

Extra Slides, week 1

CSE 190 M (Web Programming), Spring 2008
University of Washington

Reading: Chapter 1, sections 1.3 - 1.5

Except where otherwise noted, the contents of this presentation are © Copyright 2008 Marty Stepp and Jessica Miller and are licensed under the Creative Commons Attribution 2.5 License.



Additional XHTML elements

Not needed for our homework, but useful nonetheless

Definition list: <dl>, <dt>, <dd>

dl represents a list of definitions of terms (block)

dt represents each term, and *dd* its definition

```
<dl>
  <dt>newbie</dt><dd>one who does not have mad skills</dd>
  <dt>own</dt><dd>to soundly defeat
    (e.g. <q>I owned that newbie!</q>)</dd>
  <dt>frag</dt><dd>a kill in a shooting game</dd>
</dl>
```

HTML

newbie
one who does not have mad skills

own
to soundly defeat (e.g. "I owned that newbie!")

frag
a kill in a shooting game

Computer code: <code>

code: a short section of computer code (usually rendered in a fixed-width font)

```
<p>
  The <code>ul</code> and <code>ol</code>
  tags make lists.
</p>
```

HTML

The `ul` and `ol` tags make lists.

Preformatted text: `<pre>`

a large section of pre-formatted text (block)

```
<pre>
```

```
Steve Jobs speaks loudly
  reality distortion
  Apple fans bow down
```

```
</pre>
```

HTML

```
Steve Jobs speaks loudly
  reality distortion
  Apple fans bow down
```

-
- displayed with exactly the whitespace / line breaks given in the text
 - shown in a fixed-width font by default
 - how would it look if we had instead enclosed it in code tags?

Using `pre` and `code` together

```
<pre><code>
```

```
public static void main(String[] args) {
    System.out.println("Hello, world!");
}
```

```
</code></pre>
```

HTML

```
public static void main(String[] args) {
    System.out.println("Hello, world!");
}
```

-
- When showing a large section of computer code, enclose it in a `pre` to preserve whitespace and a `code` to describe the semantics of the content

Additional CSS

Embedding style sheets: `<style>`

```
<head>
  <style type="text/css">
    p { font-family: sans-serif; color: red; }
    h2 { background-color: yellow; }
  </style>
</head>
```

HTML

- CSS code can be embedded within the head of an HTML page
- this is considered *bad style* in this course (why?)
- linking to an external style sheet file is preferred, especially when you have many styles

Inline styles: the `style` attribute

```
<p style="font-family: sans-serif; color: red;">
This is a paragraph</p>
```

HTML

This is a paragraph

- higher precedence than embedded or linked styles
- used for one-time overrides and styling a particular element
- this is considered *bad style* in this course (why?)

Cascading style sheets

- it's called Cascading Style Sheets because the properties of an element *cascade* together in this order:
 - browser's default styles
 - external style sheet files (in a `<link>` tag)
 - internal style sheets (inside a `<style>` tag in the page's header)
 - inline style (the `style` attribute of the HTML element)

Inheriting styles (explanation)

```
body { font-family: sans-serif; background-color: yellow; }
p { color: red; background-color: aqua; }
a { text-decoration: overline underline; }
h2 { font-weight: bold; text-align: center; }
```

CSS

This is a heading.

A styled paragraph. [Previous slides](#) are available on the web site.

- a bullet list

-
- when multiple styles apply to an element, they are **inherited**
 - a more tightly matching rule can override a more general inherited rule
 - not all properties are inherited (notice link's color above)

Styles that conflict

```
p,h1,h2 { color: blue; font-style: italic; }
h2 { color: red; background-color: yellow; }
```

CSS

This paragraph uses the first style above.

This heading uses both styles above.

-
- when two styles set conflicting values for the same property, the latter style takes precedence