

## Quick Reference Library

### CSS Quick Reference

Copyright 1999-2002 by Infinite Software Solutions, Inc. All rights reserved.

Trademark Information.

Welcome to the **DevGuru Cascading Style Sheets Quick Reference** guide. This is a useful 166 page reference source that defines and explains all of the various style sheet properties, values, and displays sample code.

Cascading Style Sheets (CSS) is a declarative language that is used to enhance the HTML language. CSS is user friendly and was purposely designed to be very readable and writable. The terminology of CSS is lifted from the terminology of desktop publishing. A minimum amount of coding can create sophisticated web pages that have a common theme in appearance.

The early history of style sheets can be described as slightly chaotic. In particular, browser compatibility proved to be a major issue for the first developers who delved into CSS. To help rectify this problem, the World Wide Web Consortium (W3C) published a set of standards in December 1996 referred to as the "Cascading Style Sheets, Level 1" standards. This was followed in December 1998 by the "Cascading Style Sheets, Level 2" standards. Level 2 was a major revision that almost doubled the size of the CSS language by adding 42 new properties.

This Quick Reference documents the Level 2 standards, which are commonly referred to as CSS2.

Internet Explorer Version 5.5, and to a lesser extent Netscape 6, are fairly compatible with most of the W3C Cascading Style Sheet Level 2 standards. The key word here is most. Neither Internet Explorer nor Netscape recognize all of the W3C standards for Level 1 or Level 2. Further complicating matters is the fact that some companies have created proprietary properties that only work on their browser. So, browser compatibility issues still remain a problem for all versions of all browsers available on the Internet. Therefore, while coding, a developer would be wise to view his or her Web pages on a variety of browsers.

Here is a very simple example of using CSS: **The word red is red.**

Below is the code. Note how the red color is declared using a style statement inside a pair of HTML span tags. The style is applied to all text between the opening and closing <span> tags. Therefore, the closing </span> tag is mandatory.

```
<b>The word <span style="color: red;">red</span> is red.</b>
```

However, a far more useful approach is to declare style properties inside the <head> ... </head> portion of an HTML file and associate that property with some specified selector keyword.

The effects of declaring several style properties can cascade together into creating the final appearance of the page. Used properly, CSS can allow you to create an entire Web site (such as **DevGuru**) with a consistent appearance in design. This is done by creating one .css file that contains all of the style rules for the entire site. Each page is linked to the style file using the HTML link tag:

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

While CSS is not case sensitive, it is recommended that you use only lower case.

Meanwhile, on 19 January 2001, W3C released a working draft on [CSS3](#) which proposes to modularize the CSS specifications. So, the future of Cascading Style Sheets promises to be very evolutionary.

# INDEX

Scroll Down to View Index Sorted on Element Type

<a href="#">background</a>	<a href="#">color</a>	<a href="#">max-height</a>	PSEUDO-CLASSES	
<a href="#">background-attachment</a>	<a href="#">content</a>	<a href="#">max-width</a>		<a href="#">: active</a>
<a href="#">background-color</a>	<a href="#">counter-increment</a>	<a href="#">min-height</a>		<a href="#">: first</a>
<a href="#">background-image</a>	<a href="#">counter-reset</a>	<a href="#">min-width</a>		<a href="#">: first-child</a>
<a href="#">background-position</a>	<a href="#">cursor</a>	<a href="#">outline</a>		<a href="#">: focus</a>
<a href="#">background-repeat</a>	<a href="#">direction</a>	<a href="#">outline-color</a>		<a href="#">: hover</a>
<a href="#">border</a>	<a href="#">display</a>	<a href="#">outline-style</a>		<a href="#">: lang</a>
<a href="#">border-bottom</a>	<a href="#">empty-cells</a>	<a href="#">outline-width</a>		<a href="#">: left</a>
<a href="#">border-bottom-color</a>	<a href="#">float</a>	<a href="#">overflow</a>		<a href="#">: link</a>
<a href="#">border-bottom-style</a>	<a href="#">font</a>	<a href="#">padding</a>		<a href="#">: right</a>
<a href="#">border-bottom-width</a>	<a href="#">font-family</a>	<a href="#">padding-bottom</a>	<a href="#">: visited</a>	
<a href="#">border-collapse</a>	<a href="#">font-size</a>	<a href="#">padding-left</a>		
<a href="#">border-color</a>	<a href="#">font-size-adjust</a>	<a href="#">padding-right</a>	PSEUDO-ELEMENTS	
<a href="#">border-left</a>	<a href="#">font-stretch</a>	<a href="#">padding-top</a>		<a href="#">: after</a>
<a href="#">border-left-color</a>	<a href="#">font-style</a>	<a href="#">position</a>		<a href="#">: before</a>
<a href="#">border-left-style</a>	<a href="#">font-variant</a>	<a href="#">quotes</a>		<a href="#">: first-letter</a>
<a href="#">border-left-width</a>	<a href="#">font-weight</a>	<a href="#">right</a>		<a href="#">: first-line</a>
<a href="#">border-right</a>	<a href="#">height</a>	<a href="#">table-layout</a>		
<a href="#">border-right-color</a>	<a href="#">left</a>	<a href="#">text-align</a>		
<a href="#">border-right-style</a>	<a href="#">letter-spacing</a>	<a href="#">text-decoration</a>		
<a href="#">border-right-width</a>	<a href="#">line-height</a>	<a href="#">text-indent</a>		
<a href="#">border-spacing</a>	<a href="#">list-style</a>	<a href="#">text-shadow</a>		AT-RULES
<a href="#">border-style</a>	<a href="#">list-style-image</a>	<a href="#">text-transform</a>	<a href="#">@ charset</a>	
<a href="#">border-top</a>	<a href="#">list-style-position</a>	<a href="#">top</a>	<a href="#">@ font-face</a>	
<a href="#">border-top-color</a>	<a href="#">list-style-type</a>	<a href="#">unicode-bidi</a>	<a href="#">@ import</a>	
<a href="#">border-top-style</a>	<a href="#">margin</a>	<a href="#">vertical-align</a>	<a href="#">@ media</a>	
<a href="#">border-top-width</a>	<a href="#">margin-bottom</a>	<a href="#">visibility</a>	<a href="#">@ page</a>	
<a href="#">border-width</a>	<a href="#">margin-left</a>	<a href="#">white-space</a>		
<a href="#">bottom</a>	<a href="#">margin-right</a>	<a href="#">width</a>		
<a href="#">caption-side</a>	<a href="#">margin-top</a>	<a href="#">word-spacing</a>		
<a href="#">clear</a>	<a href="#">marker-offset</a>	<a href="#">z-index</a>		
<a href="#">clip</a>				

# INDEX

The Same Index as Above Sorted by Element Type

<b>BACKGROUND</b>	<b>DIMENSIONS</b>	<b>OUTLINES</b>
<a href="#">background</a>	<a href="#">height</a>	<a href="#">outline</a>
<a href="#">background-attachment</a>	<a href="#">line-height</a>	<a href="#">outline-color</a>
<a href="#">background-color</a>	<a href="#">max-height</a>	<a href="#">outline-style</a>
<a href="#">background-image</a>	<a href="#">max-width</a>	<a href="#">outline-width</a>
<a href="#">background-position</a>	<a href="#">min-height</a>	
<a href="#">background-repeat</a>	<a href="#">min-width</a>	<b>PADDING</b>
	<a href="#">width</a>	<a href="#">padding</a>
<b>BORDER</b>	<b>FONT</b>	<a href="#">padding-bottom</a>
<a href="#">border</a>	<a href="#">font</a>	<a href="#">padding-left</a>
<a href="#">border-bottom</a>	<a href="#">font-family</a>	<a href="#">padding-right</a>
<a href="#">border-bottom-color</a>	<a href="#">font-size</a>	<a href="#">padding-top</a>
<a href="#">border-bottom-style</a>	<a href="#">font-size-adjust</a>	
<a href="#">border-bottom-width</a>	<a href="#">font-stretch</a>	<b>POSITIONING</b>
<a href="#">border-color</a>	<a href="#">font-style</a>	<a href="#">bottom</a>
<a href="#">border-left</a>		<a href="#">clip</a>

[border-left-color](#)  
[border-left-style](#)  
[border-left-width](#)  
[border-right](#)  
[border-right-color](#)  
[border-right-style](#)  
[border-right-width](#)  
[border-style](#)  
[border-top](#)  
[border-top-color](#)  
[border-top-style](#)  
[border-top-width](#)  
[border-width](#)

#### **CLASSIFICATION**

[clear](#)  
[cursor](#)  
[display](#)  
[float](#)  
[position](#)  
[visibility](#)

[font-variant](#)  
[font-weight](#)

#### **GENERATED CONTENT**

[content](#)  
[counter-increment](#)  
[counter-reset](#)  
[quotes](#)

#### **LIST and MARKER**

[list-style](#)  
[list-style-image](#)  
[list-style-position](#)  
[list-style-type](#)  
[marker-offset](#)

#### **MARGIN**

[margin](#)  
[margin-bottom](#)  
[margin-left](#)  
[margin-right](#)  
[margin-top](#)

[left](#)  
[overflow](#)  
[right](#)  
[top](#)  
[vertical-align](#)  
[z-index](#)

#### **TABLE**

[border-collapse](#)  
[border-spacing](#)  
[caption-side](#)  
[empty-cells](#)  
[table-layout](#)

#### **TEXT**

[color](#)  
[direction](#)  
[letter-spacing](#)  
[text-align](#)  
[text-decoration](#)  
[text-indent](#)  
[text-shadow](#)  
[text-transform](#)  
[unicode-bidi](#)  
[white-space](#)  
[word-spacing](#)

# PROPERTY: background

---

**background** : background-color &| background-image &| background-repeat &| background-attachment &| background-position

**Compatibility:** IE4+ N4+

**Version:** Level 1

**Inherited:** No

The **background** property allows you to combine together in only one declaration any, or all, of five individual Cascading Style Sheet properties related to background.

Each of these five individual properties are discussed on their own page.

Code:

```
blockquote { background: red no-repeat scroll 5% 60%; }
body { background: url("images/aardvark.gif") repeat-y; }
pre { background: url("images/aardvark.gif") top; }
caption { background: fuchsia; }
```

# PROPERTY: background-attachment

---

**background-attachment** : scroll | fixed

**Compatibility:** IE4+ N6

**Version:** Level 1

**Inherited:** No

The **background-attachment** property allows you to choose if the background is fixed or scrolls.

## scroll

The **scroll** value causes the background image to move along with the foreground text and images as you scroll up or down or right or left.

## fixed

The **fixed** value causes the background to remain fixed in place. The foreground text and images move over the fixed background as you scroll up or down or right or left.

Code:

```
body
{
  background-image: url("anasazi.tif");
  background-attachment: scroll;
}
```

```
html
{
  background-image: url("anasazi.tif");
  background-attachment: fixed;
}
```

# PROPERTY: background-color

**background-color** : **color** | **transparent**

**Compatibility:** IE4+ N4+

**Version:** Level 1

**Inherited:** No

The **background-color** property allows you to select the background color. Since, you can assign the **background-color** to individual elements, you can have more than one background color on a page.

## color

The **color** value can be the keyword color name, the hex six-digit number (#FFFFFF), or the rgb three-digit value (255,255,255). There are sixteen standard colors in html:

<b>aqua</b> #00FFFF rgb(0,255,255)	<b>gray</b> #808080 rgb(128,128,128)	<b>silver</b> #C0C0C0 rgb(192,192,192)	<b>navy</b> #000080 rgb(0,0,128)
<b>black</b> #000000 rgb(0,0,0)	<b>green</b> #800000 rgb(128,0,0)	<b>teal</b> #008080 rgb(0,128,128)	<b>olive</b> #808000 rgb(128,128,0)
<b>blue</b> #0000FF rgb(0,0,255)	<b>lime</b> #00FF00 rgb(0,255,0)	<b>white</b> #FFFFFF rgb(255,255,255)	<b>purple</b> #800080 rgb(128,0,128)
<b>fuchsia</b> #FF00FF rgb(255,0,255)	<b>maroon</b> #800000 rgb(128,0,0)	<b>yellow</b> #FFFF00 rgb(255,255,0)	<b>red</b> #FF0000 rgb(255,0,0)

The **Guru** has also created a [Color Chart](#) that displays all of the named HTML colors. In addition the hexadecimal code is provided for 256 colors.

## transparent

The **transparent** value allows the background color to be visible through the foreground images. This is how an area inside any character, such as the white space inside this O, is the same as this white background.

Code:

```
p {background-color: silver}
br {background-color: rgb(223,71,177) }
body {background-color: #98AB6F}
pre {background-color: transparent}
```

or

```
<span style="background-color: yellow;">A yellow background</span>
```

output:

A yellow background

# COLOR CHARTS

## Hexadecimal and HTML Named Color Charts.

The chart below gives the hexadecimal numbers for the RGB codes that define the displayed colors.

#EEEEEE	#DDDDDD	#CCCCCC	#BBBBBB	#AAAAAA	#999999	
#888888	#777777	#666666	#555555	#444444	#333333	
#000000	#000033	#000066	#000099	#0000CC	#0000FF	
#330000	#330033	#330066	#330099	#3300CC	#3300FF	
#660000	#660033	#660066	#660099	#6600CC	#6600FF	
#990000	#990033	#990066	#990099	#9900CC	#9900FF	
#CC0000	#CC0033	#CC0066	#CC0099	#CC00CC	#CC00FF	
#FF0000	#FF0033	#FF0066	#FF0099	#FF00CC	#FF00FF	
#003300	#003333	#003366	#003399	#0033CC	#0033FF	
#333300	#333333	#333366	#333399	#3333CC	#3333FF	
#663300	#663333	#663366	#663399	#6633CC	#6633FF	
#993300	#993333	#993366	#993399	#9933CC	#9933FF	
#CC3300	#CC3333	#CC3366	#CC3399	#CC33CC	#CC33FF	
#FF3300	#FF3333	#FF3366	#FF3399	#FF33CC	#FF33FF	
#006600	#006633	#006666	#006699	#0066CC	#0066FF	
#336600	#336633	#336666	#336699	#3366CC	#3366FF	
#666600	#666633	#666666	#666699	#6666CC	#6666FF	
#996600	#996633	#996666	#996699	#9966CC	#9966FF	
#CC6600	#CC6633	#CC6666	#CC3399	#CC66CC	#CC66FF	
#FF6600	#FF6633	#FF6666	#FF6699	#FF66CC	#FF66FF	
#009900	#009933	#009966	#009999	#0099CC	#0099FF	
#339900	#339933	#339966	#339999	#3399CC	#3399FF	
#669900	#669933	#669966	#669999	#6699CC	#6699FF	
#999900	#999933	#999966	#999999	#9999CC	#9999FF	
#CC9900	#CC9933	#CC9966	#CC9999	#CC99CC	#CC99FF	
#FF9900	#FF9933	#FF9966	#FF9999	#FF99CC	#FF99FF	
#00CC00	#00CC33	#00CC66	#00CC99	#00CCCC	#00CCFF	
#33CC00	#33CC33	#33CC66	#33CC99	#33CCCC	#33CCFF	
#66CC00	#66CC33	#66CC66	#66CC99	#66CCCC	#66CCFF	
#99CC00	#99CC33	#99CC66	#99CC99	#99CCCC	#99CCFF	
#CCCC00	#CCCC33	#CCCC66	#CCCC99	#CCCCCC	#CCCCFF	
#FFCC00	#FFCC33	#FFCC66	#FFCC99	#FFCCCC	#FFCCFF	
#00FF00	#00FF33	#00FF66	#00FF99	#00FFCC	#00FFFF	
#33FF00	#33FF33	#33FF66	#33FF99	#33FFCC	#33FFFF	
#66FF00	#66FF33	#66FF66	#66FF99	#66FFCC	#66FFFF	

#99FF00	#99FF33	#99FF66	#99FF99	#99FFCC	#99FFFF
#CCFF00	#CCFF33	#CCFF66	#CCFF99	#CCFFCC	#CCFFFF
#FFFF00	#FFFF33	#FFFF66	#FFFF99	#FFFFCC	#FFFFFF

The following chart shows the HTML named colors in alphabetical order.

Aliceblue	Antiquewhite	Aqua
Aquamarine	Azure	Beige
Bisque	Black	Blanchedalmond
Blue	Blueviolet	Brown
Burlywood	Cadetblue	Chartreuse
Chocolate	Coral	Cornflowerblue
Cornsilk	Crimson	Cyan
Darkblue	Darkcyan	Darkgoldenrod
Darkgray	Darkgreen	Darkkhaki
Darkmagenta	Darkolivegreen	Darkorange
Darkorchid	Darkred	Darksalmon
Darkseagreen	Darkslateblue	Darkslategray
Darkturquoise	Darkviolet	Deeppink
Deepskyblue	Dimgray	Dodgerblue
Firebrick	Floralwhite	Forestgreen
Fuchsia	Gainsboro	Ghostwhite
Gold	Goldenrod	Gray
Green	Greenyellow	Honeydew
Hotpink	Indianred	Indigo
Ivory	Khaki	Lavender
Lavenderblush	Lawngreen	Lemonchiffon
Lightblue	Lightcoral	Lightcyan
Lightgoldenrodyellow	Lightgreen	Lightgrey
Lightpink	Lightsalmon	Lightseagreen
Lightskyblue	Lightslategray	Lightsteelblue
Lightyellow	Lime	Limegreen
Linen	Magenta	Maroon
Mediumaquamarine	Mediumblue	Mediumorchid

Mediumpurple		Mediumseagreen		Mediumslateblue	
Mediumspringgreen		Mediumturquoise		Mediumvioletred	
Midnightblue		Mintcream		Mistyrose	
Moccasin		Navajowhite		Navy	
Oldlace		Olive		Olivedrab	
Orange		Orangered		Orchid	
Palegoldenrod		Palegreen		Paleturquoise	
Palevioletred		Papayawhip		Peachpuff	
Peru		Pink		Plum	
Powderblue		Purple		Red	
Rosybrown		Royalblue		Saddlebrown	
Salmon		Sandybrown		Seagreen	
Seashell		Sienna		Silver	
Skyblue		Slateblue		Slategray	
Snow		Springgreen		Steelblue	
Tan		Teal		Thistle	
Tomato		Turquoise		Violet	
Wheat		White		Whitesmoke	

# PROPERTY: background-image

---

**background-image** : **url** | **none**

**Compatibility:** IE4+ N4+

**Version:** Level 1

**Inherited:** No No

The **background-image** property allows you to select an image to use as the background.

## **url**

The **url** value is the full path to a named image, or the name of the image. The syntax is demonstrated in the code examples.

## **none**

The **none** value is the default. No background image is selected. By declaring **none**, you can ensure that any previous declaration will not effect a specific background.

Code:

```
code { background-image: url("comet.jpg"); }
blockquote { background-image: url("c:\InetPub\MyPixs\comet.jpg"); }
br { background-image: url("http://Fred.com/ImageFile/Q.gif"); }
body { background-image: none; }
```

# PROPERTY: background-position

---

**background-position** : [ **percentage** | **length** ] |  
[ [ **top** | **center** | **bottom** ] & [ **left** | **center** | **right** ] ]

**Compatibility:** IE4+ N6

**Version:** Level 1

**Inherited:** No

The **background-position** property allows you to determine the initial position of a background image.

You can use any combination of a **percentage** or a **length**. Alternately, you can use one of the keywords [**top**, **center**, **bottom**] and/or one of the keywords [**left**, **right**, **center**] to define the position. You cannot mix the keywords with **percentage** or **length**.

[ **percentage** &| **length** ]

The **percentage** value is usually listed as a pair of values. Place a blank (white) space between the pair of percents. The first **percentage** in the pair refers to the horizontal axis and the second **percentage** to the vertical axis. 0% 0% is defined as the left top corner and 100% 100% as the right bottom corner. If you only give one **percentage**, it will only set the horizontal axis and the vertical axis is defaulted to 50%.

Other examples of percentage pairs:

% Pair	Position
0% 50%	left center
50% 0%	center top
50% 50%	center
0% 100%	left bottom corner
100% 0%	right top corner
50% 100%	center bottom
100% 50%	right center

The **length** value can be in any of the following seven units. Use the abbreviation.

Length	Abbrev.
centimeters	cm
ems	em
inches	in

millimeters	mm
picas	pc
pixels	px
points	pt

Like the **percentage** pair, you usually use a pair of **length** values. The first **length** in the pair refers to the horizontal axis and the second **length** to the vertical axis. The (0,0) position is defined as the top left corner. If you only give one length, it will only set the horizontal axis and the vertical axis is defaulted to 50%.

