How the Web Works, Part VI: Using Style Sheets

Agenda

- Introduction
- Examples
- Style Sheet Syntax
- Sources
- Tools

Terms

- Cascading style sheets
- Style rule
- Selector
- Declaration
- Property
- Value
- Pseudo-class

I. Introduction

A. What are style sheets and why should I use them?

Simply put, style sheets are a better way to make your Web pages pretty.

A style sheet is a collection of rules that govern the presentation of the document.

B. Where can style information go?

1. Inline

```
<P STYLE="color: red; background: white;">
Blah blah blah.
</P>
```

2. Embedded

```
<STYLE TYPE="text/css">
<!--
P {
color: red;
background: white;
}
-->
</STYLE>
```

3. External

```
<LINK REL="stylesheet" TYPE="text/css" HREF="mystyle.css">
```

Note: The LINK element shouldn't be confused with the A (anchor) element, which is used to embed links in a Web page. No, the LINK element goes in the document HEAD and is therefore invisible. This element may do all sorts of wonderful things in browsers of the future, but right now, its only useful purpose is to hook your page up to an external style sheet.

Netscape Composer tips

Don't confuse Composer's Format > Style submenu with style sheets -- there's no connection. Composer just plain doesn't understand style sheets yet.

You can easily link to an external style sheet by editing the HTML source code, if you have specified an external editor in your Composer preferences. Just choose Edit > HTML Source and type the LINK element in manually.

Microsoft FrontPage tips

You can link to a style sheet by choosing Format > Style Sheet Links. FrontPage can also generate style sheets.

II. Example

```
BODY {
background-color: black;
padding: 10px;
}

DIV.NARROW {
background-color: tan;
font-size: .95em;
border: none;
padding: 20px;
font-family: "Bookman Old Style", Times, serif;
width: 400px;
}

P {
margin-left: 4em;
margin-right: 2em;
}

ADDRESS {
font-size: small;
font-style: normal;
color: purple;
text-align: right;
}

PRE, BLOCKQUOTE {
line-height: .9em;
margin-left: 6em;
margin-right: 3em;
text-align: left;
color: red;
}

H1, H2, H3, H4 {
color: brown;
font-family: Arial, Helvetica, sans-serif;
}

H1 { font-size: 2em }
H2 { font-size: 1.8em }
H3 { font-size: 1.6em }
H4 { font-size: 1.4em }

A { text-decoration: none }
A:LINK { font-weight: bold; color: red; }
A:VISITED { font-weight: normal; color: brown; }
```

III. Style Sheet Syntax

A. Basic syntax

Each style sheet *rule* has two parts: a *selector*, and a *declaration*.

```
SELECTOR { declaration }
```

The selector indicates an HTML element, while the declaration specifies the actual style to be assigned that element. Declarations take the form of *property-value* pairs. Here's a simple example:

```
P { color: red }
```

Rules that are more complex can be built by combining multiple declarations, which must be separated with semicolons:

```
SELECTOR {
property: value;
property: value;
property: value;
}
```

Furthermore, selectors can target HTML elements by *class*:

```
SELECTOR.CLASS
```

Here's a simple example:

```
P.warning { color: red }
```

This will affect any paragraph element with a class of "warning."
Example: `<P CLASS="warning">`

To target all elements of this class, whether they are paragraphs or otherwise, omit the first part of the selector:

```
.warning { color: red }
```

In addition to genuine classes , there are a few *pseudo-classes*, which apply only to the anchor element. These can be used to set the color of links and are easy to understand:

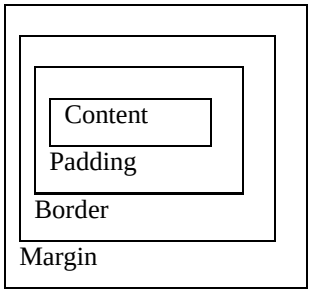
```
A:link { color: red }
A:visited { color: blue }
A:active { color: orange }
```

Note that a colon is used instead of a period.

