Cascading Style Sheet

(CSS)

INDEX

What is CSS?

- **CSS** stands for **C**ascading **S**tyle **S**heets
- Styles define **how to display** HTML elements
- Styles are normally stored in **Style Sheets**
- Styles were added to HTML 4.0 **to solve a problem**
- **External Style Sheets** can save a lot of work
- External Style Sheets are stored in **CSS files**
- Multiple style definitions will **cascade** into one

Why Css?

The original HTML was never intended to contain tags for formatting a document. HTML tags were intended to define the content of a document, like:

```
<p>This is a paragraph.</p>

<h1>This is a heading</h1>
```

When tags like <font> and color attributes were added to the HTML 3.2 specification, it started a nightmare for web developers. Development of large web sites where fonts and color information had to be added to every single Web page, became a long, expensive and unduly painful process.

To solve this problem, the World Wide Web Consortium (W3C) - responsible for standardizing HTML - created CSS in addition to HTML 4.0.

With HTML 4.0, all formatting can be removed from the HTML document and stored in a separate CSS file.

All browsers support CSS today.

CSS save a lot of work

Styles sheets define HOW HTML elements are to be displayed.

Styles are normally saved in external .css files. External style sheets enable you to change the appearance and layout of all the pages in a Web site, just by editing one single CSS document!

CSS types and priority

Style sheets allow style information to be specified in many ways.

Styles can be specified:

- inside an HTML element
- inside the head section of an HTML page
- in an external CSS file

Tip: Even multiple external style sheets can be referenced inside a single HTML document.

Cascading order - What style will be used when there is more than one style specified for an HTML element?

Generally speaking we can say that all the styles will "cascade" into a new "virtual" style sheet by the following rules, where number four has the highest priority:

1. Browser default
2. External style sheet
3. Internal style sheet (in the head section)
4. Inline style (inside an HTML element)

So, an inline style (inside an HTML element) has the highest priority, which means that it will override a style defined inside the <head> tag, or in an external style sheet, or in a browser (a default value).

If the link to the external style sheet is placed after the internal style sheet in HTML <head>, the external style sheet will override the internal style sheet!

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 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 want 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 describe one property on each line, like this:

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

Grouping

You can group selectors. Separate each selector with a comma. In the example below we have grouped all the header elements. All header elements will be displayed in green text color:

```
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>
```

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 #.

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
}
```

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;
font-family:arial
}
```

CSS Types

How to Insert a Style Sheet

When a browser reads a style sheet, it will format the document according to it. There are three ways of inserting a style sheet:

External Style Sheet

An external style sheet is ideal when the style is applied to many pages. With an external style sheet, you can change the look of an entire Web site by changing one file. Each page must link to the style sheet using the <link> tag. The <link> tag goes inside the head section:

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

The browser will read the style definitions from the file mystyle.css, and format the document according to it.

An external style sheet can be written in any text editor. The file should not contain any html tags. Your style sheet should be saved with a .css extension. An example of a style sheet file is shown below:

```
hr {color:sienna}
p {margin-left:20px}
body {background-image:url("images/back40.gif")}
```

Do not leave spaces between the property value and the units! "margin-left:20 px" (instead of "margin-left:20px") will only work in IE6, but it will not work in Firefox or Opera.

Internal Style Sheet

An internal style sheet should be used when a single document has a unique style. You define internal styles in the head section by using the <style> tag, like this:

```
<head>
<style type="text/css">
hr {color:sienna}
p {margin-left:20px}
body {background-image:url("images/back40.gif")}
</style>
</head>
```

The browser will now read the style definitions, and format the document according to it.

Note: A browser normally ignores unknown tags. This means that an old browser that does not support styles, will ignore the <style> tag, but the content of the <style> tag will be displayed on the page. It is possible to prevent an old browser from displaying the content by hiding it in the HTML comment element:

```
<head>
<style type="text/css">
<!--
hr {color:sienna}
p {margin-left:20px}
body {background-image:url("images/back40.gif")}
-->
</style>
</head>
```

Inline Styles

An inline style loses many of the advantages of style sheets by mixing content with presentation. Use this method sparingly, such as when a style is to be applied to a single occurrence of an element.

