

## HTML Complete Tag List

### <!-- ... --> (Comment)

This construct encloses text comments that will not be displayed by the browser. No attributes or events are associated with this construct.

#### Standard Syntax

```
<!-- comment goes here -->
```

#### Examples

```
<!-- This is an informational comment that can occur  
anywhere in an HTML document. The next few examples  
show how style sheets and scripts are "commented out" to prevent  
older browsers from misinterpreting the content.  
-->
```

### <!DOCTYPE> (Document Type Definition)

This SGML construct specifies the document type definition corresponding to the document. There are no attributes or events associated with this element.

#### Standard Syntax

```
<!DOCTYPE "DTD IDENTIFIER">
```

#### Examples

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 TRANSITIONAL//EN">  
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 TRANSITIONAL//EN">  
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "xhtml11.dtd">
```

#### Notes

- The **doctype** statement should be used as the first line of all documents.
- Validation programs might use this construct when determining the correctness of an HTML document.
- Modern browsers may determine what rendering mode to use depending on the **doctype** statement. This is dubbed the *doctype switch*. An incorrect **doctype** that does not correspond to appropriate markup usage may result in inaccurate display.

### <a> (Anchor)

This element defines a hyperlink, the named target destination for a hyperlink, or both.

#### Standard Syntax

```
<a  
  accesskey="key"  
  charset="character code for language of linked  
  resource"  
  class="class name(s)"  
  coords="comma-separated list of numbers"
```

dir="ltr | rtl"  
href="url"  
hreflang="language code"  
id="unique alphanumeric identifier"  
lang="language code"  
name="name of target location"  
rel="comma-separated list of relationship values"  
rev="comma-separated list of relationship values"  
shape="default | circle | poly | rect"  
style="style information"  
tabindex="number"  
target="\_blank | frame-name | \_parent | \_self | \_top"  
(transitional only)  
title="advisory text"  
type="content type of linked data">

</a>

### Examples

<!-- anchor linking to external file -->

<a href="http://www.democompany.com/">External Link</a>

<!-- anchor linking to file on local file system -->

<a href="file:c:\html\index.html">local file link</a>

<!-- anchor invoking anonymous FTP -->

<a href="ftp://ftp.democompany.com/freestuff">Anonymous FTP link</a>

<!-- anchor invoking FTP with password -->

<a href="ftp://joeuser:secretpassword@democompany.com/path/file">FTP with password</a>

<!-- anchor invoking mail -->

<a href="mailto:fakeid@democompany.com">Send mail</a>

<!-- anchor used to define target destination within document -->

<a name="jump">Jump target</a>

<!-- anchor linking internally to previous target anchor -->

<a href="#jump">Local jump within document</a>

<!-- anchor linking externally to previous target anchor -->

<a href="http://www.democompany.com/document#jump">

### Notes

- The following are reserved browser key bindings for the two major browsers and should not be used as values for **accesskey**: **a**, **c**, **e**, **f**, **g**, **h**, **v**, left arrow, and right arrow.
- HTML 3.2 defines only **name**, **href**, **rel**, **rev**, and **title**.

- The **target** attribute is not defined in browsers that do not support frames, such as Netscape 1 generation browsers. Furthermore, **target** is not allowed under strict variants of XHTML but is limited to frameset or transitional forms.

## <abbr> (Abbreviation)

This element allows authors to clearly indicate a sequence of characters that define an abbreviation for a word (such as Mr. instead of Mister, Calif instead of California).

### Standard Syntax

```
<abbr
  class="class name(s)"
  dir="ltr | rtl"
  id="unique alphanumeric identifier"
  lang="language code"
  style="style information"
  title="advisory text">
```

```
</abbr>
```

### Examples

```
<abbr title="California abbreviated">Calif
</abbr>
```

Isn't `<abbr>WWW</abbr>` an acronym? Are you sure?

### Notes

- This tag is commonly confused with `<acronym>`. Debate about just what constitutes an acronym as compared with an abbreviation is common amongst very detail-oriented Web standards experts. In reality, Web developers appear to use the `<acronym>` tag more often than the `<abbr>` tag.
- With the **title** attribute set on this element, Opera and Mozilla may render a dotted underline useful to suggest the Tooltip that might contain a definition for the word.
- Because there is typically no markup-oriented presentation for this element, it is primarily used in conjunction with style sheets and scripts.

## <acronym> (Acronym)

This element allows authors to clearly indicate a sequence of characters that compose an acronym (XML, WWW, and so on).

### Standard Syntax

```
<acronym
  class="class name(s)"
  dir="ltr | rtl"
  id="unique alphanumeric identifier"
  lang="language code"
  style="style information">
```

```
    title="advisory text">
</acronym>
```

### Examples

```
<acronym title="Extensible Markup Language">XML</acronym>
<acronym lang="fr" title="Société Nationale de Chemins de
Fer">SNCF</acronym>
```

### Notes

- **<acronym>** is a new element that is not defined under HTML 2 or 3.2.
- This tag is often confused with **<abbr>**.
- With the **title** attribute set on this element, Opera and Mozilla may render a dotted underline to suggest the Tooltip that might contain a definition for the word.

## <address> (Address)

This element marks up text indicating authorship or ownership of information. It generally occurs at the beginning or end of a document.

### Standard Syntax

```
<address
  class="class name(s)"
  dir="ltr | rtl"
  id="unique alphanumeric identifier"
  lang="language code"
  style="style information"
  title="advisory text">
</address>
```

### Example

```
<address>Big Company, Inc.<br />
2105 Demo Street<br />
San Diego, CA 92109 U.S.A.</address>
```

## <applet> (Java Applet)

This element identifies the inclusion of a Java applet. The strict HTML 4.01 definition does not include this element; it has been deprecated in favor of **<object>**.

### Standard Syntax (HTML 4.01 Transitional Only)

```
<applet
  align="bottom | left | middle | right | top"
  alt="alternative text"
  archive="URL of archive file"
```

```
class="class name(s)"
code="URL of Java class file"
codebase="URL for base referencing"
height="pixels"
hspace="pixels"
id="unique alphanumeric identifier"
name="unique name for scripting reference"
object="filename"
style="style information"
title="advisory text"
vspace="pixels"
width="pixels">
```

</applet>

### Element-Specific Attributes

**align** This attribute is used to position the applet on the page relative to content that might flow around it. The transitional specifications define values of **bottom**, **left**, **middle**, **right**, and **top**, whereas Microsoft and Netscape also might support **absbottom**, **absmiddle**, **baseline**, **center**, and **texttop**.

**alt** This attribute causes a descriptive text alternative to be displayed in browsers that do not support Java. Page designers should also remember that content enclosed within an **<applet>** tag may also be rendered as alternative text.

**archive** This attribute refers to an archived or compressed version of the applet and its associated class files, which might help reduce download time.

**code** This attribute specifies the URL of the applet's class file to be loaded and executed. Applet filenames are identified by a .class filename extension. The URL specified by **code** might be relative to the **codebase** attribute.

**codebase** This attribute gives the absolute or relative URL of the directory where applets' .class files referenced by the **code** attribute are stored.

**datafld** This attribute, supported by Internet Explorer 4 and higher, specifies the column name from the data source object that supplies the bound data. This attribute might be used to specify the various **<param>** tags passed to the Java applet.

**datasrc** Like **datafld**, this attribute is used for data binding under Internet Explorer 4. It indicates the **id** of the data source object that supplies the data that is bound to the **<param>** tags associated with the applet.

**height** This attribute specifies the height, in pixels, that the applet needs.

**hspace** This attribute specifies additional horizontal space, in pixels, to be reserved on either side of the applet.

**mayscript** In the Netscape implementation, this attribute allows access to an applet by programs in a scripting language embedded in the document.

**name** This attribute assigns a name to the applet so that it can be identified by other resources, particularly scripts.

**object** This attribute specifies the URL of a serialized representation of an applet.

**src** As defined for Internet Explorer 4 and higher, this attribute specifies a URL for an associated file for the applet. The meaning and use is unclear and not part of the HTML standard.

**vspace** This attribute specifies additional vertical space, in pixels, to be reserved above and below the applet.

**width** This attribute specifies the width, in pixels, that the applet needs.

### Example

```
<applet code="game.class" align="left" archive="game.zip"
  height="250" width="350">
  <param name="difficulty" value="easy">
  <b>Sorry, you need Java to play this game.</b>
</applet>
```

## <area> (Image Map Area)

Defines a hot-spot region on an image, and associates it with a hypertext link. This element is used only within a <map> tag.

### Standard Syntax

```
<area
  accesskey="character"
  alt="alternative text"
  class="class name(s)"
  coords="comma separated list of values"
  dir="ltr | rtl"
  href="url"
  id="unique alphanumeric identifier"
  lang="language code"
  nohref="nohref"
  shape="circle | default | poly | rect"
  style="style information"
  tabindex="number"
```

```
target="_blank | frame-name | _parent | _self |
_top" (transitional or frameset only)
title="advisory text" />
```

### Element-Specific Attributes

**accesskey** This attribute specifies a keyboard navigation accelerator for the element. Pressing ALT or a similar key in association with the specified character selects the form control correlated with that key sequence. Page designers are forewarned to avoid key sequences already bound to browsers.

**alt** This attribute contains a text string alternative to display on browsers that cannot display images.

**coords** This attribute contains a set of values specifying the coordinates of the hot-spot region. The number and meaning of the values depend upon the value specified for the **shape** attribute. For a **rect** or **rectangle** shape, the **coords** value is two *x,y* pairs: **left**, **top**, **right**, and **bottom**. For a **circ** or **circle** shape, the **coords** value is *x,y,r* where *x,y* is a pair specifying the center of the circle and *r* is a value for the radius. For a **poly** or **polygon** shape, the **coords** value is a set of *x,y* pairs for each point in the polygon: *x1,y1,x2,y2,x3,y3*, and so on.

**href** This attribute specifies the hyperlink target for the area. Its value is a valid URL. Either this attribute or the **nohref** attribute must be present in the element.

**name** This attribute is used to define a name for the clickable area so that it can be scripted by older browsers.

**nohref** This attribute indicates that no hyperlink exists for the associated area. Either this attribute or the **href** attribute must be present in the element.

**shape** This attribute defines the shape of the associated hot spot. HTML 4 defines the values **rect**, which defines a rectangular region; **circle**, which defines a circular region; **poly**, which defines a polygon; and **default**, which indicates the entire region beyond any defined shapes. Many browsers, notably Internet Explorer 4 and higher, support **circ**, **polygon**, and **rectangle** as valid values for **shape**.

**tabindex** This attribute uses a numeric value to specify the position of the defined area in the browser tabbing order.

**target** This attribute specifies the target window for hyperlink referencing frames. The value is a frame name or one of several special names. A value of **\_blank** indicates a new window. A value of **\_parent** indicates the parent frame set containing the source link. A value of **\_self** indicates the frame containing the source link. A value of **\_top** indicates the full browser window.

### Example

```
<map name="primary">
  <area shape="circle" coords="200,250,25" href="another.html" />
  <area shape="default" nohref="nohref" />
</map>
```

### Notes

- Under the HTML 3.2 and 4.0 specifications, the closing tag **</area>** is forbidden.
- The XHTML 1.0 specification requires a trailing slash: **<area />**.

- The **id**, **class**, and **style** attributes have the same meaning as the core attributes defined in the HTML 4 specification, but only Netscape and Microsoft define them.
- Netscape 1-level browsers do not understand the **target** attribute as it relates to frames.
- HTML 3.2 defines only **alt**, **coords**, **href**, **nohref**, and **shape**.

## <b> (Bold)

This element indicates that the enclosed text should be displayed in boldface.

### Standard Syntax

```
<b
  class="class name(s)"
  dir="ltr | rtl"
  id="unique alphanumeric identifier"
  lang="language code"
  style="style information"
  title="advisory text">
```

```
</b>
```

### Example

This text is **bold** for emphasis.

## <base> (Base URL)

This element specifies the base URL to be used for all relative URLs contained within a document.

### Standard Syntax

```
<base
  href="url"
  id="unique alphanumeric identifier"
  target="_blank | frame-name | _parent | _self | top" (transitional only)
/>
```

### Element-Specific Attributes

**href** This attribute specifies the base URL to be used throughout the document for relative URL addresses.

**target** For documents containing frames, this attribute specifies the default target window for every link that does not have an explicit target reference. Aside from named frames, several special values exist. A value of **\_blank** indicates a new window. A value of **\_parent** indicates the parent frame set containing the source link. A value of **\_self** indicates the frame containing the source link. A value of **\_top** indicates the full browser window.

## Examples

```
<base href="http://www.democompany.com/" />
```

```
<base target="_blank" href="http://www.democompany.com/" />
```

## <basefont> (Base Font)

This element establishes a default font size for a document. Font size then can be varied relative to the base font size using the **font** element.

### Standard Syntax (Transitional Only)

```
<basefont  
  color="color name | #RRGGBB"  
  face="font name(s)"  
  id="unique alphanumeric identifier"  
  size="1-7 | +/-int" />
```

### Element-Specific Attributes

**color** This attribute sets the text color using either a named color or a color specified in the hexadecimal *#RRGGBB* format.

**face** This attribute contains a list of one or more font names. The document text in the default style is rendered in the first font face that the client's browser supports. If no font listed is installed on the local system, the browser typically defaults to the proportional or fixed-width font for that system.

**size** This attribute specifies the font size as either a numeric or relative value. Numeric values range from **1** to **7** with **1** being the smallest and **3** the default.

### Example

```
<basefont color="#ff0000" face="Helvetica" size="+2" />
```

### Notes

- HTML 3.2 supports the **basefont** element but only with the **size** attribute.
- The strict HTML and XHTML specifications do not support this element.
- Despite being part of transitional standards, some standards-focused browsers like Mozilla and Opera do not support this element.
- This element can be imitated with a CSS rule on the body element.
- XHTML 1.0 requires a trailing slash for this element: **<basefont />**.

## <bdo> (Bidirectional Override)

This element is used to override the current directionality of text.

### Standard Syntax

```
<bdo  
  class="class name(s)"  
  dir="ltr | rtl"  
  id="unique alphanumeric identifier"  
  lang="language code"
```

```
style="style information"
title="advisory text">
```

```
</bdo>
```

### Example

```
<!-- Switch text direction -->
```

```
<bdo dir="rtl">This text will go right to left if you can
find a browser that supports this element.
```

```
</bdo>
```

## <bgsound> (Background Sound)

This Internet Explorer element associates a background sound with a page.

### Standard Syntax (Defined by Internet Explorer 4)

```
<bgsound
  balance="number"
  id="unique alphanumeric identifier"
  loop="number"
  src="url of sound file"
  volume="number">
```

### Element-Specific Attributes

**balance** This attribute defines a number between -10,000 and +10,000 that determines how the volume will be divided between the speakers.

**loop** This attribute indicates the number of times a sound is to be played and either has a positive numeric value or **-1** to specify that it will continuously loop. The keyword **infinite** is also supported in many Internet Explorer implementations.

**src** This attribute specifies the URL of the sound file to be played, which must be one of the following types: .wav, .au, or .mid.

**volume** This attribute defines a number between -10,000 and 0 that determines the loudness of a page's background sound.

### Examples

```
<bgsound src="sound1.mid">
<bgsound src="sound2.au" loop="infinite">
```

### Notes

- Similar functionality can be achieved in some versions of Netscape using the **<embed>** tag to invoke an audio player.
- You can write **bgsound** with a self-closing tag **<bgsound />**. However, since this element is not part of a standard, making it XHTML-like will not make it validate.

## <big> (Big Font)

Indicates that the enclosed text should be displayed in a larger font relative to the current font.

### Standard Syntax

```
<big
  class="class name(s)"
  dir="ltr | rtl"
  id="unique alphanumeric identifier"
  lang="language code"
  style="style information"
  title="advisory text">
</big>
```

### Example

This text is regular size. **<big>**This text is larger.**</big>**

## <blink> (Blinking Text)

This Netscape-specific element causes the enclosed text to flash slowly.

### Syntax (Defined by Netscape)

```
<blink
  class="class name(s)"
  id="unique alphanumeric identifier"
  lang="language code"
  style="style information">

</blink>
```

### Example

**<blink>**Annoying, isn't it?**</blink>**

## <blockquote> (Block Quote)

This block element indicates that the enclosed text is an extended quotation. Usually, this is rendered visually by indentation.

### Standard Syntax

```
<blockquote
  cite="url of source information"
  class="class name(s)"
  dir="ltr | rtl"
  id="unique alphanumeric identifier"
  lang="language code"
  style="style information"
  title="advisory text">
```

</blockquote>

### Element-Specific Attributes

**cite** The value of this attribute should be a URL for the document in which the information cited can be found.

### Example

The following paragraph is taken from our March report:

```
<blockquote cite="marchreport.html"> ... text ...
```

```
</blockquote>
```

## <body> (Document Body)

This element encloses a document's displayable content.

### Standard Syntax

```
<body
```

```
  alink="color name | #RRGGBB" (transitional only)
```

```
  background="url of background image" (transitional only)
```

```
  bgcolor="color name | #RRGGBB" (transitional only)
```

```
  class="class name(s)"
```

```
  dir="ltr | rtl"
```

```
  id="unique alphanumeric identifier"
```

```
  lang="language code"
```

```
  link="color name | #RRGGBB" (transitional only)
```

```
  style="style information"
```

```
  text="color name | #RRGGBB" (transitional only)
```

```
  title="advisory text"
```

```
  vlink="color name | #RRGGBB"> (vlink attribute transitional only)
```

```
</body>
```

### Element-Specific Attributes

**alink** This attribute sets the color for active links within the document. Active links represent the state of a link as it is being clicked. The value of the attribute can be either a browser-dependent named color or a color specified in the hexadecimal *#RRGGBB* format.

**background** This attribute contains a URL for an image file, which will be tiled to provide the document background.

**bgcolor** This attribute sets the background color for the document. Its value can be either a browser-dependent named color or a color specified using the hexadecimal *#RRGGBB* format.

**bgproperties** This attribute, first introduced in Internet Explorer 2, has one value, **fixed**, which causes the background image to act as a fixed watermark and not to scroll.

**bottommargin** This attribute specifies the bottom margin for the entire body of the page and overrides the default margin. When set to **0** or " ", the bottom margin is the bottom edge of the window or frame the content is displayed in.

**leftmargin** This Internet Explorer-specific attribute sets the left margin for the page in pixels, overriding the default margin. When set to **0** or " ", the left margin is the left edge of the window or the frame.

**link** This attribute sets the color for hyperlinks within the document that have not yet been visited. Its value can be either a browser-dependent named color or a color specified using the hexadecimal *#RRGGBB* format.

**marginheight** This Netscape-specific attribute sets the top margin for the document in pixels. If set to **0** or " ", the top margin will be exactly on the top edge of the window or frame. It is equivalent to combining the Internet Explorer attributes **bottommargin** and **topmargin**.

**marginwidth** This Netscape-specific attribute sets the left and right margins for the page in pixels, overriding the default margin. When set to **0** or " ", the left margin is the left edge of the window or the frame. It is equivalent to combining the Internet Explorer attributes **leftmargin** and **rightmargin**.

**nowrap** This Internet Explorer-specific attribute is used to control the wrapping of text body width. If set to **yes**, text should not wrap. The default is **no**. CSS rules should be used instead of this attribute.

**rightmargin** This attribute, specific to Internet Explorer, sets the right margin for the page in pixels, overriding the default margin. When set to **0** or "", the right margin is the right edge of the window or the frame.

**scroll** This Internet Explorer attribute turns the scroll bars on or off. The default value is **yes**.

**text** This attribute sets the text color for the document. Its value can be either a browser-dependent named color or a color specified using the hexadecimal *#RRGGBB* format.

**topmargin** This Internet Explorer-specific attribute sets the top margin for the document in pixels. If set to **0** or "", the top margin will be exactly on the top edge of the window or frame.

#### Example

```
<body background="checkered.gif"
  bgcolor="white"
  alink="red"
  link="blue"
  vlink="red"
  text="black"> ... </body>
```

```
<body onload="myLoadFunction()"> ... </body>
```

#### Notes

- When defining text colors, it is important to be careful to specify both foreground and background explicitly so that they are not masked out by browser defaults set by the user.
- Under the strict HTML and XHTML definitions, CSS should be used in place of presentation attributes like **alink**, **background**, **bgcolor**, **link**, **text**, and **vlink**.
- This element must be present in all documents except those declaring a frame set.
- Under XHTML 1.0, the closing **</body>** tag is mandatory.

## <br> (Line Break)

This empty element forces a line break.

#### Standard Syntax

```
<br
  class="class name(s)"
  clear="all | left | none | right" (transitional only)
  id="unique alphanumeric identifier"
  style="style information"
  title="advisory text" />
```

### Element-Specific Attributes

**clear** This attribute forces the insertion of vertical space so that the tagged text can be positioned with respect to images. A value of **left** clears text that flows around left-aligned images to the next full left margin, a value of **right** clears text that flows around right-aligned images to the next full right margin, and a value of **all** clears text until it can reach both full margins. The default value according to the transitional HTML and XHTML specifications is **none**, but its meaning generally is supported as just introducing a return and nothing more.

### Examples

This text will be broken here `<br />` and continued on a new line.

