B
APPENDIX

CSS1 and CSS2 Reference

Cascading style sheets, covered in Chapters 10 and 11, offer a powerful new tool for Web page layout. When used properly, style sheets separate style from document structure, as was originally intended for HTML. Most style properties defined by the Cascading Style Sheets 1 (CSS1) specification are supported by major browsers. However, many of the properties in the CSS2 specification are not. A full listing of CSS1 is presented in this appendix. The visual properties of CSS2 are presented in their entirety; however, the aural properties are only briefly summarized given their lack of use. The appendix concludes with an overview of some of the proprietary CSS properties supported by Internet Explorer, some of which appear slated for inclusion in CSS3.

Style Inclusion Methods

This section defines the basic methods to associate CSS-based style information with HTML or XHTML documents.

Embedded Styles

Document-wide styles can be embedded in a document's **head** element using the **<style>** tag. Note that styles should be commented out to avoid interpretation by nonstyle-aware browsers. However, be aware that comment masking is frowned upon in XHTML and linked styles should be used or a CDATA section employed.

Example

```
<style type="text/css">
<!--
p  {font-size: 14pt; font-face: Times; color: blue;
    background-color: yellow;}
em {font-size: 16pt; color: green;}
-->
</style>
```

Inline Styles

You can apply styles directly to elements in a document using the core attribute **style**. As the closest style inclusion method to a tag, inline styles will take precedence over document wide or linked styles.

Example

```
<h1 style="font-size: 48pt; font-family: Arial;
           color: green;">CSS1 Test</h1>
```

Linked Styles

Styles can be contained in an external style sheet linked to a document or a set of documents (see Chapter 10), as shown in the following example. Linked information should be placed inside the **<head>** tag.

Example

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

The **rel** attribute is generally set to the value **stylesheet** but may also have a value of **alternate stylesheet** with an associated *title* value to provide different looks for the same page.

Examples

```
<link rel="stylesheet" href="standard.css" title="standard" />
<link rel="alternate stylesheet" href="bigred.css" title="Red Sheet" />
```

Chapter 11 has examples and more information on using alternative style sheets.

The **media** attribute may also be used to define the media to which a style sheet is applied. The keyword values **screen** or **print** are commonly used. The default value of **all** is applied when media is not specified.

Examples

```
<link rel="stylesheet" href="screenstyle.css" media="screen"
      type="text/css" />
<link rel="stylesheet" href="printstyle.css" media="print"
      type="text/css" />
```

CSS2 does define a rich set of media values as shown in Table B-1, but practice shows few are supported.

Imported Styles

Styles can be imported from an external file and expanded in place, similar to a macro. Importing can be used to include multiple style sheets. An imported style is defined within a **<style>** tag using **@import** followed optionally by a **type** attribute and a URL for the style sheet.

Media Type	Definition
all	For use with all devices
aural	For use with speech synthesizers
Braille	For use with tactile Braille devices
embossed	For use with Braille printers
handheld	For use with handheld devices
print	For use with printed material and documents viewed onscreen in print preview mode
projection	For use with projected media (direct computer-to-projector presentations), or printing transparencies for projection
screen	For use with color computer screens
tty	For use with low-resolution teletypes, terminals, or other devices with limited display capabilities
tv	For use with television-type devices

TABLE B-1 Media Types Defined Under CSS2

Example

```
<style type="text/css">
@import url(newstyle.css)
</style>
```

The **@import** directive allows style sheets to be grouped and joined together. While this was the design of the feature, unfortunately most CSS developers use it to perform a weak form of browser selection because many older CSS implementations do not support the directive. The basic idea of the trick is to put sophisticated style rules in an **@import** style sheet and leave basic styles in the style block. This trick should be avoided, particularly given that some browsers, notably versions of Internet Explorer, will cause a disturbing flash effect when loading imported styles.

CSS Measurements

CSS supports a number of measurements including absolute units such as inches, centimeters, points, and so on, as well as relative measures such as percentages and em units. Table B-2 summarizes these units of measure.

CSS Color Values

Style sheets support a variety of color measurement values, as shown in Table B-3. Appendix E provides a greater discussion of possible color values and names.

Measurement	Description	Example

%	Defines a measurement as a percentage relative to another value, typically an enclosing element.	p {font-size: 14pt;
		line-height: 150%;}

cm	Defines a measurement in centimeters.	div {margin-bottom: 1cm;}
em	A relative measurement for the height of a font in em spaces. Because an **em** unit is equivalent to the size of a given font, if you assign a font to 12pt, each "**em**" unit would be 12pt; thus, 2em would be 24pt.	p {letter-spacing: 5em;}

ex	This value defines a measurement relative to a font's x-height. The x-height is determined by the height of the font's lowercase letter *x*.	p {font-size: 14pt;
		line-height: 2ex;}

in	Defines a measurement in inches.	p {word-spacing: .25in;}
mm	Defines a measurement in millimeters.	p {word-spacing: 12mm;}
pc	Defines a measurement in picas. A pica is equivalent to 12 points; thus, there are 6 picas per inch.	p {font-size: 10pc;}
pt	Defines a measurement in points. A point is defined as 1/72nd of an inch.	body {font-size: 14pt;}
px	Defines a measurement in screen pixels.	p {padding: 15px;}

TABLE B-2 CSS1 and CSS2 Length Measurement Units

Color Format	Description	Example

Specification-defined named colors	CSS color values can be defined using 16 color names: **aqua**, **black**, **blue**, **fuchsia**, **gray**, **green**, **lime**, **maroon**, **navy**, **olive**, **purple**, **red**, **silver**, **teal**, **white**, and **yellow.**	body {font-family: Arial;
		font-size: 12pt;
		color: red;}

Browser-defined named colors	Browsers define a variety of their own colors like **mintcream**. Appendix E provides a complete list of these extended colors, which should be avoided because of potential browser compatibility problems.	#gap {color: khaki;}

Six-digit hexadecimal	CSS's six-digit hexadecimal format is the same as HTML/XHTML. The format specifies color as #RRGGBB where RR is the amount of red, GG the amount of green, and BB the amount of blue all specified in a hexadecimal value ranging from 00 to FF.	div {font-family: Courier;
		font-size: 10pt;
		color: #00CCFF;}

Three-digit hexadecimal	The three-digit hexadecimal format is unique to CSS and specifies color in the format of #RGB where R,G, and B are hexadecimal ranging from 0 to F. Given the lesser number of digits, it is obviously less expressive than the six-digit format.	span {font-family: Helvetica;
		font-size: 14pt;
		color: #0CF;}

RGB	CSS colors can also be defined using the keyword **rgb***,* followed by three numbers between 0 and 255, contained in parentheses and separated by commas, with no spaces between them. RGB color values can also be defined using percentages. The format is the same, except that the numbers are replaced by percentage values between **0%** and **100%**.	#p1 {color: rgb(204,0,51);}
		p {color: rgb(0%,10%,50%);}

TABLE B-3 CSS1 and CSS2 Color Measurement Units

TIP *For maximum compatibility with older CSS implementations, developers are encouraged to use known name values or the six-digit hexadecimal format when specifying color.*

Selectors

CSS1 and 2 support a rich set of selectors for specifying which particular element(s) that a CSS rule applies to. CSS1 selectors are presented in Table B-4.

CSS2 introduces a number of new selectors as summarized in Table B-5. Many of these selectors use document context to determine how styles should be applied to elements, potentially reducing reliance on HTML/XHTML selectors such as **class** and **id**. For more information, see Chapter 11.

Selector	Description	Example
element	Selects all elements of the name specified in the rule.	h1 {color: red;} /* makes all h1 tags red */
#id	Selects any tag with an **id** attribute set.	#test {color: green;} /* makes a tag with id='test' green */
.class	Selects any tag with the specified class value.	.note {color: yellow;} /* makes all tags with class='note' yellow */
element.class	Selects the specified elements with a particular class value.	h1.note {text-decoration: underline;} /* underlines all H1 tags with class='note' */
Grouping	Applies the same rules to a group of tags.	h1,h2,h3 {background-color: orange;} /* sets the background color of all h1,h2, and h3 elements to orange */
Contextual	Selects descendent tags.	p strong {color: purple;} /* sets all strong tags that are descendents of p tags purple */
:first-line	Selects the first line of an element.	p:first-line {color: red;} /* makes the first lines of a paragraph red */
:first-letter	Selects the first letter of an element.	p:first-letter {font-size: larger;} /* makes the first letter of a paragraph larger */
a:link	Specifies the unvisited link.	a:link {font-weight: bold;} /* makes unvisited links bold */
a:active	Specifies the link as it is being pressed.	a:active {color: red;} /* makes links red as they are pressed */
a:visited	Specifies the link after being pressed.	a:visited {text-decoration: line-through;} /* puts a line through visited links */

TABLE B-4 CSS1 and CSS2 Color Measurement Units

Selector	**Description**	**Examples**

*	The wildcard selector is used to apply a match to any element. It can be used for a global rule or, more commonly, in contextual or child selection rules.	* {background-color: red;}
		div * span {background-color: yellow;}

>	This selector defines a rule that matches only elements that are directly enclosed within another element, such as a **<p>** tag with a document body.	body > p {background-color: yellow;}
+	The adjacent sibling selector defines a rule that applies a style to the first incidence of an element immediately after the first element. In other words, the two tags are adjacent siblings in their parse tree.	h1 + p {color: red;}

[ ]	The attribute selector has many uses. Unfortunately, many browsers do not support it. The basic inclusion of an attribute name in brackets matches when the attribute is used on the selected element. A specific attribute value can be matched with = and pieces of an attribute can also be matched. The symbol ~= can be used to match space-separated attribute values while |= is used to match dash-separated attribute values. It is also possible to match multiple attribute values at once.	a[href] {background-color: yellow;}
		a[href="http://www.htmlref.com"] {font-weight: bold;}
		p[title~="Test match"] { font-style: italic; }
		p[lang|="en"] { color: red; } /* English text in red */
		p[title="Test Selector"][lang|="en"] {border-style: dashed; border-width: 1px; }

@media	Defines style rules for multiple media types in a single style sheet. See "Media-Dependent Style Sheets" in Chapter 11.	@media screen {body
		{font-family: sans-serif;
		font-size: 18 pt;}
		}

@page	Used to define rules for page sizing and orientation rules for printing.	@page {size: 8.5in 11in;}
:left	Sets page layout rules for a left-hand page when printing.	@page :left {margin-left: 4cm; margin-right: 3cm;}
:right	Sets page layout rules for a right-hand page when printing.	@page :right {margin-left: 3cm; margin-right: 4cm;}
:first	Sets page layout rules for the first page in a document when printing.	/* Top margin on first page 10cm */ @page :first {margin-top: 10cm;}
:first-child	Applies a style to the first child element of an element.	p:first-child {color: red;}
:focus	Changes the display of an element when the element receives focus (generally, **<input="text"**>).	input:focus{background-color: yellow;}

:hover	Changes the display of an element when a cursor passes over the element. Commonly supported for links, but defined for nearly every element.	a:hover {text-decoration: underline;}
		p:hover {background-color: yellow;}

:lang	Applies style to an element according to what language the element is in.	*:lang(fr) {color: blue;}
		*:lang(en) {color: red;}

:before	Defines content to be placed before an element. See the **content** property for more information on use.	div:before {content: url(sectionstart.gif);}
:after	Defines content to be placed after an element. See the **content** property for more information on use.	div:after {content: url(sectionend.gif);}

Table B-5 CSS2 Selector Summary

TIP *Developers should proceed with caution when using CSS2 selectors; many of the more complex ones are buggy or not supported even in modern browsers. However, they should not be avoided because when supported they can be used to create very powerful rules.*

Miscellaneous CSS Constructs

This section discusses some miscellaneous constructs associated with style sheets.