[ **top** | **center** | **bottom** ] & [ **left** | **center** | **right** ]

Instead of the **length** or the **percentage** values, you can use keyword values to set the position. Again, you usually use a pair of keywords. If you only use one keyword, the second value of the pair is defaulted to **center**. Note that the order of the pair of keyword values is not important. Thus, **top center top**, and **top center** are equivalent and define the same image position. You cannot mix keyword values, such as **top**, with **percentage** or **length** values.

Code:

```
img {  
background: url("images/aardvark.gif");  
background-position: 35% 80%;  
}
```

```
menu {  
background: url("images/aardvark.gif");  
background-position: 35% 2.5cm;  
}
```

```
a {  
background: url("images/aardvark.gif");  
background-position: 3.25in;  
}
```

```
body {  
background: url("images/aardvark.gif");  
background-position: top right;  
}
```

# PROPERTY: background-repeat

---

**background-repeat** : **repeat** | **repeat-x** | **repeat-y** | **no-repeat**

**Compatibility:** IE4+ N4+

**Version:** Level 1

**Inherited:** No

The **background-repeat** property allows you to repeat a background image vertically and/or horizontally. The repetition of an image is called tiling.

## **repeat**

The **repeat** value causes the background image to be tiled both vertically and horizontally.

## **repeat-x**

The **repeat-x** value causes the background image to be tiled horizontally (along the x-axis).

## **repeat-y**

The **repeat-y** value causes the background image to be tiled vertically (along the y-axis).

## **no-repeat**

The **no-repeat** value is the default. No tiling occurs. By declaring **no-repeat**, you can ensure that any previous declaration will not effect a specific imaged.

Code:

```
body {  
background: url("images/aardvark.gif");  
background-repeat: repeat;  
}
```

```
pre {  
background: url("images/aardvark.gif");  
background-repeat: repeat-x;  
}
```

```
menu {  
background: url("images/aardvark.gif");  
background-repeat: repeat-y;  
}
```

```
p {  
background: url("images/aardvark.gif");  
background-repeat: no-repeat;  
}
```

# PROPERTY: border

---

**border** : **border-width** &| **border-style** &| **border-color**

**Compatibility:** IE4+ N4+

**Version:** Level 1

**Inherited:** No

The **border** property allows you to combine into one declaration the **border-width**, and/or the **border-style**, and/or the **border-color** properties.

Each of these three individual properties are discussed on their own page.

Code:

```
body { border: thick dashed yellow; }  
p { border: thick double yellow; }  
blockquote { border: dotted gray; }
```

or

```
p.bodr { border: 25px solid red; }
```

...

```
<p class="bodr">  
DevGuru is great!  
</p>
```

Output:



DevGuru is great!

# PROPERTY: border-bottom

---

**border-bottom** : **border-bottom-width** &| **border-style** &| **border-color**

**Compatibility:** IE4+ N6

**Version:** Level 1

**Inherited:** No

The **border-bottom** property is a shortcut that allows you to declare for the bottom border the **border-bottom-width**, and/or the **border-style**, and/or the **border-color** properties in only one declaration. The color value uses the same values as discussed in **border-color**.

Code:

```
body { border-bottom: thick dashed yellow; }
body { border-bottom: 23px double yellow; }
body { border-bottom: dotted gray; }
```

or

```
p.bodr
{
border-bottom: 25px solid red;
border-left: 25px solid yellow;
border-right: 25px solid blue;
border-top: 25px solid green;
}
```

...

```
<p class="bodr">
DevGuru is great!
</p>
```

Output:



# PROPERTY: border-bottom-color

**border-bottom-color** : **color**

**Compatibility:** IE4+ N6

**Version:** Level 2

**Inherited:** No

The **border-bottom-color** property allows you to set the color of the bottom border.

There are similar properties that allow you to set the color for the left, right, and top, as well as all four borders.

## color

The **color** value can be the keyword color name, the hex six-digit number (#FFFFFF), or the RGB three-digit value (255,255,255). There are sixteen standard colors in HTML:

<b>aqua</b> #00FFFF rgb(0,255,255)	<b>gray</b> #808080 rgb(128,128,128)	<b>silver</b> #C0C0C0 rgb(192,192,192)	<b>navy</b> #000080 rgb(0,0,128)
<b>black</b> #000000 rgb(0,0,0)	<b>green</b> #008000 rgb(128,0,0)	<b>teal</b> #008080 rgb(0,128,128)	<b>olive</b> #808000 rgb(128,128,0)
<b>blue</b> #0000FF rgb(0,0,255)	<b>lime</b> #00FF00 rgb(0,255,0)	<b>white</b> #FFFFFF rgb(255,255,255)	<b>purple</b> #800080 rgb(128,0,128)
<b>fuchsia</b> #FF00FF rgb(255,0,255)	<b>maroon</b> #800000 rgb(128,0,0)	<b>yellow</b> #FFFF00 rgb(255,255,0)	<b>red</b> #FF0000 rgb(255,0,0)

The **Guru** has also created a [Color Chart](#) that displays all of the named HTML colors. In addition the hexadecimal code is provided for 256 colors.

Code:

```
table { border-bottom-color: RGB(201, 94, 177); }  
body { border-bottom-color: #E95A2B; }  
textarea { border-bottom-color: silver; }
```

or

```
p.bodr  
{  
border-bottom-color: red; border-bottom-style: double; border-bottom-width: 15px;  
border-left-color: yellow; border-left-style: double; border-left-width: 15px;  
border-right-color: blue; border-right-style: double; border-right-width: 15px;  
border-top-color: green; border-top-style: double; border-top-width: 15px;  
}  
...  
<p class="bodr">
```

```
<b>DevGuru is great!</b>  
</p>
```

Output:



# PROPERTY: border-bottom-style

---

**border-bottom-style** : **none** | **hidden** | **dotted** | **dashed** | **solid** | **double** | **groove** | **ridge** | **inset** | **outset**

**Compatibility:** IE4+ N6

**Version:** Level 2

**Inherited:** No

The **border-bottom-style** property allows you to set the appearance of the border line style for the bottom border. Many of the possible styles are not recognized by various browsers.

There are similar properties to set the style for the bottom, right, and top, as well as all four borders.

## **dashed**

A dashed line appears on the background. Many browsers do not recognize this border style.

## **dotted**

A dotted line appears on the background. Many browsers do not recognize this border style.

## **double**

A double solid line appears on the background which has a total width equal to **border-width**.

## **groove**

A 3-D grooved line appears on the background. The exact appearance of the line depends on the selected **color** value.

## **hidden**

The same as **none**, except that it can be used to resolve border conflicts when two different types of borders join or intersect.

## **inset**

A 3-D inset line appears on the background. The exact appearance of the line depends on the selected **color** value.

## **none**

The **none** value is the default. If no style is declared, no border will appear even if other border properties have been set. By declaring **none**, you can ensure that any previous declaration will not effect the four borders.

## **outset**

A 3-D outset line appears on the background. The exact appearance of the line depends on the selected **color** value.

## **ridge**

A 3-D ridged line appears on the background. The exact appearance of the line depends on the selected **color** value.

## **solid**

A solid line appears on the background.

Code:

```
body { border-bottom-style: groove; }
```

or

```
p.bordr
```

```
{  
border-bottom-color: red; border-bottom-style: double; border-bottom-width: 15px;  
border-left-color: yellow; border-left-style: double; border-left-width: 15px;  
border-right-color: blue; border-right-style: double; border-right-width: 15px;  
border-top-color: green; border-top-style: double; border-top-width: 15px;  
}
```

```
...
```

```
<p class="bordr">  
<b>DevGuru is great!</b>  
</p>
```

Output:



# PROPERTY: border-bottom-width

---

**border-bottom-width** : **thin** | **medium** | **thick** | **length**

**Compatibility:** IE4+ N4+

**Version:** Level 1

**Inherited:** No

The **border-bottom-width** property sets the width of the bottom border.

**thin** | **medium** | **thick**

The keyword values of **thin**, **medium**, or **thick** can be used to set the width of the border line. The exact thickness of the border created by using these keyword values will be determined by your computer/browser combination.

**length**

The **length** value can be in any of the following seven units. Use the abbreviation.

Length	Abbrev.
centimeters	cm
ems	Em
inches	in
millimeters	mm
picas	pc
pixels	px
points	pt

Code:

```
body { border-bottom-width: thick; }  
body { border-bottom-width: 0.25in; }  
body { border-bottom-width: 5mm; }
```

or

```
p.bordr  
{  
border-bottom-color: red; border-bottom-style: double; border-bottom-width: 15px;  
border-left-color: yellow; border-left-style: double; border-left-width: 15px;  
border-right-color: blue; border-right-style: double; border-right-width: 15px;  
border-top-color: green; border-top-style: double; border-top-width: 15px;  
}  
...  
<p class="bordr">  
<b>DevGuru is great!</b>
```

</p>

Output:



# PROPERTY: border-collapse

---

**border-collapse** : collapse | separate

**Compatibility:** IE5+

**Version:** Level 2

**Inherited:** Yes

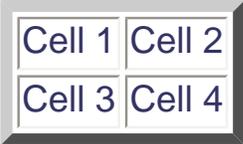
The **border-collapse** property allows you to choose the border model for a table.

There are two types of table border models, **collapse** and **separate**. (Not all browsers support both models.)

This is one of five Level 2 properties that specifically apply style to tables. The other four are **border-spacing**, **caption-side**, **empty-cells**, and **table-layout**.

## collapse

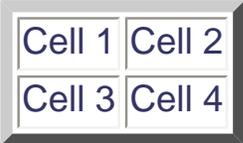
The **collapse** value dictates the use of the **collapse** model. In this model, the table has an outer border and adjacent cells share a common internal border. (In Netscape, this table will appear as a **separate** model.)



Cell 1	Cell 2
Cell 3	Cell 4

## separate

The **separate** value dictates the use of the **separate** model. In this model, the table has a separate outer border and each individual cell has its own separate internal border. The table outer border does not physically touch the individual internal cell borders.



Cell 1	Cell 2
Cell 3	Cell 4

Remember that a border is optional in a table and that the default is no border.

Here is the code for the above two tables:

Code:

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

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



# PROPERTY: border-color

**border-color** : color

**Compatibility:** IE4+ N6

**Version:** Level 1

**Inherited:** No

The **border-color** property allows you to set the color of the border.

You can declare one, two, three or four colors:

If you declare one color, all four borders will be that color.

If you declare two colors, the top and bottom borders will be the first color, the right and left borders will be the second color.

If you declare three colors, the top border will be the first, the right border the second, and the bottom and left borders the third color.

If you declare four colors, the order is top, right, bottom, left.

## color

The **color** value can be the keyword color name, the hex six-digit number (#FFFFFF), or the RGB three-digit value (255,255,255). There are sixteen standard colors in HTML:

<b>aqua</b> #00FFFF rgb(0,255,255)	<b>gray</b> #808080 rgb(128,128,128)	<b>silver</b> #C0C0C0 rgb(192,192,192)	<b>navy</b> #000080 rgb(0,0,128)
<b>black</b> #000000 rgb(0,0,0)	<b>green</b> #008000 rgb(128,0,0)	<b>teal</b> #008080 rgb(0,128,128)	<b>olive</b> #808000 rgb(128,128,0)
<b>blue</b> #0000FF rgb(0,0,255)	<b>lime</b> #00FF00 rgb(0,255,0)	<b>white</b> #FFFFFF rgb(255,255,255)	<b>purple</b> #800080 rgb(128,0,128)
<b>fuchsia</b> #FF00FF rgb(255,0,255)	<b>maroon</b> #800000 rgb(128,0,0)	<b>yellow</b> #FFFF00 rgb(255,255,0)	<b>red</b> #FF0000 rgb(255,0,0)

The **Guru** has also created a [Color Chart](#) that displays all of the named HTML colors. In addition the hexadecimal code is provided for 256 colors.

Code:

```
body { border-color: RGB(201, 94, 177); }  
body { border-color: #E95A2B; }  
body { border-color: silver; }  
body { border-color: silver ;red; }  
body { border-color: silver red RGB(223, 94, 77); }  
body { border-color: silver red RGB(223, 94, 77) black; }
```

or

```
p.bordr
{
border-color: red red red red;
border-style: solid solid solid solid;
border-width: 30px 30px 30px 30px;
}
```

```
...
<p class="bordr">
DevGuru is great!
</p>
```

Output:



DevGuru is great!

# PROPERTY: border-left

---

**border-left** : **border-left-width** &| **border-style** &| **border-color**

**Compatibility:** IE4+ N6

**Version:** Level 1

**Inherited:** No

The **border-left** property allows you to declare for the left border the **border-left-width**, and/or the **border-style**, and/or the **border-color** properties in only one declaration. The color value uses the same values as discussed in **border-color**.

Code:

```
body { border-left: thick dashed yellow; }  
body { border-left: thick double yellow; }  
body { border-left: dotted gray;
```

or

```
p.bodr  
{  
border-bottom: 25px solid red;  
border-left: 25px solid yellow;  
border-right: 25px solid blue;  
border-top: 25px solid green;  
}
```

...

```
<p class="bodr">  
DevGuru is great!  
</p>
```

Output:



# PROPERTY: border-left-color

**border-left-color** : **color**

**Compatibility:** IE4+ N6

**Version:** Level 2

**Inherited:** No

The **border-left-color** property allows you to set the color of the left border.

There are similar properties that allow you to set the color for the bottom, right, and top, as well as all four borders.

## color

The **color** value can be the keyword color name, the hex six-digit number (#FFFFFF), or the RGB three-digit value (255,255,255). There are sixteen standard colors in HTML:

<b>aqua</b> #00FFFF rgb(0,255,255)	<b>gray</b> #808080 rgb(128,128,128)	<b>silver</b> #C0C0C0 rgb(192,192,192)	<b>navy</b> #000080 rgb(0,0,128)
<b>black</b> #000000 rgb(0,0,0)	<b>green</b> #008000 rgb(128,0,0)	<b>teal</b> #008080 rgb(0,128,128)	<b>olive</b> #808000 rgb(128,128,0)
<b>blue</b> #0000FF rgb(0,0,255)	<b>lime</b> #00FF00 rgb(0,255,0)	<b>white</b> #FFFFFF rgb(255,255,255)	<b>purple</b> #800080 rgb(128,0,128)
<b>fuchsia</b> #FF00FF rgb(255,0,255)	<b>maroon</b> #800000 rgb(128,0,0)	<b>yellow</b> #FFFF00 rgb(255,255,0)	<b>red</b> #FF0000 rgb(255,0,0)

The **Guru** has also created a [Color Chart](#) that displays all of the named HTML colors. In addition the hexadecimal code is provided for 256 colors.

Code:

```
table { border-left-color: RGB(201, 94, 177); }  
body { border-left-color: #E95A2B; }  
textarea { border-left-color: silver; }
```

or

```
p.bordr  
{  
border-bottom-color: red; border-bottom-style: double; border-bottom-width: 15px;  
border-left-color: yellow; border-left-style: double; border-left-width: 15px;  
border-right-color: blue; border-right-style: double; border-right-width: 15px;  
border-top-color: green; border-top-style: double; border-top-width: 15px;  
}  
...  
<p class="bordr">
```

```
<b>DevGuru is great!</b>  
</p>
```

Output:



# PROPERTY: border-left-style

---

**border-left-style** : none | hidden | dotted | dashed | solid | double | groove | ridge | inset | outset

**Compatibility:** IE4+ N6

**Version:** Level 2

**Inherited:** No

The **border-left-style** property allows you to set the appearance of the border line style for the left border. Many of the possible styles are not recognized by various browsers.

There are similar properties to set the style for the bottom, right, and top, as well as all four borders.

## dashed

A dashed line appears on the background. Many browsers do not recognize this border style.

## dotted

A dotted line appears on the background. Many browsers do not recognize this border style.

## double

A double solid line appears on the background which has a total width equal to **border-width**.

## groove

A 3-D grooved line appears on the background. The exact appearance of the line depends on the selected **color** value.

## hidden

The same as **none**, except that it can be used to resolve border conflicts when two different types of borders join or intersect.

## inset

A 3-D inset line appears on the background. The exact appearance of the line depends on the selected **color** value.

## none

The **none** value is the default. If no style is declared, no border will appear even if other border properties have been set. By declaring **none**, you can ensure that any previous declaration will not effect the four borders.

## outset

A 3-D outset line appears on the background. The exact appearance of the line depends on the selected **color** value.

## ridge

A 3-D ridged line appears on the background. The exact appearance of the line depends on the selected **color** value.

## **solid**

A solid line appears on the background.

Code:

```
body {border-left-style: double}
```

or

```
p.bordr
{
border-bottom-color: red; border-bottom-style: double; border-bottom-width: 15px;
border-left-color: yellow; border-left-style: double; border-left-width: 15px;
border-right-color: blue; border-right-style: double; border-right-width: 15px;
border-top-color: green; border-top-style: double; border-top-width: 15px;
}
```

```
...
<p class="bordr">
<b>DevGuru is great!</b>
</p>
```

Output:



# PROPERTY: border-left-width

---

**border-left-width** : **thin** | **medium** | **thick** | **length**

**Compatibility:** IE4+ N4+

**Version:** Level 1

**Inherited:** No

The **border-left-width** property sets the width of the left border.

**thin** | **medium** | **thick**

The keyword values of **thin**, **medium**, or **thick** can be used to set the width of the border line. The exact thickness of the border created by using these keyword values will be determined by your computer/browser combination.

**length**

The **length** value can be in any of the following seven units. Use the abbreviation.

Length	Abbrev.
centimeters	cm
ems	em
inches	in
millimeters	mm
picas	pc
pixels	px
points	pt

Code:

```
body { border-left-width: thick; }  
body { border-left-width: 0.25in; }  
body { border-left-width: 5mm; }
```

or

```
p.bordr  
{  
border-bottom-color: red; border-bottom-style: double; border-bottom-width: 15px;  
border-left-color: yellow; border-left-style: double; border-left-width: 15px;  
border-right-color: blue; border-right-style: double; border-right-width: 15px;  
border-top-color: green; border-top-style: double; border-top-width: 15px;  
}  
...  
<p class="bordr">  
<b>DevGuru is great!</b>
```

</p>

Output:



# PROPERTY: border-right

---

**border-right** : **border-right-width** &| **border-style** &| **border-color**

**Compatibility:** IE4+ N6

**Version:** Level 1

**Inherited:** No

The **border-right** property allows you to declare for the right border the **border-right-width**, and/or the **border-style**, and/or the **border-color** properties in only one declaration. The **color** value uses the same values as discussed in **border-color**.

Code:

```
body { border-right: thick dashed yellow; }
body { border-right: thick double yellow; }
body { border-right: dotted gray; }
```

or

```
p.bodr
{
border-bottom: 25px solid red;
border-left: 25px solid yellow;
border-right: 25px solid blue;
border-top: 25px solid green;
}
```

...

```
<p class="bodr">
DevGuru is great!
</p>
```

Output:



## PROPERTY: border-right-color

**border-right-color** : **color**

**Compatibility:** IE4+ N6

**Version:** Level 2

**Inherited:** No

The **border-right-color** property allows you to set the color of the right border.

There are similar properties that allow you to set the color for the bottom, left, and top, as well as all four borders.

### color

The **color** value can be the keyword color name, the hex six-digit number (#FFFFFF), or the RGB three-digit value (255,255,255). There are sixteen standard colors in HTML:

<b>aqua</b> #00FFFF rgb(0,255,255)	<b>gray</b> #808080 rgb(128,128,128)	<b>silver</b> #C0C0C0 rgb(192,192,192)	<b>navy</b> #000080 rgb(0,0,128)
<b>black</b> #000000 rgb(0,0,0)	<b>green</b> #008000 rgb(128,0,0)	<b>teal</b> #008080 rgb(0,128,128)	<b>olive</b> #808000 rgb(128,128,0)
<b>blue</b> #0000FF rgb(0,0,255)	<b>lime</b> #00FF00 rgb(0,255,0)	<b>white</b> #FFFFFF rgb(255,255,255)	<b>purple</b> #800080 rgb(128,0,128)
<b>fuchsia</b> #FF00FF rgb(255,0,255)	<b>maroon</b> #800000 rgb(128,0,0)	<b>yellow</b> #FFFF00 rgb(255,255,0)	<b>red</b> #FF0000 rgb(255,0,0)

The **Guru** has also created a [Color Chart](#) that displays all of the named HTML colors. In addition the hexadecimal code is provided for 256 colors.

