TEXTAREA|PASSWORD|CHECKBOX|RADIO|SUBMIT|RESET|FILE|HIDDEN|IMAGE|BUTTON
VALUE=*value*>

Specifies a form control.

ALIGN=TOP|MIDDLE|BOTTOM|LEFT|RIGHT

Aligns text to the form element. The default is TOP.

TOP	Surrounding text is aligned to the top of the form element.
MIDDLE	Surrounding text is aligned to the middle of the form element.
BOTTOM	Surrounding text is aligned to the bottom of the form element.
LEFT	Surrounding text is aligned to the left of the form element.
RIGHT	Surrounding text is aligned to the right of the form element.

CHECKED

Sets a check box or radio button to "selected" when the form first loads. Applies only to radio buttons and check boxes.

CLASS=*type*

Indicates the class to which the element belongs.

ID=*value*

Specifies a unique value for the element over the document.

MAXLENGTH=*length*

Indicates the maximum number of characters that can be entered into a text control.

NAME=*name*

Specifies the name of the control.

NOTAB

Specifies that the element does not appear in the tabbing order.

onBlur=*event*

Specifies an event to occur when the element loses focus.

onChange=*event*

Specifies an event to occur when the value of the element is changed.

onClick=*event*

Specifies an event to occur when the mouse is clicked on the element.

onFocus=*event*

Specifies an event to occur when the cursor focus is on the element.

onSelect=*event*

Specifies an event to occur when the element is selected.

SIZE=*size*

Specifies the size of the control (in characters). For TEXTAREA-type controls, both width and height can be specified using this format: "*width*,*height*".

SRC=*url*

Specifies the address of the image to be used. Used when **TYPE=**IMAGE. I>**STYLE=***css1 properties*

Specifies style information.

TABINDEX=*n*

Specifies the order of the element in the tabbing sequence.

TITLE=*text*

Specifies an advisory title.

TYPE=*type*

Specifies what type of control to use. The default is TEXT.

TEXT — Used for a single-line text-entry field. Use in conjunction with the **SIZE=** and **MAXLENGTH=** attributes.

TEXTAREA — Used for a multiline text-entry field.

PASSWORD — The same as the TEXT= attribute, except that text is not displayed as the user enters it.

CHECKBOX — The element appears as a check box.

RADIO — Used for attributes that accept a single value from a set of alternatives. Each radio button field in the group should be given the same name. Only the selected radio button in the group generates a name/value pair in the submitted data. Radio buttons require an explicit **VALUE=** attribute.

SUBMIT — A button that, when clicked, submits the form. You can use the **VALUE=** attribute to provide a non-editable label to be displayed on the button. The default label is application-specific. If a SUBMIT button is clicked to submit the form, and that button has a **NAME=** attribute specified, that button contributes a name/value pair to the submitted data. Otherwise, a SUBMIT button makes no contribution to the submitted data.

RESET — A button that, when clicked, resets the form's fields to their specified initial values. The label to be displayed on the button can be specified just as for the SUBMIT button.

FILE — Used to insert a file.

HIDDEN — No field is presented to the user, but the content of the field is sent with the submitted form. This value can be used to transmit state information about client/server interaction.

IMAGE — An image field that you can click, causing the form to be immediately submitted. The coordinates of the selected point are measured in pixel units from the upper-left corner of the image, and are returned (along with the other contents of the form) in two name/value pairs. The x-coordinate is submitted under the name of the field with ".x" appended, and the y-coordinate is submitted under the name of the field with ".y" appended. Any **VALUE=** attribute is ignored. The image itself is specified by the **SRC=** attribute, exactly as for the image element.

BUTTON — The element appears as a button.

VALUE=*value*

Specifies the default value of textual/numerical controls or specifies the value to be returned when the Boolean controls are turned on.

Example

```
<FORM ACTION="http://intranet/survey" METHOD=POST>
<P>Name
<BR><INPUT NAME="CONTROL1" TYPE=TEXT VALUE="Your Name">
<P>Password
<BR><INPUT TYPE="PASSWORD" NAME="CONTROL2">
<P>Color
<BR><INPUT TYPE="RADIO" NAME="CONTROL3" VALUE="0" CHECKED>Red
<INPUT TYPE="RADIO" NAME="CONTROL3" VALUE="1">Green
<INPUT TYPE="RADIO" NAME="CONTROL3" VALUE="2">Blue
<P>Comments
<BR><INPUT TYPE="TEXTAREA" NAME="CONTROL4" SIZE="20,5" MAXLENGTH="250">
<P><INPUT NAME="CONTROL5" TYPE=CHECKBOX CHECKED>Send receipt
<P><INPUT TYPE="SUBMIT" VALUE="OK"><INPUT TYPE="RESET" VALUE="Reset">
</FORM>
```

ISINDEX

<ISINDEX
ACTION=*url*
PROMPT=*prompt-text***>**

Indicates the presence of a searchable index.

ACTION=*url*
Specifies the gateway program to which the string in the text box should be passed.

PROMPT=*prompt-text*
Specifies a prompt to be used instead of the default prompt.

If the **PROMPT=** attribute is not used, the element displays the following message:

"You can search this index. Type the keyword(s) you want to search for:"

followed by a text box.

When the user enters text and presses ENTER, that text is posted back to the page's URL as a query.

Example

```
<ISINDEX "http://intranet/search" PROMPT="Type keywords here.">
```

KBD

<KBD>

Indicates text to be entered at the keyboard. Renders text in fixed-width and bold type. The end-tag is required.

