**What is CSS?**

* **CSS** stands for **C**ascading **S**tyle **S**heets
* Styles define **how to display** HTML elements
* 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**

**CSS Demo**

An HTML document can be displayed with different styles: [See how it works](http://www.w3schools.com/css/demo_default.htm)

**Styles Solved a Big Problem**

HTML was never intended to contain tags for formatting a document.

HTML was intended to define the content of a document, like:

<h1>This is a heading</h1>

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

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 were added to every single page, became a long and expensive process.

To solve this problem, the World Wide Web Consortium (W3C) created CSS.

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

All browsers support CSS today.

**CSS Saves a Lot of Work!**

CSS defines HOW HTML elements are to be displayed.

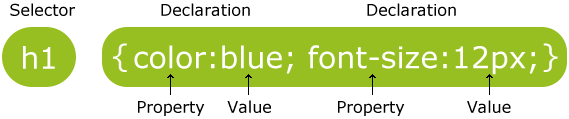
Styles are normally saved in external .css files. External style sheets enable you to change the appearance and layout of all the pages in a Web site, just by editing one single file!

**examples**

* Look at [Example 1](http://www.w3schools.com/css/tryit.asp?filename=ex1)
* Look at [Example 2](http://www.w3schools.com/css/tryit.asp?filename=ex2)

**CSS Syntax**

A CSS rule has two main parts: a selector, and one or more declarations:



The selector is normally the HTML element you want to style.

Each declaration consists of a property and a value.

The property is the style attribute you want to change. Each property has a value.

**CSS Example**

A CSS declaration always ends with a semicolon, and declaration groups are surrounded by curly brackets:

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

To make the CSS more readable, you can put one declaration on each line, like this:

**Example**

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

**CSS Comments**

Comments are used to explain your code, and may help you when you edit the source code at a later date. Comments are 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;  
}

## The id and class Selectors

In addition to setting a style for a HTML element, CSS allows you to specify your own selectors called "id" and "class".

## The id Selector

The id selector is used to specify a style for a single, unique element.

The id selector uses the id attribute of the HTML element, and is defined with a "#".

The style rule below will be applied to the element with id="para1":

## Example

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

## The class Selector

The class selector is used to specify a style for a group of elements. Unlike the id selector, the class selector is most often used on several elements.

This allows you to set a particular style for many HTML elements with the same class.

The class selector uses the HTML class attribute, and is defined with a "."

In the example below, all HTML elements with class="center" will be center-aligned:

## Example

.center {text-align:center;}

You can also specify that only specific HTML elements should be affected by a class.

In the example below, all p elements with class="center" will be center-aligned:

## Example

p.center {text-align:center;}

## Three Ways to Insert CSS

There are three ways of inserting a style sheet:

* External style sheet
* Internal style sheet
* Inline style

## 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>

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");}

RemarkDo not add a space between the property value and the unit (such as margin-left:20 px). The correct way is: margin-left:20px

## 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 of an HTML page, 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>

## Inline Styles

An inline style loses many of the advantages of style sheets by mixing content with presentation. Use this method sparingly!

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.

## Multiple Styles Will Cascade into One

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).

SS background properties are used to define the background effects of an element.

CSS properties used for background effects:

* background-color
* background-image
* background-repeat
* background-attachment
* background-position

**Background Color**

The background-color property specifies the background color of an element.

The background color of a page is defined in the body selector:

**Example**

body {background-color:#b0c4de;}

With CSS, a color is most often specified by:

* a HEX value - like "#ff0000"
* an RGB value - like "rgb(255,0,0)"
* a color name - like "red"

Look at [CSS Color Values](http://www.w3schools.com/cssref/css_colors_legal.asp) for a complete list of possible color values.

In the example below, the h1, p, and div elements have different background colors:

**Example**

h1 {background-color:#6495ed;}  
p {background-color:#e0ffff;}  
div {background-color:#b0c4de;}

**Background Image**

The background-image property specifies an image to use as the background of an element.

By default, the image is repeated so it covers the entire element.

The background image for a page can be set like this:

**Example**

body {background-image:url('paper.gif');}

Below is an example of a bad combination of text and background image. The text is almost not readable:

**Example**

body {background-image:url('bgdesert.jpg');}

**Background Image - Repeat Horizontally or Vertically**

By default, the background-image property repeats an image both horizontally and vertically.