B. Properties

1. *Box Properties*

HTML elements can be thought of as boxes, constructed as indicated by the following diagram:



`margin`

Value: a length or a percentage

The 'margin' property sets all four margins of an element at the same time.

```
BODY { margin: 2em }
```

Because of bugs in Navigator, it's best to avoid setting the margin of inline elements.

`padding`

Value: a length or a percentage

The 'padding' property sets all four paddings of an element at the same time.

```
H1 { padding: 1em }
```

Because of bugs in Navigator, it's best to avoid setting the padding of inline elements.

`margin-left, margin-right, margin-top, margin-bottom, padding-left, padding-right, padding-top, padding-bottom`

Value: a length or a percentage

Use these to individually set the margins or paddings of an element:

```
BODY { margin-left: 2em }
```

`border-color`

Value: a color

The 'border-color' property sets the color of the border. If no color value is specified, the value of the 'color' property of the element itself will take its place:

```
P {
color: black;
background: white;
border: solid;
}
```

`border-style`

Value: none | solid | double | groove | ridge | inset | outset

The 'border-style' property sets the style of the four borders.

```
H1 { border-style: solid }
```

`border-width`

Value: thin | medium | thick | <length>

The 'border-width' property sets the width of the four borders.

```
H1 { border-width: 5px }
```

`width`

Value: a length or percentage

Sets the width of the element.

```
DIV { width: 100px }
```

Caveat: don't use this property for any page that you want users to be able to print, as it messes up the printing process for the Windows version of Navigator.

2. Color and Background Properties

`color`

This property describes the text color of an element (often referred to as the *foreground* color). There are different ways to specify red:

```
EM { color: red }
EM { color: rgb(255,0,0) }
```

`background-color`

This property sets the background color of an element. This is straightforward enogh when setting the background color of the BODY, but there is a Netscape bug that crops up when setting the background color of any element within the body. It's usually most notable in paragraphs -- the background color is applied only to the text and not to the entire box. Workaround: set the border to "none".

```
P { background-color: yellow; border: none; }
```

`background-image`

This property sets the background image of an element. When setting a background image, one should also set a background color that will be used when the image is unavailable. When the image is available, it is overlaid on top of the background color.

```
BODY { background-image: url(marble.gif) }
P { background-image: none }
```

3. *Text Properties*

`text-align`

Value: left | right | center | justify

This property describes how text is aligned within the element:

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

All block-level elements inside the 'DIV' element with 'CLASS=bingo' will be centered.

`text-decoration`

Value: none | underline | overline | line-through

This property describes decorations that are added to the text of an element:

```
A:link, A:visited, A:active { text-decoration: underline }
```

The example above would underline the text of all links (i.e., all 'A' elements with a 'HREF' attribute).

`text-indent`

Value: a length or percentage

The property specifies the indentation that appears before the first formatted line. The value of 'text-indent' may be negative, but there may be implementation-specific limits.

Example:

```
P { text-indent: 3em }
```

`text-transform`

Value: capitalize | uppercase | lowercase | none

Changes the case of the selected element:

```
H1 { text-transform: uppercase }
```

The example above would put 'H1' elements in uppercase text.

`line-height`

Value: normal | <number> | <length> | <percentage>

The property sets the distance between lines of text.

When a numerical value is specified, the line height is given by the font size of the current element multiplied with the numerical value.

The three rules in the example below have the same resultant line height:

```
DIV { line-height: 1.2; font-size: 10pt }
DIV { line-height: 1.2em; font-size: 10pt }
DIV { line-height: 120%; font-size: 10pt }
```

A value of 'normal' sets the 'line-height' to a reasonable value for the element's font.

4. *Font Properties*

`font-family`

Value: a prioritized list of specific or generic font-family names

```
BODY { font-family: "new century schoolbook", times, serif }
```

Separate the font-family names with commas. Font-family names with spaces must be enclosed in quotes. You should offer a generic font family as a last alternative. Generic font-families include the following:

- 'serif' (e.g. Times)
- 'sans-serif' (e.g. Helvetica)
- 'monospace' (e.g. Courier)

`font-size`

Value: an absolute or relative size, or a length or percentage value

Absolute size keywords include:
xx-small | x-small | small | medium | large | x-large | xx-large

Relative size keyword include: larger | smaller

Length and percentage values should not take the font size table into account when calculating the font size of the element.