/* comments */

Comments can be placed within style sheets. HTML comment syntax (**<!-- comment -->**) does not apply. However, HTML comments are often used to mask style blocks. Style sheets use the comment syntax used in C programming (**/*comment*/**).

Example

```
<style type="text/css">
p {font-face: Courier; font-size: 14pt; font-weight: bold;
   background-color: yellow;}
/*This style sheet was created at Demo Company, Inc.
  All rights reserved.*/
</style>
```

! Important

This property specifies that a style takes precedence over any different, conflicting styles. A style specified as important by an author takes precedence over a rule set by an end user. This construct should be used sparingly.

Example

```
div {font-size: 14pt; line-height: 150%; font-family: Arial ! important;}
```

CSS1 and CSS2 Properties

This section presents the CSS1 and 2 properties in alphabetical order. Readers should note that the properties tend to come in groups and that most groups have shorthand notation. For example, the **background** property is shorthand for **background-color**, **background-image**, **background-position**, and **background-attachment**. The primary property for a set may contain extra details that should be noted.

background

This property sets any or all background properties in a shorthand form. Property order should not matter but the typical syntax is

```
background:  background-color  background-image  background-repeat
             background-attachment  background-position;
```

Any properties not specified use their default values. As with all shorthand forms, document authors should experiment with individual property values before condensing to a short form.

Examples

```
body {background: white url(picture.gif) repeat-y center;}
.red {background: #ff0000;}
#div1 {background: white url(logo.gif) no-repeat fixed 10px 10px;}
```

Browser and CSS Support Notes

CSS1	IE 4, 5, 5.5, 6	Nav 4, 4.5–4.8 (buggy), 6, 7	Opera 4, 5, 6, 7

background-attachment

This property sets the background image to scroll or not to scroll with its associated element's content. The default value is **scroll**, which sets the background to scroll with the associated content, typically text. The alternate value, **fixed**, is intended to make the background static with associated content such as text scrolls on top. This property is often used to create a watermark effect similar to the proprietary attribute, **bgproperties**, of the **<body>** tag introduced by Microsoft.

Examples

```
body {background-image: url(tile.gif); background-attachment: scroll;}
#logo {background-image: url(logo.gif); background-attachment: fixed;}
```

Browser and CSS Support Notes

CSS1	IE 4, 5, 5.5 ,6	Nav 6, 7	Opera 4, 5, 6, 7

background-color

This property sets an element's background color. It is often used in conjunction with the **color** property, which sets text color. If both are not set it is possible to have rendering problems and the W3C CSS validator will warn of this issue. Used with block elements, this property colors content and padding but not margins. The default value, **transparent**, allows any underlying content to show through. See Appendix E for browser support of color values.

Examples

```
p    {background-color: #00CCFF;}
body {background-color: orange;}
.red {background-color: rgb(255, 0, 0;}
```

Browser and CSS Support Notes

CSS1	IE 4, 5, 5.5, 6	Nav 4, 4.5–4.7 (buggy; may not fit entire region), 6, 7	Opera 4, 5, 6, 7

background-image

This property associates a background image with an element. Underlying content may show through transparent regions in the source image. The background image requires a URL (complete or relative) to link it to the source image specified with the **url( )** syntax. The default value is **none** and sets the background so that it doesn't display an image.

Examples

```
body   {background-image: url(yellowpattern.gif);}
p      {background-image: none;}
.robot {background-image: url(http://www.democompany.com/images/robot.gif);}
```

Browser and CSS Support Notes

CSS1	IE 4, 5, 5.5, 6	Nav 4, 4.5–4.7 (buggy; may not fit entire region), 6, 7	Opera 4, 5, 6, 7

background-position

This property determines how a background image is positioned within the canvas space used by its associated element. The position of the background image's upper-left corner can be specified as an absolute distance in pixels from the surrounding element's origin. It can also be specified as a percentage along the horizontal and vertical dimensions. Finally, the position can be specified as named values that describe the horizontal and vertical dimensions. The named values for the horizontal axis are **center**, **left**, and **right**; those for the vertical axis are **top**, **center**, and **bottom**. The default value for an unspecified dimension is assumed to be **center**.

Examples

```
body  {background-image: url(yellowpattern.gif);
       background-position: 50px 100px;}
#div1 {background-image: url(bricks.gif); background-position: 10% 45%;}
body  {background-image: url(logo.gif); background-position: top center;}
```

Browser and CSS Support Notes

CSS1	IE 4, 5, 5.5, 6	Nav 6, 7	Opera 4, 5, 6, 7

background-repeat

This value determines how background images tile when they are smaller than the canvas space used by their associated elements. It is used in conjunction with the **background-image** property. Possible values are **repeat** (repeats in both direction), **repeat-x** (repeats only horizontally), **repeat-y** (repeats vertically), and **no-repeat**. The default value is **repeat**.

Examples

```
body  {background-image: url(yellowpattern.gif) background-repeat: repeat;}
#div1 {background-image: url(tile.gif); background-repeat: repeat-x;}
p     {background-image: url(tile2.gif); background-repeat: repeat-y;}
.mark {background-image: url(logo.png); background-repeat: no-repeat;}
```

Browser and CSS Support Notes

CSS1	IE 4, 5, 5.5, 6	Nav 4, 4.5–4.8 (buggy may not fit entire region), 6, 7	Opera 4, 5, 6, 7

border

This property defines the width, style, and color for all four sides of an element's border in a shorthand form. There are five specific properties for setting the width of borders: **border-top-width**, **border-bottom-width**, **border-right-width**, **border-left-width**, and **border-width**. The first four set the width of specific borders; **border-width** is used to set all four. The **border** property by itself can set all of them. Values for border widths can be set in numeric measurements or with the named values **thin**, **medium**, or **thick**. Border colors and styles can be set with the properties **border-color** and **border-style**, respectively, as well as with the shorthand. The properties **border-top**, **border-bottom**, **border-right**, and **border-left** can be used to set width, style, and color values for different sides of a border. Given all these properties, the **border** property sets the width, style, and color of all sides of an element's border, typically in that order. The syntax is shown here:

```
border: border-width border-style border-color;
```

Given the complexity of the rule, like many shorthand properties, individual properties should probably be set and then condensed to shorthand form.

Examples

```
div     {border: 2px double red;}
.dashed {border: .5em dashed #f00;}
```

Browser and CSS Support Notes

CSS1,	IE 4, 5 (buggy), 5.5, 6	Nav 4, 4.5–4.8 (buggy), 6, 7	Opera 5, 6, 7

border-bottom

This property defines the width, style, and color for the bottom border of an element in a shorthand form. The syntax for the rules should be in order, as follows:

```
border-bottom: border-width border-style border-color;
```

Given that CSS1 did not support **border-bottom-color** and **border-bottom-style,** this property is useful for setting the characteristics of the bottom of boxes for older browsers.

Example

```
#redbottom  {border-top: thin solid red;}
```

Browser and CSS Support Notes

CSS1	IE 4, 5, 5.5, 6	Nav 6, 7	Opera 4, 5, 6, 7

border-bottom-color

This property defines the color of an element's bottom border. See Appendix E for information on browser support of color values.

Example

```
p {border-style: solid; border-width: thin; border-bottom-color: orange;}
```

Browser and CSS Support Notes

CSS2	IE 4, 5, 5.5, 6	Nav 6, 7	Opera 7

border-bottom-style

This property defines the style for the bottom border of an element. Allowed values are **none**, **dotted**, **dashed**, **solid**, **double**, **groove**, **ridge**, **inset**, and **outset**. See **border-style** property for specific information on each style.

Example

```
#box {border-width: 10px; border-style: solid; border-bottom-style: double;}
```

Browser and CSS Support Notes

CSS2	IE 4, 5, 5.5, 6	Nav 6, 7	Opera 7

border-bottom-width

This property sets the width of an element's bottom border. Values can be keywords (**thin**, **medium**, or **thick**) and numerical lengths such as pixels (**px**), inches (**in**), and so on.

Examples

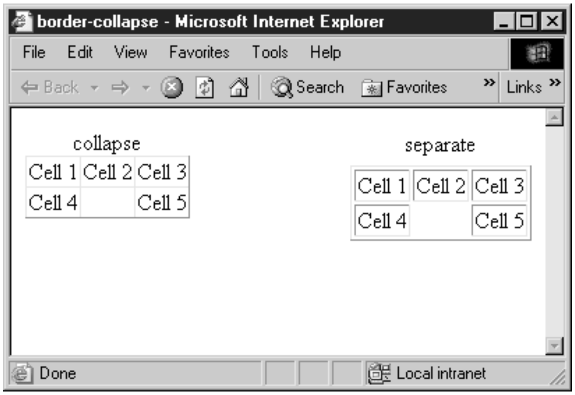
```
.low {border-bottom-width: thick;}
p    {border-bottom-width: 15px;}
```

Browser and CSS Support Notes

CSS1	IE 4, 5 (buggy), 5.5, 6	Nav 4, 4.5–4.8 (buggy), 6, 7	Opera 5, 6, 7

border-collapse

This CSS2 property defines if table cell borders are connected or separate. Allowed values are **separate** and **collapse**. With a value of **collapse**, the borders appear to collapse on each other so that there's no more spacing between the borders. A rendering here in Internet Explorer 6 should illustrate the idea of the property.



collapse

Cell 1	Cell 2	Cell 3
Cell 4		Cell 5

separate

Cell 1	Cell 2	Cell 3
Cell 4		Cell 5

Example

```
<table border="1" style="border-collapse: collapse;">
<tr>
 <td>Cell 1</td><td>Cell 2</td><td>Cell 3</td>
</tr>
 <tr>
   <td>Cell 4</td><td></td><td>Cell 5</td>
</tr>
</table>
```

Browser and CSS Support Notes

CSS2	IE 5, 5.5, 6 (buggy)	Nav 6, 7	Opera 5, 6, 7

TIP *There are significant rendering differences in browsers even when the border-collapse property is supported.*

border-color

This property defines the color of an element's border. All borders are set at once but individual color values can be set with the shorthand **border-top**, **border-right**, **border-bottom**, and **border-left**. See Appendix E for information on browser support of color values.

Examples

```
p   {border-style: solid; border-width: thin; border-color: blue;}
#d1 {border-style: double; border-color: #0000EE;}
```

Browser and CSS Support Notes

CSS1	IE 4, 5 (buggy), 5.5, 6	Nav 4, 4.5–4.8 (buggy), 6, 7	Opera 4, 5, 6, 7

border-left

This property defines the width, style, and color for the left border of an element in a shorthand form. The syntax for the rules should be in order, as follows:

```
border-left: border-width border-style border-color;
```

Given that CSS1 did not support **border-left-color** and **border-left-style**, this property is useful for setting the characteristics of the left of boxes for older browsers.

Example

```
#leftout  {border-left: thin dashed red;}
```

Browser and CSS Support Notes

CSS1	IE 4, 5, 5.5, 6	Nav 6, 7	Opera 4, 5, 6, 7

border-left-color

This property defines the color of an element's left border. See Appendix E for information on browser support of color values.

Example

```
p {border-style: solid; border-width: thin; border-left-color: red;}
```

Browser and CSS Support Notes

CSS2	IE 4, 5, 5.5, 6	Nav 6, 7	Opera 7

border-left-style

This property defines the style for the left border of an element. Allowed values are **none**, **dotted**, **dashed**, **solid**, **double**, **groove**, **ridge**, **inset**, and **outset**. See **border-style** property for specific information on each style.