Some images should be repeated only horizontally or vertically, or they will look strange, like this:

**Example**

body  
{  
background-image:url('gradient2.png');  
}

If the image is repeated only horizontally (repeat-x), the background will look better:

**Example**

body  
{  
background-image:url('gradient2.png');  
background-repeat:repeat-x;  
}

**Background Image - Set position and no-repeat**

RemarkWhen using a background image, use an image that does not disturb the text.

Showing the image only once is specified by the background-repeat property:

**Example**

body  
{  
background-image:url('img\_tree.png');  
background-repeat:no-repeat;  
}

In the example above, the background image is shown in the same place as the text. We want to change the position of the image, so that it does not disturb the text too much.

The position of the image is specified by the background-position property:

**Example**

body  
{  
background-image:url('img\_tree.png');  
background-repeat:no-repeat;  
background-position:right top;  
}

**Background - Shorthand property**

As you can see from the examples above, there are many properties to consider when dealing with backgrounds.

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

The shorthand property for background is simply "background":

**Example**

body {background:#ffffff url('img\_tree.png') no-repeat right top;}

When using the shorthand property the order of the property values is:

* background-color
* background-image
* background-repeat
* background-attachment
* background-position

It does not matter if one of the property values is missing, as long as the ones that are present are in this order.

This example uses more advanced CSS. Take a look: [Advanced example](http://www.w3schools.com/css/tryit.asp?filename=trycss_background_shorthand2)

**More Examples**

[How to set a fixed background image](http://www.w3schools.com/css/tryit.asp?filename=trycss_background-attachment)  
This example demonstrates how to set a fixed background image. The image will not scroll with the rest of the page.

**All CSS Background Properties**

|  |  |
| --- | --- |
| **Property** | **Description** |
| [background](http://www.w3schools.com/cssref/css3_pr_background.asp) | Sets all the background properties in one declaration |
| [background-attachment](http://www.w3schools.com/cssref/pr_background-attachment.asp) | Sets whether a background image is fixed or scrolls with the rest of the page |
| [background-color](http://www.w3schools.com/cssref/pr_background-color.asp) | Sets the background color of an element |
| [background-image](http://www.w3schools.com/cssref/pr_background-image.asp) | Sets the background image for an element |
| [background-position](http://www.w3schools.com/cssref/pr_background-position.asp) | Sets the starting position of a background image |
| [background-repeat](http://www.w3schools.com/cssref/pr_background-repeat.asp) | Sets how a background image will be repeated |

# text formatting

This text is styled with some of the text formatting properties. The heading uses the text-align, text-transform, and color properties. The paragraph is indented, aligned, and the space between characters is specified. The underline is removed from the ["Try it yourself"](http://www.w3schools.com/css/tryit.asp?filename=trycss_text) link.

## Text Color

The color property is used to set the color of the text.

With CSS, a color is most often specified by:

* a HEX value - like "#ff0000"
* an RGB value - like "rgb(255,0,0)"
* a color name - like "red"

Look at [CSS Color Values](http://www.w3schools.com/cssref/css_colors_legal.asp) for a complete list of possible color values.

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);}

RemarkFor W3C compliant CSS: If you define the color property, you must also define the background-color property.

## 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

h1 {text-align:center;}  
p.date {text-align:right;}  
p.main {text-align:justify;}

## 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

a {text-decoration:none;}

It can also be used to decorate text:

## Example

h1 {text-decoration:overline;}  
h2 {text-decoration:line-through;}  
h3 {text-decoration:underline;}  
h4 {text-decoration:blink;}

RemarkIt is not recommended to underline text that is not a link, as this often confuses users.

## 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

p.uppercase {text-transform:uppercase;}  
p.lowercase {text-transform:lowercase;}  
p.capitalize {text-transform:capitalize;}

