Web Site Design

Stanford University Continuing Studies CS 21

Mark Branom
branom@alumni.stanford.edu
http://web.stanford.edu/people/markb/

Course Web Site: http://web.stanford.edu/group/csp/cs21/

Week 3 Agenda

- Using Cascading Style Sheets (CSS) – The Basics

Cascading Style Sheets

- Established and maintained by the World Wide Web Consortium (W3C) and the WHATWG (Web Hypertext Application Technology Working Group).
- Controls the look, or *style*, of web pages or other XML files
- Provides a set of rules, or *sheet*, where you can define how certain HTML (Hyper-Text Markup Language) or XML (eXtensible Markup Language) tags are going to be interpreted by the browser.
- Called “cascading” because the browser interprets these styles in an aggregating, cascading fashion (e.g, if one rule tells a tag to be blue, and a later rule tells it to be italics, the tag will be both blue and italics).
- If rules are in conflict with one another, the last rule listed generally wins (e.g., if one rule tells a tag to be green, and a later rule tells the tag to be purple, the tag will be purple).

CSS Basics

- Using just HTML, to create a web page with `<h2>` tags that have the standard features of a Header tag (that is, their own paragraph, bold, with a size change) and also are dark blue, you would have to code each one as follows:

```
<h2>
   <font color="darkblue">This is a darkblue H2 tag</font>
</h2>
```

- That's a lot of information to type every time you want to use a dark blue `<h2>` tag. Using CSS, all you need to do is use a regular `<h2>` tag. The style information could then be included in the Style Sheet (and the resultant web page) as follows:

```
CSS File  -- h2 { color: darkblue;}
HTML File -- <h2>This is a darkblue H2 tag</h2>
```

CSS Rules

- To change the color of ALL `<h2>` tags from darkblue to green, simply change darkblue to green in the CSS. The next time anyone sees the site, all the `<h2>` tags on all the pages will be green instead of darkblue.
- These styles are called *rules*. Each rule consists of a *selector* and a *declaration* (which is made up of a *property* and a *value)*.
- In the example below, *h2* is the selector, *color* is the property, and *darkblue* is the value. Generally, selectors are usually HTML elements (tags), classes, or IDs (more on classes and IDs in a few slides).

```
h2 { color: darkblue;}
```

- Syntax for a CSS rule:

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

Where do you put the styles?

- Style information can be located in three places:
  - External to the pages in a site
  - Internal to each page
  - Inline with individual tags

- Generally, creating an external style sheet file is the preferred method. To take full advantage of CSS, the Style Sheet for a site should be in this one external file, so that any changes will apply throughout the site. This also means that only one style document has to be downloaded for a single site.

Style Location: External

- The most common place to put style information is in an external document that each page of a web site points to directly.
- Any changes made to this single document will then be applied throughout the entire web site as each page is accessed by users.
- External Style Sheets have a .css extension.
- When linking to an external style sheet, you can also specify separate style sheets by media type:

all - Suitable for all devices. **aural** - Intended for speech synthesizers. **braille** - Intended for braille tactile feedback devices. **embossed** - Intended for paged braille printers. **handheld** - Intended for handheld devices (typically small screen, monochrome, limited bandwidth). **print** - Intended for paged, opaque material and for pages viewed on screen in print preview mode.	**projection** - Intended for projected presentations **screen** - Intended primarily for color computer screens. **tty** - Intended for media using a fixed-pitch character grid, such as teletypes, terminals, or portable devices with limited display capabilities. **tv** - Intended for television-type devices

External example

Text that appears in the *basic.css* style sheet document:

```
h2 {font-family: Arial, sans-serif; font-style: italic; color: green;}

p {font-family: Courier, monotype; font-style: bold; color: red; }
```

Text that appears in the *print.css* style sheet document:

```
h2 {font-family: Book Antiqua, Times, serif; font-style: italic; }

p {font-family: Courier, monotype; font-style: bold; }
```

HTML document, using the <link> tag method

```
<head>

<link rel="stylesheet" type="text/css"
    href="basic.css" media="all" />

<link rel="stylesheet" type="text/css"
    href="print.css" media="print" />

</head>
```

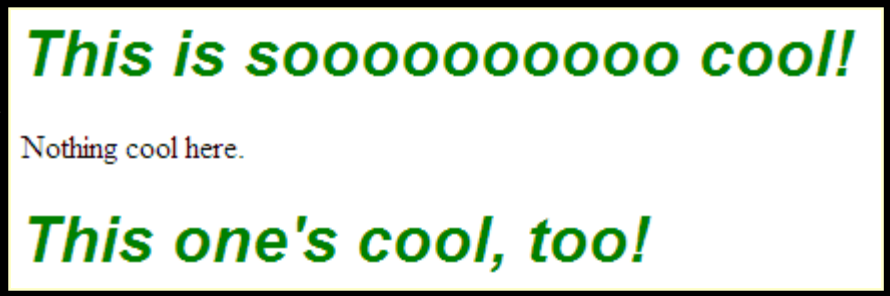
HTML document, using the @import method

```
<head>
<style type="text/css">
<!--
@import url("basic.css") all;
@import url("print.css") print;
-->
</style>
</head>
```

