

# CGS 3175: Internet Applications Fall 2007

## Cascading Style Sheets (CSS) – Part 1

Instructor :     Dr. Mark Llewellyn  
                  markl@cs.ucf.edu  
                  HEC 236, 407-823-2790  
                  <http://www.cs.ucf.edu/courses/cgs3175/fall2007>

School of Electrical Engineering and Computer Science  
University of Central Florida

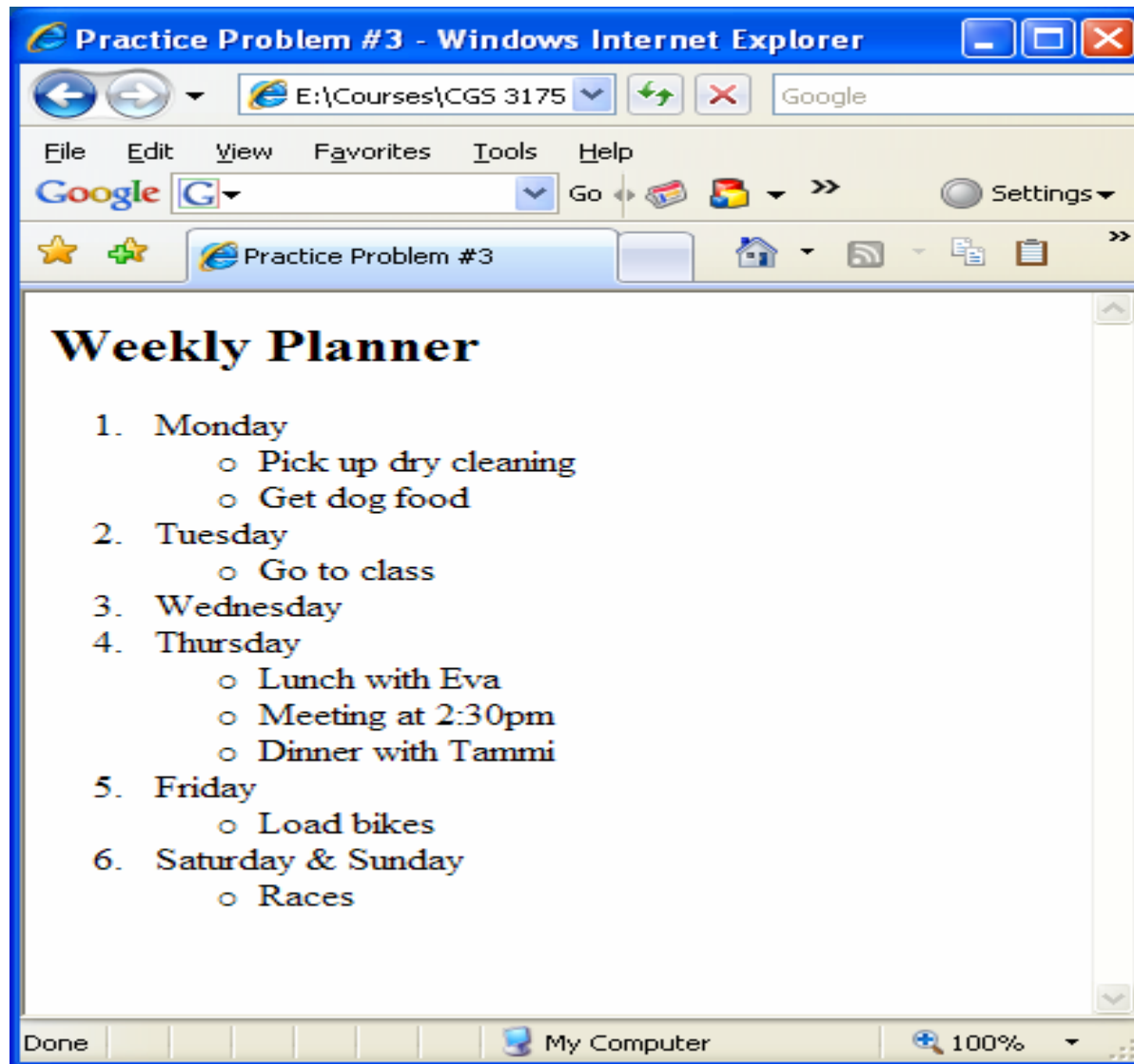


## Some Practice – From XHTML Part 3

3. Create an XHTML document when viewed with a browser will produce a page exactly like to the one shown on page 33.
4. Create an XHTML document that contains three hyperlinks to any Web pages of your choice and then verify your code at the W3C Markup Validation Web site. When viewed with a browser and you click on the links it will take you to the correct Web page. See page 34.