```

```

This is the image caption. `<br clear="right" />`

### Notes

- This is an empty element. A closing tag is illegal under all HTML specifications. For XHTML compatibility, a closing slash is required: `<br />`.
- Under the strict HTML and XHTML specifications, the **clear** attribute is not valid. The style property **clear** provides the same functionality as the **clear** attribute.

## <button> (Form Button)

This element defines a nameable region known as a button, which can be used together with scripts.

### Standard Syntax

```
<button
  accesskey="key"
  class="class name(s)"
  dir="ltr | rtl"
  disabled="disabled"
  id="unique alphanumeric identifier"
  lang="language code"
  name="button name"
  style="style information"
  tabindex="number"
  title="advisory text"
  type="button | reset | submit"
  value="button value">
```

```
</button>
```

### Element-Specific Attributes

**accesskey** This attribute specifies a keyboard navigation accelerator for the element. Pressing ALT or a similar key in association with the specified key selects the anchor element correlated with that key.

**datafld** This Internet Explorer-specific attribute specifies the column name from the data source object that supplies the bound data that defines the information for the **<button>** tag's content.

**dataformatas** This Internet Explorer-specific attribute indicates whether the bound data is plain text or HTML.

**datasrc** This Internet Explorer-specific attribute indicates the **id** of the data source object that supplies the data that is bound to the **<button>** tag.

**disabled** This attribute is used to disable the button.

**name** This attribute is used to define a name for the button so that it can be scripted by older browsers or used to provide a name for submit buttons when there is more than one in a page.

**tabindex** This attribute uses a number to identify the object's position in the tabbing order.

**type** This attribute defines the action of the button. Possible values include **button**, **reset**, and **submit**, which are used to indicate that the button is a plain button, form reset button, or form submission button, respectively. The XHTML specification suggests **submit** is the default, but browsers may not enforce this in practice.

**value** Defines the value that is sent to the server when the button is clicked. This might be useful when using multiple **submit** buttons that perform different actions to indicate which button was pressed to the handling server-side program.

#### Examples

```
<button name="Submit"
  value="Submit"
  type="Submit">Submit Request</button>
```

```
<button type="button"
  onclick="doSomething()">Click This Button</button>
```

```
<button type="button">
</button>
```

#### Notes

- It is illegal to associate an image map with an **<img>** tag that appears as the content of a **button** element.
- The HTML 4.01 specification reserves the data-binding attributes **datafld**, **dataformatas**, and **datasrc** for future use. They were dropped from XHTML but Internet Explorer does support them.
- Developers may want to consider using the markup **<input type="submit">** instead of a **<button>** tag for complete browser backward compatibility.

## **<caption>** (Table Caption)

This element is used within the table element to define a caption.

#### Standard Syntax

```
<caption
```

```
align="bottom | left | right | top" (transitional only)
class="class name(s)"
dir="ltr | rtl"
id="unique alphanumeric identifier"
lang="language code"
style="style information"
title="advisory text">
```

</caption>

### Element-Specific Attributes

**align** This attribute specifies the alignment of the caption. HTML 4 defines **bottom**, **left**, **right**, and **top** as legal values. Internet Explorer also supports **center**. Because this does not provide the possibility to combine vertical and horizontal alignments, Microsoft has introduced the **valign** attribute for the **caption** element.

**valign** This Internet Explorer-specific attribute specifies whether the table caption appears at the **top** or **bottom**. The default is top.

### Example

```
<table>
  <caption align="top">Our High-Priced Menu</caption>
  <tr>
    <td>Escargot</td>
    <td>Filet Mignon</td>
    <td>Big Mac</td>
  </tr>
</table>
```

### Notes

- There should be only one caption per table.
- HTML 3.2 defines only the **align** attribute with values of **bottom** and **top**. No other attributes are defined prior to HTML 4.

## <center> (Center Alignment)

This element causes the enclosed content to be centered within the margins currently in effect. Margins are either the default page margins or those imposed by overriding elements, such as tables.

### Standard Syntax (Transitional Only)

```
<center
  class="class name(s)"
  dir="ltr | rtl"
  id="unique alphanumeric identifier"
  lang="language code"
  style="style information"
  title="advisory text">
```

</center>

## Example

`<center>`This is in the center of the page.`</center>`

## Notes

- The **center** element defined by the W3C is a shorthand notation for `<div align="center">`.
- The strict versions of HTML and XHTML do not include the **center** element, but it is easily imitated with the **text-align** CSS property.
- HTML 3.2 does not support any attributes for this element.

## `<cite>` (Citation)

This element indicates a citation from a book or other published source and usually is rendered in italics by a browser.

### Standard Syntax

```
<cite
  class="class name(s)"
  dir="ltr | rtl"
  id="unique alphanumeric identifier"
  lang="language code"
  style="style information"
  title="advisory text">
```

```
</cite>
```

### Example

This example is taken from `<cite>`HTML: The Complete Reference`</cite>`.

## `<code>` (Code Listing)

This element indicates that the enclosed text is source code in a programming language. Usually, it is rendered in a monospaced font.

### Standard Syntax

```
<code
  class="class name(s)"
  dir="ltr | rtl"
  id="unique alphanumeric identifier"
  lang="language code"
  style="style information"
  title="advisory text">
```

```
</code>
```

### Example

To increment a variable called count, use `<code>` count++ `</code>`

## <col> (Table Column)

This element defines a column within a table and is used for grouping and alignment purposes. It generally is found within a **colgroup** element.

### Standard Syntax

```
<col
  align="center | char | justify | left | right"
  char="character"
  charoff="number"
  class="class name(s)"
  dir="ltr | rtl"
  id="unique alphanumeric identifier"
  lang="language code"
  span="number"
  style="style information"
  title="advisory text"
  valign="baseline | bottom | middle | top"
  width="column width specification" />
```

### Element-Specific Attributes

**align** This attribute specifies horizontal alignment of a cell's contents.

**bgcolor** This Internet Explorer-specific attribute sets the background color for the column. Its value can be either a browser-dependent named color or a color specified using the hexadecimal #RRGGBB format.

**char** This attribute is used to set the character on which the cells in a column should be aligned. A typical value for this is a period (.) for aligning numbers or monetary values.

**charoff** This attribute is used to indicate the number of characters by which the column data should be offset from the alignment characters specified by the **char** value.

**span** When present, this attribute applies the attributes of the **col** element to additional consecutive columns.

**valign** This attribute specifies the vertical alignment of the text within the cell. Possible values for this attribute are **baseline**, **bottom**, **middle**, and **top**.

**width** This attribute specifies a default width for each column in the current column group. In addition to the standard pixel and percentage values, this attribute might take the special form **0\***, which means that the width of each column in the group should be the minimum width necessary to hold the column's contents.

### Example

```
<table border="1" width="400">
<colgroup>
  <col align="center" width="150" />
  <col align="right" />
</colgroup>
<tr>
  <td>This column is aligned to the center.</td>
  <td>This one is aligned to the right.</td>
</tr>
<tr><td>!</td><td>?</td></tr>
```

```
<tr><td>!</td><td>?</td></tr>
</table>
```

### Notes

- Under XHTML 1.0, `<col>` requires a trailing slash: `<col />`.
- This element should appear within a `<colgroup>` element, and like that element, it is somewhat of a convenience feature used to set attributes with one or more table columns. In practice, few developers seem to use it.

## <colgroup> (Table Column Group)

This element creates an explicit group of table columns to allow group level scripting or formatting.

### Standard Syntax

```
<colgroup
  align="center | char | justify | left | right"
  char="character"
  charoff="number"
  class="class name(s)"
  dir="ltr | rtl"
  id="unique alphanumeric identifier"
  lang="language code"
  span="number"
  style="style information"
  title="advisory text"
  valign="baseline | bottom | middle | top"
  width="column width specification">
```

col elements only

```
</colgroup>
```

### Element-Specific Attributes

**align** This attribute specifies horizontal alignment of the contents of the cells in the column group. The values of **center**, **left**, and **right** have obvious meanings. A value of **justify** for the attribute should attempt to justify all the column's contents. A value of **char** attempts to align the contents based on the value of the **char** attribute in conjunction with **charoff**.

**bgcolor** This Internet Explorer-specific attribute sets the background color for the columns in the column group. Its value can be either a browser-dependent named color or a color specified using the hexadecimal #RRGGBB format.

**char** This attribute is used to set the character on which the cells in a column should be aligned. A typical value for this attribute is a period (.) for aligning numbers or monetary values.

**charoff** This attribute is used to indicate the number of characters by which the column data should be offset from the alignment characters specified by the **char** value.

**span** When present, this attribute specifies the default number of columns in this group. Browsers should ignore this attribute if the current column group contains one or more `<col>` tags. The default value of this attribute is **1**.

**valign** This attribute specifies the vertical alignment of the contents of the cells within the column group.

**width** This attribute specifies a default width for each column and its cells in the current column group. In addition to the standard pixel and percentage values, this attribute can take the special form **0\***, which means that the width of each column in the group should be the minimum width necessary to hold the column's contents.

#### Examples

```
<colgroup span="2" align="char" char=":" valign="center">
  <col /><col /><col />
</colgroup>
```

```
<colgroup style="background-color: green;">
  <col align="left" />
  <col align="center" />
</colgroup>
```

#### Notes

- Each column group defined by a **<colgroup>** tag can contain zero or more **<col>** tags.
- Under XHTML 1.0, the closing **</colgroup>** tag is mandatory.

## <comment> (Comment Information)

This nonstandard element treats enclosed text as comments. This element should not be used.

#### Element-Specific Attributes

**data** References a URL that contains the comment information.

#### Example

```
<comment>This is not the proper way to form comments!!!</comment>
```

#### Notes

- It is better to use the **<!--. .-->** syntax for specifying comments.
- Because the **comment** element is not supported by all browsers, commented text done in this fashion will appear in other browsers.

## <dd> (Definition in a Definition List)

This element indicates the definition of a term within a list of defined terms (**<dt>**) enclosed by a definition list (**<dl>**).

#### Standard Syntax

```
<dd
  class="class name(s)"
  dir="ltr | rtl"
  id="unique alphanumeric identifier"
  lang="language code"
  style="style information"
  title="advisory text">
```

</dd>

### Element-Specific Attributes

**nowrap** This Internet Explorer-specific attribute is used to control the wrapping of text within a <dd> tag. If set to **yes**, text should not wrap. The default is **no**. CSS rules should be used instead of this attribute.

### Example

<dl>

<dt>DOG</dt>

<dd>A domesticated animal that craves attention all the time</dd>

<dt>CAT</dt>

<dd>An animal that would just as soon ignore you until it gets hungry</dd>

</dl>

### Notes

- Under HTML specifications, the closing tag for this element is optional, though it is encouraged when it will help make the list more understandable.
- Under XHTML 1.0, the closing </dd> tag is mandatory.
- This element occurs within a list of defined terms enclosed by a <dl> tag. Typically associated with it is the term it defines, indicated by the <dt> tag that precedes it.
- HTML 2 and 3.2 define no attributes for this element.

## <del> (Deleted Text)

This element is used to indicate that text has been deleted from a document. A browser might render deleted text as strikethrough text.

### Standard Syntax

<del

  cite="url"

  class="class name(s)"

  datetime="date"

  dir="ltr | rtl"

  id="unique alphanumeric identifier"

  lang="language code"

  style="style information"

  title="advisory text">

</del>

### Element-Specific Attributes

**cite** The value of this attribute is a URL that designates a source document or message that might give a reason that the information was deleted.

**datetime** This attribute is used to indicate the date and time the deletion was made. The value of the attribute is a date in a special format as defined by ISO 8601. The basic date format is YYYY-MM-DDThh:mm:ssTZD

where the following is true:

YYYY=four-digit year such as 1999

MM=two-digit month (01=January, 02=February, and so on.)  
DD=two-digit day of the month (01 through 31)  
hh=two digit hour (00 to 23) (24-hour clock, not AM or PM)  
mm=two digit minute (00 through 59)  
ss=two digit second (00 through 59)  
TZD=time zone designator

The time zone designator is either **Z**, which indicates Universal Time Coordinate or coordinated universal time format (UTC), or **+hh:mm**, which indicates that the time is a local time that is hh hours and mm minutes ahead of UTC. Alternatively, the format for the time zone designator could be **-hh:mm**, which indicates that the local time is behind UTC. Note that the letter "T" actually appears in the string, all digits must be used, and **00** values for minutes and seconds might be required. An example value for the **datetime** attribute might be **1999-10-6T09:15:00-05:00**, which corresponds to October 6, 1999, 9:15 A.M., U.S. Eastern Standard Time.

#### Example

```
<del cite="http://www.bigcompany.com/changes/oct97.htm"
  datetime="1998-10-06T09:15:00-05:00">
```

The penalty clause applies to client lateness as well.

```
</del>
```

#### Notes

- Browsers can render deleted (**<del>**) text in a different style to show the changes that have been made to the document. Internet Explorer renders the text as strikethrough text. Eventually, a browser could have a way to show a revision history on a document.
- User agents that do not understand **<del>** or **<ins>** will show the information anyway, so there is no harm in adding information-only in deleting it. Because of the fact that **<del>**-enclosed text might show up, it might be wise to comment it out within the element, as shown here:
  - **<del>**
  - **<!-- This is old information. -->**
  - **</del>**

## **<dfn>** (Definition)

This element encloses the defining instance of a term. It usually is rendered as bold or bold italic text.

#### Standard Syntax

```
<dfn
  class="class name(s)"
  dir="ltr | rtl"
  id="unique alphanumeric identifier"
  lang="language code"
  style="style information"
  title="advisory text">
```

```
</dfn>
```

### Example

`<p>An <dfn>elephant</dfn> is too large to make a viable pet for anyone poorer than Bill Gates.</p>`

## `<dir>` (Directory List)

This element encloses a list of brief, unordered items, such as might occur in a menu or directory.

### Standard Syntax (Transitional Only-Deprecated)

```
<dir  
  class="class name(s)"  
  compact="compact"  
  dir="ltr | rtl"  
  id="unique alphanumeric identifier"  
  lang="language code"  
  style="style information"  
  title="advisory text">
```

li elements only

```
</dir>
```

### Element-Specific Attributes

**compact** This attribute reduces the white space between list items.

### Example

```
<dir>  
  <li>Header Files</li>  
  <li>Code Files</li>  
  <li>Comment Files</li>  
</dir>
```

### Notes

- Because the `<dir>` tag is supposed to be used with short lists, the items in the list should have a maximum width of 20 characters. This is rarely respected.
- The HTML and XHTML strict specifications do not support this element.
- Most browsers will not render the `<dir>` tag any differently from the **ul** element.
- HTML 2 and 3.2 define only the **compact** attribute.
- Most browsers will not render the **compact** list style.
- For XHTML compatibility, the **compact** attribute must have a value: `<dir compact="compact">`.

## `<div>` (Division)

This element indicates a generic block of document content that should be treated as a logical unit and will have no default rendering or meaning.

## Standard Syntax

```
<div  
  align="center | justify | left | right" (transitional only)  
  class="class name(s)"  
  dir="ltr | rtl"  
  id="unique alphanumeric identifier"  
  lang="language code"  
  style="style information"  
  title="advisory text">
```

```
</div>
```

### Element-Specific Attributes

**align** This attribute indicates how the tagged text should be horizontally aligned on the page. The default value is **left**.

**nowrap** This Internet Explorer-specific attribute is used to control the wrapping of text within a **<div>** tag. If set to **yes**, text should not wrap. The default is **no**. CSS rules should be used instead of this attribute.

### Examples

```
<div align="justify">
```

All text within this division will be justified

```
</div>
```

```
<div class="special" id="div1" style="background: yellow;">
```

Divs are useful for setting arbitrary style

```
</div>
```

### Notes

- A **<div>** tag is a generic block tag and is very useful for binding scripts or styles to an arbitrary section of a document. It complements **<span>**, which is used inline.
- The HTML 4.01 specification specifies that the **datafld**, **dataformatas**, and **datasrc** attributes are reserved for **<div>** and might be supported in the future. They were removed from XHTML but Internet Explorer supports them for data binding.
- Under the HTML 4.01 strict specification, the **align** attribute is not supported.
- HTML 3.2 supports only the **align** attribute.

## <dl> (Definition List)

This element encloses a list of terms and definition pairs. A common use for this element is to implement a glossary.

### Standard Syntax

```
<dl  
  class="class name(s)"  
  compact="compact" (transitional only)  
  dir="ltr | rtl"  
  id="unique alphanumeric identifier"  
  lang="language code"  
  style="style information"
```

```
title="advisory text">
```

dt and dd elements only

```
</dl>
```

### Element-Specific Attributes

**compact** This attribute reduces the white space between list items.

### Example

```
<dl>
```

```
<dt>Cat</dt>
```

```
<dd>A domestic animal that likes fish.</dd>
```

```
<dt>Skunk</dt>
```

```
<dd>A wild animal that needs deodorant.</dd>
```

```
</dl>
```

### Notes

- The items in the list comprise two parts: the term, indicated by the **dt** element, and its definition, indicated by the **dd** element.
- Some page designers might use a **<dl>** tag or **<ul>** tag to create text indentation. Although this is a common practice on the Web, it is not advisable because it confuses the meaning of the element by making it a physical layout device rather than a list.
- HTML 2 and 3.2 support only the **compact** attribute for this element.
- For XHTML compatibility, the **compact** attribute must be expanded: **<dl compact="compact">** under the transitional form. It is deprecated under the strict specification.

## <dt> (Term in a Definition List)

Identifies a definition list term in a list of terms and definitions.

### Standard Syntax

```
<dt
```

```
class="class name(s)"
```

```
dir="ltr | rtl"
```

```
id="unique alphanumeric identifier"
```

```
lang="language code"
```

```
style="style information"
```

```
title="advisory text">
```

```
</dt>
```

### Element-Specific Attributes

**nowrap** This Internet Explorer-specific attribute is used to control the wrapping of text within a **<dt>** tag. If set to **yes**, text should not wrap. The default is **no**. CSS rules should be used instead of this attribute.

### Example

```
<dl>
```

```
<dt>Vole</dt>
```

```
<dd>Small creature related to the weasel</dd>
```

```
<dt>Weasel</dt>
  <dd>Small creature related to the vole</dd>
</dl>
```

### Notes

- This element occurs within a list of defined terms enclosed by a **<dl>** tag. It generally is used in conjunction with a **<dd>** tag, which indicates its definition. However, **<dt>** tags do not require a one-to-one correspondence with **<dd>** tags.
- The close tag for the element is optional under older versions of HTML but is suggested when it will make things more clear, particularly with multiple-line definitions.
- Under XHTML 1.0, the closing **</dt>** tag is mandatory.
- HTML 2 and 3.2 support no attributes for this element.

## <em> (Emphasis)

This element indicates emphasized text, which many browsers will display as italic text.

### Standard Syntax

```
<em
  class="class name(s)"
  dir="ltr | rtl"
  id="unique alphanumeric identifier"
  lang="language code"
  style="style information"
  title="advisory text">
```

```
</em>
```

### Example

This is an **<em>**important point**</em>** to consider.

### Notes

- As a logical element, **em** is a prime candidate to bind style information to. For example, to define emphasis to mean a larger font size in the Impact font, you might use a CSS rule like the following in a document-wide style sheet:  
em {font-size: larger; font-family: Impact;}
- HTML 2 and 3.2 support no attributes for this element.

## <embed> (Embedded Object)

This widely supported nonstandard element specifies an object, typically a multimedia element, to be embedded in an HTML document.

### Proprietary Syntax (Commonly Supported)

```
<embed
  accesskey="key" (5.5)
  align="absbottom | absmiddle | baseline | bottom |
    left | middle | right | texttop | top" (4)
```

alt="alternative text" (4)  
class="class name(s)" (4)  
code="filename" (4)  
codebase="url" (4)  
height="pixels" (4)  
hspace="pixels" (4)  
id="unique alphanumeric identifier" (4)  
language="javascript | jscript | vbs | vbscript | xml" (5.5)  
name="string" (4)  
src="url" (4)  
style="style information" (4)  
title="advisory text" (4)  
unselectable="on | off" (5.5)  
vspace="pixels" (4)  
width="pixels" (4)>

</embed>

### Element-Specific Attributes

**align** This attribute controls the alignment of adjacent text with respect to the embedded object. The default value is **left**.

**alt** This attribute indicates the text to be displayed if the included object cannot be executed.

**border** This attribute specifies the size in pixels of the border around the embedded object.

**code** This attribute specifies the name of the file containing the compiled Java class if the **embed** element is used to include a Java applet. This is a strange alternative form of Java inclusion documented by Microsoft.

**codebase** This specifies the base URL for the plug-in or potential applet in the case of the alternative form under Internet Explorer.

**height** This attribute sets the height of the embedded object in pixels.

**hidden** If this attribute is set to the value **true**, the embedded object is not visible on the page and implicitly has a size of zero.

**hspace** This attribute specifies, in pixels, the size of the left and right margins between the embedded object and surrounding text.

**name** This attribute specifies a name for the embedded object, which can be referenced by client-side programs in an embedded scripting language.

**palette** This attribute is used only on Windows systems to select the color palette used for the plug-in and might be set to **background** or **foreground**. The default is **background**.

**pluginspage** This attribute contains the URL of instructions for installing the plug-in required to render the embedded object.

**src** This attribute specifies the URL of source content for the embedded object.

**type** This attribute specifies the MIME type of the embedded object. It is used by the browser to determine an appropriate plug-in for rendering the object. It can be used instead of the **src** attribute for plug-ins that have no content or that fetch it dynamically.

**units** This Netscape-specific attribute is used to set the units for measurement for the embedded object either in **en** or in the default, **pixels**.

**vspace** This attribute specifies, in pixels, the size of the top and bottom margins between the embedded object and surrounding text.

**width** This attribute sets the width, in pixels, of the embedded object.

### Examples

```
<!-- embed without a close tag -->
```

```
<embed src="testmovie.mov" height="150" width="150">
```

```
</noembed>
```

```
  
```

```
</noembed>
```

```
<!-- embed with a close tag -->
```

```
<embed src="testmovie.mov" height="150" width="150">
```

```
</noembed>
```

```
  
```

```
</noembed>
```

```
</embed>
```

### Notes

- It is actually unclear whether or not the close tag for **<embed>** is required. Many sites tend not to use it, and documentation is not consistent. Some people claim that a close tag is required and should surround any alternative content in a **noembed** element; others do not use a close tag. Whatever the case, this element should be phased out in favor of **object**, so this might be a moot issue.
- The **embed** element is not favored by the W3C and is not part of any official HTML or XHTML specification; however, it is very common. The HTML specification says to use the **object** element, which can be used in conjunction with the **embed** element to provide backward compatibility.
- Embedded objects are multimedia content files of arbitrary type that are rendered by browser plug-ins. The **type** attribute uses a file's MIME type to determine an appropriate browser plug-in. Any attributes not defined are treated as object-specific parameters and are passed through to the embedded object. Consult the plug-in or object documentation to determine these. The standard parameters supported by the Microsoft implementation are **height**, **name**, **palette**, **src**, **units**, and **width**.

## <fieldset> (Form Field Grouping)

This element allows form designers to group thematically related controls together.

### Standard Syntax

```
<fieldset
```

```
  class="class name(s)"
```

```
  dir="ltr | rtl"
```

```
  id="unique alphanumeric identifier"
```

```
  lang="language code"
```

```
  style="style information"
```

```
  title="advisory text">
```

```
</fieldset>
```

### Element-Specific Attributes

**align** Internet Explorer defines the **align** attribute, which sets how the element and its contents are positioned in a table or the window.

**datafld** This attribute specifies the column name from the data source object that supplies the bound data. This attribute is specific to Microsoft's data binding.

### Example

```
<fieldset>
<legend>Customer Identification</legend>
<br />
<label>Customer Name:
<input type="text" id="CustName" size="25" />
</label>
</fieldset>
```

### Notes

- Grouping controls makes it easier for users to understand the purposes of the controls while simultaneously facilitating tabbing navigation for visual user agents and speech navigation for speech-oriented user agents. The proper use of this element makes documents more accessible to people with disabilities.
- The caption for a **<fieldset>** tag can be defined by the **legend** element.
- The typical visual rendering of a fieldset is a boxed grouping of form fields with a label defined by the **legend** element.

## <font> (Font Definition)

This element allows specification of the size, color, and font of the text it encloses.

### Standard Syntax (Transitional Only)

```
<font
  class="class name(s)"
  color="color name | #RRGGBB"
  dir="ltr | rtl"
  face="font name"
  id="unique alphanumeric identifier"
  lang="language code"
  size="1 to 7 | +1 to +6 | -1 to -6"
  style="style information"
  title="advisory text">
```

```
</font>
```

### Element-Specific Attributes

**color** This attribute sets the text color using either a browser-dependent named color or a color specified in the hexadecimal #RRGGBB format.

**face** This attribute contains a list of one or more font names separated by commas. The user agent looks through the specified font names and renders the text in the first font that is supported.

**point-size** This Netscape 4-specific attribute specifies the point size of text and is used with downloadable fonts.

**size** This attribute specifies the font size as either a numeric or relative value. Numeric values range from **1** to **7** with **1** being the smallest and **3** the default. The relative values, + and -, increment or decrement the font size relative to the current size. The value for increment or decrement should range only from **+1** to **+6** or **-1** to **-6**.

**weight** Under Netscape 4, this attribute specifies the weight of the font, with a value of **100** being lightest and **900** being heaviest.

#### Example

```
<font color="#FF0000" face="Helvetica, Times Roman" size="+1">
```

Relatively large red text in Helvetica or Times.

