

# CSS and Intro to Javascript

CS174

Chris Pollett

Sep 17, 2007.

# Outline

- Finish up with CSS
- Begin Javascript

# Pseudo Classes

- Last day, we learned about class, generic, and id selectors in CSS.
- These allow you to select which tags a given style applies to.
- In contrast, pseudo selectors allow you to apply styles in the case that certain events happen.
- Two pseudo-classes are supported by Firefox and IE7, but not 6 are: hover and focus.
- An example of using these is:

```
input:hover {color:red} /* input tag will turn red when mouse over. In  
class I demo'd that the hover tag can be used to create menus only  
using CSS*/
```

```
input:focus {color:green} /* when click into input field, it will turn green  
*/
```

# What values can a property have?

- CSS1 supports over 60 properties for tags (the next few slides will discuss these).
- First, though, the values one can assign a property can take one of several forms:
  - Predefined values:
    - For example, bold, red, disc. These are case insensitive
  - Signed Number values:
    - -17, 17em /\*1 em =height of letter m \*/, 17ex, 17mm, 17px, 17cm, 17pt
    - These may or may not have a unit after them.
    - Percentage values 75%
  - URL values:
    - For example: url(mypic.jpg)
  - Color values:
    - These can be predefined as for example the keyword *white*
    - Or you can specify rgb values in one of two formats:
      - rgb(0, 0, 0) /\*decimal \*/
      - #C8C8C8 /\*hex \*/

# Color, Fonts, and Text Decoration

- Colors
  - You should remember when specifying colors that they might look different on different machines / monitors. There is a list of 216 so called web-safe colors in the book.
  - To specify the foreground color of a tag you can set the color property. For example, `div.changed {color: red}`  
`<div class="changed"><p>first paragraph that's new</p><p>2nd...</p></div>`
  - To specify the background you use the background-color property
- Fonts
  - There are several properties of font you use that you can control.
    - font-family (examples: Arial, Times, Courier)
    - font-size (examples: 20pt, 15px, xx-small, x-small, small, medium, large, smaller, larger, etc)
    - font-style (example: italic, oblique)
    - font-weight (example: bold, bolder)
- Text-Decoration
  - The text-decoration property can be further used to control how text looks. Some values are *line-through*, *overline*, *underline*, and *none*.

# List Properties

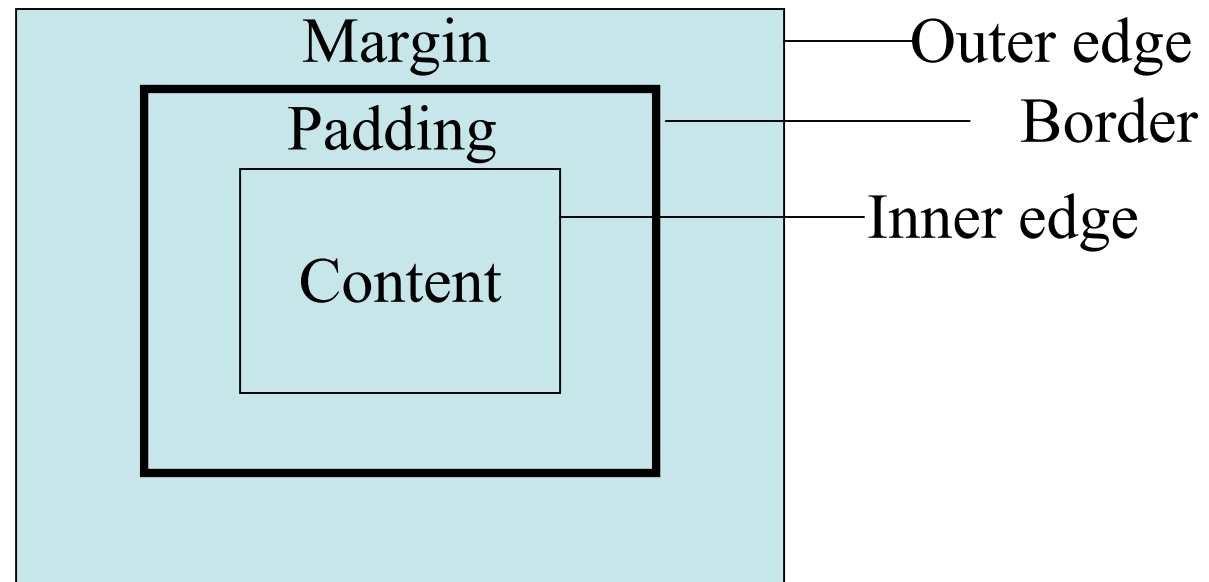
- Here are some properties which can be used to control how ordered and unordered lists are displayed:
  - `list-style-type`: For unordered lists one can have values *disc*, *square*, *circle*; for ordered lists one can have values *decimal*, *upper-alpha*, *lower-alpha*, *upper-roman*, *lower-roman*.
  - `list-style-image`: Can be set to the url of an image to be used for the bullet. For example,  
`<li style="list-style-image: url(mybullet.png)"> item </li>`