To use inline styles you use the style attribute in the relevant tag. The style attribute can contain any CSS property. The example shows how to change the color and the left margin of a paragraph:

```
<p style="color:sienna;margin-left:20px">This is a paragraph.</p>
```

Multiple Style Sheets

If some properties have been set for the same selector in different style sheets, the values will be inherited from the more specific style sheet.

For example, an external style sheet has these properties for the h3 selector:

```
h3
{
color:red;
text-align:left;
font-size:8pt
}
```

And an internal style sheet has these properties for the h3 selector:

```
h3
{
text-align:right;
font-size:20pt
}
```

If the page with the internal style sheet also links to the external style sheet the properties for h3 will be:

color:red;
text-align:right;
font-size:20pt
The color is inherited from the external style sheet and the text-alignment and the font-size is replaced by the internal style sheet.

CSS Background

1. **Color as a background**

```
<style type="text/css">

body

{

background-color:yellow;

}

h1

{

background-color:#00ff00;

}

p

{

background-color:rgb(255,0,255);

}

</style>
```

Note: Default value is transparent.

2. **Image as a background**

```
<style type="text/css">
body {background-image:url('paper.gif')}
</style>
```

---

3. **How to repeat background Image?**

Note: By default, a background-image is placed at the top-left corner of an element, and repeated both vertically and horizontally.

Background Image Vertical Repeat

```
<style type="text/css">
body
{
background-image:url('paper.gif');
background-repeat:repeat-x;
}
</style>
```

Background Image Horizontal Repeat

```
<style type="text/css">
body
{
background-image:url('paper.gif');
background-repeat:repeat-y;
}
</style>
```

Display a Background Image only one time

```
<style type="text/css">
body
{
background-image:url('paper.gif');
background-repeat:no-repeat;
}
</style>
```

4. **Image as a background**

```
<style type="text/css">
body
{
background-image:url('smiley.gif');
background-repeat:no-repeat;
background-attachment:fixed;
background-position:center;
}
</style>
```

Note: For this to work in Firefox and Opera, the background-attachment property must be set to "fixed". Background will not move scroll.

Background-attachment: Sets whether a background image is fixed or scrolls with the rest of the pages. **Default value is Scroll.**

5. **Background Image Position**

```
<style type="text/css">
body {background-image:url('paper.gif')}
</style>
```

Property Values

Value	Description
top left top center top right center left center center center right bottom left bottom center bottom right	If you only specify one keyword, the second value will be "center". Default value is: 0% 0%
x% y%	The first value is the horizontal position and the second value is the vertical. The top left corner is 0% 0%. The right bottom corner is 100% 100%. If you only specify one value, the other value will be 50%.
xpos ypos	The first value is the horizontal position and the second value is the vertical. The top left corner is 0 0. Units can be pixels (0px 0px) or any other CSS units. If you only specify one value, the other value will be 50%. You can mix % and positions
inherit	Specifies that the setting of the background-position property should be inherited from the parent element

Example:

```
<style type="text/css">
body
{
background-image: url('smiley.gif');
background-repeat: no-repeat;
background-attachment:fixed;
background-position: 0% 20%;
background-position: 50px 100px;
}
</style>
```

CSS Text

---

1. **Text Color**

The color property is used to set the color of the text. The color can be set by:

- name - specify a color name, like "red"
- RGB - specify an RGB value, like "rgb(255,0,0)"
- Hex - specify a hex value, like "#ff0000"

The default color for a page is defined in the body selector.

Example

```
body {color:blue}
h1 {color:#00ff00}
h2 {color:rgb(255,0,0)}
```

2. **Text Alignment**

The text-align property is used to set the horizontal alignment of a text.

Text can be centered, or aligned to the left or right, or justified.

When text-align is set to "justify", each line is stretched so that every line has equal width, and the left and right margins are straight (like in magazines and newspapers).

Example:

```
<html>

<head>

<style type="text/css">

h1 {text-align:center}

p.date {text-align:right}

p.main {text-align:justify}

</style>

</head>

<body>

<h1>CSS text-align Example</h1>
```

```
<p class="date">May, 2009</p>

<p class="main">In my younger and more vulnerable years my father gave me
some advice that I've been turning over in my mind ever since. 'Whenever
you feel like criticizing anyone,' he told me, just remember that all the
people in this world haven't had the advantages that you've had.'</p>

</body>

</html>
```

Note: Try to resize the browser window to see how justify works.