Example

```
<KBD>The user should enter this text.</KBD>
```

LI

<LI
  CLASS=*type*
  ID=*value*
  STYLE=*css1* **properties**
  TYPE=*order-type*
  VALUE=*n***>**

Denotes one item of a list. Denotes a new list item in a DIR, MENU, OL, or UL block.

CLASS=*type*
Indicates the class to which the element belongs.

ID=*value*
Specifies a unique value for the element over the document.

STYLE=*css1 properties*
Specifies style information.

TYPE=*order-type*
Changes the style of an ordered list. The *order-type* can be one of these values:

A	Use large letters.
a	Use small letters.
I	Use large Roman numerals.
i	Use small Roman numerals.
1	Use numbers.

VALUE=*n*
Changes the count of ordered lists as they progress.

Example

```
<DIR> <LI>Art
<LI>History
<LI>Literature
<LI>Sports
<LI>Entertainment
<LI>Science</DIR>
```

LINK

<LINK
    HREF=*url*
    REL=*forward* **link type**
    REV=*reverse* **link type**
    TITLE=*title*
    TYPE=*type*>

Establishes a hierarchical organization for navigating between documents. The LINK element must reside within the HEAD element. The HEAD element may contain several LINK elements.

HREF=*url*

Specifies the URL that has a relationship to the current document.

REL=*forward link type*

Specifies the forward link type, that is, the type of document to which the link is being made. Link type "stylesheet" signifies that the associated LINK element specifies a link to a style sheet that may be applied to the HTML document containing the LINK element.

REV=*reverse link type*

Indicates that the document to which there is a link has a reverse link from the target back to the source document.

TITLE=*title*

Indicates an advisory title string. The character string supplied with this attribute is recommended for use in building a menu of alternative styles.

TYPE=*type*

Specifies the Internet media type and associated parameters for the linked style sheet. This allows the user agent to disregard style sheets in unsupported notations, without the need to first make a remote query across the network.

Example

```
<LINK HREF="http://www.microsoft.com/newdocnewdoc.htm">
```

LISTING

<LISTING>

Renders text in fixed-width type.

Example

```
<LISTING>Here's some plain text.</LISTING>
```

MAP

<**MAP**
    NAME=*name*>

Specifies a collection of hot spots for a client-side image map.

NAME=*name*
    Names a MAP so it can be referenced. The following example shows a client-side image map.

Example

```
<MAP NAME="map1">
     <AREA ... >
     <AREA ... >
</MAP>
```

MARQUEE

<**MARQUEE**
ALIGN=LEFT|CENTER|RIGHT|TOP|BOTTOM
BEHAVIOR=*type*
BGCOLOR=*color*
DIRECTION=*direction*
HEIGHT=*n*
HSPACE=*n*
LOOP=*n*
SCROLLAMOUNT=*n*
SCROLLDELAY=*n*
VSPACE=*n*
WIDTH=*n*>
</**MARQUEE>**

Creates a scrolling text marquee. The scrolling text appears in the container.

ALIGN=LEFT|CENTER|RIGHT|TOP|BOTTOM

Specifies how the surrounding text should align with the marquee. The default is LEFT.

LEFT	Surrounding text aligns with the left of the marquee.
CENTER	Surrounding text aligns with the center of the marquee.
RIGHT	Surrounding text aligns with the right of the marquee.
TOP	Surrounding text aligns with the top of the marquee.
BOTTOM	Surrounding text aligns with the bottom of the marquee.

BEHAVIOR=*type*

Specifies how the text should behave. The *type* can be one of these values:

SCROLL	Start completely off one side, scroll all the way across and completely off, and then start again. This is the default.
SLIDE	Start completely off one side, scroll in, and stop as soon as the text touches the other margin.
ALTERNATE	Bounce back and forth within the marquee.

BGCOLOR=*color*

Specifies a background color for the marquee. The *color* can be either a hexadecimal number (optionally preceded by a #) specifying a red-green-blue color value, or a predefined color name as described in Color.

DIRECTION=*direction*

Specifies in which direction the text should scroll. The *direction* can be LEFT or RIGHT. The default is LEFT, which means scrolling from right to left.

HEIGHT=*n*

Specifies the height of the marquee, either in pixels or as a percentage of the screen height. To specify a percentage, the *n* must end with a percent sign (%).

HSPACE=*n*

Specifies left and right margins for the outside of the marquee, in pixels.

LOOP=*n*

Specifies how many times a marquee will loop when activated. If **LOOP=**-1, or if **LOOP=**INFINITE, the marquee will loop indefinitely.

SCROLLAMOUNT=*n*

Specifies the number of pixels between each successive draw of the marquee text.

SCROLLDELAY=*n*

Specifies the number of milliseconds between each successive draw of the marquee text.

VSPACE=*n*

Specifies top and bottom margins for the outside of the marquee, in pixels.

WIDTH=*n*

Sets the width of the marquee, either in pixels or as a percentage of the screen width. To specify a percentage, the *n* must end with a percent sign (%).

Example

```
<MARQUEE DIRECTION=RIGHT BEHAVIOR=SCROLL SCROLLAMOUNT=10 SCROLLDELAY=200>
This is a scrolling marquee.
</MARQUEE>
```

MENU

<MENU
COMPACT
... >
</MENU>

Specifies that the following list of items contains individual items that begin with an LI element. The end-tag is required.

COMPACT
    Items in the list are displayed with more compact line spacing.

Example

```
<MENU>
<LI>This is the first item in the menu.
<LI>And this is the second item in the menu.
</MENU>
```

META

<META
CHARSET=*charset*
CONTENT=*description*
HTTP-EQUIV=*response*
NAME=*description*
URL=*url***>**

Provides information about an HTML document to browsers, search engines, and other applications. For instance, **META HTTP-EQUIV=REFRESH CONTENT=2** reloads a document every two seconds. Setting the **HTTP-EQUIV=** attribute to REFRESH gives the instruction to reload. The **CONTENT=** attribute specifies the time in seconds that the page refreshes. You can specify any URL in the element. If no URL is specified, the current document is reloaded.