# Alignment of Text

- There are several properties which control the flow of text and images in a given set of tags.
  - text-indent --- usually used to control how much the first line of a paragraph is indented.
    - Ex: p.indent {text-indent: 0.5in}
  - text-align --- this controls how the text is aligned within margins. It can have the values: *left*, *right*, *center*, and *justify*.
  - float --- this a property often used with images and tables to control how text flows around the element. possible values are *left*, *right*, and *none*. We showed in class that the float tag can be used to make a simple two column layout.

# Box Model

- Each element in XHTML has properties which control its border and the amount of white space around it



- This model is also used when you use CSS with other kinds of XML documents



# More on the Box Model

- Controlling the border:
  - `border-style`: This can be *none*, *solid*, *dotted*, *dashed*, *double*
  - `border-width`: This can be a length in pixels or *thin*, *medium*, or *thick*. You can more finely control each side of the border by using the properties: *border-top-width*, *border-bottom-width*, *border-left-width*, *border-right-width*
  - `border-color`: This sets the color of the border. This can also be more finely controlled with *border-top-color*, *border-bottom-color*, *border-left-color*, and *border-right-color*.
- Controlling margins and padding:
  - These can be controlled using the properties: *margin-left*, *margin-right*, *margin-top*, *margin-bottom*, *padding-left*, *padding-right*, *padding-top*, *padding-bottom*.

# Background Images

- The background-image property is used to place an image in the background of an element:  
body {background-image: url(background.jpg);}
- The way this image is displayed can be further controlled using the properties:
  - background-repeat: this may have values: *no-repeat*, *repeat-x*, *repeat-y*
  - background-position: this may have values which are space separated combinations of *top*, *center*, *bottom*, *left*, and *right*.

# Position and Display Properties

- The *position* property can have the value *relative* (says to place element relative to the last element) or *absolute* (says to place element at some offset from top left of display).
- For example,

```
<style type="text/css">
  p.first {position: absolute; top:1in; left:2in;}
  p.second {position: relative; left:-1in;}
</style>
<p class="first">displays one inch down two inches from left of top of screen</p>
<p class="second">displays 1inch to the left of where the first paragraph finished</p>
```
- The *display* property can be one of *block*, *inline*, *hidden*, or *none*. *block* causes the box model to be used, *inline* makes the element data flow like text. *hidden* renders the element in the background color, and *none* does not render the element.
- Other useful properties are *width* and *height* which set and control respectively the width and height of an element.

# Keyboard Shortcuts

- We are now done talking about CSS.
- As a brief interlude before beginning our next major topic -- which is Javascript -- I thought I would point out a useful way to make web pages more accessible.
- This is to use the `accesskey` attribute of an element within the web page:  

```
<input type="text" name="fname" accesskey="f" />
```
- Now when ALT-f is typed the focus will go directly to this element.

# Javascript

- This language was originally developed at Netscape and first appeared in Netscape2.
- It has gone through version 1.0 to 1.5 and has been standardized as ECMA-262 and ISO-16262.
- The standardized version is sometimes called ECMAScript; the Microsoft version is sometimes called JScript.
- It can be used on both the client and on the server.
- We will use it mainly on the client except for a little bit at the end of the semester.
- Only the name is similar to Java. Unlike Java:
  - it is a loosely rather than strongly type language.
  - the object model is simpler
  - it is interpreted

# Uses of Javascript

- To do some of the processing of web pages on the client-side rather than make the server do it. For instance one might want to dynamically change the look of a page without going back to the server.
- To perform actions when certain events occur on the client-side. For instance, when a web-page is submitted it could be used to validate the page.
- To do simple networking in the background transparently to the user.
- It can be often used as a smaller alternative to a Java applet.

# Objects in Javascript

- Javascript objects are collections of properties.
- Each property is either a data property or a method property (*a method*).
- Data properties appear in two categories: primitive values and references to other objects.
- We access objects in Javascript by using variables by reference.
- All primitive values are accessed directly by value.
- There is a root object in Javascript called Object. It is the ancestor through something know as prototype inheritance of all other objects in Javascript.