

# A User's Guide to Style Sheets

---

Handan Selamoglu  
Microsoft Corporation

Last updated: October 24, 1996

## Abstract

Style sheets, introduced in Microsoft Internet Explorer 3.0, are about to change the way people build and maintain their Web pages. A style sheet may not be obvious to readers of Web pages because it works behind the scenes to give Web authors and designers more control and flexibility in their word- and design-smithing on the Web.

Style sheets deliver some of the formatting features and control authors and designers have been enjoying for years in traditional desktop publishing—features that have been sorely missed on the Web. With style sheets, you can finally specify point sizes, page margins, and leading (spacing between lines). You can also create any number of formatting variations for a single HTML tag.

In addition to this new functionality, a style sheet provides a convenient service: It separates the formatting information from the actual content on your HTML pages, so it becomes much easier to design and revise your pages.

This document provides an introduction to style sheets and explains usage and syntax, as defined by the World Wide Web Consortium (W3C) and supported by Microsoft Internet Explorer 3.0.

## New in Internet Explorer 3.0 final release:

- New attribute: margin-top.
- Negative values for the margin-left, margin-right, margin-top, and text-indent attributes, allowing "outdents" and creation of special effects.
- Style sheets on/off option: Readers have the option to turn off style sheets through the View Options dialog box (Advanced tab). This allows you to use your browser's default values if your system does not support fonts specified by the style sheet.
- You can now use background colors and images on any element. Background repetition, scrolling, and positioning are also supported.

**Note on Internet Explorer 3.01 release:** Please see "Updated Margin Usage in Style Sheets" on this site for information on the Internet Explorer 3.01 fix for margin attributes.

## Contents

Introduction

    Browser support

    Cascading style sheets

Three Ways to Add Style Sheets to Your Web Pages

    Basic syntax

- Linking to a style sheet
- Embedding a style block
- Using inline styles
- Mixing methods
- Additional options

#### Style Reference Guide

- Quick Reference
- Point Size
- Typeface
- Weight
- Style
- Line height
- Grouping font attributes
- Colors
- Special text effects
- Margins
- Grouping margin attributes
- Alignment
- Indentation
- Background colors and images

#### Tips & Tricks

- Designing for multiple browsers
- Taking advantage of inheritance

#### Samples

---

## Introduction

---

A style sheet is a template that controls the formatting of HTML tags on your Web pages. If you use Microsoft Word, you will find the concept of style sheets very similar to Word templates: You can alter the appearance of a Word document by changing the formatting assigned to styles in your document. Similarly, you can alter the appearance of a Web page by changing the formatting assigned to HTML tags through a style sheet, and thus override the browser's default specification for those tags.

If you publish on the Web, you may want to take advantage of style sheets for three main reasons: additional formatting, better control, and easier customization. Specifically, style sheets allow you to:

- Use leading, margins, indents, point sizes, and text background colors on your Web pages. Authors and designers take these formatting features for granted, but HTML does not support these directly through its standard set of tags.
- Stop using awkward workarounds for basic formatting tasks (such as <BLOCKQUOTE> for margins).
- Change the formatting of individual Web pages or your entire site without touching every single line in your HTML files.
- Reduce the clutter of tags on your HTML pages (because you don't need to turn tags on and off to change the design of a single element).

- Define design variations through classes (for example, if you need five different paragraph styles, you can specify five classes for the <P> tag, e.g., *P.normal*, *P.indented*, *P.underlined*, *P.red*, *P.dblspaced*).

## Browser support

Style sheets are fairly new to the Web. Internet Explorer 3.0 is the first Web browser to support style sheets. The Internet Explorer 3.0 implementation of style sheets is based on the W3C working draft called "Cascading Style Sheets, Level 1" (dated May 1996, <http://www.w3.org/pub/WWW/TR/WD-css1.html>). Spyglass and Netscape have also indicated that they will support style sheets in future versions of their browsers. Browsers that don't support style sheets display the pages using the browser's default specs instead. (See "Tips & Tricks" for instructions on how to set up your style sheets so they don't cause problems in other browsers.)