CHARSET=*charset*

Describes the character set to be used with the document. The preferred usage is the **HTTP-EQUIV=** attribute.

CONTENT=*description*

Defines the meta-information content to be associated with the given name or HTTP response header. Can be used with the **URL=** attribute and a date and time specification to reload a document at a specified interval. See the Client Pull section in the Author's Guide or the examples that follow.

HTTP-EQUIV=*response*

Binds the element to an HTTP response header. This information is then used based on the application reading the header (examples follow). This attribute is the preferred usage for describing a character set.

NAME=*description*

Contains a description of the document.

URL=*url*

Indicates the document's URL.

Examples

If the document contains:

```
<META HTTP-EQUIV="Expires"
      CONTENT="Tue, 04 Dec 1996 21:29:02 GMT">
<meta http-equiv="Keywords" CONTENT="HTML, Reference">
<META HTTP-EQUIV="Reply-to"
      content="anybody@microsoft.com">
<Meta Http-equiv="Keywords" CONTENT="HTML Reference Guide">
```

then the server would include the following header fields:

Expires: Tue, 04 Dec 1996 21:29:02 GMT

Keywords: HTML, Reference

Reply-to: anybody@microsoft.com

as part of the HTTP response to a GET or HEAD request for that document.

The following example shows the correct usage for describing a character set. For more information, see the Character Sets section in the Author's Guide.

```
<META HTTP-EQUIV="Content-Type"
  CONTENT="text/html; charset=Windows-1251">

<HTML>
<HEAD>
<META HTTP-EQUIV="REFRESH" CONTENT=2>
<TITLE>Reload Document</TITLE>
</HEAD>
```

```
<BODY>
<P>This document will be reloaded every two seconds.
</BODY>
</HTML>
```

```
<HTML>
<HEAD>
<META HTTP-EQUIV="REFRESH" CONTENT="5; URL=http://www.sample.com/next.htm">
<TITLE>Load Next Document</TITLE>
</HEAD>
<BODY>
<P>After five seconds have elapsed, the document
"http://www.sample.com/next.htm" will be loaded.
</BODY>
</HTML>
```

NOBR

<NOBR>

Turns off line breaking. Renders text without line breaks.

Example

```
<NOBR>Here's a line of text I don't want to be broken . . . here's the end
of the line.</NOBR>
```

NOFRAMES

<NOFRAMES>

Indicates content viewable only by browsers that do not support frames. Browsers that support frames will not display content between the beginning and ending NOFRAMES tags. You can create a page that is compatible with both browser types by using NOFRAMES.

Example

```
<FRAMESET>
 <NOFRAMES>You need Internet Explorer 3.0 to view frames!</NOFRAMES>
</FRAMESET>
```

OBJECT

<OBJECT
ALIGN=LEFT|TEXTTOP|MIDDLE|TEXTMIDDLE|BASELINE|TEXTBOTTOM|CENTER|RIGHT
BORDER=*n*
CLASSID=*url*
CODEBASE=*url*
CODETYPE=*codetype*
DATA=*url*
DECLARE
HEIGHT=*n*
HSPACE=*n*
NAME=*url*
NOTAB
SHAPES
STANDBY=*message*
TABINDEX=*n*
TITLE=*text*
TYPE=*type*
USEMAP=*url*
VSPACE=*n*
WIDTH=*n***>**
</OBJECT>

Inserts an object, such as an image, document, applet, or control, into the HTML document. An object can contain any elements ordinarily used within the body of an HTML document, including section headings, paragraphs, lists, forms, and nested objects.

The end-tag is required.

ALIGN=LEFT|TEXTTOP|MIDDLE|TEXTMIDDLE|BASELINE|TEXTBOTTOM|CENTER|RIGHT

Sets the alignment for text surrounding the object. The default is LEFT.

LEFT	The object is drawn as a left-flush "floating object," and text flows around it.
TEXTTOP	Surrounding text is aligned with the top of the object.
MIDDLE	The object is drawn as a middle-centered "floating object," and text flows around it.
TEXTMIDDLE	Surrounding text is aligned with the middle of the object.
BASELINE	The object is drawn with its bottom aligned with the baseline of the continuous text.
TEXTBOTTOM	Surrounding text is aligned with the bottom of the object.
CENTER	Surrounding text is aligned with the center of the object.
RIGHT	The object is drawn as a right-flush "floating object," and text flows around it.

BORDER=*n*

Specifies the width of the border if the object is defined to be a hyperlink.

CLASSID=*url*

Identifies the object implementation. The syntax of the *url* depends on the object type. For example, the syntax is **CLSID:***class-identifier* for registered ActiveX controls.

CODEBASE=*url*

Identifies the code base for the object. The syntax of the *url* depends on the object.

CODETYPE=*codetype*

Specifies the Internet media type for code.

DATA=*url*

Identifies data for the object. The syntax of the *url* depends on the object.

DECLARE

Declares the object without instantiating it. Use this when creating cross-references to the object later in the document or when using the object as a parameter in another object.

HEIGHT=*n*

Specifies the suggested height for the object.

HSPACE=*n*

Specifies the horizontal gutter. This is the extra, empty space between the object and any text or images to the left or right of the object.

NAME=*url*

Sets the name of the object when submitted as part of a form.

NOTAB

Excludes the object from the tabbing order.

SHAPES

Specifies that the object has shaped hyperlinks.