## Text Indentation

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

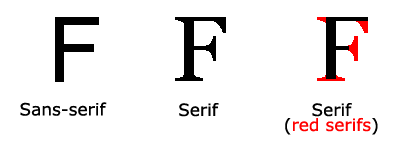
## Example

p {text-indent:50px;}

## All CSS Text Properties

|  |  |
| --- | --- |
| **Property** | **Description** |
| [color](http://www.w3schools.com/cssref/pr_text_color.asp) | Sets the color of text |
| [direction](http://www.w3schools.com/cssref/pr_text_direction.asp) | Specifies the text direction/writing direction |
| [letter-spacing](http://www.w3schools.com/cssref/pr_text_letter-spacing.asp) | Increases or decreases the space between characters in a text |
| [line-height](http://www.w3schools.com/cssref/pr_dim_line-height.asp) | Sets the line height |
| [text-align](http://www.w3schools.com/cssref/pr_text_text-align.asp) | Specifies the horizontal alignment of text |
| [text-decoration](http://www.w3schools.com/cssref/pr_text_text-decoration.asp) | Specifies the decoration added to text |
| [text-indent](http://www.w3schools.com/cssref/pr_text_text-indent.asp) | Specifies the indentation of the first line in a text-block |
| text-shadow | Specifies the shadow effect added to text |
| [text-transform](http://www.w3schools.com/cssref/pr_text_text-transform.asp) | Controls the capitalization of text |
| unicode-bidi |  |
| [vertical-align](http://www.w3schools.com/cssref/pr_pos_vertical-align.asp) | Sets the vertical alignment of an element |
| [white-space](http://www.w3schools.com/cssref/pr_text_white-space.asp) | Specifies how white-space inside an element is handled |
| [word-spacing](http://www.w3schools.com/cssref/pr_text_word-spacing.asp) | Increases or decreases the space between words in a text |

**Difference Between Serif and Sans-serif Fonts**



RemarkOn computer screens, sans-serif fonts are considered easier to read than serif fonts.

**CSS Font Families**

In CSS, there are two types of font family names:

* **generic family** - a group of font families with a similar look (like "Serif" or "Monospace")
* **font family** - a specific font family (like "Times New Roman" or "Arial")

|  |  |  |
| --- | --- | --- |
| **Generic family** | **Font family** | **Description** |
| Serif | Times New Roman Georgia | Serif fonts have small lines at the ends on some characters |
| Sans-serif | Arial Verdana | "Sans" means without - these fonts do not have the lines at the ends of characters |
| Monospace | Courier New Lucida Console | All monospace characters have the same width |

**Font Family**

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

The font-family property should hold several font names as a "fallback" system. If the browser does not support the first font, it tries the next font.

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", Times, serif;}

**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**

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

**Font Size**

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

Being able to manage the text size is important in web design. However, you should not use font size adjustments to make paragraphs look like headings, or headings look like paragraphs.

Always use the proper HTML tags, like <h1> - <h6> for headings and <p> for paragraphs.

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

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

**Set Font Size With Pixels**

Setting the text size with pixels gives you full control over the text size:

**Example**

h1 {font-size:40px;}  
h2 {font-size:30px;}  
p {font-size:14px;}

**Set Font Size With Em**

To avoid the resizing problem with older versions of 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 \*/  
h2 {font-size:1.875em;} /\* 30px/16=1.875em \*/  
p {font-size:0.875em;} /\* 14px/16=0.875em \*/

In the example above, the text size in em is the same as the previous example in pixels. However, with the em size, it is possible to adjust the text size in all browsers.

Unfortunately, there is still a problem with older versions of IE. The text becomes larger than it should when made larger, and smaller than it should when made smaller.

**Use a Combination of Percent and Em**

The solution that works in all browsers, is to set a default font-size in percent for the <body> element:**Example**

body {font-size:100%;}  
h1 {font-size:2.5em;}  
h2 {font-size:1.875em;}  
p {font-size:0.875em;}

**All CSS Font Properties**

|  |  |
| --- | --- |
| **Property** | **Description** |
| [font](http://www.w3schools.com/cssref/pr_font_font.asp) | Sets all the font properties in one declaration |
| [font-family](http://www.w3schools.com/cssref/pr_font_font-family.asp) | Specifies the font family for text |
| [font-size](http://www.w3schools.com/cssref/pr_font_font-size.asp) | Specifies the font size of text |
| [font-style](http://www.w3schools.com/cssref/pr_font_font-style.asp) | Specifies the font style for text |
| [font-variant](http://www.w3schools.com/cssref/pr_font_font-variant.asp) | Specifies whether or not a text should be displayed in a small-caps font |
| [font-weight](http://www.w3schools.com/cssref/pr_font_weight.asp) | Specifies the weight of a font |

**Styling Links**

Links can be styled with any CSS property (e.g. color, font-family, background, etc.).

Special for links are that they can be styled differently depending on what state they are in.

The four links states are:

* a:link - a normal, unvisited link
* a:visited - a link the user has visited
* a:hover - a link when the user mouses over it
* a:active - a link the moment it is clicked

**Example**

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 \*/

When setting the style for several link states, there are some order rules:

* a:hover MUST come after a:link and a:visited
* a:active MUST come after a:hover

**Common Link Styles**

In the example above the link changes color depending on what state it is in.