```
</font>
```

- Interestingly, the transitional specification for some reason does not define core events for this element. In practice, they are supported by major browsers.
- The default text size for a document can be set using the **size** attribute of the **basefont** element.
- The HTML 3.2 specification supports only the **color** and **size** attributes for this element.

## <form> (Form for User Input)

The element defines a fill-in form that can contain labels and form controls, such as menus and text entry boxes that might be filled in by a user.

### Standard Syntax

```
<form
  accept-charset="list of supported character sets"
  action="url"
  class="class name(s)"
  dir="ltr | rtl"
  enctype="application/x-www-form-urlencoded |
    multipart/form-data | text/plain |
    Media Type as per RFC 2045"
  id="unique alphanumeric identifier"
  lang="language code"
  method="get | post"
  name="form's name for scripting"
  style="style information"
  target="_blank | frame name | _parent | _self |
    _top" (transitional only)
  title="advisory text">
```

```
</form>
```

### Element-Specific Attributes

**accept-charset** This attribute specifies the list of character encodings for input data that must be accepted by the server processing the form. The value is a space- or comma-delimited list of character sets as defined in RFC 2045. The default value for this attribute is the reserved value **unknown**.

**action** This attribute contains the URL of the server program that will process the contents of the form. Some browsers also might support a mailto URL, which can mail the results to the specified address.

**autocomplete** This Microsoft proprietary attribute, introduced in Internet Explorer 5.0, will automatically finish filling in information that the user has previously input into an input field, and which has been encrypted and stored by the browser.

**enctype** This attribute indicates how form data should be encoded before being sent to the server. The default is **application/x-www-form-urlencoded**. This encoding replaces blank characters in the data with a plus character (+) and all other nonprinting characters with a percent sign (%) followed by the character's ASCII HEX representation. The multipart/form-data option does not perform character conversion and transfers the information as a compound MIME document. This must be used when using **<input-type="file">**. It also might be possible to use another encoding, such as text/plain, to avoid any form of hex encoding; this might be useful with mailed forms.

**method** This attribute indicates how form information should be transferred to the server. The **get** option appends data to the URL specified by the **action** attribute. This approach gives the best performance but imposes a size limitation determined by the command line length supported by the server. The **post** option transfers data using a HTTP post transaction. This approach is more secure and imposes no data size limitation.

**name** This attribute specifies a name for the form and can be used by client-side programs to reference form data.

**target** In documents containing frames, this attribute specifies the target frame that will display the results of a form submission. In addition to named frames, several special values exist. The **\_blank** value indicates a new window. The **\_parent** value indicates the parent frame set containing the source link. The **\_self** value indicates the frame containing the source link. The **\_top** value indicates the full browser window.

#### Example

```
<form action="http://www.bigcompany.com/cgi-bin/processit.exe"
method="post" name="testform" onsubmit="return validate();"
Enter your comments here:<br />
<textarea name="comments" cols="30" rows="8">
</textarea>
<br /><br />
<input type="submit" value="send" />
<input type="reset" value="clear" />
</form>
```

#### Notes

- Form content is defined using the **<button>**, **<input>**, **<select>**, and **<textarea>** tags, as well as other HTML formatting and structuring elements.
- Special grouping elements, such as **fieldset**, **label**, and **legend** are provided to structure form fields, but more often tags like **<div>** and **<table>** are used to improve form layout.
- HTML 2 and 3.2 support only the **action**, **enctype**, and **method** attributes for the **form** element.

## <frame> (Window Region)

This element defines a nameable window region, known as a frame, that can independently display its own content.

### Standard Syntax

```
<frame
  class="class name(s)"
  frameborder="0 | 1"
  id="unique alphanumeric identifier"
  longdesc="url of description"
  marginheight="pixels"
  marginwidth="pixels"
  name="frame name"
  noresize="noresize"
  scrolling="auto | no | yes"
  src="url" of frame contents
  style="style information"
  title="advisory text">
```

### Element-Specific Attributes

**allowtransparency** This Internet Explorer-specific attribute determines whether the <frame> is transparent or opaque. The default value is **false**, which means it is opaque.

**application** This Microsoft-specific attribute is used to indicate whether the content of an <frame> is to be considered an HTA application. HTA applications are applications that use HTML, JavaScript, and Internet Explorer, but are not limited to the typical type of security considerations of a Web page. Given its security implications, this attribute should only be set if the developer is familiar with HTAs.

**bordercolor** This attribute sets the color of the frame's border using either a named color or a color specified in the hexadecimal #RRGGBB format.

**datafld** This Internet Explorer attribute specifies the column name from the data source object that supplies the bound data.

**datasrc** This Internet Explorer attribute indicates the **id** of the data source object that supplies the data that is bound to this element.

**frameborder** This attribute determines whether the frame is surrounded by an outlined three-dimensional border. The HTML specification prefers the use of **1** for the frame border on, and **0** for off; most browsers also acknowledge the use of **no** and **yes**.

**longdesc** This attribute specifies the URL of a document that contains a long description of the frame's content. This attribute should be used in conjunction with the **title** element.

**marginheight** This attribute sets the height in pixels between the frame's contents and its top and bottom borders.

**marginwidth** This attribute sets the width in pixels between the frame's contents and its left and right borders.

**name** This attribute assigns the frame a name so that it can be the target destination of hyperlinks as well as being a possible candidate for manipulation via a script.

**noresize** This attribute overrides the default ability to resize frames and gives the frame a fixed size.

**scrolling** This attribute determines whether the frame has scroll bars. A **yes** value forces scroll bars, a **no** value prohibits them, and an **auto** value lets the browser decide. When not specified,

the default value of **auto** is used. Authors are recommended to leave the value as **auto**. If you turn off scrolling and the contents end up being too large for the frame (due to rendering differences, window size, and so forth), the user will not be able to scroll to see the rest of the contents. If you turn scrolling on and the contents all fit in the frame, the scroll bars will needlessly consume screen space. With the **auto** value, scroll bars appear only when needed.

**security** This attribute sets the value indicating whether the source file of a frame has security restrictions applied. The only allowed value is **restricted**.

**src** This attribute contains the URL of the contents to be displayed in the frame. If it is absent, nothing will be loaded in the frame.

#### Example

```
<frameset rows="20%,80%">
  <frame src="controls.html" name="controls" noresize="noresize" scrolling="no" />
  <frame src="content.html" />
</frameset>
```

#### Notes

- XHTML 1.0 requires a trailing slash for this element: **<frame />**.
- A frame must be declared as part of a frame set by using the **<frameset>** tag, which specifies the frame's relationship to other frames on a page. A frame set occurs in a special HTML document, in which the **frameset** element replaces the **body** element. Another form of frames called independent frames, or floating frames, also is supported. Floating frames can be directly embedded in a document without belonging to a frame set. These are defined with the **iframe** element.
- Many browsers do not support frames and require the use of the **<noframes>** tag.
- Frames introduce potential navigation difficulties; their use should be limited to instances in which they can be shown to help navigation rather than hinder it.

## <frameset> (Frameset Definition)

This element is used to define the organization of a set of independent window regions known as frames as defined by the **frame** element. This element replaces the **body** element in framing documents.

#### Standard Syntax

```
<frameset
  class="class name(s)"
  cols="list of columns"
  id="unique alphanumeric identifier"
  rows="list of rows"
  style="style information"
  title="advisory text">
```

```
</frameset>
```

#### Element-Specific Attributes

**border** This attribute sets the width in pixels of frame borders within the frame set. Setting **border="0"** eliminates all frame borders. This attribute is not defined in the HTML or XHTML specification but is widely supported.

**bordercolor** This attribute sets the color for frame borders within the frame set using either a named color or a color specified in the hexadecimal #RRGGBB format.

**cols** This attribute contains a comma-delimited list that specifies the number and size of columns contained within a set of frames. List items indicate columns from left to right. Column size is specified in three formats, which might be mixed. A column can be assigned a fixed width in pixels. It also can be assigned a percentage of the available width, such as 50%. Finally, a column can be set to expand to fill the available space by setting the value to \*, which acts as a wildcard.

**frameborder** This attribute controls whether or not frame borders should be displayed. Netscape supports **no** and **yes** values. Microsoft uses **1** and **0** as well as **no** and **yes**.

**framespacing** This attribute indicates the space between frames in pixels.

**rows** This attribute contains a comma-delimited list that specifies the number and size of rows contained within a set of frames. The number of entries in the list indicates the number of rows. Row size is specified with the same formats used for columns.

### Examples

<!-- This example defines a frame set of three columns. The middle column is 50 pixels wide. The first and last columns fill the remaining space.

-->

```
<frameset cols="*,50,*">
  <frame src="column1.html">
  <frame src="column2.html">
  <frame src="column3.html">
</frameset>
```

<!-- This example defines a frame set of two columns, one of which is 20% of the screen, and the other, 80%. -->

```
<frameset cols="20%, 80%">
  <frame src="controls.html">
  <frame src="display.html">
</frameset>
```

<!-- This example defines two rows, one of which is 10% of the screen, and the other, whatever space is left. -->

```
<frameset rows="10%, *">
  <frame src="adbanner.html" name="ad_frame">
  <frame src="contents.html" name="content_frame">
</frameset>
```

### Notes

- The content model says that the **<frameset>** tag contains one or more **<frame>** tags, which are used to indicate the framed contents. A **<frameset>** tag also might contain a **<noframes>** tag whose contents will be displayed by browsers that do not support frames.
- The **<frameset>** tag replaces the **<body>** tag in a framing document, as shown here:

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en">
<head>
<title>Collection of Frames</title>
<meta http-equiv="content-type" content="text/html; charset=ISO-8859-1" />
</head>
<frameset cols="*,50,*">
  <frame src="column1.htm" name="col1" />
  <frame src="column2.htm" name="col2" />
  <frame src="column3.htm" name="col3" />
</frameset>
<body>
Please visit our <a href="noframes.html">no frames</a> site.
</body>
</noframes>
</frameset>
</html>

```

## <h1> Through <h6> (Headings)

These tags implement six levels of document headings; <h1> is the most prominent and <h6> is the least prominent.

### Standard Syntax

```

<h1
  align="center | justify | left | right"
    (transitional only)
  class="class name(s)"
  dir="ltr | rtl"
  id="unique alphanumeric identifier"
  lang="language code"
  style="style information"
  title="advisory text">

```

```
</h1>
```

### Element-Specific Attributes

**align** This attribute controls the horizontal alignment of the heading with respect to the page. The default value is **left**.

### Example

```

<h1 align="justify">This is a Major Document Heading</h1>
<h2 align="center">Second heading, aligned to the center</h2>
<h3 align="right">Third heading, aligned to the right</h3>
<h4>Fourth heading</h4>
<h5 style="font-size: 20pt;">Fifth heading with style information</h5>
<h6>The least important heading</h6>

```

## Notes

- In most implementations, heading numbers correspond inversely with the six font sizes supported by the **font** element. For example, **<h1>** corresponds to **<font size="6">**. The default font size is **3**. However, this approach to layout is not encouraged, and page designers should consider using styles to set even relative sizes.
- HTML 3.2 supports only the **align** attribute. HTML 2 does not support any attributes for headings.
- The strict definitions of HTML 4 and XHTML do not include support for the **align** attribute. Style sheet properties like **text-align** should be used instead.

## <head> (Document Head)

This element indicates the document head that contains descriptive information about the HTML document as well as other supplementary information, such as style rules or scripts.

### Standard Syntax

```
<head
  dir="ltr | rtl"
  id="unique alphanumeric identifier"
  lang="language code"
  profile="url">
```

```
</head>
```

### Element-Specific Attributes

**profile** This attribute specifies a URL for a meta-information dictionary. The specified profile should indicate the format of allowed metadata and its meaning.

### Example

```
<head>
<title>Big Company Home Page</title>
<base href="http://www.bigcompany.com" />
<meta name="Keywords"content="BigCompany, SuperWidget" />
</head>
```

```
<head profile="http://www.bigcompany.com/metadict.xml">
```

### Notes

- The **<head>** tag must contain a **<title>** tag. It also might contain the **<base>**, **<isindex>**, **<link>**, **<meta>**, **<script>**, and **<style>** tags. Internet Explorer 4 supports the inclusion of the **<basefont>** tag in the **<head>**, but **<basefont>** has been deprecated under HTML 4.
- Under the XHTML 1.0 specification, the **head** element no longer can be implied, but must be used in all documents.
- Under XHTML 1.0, the closing **</head>** tag is mandatory.
- The meaning of the **profile** attribute is somewhat unclear, and no browsers appear to support it in any meaningful way.
- Internet Explorer may allow the **<bgsound>** tag within **<head>**.
- HTML 2 and 3.2 support no attributes for this element.

## <hr> (Horizontal Rule)

This element is used to insert a horizontal rule to visually separate document sections. Rules usually are rendered as a raised or etched line.

### Standard Syntax

```
<hr
  align="center | left | right" (transitional only)
  class="class name(s)"
  dir="ltr | rtl"
  id="unique alphanumeric identifier"
  lang="language code" id="unique alphanumeric identifier"
  noshade="noshade " (transitional only)
  size="pixels" (transitional only)
  style="style information"
  title="advisory information"
  width="percentage | pixels" /> (transitional only)
```

### Element-Specific Attributes

**align** This attribute controls the horizontal alignment of the rule with respect to the page. The default value is **left**.

**color** This attribute sets the rule color using either a named color or a color specified in the hexadecimal *#RRGGBB* format. This attribute currently is supported only by Internet Explorer.

**noshade** This attribute causes the rule to be rendered as a solid bar without shading.

**size** This attribute indicates the height in pixels of the rule.

**width** This attribute indicates how wide the rule should be, specified either in pixels or as a percent of screen width, such as 80%.

### Examples

```
<hr align="left" noshade="noshade" size="1" width="420" />
```

```
<hr align="center" width="100%" size="3" color="#000000" />
```

## <html> (HTML Document)

This element identifies an HTML or XHTML document.

### Standard Syntax

```
<html
  dir="ltr | rtl"
  id="unique alphanumeric identifier"
  lang="language code"
  xmlns="http://www.w3.org/1999/xhtml">
```

```
</html>
```

### Element-Specific Attributes

**scroll** This attribute is used to set whether scroll bars should show for the document or not. The default value of **auto** puts in scroll bars as needed. This attribute, while documented by Microsoft, does not appear to work properly and should be avoided.

**xmlns** This attribute declares a namespace for XML-based custom tags in the document. For XHTML, this value is always <http://www.w3.org/1999/xhtml>.

### Example

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en">
<head>
<title>Minimal Document</title>
<meta http-equiv="content-type" content="text/html; charset=ISO-8859-1" />
</head>
<body>
<p>Hello world!</p>
</body>
</html>
```

### Notes

- The **html** element is the first element in a document. Except for comments, the only tags it directly contains are **<head>** followed by either **<body>** or **<frameset>**.
- Because it is the outermost tag in a document, the **html** element is called the root element.
- Under the XHTML 1.0 specification, **<html>** can no longer be implied.
- The **<html>** tag and its closing tag **</html>** are both mandatory under XHTML.
- Under HTML 4 transitional, a **version** attribute is supported.

## **<i>** (Italic)

Indicates that the enclosed text should be displayed in an italic typeface.

### Standard Syntax

```
<i
  class="class name(s)"
  dir="ltr | rtl"
  id="unique alphanumeric identifier"
  lang="language code"
  style="style information"
  title="advisory text">
```

```
</i>
```

### Examples

Here is some **<i>**italicized**</i>** text.

This is also **<i style="color:red;" id="myItalic">**italic**</i>**.

## <iframe> (Floating Frame)

This element indicates a floating frame, an independently controllable content region that can be embedded in a page.

### Standard Syntax (Transitional and Frameset Only)

```
<iframe
  align="bottom | left | middle | right | top"
  class="class name(s)"
  dir="ltr | rtl"
  frameborder="1 | 0"
  height="percentage | pixels"
  id="unique alphanumeric identifier"
  lang="language code"
  longdesc="url of description"
  marginheight="pixels"
  marginwidth="pixels"
  name="string"
  scrolling="auto | no | yes"
  src="url of frame contents"
  style="style information"
  title="advisory text"
  width="percentage | pixels">
```

</iframe>

### Element-Specific Attributes

**align** This attribute controls the horizontal alignment of the floating frame with respect to the page. The default is **left**.

**allowtransparency** This Internet Explorer-specific attribute determines whether the <iframe> is transparent or opaque. The default value is **false**, which means it is opaque.

**application** This Microsoft-specific attribute is used to indicate whether the contents of an <iframe> are to be considered an HTA application. HTA applications are applications that use HTML, JavaScript, and Internet Explorer, but are not limited to the typical type of security considerations of a Web page. Given its security implications, this attribute should only be set if the developer is familiar with HTAs.

**border** This attribute specifies the thickness of the border in pixels.

**bordercolor** This attribute specifies the color of the border.

**datafld** This attribute specifies the column name from the data source object that supplies the bound data.

**datasrc** This attribute indicates the **id** of the data source object that supplies the data that is bound to this element.

**frameborder** This attribute determines whether the **iframe** is surrounded by a border. The HTML 4 specification defines **0** to be off and **1** to be on. The default value is **1**. Internet Explorer also defines the values **no** and **yes**.

**framespacing** This attribute creates additional space between the frames.

**height** This attribute sets the floating frame's height in pixels.

**hspace** This attribute specifies horizontal padding between the **iframe** and any contents that may flow around it.

**longdesc** This attribute specifies the URL of a document that contains a long description of the frame's contents.

**marginheight** This attribute sets the height in pixels between the floating frame's content and its top and bottom borders.

**marginwidth** This attribute sets the width in pixels between the floating frame's content and its left and right borders.

**name** This attribute assigns the floating frame a name so that it can be the target destination of hyperlinks.

**scrolling** This attribute determines whether the frame has scroll bars. A **yes** value forces scroll bars; a **no** value prohibits them. The default value is **auto**.

**security** This attribute sets the value indicating whether the source file of an **iframe** has security restrictions applied. The only allowed value is **restricted**.

**src** This attribute contains the URL of the content to be displayed in the floating frame. If absent, the frame is blank.

**vspace** This attribute specifies vertical padding between an **iframe** and any content that may flow around it.

**width** This attribute sets the floating frame's width in pixels.

#### Example

```
<iframe src="http://www.democompany.com" height="150" width="200"
  name="FloatingFrame1">
```

Sorry, your browser doesn't support inline frames.

```
</iframe>
```

#### Notes

- Under the HTML 4 strict specification, the **iframe** element is not defined. However, under XHTML transitional and XHTML frameset, **iframe** is allowed. XHTML 1.1 does not allow it either. Floating frames can be somewhat imitated using the **div** element and CSS positioning facilities.
- When a browser does not understand an **<iframe>** tag, it displays the text included within it as an alternate rendering.

## <ilayer> (Inflow Layer)

This Netscape 4-specific element allows the definition of overlapping content layers that can be positioned, hidden or shown, rendered transparent or opaque, reordered front to back, and nested. An *inflow layer* is a layer with a relative position that appears where it would naturally occur in the document, in contrast to a *general layer*, which might be positioned absolutely, regardless of its location in a document. The functionality of layers is available using CSS positioning, and page developers are advised not to use this element.

#### Syntax (Netscape 4 Only)

```
<ilayer
  above="layer"
  background="url of image"
  below="layer"
  bgcolor="color name | #RRGGBB"
  class="class name(s)"
```

```
clip="x1, y1, x2, y2"  
height="percentage | pixels"  
id="unique alphanumeric identifier"  
left="pixels"  
name="string"  
pagex="pixels"  
pagey="pixels"  
src="url of layer contents"  
style="style information"  
top="pixels"  
visibility="hide | inherit | show"  
width="percentage | pixels"  
z-index="number">
```

</ilayer>

### Element-Specific Attributes

**above** This attribute contains the name of the layer to be rendered above the current layer.

**background** This attribute contains the URL of a background image for the layer.

**below** This attribute contains the name of the layer to be rendered below the current layer.

**bgcolor** This attribute specifies a layer's background color. Its value can be either a named color or a color specified in the hexadecimal *#RRGGBB* format.

**clip** This attribute specifies the clipping region or viewable area of the layer. All layer content outside that rectangle will be rendered as transparent. The **clip** rectangle is defined by two *x,y* pairs: top *x*, left *y*, bottom *x*, and right *y*. Coordinates are relative to the layer's origin point, **0,0**, in its top-left corner.

**height** This attribute specifies the height of a layer in pixels or percentage values.

**left** This attribute specifies, in pixels, the horizontal offset of the layer. The offset is relative to its parent layer if it has one or to the left page margin if it does not.

**name** This attribute assigns the layer a name that can be referenced by programs in a client-side scripting language. The **id** attribute also can be used.

**pagex** This attribute specifies the horizontal position of the layer relative to the browser window.

**pagey** This attribute specifies the vertical position of the layer relative to the browser window.

**src** This attribute is used to set the URL of a file that contains the content to be loaded into the layer.

**style** This attribute specifies an inline style for the layer.

**top** This attribute specifies, in pixels, the top offset of the layer. The offset is relative to its parent layer if it has one or the top page margin if it does not.

**visibility** This attribute specifies whether a layer is hidden, shown, or inherits its visibility from the layer that includes it.

**width** This attribute specifies a layer's width in pixels.

**z-index** This attribute specifies a layer's stacking order relative to other layers. Position is specified with positive integers, with **1** indicating the bottommost layer.

### Example

```
<p>Content comes before.</p>
```

```
<ilayer name="background" bgcolor="green">
```

```
<p>Layered information goes here.</p>
</ilayer>
<p>Content comes after.</p>
```

## <img> (Image)

This element indicates a media object to be included in an HTML document. Usually, the object is a graphic image, but some implementations support movies and animations.

### Standard Syntax

