

Introduction to Cascading Style Sheets

Pat Morin
COMP 2405

Outline

- Motivation for CSS
- How to use CSS in your documents
- Formatting text with CSS
 - Font properties
 - Text properties
- Formatting lists with CSS
- Summary

HTML

- HTML (when used correctly) describes the different parts of a document
 - Paragraphs, section headings, quotes, images, ...
- HTML (when used correctly) does not describe the formatting of a document
- HTML is a *logical* markup language not a *physical* markup language
- HTML was designed to be rendered on a wide variety of devices
 - Graphical web browsers, text terminals, screen readers, ...

CSS

- CSS (Cascading Style Sheets) is for the formatting side of the Web
- CSS describes how rendered HTML documents should look
- CSS considers the physical and visual display of the document (the *Style*)

Advantages of CSS

- The use of CSS separates document layout from document content
 - Different people can be responsible for the two parts
 - Document author can focus on content
 - Graphic designer can focus on layout
- A single file can control the look of an entire web site
 - Easy to modify look of web site without affecting its contents
 - Easy to obtain a consistent look (the R in CRAP)
- If done correctly, documents degrade gracefully on platforms that don't support visual formatting

Disadvantages of CSS

- More to learn
 - CSS is powerful but complex
- Not fully supported on some browsers
 - Even some modern browsers are not fully CSS 2 compliant
- The formatting is separated from the document
 - Makes it hard to write a document and format it simultaneously

How to Write a Document

1. Decide what you want to write

- First and foremost the content of the document is important
- Decide on the logical structure of the document

2. Write it

- Write the document content
- Markup the document's logical structure

3. Format it

- Use CSS to do formatting
- Add other formatting-specific data (e.g., navigation)
- Points 1 and 2 should take the majority of the time

Using CSS

- There are three ways to use CSS
- External Style Sheets
 - Uses the LINK tag (in the document HEAD)

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

- Inline Style Sheets
 - The STYLE tag (in the document HEAD)

```
<style type="text/css">
  /* CSS information goes here */
</style>
```


Using CSS (Cont'd)

- Inline STYLE Attributes

- The STYLE attribute (within another HTML tag)

```
<a style="font-size: 10pt;" href="xsk.html">
```

- Don't use these

- Gives the disadvantages of CSS without the most important advantages

Simple CSS

- An Inline CSS example that modifies some of the common HTML tags

```
<style type="text/css">
  h1 {
    font-size: 12pt;
    font-family: "Luxi Sans", sans-serif;
    font-weight: bold;
  }

  p {
    font-size: 12pt;
    font-family: "Luxi Serif", serif;
  }
</style>
```

CSS with Class

- (Almost) all HTML tags can have an optional CLASS attribute
- This works nicely in conjunction with CSS:

```
<style type="text/css">
  p.blogentry {
    font-size: 12pt;
    font-family: "Luxi Serif", serif;
  }
</style>
...
<p class="blogentry">
  Today was the best day ever ...
</p>
```

CSS with IDs

- Similarly, almost all HTML tags can have an ID attribute:

```
<style type="text/css">
  #menubar {
    font-size: 12pt;
    font-family: "Luxi Serif", serif;
  }
</style>
...
<ul id="menubar">
  <li>Item 1</li> ...
</ul>
```

Notes on CLASS and ID

- Use CLASS when a document contains (or may contain) many items of this type
 - E.g.: `blogEntry`, `nameList`, `properName`
- Use ID when a document can only contain one item of this type
 - E.g. `pageTitle`, `mainNavigationBar`
- Choose CLASS and ID names wisely
 - Good: `Pat Morin`
 - Bad: `Pat Morin`

The DIV and SPAN tags

- HTML has 2 tags, DIV and SPAN, that are specifically used with CSS and CLASS and ID attributes
- A DIV tag contains a block of text
 - Like a paragraph, section heading, or title
- A SPAN tag contains a short piece of text within a block
 - Like a proper name, or a dollar amount

Example of DIV and SPAN

```
<div class="mainDocument">  
  On his way to school,  
  <span class="properName">Uncle Rico</span>  
  stopped at the local groceteria and spent  
  <span class="money">$4.55</span> on milk,  
  bread, and honey.  
</div>
```

```
<div class="about">  
  This document was written by  
  <span class="properName">Pedro Sanchez</span>  
  on <span class="date">Jan. 9th, 2007</span>.  
</div>
```

Typeface: font-family

- Consists of a comma-separated list of font names

```
h1 { font-family: "Times New Roman",  
          "Luxi Serif",  
          "DejaVu LGC Serif",  
          serif;  
}
```

- Not all fonts are available on all systems
 - Browser will try fonts in order that they are specified
- Always make the last option one of the *generic fonts*: serif, sans-serif, *curative*, fantasy, or monospace

Font Size: *font-size*

- Specified in one of the CSS size units
 - 1em is the width of a letter m (relative to surrounding text)
 - 1pt is a standard typographic point (1/72 inches)
 - 1px is one screen pixel
 - Keywords: xx-small, x-small, small, medium, large, x-large, xx-large are browser dependant
 - Percentages: XXX% relative to the surrounding code
- Points and pixels are extremely accurate but not resizable

```
h1 { font-size: 200%; }  
h2 { font-size: large; }  
h3 { font-size: medium; }
```

