Formatting Web Pages: Cascading Style Sheets

Dr Enrico Gerding

eg@ecs.soton.ac.uk
Overview

• What are cascading style sheets
• Basic elements
• Types of style sheets
Introduction

• Formatting of text used to be done like this:

<font size="7" color="red" face="verdana"><p>I am big red!</p></font>

<h1 align="right">text</h1>

These and similar elements have been deprecated and replaced by a single style attribute!
Example

Inline style:

```html
<p style="font-size: xx-large; color:red; font-family:Verdana"> Hello</p>
```

Style value can contain multiple properties:

```html
property-name1: value1; property-name2: value2; etc...
```

Properties have several types of values

- Predefined keywords, percentage values, Specific lengths measurements, colour values, Integers and URLs.
Examples

<p style="font-size:24px;">Hello</p>
<p style="color:#ee0;">Hello</p>
<p style="font-family:'Times New Roman'">Hello</p>

• Note use of quotes
  – Combination of single and double
Activity

Why is having styles inline a good or bad thing to do?
Adding styles: Inline

• All perfectly acceptable
  – Does pollute the document with presentation information
  – Tied to a single tag,
  – if changes required every one of them must be found.
  – If you must, use it to override higher level rule.
Using Style Sheets

• Often you want to apply a style to all element in the body.
  – All headings to be the same
  – All paragraphs to be formatted the same
  – Etc.

This is where style can be used.
Using Style Sheets

• Styles can be embedded into a page to format the elements in a page
• Alternatively they can be used by a set of pages by writing them into style sheet and referring to this file in the header.
• also allows the same page to be presented in different styles for different rendering methods,
  – Printing, screen reader
  – A number of rules can be applied to the same tag, the rule cascade, CSS specifies how to decide on which one to apply.
Advantages

• Greater typographical and page layout control
• Style is separate from contents
• Less Work
  – Not only for a page but an entire site
• Consistency across web pages
• Easier to maintain
• Potentially smaller documents
• Well supported
• Way of the future
  – style tag is no longer in latest XHTML specification.
Other ways of inserting styles

- Embedded
- External

Rules are specified which *bind* formatting to specific elements, classes, and/or IDs
• Style sheets are composed of "Rules" that describe the styling to be applied.

• Each rule contains a Selector and a Declaration.
Rule Syntax

selector  {property1: value1;
property2: value2; ...}

Or similarly:

selector  {
  property1: value1;
  property2: value2;
  ...
}


Examples

h1 { color: #eee; }
p {
    font-size: 12px;
    font-family: Verdana;
}
body { background-color: green; }
### CSS Syntax for Color Values

**TABLE 4.2 Syntax to configure a paragraph with red text**

<table>
<thead>
<tr>
<th>CSS Syntax</th>
<th>Color Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>p { color: red }</td>
<td>Color name</td>
</tr>
<tr>
<td>p { color: #FF0000 }</td>
<td>Hexadecimal color value</td>
</tr>
<tr>
<td>p { color: #F00 }</td>
<td>Shorthand hexadecimal (one character for each hexadecimal pair – only used with web safe colors)</td>
</tr>
<tr>
<td>p { color: rgb(255,0,0) }</td>
<td>Decimal color value (RGB triplet)</td>
</tr>
<tr>
<td>p { color: rgba(255,0,0,0.5) }</td>
<td>CSS3: Decimal color value (RGB triplet) followed by the alpha opacity (a value from 0 to 1). <em>The CSS3 Color Module is in draft status and is not yet uniformly supported by browsers.</em></td>
</tr>
<tr>
<td>p { color: hsl(0, 100%, 50%) }</td>
<td>HSL color values. <em>The CSS3 Color Module is in draft status and is not yet uniformly supported by browsers.</em> See <a href="http://www.w3.org/TR/css3-color/#hsl-color">http://www.w3.org/TR/css3-color/#hsl-color</a></td>
</tr>
</tbody>
</table>
Adding styles: Embedded

Embed a style block in the document header:
<style type="text/css">........</style>

Example:
<head>
  <style type="text/css">
    h1 {color: #666; text-align: center}
    p {font-size: 150%; font-family: Arial; background-color: green}
  </style>
</head>
Adding styles: External

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

<link> tag specifies a relationship between document and external resource

Alternative:
<head>
  <style type = "text/css">
    @import url(http://pathname/stylesheet.css);
  </style>
</head>
External

- Files should have extension .css
- .css files do not contain the style tag!
Combinations

Can also combine external, embedded, and inline styles

Example:

```html
<style type = "text/css">
    @import url (http://pathname/stylesheet.css);
    p {font-family: Verdana;}
</style>
```

• External style must come before embedded (otherwise overridden)
Adding styles: Embedded

