

CSS - Cascading Style Sheets

CSS is used by both the authors and readers of web pages to define colors, fonts, layout, and other aspects of document presentation. It is designed primarily to enable the separation of document content (written in HTML or a similar markup language) from document presentation (written in CSS). This separation can improve content accessibility, provide more flexibility and control in the specification of presentational characteristics, and reduce complexity and repetition in the structural content.

How to use this CSS Properties Table:

You need to match the appropriate property with the appropriate value. Separate the property and value with a colon : and put a semi-colon ; after the value. The : separates the property from the value and the ; separates all property:value pairs from each other. For example, the first property in this table is *letter-spacing*. To pair it to the *normal* value you would write **letter-spacing:normal;**

1) Using the Style Attribute or In-Line Styles :

The first way to add a style to your page is to simply declare it inside an HTML tag. The way to do this is to add the `style=""` attribute to an HTML tag. The general form to add one style property looks like this:

```
style="property:value;"
```

To help clarify this, let's look at an example. If you want the color of some text to look red AND the text size to be 20 points, the style attribute would look like this:

```
<div style="color:red; font-size: 20pt;">
```

2) Styles in the Head Section:

A style in the HEAD section must begin with the `<style>` tag and end with the `</style>` tag. You also need a comment between the property:value;.

```
<style>
<!--
p { color:red; font-style:italic; }
-->
</style>
```

The squiggly brackets {} hold the property:value; combinations of the selector (which is the P tag). Now all text in a paragraph will be red and italic.

3) Using External Style Sheets:

Open your text editor and create a new blank document. Type in just CSS like the following information:

```
p { color:red; font-style:italic; }
.redfont { color:red; }
```

Now, save the file as "style.css" (You can use any name you wish, just be sure you have the .css extension at the end). This creates an external style sheet.

To link to a style sheet, you need to add the `<link>` tag inside the head section of your HTML document. Here is the general form:

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

Now all CSS information will be applied to the HTML document even though the CSS is contained in another external file.

CSS Syntax

The CSS syntax is made up of three parts: a selector, a property and a value:

```
selector {property: value}
```

The selector is normally the HTML element/tag you wish to define, the property is the attribute you wish to change, and each property can take a value. The property and value are separated by a colon (and called a declaration), and surrounded by curly braces:

```
body {color: black}
```

Note: If the value is multiple words, put quotes around the value:

```
p {font-family: "sans serif"}
```

Note: If you wish to specify more than one property, you must separate each property with a semicolon. The example below shows how to define a center aligned paragraph, with a red text color:

```
p {text-align: center; color: red}
```

To make the style definitions more readable, you can (should :-)) describe one property on each line, like this:

```
p {
  text-align: center;
  color: black;
  font-family: arial;
}
```

Grouping

```
h1,h2,h3,h4,h5,h6 {
  color: green }
```

The class Selector

With the class selector you can define different styles for the same type of HTML element.

Say that you would like to have two types of paragraphs in your document: one right-aligned paragraph, and one center-aligned paragraph. Here is how you can do it with styles:

```
p.right {text-align: right}
p.center {text-align: center}
```

You have to use the class attribute in your HTML document:

```
<p class="right">This paragraph will be right-aligned.</p>
<p class="center">This paragraph will be center-aligned.</p>
```

Note: To apply more than one class per given element, the syntax is:

```
<p class="center bold">This is a paragraph.</p>
```

The paragraph above will be styled by the class "center" AND the class "bold".

You can also omit the tag name in the selector to define a style that will be used by all HTML elements that have a certain class. In the example below, all HTML elements with class="center" will be center-aligned:

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

In the code below both the h1 element and the p element have class="center". This means that both elements will follow the rules in the ".center" selector:

```
<h1 class="center">This heading will be center-aligned</h1>
<p class="center">This paragraph will also be center-aligned.</p>
```

Note: Do NOT start a class name with a number! It will not work in Mozilla/Firefox.

Add Styles to Elements with Particular Attributes

You can also apply styles to HTML elements with particular attributes.

The style rule below will match all input elements that have a type attribute with a value of "text":

```
input[type="text"] {background-color: blue}
```

The id Selector

You can also define styles for HTML elements with the id selector. The id selector is defined as a #.