3. **Text Decoration**

The text-decoration property is used to set or remove decorations from text.

The text-decoration property is mostly used to remove underlines from links for design purposes:

Example:

```
<style type="text/css">

h1 {text-decoration:overline}

h2 {text-decoration:line-through}

h3 {text-decoration:underline}

h4 {text-decoration:blink}

</style>
```

Note: The "blink" value is not supported in IE, Chrome, or Safari.

4. **Text Transformation**

The text-transform property is used to specify uppercase and lowercase letters in a text.

It can be used to turn everything into uppercase or lowercase letters, or capitalize the first letter of each word.

Example:

```
<html>
<head>
<style type="text/css">
```

```
p.uppercase {text-transform:uppercase}
p.lowercase {text-transform:lowercase}
p.capitalize {text-transform:capitalize}
</style>
</head>

<body>
<p class="uppercase">This is some text in a paragraph</p>
<p class="lowercase">This is some text in a paragraph</p>
<p class="capitalize">This is some text in a paragraph</p>
</body>
</html>
```

5. **Text Indentation**

The text-indentation property is used to specify the indentation of the first line of a text.

Example:

```
<html>

<head>
<style type="text/css">
      p {text-indent:50px}
</style>
</head>

<body>
<p>In my younger and more vulnerable years my father gave me some advice that I've been turning over in my mind ever since.</p>
</body>
</html>
```

6. **Specify the space between characters**

This property is used to specify the space between characters. Negative value will decrease the space while positive value will increase the space.

Example:

```
<html>

<head>
<style type="text/css">
h1 {letter-spacing: -3px}
h4 {letter-spacing: 0.5cm}
</style>
</head>

<body>
<h1>This is header 1</h1>
```

```
<h4>This is header 4</h4>
</body>
</html>
```

7. **Specify the space between lines**

This property is used to specify the space between lines. Negative value will decrease the space while positive value will increase the space.

Example:

```
<html>
<head>
<style type="text/css">
p.small {line-height: 90%}
p.big {line-height: 20px}
</style>
</head>

<body>
<p>
This is a paragraph with a standard line-height.
This is a paragraph with a standard line-height.
This is a paragraph with a standard line-height.
</p>

<p class="small">
This is a paragraph with a smaller line-height.
This is a paragraph with a smaller line-height.
This is a paragraph with a smaller line-height.
This is a paragraph with a smaller line-height.
</p>

<p class="big">
This is a paragraph with a bigger line-height.
This is a paragraph with a bigger line-height.
This is a paragraph with a bigger line-height.
This is a paragraph with a bigger line-height.
</p>

</body>
</html>
```

Note: **line-height** can be given in % and pixel. The default line height in most browsers is about 110% to 120%.

8. **Text direction of an element**

This property is used to specify the direction of an element.

ltr: Text will start from left side.

rtl: Text will start from left side.

Example:

```
<html>
<head>
<style type="text/css">
div.one
{
direction:rtl;
}
div.two
{
direction:ltr;
}
</style>
</head>

<body>
<div class="one">Some text. Right-to-left direction.</div>
<div class="two">Some text. Left-to-right direction.</div>
</body>
</html>
```

9. **Increase the white space between words**

This property is used to specify the white space between words. Positive value increases the space while negative value decreases the space.

Example:

```
<html>
<head>
<style type="text/css">
p
{
word-spacing:30px;
}
</style>
</head>
<body>

<p>
This is some text. This is some text.
</p>

</body>
</html>
```

10. **Text wrapping**

White-space: nowrap – text will not wrap horizontal scroll will be visible.

White-space: wrap – text will wrap horizontal scroll will not be visible.

Example:

```
<html>
<head>
<style type="text/css">
p
{
white-space:nowrap;
}
</style>
</head>
<body>

<p>
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
</p>

</body>
</html>
<style type="text/css">
p
{
white-space:nowrap;
}
</style>
```

CSS Font

1. **Font Family**

The font family of a text is set with the font-family property.

Start with the font you want, and end with a generic family, to let the browser pick a similar font in the generic family, if no other fonts are available.

Note: If the name of a font family is more than one word, it must be in quotation marks, like font-family: "Times New Roman".