```

```

### Element-Specific Attributes

**align** This attribute controls the horizontal alignment of the image with respect to the page. The default value is **left**. Netscape and Internet Explorer implementations support the **absbottom**, **absmiddle**, **baseline**, and **texttop** values. This attribute is deprecated under strict variants of HTML and XHTML.

**alt** This attribute contains a string to be displayed instead of the image for browsers that cannot display images.

**border** This attribute indicates the width, in pixels, of the border surrounding the image.

**datafld** This attribute specifies the column name from the data source object that supplies the bound data. In this situation, the bound data is used to set the **src** of an **<img>** tag.

**datasrc** This attribute indicates the **id** of the data source object that supplies the data that is bound to this **<img>** tag.

**dynsrc** In the Microsoft implementation, this attribute indicates the URL of a movie file and is used instead of the **src** attribute.

**galleryimg** This Microsoft attribute is used to control whether the gallery image menu should appear when the mouse pointer hovers over an image. The default value is **true** or **yes**. A value

of **no** or **false** suppresses the menu. A meta tag like `<meta http-equiv="imagemetoolbar" content="no" />` can be used to suppress the image toolbar document wide.

**height** This attribute indicates the height, in pixels or percentage of the screen, of the image.

**hspace** This attribute indicates the horizontal space, in pixels, between the image and surrounding text.

**ismap** This attribute indicates that the image is a server-side image map. User mouse actions over the image are sent to the server for processing.

**longdesc** This attribute specifies the URL of a document that contains a long description of the image. This attribute is used as a complement to the **alt** attribute.

**loop** In the Microsoft implementation, this attribute is used with the **dynsrc** attribute to cause a movie to loop. Its value is either a numeric loop count or the keyword **infinite**. Later versions of Internet Explorer suggest using **-1** to suggest infinite.

**lowsrc** This nonstandard attribute supported in most browsers contains the URL of an image to be initially loaded. Typically, the **lowsrc** image is a low-resolution or black-and-white image that provides a quick preview of the image to follow. Once the primary image is loaded, it replaces the **lowsrc** image.

**name** This common attribute is used to bind a name to the image. Older browsers understand the **name** field, and in conjunction with scripting languages, it is possible to manipulate images by their defined names to create effects such as "rollover" buttons. While under future versions of HTML and XHTML the **id** attribute specifies element identifiers, **name** can still be used for backward compatibility.

**src** This attribute indicates the URL of an image file to be displayed.

**usemap** This attribute makes the image support client-side image mapping. Its argument is a URL specifying the map file, which associates image regions with hyperlinks.

**vspace** This attribute indicates the vertical space in pixels between the image and surrounding text.

**width** This attribute indicates the width in pixels of the image.

### Examples

```

```

```

```

```
<a href="home.htm"></a>
```

### Notes

- Typically, when you use the **usemap** attribute, the URL is a fragment, such as #map1, rather than a full URL. Some browsers do not support external client-side map files.
- Under the strict HTML and XHTML definitions, the `<img>` tag does not support **align**, **border**, **height**, **hspace**, **vspace**, and **width**. The functionality of these attributes should be possible using style sheet rules.
- Whereas the HTML 4 specification reserves data-binding attributes such as **datafld** or **datasrc**, it is not specified for `<img>`, although Internet Explorer provides support for these attributes.
- XHTML 1.0 requires a trailing slash for this element: `<img />`.

- Under future versions of XHTML such as 2, `<img>` may be dropped in favor of `<object>`.

## `<input>` (Input Form Control)

This element specifies an input control for a form. The type of input is set by the **type** attribute and can be a variety of different types, including single-line text field, password field, hidden, check box, radio button, or push button.

### Standard Syntax

`<input`

```
  accept="MIME types"
  accesskey="character"
  align="bottom | left | middle | right | top" (transitional only)
  alt="text"
  checked="checked"
  class="class name(s)"
  dir="ltr | rtl"
  disabled="disabled"
  id="unique alphanumeric identifier"
  lang="language code"
  maxlength="maximum field size"
  name="field name"
  readonly="readonly"
  size="field size"
  src="url of image file"
  style="style information"
  tabindex="number"
  title="advisory text"
  type="button | checkbox | file | hidden | image |
      password | radio | reset | submit | text"
  usemap="url of map file"
  value="field value" />
```

### Element-Specific Attributes

**accept** This attribute is used to list the MIME types accepted for file uploads when using a file upload control (`<input type="file">`).

**accesskey** This attribute specifies a keyboard navigation accelerator for the element. Pressing ALT or a similar key in association with the specified character selects the form control correlated with that key sequence. Page designers are forewarned to avoid key sequences already bound to browsers.

**align** With image form controls (`type="image"`), this attribute aligns the image with respect to surrounding text. The HTML 4.01 transitional specification defines **bottom**, **left**, **middle**, **right**, and **top** as allowable values. Netscape and Microsoft browsers might also allow the use of attribute values such as **absbottom** or **absmiddle**. Like other presentation-specific aspects of HTML, the **align** attribute is dropped under the strict HTML 4.01 specification.

**alt** This attribute is used to display an alternative description of image buttons for text-only browsers. The meaning of **alt** for forms of `<input>` beyond `type="image"` is unclear.

**autocomplete** This Microsoft-specific attribute is used to indicate whether the form field should be automatically filled in or not. The default value is **no**.

**checked** The **checked** attribute should be used only for check box (`type="checkbox"`) and radio (`type="radio"`) form controls. The presence of this attribute indicates that the control should be displayed in its checked state.

**disabled** This attribute is used to turn off a form control. Elements will not be submitted, nor will they receive any focus from the keyboard or mouse. Disabled form controls will not be part of the tabbing order. The browser also might gray out the form that is disabled, in order to indicate to the user that the form control is inactive. This attribute requires no value.

**dynsrc** In the Microsoft implementation, this attribute indicates the URL of a movie file and is used instead of the **src** attribute for `<input type="image">`.

**loop** In the Microsoft implementation, this attribute is used with `<input type="image">` and the **dynsrc** attribute to cause a movie to loop. Its value is either a numeric loop count or the keyword **infinite**. Later versions of Internet Explorer suggest using **-1** to suggest infinite.

**hspace** This Internet Explorer-specific attribute indicates the horizontal space in pixels between the image and surrounding text when using `<input type="image">`.

**lowsrc** This Microsoft-supported attribute contains the URL of an image to be initially loaded when using `<input type="image">`. Typically, the **lowsrc** image is a low-resolution or black-and-white image that provides a quick preview of the image to follow. Once the primary image is loaded, it replaces the **lowsrc** image.

**maxlength** This attribute indicates the maximum content length that can be entered in a text form control (`type="text"`). The maximum number of characters allowed differs from the visible dimension of the form control, which is set with the **size** attribute.

**name** This attribute allows a form control to be assigned a name so that it can be referenced by a scripting language. **Name** is supported by older browsers, such as Netscape 2-generation browsers, but the W3C encourages the use of the **id** attribute. For compatibility purposes, both might have to be used.

**readonly** This attribute prevents the form control's value from being changed. Form controls with this attribute set might receive focus from the user but might not be modified. Because it receives focus, a **readonly** form control will be part of the form's tabbing order. The control's value will be sent on form submission. This attribute can be used only with `<input>` when **type** is set to **text** or **password**. The attribute also is used with the **textarea** element.

**size** This attribute indicates the visible dimension, in characters, of a text form control (`type="text"`). This differs from the maximum length of content, which can be entered in a form control set by the **maxlength** attribute.

**src** This attribute is used with image form controls (`type="image"`) to specify the URL of the image file to load.

**tabindex** This attribute takes a numeric value that indicates the position of the form control in the tabbing index for the form. Tabbing proceeds from the lowest positive **tabindex** value to the highest. Negative values for **tabindex** will leave the form control out of the tabbing order. When tabbing is not explicitly set, the browser tabs through items in the order they are encountered. Disabled form fields will not be part of the tabbing index, although read-only controls will be.

**type** This attribute specifies the type of the form control. A value of **button** indicates a general-purpose button with no well-defined meaning. However, an action can be associated with the

button by using an event handler attribute, such as **onclick**. A value of **checkbox** indicates a check box control. Check box form controls have a checked and unchecked setting, but even if these controls are grouped together, they allow a user to select multiple check boxes at once. In contrast, a value of **radio** indicates a radio button control. When grouped, radio buttons allow only one of the many choices to be selected at a given time.

A form control type of **hidden** indicates a field that is not visible to the viewer but is used to store information. A hidden form control often is used to preserve state information between pages.

A value of **file** for the **type** attribute indicates a control that allows the viewer to upload a file to a server. The filename can be entered in a displayed field, or a user agent might provide a special browse button allowing the user to locate the file. A value of **image** indicates a graphic image form control that a user can click on to invoke an associated action. (Most browsers allow the use of **img**-associated attributes such as **height**, **width**, **hspace**, **vspace**, and **alt** when the **type** value is set to **image**.) A value of **password** for the **type** attribute indicates a password entry field. A password field will not display text entered as it is typed; it might instead show a series of dots. Note that password-entered data is not transferred to the server in any secure fashion. A value of **reset** for the **type** attribute is used to insert a button that resets all controls within a form to their default values. A value of **submit** inserts a special submission button that, when clicked, sends the contents of the form to the location indicated by the **action** attribute of the enclosing **<form>** tag. Lastly, a value of **text** (the default) for the **type** attribute indicates a single-line text input field.

**usemap** This HTML 4.0 attribute is used to indicate the map file to be associated with an image when the form control is set with **type="image"**. The value of the attribute should be a URL of a map file, but generally will be in the form of a URL fragment referencing a map file within the current file.

**value** This attribute has two different uses, depending on the value for the **type** attribute. With data-entry controls (**type="text"** and **type="password"**), this attribute is used to set the default value for the control. When used with check box or radio form controls, this attribute specifies the return value for the control. If not set for these fields, a default value of **on** will be submitted when the control is activated.

**vspace** This Internet Explorer-specific attribute indicates the vertical space in pixels between the image and surrounding text when using **<input type="image">**.

**width** This attribute, supported by Internet Explorer, is used to set the size of the form control in pixels.

### Examples

```
<form action="#" method="get">
```

```
Enter your name: <input type="text" maxlength="35" size="20" /><br />
```

```
Enter your password: <input type="password" maxlength="35" size="20" />
```

```
<br /><br />
```

Which is your favorite food?

```
<input type="radio" name="favorite" value="Mexican" />Mexican
```

```
<input type="radio" name="favorite" value="Russian" />Russian
```

```
<input type="radio" name="favorite" value="Japanese" />Japanese
```

```
<input type="radio" checked name="favorite" value="Other" />Other
```

```
<br /><br />
```

```
<input type="submit" value="Submit" />
```

```
<input type="reset" value="Reset" />
</form>
```

#### Notes

- Some documents suggest the use of **type="textarea"**. Even if this strange form is supported, it should be avoided in favor of the `<textarea>` tag, which is common to all browsers.
- The HTML 2.0 and 3.2 specifications support only the **align**, **checked**, **maxlength**, **name**, **size**, **src**, **type**, and **value** attributes for the **input** element.
- The HTML 4.01 specification also reserves the use of the **datafld**, **dataformatas**, and **datasrc** data-binding attributes. They were not included in the XHTML specification but are supported by Internet Explorer.
- Use of **autocomplete** may have security implications. Use with caution.
- Under the strict HTML and XHTML specifications, the **align** attribute is not allowed.
- As an empty element under XHTML, `<input />` requires the trailing slash.

## <ins> (Inserted Text)

This element is used to indicate that text has been added to the document.

### Standard Syntax

```
<ins
  cite="URL"
  class="class name(s)"
  datetime="date"
  dir="ltr | rtl"
  id="unique alphanumeric identifier"
  lang="language code"
  style="style information"
  title="advisory text">
```

```
</ins>
```

### Element-Specific Attributes

**cite** The value of this attribute is a URL that designates a source document or message for the information inserted. This attribute is intended to point to information explaining why the text was changed.

**datetime** This attribute is used to indicate the date and time the insertion was made. The value of the attribute is a date in a special format as defined by ISO 8601. The basic date format is

```
yyyy-mm-ddthh:mm:ssTZD
```

where the following is true:

yyyy=four-digit year such as 1999

mm=two-digit month (01=January, 02=February, and so on)

dd=two-digit day of the month (01 through 31)

hh=two-digit hour (00 to 23) (24-hour clock not AM or PM)

mm=two-digit minute (00 to 59)

ss=two-digit second (00 to 59)

tzd=time zone designator

The time zone designator is either **z**, which indicates Universal Time Coordinate, or coordinated universal time format (UTC), or **+hh:mm**, which indicates that the time is a local time that is *hh* hours and *mm* minutes ahead of UTC. Alternatively, the format for the time zone designator could be **-hh:mm**, which indicates that the local time is behind UTC. Note that the letter "T" actually appears in the string, all digits must be used, and **00** values for minutes and seconds might be required. An example value for the **datetime** attribute might be **1999-10-6T09:15:00-05:00**, which corresponds to October 6, 1999, 9:15 A.M., U.S. Eastern Standard Time.

#### Example

```
<ins cite="http://www.bigcompany.com/changes/oct99.html"
      date="1999-10-06T09:15:00-05:00">
```

The penalty clause applies to client lateness as well.

```
</ins>
```

#### Notes

- Browsers can render inserted (**<ins>**) or deleted (**<del>**) text in a different style to show the changes that have been made to the document. Eventually, a browser could have a way to show a revision history on a document. User agents that do not understand **<del>** or **<ins>** will show the information anyway, so there is no harm in adding information, only in deleting it.

## <isindex> (Index Prompt)

This element indicates that a document has an associated searchable keyword index. When a browser encounters this element, it inserts a query entry field at that point in the document. The viewer can enter query terms to perform a search. This element is deprecated under the strict HTML and XHTML specifications and should not be used.

#### Standard Syntax (Transitional Only)

```
<isindex
  class="class name(s)"
  dir="ltr | rtl"
  href="url" (nonstandard but common)
  id="unique alphanumeric identifier"
  lang="language code"
  prompt="string"
  style="style information"
  title="advisory text" />
```

#### Element-Specific Attributes

**action** This attribute specifies the URL of the query action to be executed when the viewer presses the ENTER key. Although this attribute is not defined under any HTML specification, it is common to many browsers, particularly Internet Explorer 3, which defined it.

**prompt** This attribute allows a custom query prompt to be defined. The default prompt is "This is a searchable index. Enter search keywords."

#### Examples

```
<isindex action="cgi-bin/search" prompt="Enter search terms" />
```

```
<!-- very old HTML style syntax below -->
```

```
<base href='cgi-bin/search'>
<isindex prompt='Enter search terms' />
```

```
<isindex href='cgi-bin/search' prompt='Keywords:' />
```

### Notes

- Originally, the W3C intended this element to be used in a document's **head**. Browser vendors have relaxed this usage to allow the element in a document's **body**. Early implementations did not support the **action** attribute and used a **<base>** tag or an **href** attribute to specify a search function's URL.
- As an empty element, **<isindex>** requires no closing tag under HTML specifications. However, under the XHTML specification, a trailing slash **<isindex />** is required.
- The HTML 3.2 specification only allows the **prompt** attribute, whereas HTML 2 expected a text description to accompany the search field.
- Netscape 1.1 originated the use of the **prompt** attribute.

## <kbd> (Keyboard Input)

This element logically indicates text as keyboard input. A browser generally renders text enclosed by this element in a monospaced font.

### Standard Syntax

```
<kbd
  class="class name(s)"
  dir="ltr | rtl"
  id="unique alphanumeric identifier"
  lang="language code"
  style="style information"
  title="advisory text">
```

```
</kbd>
```

### Example

Enter the change directory command at the prompt as shown below:

```
<br /><br />
<kbd>CD .. </kbd>
```

## <label> (Form Control Label)

This element is used to relate descriptions to form controls.

### Standard Syntax

```
<label
  accesskey="key"
  class="class name(s)"
  dir="ltr | rtl"
  for="id of form field">
```

```
id="unique alphanumeric identifier"  
lang="language code"  
style="style information"  
title="advisory text">
```

```
</label>
```

### Element-Specific Attributes

**accesskey** This attribute specifies a keyboard navigation accelerator for the element. Pressing ALT or a similar key in association with the specified key selects the anchor element correlated with that key.

**datafld** This attribute is used to indicate the column name in the data source that is bound to the content of a **<label>** tag.

**dataformatas** This attribute indicates whether the bound data is plain text (**text**) or HTML (**html**). The data bound with **<label>** is used to set the content of the label.

**datasrc** The value of this attribute is an identifier indicating the data source to pull data from.

**for** This attribute specifies the **id** for the form control element the label references. This is optional when the label encloses the form control it is bound to. In many cases, particularly when a table is used to structure the form, a **<label>** tag will not be able to enclose the associated form control, so the **for** attribute should be used. This attribute allows more than one label to be associated with the same control by creating multiple references.

### Examples

```
<form action="#" method="get">  
  <label id="username" >Name  
  <input type="text" id="username" name="username" />  
  </label>  
</form>
```

```
<form>  
  <table>  
    <tr>  
      <td><label for="username">Name</label></td>  
      <td><input type="text" id="username" name="username"></td>  
    </tr>  
  </table>  
</form>
```

## <layer> (Positioned Layer)

This Netscape-specific element allows the definition of overlapping content layers that can be exactly positioned, hidden or shown, rendered transparent or opaque, reordered front to back, and nested. Most of the functionality of layers is available using CSS positioning facilities. Developers are strongly advised not to use the **layer** element.

### Syntax (Defined by Netscape 4 Only)

```
<layer  
  above="layer name"  
  background="URL of background image"
```

below="layer name"  
bgcolor="color value"  
class="class name(s)"  
clip="clip region coordinates in x1, y1, x2, y2 form"  
height="percentage | pixels"  
id="unique alphanumeric identifier"  
left="pixels"  
name="string"  
overflow="none | clip"  
pagex="horizontal pixel position of layer"  
pagey="vertical pixel position of layer"  
src="url of layer's contents"  
style="style information"  
title="advisory text"  
top="pixels"  
visibility="hide | inherit | show"  
width="percentage | pixels"  
z-index="number">

</layer>

### Element-Specific Attributes

**above** This attribute contains the name of the layer (as set with the **name** attribute) to be rendered directly above the current layer.

**background** This attribute contains the URL of a background pattern for the layer. Like backgrounds for the document as a whole, the image might tile.

**below** This attribute specifies the name of the layer to be rendered below the current layer.

**bgcolor** This attribute specifies a layer's background color. The attribute's value can be either a named color, such as **red**, or a color specified in the hexadecimal **#RRGGBB** format, such as **#FF0000**.

**clip** This attribute clips a layer's content to a specified rectangle. All layer content outside that rectangle will be rendered transparent. The **clip** rectangle is defined by two *x,y* pairs that correspond to the top *x*, left *y*, bottom *x*, and right *y* coordinate of the rectangle. The coordinates are relative to the layer's origin point, **0,0**, in its top-left corner, and might have nothing to do with the pixel coordinates of the screen.

**height** This attribute is used to set the height of the layer, either in pixels or as a percentage of the screen or region the layer is contained within.

**left** This attribute specifies, in pixels, the left offset of the layer. The offset is relative to its parent layer, if it has one, or to the left browser margin if it does not.

**name** This attribute assigns the layer a name that can be referenced by programs in a client-side scripting language. The **id** attribute also can be used.

**overflow** This attribute specifies what should happen when the layer's content exceeds its rendering box and clipping area. A value of **none** does not clip the content, while **clip** clips the content to its dimensions or defined clipping area.

**pagex** This attribute is used to set the horizontal pixel position of the layer relative to the document window rather than any enclosing layer.

**pagey** This attribute is used to set the vertical pixel position of the layer relative to the document window rather than any enclosing layer.

**src** This attribute specifies the URL that contains the content to be included in the layer. Using this attribute with an empty element is a good way to preserve layouts under older browsers.

**top** This attribute specifies, in pixels, the top offset of the layer. The offset is relative to its parent layer if it has one, or to the top browser margin if it is not enclosed in another layer.

**visibility** This attribute specifies whether a layer is hidden (**hidden**), shown (**show**), or inherits (**inherits**) its visibility from the layer enclosing it.

**width** This attribute specifies a layer's width in pixels or as a percentage value of the enclosing layer or browser width.

**z-index** This attribute specifies a layer's stacking order relative to other layers. Position is specified with positive integers, with **1** indicating the bottommost layer.

### Examples

```
<layer name="scene" bgcolor="#00FFFF>
  <layer name="Shaq" left="100" top="100">
    
  </layer>
  <layer name="Rodman" left="200" top="100"
    visibility="hidden">
    
  </layer>
</layer>
```

```
<!-- The better way to do layers -->
<layer src="contents.html" left="20" top="20"
  height="80%" width="80%">
</layer>
```

### Notes

- Because this element is specific to Netscape 4, it should never be used and is discussed only for readers supporting existing **<layer>**-filled pages.
- Applets, plug-ins, and other embedded media forms, generically called *objects*, can be included in a layer; however, they will float to the top of all other layers, even if their containing layer is obscured.

## <legend> (Field Legend)

Used to assign a caption to a set of form fields as defined by a **fieldset** element.

### Standard Syntax

```
<legend
  accesskey="character"
  align="bottom | left | right | top" (transitional only)
  class="class name(s)"
  dir="ltr | rtl"
  id="unique alphanumeric identifier"
  lang="language code"
  style="style information"
```

```
title="advisory text">
```

```
</legend>
```

### Element-Specific Attributes

**accesskey** This attribute specifies a keyboard navigation accelerator for the element. Pressing ALT or a similar key in association with the specified key selects the form section or the legend itself.

**align** This attribute indicates where the legend value should be positioned within the border created by a **<fieldset>** tag. The default position for the legend is the upper-left corner. It also is possible to position the legend to the right by setting the attribute to **right**. The specification defines **bottom** and **top**, as well. Microsoft also defines the use of the value **center**.

### Example

```
<form action="#" method="get">
<fieldset>
  <legend align="top">User Information</legend>
  <label>First Name:
  <input type="text" id="firstname" name="firstname" size="20" />
</label><br />
  <label>Last Name:
  <input type="text" id="lastname" name="lastname" size="20" />
</label><br />
</fieldset>
</form>
```

### Notes

- A **<legend>** tag should occur only within a **<fieldset>** tag. There should be only one **legend** per **fieldset** element.
- The **<legend>** tag improves accessibility when the **fieldset** is not rendered visually.
- Some versions of Microsoft documentation show a **valign** attribute for **<legend>** positioning. However, the **valign** attribute does not appear to work consistently and has since been dropped from the official documentation.

## <li> (List Item)

This element is used to indicate a list item as contained in an ordered list (**<ol>**), unordered list (**<ul>**), or older list styles such as **<dir>** and **<menu>**.

### Standard Syntax

