

CSE417: WEB ENGINEERING

Daffodil International University

You Will Be Able To

- ✓ Understand and Apply CSS

Contents

- Basics of *Cascading Style Sheets*
 - Inline Style
 - Document Style Sheets
 - External Style Sheets

Content vs. Presentation

- most HTML tags define content type, independent of presentation
 - exceptions?
- style sheets associate presentation formats with HTML elements
 - CSS1: developed in 1996 by W3C
 - CSS2: released in 1998
 - CSS2.1: released in 2011, Only IE8 supports fully
 - CSS3: modularized release started 2011, Not fully supported by browsers
- HTML style sheets are known as *Cascading Style Sheets*, since can be defined at three different levels
 1. *inline* style sheets apply to the content of a single HTML element
 2. *document* style sheets apply to the whole BODY of a document
 3. *external* style sheets can be linked and applied to numerous documents

lower-level style sheets can override higher-level style sheets

Inline Style Sheets

```
<html>
<head>
  <title>Inline Style Sheets</title>
</head>

<body>
  <p style="font-family:Arial,sans-serif;
    text-align:right">This is a
    right-justified paragraph in a sans serif
    font (preferably Arial), with some
    <span style="color:green">green text</span>.
  </p>

  <p>And <a style="color:red;
    text-decoration:none;
    font-size:larger;"
    href="page01.html">here</a>
    is a formatted link.
  </p>
</body>
</html>
```

Using the `style` attribute, you can specify presentation style for a single HTML element

- within tag, list sequence of `property:value` pairs

`font-family:Courier,monospace`

`font-style:italic`

`font-weight:bold`

`font-size:12pt font-size:large font-size:larger`

`color:red color:#000080`

`background-color:white`

`text-decoration:underline`

`text-decoration:none`

`text-align:left text-align:center`

`text-align:right text-align:justify`

`vertical-align:top vertical-align:middle`

`vertical-align:bottom`

`text-indent:5em text-indent:0.2in`

Inline Style Sheets (cont.)

```
<html>

<head>
  <title>Inline Style Sheets</title>
</head>

<body>
  <p>Here is an image
    
    embedded in text.
  </p>

  <ol style="list-style-type:upper-alpha">
    <li> one thing</li>
    <li> or another</li>
    <ul style="list-style-type:square;
              whitespace:pre">
      <li> with   this</li>
      <li> or     that</li>
    </ul>
  </ol>
</body>
</html>
```

more style properties & values

margin-left:0.1in margin-right:5%
margin:3em
padding-top:0.1in padding-bottom:5%
padding:3em

border-width:thin border-width:thick
border-width:5
border-color:red
border-style:dashed border-style:dotted
border-style:double border-style:none

whitespace:pre

list-style-type:square
list-style-type:decimal
list-style-type:lower-alpha
list-style-type:upper-roman

Inline Style Sheets (cont.)

```
<html>

<head>
  <title> Inline Style Sheets </title>
</head>

<body>
  <table style="font-family:Arial,sans-serif">
    <caption style="color:red;
                  font-style:italic;
                  text-decoration:underline">
      Student data. </caption>
    <tr style="background-color:red">
      <th> name </th> <th> age </th>
    </tr>
    <tr>
      <td> Abdul </td> <td> 19 </td>
    </tr>
    <tr>
      <td> Kuddus </td> <td> 20 </td>
    </tr>
    <tr>
      <td> Makhon </td> <td> 9 </td>
    </tr>
  </table>
</body>
</html>
```

style sheets can be applied to tables for interesting effects

Document Style Sheets

- inline style sheets apply to individual elements in the page
 - using inline style directives can lead to inconsistencies, as similar elements are formatted differently
e.g., we might like for all `<h1>` elements to be centered
 - inline definitions mix content & presentation
➔ violates the general philosophy of HTML
- alternatively, document style sheets allow for a cleaner separation of content and presentation
 - style definitions are placed in the `<head>` of the page (within `STYLE` tags)
 - can apply to all elements, or a subclass of elements, throughout the page