# Practice Problem #3 – Output



# Practice Problem #3 – The XHTML File

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Practice Problem #3 </title>
  </head>
  <body>
    <h2>Weekly Planner</h2>
    <ol>
      <li>Monday
        <ul><li>Pick up dry cleaning</li> <li>Get dog food</li> </ul>
      </li>
      <li> Tuesday
        <ul><li>Go to class</li> </ul>
      </li>
      <li> Wednesday </li>
      <li> Thursday
        <ul><li>Lunch with Eva</li><li>Meeting at 2:30pm</li><li>Dinner with Tammi</li>
        </ul>
      </li>
      <li> Friday
        <ul>
          <li>Load bikes</li>
        </ul>
      </li>
      <li> Saturday & Sunday
        <ul>
          <li>Races</li>
        </ul>
      </li>
    </ol>
  </body>
</html>
```



# Practice Problem #4 – Output

The image shows two browser windows side-by-side. The left window is displaying the CNN.com homepage, which includes a navigation menu with 'HOME', 'WORLD', 'U.S.', 'POLITICS', and 'ENTERTAINMENT'. A 'Hot Topics' section lists 'Iraq Report Card' and 'Madeleine McCann'. Below this, a 'DEVELOPING STORY' section features a photo of a man in a military uniform. The right window is displaying the W3C Markup Validation Service page, which shows a green banner stating 'This Page Is Valid XHTML 1.0 Strict!' and a table of validation details.

<b>Result:</b>	Passed validation
<b>File:</b>	E:\Courses\CGS 3175 - Internet Applications\practice problem 4.html
<b>Encoding:</b>	utf-8
<b>Doctype:</b>	XHTML 1.0 Strict



# Practice Problem #4 – The XHTML File

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Practice Problem #4</title>
  </head>
  <body>
    <h3>Here are some examples of hyperlinks in XHTML</h3>
    <p>
      <a
href="http://www.cs.ucf.edu/courses/cgs3175/fall2007/index.html">CGS 3175 -
Fall 2007 Home Page</a>
      <br /><br />
      <a href="http://www.w3c.org">W3C Home Page</a>
      <br /><br />
      <a href="http://www.cnn.com">CNN News</a>
    </p>
  </body>
</html>
```



# Cascading Style Sheets (CSS)

- In 1996 the W3C recommended the adoption of a standard set to style sheets, **Cascading Style Sheets** level 1 (CSS1).
- The original purpose of CSS was to provide HTML authors with more formatting support and give them a way to apply uniform styles to multiple documents.
- Cascading Style Sheets level 2 (CSS2), introduced in 1998, included additional features and functionality.
- CSS work with XHTML the same way they work with HTML. Web browser support is widespread with Netscape Navigator 4.5 and above, and Internet Explorer 3.0 and above all supporting CSS.



# Why Use Cascading Style Sheets?

- The primary reason for using CSS is to separate a document's content from its presentation. In so doing it provides the document author with much greater control over the document's format.
- Keeping the content and presentation information separate also allows you to change your presentation layout or method without having to modify the documents themselves, and allows you to apply one style sheet to any number of documents.
  - For example, an organization could produce a price list document and then develop different style sheets depending on the type of user who is viewing the content whether they are using a web browser or a palm pilot.





# Separating Content From Presentation

- Web development is rapidly heading toward this idea of separating content from presentation. The XML family of technologies already clearly defines the boundaries between content and presentation, as we've already discussed. XHTML Strict does not provide support for many of the strictly presentational elements, such as HTML's `<font>` element.
- The strictly presentational elements that are part of the XHTML Transitional and XHTML Frameset versions of XHTML 1.0, including the `<font>` element, are primarily included for backward compatibility with existing HTML content. Because XHTML Strict does not include many of these presentational elements in its element set, it relies on style sheets to define the presentational styles. XHTML Strict most closely represents the direction XHTML is heading. The W3C recommends that new content development should center around XHTML Strict whenever possible in order to be most compatible with future technologies.



# CSS Syntax

- CSS contain rules and declarations that instruct a program, such as a Web browser, how to display certain elements.
- There are many styles that can be applied to XHTML documents. The CSS specification is extremely large, so we will cover only a subset of its styles and declarations in this course. As you get more experience you will add more and more features from CSS to be incorporated into your Web documents.



# Defining Styles

- In order to use a style sheet with your XHTML document, you need to tell your document where to locate the style definitions. There are three ways to define styles:
  1. **Linked Style Sheets:** Style definitions in **linked style sheets** are stored in a file separate from the XHTML document.
  2. **Global Style Sheets:** Style definitions in global style sheets are stored in the XHTML document itself within the `<style>` element.
  3. **Inline Styles:** Inline styles are applied to a single element within the start tag of the element.



# Linked Style Sheets

- **Linked Style Sheets:** Style definitions in **linked style sheets** are stored in a file separate from the XHTML document. Linked style sheets provide style definitions to many documents – each document simply has to reference a single style sheet.

- The syntax for a linked style sheet declaration in an XHTML document is:

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

- The `<link>` element is used to define the style sheet. The `<link>` element is an empty element and **must** be contained within the `<head>` element of an XHTML document.



# Linked Style Sheets

- The `rel` attribute specifies this `<link>` element to be a link to a style sheet.
- The `href` attribute, like that for the `<a>` (anchor) element, specifies the location of the style sheet file on the system. Both relative and absolute URLs can be used as the value for the `href` attribute.
- The `type` attribute declares this style sheet to be a plain-text file containing CSS styles.



# Global Style Sheets

- **Global Style Sheets:** Style definitions in global style sheets are stored in the XHTML document itself within the `<style>` element. They `<style>` element **must** be contained within the `<head>` element.
- The syntax of a global style sheet is:

```
<style type="text/css"  
  
<!-- style definitions go here  
-->  
  
</style>
```

Notice that the section where the global styles are defined is contained within an XHTML comment. This is to hide the actual contents of the style definition from older browsers that don't support CSS. The comment tags are not necessary to make CSS work, but if they are not provided in an XHTML document, older Web browsers may actually display the style property definitions on the Web page!



