CS 312
Internet Concepts

Developing Web Content:
Cascading Style Sheets (CSS)

Dr. Michele Weigle
Department of Computer Science
Old Dominion University
mweigle@cs.odu.edu

http://www.cs.odu.edu/~mweigle/CS312-F09/

Styling HTML Pages

- Distributed, local styling
  - » Definitions of the presentation of individual items
    - ❖ By default, defined on the spot
  - » Great flexibility, but a lot of work
    - ❖ Items need repeated definitions for the same non-default presentation
    - ❖ Inconsistencies or errors may cause confusion
  - » Modifying a style
    - ❖ Requires individual effort
    - ❖ Hard to maintain consistency, particularly hard in large pages, or sites with multiple pages

Styling HTML Pages

- Presentation lacks a dynamic aspect
  - Regarding user action
    - No reaction to user actions
  - Regarding user input
    - All actions depend on the server, not the client

Styling Multiple HTML Pages at Same Site

- Without consistent styles
  - Easy to implement
  - Can be quite confusing
  - Typically, users dislike such sites
- With consistent styles
  - Typically welcomed by users
  - To implement in HTML
    - requires much effort to coordinate and to enforce
    - harder to update

Styling HTML Pages

- How to alleviate the previously mentioned problems?
  - » Separate specifications of content and style
  - » Use additional facilities, e.g. cascading style sheets
    - To define how and what to display consistently
      - Top-down approach
      - Requires minor coding efforts
      - Easier to update
    - Flexibility provided
      - Inheritance vs. overriding

Cascading Style Sheets (CSS)

- A computer language
  - » Separate layout from content
  - » Describe the stylistic presentation of documents with a very simple syntax
    - e.g., colors, fonts, images, borders, margins
- Characteristics
  - » Define non-default presentation styles for objects, or classes of objects
  - » Allows centralized definition, but also allows overriding at different places
  - » Implemented on newer browsers
    - newer browsers may still act inconsistently, even with legal coding.

Benefits of CSS

- Separate document content and structure
  - » Reduced complexity of structural content
  - » Easier design, implementation, and maintenance
    - Easier Design - consistent styles
      - For a single page
      - For multiple pages
    - Easier implementation and modification
      - May be strategically located
      - In a separate CSS file
      - Inside pages where desired

Benefits of CSS

  - » Greater control of presentation characteristics
    - Using profiles for different devices, different users
  - » Improved content accessibility

- Broad applicability, may be used with
  - » HTML
  - » XML
  - » other structured document format

Flexibility in CSS Definition

- External: in a separate CSS file
  - » Referenced from inside the <head> </head> section of (multiple) HTML pages, e.g.,
  - » <link rel="stylesheet" type="text/css" href="styleOne.css" />
  - » *Preferred style, what we'll focus on*
- Embedded:
  - » In the <head> section of a HTML page in a <style> </style> block
  - » Can override external definition
- Inline:
  - » In a HTML <body> </body> section, exactly where it is needed
  - » Can override external and embedded definitions

CSS Syntax

- Comments are enclosed in /* */
- Every style sheet consists of a set of *rules*
- A *rule* is made up of three parts:
  - » a selector
    - ❖ specifies the items for which the rule holds, generally tag names (*e.g.*, p, body, h2)
  - » a property
    - ❖ attribute (*e.g.*, color, font-family)
  - » a value
- Example syntax:
  - » selector {property**:** value**;**}
  - » selector {property1: value1; property2: value2;}
- Validator
  http://jigsaw.w3.org/css-validator/

Examples

- body {color: black}
  - » Sets the body text color to black
- p {font-size: 10pt; font-family: "sans serif"}
  - » Sets anything with the <p> tag to the 10-point sans serif font
- h1,h2 {color: green;}
  - » Sets the level 1 and 2 heading text color to green

Another Example

```
p
{
   text-align: center;
   color: black;
   font-family: arial
}
```

- Everything inside of <p> tags is centered, black, and arial text

Applying CSS to HTML

- In-line
  - » Use the style attribute

    my red text
  - » Turns only this paragraph red
- Embedded, or Internal

  <head>

  <style type="text/css">

  p { color: red;}

  </style>

  </head>
  - » Turns all paragraphs in the page red

http://www.htmldog.com/guides/cssbeginner/applyingcss/

Applying CSS to HTML

- External
  - » Separate CSS file (ex: web.css)

    p {color: red;}
  - » Load into HTML file

    <head>

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

    </head>
  - » To make changes to style, only need to edit CSS file, not HTML

http://www.htmldog.com/guides/cssbeginner/applyingcss/

Demonstrating CSS

- We're going to start with a simple example and then build the CSS as we learn new selectors and properties