Example

```
#box {border-width: 5px; border-style: solid; border-left-style: dotted;}
```

Browser and CSS Support Notes

CSS2	IE 4, 5, 5.5, 6	Nav 6, 7	Opera 7

border-left-width

This property sets the width of an element's left border. Values can be keywords (**thin**, **medium**, or **thick**) and numerical lengths such as pixels (**px**), inches (**in**), and so on.

Examples

```
.thin {border-style: dashed; border-left-width: thin;}
p {border-style: solid; border-left-width: 5px;}
```

Browser and CSS Support Notes

CSS1	IE 4, 5 (buggy), 5.5, 6	Nav 4, 4.5–4.8 (buggy), 6, 7	Opera 5, 6, 7

border-right

This property defines the width, style, and color for the bottom border of an element in a shorthand form. The syntax for the rules should be in order, as follows:

`border-right`: *`width style color`*;

Given that CSS1 did not support **border-right-color** and **border-right-style**, this property is useful for setting the characteristics of the right of boxes for older browsers.

Example

```
#greenzone  {border-right: thick dashed green;}
```

Browser and CSS Support Notes

CSS1	IE 4, 5, 5.5, 6	Nav 6, 7	Opera 4, 5, 6, 7

border-right-color

This property defines the color of an element's right border. See Appendix E for information on browser support of color values.

Example

```
p {border-style: solid; border-width: thin; border-right-color: green;}
```

Browser and CSS Support Notes

CSS2	IE 4, 5, 5.5, 6	Nav 6, 7	Opera 7

border-right-style

This property defines the style for the right border of an element. Allowed values are **none**, **dotted**, **dashed**, **solid**, **double**, **groove**, **ridge**, **inset**, and **outset**. See **border-style** property for specific information on each style.

Example

```
#box {border-width: 5px; border-style: solid; border-right-style: ridge;}
```

Browser and CSS Support Notes

CSS2	IE 4, 5, 5.5, 6	Nav 6, 7	Opera 7

border-right-width

This property sets the width of an element's right border. Values can be keywords (**thin**, **medium**, or **thick**) and numerical lengths such as pixels (**px**), inches (**in**), and so on.

Examples

```
div         {border-right-width: medium;}
.superfat   {border-right-width: 40px;}
```

Browser and CSS Support Notes

CSS1	IE 4, 5 (buggy), 5.5, 6	Nav 4,4.5–4.8 (buggy), 6, 7	Opera 5, 6, 7

border-spacing

This CSS2 property is similar to the **cellspacing** attribute on the **table** element in HTML/XHTML and defines the space between cells in a table. Its value can be an arbitrary length, but not negative. If one length is specified, it gives both the horizontal and vertical spacing. If two are specified, the first gives the horizontal spacing and the second the vertical spacing between cells.

Examples

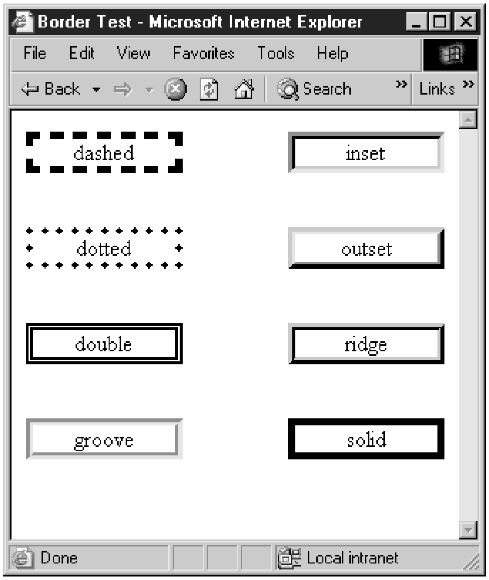
```
table        {border-spacing: 10px;}
#table2      {border-spacing: 10px 5px;}
```

Browser and CSS Support Notes

CSS2	No IE support	Nav 6, 7	Opera 5, 6, 7

border-style

The **border-style** property defines the style of up to four different sides of a border, using the values **none**, **dotted**, **dashed**, **solid**, **double**, **groove**, **ridge**, **inset**, and **outset**. A value of **none** overrides any borders currently set. **Dashed** and **dotted** values are distinctly different but many browsers will set them the same. **Solid** is the default value. **Double** sets a double line border. **Groove** sets an edge line style border while a **ridge** value sets the border to resemble a raised ridge by reversing the shading of the grooved rendering. An **inset** value sets the border to display a lighter shade of the border color on its right and bottom sides while an **outset** value sets the border to display a lighter shade of the border color on its top and left sides. Examples of these styles are shown here:



The shorthand style allows individual borders to be set. A single value copies to all border sides. With two values, the first sets the border style of top and bottom, and the second sets the right and left values. With three values, the first sets the top style, the second sets the right and left, and the third sets the bottom style. With four values, each is set individually in the order top, right, bottom, and left. In general, missing values are inferred from the value defined for the opposite side.

Examples

```
p           {border-style: solid;}
.twosides   {border-style: dashed solid;}
.allsides   {border-style: solid dashed groove inset;}
```

Browser and CSS Support Notes

CSS1	IE 4, 5 (no dotted/dashed), 5.5, 6	Nav 4, 4.5–4.8 (buggy), 6, 7	Opera 5, 6, 7

NOTE *Netscape 4 supports only one value for* ***border-style****. Use of multiple values will create erratic display under that browser.*

border-top

This property defines the width, style, and color for the top border of an element in a shorthand form. The syntax for the rules should be in order, as follows:

border-top: *border-width border-style border-color*;

Given that CSS1 did not support **border-top-color** and **border-top-style**, this property is useful for setting the characteristics of the top of boxes for older browsers.

Example

```
#boxtop  {border-top: thin solid blue;}
```

Browser and CSS Support Notes

CSS1	IE 4, 5, 5.5, 6	Nav 6, 7	Opera 4, 5, 6, 7

border-top-color

This property defines the color of an element's top border. See Appendix E for information on browser support of color values.

Example

```
p {border-style: solid; border-width: thin; border-top-color: red;}
```

Browser and CSS Support Notes

CSS2	IE 4, 5, 5.5, 6	Nav 6, 7	Opera 7

border-top-style

This property defines the style for the top border of an element. Allowed values are **none**, **dotted**, **dashed**, **solid**, **double**, **groove**, **ridge**, **inset**, and **outset**. See **border-style** property for specific information on each style.

Example

```
#box {border-width: 1px; border-style: solid; border-top-style: dashed;}
```

Browser and CSS Support Notes

CSS2	IE 4, 5, 5.5, 6	Nav 6, 7	Opera 7

border-top-width

This property sets the width of an element's top border. Values can be keywords (**thin**, **medium**, or **thick**) and numerical lengths such as pixels (**px**), inches (**in**), and so on.

Examples

```
p         {border-top-width: thin;}
#thicktop {border-top-width: 25px;}
```

Browser and CSS Support Notes

CSS1	IE 4, 5 (buggy), 5.5, 6	Nav 4, 4.5–4.8 (buggy), 6, 7	Opera 5, 6, 7

border-width

This property sets the width of an element's complete border. Values can be keywords (**thin**, **medium**, or **thick**) and numerical lengths. The **border-width** property can also be used to specify all four borders individually in the standard top, right, bottom, left style. A single value copies to all border sides. With two values, the first sets the border width of top and bottom, and the second sets the right and left values. With three values, the first sets the top width, the second sets the right and left, and the third sets the bottom width. With four values, each is set individually in the order top, right, bottom, and left.

Examples

```
div {border-width: medium;}   /* all sides set medium */
#d1 {border-width: 10px 5px;} /* 10px top-bottom, 5px right and left */
#fun {border-width: 10px 1px 4px 50px;} /* sides set individually */
```

Browser and CSS Support Notes

CSS1	IE 4, 5 (buggy), 5.5, 6	Nav 4, 4.5–4.8 (buggy), 6, 7	Opera 5, 6, 7

bottom

This property defines the *y* (vertical) coordinate for a positioned element, relative to the bottom of the enclosing element or browser window. Values can be specified as lengths (inches, pixels, and so on), as a percentage of the containing object's dimensions, or as **auto**, which lets this property function as determined by the browser or as defined by the parent element.

Examples

```
#div1 {position: absolute; left: 100px; bottom: 150px;}
#div2 {position: absolute; left: 50%; bottom: 30%;}
#div3 {position: absolute; left: auto; bottom: auto;}
```

Browser and CSS Support Notes

CSS2	IE 5, 5.5, 6	Nav 6, 7	Opera 5, 6, 7

caption-side

This property defines the position of a **caption** element within a **table**. It may take **left**, **right**, **top** or **bottom** as a value. A value of **top** will typically default in a browser.

Examples

```
caption {caption-side: bottom;}
.right  {caption-side: right;}
```

Browser and CSS Support Notes

CSS2	No IE support	Nav 6, 7	Opera 6, 7

clear

This property specifies the placement of an element in relation to floating objects. Possible values are **left**, **right**, **both**, and **none**. The property acts much like the **clear** attribute for the **<br>** tag and continues to return until the left, right, or both columns are clear. The default value is **none**.

Examples

```
br.clearright {clear: right;}
#clearboth    {clear: all;}
```

Browser and CSS Support Notes

CSS1	IE 4, 5 (buggy), 5.5, 6	Nav 4, 4.5–4.8 (buggy), 6, 7	Opera 5, 6, 7

clip

This property sets the coordinates of the clipping rectangle that houses the content of elements. The allowed clipping shape is a rectangle defined **rect (***top right bottom left***)** where the values specify offsets from the respective sides of the containing box. Eventually, other shapes may be possible.

Example

```
#div1 {position: absolute; left: 20px; top: 20px;
       width:100px; height:100; clip: rect(10px 90px 90px 10px);}
```

Browser and CSS Support Notes

CSS2	IE 4, 5 (buggy), 5.5, 6	Nav 6, 7	Opera 5, 6, 7

color

This property sets the color of text. Values can be specified as color names, hex values in three or six-digit format, or red-green-blue (RGB) values (numbers or percentages). For more information on color values, see Appendix E.

Examples

```
.sunflower {color: yellow;}
#sunburn   {color: #FF0000;}
p          {color: #FF0;}
#sunburn2  {color: rgb(255,0,0);}
body       {color: rgb(100%,100%,100%);}
```

Browser and CSS Support Notes

CSS1	IE 3, 4, 5, 5.5, 6	Nav 4, 4.5–4.7 6, 7	Opera 4, 5, 6, 7

content

This property generates content in a document and is used with the **:before** and **:after** pseudo-elements. Values for the property include ***string***, ***url***, ***counter***, **open-quote**, **close-quote**, **no-open-quote**, **no-close-quote**, and **attr(*X*)**. The ***string*** value simply inserts the defined string either before or after the selected element, depending on the rule in use. The ***url*** value is used to insert an external resource, typically an image. ***Counter*** values can be specified and used to automatically add a sequential indicator. The values **open-quote** and **close-quote** insert quotation symbols specified by the **quotes** property or, if undefined, default to the user-agent's default quote style. The **no-open-quote** and **no-close-quote** values do not insert quotation symbols but do increment or decrement the nesting level for quotes. The **attr(*X*)** returns a string value for the attribute ***X*** for the element the rule is associated with. If the related element does not have an attribute named ***X*,** an empty string is used.