More than one font family is specified in a comma-separated list:

Example:

```
p{font-family:"Times New Roman",Georgia,Serif}
```

2. **Font Style**

The font-style property is mostly used to specify italic text.

This property has three values:

- normal - The text is shown normally
- italic - The text is shown in italics
- oblique - The text is "leaning" (oblique is very similar to italic, but less supported)

Example:

```
<html>

<head>

<style type="text/css">

p.normal {font-style:normal}

p.italic {font-style:italic}

p.oblique {font-style:oblique}

</style>

</head>

<body>

<p class="normal">This is a paragraph, normal.</p>

<p class="italic">This is a paragraph, italic.</p>

<p class="oblique">This is a paragraph, oblique.</p>

</body>

</html>
```

3. **Font Size**

The font-size property sets the size of the text.

The font-size value can be an absolute, or relative size.

Absolute size:

- Sets the text to a specified size
- Does not allow a user to change the text size in all browsers (bad for accessibility reasons)
- Absolute size is useful when the physical size of the output is known

Relative size:

- Sets the size relative to surrounding elements
- Allows a user to change the text size in browsers

Note: If you do not specify a font size, the default size for normal text, like paragraphs, is 16px **(16px=1em)**.

Example:

```
<html>

<head>

<style>

h1 {font-size:40px}

h2 {font-size:30px}

p {font-size:14px}

</style>

</head>

<body>

<h1>This is heading 1</h1>

<h2>This is heading 2</h2>

<p>This is a paragraph.</p>

</body>

</html>
```

---

4. **Why font Size in em ?**

To avoid the resizing problem with Internet Explorer, many developers use em instead of pixels.

The em size unit is recommended by the W3C.

1em is equal to the current font size. The default text size in browsers is 16px. So, the default size of 1em is 16px.

The size can be calculated from pixels to em using this formula: *pixels*/16=*em*

Example:

```
h1 {font-size:2.5em} /* 40px/16=2.5em */
```

5. **Font Weight**

Sets the weight of a font.

Values: normal,bold,bolder,lighter,100,200,300,400,500,600,700,800,900

Example:

```
<html>

<head>

<style type="text/css">

p.normal {font-weight:normal}

p.thick {font-weight:bold}

p.thicker {font-weight:900}

</style>

</head>

<body>

<p class="normal">This is a paragraph.</p>

<p class="thick">This is a paragraph.</p>

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

```
</body>

</html>
```

6. **Font Variant**

Displays text in a small-caps font or a normal font.

Example:

```
<html>

<head>

<style type="text/css">

p.normal {font-variant:normal}

p.small {font-variant:small-caps}

</style>

</head>

<body>

<p class="normal">This is a paragraph.</p>

<p class="small">This is a paragraph.</p>

</body>

</html>
```

CSS Units

Measurements

Unit	Description
%	percentage
in	inch
cm	centimeter
mm	millimeter
em	1em is equal to the current font size. 2em means 2 times the size of the current font. E.g., if an element is displayed with a font of 12 pt, then '2em' is 24 pt. The 'em' is a very useful unit in CSS, since it can adapt automatically to the font that the reader uses
ex	one ex is the x-height of a font (x-height is usually about half the font-size)

pt	point (1 pt is the same as 1/72 inch)
pc	pica (1 pc is the same as 12 points)
px	pixels (a dot on the computer screen)

Colors

Unit	**Description**
color name	A color name (e.g. red)
rgb(x,x,x)	An RGB value (e.g. rgb(255,0,0))
rgb(x%, x%, x%)	An RGB percentage value (e.g. rgb(100%,0%,0%))
#rrggbb	A HEX number (e.g. #ff0000)

CSS Tables

1. **Set the layout of a table**

Example:

```
<html>

<head>

<style type="text/css">

table.ex1 {table-layout:auto}

table.ex2 {table-layout:fixed}

</style>

</head>

<body>

<table class="ex1" border="1" width="100%">

<tr>

<td width="5%">10000000000000000000000000000</td>

<td width="95%">10000000</td>

</tr>

</table>

<br />

<table class="ex2" border="1" width="100%">
```

```
<tr>

<td width="5%">10000000000000000000000000000</td>

<td width="95%">10000000</td>

</tr>

</table>

</body>

</html>
```

2. **Collapse a table border**

Sets whether the table borders are collapsed into a single border or detached as in standard HTML

Example:

```
<html>

<head>

<style type="text/css">

table {border-collapse:collapse}

</style>

</head>

<body>

<table border="1">

<tr>

<td>Peter</td>

<td>Griffin</td>

</tr>

<tr>

<td>Lois</td>
```

```
<td>Griffin</td>

</tr>

</table>

</body>

</html>
```

3. **Empty cells in a table border**

Sets whether or not to show empty cells in a table (only for the "separated borders" model)

Example:

```
<html>

<head>

<style type="text/css">

table

{

border-collapse:separate;

empty-cells:hide;

}

</style>

</head>

<body>

<table border="1">

<tr>

<td>Peter</td>

<td>Griffin</td>

</tr>
```

```
<tr>

<td>Lois</td>

<td></td>

</tr>

</table>

</body>

</html>
```

CSS Image Opacity

Note: This is not yet a CSS standard. However, it works in all modern browsers, and is a part of the W3C CSS 3 recommendation.

1. **Transparent Image**

```
<img src="klematis.jpg" width="150" height="113" alt="klematis"
style="opacity:0.4;filter:alpha(opacity=40)" />
```

Firefox uses the property **opacity:x** for transparency, while IE uses **filter:alpha(opacity=x)**.

Tip: The CSS3 syntax for transparency is **opacity:x**.

In Firefox (opacity:x) x can be a value from 0.0 - 1.0. A lower value makes the element more transparent.

In IE (filter:alpha(opacity=x)) x can be a value from 0 - 100. A lower value makes the element more transparent.

2. **Transparent Image – Mouseover effect**

```
<img src="klematis.jpg" style="opacity:0.4;filter:alpha(opacity=40)"
onmouseover="this.style.opacity=1;this.filters.alpha.opacity=100"
onmouseout="this.style.opacity=0.4;this.filters.alpha.opacity=40" />
```

The syntax for this in Firefox is: **this.style.opacity=1** and the syntax in IE is: **this.filters.alpha.opacity=100**.

When the mouse pointer moves away from the image, we want the image to be transparent again. This is done in the onmouseout attribute.

CSS Cursor

This property displays different types of cursor by moving on the text.

Example:

```
<html>

<body>

<p>Mouse over the words to change the cursor.</p>

<span style="cursor:auto">auto</span><br />

<span style="cursor:crosshair">crosshair</span><br />

<span style="cursor:default">default</span><br />

<span style="cursor:e-resize">e-resize</span><br />

<span style="cursor:help">help</span><br />

<span style="cursor:move">move</span><br />

<span style="cursor:n-resize">n-resize</span><br />

<span style="cursor:ne-resize">ne-resize</span><br />

<span style="cursor:nw-resize">nw-resize</span><br />

<span style="cursor:pointer">pointer</span><br />

<span style="cursor:progress">progress</span><br />

<span style="cursor:s-resize">s-resize</span><br />

<span style="cursor:se-resize">se-resize</span><br />

<span style="cursor:sw-resize">sw-resize</span><br />

<span style="cursor:text">text</span><br />

<span style="cursor:w-resize">w-resize</span><br />

<span style="cursor:wait">wait</span><br />

</body>

</html>
```

CSS Invisible Tags

Notice that the invisible heading still takes up space. Marked(**) statement will not be dispalyed

Example:

```
<style type="text/css">

h1.visible {visibility:visible}

h1.hidden {visibility:hidden}

</style>

<body>

<h1 class="visible">This is a visible heading</h1>

**<h1 class="hidden">This is an invisible heading</h1>

</body>
```

CSS Horizontal Menu

Example:

```
<html>

<head>

<style type="text/css">

ul

{

float:left;

width:100%;

padding:0;

margin:0;

list-style-type:none;
```

```
}
a
{
float:left;
width:6em;
text-decoration:none;
color:white;
background-color:#aaaaaa;
padding:0.2em 0.6em;
border-right:1px solid white;
}
a:hover {background-color:#ff3300}
li {display:inline}
</style>
</head>
<body>
<ul>
<li><a href="#">Link one</a></li>
<li><a href="#">Link two</a></li>
<li><a href="#">Link three</a></li>
<li><a href="#">Link four</a></li>
</ul>
</body>
</html>
```

Output:

CSS Margin

The margin clears an area around an element (outside the border). The margin does not have a background color, and is completely transparent.

The top, right, bottom, and left margin can be changed independently using separate properties. A shorthand margin property can also be used, to change all margins at once.

1.Possible Values

Value	**Description**
auto	The browser sets the margin. The result of this is dependant of the browser
length	Defines a fixed margin (in pixels, pt, em, etc.)
%	Defines a margin in % of the containing element

It is possible to use negative values, to overlap content.

In CSS, it is possible to specify different margins for different sides:

Example:

```
margin-top:100px;
margin-bottom:100px;
margin-right:50px;
margin-left:50px;
```

1.Margin Shorthand Property

To shorten the code, it is possible to specify all the margin properties in one property. This is called a shorthand property.

The shorthand property for all the margin properties is "margin":

The margin property can have from one to four values.

- **margin:25px 50px 75px 100px;**
  - top margin is 25px
  - right margin is 50px
  - bottom margin is 75px
  - left margin is 100px

- **margin:25px 50px 75px;**
  - top margin is 25px

- right and left margins are 50px
- bottom margin is 75px

- **margin:25px 50px;**
  - top and bottom margins are 25px
  - right and left margins are 50px

- **margin:25px;**
  - all four margins are 25px

Note: Margin can be set in cm and %.

CSS Border

1. Border Style

The border-style property specifies what kind of border to display.

None of the other border properties will have any effect unless border-style is set.

border-style Values

none: Defines no border

dotted: Defines a dotted border

dashed: Defines a dashed border

solid: Defines a solid border

double: Defines two borders. The width of the two borders are the same as the border-width value

groove: Defines a 3D grooved border. The effect depends on the border-color value

ridge: Defines a 3D ridged border. The effect depends on the border-color value

inset: Defines a 3D inset border. The effect depends on the border-color value

outset: Defines a 3D outset border. The effect depends on the border-color value

Example:

```
<style type="text/css">

p.none {border-style:none}

p.dotted {border-style:dotted}

p.dashed {border-style:dashed}

p.solid {border-style:solid}

p.double {border-style:double}

p.groove {border-style:groove}

p.ridge {border-style:ridge}

p.inset {border-style:inset}

p.outset {border-style:outset}

p.hidden {border-style:hidden}

</style>
```

2. Border width

The border-width property is used to set the width of the border.

The width is set in pixels, or by using one of the three pre-defined values: thin, medium, or thick.

Note: The "border-width" property does not work if it is used alone. Use the "border-style" property to set the borders first.

Example:

```
p.one
{
border-style:solid;
border-width:5px;
```

```
}
p.two
{
border-style:solid;
border-width:medium;
}
```

3. Border color

The border-color property is used to set the color of the border. The color can be set by:

- name - specify a color name, like "red"
- RGB - specify a RGB value, like "rgb(255,0,0)"
- Hex - specify a hex value, like "#ff0000"

You can also set the border color to "transparent".

Note: The "border-color" property does not work if it is used alone. Use the "border-style" property to set the borders first.

Example:

```
p.one
{
border-style:solid;
border-color:red;
}
p.two
{
border-style:solid;
border-color:#98bf21;
}
```

3. Border Individual Side

In CSS it is possible to specify different borders for different sides:

Example:

```
p
{
border-top-style:dotted;
border-right-style:solid;
border-bottom-style:dotted;
border-left-style:solid;
}
```

The border-style property can have from one to four values.

- **border-style:dotted solid double dashed;**

    - top border is dotted
    - right border is solid
    - bottom border is double
    - left border is dashed

- **border-style:dotted solid double;**
    - top border is dotted
    - right and left borders are solid
    - bottom border is double

- **border-style:dotted solid;**
    - top and bottom borders are dotted
    - right and left borders are solid

- **border-style:dotted;**
    - all four borders are dotted

The border-style property is used in the example above. However, it also works with border-width and border-color.

Note: When using the border property, the orders of the values are:

- border-width
- border-style
- border-color

It does not matter if one of the values above are missing (although, border-style is required), as long as the rest are in the specified order.

All CSS Border Properties

The number in the "CSS" column indicates in which CSS version the property is defined (CSS1 or CSS2).

Property	Description	Values	CSS
border	Sets all the border properties in one declaration	*border-width border-style border-color*	1
border-bottom	Sets all the bottom border properties in one declaration	*border-bottom-width border-bottom-style border-bottom-color*	1
border-bottom-color	Sets the color of the bottom border	*border-color*	2
border-bottom-style	Sets the style of the bottom border	*border-style*	2

border-bottom-width	Sets the width of the bottom border	*border-width*	1
border-color	Sets the color of the four borders	*color_name* *hex_number* *rgb_number* transparent inherit	1
border-left	Sets all the left border properties in one declaration	*border-left-width* *border-left-style* *border-left-color*	1
border-left-color	Sets the color of the left border	*border-color*	2
border-left-style	Sets the style of the left border	*border-style*	2
border-left-width	Sets the width of the left border	*border-width*	1
border-right	Sets all the right border properties in one declaration	*border-right-width* *border-right-style* *border-right-color*	1
border-right-color	Sets the color of the right border	*border-color*	2
border-right-style	Sets the style of the right border	*border-style*	2
border-right-width	Sets the width of the right border	*border-width*	1
border-style	Sets the style of the four borders	none hidden dotted dashed solid double groove ridge inset outset inherit	1
border-top	Sets all the top border properties in one declaration	*border-top-width* *border-top-style* *border-top-color*	1
border-top-color	Sets the color of the top border	*border-color*	2
border-top-style	Sets the style of the top border	*border-style*	2
border-top-width	Sets the width of the top border	*border-width*	1
border-width	Sets the width of the four borders	thin medium thick *length* inherit	1

CSS Padding

The padding clears an area around the content (inside the border) of an element. The padding is affected by the background color of the element.

The top, right, bottom, and left padding can be changed independently using separate properties. A shorthand padding property can also be used, to change all paddings at once.

Possible Values

Value	Description
length	Defines a fixed padding (in pixels, pt, em, etc.)
%	Defines a padding in % of the containing element

Padding - Individual sides

In CSS, it is possible to specify different padding for different sides:

Example

```
padding-top:25px;
padding-bottom:25px;
padding-right:50px;
padding-left:50px;
```

Padding - Shorthand property

To shorten the code, it is possible to specify all the padding properties in one property. This is called a shorthand property.

The shorthand property for all the padding properties is "padding":

Example

```
padding:25px 50px;
```

The margin property can have from one to four values.

- **padding:25px 50px 75px 100px;**
  - top padding is 25px
  - right padding is 50px
  - bottom padding is 75px
  - left padding is 100px

- **padding:25px 50px 75px;**
  - top padding is 25px
  - right and left paddings are 50px
  - bottom padding is 75px

- **padding:25px 50px;**

- top and bottom paddings are 25px
- right and left paddings are 50px

- **padding:25px;**
  - all four paddings are 25px

CSS Pseudo-classes

CSS pseudo-classes are used to add special effects to some selectors.

Syntax

The syntax of pseudo-classes:

```
selector:pseudo-class {property:value}
```

CSS classes can also be used with pseudo-classes:

```
selector.class:pseudo-class {property:value}
```

Anchor Pseudo-classes

Links can be displayed in different ways in a CSS-supporting browser:

```
a:link {color:#FF0000} /* unvisited link */
a:visited {color:#00FF00} /* visited link */
a:hover {color:#FF00FF} /* mouse over link */
a:active {color:#0000FF} /* selected link */
```

Note: a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective!!

Note: a:active MUST come after a:hover in the CSS definition in order to be effective!!

Note: Pseudo-class names are not case-sensitive.

Pseudo-classes and CSS Classes

Pseudo-classes can be combined with CSS classes:

```
a.red:visited {color:#FF0000}

<a class="red" href="css_syntax.asp">CSS Syntax</a>
```

If the link in the example above has been visited, it will be displayed in red.

CSS - The :first-child Pseudo-class

The :first-child pseudo-class matches a specified element that is the first child of another element.

Note: For :first-child to work in IE a <!DOCTYPE> must be declared.

Match the first <p> element

In the following example, the selector matches any <p> element that is the first child of any element:

Example