Code:

```
table { border-right-color: RGB(201, 94, 177); }  
body { border-right-color: #E95A2B; }  
textarea { border-right-color: silver; }
```

or

```
p.bordr  
{  
border-bottom-color: red; border-bottom-style: double; border-bottom-width: 15px;  
border-left-color: yellow; border-left-style: double; border-left-width: 15px;
```

```
border-right-color: blue; border-right-style: double; border-right-width: 15px;  
border-top-color: green; border-top-style: double; border-top-width: 15px;  
}  
...  
<p class="bordr">  
<b>DevGuru is great!</b>  
</p>
```

Output:



# PROPERTY: border-right-style

---

**border-right-style** : **none** | **hidden** | **dotted** | **dashed** | **solid** | **double** | **groove** | **ridge** | **inset** | **outset**

**Compatibility:** IE4+ N6

**Version:** Level 2

**Inherited:** No

The **border-right-style** property allows you to set the appearance of the border line style for the right border. Many of the possible styles are not recognized by various browsers.

There are similar properties to set the style for the bottom, left, and right, as well as all four borders.

## **dashed**

A dashed line appears on the background. Many browsers do not recognize this border style.

## **dotted**

A dotted line appears on the background. Many browsers do not recognize this border style.

## **double**

A double solid line appears on the background which has a total width equal to **border-width**.

## **groove**

A 3-D grooved line appears on the background. The exact appearance of the line depends on the selected **color** value.

## **hidden**

The same as **none**, except that it can be used to resolve border conflicts when two different types of borders join or intersect.

## **inset**

A 3-D inset line appears on the background. The exact appearance of the line depends on the selected **color** value.

## **none**

The **none** value is the default. If no style is declared, no border will appear even if other border properties have been set. By declaring **none**, you can ensure that any previous declaration will not effect the four borders.

## **outset**

A 3-D outset line appears on the background. The exact appearance of the line depends on the selected **color** value.

## **ridge**

A 3-D ridged line appears on the background. The exact appearance of the line depends on the selected **color** value.

## **solid**

A solid line appears on the background.

Code:

```
body {border-right-style: double}
```

or

```
p.bordr
{
border-bottom-color: red; border-bottom-style: double; border-bottom-width: 15px;
border-left-color: yellow; border-left-style: double; border-left-width: 15px;
border-right-color: blue; border-right-style: double; border-right-width: 15px;
border-top-color: green; border-top-style: double; border-top-width: 15px;
}
```

```
...
<p class="bordr">
<b>DevGuru is great!</b>
</p>
```

Output:



# PROPERTY: border-right-width

---

**border-right-width** : **thin** | **medium** | **thick** | **length**

**Compatibility:** IE4+ N4+

**Version:** Level 1

**Inherited:** No

The **border-right-width** property sets the width of the right border.

**thin** | **medium** | **thick**

The keyword values of **thin**, **medium**, or **thick** can be used to set the width of the border line. The exact thickness of the border created by using these keyword values will be determined by your computer/browser combination.

**length**

The **length** value can be in any of the following seven units. Use the abbreviation.

Length	Abbrev.
centimeters	cm
ems	em
inches	in
millimeters	mm
picas	pc
pixels	px
points	pt

Code:

```
body { border-right-width: thick; }
body { border-right-width: 0.25in; }
body { border-right-width: 5mm; }
```

or

```
p.bodr
{
border-bottom-color: red; border-bottom-style: double; border-bottom-width: 15px;
border-left-color: yellow; border-left-style: double; border-left-width: 15px;
border-right-color: blue; border-right-style: double; border-right-width: 15px;
border-top-color: green; border-top-style: double; border-top-width: 15px;
}
...
<p class="bodr">
<b>DevGuru is great!</b>
```

</p>

Output:



# PROPERTY: border-spacing

---

**border-spacing** : length length

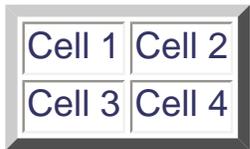
**Compatibility:** Currently not supported by any browser

**Version:** Level 2

**Inherited:** Yes

The **border-spacing** property is used to set the vertical and horizontal spacing between adjacent cells in a table using the **separate** border model.

In the **separate** model, the table has a separate outer border and each individual cell has its own separate internal border. The table outer border does not physically touch the individual internal cell borders.



Remember that a border is optional in a table and that the default is no border.

This is one of five Level 2 properties that specifically apply style to tables. The other four are **border-collapse**, **caption-side**, **empty-cells**, and **table-layout**.

## length length

The **length length** values set the spacing. The second **length** is optional. If only the first **length** is used, then it sets the same value for both the vertical and horizontal spacing between adjacent cells. If both **length length** values are used, the first gives the horizontal spacing and the second the vertical spacing. No negative values are permitted. The values can be in any of the following seven units. Use the abbreviation.

Length	Abbrev.
centimeters	cm
ems	em
inches	in
millimeters	mm
picas	pc
pixels	px
points	pt

Code:

```
<table style="border-collapse: separate; border-spacing: 10px">  
<tr><td>Cell 1</td><td>Cell 2</td></tr>  
<tr><td>Cell 3</td><td>Cell 4</td></tr>
```

</table>

# PROPERTY: border-style

---

**border-style** : none | hidden | dotted | dashed | solid | double | groove | ridge | inset | outset

**Compatibility:** IE4+ N4+

**Version:** Level 1

**Inherited:** No

The **border-style** property allows you to select the same line style for all four borders. The **border-style** property is also used in **border-left**, **border-right**, and **border-top**.

You can declare one, two, three or four styles:

If you declare one style, all four borders will be that style.

If you declare two styles, the top and bottom borders will be the first style, the right and left borders will be the second style.

If you declare three styles, the top border will be the first, the right border the second, and the bottom and left borders the third style.

If you declare four styles, the order is top, right, bottom, left.

## none

The **none** value is the default. If no style is declared, no border will appear even if other border properties have been set. By declaring **none**, you can ensure that any previous declaration will not effect the four borders.

## hidden

The same as **none**, except that it can be used to resolve border conflicts when two different types of borders join or intersect.

## dotted

A dotted line appears on the background. This border style is not recognized by many browsers.

## dashed

A dashed line appears on the background. This border style is not recognized by many browsers.

## solid

A solid line appears on the background.

## double

A double solid line appears on the background which has a total width equal to **border-width**.

## groove

A 3-D grooved line appears on the background. The exact appearance depends on the selected **color** value.

## ridge

A 3-D ridged line appears on the background. The exact appearance depends on the selected **color** value.

## inset

A 3-D inset line appears on the background. The exact appearance depends on the selected **color** value.

## outset

A 3-D outset line appears on the background. The exact appearance depends on the selected **color** value.

Code:

```
body { border-style: double; }  
body { border-style: double groove; }  
body { border-style: double groove dashed; }  
body { border-style: double groove none solid; }
```

or

```
p.bordr  
{  
border-color: red red red red;  
border-style: solid solid solid solid;  
border-width: 30px 30px 30px 30px;  
}
```

...

```
<p class="bordr">  
DevGuru is great!  
</p>
```

Output:



DevGuru is great!

# PROPERTY: border-top

---

**border-top** : **border-top-width** &| **border-style** &| **border-color**

**Compatibility:** IE4+ N6

**Version:** Level 1

**Inherited:** No

The **border-top** property allows you to declare for the top border the **border-top-width**, and/or the **border-style**, and/or the **color** properties in only one declaration. The **color** value uses the same values as discussed in **border-color**.

Code:

```
body { border-top: thick dashed yellow; }
body { border-top: thick double green; }
body { border-top: dotted gray;
```

or

```
p.bodr
{
border-bottom: 25px solid red;
border-left: 25px solid yellow;
border-right: 25px solid blue;
border-top: 25px solid green;
}
```

...

```
<p class="bodr">
DevGuru is great!
</p>
```

Output:



# PROPERTY: border-top-color

**border-top-color** : color

**Compatibility:** IE4+ N6

**Version:** Level 2

**Inherited:** No

The **border-top-color** property allows you to set the color of the top border.

There are similar properties that allow you to set the color for the bottom, left, and right, as well as all four borders.

## color

The **color** value can be the keyword color name, the hex six-digit number (#FFFFFF), or the RGB three-digit value (255,255,255). There are sixteen standard colors in HTML:

<b>aqua</b> #00FFFF rgb(0,255,255)	<b>gray</b> #808080 rgb(128,128,128)	<b>silver</b> #C0C0C0 rgb(192,192,192)	<b>navy</b> #000080 rgb(0,0,128)
<b>black</b> #000000 rgb(0,0,0)	<b>green</b> #008000 rgb(128,0,0)	<b>teal</b> #008080 rgb(0,128,128)	<b>olive</b> #808000 rgb(128,128,0)
<b>blue</b> #0000FF rgb(0,0,255)	<b>lime</b> #00FF00 rgb(0,255,0)	<b>white</b> #FFFFFF rgb(255,255,255)	<b>purple</b> #800080 rgb(128,0,128)
<b>fuchsia</b> #FF00FF rgb(255,0,255)	<b>maroon</b> #800000 rgb(128,0,0)	<b>yellow</b> #FFFF00 rgb(255,255,0)	<b>red</b> #FF0000 rgb(255,0,0)

The **Guru** has also created a [Color Chart](#) that displays all of the named HTML colors. In addition the hexadecimal code is provided for 256 colors.

Code:

```
table { border-top-color: RGB(201, 94, 177); }  
body { border-top-color: #E95A2B; }  
textarea { border-top-color: silver; }
```

or

```
p.bodr  
{  
border-bottom-color: red; border-bottom-style: double; border-bottom-width: 15px;  
border-left-color: yellow; border-left-style: double; border-left-width: 15px;  
border-right-color: blue; border-right-style: double; border-right-width: 15px;  
border-top-color: green; border-top-style: double; border-top-width: 15px;  
}  
...  
<p class="bodr">
```

```
<b>DevGuru is great!</b>  
</p>
```

Output:



# PROPERTY: border-top-style

---

**border-top-style** : **none** | **hidden** | **dotted** | **dashed** | **solid** | **double** | **groove** | **ridge** | **inset** | **outset**

**Compatibility:** IE4+ N6

**Version:** Level 2

**Inherited:** No

The **border-top-style** property allows you to set the appearance of the border line style for the top border. Many of the possible styles are not recognized by various browsers.

There are similar properties to set the style for the bottom, left, and right, as well as all four borders.

## **dashed**

A dashed line appears on the background. Many browsers do not recognize this border style.

## **dotted**

A dotted line appears on the background. Many browsers do not recognize this border style.

## **double**

A double solid line appears on the background which has a total width equal to **border-width**.

## **groove**

A 3-D grooved line appears on the background. The exact appearance of the line depends on the selected **color** value.

## **hidden**

The same as **none**, except that it can be used to resolve border conflicts when two different types of borders join or intersect.

## **inset**

A 3-D inset line appears on the background. The exact appearance of the line depends on the selected **color** value.

## **none**

The **none** value is the default. If no style is declared, no border will appear even if other border properties have been set. By declaring **none**, you can ensure that any previous declaration will not effect the four borders.

## **outset**

A 3-D outset line appears on the background. The exact appearance of the line depends on the selected **color** value.

## **ridge**

A 3-D ridged line appears on the background. The exact appearance of the line depends on the selected **color** value.

## **solid**

A solid line appears on the background.

Code:

```
body { border-top-style: solid; }
```

or

```
p.bordr  
{  
border-bottom-color: red; border-bottom-style: double; border-bottom-width: 15px;  
border-left-color: yellow; border-left-style: double; border-left-width: 15px;  
border-right-color: blue; border-right-style: double; border-right-width: 15px;  
border-top-color: green; border-top-style: double; border-top-width: 15px;  
}
```

```
...  
<p class="bordr">  
<b>DevGuru is great!</b>  
</p>
```

Output:



# PROPERTY: border-top-width

---

**border-top-width** : **thin** | **medium** | **thick** | **length**

**Compatibility:** IE4+ N4+

**Version:** Level 1

**Inherited:** No

The **border-top-width** property sets the width of the top border.

**thin** | **medium** | **thick**

The keyword values of **thin**, **medium**, or **thick** can be used to set the width of the border line. The exact thickness of the border created by using these keyword values will be determined by your computer/browser combination.

**length**

The **length** value can be in any of the following seven units. Use the abbreviation.

Length	Abbrev.
centimeters	cm
ems	em
inches	in
millimeters	mm
picas	pc
pixels	px
points	pt

Code:

```
body { border-top-width: thick; }
body { border-top-width: 0.25in; }
body { border-top-width: 5mm; }
```

or

```
p.bodr
{
border-bottom-color: red; border-bottom-style: double; border-bottom-width: 15px;
border-left-color: yellow; border-left-style: double; border-left-width: 15px;
border-right-color: blue; border-right-style: double; border-right-width: 15px;
border-top-color: green; border-top-style: double; border-top-width: 15px;
}
...
<p class="bodr">
<b>DevGuru is great!</b>
```

</p>

Output:



# PROPERTY: border-width

---

**border-width** : **thin** | **medium** | **thick** | **length**

**Compatibility:** IE4+ N4+

**Version:** Level 1

**Inherited:** No

The **border-width** property allows you to set the width of all four borders with only one declaration.

You can declare one, two, three or four widths:

If you declare one width, all four borders will be that width.

If you declare two widths, the top and bottom borders will be the first width, the right and left borders will be the second width.

If you declare three widths, the top border will be the first, the right border the second, and the bottom and left borders the third width.

If you declare four widths, the order is top, right, bottom, left.

**thin** | **medium** | **thick**

The keyword values of **thin**, **medium**, or **thick** can be used to set the width of the border line. The exact thickness of the border created by using these keyword values will be determined by your computer/browser combination.

**length**

The **length** value can be in any of the following seven units. Use the abbreviation.

Length	Abbrev.
centimeters	cm
ems	em
inches	in
millimeters	mm
picas	pc
pixels	px
points	pt

Code:

```
body { border-width: thick; }
```

```
body { border-width: 0.25in; }  
body { border-width: 5mm; }
```

or

```
p.bordr  
{  
border-color: red red red red;  
border-style: solid solid solid solid;  
border-width: 30px 30px 30px 30px;  
}
```

```
...  
<p class="bordr">  
DevGuru is great!  
</p>
```

Output:



DevGuru is great!

# PROPERTY: bottom

---

**bottom** : auto | length | percentage

**Compatibility:** IE5+

**Version:** Level 2

**Inherited:** No

The **bottom** property sets the physical distance of how far the bottom content edge of an element is above the bottom content edge of the containing block.

A containing block is simply an element that contains one or more related elements.

There are three other properties that allow you to set the distance for the **left**, **right**, and **top** content edges. All four properties are used in conjunction with the **position** property. Note that if the **position** property is set to the **static** value, setting the **bottom** property has no effect.

## auto

The **auto** value dictates that the browser sets the distance between the bottom content edges.

## length

The **length** value sets the distance between the bottom content edges and can be in any of the following seven units. Use the abbreviation.

Length	Abbrev.
centimeters	cm
ems	em
inches	in
millimeters	mm
picas	pc
pixels	px
points	pt

## percentage

The **percentage** value sets the distance between the bottom content edges as a percentage of the overall height of the parent block. It is a positive integer number. The percent sign is mandatory.

Code:

```
p.one {  
position: absolute;  
bottom: 1in;  
left: 1in;
```

```
right: 1in;  
top: 1in;  
}
```

# PROPERTY: caption-side

---

**caption-side** : **bottom** | **left** | **right** | **top**

**Compatibility:** Currently not supported by any browser

**Version:** Level 2

**Inherited:** Yes

The **caption-side** property specifies if the caption box will be on the bottom, left, right, or top of a table. This property is designed to be used with the HTML [caption](#) tag.

It is recommended that special consideration be paid to the width of the table when using a left or right sided caption box. You can vertically align a left or right side caption box to a table by using the **vertical-align** property with the **bottom**, **middle**, or **top** values.

This is one of five Level 2 properties that specifically applies style to tables. The other four are **border-collapse**, **border-spacing**, **empty-cells**, and **table-layout**.

## **bottom**

The **bottom** value places a caption box below the table.

## **left**

The **left** value places a caption box on the left side of the table.

## **right**

The **right** value places a caption box on the right side of the table.

## **top**

The **top** value places a caption box above the table.

Code:

```
caption
{
  caption-side: top;
  width: auto;
  text-align: left;
}
```

# PROPERTY: clear

---

**clear** : none | left | right | both

**Compatibility:** IE4+ N4+

**Version:** Level 1

**Inherited:** No

The **clear** property sets restrictions on where a floating element can be placed with respect to previously placed elements. It should only be applied to block-level elements.

## none

The **none** value places no restriction on where the element can appear in the text.

## left

The **left** value requires that a floating element must be placed lower than the previously occurring element on the left side.

## right

The **right** value requires that a floating element must be placed lower than the previously occurring element on the right side.

## both

The **both** value requires that a floating element must be placed below all previous elements.

Code:

```
a { clear: none; }
b { clear: left; }
li { clear: right; }
p { clear: both; }
```

or

```
p.one { clear: left; }
p.two { clear: right; }
img { float: right; }
```

```
...

<b class="one">
```

First text appears to the left of the Guru image since it is only required to be below the previous element on the left side (which is "Output:") and the Guru is on the right side. Yet in the flow of the code, this text actually occurs after the Guru image.

```
</b>
<b class="two">
```

Second text appears completely below the Guru image since it is required to be below the previous element on the right side which is the Guru.

```
</b>
```

Output:

First text appears to the left of the Guru image since it is only required to be below the previous element on the left side (which is "Output:") and the Guru is on the right side. Yet in the flow of the code, this text actually occurs after the Guru image. Second text appears completely below the Guru image since it is required to be below the previous element on the right side which is the Guru.



# PROPERTY: clip

---

**clip** : auto | rect(top, right, bottom, left)

**Compatibility:** IE4+ N6

**Version:** Level 2

**Inherited:** No

The **clip** property can be used when an element is too big to fit inside the containing block.

A containing block is simply an element that contains related sub-elements. A problem arises if a sub-element is too big in size to fit inside the dimensions of the containing block. By default, when an element is too large to fit in the allotted space, portions will be chopped off so that the remaining portion of the element will fit inside the containing box. The portions that are chopped off are simply not displayed.

The **clip** property allows you to specify what portion of the over-sized sub-element will be visible. This is referred to as clipping the sub-element.

The related **overflow** property allows you to specify if an over-size element is to be clipped. If the **overflow** property is set to the **visible** value, then setting the **clip** property should have no effect. In this case, the over-sized element is completely displayed even though it overflows the boundaries of the containing block.

## auto

The **auto** value dictates that the browser will determine which portion of the element will be visible.

## rect(top, right, bottom, left)

The **rect(top, right, bottom, left)** value sets the shape of the element that is to be visible as a rectangle (currently, this is the only permitted shape). The edges of the rectangle are defined with a blank space delimited list of the mandatory **top**, **right**, **bottom**, and **left** arguments which are the respective dimensional offsets from the four edges of the containing block. The offset can be either a positive or negative value, where a negative value extends beyond the containing block. Each of the **top**, **right**, **bottom**, and **left** arguments can be in any of the following seven units. Use the abbreviation.

Length	Abbrev.
centimeters	cm
ems	em
inches	in
millimeters	mm
picas	pc
pixels	px

points	pt
--------	----

In this example, the containing element is the HTML img tag which is sized as a square with dimensions of 75 pixels by 75 pixels.

Code:

```
img.clp {  
clip: rect(10px 60px 60px 10px);  
height="75px";  
width="75px";  
position: absolute;  
}  
...  
  
<br>  

```

Output:



# PROPERTY: color

**color** : color

**Compatibility:** IE4+ N4+

**Version:** Level 1

**Inherited:** Yes

The **color** property allows you to select the color of the text. The default is black.

The **color** value can be the keyword color name, the hex six-digit number (#FFFFFF), or the RGB three-digit value (255,255,255). There are sixteen standard colors in HTML:

<b>aqua</b> #00FFFF rgb(0,255,255)	<b>gray</b> #808080 rgb(128,128,128)	<b>silver</b> #C0C0C0 rgb(192,192,192)	<b>navy</b> #000080 rgb(0,0,128)
<b>black</b> #000000 rgb(0,0,0)	<b>green</b> #800000 rgb(128,0,0)	<b>teal</b> #008080 rgb(0,128,128)	<b>olive</b> #808000 rgb(128,128,0)
<b>blue</b> #0000FF rgb(0,0,255)	<b>lime</b> #00FF00 rgb(0,255,0)	<b>white</b> #FFFFFF rgb(255,255,255)	<b>purple</b> #800080 rgb(128,0,128)
<b>fuchsia</b> #FF00FF rgb(255,0,255)	<b>maroon</b> #800000 rgb(128,0,0)	<b>yellow</b> #FFFF00 rgb(255,255,0)	<b>red</b> #FF0000 rgb(255,0,0)

The **Guru** has also created a [Color Chart](#) that displays all of the named HTML colors. In addition the hexadecimal code is provided for 256 colors.