Examples

```
div.section:before {content: "Section: ";}
div:before         {content: url(sectionstart.gif);}
div:after          {content: url(sectionend.gif);}
blockquote:before  {content: open-quote;}
blockquote:after   {content: close-quote;}
p:before           {content: counter(par-num, upper-roman); ". " }
p:before           {content: attr(title); }
```

Browser and CSS Support Notes

CSS2	IE not yet supported	Nav 6, 7 (no counter support)	Opera 7

counter-increment

This property accepts one or more names of counter names, each one optionally followed by an integer. The integer indicates by how much the counter is incremented or decremented for every occurrence of the element. The default increment is 1. Zero and negative integers are allowed. By default, counters are formatted as decimal numbers, but the styles supported by the **list-style-type** property are also available for counters and are specified with the syntax **counter(*name, list-style-type*)**.

Examples

```
div.section:before {content: "Section: " counter(section) ". ";
                    counter-increment: section;}  /* Add 1 to section */
h1.chapter:before  {content: counter(chno, upper-latin) ". ";
                    counter-increment: chno; }
.topten:before     {content: counter(countdown) ". ";
                    counter-increment: countdown -1; }
```

Browser and CSS Support Notes

CSS2	IE not yet supported	No Mozilla or Nav support yet	Opera 7

counter-reset

This property contains a list of one or more counter names, each one optionally followed by an integer. The integer gives the value that the counter is set to on each occurrence of the element. The default value is 0.

Examples

```
div.section:before {content: "Section: " counter(section) ". ";
                    counter-increment: section;}  /* Add 1 to section */
div.chapter:before {counter-reset: section;}      /* Set section to 0 */
```

Browser and CSS Support Notes

CSS2	IE not yet supported	Nav and Mozilla not yet supported	Opera 7

cursor

This property determines how the cursor displays when passed over the affected element. The **auto** value leaves the display to be determined by the user agent, so the cursor will display according to either the browser default settings or the user settings. The **crosshair** value renders the cursor as a simple cross, whereas **default** displays the system's default cursor (usually an arrow). Various other values listed in the CSS2 specification can indicate that something is a link (**pointer**), that text can be selected (**text**), that something can be resized in various directions (**e-resize**, **ne-resize**, **nw-resize**, **n-resize**, **se-resize**, **sw-resize**, **s-resize**, **w-resize**), or that the user must wait while a program is busy (**wait**). Table B-6 details the cursors supported in CSS2.

The value **url** can be used to reference a cursor source; multiple cursor sources can be listed. As with fonts, the user agent should attempt to render the first cursor listed, try the second one if necessary, and ultimately default to the generic cursor value listed last.

Examples

```
.help {cursor: help;}
p.clickable {cursor: hand;} /* non-standard */
a:longload  {cursor: wait;}
p {cursor:url("mything.cur"), url("second.cur"), text; }
```

Browser and CSS Support Notes

CSS2	IE 5, 5.5, 6	Nav 6, 7	Opera 7

TIP *Internet Explorer 6 custom cursor support includes animated cursors. While custom cursors may not be supported in all browsers, a variety of JavaScript tricks are often employed to imitate this CSS2 property.*

CSS Cursor Property Values	Description	Typical Rendering
auto	The browser determines the cursor to display based on the current context.	N/A
crosshair	A simple crosshair generally resembles a plus symbol.	+
default	The browser's default cursor is generally an arrow.	↖
hand	A hand pointer (nonstandard but commonly supported).	☝
move	This indicates something is to be moved; usually rendered as four arrows together.	✥
e-resize	This indicates resizing as a double arrow pointing east-west (left-right).	↔
ne-resize	This indicates resizing as a double arrow pointing northeast-southwest.	⤢
nw-resize	This indicates resizing as a double arrow pointing northwest-southeast.	⤡
n-resize	This indicates resizing as a double arrow pointing north-south.	↕
pointer	Typically renders similar to the browser's default pointing cursor, which is generally a hand.	☝
se-resize	This indicates resizing as a double arrow pointing southeast-northwest.	⤡
sw-resize	This indicates resizing as a double arrow pointing southwest-northeast.	⤢
s-resize	This indicates resizing as a double arrow pointing north-south.	↕
w-resize	This indicates resizing as a double arrow pointing west-east.	↔

TABLE B-6 CSS2 Cursor Properties

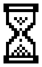
CSS Cursor Property Values	Description	Typical Rendering
text	This indicates text that may be selected or entered; generally rendered as an I-bar.	
wait	This indicates that the page is busy; generally rendered as an hourglass.	
help	This indicates that Help is available; the cursor is generally rendered as an arrow and a question mark.	

TABLE B-6 CSS2 Cursor Properties *(continued)*

direction

The **direction** property is used to control the text direction much like the attribute **dir** for various HTML/XHTML tags. The allowed values are **rtl** (right to left) and **ltr** (left to right). While the direction property can easily affect block elements, for it to affect inline-level elements, the **unicode-bidi** property value must be **embed** or **override**.

Example

```
<div><span style="unicode-bidi: embed; direction: rtl;
background-color: yellow;">here doing I am  What!
</span> This is just a test</div>
```

Browser and CSS Support Notes

CSS2	IE 5, 5.5, 6	Nav 6, 7	No Opera support

display

This property specifies an element's display type and can override an element's defined display type. For example, block-level elements can be redefined as inline elements so that extra lines will not be placed between them. The allowed values for display under CSS1 are **inline**, **block**, **list-item**, and **none**. The value of **none** completely removes an element from the document tree and unlike the **hidden** value of the **visibility** property, **none** does not preserve an element's canvas space.

CSS2 adds **run-in**, **compact**, **marker**, **table**, **inline-table**, **table-row-group**, **table-header-group**, **table-footer-group**, **table-row**, **table-column-group**, **table-column**, **table-cell**, and **table-caption**.

Examples

```
p        {display: inline;}
b        {display: block;}
.removed {display: none;}
```

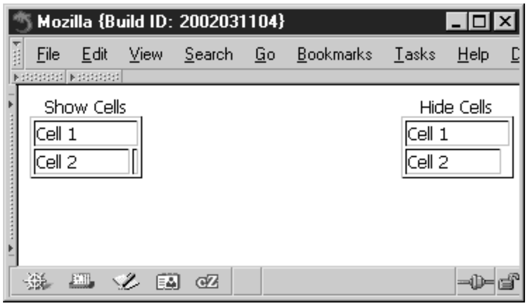
Browser and CSS Support Notes

CSS1	IE 4, 5, 5.5, 6	Nav 4, 4.5–4.8 (buggy), 6, 7	Opera 5, 6, 7

TIP *While the display property itself is generally implemented, most of the CSS2 display properties are not widely supported.*

empty-cells

This CSS2 property is used to control whether or not borders show on empty table cells. It takes two values: **show** or **hide**. The example here shows the subtle difference.


Mozilla {Build ID: 2002031104}
File Edit View Search Go Bookmarks Tasks Help D
Show Cells
Cell 1
Cell 2
Hide Cells
Cell 1
Cell 2

In most browsers that do not support this property, empty table cell borders are generally hidden.

Examples

```
<table border="1" style="empty-cells:show; width: 80px;">
<caption>Show Cells</caption>
<tr>
    <td colspan="2">Cell 1</td>
 </tr>
 <tr>
    <td>Cell 2</td>
    <td></td>
 </tr>
</table>
```

Browser and CSS Support Notes

CSS2	IE does not support except IE Mac	Nav 6, 7	Opera 5, 6, 7

float

This property influences the horizontal alignment of an element, making it "float" toward the left or right margin of its containing element. Possible values are **left**, **right**, and **none**. Floated regions act much like **<img>** tags that have been aligned left or right with respect to text.

Examples

```
#myimage {float: left;}
#pullquote {border-style: double; border-width: 5px;
            background-color: yellow; float: right;}
```

Browser and CSS Support Notes

CSS1	IE 4, 5 (buggy), 5.5, 6	Nav 4, 4.5–4.8 (buggy), 6, 7	Opera 6, 7

font

This property provides a shorthand way to specify all font properties with one style rule. The shorthand syntax is shown here:

```
font: font-style font-variant font-weight font-size/line-height font-family;
```

It is not necessary to include all properties and the lists of variant fonts should be separated by commas with those font names consisting of more than one word placed in quotes. You shouldn't use shorthand rules until the individual property values are correctly set.

Example

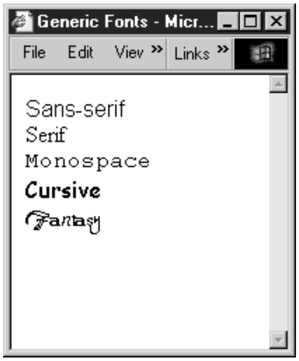
```
p {font: normal small-caps bold 12pt/18pt "Times New Roman", Courier, serif;}
.super {font: italic 18pt sans-serif;}
```

Browser and CSS Support Notes

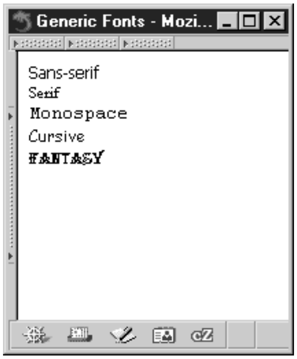
CSS1	IE 3 (incomplete) 4, 5, 5.5, 6	Nav 4 (incomplete on Mac), 4.5–4.8, 6, 7	Opera 6, 7

font-family

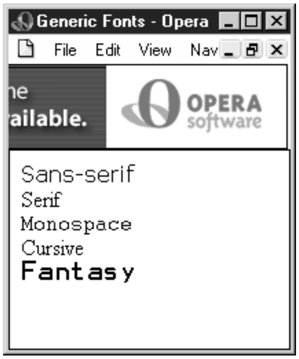
This property sets the font face for text. It is equivalent to the **face** attribute of a **<font>** tag. Fonts may be named specifically or a generic font family name may be used. When multiple font names are specified and separated by commas, they are read in descending order looking for the first match. Generally, a generic font name will be listed at the end of a font list. There are five generic font names currently available: **serif**, **sans-serif**, **cursive**, **fantasy**, and **monospace**. Their renderings under modern browsers are shown here, but beware that they may not render the same in all browsers.



Sans-serif
Serif
Monospace
Cursive
Fantasy



Sans-serif
Serif
Monospace
Cursive
FANTASY



Sans-serif
Serif
Monospace
Cursive
Fantasy

Examples

```
.modern  {font-family: "Arial, Helvetica, sans-serif";}
p        {font-family: "Times New Roman";}
body     {font-family: "Times New Roman, Courier";}
#special {font-family: fantasy;}
```

Browser Support Notes

CSS1	IE 4, 5, 5.5, 6	Nav 4, 4.5–4.8, 6, 7	Opera 4, 5, 6, 7

font-size

This property sets the font size of text. Options include exact sizes set with points (**pt**), pixels (**px**), picas (**pc**), inches (**in**), millimeters (**mm**), and centimeters (**cm**). Standard relative sizing units in em (**em**) and x-height (**ex)** may also be used, as well as the relative size keywords **larger** and **smaller** and percentage value (for example, 90%). Percentage values set the font size to a percentage of the primary font size of a section or document. For example, if the **font-size** for the **body** element was set to 12pt, and font size for a <**p>** tag inside the **body** was set to 200%, the text within the **<p>** tag would be 24pt. The property also supports size keywords (**xx-small**, **x-small**, **small**, **medium**, **large, x-large**, **xx-large**). The size keywords are roughly equivalent to the 1–7 size values for a **<font>** tag and the use of **larger** or **smaller** will increase or decrease the text size one relative size.