```
<li
  class="class name(s)"
  dir="ltr | rtl"
  id="unique alphanumeric identifier"
  lang="language code"
  style="style information"
  title="advisory text"
  type="circle | disc | square | a | A | i | I | 1"
  value="number"> (value attribute transitional only)
```

</li>

### Element-Specific Attributes

**type** This attribute indicates the bullet type used in unordered lists or the numbering type used in ordered lists. For ordered lists, a value of **a** indicates lowercase letters, **A** indicates uppercase letters, **i** indicates lowercase Roman numerals, **I** indicates uppercase Roman numerals, and **1** indicates numbers. For unordered lists, values are used to specify bullet types. Although the browser is free to set bullet styles, a value of **disc** generally specifies a filled circle, a value of **circle** specifies an empty circle, and a value of **square** specifies a filled square.

**value** This attribute indicates the current number of items in an ordered list as defined by an <ol> tag. Regardless of the value of **type** being used to set Roman numerals or letters, the only allowed value for this attribute is a number. List items that follow will continue numbering from the value set. The **value** attribute has no meaning for unordered lists.

### Examples

<ul>

<li type="circle">First list item is a circle</li>

<li type="square">Second list item is a square</li>

<li type="disc">Third list item is a disc</li>

</ul>

<ol>

<li type="i">Roman Numerals</li>

<li type="a" value="3">Second list item is letter C</li>

<li type="a">Continue list in lowercase letters</li>

</ol>

### Notes

- Under the strict HTML and XHTML definitions, the **li** element loses the **type** and **value** attributes, as these presentation styles can be emulated with style sheets.
- Whereas bullet styles can be set explicitly, browsers tend to change styles for bullets when <ul> lists are nested. However, ordered lists generally do not change style automatically, nor do they support outline-style numbering (1.1, 1.1.1, and so on).
- The closing tag </li> is optional under HTML specifications and is not commonly used. However, it is required under XHTML and should always be used.

## <link> (Link to External Files or Set Relationships)

This empty element specifies relationships between the current document and other documents. Possible uses for this element include defining a relational framework for navigation and linking the document to a style sheet.

### Standard Syntax

<link

charset="charset list from RFC 2045"

class="class name(s)"

dir="ltr | rtl"

href="URL"

hreflang="language code"

id="unique alphanumeric identifier"  
lang="language code"  
media="all | aural | braille | print | projection |  
screen | other"  
rel="relationship value"  
rev="relationship value"  
style="style information"  
target="frame name" (transitional only)  
title="advisory information"  
type="content type" />

### Common Attributes

disabled="disabled " (from DOM Level 1)

### Element-Specific Attributes

**charset** This attribute specifies the character set used by the linked document. Allowed values for this attribute are character set names, such as EUC-JP, as defined in RFC 2045.

**disabled** This DOM Level 1 defined attribute is used to disable a link relationship. The presence of the attribute is all that is required to remove a linking relationship. In conjunction with scripting, this attribute could be used to turn on and off various style sheet relationships. It appears to be an oversight in the HTML and XHTML specifications.

**href** This attribute specifies the URL of the linked resource. A URL might be absolute or relative.

**hreflang** This attribute is used to indicate the language of the linked resource.

**media** This attribute specifies the destination medium for any linked style information, as indicated when the **rel** attribute is set to **stylesheet**. The value of the attribute might be a single media descriptor, such as **screen**, or a comma-separated list. Possible values for this attribute include **all**, **aural**, **braille**, **print**, **projection**, and **screen**. Other values also might be defined, depending on the browser. Internet Explorer supports **all**, **print**, and **screen** as values for this attribute.

**rel** This attribute names a relationship between the linked document and the current document. Possible values for this attribute include **alternate**, **bookmark**, **chapter**, **contents**, **copyright**, **glossary**, **help**, **index**, **next**, **prev**, **section**, **start**, **stylesheet**, and **subsection**.

The most common use of this attribute is to specify a link to an external style sheet. The **rel** attribute is set to **stylesheet**, and the **href** attribute is set to the URL of an external style sheet to format the page.

**rev** The value of the **rev** attribute shows the relationship of the current document to the linked document, as defined by the **href** attribute. The attribute thus defines the reverse relationship compared to the value of the **rel** attribute. Values for the **rev** attribute are similar to the possible values for **rel**. They might include **alternate**, **bookmark**, **chapter**, **contents**, **copyright**, **glossary**, **help**, **index**, **next**, **prev**, **section**, **start**, **stylesheet**, and **subsection**.

**target** The value of the **target** attribute defines the frame or window name that has the defined linking relationship or that will show the rendering of any linked resource.

**type** This attribute is used to define the type of the content linked to. The value of the attribute should be a MIME type, such as **text/html**, **text/css**, and so on. The common use of this attribute is to define the type of style sheet linked, and the most common current value is **text/css**, which indicates a cascading style sheet format.

### Examples

```
<link href="products.html" rel="parent" />
```

```
<link href="corpstyle.css" rel="stylesheet" type="text/css" media="all" />
```

```
<link href="nextpagetoload.html" rel="next" />
```

#### Notes

- Under XHTML 1.0, empty elements such as **<link>** require a trailing slash: **<link />**.
- A **<link>** tag can occur only in the **head** element; however, there can be multiple occurrences of **<link>**.
- HTML 3.2 defines only the **href**, **rel**, **rev**, and **title** attributes for the **link** element.
- HTML 2 defines the **href**, **methods**, **rel**, **rev**, **title**, and **urn** attributes for the **link** element. The **methods** and **urn** attributes were later removed from the specifications.
- The HTML and XHTML specifications define event handlers for the **link** element, but it is unclear how they would be used.

## <listing> (Code Listing)

This deprecated element from HTML 2 is used to indicate a code listing; it is no longer part of the HTML standard. Text tends to be rendered in a smaller size within this element. The **pre** element should be used instead of **listing** to indicate preformatted text.

#### Standard Syntax (HTML 2 Only; Deprecated)

```
<listing>  
</listing>
```

#### Example

```
<listing>
```

This is a code listing. The preformatted text element `&lt;PRE&gt;` should be used instead of this deprecated element.

```
</listing>
```

#### Notes

- As a deprecated element, this element should not be used. This element is not supported by HTML 4, XHTML 1.0, or 1.1. It is still documented and supported by many browser vendors, however, and does creep into some pages. The **pre** element should be used instead of **<listing>**.
- It appears that Internet Explorer browsers also make text within **<listing>** one size smaller than normal text, probably because the HTML 2 specification suggested that 132 characters fit to a typical line rather than 80.

## <map> (Client-Side Image Map)

This element is used to implement client-side image maps. The element is used to define a map that associates locations on an image with a destination URL. Each hot spot or hyperlink mapping is defined by an enclosed **area** element. A map is bound to a particular image through the use of the **usemap** attribute in the **img** element, which is set to the name of the map.

#### Syntax

```
<map
```

```
class="class name(s)"
dir="ltr | rtl"
id="unique alphanumeric identifier"
lang="language code"
name="unique alphanumeric identifier"
style="style information"
title="advisory text">
```

</map>

### Element-Specific Attributes

**name** Like **id**, this attribute is used to define a name associated with the element. In the case of the **map** element, the **name** attribute is the common way to define the name of the image map to be referenced by the **usemap** attribute within an **<img>** tag.

### Example

```
<map name="mainmap" id="mainmap">
  <area shape="circle" coords="200,250,25"
    href="file1.html" />
  <area shape="rectangle" coords="50,50,100,100"
    href="file2.html#important" />
  <area shape="default" nohref="nohref" />
</map>
```

### Notes

- HTML 3.2 supports only the **name** attribute for the **map** element.
- Client-side image maps are not supported under HTML 2. They were first suggested by Spyglass and later incorporated in Netscape and other browsers.

## <marquee> (Marquee Display)

This proprietary element specifies a scrolling, sliding, or bouncing text marquee.

### Proprietary Syntax (Defined by Internet Explorer)

```
<marquee
  accesskey="key" (5.5)
  behavior="alternate | scroll | slide" (3)
  bgcolor="color name | #RRGGBB" (3)
  class="class name(s)" (4)
  contenteditable=" false | true | inherit" (5.5)
  datafld="column name" (4)
  dataformatas="html | text" (4)
  datasrc="data source id" (4)
  direction="down | left | right | up" (3)
  dir="ltr | rtl" (5.0)
  disabled="false | true" (5.5)
  height="pixels or percentage"
  hidefocus="true | false" (5.5)
  hspace="pixels" (3)
```

id="unique alphanumeric identifier" (4)  
lang="language code" (4)  
language="javascript | jscript | vbs | vbscript" (4)  
loop="infinite | number" (3)  
scrollamount="pixels" (3)  
scrolldelay="milliseconds" (3)  
style="style information" (4)  
tabindex="number" (5.5)  
title="advisory text" (4)  
truespeed="false | true" (4)  
unselectable="on | off" (5.5)  
vspace="pixels" (3)  
width="pixels or percentage" (3)

>

</marquee>

### Element-Specific Attributes

**behavior** This attribute controls the movement of marquee text across the marquee. The **alternate** option causes text to completely cross the marquee field in one direction and then cross in the opposite direction. A value of **scroll** for the attribute causes text to wrap around and start over again. This is the default value for a marquee. A value of **slide** for this attribute causes text to cross the marquee field and stop when its leading character reaches the opposite side.

**bgcolor** This attribute specifies the marquee's background color. The value for the attribute can either be a color name or a color value defined in the hexadecimal **#RRGGBB** format.

**datafld** This attribute is used to indicate the column name in the data source that is bound to the **marquee** element.

**dataformatas** This attribute indicates whether the bound data is plain text (**text**) or HTML (**html**). The data bound with **marquee** is used to set the message that is scrolled.

**datasrc** The value of this attribute is set to an identifier indicating the data source from which data is to be pulled. Bound data is used to set the message that is scrolled in the **marquee**.

**direction** This attribute specifies the direction in which the marquee should scroll. The default is **left**. Other possible values for **direction** include **down**, **right**, and **up**.

**height** This attribute specifies the height of the marquee in pixels or as a percentage of the window.

**hspace** This attribute indicates the horizontal space in pixels between the marquee and surrounding content.

**loop** This attribute indicates the number of times the marquee content should loop. By default, a marquee loops infinitely unless the **behavior** attribute is set to **slide**. It also is possible to use a value of **infinite** or **-1** to set the text to loop indefinitely.

**scrollamount** This attribute specifies the width in pixels between successive displays of the scrolling text in the marquee.

**scrolldelay** This attribute specifies the delay in milliseconds between successive displays of the text in the marquee.

**truespeed** When this attribute is present, it indicates that the **scrolldelay** value should be honored for its exact value. If the attribute is not present, any values less than 60 are rounded up to 60 milliseconds.

**vspace** This attribute indicates the vertical space in pixels between the marquee and surrounding content.

**width** This attribute specifies the width of the marquee in pixels or as a percentage of the enclosing window.

### Examples

```
<marquee behavior="alternate">
SPECIAL VALUE !!! This week only !!!
</marquee>
```

```
<marquee id="marquee1" bgcolor="red" direction="right" height="30"
width="80%" hspace="10" vspace="10">
```

The super scroller scrolls again!!

More fun than a barrel of **&lt;BLINK&gt;** elements.

```
</marquee>
```

### Notes

- This is primarily a Microsoft-specific element, although a few other browsers, notably MSN TV and later Netscape and Mozilla versions, support it to some degree.

## <menu> (Menu List)

This element is used to indicate a short list of items that can occur in a menu of choices.

### Syntax (Transitional Only)

```
<menu
class="class name(s)"
compact="compact"
dir="ltr | rtl"
id="unique alphanumeric string"
lang="language code"
style="style information"
title="advisory text">
</menu>
```

### Element-Specific Attributes

**compact** This attribute indicates that the list should be rendered in a compact style. Few browsers actually change the rendering of the list regardless of the presence of this attribute. The **compact** attribute requires no value under traditional HTML but should be set to a value of **compact** under XHTML transitional.

### Example

```
<h2>Taco List</h2>
<menu>
<li>Fish</li>
<li>Pork</li>
<li>Beef</li>
<li>Chicken</li>
</menu>
```

### Notes

- Under the strict HTML and XHTML specifications, this element is not defined. Because most browsers simply render this style of list as an unordered list, using the **<ul>** tag instead is preferable.
- The HTML 2.0 and 3.2 specifications support only the **compact** attribute.
- Most browsers tend not to support the **compact** attribute.

## <meta> (Meta-Information)

This element specifies general information about a document that can be used in document indexing. It also allows a document to simulate HTTP response headers which are useful for cache control, page ratings, page refresh time, and other useful things.

### Standard Syntax

```
<meta
  content="string"
  dir="ltr | rtl"
  http-equiv="http header string"
  id="unique alphanumeric string"
  lang="language code"
  name="name of meta-information"
  scheme="scheme type" />
```

### Element-Specific Attributes

**content** This attribute contains the actual meta-information. The form of the meta-information varies greatly, depending on the value set for **name**.

**http-equiv** This attribute binds the meta-information in the **content** attribute to an HTTP response header. If this attribute is present, the **name** attribute should not be used.

**lang** This attribute is the language code associated with the language used in the **content** attribute.

**name** This attribute associates a name with the meta-information contained in the **content** attribute. If present, the **http-equiv** attribute should not be used.

**scheme** The scheme attribute is used to indicate the expected format of the value of the **content** attribute. The particular scheme also can be used in conjunction with the metadata profile, as indicated by the **profile** attribute for the **head** element.

### Examples

```
<!-- Use of the meta element to assist document indexing -->
```

```
<meta name="keywords" content="html, scripting"
  scheme="Lycos" />
```

```
<!-- Use of the meta element to implement client-pull to automatically
  load a page -->
```

```
<meta http-equiv="refresh"
  content="3;URL='http://www.pint.com/'" />
```

```
<!-- Use of the META element to add rating information -->
```

```
<meta http-equiv="PICS-Label" content="(PICS-1.1
  'http://www.rsac.org/ratingsv01.html'
  1 gen true comment 'RSACi North America
```

```
Server' by 'webmaster@bigcompany.com'  
for 'http://www.bigcompany.com' on  
'1999.05.26T13:05-0500'  
r (n 0 s 0 v 0 1 1))" />
```

## Notes

- The **meta** element can occur only in the **head** element. It can be defined multiple times.
- The **meta** element is an empty element (as defined in the HTML specifications) and does not have a closing tag nor contain any content. However, under XHTML 1.0, empty elements such as `<meta>` require a trailing slash: `<meta />`.
- A common use of the **meta** element is to set information for indexing tools, such as search engines. Common values for the **name** attribute when performing this function include **author**, **description**, and **keywords**; other attributes also might be possible.
- The **http-equiv** attribute is often used to create a document that automatically loads another document after a set time. This is called *client-pull*. An example of a client-pull **meta** element is `<meta http-equiv="refresh" content="10;URL='nextpage.html'" />`. Note that the **content** attribute contains two values. The first is the number of seconds to wait, and the second is the identifier URL and the URL to load after the specified time.
- The **http-equiv** attribute is also used for page ratings, cache control, setting defaults such as language or scripting, and a variety of other tasks. In many cases, it would be better to set these values via the actual HTTP headers rather than via a `<meta>` tag.
- The HTML 2.0 and 3.2 specifications define only the **content**, **http-equiv**, and **name** attributes.

## <multicol> (Multiple Column Text)

This Netscape-specific element renders the enclosed content in multiple columns. This element should not be used; a table is a more standard way to render multiple columns of text across browsers.

### Proprietary Syntax (Defined by Netscape)

```
<multicol  
  class="class name(s)"  
  cols="number of columns"  
  gutter="pixels"  
  id="unique alphanumeric identifier"  
  style="style information"  
  width="pixels">
```

```
</multicol>
```

### Element-Specific Attributes

**cols** This attribute indicates the number of columns in which to display the text. The browser attempts to fill the columns evenly.

**gutter** This attribute indicates the width in pixels between the columns. The default value for this attribute is **10** pixels.

**width** This attribute indicates the column width for all columns. The width of each column is set in pixels and is equivalent for all columns in the group. If the attribute is not specified, the width of columns will be determined by taking the available window size, subtracting the

number of pixels for the gutter between the columns (as specified by the **gutter** attribute), and evenly dividing the result by the number of columns in the group (as set by the **cols** attribute).

#### Example

```
<multicol cols="3" gutter="20">
```

Put a long piece of text here....

```
</multicol>
```

#### Notes

- Page developers are strongly encouraged not to use this element. Netscape dropped this element for browser versions 6.0 and higher. Its inclusion in this book is for support of existing documents only.

## <noabr> (No Breaks)

This proprietary element renders enclosed text without line breaks. Break points for where text may wrap can be inserted using the **wbr** element.

#### Common Syntax

```
<noabr
```

```
  class="class name(s)"
```

```
  id="unique alphanumeric identifier"
```

```
  style="style information"
```

```
  title="advisory text">
```

```
</noabr>
```

#### Examples

```
<noabr>This really long text ... will not be broken.</noabr>
```

```
<noabr>With this element it is often important to hint where a line may  
be broken using &lt;wbr&gt;.<wbr> This element acts as a soft return.</noabr>
```

## <noembed> (No Embedded Media Support)

This Netscape-specific element is used to indicate alternative content to be displayed on browsers that cannot support an embedded media object. It should occur in conjunction with the **embed** element.

#### Proprietary Syntax (Defined by Netscape)

```
<noembed>
```

```
  Alternative content for non-embed supporting browsers
```

```
</noembed>
```

#### Element-Specific Attributes

Netscape does not specifically define attributes for this element; however, Netscape documentation suggests that **class**, **id**, **style**, and **title** might be supported for this element.

#### Example

```
<embed src="trailer.mov" height="150" width="150">
```

```
<noembed>
```

```
  
```

```
  <br />
```

Sorry, this browser is not configured to display video.

```
</noembed>
```

```
</embed>
```

### Notes

- This element will disappear as the **<object>** style of inserting media into a page becomes more common.
- Even if other browsers do not support the tag and render its contents, it works in the manner it was designed.

## <noframes> (No Frame Support Content)

This element is used to indicate alternative content to be displayed on browsers that do not support frames.

### Standard Syntax

```
<noframes  
  class="class name(s)"  
  dir="ltr | rtl"  
  id="unique alphanumeric identifier"  
  lang="language code"  
  style="style information"  
  title="advisory text">
```

*Alternative content for non-frame-supporting browsers*

```
</noframes>
```

### Example

```
<frameset rows="100,*">  
  <frame src="controls.html"> name="frame1" id="frame1"  
  <frame src="content.html"> "name=frame2" id="frame2"  
  <noframes>  
    Sorry, this browser does not support frames.  
  </noframes>  
</frameset>
```

### Notes

- This element should be used within the scope of the **frameset** element.
- The benefit of events and sophisticated attributes, such as **style**, is unclear for browsers that would use content within **<noframes>**, given that older browsers that don't support frames probably would not support these features.

## <noscript> (No Script Support Content)

This element is used to enclose content that should be rendered on browsers that do not support scripting or that have scripting turned off.

### Syntax

```
<noscript  
  class="class name(s)"
```

```
dir="ltr | rtl"
id="unique alphanumeric identifier"
lang="language code"
style="style information"
title="advisory text">
```

*Alternative content for non-script-supporting browsers*

```
</noscript>
```

### Example

```
<script language="type/javascript">
```

```
<!--
```

```
window.location="http://www.pint.com";
```

```
// -->
```

```
</script>
```

```
<noscript>
```

JavaScript is not supported. Follow this

[link](http://www.pint.com) instead.

```
</noscript>
```

### Notes

- Improved functionality for the **noscript** element might come if it is extended to deal with the lack of support for one scripting language or another. Currently, the element is used only to indicate whether any scripting is supported or not.
- Oddly **noscript** is not allowed in the head even though **script** is.

## <object> (Embedded Object)

This element specifies an arbitrary object to be included in an HTML document. Initially, this element was used to insert ActiveX controls, but according to the specification, an object can be any media object, document, applet, ActiveX control, or even image.

### Standard Syntax

```
<object
```

```
align="bottom | left | middle | right | top" (transitional only)
```

```
archive="url"
```

```
border="percentage | pixels" (transitional only)
```

```
class="class name(s)"
```

```
classid="id"
```

```
codebase="URL"
```

```
codetype="MIME Type"
```

```
data="URL of data"
```

```
declare="declare"
```

```
dir="ltr | rtl"
```

```
height="percentage | pixels"
```

```
hspace="percentage | pixels" (transitional only)
```

```
id="unique alphanumeric identifier"
```

```
lang="language code"
```

```
name="unique alphanumeric name"
standby="standby text string"
style="style information"
tabindex="number"
title="advisory text"
type="MIME Type"
usemap="URL"
vspace="percentage | pixels" (transitional only)
width="percentage | pixels">
```

*param elements and alternative rendering*

</object>

### Element-Specific Attributes

**align** This attribute aligns the object with respect to the surrounding text. The default is **left**. The HTML specification defines **bottom**, **middle**, **right**, and **top**, as well. Browsers might provide an even richer set of alignment values. The behavior of alignment for objects is similar to images. Under the strict HTML and XHTML specifications, the **object** element does not support this attribute.

**archive** This attribute contains a URL for the location of an archive file. An archive file typically is used to contain multiple object files to improve the efficiency of access.

**border** This attribute specifies the width of the object's borders in pixels or as a percentage.

**classid** This attribute contains a URL for an object's implementation. The URL syntax depends upon the object's type. With ActiveX controls, the value of this attribute does not appear to be a URL but something of the form *CLSID: object-id*; for example, **CLSID: 99B42120-6EC7-11CF-A6C7-00AA00A47DD2**.

**code** Under the old Microsoft implementation, this attribute contains the URL referencing a Java applet class file. The way to access a Java applet under the HTML/XHTML specification is to use **<object classid="java: classname.class">**. The pseudo-URL *java:* is used to indicate a Java applet. Microsoft Internet Explorer 4 and beyond support this style, so **code** should not be used.

**codebase** This attribute contains a URL to use as a relative base to access the object specified by the **classid** attribute.

**codetype** This attribute specifies an object's MIME type. Do not confuse this attribute with **type**, which specifies the MIME type of the data the object may use, as defined by the **data** attribute.

**data** This attribute contains a URL for data required by an object.

**datafld** This Microsoft-specific attribute is used to indicate the column name in the data source that is bound to the **object** element.

**datasrc** The value of this Microsoft-specific attribute is set to an identifier indicating the data source to pull data from.

**declare** This attribute declares an object without instantiating it. This is useful when the object will be a parameter to another object. In traditional HTML, this attribute takes no value; under XHTML, set it equal to **declare**.

**height** This attribute specifies the height of the object in pixels or as a percentage of the enclosing window.

**hspace** This attribute indicates the horizontal space, in pixels or percentages, between the object and surrounding content.

**name** Under the Microsoft definition, this attribute defines the name of the control so scripting can access it. Older HTML specifications suggest that it is a name for form submission, but this meaning is unclear and not supported by browsers.

**standby** This attribute contains a text message to be displayed while the object is loading.

**tabindex** This attribute takes a numeric value indicating the position of the object in the tabbing index for the document. Tabbing proceeds from the lowest positive **tabindex** value to the highest. Negative values for **tabindex** will leave the object out of the tabbing order. When tabbing is not explicitly set, the browser can tab through items in the order they are encountered.

**type** This attribute specifies the MIME type for the object's data. This is different from the **codetype**, which is the MIME type of the object and not of the data it uses.

**usemap** This attribute contains the URL of the image map to be used with the object. Typically, the URL will be a fragment identifier referencing a **map** element somewhere else within the file. The presence of this attribute indicates that the type of object being included is an image.

**vspace** This attribute indicates the vertical space, in pixels or percentages, between the object and surrounding text.

**width** This attribute specifies the width of the object in pixels or as a percentage of the enclosing window or block element.

### Examples

```
<object id="IeLabel1" width="325" height="65"
  classid="CLSID:99B42120-6EC7-11CF-A6C7-00AA00A47DD2">
  <param name="_ExtentX" value="6879" />
  <param name="_ExtentY" value="1376" />
  <param name="Caption" value="Hello World" />
  <param name="Alignment" value="4" />
  <param name="Mode" value="1" />
  <param name="ForeColor" value="#FF0000" />
  <param name="FontName" value="Arial" />
  <param name="FontSize" value="36" />
  <b>Hello World for non-ActiveX users!</b>
</object>
```

```
<object classid="java:Blink.class"
  standby="Here it comes"
  height="100" width="300">
  <param name="lbl" value="Java is fun, exciting, and new." />
  <param name="speed" value="2" />
```

This will display in non-Java-aware or -enabled browsers.