Code:

```
p { color: RGB(100, 14, 107); }  
p { color: #E9ACB2; }  
p { color: teal; }
```

or

```
<b>  
<span style="color: fuchsia;">DevGuru</span>  
<span style="color: #00FF00;"> is</span>  
<span style="color: RGB(256, 11, 17);"> great!</span>  
</b>
```

Output:

**DevGuru is great!**

# PROPERTY: content

---

**content** : `attr(alt)` | `counter(name)` | `counter(name, list-style-type)` | `counters(name, string)` | `counters(name, string, list-style-type)` | `no-close-quote` | `no-open-quote` | `close-quote` | `open-quote` | `string` | `url()`

**Compatibility:** N6+

**Version:** Level 2

**Inherited:** No

The **content** property is used with the **:before** and **:after** psuedo-elements to display content either before or after a specified CSS selector. This content can include strings, quotations marks, URLs, and counters. Simply put, this allows you to repetitiously add text and pictures before and after HTML elements.

Note that the **content** property does not inherit. However, the **:before** and **:after** psuedo-elements can inherit any inheritable styles that are in effect, or you can specify various CSS properties to effect the appearance of the content.

## **:after**

The **:after** psuedo-element dictates that the content is to be added after the element specified by the selector. The colon is mandatory.

Syntax: `selector:after { ... }`.

## **:before**

The **:before** psuedo-element dictates that the content is to be added before the element specified by the selector. The colon is mandatory.

Syntax: `selector:before { ... }`.

The **content** property has eleven possible values.

## **attr(alt)**

The **attr** value inserts the text of the **alt** argument as content.

## **counter(name)**

The **counter** value inserts the named counter as content.

## **counter(name, list-style-type)**

The **counter** value inserts the named counter as content rendered in the specified list-style-type.

## **counters(name, string)**

The **counters** value inserts all named counters as content.

## **counters(name, string, list-style-type)**

The **counters** value inserts all named counters as content rendered in the specified list-style-

type.

### close-quote

The **close-quote** value inserts the closing quotation values specified by the nesting level of the **quotes** property.

### no-close-quote

The **no-close-quote** value does not insert a closing quotation mark, but it does increment the nesting level of the **quotes** property.

### no-open-quote

The **no-open-quote** value does not insert an opening quotation mark, but it does increment the nesting level of the **quotes** property.

### open-quote

The **open-quote** value inserts the opening quotation values specified by the nesting level of the **quotes** property.

### string

The **string** value is a string or text enclosed within a pair of quotes.

### url( )

The **url** value is a URL address enclosed within a pair of quotes.

This example adds opening and closing quotes.

Code:

```
<html>
<head>
<title>content test</title>
<style type="text/css">
p:before {content: open-quote;}
p:after {content: close-quote;}
</style>
</head>
<body>
<p>DevGuru is the best reference source on the web.</p>
</body>
</html>
```

Simulated output, however, this code example works on Netscape 6+.

Output:

"DevGuru is the best reference source on the web."

# PROPERTY: counter-increment

---

**counter-increment** : none | identifier number

**Compatibility:** Currently not supported by any browser

**Version:** Level 2

**Inherited:** No

The **counter-increment** property is a blank space delimited list of one or more identifier/number pairs associated with a selector. The pair sets the numeric value by which the named counter will be incremented each time the specified selector is encountered. The number part of the pair is optional and the default is to increment by one.

For example, you could consecutively number paragraphs, sections of a text, and images.

If the **display** property is set to **none**, then you cannot increment. If the **visibility** property is set to **hidden**, then you can increment.

You can use the **content** property to insert counters either before or after a specified element.

## identifier number

The **identifier** value part of the pair is mandatory. It identifies the counter. It can refer to a class, id, or selector. The optional **number** value is the numeric value used to increment the counter. It can be zero or any positive or negative integer. If omitted, the default is one.

## none

The **none** value prevents the counter from incrementing.

Code:

```
p:before
{
  content: "paragraph" counter(paragraph);
  counter-increment: paragraph;
}
```

# PROPERTY: counter-reset

---

**counter-reset** : **none** | **identifier number**

**Compatibility:** Currently not supported by any browser

**Version:** Level 2

**Inherited:** No

The **counter-reset** property is a blank space delimited list of one or more identifier/number pairs associated with a selector. The pair sets the numeric value that the named counter will be reset to when the specified selector is encountered. The number part of the pair is optional and the default is zero.

For example, you could consecutively number nesting paragraphs, sub-sections of a text, and images in a chapter as 1.0, 1.1, 1.2, etc.

If the **display** property is set to **none**, then you cannot reset. If the **visibility** property is set to **hidden**, then you can reset.

You can use the **content** property to insert counters either before or after a specified element.

## **none**

The **none** value prevents the counter from resetting.

## **identifier number**

The **identifier** value part of the pair is mandatory and identifies the counter. It can refer to a class, id, or selector. The optional **number** value is the numeric value to reset to. It can be zero or any positive or negative integer. If omitted, the default is zero.

This code from W3C numbers paragraphs as 1, 1.1, 1.1.1, etc.

Code:

```
ol { counter-reset: item }  
li { display: block }  
li:before { content: counters(item, "."); counter-increment: item }
```

# PROPERTY: cursor

---

**cursor** : **auto** | **crosshair** | **default** | **help** | **move** | **pointer** | **e-resize** | **n-resize** | **ne-resize** | **nw-resize** | **s-resize** | **se-resize** | **sw-resize** | **w-resize** | **text** | **url( )** | **wait**

**Compatibility:** IE4+ N6

**Version:** Level 2

**Inherited:** No

The **cursor** property allows you to set the type of cursor that will be visible. Most browsers do not recognize all of the possible types of cursors.

## **auto**

The **auto** value is a default that allows the browser to set the cursor type.

## **crosshair**

The **crosshair** value renders the cursor as a crosshair (large plus sign).

## **default**

The **default** value allows the browser to set the cursor type.

## **help**

The **help** value usually renders the cursor as a question mark or a balloon.

## **move**

The **move** value is used to signify that something is to be moved. Usually, it looks like a plus sign with arrows at the four tips.

## **pointer**

The **pointer** value renders the cursor as an arrow to signify a link.

## **e-resize**

The **e-resize** value renders the cursor as an arrow pointing to the right.

## **n-resize**

The **n-resize** value renders the cursor as an arrow pointing up.

## **ne-resize**

The **ne-resize** value renders the cursor as an arrow pointing to the upper right.

## **nw-resize**

The **nw-resize** value renders the cursor as an arrow pointing to the upper left.

## **s-resize**

The **s-resize** value renders the cursor as an arrow pointing down.

## **se-resize**

The **se-resize** value renders the cursor as an arrow pointing to the lower right

## **sw-resize**

The **sw-resize** value renders the cursor as an arrow pointing to the lower left.

## **w-resize**

The **w-resize** value renders the cursor as an arrow pointing to the left.

## **text**

The **text** value usually renders the cursor as an I-bar to signify text.

## **url( ), ...**

The **url** value is a comma delimited list of one or more URLs that are the locations of custom cursors. Each URL must be enclosed within a pair of quotes. It is possible that a browser may not recognize a custom cursor. Therefore, it is recommended that a generic cursor be placed at the end of the list.

## **wait**

The **wait** value usually renders the cursor as a hourglass or watch to signify that the user should wait until the process is completed.

Code:

```
p {cursor: text;}
a {cursor: pointer;}
body {cursor: url("mycursor.gif"), url("images/cursors/footcursor.jpg"), default;}
```

# PROPERTY: direction

---

**direction** : `ltr` | `rtl`

**Compatibility:** IE5+

**Version:** Level 2

**Inherited:** Yes

The **direction** property allows you to choose the direction that the text will flow (i.e., left-to-right or right-to-left). For example, this can be applied to embedded text, quotes, and strings. It can also be applied to the order of the columns in a table or to dictate the placement of text when using the **text-align** property.

The default for the flow of text is left to the right, such as for English or Spanish. However, certain languages, such as Chinese and Hebrew, flow from the right to the left.

If you wish to apply the **direction** property to an inline text, you must set the **unicode-bidi** property either to the **bidi-override** or **embed** values.

## `ltr`

The **ltr** value sets the direction to be left-to-right. This is the default.

## `rtl`

The **rtl** value sets the direction to be right-to-left.

Code:

```
blockquote.eng { direction: ltr; }  
blockquote.heb { direction: rtl; }
```

or

```
<blockquote style="direction: rtl; unicode-bidi: bidi-override;">  
ABC DEF GHI JKL MNO PQR STV UWX YZ  
</blockquote>
```

Output:

ABC DEF GHI JKL MNO PQR STV UWX YZ

# PROPERTY: display

---

**display** : **block** | **compact** | **inline** | **inline-table** | **list-item** | **marker** | **none** | **run-in** | **table** | **table-caption** | **table-cell** | **table-column** | **table-column-group** | **table-footer-group** | **table-header-group** | **table-row** | **table-row-group**

**Compatibility:** IE4+ N4+

Most of the Level 2 additions are currently not supported by any browser

**Version:** Level 1/Expanded in Level 2

**Inherited:** No

The **display** property controls if or how an element will be displayed.

**block** Level 1

The **block** value is the default. It treats the element to be displayed as a block-level element, or block-level box, that has a line break before and after the element.

**compact** Level 2

The **compact** value assigns the element to be block-level or inline based upon the context.

**inline** Level 1

The **inline** value treats the element to be displayed as an inline element, or inline box, that does not have a line break before or after the element. In other words, everything is displayed on the same line.

**inline-table** Level 2

The **inline-table** value displays a table as an inline element, or inline box, that does not have a line break before or after the table.

**list-item** Level 1

The **list-item** value treats the element to be displayed as a list. Each item in the list can be preceded by an optional marker, such as the solid black circle,

- associated with the HTML tag.< li >

**marker** Level 2

The **marker** value assigns the content, before or after a box element, to be a marker. It is used with the **:after** and **:before** pseudo-elements.

**none** Level 1

The **none** value prevents the display of the element.

**run-in** Level 2

The **run-in** value assigns the element to be block-level or inline based upon the context.

## **table** Level 2

The **table** value displays the element as if it were a block-level table.

## **table-caption** Level 2

The **table-caption** value displays the element as if it were a table caption.

## **table-cell** Level 2

The **table-cell** value displays the element as if it were a table cell.

## **table-column** Level 2

The **table-column** value displays the element as if it were a table column.

## **table-column-group** Level 2

The **table-column-group** value displays the element as if it were a group of table columns.

## **table-footer-group** Level 2

The **table-footer-group** value displays the element as if it were a group of table footers.

## **table-header-group** Level 2

The **table-header-group** value displays the element as if it were a group of table headers.

## **table-row** Level 2

The **table-row** value displays the element as if it were a table row.

## **table-row-group** Level 2

The **table-row-group** value displays the element as if it were a group of table rows.

Code:

```
p { display: block; }
img { display: inline; }
li { display: list-item; }
img { display: none; }
```

or

In this example, the second image is not displayed:

Code:

```



```

Output:



# PROPERTY: empty-cells

---

**empty-cells** : [hide](#) | [show](#)

**Compatibility:** Currently not supported by any browser

**Version:** Level 2

**Inherited:** Yes

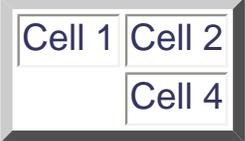
The **empty-cells** property determines whether a border will appear around any empty cells in a table. An empty cell has no visible content. This only applies to tables using the **separate** border model.

(While this property is currently not supported, by default, both Explorer and Netscape do not render an internal border around an empty cell.)

A cell that has the **visibility** property set to the **hidden** value is considered to not have visible content. Also, the carriage return, line feed, tab, and space are not considered to be visible content. An `&nbsp;` is visible content.

This is one of five Level 2 properties that specifically apply style to tables. The other four are **border-collapse**, **border-spacing**, **caption-side**, and **table-layout**.

In the **separate** model, the table has a separate outer border and each individual cell has its own separate internal border. The table outer border does not physically touch the individual internal cell borders.



Cell 1	Cell 2
	Cell 4

Remember that a border is optional in a table and that the default is no border.

## hide

The **hide** value dictates that a border will not appear around empty cells.

## show

The **show** value dictates that a border will appear around empty cells.

Code:

```
<table style="border-collapse: separate; empty-cells: hide;">
<tr><td>Cell 1</td><td>Cell 2</td></tr>
<tr><td></td><td>Cell 4</td></tr>
</table>
```

# PROPERTY: float

---

**float** : left | right | none

**Compatibility:** IE4+ N4+

**Version:** Level 1

**Inherited:** No

The **float** property allows you to determine where an image will appear within a text or element.

Image and text elements that appear embedded within a text or another element are called floating elements.

## left

The **left** value embeds the image within the text and places the left edge of the image on the left side of the text.

## right

The **right** value embeds the image within the text and places the right edge of the image on the right side of the text.

## none

The **none** value allows the image to simply appear where it occurs within a text.

Code:

```
img { float: left; }  
img { float: right; }  
img { float: none; }
```

or

Code:

```
  

```

Output:



# PROPERTY: font

---

**font** : font-style &| font-variant &| font-weight &| font-size &| | / line-height &| font-family | caption | icon | menu | message-box | small-caption | status-bar

**Compatibility:** IE4+ N4+

**Version:** Level 1/Expanded in Level 2

**Inherited:** Yes

The **font** property allows you to combine together in only one declaration any, or all of six individual CSS properties related to font. Each of these six individual properties is discussed on its own page.

In addition, effective with Level 2, the **font** property can also be used to declare six values that set system font settings.

**font-style font-variant font-weight font-size / line-height font-family**

For values, see the individual home pages.

**caption** Level 2

The **caption** value sets the font used by caption controls, such as buttons.

**icon** Level 2

The **icon** value sets the font used by icon labels.

**menu** Level 2

The **menu** value sets the font used in drop-down menu boxes.

**message-box** Level 2

The **message-box** value sets the font used in dialog boxes.

**small-caption** Level 2

The **small-caption** value sets the font used for small controls.

**status-bar** Level 2

The **status-bar** value sets the font used in the window status bars.

Note the forward slash before **/ line-height**. It must appear and is used in association with the font size. The use of the **/** is demonstrated in the first five code examples.

Code:

```
p { font: italic small-caps 600 12pts/18pts Courier; }
p { font: italic small-caps 600 12pts/150% Courier; }
p { font: italic small-caps 600 12pts/1.5 Courier; }
p { font: italic small-caps 600 12pts/18pts Courier; }
p { font: /18pts serif; }
p { font: oblique 100 24pts; }
```



# PROPERTY: font-family

---

**font-family** : family-name &| generic-family

**Compatibility:** IE4+ N4+

**Version:** Level 1

**Inherited:** Yes

The **font-family** property allows you to select specific typefaces or a generic family of typefaces.

You can list a prioritized selection of **family-name** and/or **generic-name** typefaces. The program will consider each typeface value in turn and will use the first value it finds that is recognized by the browser. In the list, each typeface value must be separated by a comma. This is one of the few places where commas are used in declaring values for a property. The use of a pair of double quotes to enclose the typeface name is optional. However, it is recommended that any typeface name, such as "helvetica extra bold", that contains blank (white) spaces, should be enclosed by double quotes. Note that single quotes must be used in place of double quotes when using certain HTML code.

## family-name

The **family-name** typeface values are the font names, such as: Courier, Arial, Times, Roman, etc. It is not possible to present a definitive list since not all computers have all typefaces installed.

## generic-family

There are five values of generic-name typefaces to choose from:

cursive  
fantasy  
monospace  
serif  
sans-serif

It is recommended that **generic-name** typeface values be used as a last resort.

Code:

```
p { font-family: Courier, "Zapf Dingbat", serif }
```

or

```
<span style="font-family: arial;">Arial </span>  
<span style="font-family: courier;">Courier </span>  
<span style="font-family: helvetica;">Helvetica </span>  
<span style="font-family: palatino;">Palatino </span>  
<span style="font-family: symbol;">Symbol </span>  
<span style="font-family: times;">Times </span>
```

output:

Arial Courier Helvetica Palatino Συμβολ Times

# PROPERTY: font-size

---

**font-size** : **absolute-size** | **relative-size** | **length** | **percentage**

**Compatibility:** IE4+ N4+

**Version:** Level 1

**Inherited:** Yes

The **font-size** property allows you to change the size of the font (i.e., characters in a text) by selecting from four different categories of values. The amount of change in size is a relative, not fixed, quantity that is determined by your individual computer and/or browser.

## **absolute-size**

There are seven steps in the **absolute-size** values to choose from:

xx-small  
x-small  
small  
medium  
large  
x-large  
xx-large

The amount of change caused by each step will be determined by the individual computer/browser.

## **relative-size**

There are two **relative-size** values to choose from:

**larger**  
**smaller**

The amount of change will be determined by the individual computer/browser.

## **length**

The **length** value can be in any of the following seven units. Use the abbreviation.

<b>Length</b>	<b>Abbrev.</b>
centimeters	cm
ems	em
inches	in
millimeters	mm
picas	pc
pixels	px
points	pt

## percentage

The **percentage** value is the percent amount you wish to increase or decrease the font size. Values less than 100% reduce the size of the font. 100% gives no change. Values greater than 100% increase the font size.

Code:

```
p { font-size: x-large; }  
p { font-size: smaller; }  
p { font-size: 18pt; }  
p { font-size: 3em; }  
p { font-size: 75%; }  
p { font-size: 150%; }
```

or

```
<span style="font-size: xx-small;">A </span>  
<span style="font-size: x-small;">A </span>  
<span style="font-size: small;">A </span>  
<span style="font-size: medium;">A </span>  
<span style="font-size: large;">A </span>  
<span style="font-size: x-large;">A </span>  
<span style="font-size: xx-large;">A </span>
```

A A A A A A A

# PROPERTY: font-size-adjust

---

**font-size-adjust** : none | number

**Compatibility:** Currently not supported by any browser

**Version:** Level 2

**Inherited:** Yes

The **font-size-adjust** property is used to assign the same aspect value for all of the fonts types assigned to an element.

Using such properties as **font** and **font-family**, you can create a prioritized listing of font types (arial, courier, etc.) that can be assigned to an element. When an .asp or .html page is to be displayed on a Web site, the browser will consider each typeface value in the list in turn and will use the first value that it recognizes.

When dealing with small font sizes, some fonts are easier to read than others. How easy it will be to read a small font is determined quantitatively by the font's aspect value, which is simply the ratio of the height of the lower case letter x over the height of the font size. A font with an aspect value of .58, such as Verdana, will be much easier to read at small sizes than a font with an aspect value of .28, such as Flemish Script.

The **font-size-adjust** property forces all of the fonts in the list to have the same aspect value. Therefore, regardless of which font is selected from the list, at small sizes it will be just as readable as any other font in the list.

## none

The **none** value allows each font to keep its own aspect value.

## number

The **number** value assigns the same aspect value to all fonts in the list.

Code:

```
p {font-family: arial, courier; font-size-adjust: none;}  
p {font-family: verdana, courier; font-size-adjust: .56;}
```

# PROPERTY: font-stretch

---

**font-stretch** : narrower | wider | ultra-condensed | extra-condensed | condensed | semi-condensed | normal | semi-expanded | expanded | extra-expanded | ultra-expanded

**Compatibility:** Currently not supported by any browser

**Version:** Level 2

**Inherited:** Yes

The **font-stretch** property is used to expand or contract (condense) the horizontal width of the font. The change is relative to the normal width of the font as displayed by the browser.

## narrower

The **narrower** value contracts the font to the next smaller width.

## wider

The **wider** value expands the font to the next larger width.

## ultra-condensed

## extra-condensed

## condensed

## semi-condensed

## normal

## semi-expanded

## expanded

## extra-expanded

## ultra-expanded

The order descends from narrowest to widest in value. The **normal** value is the normal width of the font as displayed by the browser.

Code:

```
p {font-stretch: wider;}
```

```
p {font-stretch: ultra-expanded;}
```

# PROPERTY: font-style

---

**font-style** : **normal** | **italic** | **oblique**

**Compatibility:** IE4+ N4+

**Version:** Level 1

**Inherited:** Yes

The **font-style** property allows you to designate a font to be **normal**, **italic** or **oblique**.