Examples

```
body      {font-size: 18pt;}
#heading1 {font-size: 36px;}
p         {font-size: 2em;}
h6        {font-size: xx-small;}
.special  {font-size: 75%;}
```

Browser Support Notes

CSS1	IE 4, 5, 5.5, 6	Nav 4, 4.5–4.8, 6, 7	Opera 4, 5, 6, 7

TIP *Keyword sizes such as **xx-large** may vary noticeably in browser implementations. Point (**pt**) size value may also vary across systems that measure screen resolution differently.*

font-size-adjust

This property is used for scaled fonts to make sure that text takes up the same amount of room regardless of the availability of a particular font or not. According to specification, when used properly the property helps ensure the x-height of the first font in a font selection list is maintained for subsequent fonts in the list. This property takes a number, which specifies the scaling factor to be used to adjust fonts. The default value of **none** does not preserve the font's x-height.

Browser Support Notes

CSS2	No IE support	No Nav or Mozilla support	No Opera support

font-size-stretch

This property is used to stretch or condense a font and takes values of **ultra-condensed**, **extra-condensed**, **condensed**, **semi-condensed**, **normal**, **semi-expanded**, **expanded**, **extra-expanded**, and **ultra-expanded**. The property also can take a relative value of **wider** or **narrower** to modify the appearance of text relative to a parent font. The default value is **normal** and is used to override inherited **font-size-stretch** values. At the time of this edition's writing, this property was yet to be supported by any major browser.

Examples

```
.narrow        {font-stretch: narrower;}
#arialstretch  {font-family: Arial; font-stretch: ultra-expanded;}
```

Browser Support Notes

CSS2	No IE support	No Nav or Mozilla support	No Opera support

font-style

This property sets the style of a font to **normal**, **oblique**, or **italic**. Sometimes font style can be controlled using a specific font (for example, Times New Roman Italic).

Examples

```
.backToNormal {font-style: normal;}
#special      {font-style: oblique;}
p.emphasis    {font-style: italic;}
```

Browser Support Notes

CSS1	IE 4, 5, 5.5, 6	Nav 4, 4.5–4.8, 6, 7	Opera 4, 5, 6, 7

font-variant

This property sets a variation of the specified or default font family. Values currently supported are **normal** and **small-caps**. The **small-caps** value sets text in smaller size all capitals. This style is often used in legal documents and license agreements. The **normal** value would be used to override any inherited font-variant value.

Examples

```
.legalese  {font-variant: small-caps;}
.nolegal   {font-variant: normal;}
```

Browser Support Notes

CSS1	IE 4, 5, 5.5, 6	Nav 4, 4.5–4.8, 6, 7	Opera 4, 5, 6, 7

font-weight

This property sets the weight, or relative boldness, of a font. Values can be set with named values (**normal** or **bold**) or with numbered values (**100**–**900**). Numeric weight values run from lightest (**100**) to boldest (**900**) in increments of 100. In practice, under most browsers the values **100**–**500** display as normal text; **600**–**900** display as bold. Browser screen support for other values is inconsistent at best, though print output may show variations. Relative values of **lighter** or **bolder** will increase or decrease the **font-weight** value relative to its surrounding weight. However, onscreen this generally means simply toggling the boldness on or off. There is some consideration for mapping various weight fonts into associated font families. For example, text in Helvetica at **font-weight: 900** would be mapped into Helvetica Black. However, while described in the specification, the support for weight to font family mapping appears nonexistent in browsers at this edition's writing.

Examples

```
em          {font-weight: bold;}
#light      {font-weight: 300;}
.superbold  {font-weight: 900;}
strong      {font-weight: normal; color: red;}
/* note override of default tag presentation */
```

Browser Support Notes

CSS1	IE 4, 5, 5.5, 6	Nav 4, 4.5–4.8, 6, 7	Opera 4, 5, 6, 7

height

This property sets the height of an element's content region (excluding padding, border, and margin). It is generally used on positioned block elements. Absolute measurements as well as percentage values are allowed. Percentage values are based on the height of the containing element. The default value **auto** automatically calculates the height of an element, based on the height of the containing element and the size of the content. Negative values are not allowed.

Examples

```
p {height: 200px; padding: 10px; border: solid 5px;}
#div1 {height: 50%; width: 50%;}
```

Browser Support Notes

CSS1	IE 4, 5, 5.5, 6	Nav 4, 4.5–4.8, 6, 7	Opera 4, 5, 6, 7

left

This property defines the *x* (horizontal) coordinate for a positioned element, relative to the left side of the containing element or browser window. Given that the containing element itself may be positioned, the value of left does not always relate to the distance from the browser window. Values can be specified as lengths (inches, pixels, and so on), as a percentage of the containing object's dimensions, or as **auto**.

Examples

```
#div1 {position: absolute; left: 120px; top: 50px;}
#div2 {position: absolute; left: 30%; top: 50%;}
#div3 {position: absolute; left: auto; top: auto;}
```

Browser Support Notes

CSS2	IE 4, 5, 5.5, 6	Nav 4, 4.5–4.8, 6, 7	Opera 5, 6, 7

letter-spacing

This property sets the amount of spacing between letters. Values can be set in various units (negative values are permitted) or to the default value **normal**. Because you can't cross tags, you may not have full control over inter-character spacing; in other words, the **letter-spacing** property should not be confused with the ability to fully kern text.

Example

```
.tight {font-family: Arial; font-size: 14pt; letter-spacing: 2pt;}
p      {letter-spacing: 1em;}
p.norm {letter-spacing: normal;}
```

Browser and CSS Support Notes

CSS1	IE 4, 5, 5.5, 6	Nav 6, 7	Opera 4, 5, 6, 7

line-height

This property sets the height (leading) between lines of text in a block-level element such as a paragraph. Values can be specified as a number of lines, a number of units (pixels, points, inches, centimeters, and so on), or a percentage of the font size. This property is often used in conjunction with the **font-size** property. Alternatively, the **line-height** can be set through the shorthand **font** property.

Examples

```
.double {line-height: 2;}
p       {font-size: 14px; line-height: 16px;}
p.norm  {line-height: normal;}
body    {line-height: 4ex;}
div     {line-height: 125%;}
```

Browser and CSS Support Notes

CSS1	IE 3, 4, 5, 5.5, 6	Nav 4, 4.5–4.7 (some problems with number of lines value), 6, 7	Opera 4, 5, 6, 7

list-style

This shorthand property sets **list-style-type, list-style-position** and **list-style-image**. In practice, the properties can appear in any order, but they ought to be written in the presented order.

Examples

```
ul       {list-style: inside url("bullet.gif");}
#square  {list-style: outside square;}
ol       {list-style: lower-roman inside;}
```

Browser and CSS Support Notes

CSS1	IE 4, 5, 5.5, 6	Nav 6, 7	Opera 4, 5, 6, 7

list-style-image

This property assigns a graphic image to a list label, using the URL of the image. The value for **list-style-image** other than a URL is **none**.

Example

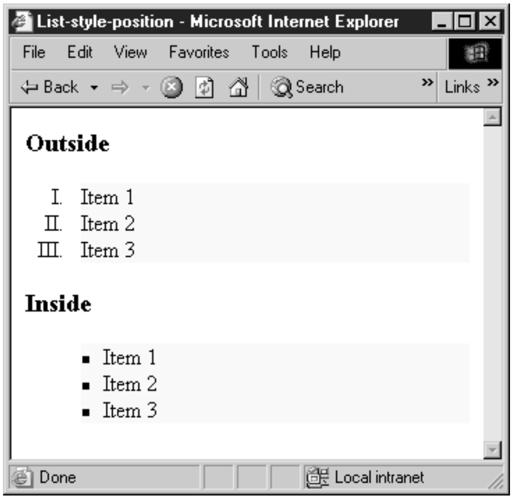
```
ul {list-style-image: url(ball.gif);}
```

Browser and CSS Support Notes

CSS1	IE 4, 5, 5.5, 6	Nav 6, 7	Opera 4, 5, 6, 7

list-style-position

This property specifies whether the labels for an element's list items are positioned **inside** or **outside** the "box" defined by the list. The difference is illustrated clearly here.



Outside

I. Item 1
II. Item 2
III. Item 3

Inside

- Item 1
- Item 2
- Item 3

Example

```
ol {list-style-type: upper-roman; list-style-position: outside;
    background: yellow;}
ul {list-style-type: square; list-style-position: inside;
    background: yellow;}
```

Browser and CSS Support Notes

CSS1	IE 4, 5, 5.5, 6	Nav 6, 7	Opera 4, 5, 6, 7

list-style-type

This property defines labels for ordered and unordered lists or elements that have their **display** property set to **list-item**. The value **none** prevents a list label from displaying. CSS1 defines **disc**, **circle**, and **square**, which are typically used on unordered lists (**<ul>**). The values **decimal**, **lower-roman**, **upper-roman**, **lower-alpha**, and **upper-alpha** are typically used on ordered lists **(<ol>).** These property types correspond to the HTML/XHTML type attributes for lists. CSS2 adds more type values primarily for ordered lists in foreign languages. The CSS2 values include **decimal-leading-zero**, **lower-greek**, **lower-latin**, **upper-latin**, **hebrew**, **armenian**, **georgian**, **cjk-ideographic**, **hiragana**, **katakana**, **hiragana-iroha**, and **katakana-iroha**.

Examples

```
ol       {list-style-type: upper-roman;}
ol.none  {list-style-type: none;}
.ichi-ni {list-style-type: hiragana;}
```

Browser and CSS Support Notes

CSS1	IE 4, 5, 5.5, 6	Nav 4, 4.5–4.8, 6, 7	Opera 4, 5, 6, 7

TIP *Many of the CSS2 values for **list-style-type** are not supported in browsers.*

margin

The **margin** property sets a consistent margin on all four sides of the affected element. Margin values can be set to a specific length (**15pt**, **2em**, and so on) or to a percentage value of the block element's width. Margins can be set to negative values but content may be clipped. The value **auto** attempts to calculate the margin automatically. However, this value is buggy in many CSS implementations, particularly older browser versions, and might be avoided for compatibility reasons.

As a shorthand form, it is possible to set the four margins (**margin-top, margin-bottom**, **margin-right**, and **margin-left**) independently with this property. A single value will be copied to all four margins. With two values, the first value will specify top and bottom margins and the second value will specify the top right and left margins. If three values are specified, the first defines the top margin, the second defines the left and right margins, and the third defines the bottom margin. Note that the unspecified margin is inferred from the value defined for its opposite side. Lastly, all four values can also be set in order of top, right, bottom, and left.

Examples

```
p {margin: 15pt;} /* all sides  15pt */
#div1 {margin: 20px 10px;} /* 20px top-bottom, 10px left-right */
#div2 {margin: 10px 10px 5px 10px;}
/* 10px top, 10px right 5px bottom, 10px left */
```

Browser and CSS Support Notes

CSS1	IE 4 (buggy), 5, 5.5, 6	Nav 4,4.5–4.8 (buggy), 6, 7	Opera 4, 5, 6, 7

margin-bottom

This property sets an element's bottom margin and can be any normal measurement value as well as a negative value.

Example

```
p {margin-bottom: 10pt;}
```

Browser and CSS Support Notes

CSS1	IE 4 (buggy), 5, 5.5, 6	Nav 4, 4.5–4.8 (buggy), 6, 7	Opera 4, 5, 6, 7

margin-left

This property sets an element's left margin and can be any normal measurement value as well as a negative value.