# Inline Styles

- **Inline Styles:** Inline styles are applied to a single element within the start tag of the element.
- For example, if you wanted to assign certain properties to the text within a paragraph, you would include style definitions like the following:

```
<p style="color:red; font-family:arial">  
    paragraph text  
</p>
```



# Style Precedence

- An XHTML document can get its style information from any of the three style definition methods, or from a combination of any or all of them.
- When a style is defined in more than one place for a particular element the definition that is closest to the element itself is used.
  - For example, suppose that an XHTML document references a linked CSS file that defines style properties for the `<h1>` element, setting its font color to blue. The document then defines a global property within its `<style>` element that sets the font color for `<h1>` to red. Finally, a particular `<h1>` element within the document defines its font color to be purple by using the `style` attribute. Which style will be displayed in the browser? Will the font appear in blue, red, or purple? Answer: **purple**.





# Style Precedence

- Inline styles on particular elements override global styles defined in the `<style>` element or in linked CSS files.
- In turn, global styles defined in the `<style>` element of an XHTML document override style settings from linked CSS files.



# Style Precedence

- The ability to override styles gives developers a lot of power.
- For example, a developer could use a linked CSS file for the common formatting properties for all of the documents on a Web site. If one particular document needed special formatting properties, the developer could define global styles within that document without affecting the rest of the documents. The same is true for individual elements. A developer who wanted all of the `<h1>` elements in a document to be blue would define this in the `<style>` Element or in a linked CSS file. If one particular `<h1>` element needed red text, the developer could define the red style on that particular element, which would override the blue setting of the other `<h1>` elements.



# CSS Properties

- There are many types of CSS properties:
  1. **Font properties** define font styles such as font family or type, size, weight, and variant.
  2. **Text properties** define the layout of blocks of text, words, or characters, including spacing, alignment, transformation (forced uppercase or lowercase), and decoration (such as underline, overline, strikethrough, and blinking).
  3. **Color and image properties** define the color and background formatting of text and images. These properties can also define the position of a background image and whether the image is repeated (tiled).
  4. **Border properties** define the style of borders for elements like tables and images, as well as for the entire document. Border properties include width, height, style, color, margins, and padding.
  5. **Display properties** define the style for the structure of the document. These properties can define the placement of elements within the document, such as block or inline, and how whitespace is formatted within the document.



# Font Properties

Property	Description	Value Example(s)
<code>font</code>	Global font declaration. Can define all font properties in one property.	<code>font-family, font-style, font-weight, font-size, font-style</code>
<code>font-family</code>	Font type to display text.	<code>arial, courier</code>
<code>font-size</code>	Size of font in pixels or as a percentage.	<code>small, x-small, medium, large, x-large</code>
<code>font-style</code>	Style of font.	<code>italic, bold, oblique</code>
<code>font-variant</code>	Font rendering.	<code>normal, small-caps</code>
<code>font-weight</code>	Darkness of font. Uses name or number.	<code>normal, light, bold, bolder, 100, 200, 300, 400, etc.</code>



# Text Properties

Property	Description	Value Example(s)
word-spacing	Amount of space between words in an element.	normal, number of pixels
letter-spacing	Amount of space between letters.	normal, number of pixels
text-align	Horizontal alignment of text on page.	right, left, center
vertical-align	Vertical alignment of text on page.	baseline, sub, super, top, text-top, middle, bottom, text-bottom, percentage
text-indent	How much first line is indented.	0, number of pixels, percentage
text-transform	Change case of text.	uppercase, lowercase, capitalize, none
line-height	Amount of space between lines of text.	normal, number of pixels
text-decoration	Special controls of text appearance	underline, overline, blink, line-through, none



# Color Properties

Property	Description	Value Example(s)
color	Text color	red, blue, color code
background	Global background declaration. Can define all background properties in one property.	background-color, background-image, background-position, background-repeat, background-attachment
background-color	Color of element's background	color name, transparent, inherit
background-image	Image to be used as a background.	URL, name of local file.
background-attachment	Scrolling of background image with the element.	scroll, fixed, inherit
background-position	Position of element's background.	top, center, bottom, left, right, percentage, number of pixels
background-repeat	Repeat pattern for background image (tiling).	repeat, repeat-x, repeat-y, no-repeat



# Border Properties

Property	Description	Value Example(s)
<code>border-color</code>	Color of the border element.	red, blue, color code
<code>border-width</code>	Width of the border.	medium, thin, thick, number of pixels.
<code>border-style</code>	Style of the border.	none, solid, double
<code>margin-top</code>	Width of margin at the top of element.	0, number of pixels, percentage.
<code>margin-bottom</code>	Width of margin at the bottom of element.	0, number of pixels, percentage.
<code>margin-left</code>	Width of margin at the left side of element.	0, number of pixels, percentage.
<code>margin-right</code>	Width of margin at the right side of element.	0, number of pixels, percentage.
<code>padding-top</code>	Amount of padding at top of element.	0, number of pixels, percentage.