## normal

The **normal** value is the default and refers to the characters being upright. By declaring **normal**, you can ensure that any previous declaration will not effect a selected text.

## italic

If you select the **italic** value, and it is not available for a particular font, then the **oblique** value will be applied.

## oblique

The **oblique** value slants the characters.

Code:

```
p { font-style: normal; }  
p { font-style: italic; }  
p { font-style: oblique; }
```

or

```
<span style="font-style: normal;">Normal </span>  
<span style="font-style: italic;">Italic </span>  
<span style="font-style: oblique;">Oblique </span>
```

Output:

Normal Italic Oblique

# PROPERTY: font-variant

---

**font-variant** : normal | small-caps

**Compatibility:** IE4+ N6

**Version:** Level 1

**Inherited:** Yes

The **font-variant** property allows you to choose between a normal font that has both upper and lower case letters or a font that is composed of small capital letters.

## normal

The **normal** value is the default. By declaring **normal**, you can ensure that any previous declaration will not effect a selected text.

## small-caps

The **small-caps** value renders the entire text as small capital letters.

Code:

```
p { font-variant: normal; }  
p { font-variant: small-caps; }
```

or

```
<span style="font-variant: normal;">Normal </span>  
<span style="font-variant: small-caps;">Small Caps </span>
```

Output:

Normal Small Caps

# PROPERTY: font-weight

---

**font-weight** : **normal** | **bold** | **bolder** | **lighter** | **100** | **200** | **300** | **400** | **500** | **600** | **700** | **800** | **900**

**Compatibility:** IE4+ N4+

**Version:** Level 1

**Inherited:** Yes

The **font-weight** property allows you to choose how thick or thin the characters of a text appear. This is referred to as boldness. Boldness is a relative, not fixed, quantity which is determined by your individual computer and/or browser.

Many browsers only recognize normal and **bold** font weight.

## **normal**

The **normal** value is the default. On the numeric scale running from **100** to **900**, **normal** is usually considered to be **400**. By declaring **normal**, you can ensure that any previous declaration will not effect a selected text.

## **bold**

The **bold** value corresponds to the <B> tag in HTML. On the numeric scale running from **100** to **900**, **bold** is usually considered to be **700**.

## **bolder**

The **bolder** value makes the character appear thicker in size and, hence, increases the boldness. The amount of change will be determined by the individual computer/browser.

## **lighter**

The **lighter** value makes the character appear thinner in size and, hence, decreases the boldness. The amount of change will be determined by the individual computer/browser.

## **100 ... 900**

The **100,200,300,400,500,600,700,800,900** values provide nine steps in range of boldness. The larger the number, the greater the boldness. This is a relative numeric scale where **400** usually equates to **normal** and **700** to **bold**. The amount of change caused by each step will be determined by the individual computer/browser.

Code:

```
p { font-weight: normal; }
p { font-weight: bold; }
p { font-weight: bolder; }
p { font-weight: lighter; }
p { font-weight: 600; }
```

or

```
<span style="font-weight: 100;">100 </span>
<span style="font-weight: 200;">200 </span>
<span style="font-weight: 300;">300 </span>
```

```
<span style="font-weight: 400;">400 </span>  
<span style="font-weight: 500;">500 </span>  
<span style="font-weight: 600;">600 </span>  
<span style="font-weight: 700;">700 </span>  
<span style="font-weight: 800;">800 </span>  
<span style="font-weight: 900;">900 </span>
```

Output:

lighter

normal

**bold**

bolder

100

200

300

400

500

**600**

**700**

**800**

**900**

# PROPERTY: height

---

**height** : length | percentage | auto

**Compatibility:** IE4+ N6

**Version:** Level 1

**Inherited:** No

The **height** property allows you to set the height of an element of text or an image. For example, you could embed an image into a specified position within a text using the **float** property. Then use the **height** and/or **width** properties to specify an exact size for the image. If you set the **height** of an element to a specified value and set the **width** property to **auto**, the element will be scaled proportionally (i.e., you maintain the aspect ratio).

## length

The **length** value can be in any of the following seven dimensions. Use the abbreviation.

Length	Abbrev.
centimeters	cm
ems	em
inches	in
millimeters	mm
picas	pc
pixels	px
points	pt

## percentage

The **percentage** value is based upon the total height being defined as 100%.

## auto

The **auto** value directs the browser to automatically calculate the height. If both the **width** and **height** are set to **auto**, the dimensions of the text or image element remains unchanged. By declaring **auto**, you can ensure that any previous declaration will not effect the height of the selected element.

Code:

```
img { height: 150mm; }  
img { height: 2.85in; }  
img { height: 45pc; }  
img { height: 36pt; }  
img { height: 50%; }  
img { height: auto; }  
img {
```

```
height: 3.0in  
width: auto  
}
```

or

Code:

```
  
<br>  

```

Output:



# PROPERTY: left

---

**left** : auto | length | percentage

**Compatibility:** IE4+ N4+

**Version:** Level 2

**Inherited:** No

The **left** property sets the physical distance of how far the left content edge of an element is to the right from the left content edge of the containing block.

A containing block is simply an element that contains one or more related elements.

There are three other properties that allow you to set the distance for the **bottom**, **right**, and **top** content edges. All four properties are used in conjunction with the **position** property. Note that if the **position** property is set to the **static** value, setting the **left** property has no effect.

## auto

The **auto** value dictates that the browser sets the distance between the left content edges.

## length

The **length** value sets the distance between the left content edges and can be in any of the following seven units. Use the abbreviation.

Length	Abbrev.
centimeters	cm
ems	em
inches	in
millimeters	mm
picas	pc
pixels	px
points	pt

## percentage

The **percentage** value sets the distance between the left content edges as a percentage of the overall width of the parent block. It is a positive integer number. The percent sign is mandatory.

Code:

```
p.one {  
position: absolute;  
bottom: 1in;  
left: 1in;  
right: 1in;
```

```
top: 1in;  
}
```

# PROPERTY: letter-spacing

---

**letter-spacing** : normal | length

**Compatibility:** IE4+ N6

**Version:** Level 1

**Inherited:** Yes

The **letter-spacing** property increases or decreases the spacing (width) of the blank (white) space between adjacent characters or letters. A positive value increases the spacing. A negative value decreases the spacing.

## normal

The **normal** value is the default. With one exception, by declaring **normal**, you can ensure that any previous declaration will not effect the spacing. Even if **normal** is declared, the spacing could still be effected by justification by the **text-align** property. See the **length** value, below, for a way of turning off justification.

## length

The **length** value can be negative or positive. A value of zero prevents justification of the text from occurring, even if justification has been declared. The **length** can be in any of the following seven units. Use the abbreviation.

Length	Abbrev.
centimeters	cm
ems	em
inches	in
millimeters	mm
picas	pc
pixels	px
points	pt

Code:

```
p { letter-spacing: normal; }  
p { letter-spacing: 0; }  
br { letter-spacing: 0.1mm; }  
blockquote { letter-spacing: -0.2em; }
```

or

```
<span style="letter-spacing: 30px;">DevGuru</span>
```

Output:



# PROPERTY: line-height

---

**line-height** : **normal** | **number** | **length** | **percentage**

**Compatibility:** IE4+ N4+

**Version:** Level 1

**Inherited:** Yes

The **line-height** property defines the distance between adjacent lines. This distance is based upon the size of the font.

## normal

The **normal** value is the default. By declaring **normal**, you can ensure that any previous declaration will not effect **line-height**.

## number

The **number** value is multiplied times the font size to get the **line-height** dimension.

## length

The **length** value can be in any of the following seven units. Use the abbreviation.

Length	Abbrev.
centimeters	cm
ems	em
inches	in
millimeters	mm
picas	pc
pixels	px
points	pt

## percentage

The **percentage** value is based upon the normal line height being defined as 100%. A **percentage** value below 100% decreases the line height, 100% causes no change, and a value greater than 100% increases the line height.

Code:

```
p { line-height: normal; }  
p { line-height: 1.25; }  
br { line-height: 1.5in; }  
blockquote { line-height: ;85%; }
```

or

```
<span style="line-height: 30px;">  
DevGuru<br>  
is great!  
</span>
```

Output:

DevGuru

is great!

# PROPERTY: list-style

---

**list-style** : **list-style-image** &| **list-style-position** &| **list-style-type**

**Compatibility:** IE4+ N6

**Version:** Level 1

**Inherited:** Yes

The **list-style** property allows you to combine into one declaration the **list-style-image**, and/or the **list-style-position**, and/or the **list-style-type** properties.

Each of these three individual properties are discussed on their own page.

Code:

```
li { list-style: url("http://devguru.com/devgurupix.gif"), inside, circle; }
ul { list-style: outside, upper-roman; }
ol { list-style: square; }
```

# PROPERTY: list-style-image

---

**list-style-image** : `url("urladdress")` | `none`

**Compatibility:** IE4+ N6

**Version:** Level 1

**Inherited:** Yes

The **list-style-image** property is used to provide the location of an image that is used as a marker in a list. This property only applies to elements with **display** values of **list-item**.

**url("urladdress")**

The **url("urladdress")** value gives the address where the image is stored. You must enclose the url address inside a pair of parentheses and within a pair of quotes.

**none**

The default **none** value indicates that no image will be displayed. This is primarily used to turn off a previously selected image so that it will not be reused in another listing.

Code:

```
ul.out { list-style-position: outside; list-style-image: url("images/ie.gif"); }
```

...

```
<ul class="ie">
```

```
<li> ADO
```

```
<li> ASP
```

```
<li> Jet SQL
```

```
<li> WSH
```

```
</ul>
```

Output:

- ADO
- ASP
- Jet SQL
- WSH

# PROPERTY: list-style-position

---

**list-style-position** : **inside** | **outside**

**Compatibility:** IE4+ N6

**Version:** Level 1

**Inherited:** Yes

The **list-style-position** property determines where the list-item marker appears in a list with respect to each item in the list. This property only applies to elements with **display** values of **list-item**.

## **inside**

The **inside** value indents the list-item marker to the right into the list.

## **outside**

The default **outside** value keeps the list-item marker extended to the left out of the list.

Code:

```
ul.in { display: list-item; list-style-position: inside; }
```

Output:

- The effects of declaring several style properties can cascade together into creating the final appearance of the page.
- Neither Internet Explorer nor Netscape recognize all of the W3C standards for Level 1 or Level 2.

Code:

```
ul.out { display: list-item; list-style-position: outside; }
```

Output:

- The effects of declaring several style properties can cascade together into creating the final appearance of the page.
- Neither Internet Explorer nor Netscape recognize all of the W3C standards for Level 1 or Level 2.

# PROPERTY: list-style-type

---

**list-style-type** : **disc** | **circle** | **square** | **decimal** | **lower-roman** | **upper-roman** | **lower-alpha** | **upper-alpha** | **none** | **armenian** | **CJK-ideographic** | **georgian** | **lower-greek** | **hebrew** | **hiragana** | **hiragana-iroha** | **katakana** | **katakana-iroha** | **lower-latin** | **upper-latin**

**Compatibility:** IE4+ N4+

The level 2 values are currently not recognized by any browser

**Version:** Level 1/Expanded in Level2

**Inherited:** Yes

The **list-style-type** property allows you to select the type of list-item marker that you use in a listing. There are three types of list-item markers: alphabetic, glyphs, and numeric. For unordered lists, you can choose **disc**, **circle**, or **square**. For ordered lists, you can select any of the alphabetic or numeric values. For any type of list, you can choose **none**.

This property only applies to elements with **display** values of **list-item**.

Many browsers will only recognize and display the solid black **disc**, regardless of the list-item marker you select.

**lower-alpha** Level 1

- This is lower-alpha (a, b, c, etc.).

**upper-alpha** Level 1

- This is upper-alpha (A, B, C, etc.).

**circle** Level 1

- This is a circle.

**decimal** Level 1

- This is decimal (1, 2, 3, etc.).

**disc** Level 1

- This is a disc marker. It is the default marker.

**none** Level 1

- This is no marker.

**lower-roman** Level 1

- This is lower-Roman (i, ii, iii, iv, etc.).

## **upper-roman** Level 1

- This is upper-Roman (i, li, Ili, IV, etc.).

## **square** Level 1

- This is a square.

## **armenian** Level 2

The **armenian** value uses traditional Armenian numbers as markers.

## **cjk-ideographic** Level 2

The **cjk-ideographic** value uses plain ideographic numbers as markers.

## **georgian** Level 2

The **georgian** value uses tradition Georgian numbers as markers.

## **lower-greek** Level 2

The **lower-greek** value uses classic lower-case Greek characters as markers.

## **hebrew** Level 2

The **hebrew** value uses traditional Hebrew numbers as markers.

## **hiragana** Level 2

The **hiragana** value uses the Japanese Hiragana character list as markers.

## **hiragana-iroha** Level 2

The **hiragana-iroha** value uses the Japanese Hiragana-iroha ordering as markers.

## **katakana** Level 2

The **katakana** values uses the Japanese Katakana character list as markers.

## **katakana-iroha** Level 2

The **katakana-iroha** value uses the Japanese Katakana-iroha ordering as markers.

## **lower-latin** Level 2

The **lower-latin** value uses lower-case Latin characters as markers.

## **upper-latin** Level 2

The **upper-latin** value uses upper-case Latin characters as markers.

Code:

```
li { list-style-type: disc; }
```

```
li { list-style-type: circle; }  
li { list-style-type: square; }  
li { list-style-type: decimal; }  
li { list-style-type: lower-roman; }  
li { list-style-type: upper-roman; }  
li { list-style-type: lower-alpha; }  
li { list-style-type: upper-alpha; }  
li { list-style-type: none; }
```

or

```
<li style="list-style-type: square"> ADO </li>  
<li style="list-style-type: square"> ASP </li>  
<li style="list-style-type: square"> WML </li>  
<li style="list-style-type: square"> WSH </li>
```

Output:

- ADO
- ASP
- WML
- WSH

# PROPERTY: margin

---

**margin** : **length** | **percentage** | **auto**

**Compatibility:** IE4+ N4+

**Version:** Level 1

**Inherited:** No

The **margin** property is a shortcut for setting the **margin-bottom**, **margin-left**, **margin-right**, and/or **margin-top** properties in one declaration.

You can declare one, two, three or four values:

If you declare one value, the size for all four margins will be that value.

If you declare two values, the top and bottom margins will be set by the first value, the right and left margins by the second.

If you declare three values, the top margin will be set by the first value, the right margin by the second, and the bottom and left margins by the third.

If you declare four values, the order is top, right, bottom, left.

## **length**

The **length** value can be in any of the following seven units. Use the abbreviation.

<b>Length</b>	<b>Abbrev.</b>
centimeters	cm
ems	em
inches	in
millimeters	mm
picas	pc
pixels	px
points	pt

## **percentage**

The **percentage** value is based upon the total width being defined as 100%.

## **auto**

The **auto** value directs the browser to automatically calculate all four margins. By declaring **auto**, you can ensure that any previous declaration will not effect the margins.

Code:

```
body { margin: 25mm; }
body { margin: 1.0in; }
body { margin: 15pc 25pc; }
body { margin: 36pt 24pt 36pt; }
body { margin: 11.5%; }
body { margin: 10% 10% 10% 10%; }
body { margin: auto; }
```

# PROPERTY: margin-bottom

---

**margin-bottom** : **length** | **percentage** | **auto**

**Compatibility:** IE4+ N4+

**Version:** Level 1

**Inherited:** No

The **margin-bottom** property allows you to set the dimensions for the bottom margin.

## length

The **length** value can be in any of the following seven units. Use the abbreviation.

Length	Abbrev.
centimeters	cm
ems	em
inches	in
millimeters	mm
picas	pc
pixels	px
points	pt

## percentage

The **percentage** value is based upon the total height being defined as 100%.

## auto

The **auto** value directs the browser to automatically calculate the bottom margin. By declaring **auto**, you can ensure that any previous declaration will not effect the bottom margin.

Code:

```
p { margin-bottom: 25mm; }  
p { margin-bottom: 1.0in; }  
p { margin-bottom: 15pc; }  
p { margin-bottom: 36pt; }  
p { margin-bottom: 11.5%; }  
p { margin-bottom: auto; }
```

# PROPERTY: margin-left

---

**margin-left** : **length** | **percentage** | **auto**

**Compatibility:** IE4+ N4+

**Version:** Level 1

**Inherited:** No

The **margin-left** property allows you to set the dimensions for the left margin.

## length

The **length** value can be in any of the following seven units. Use the abbreviation.

Length	Abbrev.
centimeters	cm
ems	em
inches	in
millimeters	mm
picas	pc
pixels	px
points	pt

## percentage

The **percentage** value is based upon the total width being defined as 100%.

## auto

The **auto** value directs the browser to automatically calculate the left margin. By declaring **auto**, you can ensure that any previous declaration will not effect the left margin.

Code:

```
p { margin-left: 25mm; }  
p { margin-left: 1.0in; }  
p { margin-left: 15pc; }  
p { margin-left: 36pt; }  
p { margin-left: 11.5%; }  
p { margin-left: auto; }
```

# PROPERTY: margin-right

---

**margin-right** : **length** | **percentage** | **auto**

**Compatibility:** IE4+ N4+

**Version:** Level 1

**Inherited:** No

The **margin-right** property allows you to set the dimensions for the right margin.

## length

The **length** value can be in any of the following seven units. Use the abbreviation.

Length	Abbrev.
centimeters	cm
ems	em
inches	in
millimeters	mm
picas	pc
pixels	px
points	pt

## percentage

The **percentage** value is based upon the total width being defined as 100%.

## auto

The **auto** value directs the browser to automatically calculate the right margin. By declaring **auto**, you can ensure that any previous declaration will not effect the right margin.

Code:

```
p { margin-right: 25mm; }  
p { margin-right: 1.0in; }  
p { margin-right: 15pc; }  
p { margin-right: 36pt; }  
p { margin-right: 11.5%; }  
p { margin-right: auto; }
```

# PROPERTY: margin-top

---

**margin-top** : length | percentage | auto

**Compatibility:** IE4+ N4+

**Version:** Level 1

**Inherited:** No

The **margin-top** property allows you to set the dimensions for the top margin.

## length

The **length** value can be in any of the following seven units. Use the abbreviation.

Length	Abbrev.
centimeters	cm
ems	em
inches	in
millimeters	mm
picas	pc
pixels	px
points	pt

## percentage

The **percentage** value is based upon the total height being defined as 100%.

## auto

The **auto** value directs the browser to automatically calculate the top margin. By declaring **auto**, you can ensure that any previous declaration will not effect the top margin.

Code:

```
p { margin-top: 25mm; }  
p { margin-top: 1.0in; }  
p { margin-top: 15pc; }  
p { margin-top: 36pt; }  
p { margin-top: 11.5%; }  
p { margin-top: auto; }
```

# PROPERTY: marker-offset

---

**marker-offset** : **auto** | **length**

**Compatibility:** Currently not supported by any browser

**Version:** Level 2

**Inherited:** No

The **marker-offset** property specifies the horizontal distance (called the offset) between a marker box and the principal box. This distance is measured between the nearest border edges of the two boxes.

One method for displaying a list is to create a principal box and a marker box. An item in the list is displayed in the principal box, while the associated marker box contains a number, image, or decoration such a bullet. This offers the possibility of creating unique list styles. Also, the marker boxes can be used with counters (via the **content** property).

The **marker-offset** property is used to position the marker box precisely in relation to the principal box. The marker box can occur either inside or outside the principal box, but the position of the marker box does effect the position of the principal box.

In contrast, the various CSS list-style properties offer the ability to create simple lists, but lack the potential sophistication of marker/principal boxes.

## auto

The **auto** value allows the browser to automatically set the distance.

## length

The **length** value is the horizontal distance between the two boxes. It can be positive or negative and can be in any of the following seven units. Use the abbreviation.

Length	Abbrev.
centimeters	cm
ems	em
inches	in
millimeters	mm
picas	pc
pixels	px
points	pt

Code:

```
<style type="text/css">
li:before
{
```

```
display: marker;  
marker-offset: 5px;  
}  
</style>
```

# PROPERTY: max-height

---

**max-height** : none | length | percentage

**Compatibility:** Currently not supported by any browser

**Version:** Level 2

**Inherited:** No

The **max-height** property is used to set the maximum height of an element. Other properties can be used to set the height, width, maximum width, minimum height and the minimum width.

If a specified value for **max-height** is less than a previously specified value for **min-height** for the same element, then **max-height** is set to the value of **min-height**. If the specified value for **min-height** is greater than the **height** value for the same element, then **height** is set to the value of **min-height**.

