

IIST 361 - Web Development

Class 4 – Lecture notes

Anna Arzrumtsyan

Review – Common mistakes with XHTML

Information Inside the Head Element

- The elements inside the head element should not be displayed by a browser.
- According to the HTML standard, only a few tags are legal inside the head section. These are: `<base>`, `<link>`, `<meta>`, `<title>`, `<style>`, and `<script>`.
- Look at the following illegal construct:

```
<head>  
<p>This is some text</p>  
</head>
```

In this case the browser has two options:

- Display the text because it is inside a paragraph element
- Hide the text because it is inside a head element
- If you put an HTML element like `<h1>` or `<p>` inside a head element like this, most browsers will display it, even if it is illegal.
 - Should browsers forgive you for errors like this?
 - W3C doesn't think so, but Others do.

Review – Common mistakes with XHTML

- XHTML is the "new" HTML.
- The most important thing you can do is to start writing valid HTML 4.01
 - always write your tags in lower case
 - always close your tag elements
 - always include “alt” attribute for tag
 - never end a paragraph without </p>
 - never have more than one <body> tag inside your document

Designing with Style Sheets

Overview

- Page layouts
- What is CSS?
- Why use style sheets?
- Types of CSS
 - Exercise 1 – creating an embedded style
- CSS syntax
- Classes and ID
- Alignment and text positioning
- Borders and backgrounds
 - Exercise 2 – using CSS layouts

Page Layout

- Everywhere on the Web you will find pages that are formatted like newspaper pages using HTML columns.
- HTML Layout - Using Tables
 - One very common practice with HTML, is to use HTML tables to format the layout of an HTML page.
- The World Wide Web Consortium (W3C) recommends Cascading Style Sheets (CSS) to define the layout and display properties of HTML elements.

What is CSS?

- Cascading Style Sheets are a tool used to separate the *presentation* of Web page elements from the HTML or XHTML code.
- Style sheets may make subsequent editing of a Web site easier and quicker. In other instances they can also be used to enhance the browsing experience of Web site visitors.
- The many facets of Cascading Style Sheets allow a Web design extra freedom that might not be available in traditional HTML designs.

Why use CSS?

- The original HTML was **never intended** to contain tags for **formatting** a document.
- HTML tags were intended to define the **content** of the document like:

```
<p>This is a paragraph</p>  
<h1>This is a heading</h1>
```
- When tags like `` and color attributes were added to the HTML 3.2 specification, it started a **nightmare** for web developers.
- Development of large web sites where fonts and color information had to be added to every single Web page, became a long, expensive and painful process.
 - **CSS controls presentation, assures consistency, and saves time**

What is so Great about CSS?

- In HTML 4.0 all formatting can be removed from the HTML document and stored in a separate *style sheet*.
- Because HTML 4.0 separates the *presentation* from the *document structure*, we have what we always needed: total control of presentation layout without messing up the document content.
- **What Should You do About it ?**
- Do not use *presentation attributes* such as color & font-size inside your HTML tags if you can avoid it.
- Start using styles!

Style sheets & accessibility

- Your content can be made more accessible if you use Cascading Style Sheets (CSS) to style your pages.
- With CSS-styled pages, users can easily apply personalized formatting to Web documents.
 - E.g. a page designed using red text against a green background presents a problem for users with red-green color blindness: the contrast between text and background may not be great enough for the text to be distinguishable.
- If the colors are set via a style sheet, users can set their browser preferences to override your settings and can apply their own style sheet to the page instead.
- With CSS-styled pages, the user can transform Web content into a format that addresses their requirements for accessibility.
- Source: <http://www.webstyleguide.com/>