## Cascading style sheets

The W3C draft refers to "cascading style sheets" because you can use multiple styles to control the appearance of your page, and the browser follows rules (a "cascading order") to determine precedence and to resolve conflicts. For example, the Web author can use linked, embedded, **and** inline styles (described below) in one document, and Web readers can have their own personal style sheets. If the linked style sheet defines a heading tag as blue, the top-of-page style sheet defines it as red, and the reader wants to see purple, the browser has to decide what to display. Cascading rules are discussed in "Mixing Methods" later in this article.

## Three Ways to Add Styles to Your Web Pages

---

You can use style sheets in three ways, depending on your design needs:

- By **linking** to a style sheet from your HTML file. This method allows you to change the appearance of *multiple Web pages* by tweaking a single file.
- By **embedding** a style sheet in your HTML file. This method allows you to change the appearance of a *single Web page* by tweaking a few lines.
- By adding **inline** styles to your HTML file. This gives you a quick way to change the appearance of a *single tag*, a *group of tags*, or a *block of information* on your page.

You can use one, two, or all three of these methods on your pages. The sections below discuss each method and syntax in detail. For detailed information, see the W3C documents: HTML3 and Style Sheets and Cascading Style Sheets, Level 1.

**Note:** Internet Explorer 3.0 does not support imported style sheets.

## Basic Syntax

Both linked and embedded style sheets include one or more style definitions. (The inline style syntax is somewhat different, as we'll discuss later.) A style definition consists of an HTML tag (*any tag*), followed by a listing of properties for that tag within curly braces. Each property is identified by the property name, followed by a colon and the property value. Multiple properties

are separated by semicolons. For example, the following style definition assigns the <H1> tag a specific font size (15 points) and font weight (boldface):

```
H1 {font-size: 15pt;
    font-weight: bold}
```

You can thus create style definitions for any number of HTML tags and either place them in a separate file or embed them directly within your Web pages. For a list of properties, see the "Style Reference Guide" section below.

## Linking to a style sheet

To link to an external style sheet, you simply create a file with your style definitions (as explained below for embedded styles), save it with a .CSS file extension, and link to it from your Web page. This way, you can use the same style sheet for any number of pages on your site.

For example, if your style sheet is called MYSTYLES.CSS and is located at the address <http://internet-name/mystyles.css>, you would add the following to your Web page, within the <HEAD> tag:

```
<HEAD>
<TITLE>Title of article</TITLE>
<LINK REL=STYLESHEET HREF="http://internet-name/mystyles.css" TYPE="text/css">
</HEAD>
```

**Note:** Internet Explorer 3.0 (final release) now registers the Internet Media (MIME) type for style sheets, so you don't have to register the "text/css" type on the server yourself. This version of the browser allows you to link to a single style sheet. The TYPE attribute is ignored.

## Embedding a STYLE block

To embed a style sheet, you add a <STYLE> </STYLE> block at the top of your document, between the <HTML> and <BODY> tags. This allows you to change the appearance of a *single* Web page. The <STYLE> tag has one parameter, TYPE, which specifies the Internet Media type as "text/css" (allowing browsers that do not support this type to ignore style sheets). The <STYLE> tag is followed by any number of style definitions and terminated with the </STYLE> tag.

The following example specifies styles for the <BODY>, <H1>, <H2>, and <P> tags:

```
<HTML>
<STYLE TYPE="text/css">
<!--
  BODY {font: 10pt "Arial"}
  H1 {font: 15pt/17pt "Arial";
     font-weight: bold;
     color: maroon}
  H2 {font: 13pt/15pt "Arial";
     font-weight: bold;
     color: blue}
  P {font: 10pt/12pt "Arial";
     color: black}
-->
```

```
</STYLE>
<BODY>
...
</BODY>
</HTML>
```

See the "Tips & Tricks" section to find out why I've added HTML comments (<!-- and -->) within the style block.

## Using inline styles

If you want to take advantage of point sizes, indentation, or other styles in only a few sections of your Web page, you can use inline styles.

Inline style definitions affect individual occurrences of a tag. These are embedded within the tag itself using the STYLE parameter. The following HTML code indents a specific <P> tag:

```
<P STYLE="margin-left: 0.5in; margin-right: 0.5in">
This line will be indented on the left and right.
<P>
This line will receive no indentation.
```

Here's the result (this display requires Internet Explorer 3.0):

This line will be indented on the left and right.