- mystyle.css
  http://www.cs.odu.edu/~mweigle/cs312/css/mystyle.css

- mypage.html
  http://www.cs.odu.edu/~mweigle/cs312/css/mypage.html

Lengths and Percentages

- em – calculated size of a font
  - » 2em is two times the current font size
- px – pixels
- pt – points
  - » ex: font-size: 12pt
- % - percentages
- others: cm, mm, in

- *em and % are recommended over px, in, cm because they are resizeable*

http://www.htmldog.com/guides/cssbeginner/

Colors

- name
  - » 16 color names – validate with W3C validator
  - » other names are available – won't validate
    http://www.w3schools.com/css/css_colornames.asp
- red-green-blue
  - » absolute values: rgb(255,0,0)
  - » percentages: rgb(100%, 0%, 0%)
- hex value
  - » 6-digits: #ff0000
  - » 3-digits: #f00
    - ❖ each digit represents 2 of the same digit in the 6-digit form

http://www.htmldog.com/guides/cssbeginner/

Colors

- Can specify both text color and background color
  - » color
  - » background-color

- Add to mystyle.css

```
body {
   font-size: 1.4em;
   color: navy;
}

h1 {
   color: #ffc;
   background-color: #009;
}
```

http://www.htmldog.com/guides/cssbeginner/

Text

- font-family
  - » font must be on the user's computer, so don't specify something obscure
  - » safe fonts: arial, verdana, "times new roman"
  - » if you specify more than one font, the browser will display in the first font that it actually has
- font-size
  - » xx-small, x-small, small, medium, large, x-large, xx-large
  - » use percentage or em
- font-weight
  - » bold, normal
- font-style
  - » italic, normal

http://www.htmldog.com/guides/cssbeginner/

Text

- text-decoration
  - » overline, line-through, underline, none
- text-transform
  - » capitalize, uppercase, lowercase, none
- text spacing
  - » *letter-spacing* – spacing between letters (length or normal)
  - » *word-spacing* – spacing between words
  - » *line-height* – height of lines, doesn't adjust font size
  - » *text-align* – left, right, center, justify
  - » *text-indent* – indent first line of a paragraph to given length or percentage

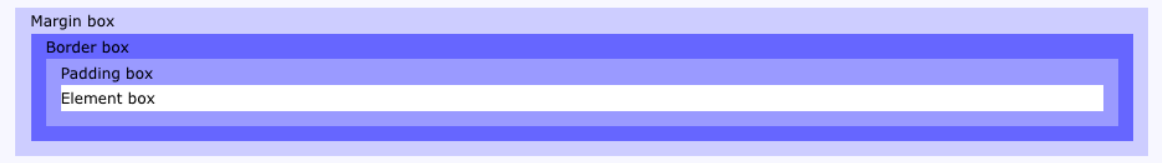
http://www.htmldog.com/guides/cssbeginner/

Edit mystyle.css

```
body {
    font-family: arial, helvetica,
    sans-serif;
    font-size: 0.8em;
    color: navy;
}
h1 {
    color: #ffc;
    background-color: #009;
    font-size: 2em;
}
h2 {
    font-size: 1.5em;
}
strong {
    font-style: italic;
    text-transform:
    uppercase;
}
p {
    line-height: 1.5;
}
```

http://www.htmldog.com/guides/cssbeginner/

Box Model



- Content (element box) is in the middle
- Surrounding content is the *padding*
- Surrounding padding is the *border*
- Surrounding border is the *margin*

http://www.htmldog.com/guides/cssbeginner/

Margins and Padding

- margin
  - » space outside of the element
  - » *margin*: sets all margins to same value
  - » *margin-top*, *margin-right*, *margin-bottom*, *margin-left*

