



# Updating and Replacing HTML in Web Pages

by Michael Foster, STC Network Systems

Many traditional HyperText Markup Language (HTML) commands, most used to modify type style and appearance, have – or are being – removed. The term for this is “deprecate” and the rule for this is to migrate or convert to using newer options. These replacements are mostly provided using Cascading Style Sheets (CSS) in the form of properties and classes (or “pseudo-classes”).

## Deprecated HTML Elements

- **<applet></applet>**  
The applet element has been deprecated in favor of the `<object></object>` tag.
- **<basefont />**  
The basefont element has been deprecated in favor of the CSS `font` and `color` properties.
- **<blackface></blackface>**  
The blackface element is a MSNTV/WebTV element that has been deprecated in favor of the CSS `font-weight` property.
- **<blockquote></blockquote>**  
The blockquote element is deprecated in favor of CSS properties like `margin` *when used to indent text*.
- **<center></center>**  
The center element has been deprecated in favor of using CSS to center elements.
- **<dir></dir>**  
The dir element has been deprecated in favor of the `<ul></ul>` tag.
- **<embed />**  
The embed element has been deprecated in favor of the `<object></object>` tag. This tag was reinstated in HTML 5.
- **<font></font>**  
The font element has been deprecated in favor of the `font` and `color` style properties.
- **<i></i>**  
The i element has been deprecated in favor of the CSS `font-style` property.
- **<isindex />**  
The isindex element has been deprecated in favor of the `input` tag.
- **<layer></layer>**  
The layer element has been deprecated in favor of CSS positioning.
- **<menu></menu>**  
The menu element has been deprecated in favor of the `<ul></ul>` tag.
- **<noembed></noembed>**  
The noembed element has been deprecated in favor of the `<object></object>` tag.
- **<s></s>**  
The s element has been deprecated in favor of the `text-decoration` CSS property.

- **<shadow></shadow>**  
The `shadow` element is a MSN TV/WebTV element that has been deprecated in favor of the `text-shadow` style property.
- **<strike></strike>**  
The `strike` element has been deprecated in favor of the `text-decoration` CSS property.
- **<u></u>**  
The `u` element has been deprecated in favor of the `text-decoration` CSS property.

## Deprecated HTML Attributes

- **alink**  
The `alink` attribute defines the color of an active link on the web page. It has been deprecated in favor of the `color` style property assigned to the `:active` pseudo-class.
- **align**  
Used to align elements vertically and horizontally on the document page, the `align` attribute has been deprecated in favor of the `float` CSS style property.
- **background**  
The `background` attribute is used to define a background image for the element. It has been deprecated in favor of the `background-image` style property.
- **border**  
Used to define a border around the element, the `border` attribute has been deprecated in favor of the `border` style property.
- **color**  
Used to define the color of the enclosed text, the `color` attribute has been deprecated in favor of the `color` property.
- **compact**  
The `compact` attribute is used in list tags to create a list that takes up less space. It has been deprecated in favor of the `list-style` property.
- **face**  
Used to define the font face of the enclosed text, the `face` attribute has been deprecated in favor of the `font-family` CSS style property.
- **height**  
Used to define the height of the element, the `height` attribute has been deprecated in favor of the `height` CSS property.
- **language**  
The `language` attribute is used to define the language used by the script element. It has been deprecated in favor of the `type` attribute.
- **link**  
The `link` attribute defines the color of a link on the Web page. It has been deprecated in favor of the `color` style property assigned to the `:link` pseudo-class.
- **name**  
The `name` attribute names the element for use with dynamic content, it has been deprecated in favor of the `id` attribute.
- **noshade**  
The `noshade` attribute removes the 3-D effect from horizontal rules. It has been deprecated in favor of style sheets.

- **nowrap**  
Used to stop table contents from wrapping, the `nowrap` attribute has been deprecated in favor of the `white-space` style property.
- **size**  
Used to define the size of the enclosed text, the `size` attribute has been deprecated in favor of `font-size` CSS property.
- **start**  
The `start` attribute of list tags defines the starting number of the list. This attribute has been deprecated in favor of the `list-style-type` property.
- **text**  
The `text` attribute defines the color of the text on the Web page. It has been deprecated in favor of the `color` style property.
- **type**  
The `type` attribute of list tags defines what type of list should be used. This attribute has been deprecated in favor of the `list-style-type` property.
- **value**  
The `value` attribute of list tags sets the value of the list item. This attribute has been deprecated in favor of the `list-style` CSS property.
- **version**  
The `version` attribute defines the version of HTML used by the document. It has been deprecated in favor of defining the `DOCTYPE`.
- **vlink**  
The `vlink` attribute defines the color of a visited link on the Web page. It has been deprecated in favor of color style property assigned to the `:visited` pseudo-class.
- **width**  
The `width` attribute defines the width in pixels of the element. It has been deprecated in favor of the `width` property.

## Use CSS Instead of Deprecated HTML

- `<basefont>`  
You can change the color, face, and size of the fonts on your Web page with CSS. Create a base font for your body and paragraph tags with the following CSS:  

```
body, p
{
  font: 1em Geneva;
  color: #000;
}
```

  
This will create a page with a 1em black Geneva font.
- `<blackface>` (WebTV)  
In WebTV/MSNTV browsers the `blackface` tag makes the text double-weight bold. This is easy with CSS:

```
.blackface { font-weight: bolder; }
```

Not all browsers support the `bolder` designation, but they will all turn it bolder than the surrounding text.