This line will receive no indentation.

If the inline style conflicts with an embedded or linked style, the inline style overrides for that particular occurrence. For example, if the line above appears on a Web page that defines <P> with 1-inch margins through a linked style sheet, all paragraphs on the Web page will get 1-inch margins except for the <P> above, which will get 0.5-inch margins.

If you want to change the appearance of an entire section, you can use the <DIV> tag to define styles globally for that section. The following example changes the color and point size of a block of text by using the <DIV> tag (this has the same effect as assigning these styles separately for the <P>, <UL>, and <LI> tags):

```
<DIV STYLE="font-size: 10pt; color: red">
<P>
The style specification affects everything on this page until the DIV close tag.
<UL>
<LI>This is red and 10 pt.
<LI>So is this.
</UL>
</DIV>
```

Result (requires Internet Explorer 3.0):

The style specification affects everything on this page until the DIV close tag.

- This is red and 10 pt.
- So is this.

The following example uses the <DIV> tag for a block of text, but overrides it for a specific <LI> tag:

```
<DIV STYLE="font-size: 10pt; color: red">
<P>
The style specification affects everything on this page until the DIV close tag,
except for the second list item.
<UL>
<LI>This is red and 10 pt.
<LI STYLE="color: blue">This is blue and 10 pt.
</UL>
</DIV>
```

Result (requires Internet Explorer 3.0):

The style specification affects everything on this page until the DIV close tag, except for the second list item.

- This is red and 10 pt.
- This is blue and 10 pt.

Inline styles are simple to use if you're focusing on a few tags or sections on your Web page (for example, you're highlighting a heading or indenting an abstract). However, inline styles clutter up your HTML pages, and revisions require more detailed attention (because you have to change multiple lines scattered throughout your HTML file). If you want to make global changes to one or more Web pages, you'll find that using a centralized STYLE block (either linked or embedded) is easier and more efficient.

The <SPAN> tag is very similar to the <DIV> tag, but is used for smaller page elements (for example, words, even letters) instead of text blocks or containers. You can use <SPAN> to change styles within a sentence or even within a word, or to create special effects. See the Microsoft Typography group's demo pages for some interesting effects created with <DIV> and <SPAN>.

## Mixing methods

If you're just starting out with style sheets, it's probably a good idea to stick to one method that best addresses your needs. However, you may have complicated requirements that can be met only by using two or all three methods. I've listed some scenarios below. (**Note:** These are future scenarios; Internet Explorer 3.0 does not yet support links to multiple style sheets, the use of linked and embedded styles within the same document, or readers' style sheets.)

- Storing your styles in separate style sheets (one storing the basic page layout for simple documents, and others customizing the layout for different types of documents), and linking to multiple .CSS files at the top of your page.

- Linking to your company-wide style sheet for consistency, but providing style variations (through classes) with an embedded style sheet.
- Linking to your company-wide style sheet, but overriding several specific elements using inline styles.
- When readers' style sheets are supported, using style sheets to provide formatting that is essential to your page design, and letting readers adjust the point size, leading, and typeface to suit their personal preferences.

So what will happen if you use multiple style sheets that have conflicting style information? That's where the "cascading order" will come into play. According to the W3C working draft, the author's style sheets will override the reader's style sheet, which in turn will override the browser's default values. If the author uses all three methods listed above, the inline styles will take precedence over the embedded <STYLE> block, which will override the linked style sheet.

## Additional options

### Simplification through grouping

If you want to assign the same formatting to multiple tags, for example:

```
H1 {font-size: 15pt;
    font-weight: bold;
    color: maroon}
H2 {font-size: 15pt;
    font-weight: bold;
    color: maroon}
H3 {font-size: 15pt;
    font-weight: bold;
    color: maroon}
```

you can group them thus:

```
H1, H2, H3 {font-size: 15pt;
            font-weight: bold;
            color: maroon}
```

You can also group formatting specifications. Take:

```
H1 {font-size: 15pt;
    line-height: 17pt;
    font-weight: bold;
    font-family: "Arial"
    font-style: normal}
```

and simplify it this way:

```
H1 {font: 15pt/17pt bold "Arial" normal}
```

(See the "Style Reference Guide" section below for more information on grouping font specs.)