STANDBY=*message*

Sets the message to show while loading the object.

TABINDEX=*n*

Sets the position of the object in the tabbing order.

TITLE=*text*

Sets an advisory title.

TYPE=*type*

Specifies the Internet media type for data.

USEMAP=*url*

Specifies the image map to use with the object.

VSPACE=*n*

Specifies the vertical gutter. This is the extra, empty space between the object and any text or images above or below the object.

WIDTH=*n*

Specifies the suggested width for the object.

OL

<OL
  CLASS=*type*
  COMPACT
  ID=*value*
  START=*n*
  STYLE=*css1* **properties**
  TYPE=*order-type*>

Specifies that the following lines of text contain individual items that begin with an LI tag. These items are numbered.

CLASS=*type*
Indicates the class to which the element belongs.

COMPACT
Reduces the spacing between items in the list.

ID=*value*
Specifies a unique value for the element over the document.

START=*n*
Specifies a starting number for the list.

STYLE=*css1 properties*
Specifies style information.

TYPE=*order-type*
Changes the style of the list. The *order-type* can be one of these values:

A	Use large letters.
a	Use small letters.
I	Use large Roman numerals.
i	Use small Roman numerals.
1	Use numbers.

Example

```
<OL>
<LI>This is the first item in the list.
<LI>And this is the second item in the list.
</OL>

<OL START=3>
<LI>This is item number 3.
</OL>

<OL TYPE=A>
<LI>This is item A.
</OL>
```

OPTION

<OPTION
SELECTED
VALUE=*value***>**

Denotes one choice in a list box. In a SELECT block, denotes one of the choices that will appear in the list.

SELECTED

Indicates that this item is the default. If not present, the first item becomes the default.

VALUE=*value*

Indicates the value that will be returned if this item is chosen.

Example

```
<SELECT NAME="Cars" SIZE="1">
    <OPTION VALUE="1">BMW
    <OPTION VALUE="2">PORSCHE
    <OPTION VALUE="3" SELECTED>MERCEDES
</SELECT>
```

P

<P
ALIGN=LEFT|CENTER|RIGHT
CLASS=*type*
ID=*value*
STYLE=*css1* **properties>**

Inserts a paragraph break and denotes a paragraph. The end-tag is optional.

ALIGN=LEFT|CENTER|RIGHT
Sets the alignment of the paragraph. The default is LEFT.

LEFT	Text is aligned to the left margin of the page.
CENTER	Text is centered on the page.
RIGHT	Text is aligned to the right margin of the page.

CLASS=*type*
Indicates the class to which the element belongs.

ID=*value*
Specifies a unique value for the element over the document.

STYLE=*css1 properties*
Specifies style information.

Example

```
<P>This is a paragraph.</P>
```

PARAM

<**PARAM**
NAME=*name*
TYPE=*type*
VALUE=*value*
VALUETYPE=*type*>

Sets property values for a given object. The end-tag is optional.

NAME=*name*

Specifies the property name.

TYPE=*type*

Specifies the Internet media type.

VALUE=*value*

Specifies the property value. The value is passed to the object without change except that any character or numeric character entities are replaced with their corresponding character values.

VALUETYPE=*type*

Specifies how to interpret the value. The *type* can be one of these values:

DATA	The value is data. This is the default value type.
REF	The value is a URL.
OBJECT	The value is a URL of an object in the same document.

This element is valid only within an OBJECT element.

PLAINTEXT

<PLAINTEXT>
</PLAINTEXT>

Renders text in fixed-width type without processing tags and disables HTML parsing until the browser encounters the </PLAINTEXT> tag. The end-tag restores the text to normal formatting.

Example

```
<PLAINTEXT>Here's a sample of HTML: <A HREF="sample.url">This is a
shortcut to a sample.</A></PLAINTEXT>
```

PRE

<PRE>
</PRE>

Renders text in fixed-width type. The end-tag restores the text to normal formatting.

This element is deprecated and is included for backward compatibility with earlier browsers.

Example

```
<PRE>Here's some plain text.</PRE>
```

S

<S>

</S>

Renders text in strikethrough type. The end-tag restores the formatting to normal.

Example

```
<S>This text has a line through it.</S>
```

SAMP

<SAMP>
</SAMP>

Specifies sample text and renders it in a small font. (If no FACE= attribute is specified in the FONT element, fixed-width font is used.) The end-tag restores the text formatting to normal.

Example

```
<SAMP>Here is some text in a small fixed-width font.</SAMP>
```

SCRIPT

<SCRIPT
    LANGUAGE=*scripting* **language**
    SRC=*filename*
    TYPE=*MIME* **type>**
    &lt/SCRIPT>

Specifies the inclusion of a script such as VBScript or JavaScript. External scripts, written as text files, can be referenced using SCRIPT with the **SRC=** attribute. The scripts themselves appear in the element's container. The end-tag is required.

LANGUAGE=*scripting language*

Indicates the ActiveX Scripting language in which the enclosed script was written. Examples of an ActiveX Scripting language are VBScript and JScript. Note: JScript is a brand name. For the **LANGUAGE=** attribute, it must always be written as JavaScript to be functional.

SRC=*filename*

Specifies an external file that contains scripts. These scripts are available just as if they had been written directly into the document, and they follow the same rules as any script that might appear within the document.

TYPE=*MIME type*

Specifies the Internet media type for style notation.

Authors often enclose the scripts in the SCRIPT element within the comment tag (<!-- ... -->). This prevents browsers which do not recognize the SCRIPT element from displaying the scripts as plain text in the browser.