The style rule below will match the element that has an id attribute with a value of "green":

```
#green {color: green}
```

The style rule below will match the p element that has an id with a value of "para1":

```
p#para1 {
  text-align: center;
  color: red
}
```

Note: Do NOT start an ID name with a number! It will not work in Mozilla/Firefox.

CSS Comments

Comments are used to explain your code, and may help you when you edit the source code at a later date. A comment will be ignored by browsers. A CSS comment begins with "/*", and ends with "*/", like this:

```
/* This is a comment */
p {
  text-align: center;      /* This is another comment */
  color: black;
}
```

CSS 2.0 Background Properties

Property	Description	Possible Values	Examples
background-color	Declares the background color.	Valid color names, RGB values, hexadecimal notation.	div { background-color:green; } div { color:#00FF00; }
background-image	Declares the background image of an element.	URL values.	div { background-image:url(images/img.jpg); } body { background-image:url(img.jpg); }
background-repeat	Declares how and/or if a background image repeats.	repeat repeat-x repeat-y no-repeat	div { background-repeat:repeat-x; } div { background-repeat:no-repeat; }
background-attachment	Declares the attachment of a background image (to scroll with the page content or be in a fixed position).	fixed scroll	div { background-attachment:fixed; } div { background-attachment:scroll; }
background-position	Declares the position of a background image.	Lengths or percentages for the x and y positions, or one of the predefined values: top left, top center, top right, center left, center center, center right, bottom left, bottom center, bottom right	div { background-position:10px 50px; } div { background-position:bottom right; }
background	Used as a shorthand property to set all the background properties at once.	Separate values by a space in the following order (those that are not defined will use inherited or default initial values): background-color background-image background-repeat background-attachment background-position	div { background:green url(image.jpg) no-repeat fixed center center; } div { background:url(image.jpg) fixed; }

CSS 2.0 Border Properties

Property	Description	Possible Values	Examples
border-top-color	Declares the color of the top border.	Valid color names, RGB values, hexadecimal notation, or the predefined value transparent.	div { border-top-color:green; } div { border-top-color:#00FF00; }
border-top-style	Declares the style of the top border.	none, hidden, dotted, dashed, solid, double, groove, ridge, inset, outset	div { border-top-style:solid; } div { border-top-style:inset; }
border-top-width	Declares the width of the top border.	Lengths or the following predefined values: thin medium thick	div { border-top-width:2px; } div { border-top-width:thin; }
border-top	Used as a shorthand property to set all the border-top properties at once.	Separate values by a space in the following order (those that are not defined will use inherited or default initial values): border-top-width border-top-style border-top-color	div { border-top:2px solid green; } div { border-top:thick double #00FF00; }
border-right-color	Declares the color of the right border.	Valid color names, RGB values, hexadecimal notation, or the predefined value transparent.	div { border-right-color:green; } div { border-right-color:#00FF00; }
border-right-style	Declares the style of the right border.	none, hidden, dotted, dashed, solid, double, groove, ridge, inset, outset	div { border-right-style:solid; } div { border-right-style:inset; }

border-right-width	Declares the width of the right border.	Lengths or the following predefined values: thin medium thick	div { border-right-width:2px; } div { border-right-width:thin; }
border-right	Used as a shorthand property to set all the border-right properties at once.	Separate values by a space in the following order (those that are not defined will use inherited or default initial values): border-right-width border-right-style border-right-color	div { border-right:2px solid green; } div { border-right:thick double #00FF00; }
border-bottom-color	Declares the color of the bottom border.	Valid color names, RGB values, hexadecimal notation, or the predefined value transparent.	div { border-bottom-color:green; } div { border-bottom-color:#00FF00; }
border-bottom-style	Declares the style of the bottom border.	none, hidden, dotted, dashed, solid, double, groove, ridge, inset, outset	div { border-bottom-style:solid; } div { border-bottom-style:inset; }
border-bottom-width	Declares the width of the bottom border.	Lengths or the following predefined values: thin medium thick	div { border-bottom-width:2px; } div { border-bottom-width:thin; }
border-bottom	Used as a shorthand property to set all the border-bottom properties at once.	Separate values by a space in the following order (those that are not defined will use inherited or default initial values): border-bottom-width border-bottom-style border-bottom-color	div { border-bottom:2px solid green; } div { border-bottom:thick double #00FF00; }
border-left-color	Declares the color of the left border.	Valid color names, RGB values, hexadecimal notation, or the predefined value transparent.	div { border-left-color:green; } div { border-left-color:#00FF00; }
border-left-style	Declares the style of the left border.	none, hidden, dotted, dashed, solid, double, groove, ridge, inset, outset	div { border-left-style:solid; } div { border-left-style:inset; }
border-left-width	Declares the width of the left border.	Lengths or the following predefined values: thin, medium, thick	div { border-left-width:2px; } div { border-left-width:thin; }
border-left	Used as a shorthand property to set all the border-left properties at once.	Separate values by a space in the following order (those that are not defined will use inherited or default initial values): border-left-width border-left-style border-left-color	div { border-left:2px solid green; } div { border-left:thick double #00FF00; }
border-color**	Declares the border color of all four borders at once.	Valid color names, RGB values, hexadecimal notation, or the predefined value transparent. Separate the color for each border by a space, declaring the colors for the borders in the following order: border-top-color border-right-color border-bottom-color border-left-color	div { border-color:green red blue olive; } div { border-color:green; } div { border-color:green red; } div { border-color:green red blue; }