```
</object>
```

```
<object data="pullinthisfile.html">
```

Data not included!

```
</object>
```

Notes

- Under the strict HTML and XHTML specifications, the **object** element loses most of its presentation attributes, including **align**, **border**, **height**, **hspace**, **vspace**, and **width**. These attributes are replaced by style sheet rules.
- The HTML 4.01 specification reserves the **datafld**, **dataformatas**, and **datasrc** attributes for future use. However, these attributes were dropped in XHTML, though they are well supported by Internet Explorer 4 and beyond.
- Alternative content should be defined within an **<object>** tag after any enclosed **<param>** tags.
- The **object** element is still mainly used to include multimedia binaries in pages. Although the specification defines that it can load in HTML files and create image maps, not every browser supports this, and few developers are aware of these features. In theory, this very versatile tag should take over duties from the venerable **<img>** tag in future XHTML specifications.

## **<ol> (Ordered List)**

This element is used to define an ordered or numbered list of items. The numbering style comes in many forms, including letters, Roman numerals, and regular numerals. The individual items within the list are specified by **li** elements included with the **ol** element.

### **Standard Syntax**

```
<ol
  class="class name(s)"
  compact="compact" (transitional only)
  dir="ltr | rtl"
  id="unique alphanumeric identifier"
  lang="language code"
  start="number" (transitional only)
  style="style information"
  title="advisory text"
  type="a | A | i | I | 1"> (type attribute transitional only)
```

*li elements only*

```
</ol>
```

### **Element-Specific Attributes**

**compact** This attribute indicates that the list should be rendered in a compact style. Few browsers actually change the rendering of the list regardless of the presence of this attribute. The **compact** attribute requires no value under traditional HTML, but under XHTML should be set to **compact**.

**start** This attribute specifies the start value for numbering the individual list items. Although the ordering type of list elements might be Roman numerals, such as XXXI, or letters, the value of **start** is always represented as a number. To start numbering elements from the letter "C," use **<ol type="A" start="3">**.

**type** This attribute indicates the numbering type: **a** indicates lowercase letters, **A** indicates uppercase letters, **i** indicates lowercase Roman numerals, **I** indicates uppercase Roman numerals,

and **1** indicates numbers. Type set in an **ol** element is used for the entire list unless a **type** attribute is used within an enclosed **li** element.

### Examples

```
<ol type="1">
  <li>First step</li>
  <li>Second step</li>
  <li>Third step</li>
</ol>
<ol compact="compact" type="I" start="30">
  <li>Clause 30</li>
  <li>Clause 31</li>
  <li>Clause 32</li>
</ol>
```

### Notes

- Under the strict HTML and XHTML specifications, the **ol** element no longer supports the **compact**, **start**, and **type** attributes. These aspects of lists can be controlled with style sheet rules.
- Under the XHTML 1.0 specification, the **compact** attribute no longer can be minimized, but must have a quoted attribute value: `<ol compact="compact">`.
- The HTML 3.2 specification supports only the **compact**, **start**, and **type** attributes.
- The HTML 2.0 specification supports only the **compact** attribute.

## <optgroup> (Option Grouping)

This element specifies a grouping of items in a selection list defined by **option** elements so that the menu choices can be presented in a hierarchical menu or similar alternative fashion to improve access through nonvisual browsers.

### Standard Syntax

```
<optgroup
  class="class name(s)"
  dir="ltr | rtl"
  disabled="disabled"
  id="unique alphanumeric identifier"
  label="text description"
  lang="language code"
  style="style information"
  title="advisory text">
```

*option elements*

```
</optgroup>
```

### Element-Specific Attributes

**disabled** Occurrence of this attribute indicates that the enclosed set of options is disabled.

**label** This attribute contains a short label that might be more appealing to use when the selection list is rendered as items in a hierarchy.

## Example

Where would you like to go for your vacation?

```
<select name="vacation" id="vacation">
  <option id="ch1" value="Hong Kong">Hong Kong</option>
  <optgroup label="South Pacific">
    <option id="ch2" label="Australia" value="Australia">
      Australia</option>
    <option id="ch3" label="Fiji" value="Fiji">
      Wakaya (Fiji Islands)</option>
    <option id="ch4" value="New Zealand">
      New Zealand</option>
  </optgroup>
  <option id="ch5" value="home" selected="selected">Your backyard</option>
</select>
```

## Notes

- This element should occur only within the context of a **select** element.
- Only the most modern browsers present this element in a visually meaningful fashion.

## <option> (Option in Selection List)

This element specifies an item in a selection list defined by the **select** element.

### Standard Syntax

```
<option
  class="class name(s)"
  dir="ltr | rtl"
  disabled="disabled"
  id="unique alphanumeric identifier"
  label="text description"
  lang="language code"
  selected="selected"
  style="style information"
  title="advisory text"
  value="option value">
```

```
</option>
```

### Element-Specific Attributes

**disabled** Presence of this attribute indicates that the particular item is not selectable. Traditional HTML did not require a value for this attribute, but it should be set to **disabled** under XHTML.

**label** This attribute contains a short label that might be more appealing to use when the selection list is rendered as a hierarchy due to the presence of an **optgroup** element.

**selected** This attribute indicates that the associated item is the default selection. If not included, the first item in the selection list is the default. If the **select** element enclosing the **option** elements has the **multiple** attribute, the **selected** attribute might occur in multiple entries. Otherwise, it should occur in only one entry. Under XHTML, the value of the selected attribute must be set to **selected**.

**value** This attribute indicates the value to be included with the form result when the item is selected.

### Example

Which is your favorite dog?:

```
<select>
  <option value="Scotty">Angus</option>
  <option value="Golden Retriever">Borrego</option>
  <option value="Choco Lab">Dutch</option>
  <option value="Mini Schnauzer" selected="selected">Tucker</option>
</select>
```

### Notes

- Under HTML specifications, the closing tag for **<option>** is optional. However, for XHTML compatibility the closing tag **</option>** is required.
- This element should occur only within the context of a **select** element.
- The HTML 2.0 and 3.2 specifications define only the **selected** and **value** attributes for this element.

## <p> (Paragraph)

This element is used to define a paragraph of text.

### Standard Syntax

```
<p
  align="center | justify | left | right" (transitional only)
  class="class name(s)"
  dir="ltr | rtl"
  id="unique alphanumeric identifier"
  lang="language code"
  style="style information"
  title="advisory text">
```

```
</p>
```

### Element-Specific Attributes

**align** This attribute specifies the alignment of text within a paragraph. The default value is **left**. The transitional specification of HTML 4.01 also defines **center**, **justify**, and **right**. However, under the strict HTML and XHTML specifications, text alignment can be handled through a style sheet rule.

### Examples

```
<p align="right">A right-aligned paragraph</p>
```

```
<p id="para1" class="defaultParagraph"
  title="Introduction Paragraph">
```

This is the introductory paragraph for a very long paper about nothing.

```
</p>
```

### Notes

- Because **p** is a block element browsers typically insert a blank line, but this rendering should not be assumed, given the rise of style sheets.

- Under the strict HTML and XHTML specifications, the **align** attribute is not supported. Alignment of text can instead be accomplished using style sheets.
- The closing tag for the `<p>` tag is optional under the HTML specification; however, under the XHTML 1.0 specification, the closing tag `</p>` is required for XHTML compatibility.
- As a logical element, empty paragraphs are ignored by browsers, so do not try to use multiple `<p>` tags in a row, like `<p><p><p><p>`, to add blank lines to a Web page. This will not work; use the `<br>` tag instead.
- Remove the empty paragraph `<p>&nbsp;</p>` markup inserted by WYSIWYG editors.
- The HTML 3.2 specification supports only the **align** attribute with values of **center**, **left**, and **right**.
- The HTML 2.0 specification supports no attributes for the **p** element.

## `<param>` (Object Parameter)

This element specifies a parameter to be passed to an embedded object that is specified with the **object** or **applet** elements. This element should occur only within the scope of one of these elements.

### Standard Syntax

```
<param
  id="unique alphanumeric identifier"
  name="parameter name"
  type="mime Type"
  value="parameter value"
  valuetype="data | object | ref" />
```

### Element-Specific Attributes

**datafld** This Internet Explorer-specific attribute is used to indicate the column name in the data source that is bound to the `<param>` tag's value.

**dataformatas** This Internet Explorer-specific attribute indicates whether the bound data is plain text (**text**) or HTML (**html**).

**datasrc** The value of this attribute is set to an identifier indicating the data source to pull data from. Bound data is used to set the value of the parameters passed to the object or applet with which this `<param>` tag is associated.

**name** This attribute contains the parameter's name. The name of the parameter depends on the particular object being inserted into the page, and it is assumed that the object knows how to handle the passed data. Do not confuse the **name** attribute for this element with the **name** attribute used for form elements. In the latter case, the **name** attribute does not have a similar meaning to **id**, but rather specifies the name of the data to be passed to an enclosing `<object>` tag.

**type** When the **valuetype** attribute is set to **ref**, the **type** attribute can be used to indicate the type of the information to be retrieved. Legal values for this attribute are in the form of MIME types, such as **text/html**.

**value** This attribute contains the parameter's value. The actual content of this attribute depends on the object and the particular parameter being passed in, as determined by the **name** attribute.

**valuetype** This attribute specifies the type of the **value** attribute being passed in. Possible values for this attribute include **data**, **object**, and **ref**. A value of **data** specifies that the

information passed in through the **value** parameter should be treated just as data. A value of **ref** indicates that the information being passed in is a URL that indicates where the data to be used is located. The information is not retrieved, but the URL is passed to the object, which then can retrieve the information if necessary. The last value of **object** indicates that the value being passed in is the name of an object as set by its **id** attribute. In practice, the **data** attribute is used by default.

### Examples

```
<applet code="plot.class">
  <param name="min" value="5" />
  <param name="max" value="30" />
  <param name="ticks" value=".5" />
  <param name="line-style" value="dotted" />
</applet>
```

```
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
  codebase="swflash.cab#version=2,0,0,0"
  height="100" width="100">
  <param id="param1" name="Movie" value="SplashLogo.swf" />
  <param id="param2" name="Play" value="True" />
</object>
```

### <plaintext> (Plain Text)

This deprecated element from the HTML 2.0 specification renders the enclosed text as plain text and forces the browser to ignore any enclosed HTML. Typically, information affected by the **<plaintext>** tag is rendered in monospaced font. This element is no longer part of the HTML standard and should never be used.

### Syntax (HTML 2; Deprecated Under HTML 4)

```
<plaintext>
```

### Example

```
<!DOCTYPE html PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html>
<head><title>Plaintext Example</title></head>
<body>
```

The rest of this file is in plain text.

```
<plaintext>
```

Even though this is supposed to be **<b>bold</b>**, the tags still show.

There is no way to turn plain text off once it is on. **</plaintext>** does nothing to help. Even **</body>** and **</html>** will show up.

### Notes

- No closing tag for this element is necessary because the browser will ignore all tags after the starting tag.

- This element should not be used. Plain text information can be indicated by a file type, and information can be inserted in a preformatted fashion using the **pre** element.
- Most browsers continue to support this tag despite documentation to the contrary.

## <pre> (Preformatted Text)

This element is used to indicate that the enclosed text is preformatted, meaning that spaces, returns, tabs, and other formatting characters are preserved. Browsers will, however, acknowledge most HTML elements that are found within a <pre> tag. Preformatted text generally will be rendered by the browsers in a monospaced font.

### Standard Syntax

```
<pre
  class="class name(s)"
  dir="ltr | rtl"
  id="unique alphanumeric value"
  lang="language code"
  style="style information"
  title="advisory text"
  width="number" (transitional only)
  xml:space="preserve">
```

```
</pre>
```

### Element-Specific Attributes

**width** This attribute should be set to the width of the preformatted region. The value of the attribute should be the number of characters to display. In practice, this attribute is not supported and is dropped under the strict HTML 4.01 specification.

**wrap** In some versions of Microsoft browsers, this attribute controls word wrap behavior within a <pre> tag. The default value of **off** for the attribute forces the element not to wrap text, so the author must manually enter line breaks. Values of **hard** or **soft** cause word wrap and set different types of line breaks in the wrapped text. Given the nature of the **pre** element, the value of this attribute is limited.

**xml:space** This attribute is included from XHTML 1.0 and is used to set whether spaces need to be preserved within the element or whether the default white space handling should be employed. It is curious that an element defined to override traditional white space rules would allow such an attribute, and in practice this attribute is not used by developers.

### Example

```
<pre>
  Within PREFORMATTED text  A L L  formatting IS  PRESERVED
  NO m a t t e r how wild it is. Remember that some
  <b>HTML</b> markup is allowed within the &lt;PRE&gt; element.
```

```
</pre>
```

### Notes

- The HTML 4.01 and XHTML 1.0 transitional specifications state that the **applet**, **basefont**, **big**, **font**, **img**, **object**, **small**, **sub**, and **sup** elements should not be used within the <pre> tag. The strict HTML and XHTML specifications state that only the <big>, <img>, <object>, <small>, <sub>, and <sup> tags should not be used within the <pre>

tag. The other excluded elements are missing, as they are deprecated from the strict specification. Although these elements should not be used, it appears that the two most popular browsers will render them anyway.

- The strict HTML and XHTML specifications drop support for the **width** attribute, which was not well-supported anyway.
- The HTML 2.0 and 3.2 specifications support only the **width** attribute for `<pre>`.

## `<q>` (Quote)

This element indicates that the enclosed text is a short inline quotation.

### Standard Syntax

```
<q  
  cite="url of source"  
  class="class name(s)"  
  dir="ltr | rtl"  
  id="unique alphanumeric string"  
  lang="language code"  
  style="style information"  
  title="advisory text">
```

```
</q>
```

### Element-Specific Attributes

**cite** The value of this attribute is a URL that designates a source document or message for the information quoted. This attribute is intended to point to information explaining the context or the reference for the quote.

### Example

```
<q style="color: green;">A few green balls and a rainbow bar will  
give you an exciting Web page Christmas Tree!</q>
```

### Notes

- This element is intended for short quotations that don't require paragraph breaks, as compared to text that would be contained within `<blockquote>`.
- Some browsers, like Internet Explorer, may not make any sort of style change for quotations, but it is possible to apply a style rule.
- Most modern standards-aware browsers, like Mozilla, Opera, and Safari, should add quotes around text enclosed within the `q` element.

## `<s>` (Strikethrough)

This element renders the enclosed text with a line drawn through it.

### Standard Syntax (Transitional Only)

```
<s  
  class="class name(s)"  
  dir="ltr | rtl"
```

```
id="unique alphanumeric identifier"
lang="language code"
style="style information"
title="advisory text">
```

</s>

### Examples

This line contains a <s>misstake</s>.

```
<s id="strike1"
  onmouseover="this.style.color='red';"
  onmouseout="this.style.color='black';">Fastball</s>
```

### Notes

- This element should act the same as the **strike** element.
- This HTML 3 element eventually was adopted by Netscape and
- Microsoft and later was incorporated into the HTML 4.01 transitional
- specification.
- This element has been deprecated by the W3C. The strict HTML
- 4.01 specification does not include the **s** element or the **strike** element. It is possible to
- indicate
- strikethrough text using a style sheet.

## <samp> (Sample Text)

This element is used to indicate sample text. Enclosed text generally is rendered in a monospaced font.

### Standard Syntax

```
<samp
  class="class name(s)"
  dir="ltr | rtl"
  id="unique alphanumeric string"
  lang="language code"
  style="style information"
  title="advisory text">
```

</samp>

### Example

Use the following salutation in all e-mail messages to the boss:

```
<samp>Please excuse the interruption, oh exalted manager.</samp>
```

### Notes

- As a logical element, **samp** is useful to bind style rules to.
- The HTML 2.0 and 3.2 specifications support
- no attributes for this element.

## <script> (Scripting)

This element encloses statements in a scripting language for client-side processing. Scripting statements can either be included inline or loaded from an external file and might be commented out to avoid execution by nonscripting-aware browsers.

### Standard Syntax

```
<script
  charset="character set"
  defer="defer"
  id="unique alphanumeric identifier"
  language="scripting language name"
  src="url of script code"
  type="mime type"
  xml:space="preserve">
</script>
```

### Element-Specific Attributes

**charset** This attribute defines the character encoding of the script. The value is a space- and/or comma-delimited list of character sets as defined in RFC 2045. The default value is **ISO-8859-1**.

**defer** Presence of this attribute indicates that the browser might defer execution of the script enclosed by the **<script>** tag. In practice, deferring code might be more dependent on the position of the **<script>** tag or the contents. Support for this attribute is inconsistent.

**event** This Microsoft-specific attribute is used to define a particular event that the script should react to. It must be used in conjunction with the **for** attribute. Event names are the same as event handler attributes; for example, **onclick**, **ondblclick**, and so on.

**for** The **for** attribute is used in Microsoft browsers to define the name or ID of the element to which an event defined by the **event** attribute is related. For example, **<script event="onclick" for="button1" language="vbscript">** defines a VBScript that will execute when a click event is issued for an element named button1.

**language** This attribute specifies the scripting language being used. The Netscape implementation supports JavaScript.

**src** This attribute specifies the URL of a file containing scripting code. Typically, files containing JavaScript code will have a .js extension, and a server will attach the appropriate MIME type; if not, the **type** attribute might be used to explicitly set the content type of the external script file. The **language** attribute also might be helpful in determining this.

**type** This attribute should be set to the MIME type corresponding to the scripting language used. For JavaScript, for example, this would be **text/javascript**. In practice, the **language**

attribute is the more common way to indicate which scripting language is in effect, but the **type** attribute is standard.

**xml:space** This attribute is included from XHTML 1.0 on, and is used to set whether spaces need to be preserved within the script element or whether the default white space handling should be employed. In practice, this attribute is not used by developers.

### Examples

```
<script type="text/javascript">
  alert("Hello World !!!");
</script>
<!-- code in external file -->
<script language="JavaScript1.2" src="superrollover.js"></script>
```

### Notes

- It is common practice to comment out statements enclosed by a **<script>** tag. Without commenting, scripts are displayed as page content by browsers that do not support scripting. The particular comment style might be dependent on the language being used. For example, in JavaScript, use  

```
<script type="text/javascript">
<!-- JavaScript code here// -->
</script>
```

  
In VBScript, use  

```
<script type="text/vbscript">
<!-- VBScript code here -->
</script>
```
- The **event** and **for** attributes are defined under transitional versions of HTML 4.01 but only as reserved values. Later specifications appear to have dropped potential support for them, though they continue to be supported by Internet Explorer. Most browsers assume JavaScript when parsing a script element without a set **type** or **language** attribute. Refer to the **<noscript>** tag reference in this appendix to see how content might be identified for nonscripting-aware browsers.

## <select> (Selection List)

This element defines a selection list within a form. Depending on the form of the selection list, the control allows the user to select one or more list options.

### Standard Syntax

```
<select
  class="class name(s)"
  dir="ltr | rtl"
  disabled="disabled"
  id="unique alphanumeric identifier"
```

```
lang="language code"  
multiple="multiple"  
name="unique alphanumeric name"  
size="number"  
style="style information"  
tabindex="number"  
title="advisory text">
```

*option and optgroup elements only*

</select>

### Element-Specific Attributes

**align** This Microsoft-specific attribute controls the alignment of the image with respect to the content on the page. The default value is **left**, but other values such as **absbottom**, **absmiddle**, **baseline**, **bottom**, **middle**, **right**, **texttop**, and **top** also might be supported. The meaning of these values should be similar to those used for inserted objects, such as images.

**datafld** This Internet Explorer-specific attribute is used to indicate the column name in the data source that is bound to the options in the **select** element.

**datasrc** The value of this Internet Explorer-specific attribute is set to an identifier indicating the data source that data should be pulled from.

**disabled** This attribute is used to turn off a form control. Elements will not be submitted nor can they receive any focus from the keyboard or mouse. Disabled form controls will not be part of the tabbing order. The browser also can gray out the form that is disabled in order to indicate to the user that the form control is inactive. This attribute requires no value under traditional HTML, but under XHTML variants should be set to **disabled**.

**multiple** This attribute allows the selection of multiple items in the selection list. The default is single-item selection. Under XHTML, this attribute must have a value set to **multiple**.

**name** This attribute allows a form control to be assigned a name so that it can be referenced by a scripting language. **Name** is supported by older browsers, such as Netscape 2-generation browsers, but the W3C encourages the use of the **id** attribute. For compatibility purposes, both might have to be used, though this may cause trouble.

**size** This attribute sets the number of visible items in the selection list. When the **multiple** attribute is not present, only one entry should show; however, when **multiple** is present, this attribute is useful for setting the size of the scrolling list box.

**tabindex** This attribute takes a numeric value indicating the position of the form control in the tabbing index for the form. Tabbing proceeds from the lowest positive **tabindex** value to the highest. Negative values for **tabindex** will leave the form control out of the tabbing order. When tabbing is not explicitly set, the browser might tab through items in the order they are

encountered. Form controls that are disabled due to the presence of the **disabled** attribute will not be part of the tabbing index.

### Examples

Choose your favorite colors:

```
<select name="colors" multiple="multiple" size="2">
  <option>Red</option>
  <option>Blue</option>
  <option>Green</option>
  <option>Yellow</option>
</select>
```

Taco Choices:

```
<select name="tacomenu">
  <option value="SuperChicken">Chicken</option>
  <option value="Baja">Fish</option>
  <option value="RX-Needed">Carnitas</option>
</select>
```

### Notes

- The HTML 4.01 specification reserves the attributes **datafld** and **datasrc** for future use, but these are removed under XHTML.
- Internet Explorer's variant of the disabled attribute allows values of **true** and **false**, as well as the standard **disabled** value.
- Under traditional HTML, the end tag **</option>** is often omitted.
- Be careful of the **name** and **id** attribute problem that may occur, particularly when setting the **multiple** attribute. It may be better to have separate values.
- The HTML 2.0 and 3.2 specifications define only **multiple**, **name**, and **size** attributes.

### <small> (Small Text)

This element renders the enclosed text one font size smaller than a document's base font size, unless it is already set to the smallest size.

### Standard Syntax

```
<small
  class="class name(s)"
  dir="ltr | rtl"
  id="unique alphanumeric string"
  lang="language code"
  style="style information"
  title="advisory text">
</small>
```

### Examples

Here is some `<small>`small text`</small>`.

This element can be applied `<small><small><small>`multiple times`</small></small></small>`to make things even smaller.

### Notes

- A `<small>` tag can be used multiple times to decrease the size of text to a greater degree. Using more than six `<small>` tags together doesn't make sense because browsers currently only support relative font sizes from **1** to **7**. As style sheets become more common, this element might fall out of favor or be used with measurements in a more logical fashion.
- The default base font size for a document typically is **3**, although it can be changed with the `<basefont>` tag.

## `<spacer>` (Extra Space)

This Netscape proprietary element specifies an invisible region that is useful for page layout.

### Proprietary Syntax (Netscape 3 and 4 Only)

```
<spacer
  align="absmiddle | absbottom | baseline | bottom |
    left | middle | right | texttop | top"
  height="pixels"
  size="pixels"
  type="block | horizontal | vertical"
  width="pixels">
```

### Element-Specific Attributes

**align** This attribute specifies the alignment of the spacer with respect to surrounding text. It is used only with spacers with `type="block"`. The default value for the **align** attribute is **bottom**. The meanings of the **align** values are similar to those used with the **img** element.

**height** This attribute specifies the height of the invisible region in pixels. It is used only with spacers with `type="block"`.

**size** Used with `type="block"` and `type="horizontal"` spacers, this attribute sets the spacer's width in pixels. Used with a `type="vertical"` spacer, this attribute is used to set the spacer's height.

**type** This attribute indicates the type of invisible region. A **horizontal** spacer adds horizontal space between words and objects. A **vertical** spacer is used to add space between lines. A **block** spacer defines a general-purpose positioning rectangle, like an invisible image that text can flow around.

**width** This attribute is used only with the `type="block"` spacer and is used to set the width of the region in pixels.

## Examples

A line of text with two `<spacer type="horizontal" size="20">`words separated by 20 pixels. Here is a line of text.`<br>`

`<spacer type="vertical" size="50">`

Here is another line of text with a large space between the two lines.`<spacer align="left" type="block" height="100" width="100">` This is a bunch of text that flows around an invisible block region. You could have easily performed this layout with a table.

## `<span>` (Text Span)

This element typically is used to group inline text so scripting or style rules can be applied to the content. As it has no preset or rendering meaning, this is the most useful inline element for associating style and script with content.

### Syntax

```
<span
  class="class name(s)"
  dir="ltr | rtl"
  id="unique alphanumeric string"
  lang="language code"
  style="style information"
  title="advisory text">
</span>
```

### Examples

Here is some `<span style="font-size: 14pt; color: purple;">`very strange`</span>` text.

