

CSSS

Peter Cho | 161A

Notes from Jennifer Niederst: Web Design in a Nutshell
Thomas A. Powell: HTML & XHTML, Fourth Edition
and <http://www.w3schools.com>

Based on a tutorials by Prof. Daniel Sauter / Prof. Casey Reas

What is CSS?

CSS = Cascading Style Sheets

Official W3C standard for controlling presentation, added to HTML 4.0

Styles define how to display HTML elements

Style sheets rely on underlying markup structure

Greater typography and page layout control

Style is separate from structure

Potentially smaller documents

Allow for easier site maintenance

Multiple style definitions “cascade” into one

First example

```
<html>
<head>
<title>CSS Example</title>
<style type="text/css">
  h1 { font-size: 32pt;
        font-family: Helvetica,Arial;
        color: gray;
        margin: 30px; }
  p { font-size: 12pt;
        font-family: Helvetica,Arial;
        font-style: italic;
        line-height: 1.7;
        margin-left: 30px;
        width: 400px; }
</style>
</head>
<body>
<h1>New and improved markup using CSS Styles</h1>
<p>CSS provides a powerful set of properties for
manipulating the look of HTML elements</p>
</body>
</html>
```

Adding CSS to a document

1. Inline styles on one HTML tag

```
<h1 style="color: red">Red Herring</h1>
```

2. Internal style sheets

```
<head>  
<style type="text/css">  
<!--  
h1 { color: red; font-size: 36pt; }  
-->  
</style>  
</head>
```

3. Linking to External Style Sheets

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

Cascading order

Styles will “cascade” into a new “virtual” style sheet by the following rules, in order from highest to lowest priority:

1. Inline style (inside an HTML element)
2. Internal style sheet (inside the `<head>` tag)
3. External style sheet
4. Browser default

CSS Syntax

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

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

Apply styles to HTML elements/tags

```
H1 { color: blue }  
p { font-family: "sans serif"; font-size: 10pt; }  
body { background-color:#FFFFFFCC;  
       margin-left:100px; }
```

Selectors can be grouped together

```
H1, H2, P  
{  
  font-style: bold;  
  color: red  
}
```

class selectors

In the stylesheet:

```
p.important { color: red }
```

```
p.caption { font-size:12pt; line-height:14pt;  
font-family:Helvetica,Arial,sans-serif; }
```

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

In the HTML:

```
<p class="important">Attention!</p>
```

```
<p class="caption">The class' name doesn't  
matter! </p>
```

```
<h1 class="right">Right-aligned headline</h1>
```

```
<p class="right">Right-aligned paragraph</p>
```

id selectors

In the stylesheet:

```
#st01 { color: green t}  
#firstheading {color: magenta; font-weight: bold; }
```

In the HTML:

```
<p id="st01">Attention!</p>  
<p id="firstheading">Attention!</p>
```

The ID must be unique and should be used only once in the respective document

Color

```
<style type="text/css">
h1 { color:#FFCCDD; background-color:black; }
body { background-color:rgb(51,0,102); }
blockquote { background-color:rgb(60%,90%,75%); }
</style>
```

These definitions are equivalent:

```
color:white;
color:rgb(255, 255, 255);
color:#FFFFFF;
```

Background

```
<style type="text/css">
body
{
background-image:url('imgs/gradient2.gif');
background-repeat: repeat-x;
}
</style>
```

Background options

background-color	#00FF00
background-image	url('stars.gif');
background-attachment	scroll fixed
background-position	x-pos y-pos
background-repeat	repeat repeat-x repeat-y no-repeat

Type properties

font-family	p {font-family: Verdana,sans-serif}
font-style	h1 {font-style: italic}
font-weight	h2 {font-weight: bold}
font-size	p {font-size: 13px}
font	h1 {font: 12pt Times, serif}
word-spacing	h3 {word-spacing: 3px}
letter-spacing	.subhead {letter-spacing: 2pt}
text-decoration	a:link {text-decoration: underline}
text-transform	h1 {text-transform: uppercase}
text-align	p {text-align: justify}
text-indent	p {text-indent: 10px}
color	.caption {color: #6644EE}
line-height	p {line-height: 14pt}

