

# CIS 4004: Web Based Information Technology Summer 2014

## Cascading Style Sheets (CSS)

Instructor : Dr. Mark Llewellyn  
markl@cs.ucf.edu  
HEC 236, 407-823-2790  
<http://www.cs.ucf.edu/courses/cis4004/sum2014>

Department of Electrical Engineering and Computer Science  
University of Central Florida



# 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.
- Cascading Style Sheets level 3 (CSS3), was started in 1998 and like HTML5 is still a standard in progress, but is supported (to some extent) by most current web browsers.
- CSS4 began as a W3C working draft in September 2009 and is not sufficiently developed yet to be supported by any major browser.



# Why Use Cascading Style Sheets?

- The primary reason for using CSS is to separate a document's content and structure 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 desktop browser or a mobile device.



# 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 did not provide support for many of the strictly presentational elements, such as HTML's `<font>` element.
- Like XHTML Strict, HTML5 was designed to specify the content and structure of a document. Though HTML5 contains some attributes that control presentation (as did XHTML Strict), it is best to separate the content and presentation. If a webpage's presentation is determined entirely by a stylesheet, you can simply change the stylesheet to completely change the page's appearance.



# 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 HTML5 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 HTML5 document, you need to tell your document where to locate the style definitions. There are three ways to define styles:
  1. **Linked (External) Style Sheets:** Style definitions in **linked style sheets** are stored in a file separate from the HTML5 document.
  2. **Global (Embedded or Internal) Style Sheets:** Style definitions in are stored in the HTML5 document **global style sheet** itself within the `<style>` element in the `<head>` of the document.
  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 HTML5 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 HTML5 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 HTML5 document.



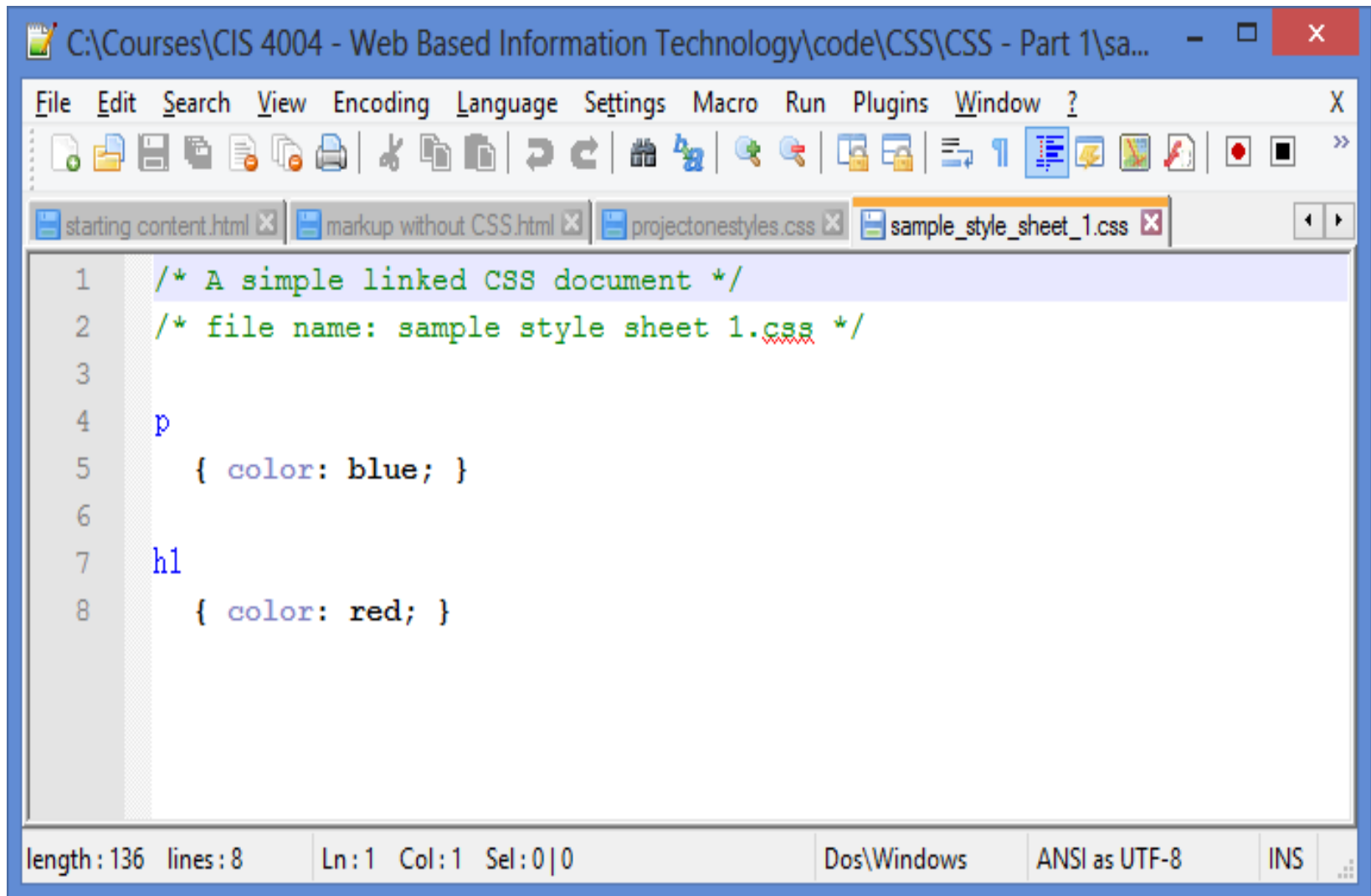
# Linked Style Sheets

- 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.