## Border Properties (continued)

Property	Description	Value Example(s)
padding-bottom	Amount of padding at bottom of element.	0, number of pixels, percentage.
padding-left	Amount of padding on left side of element.	0, number of pixels, percentage.
padding-right	Amount of padding on right side of element.	0, number of pixels, percentage.
clear	Whether an element permits other elements on its sides.	none, left, right
float	Floating element.	none, left, right
height	Height of an element.	auto, number of pixels, percentage
width	Width of section	auto, number of pixels, percentage





# Display Properties

Property	Description	Value Example(s)
<code>display</code>	Controls display of an element.	<code>block</code> , <code>inline</code> , <code>list-item</code>
<code>white-space</code>	Whitespace formatting.	<code>normal</code> , <code>pre</code> , <code>nowrap</code>
<code>visibility</code>	Controls visibility of element.	<code>inherit</code> , <code>visible</code> , <code>hidden</code>



# CSS Rules

- CSS rules have two parts: a **selector** and a set of **property declarations** that define the style or styles that will apply to the selector.
- The selector can contain a single element, a class/id selector, or a list of selectors. Multiple selectors are separated by commas. For styles that have more than one property defined, each property is separated by a semicolon.
- The following page shows two examples of CSS rules, the first is an example of a rule that applies to a single element, in this case, the `<h1>` element, and the second rule applies to three elements, the `<h1>` , `<h2>` , and `<p>` elements.



# CSS Rules – Examples

CSS rule applying to a single element.

```
h1 {  
    color: black;  
    font-size: 12pt;  
    font-family: arial  
}
```

CSS rule applying to a three elements.

```
h1, h2, p {  
    color: red;  
    font-size: 12pt;  
    font-family: arial  
}
```



# CSS Comments

- The syntax for comments in CSS is different than we've seen so far for XHTML documents (recall XHTML comments begin with `<!--` and end with `-->`). Comments in CSS begin with `/*` and end with `*/`. The following is an example of a CSS comments:

```
/* This is a comment in CSS */
```

```
/* Comments can also
```

```
span multiple lines */
```

- Web browsers and other processing applications ignore comments in CSS files.
- As with your XHTML documents, you should comment your CSS files.



## CSS Example

- Before we go too much further in examining CSS, let's rework one of our earlier examples to include CSS.
- The very first XHTML document we created was a course description document (see XHTML – Part 1 page 37). Let's create a CSS for this document.
- For this first example of CSS, we'll use a **linked style sheet**. Recall that this means the style definitions are stored in a file separate from the XHTML document. So let's create the CSS file and name it "firstCSS.css".



# firstCSS.css

```
/* My first Cascading Style Sheet */  
  
strong {  
    font-weight:bold;  
    text-align: left;  
    background-color: yellow;  
    text-decoration: underline;  
}  
li {  
    font-style:italic;  
    color: purple;  
}
```