Type properties

In the stylesheet:

```
.fancy {  
  font-family:'Times New Roman',Times,serif;  
  text-decoration:underline;  
  font-style:italic;  
  font-variant:small-caps;  
  font-size:10pt;  
  font-weight:bold;  
  word-spacing:8pt;  
  letter-spacing: 1pt;  
  color:blue; }
```

in the HTML:

```
<p class="fancy">  
CSS allows extended type control.  
</p>
```

Pseudoclasses (anchor styles)

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

```
<style type="text/css">
a { font-weight:bold; }
a:link {
  color:#0000E0;
  text-decoration:none }
a:visited {
  color:#000080;
  text-decoration:none }
a:hover {
  color:#FFFFFF;
  background-color:#E00000;
  text-decoration:underline }
a:active {
  color:#FFFFFF;
  background-color:#E00000;
  text-decoration:underline }
a:focus {
  color:#FFFF00;
  text-decoration:none }
</style>
```

Length units

p t = point (absolute). Typographical unit. 1 point equals 1/72 inches.

p c = pica (absolute). Typographical unit. 1 pica is equivalent to 12 point.

i n = inch (absolute). 1 inch is equivalent to 2.54 centimeters.

m m = millimeters (absolute).

c m = centimeters (absolute).

p x = pixel (absolute/relative). Depends on the output device.

e m = em (relative). 1 em in a 24pt size type is 24pt.

e x = ex (relative). Refers to height of the small character x.

% = percent (relative). Can be relative to the size of the element itself, to the size of the parent element, etc.

Do not leave spaces between the property value and the units!

If you use “`margin-left: 20 px`” instead of “`margin-left: 20px`”

it will not work in Firefox/Netscape.

<div> and

Generic HTML tags

<div> = block element

 = inline element

Useful for applying styles

```
<div style="background-color: #DDDDDD; font-weight: bold;
color: black; padding: 20pt;">
<p>Style sheets separate the structure of a document from its
presentation.</p>
</div>
```

```
<p>Calling out <span style="background-color: yellow; color:
black;">special sections of text</span> by using span</p>
```

Box properties

width

height

margin

margin-top, -bottom, -left, -right

border

border-top, -bottom, -left, -right

border-style

border-width

border-color

padding

padding-top, -bottom, -left, -right

background

background-color

background-image

background-repeat

Positioning with CSS

`position` = position type

`top` = start position from the top

`left` = start position from the left

`bottom` = start position from the bottom

`right` = start position from the right

`width` = width

`height` = height

`overflow` = content exceeds the size of the element

`direction` = content direction

`float` = text flow around elements

`z-index` = layer position of overlapping areas

`display` = display behavior without reserved space

`visibility` = display behavior with reserved space

`clip` = limit displayed area

Absolute vs. Relative

absolute: An element can be placed anywhere on a page. The element's position is specified with the "left", "top", "right", and "bottom" properties.

```
.abs
{
  position: absolute;
  top: 140px;
  left: 150px;
  width: 230px;
}
```

relative: Moves an element relative to its normal position, so "left:20px" adds 20 pixels to the element's current "left" position.

```
.rel
{
  position: relative;
  left: 30px;
  width: 230px;
}
```

External/Internal styles

External CSS

page.htm

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

mystyles.css

```
hr {color: #4400CC}  
p {margin-left: 20px}  
body {background-image:  
url("images/back40.gif")}
```

Internal CSS

page.htm

```
...  
<head>  
<style type="text/css">  
hr {color: #4400CC}  
p {margin-left: 20px}  
body {background-image: url("images/back40.gif")}  
</style>  
</head>  
...
```

Using CSS for different output media types

```
<html><head>  
<link rel="stylesheet" media="screen" href="website.css">  
<link rel="stylesheet" media="print" href="print.css">  
<link rel="stylesheet" media="aural"  
href="speaker.css">  
<link rel="stylesheet" media="braille"  
href="braille.css">  
</head>  
<body></body></html>
```

Each of these CSS files must contain only the format definitions for the respective output medium