Bold Fonts: font-weight

- Can be one of normal, bold, bolder, lighter, 100, 200, ..., 900

```
h1 { font-weight: bold;
      font-size: 200%;
    }
```

```
h2 { font-weight: bold;
      font-size: medium;
    }
```

The Font Style: font-style

- Can be one of `normal`, `italic`, or `oblique`
- An *italic* font is usually slanted and maybe has some curls added
- An *oblique* font is a skew transformation of a regular font (not well supported)

```
em { font-style: italic; }  
h1 { font-family: sans-serif;  
      font-weight: bold;  
      font-style: italic;  
}
```

Other Font Properties

- Use `font-variant` for producing small caps
- Use `font-stretch` to expand or condense a font
- Use `font` as a shorthand to set all font properties at once

```
h1 { font: italic
      small-caps
      bold
      12pt
      sans-serif;
}
```

Text Properties: color

- Can be a color name, a rgb value, or a hexadecimal value
- 147 color names are at
 - http://www.w3schools.com/css/css_colornames.asp

```
/* Make all headings pure red */  
H1 { color: red; }  
H2 { color: rgb(255,0,0); }  
H3 { color: #ff0000; }
```

Text Properties: background

- The background property specifies the background color and/or image
- background-color specifies a color (named, RGB, or hex)
- background-image, background-position, background-repeat, and/or background-attachment to specify a background image
- Usually used on the body tag

Text Properties: text-align

- Determine which side of the page (or containing box) that text lines up with
- Can be one of left, right, center, or justify

```
h1 { text-align: center; }  
p { text-align: left; }
```

Text Properties: text-decoration

- Add some decoration to the text
- Can be one of none, underline, overline, line-through, or blink
- Treat blink like a contagious disease

```
h1 { text-decoration: underline; }  
a { text-decoration: none; }
```


Text Properties: text-indent

- Specifies a length by which the first line of text should be indented
- Length can be measured in a relative unit
 - em: the font-size of the current font (width of a letter m)
 - ex: the x-height of the current font (height of a letter x)
 - px: one screen pixel
- Or an absolute unit
 - Inches (in) centimeters (cm) millimeters (mm), points (pt), or picas (pc)
 - 1pt = (1/72)in and 1pc=12pt

```
P { text-indent: 5em; }
```

Text Properties: text-transform

- Can actually transform the text
- Can be one of
 - none: don't do anything to the text
 - capitalize: Start each word with an uppercase letter
 - uppercase: Make everything uppercase (allcaps)
 - lowercase: Make everything lowercase

```
h1 { font-size: large;
      text-transform: capitalize;
}

h2 { font-size: normal;
      text-transform: uppercase;
}
```

Text Properties: white-space

- Defines how white space (space, tabs, carriage-returns) is handled
- Can be one of
 - `normal`: Normal formatting, wrap at white space, merge multiple whitespace
 - `pre`: treat as preformatted text
 - `nowrap`: don't wrap at whitespace

```
a.properName { white-space: nowrap; }
```

Other Text Properties

- `direction`: Sets the text direction
- `line-height`: Sets the distance between lines
- `letter-spacing`: Sets the space between characters
- `text-shadow`: Make a colored text shadow
- `unicode-bidi`
 - Allows english blocks to flow left-to-right and Arabic blocks flow right-to-left on the same page
- `word-spacing`: Sets the space between words
- These probably should not be used unless you know a lot about typesetting.

CSS and Lists

- Several properties are specific to lists and list items
- `list-style-image`: sets an image to use as the bullet marker
- `list-style-position`: controls where the bullet appears
- `list-style-type`: specifies what kind of bullet to use
 - none, disc, circle, square, decimal, decimal-leading-zero, lower-roman, upper-roman, lower-alpha, upper-alpha, lower-greek, lower-latin, upper-latin, hebrew, armenian, georgian, cjk-ideographic, hiragana, katakana, hiragana-iroha, katakana-iroha

CSS Lookup

- We can specify styles that apply to tags or identifiers only when contained in other tags

```
ul.menu { padding: 0; }
```

```
ul.menu li { text-decoration: none; }
```

```
...
```

```
<ul class="menu">  
  <li>Menu item1</li>  
  <li>Menu item2</li>  
</ul>
```

CSS and the A Tag

- CSS has several options for the A tag
 - A:link defines the style for normal unvisited links
 - A:visited defines the style for links that have already been visited in the past
 - A:active defines the style for links after the user clicks on them (usually while the next page loads)
 - A:hover defines the style for links when the mouse pointer is hovering over them
- A:hover is useful for making things that look like menus

Summary

- CSS provides fine-grained control over
 - fonts
 - text
 - the display of lists
- The CSS lookup mechanism allows us to apply styles to tags that only appear within other tags
- In practice, many authors ignore most HTML tags other than DIV and SPAN