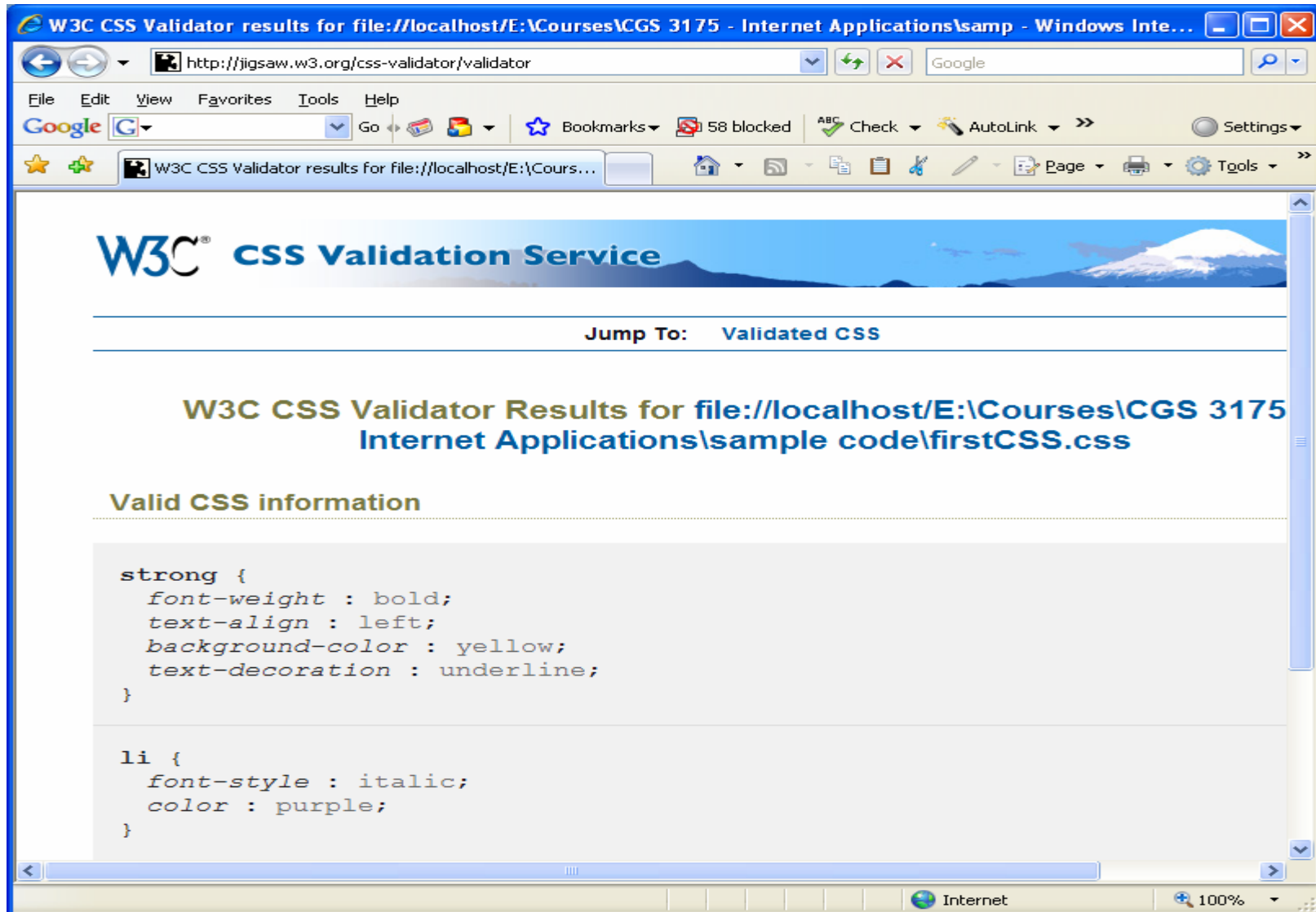
## Validating Cascading Style Sheets

- The W3C provides a tool on its Web site that will validate CSS documents, much like the XHTML validator that we discussed in the first part of the XHTML notes.
- The validator is available at:

<http://.www.jigsaw.w3.org/css-validator>.



# Validating Cascading Style Sheets



The screenshot shows a web browser window displaying the W3C CSS Validation Service. The browser's address bar shows the URL `http://jigsaw.w3.org/css-validator/validator`. The page title is "W3C CSS Validator results for file://localhost/E:\Courses\CGS 3175 - Internet Applications\samp - Windows Inte...". The browser's menu bar includes "File", "Edit", "View", "Favorites", "Tools", and "Help". The toolbar shows "Go", "Bookmarks", "58 blocked", "Check", "AutoLink", and "Settings". The main content area features the W3C logo and the text "CSS Validation Service". Below this, there is a "Jump To:" section with a link to "Validated CSS". The main heading reads "W3C CSS Validator Results for file://localhost/E:\Courses\CGS 3175 Internet Applications\sample code\firstCSS.css". Underneath, there is a section titled "Valid CSS information" which contains the following CSS code:

```
strong {  
  font-weight : bold;  
  text-align : left;  
  background-color : yellow;  
  text-decoration : underline;  
}  
  
li {  
  font-style : italic;  
  color : purple;  
}
```





# markup with css.html

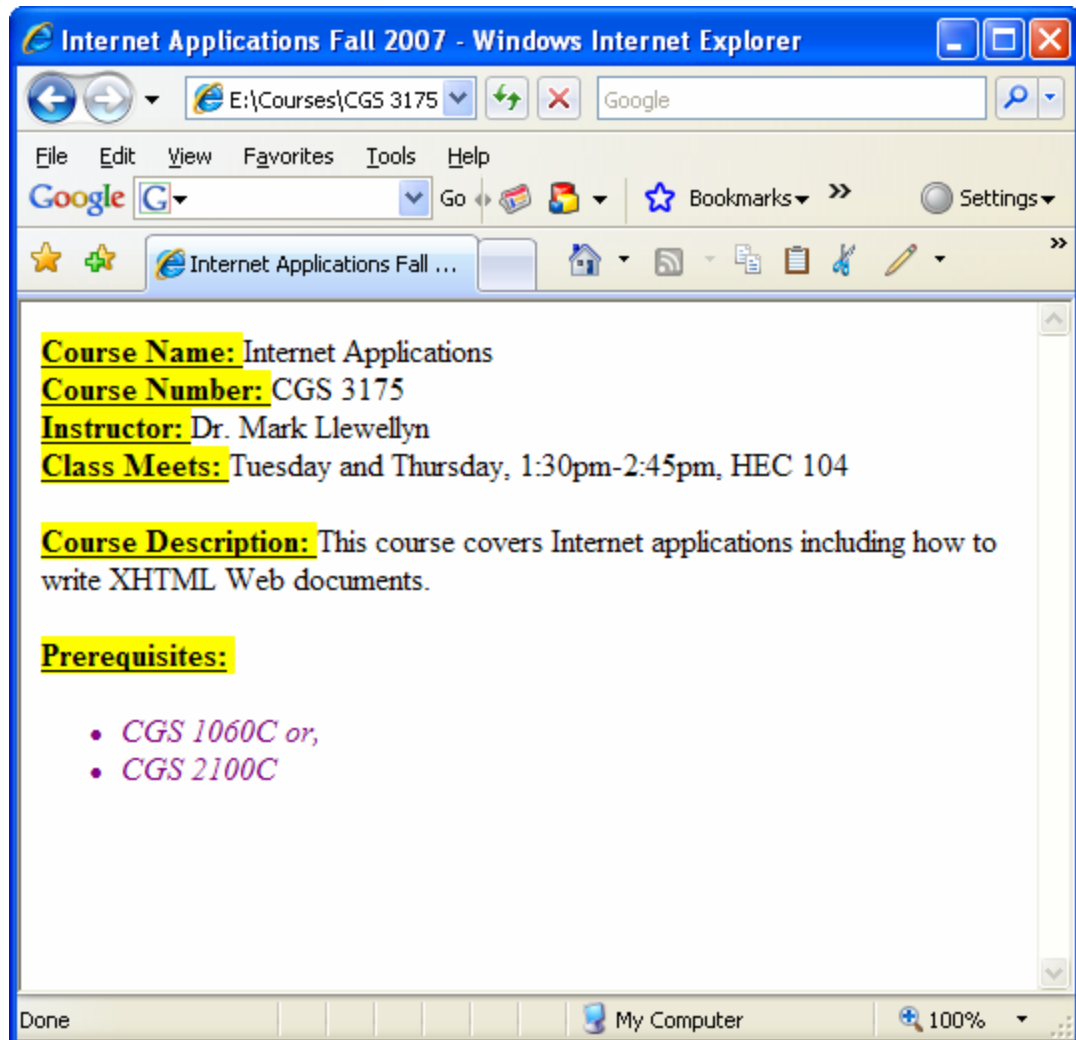
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Internet Applications Fall 2007</title>
    <link rel="stylesheet" href="firstCSS.css" type="text/css" />
  </head>
  <body>
    <strong>Course Name: </strong> Internet Applications <br />
    <strong>Course Number: </strong> CGS 3175 <br />
    <strong>Instructor: </strong> Dr. Mark Llewellyn <br />
    <strong>Class Meets: </strong> Tuesday and Thursday, 1:30pm-