Example

```
<SCRIPT>
<SCRIPT language="VBScript">
    '... Additional VBScript statements ...
</SCRIPT>
```

SELECT

<SELECT
NAME=*name*
SIZE=*n***>**
            &lt/SELECT>

Denotes a list box or drop-down list. The end-tag encloses any OPTION elements that may appear within the SELECT element.

NAME=*name*
    Specifies a name for the list box or drop-down list.

SIZE=*n*
    Specifies the height of the list control.

Example

```
<SELECT NAME="Cars" SIZE="1">
    <OPTION VALUE="1">BMW
    <OPTION VALUE="2">PORSCHE
    <OPTION VALUE="3" SELECTED>MERCEDES
</SELECT>
```

SMALL

<SMALL>

Reduces text by one size. Size is relative to the normal text size and is typically defined by the browser.

Example

```
<SMALL>This text is smaller.</SMALL>
```

SPAN

<SPAN
    STYLE=*css1* **properties>**
</SPAN>

Applies style information to text and other HTML elements within a document. SPAN can be used for localized formatting of text by using **STYLE=** as an attribute. For the application of CSS1 properties to text elements, STYLE is the preferred element.

The end-tag is required.

STYLE=*css1 properties*
    Specifies style information.

Example

```
<SPAN STYLE="margin-left: 1.0in"> This paragraph is 1.0 inches from the
left margin.</SPAN>
```

STRIKE

<STRIKE>
</STRIKE>

Renders text in strikethrough type. The end-tag returns formatting to normal.

Example

```
<STRIKE>This text has a line through it.</STRIKE>
```

STRONG

<STRONG>
    </STRONG>

Emphasizes the text. Usually displays the text in bold. The end-tag returns formatting to normal.

Example

```
<STRONG>This text will be bold.</STRONG>
```

STYLE

<STYLE
TITLE=*title*
TYPE=*MIME* **type>**
&lt/STYLE>

Allows authors to include rendering information by using a specified style notation. Information in the STYLE element overrides client defaults and linked style sheets. The end-tag is required.

Microsoft Internet Explorer supports most of the style sheet properties defined in the CSS1 specifications. For a full description of these properties, see *A User's Guide to Style Sheets*.

TITLE=*title*

Specifies an advisory title.

TYPE=*MIME type*

Specifies the file type for the style information.

Example

```
<HTML>
<HEAD>
<STYLE>
BODY {background: white; color: black}
H1 {font: 8pt Arial bold}
P {font: 10pt Arial; text-indent: 0.5in}
A {text-decoration: none; color: blue}
</STYLE>
<BODY>
<H1>The background color of this document is white.   The color of text
is black.  This text, formatted with the heading one element, is in
8-point Arial bold.&lt/H1>
<P>This text is formatted with 10-point Arial and an indented margin
of 0.5 inches.
<P>The hyperlink created by the text &A HREF=>click here&/A>appears
in blue.
<BODY>
<HTML>
```

SUB

<SUB>
</SUB>

Renders text in subscript. The end-tag restores normal formatting.

Example

```
<SUB>This text is rendered as subscript.</SUB>
```

SUP

<SUP>
</SUP>

Renders text in superscript. The end-tag restores normal formatting.

Example

```
<SUP>This text is rendered as superscript.</SUP>
```

TABLE

<TABLE
ALIGN=LEFT|CENTER|RIGHT|BLEEDLEFT|BLEEDRIGHT|JUSTIFY
BACKGROUND=*url*
BGCOLOR=*color*
BORDER=*n*
BORDERCOLOR=*color*
BORDERCOLORDARK=*color*
BORDERCOLORLIGHT=*color*
CELLPADDING=*n*
CELLSPACING=*n*
CLASS=*type*
CLEAR=LEFT|RIGHT|ALL|NO
COLS=*n*
FRAME=*frame-type*
ID=*value*
NOWRAP
RULES=*rule-type*
STYLE=*css1* **properties**
VALIGN=TOP|MIDDLE|BOTTOM|BASELINE
WIDTH=*n***>**
</TABLE>

Defines a table. Use the TR, TD, and TH elements in the container to create the rows, columns, and cells. The end-tag is required.

The optional THEAD, TBODY, TFOOT, COLGROUP, and COL elements can be used to organize a table and apply attributes across columns and groups of columns.

ALIGN=LEFT|CENTER|RIGHT|BLEEDLEFT|BLEEDRIGHT|JUSTIFY

Specifies the table alignment. The default is LEFT.

LEFT	The table is left-aligned.
CENTER	The table is centered on the page.
RIGHT	The table is right-aligned. If the table is less than the width of the window, text following the table wraps along the left side of the table.
BLEEDLEFT	The table bleeds over the margin into the left side of the document.
BLEEDRIGHT	The table bleeds over the margin into the right side of the document.
JUSTIFY	The table fits the left and right margin definitions.

BACKGROUND=*url*

Specifies a background picture. The picture is tiled behind the text and graphics in the table, table head, or table cell.

BGCOLOR=*color*

Sets background color. The *color* is either a hexadecimal, red-green-blue color value or a predefined color name. See Color.

BORDER=*n*

Sets the size, in pixels, of the table border. The default is zero.

BORDERCOLOR=*color*

Sets border color. Must be used with the **BORDER=** attribute. The *color* is either a hexadecimal, red-green-blue color value or a predefined color name. See Color.

BORDERCOLORDARK=*color*

Sets independent border color control over one of the two colors used to draw a 3-D border, opposite of **BORDERCOLORLIGHT=**. Must be used with the **BORDER=** attribute. The *color* is either a hexadecimal, red-green-blue color value or a predefined color name. See Color.