Examples

```
p    {margin-right: 15pt;}
.off {margin-left: -10px;}
```

Browser and CSS Support Notes

CSS1	IE 4 (buggy), 5, 5.5, 6	Nav 4, 4.5–4.8 (buggy), 6, 7	Opera 4, 5, 6, 7

margin-right

This property sets an element's right margin and can be any normal measurement value as well as a negative value.

Example

```
p {margin-right: 15pt;}
```

Browser and CSS Support Notes

CSS1	IE 4 (buggy), 5, 5.5, 6	Nav 4, 4.5–4.8 (buggy), 6, 7	Opera 4, 5, 6, 7

margin-top

This property sets an element's top margin and can be any normal measurement value as well as a negative value.

Example

```
p {margin-top: 15pt;}
```

Browser and CSS Support Notes

CSS1	IE 4 (buggy), 5, 5.5, 6	Nav 4, 4.5–4.8 (buggy), 6, 7	Opera 4, 5, 6, 7

marker-offset

This property specifies the distance between the nearest border edges of a marker box and its associated surrounding box. The value may be specified in any length unit either relative or absolute. Lengths may be negative. So far, this property is not supported by any major browser.

Example

```
li:before { display: marker; marker-offset: 3em;
            content: counter(mycounter, lower-roman) ".";
            counter-increment: mycounter;}
```

Browser and CSS Support Notes

CSS2	No IE support	No Nav or Mozilla support	No Opera support

marks

Defines if during printing cross marks (**cross**) or crop marks (**crop**) are rendered just outside a page box edge. In order to see these indications, your printed page size may have to be adjusted smaller.

Example

```
body {marks: crop;}
```

Browser and CSS Support Notes

CSS2	No IE support	No Netscape or Mozilla support	No Opera support

max-height

Defines the maximum height a region may expand to if it is relatively sized. Measurements are generally in fixed values although relative units such as percentages may be used.

Example

```
#div1 {height: 50%; max-height: 500px; overflow: hidden;}
```

Browser and CSS Support Notes

CSS2	No IE support	Nav 6, 7	Opera 7

max-width

Defines the maximum width a region may expand to if it is relatively sized. Useful to constrain large regions of text from having overly long lines, which can result in readability problems. Measurements are generally in fixed values although relative units such as percentages may be used.

Example

```
#div1 {width: 50%; max-width: 800px;}
```

Browser and CSS Support Notes

CSS2	No IE support	Nav 6, 7	Opera 7

min-height

Defines the minimum height a region may reduce to if it is relatively sized and the browser window is adjusted. Measurements are generally in fixed values although relative units such as percentages may be used.

Example

```
#div1 {min-height: 100px; height: 50%;}
```

Browser and CSS Support Notes

CSS2	No IE support	Nav 6, 7	Opera 7

min-width

Defines the minimum width a region may reduce to if it is relatively sized and the browser window is adjusted. Measurements are generally in fixed values although relative units such as percentages may be used.

Example

```
#div1 {width: 50%; max-width: 800px; min-width: 400px;}
```

Browser and CSS Support Notes

CSS2	No IE support	Nav 6, 7	Opera 7

orphans

Defines the minimum number of lines of a paragraph that must be left at the bottom of a page. This property is really only meaningful in a paged environment such as print output. The default value is **2** if unspecified.

Examples

```
#hateorphans  {orphans: 5;}
.orphaned     {orphans: 1;}
```

Browser and CSS Support Notes

CSS2	No IE support	Nav 6, 7	Opera 7

outline

This property is a shorthand form that sets all outline properties at once. Outlines resemble borders but take up no additional space, and can be set to a shape different from that of the image, form field, or other element to which they are applied. Outlines are drawn over an item, rather than around it, thus causing no reflow. The shorthand syntax is as follows:

`outline``:` *`outline-color outline-style outline-width`*`;`

While outlines are similar to borders, their individual sides cannot be set. All outline properties, both specific and shorthand, apply to the whole outline.

Examples

```
p:hover  {outline: dashed;}
.test    {outline: green solid 10px;}
```

Browser and CSS Support Notes

CSS2	No IE support	Nav 6, 7 use a proprietary style	Opera 7

TIP *Mozilla also supports an outline style with a proprietary syntax:* ***-moz-outline****.*

outline-color

This property accepts all CSS color values as discussed earlier in the appendix as well as in Appendix E. The keyword **invert** is also supported and should perform a color inversion on the pixels on the screen.

Examples

```
p:hover  {outline-style: dashed; outline-color: green;}
.test    {outline-width: 10px; outline-style: solid; outline-color: #f00;}
```

Browser and CSS Support Notes

CSS2	No IE support	Nav 6, 7 use a proprietary style	Opera 7

outline-style

This property defines a style for an element's outline, which is generally the same as its **border-style**. While outlines resemble borders, they are not supposed to take up canvas space, and theoretically could appear as a different shape than borders, potentially wrapping around the edges of content. Since outlines are drawn over an item, rather than around it, there is no reflow. The allowed values for this property are the same as **border-style** and include **none**, **dotted**, **dashed**, **solid**, **double**, **groove**, **ridge**, **inset**, and **outset**. See **border-style** property for specific information on each style.

Examples

```
p:hover  {outline-style: dashed;}
.test    {outline-width: 10px; outline-style: solid; outline-color: black;}
```

Browser and CSS Support Notes

CSS2	No IE support	Nav 6, 7 use a proprietary style	Opera 7

outline-width

This property defines a width for an element's outline, which is generally the same as its **border-width**. While outlines resemble borders, they are not supposed to take up canvas space, and theoretically could appear as a different shape than borders. Furthermore, outlines are drawn over an item, rather than around it, thus causing no reflow. Like **border-width**, this property's values can be keywords (**thin**, **medium**, or **thick**) and numerical lengths such as pixels (**px**), inches (**in**), and so on.

Examples

```
p     {outline-style: dashed; outline-width: thick;}
.test {outline-width: 10px; outline-style: solid; outline-color: black;}
```

Browser and CSS Support Notes

CSS2	No IE support	Nav 6, 7 use a proprietary style	Opera 7

overflow

This property determines an element's behavior when its content doesn't fit into the space defined by the element's other properties. Possible values are **visible**, **hidden**, **scroll**, **auto** or **inherit**. By default, content will be visible, but a value of **hidden** will clip content that extends past the defined region size. A value of **scroll** adds scroll bars appropriately so content can be viewed.

Examples

```
#div1 {position: absolute; left: 20px; top: 20px;
       width: 100px; height: 100px; overflow: scroll;}
#div2 {height: 100px; width: 100px; overflow: hidden;}
```

Browser and CSS Support Notes

CSS2	IE 4, 5, 5.5 (buggy), 6	Nav 4, 4.5–4.8 (buggy), 6, 7	Opera 4, 5, 6, 7

TIP *Printing content in a region with **overflow** set to **scroll** can be troublesome.*

padding

The padding properties set the space between an element's border and its content. The **padding** property sets the padding for all four sides; the other four properties set the padding for specific sides. Values can be specified as specific values (pixels, points, and so on) or as a percentage of the element's overall width. The shorthand **property** is similar to **margin**. A single value creates equal padding on all sides. Up to four values can be used, in the following clockwise order: **top**, **right**, **bottom**, and **left**. Any missing value defaults to the value defined for the side opposite to it. However, unlike the **margin** property, the **padding** property cannot take negative values.

Examples

```
#div1 {border-style: solid; padding: 10px 20px 10px;}
#div2 {border-style: dashed; padding: 50px;}
#div3 {padding: 10px 20px;}
```

Browser and CSS Support Notes

CSS1	IE 4, 5, 5.5 (buggy), 6	Nav 4, 4.5–4.8 (buggy), 6, 7	Opera 4, 5, 6, 7

padding-bottom

This property sets the distance between an element's bottom border and the bottom of its contained content.

Example

```
#div1 {padding-bottom: 5px;}
```

Browser and CSS Support Notes

CSS1	IE 4, 5 (buggy), 5.5, 6	Nav 4, 4.5–4.8 (buggy), 6, 7	Opera 4, 5, 6, 7

padding-left

This property sets the distance between an element's left border and the left edge of its content.

Example

```
#div1 {padding-left: 3em;}
```

Browser and CSS Support Notes

CSS1	IE 4, 5 (buggy), 5.5, 6	Nav 4, 4.5–4.8 (buggy), 6, 7	Opera 4, 5, 6, 7

padding-right

This property sets the distance between an element's right border and the rightmost edge of its content.

Example

```
#div1 {padding-right: .5in;}
```

Browser and CSS Support Notes

CSS1	IE 4, 5 (buggy), 5.5, 6	Nav 4, 4.5–4.8 (buggy), 6, 7	Opera 4, 5, 6, 7

padding-top

This property sets the distance between an element's top border and the top of its content.

Example

```
#div1 {padding-top: 25px;}
```

Browser and CSS Support Notes

CSS1	IE 4, 5 (buggy), 5.5, 6	Nav 4, 4.5–4.8 (buggy), 6, 7	Opera 4, 5, 6, 7

page

Defines the type of page where an element should be displayed. The page is defined and named with an **@page** selector.

Example

```
/* sets tables to be on landscape pages */
@page land {size: landscape;}
table {page: land;}
```

Browser and CSS Support Notes

CSS2	IE 5.5, 6	No Nav or Mozilla support	No Opera support

page-break-after

This property is used to control page breaks when printing a document. The property is set relative to the end of an element. A value of **always** forces a page break after the element. A value of **avoid** attempts to avoid a page break after the element. A value of **left** forces one or two page breaks after the element so that the next page is considered a left page. A value of **right** forces one or two page breaks after the element so the next page is considered a right page. The default value of **auto** neither forces nor forbids a page break.

Examples

```
#breakitdown {page-break-after: always;}
.getitright  {page-break-after: right;}
```

Browser and CSS Support Notes

CSS2	IE 4, 5, 5.5, 6	Nav 6, 7	Opera 7

page-break-before

This property is used to control page breaks when printing a document. This property is set relative to the start of an element. A value of **always** forces a page break before the element. A value of **avoid** attempts to avoid a page break before the element. A value of **left** forces one or two page breaks before the element so that the next page is considered a left page. A value of **right** forces one or two page breaks before the element so the next page is considered a right page. The default value of **auto** neither forces nor forbids a page break.

Examples

```
#breakitdown {page-break-before: always;}
.lefty       {page-break-before: left;}
```

Browser and CSS Support Notes

CSS2	IE 4, 5, 5.5, 6	Nav 6, 7	Opera 7

page-break-inside

This property is used to force or prohibit a printing page break within an element. A value of **always** forces a page break within the element. A value of **avoid** attempts to avoid a page break within the element. A value of **left** forces one or two page breaks within the element so that the next page is considered a left page. A value of **right** forces one or two page breaks within the element so the next page is considered a right page. The default value of **auto** neither forces nor forbids a page break.

Examples

```
#breakitdown {page-break-inside: always;}
.nobreaks    {page-break-inside: avoid;}
```

Browser and CSS Support Notes

CSS2	No IE support	Nav 6, 7	Opera 7

position

This property defines how an element is positioned relative to other elements, using the values **static**, **absolute**, **fixed**, and **relative**. When positioned **absolute**, the **left**, **right**, **top**, and **bottom** properties define the element's precise location, using the affected element's upper-left corner (0,0) as reference. Because elements can contain other elements, 0,0 is not necessarily the upper-left corner of the browser window. When a **relative** position is used, offsets will be related to the object's natural position in the document flow. An element with absolute position will be set at the defined coordinates regardless to its position within the document, but will scroll with a window. An object with a **fixed** position will stay in position onscreen as things scroll. The default value static places elements according to the natural order in which they occur in a document.