```
<html>
<head>
<style type="text/css">
p:first-child
{
color:blue
}
</style>
</head>

<body>
<p>I am a strong man.</p>
<p>I am a strong man.</p>
</body>
</html>
```

Match the first <i> element in all <p> elements

In the following example, the selector matches the first <i> element in all <p> elements:

Example

```
<html>
<head>
<style type="text/css">
p > i:first-child
{
font-weight:bold
}
</style>
</head>

<body>
<p>I am a <i>strong</i> man. I am a <i>strong</i> man.</p>
<p>I am a <i>strong</i> man. I am a <i>strong</i> man.</p>
</body>
</html>
```

Match all <i> elements in all first child <p> elements

In the following example, the selector matches all <i> elements in <p> elements that are the first child of another element:

Example

```
<html>
<head>
<style type="text/css">
p:first-child i
{
color:blue
}
</style>
</head>

<body>
<p>I am a <i>strong</i> man. I am a <i>strong</i> man.</p>
<p>I am a <i>strong</i> man. I am a <i>strong</i> man.</p>
</body>
</html>
```

CSS - The :lang Pseudo-class

The :lang pseudo-class allows you to define special rules for different languages. In the example below, the :lang class defines the type of quotation marks for q elements with a lang attribute with a value of "no":

```
<html>

<head>
<style type="text/css">
q:lang(no)
{
quotes:"~" "~"
}
</style>
</head>

<body>
<p>Some text <q lang="no">A quote in a paragraph</q>
Some text.</p>
</body>

</html>
```

CSS Pseudo-elements

CSS pseudo-elements are used to add special effects to some selectors.

Syntax

The syntax of pseudo-elements:

```
selector:pseudo-element {property:value}
```

CSS classes can also be used with pseudo-elements:

```
selector.class:pseudo-element {property:value}
```

The :first-line Pseudo-element

The "first-line" pseudo-element is used to add special styles to the first line of the text in a selector:

```
p:first-line {color:#0000ff;font-variant:small-caps}

<p>Some text that ends up on two or more lines</p>
```

The output could be something like this:

```
Some text that ends
up on two or more lines
```

In the example above the browser displays the first line formatted according to the "first-line" pseudo element. **Where** the browser breaks the line depends on the size of the browser window.

Note: The "first-line" pseudo-element can only be used with block-level elements.

Note: The following properties apply to the "first-line" pseudo-element:;

- font properties
- color properties
- background properties
- word-spacing
- letter-spacing
- text-decoration
- vertical-align
- text-transform
- line-height
- clear

The :first-letter Pseudo-element

The "first-letter" pseudo-element is used to add special style to the first letter of the text in a selector:

```
p:first-letter {color:#ff0000;font-size:xx-large}

<p>The first words of an article...</p>
```

The output could be something like this:

```
he first words of an article...
```

Note: The "first-letter" pseudo-element can only be used with block-level elements.

Note: The following properties apply to the "first-letter" pseudo-element:

- font properties
- color properties
- background properties
- margin properties
- padding properties
- border properties
- text-decoration
- vertical-align (only if "float" is "none")
- text-transform
- line-height
- float
- clear

Pseudo-elements and CSS Classes

Pseudo-elements can be combined with CSS classes:

```
p.article:first-letter {color:#ff0000}

<p class="article">A paragraph in an article</p>
```

The example above will make the first letter of all paragraphs with class="article" red.

Multiple Pseudo-elements

Several pseudo-elements can be combined:

```
p:first-letter {color:#ff0000;font-size:xx-large}
p:first-line {color:#0000ff}

<p>The first words of an article...</p>
```

The output could be something like this:

```
he first
words of an
article...
```

In the example above the first letter of the paragraph will be red with a font size of 24pt. The rest of the first line would be blue while the rest of the paragraph would be the default color.

CSS - The :before Pseudo-element

The ":before" pseudo-element can be used to insert some content before the content of an element.

The style below will play a sound before each occurrence of an <h1> element:

```
h1:before
{
content:url(beep.wav)
}
```

CSS - The :after Pseudo-element

The ":after" pseudo-element can be used to insert some content after the content of an element.

The style below will play a sound after each occurrence of an <h1> element:

```
h1:after
{
content:url(beep.wav)
}
```