Types of CSS

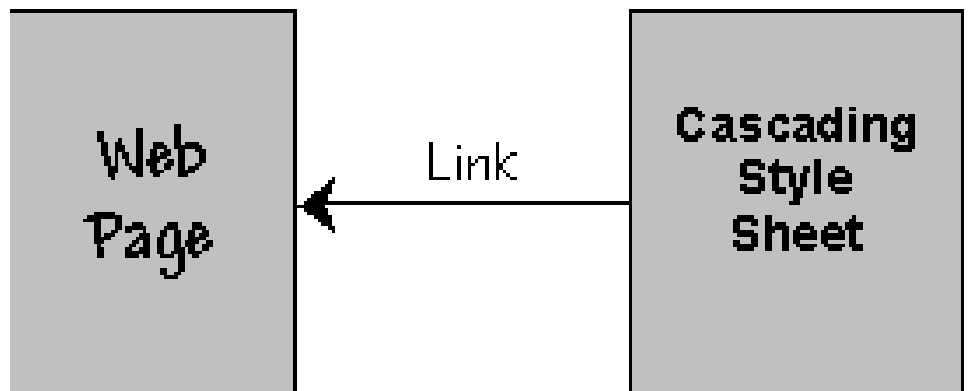
The different types of style sheets

- Linked
- Embedded
- Inline

Types of CSS - Linked

Linked style sheets feature one or more Web page files that are linked to a separate CSS document.

A linked style sheet can affect one, some, or all pages within a Web site.



Source: <http://library.albany.edu/imc/csstut/index.htm>

Types of CSS - Linked

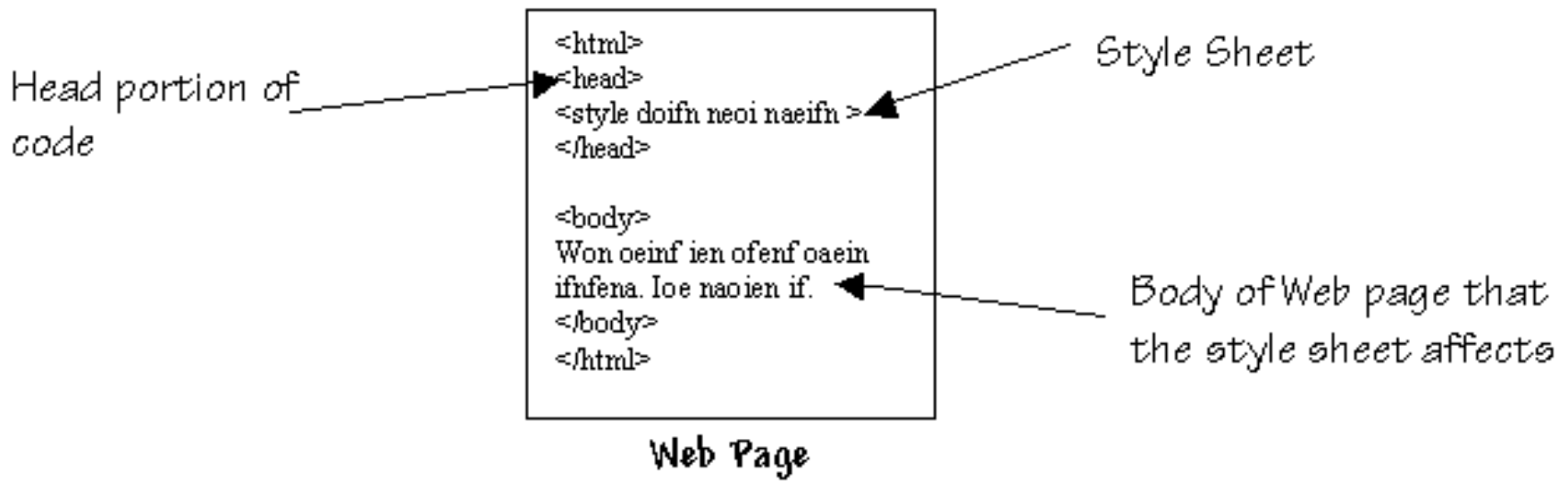
- An external (linked) style sheet is ideal when the style is applied to many pages.
- With an external style sheet, you can change the look of an entire Web site by changing one file.
- Each page must link to the style sheet using the <link> tag. The <link> tag goes inside the head section.

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

- Example of linked CSS (note, there are no tables, <div> tags)
 - <http://library.albany.edu/imc/csstut/index.htm>
 - View page source

Types of CSS - Embedded

Embedded style sheets are written within the head `<head></head>` portion of a Web page. They affect only the page in which they are written. An internal style sheet should be used when a single document has a unique style.



Source: <http://library.albany.edu/imc/csstut/index.htm>

Types of CSS - Inline

- Inline Style sheets are styles that are located directly within the code of a page at the place where they are applied.
- An inline style should be used when a unique style is to be applied to a single occurrence of an element.
- To use inline styles you use the style attribute in the relevant tag. The style attribute can contain any CSS property.