**Undeclared values work as further shorthand notation. If only one color value is declared, all four borders will use that color. If two colors are declared, the top and bottom borders will use the first color while the right and left borders will use the second color. If three colors are declared, the top border will use the first color, the right and left borders will use the second color, and the bottom border will use the third color.

border-style	Declares the border style of all four borders at once.	none, hidden, dotted, dashed, solid, double, groove, ridge, inset, outset	div { border-style:solid dotted dashed double; } div { border-style:solid; } div { border-style:solid dotted; } div { border-style:solid dotted dashed; }
border-width	Declares the width of all four borders at once.	Lengths or the following predefined values: thin, medium, thick	div { border-width:1px 3px 5px 2px; } div { border-width:thin; } div { border-width:2px 4px; } div { border-width:2px 4px 5px; }
border	Used as a shorthand to declare the border properties when all four borders will have the same appearance.	Separate values by a space in the following order (those that are not defined will use inherited or default initial values): border-width border-style border-color	div { border:1px double green; } div { border:thin solid #00FF00; }

CSS 2.0 Classification/Positioning Properties

Property	Description	Possible Values	Examples
clear	Declares the side(s) of an element where no previous floating elements are allowed to be adjacent.	left, right, both, none	div { clear:right; } div { clear:both; }
cursor	Declares the type of cursor to be displayed.	URL values, and the following predefined values: auto, crosshair, default, pointer, move, e-resize, ne-resize, nw-resize, n-resize, se-resize, sw-resize, s-resize, w-resize, text, wait, help	div { cursor:crosshair; } div { cursor:url(image.csr); } div { cursor:url(image.csr), pointer; }
display	Declares if/how the element displays.	none, inline, block, list-item, run-in, compact, marker, table, inline-table, table-row-group, table-header-group, table-footer-group, table-row, table-column-group, table-column, table-cell, table-caption,	div { display:none; } div { display:inline; } div { display:marker; }
float	Declares whether a box should float to the left or right of other content, or whether it should not be floated at all.	left, right, none	div { float:left; } div { float:right; }
visibility	Declares the visibility of boxes generated by an element.	visible, hidden, collapse	div { visibility:visible; } div { visibility:hidden; }
top	Declares the distance that the top content edge of the element is offset below the top edge of its containing block. The position property of the element must also be set to a value other than static.	Lengths, percentages, and the predefined value auto.	div { top:15px; } div { top:2%; }
right	Declares the distance that the right content edge of the element is offset to the left of the right edge of its containing block. The position property of the element must also be set to a value other than static.	Lengths, percentages, and the predefined value auto.	div { right:15px; } div { right:2%; }