Document Style Sheets

```
<html>
<head>
  <title>Document Style Sheets</title>
  <style type="text/css">
    h1 {color:blue;
        text-align:center}
    p.indented {text-indent:0.2in}
  </style>
</head>

<body>
  <h1> Centered Title </h1>

  <p class="indented">This paragraph will
have the first line indented, but
subsequent lines will be flush. </p>

  <p>This paragraph will not be indented.
</p>

  <h1> The End </h1>

</body>
</html>
```

document style sheets ensure that similar elements are formatted similarly

- can even define **subclasses** of elements and specify formatting

p.indented defines subclass of paragraphs

- inherits all defaults of **<p>**
- adds new features

to specify this newly defined class, place **class="ID"** attribute in tag

note how "clean" the **<body>** is

Document Style Sheets (cont.)

```
<html>
<head>
  <title> Inline Style Sheets </title>
  <style type="text/css">
    table {font-family:Arial,sans-serif}
    caption {color:red;
              font-style:italic;
              text-decoration:underline}
    th {background-color:red}
  </style>
</head>

<body>
  <table>
    <caption> Student data. </caption>
    <tr><th> name </th>      <th> age</th></tr>
    <tr><td> Abdul Alim </td>  <td> 19 </td></tr>
    <tr><td> Kuddus Miah </td> <td> 20 </td></tr>
    <tr><td> Makhon Sorkar </td> <td> 9 </td></tr>
  </table>
</body>
</html>
```

document style sheets are especially useful in formatting tables

effectively separates content from presentation

what if you wanted to right-justify the column of numbers?

what if you changed your mind?

Pseudo-Elements

```
<html>

<head>
  <title>Title for Page</title>
  <style type="text/css">
    a {color : red;
      text-decoration : none;
      font-size : larger}
    a:visited {color : black}
    a:active {color : orange}
    a:hover {color : blue}
    p:first-letter {font-size : large;
      color : white;
      background-color : darkblue}

  </style>
</head>

<body>
  <p> Welcome to my Web page.  I am so
  happy you are here.
  </p>
  <p> Be sure to visit
  <a href="http://bdnews24.com">BDNEWS24</a>
  for late-breaking news.
  </p>
</body>
</html>
```

pseudo-elements are used to address sub-parts of elements

- can specify appearance of link in various states
:visited :active :hover
- can specify format of first line in page or paragraph
:first-line
- can specify format of first letter in page or paragraph
:first-letter

Danger : changing the look of familiar elements is confusing

Careful : current browsers do not support all CSS3 features

Selectors, Properties, and Values

- Document (and external) style sheet directives consist of a “selector”, together with one or more “property: value” pairs, where the pairs are enclosed inside of braces, and separated by semi-colons.
- Examples include:

```
h1 { color: blue; text-align: center; }
```

```
.alert { text-decoration: underline;  
        color: red;  
        font-size: 150%; }
```

```
ol, a { background-color: yellow;  
        font-style: bold;  
        font-family: "Times New Roman"; }
```

These directives are placed in a `<style>` element in the `<head>` element, in a manner similar to the previous example.

Selectors, Properties, and Values (cont.)

- Note that the example

```
.alert { text-decoration: underline; color: red; font-size: 150%; }
```

- creates a class, which can (in principle) be applied to any HTML element, in a manner similar to the example below (assumed to be part of a larger valid HTML document).

```
<p class="alert">Help me! I'm falling down!</p>
```

- This would create a paragraph with text that is red, underlined, and 150% of the normal text size of the webpage.

Selectors, Properties, and Values (cont.)

- The first example below would apply the “alert” class to both items in the list, so both would be red, underlined, and in a larger font. (The number of the list item is also in a larger red font and could appear underlined or not, depending upon the browser used, it seems.)

```
<ol class="alert">  
  <li>Help!</li>  
  <li>Beware!</li>  
</ol>
```

- This second example would apply the “alert” to only the first item in the list.

```
<ol>  
  <li class="alert">Help!</li>  
  <li>Beware!</li>  
</ol>
```

Selectors, Properties, and Values (cont.)

- Also note the very subtle (but important!) distinction between the following two selector definitions:

```
ol, a { background-color: yellow;  
        font-style: bold;  
        font-family: "Times New Roman"; }
```

- This makes a style declaration that applies to both `` and `<a>` elements. (Note the **comma** between the `ol` and `a`.)

```
ol a { background-color: yellow;  
        font-style: bold;  
        font-family: "Times New Roman"; }
```

- This second example makes a style declaration that applies to `<a>` elements that are children of `` (i.e. contained in) elements. (There is no comma between the `ol` and `a`.)

Classes and IDs

- It is possible to define classes that apply only to specific HTML elements. One example like this was shown already.

```
p.indented { text-indent: 0.2in; }
```

- This allows you to ensure that certain styles aren't accidentally applied where they shouldn't be (for example).
- An id can also be defined, which is supposed to be used to specify a unique element within a webpage. For example:

```
#bigorange { color: #ffa500;  
             font-size: 200%;  
             font-weight: bold; }
```

- Then apply it to the HTML element using id="bigorange" in the HTML element's tag.