## none

The **none** value specifies that there are no height limits imposed on the element.

## length

The **length** value specifies the maximum height of the element and can be in any of the following seven units. Use the abbreviation.

Length	Abbrev.
centimeters	cm
ems	em
inches	in
millimeters	mm
picas	pc
pixels	px
points	pt

## percentage

The **percentage** value specifies the height of the element as a percentage of the height of the containing block (i.e., the block that contains the element). It is a positive integer number. The percent sign is mandatory.

Code:

```
p {max-height: 175px;}  
p {max-height: 200%;}
```



# PROPERTY: max-width

---

**max-width** : none | length | percentage

**Compatibility:** Currently not supported by any browser

**Version:** Level 2

**Inherited:** No

The **max-width** property is used to set the maximum width of an element. Other properties can be used to set the height, width, maximum height, minimum height and the minimum width.

If a specified value for **max-width** is less than a previously specified value for **min-width** for the same element, then **max-width** is set to the value of **min-width**. If the specified value for **min-width** is greater than the **width** value for the same element, then **width** is set to the value of **min-width**.

## none

The **none** value specifies that there are no width limits imposed on the element.

## length

The **length** value specifies the maximum width of the element and can be in any of the following seven units. Use the abbreviation.

Length	Abbrev.
centimeters	cm
ems	em
inches	in
millimeters	mm
picas	pc
pixels	px
points	pt

## percentage

The **percentage** value specifies the width of the element as a percentage of the width of the containing block (i.e., the block that contains the element). It is a positive integer number. The percent sign is mandatory.

Code:

```
p {max-width: 210px;}  
p {max-width: 200%;}
```



# PROPERTY: min-height

---

**min-height** : **length** | **percentage**

**Compatibility:** Currently not supported by any browser

**Version:** Level 2

**Inherited:** No

The **min-height** property is used to set the minimum height of an element. Other properties can be used to set the height, width, maximum height, maximum width and the minimum width.

If a specified value for **max-height** is less than a previously specified value for **min-height** for the same element, then **max-height** is set to the value of **min-height**. If the specified value for **min-height** is greater than the **height** value for the same element, then **height** is set to the value of **min-height**.

## length

The **length** value specifies the maximum height of the element and can be in any of the following seven units. Use the abbreviation.

Length	Abbrev.
centimeters	cm
ems	em
inches	in
millimeters	mm
picas	pc
pixels	px
points	pt

## percentage

The **percentage** value specifies the height of the element as a percentage of the height of the containing block (i.e., the block that contains the element). It is a positive integer number. The percent sign is mandatory.

Code:

```
p {min-height: 175px;}  
p {min-height: 200%;}
```

# PROPERTY: min-width

---

**min-width** : **length** | **percentage**

**Compatibility:** Currently not supported by any browser

**Version:** Level 2

**Inherited:** No

The **min-width** property is used to set the minimum width of an element. Other properties can be used to set the height, width, maximum height, maximum width, and the minimum height.

If a specified value for **max-width** is less than a previously specified value for **min-width** for the same element, then **max-width** is set to the value of **min-width**. If the specified value for **min-width** is greater than the **width** value for the same element, then **width** is set to the value of **min-width**.

## length

The **length** value specifies the minimum width of the element and can be in any of the following seven units. Use the abbreviation.

Length	Abbrev.
centimeters	cm
ems	em
inches	in
millimeters	mm
picas	pc
pixels	px
points	pt

## percentage

The **percentage** value specifies the width of the element as a percentage of the width of the containing block (i.e., the block that contains the element). It is a positive integer number. The percent sign is mandatory.

Code:

```
p {min-width: 94px;}  
p {min-width: 150%;}
```

# PROPERTY: outline

---

**outline** : **outline-color** &| **outline-style** &| **outline-width**

**Compatibility:** Currently not supported by any browser

**Version:** Level 2

**Inherited:** No

The **outline** property is a shortcut that allows you to combine the **outline-color**, **outline-style**, and **outline-width** properties in one declaration.

Each of these three individual properties is discussed on its own page.

An outline is a line drawn around an element with the purpose of making that element stand out (i.e., become more noticeable on a page). The outline is drawn outside of the border edge of the element. Therefore, an element can have both a border and an outline. Unlike a border, an outline does not have to have a rectangular shape.

An interesting use of an outline is to have it display only when the element comes into focus or when the element is active. This is accomplished with the following code fragment by using the psuedo-elements **:focus** and **:active**.

```
p:focus { outline: blue solid thin }  
p:active { outline: red solid thick }
```

Remember that each of these three properties is optional.

Code:

```
img { outline: red }  
p { outline: double 5px }  
button { outline: #E9E9E9 double thin }
```

# PROPERTY: outline-color

**outline-color** : **color** | **invert**

**Compatibility:** Currently not supported by any browser

**Version:** Level 2

**Inherited:** No

The **outline-color** property sets the color of the outline.

An outline is a line drawn around an element with the purpose of making that element stand out (i.e., become more noticeable on a page). You can set the color, style, and width of the line. The outline is drawn outside of the border edge of the element. Therefore, an element can have both a border and an outline. Unlike a border, an outline does not have to have a rectangular shape.

## color

The **color** value can be the keyword color name, the hex six-digit number (#FFFFFF), or the RGB three-digit value (255,255,255). There are sixteen standard colors in HTML:

<b>aqua</b> #00FFFF rgb(0,255,255)	<b>gray</b> #808080 rgb(128,128,128)	<b>silver</b> #C0C0C0 rgb(192,192,192)	<b>navy</b> #000080 rgb(0,0,128)
<b>black</b> #000000 rgb(0,0,0)	<b>green</b> #800000 rgb(128,0,0)	<b>teal</b> #008080 rgb(0,128,128)	<b>olive</b> #808000 rgb(128,128,0)
<b>blue</b> #0000FF rgb(0,0,255)	<b>lime</b> #00FF00 rgb(0,255,0)	<b>white</b> #FFFFFF rgb(255,255,255)	<b>purple</b> #800080 rgb(128,0,128)
<b>fuchsia</b> #FF00FF rgb(255,0,255)	<b>maroon</b> #800000 rgb(128,0,0)	<b>yellow</b> #FFFF00 rgb(255,255,0)	<b>red</b> #FF0000 rgb(255,0,0)

The **Guru** has also created a [Color Chart](#) that displays all of the named HTML colors. In addition the hexadecimal code is provided for 256 colors.

## invert

The **invert** value performs a color inversion on the background color and uses the inverted color for the outline. This insures that the outline color will stand out from the background.

Code:

```
img { outline-color: red }  
p { outline-color: #E9E9E9 }
```

# PROPERTY: outline-style

---

**outline-style** : none | dotted | dashed | solid | double | groove | ridge | inset | outset

**Compatibility:** Currently not supported by any browser

**Version:** Level 2

**Inherited:** No

The **outline-style** property sets the physical appearance of the outline (i.e., solid line, dashed, etc.).

An outline is a line drawn around an element with the purpose of making that element stand out (i.e., become more noticeable on a page). You can set the color, style, and width of the line. The outline is drawn outside of the border edge of the element. Therefore, an element can have both a border and an outline. Unlike a border, an outline does not have to have a rectangular shape.

## none

The **none** value is the default. If no style is declared, no outline will appear even if other outline properties have been set. By declaring **none**, you can ensure that any previous outline declarations will not affect the outline.

**dotted** A dotted line outlines the element.

**dashed** A dashed line outlines the element.

**double** A double solid line outlines the element.

**groove** A 3-D grooved line outlines the element. The exact appearance of the line depends on the selected **outline-color** value.

**inset** A 3-D inset line outlines the element. The exact appearance of the line depends on the selected **outline-color** value.

**outset** A 3-D outset line outlines the element. The exact appearance of the line depends on the selected **outline-color** value.

**ridge** A 3-D ridged line outlines the element. The exact appearance of the line depends on the selected **outline-color** value.

**solid** A solid line outlines the element.

Code:

```
img
{
  outline-color: orange
  outline-style: solid
  outline-width: medium
}
```

# PROPERTY: outline-width

---

**outline-width** : **thin** | **medium** | **thick** | **length**

**Compatibility:** Currently not supported by any browser

**Version:** Level 2

**Inherited:** No

The **outline-width** property sets the physical width of the outline.

An outline is a line drawn around an element with the purpose of making that element stand out (i.e., become more noticeable on a page). You can set the color, style, and width of the line. The outline is drawn outside of the border edge of the element. Therefore, an element can have both a border and an outline. Unlike a border, an outline does not have to have a rectangular shape.

## length

The **length** value specifies the width of the outline and can be in any of the following seven units. Use the abbreviation.

Length	Abbrev.
centimeters	cm
ems	em
inches	in
millimeters	mm
picas	pc
pixels	px
points	pt

## medium

The **medium** value creates a medium width outline.

## thick

The **thick** value creates a thick width outline.

## thin

The **thin** value creates a thin width outline.

Code:

```
img  
{  
  outline-color: orange  
  outline-style: solid
```

```
outline-width: medium
```

```
}
```

# PROPERTY: overflow

---

**overflow** : **auto** | **hidden** | **scroll** | **visible**

**Version:** Level 2

**Inherited:** No

The **overflow** property allows you to select whether an over-sized element in a containing block is clipped or not.

A containing block is simply an element that contains related sub-elements. A problem arises if a sub-element is too big in size to fit inside the dimensions of the containing block. By default, when an element is too large to fit in the allotted space, portions will be chopped off so that the remaining portion of the element will fit inside the containing box. The portions that are chopped off are simply not displayed.

The **clip** property allows you to specify what portion of the over-sized sub-element will be visible. This is referred to as clipping the sub-element.

## auto

The **auto** value allows the over-sized element to be clipped and scroll bars to be used to see the rest of the element.

## hidden

The **hidden** value dictates that only the clipped portion of an over-sized element will be visible. It will be completely contained inside the containing block, and scroll bars will not be displayed.

## scroll

The **scroll** value dictates that the over-sized element is clipped and that scroll bars will be displayed that allow you to view the rest of the element by scrolling.

## visible

The **visible** value dictates that the over-sized element is not clipped. The entire element will be displayed even though it extends beyond the boundaries of the containing block. If this value is selected, then setting the **clip** property should have no effect.

Code:

```
div.over {  
  overflow: scroll;  
  height: 100px;  
  width: 100px;  
}  
...  
<div class="over">  
  
</div>
```

Output:





# PROPERTY: padding

---

**padding** : **length** | **percentage**

**Compatibility:** IE4+ N4+

**Version:** Level 1

**Inherited:** No

The **padding** property is a shortcut that allows you to set the values for **padding-bottom**, **padding-left**, **padding-right**, and **padding-top** in only one declaration.

You can declare one, two, three, or four values:

If you declare one value, the amount of padding between all four borders and the contents will be that value.

If you declare two values, the top and bottom padding will be set by the first value, the right and left padding by the second value.

If you declare three values, the top padding will be set by the first value, the right padding by the second, and the bottom and left padding by the third.

If you declare four values, the order is top, right, bottom, left.

## length

The **length** value can be in any of the following seven units. Use the abbreviation.

Length	Abbrev.
centimeters	cm
ems	em
inches	in
millimeters	mm
picas	pc
pixels	px
points	pt

## percentage

The **percentage** value is based upon the total width being defined as 100%.

Code:

```
body { padding: 25mm; }
```

```
body { padding: 1.0in; }
```

```
body { padding: 16pc 25pc; }
```

```
body { padding: 36pt 24pt 36pt; }
```

```
body { padding: 11.5%; }  
body { padding: 10% 10% 10% 10%; }
```

or

```
<table border="2">  
<tr>  
<td style="padding: 22px 22px 22px 22px;">  
22 pixels padding on all four sides  
</td>  
</tr>  
</table>
```

Output:

22 pixels padding on all four sides
-------------------------------------

# PROPERTY: padding-bottom

---

**padding-bottom** : length | percentage

**Compatibility:** IE4+ N4+

**Version:** Level 1

**Inherited:** No

The **padding-bottom** property allows you to insert padding (space) between the bottom border and the enclosed text or images.

## length

The **length** value can be in any of the following seven units. Use the abbreviation.

Length	Abbrev.
centimeters	cm
ems	em
inches	in
millimeters	mm
picas	pc
pixels	px
points	pt

## percentage

The **percentage** value is based upon the total height being defined as 100%.

Code:

```
body { padding-bottom: 25mm; }
body { padding-bottom: 1.0in; }
body { padding-bottom: 15pc; }
body { padding-bottom: 36pt; }
body { padding-bottom: 11.5%; }
```

or

```
<table border="2">
<tr>
<td style="padding-bottom: 22px;">
Bottom padding
</td>
</tr>
</table>
```

Output:

Bottom padding

# PROPERTY: padding-left

---

**padding-left** : **length** | **percentage**

**Compatibility:** IE4+ N4+

**Version:** Level 1

**Inherited:** No

The **padding-left** property allows you to insert padding (space) between the left border and the enclosed text or images.

## length

The **length** value can be in any of the following seven units. Use the abbreviation.

Length	Abbrev.
centimeters	cm
ems	em
inches	in
millimeters	mm
picas	pc
pixels	px
points	pt

## percentage

The **percentage** value is based upon the total width being defined as 100%.

Code:

```
body { padding-left: 25mm; }
body { padding-left: 1.0in; }
body { padding-left: 15pc; }
body { padding-left: 36pt; }
body { padding-left: 11.5%; }
```

or

```
<table border="2">
<tr>
<td style="padding-left: 22px;">
left padding
</td>
</tr>
</table>
```

Output:

left padding

# PROPERTY: padding-right

---

**padding-right** : length | percentage

**Compatibility:** IE4+ N4+

**Version:** Level 1

**Inherited:** No

The **padding-right** property allows you to insert padding (space) between the right border and the enclosed text or images.

## length

The **length** value can be in any of the following seven units. Use the abbreviation.

Length	Abbrev.
centimeters	cm
ems	em
inches	in
millimeters	mm
picas	pc
pixels	px
points	pt

## percentage

The **percentage** value is based upon the total width being defined as 100%.

Code:

```
body { padding-right: 25mm; }
body { padding-right: 1.0in; }
body { padding-right: 15pc; }
body { padding-right: 36pt; }
body { padding-right: 11.5%; }
```

or

```
<table border="2">
<tr>
<td style="padding-right: 22px;">
right padding
</td>
</tr>
</table>
```

Output:

right padding

# PROPERTY: padding-top

---

**padding-top** : **length** | **percentage**

**Compatibility:** IE4+ N4+

**Version:** Level 1

**Inherited:** No

The **padding-top** property allows you to insert padding (space) between the top border and the enclosed text or images.

## length

The **length** value can be in any of the following seven units. Use the abbreviation.

Length	Abbrev.
centimeters	cm
ems	em
inches	in
millimeters	mm
picas	pc
pixels	px
points	pt

## percentage

The **percentage** value is based upon the total height being defined as 100%.

Code:

```
body { padding-top: 25mm; }
body { padding-top: 1.0in; }
body { padding-top: 15pc; }
body { padding-top: 36pt; }
body { padding-top: 11.5%; }
```

or

```
<table border="2">
<tr>
<td style="padding-top: 22px;">
top padding
</td>
</tr>
</table>
```

Output:

top padding

# PROPERTY: position

---

**position** : **absolute** | **fixed** | **relative** | **static**

**Compatibility:** IE4+ N4+

**Version:** Level 2

**Inherited:** No

The **position** property is one of two properties that can be used to set the physical position of an element in terms of where and how it will appear on a page. The other positioning property is **float**. These two properties allow you to select among the three possible positioning schemes in CSS, which are:

- **normal flow** The elements are displayed in the normal order that they occur.
- **floats** The element is shifted to the right or left. Other elements, such as text, can optionally flow around the side.
- **absolute positioning** The element is offset a specified distance with respect to the position it has in the normal flow.

The **position** property can be used in conjunction with the **bottom**, **left**, **right**, and the **top** properties.

## **absolute**

The **absolute** value allows an element to be placed anywhere on a page. The position is set using the **bottom**, **left**, **right**, or **top** properties.

## **fixed**

The **fixed** value is currently not recognized by any browser.

## **relative**

The **relative** value moves the element an offset distance relative to the position it has in the normal flow of the display of the page.

## **static**

The **static** value dictates that an element will be positioned as it occurs in the normal flow of the display of the page.

Code:

```
p.one {  
position: absolute;  
bottom: 1in;  
left: 1in;  
right: 1in;  
top: 1in;  
}
```

# PROPERTY: quotes

---

**quotes** : none | string string ... ..

**Compatibility:** Currently not supported by any browser

**Version:** Level 2

**Inherited:** Yes

The **quotes** property is a blank space delimited list of one or more types of opening and closing quotations marks that will be used in each successive level of embedded quotes. You use the **content** property to set the quotation marks before and after a specified element. This allows the selection of various types of quotation marks in a style-sensitive and context-dependent manner. (Note that you do not have to just use quotation marks. Since you are simply providing a string, you could use <, >, \*, ?, and other characters.)

## none

The **none** value prevents the **content** property from displaying quotation marks.

## string string

The **string string** values occur in pairs and are used to define pairs of opening and closing quotation marks. The first string in the pair defines the opening quotation mark. The second string in the pair defines the closing quotation mark. Each quotation mark being defined is treated as a string and must be enclosed within a pair of double quotes with no white space. If you include white space, it will appear in the output.

When a new level of embedded quotes is encountered, the next pair of **string** values in the list is referenced to provide the type of quotation marks to use.

In this example, the first pair are double quotes, the second pair are single quotes.  
(i.e., quotes: DDD DDD DSD DSD;)

Code:

```
<html>
<style type="text/css">
q { quotes: "" "" "" ""; }
</style>
<q>
This is an <q>emdedded</q> quote.
</q>
</html>
```

Expected Output:

"This is an 'emdedded' quote."

# PROPERTY: right

---

**right** : auto | length | percentage

**Compatibility:** IE5+

**Version:** Level 2

**Inherited:** No

The **right** property sets the physical distance of how far the right content edge of an element is to the left from the right content edge of the containing block.

A containing block is simply an element that contains one or more related elements.

There are three other properties that allow you to set the distance for the **bottom**, **left**, and **top** content edges. All four properties are used in conjunction with the **position** property. Note that if the **position** property is set to the **static** value, setting the **right** property has no effect.

## auto

The **auto** value sets the distance between the right content edges.

## length

The **length** value sets the distance between the right content edges and can be in any of the following seven units. Use the abbreviation.

Length	Abbrev.
centimeters	cm
ems	em
inches	in
millimeters	mm
picas	pc
pixels	px
points	pt

## percentage

The **percentage** value sets the distance between the right content edges as a percentage of the overall width of the parent block. It is a positive integer number. The percent sign is mandatory.

Code:

```
p.one {  
position: absolute;  
bottom: 1in;  
left: 1in;  
right: 1in;
```

```
top: 1in;  
}
```

# PROPERTY: table-layout

---

**table-layout** : **auto** | **fixed**

**Compatibility:** IE5.5

**Version:** Level 2

**Inherited:** Yes

The **table-layout** property allows you to choose the layout algorithm for constructing the table.

This is one of five Level 2 properties that specifically apply style to tables. The other four are **border-collapse**, **border-spacing**, **caption-side**, and **empty-cells**.

There are two type of layout algorithms, **auto** and **fixed**.

## **auto**

The **auto** value dictates the use of the **auto** layout algorithm. In this algorithm, the layout is dependent upon the contents of the individual cells. This requires that the browser first access all of the table contents to determine the layout for each column before the table can be rendered. As a consequence, this algorithm is considered slower than the **fixed** algorithm.

## **fixed**

The **fixed** value dictates the use of the **fixed** layout algorithm. In this algorithm, the horizontal layout is only dependent on the table's width, the width of the columns, the width of any borders, and the cell spacing. It does not depend on the content of the table cells.

Code:

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

or

```
<table style="table-layout: auto;">
```

```
...
```

```
</table>
```

# PROPERTY: text-align

---

**text-align** : left | right | center | justify

**Compatibility:** IE4+ N4+

**Version:** Level 1

**Inherited:** Yes

The **text-align** property aligns a specified selection of text.

## left

The **left** value is the default. The text will be aligned along its left side.

## right

The **right** value aligns the text along its right side.

## center

The **center** value aligns each line of the text in the center with an equal amount of blank (white) space on each side of the line.

## justify

The **justify** value adds blank (white) space between adjacent words and characters in order to align both the right and left sides of a block of text. The **letter-spacing** property can override justification.