2:45pm, HEC 104 <br />
    <p />
    <strong>Course Description: </strong> This course covers Internet
applications including how to write XHTML Web documents.
    <p />
    <strong>Prerequisites: </strong>
    <ul>
      <li> CGS 1060C or,</li>
      <li> CGS 2100C</li>
    </ul>
  </body>
</html>
```

This is the same file as before with the exception of the highlighted line that we added to link to the stylesheet.



# Viewing the XHTML With CSS

Just as before, load the new "markup with css.html" file in your browser and you should see the course description document with the CSS style applied, as shown.



## CSS Example 2

- Using the same XHTML file from Example 1, this time we'll use the global style sheet form, in which the style definitions are contained within the XHTML document.
- Notice that the output when viewed from a browser is no different than that when a linked style sheet was used.



# markup using global style css.html

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Internet Applications Fall 2007</title>
    <style type="text/css">
      <!--
        strong {
          font-weight:bold;
          text-align: left;
          background-color: yellow;
          text-decoration: underline;
        }
        li {
          font-style:italic;
          color: purple;
        }
      -->
    </style>
  </head>
  <body>
    <strong>Course Name: </strong> Internet Applications <br />
    <strong>Course Number: </strong> CGS 3175 <br />
    <strong>Instructor: </strong> Dr. Mark Llewellyn <br />
    <strong>Class Meets: </strong> Tuesday and Thursday, 1:30pm-2:45pm, HEC 104 <br />
    <p />
    <strong>Course Description: </strong> This course covers Internet applications including how
to write XHTML Web documents.
    <p />
    <strong>Prerequisites: </strong>
    <ul>
      <li> CGS 1060C or,</li>
      <li> CGS 2100C</li>
    </ul>
  </body>
</html>
```