- More than one style sheet can be applied to your document
- Browsers have built in style sheets.
- Individual may apply their own style sheets, so personalise their visit to ours and many other sites.
Inheritance Rules

• Browser default
• Imported
  – If multiple the last in the list has priority.
• Embedded
  – Those applied to a specific document override imported ones
• Inline
  – This overrides all the above

→ That’s why it’s called cascading
Other things to use with CSS

- As well as applying style sheets to tags we can also apply CSS to other elements on in the web page.
Id and class

• Id give an element a specific name
  – Must be unique in the document
  – Can occur in other documents on the site

• Class used to group similar elements.
  – Allows style sheet rules to be applied to all those elements of that class
  – All “telephone” elements can be formatted the same way

• All elements can be assigned an id and/or class
Examples: id

- **CSS code:**
  ```
  #example1  { background-color: white; } 
  #example2  { text-transform: uppercase; } 
  ```

- **HTML usage:**
  ```html
  <p id="example1">This paragraph has an ID name of "example1" and has a white CSS defined background</p>
  <p id="example2">This paragraph has an ID name of "example2" and has had its text transformed to uppercase letters. </p>
  ```
Examples: class

• CSS code:
  
  .example1 { background-color: white; }  
  .example2 { text-transform: uppercase; }  

• HTML usage:
  
  <p class="example1">This paragraph has a class name of "example1" and has a white CSS defined background</p>  
  
  <p class="example2">This paragraph has a class name of "example2" and has had its text transformed to uppercase letters.  
  </p>
Tags and classes

• Often, you want to treat different tags of the same class differently
  – For example: headers are always green, paragraphs always red

p.example1 {color: green}
h1.example1 {color: red}
Groupings

• Group a number of selectors to give the same style.

```html
h1, h2, h3, h4, h5, h6 {
  color: green
}
```
Combining Classes

• Apply styles to combination of classes

```css
h1.veryimportant {background-color:red;}
h1.veryimportant.stuff {background-color: green;}

<h1 class="veryimportant">This is red</h1>
<h1 class="veryimportant dummy">also red</h1>
<h1 class="veryimportant stuff">This is green</h1>
<h1 class="veryimportant dummy stuff">This is also green</h1>
```
Contextual Selection

• Set style of tags within other tags
  – For example: strong text within <p></p> can be displayed differently from strong text within <h1></h1>

• CSS Usage:
  
  p strong {color: yellow;}

• HTML example:

  <p>The next word is
    <strong>yellow</strong></p>

  <h1>But this word is
    <strong>not</strong></h1>
Pseudo-elements

Examples:

ul:first-child
p:first-letter
p:first-line
a:hover
a:visited
....
Exercise

<style type="text/css">
  h1 {color:red;}
  #s1 {color:yellow;}
  .s2 {color:orange;}
  h1.s2 {color:blue;}
</style>

<h1 class="s2">Colour?</h1>
<p class="s2">Colour?</p>
<h1>Colour?</h1>
<h1 id="s1" class="s2">Colour?</h1>

How do I get an orange paragraph with box?
Other tags used often with CSS

```html
<div> ......</div>
```

• Denotes a generic division within a document
  – Used to add a customised block element into a document.
  – Starts on a new line
  – The content within the div element is typically given a `class` or `id` attribute and then formed in a style sheet.

```html
<div id="sidebar"> content of Sidebar...</div>
```
Span

- Similar to `div` but no new line
  - used mainly as an inline “hook” to apply rules to (for instance CSS)
  
  Enrico: `<span class= "telephone" > 02380599201 </span>`

- Allows all “telephone” elements to be formatted correctly
More on inheritance

• Inheritance is why styles are called cascading
• What happens when you combine different rules for different tags

Example:

div {color:green; background-color:yellow}
p {color:red}

<div><p>Hello World</p></div>
General Inheritance Rule

• The more specific the rule, the more powerful
• The closer the rule is to the tag, the more powerful

Gets complicated when rule is closer to tag but less specific or vice versa – better to avoid
Examples and tools

- **Free css templates** large number of free templates
  http://www.freecsstemplates.org/css-templates/

- **CSS beauty** a site full of ideas of what can be done with CSS
  http://www.cssbeauty.com/

- **Web safe colours**
  http://cloford.com/resources/colours/websafe1.htm

- **CSS validation**
  http://jigsaw.w3.org/css-validator/
Summary

• Style sheets
  – separate content and presentation
  – add consistency to you document/site
  – easier to maintain

• A set of rules determine which style apply in the event of a conflict.
Recommended reading

• Learning Web Design (4th Edition), Chapters 11,12,13 or Felke-Morris, T., Basics of Web Design, Chapter 3

• Full CSS tutorials
  http://www.w3schools.com/css/default.asp