Code:

```
p.one { text-align: left; }
p.two { text-align: center; }
p.three { text-align: right; }
...
<p class="one">DevGuru</p>
<p class="two">is</p>
<p class="three">great!</p>
```

Output:

DevGuru

is

great!

# PROPERTY: text-decoration

---

**text-decoration** : none | [ underline &| overline &| line-through &| blink ]

**Compatibility:** IE4+ N4+

**Version:** Level 1

**Inherited:** No

The **text-decoration** property allows you to underline, and/or overline, and/or draw a line through a specified selection of text, and/or cause the text to blink.

## none

The **none** value is the default. By declaring **none**, you can ensure that any previous declaration will not effect the selected text.

## underline

The **underline** value draws a line under the text. If the text is composed of more than one color, the **underline** will be the color of the first element of the text.

## overline

The **overline** value draws a line over the text. If the text is composed of more than one color, the **overline** will be the color of the first element of the text.

## line-through

The **line-through** value draws a line through the middle the text. If the text is composed of more than one color, the **line-through** will be the color of the first element of the text.

## blink

The **blink** value causes the selected text to flash in and out of view. It is recommended that you use the **blink** value sparingly. Many people dislike blinking text on a web site.

Code:

```
p { text-decoration: none; }
p { text-decoration: underline; }
p { text-decoration: overline; }
p { text-decoration: line-through; }
pre { text-decoration: blink; }
q { text-decoration: underline blink; }
blockquote { text-decoration: underline line-through blink; }
br {text-decoration: underline overline line-through blink}
```

or

The answer is a firm NO!

Output:

The answer is a firm NO!

# PROPERTY: text-indent

---

**text-indent** : **length** | **percentage**

**Compatibility:** IE4+ N4+

**Version:** Level 1

**Inherited:** Yes

The **text-indent** property indents the first line of text to the right or left by the specified **length** or **percentage**. If the **length** or **percentage** is a negative number, the line is indented to the left. A positive number indents to the right.

## length

The **length** value can be in any of the following seven units. Use the abbreviation.

Length	Abbrev.
centimeters	cm
ems	em
inches	in
millimeters	mm
picas	pc
pixels	px
points	pt

## percentage

The **percentage** value is based upon the total width being defined as 100%.

Code:

```
p { text-indent: 0.5in; }
p { text-indent: -15pc; }
p { text-indent: 24pt; }
p { text-indent: 5.5%; }
p { text-indent: -8%; }
```

or

```
pone { text-indent: 25px; }
```

...

```
<p>
```

We are a premier developers' resource featuring over three thousand pages containing comprehensive quick reference guides, tutorials, knowledge base articles, and useful products to serve a wide range of developers' needs.

```
</p>
```

Output:

We are a premier developers' resource featuring over three thousand pages containing comprehensive quick reference guides, tutorials, knowledge base articles, and useful products to serve a wide range of developers' needs.

# PROPERTY: text-shadow

---

**text-shadow** : **color** &| **length length length**, ... &| ... .. ,

**Compatibility:** Currently not supported by any browser

**Version:** Level 2

**Inherited:** No

The **text-shadow** property is a comma delimited list that sets each color, blur, and shadow effect you wish to apply to the associated character, word, line, or text. Each effect is applied in the provided order (i.e., the listed order is also the stacking order where the subsequent effect lays on top of the previous effect). The shadow effects always appear to lay under the text. You can optionally set a color for each effect. You can also set a horizontal and vertical distance that the effects will extend either to the right or left and/or either above or below the text. Further, you can optionally set a blur distance around the text.

## color

The optional **color** value specifies the color of the shadow effect. If a **color** value is not specified for the **text-shadow** property and if a color has been set by the **color** property for the font, then that color will be used. The **color** value can appear in order either before or after the **length length length** values.

## length length length

The first **length** value sets the horizontal distance to the right or left that the shadow effect will extend. A positive value extends to the right. A negative value extends to the left. A value of zero dictates no horizontal shadow effect.

The second **length** value sets the vertical distance above or below that the shadow effect will extend. A positive value extends above. A negative value extends below. A value of zero dictates no vertical shadow effect.

The optional third **length** value sets the radius of the distance that a blur effect will extend around the associated text. If you only want a blur effect, set both the first and second **length** to zero.

Each individual **length** value can be in any of the following seven units. Use the abbreviation.

Length	Abbrev.
centimeters	cm
ems	em
inches	in
millimeters	mm
picas	pc
pixels	px

points	pt
--------	----

Text shadows can also be used with the **:first-letter** and **:first-line** pseudo-elements.

### **:first-letter**

The **:first-letter** pseudo-element is used to apply the shadow effect to the first letter of a paragraph. It does this by the use of what is called a fictional tag sequence. In this code fragment, the fictional tag sequence is: `<p:first-letter> ... </p:first-letter>`

The closing tag is mandatory.

Code:

```
p:first-letter { font-size: 36px; color: red; text-shadow: red 0px 0px 5px;}
```

```
...
```

```
<p>
```

```
<p:first-letter>D</p:first-letter>evGuru is the best reference site on the Web.
```

```
<p>
```

### **:first-line**

The **:first-line** pseudo-element is used to apply the shadow effect to the first line of a paragraph. It requires the use of a fictional tag sequence (see above).

In this example, three shadow effects are specified. The list order dictates which shadows effects lay on top of other effects. The stacking order is:

- The first effect is on the bottom,
- the second effect lies on top of the first,
- and the third effect lies on top of the second.

Code:

```
p { text-shadow: 0px 0px 20px yellow, 0px 0px 10px orange, red 5px -5px; }
```

# PROPERTY: text-transform

---

**text-transform** : **capitalize** | **uppercase** | **lowercase** | **none**

**Compatibility:** IE4+ N4+

**Version:** Level 1

**Inherited:** Yes

The **text-transform** property allows you to control capitalization in a selected text.

## **capitalize**

The **capitalize** value will capitalize the first letter in each word in a selected text.

## **uppercase**

The **uppercase** value capitalizes all letters in a selected text.

## **lowercase**

The **lowercase** value makes all letters lower case (no capitals) in a selected text.

## **none**

The **none** value is the default. By declaring **none**, you can ensure that any previous declaration will not effect a selected text.

Code:

```
p { text-transform: none; }  
p { text-transform: capitalize; }  
p { text-transform: lowercase; }  
p { text-transform: uppercase; }
```

or

```
<span style="text-transform: none;">How now brown cow?</span>  
<span style="text-transform: capitalize;">How now brown cow?</span>  
<span style="text-transform: lowercase;">How now brown cow?</span>  
<span style="text-transform: uppercase;">How now brown cow?</span>
```

Output:

```
How now brown cow?  
How now brown cow?  
How now brown cow?  
How now brown cow?
```

# PROPERTY: top

---

**top** : auto | length | percentage

**Compatibility:** IE4+ N4+

**Version:** Level 2

**Inherited:** No

The **top** property sets the physical distance of how far the top content edge of an element is below the top content edge of the containing block.

A containing block is simply an element that contains one or more related elements.

There are three other properties that allow you to set the distance for the **bottom**, **left**, and **right** content edges. All four properties are used in conjunction with the **position** property. Note that if the **position** property is set to the **static** value, setting the **top** property has no effect.

## auto

The **auto** value dictates that the browser sets the distance between the top content edges.

## length

The **length** value sets the distance between the top content edges and can be in any of the following seven units. Use the abbreviation.

Length	Abbrev.
centimeters	cm
ems	em
inches	in
millimeters	mm
picas	pc
pixels	px
points	pt

## percentage

The **percentage** value sets the distance between the top content edges as a percentage of the overall height of the parent block. It is a positive integer number. The percent sign is mandatory.

Code:

```
p.one {  
position: absolute;  
bottom: 1in;  
left: 1in;  
right: 1in;
```

```
top: 1in;  
}
```

# PROPERTY: unicode-bidi

---

**unicode-bidi** : **bidirectional-override** | **embed** | **normal**

**Compatibility:** IE5+

**Version:** Level 2

**Inherited:** No

The **unicode-bidi** property is used when you need to display text that is read in different directions on the same Web page. It is used in conjunction with the **direction** property which sets the direction that the text flows. These two properties specify how the elements and attributes of a document language map to the bidirectional algorithm that controls text flow.

As an example, consider displaying a Hebrew text (which is read left-to-right) on a Web page that is otherwise in English (which reads right-to-left).

If you wish to apply the **direction** property to an inline-level text, you must set the **unicode-bidi** property either to the **bidirectional-override** or **embed** values.

## **bidirectional-override**

The **bidirectional-override** value allows the text direction to be set for an inline-level element or for a block-level element that contains only inline-level elements. The implicit portion of the bidirectional algorithm is ignore.

## **embed**

The **embed** value is used with inline-level elements to set levels of embedding in the bidirectional algorithm that controls text flow. You can set up to 15 embedded levels.

## **normal**

The **normal** value allows the implicit use of the bidirectional algorithm that controls text flow.

This example is best viewed in IE5.0 or IE5.5:

Code:

```
<blockquote style="direction: rtl; unicode-bidi: bidirectional-override;">  
ABC DEF GHI JKL MNO PQR STV UWX YZ  
</blockquote>
```

Output:

ABC DEF GHI JKL MNO PQR STV UWX YZ

# PROPERTY: vertical-align

---

**vertical-align** : **baseline** | **sub** | **super** | **top** | **text-top** | **middle** | **bottom** | **text-bottom** | **percentage** | **length**

**Compatibility:** IE4+ N4+

**Version:** Level 1/Expanded in Level 2

**Inherited:** No

The **vertical-align** property allows a wide variety of choices for aligning characters, letters, words, and text with regard to the baseline of a selected line of characters, letters, words, or text.

**baseline** Level 1

The **baseline** value is the default. It aligns the selected text to the baseline (bottom) of a line. By declaring **baseline**, you can ensure that any previous declaration will not effect a selected text.

**bottom** Level 1

The **bottom** value aligns selected characters, letters, words, or text with the lowest character on the same line.

**length** Level 2

The **length** values raises or lowers the element above or below the baseline by the specified amount. A positive value raises and a negative lowers. A value of zero is the baseline. You can use any of the following seven units. Use the abbreviation.

Length	Abbrev.
centimeters	cm
ems	em
inches	in
millimeters	mm
picas	pc
pixels	px
points	pt

**middle** Level 1

The **middle** value aligns selected characters, letters, words, or text with the mid-point (middle) of the same line.

**percentage** Level 1

The **percentage** value is based upon the line height. The bottom of the line (baseline) is designated 0% and the top is 100%. The **percentage** can be positive or negative and can

exceed 100% in magnitude. A negative **percentage** aligns below the baseline and a positive above.

### **sub** Level 1

The **sub** value places selected characters, letters, words, or text as a subscript.

### **super** Level 1

The **super** value places selected characters, letters, words, or text as a superscript.

### **text-top** Level 1

The **text-top** value aligns selected characters, letters, words, or text with the tallest character on a selected line.

### **text-bottom** Level 1

The **text-bottom** value aligns selected characters, letters, words, or text with the tallest character on a selected line.

### **top** Level 1

The **top** value aligns selected characters, letters, words, or text with the tallest character on the same line.

Code:

```
img { vertical-align: baseline; }
img { vertical-align: sub; }
img { vertical-align: super; }
img { vertical-align: top; }
img { vertical-align: text-top; }
img { vertical-align: middle; }
img { vertical-align: bottom; }
img { vertical-align: text-bottom; }
img { vertical-align: -50%; }
img { vertical-align: 125%; }
img { vertical-align: -5px; }
img { vertical-align: 10mm; }
```

or

```
img.tp { vertical-align: top; }
img.md { vertical-align: middle; }
img.bt { vertical-align: bottom; }
...
<b>The Guru</b> 
<p>
<b>The Guru</b> 
<p>
<b>The Guru</b> 
```

Output:



The Guru



The Guru



The Guru

# PROPERTY: visibility

---

**visibility** : collapse | hidden | visible

**Compatibility:** IE4+ N6

**Version:** Level 2

**Inherited:** No

The **visibility** property is used to create dynamic displays. It determines whether an element is visible or invisible. By invisible, we mean that the element is fully transparent, but that the element still occupies a place on the page and other content may not fill that space.

The **display** property can be used to make an element invisible and also specifies that nothing can take up space in the layout.

## collapse

The **collapse** value is primarily used to hide rows or columns in table elements. The hidden row or column space can be used for other content. For other types of elements, it has the same effect as the **hidden** value.

## hidden

The **hidden** value specifies that the element is not visible on the page.

## visible

The **visible** value specifies that the element is visible on the page.

Note that only the first and third **Guru** images are visible, and that the space where the second image should be, while reserved for the image, is empty.

Code:

```
p {visibility: hidden;}  
br {visibility: visible;}
```

or

```
  
  

```

Output:



# PROPERTY: white-space

---

**white-space** : normal | pre | nowrap

**Compatibility:** IE5.5 N4+

**Version:** Level 1

**Inherited:** Yes

The **white-space** property determines the use of white space (blank space) inside an element.

## normal

The **normal** value is the default.

## pre

The **pre** value behaves like the HTML tag **<pre>**.

## nowrap

The **nowrap** value allows a text to continue on the same line until the end of the text or until an HTML tag **<br>** is encountered. The **<br>** tag causes the text to wrap down to the next line.

Code:

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

or

```
<p style="white-space: pre;"> How now purple cow? </p>
```

Output:

```
How now purple cow?
```

# PROPERTY: width

---

**width** : **length** | **percentage** | **auto**

**Version:** Level 1

**Inherited:** No

The **width** property allows you to set the width of an element of text or an image. For example, you could embed an image into a specified position within a text using the float property. Then use the **width** and/or **height** properties to specify an exact size for the image. If you set the **width** of an element to a specified value and set the **height** property to **auto**, the element will be scaled proportionally (i.e., you maintain the aspect ratio).

## length

The **length** value can be in any of the following seven dimensions. Use the abbreviation.

Length	Abbrev.
centimeters	cm
ems	em
inches	in
millimeters	mm
picas	pc
pixels	px
points	pt

## percentage

The **percentage** value is based upon the total width being defined as 100%.

## auto

The **auto** value directs the browser to automatically calculate the width. If both the **width** and **height** are set to **auto**, the dimensions of the text or image element remains unchanged. By declaring **auto**, you can ensure that any previous declaration will not effect the width of the selected element.

Code:

```
img { width: 150mm; }  
img { width: 2.85in; }  
img { width: 45pc; }  
img { width: 36pt; }  
img { width: 140%; }  
img { width: auto; }  
img {  
width: 3.0in
```

```
height: auto  
}
```

or

Code:

```
  
<br>  

```

Output:



# PROPERTY: word-spacing

---

**word-spacing** : normal | length

**Compatibility:** IE6 N6

**Version:** Level 1

**Inherited:** Yes

The **word-spacing** property increases or decreases the width of the blank (white) space between adjacent words. A positive value increases the width. A negative value decreases the width.

## normal

The **normal** value is the default. By declaring **normal**, you can ensure that any previous declaration will not effect the selected text.

## length

The **length** value can be in any of the following seven units. Use the abbreviation.

Length	Abbrev.
centimeters	cm
ems	em
inches	in
millimeters	mm
picas	pc
pixels	px
points	pt

Code:

```
p { word-spacing: normal; }  
p { word-spacing: 1mm; }  
p { word-spacing: 2.5em; }
```

or

```
<p style="word-spacing: 50px;"> How now purple cow? </p>
```

Output:

How now purple cow?



# PROPERTY: z-index

---

**z-index** : **auto** | **number**

**Compatibility:** IE4+ N6

**Version:** Level 2

**Inherited:** No

The **z-index** property sets the stacking order for a group of elements whose x/y coordinates overlap the same area.

Elements can have 3-dimensions. The x- and y-components set the element position as viewed on the monitor. The z-component determines which elements can appear to lie on top of other elements, or conversely, which elements appear to lie under other elements.

The higher (or more positive) the number, the higher the element is in the stacking order. Higher numbered elements overlay lower numbered elements (2 overrides 1).

## **auto**

The **auto** value sets the stacking order number to the value of the parent element.

## **number**

The **number** value can be zero, a positive integer, or a negative integer. This sets the local stacking order. An element with a higher number will be in front of all elements with lower numbers.

In this example, the blue colored word Dev has a higher z-index and is superimposed on top of the red colored word Guru.

Code:

```
<html>
<head>
<title>z-index example</title>
<style type="text/css">
.logo {
position: absolute;
left: 0.5in;
top: 0.5in;
}
</style>
</head>
<body>

<div id="word1" class="logo" style="z-index: 1">
<br>
<span style="color: red; font-size: 100px;">Guru</span>
</div>

<div id="word2" class="logo" style="z-index: 2">
<span style="color: blue; font-size: 70px;">Dev</span>
</div>

</body>
```

</html>

# PSEUDO-CLASSES: active

---

## active

**Compatibility:** IE4+ N6+

**Version:** Level 1

The **active** pseudo-class is used to apply a style to a link element when that link is selected either by coming into focus or being clicked upon. Under those circumstances, the link is said to be active.

All together, there are four pseudo-classes that are reserved for use with the HTML anchor tag (**a**). They are used to assign any appropriate CSS property:value pair to a link. Most typically, they are used to designate a font color, or background color, or to remove the underline.

Please refer to the HTML Quick Reference for more information about the [anchor tag](#).

The three other pseudo-classes used with the **a** tag are:

## hover

The **hover** pseudo-class effects the style when the mouse is hovering over the link (mouse over).

## link

The **link** pseudo-class effects a style for a link that is not active and has not been visited.

## visited

The **visited** pseudo-class effects the style for a link after the link has been visited.

A pseudo-class is assigned to a selector via following syntax:

```
selector:pseudo-class { property: value; ...; }
```

A pseudo-class can also be used with the value assigned as a class to a selector:

```
selector.class:pseudo-class { property: value; ...; }
```

Code:

```
a:active { background-color: RGB(100, 14, 107); text-decoration: none; }
a:hover { color: red; text-decoration: none; }
a:link { color: teal; text-decoration: none; }
a:visited { visibility: hidden; }
```

or

```
a.example:active {
color: yellow; font-size: 50px; background-color: red; text-decoration: none;
}
```

```
<a class="example" href="http://www.devguru.com" target="_blank">DevGuru</a>
```

# AT-RULES: @page

## PSEUDO-CLASSES: :first :left :right

---

**@page** **page-selector** **pseudo-class** { **style-rules** };

**Compatibility:** IE5+ N6+

**Version:** Level 2

The **@page** at-rule is used to set a collection of style rules which define page context on a document level.

The **page-selector** component is a string that is used to provide a unique name for this set of style rules. This name provides a simple means of uniquely referencing these rules. For example, if you were creating a birthday card, you could use birthday-card as the **page-selector**.

The **pseudo-class** is one of three pseudo-classes that are exclusively used by the **@page** at-rule. They are:

The **:first** pseudo-class refers to the first page of a document. You may wish to apply different style rules to the first page, as compared to the rest of the pages in a document.

The **:left** pseudo-class refers to the left page of a document (i.e., think of an open book which will have a left and a right page).

The **:right** pseudo-class refers to the right page of a document (i.e., think of an open book which will have a left and a right page).

The **style-rules** is a set of one or more style rules that you wish to apply to the document. The entire set is enclosed by a pair of left and right curly braces { }. In addition to the regular CSS2 style rules, there are style rules which are exclusive to the **@page** at-rule. They include:

**marks, orphans, page, page-break-after, page-break-before, page-break-inside, size, and windows.**

A full explanation of these exclusive style rules would be extremely lengthy and is beyond the scope of this Quick Reference. As a starting point, the **Guru** recommends that you refer to Chapter 13 of the W3C standard on [Paged Media](#).

The syntax for an at-rule is an @ symbol, followed immediately by an unique identifier, which in turn is followed by the block which contains content that is applied on a document level. The block syntax is either:

- all content contained between a pair of curly brackets,
- or is all content contained between the unique identifier and a semi-colon.

There are five at-rules: **@charset**, **@font-face**, **@import**, **@media**, and **@page**.

Code:

```
@page birthday-card:right { margin-bottom: 3in; margin-left: 2in; margin-right: 2in; margin-top: 1in; }
```

# PSEUDO-CLASSES: first-child

---

## first-child

**Compatibility:** Currently not supported by any browser

**Version:** Level 1

The **first-child** pseudo-class is used to apply style to the first element, called the first-child element, that is contained inside a specific parent element.

A pseudo-class is assigned to a selector via following syntax:

```
selector:pseudo-class { property: value; ...; }
```

A pseudo-class can also be used with the value assigned as a class to a selector:

```
selector.class:pseudo-class { property: value; ...; }
```

In this example, the first-child element are the pair of opening and closing **bold** tags that enclose **DevGuru**.