Examples

```
span    {position: relative; left: 190px; top: 30px;}
#div1   {position: absolute; left: 120px; top: 50px;}
#navbar {position:fixed; left: 0px; top: 0px;}
```

Browser and CSS Support Notes

CSS2	IE 4, 5, 5.5, 6	Nav 4, 4.5–4.8, 6, 7	Opera 4, 5, 6, 7

NOTE *The **fixed** position value is not supported in Internet Explorer versions although it is supported in later versions of standards-compliant browsers such as Mozilla and Opera.*

quotes

This property defines the style of quotation marks to be used with embedded quotations. A value of **none** produces no quotation marks. Two strings can also be given for the left and rightmost quotation symbols.

Examples

```
<style type="text/css">
 q { quotes: '"[' ']"'  }
</style>

<p>Hey <q>You can <q>quote</q> me</q> on this</p>
```

Browser and CSS Support Notes

CSS2	No IE support	Nav 6, 7	Opera 7 claims support but testing indicates it is inconsistent and occasionally broken.

right

This property defines the *x* (horizontal) coordinate for a positioned element, relative to the right side of either the containing element or browser window if directly within the **<body>**. Values can be specified as lengths (inches, pixels, and so on), as a percentage of the containing object's dimensions, or as **auto**. Most browsers assume pixels as the measure if not defined explicitly.

Examples

```
#div1 {position: absolute; right: 120px; top: 50px;}
#div2 {position: absolute; right: 30%; top: 50%;}
#div3 {position: absolute; right: auto; top: auto;}
```

Browser and CSS Support Notes

CSS2	IE 4, 5, 5.5, 6	Nav 6, 7	Opera 4, 5, 6, 7

size

This property specifies the size and orientation of a page box for printing. The **size** may either be "absolute" (fixed size) or "relative" (scalable, i.e., fitting available sheet sizes). Relative values include **landscape** and **portrait**. The default value is **auto**.

Examples

```
@page {size: landscape;}
@page {size: 6in 10in;} /* funny width and height */
```

Browser and CSS Support Notes

CSS2	No IE support but similar proprietary facilities available.	Nav 6, 7	Opera 7

table-layout

This property controls the algorithm used to lay out the table cells, rows, and columns. A value of **fixed** uses the fixed table layout algorithm, which relays not the content of the cells but simply the width of the tables, columns, borders, and defined cell spacing. This should result in faster page rendering. The default value of **auto** uses the standard automatic table layout algorithm, which may require multiple passes or take perceptible time to calculate, particularly when the table is complex or heavily nested.

Example

```
table {table-layout: fixed;}
```

Browser and CSS Support Notes

CSS2	IE 5, 5.5, 6	Nav 6, 7	Opera 7

text-align

This property sets the horizontal alignment of elements. Values are **left**, **right**, **center**, and **justify**. The default value is **left**. This property is similar to the **align** attribute available with HTML/XHTML block-level tags such as **<p>**. Justification may produce poor results: showing white space "rivers" in large text bodies because of screen resizing.

Examples

```
.goleft {text-align: left;}
p       {text-align: justify;}
h1.cent {text-align: center;}
```

Browser and CSS Support Notes

CSS1	IE 3, 4 (no justification), 5, 5.5, 6	Nav 4, 4.5–4.7 (quirks), 6, 7	Opera 4, 5, 6, 7

text-decoration

This property defines specific text effects. Possible values are **blink**, **line-through**, **overline**, **underline**, and **none**. The **blink** value is actually part of the CSS2 specification but is not supported in IE up to version 6. The **text-decoration** property is often used with the **a** element and its associated pseudoclasses (**a:active**, **a:hover**, **a:link**, and **a:visited**) to turn off link underlining or set different looks for hover or visited states.

Example

```
a           {text-decoration: none;}
a:visited   {text-decoration: line-through;}
a:hover     {text-decoration: underline;}
.onsale     {text-decoration: blink;}
.underlined {text-decoration: underline;}
.struck     {text-decoration: line-through;}
```

Browser and CSS Support Notes

CSS1	IE 4, 5, 5.5, 6 (no support for **blink**)	Nav 4, 4.5–4.8 (incomplete), 6, 7	Opera 6, 7

text-indent

This property indents the text in the first line of a block-level element. Values can be defined as length values (**.5cm**, **15px**, **12pt**, and so on) or as a percentage of the width of the block element. The default value is **0**, which indicates no indentation.

Examples

```
p          {text-indent: 5pt;}
#section1  {text-indent: 15%;}
```

Browser and CSS Support Notes

CSS1	IE 3, 4, 5, 5.5, 6	Nav 4, 4.5–4.7, 6, 7	Opera 4, 5, 6, 7

text-shadow

This as yet unimplemented CSS2 property sets a shadow effect for text. The shadow is defined by a comma-separated list of shadow effects to be applied to the text of the element. The shadow effects are applied in the order specified and may overlay each other, but they will never overlay the text itself. Each shadow effect must specify a shadow offset and may optionally specify a blur radius and a shadow color.

A shadow offset is specified with two length values usually in absolute measurement that indicate the distance from the text. The first length value specifies the horizontal distance to the right of the text. A negative horizontal length value places the shadow to the left of the text. The second length value specifies the vertical distance below the text. A negative vertical length value places the shadow above the text.

An optional blur radius may be specified after the shadow offset. The blur radius is a length value that indicates the boundaries of the blur effect.

A color value may optionally be specified before or after the length values of the shadow effect. The color value will be used as the basis for the shadow effect. If no color is specified, the value of an inherited color property is used.

Examples

```
h3 {text-shadow: 0.2em 0.2em; }
/* shadow to right and below */

span {text-shadow: 3px 3px 5px red; }
/* red blurry shadow right and below */

.solar{background: white; color: white; text-shadow: black 0px 0px 5px;}
/* sets an outline effect on the text */
```

Browser and CSS Support Notes

CSS2	No IE support	No Netscape or Mozilla support	No Opera support

text-transform

This property transforms the case of the affected text. Possible values are **capitalize**, **uppercase**, **lowercase**, and **none**. Note that a value of **capitalize** will affect every word in the selected text rather than first word only. The value of **none** is used to override any inherited **text-transform** values.

Examples

```
h1           {text-transform: capitalize;}
h1.nocap     {text-transform: none;}
.allsmall    {text-transform: lowercase;}
#bigletters {text-transform: uppercase; font-size: larger;}
```

Browser and CSS Support Notes

CSS1	IE 4, 5, 5.5, 6	Nav 4 (incomplete for Mac), 4.5–4.8, 6, 7	Opera 6, 7

top

This property defines the *y* (vertical) coordinate for a positioned element, relative to the top of the enclosing object or browser window. Values can be specified as lengths (inches, pixels, and so on), as a percentage of the containing object's dimensions, or as **auto**, which lets this property function as determined by the browser or as defined by the parent element. Browsers often assume pixels as a default measurement if none is specified.

Examples

```
#div1 {position: absolute; left: 100px; top: 150px;}
#div2 {position: absolute; left: 50%; top: 30%;}
#div3 {position: absolute; left: auto; top: auto;}
```

Browser and CSS Support Notes

CSS1	IE 4, 5, 5.5, 6	Nav 4, 4.5–4.8, 6, 7	Opera 6, 7

unicode-bidi

This property allows the text direction to be overridden to support multiple languages in the same document. The value **normal** uses the standard direction and rendering. A value of **embed** allows a new level of embedding to change direction while **bidi-override** allows the direction property to override any predefined direction.

Example

```
<div><span style="unicode-bidi: embed; direction: rtl;
      background-color: yellow;">?here doing I am  What</span>
 This is just a test.</div>
```

Browser and CSS Support Notes

CSS2	IE 5, 5.5, 6	Nav 6, 7	No Opera support

vertical-align

This property sets the vertical positioning of text and images with respect to the baseline setting. Possible values are **baseline**, **sub**, **super**, **top**, **text-top**, **middle**, **bottom**, and **text-bottom**. A value of **top** aligns the top of text or images with the top of the tallest element, relative to the baseline. A value of **text-top** aligns the top of text or images with the top of the font in the containing element while **text-bottom** aligns things with the bottom of the font. A value of **middle** aligns the middle of text or images to the middle of the x-height of the containing element. A value of **bottom** aligns the bottom of text or images with the bottom of the lowest element, relative to the baseline. The **sub** and **super** values provide subscript and superscript style. Percentages can also be given as values. The default value is **baseline**.

Examples

```
p      {vertical-align: top;}
.dive  {vertical-align: sub;}
.climb {vertical-align: super;}
```

Browser and CSS Support Notes

CSS1	IE 4, 5 (buggy), 5.5, 6	Nav 4, 4.5–4.8 (buggy), 6, 7	Opera 4, 5, 6, 7

visibility

This property determines whether or not an element is visible. Possible values are **hidden**, **visible**, and **inherit**. Be aware that a hidden element still occupies its full canvas space, but because this property is often used in conjunction with **z-index**, this may not matter. The default value **inherit** specifies that an element inherits its visibility state from the element that contains it. CSS2 defines a **collapse** value for this property that is often used with table cells. This property is commonly accessed via JavaScript to show and hide page objects in a manner often dubbed DHTML (Dynamic HTML).

Examples

```
p            {visibility: inherit;}
 peek-a-boo  {visibility: hidden;}
```

Browser and CSS Support Notes

CSS2	IE 4, 5, 5.5, 6	Nav 4, 4.5–4.8, 6, 7	Opera 4, 5, 6, 7

white-space

This property controls how spaces, tabs, and newline characters are handled in an element. Possible values are **normal**, **pre**, and **nowrap**. The normal value collapses multiple white space characters into single spaces and automatically wraps lines, as in normal HTML/XHTML. The **pre** value makes the element act much like a **<pre>** tag and preserves all white space. The value of **nowrap** prevents lines from wrapping if they exceed the element's content width.

Examples

```
p           {white-space: pre;}
pre         {white-space: normal;}
.sourcecode {white-space: nowrap;}
```

Browser and CSS Support Notes

CSS1	IE 5.5, 6	Nav 4, 4.5–4.8 (partial), 6, 7	Opera 4, 5, 6, 7

widows

Defines the minimum number of lines in a paragraph to be left at the top of a page. This property is really only meaningful in a paged environment such as print output. The default value is **2** if unspecified.

Examples

```
#hatewidows  {widows: 5;}
.widowmaker  {widows: 1;}
```

Browser and CSS Support Notes

CSS2	No IE support	Nav 6, 7	Opera 7

width

This property sets the width of an element's content region (excluding padding, border, and margin). Standard length units can be used and pixels (**px**) is often the assumed measurement in browsers. Percentage values, based on the width of the containing element, can also be used. The default value of **auto** automatically calculates the width of an element, based on the width of the containing element and the size of the content.

Examples

```
p     {width: 400px; padding: 10px; border: solid 5px;}
#div1 {width: 80%; padding: 10px; border: solid 5px;}
```

Browser and CSS Support Notes

CSS1	IE 4, 5, 5.5, 6	Nav 4, 4.5–4.8, 6, 7	Opera 4, 5, 6, 7

word-spacing

This property sets the spacing between words. Values can be set in inches (**in**), centimeters (**cm**), millimeters (**mm**), points (**pt**), picas (**pc**), em spaces (**em**), or pixels (**px**). Negative values are possible with this property, and may be used either for interesting effects or to create unreadable text. A default value of **normal** sets word spacing to the standard browser setting.