<p style="color: red; margin-left: 20px"> This is a paragraph </p>

- The example shows how to change the color and the left margin of a paragraph.

<a style-"text-decoration:none" href="about.htm">Link to our About Us page

- In this case the hyper link to a page named "about.htm" appears without underlining. But it does not affect any other link or element on the page.

How CSS work?

- The presentation styles for elements are defined in the style sheet. An element can be nearly any component on a Web page such as text, tables, and images.
 - body tag `<body></body>`
 - heading tags h1 `<h1></h1>`
 - division tags `<div></div>`.
- The collection of styles are grouped together into style sheets.
- When a browser reads the HTML code of a Web page, it "sees" the reference to the style sheet, then "goes" to the style sheet to determine how to display the element to which the style is applied.
- For example:

```
<span class="RedBold">Hello, Everyone! Welcome to our  
class today.</span>
```
- The browser examines the style sheet and displays the specified effect:
Hello, Everyone! Welcome to our class today.

CSS Rules and Components

- Each style within CSS is made of three basic parts;
 - Selector
 - Property
 - Attribute

`selector { property: attribute; }`

- The selector is the element to which the style is applied. That element can be an HTML tag, such as table <table>, or it can be something special called a "class" or an "id."
- The property is what we will specify. It might be color, size, background, or any of many other features.
- The attribute is how the property is specified. For example, color might be specified as #660066, the size as 1 em.
- An example of how a style might look is:
 - `h1 { color:#cc6600; }`
- This style says that any text inside the h1 tag must be displayed with the color cc6600. (**Which happens to be this shade of orange.**)

CSS Syntax

- A misplaced or missing bracket is often the source of a problem, and is a simple fix.
- Multiple words cannot have spaces between words
h1 { font-family: Times, "Times New Roman", serif; font-size: 12px; font-weight: bold; color: #660066; background-color: #ffffff; }
- Whenever you have specified a **color** for your font you must also specify a **background color**, and vice versa! If you leave out one or the other, the page will not validate. Most browsers will display the text correctly, but the W3C says that is against the CSS rules.
- Don't forget the **spaces!** Key to how everything goes together are two simple spaces. One space must be after the opening curly bracket, {, and the other space before the closing curly bracket }. If you leave them out, the browser will not know how to interpret the style without them, and the style will not work.

Getting started, exercise 1 – *creating an embedded style* ~ 10 minutes

- Create a new document “**css-ex1.html**” in your **ist361** folder
- Add the following code into the `<body>` of your page
`<h1>This is an example of Heading 1.</h1>`
`<h2> This is an example of Heading 2.</h2>`
- Create the style below into the `<head>` of the Web page. The code goes into the **<head>** because this is an *embedded style*.

```
<style type="text/css">
<!--
h1 {
background-color: #FFFFFF;
font-family: Arial, Helvetica, sans-serif;
font-size: 16px;
font-weight: bold;
color: #cc6600;
}
-->
</style>
```

- Add another style for h2, chose your own color, font size etc.
- Preview it in your browser to make sure the styles work

Classes and ID

- The problem with defining a standard HTML tag, such as the h1 and h2 tags you have styled so far, is that the style applies to all instances of that tag no matter where the h1 tag appears on a page. What if you need to have only certain instances of an HTML tag styled and not others?
- The problem is solved using a type of style definition called a "**class.**" Classes are specific styles that are given unique names. A class can be then applied to any HTML tag.
 - For example, you can create a class with the name ".BigTitle" that features red text displayed in bold.
 - You can use `Some Title`
 - Any HTML tag to which the class .BigTitle is applied will cause the associated text to be red & bold, in this case **Some Title**
- Note: Class names must start with a dot, "." as in **.BigTitle** You can't use spaces, but you can use hyphens or underscores .Big-Title or .Big_Title.

