

# 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-F10/>

1

## Styling HTML Pages Multiple Pages at Same Site

---

- ◆ Without consistent styles
  - » Easy to implement
  - » Can be quite confusing
    - ❖ umm, [www.odu.edu](http://www.odu.edu) anyone?
  - » 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

<p><u>Solution:</u> Separate content from layout and styling</p>
--

2

# **Cascading Style Sheets (CSS)**

---

- ◆ 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
  - » Browsers may still act inconsistently, even with legal coding (surprise!)

3

## **Benefits of CSS**

### **Separate Content and Structure**

- ◆ Reduced complexity of structural content
- ◆ Easier design, implementation, and maintenance
  - » Easier Design - consistent styles
    - ❖ for a single page and for multiple pages
  - » Easier implementation and modification
    - ❖ may be located in a separate CSS file or inside pages where desired
- ◆ Greater control of presentation characteristics
  - » Using profiles for different devices, different users
- ◆ Improved content accessibility

4

# Benefits of CSS

---

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

5

## 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="style.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

6

# 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/>

7

# 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

8

## Another Example

---

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

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

9

## Applying CSS to HTML

---

- ◆ In-line
  - » Use the style attribute  
`<p style="color: red">my red text</p>`
  - » 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

# 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/>

11

# 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>

12

## Lengths and Percentages

---

- ◆ em – calculated size of a font
  - » default font is 16px, so 1em = 16px
  - » 2em is two times the current font size (32px)
- ◆ px – pixels
- ◆ pt – points
  - » ex: font-size: 12pt
- ◆ % - percentages
- ◆ others: cm, mm, in
  
- ◆ *em and % are recommended over px, in, cm because they are resizable*

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

13

## Colors

---

- ◆ name
  - » 16 color names – validate with W3C validator
  - » other names are available – won't validate
    - [http://www.w3schools.com/css/css\\_colornames.asp](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/>

14

# Colors

---

- ◆ Can specify both text color and background color

- » color
- » background-color

- ◆ Add to mystyle.css

```
body {  
    font-size: 1.4em;  
    color: navy;  
    background-color: #ffc;  
}  
  
h1 {  
    color: #ffc;  
    background-color: navy;  
}
```

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

15

# 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/>

16



# 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/>

17

# Edit mystyle.css

---

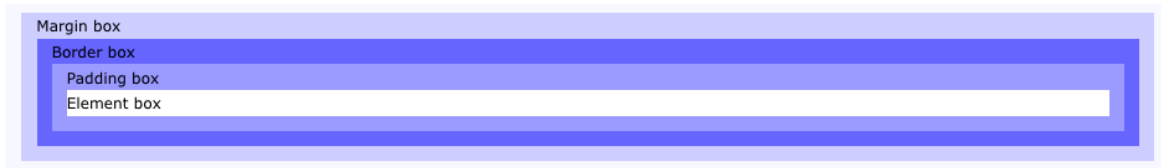
```
body {  
    font-family: arial,  
    helvetica, sans-serif;  
    font-size: 1.2em;  
    color: navy;  
    background-color: #ffc;  
}  
  
h1 {  
    color: #ffc;  
    background-color: navy;  
    font-size: 2em;  
}  
  
h2 {  
    font-size: 1.75em;  
}  
  
strong {  
    font-style: italic;  
    text-transform:  
    uppercase;  
}  
  
p {  
    line-height: 1.5;  
}
```

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

18

# 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/>

19

## 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/>

20

# 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/>

21

# Edit mystyle.css

---

- ◆ Change h2 code to

```
h2 {  
    font-size: 1.5em;  
    margin: 1em;  
    border-style: solid;  
    border-color: navy;  
}
```

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

22

# Formatting Lists

---

- ◆ **list-style-type**
  - » disc, circle, square, decimal, lower-roman, upper-roman, lower-alpha, upper-alpha, none
- ◆ **list-style-image**
  - » provide URL (relative or absolute)
- ◆ **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)

23

# Add to mystyle.css

---

```
ul {  
    list-style-type: square;  
    list-style-position: inside;  
}  
  
ul ul {  
    list-style-type: disc;  
}  
  
ol {  
    list-style-type: lower-roman;  
}
```

24



## 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***

27

## Add to mystyle.css

---

```
a {  
    color: black;  
}
```

```
a:hover {  
    text-decoration: none;  
    color: white;  
    background: navy;  
}
```

28

# Final Product

---

- ◆ Demonstrates many of the features of CSS

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

29

# 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

30

## 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}`

31

## 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**

32



## Another Example

---

In CSS File:

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

**this is text**

This Is Text

33

## 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>
```

34

# Efficiency of Including CSS

## **Most efficient: 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