Lets go through some of the other common ways to style links:

**Text Decoration**

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

**Example**

a:link {text-decoration:none;}  
a:visited {text-decoration:none;}  
a:hover {text-decoration:underline;}  
a:active {text-decoration:underline;}

**Background Color**

The background-color property specifies the background color for links:

**Example**

a:link {background-color:#B2FF99;}  
a:visited {background-color:#FFFF85;}  
a:hover {background-color:#FF704D;}  
a:active {background-color:#FF704D;}

The CSS list properties allow you to:

* Set different list item markers for ordered lists
* Set different list item markers for unordered lists
* Set an image as the list item marker

**List**

In HTML, there are two types of lists:

* unordered lists - the list items are marked with bullets
* ordered lists - the list items are marked with numbers or letters

With CSS, lists can be styled further, and images can be used as the list item marker.

**Different List Item Markers**

The type of list item marker is specified with the list-style-type property:

**Example**

ul.a {list-style-type: circle;}  
ul.b {list-style-type: square;}  
  
ol.c {list-style-type: upper-roman;}  
ol.d {list-style-type: lower-alpha;}

Some of the values are for unordered lists, and some for ordered lists.

**An Image as The List Item Marker**

To specify an image as the list item marker, use the list-style-image property:

**Example**

ul  
{  
list-style-image: url('sqpurple.gif');  
}

The example above does not display equally in all browsers. IE and Opera will display the image-marker a little bit higher than Firefox, Chrome, and Safari.

If you want the image-marker to be placed equally in all browsers, a crossbrowser solution is explained below.

**Crossbrowser Solution**

The following example displays the image-marker equally in all browsers:

**Example**

ul  
{  
list-style-type: none;  
padding: 0px;  
margin: 0px;  
}  
ul li  
{  
background-image: url(sqpurple.gif);  
background-repeat: no-repeat;  
background-position: 0px 5px;   
padding-left: 14px;   
}

Example explained:

* For ul:
  + Set the list-style-type to none to remove the list item marker
  + Set both padding and margin to 0px (for cross-browser compatibility)
* For all li in ul:
  + Set the URL of the image, and show it only once (no-repeat)
  + Position the image where you want it (left 0px and down 5px)
  + Position the text in the list with padding-left

**List - Shorthand property**

It is also possible to specify all the list properties in one, single property. This is called a shorthand property.

The shorthand property used for lists, is the list-style property:

**Example**

ul  
{  
list-style: square url("sqpurple.gif");  
}

When using the shorthand property, the order of the values are:

* list-style-type
* list-style-position (for a description, see the CSS properties table below)
* list-style-image

It does not matter if one of the values above are missing, as long as the rest are in the specified order.

Examples

**All CSS List Properties**

|  |  |
| --- | --- |
| **Property** | **Description** |
| [list-style](http://www.w3schools.com/cssref/pr_list-style.asp) | Sets all the properties for a list in one declaration |
| [list-style-image](http://www.w3schools.com/cssref/pr_list-style-image.asp) | Specifies an image as the list-item marker |
| [list-style-position](http://www.w3schools.com/cssref/pr_list-style-position.asp) | Specifies if the list-item markers should appear inside or outside the content flow |
| [list-style-type](http://www.w3schools.com/cssref/pr_list-style-type.asp) | Specifies the type of list-item marker |

## Table Borders

To specify table borders in CSS, use the border property.

The example below specifies a black border for table, th, and td elements:

## Example

table, th, td  
{  
border: 1px solid black;  
}

Notice that the table in the example above has double borders. This is because both the table and the th/td elements have separate borders.

To display a single border for the table, use the border-collapse property.

## Collapse Borders

The border-collapse property sets whether the table borders are collapsed into a single border or separated:

## Example

table  
{  
border-collapse:collapse;  
}  
table,th, td  
{  
border: 1px solid black;  
}

## Table Width and Height

Width and height of a table is defined by the width and height properties.

The example below sets the width of the table to 100%, and the height of the th elements to 50px:

## Example

table   
{  
width:100%;  
}  
th  
{  
height:50px;  
}

## Table Text Alignment

The text in a table is aligned with the text-align and vertical-align properties.

The text-align property sets the horizontal alignment, like left, right, or center:

## Example

td  
{  
text-align:right;  
}

The vertical-align property sets the vertical alignment, like top, bottom, or middle:

## Example

td  
{  
height:50px;  
vertical-align:bottom;  
}

## Table Padding

To control the space between the border and content in a table, use the padding property on td and th elements: Example

td  
{  
padding:15px;  
}

**Table Color**

The example below specifies the color of the borders, and the text and background color of th elements:

## Example

table, td, th  
{  
border:1px solid green;  
}  
th  
{  
background-color:green;color:white;  
}