Style Location: Internal

- Style information can also be included in the <head> section of an individual web page. This tends to work best when a single page needs to have a slightly different look than the rest of the site.

```
<head>
<style type="text/css">
h1 { font: italic Arial;
color: green;}
</style>
</head>
<body>
<h1>This is sooooooooo cool!</h1>
<p>Nothing cool here.</p>
<h1>This one's cool, too!</h1>
</body>
```



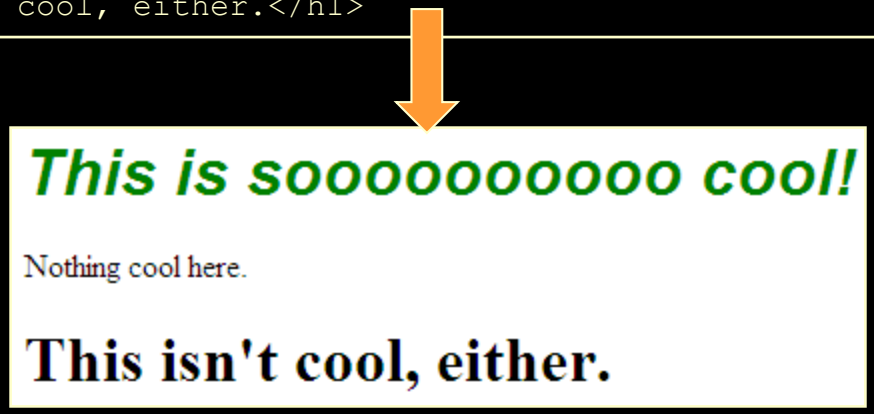
Style Location: Inline

- For extreme control, style information can be included in an individual tag. The style effects only that tag and no others in the document. This option is most useful for those rare occasions when a single tag needs to have a slightly different style, or if you are creating an HTML email newsletter.

```
<h1 style="font: italic Arial; color: green;">This is soooooooooo cool!</h1>

<p>Nothing cool here.</p>

<h1>This isn't cool, either.</h1>
```



Font and Text Styling

When choosing a font, there are several things to keep in mind:

1. Not everyone has the same set of fonts.
2. If you use a font that the visitor doesn't have, the page will display in the default font (usually Times), unless you provide more choices. To do this, add more than one font in your declaration, and always end with the font family (serif, sans-serif, or monospace):

```
font-family: Verdana, Arial, Helvetica, sans-serif
```

3. Documents designed to be printed tend to look better in Serif fonts (Times, Georgia, Book Antiqua, etc.)
4. Documents designed to be viewed onscreen tend to look better in Sans-serif fonts (Verdana, Arial, Helvetica, etc.)

To apply a font to the entire web page, modify the body tag:

```
body {font-family: Verdana;}
```

Classes and IDs: Classes

- HTML has two attributes that make CSS even more useful: *class* and *ID*. They make it easy to apply style to just about any tag.
- Classes can describe a generic style that can be applied to any HTML element, or can be created for specific elements.
- When defining a style for elements with a particular class attribute in the Style Sheet, declare a rule using a dot (.) followed by the class name. To limit the style to a particular element with that class attribute, use a selector combining the tag name with a dot followed immediately by the class name.
  - The following rule would apply to any element with the attribute **class="shade"**

    ```
    .shade { background: yellow; }
    ```
  - The following rule would apply only to paragraph tags with the class **shade** (`<p class="shade">`)

    ```
    p.shade { background: red; }
    ```

Classes and IDs: IDs

- IDs are similar to classes, but IDs are unique – they can only be used with one instance of an element within a document.
- When defining a CSS rule using an ID-based selector, use a number/pound/hash sign (#) followed by the style name. To limit the style to a particular element with that id attribute, use a selector combining the tag name with a # and then the id name.
  - The following rule would apply to any element with the attribute **id="intro"**

```
#intro { font-size: 2em; }
```

  - The following rule would apply only to heading 1 tags with the id **intro** (`<h1 id="intro">`)

```
h1#intro { color: green; }
```

Class example

- Here's an example of a web page with an internal CSS style with a class called "highlight":

```
<head>
<style type="text/css">
.highlight { background-color: #ccc;}
</style>
</head>
<body>
<p>This is paragraph is not highlit.</p>
<p class="highlight">This paragraph is highlit.</p>
</body>
```

This paragraph is not highlit.

This paragraph is highlit.

Span and Div

- There are two tags that are particularly useful when using CSS: `<span>` and `<div>`. They are both container tags that have minimal formatting associated with them.
- The `<span>` tag is an inline element that simply holds text without doing anything special to it.
- The `<div>` tag is a block element and causes the text it encloses to start on a new line.
- Using `<span>` and `<div>` tags in conjunction with classes and IDs allows for great flexibility in creating pages.
- For example, to apply a font to a specific section of text, create a class, and use the span tag with that class:

```
.neatstuff {font-family: 'Comic Sans MS';}
<span class="neatstuff">This is in Comic Sans</span>
```

Example using SPAN, DIV, Class, and ID

```
<head>
<title>CSS Example</title>
<style type="text/css">
<!--
#topic1 { background: black; color: white;}
#topic2 { background: #f00; color: yellow; margin-top: 40px;}
.important { background: #0F0; border: thin dotted #999; padding: 5px; color:
black;}
-->
</style>
</head>
<body>
<div id="topic1">This is topic number 1</div>
<div id="topic2">
   This is the <span class="important">most important</span> topic
</div>
</body>
```