Examples

```
p         {font-family: Arial; font-size: 16pt; word-spacing: 3pt;}
p.normal  {font-family: Helvetica; font-size: 12pt; word-spacing: normal;}
```

Browser and CSS Support Notes

CSS1	IE 6	Nav 4, 4.5–4.8, 6, 7	Opera 4, 5, 6, 7

z-index

This property defines a layering context with relative or absolute positioning for elements containing other elements. By default, overlapping elements stack in the order in which they are defined in an HTML/XHTML document. This property can override default layering by assigning numeric layering values to an element, with higher numbers layering above lower numbers. The **auto** value tries to determine the z-placement of an element automatically by its order in the document.

Example

```
#div1 {position: absolute; top: 20px; left: 20px; height: 50px; width: 50px;
      background-color: blue; z-index: 2;}
```

Browser and CSS Support Notes

CSS2	IE 4, 5, 5.5, 6	Nav 4, 4.5–4.8, 6, 7	Opera 4, 5, 6, 7

CSS2 Aural Style Properties

CSS2 specifies a number of properties for use with speech-based browsers, which are listed here. No major browsers currently support these properties; however, in the future they may be useful for defining how speech-enabled browsers will "read" a document, right down to rate of speech, pauses before and after words, and when reading should be cued. Table B-7 lists the CSS2 aural style properties: more details on these properties are presented in Chapter 11.

TIP *While aural properties may seem to have little use, some CSS authors like to use aural style sheet rules to confuse certain browser versions to overload properties. This technique is not suggested and scripting logic should be used instead.*

Microsoft Extensions to CSS

Microsoft has introduced a number of proprietary extensions to CSS, many of which have also been proposed as additions to the CSS specification. The following section provides a quick overview of these, and notes the earliest version of Internet Explorer to support each one. For a more comprehensive overview of these properties, and of Microsoft's CSS support in general, see http://www.msdn.microsoft.com/workshop/author/css/reference/attributes.asp.

background-position-x

This Microsoft-proposed CSS property defines the x-coordinate of the **background-position** property. Introduced in Internet Explorer 4.

TABLE B-7 CSS2 Aural Style Properties

azimuth	**cue**
cue-after	**cue-before**
elevation	**pause**
pause-after	**pause-before**
pitch	**pitch-range**
play-during	**richness**
speak	**speak-header**
speak-numeral	**speak-punctuation**
speech-rate	**stress**
voice-family	**volume**

Example

```
<body style="background-image: url(picture.gif);
      background-repeat: no-repeat;
      background-position-x: 25%;">
```

background-position-y

This Microsoft-proposed CSS property defines the y-coordinate of the **background-position** property. Introduced in Internet Explorer 4.

Example

```
<body style="background-image: url(picture.gif);
             background-repeat: no-repeat;
             background-position-y: 200px;">
```

behavior

This Microsoft-proposed CSS property is used to define the URL for a script providing DHTML behavior. Introduced in Internet Explorer 5.0.

Example

```
<h1 style="behavior: url(colorchange.js);">What a dynamic header!</h1>
```

filter

This Microsoft-proposed CSS property is used to apply filter effects to associated HTML elements. Introduced in Internet Explorer 4.

Example

```
<h2 style="filter: Blur(Add = 1, Direction = 90, Strength = 20);
    width: 100%;">This header is all blurry.</h2>
```

ime-mode

This Microsoft-proposed CSS property is used to set the state of an Input Method Editor (IME); for use with Chinese, Japanese, and Korean character sets. Introduced in Internet Explorer 5.0.

Example

```
<textarea style="ime-mode:active;"></textarea>
```

layout-grid

This Microsoft-proposed CSS property defines a grid to be used in laying out Japanese or Chinese characters in a Web document. This is a shorthand property for the layout grid properties discussed next. Introduced in Internet Explorer 5.0.

Example

```
<p style="layout-grid: char line 12px 12px .5in;">
   A short text sample.</p>
```

layout-grid-char

This Microsoft-proposed CSS property defines the size of the character grid used for laying out Japanese or Chinese characters in a Web document. Introduced in Internet Explorer 5.0.

Example

```
<p style="layout-grid-char: 50px;">
  A short text sample.</p>
```

layout-grid-line

This Microsoft-proposed CSS property defines the gridline value used for laying out Japanese or Chinese characters in a Web document. Introduced in Internet Explorer 5.0.

Example

```
<p style="layout-grid-line: 100px">
A short text sample<br />
   with line breaks so<br />
   the meaning of this<br />
   property will be obvious.</p>
```

layout-grid-mode

This Microsoft-proposed CSS property defines if the text layout grid uses one or two dimensions. Introduced in Internet Explorer 5.0.

Example

```
<p style="layout-grid-mode: none; layout-grid-line: 100px;">
   A short text sample<br />
   with layout-grid-mode<br />
   set to a value of none<br />
   to turn off the grid.</p>
```

layout-grid-type

This Microsoft-proposed CSS property defines the type of grid to be used for laying out Japanese or Chinese characters in a Web document. Introduced in Internet Explorer 5.0.

Example

```
<p style="layout-grid-type: strict; layout-grid-line: 55px;">
   A short text sample.</p>
```

line-break

This Microsoft-proposed CSS property defines line-breaking rules for Japanese text. Introduced in Internet Explorer 5.0.

Example

```
p {line-break: normal;}
```

overflow-x

This Microsoft-proposed CSS property defines how content should behave when it exceeds the width of its enclosing element. Introduced in Internet Explorer 4.

Example

```
<p style="overflow-x: scroll; width: 100px;">
ABCDEFGHIJKLMNOPQRSTUVWXYZ
</p>
```

overflow-y

This Microsoft-proposed CSS property defines how content should behave when it exceeds the height of its enclosing element. Introduced in Internet Explorer 4.

Example

```
<p style="overflow-y: scroll; height: 25px; width: 50px;
          background-color: lightblue;">
ABC<br />
DEF<br />
GHI<br />
JKL<br />
MNO<br />
PQR<br />
STU<br />
VWXYZ
</p>
```

ruby-align

This Microsoft-proposed CSS property defines the alignment of ruby text as defined by the **<rt>** element, in relation to base text defined by the **<ruby>** tag (see entries for **<ruby>** and **<rt>** in Appendix A). Introduced in Internet Explorer 5.0.

Example

```
<ruby style="ruby-align: right;">
This is the base text defined by the ruby element.

<rt>This is the ruby text defined by the rt element.</rt>
</ruby>
```

ruby-overhang

This Microsoft-proposed CSS property defines the overhang of ruby text as defined by the **<rt>** tag, in relation to base text defined by the **<ruby>** tag (see entries for **<ruby>** and **<rt>** in Appendix A). Introduced in Internet Explorer 5.0.

Example

```
<ruby style="ruby-overhang: auto;">
This is the base text defined by the ruby element

<rt>This is the ruby text defined by the rt element
and it's a lot longer than the base text in this example.
A lot longer.</rt>
</ruby>

This is text outside of the ruby element.
```

ruby-position

This Microsoft-proposed CSS property defines the position of ruby text as defined by the **<rt>** tag, in relation to base text defined by the **<ruby>** tag (see entries for **<ruby>** and **<rt>** in Appendix A). Introduced in Internet Explorer 5.0.

Example

```
<ruby style="ruby-position: inline;">
This is the base text defined by the ruby element.

<rt>This is the ruby text defined by the rt element but it won't
look that way in IE 5.5 because it has been defined as inline in
relation to the base text.
</rt>
</ruby>
```

scrollbar-3d-light-color

This Microsoft extension to CSS defines a color for the top and left edges of the scroll box in a scrollbar; this and the related scrollbar properties can be applied to the browser window scrollbar when used with the **body** element, or to other elements when used in conjunction with clipping properties. Introduced in Internet Explorer 5.5.

Example

```
body {scrollbar-3d-light-color: lightblue;}
```

scrollbar-arrow-color

This Microsoft extension to CSS defines a color for the arrows in a scrollbar. Introduced in Internet Explorer 5.5.

Example

```
body {scrollbar-arrow-color: red;}
```

scrollbar-base-color

This Microsoft extension to CSS defines the base color for a scrollbar. Introduced in Internet Explorer 5.5.

Example

```
body {scrollbar-base-color: green;}
```

scrollbar-dark-shadow-color

This Microsoft extension to CSS defines a shadow color for the right and bottom edges of a scrollbar. Introduced in Internet Explorer 5.5.

Example

```
body {scrollbar-dark-shadow-color: #0000FF;}
```

scrollbar-face-color

This Microsoft extension to CSS defines a color for the face of a scrollbar. Introduced in Internet Explorer 5.5.

Example

```
body {scrollbar-face-color: #CC00FF;}
```

scrollbar-highlight-color

This Microsoft extension to CSS defines a color for the top and left edges of a scrollbar. Introduced in Internet Explorer 5.5.

Example

```
body {scrollbar-highlight-color: #CCFFFF;}
```

scrollbar-shadow-color

This Microsoft extension to CSS defines a color for the right and bottom edges of a scrollbar. Introduced in Internet Explorer 5.5.

Example

```
body {scrollbar-shadow-color: purple;}
```

TIP *Scrollbar color changes may not show in Internet Explorer if the browser is in strict compatibility mode as set by the doctype statement.*

text-autospace

This Microsoft-proposed CSS property defines spacing values for text; used when combining different types of characters, such as regular text, ideographic text, and numeric characters. Introduced in Internet Explorer 5.0.

text-justify

This Microsoft-proposed CSS property provides greater control over how justified text should be aligned and spaced. Introduced in Internet Explorer 5.0.

Example

```
<p style="text-align: justify; text-justify: distribute-all-lines;
          width: 250px;">
This paragraph is not only justified, but the text-justify property
is set to a value that makes the last line justify as well.
</p>
```

text-kashida-space

This Microsoft-proposed CSS property defines the ratio between kashida expansion and white space expansion in justified text (kashida is a typographic effect used with Arabic writing systems). Introduced in Internet Explorer 5.5.

text-underline-position

This Microsoft-proposed CSS property defines the position of underlining set with the text-decoration property. Introduced in Internet Explorer 5.0.

Example

```
<p style="text-decoration: underline;
          text-underline-position: above;">
This example uses the text-underline-position property to
place the underlining on top of the text. Why not just set
text-decoration to overline instead?</p>
```

word-break

This Microsoft-proposed CSS property can be used to allow line breaks within words, primarily for use with Asian languages. Introduced in Internet Explorer 5.0.

Example

```
<div style="word-break: break-all; width:50px;">
Words can break in this code example. Like this one:
Sesquipedalianism</div>
```

word-wrap

This Microsoft extension to CSS can be set to allow line breaks within words when content exceeds the limits of its containing element. Introduced in Internet Explorer 5.5.

Example

```
<p style="word-wrap: break-word; width: 30px;">
Words can break in this code example. Here is another long one:
Transcendentalism</p>
```

writing-mode

This Microsoft-proposed CSS property can be used to set text flow appropriate for European alphabets or East Asian alphabets. Values are **lr-tb** (left-to-right, top-to-bottom) and **tb-rl** (top-to-bottom, right-to-left). Introduced in Internet Explorer 5.5.

Example

```
<p style="writing-mode: tb-rl;">
This example will really turn your head if you
view it in Internet Explorer 5.5.</p>
```

zoom

This Microsoft extension to CSS can be used to define a magnification scale for an element. Can be used with scripting to create rollover effects. Introduced in Internet Explorer 5.5 and appears to be slated for CSS3 inclusion.

Example

```
<p onmouseover="this.style.zoom='150%';"
   onmouseout="this.style.zoom='normal';">
    Careful, this text might jump out at you!</p>
```