bottom	Declares the distance that the bottom content edge of the element is offset above the bottom edge of its containing block. The position property of the element must also be set to a value other than static.	Lengths, percentages, and the predefined value auto.	div { bottom:15px; } div { bottom:2%; }
left	Declares the distance that the left content edge of the element is offset to the right of the left edge of its containing block. The position property of the element must also be set to a value other than static.	Lengths, percentages, and the predefined value auto.	div { left:15px; } div { left:2%; }
position	Declares the type of positioning of an element.	static relative absolute fixed	div { position:absolute; left:10px; bottom:200px; } div { position:relative; right:10px; top:200px; }
<p>static: default positioning for all elements, means the element is not positioned and occurs where it normally would in the document (top, bottom, left, right do NOT work with static)</p> <p>relative: positions element relative to its normal position (use left, right, top, bottom to move)</p> <p>absolute: is positioned relative to the first parent element that has a position other than static - is removed from the normal flow or act like they do not exist- can overlap (use left, right, top, bottom to move)</p> <p>fixed: is positioned relative to the browser window - removed from normal flow - can overlap - do NOT scroll with window (use left, right, top, bottom to move)</p>			
clip	Declares the shape of a clipped region when the value of the overflow property is set to a value other than visible.	Shapes, or the predefined value auto. In CSS 2.0 2, the only valid shape is a rectangle, using the following format to specify the offset lengths from each side of the box: rect(top, right, bottom, left)	div { clip:auto; } div { clip:rect(2px, 4px, 7px, 5px); }
overflow	Declares how content that overflows the element's box is handled.	visible hidden scroll auto	div { overflow:hidden; } div { overflow:scroll; }
vertical-align	Declares the vertical alignment of an inline-level element or a table cell.	Lengths, percentages, and the following predefined values: baseline, sub, super, top, text-top, middle, bottom, text-bottom	span { vertical-align:middle; } td { vertical-align:top; }
z-index	Declares the stack order of the element.	Integer values and the predefined value auto.	div { z-index:2; } div { z-index:auto; }

CSS 2.0 Dimension Properties

Property	Description	Possible Values	Examples
height	Declares the height of the element.	Lengths, percentages, and the predefined value auto.	div { height:200px; } div { height:50%; }
max-height	Declares the maximum height of the element.	Lengths, percentages, and the predefined value auto.	div { max-height:200px; } div { max-height:50%; }
min-height	Declares the minimum height of the element.	Lengths, percentages, and the predefined value auto.	div { min-height:200px; } div { min-height:50%; }
width	Declares the width of the element.	Lengths, percentages, and the predefined value auto.	div { width:500px; } div { width:75%; }
max-width	Declares the maximum width of the element.	Lengths, percentages, and the predefined value auto.	div { max-width:500px; } div { max-width:75%; }
min-width	Declares the minimum width of the element.	Lengths, percentages, and the predefined value auto.	div { min-width:500px; } div { min-width:75%; }

CSS 2.0 Font Properties

Property	Description	Possible Values	Examples
font-family	Declares the name of the font to be used. Previously set in HTML via the face attribute in a tag.	Valid font family names or generic family names, i.e. Arial, Verdana, sans-serif, "Times New Roman", Times, serif, etc. Font family names can be separated by a comma in the same declaration to allow additional and/or generic family names to be used if the preferred font is unable to be displayed.	div { font-family:Arial; } div { font-family:"Times New Roman"} div { font-family:Arial, Helvetica, sans-serif; } }
font-size	Declares the size of the font. Previously set in HTML via the size attribute in a tag.	Lengths (number and unit type— i.e. 1em, 12pt, 10px, 80%) or one of the following predefined values: xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger	div { font-size:70%; } div { font-size:0.85em; } div { font-size:medium; }
font-size-adjust	Limited browser support: Was part of CSS 2.0 2, but not in CSS 2.0 2.1. This property may return in CSS 2.0 3. Declares the aspect value (font size divided by x-height).	Numeric value	div { font-size-adjust:0.54; } div { font-size-adjust:0.46; }
font-stretch	Limited browser support: Was part of CSS 2.0 2, but not in CSS 2.0 2.1. This property may return in CSS 2.0 3. Declares the stretch of the font face.	normal, wider, narrower, ultra-condensed, extra-condensed, condensed, semi-condensed, semi-expanded, expanded, extra-expanded, ultra-expanded	div { font-stretch:narrower; } div { font-stretch:ultra-expanded; }
font-style	Declares the font style.	normal italic oblique	div { font-style:italic; } div { font-style:oblique; }
font-variant	Declares the font variant.	normal small-caps	div { font-variant:normal; } div { font-variant:small-caps; }
font-weight	Declares the font weight (lightness or boldness)	normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800, 900	div { font-weight:bolder; } div { font-weight:200; }
font	Used as a shorthand property to declare all of the font properties at once (except font-size-adjust and font-stretch).	Separate values by a space in the following order (those that are not defined will use inherited or default initial values): font-style font-variant font-weight font-size line-height font-family	div { font:italic small-caps bold 1em 1.2em Arial } div { font:bold 0.8em Verdana }

CSS 2.0 Box Model



Explanation of the different parts:

Margin - Clears an area around the border. The margin does not have a background color, and it is completely transparent

Border - A border that lies around the padding and content. The border is affected by the background color of the box

Padding - Clears an area around the content. The padding is affected by the background color of the box

Content - The content of the box, where text and images appear

In order to set the width and height of an element correctly in all browsers, you need to know how the box model works.