## Variations through classes

Use classes to create variations for a single HTML tag. For example, if you'd like to use three colors for your H1 headings (say, depending on context), you'd define three classes in your STYLE tag:

```
<STYLE>
  H1.red {font: 15pt/17pt;
         color: red}
  H1.green {font: 15pt/17pt;
           color: green}
  H1.blue {font: 15pt/17pt;
          color: blue}
</STYLE>
```

and use them as follows on your Web page:

```
<H1 CLASS=red>This is the red heading</H1>
...
<H1 CLASS=blue>This is the blue heading</H1>
...
<H1 CLASS=green>You get the picture...</H1>
```

## Links

The style sheet implementation also allows you to customize the appearance of links (text that the user clicks to jump to another page) by assigning two predefined classes to the <A> (anchor) tag:

- **A:link** represents a link that hasn't been visited (clicked).
- **A:visited** represents a link that the user has already clicked.

You can set any font or text formatting properties on these anchor classes, including **color**, **font-size**, **font-weight**, and **text-decoration**. For example, to assign specific colors to the three types of links, you can specify:

```
A:link {color: red}
A:visited {color: blue}
A:active {color: orange}
```

Setting **text-decoration** to "none" allows you to remove the underlining from the link text:

```
A:visited {color: blue; text-decoration: none}
```

**Notes:** Internet Explorer 3.0 does not support the A:active class, which represents a link that is currently being visited (for example, if your Web page consists of two frames—a contents frame and a viewer frame—and you click a link in the contents frame, the link will be "active" while you're viewing that article). In Internet Explorer 3.0, you'll also find that if you set a property on A:link, visited links inherit that property.

## Comments



You can add comments to your style sheet to explain your design decisions. Comments can appear on any line in the style specification, between the characters */\** and *\*/*—for example:

```
H1 {font: 20pt/22pt bold; color=#00FF00} /* Green for heading 1 */
```

## Style Reference Guide

---

This section provides a list of properties you can use in your style specifications with Internet Explorer 3.0. See the W3C "Cascading Style Sheets, Level 1" working draft for a complete list of the standard properties.

Note that Internet Explorer 3.0 does not currently support all of the attribute types and values documented in the W3C working draft. In the sections below, we've noted some of the differences in supported values.

**Note:** If you find that some of the style properties listed in this section do not work as documented, try switching the properties around. For example, if you assigned two properties, **font-weight** and **color** (in that order), to the <H1> tag and encounter problems, try switching the order.

### Quick reference

Attribute	Description	Values	Example
font-size	Sets size of text.	points (pt) inches (in) centimeters (cm) pixels (px)	{font-size: 12pt}
font-family	Sets typeface.	typeface name font family name	{font-family: courier}
font-weight	Sets thickness of type.	extra-light light demi-light medium demi-bold bold extra-bold	{font-weight: bold}
font-style	Italicizes text.	normal italic	{font-style: italic}
line-height	Sets the distance between baselines.	points (pt) inches (in) centimeters (cm) pixels (px) percentage (%)	{line-height: 24pt}
color	Sets color of text.	color-name RGB triplet	{color: blue}
text-decoration	Underlines or otherwise highlights text.	none underline italic line-through	{text-decoration: underline}
margin-left	Sets distance from left	points (pt)	{margin-left: 1in}

	edge of page.	inches (in) centimeters (cm) pixels (px)*	
margin-right	Sets distance from right edge of page.	points (pt) inches (in) centimeters (cm) pixels (px)*	{margin-right: 1in}
margin-top	Sets distance from top edge of page.	points (pt) inches (in) centimeters (cm) pixels (px)*	{margin-top: -20px}
text-align	Sets justification.	left center right	{text-align: right}
text-indent	Sets distance from left margin.	points (pt) inches (in) centimeters (cm) pixels (px)*	{text-indent: 0.5in}
background	Sets background images or colors.	URL, color-name RGB triplet	{background: #33CC00}

**Note:** Internet Explorer 3.0 (final release) supports negative values for the margin-left, margin-right, margin-top, and text-indent attributes.