BORDERCOLORLIGHT=*color*

Sets independent border color control over one of the two colors used to draw a 3-D border, opposite of **BORDERCOLORDARK=**. Must be used with the **BORDER=** attribute. The *color* is either a hexadecimal, red-green-blue color value or a predefined color name. See Color.

CELLPADDING=*n*

Sets the amount of space, in pixels, between the sides of a cell and its contents.

CELLSPACING=*n*

Sets the amount of space, in pixels, between the frame (exterior) of the table and the cells in the table.

CLEAR=NO|LEFT|RIGHT|ALL

Formats text following the table. The default is NO.

NO	Text appears immediately after the table.
LEFT	Text appears as the first left-aligned line after the table.
RIGHT	Text appears as the first right-aligned line after the table.
ALL	

CLASS=*type*

Indicates the class to which the element belongs.

COLS=*n*

Sets the number of columns in the table. If given, this attribute may speed up processing of tables, especially lengthy ones.

FRAME=*frame-type*

Specifies which sides of a frame (outer borders) are displayed. The *frame-type* can be one of these values:

BORDER	Displays a border on all sides of the table frame. This is the default.
VOID	Removes all outside table borders.
ABOVE	Displays a border on the top side of the table frame.
BELOW	Displays a border on the bottom side of the table frame.
HSIDES	Displays a border on the top and bottom sides of the table frame.
LHS	Displays a border on the left-hand side of the table frame.
RHS	Displays a border on the right-hand side of the table frame.
VSIDES	Displays a border on the left and right sides of the table frame.
BOX	Displays a border on all sides of the table frame.

ID=*value*

Specifies a unique value for the element over the document.

NOWRAP Keeps table rows from wrapping if they extend beyond the right margin.

RULES=*rule-type*

Specifies which dividing lines (inner borders) are displayed. The *rule-type* can be one of these values:

NONE	Removes all interior table borders. This is the default.
GROUPS	Displays horizontal borders between all table groups. Groups are specified by the THEAD, TBODY, TFOOT, and COLGROUP elements.
ROWS	Displays horizontal borders between all table rows.
COLS	Displays vertical borders between all table columns.
ALL	Displays a border on all rows and columns.

STYLE=*css1 properties*

Specifies style information.

VALIGN=TOP|MIDDLE|BOTTOM|BASELINE

Specifies the vertical table alignment. The default is TOP.

TOP	The table is aligned at the top.
MIDDLE	The table is vertically aligned at the middle.
BOTTOM	The table is vertically aligned with the bottom.
BASELINE	The table is aligned with the text baseline.

WIDTH=*n*

Sets the width of the table in pixels or as a percentage of the window. To set a percentage, the *n* must end with a percent sign (%).

Example

```
<TABLE BORDER=1 WIDTH=80%>
<THEAD>
<TR>
    <TH>Heading 1</TH>
    <TH>Heading 2</TH>
</TR>
<TBODY>
<TR>
    <TD>Row 1, Column 1 text.</TD>
    <TD>Row 1, Column 2 text.</TD>
</TR>
<TR>
    <TD>Row 2, Column 1 text.</TD>
    <TD>Row 2, Column 2 text.</TD>
</TR>
</TABLE>
```

TBODY

<TBODY
    CLASS=*type*
    ID=*value*
    STYLE=*css1* **properties>**
</TBODY>

Creates multiple sections when rules are needed between groups of table rows. The end-tag is required.

CLASS=*type*

Indicates the class to which the element belongs.

ID=*value*

Specifies a unique value for the element over the document.

STYLE=*css1 properties*

Specifies style information.

If a table does not have a header or footer (does not have a THEAD or TFOOT element), the TBODY element is optional. The end-tag is optional.

You can use the TBODY element more than once in a table. This is useful for dividing lengthy tables into smaller units and for controlling the placement of horizontal rules.

Example

```
<TABLE>
<THEAD>
<TR>
    ...
</TR>
<TBODY>
<TR>
    ...
</TR>
</TBODY>
</TABLE>
```

TD

<TD
ALIGN=CENTER|LEFT|RIGHT|JUSTIFY
BACKGROUND=*url*
BGCOLOR=*color*
BORDERCOLOR=*color*
BORDERCOLORDARK=*color*
BORDERCOLORLIGHT=*color*
CLASS=*type*
COLSPAN=*n*
HEIGHT=*n*
ID=*value*
NOWRAP
ROWSPAN=*n*
STYLE=*css1* **properties**
VALIGN=MIDDLE|TOP|BOTTOM|BASELINE
WIDTH=*n*>

Creates a cell in a table. The end-tag is optional.

ALIGN=CENTER|LEFT|RIGHT|JUSTIFY

Specifies the horizontal alignment of text in a cell. The default is CENTER.

CENTER	The text is centered in the cell.
LEFT	The text is aligned with the left side of the cell.
RIGHT	The text is aligned with the right side of the cell.
JUSTIFY	The text in the cell is justified.

BACKGROUND=*url*

Specifies a background picture. The picture is tiled behind the text and graphics in the table, table head, or table cell.

BGCOLOR=*color*

Sets background color. The *color* is either a hexadecimal, red-green-blue color value or a predefined color name. See Color.

BORDERCOLOR=*color*

Sets border color. Must be used with the BORDER= attribute of the TABLE element. The *color* is either a hexadecimal, red-green-blue color value or a predefined color name. See Color.

BORDERCOLORDARK=*color*

Sets independent border color control over one of the two colors used to draw a 3-D border, opposite of **BORDERCOLORLIGHT=**. Must be used with the BORDER= attribute of the TABLE element. The *color* is either a hexadecimal, red-green-blue color value or a predefined color name. See Color.