Code:

```
<html>
<head>
<title>first-child test</title>
<style type="text/css">
p:first-child {color: red;}
</style>
</head>
<body>
<p>
I love <b>DevGuru</b> more than I love <i>ice cream</i>!
</p>
</body>
</html>
```

Simulated output:

Output:

I love **DevGuru** more than I love ice cream!

# PSEUDO-CLASSES: focus

---

## focus

**Compatibility:** N6+

**Version:** Level 2

The **focus** pseudo-class is used to effect style when an HTML anchor or form element come into focus. As an example, you could highlight each form element with a background color as it came into focus. On a complicated form, this highlight draws the user's eye to the element.

By focus, we mean that the cursor is at that element. For example, if the element was an input text box in a form, the cursor would appear inside the element and you could type and enter data into the text box.

There are several ways that an element can come into focus. Perhaps the most common way, is that the user simply clicks the mouse onto the element. A more complex example occurs when a JavaScript function is used to check a submitted form, and then assigns focus to a specific element contained in the form that is not filled out properly. This concept is exemplified in detail in the [Client-side Form Verification in JavaScript](#) Knowledge Base article.

A pseudo-class is assigned to a selector via following syntax:

```
selector:pseudo-class { property: value; ...; }
```

A pseudo-class can also be used with the value assigned as a class to a selector:

```
selector.class:pseudo-class { property: value; ...; }
```

The **focus** pseudo-class is poorly implemented.

The **focus** pseudo-class can also be combined with other pseudo-classes, such as **hover**. Here is a code example (however this feature is very poorly implemented):

```
a.example:focus:hover {color: red; background-color: yellow; text-decoration: none;}
```

```
Code:
<html>
<head>
<title>ocus test</title>
<style type="text/css">
a.example:focus {color: red; background-color: yellow; text-decoration: none;}
</style>
</head>
<body>
<a href="http://www.devguru.com">
DevGuru</a>
</body>
</html>
```

# PSEUDO-CLASSES: hover

---

## hover

**Compatibility:** IE4+ N6+

**Version:** Level 2

The **hover** pseudo-class is used to apply a style to a link element when the cursor passes over the link (i.e., a mouse-over). Under those circumstances, the cursor is said to hover over the link.

All together, there are four pseudo-classes that are reserved for use with the HTML anchor tag (**a**). They are used to assign any appropriate CSS property:value pair to a link. Most typically, they are used to designate a font color, or background color, or to remove the under lining.

The three other pseudo-classes used with the **a** tag are:

## active

The **active** pseudo-class effects a style when the link is selected (clicked on).

## link

The **link** pseudo-class effects a style for a link that is not active and has not been visited.

## visited

The **visited** pseudo-class effects the style for a link after the link has been visited.

A pseudo-class is assigned to a selector via following syntax:

```
selector:pseudo-class { property: value; ...; }
```

A pseudo-class can also be used with the value assigned as a class to a selector:

```
selector.class:pseudo-class { property: value; ...; }
```

Code:

```
a:active { background-color: RGB(100, 14, 107); text-decoration: none; }
a:hover { color: red; text-decoration: none; }
a:link { color: teal; text-decoration: none; }
a:visited { visibility: hidden; }
```

or

```
a.example:hover {
color: yellow; font-size: 50px; background-color: red; text-decoration: none;
}
```

```
<a class="example" href="http://www.devguru.com" target="_blank">DevGuru</a>
```

# PSEUDO-CLASSES: lang

---

## **lang**(language code)

**Compatibility:** Currently not supported by any browser

**Version:** Level 2

The **lang** pseudo-class is used to apply style to the contents of an element which is written in a foreign language. You must declare the language code by enclosing the language code within a pair of parentheses.

The language code is set by the [ISO 639 and RFC 1776 standards](#).

You can select the language for the element or document by:

- setting the language in the HTTP header,
- or by using the **lang** attribute that is available in [HTML](#) and [XHTML](#),
- or by using a **meta** element that is available in [HTML](#) and [XHTML](#).

A pseudo-class is assigned to a selector via following syntax:

```
selector:pseudo-class(language code) { property: value; ...; }
```

A pseudo-class can also be used with the value assigned as a class to a selector:

```
selector.class:pseudo-class(language code) { property: value; ...; }
```

Code:

```
<html>
<head>
<title>lang test</title>
<style type="text/css">
p:lang(fr) {color: red;} </style>
</head>
<body>
<p lang="fr">
Bonjour le monde!
</p>
</body>
</html>
```

Simulated output.

Output:

**Bonjour le monde!**

# PSEUDO-CLASSES: link

---

## link

**Compatibility:** IE4+ N6+

**Version:** Level 1

The **link** pseudo-class is used to apply style to a link that is unvisited. In other words, the link has not been clicked upon or visited by the user. (Note: the fact that a link has been visited is stored in the browser history. If the history is plurged, the fact that the link has been visited may be lost and the link will be treated as unvisited.)

All together, there are four pseudo-classes that are reserved for use with the HTML anchor tag (**a**). They are used to assign any appropriate CSS property:value pair to a link. Most typically, they are used to designate a font color, or background color, or to remove the underline.

The three other pseudo-classes used with the **a** tag are:

## active

The **active** pseudo-class effects a style when the link is selected (clicked on).

## hover

The **hover** pseudo-class effects the style when the mouse is hovering over the link (mouse over).

## visited

The **visited** pseudo-class effects the style for a link after the link has been visited.

A pseudo-class is assigned to a selector via following syntax:

```
selector:pseudo-class { property: value; ...; }
```

A pseudo-class can also be used with the value assigned as a class to a selector:

```
selector.class:pseudo-class { property: value; ...; }
```

Code:

```
a:active { background-color: RGB(100, 14, 107); text-decoration: none; }
a:hover { color: red; text-decoration: none; }
a:link { color: teal; text-decoration: none; }
a:visited { visibility: hidden; }
```

or

```
a.example:link {
color: blue; background-color: red; text-decoration: none;
}
a.example:visited {color: yellow; text-decoration: none;}
```

```
<a class="example" href="http://www.facade.com/biorhythm" target="_blank">Biorythem</a>
```

Output:

Biorythem



# PSEUDO-CLASSES: visited

---

## visited

**Compatibility:** IE4+ N6+

**Version:** Level 1

The **visited** pseudo-class is used to apply style to a link element after that link has been visited by the user. Commonly, a visited link will be assigned a different color than an unvisited link. This makes it very easy for a user to determine which links have or have not been visited. (Note: the fact that a link has been visited is stored in the browser history. If the history is purged, the fact that the link has been visited may be lost and the link will be treated as unvisited.)

All together, there are four pseudo-classes that are reserved for use with the HTML anchor tag (**a**). They are used to assign any appropriate CSS property:value pair to a link. Most typically, they are used to designate a font color, or background color, or to remove the underline.

The three other pseudo-classes used with the **a** tag are:

## active

The **active** pseudo-class effects a style when the link is selected (clicked on).

## hover

The **hover** pseudo-class effects the style when the mouse is hovering over the link (mouse over).

## link

The **link** pseudo-class effects a style for a link that is not active and has not been visited.

A pseudo-class is assigned to a selector via following syntax:

```
selector:pseudo-class { property: value; ...; }
```

A pseudo-class can also be used with the value assigned as a class to a selector:

```
selector.class:pseudo-class { property: value; ...; }
```

Code:

```
a:active { background-color: RGB(100, 14, 107); text-decoration: none; }  
a:hover { color: red; text-decoration: none; }  
a:link { color: teal; text-decoration: none; }  
a:visited { visibility: hidden; }
```

or

```
a.example:visited {color: yellow; text-decoration: none;}
```

```
<a class="example" href="http://www.facade.com/biorhythm/" target="_blank">Biorythem</a>
```

# PSEUDO-ELEMENTS: after

---

**Compatibility:** N6+

**Version:** Level 2

The **:after** pseudo-element is used to insert content after an element. This is done via the [content](#) property. The content assigned by the **content** property can be characters, a string, text, or an image. Further, you can apply style to the content, such as setting font and color. This is a convenient way to suffix the same text to large number of related text elements which have the same **class** value.

The similar **:after** pseudo-element is used to insert content immediately after an element.

CSS2 has four pseudo-elements: **:after**, **:before**, **:first-letter**, and **:first-line**. Pseudo-elements allow you to create element-like structures which permit you to apply style to parts of a document that normally cannot be accessed using HTML. Specifically, you can add styled content before and after an element, or effect the style of the first letter or first line of an element.

Note that while the **content** property does not inherit, the **:before** and **:after** psuedo-elements can inherit any inheritable styles that are in effect, or you can specify various CSS properties to effect the appearance of the content.

A pseudo-element is assigned to a selector via following syntax:

```
selector:pseudo-element {property: value; ...;}
```

A pseudo-element can also be used with the value assigned as a class to a selector:

```
selector.classvalue:pseudo-element {property: value; ...;}
```

Code:

```
<html>
<head>
<title>after and before test</title>
<style type="text/css">
p.red:before {content: "FOR SALE!"; color: red;}
p.red:after {content: "No implied warrenty"; color: blue;}
</style>
</head>
<body>
<p class="red">
1954 Ford Maroon 2-door Sedan, as is, $399.00
</p>
<p class="red">
1955 Ford Black 4-door Sedan, rusted, $299.00
</p>
</body>
</html>
```

Simulated output, however, the above code will work on Netscape 6+.

Output:

**FOR SALE!** 1954 Ford Maroon 2-door Sedan, as is, \$399.00 **No implied warrenty**

**FOR SALE!** 1955 Ford Black 4-door Sedan, rusted, \$299.00 [No implied warrenty](#)

# PSEUDO-ELEMENTS: before

---

**Compatibility:** N6+

**Version:** Level 2

The **:before** pseudo-element is used to insert content immediately before an element. This is done via the [content](#) property. The content assigned by the **content** property can be characters, a string, text, or an image. Further, you can apply style to the content, such as setting font and color. This is a convenient way to prefix the same text to large number of related text elements which have the same **class** value.

The similar **:after** pseudo-element is used to insert content immediately after an element.

CSS2 has four pseudo-elements: **:after**, **:before**, **:first-letter**, and **:first-line**. Pseudo-elements allow you to create element-like structures which permit you to apply style to parts of a document that normally cannot be accessed using HTML. Specifically, you can add styled content before and after an element, or effect the style of the first letter or first line of an element.

Note that while the **content** property does not inherit, the **:before** and **:after** psuedo-elements can inherit any inheritable styles that are in effect, or you can specify various CSS properties to effect the appearance of the content.

A pseudo-element is assigned to a selector via following syntax:

```
selector:pseudo-element {property: value; ...;}
```

A pseudo-element can also be used with the value assigned as a class to a selector:

```
selector.classvalue:pseudo-element {property: value; ...;}
```

Code:

```
<html>
<head>
<title>after and before test</title>
<style type="text/css">
p.red:before {content: "FOR SALE!"; color: red;}
p.red:after {content: "No implied warrenty"; color: blue;}
</style>
</head>
<body>
<p class="red">
1954 Ford Maroon 2-door Sedan, as is, $399.00
</p>
<p class="red">
1955 Ford Black 4-door Sedan, rusted, $299.00
</p>
</body>
</html>
```

Simulated output, however, the above code will work on Netscape 6+.

Output:

**FOR SALE!** 1954 Ford Maroon 2-door Sedan, as is, \$399.00 **No implied warrenty**

**FOR SALE!** 1955 Ford Black 4-door Sedan, rusted, \$299.00 [No implied warrenty](#)

# PSEUDO-ELEMENT: first-letter

---

**Compatibility:** IE5+ N6+

**Version:** Level 1

The **:first-letter** pseudo-element allows you to apply style to the first letter of an element. This is a very convenient way to enhance the appearance of the beginning of paragraphs throughout a document.

CSS2 has four pseudo-elements: **:after**, **:before**, **:first-letter**, and **:first-line**. Pseudo-elements allow you to create element-like structures which permit you to apply style to parts of a document that normally cannot be accessed using HTML. Specifically, you can add styled content before and after an element, or effect the style of the first letter or first line of an element.

Only certain CSS properties can be applied using **:first-letter**.

They are:

<b>background</b>	<b>background-color</b>	<b>background-image</b>
<b>background-repeat</b>	<b>clear</b>	<b>color</b>
<b>float</b>	<b>font</b>	<b>font-family</b>
<b>font-size</b>	<b>font-style</b>	<b>font-variant</b>
<b>font-weight</b>	<b>letter-spacing</b>	<b>line-height</b>
<b>margin</b>	<b>margin-bottom</b>	<b>margin-left</b>
<b>margin-right</b>	<b>margin-top</b>	<b>padding</b>
<b>padding-bottom</b>	<b>padding-left</b>	<b>padding-right</b>
<b>padding-top</b>	<b>text-decoration</b>	<b>text-shadow</b>
<b>text-transform</b>	<b>vertical-align</b>	<b>word-spacing</b>

A pseudo-element is assigned to a selector via following syntax:

```
selector:pseudo-element {property: value; ...;}
```

A pseudo-element can also be used with the value assigned as a class to a selector:

```
selector.classvalue:pseudo-element {property: value; ...;}
```

Code:

```
<html>
<head>
<title>first-letter test</title>
<style type="text/css">
h1.red:first-letter {color: #FF0000;}
h1.yellow:first-letter {color: yellow;}
</style>
```

```
</head>
<body>
<h1 class="red"> The first letter is red.</h1>
<h1 class="yellow"> The first letter is yellow.</h1>
<h1> The first letter is black.</h1>
</body>
</html>
```

Output:

**The first letter is red.**

**The first letter is yellow.**

**The first letter is black.**

# PSEUDO-ELEMENT: first-line

---

**Compatibility:** IE5+ N6+

**Version:** Level 1

The **:first-line** pseudo-element allows you to apply style to the first line of an element. This is a very convenient way to enhance the appearance of the beginning of paragraphs throughout a document.

CSS2 has four pseudo-elements: **:after**, **:before**, **:first-letter**, and **:first-line**. Pseudo-elements allow you to create element-like structures which permit you to apply style to parts of a document that normally cannot be accessed using HTML. Specifically, you can add styled content before and after an element, or effect the style of the first letter or first line of an element.

Only certain CSS properties can be applied using "**first-line**."

They are:

<b>background</b>	<b>background-attachment</b>	<b>background-color</b>
<b>background-image</b>	<b>background-repeat</b>	<b>clear</b>
<b>color</b>	<b>font</b>	<b>font-family</b>
<b>font-size</b>	<b>font-style</b>	<b>font-variant</b>
<b>font-weight</b>	<b>letter-spacing</b>	<b>line-height</b>
<b>text-decoration</b>	<b>text-shadow</b>	<b>text-transform</b>
<b>vertical-align</b>	<b>word-spacing</b>	

A pseudo-element is assigned to a selector via following syntax:

```
selector:pseudo-element {property: value; ...;}
```

A pseudo-element can also be used with the value assigned as a class to a selector:

```
selector.classvalue:pseudo-element {property: value; ...;}
```

Code:

```
<html>
<head>
<title>first-line test</title>
<style type="text/css">
p.red:first-line {color: #ff0000; font-weight: bold;}
</style>
</head>
<body>
<p class="red">
```

Only the first line is red and in bold.

```
<br>
```

The second line remains the default black.

```
<br>
```

Ditto for the third line and so on...

```
</p>  
<p class="red">  
When there is a sentence that is so long that it continues beyond the first line, then only the first  
line is affected by the style. No additional lines are affected.  
</p>  
</body>  
</html>
```

Output:

Only the first line is red and in bold.  
The second line remains the default black.  
Ditto for the third line and so on...

When there is a sentence that is so long that it continues beyond the first line, then only the first  
line is affected by the style. No additional lines are affected.

# AT-RULES: @charset

---

**@charset "charset";**

**Compatibility:** IE5+ N6+

**Version:** Level 2

The **@charset** at-rule allows you to designate a character set be used with an HTML document.

Note that the **@charset** at-rule can **ONLY** appear in an external style sheet file. It cannot appear inside a **style** element embedded inside an HTML document.

The W3C officially defines the character sets at:

<http://www.iana.org/assignments/character-sets>.

The syntax for an at-rule is an @ symbol, followed immediately by an unique identifier, which in turn is followed by the block which contains content that is applied on a document level. The block syntax is either:

- all content contained between a pair of curly brackets,
- or is all content contained between the unique identifier and a semi-colon.

There are five at-rules: **@charset**, **@font-face**, **@import**, **@media**, and **@page**.

The mandatory **"charset"** component is the name of the character set enclosed between a pair of double quotes.

For example, the character set for Arabic is designsted by:

Code:

```
@charset "ISO-8859-6"
```

# AT-RULES: @font-face

---

```
@font-face { descriptor: value; ...; };
```

**Compatibility:** IE5+ N6+

**Version:** Level 2

The **@font-face** at-rule is used to comprehensively describe the font-face used in a document.

A list of one or more **descriptor: value;** components delineated (separated) by blank space are used to specify the font. You may choose from among a rather large number of such **descriptor: value;** components. This permits you to describe the font in detail.

A full explanation of the **@font-face** at-rule would be extremely lengthy and is beyond the scope of this Quick Reference. As a starting point, the **Guru** recommends that you refer to Chapter 15 of the [W3C standard](#).

The syntax for an at-rule is an **@** symbol, followed immediately by an unique identifier, which in turn is followed by the block which contains content that is applied on a document level. The block syntax is either:

- all content contained between a pair of curly brackets,
- or is all content contained between the unique identifier and a semi-colon.

There are five at-rules: **@charset**, **@font-face**, **@import**, **@media**, and **@page**.

Code:

```
@font-face { font-family: swiss721 };
```

# AT-RULES: @import

---

**@import** url(address) media-type, ... ;

**Compatibility:** IE5+ N6+

**Version:** Level 2

The **@import** at-rule allows you to designate an external style sheet file to be used with an HTML document.

The syntax for an at-rule is an @ symbol, followed immediately by an unique identifier, which in turn is followed by the block which contains content that is applied on a document level. The block syntax is either:

- all content contained between a pair of curly brackets,
- or is all content contained between the unique identifier and a semi-colon.

There are five at-rules: **@charset**, **@font-face**, **@import**, **@media**, and **@page**.

The mandatory **url(address)** component is the url address of the external style sheet file. The opening and closing parentheses are required.

The optional **media-type** component is a list of zero or more media types which are delimited (separated) by commas. Some of the permitted values are:

- **all** - the default.
- **aural** - speech synthesizers.
- **braille** - braille tactile feedback devices.
- **embossed** - paged braille printers.
- **handheld** - handheld devices.
- **print** - printed paged and print preview.
- **projection** - projectors or print to transparencies.
- **screen** - color computer screens.
- **tty** - teletypes.
- **tv** - television-type devices.

This file has the same CSS style rules as the **DevGuru** site.

Code:

```
<html>
<head>
<title>@media test</title>
<style type="text/css">
@import url(http://www.devguru.com/include/STYLERULES.css);
</style>
</head>
<body>
<p>
I love <a href="http://www.devguru.com">DevGuru</a>!
</p>
</body>
</html>
```



# AT-RULES: @media

---

**@media** **media-type**, ... { **selector { property: value; ...; } ...** };

**Compatibility:** IE5+ N6+

**Version:** Level 2

The **@media** at-rule is used to set the target media for a collection of style rules on a document level.

The optional **media-type** component is a list of zero or more media types which are delineated (separated) by commas. If left blank, the default is all media types. Some of the permitted values are:

- **all** - the default.
- **aural** - speech synthesizers.
- **braille** - braille tactile feedback devices.
- **embossed** - paged braille printers.
- **handheld** - handheld devices.
- **print** - printed paged and print preview.
- **projection** - projectors or print to transparencies.
- **screen** - color computer screens.
- **tty** - teletypes.
- **tv** - television-type devices.

The mandatory **selector { property: value; ...; }** component is a list of one or more style rules. Each style rule is a property/value pair. You can have a unlimited number of selector components. They are delineated (separated) by blank space.

The syntax for an at-rule is an **@** symbol, followed immediately by an unique identifier, which in turn is followed by the block which contains content that is applied on a document level. The block syntax is either:

- all content contained between a pair of curly brackets,
- or is all content contained between the unique identifier and a semi-colon.

There are five at-rules: **@charset**, **@font-face**, **@import**, **@media**, and **@page**.

Code:

```
@media print {  
  p { color: black; font-face: arial; }  
  a:visited { text-decoration: underline; font-weight: bold;}  
}
```