

CSS Layout Fundamentals

ASP.NET Lesson 2:

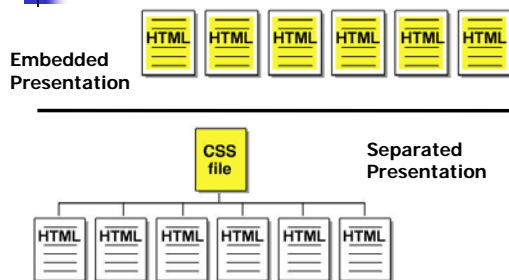
Supplemental Material:

Using HTML and CSS in ASP.NET

Separating Content from Presentation

- All web pages have two parts
 - The page content
 - The presentation of that content
- Separating these two parts will allow:
 - Presentation that is modifiable in one place
 - Code that is more accessible to many device types
 - Less code on the web page that is easier to understand and maintain

Embedded Presentation vs. Separated Presentation



External CSS File

- A website can contain any number of stylesheets
- Must be linked to the web page
- Usually contains the following types of style rules:
 - Selectors
 - IDs
 - Classes
 - Pseudo-classes

Selectors in CSS

- If you want the style of every HTML tag on your page to have a certain look or style add that selector
- Most common selectors
 - body
 - p
 - Headings: h1-h6
 - ul, ol
 - table, tr, th, td

Everybody needs some "body"

- Every external CSS file should have a body selector
- What goes in it?
 - margin for the page
 - background color
 - Default font, font-family, font-size, color
 - Default alignment (text-align)
 - Maybe a border

Selectors in a CSS file

```
body
{ margin: 5px; font-family: Arial;
  font-size: 12px; color: #006600; }

h3
{ text-align: center; color: #CCCCCC; }

ul
{ font-family: Times New Roman; color: #FF22FF;
  }
```

IDs in CSS

- An ID is used to define a style on a web page element or group of elements
 - <div id="header">
- An ID must be **unique** on a page
- IDs are usually used to define layout areas on a web page
- In the stylesheet, an ID uses the # character
 - #header { color: #CCCCCC; }

IDs in a CSS file

```
#container
{ text-align: center; width: 955px;
  margin: 0px auto; }

#header
{ background-color: #006600; padding: 5px; }

#content
{ text-align: left; background-color: #FFFFFF; }

#footer
{ text-align: center; background-color: #00CC00; }
```

Classes in CSS

- A class is a user-defined style that is applied to any element or group of elements on a page
 - <p class="bigtext"> My text </p>
- Unlike an ID, a class can be used multiple times on a web page
- In the stylesheet, classes must begin with the . character
 - For example - .bigtext { }

Classes in a CSS file

```
.centerthis
{ text-align: center; }

.verybigtext
{ font-size: xx-large; }

.redtext
{ color: #FFFFFF; }

.nounderline
{ text-decoration: none; }
```

Pseudo-Classes in CSS

- The anchor tag can have several states that can each have a style associated with it
- Pseudo-classes for default anchor tag styles should always be added to the stylesheet
- You can also add pseudo-classes to other classes or selectors

Pseudo-Classes

- The order of the pseudo-classes appear in the CSS file is important
- The a:link must always appear first
 - **a: link** – the non-visited appearance
 - **a: visited** – the visited appearance
 - **a: hover** – appearance when the mouse hovers over the link
 - **a: active** – appearance when the link is clicked

CSS file with Pseudo-Classes

```
a: link
{ color: #FFFFFF; text-decoration: none; font-size: 12px; }

a: visited
{ color: #006600; text-decoration: none; }

a: hover
{ color: #CCCCCC; text-decoration: underline; }

a: active
{ color: #2233CC; text-decoration: underline; }
```

Pseudo-Classes combined with a Class

```
a.button: link
{ color: #FFFFFF; text-decoration: none; font-size: 12px; }

a.button: visited
{ color: #006600; text-decoration: none; }

a.button: hover
{ color: #CCCCCC; text-decoration: underline; }

a.button: active
{ color: #2233CC; text-decoration: underline; }
```

Steps for a CSS based site

1. Add **unstyled** content to the web page using HTML tags
 - Remember HTML formatting like tags and tag attributes like align will hide CSS styles, so don't use them!
2. Place content between **<div>** and **</div>** containers
 - Suggested containers: surrounding container, header, content, footer
 - May also have column and/or row containers

Steps for a CSS based site

3. Create an **new** external stylesheet and give it a name and file extension of .css
 - Use the File=>New=>File menu option and choose Stylesheet
 - Drag the stylesheet filename from the Solution Explorer to the **<head>** area of your source code
 - This will automatically add the **<link>** tag

Steps for a CSS based site

4. Add the **body** selector to the external CSS file
 - Place page default styles like font-family, font-size, color in this selector
 - Add other body styles to the selector as needed: margin, background-color, text-align, etc.