Formatting Tables Using CSS (an example)

```
<html>
<!-- COMP519 page21.html 2015.10.01 -->
<head>
    <title> Inline Style Sheets </title>
<style type="text/css">
    table { font-family: Arial,sans-serif; }
    caption { color: red; font-style: italic;
text-decoration: underline; }
    th { background-color:red; }
    td { padding: 0 1em 0 2em; }
</style>
</head>
<body>
    <table>
        <caption> Student data. </caption>
        <tr><th> name </th> <th> age</th></tr>
        <tr><td> Chris Smith </td> <td> 19 </td></tr>
        <tr><td> Pat Jones </td> <td> 20 </td></tr>
        <tr><td> Doug MacKenzie </td> <td> 32 </td></tr>
    </table>
</body>
</html>
```

External Style Sheets

- modularity is key to the development and reuse of software
 - design/implement/test useful routines and classes
 - package and make available for reuse
 - saves in development cost & time
 - central libraries make it possible to make a single change and propagate
- external style sheets place the style definitions in separate files
 - multiple pages can link to the same style sheet, consistent look across a site
 - possible to make a single change and propagate automatically
 - represents the ultimate in content/representation separation

Modularity & Style Sheets

```
<html>
<head>
  <title>Title for Page</title>
  <link rel="stylesheet"
        type="text/css"
        href="myStyle.css"
        title="myStyle" />
</head>

<body>
  <h1>Centered Title</h1>

  <p class="indented">This paragraph will
have the first line indented, but
subsequent lines will be flush.</p>

  <p>This paragraph will not be indented.
</p>

  <h1>The End</h1>

</body>
</html>
```

```
/* myStyle.css  COMP519 02.09.05 */

h1 {color : blue; text-align : center}
p.indented {text-indent:0.2in}
```

ideally, the developer(s) of a Web site would place all formatting options in an external style sheet

all Web pages link to that same style sheet for a uniform look

- simplifies Web pages since only need to specify structure/content tags

Media Queries (or Media Rules)

- In addition to (or instead of) style sheets for different types of media, you can also include “media queries” or “media rules” in a style sheet that can be used to define styles for different types of media and/or devices.
- Media queries were introduced in the CSS3 specification, extending the media types of CSS2.
- Media queries look at the capability of the device being used by the client, and can check many things including (according to the W3C explanation of the “media rule”):
 - width and height of the viewport
 - width and height of the device
 - orientation (is the phone/tablet in portrait or landscape mode?)
 - resolution

Media Queries (cont.)

- Interested people can find more information about media queries using the provided link (and, of course, other online searches).
- A typical example might be something like the following:

```
<style type="text/css">
    . . . .
    @media screen and (max-width: 500px) {
        body { background-color: green; }
        ul { list-style-type: square; } }
</style>
```

Bootstrap

- Bootstrap is a free and open-source framework and collection of tools that can be used for creating websites.
- Bootstrap has design templates using HTML and CSS declarations for forms, navigation elements, buttons, typography, and other interface elements. It also includes optional JavaScript extensions (utilizing jQuery).
- Bootstrap has been designed to work with all major browsers (but the alpha release, in August 2015, of Bootstrap 4 has dropped support for Internet Explorer version 8). And it has also been designed to support mobile devices through the use of media queries in all of its CSS templates.

Web rules of thumb

- HTML and CSS provide lots of neat features,
but just because you can add a feature doesn't mean you should!

don't add features that distract from the content of the page

- use color & fonts sparingly and be careful how elements fit together
e.g, no purple text on a pink background, no weird fonts
e.g. I find white text on a black background difficult to read
- use images only where appropriate
e.g., bright background images can make text hard to read
e.g., the use of clickable images instead of buttons or links can slow access
- don't rely on window or font size for layout
e.g., font size may be adjusted by viewer, window constrained
- don' t be annoying
e.g., no pop-up windows, excessive advertising, silly music
- break a large document into several smaller ones or provide a menu for navigation
- stick to standard features and test several browsers if possible
- utilize style sheets to make changes easy & ensure consistency

Exercise

- Apply CSS to your homepage for assignment -1(LAB)
- What is layout?
- **READINGS**
 - M Schafer: Ch. 12-18
 - <https://www.w3schools.com/html>

Acknowledgement

- This module is designed and created with the help from following sources-
 - <https://cgi.csc.liv.ac.uk/~ullrich/COMP519/>
 - <http://www.csc.liv.ac.uk/~martin/teaching/comp519/>