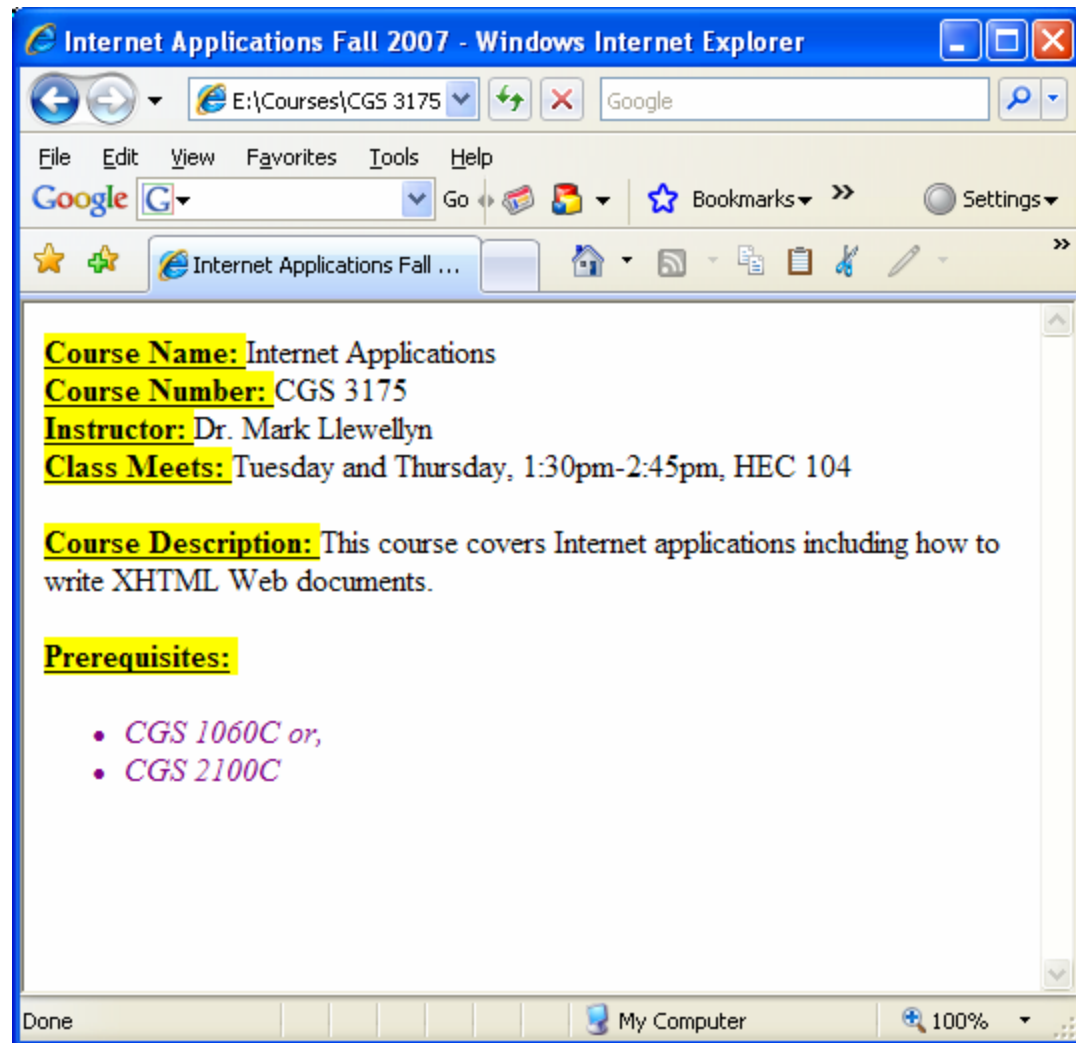
Global style sheet format – embedded in XHTML document.