BORDERCOLORLIGHT=*color*

Sets independent border color control over one of the two colors used to draw a 3-D border, opposite of **BORDERCOLORDARK=**. Must be used with the BORDER= attribute of the TABLE element. The *color* is either a hexadecimal, red-green-blue color value or a predefined color name. See Color.

CLASS=*type*

Indicates the class to which the element belongs.

COLSPAN=*n*

Extends the content of a cell into cells of the adjoining column(s). The number of columns that are overlapped is indicated by *n*.

HEIGHT=*n*

Specifies the height of the table cell.

ID=*value*

Specifies a unique value for the element over the document.

NOWRAP

Prevents table columns from wrapping if they extend beyond the page.

ROWSPAN=*n*

Extends the content of a cell into cells in the adjoining row(s). The number of rows that are overlapped is indicated by *n*.

STYLE=*css1 properties*

Specifies style information.

VALIGN=MIDDLE|TOP|BOTTOM|BASELINE

Specifies the vertical alignment of text in the cell. The default is MIDDLE.

MIDDLE	Text is aligned in the middle of each cell.
TOP	Text is aligned with the top of each cell.
BOTTOM	Text is aligned with the bottom of each cell.
BASELINE	Text in adjoining cells in a row is aligned along a common baseline.

WIDTH=*n*

Specifies the width of the table cell.

This element is valid only within a row in a table, that is, you must use a TR element before using TD. All attributes are optional.

TEXTAREA

<TEXTAREA
    COLS=*n*
    NAME=*name*
    ROWS=*n*>
</TEXTAREA>

Creates a multiline text entry control in which the user can type and edit text. The end-tag is required.

COLS=*n*
Sets the width, in characters, of the text area.

NAME=*name*
Sets the name of the text area. This name is used when the element is used within a FORM element.

ROWS=*n*
Sets the height, in characters, of the text area.

Any text between the start-tag and end-tag is used as the initial value for the control.

TFOOT

<TFOOT
CLASS=*type*
ID=*value*
STYLE=*css1* **properties>**
</TFOOT>

Defines the table footer. Use TFOOT to duplicate footers when breaking a table across page boundaries, or for static headers when body sections are rendered in a scrolling panel.

CLASS=*type*

Indicates the class to which the element belongs.

ID=*value*

Specifies a unique value for the element over the document.

STYLE=*css1 properties*

Specifies style information.

The table footer is optional; if given, only one footer is allowed. The TFOOT element is valid only within a table; you must use a TABLE element before using this element. The end-tag is optional.

Example

```
<TABLE>
<TBODY>
    <TR>
    ...
    </TR>
<TFOOT>
    <TR>
    ...
    </TR>
</TABLE>
```

TH

<TH
ALIGN=CENTER|LEFT|RIGHT|JUSTIFY
BACKGROUND=*url*
BGCOLOR=*color*
BORDERCOLOR=*color*
BORDERCOLORDARK=*color*
BORDERCOLORLIGHT=*color*
CLASS=*type*
COLSPAN=*n*
ID=*value*
NOWRAP
ROWSPAN=*n*
STYLE=*css1* **properties**
VALIGN=TOP|MIDDLE|BOTTOM|BASELINE
WIDTH=*n***>**

Creates a row or column heading in a table. The element is similar to the TD element but emphasizes the text in the cell to distinguish it from text in TD cells. The end-tag is optional.

ALIGN=CENTER|LEFT|RIGHT|JUSTIFY

Specifies the alignment of text in the cell. The default is CENTER.

CENTER	The text in the header is aligned in the center of the cell.
LEFT	The text in the header is aligned with the left margin.
RIGHT	The text in the header is aligned with the right margin.
JUSTIFY	The text in the header is justified.

BACKGROUND=*url*

Specifies a background picture. The picture is tiled behind the text and graphics in the table, table head, or table cell.

BGCOLOR=*color*

Sets background color. The *color* is either a hexadecimal, red-green-blue color value or a predefined color name. See Color.

BORDERCOLOR=*color*

Sets border color. Must be used with the BORDER= attribute of the TABLE element. The *color* is either a hexadecimal, red-green-blue color value or a predefined color name. See Color.

BORDERCOLORDARK=*color*

Sets independent border color control over one of the two colors used to draw a 3-D border, opposite of **BORDERCOLORLIGHT=**. Must be used with the BORDER= attribute of the TABLE element. The *color* is either a hexadecimal, red-green-blue color value or a predefined color name. See Color.

BORDERCOLORLIGHT=*color*

Sets independent border color control over one of the two colors used to draw a 3-D border, opposite of **BORDERCOLORDARK=**. Must be used with the BORDER= attribute of the TABLE element. The *color* is either a hexadecimal, red-green-blue color value or a predefined color name. See Color.

CLASS=*type*

Indicates the class to which the element belongs.

COLSPAN=*n*

Indicates the number of table columns this cell spans.

ID=*value*

Specifies a unique value for the element over the document.

NOWRAP

Prevents word wrapping within the cell. Lines of text appear as given in the HTML document.

ROWSPAN=*n*

Indicates the number of table rows this cell spans.

STYLE=*css1 properties*

Specifies style information.

VALIGN=TOP|MIDDLE|BOTTOM|BASELINE

Specifies the vertical alignment of text in the table. The default is TOP.

TOP	Text is aligned with the top of each cell.
MIDDLE	Text is aligned in the middle of each cell.
BOTTOM	Text is aligned with the bottom of each cell.
BASELINE	Text in adjoining cells in a row is aligned along a common baseline.

WIDTH=*n*

Specifies the width of the table heading, in columns.

This element is valid only within a row in a table, that is, you must use a TR element before using TH. All attributes are optional.