Steps for a CSS based site

5. Plan and execute **layout** of <div> containers using inline or embedded CSS
6. Move CSS to external stylesheet as **IDs**
 - Remember an ID has the name preceded by the # character
7. Delete the inline or embedded CSS for the layout

Steps for a CSS based site

8. Style **HTML tags** in the source file with inline or embedded CSS
9. Move CSS to external stylesheet as XHTML tag **selectors**
 - Make sure you really want every occurrence of that HTML tag to have this style
 - If not, use a **class** rather than a selector in the stylesheet for these styles

Steps for a CSS based site

10. Delete the inline or embedded CSS used to style the HTML tags
11. Apply CSS inline styles that may be applied more than once to a group of text or multiple XHTML objects
12. Move these inline styles to the external stylesheet as **classes**
 - Remember, a class style has a name with a . preceding it

Printing Friendly Version

- Problem: Sites that have a Print Version of the page usually created a separate file that is soon outdated
- By separating the page content (HTML tags) from the presentation (styles and layout)
 - The web page will print nicely because it is just HTML tags.
 - The page content will be the same for the printed page as it is for the page displayed with the presentation

Creating Semantically Correct HTML Markup

- **Semantically** correct markup uses HTML elements for their given purpose.
- Ensures HTML has appropriate meaning for a wide range of user agents
 - Screen readers
 - Printers
 - PDAs
 - Search Engines
 - etc.

Creating Semantically Correct Markup

- You should avoid styling HTML elements to look like other HTML elements. In simple terms, this means:
 - For headings, use heading elements starting with h1 – h6
 - For paragraphs of text, use a paragraph element
 - For lists, use a list elements

"Just Because You Can Doesn't Mean You Should"

- CSS is very powerful in that you can style any HTML element to look exactly how you want
- There are some best practices that should be used when using CSS
- One place to look for "great" examples for artistically designed CSS is:
<http://www.csszengarden.com/>

Layout Best Practices

- Choose **absolute** positioning only when you want page elements to always be in the same place
- Choose **relative** position for elements in when they will follow the regular flow of the page
- Try not to mix these two!!!

Two Column Layout

- Place both columns of content in a container <div>
- Place each column in its own <div> with the left ID
- Use **float: left;** in the left ID
- The second column will float left of the first

```
<div id="container">
  <div id="left">
    Content in first column is floated to the left of everything
  </div>
  <div id="left">
    Content inside the second column is floated to the left of the first column and is longer
  </div>
</div>
<div id="footer">
</div>
```

Two Column Layout

- You may want both columns to end at the same place
- Notice one column as more text than the other
- Need to add **clear:left;** to the footer ID

```
<div id="container">
  <div id="left">
    Content in first column is floated to the left of everything
  </div>
  <div id="left">
    Content inside the second column is floated to the left of the first column and is longer
  </div>
</div>
<div id="footer">
</div>
```

CSS for IDs for Two Column Layout

```
#container
{ margin: 0px auto; width: 955px; }

#left
{ float: left; }

#footer
{ clear: left; }
```

Three Column Layout

```
<div id="container">
  <div id="left">
    Float this column to the left
  </div>
  <div id="middle">
    Unless given a fixed width, content in this column will be flexible and will fill the middle column if the window is resized
  </div>
  <div id="right">
    Float this column to the right
  </div>
</div>
```

- If you give the left and right columns a fixed width and float them left and right in the CSS file IDs.
- The middle column will be flexible in width

CSS IDs for Three Column Layout

```
#container { margin: 0px auto; }  
#left { float: left; width: 150px; }  
#right { float: right; width: 150px; }  
#middle { }  
#footer { clear: both; }
```

Practice Makes Perfect

- Use CSS to separate page content from presentation
- Do not use CSS to give HTML tags an unfamiliar appearance
- The more you use CSS the better skilled you will become
- Examine CSS used by others to extend your knowledge, i.e.
<http://www.csszengarden.com>