## Point size

The **font-size** attribute sets the size of the text in points, inches, centimeters, or pixels. For example:

```
{font-size: 12pt}
{font-size: 1in}
{font-size: 5cm}
{font-size: 24px}
```

You can also use a percentage value, which will be evaluated based on the default point size.

## Typeface

The **font-family** attribute sets the typeface used for text. You can specify a single typeface:

```
{font-family: Arial}
```

or alternatives, separated by commas:

```
{font-family: Arial, Helvetica}
```

The line above ensures that Helvetica is used on systems that don't support Arial. Specifying a generic family name as a last alternative is good practice:

```
{font-family: Arial, Helvetica, sans-serif}
```

The generic family names (*serif*, *sans-serif*, *cursive*, *fantasy*, or *monospace*) are mapped to the fonts installed on the user's system.

If you're referencing a typeface that consists of multiple words, use quotation marks:

```
{font-family: "Courier New" }
```

## Weight

The **font-weight** attribute sets the thickness of type.

```
{font-weight: medium}  
{font-weight: bold}
```

The accepted values (*extra-light*, *light*, *demi-light*, *medium*, *demi-bold*, *bold*, and *extra-bold*) depend on the fonts available on the user's system. (For example, the user's system may only allow medium and bold for a given font.)

## Style

The **font-style** attribute sets italic text:

```
{font-style: italic}
```

(Note that the W3C working draft also references small caps, oblique, and a few other styles. Internet Explorer 3.0 currently supports only normal and italic.)

## Line height

The **line-height** attribute sets "leading" (the distance between the baselines of two lines). You can specify leading in points, inches, centimeters, or pixels. For example:

```
{line-height: 20pt}
```

You can also specify a percentage value:

```
{line-height: 150%}
```

(Note that in Internet Explorer 3.0, the spacing is added before lines, not after lines. Also, this attribute behaves inconsistently on text that uses different point sizes on the same line.)

## Grouping font attributes

The attributes listed above are ones that you will probably use frequently in your style sheets, so the W3C specifies (and Internet Explorer supports) a shortcut notation. Instead of setting the attributes separately, you can combine them into one attribute called **font**.

Thus, instead of:

```
P {font-family: Times, serif;
  font-size: 12pt;
  line-height: 20pt;
  font-weight: bold;
  font-style: italic}
```

you can simply use:

```
P {font: bold italic 12pt/20pt Times, serif}
```

**Note:** The order of attributes is significant: the font weight and style must be specified before the others.

## Colors

The **color** attribute sets the type to a named color or RGB value. You can use the 16 named colors or hexadecimal red-green-blue (RGB) color values:

```
{color: teal}
{color: #33CC00}
```

Named colors consist of the following:

black	silver	gray	white
maroon	red	purple	fuchsia
green	lime	olive	yellow
navy	blue	teal	aqua

For more information on colors, see the articles by Robert Hess on the Site Builder Workshop. The "Safety Palette Color Picker" article provides a complete color chart with hexadecimal RGB values, and "Using Named Colors in Internet Explorer" lists a much larger set of named colors you can use: the X11 named color set.

## Special text effects

The **text-decoration** attribute allows you to use underlining and strike-through for text. The supported values are underline, line-through, none, and italic.

```
{text-decoration: underline}
{text-decoration: line-through}
```

(The W3C working draft also references overline and blink, which are not currently supported by Internet Explorer.) To see an example of how you can use **text-decoration** to turn underlining off for links, see "Links" above.

## Margins

The margin-left, margin-right, and margin-top attributes set the side margins and top-margin for a tag. You can specify the margins in points, inches, centimeters, or pixels. For example:

```
BODY {margin-left: -10px;
```

```
margin-right: -10px;  
margin-top: 20px}
```

(You can use negative values for margin settings to create special effects. For examples, see the cascading style sheet demo pages created by the Microsoft Typography group.)