THEAD

<THEAD
ALIGN=LEFT|CENTER|RIGHT|JUSTIFY
CLASS=*type*
ID=*value*
STYLE=*css1* **properties**
VALIGN= MIDDLE|TOP|BOTTOM>

Defines the table heading. Use THEAD to duplicate headings when breaking tables across page boundaries, or for static headings when body sections are rendered in a scrolling panel.

ALIGN=CENTER|LEFT|RIGHT|JUSTIFY

Specifies the alignment of text in the heading. The default is CENTER.

CENTER The text in the heading is aligned in the center of the page.
LEFT The text in the heading is aligned with the left margin.
RIGHT The text in the heading is aligned with the right margin.
JUSTIFY The text in the heading is justified.

CLASS=*type*

Indicates the class to which the element belongs.

ID=*value*

Specifies a unique value for the element over the document.

STYLE=*css1 properties*

Specifies style information.

VALIGN=MIDDLE|TOP|BOTTOM

Specifies the vertical alignment of text in the heading. The default is MIDDLE.

MIDDLE Text is aligned in the middle of the heading.
TOP Text is aligned at the top of the heading.
BOTTOM Text is aligned at the bottom of the heading.

The table heading is optional; if given, only one heading is allowed. The THEAD element is valid only within a table; you must use a TABLE element before using this element. The end-tag is optional.

Example

```
<TABLE>
<THEAD>
    <TR>
    ...
    </TR>
<TBODY>
    <TR>
    ...
    </TR>
</TABLE>
```

TITLE

<TITLE>
</TITLE>

Specifies a title for the document. Internet Explorer uses this for the window caption.

This element is valid only within the HEAD element. The end-tag is required.

Example

```
<HEAD>
<TITLE>"Welcome To Internet Explorer!"</TITLE>
</HEAD>
```

TR

<TR
  ALIGN=CENTER|LEFT|RIGHT|JUSTIFY
  BGCOLOR=*color*
  BORDERCOLOR=*color*
  BORDERCOLORDARK=*color*
  BORDERCOLORLIGHT=*color*
  CLASS=*type*
  ID=*value*
  NOWRAP
  STYLE=*css1* **properties**
  VALIGN=MIDDLE|TOP|BOTTOM|BASELINE>
  </TR>

Creates a row in a table.

ALIGN=CENTER|LEFT|RIGHT|JUSTIFY

Sets the alignment for text in the row. The default is CENTER.

CENTER	Text in the row is centered.
LEFT	Text in the row is aligned to the left.
RIGHT	Text in the row is aligned to the right.
JUSTIFY	Text in the row is justified.

BGCOLOR=*color*

Sets background color. The *color* is either a hexadecimal, red-green-blue color value or a predefined color name. See Color.

BORDERCOLOR=*color*

Sets border color. Must be used with the BORDER= attribute of the TABLE element. The *color* is either a hexadecimal, red-green-blue color value or a predefined color name. See Color.

BORDERCOLORDARK=*color*

Sets independent border color control over one of the two colors used to draw a 3-D border, opposite of **BORDERCOLORLIGHT=**. Must be used with the BORDER= attribute of the TABLE element. The *color* is either a hexadecimal, red-green-blue color value or a predefined color name. See Color.

BORDERCOLORLIGHT=*color*

Sets independent border color control over one of the two colors used to draw a 3-D border, opposite of **BORDERCOLORDARK=**. Must be used with the BORDER= attribute of the TABLE element. The *color* is either a hexadecimal, red-green-blue color value or a predefined color name. See Color.

CLASS=*type*

Indicates the class to which the element belongs.

ID=*value*

Specifies a unique value for the element over the document.

NOWRAP

Prevents the table rows from wrapping if they over-extend the margins of the browser.

STYLE=*css1 properties*

Specifies style information.

VALIGN=MIDDLE|TOP|BOTTOM|BASELINE

Specifies the vertical alignment of text in the row. The default is MIDDLE.

MIDDLE	Text is aligned in the middle of each cell.
TOP	Text is aligned with the top of each cell.
BOTTOM	Text is aligned with the bottom of each cell.
BASELINE	Text in adjoining cells in a row is aligned along a common baseline.

TT

<TT>

</TT>

Indicates teletype. Renders text in fixed-width type. The end-tag returns the text formatting to normal.

Example

```
<TT>Here's some plain text.</TT>
```

U

<U>
    </U>

Renders underlined text. The end-tag restores the text to normal.

Example

```
<U>This text is underlined.</U>
```

UL

<UL
    CLASS=*type*
    COMPACT
    ID=*value*
    STYLE=*css1* **properties>**
</UL>

Specifies that the following block of text contains individual items that begin with an LI tag. These items are bulleted. The end-tag is required.

CLASS=*type*

Indicates the class to which the element belongs.

COMPACT

Reduces the spacing between items in the list.

ID=*value*

Specifies a unique value for the element over the document.

STYLE=*css1 properties*

Specifies style information.

Example

```
<UL>
<LI>This is the first bulleted item in the list.
<LI>And this is the second bulleted item in the list.
</UL>
```

VAR

<VAR>
</VAR>

Indicates placeholder text for a variable. Displays text in a small, fixed-width type. The end-tag (required) restores the formatting to normal.

Example

```
Enter the <VAR>filename</VAR> in the dialog box.
```

WBR

Inserts a soft line break in a block of NOBR text.

Example

```
<NOBR>This line of text will not break, no matter how narrow the window
gets.
<WBR>This one, however, will.</NOBR>
```

XMP

<XMP>
</XMP>

Indicates example text by displaying it in fixed-width type. The end-tag (required) restores the text to normal.

Example

```
<XMP>Here's some plain text.</XMP>
```