```
<span id="toggletext"
  onclick="this.style.color='red';"
  ondblclick="this.style.color='black';">
```

Click and Double Click Me

```
</span>
```

### Notes

- The HTML 4.01 specification reserved the **datafld**, **dataformatas**, and **datasrc** attributes for future use. They were later dropped from XHTML.
- As a generic element, **span**, like **div**, is useful for binding style to arbitrary content. However, **span** is an inline element and does not cause a return by default like **div**.

## `<strike>` (Strikeout Text)

This element is used to indicate strikethrough text, namely text with a line drawn through it. The `s` element provides shorthand notation for this element.

### Syntax (Transitional Only)

```
<strike
  class="class name(s)"
```

```
dir="ltr | rtl"
id="unique alphanumeric string"
lang="language code"
style="style information"
title="advisory text">
```

</strike>

### Examples

This line contains a spelling <strong>misstake</strong> mistake.

Price: \$<strong style="color: red;">5.00</strong>3.00

### Notes

- This tag should act the same as the <s> tag.
- This element has been deprecated by the W3C. The strict HTML and XHTML specifications do not include the <strong> tag nor the <s> tag as it is possible to indicate strikethrough text using the style sheet property **text-decoration**.

## <strong> (Strong Emphasis)

This element indicates strongly emphasized text. It usually is rendered in a bold typeface, but its rendering is not guaranteed, as it is a logical element.

### Syntax

```
<strong
  class="class name(s)"
  dir="ltr | rtl"
  id="unique alphanumeric string"
  lang="language code"
  style="style information"
  title="advisory text">
</strong>
```

### Examples

It is really <strong>important</strong> to pay attention.

<strong style="font-family: impact; font-size: 28pt;">

Important Info

</strong>

### Notes

- This element generally renders as bold text. As a logical element, however, **strong** is useful to bind style rules to.
- As compared to **b**, this element does have meaning. For example, voice browsers may speak <strong> enclosed text in a different voice than text that is enclosed by <b>.

## <style> (Style Information)

This element is used to surround style sheet rules for a document. This element should be found only in the **head** element. Style rules within a document's <body> should be set with the style attribute for a particular element.

### Syntax

```
<style
  dir="ltr | rtl"
  id="unique alphanumeric string"
  lang="language code"
  media="all | print | screen | others"
  title="advisory text"
  type="MIME Type"
  xml:space="preserve">
  CSS properties
</style>
```

### Common Attributes

disabled="disabled" (DOM Level 1)

### Element-Specific Attributes

**disabled** This initially Microsoft-defined attribute is used to disable a style sheet. The presence of the attribute is all that is required to disable the style sheet. In conjunction with scripting, this attribute could be used to turn on and off various style sheets in a document. While not documented in later versions of Internet Explorer, this attribute is very much supported and used, since it is part of the DOM standard. Internet Explorer may also support values of **true** and **false**.

**media** This attribute specifies the destination medium for the style information. The value of the attribute can be a single media descriptor, like **screen** or a comma-separated list. Possible values for this attribute include **all**, **aural**, **braille**, **print**, **projection**, **screen**, and **tv**. Other values also might be defined, depending on the browser. Internet Explorer supports **all**, **print**, and **screen** as values for this attribute.

**type** This attribute is used to define the type of style sheet. The value of the attribute should be the MIME type of the style sheet language used. The most common current value for this attribute is **text/css**, which indicates a cascading style sheet format.

**xml:space** This attribute is included from XHTML 1.0 and is used to specify whether spaces need to be preserved within the script element or whether the default white space handling should be employed. In practice, this attribute is not used by developers.

### Example

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
```

```

<title>Style Sheet Example</title>
<meta http-equiv="content-type" content="text/html; charset=ISO-8859-1" />
<style type="text/css">
  body { background: black; color: white;
        font: 12pt Helvetica;}
  h1 { color: red; font: 14pt Impact;}
</style>
</head>
<body>
<h1>A 14-point red Impact heading on a black
background</h1>
<p>Regular body text, which is 12 point white Helvetica.</p>
</body>
</html>

```

### Notes

- Style information also can be specified in external style sheets as defined by the `<link>` tag.
- Style information can also be associated with a particular element using the `style` attribute.
- Style rules are often commented out within a `<style>` tag to avoid interpretation by nonconforming browsers.
- `<style type="text/css">`  

```

<!--
  body { background-color: red;
-->
</style>

```

## `<sub>` (Subscript)

This element renders its content as subscripted text.

### Syntax

```

<sub
  class="class name(s)"
  dir="ltr | rtl"
  id="unique alphanumeric string"
  lang="language code"
  style="style information"
  title="advisory text">

```

```

</sub>

```

### Examples

Here is some `<sub>`subscripted`</sub>` text.

The secret formula is X`<sub>``<small>`2`</small>``</sub>`

## Notes

- The HTML 3.2 specification supports no attribute for **<sub>**.
- The **<sub>** tag is removed from XHTML 1.1 since it can be created using the CSS property **vertical-align**.
- Most browsers may slightly shift text lines below a **<sub>** tag.

## <sup> (Superscript)

This element renders its content as superscripted text.

## Syntax

```
<sup  
  class="class name(s)"  
  dir="ltr | rtl"  
  id="unique alphanumeric string"  
  lang="language code"  
  style="style information"  
  title="advisory text">
```

```
</sup>
```

## Examples

Here is some **<sup>**superscripted**</sup>** text.

$x^2 = 4$  when  $x = 2$

## Notes

- The HTML 3.2 specification supports no attribute for **<sup>**.
- The **<sup>** tag is removed from XHTML 1.1 since it can be created using the CSS property **vertical-align**.
- Most browsers may slightly shift text lines above a **<sup>** tag.

## <table> (Table)

This element is used to define a table. Tables are used to organize data as well as to provide structure for laying out pages when not using CSS.

## Standard Syntax

```
<table  
  align="center | left | right" (transitional only)  
  bgcolor="color name | #RRGGBB" (transitional only)  
  border="pixels"  
  cellpadding="pixels"  
  cellspacing="pixels"  
  class="class name(s)"  
  dir="ltr | rtl"
```

```
frame="above | below | border | box | hside |  
      lhs | rhs | void | vside"  
id="unique alphanumeric identifier"  
lang="language code"  
rules="all | cols | groups | none | rows"  
style="style information"  
summary="summary information"  
title="advisory text"  
width="percentage | pixels">  
caption, col, colgroup, thead, tbody, tfoot, and tr elements only  
</table>
```

### Nonstandard Attributes Commonly Supported

```
background="url of image" file  
bordercolor="color name | #RRGGBB"  
cols="number of columns"  
height="percentage | pixels"  
hspace="pixels" (Netscape variants only)  
vspace="pixels" (Netscape variants only)
```

### Element-Specific Attributes

**align** This attribute specifies the alignment of the table with respect to surrounding text. The HTML 4.01 specification defines **center**, **left**, and **right**. Some browsers also might support alignment values, such as **absmiddle**, that are common to block objects.

**background** This nonstandard attribute, which is supported by nearly every browser, specifies the URL of a background image for the table. The image is tiled if it is smaller than the table dimensions. Note that some early versions of Netscape display the background image in each table cell, rather than behind the complete table, as in Internet Explorer.

**bgcolor** This attribute specifies a background color for a table. Its value can be either a named color, such as **red**, or a color specified in the hexadecimal *#RRGGBB* format, such as **#FF0000**.

**border** This attribute specifies, in pixels, the width of a table's borders. A value of **0** makes a borderless table, which is useful for graphic layout.

**bordercolor** This attribute, supported by Internet Explorer and Netscape, is used to set the border color for a table. The attribute should be used only with a positive value for the **border** attribute. The value of the attribute can be either a named color, such as **green**, or a color specified in the hexadecimal *#RRGGBB* format, such as **#00FF00**. The color applications may be slightly different in the two browsers, since Netscape colors only the outer border of the table.

**bordercolordark** This Internet Explorer-specific attribute specifies the darker of two border colors used to create a three-dimensional effect for cell borders. It must be used with the **border** attribute set to a positive value. The attribute value can be either a named color, such as **blue**, or a color specified in the hexadecimal *#RRGGBB* format, such as **#00FF00**.

**bordercolorlight** This Internet Explorer-specific attribute specifies the lighter of two border colors used to create a three-dimensional effect for cell borders. It must be used with the **border** attribute set to a positive value. The attribute value can be either a named color, such as **red**, or a color specified in the hexadecimal *#RRGGBB* format, such as **#FF0000**.

**cellpadding** This attribute sets the width, in pixels, between the edge of a cell and its content.

**cellspacing** This attribute sets the width, in pixels, between individual cells.

**cols** This attribute specifies the number of columns in the table and is used to help quickly calculate the size of the table. This attribute was part of the preliminary specification of HTML 4.0, but was later dropped. A few browsers, notably Netscape and Internet Explorer, support it.

**datapagesize** The value of this Microsoft-specific attribute is the number of records that can be displayed in the table when data binding is used.

**datasrc** The value of this Microsoft-specific attribute is an identifier indicating the data source that data will be pulled from for data binding.

**frame** This attribute specifies which edges of a table are to display a border frame. A value of **above** indicates only the top edge; **below** indicates only the bottom edge; and **border** and **box** indicate all edges, which is the default when the **border** attribute is a positive integer. A value of **hsides** indicates only the top and bottom edges should be displayed, **lhs** indicates the left-hand edge should be displayed, **rhs** indicates the right-hand edge should be displayed, **vsides** indicates the left and right edges both should be displayed, and **void** indicates no border should be displayed.

**height** This attribute specifies the height of the table in pixels or percentage of the browser window. Be careful, because some browser versions may not support percentage values on height or may have variations in this calculation when they do support it.

**hspace** This Netscape-specific attribute indicates the horizontal space, in pixels, between the table and surrounding content, similar to the same attribute on **<img>**.

**rules** This attribute controls the display of dividing rules within a table. A value of **all** specifies dividing rules for rows and columns. A value of **cols** specifies dividing rules for columns only. A value of **groups** specifies horizontal dividing rules between groups of table cells defined by the **thead**, **tbody**, **tfoot**, or **colgroup** elements. A value of **rows** specifies dividing rules for rows only. A value of **none** indicates no dividing rules and is the default.

**summary** This attribute is used to provide a text summary of the table's purpose and structure. This element is used for accessibility, and its presence is important for non-visual user agents.

**vspace** This Netscape attribute indicates the vertical space in pixels between the table and surrounding content, similar to the same attribute on **<img>**.

**width** This attribute specifies the width of the table either in pixels or as a percentage of the enclosing window.

### Examples

```
<table bgcolor="white" border="2">
  <tr>
    <td>Cell 1</td>
    <td>Cell 2</td>
    <td>Cell 3</td>
    <td>Cell 4</td>
  </tr>

  <tr>
    <td>Cell 5</td>
    <td>Cell 6</td>
  </tr>
</table>
<table rules="all" bgcolor="yellow">
<caption>Widgets by Area</caption>
<thead align="center" bgcolor="green" valign="middle">
  <tr>
    <td>This is a Header</td>
  </tr>
</thead>

<tfoot align="right" bgcolor="red" valign="bottom">
  <tr>
    <td colspan="2">This is part of the footer.</td>
    <td>This is also part of the footer.</td>
  </tr>
</tfoot>

<tbody>
  <tr>
    <td>&nbsp;</td>
    <th>Regular Widget</th>
    <th>Super Widget</th>
  </tr>

  <tr>
    <th>West Coast</th>
    <td>10</td>
    <td>12</td>
  </tr>
```

```
<tr>
  <th>East Coast</th>
  <td>1</td>
  <td>20</td>
</tr>
</tbody>
</table>
```

## Notes

- In addition to displaying tabular data, tables are used to support graphic layout and design.
- The HTML 4 specification reserved the attributes **datasrc**, **datafld**, **dataformatas**, and **datapagesize** for future versions. However, XHTML dropped these attributes. They are supported in Internet Explorer 4 and later.
- At the time of this writing, most browsers have problems with **char** and **charoff** attributes in all table-related tags.
- The HTML 3.2 specification defines only the **align**, **border**, **cellpadding**, **cellspacing**, and **width** attributes for the **table** element.
- The **cols** attribute might provide an undesirable result under some versions of Netscape, which assumes the size of each column in the table is exactly the same.

## <tbody> (Table Body)

This element is used to group the rows within the body of a table so that common alignment and style defaults can easily be set for numerous cells.

### Standard Syntax

```
<tbody align="center | char | justify | left | right"
  char="character"
  charoff="offset"
  class="class name(s)"
  dir="ltr | rtl"
  id="unique alphanumeric identifier"
  lang="language code"
  style="style information"
  title="advisory text"
  valign="baseline | bottom | middle | top">
  tr elements only
</tbody>
```

### Element-Specific Attributes

**align** This attribute is used to align the contents of the cells within a **<tbody>** tag. Common values are **center**, **justify**, **left**, and **right**. The specification also defines a value of **char**. When **align** is set to **char**, the attribute **char** must be present and set to the character to which cells

should be aligned. A common use of this approach would be to set cells to align on a decimal point. Unfortunately, browsers do not support the **char** align value well.

**bgcolor** This attribute specifies a background color for the cells within a **<tbody>** tag. Its value can be either a named color, such as **red**, or a color specified in the hexadecimal **#RRGGBB** format, such as **#FF0000**.

**char** This attribute is used to define the character to which element contents are aligned when the **align** attribute is set to the **char** value.

**charoff** This attribute contains an offset as a positive or negative integer to align characters as related to the **char** value. A value of **2** would align characters in a cell two characters to the right of the character defined by the **char** attribute.

**valign** This attribute is used to set the vertical alignment for the table cells within a **<tbody>** tag. The HTML specification defines **baseline**, **bottom**, **middle**, and **top**. Internet Explorer also supports **center**, which should act like **middle**.

### Example

```
<table rule="all" bgcolor="yellow">
<tbody align="center" bgcolor="red" style="bodystyle"
  valign="baseline">
  <tr>
    <td>&nbsp;</td>
    <th>Regular Widget</th>
    <th>Super Widget</th>
  </tr>
  <tr>
    <th>West Coast</th>
    <td>10</td>
    <td>12</td>
  </tr>
  <tr>
    <th>East Coast</th>
    <td>1</td>
    <td>20</td>
  </tr>
</tbody>
</table>
```

### Notes

- This element is found only in a **<table>** tag and contains one or more table rows, as indicated by **<tr>** tags.
- For XHTML compatibility, the closing **</tbody>** tag must be used with this element; however, it is optional under traditional HTML.

## <td> (Table Data)

This element specifies a data cell in a table. The element should occur within a table row as defined by the **tr** element.

### Standard Syntax

```
<td
  abbr="abbreviation"
  align="center | justify | left | right"
  axis="group name"
  bgcolor="color name | #RRGGBB" (transitional only)
  char="character"
  charoff="offset"
  class="class name"
  colspan="number of columns to span"
  dir="ltr | rtl"
  headers="space-separated list of associated header
           cells' id values"
  height="pixels or percentage" (transitional only)
  id="unique alphanumeric identifier"
  lang="language code"
  nowrap="nowrap" (transitional only)
  rowspan="number or rows to span"
  scope="col | colgroup | row | rowgroup"
  style="style information"
  title="advisory text"
  valign="baseline | bottom | middle | top"
  width="pixels or percentage"> (transitional only)
```

```
</td>
```

### Nonstandard Attributes Commonly Supported

```
background="url of image file"
bordercolor="color name | #RRGGBB"
```

### Element-Specific Attributes

**abbr** The value of this attribute is an abbreviated name for a header cell. This might be useful when attempting to display large tables on small screens.

**align** This attribute is used to align the contents of the cells. Supported values are **center**, **justify**, **left**, and **right**.

**axis** This attribute is used to provide a name for a group of related headers.

**background** This nonstandard attribute, which is supported by major browsers, specifies the URL of a background image for the table cell. The image is tiled if it is smaller than the cell's dimensions.

**bgcolor** This attribute specifies a background color for a table cell. Its value can be either a named color, such as **red**, or a color specified in the hexadecimal *#RRGGBB* format, such as **#FF0000**. Note that some older versions of Netscape Navigator may not render a cell with a colored background unless a nonbreaking space, at least, is inserted in the cell.

**bordercolor** This attribute, supported by Internet Explorer and Netscape, is used to set the border color for a table cell. The attribute should be used only with a positive value for the **border** attribute. The value of the attribute can be either a named color, such as **green**, or a color specified in the hexadecimal *#RRGGBB* format, such as **#00FF00**.

**bordercolordark** This Internet Explorer-specific attribute specifies the darker of two border colors used to create a three-dimensional effect for a cell's borders. It must be used with the **border** attribute set to a positive value. The attribute value can be either a named color, such as **blue**, or a color specified in the hexadecimal *#RRGGBB* format, such as **#00FF00**.

**bordercolorlight** This Internet Explorer-specific attribute specifies the lighter of two border colors used to create a three-dimensional effect for a cell's borders. It must be used with the **border** attribute set to a positive value. The attribute value can be either a named color, such as **red**, or a color specified in the hexadecimal *#RRGGBB* format, such as **#FF0000**.

**char** This attribute is used to define the character to which element contents are aligned when the **align** attribute is set to the **char** value.

**charoff** This attribute contains an offset, specified as a positive or negative integer, to align characters as related to the **char** value. A value of **2**, for example, would align characters in a cell two characters to the right of the character defined by the **char** attribute.

**colspan** This attribute takes a numeric value that indicates how many columns wide a cell should be. This is useful for creating tables with cells of different widths.

**headers** This attribute takes a space-separated list of **id** values that correspond to the header cells related to this cell.

**height** This attribute indicates the height of the cell in pixels or percentage. Some browsers may have rendering problems with percentage values.

**nowrap** This attribute keeps the content within a table cell from automatically wrapping. The **nowrap** attribute takes no value under HTML but should be set to the value **nowrap** under XHTML.

**rowspan** This attribute takes a numeric value that indicates how many rows high a table cell should span. This attribute is useful in defining tables with cells of different heights.

**scope** This attribute specifies the table cells that the current cell provides header information for. A value of **col** indicates that the cell is a header for the rest of the column below it. A value of **colgroup** indicates that the cell is a header for its current column group. A value of **row** indicates that the cell contains header information for the rest of the row it is in. A value of **rowgroup** indicates that the cell is a header for its row group. This attribute might be used in place of the **header** attribute and is useful for rendering assistance by nonvisual browsers. This attribute was added very late to the HTML 4 specification, and support for this attribute is still minimal.

**valign** This attribute is used to set the vertical alignment for the table cell. The specification defines **baseline**, **bottom**, **middle**, and **top**. Internet Explorer also supports **center**, which should be the same as **middle**.

**width** This attribute specifies the width of a cell in pixels or percentage value.

### Examples

```
<table>
<tr>
  <td align="left" valign="top" width="100">
    Put me in the top left corner.
  </td>
  <td align="right" bgcolor="red" valign="bottom" width="100">
    Put me in the bottom right corner.
  </td>
</tr>
</table>
```

```
<table border="1" width="80%">
  <tr>
    <td colspan="3">
      A pretty wide cell
    </td>
  <tr>
    <td>Item 2</td>
    <td>Item 3</td>
    <td>Item 4</td>
  </tr>
</table>
```

### Notes

- Under the XHTML 1.0 specification, the closing `</td>` tag ceases to be optional.
- The HTML 3.2 specification defines only **align**, **colspan**, **height**, **nowrap**, **rowspan**, **valign**, and **width** attributes.
- This element should always be within the **tr** element.

## <textarea> (Multiline Text Input)

This element specifies a multiline text input field contained within a form.

### Standard Syntax

```
<textarea
  accesskey="character"
  class="class name"
  cols="number"
  dir="ltr | rtl"
  disabled="disabled"
  id="unique alphanumeric identifier"
  lang="language code"
  name="unique alphanumeric identifier"
  readonly="readonly"
  rows="number"
  style="style information"
  tabindex="number"
  title="advisory text">

</textarea>
```

### Element-Specific Attributes

**accesskey** This attribute specifies a keyboard navigation accelerator for the element. Pressing ALT or a similar key in association with the specified character selects the form control correlated with that key sequence. Page designers are forewarned to avoid key sequences already bound to browsers.

**cols** This attribute sets the width in characters of the text area. The typical default value for the size of a <textarea> tag when this attribute is not set is **20** characters.