# Viewing the XHTML With Global Style Format

Just as before, load the new "markup using global style.html" file in your browser and you should see the course description document with the CSS style applied, as shown.

Note that it is displayed exactly the same as was the case for the linked style sheet.



# Universal Value – secondCSS.css

```
/* My second Cascading Style Sheet */

strong {
    font-weight:bold;
    text-align: left;
    background-color: yellow;
    text-decoration: underline;
}
li {
    font-style:italic;
    color: purple;
}
/* Note: * is a universal value and applies to all elements in a
document unless it is specifically overridden by an element.
In this case, everything will be green. Note, however, that
the <li> element will override this case by making list
element text appear in purple. */

* {
    color: green;
}
```



# markup with css.html

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Internet Applications Fall 2007</title>
    <link rel="stylesheet" href="secondCSS.css" type="text/css" />
  </head>
  <body>
    <strong>Course Name: </strong> Internet Applications <br />
    <strong>Course Number: </strong> CGS 3175 <br />
    <strong>Instructor: </strong> Dr. Mark Llewellyn <br />
    <strong>Class Meets: </strong> Tuesday and Thursday, 1:30pm-
2:45pm, HEC 104 <br />
    <p />
    <strong>Course Description: </strong> This course covers Internet
applications including how to write XHTML Web documents.
    <p />
    <strong>Prerequisites: </strong>
    <ul>
      <li> CGS 1060C or,</li>
      <li> CGS 2100C</li>
    </ul>
  </body>
</html>
```

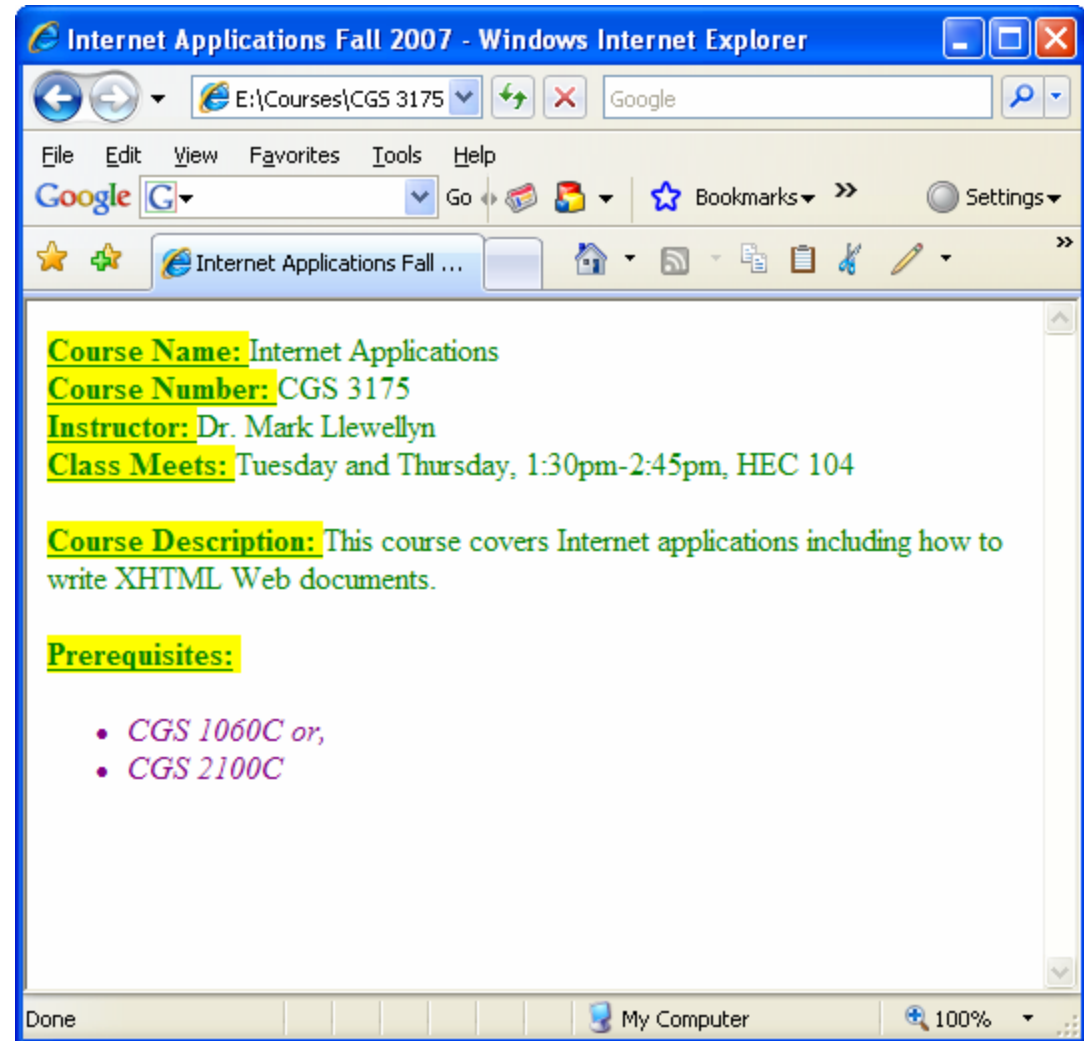
This is the same file as before with the exception of using the second CSS file we just created with the universal value.



# Viewing the XHTML With CSS

Just as before except using the style sheet "secondCSS.css:.

Note that all the text is green except for that in the list elements which was overridden by the <li> declarations in secondCSS.css.





## Some More Practice

5. Modify the ordered list example from Page 9 of XHTML – Part 3 so that it uses the “firstCSS.css” file to print the list elements in purple.
6. Modify the hyperlink example from page 22 of XHTML – Part 3, by creating a linked style sheet that will set the links to the colors below and turns off the underlining for all links. Validate your CSS document on the W3C site using the CSS validator.

unvisited links: green

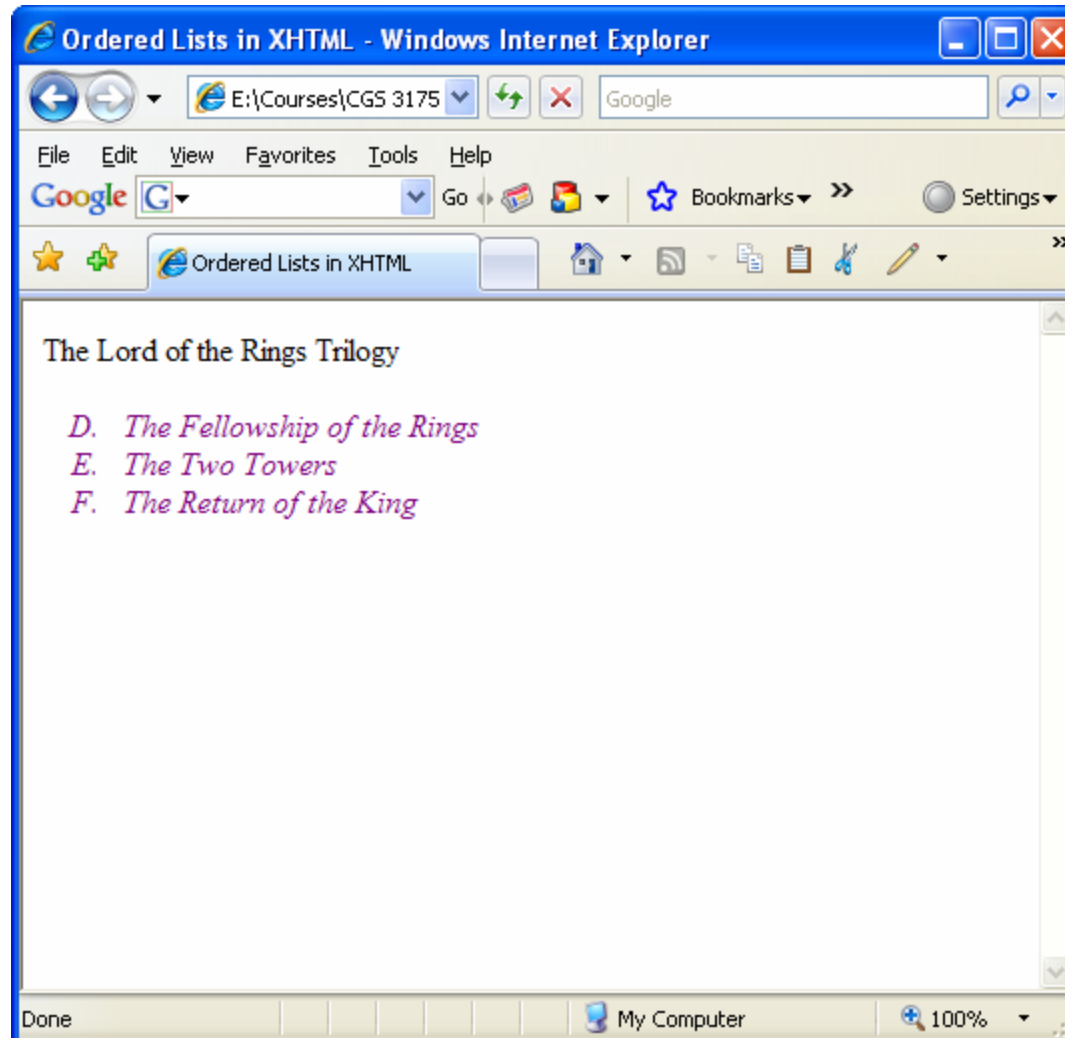
visited links: purple

active link: red

hover: blue



# Practice Problem #5 – Output



# Practice Problem #6 – Output