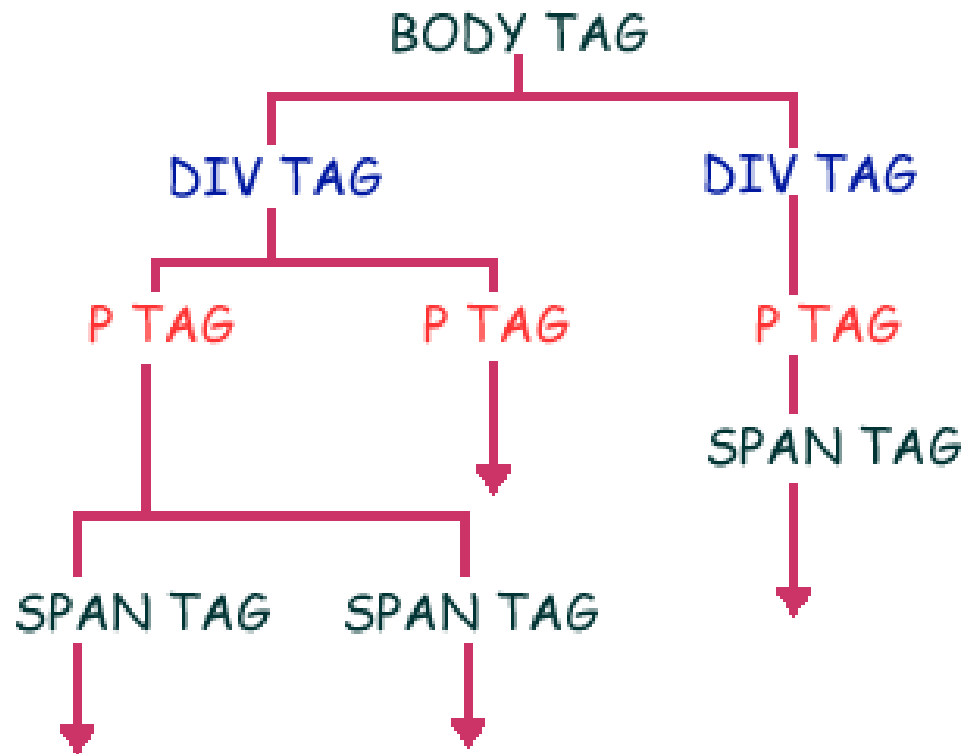
Classes and ID

- ID's are often part of certain elements of a Web page, such as images or tables. Examples:
 - ``
 - `<table width="90%" border="0" cellpadding="0" cellspacing="0" id="MainTable">`
- Where the style of MainTable is specified as

```
#MainTable {  
    color: #003300;  
    border: 1px solid #006666;  
    padding: 8px; }
```
- The "number sign" (#) before the name of the style indicates that it is for an ID attribute.

Inheritance and the Cascade

The term "cascade" in Cascading Style Sheets refers to how styles affect other styles, precedence, and inheritance.



Source: <http://library.albany.edu/imc/csstut/page7.htm>

Alignment and Text Positioning

- CSS allow you to set attributes for the way text is aligned on the page and its position.

```
.BlueText {  
  color: #000066;  
  background-color: #FFFFFF;  
  text-align: center;  
}
```

- One important attribute of the style is the **float**. It is written as : **float: right;**
- This attribute causes the text box to be positioned on the right hand side of the screen.

Borders, Backgrounds

[Border Examples](#)

Using CSS Layouts

Exercise 2 ~ 30 minutes

Floating text box – sample code

```
.ExampleBox {  
  float: right;  
  margin-left: 6px;  
  visibility: inherit;  
  z-index: auto;  
  position: relative;  
  font-family: "Times New Roman", Times, serif;  
  font-size: 14px;  
  font-style: normal;  
  color: #000000;  
  background-color: #FFFFFF;  
  vertical-align: top;  
  padding: 5px;  
  height: auto;  
  width: 200px;  
  margin-top: 6px;  
  border: 1px solid #999999;  
}
```

To include it in your html file use:

```
<div class="ExampleBox">your text goes here</div>
```

Useful links

- **Overview**
- [Introduction to CSS in Dreamweaver 8](#)
- [On Demand Seminar: Advanced Support for CSS](#)
- [Using CSS to develop big-time magazine site](#)
- **Designing with CSS—An introduction**
- [Part 1: Understanding CSS design concepts](#)
- [Part 2: Defining style properties and working with floats](#)
- [Part 3: Creating your first design without tables](#)
- [Part 4: Creating a two-column layout](#)
- [Part 5: Defining columns and vertical list navigation](#)
- [Part 6: Deciding whether to float or position columns](#)

Team 1 – meet me after class to discuss your ideas about the team project (next class).

- Next class – Fireworks

– Questions?/Problems – see me during office hours.