Important: This element `div {width:250px; padding:10px; border:5px solid gray; margin:10px;}` has a TOTAL width of 300px when added together.

CSS 2.0 List Properties

Property	Description	Possible Values	Examples
list-style-type	Declares the type of list marker used.	disc, circle, square, decimal, decimal-leading-zero, lower-roman, upper-roman, lower-alpha, upper-alpha, lower-greek, lower-latin, upper-latin, hebrew, armenian, georgian, cjk-ideographic, hiragana, katakana, hiragana-iroha, katakana-iroha	ol { list-style-type:upper-roman; } ul { list-style-type:square; }
list-style-position	Declares the position of the list marker.	inside outside	ol { list-style-position:inside; } ul { list-style-position:outside; }
list-style-image	Declares an image to be used as the list marker.	URL values.	ul { list-style-image:url(image.jpg); }
list-style	Shorthand property to declare three list properties at once.	Separate values by a space in the following order (those that are not defined will use inherited or default initial values): list-style-type list-style-position list-style-image	ul { list-style:disc inside url(image.gif); } ol { list-style:upper-roman outside; }
marker-offset	Declares the marker offset for elements with a value of marker set for the display property.	Lengths and the predefined value auto.	li:before { display:marker; marker-offset:5px; }

CSS 2.0 Margin Properties

Property	Description	Possible Values	Examples
margin-top	Declares the top margin for the element.	Lengths, percentages, and the predefined value auto.	div { margin-top:5px; } div { margin-top:15%; }
margin-right	Declares the right margin for the element.	Lengths, percentages, and the predefined value auto.	div { margin-right:5px; } div { margin-right:15%; }
margin-bottom	Declares the bottom margin for the element.	Lengths, percentages, and the predefined value auto.	div { margin-bottom:5px; } div { margin-bottom:15%; }
margin-left	Declares the left margin for the element.	Lengths, percentages, and the predefined value auto.	div { margin-left:5px; } div { margin-left:15%; }
margin	Shorthand property used to declare all the margin properties at once.	Separate values by a space in the following order (those that are not defined will use inherited or default initial values): margin-top margin-right margin-bottom margin-left	div { margin:5px 12px 4px 7px; } div { margin:5px; } div { margin:5px 10px; } div { margin:5px 7px 4px auto; }

Undeclared values work as further shorthand notation. If only one length value is declared, all four margins will use that length. If two lengths are declared, the top and bottom margins will use the first length while the right and left margins will use the second length. If three lengths are declared, the top margin will use the first length, the right and left margins will use the second length, and the bottom margin will use the third length.

Pseudo-classes

Property	What it Does	Possible Values	Examples
:link	Adds special style to an unvisited link	define look	a:link {color:red;}
:visited	Adds special style to a visited link	define look	a:visited {color:red;}
:hover	Adds special style to an element when you mouse over it	define look	a:hover {color:red;}
:active	Adds special style to an activated element	define look	a:active {color:red;}

CSS 2.0 Outline Properties

Property	Description	Possible Values	Examples
outline-color	Declares the outline color.	Valid color names, RGB values, hexadecimal notation.	div { outline-color:green; } div { outline-color:#00FF00; }
outline-style	Declares the style of the outline.	none, dotted, dashed, solid, double, groove, ridge, inset, outset	div { outline-style:solid; } div { outline-style:inset; }
outline-width	Declares the width of the outline.	Lengths or the following predefined values: thin, medium, thick	div { outline-width:2px; } div { outline-width:thin; }
outline	Used as a shorthand property to set all the background properties at once.	Separate values by a space in the following order (those that are not defined will use inherited or default initial values): outline-color outline-style outline-width	div { outline:green solid 2px; } div { outline:#00FF00 double thick; }