**datafld** This attribute is used to indicate the column name in the data source that is bound to the content enclosed by the <textarea> tag.

**datasrc** The value of this attribute is an identifier indicating the data source to pull data from.

**disabled** This attribute is used to turn off a form control. Elements will not be submitted, nor can they receive any focus from the keyboard or mouse. Disabled form controls will not be part of the tabbing order. The browser also can gray out the form that is disabled in order to indicate to the user that the form control is inactive. This attribute requires no value.

**name** This attribute allows a form control to be assigned a name so that it can be referenced by a scripting language. **Name** is supported by older browsers, such as Netscape 2-generation browsers, but the W3C encourages the use of the **id** attribute. For compatibility purposes, both attributes might have to be used.

**readonly** This attribute prevents the form control's value from being changed. Form controls with this attribute set might receive focus from the user but might not be modified. Because they receive focus, a **readonly** form control will be part of the form's tabbing order. Finally, the control's value will be sent on form submission. Under XHTML, the value of the **readonly** attribute should be set to **readonly**.

**rows** This attribute sets the number of rows in the text area. The value of the attribute should be a positive integer.

**tabindex** This attribute takes a numeric value indicating the position of the form control in the tabbing index for the form. Tabbing proceeds from the lowest positive **tabindex** value to the highest. Negative values for **tabindex** will leave the form control out of the tabbing order. When tabbing is not explicitly set, the browser can tab through items in the order they are encountered. Form controls that are disabled due to the presence of the **disabled** attribute will not be part of the tabbing index, although read-only controls will be.

**wrap** In some versions of Netscape and Microsoft browsers, this attribute controls word-wrap behavior. A value of **off** for the attribute forces the `<textarea>` not to wrap text, so the viewer must manually enter line breaks. A value of **hard** causes word wrap and includes line breaks in text submitted to the server. A value of **soft** causes word wrap but removes line breaks from text submitted to the server. Internet Explorer supports a value of **physical**, which is equivalent to Netscape's **hard** value, and a value of **virtual**, which is equivalent to Netscape's **soft** value. If the **wrap** attribute is not included, text will still wrap under Internet Explorer, but older versions of Netscape, notably Netscape 4, will scroll horizontally in the text box. Given this problem, even though it is nonstandard, it may be a good idea to include the **wrap** attribute.

### Examples

```
<textarea name="CommentBox" id="CommentBox" cols="40" rows="8">  
Default text in field  
</textarea>
```

```
<textarea name="comment" id="comment" rows="10" cols="40" wrap="virtual"  
    align="center">  
</textarea>
```

### Notes

- Any text between the `<textarea>` and `</textarea>` tags is rendered as the default entry for the form control. Content within a **textarea** is not interpreted, so white space is preserved and tags themselves are ignored.
- The HTML 2.0 and 3.2 specifications define only the **cols**, **name**, and **rows** attributes for this element.
- The **textarea** element lacks a **maxlength** attribute, which causes a potential security risk. Potential intruders to a site may copy and paste large amounts of text or script code into a **textarea** element in an attempt to break a Web application. All submissions from a **textarea** should be checked carefully to avoid security risks.

- The HTML 4.01 specification reserves the **datafld** and **datasrc** attributes for future use with the **textarea** element.

## <tfoot> (Table Footer)

This element is used to group the rows within the footer of a table so that common alignment and style defaults can easily be set for numerous cells. This element might be particularly useful when setting a common footer for tables that are dynamically generated.

### Standard Syntax

```
<tfoot
  align="center | char | justify | left | right"
  char="character"
  charoff="offset"
  class="class name(s)"
  dir="ltr | rtl"
  id="unique alphanumeric identifier"
  lang="language code"
  style="style information"
  title="advisory text"
  valign="baseline | bottom | middle | top">
```

*tr elements only*

```
</tfoot>
```

### Element-Specific Attributes

**align** This attribute is used to align the contents of the cells within a **<tfoot>** tag. Common values are **center**, **justify**, **left**, and **right**. The HTML and XHTML specifications also define a value of **char**. When **align** is set to **char**, the attribute **char** must be present and set to the character to which cells should be aligned. A common use of this approach would be to set cells to align on a decimal point.

**char** This attribute is used to define the character to which element contents are aligned when the **align** attribute is set to the **char** value.

**charoff** This attribute contains an offset, as a positive or negative integer, for aligning characters as related to the **char** value. A value of **2**, for example, would align characters in a cell two characters to the right of the character defined by the **char** attribute.

**valign** This attribute is used to set the vertical alignment for the table cells within a **<tfoot>** tag. The specification defines **baseline**, **bottom**, **middle**, and **top**. Internet Explorer also supports **center**, which should be the same as **middle**.

### Example

```
<table border="1" bgcolor="yellow" width="80%">
<tfoot align="center" bgcolor="red" class="footer">
```

```

    valign="bottom">
<tr>
  <td>This is part of the footer.</td>
  <td>This is also part of the footer.</td>
</tr>
</tfoot>
<tbody class="tbody">
  <tr>
    <td>The contents of the table!</td>
  </tr>
</tbody>
</table>

```

## Notes

- This element is contained only by the **table** element and contains table rows as delimited by **tr** elements.
- While it would seem that this element should come after a **<tbody>** tag, it actually should come before it within a **<table>** tag.
- Under the XHTML 1.0 specification, the closing **</tfoot>** tag ceases to be optional.

## <th> (Table Header)

This element specifies a header cell in a table. The element should occur within a table row as defined by a **tr** element. The main difference between this element and **td** is that browsers might render table headers slightly differently, usually bolding and centering contents. However, the element is logical in nature and should be used to structure tables.

### Standard Syntax

```

<th
  abbr="abbreviation"
  align="center | justify | left | right"
  axis="group name"
  bgcolor="color name | #RRGGBB" (transitional only)
  char="character"
  charoff="offset"
  class="class name"
  colspan="number"
  dir="ltr | rtl"
  headers="space-separated list of associated header
    cells' id values"
  height="pixels" (transitional only)
  id="unique alphanumeric identifier"
  lang="language code"
  nowrap="nowrap" (transitional only)
  rowspan="number"
  scope="col | colgroup | row | rowgroup"
  style="style information"

```

title="advisory text"  
valign="baseline | bottom | middle | top"  
width="pixels"> (transitional only)

</th>

### **Nonstandard Attributes Commonly Supported**

background="url of image file"  
bordercolor="color name | #RRGGBB"

### **Element-Specific Attributes**

**abbr** The value of this attribute is an abbreviated name for a header cell. This might be useful when attempting to display large tables on small screens.

**align** This attribute is used to align the contents of the cells within a <tbody> tag. Common values are **center**, **justify**, **left**, and **right**.

**axis** This attribute is used to provide a name for a group of related headers.

**background** This nonstandard attribute, which is supported by most browsers, specifies the URL of a background image for the table cell. The image is tiled if it is smaller than the cell's dimensions.

**bgcolor** This attribute specifies a background color for a table cell. Its value can be either a named color, such as **red**, or a color specified in the hexadecimal *#RRGGBB* format, such as **#FF0000**.

**bordercolor** This attribute, supported by Internet Explorer and Netscape, is used to set the border color for a table cell. The attribute should be used only with a positive value for the **border** attribute. The value of the attribute can be either a named color, such as **green**, or a color specified in the hexadecimal *#RRGGBB* format, such as **#00FF00**.

**bordercolordark** This Internet Explorer-specific attribute specifies the darker of two border colors used to create a three-dimensional effect for a cell's borders. It must be used with the **border** attribute set to a positive value. The attribute value can be either a named color, such as **blue**, or a color specified in the hexadecimal *#RRGGBB* format, such as **#00FF00**.

**bordercolorlight** This Internet Explorer-specific attribute specifies the lighter of two border colors used to create a three-dimensional effect for a cell's borders. It must be used with the **border** attribute set to a positive value. The attribute value can be either a named color, such as **red**, or a color specified in the hexadecimal *#RRGGBB* format, such as **#FF0000**.

**char** This attribute is used to define the character to which element contents are aligned when the **align** attribute is set to the **char** value.

**charoff** This attribute contains an offset, specified as a positive or negative integer, for aligning characters as related to the **char** value. A value of **2**, for example, would align characters in a cell two characters to the right of the character defined by the **char** attribute.

**colspan** This attribute takes a numeric value that indicates how many columns wide a cell should be. This is useful for creating tables with cells of different widths.

**headers** This attribute takes a space-separated list of **id** values that correspond to the header cells related to this cell.

**height** This attribute indicates the height of the cell in pixels or percentage. Some browsers may have rendering problems with percentage values.

**nowrap** This attribute keeps the content within a table cell from automatically wrapping. The **nowrap** attribute takes no value under HTML but should be set to the value **nowrap** under XHTML.

**rowspan** This attribute takes a numeric value that indicates how many rows high a table cell should span. This attribute is useful in defining tables with cells of different heights.

**scope** This attribute specifies the table cells for which the current cell provides header information. A value of **col** indicates that the cell is a header for the rest of the column below it. A value of **colgroup** indicates that the cell is a header for its current column group. A value of **row** indicates that the cell contains header information for the rest of the row it is in. A value of **rowgroup** indicates that the cell is a header for its row group. This attribute can be used in place of the **header** attribute and is useful for rendering assistance by nonvisual browsers. This attribute was added very late to the HTML 4.0 specification, and support for this attribute is still minimal in browsers.

**valign** This attribute is used to set the vertical alignment for the table cell. The specification defines **baseline**, **bottom**, **middle**, and **top**. Internet Explorer also supports **center**, which should be the same as **middle**.

**width** This attribute specifies the width of a cell in pixels or percentage value.

### Examples

```
<table border="1">
  <tr>
    <th>Names</th>
    <th>Apples</th>
    <th>Oranges</th>
  </tr>
  <tr>
    <td>Bobby</td>
    <td>10</td>
    <td>5</td>
  </tr>
</table>
```

```

</tr>
<tr>
  <td>Ruby Sue</td>
  <td>20</td>
  <td>3</td>
</tr>
</table>

```

### Notes

- The HTML 3.2 specification defines only **align**, **colspan**, **height**, **nowrap**, **rowspan**, **valign**, and **width** attributes.
- This element should always be within the **tr** element.
- Under the XHTML 1.0 specification, the closing **</th>** tag ceases to be optional.

## <thead> (Table Header)

This element is used to group the rows within the header of a table so that common alignment and style defaults can easily be set for numerous cells. This element might be particularly useful when setting a common head for tables that are dynamically generated.

### Standard Syntax

```

<thead
  align="center | char | justify | left | right"
  char="character"
  charoff="offset"
  class="class name(s)"
  dir="ltr | rtl"
  id="unique alphanumeric identifier"
  lang="language code"
  style="style information"
  title="advisory text"
  valign="baseline | bottom | middle | top">
  tr elements only
</thead>

```

### Element-Specific Attributes

**align** This attribute is used to align the contents of the cells within a **<thead>** tag. Common values are **center**, **justify**, **left**, and **right**. The specification also defines a value of **char**. When **align** is set to **char**, the attribute **char** must be present and set to the character to which cells should be aligned. A common use of this approach would be to set cells to align on a decimal point.

**char** This attribute is used to define the character to which element contents are aligned when the **align** attribute is set to the **char** value.

**charoff** This attribute contains an offset, specified as a positive or negative integer, for aligning characters as related to the **char** value. A value of **2**, for example, would align characters in a cell two characters to the right of the character defined by the **char** attribute.

**valign** This attribute is used to set the vertical alignment for the table cells with a **<thead>** tag. The specification defines **baseline**, **bottom**, **middle**, and **top**. Internet Explorer also supports **center**, which should be the same as **middle**.

### Example

```
<table border="1" bgcolor="yellow" width="80%">
<thead align="center" bgcolor="red" class="footer"
  valign="bottom">
  <tr>
    <td>This is the Important Table Headline</td>
  </thead>

<tbody class="tbody">
  <tr>
    <td>The contents of the table!</td>
  </tr>
</tbody>
</table>
```

### Notes

- This element is contained only by a **<table>** tag and contains table rows as delimited by **<tr>** tags.
- Under the XHTML 1.0 specification, the closing **</thead>** tag ceases to be optional.

## <title> (Document Title)

This element encloses the title of an HTML document. It must occur within a document's **head** element and must be present in all valid documents. Meaningful titles are very important because they are used for bookmarking a page and might be used by search engines attempting to index the document.

### Standard Syntax

```
<title
  dir="ltr | rtl"
  id="unique alphanumeric identifier"
  lang="language code">

</title>
```

### Example

```
<head><title>Big Company: Products: Super Widget</title></head>
```

## Notes

- The **title** should be the first element found in the **head**.
- Meaningful names should provide information about the document. A poor title would be something like "My Home Page," whereas a better title would be "Joe Smith Home."
- Browsers can be extremely sensitive to the `<title>` tag. If the **title** element is malformed or not closed, the page might not even render in the browser.
- The HTML 2.0 and 3.2 specifications define no attributes for the **title** element.

## `<tr>` (Table Row)

This element specifies a row in a table. The individual cells of the row are defined by the **th** and **td** elements.

### Syntax

```
<tr
  align="center | justify | left | right"
  bgcolor="color name | #RRGGBB" (transitional only)
  char="character"
  charoff="offset"
  class="class name(s)"
  dir="ltr | rtl"
  id="unique alphanumeric identifier"
  lang="language code"
  style="style information"
  title="advisory text"
  valign="baseline | bottom | middle | top">
  td or th elements only
</tr>
```

### Element-Specific Attributes

**align** This attribute is used to align the contents of the cells within the element. Common values are **center**, **justify**, **left**, and **right**.

**bgcolor** This attribute specifies a background color for all the cells in a row. Its value can be either a named color, such as **red**, or a color specified in the hexadecimal `#RRGGBB` format, such as **#FF0000**.

**bordercolor** This attribute, supported by Internet Explorer and Netscape, is used to set the border color for table cells in the row. The attribute should be used only with a positive value for the **border** attribute. The value of the attribute can be either a named color, such as **green**, or a color specified in the hexadecimal `#RRGGBB` format, such as **#00FF00**.

**bordercolordark** This Internet Explorer-specific attribute specifies the darker of two border colors used to create a three-dimensional effect for the cell's borders. It must be used with the **border** attribute set to a positive value. The attribute value can be either a named color, such as **blue**, or a color specified in the hexadecimal `#RRGGBB` format, such as **#00FF00**.

**bordercolorlight** This Internet Explorer-specific attribute specifies the lighter of two border colors used to create a three-dimensional effect for a cell's borders. It must be used with the **border** attribute set to a positive value. The attribute value can be either a named color, such as **red**, or a color specified in the hexadecimal *#RRGGBB* format, such as **#FF0000**.

**char** This attribute is used to define the character to which element contents are aligned when the **align** attribute is set to the **char** value.

**charoff** This attribute contains an offset, specified as a positive or negative integer, for aligning characters as related to the **char** value. A value of **2**, for example, would align characters in a cell two characters to the right of the character defined by the **char** attribute.

**valign** This attribute is used to set the vertical alignment for the table cells with a **<tr>** tag. The specification defines **baseline**, **bottom**, **middle**, and **top**. Internet Explorer also allows **center**, which should be the same as **middle**.

#### Example

```
<table width="300" border="1">
  <tr bgcolor="red" align="center" valign="middle">
    <td>3</td>
    <td>5.6</td>
    <td>7.9</td>
  </tr>
</table>
```

#### Notes

- This tag is contained only in the **<table>**, **<thead>**, **<tbody>**, and **<tfoot>** tags. It contains the **<th>** and **<td>** tags.
- The HTML 3.2 specification defines only the **align** and **valign** attributes for this element.
- Under the XHTML 1.0 specification, the closing **</tr>** tag ceases to be optional.

## <tt> (Teletype Text)

This element is used to indicate that text should be rendered in a monospaced font similar to teletype text.

#### Standard Syntax

```
<tt
  class="class name(s)"
  dir="ltr | rtl"
  id="unique alphanumeric identifier"
  lang="language code"
  style="style information"
  title="advisory text">
</tt>
```

## Examples

Here is some `<tt>monospaced text</tt>`.

Source code in this tag: `<tt>main() { printf("hello world"); }</tt>`

## Notes

- This element has been deprecated by the W3C under XHTML 1.1.
- The look of the tag can be replicated with the CSS properties **font** or **font-family**.

## <u> (Underline)

This element indicates that the enclosed text should be displayed underlined.

## Standard Syntax (Transitional Only)

```
<u
  class="class name(s)"
  dir="ltr | rtl"
  id="unique alphanumeric string"
  lang="language code"
  style="style information"
  title="advisory text">
</u>
```

## Examples

Here is some `<u>underlined text</u>`.

Be careful with `<u>underlined</u>` text; it looks like  
`<a href="http://www.yahoo.com/">links</a>`.

## Notes

- This element has been deprecated by the W3C. Under the strict HTML and XHTML specifications, the `<u>` tag is not defined. The look provided by this element is supported by the CSS property **text-decoration**.
- Underlining text can be problematic because it looks similar to a link, especially in a black-and-white environment.

## <ul> (Unordered List)

This element is used to indicate an unordered list, namely a collection of items that do not have a numerical ordering. The individual items in the list are defined by the **li** element, which is the only allowed element within a `<ul>` tag.

## Standard Syntax

```
<ul
  class="class name(s)"
  compact="compact" (transitional only)
  dir="ltr | rtl"
  id="unique alphanumeric identifier"
```

```
lang="language code"
style="style information"
title="advisory text"
type="circle | disc | square"> (transitional only)
    List items specified by <li> tags
</ul>
```

### Element-Specific Attributes

**compact** This attribute indicates that the list should be rendered in a compact style. Few browsers actually change the rendering of the list, regardless of the presence of this attribute. The **compact** attribute requires no value.

**type** The **type** attribute is used to set the bullet style for the list. The values defined under HTML 3.2 and the transitional version of HTML and XHTML are **circle**, **disc**, and **square**. A user agent might decide to use a different bullet depending on the nesting level of the list, unless the **type** attribute is used. The **type** attribute is dropped under the strict versions of HTML 4 and XHTML because style sheets can provide richer bullet control.

### Examples

```
<ul compact="compact" title="Sushi Short List" type="circle">
  <li>Maguro</li>
  <li>Ebi</li>
  <li>Hamachi</li>
</ul>
```

```
<!-- Common but bad example -->
```

```
<ul>Indentation using lists should not be used, though it is common.</ul>
```

```
<!-- Correct list nesting -->
```

```
<ul compact title="Sushi Short List" type="circle">
  <li>Item 1
  <ul>
    <li>Item A</li>
    <li>Item B</li>
  </ul></li>
  <li>Item 2</li>
</ul>
```

### Notes

- HTML 2.0 supports only the **compact** attribute.
- The HTML 3.2 specification supports **compact** and **type**.
- Under the strict HTML and XHTML specifications, the **ul** element does not support the **compact** attribute or the **type** attribute. Both of these attributes can be safely replaced with style rules.

- Due to XHTML's deprecation of attribute minimization, the **compact** attribute must have a quoted attribute when used in the transitional variant: `<ul compact="compact"></ul>`
- Many Web page designers and page development tools use the `<ul>` tag to indent text. The only element that should occur within a **ul** element is **li**, so such markup does not conform to standards. However, this common practice is likely to continue.
- Since the content model of **ul** says list items should be the only item within `<ul>` tags. Nested lists should occur within `<li>` tags rather than outside them as they are commonly found.
- Style sheets provide much better support for bullet control.

## <var> (Variable)

This element is used to indicate a variable (an identifier that occurs in a programming language or a mathematical expression). The element is logical, although enclosed text is generally rendered in italics.

### Standard Syntax

```
<var
  class="class name(s)"
  dir="ltr | rtl"
  id="unique alphanumeric value"
  lang="language code"
  style="style information"
  title="advisory text">
</var>
```

### Examples

Assign the value 5 to the variable `<var>x</var>`.

The variable `<var>total</var>` holds the total order value.

### Notes

- As a logical element, **var** is a perfect candidate for style sheet binding.
- The HTML 2.0 and 3.2 specifications support no attributes for this element.

## <wbr> (Word Break)

This nonstandard element is used to indicate a place where a line break can occur if necessary. This element is used in conjunction with the **nobr** element, which is used to keep text from wrapping. When used this way, **wbr** can be thought of as a soft line break in comparison to a `<br>` tag. This element is common to many Netscape and Microsoft implementations, though it is not part of any HTML standard.

### Proprietary Syntax

```
<wbr
  id="unique alphanumeric value">
```

### Example

`<noobr>`A line break can occur here`<wbr>`but not elsewhere, even if the line is really long.`</noobr>`

### Notes

- Older versions of Internet Explorer documentation defined **class**, **language**, **style**, and **title** for this tag. However, they have little meaning, given this tag's purpose, and have since been eliminated from the documentation, though they may effectively be recognized in some manner by the browser parser.
- Though this is an empty element and should be written as `<wbr />` under XHTML, it does not need to be. It is not standard and will not validate anyway.
- Standards-based browsers, such as Mozilla and Opera, do not support this tag but oddly seem to support `<noobr>`.

## xml (XML Data Island)

This proprietary element introduced by Microsoft can be used to embed islands of XML (Extensible Markup Language) data into HTML documents; this will work only under Internet Explorer 5.0 or later. An `<xml>` tag can be used to reference outside data sources using the **src** attribute, or surround XML data in the HTML document itself.

### Internet Explorer Syntax

```
<xml
  id="unique alphanumeric value"
  src="url of xml data file">
```

*...embedded xml code...*

```
</xml>
```

### Element-Specific Attributes

**src** This attribute references an external XML data file.

### Examples

```
<!-- This code embeds xml data directly into a document.
All code between the xml tags is not HTML, but a
hypothetical example of xml. -->
```

```
<xml id="tasty">
  <combomeal>
    <burger>
      <name>Tasty Burger</name>
      <bun bread="white">
        <meat />
        <cheese />
        <meat />
      </bun>
    </burger>
  </combomeal>
</xml>
```

```
</burger>
<fries size="large" />
<drink size="large" flavor="Cola" />
</combomeal>
</xml>
```

<!-- This code fragment uses the src attribute to reference an external file containing xml data. -->

```
<xml src="combomeal.xml"></xml>
```

### Notes

- Support for the **<xml>** tag is limited to Internet Explorer 5 or better. Mozilla's approach is more focused on embedding HTML within XML documents.

### <xmp> (Example)

This deprecated but still widely supported element indicates that the enclosed text is an example. Example text generally is rendered in a monospaced font, and the spaces, tabs, and returns are preserved, as with the **pre** element.

### Syntax (Defined by HTML 2; Deprecated Under HTML 4)

```
<xmp>
</xmp>
```

### Example

**<xmp>**This is a large block of text used as an example. Note that returns as well as S P A C E S are preserved.**</xmp>**

### Note

- This element was first deprecated under HTML 3.2, yet all major browsers continue to support it, and it is well documented and even extended for Internet Explorer. The **<pre>** tag or style sheets should be used instead of this tag.