- padding
  - » space inside the element
  - » padding setting is in the same manner as margin

http://www.htmldog.com/guides/cssbeginner/

Edit mystyle.css

- Change h2 code to

```
h2 {
   font-size: 1.5em;
   background-color: #ccc;
   margin: 1em;
   padding: 1.5em;
}
```

http://www.htmldog.com/guides/cssbeginner/

Borders

- border-style
  - » solid, dotted, dashed, double, groove, ridge, inset, outset
  - » also *border-bottom-style, border-top-style, border-left-style, border-right-style*
- border-width
  - » width of border, usually in pixels
  - » can set for any border side (like border-style)
- border-color
  - » can set for any border side (like border style)

http://www.htmldog.com/guides/cssbeginner/

Edit mystyle.css

```
h2 {
   font-size: 1.5 em;
   background-color: #ccc;
   margin: 1 em;
   padding: 1.5 em;
   border-style: dashed;
   border-width: 3px;
   border-left-width: 10px;
   border-right-width: 10px;
   border-color: red;
   border-bottom-style: solid;
   border-bottom-color: black;
}
```

http://www.htmldog.com/guides/cssbeginner/

Formatting Lists

- list-style-type
  - » disc, circle, square, decimal, lower-roman, upper-roman, lower-alpha, upper-alpha, none
- list-style-image
  - » provide URL
- list-style-position
  - » where is marker placed in regard to list item
  - » inside, outside
- display: inline
  - » allows the elements of a list to appear on the same line (rather than one line per list item)

Add to mystyle.css

```
ul {
   list-style-type: square;
   list-style-position: inside;
}

ul ul {
   list-style-type: disc;
}

ol {
   list-style-type: upper-roman;
}
```

Formatting Tables

- table
  - » border-collapse
    - ❖ are borders joined or separate? *collapse, separate*
- th, td
  - » similar formatting to text
  - » padding, border, margin, alignment, colors, etc.

Add to mystyle.css

```
table {
   border-collapse: collapse;
   width: 50%;
}

td {
   border-bottom: 1px solid red;
}
```

```
th {
   padding: 0.5em;
   text-align: left;
   background: #ffc;
   border-top: 1px solid red;
   border-bottom: 1px solid red;
}
```

We can often combine several attributes in one line.

Formatting Links

- Can format links just like normal text

- a:link
  - » normal unvisited links
- a:visited
  - » visited links
- a:hover
  - » hovered links (when the mouse moves over it)
- a:active
  - » active links (when you click on it)

If these are used, must be specified in this order for consistency.

love-hate

Add to mystyle.css

```
a {
   color: black;
}

a:hover {
   text-decoration: none;
   color: white;
   background: blue;
   font-size: 1.3em;
}
```

Final Product

- May not be pretty, but demonstrates many of the features of CSS

http://www.cs.odu.edu/~mweigle/cs312/css/mypage.html

CSS Classes

- For variety on a selector, use classes

- Syntax of class declaration:
  - » [*tag*].*classname* {property.value;}
    - ❖ the square bracket above means optional.
  - » *tag* is some item/selector
  - » *tag* is optional, the declaration is for it
    - ❖ without *tag* means all tags can use it
  - » *classname* is the user defined name of a class

CSS Classes

- Typically, several classes are defined for one tag, in separate rules
  - » Often, there is a list of *property:value;* pairs in the braces

- Example rules, two classes for p:
  - » p.normal {font-weight: normal}
  - » p.thick {font-weight: bold}

Examples

In CSS file:

```
p.normal {font-weight: normal}
p.thick {font-weight: bold}
```

In HTML file:

```
<p class="normal"> This is a paragraph</p>
<p class="thick">This is a paragraph</p>
```

Displays:

This is a paragraph

This is a paragraph

Another Example

In CSS File:

```
p.thick {font-weight: bold}
p.uppercase {text-transform: uppercase;}
p.lowercase {text-transform: lowercase;}
p.captialize {text-transform: capitalize;}
```

In HTML File:

```
<p class="uppercase" This is text</p>
<p class="lowercase thick">This is text</p>
<p class="capitalize">This is text</p>
```

Displays:

THIS IS SOME TEXT

this is some text

This Is Some Text

CSS Classes

- .center {text-align: center}
  - » This allows any kind of tag to apply this class

```
<h1 class="center"> This heading will be center-
  aligned </h1>

<p class="center"> This paragraph will also be
  center-aligned. </p>
```

Which CSS is more efficient?

- Using an external CSS file

- The style sheet gets stored in the browser's cache.
- It can be ready for use immediately on all the pages without the need for the stylesheet being re-loaded.
  - » Faster than using inline or embedded stylesheets.
- It can be used by many pages
  - » Enforcing basic common style
  - » Easier implementation and update