CSS 2.0 Padding Properties

Property	Description	Possible Values	Examples
padding-top	Declares the top padding for the element.	Lengths, percentages, and the predefined value auto.	div { padding-top:5px; } div { padding-top:15%; }
padding-right	Declares the right padding for the element.	Lengths, percentages, and the predefined value auto.	div { padding-right:5px; } div { padding-right:15%; }
padding-bottom	Declares the bottom padding for the element.	Lengths, percentages, and the predefined value auto.	div { padding-bottom:5px; } div { padding-bottom:15%; }
padding-left	Declares the left padding for the element.	Lengths, percentages, and the predefined value auto.	div { padding-left:5px; } div { padding-left:15%; }
padding	Shorthand property used to declare all the margin properties at once.	Separate values by a space in the following order (those that are not defined will use inherited or default initial values): padding-top padding-right padding-bottom padding-left	div { padding:5px 12px 4px 7px; } div { padding:5px; } div { padding:5px 10px; } div { padding:5px 7px auto 4px; }

Undeclared values work as further shorthand notation. If only one length value is declared, all four sides will use that length. If two lengths are declared, the top and bottom sides will use the first length while the right and left sides will use the second length. If three lengths are declared, the top side will use the first length, the right and left sides will use the second length, and the bottom side will use the third length.

CSS 2.0 Table Properties

Property	Description	Possible Values	Examples
border-collapse	Declares the way borders are displayed.	collapse separate	table { border-collapse:collapse; } table { border-collapse:separate; }
border-spacing	Declares the distance separating borders (if border-collapse is separate).	Lengths for the horizontal and vertical spacing, separated by a space.	table { border-spacing:5px; } table { border-spacing:5px 10px; }
If one length is value is declared, that length is used for both the horizontal and vertical spacing. If two lengths are declared, the first one is used for horizontal spacing and the second one is used for vertical spacing.			
caption-side	Declares where the table caption is displayed in relation to the table.	top, bottom, left, right	caption { caption-side:top; } caption { caption-side:right; }
empty-cells	Declares the way empty cells are displayed (if border-collapse is separate).	show hide	table { empty-cells:show; } table { empty-cells:hide; }
table-layout	Declares the type of table layout.	auto fixed	table { table-layout:auto; } table { table-layout:fixed; }

CSS 2.0 Text Properties

Property	Description	Possible Values	Examples
color	Declares the color of the text.	Valid color names, RGB values, hexadecimal notation.	div { color:green; } div { color:rgb(0,255,0); } div { color:#00FF00; } or div { color:#0F0; }
direction	Declares the reading direction of the text.	ltr (= left-to-right) rtl (= right-to-left)	div { direction:ltr; } div { direction:rtl; }
line-height	Declares the distance between lines.	Numbers, percentages, lengths, and the predefined value of normal.	div { line-height:normal; } div { line-height:2em; } div { line-height:125%; }
letter-spacing	Declares the amount of space between text characters.	A length (in addition to the default space) or the predefined value of normal.	div { letter-spacing:normal; } div { letter-spacing:5px; } div { letter-spacing:-1px; }
text-align	Declares the horizontal alignment of inline content.	left, right, center , justify	div { text-align:center; } div { text-align:right; }
If used on a set of table cells, this property can be given a string value to which the text of each row of the column will be aligned.			td { text-align:"."; }
text-decoration	Declares the text decoration.	none underline overline line-through blink	div { text-decoration:none; } div { text-decoration:underline; }
text-indent	Declares the indentation of the first line of text.	Lengths and percentages.	div { text-indent:12px; } div { text-indent:2%; }
text-shadow	Declares shadow effects on the text.	A list containing a color followed by numeric values (separated by spaces) that specify: 1. The color for the shadow effect 2. Horizontal distance to the right of the text 3. Vertical distance below the text 4. Blur radius	div { text-shadow:green 2px 2px 7px; } div { text-shadow:olive -3px -4px 5px; }
text-transform	Declares the capitalization effects on the letters in the text.	none, capitalize, uppercase, lowercase	div { text-transform:uppercase; } div { text-transform:lowercase; }
unicode-bidi	Declares values relating to bidirectional text. May be used in conjunction with the the direction property.	normal embed bidi-override	div { unicode-bidi:embed; } div { unicode-bidi:bidi-override; }
white-space	Declares how white space is handled in an element.	normal pre nowrap	div { white-space:pre; } div { white-space:nowrap; }
word-spacing	Declares the space between words in the text.	A length (in addition to the default space) or the predefined value of normal.	div { word-spacing:normal; } div { word-spacing:1.5em; }