- `<font>`  
Just like base font, you can specify the font face, color, and size with CSS. You can also define the background color behind the font, the line-height of the font, and the tracking of the font:

```
body, p
{
  font: 1em Geneva;
  color: #000;
  background-color: #ccc;
  line-height: 1.2em;
  letter-spacing: .2em;
}
```

### Text Attributes

Some of these aren't deprecated (like `<b>` and `<i>`) but it is still possible, and often better, to use style sheets for them. These affect how the text will look, not just details about the font.

- `<b>`  
Bold text is created with the `font-weight` property in CSS.  

```
.bold { font-weight: bold; }
```
- `<i>`  
Italic text is created with the `font-style` property in CSS.  

```
.italics { font-style: italic; }
```
- `<shadow>` (WebTV)  
The shadow tag in WebTV creates a drop-shadow on the text. There is a CSS 2 property to support shadows, but currently it's not supported by any browser.  

```
.shadow { text-shadow: .2em .2em #ccc; }
```
- `<strike>`  
You can do more than strike-through your text with CSS. Use the `text-decoration` property to do even more.

```
.strike { text-decoration: line-through; }
```

### Affect More than Text and Fonts

CSS fixes more than just text and fonts. You can also do things like indenting, centering, and layering with CSS properties.

- `<blockquote>` to indent text  
You shouldn't indent with the blockquote tag. That tag is only for long quotations.

```
.indentall { margin: 15px; }

.indentleft { margin-left: 15px; }
```

- `<center>`  
Centering in CSS is a bit trickier, because you can only center inline content. Block content you need to center in a different way.
- `<layer>`  
Layers are the trickiest because there isn't a specific property to replace them. Instead, you should define the styles of your elements and then move them around with positioning CSS properties. Use the z-index property to move your elements higher or lower in the stack order.

```
.center { text-align: center; }
```

```
.onTop { z-index: 100; }
```

```
.onBottom { z-index: 0; }
```

## CSS Rules, Options and Settings

**SIZES:** These apply to character, line, image, row, column, and page sizes (basically any element). Some of these measurements should be used for print style sheets, not for display.

%	percentage (of current size)
in	inch
cm	centimeter
mm	millimeter
em	1em is equal to the current font size. ("M Space" based on the <i>traditional print</i> size of the upper-case or capital letter "M" in the current font. 2em means 2 times the size of the current font. e.g., if an element is displayed with a font of 12 pt, then '2em' is 24 pt. The 'em' is a very useful unit in CSS, since it can adapt automatically to the font that the reader uses. An "em dash" is a long dash or hyphen.)
ex	one ex is the x-height of a font (x-height is usually about half the font-size based on the size of a lowse-case "x")
pt	point. Normally row height (1pt is the same as 1/72 inch, i.e. 72pt = 1in). Mechanical typewriters used 10pt aka "Courier" 12pt aka "Elite" and later, electronic typewriters and printers had 16.66pt or 17pt aka "Condensed". <i>Traditional print sizes are:</i>
	6pt, 8pt      footnotes, attributions
	10pt, 12pt    captions, text
	14pt, 18pt    sub-titles
	24pt, 36pt    titles
	48pt, 60pt    headlines
	72pt            banner headlines
pc	pica. Normally column width (1pc is the same as 12 points, i.e. 6pc = 1in)

*Traditional print sizes are:*

11pc	Standard newspaper columns (2in with margins or “gutters”)
22pc	Standard magazine or feature columns (4in with margins)
51pc	Standard U.S. Letter-size width (8.5in)
66pc	Standard U.S. Letter-size height (11in)
84pc	Standard U.S. Legal-size height (14in)

px      pixels (a dot on the computer screen)

**LOCATIONS:** These apply to borders, padding, margins, etc. when not specifically listed out in the style sheet. Locations relative to an element are in “clockwise” order: TOP RIGHT BOTTOM LEFT

example:

```
margin: 10px 1.5em 14pt 20px;
```

is equivalent to:

```
margin-top: 10px;  
margin-right: 1.5em;  
margin-bottom: 14pt;  
margin-left: 20px;
```

**BORDERS:** Several options (supported by most browsers) which apply to any element (div, img, table, etc.)

```
solid none hidden inherit dotted dashed groove ridge inset outset  
double
```

**LINKS:** Link rules (shown here as “pseudo-classes”) should assigned in order: link visited hover active as CSS rules and in style sheets.

```
.lnkClass  
{  
    color: blue;  
}  
.lnkClass a:link  
{  
    text-decoration: underline;  
}  
.lnkClass a:visited  
{  
    color: black;  
}  
.lnkClass a:hover  
{  
    color: red;  
}
```

```
.lnkClass a:active
{
    color: orange;
}
```

**POSITIONS:** To center elements horizontally, ensure enclosing element is a “block”, set margins to “auto”, and establish some sort of width (including 100% of enclosure width)

```
.horz
{
    display: block;
    margin-left: auto;
    margin-right: auto;
    width: 8em; (or ???)
}
```

To center or otherwise position text horizontally within a surrounding element or enclosure, set the alignment.

```
text-align: center; (also left, right, justified)
```

To center elements vertically, ensure enclosing element is a “table cell” (which does allow vertical positioning), try also using the new `vertical-align` rule, and establish some sort of height (including 100% of enclosure height)

```
.vert
{
    display: table-cell;
    vertical-align: middle;
    min-height: 10em; (or 12em, or ???)
}
```

To center or otherwise position text vertically within a surrounding element or enclosure, set the alignment.

```
vertical-align: middle;
```

Standardizing on the accepted and expected HTML instructions will make Web pages uniform across multiple browsers and operating systems and will make editing even unfamiliar pages easier for Web developers.

-- Michael Foster <mfoster@stcllp.com>