## Grouping margin attributes

Instead of setting the margin attributes separately, you can combine them into one attribute called **margin**. Thus, instead of the example above, you can simply use:

you can simply use:

```
P {margin: 20px -10px-10px}
```

The order is top, right, and left. If you specify a single value, it will be applied to all three margins.

## Alignment

The **text-align** attribute lets you left-justify, center, or right-justify HTML elements:

```
{text-align: left}  
{text-align: center}  
{text-align: right}
```

## Indentation

In addition to using margins, you can also set additional indentation for sections of your page using the **text-indent** attribute. Again, you can specify indentation in points, inches, centimeters, or pixels. For example:

```
H2 {text-indent: 0.5cm}
```

causes your level-2 headings to be indented 0.5 centimeters from the left margin. You can also use negative values to pull the text out into the margins.

## Background colors and images

You can use the **background** attribute to highlight sections of your page by setting a background color or background image. To set a color, you specify a named color (see the color attribute above) or use an RGB color value:

```
{background: red}  
{background: #6633FF}
```

To place an image, you specify the URL in parentheses:

```
{background: URL(http://www.microsoft.com/intdev/boxes.gif)}
```

In Internet Explorer 3.0, you can use backgrounds on any element. You can also use the *repeat*, *scroll*, and *position* parameters as described below to create special effects with background images.

**Repetitions:** The values *repeat-x*, *repeat-y*, *repeat*, and *no-repeat* determine, respectively, whether the background image is repeated horizontally, vertically, in both directions (default), or not at all.

**Scrolling:** The values *fixed* and *scroll* (default) determine whether the image is fixed on the page or scrolls with content.

**Position:** You can specify the position of the background image with respect to the element by specifying horizontal and vertical offsets. You can either use the keywords *left/center/right* and *top/middle/bottom*, percentage values, or distances in points, centimeters, inches, or pixels. For example:

```
{background: URL(http://www.microsoft.com/intdev/boxes.gif) 0% 100%}
```

or:

```
{background: URL(http://www.microsoft.com/intdev/boxes.gif) left bottom}
```

places the image at the lower left. See Cascading Style Sheets, Level 1 for additional examples of background image positioning.

## Tips & Tricks

---

### Designing for multiple browsers

If you're using embedded styles, you need to make sure that your style definitions do not get displayed as regular text in browsers (such as Netscape) that don't support style sheets. (Netscape will ignore the `<STYLE>` and `</STYLE>` tags, but will interpret the style definitions in between these as regular text, because they are not enclosed in angle brackets. To fix this, embed your style block within a comment, as follows:

```
<STYLE>
<!--
  BODY {font: 10pt "Arial"}
  H1 {font: 15pt/17pt "Arial";
     font-weight: bold;
     color: maroon}
  H2 {font: 13pt/15pt "Arial";
     font-weight: bold;
     color: blue}
  P {font: 10pt/12pt "Arial";
     color: black}
-->
</STYLE>
```

### Taking advantage of inheritance

The HTML tags on your page follow a chain of inheritance. The top-level (parent) tag is <HTML> followed by <BODY>. If you assign styles to the <BODY> tag, your tables, lists, paragraphs, and all other lower-level HTML elements will inherit from it. Thus, to set the global layout and formatting for your entire page, you can simply assign the appropriate properties to the <BODY> tag. For example:

```
BODY {font: 10pt/11pt Arial, Helvetica, sans-serif;
      background: url(clouds.gif);
      margin-left: 0.5in;
      margin-right: 0.5in}
```

sets the default font, leading, background image, and margins for your entire page. If you need to override the default for specific elements within your page (for example, you need tables indented further or your abstract paragraph in smaller type), you can add style specifications for those tags in your style sheet. The tags you haven't set in your style sheet will inherit from their parent tags or from <BODY>.

## Samples

---

- Your first stop should be the Style Sheets section of the Web Gallery, which provides a collection of style sheet samples and templates ready for you to use. We've started out with just a few examples, but the list is growing.
- The Microsoft Typography group has put together an impressive gallery of demo pages that demonstrate special effects created exclusively with text and style sheets.
- In *Style Sheets: A Brief Overview for Designers*, Nadja Vol Ochs provides several short examples of using styles for formatting and design on your Web pages.