Computer Science E-75 April 21
- [ ] David Heitmeyer guest lecturer: Cascading Style Sheets (CSS)
- [ ] What happens on the browser
  - [ ] Markup (Structure)
  - [ ] CSS (Presentation)
  - [ ] JavaScript (Functionality)
- [ ] Advantages of using CSS
  - [ ] Mixing up CSS and HTML can lead to problems later on, in maintenance and in code readability
  - [ ] Lots more control over the presentation
  - [ ] Accessibility (Individuals, devices)
  - [ ] You can take one website, change the CSS, and get radically different styles

- Separation of CSS and XHTML (presentation and markup)
    - allows easier updating
    - makes things clearer
    -
- Headers and Lists are great in CSS for navigation.
- Just changing the rules (with the same markup), can result in vastly different pages

- CSS is a text document with selectors (element names like body, h1, ul, li, etc.)
    - Within the curly braces we have properties and their values.

```
img {
  border: thin solid black;
}
```

- "img" is a selector, "border" is a property, "thin solid black" are the values (whitespace is unimportant, but used for readability).

- CSS Resources
www.alistapart.com
www.simplebits.com
meyerweb.com/eric/css/
www.westciv.com/style_master/house

There are three ways to get style rules to bind to the markup:
1. style attribute in element - inline
    - <body style="YOURCSS">

2. style element in XHTML/HTML head - internal
    - in the head tags:

```
<style type="text/css">
YOURCSS
</style>
```

3. refer to a separate css file in the head - external

```
<link rel="stylesheet" type="text/css" href="example3.css" />
```

- rel specifies the way that the link element is used
    - (like rss, site icons)

A good practice is to use external style sheets, and only override them when necessary.

COMBINING RULES AND SELECTORS

Rules can be separate

```
p {color: black;}
p {background-color: teal;}
p {font-family: helvetica, sans-serif;}
```

- multiple font families are listed because we don't know what will be available, least specific comes last (sans-serif)

Or grouped

```
p {
   color: black;
   background-color: teal;
   font-family: helvetica, sans-serif;
}
```

Can also combine selectors

```
h1 { color: maroon; }
h2 { color: maroon; }
h3 { color: maroon; }
h4 { color: maroon; }
```

becomes

```
h1, h2, h3, h4 { color: maroon; }
```

USE CLASSES AND ID'S

- dot operator used for class

```
div.withstyle {
        CSS
}
```

will apply to elements

- pound operator used for id
#legal {
	CSS
}
will apply to elements

CONTEXTUAL SELECTORS
li em { color: maroon; }
- will apply to all em within context of li
applies to this
- does not have to be a direct child

CHILD
body > p
- applies to p that are direct children of body

ATTRIBUTE
input[type=text]
- specifies which input based on their attributes

WILDCARD
wildcard (*)
- matches any element

INHERITANCE
- properties are typically inherited by child elements
	- they take the attributes given to their parents
- Firebug can be a useful tool to determine inheritance (clears things up)

WHICH CASCADING STYLE SHEETS WILL TAKE EFFECT?

STYLESHEET ORIGIN (in order of priority - author is first)
- author’s stylesheet - creator makes style sheets
- reader's stylesheet - viewer changes style sheet (making text larger, etc)
- UA's stylesheet - user agent (web browser) stylesheet
	- With no style rules, there's a way that the browser will lay markup

SPECIFICITY OF SELECTOR
- "id" attributes - most specific
- "class" attributes - somewhat specific
- element names - least specific

ORDER
- last occurrence has higher preference

FONT PROPERTIES
- font-family
- font-style (normal, italic, oblique)
- font-variant (small-caps, normal)
- font-weight (normal, bold, bolder, lighter, 100, 200 ... 900)
- font-size
    - can specify font size many different ways
    1. relative sizes (UA) (xx-small, x-small, small, medium.. xx-large)
    2. relative sizes (context) (smaller, larger)
    3. relative units (context) (percent (%), em unit (length of an m)
    4. absolute unit (pt)
    - RELATIVE SIZES ARE RELATIVE TO PARENT
        - nested divs will be relative to the parent div
- font
- font shorthand property
[font-styles | font-variant | font-weigh]? font-size[/line-height]? font-family

TEXT PROPERTIES
- word-spacing
- letter-spacing

CSS UNITS
- Length: em, pt, ex, mm, cm, in, px
- Percentage
- URL: url(url goes here)
- Color: name, rgb value

BLOCK MODEL
- margin - space between border and outside edge
- border - divides margin and padding
- padding - space between where content ends and border begins
- content - innermost part (where the meat of the webpages go)

TRBL FOR PADDING AND MARGIN SHORTHAND
"top right bottom left"
p.ex1 { padding: 0.5em 0.25em 0.5em 0.25em; }
is equal to
p.ex1 { padding: 0.5em 0.25em; }

BACKGROUND IMAGES AREN'T JUST FOR THE BODY

- with css, we can apply backgrounds and images to elements (like div)
- instead of having an img element within the XHTML, you can define one in CSS
    - bring visual elements in through css
- using tiling and image size to change images

PSEUDOCLASSES AND PSUEDOELEMENTS
Class: first-child, link, visit, hover, active, focus, lang
Elements: first-letter, first-line
ex.- p:first-line

CLASSNAMES
- be logical, not descriptive
    - keep markup and presentation separate
- choosing class and id names appropriately will help with
    - evolution (as things change on your website)
    - specificity and semantics

FLOAT
- takes the block out of the flow of the containing block and moves it (left or right) within the containing block.
CLEAR
- defines the sides of a block where floated blocks cannot occur

display:none is used to turn off the visibility of certain elements.

LISTS
- lists are controlled with a lot of granularity in CSS
- display lists inline (useful for navigation) (display: inline)