On all other properties, 'em' and 'ex' length values refer to the font size of the current element. On the 'font-size' property, these length units refer to the font size of the parent element.

Examples:

```
P { font-size: 12pt; }
BLOCKQUOTE { font-size: larger }
EM { font-size: 150% }
EM { font-size: 1.5em }
```

`font-style`

Value: normal | italic | oblique

The 'font-style' property selects between normal (sometimes referred to as "roman" or "upright"), italic and oblique faces within a font family.

```
H1, H2, H3 { font-style: italic }
```

`font-weight`

Value: normal | bold | bolder | lighter | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900

The 'font-weight' property selects the weight of the font. The values '100' to '900' form an ordered sequence, where each number indicates a weight that is at least as dark as its predecessor. The keyword 'normal' is synonymous with '400', and 'bold' is synonymous with '700'.

```
P { font-weight: normal }
H1 { font-weight: 700 }
```

5. Classification Properties

`display`

A value of 'none' turns off the display of the element.

```
IMG { display: none }
```

`list-style-type`

Value: disc | circle | square | decimal | lower-roman | upper-roman | lower-alpha | upper-alpha

This property is used to determine the appearance of the list-item marker. It only works for list elements, such as ordered or unordered lists:

```
OL { list-style-type: decimal }
```

C. Units

1. Length Units

The format of a length value is a number immediately followed by a two-letter unit identifier.

Style sheets that use relative units will more easily scale from one medium to another (e.g. from a computer display to a laser printer). Percentage units (described below) and keyword values (e.g. 'x-large') offer similar advantages.

These relative units are supported:

- `H1 { margin: 0.5em }` ems, the height of the element's font
- `H1 { margin: 1ex }` x-height, ~ the height of the letter 'x'
- `P { font-size: 12px }` pixels, relative to canvas

Absolute length units are only useful when the physical properties of the output medium are known. It's probably best to avoid these unless you really know what you're doing:

- `H4 { font-size: 12pt }` points, 1pt = 1/72 in */

2. Percentage Units

The format of a percentage value is a number immediately followed by '%'.

Percentage values are always relative to another value, for example a length unit. Each property that allows percentage units also defines what value the percentage value refer to. Most often this is the font size of the element itself:

- `P { line-height: 120% }` 120% of the element's 'font-size'

3. Color Units

A color is a either a keyword or a numerical RGB specification.

The suggested list of keyword color names is: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, and yellow. These 16 colors are taken from the Windows VGA palette, and their RGB values are not defined in this specification.

- `BODY {color: black; background: white }`

The RGB color model specifies values for Red, Green and Blue color channels. These examples all specify the same color:

- `EM { color: #f00 }` #rgb
- `EM { color: #ff0000 }` #rrggbb
- `EM { color: rgb(255,0,0) }` integer range 0 - 255
- `EM { color: rgb(100%, 0%, 0%)` float range 0.0% - 100.0%

IV. Sources

This Workshop's On-line Materials

http://www.xula.edu/Administrative/cat/workshops/style/

Guide to Cascading Style Sheets by the Web Design Group

http://www.htmlhelp.com/reference/css/

An introduction to style sheets, a quick tutorial, a reference -- everything you need is here.

The World Wide Web Consortium

http://www.w3.org/Style/CSS

For the true inside dope, read the W3C's official recommendation. There are also plenty of links to other resources for learning about CSS.

Cascading Style Sheets: the Definitive Guide

http://www.oreilly.com/catalog/css/

Everything indicates that this will be the book to buy when it hits the shelves in April.

V. Tools

TopStyle

http://www.bradsoft.com/topstyle/

A handy tool for authoring style sheets. Sorry, this is for Windows only. There is a free version.

CSSCheck

http://www.htmlhelp.com/tools/csscheck/

Use this service to validate your style sheet. You can upload files from your computer or enter the Web address of an on-line document.