# Linked Style Sheet - Example



The image shows a screenshot of a text editor window. The title bar indicates the file path: C:\Courses\CIS 4004 - Web Based Information Technology\code\CSS\CSS - Part 1\sa... The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Macro, Run, Plugins, and Window. The toolbar contains various icons for file operations. The tab bar shows four open files: starting content.html, markup without CSS.html, projectonestyles.css, and sample\_style\_sheet\_1.css. The main text area contains the following CSS code:

```
1  /* A simple linked CSS document */
2  /* file name: sample style sheet 1.css */
3
4  p
5    { color: blue; }
6
7  h1
8    { color: red; }
```

The status bar at the bottom shows: length: 136 lines: 8 Ln: 1 Col: 1 Sel: 0|0 Dos\Windows ANSI as UTF-8 INS



# Global Style Sheets

- **Global Style Sheets:** Style definitions in global style sheets are stored in the HTML5 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
```

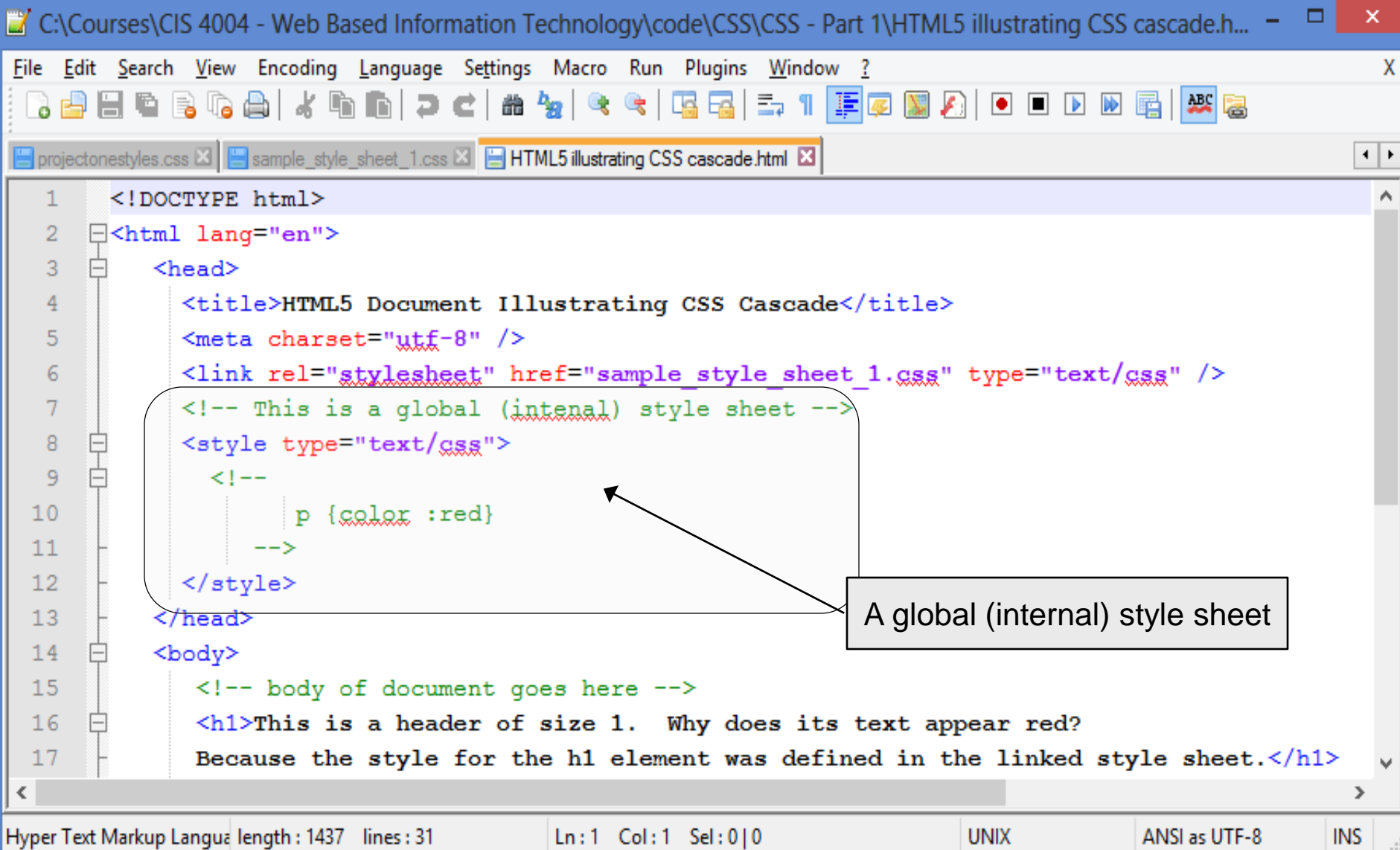
```
-->
```

```
</style>
```

Notice that the section where the global styles are defined is contained within an HTML5 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!



# Global Style Sheet - Example



The screenshot shows a web browser window with the following HTML code:

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <title>HTML5 Document Illustrating CSS Cascade</title>
5     <meta charset="utf-8" />
6     <link rel="stylesheet" href="sample_style_sheet_1.css" type="text/css" />
7     <!-- This is a global (internal) style sheet -->
8     <style type="text/css">
9       <!--
10         p {color :red}
11       -->
12     </style>
13   </head>
14   <body>
15     <!-- body of document goes here -->
16     <h1>This is a header of size 1. Why does its text appear red?
17     Because the style for the h1 element was defined in the linked style sheet.</h1>
```

A callout box with an arrow points to the internal style sheet code block, containing the text: "A global (internal) style sheet".

Hyper Text Markup Language | length: 1437 | lines: 31 | Ln: 1 | Col: 1 | Sel: 0 | 0 | UNIX | ANSI as UTF-8 | INS



# 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;">
```

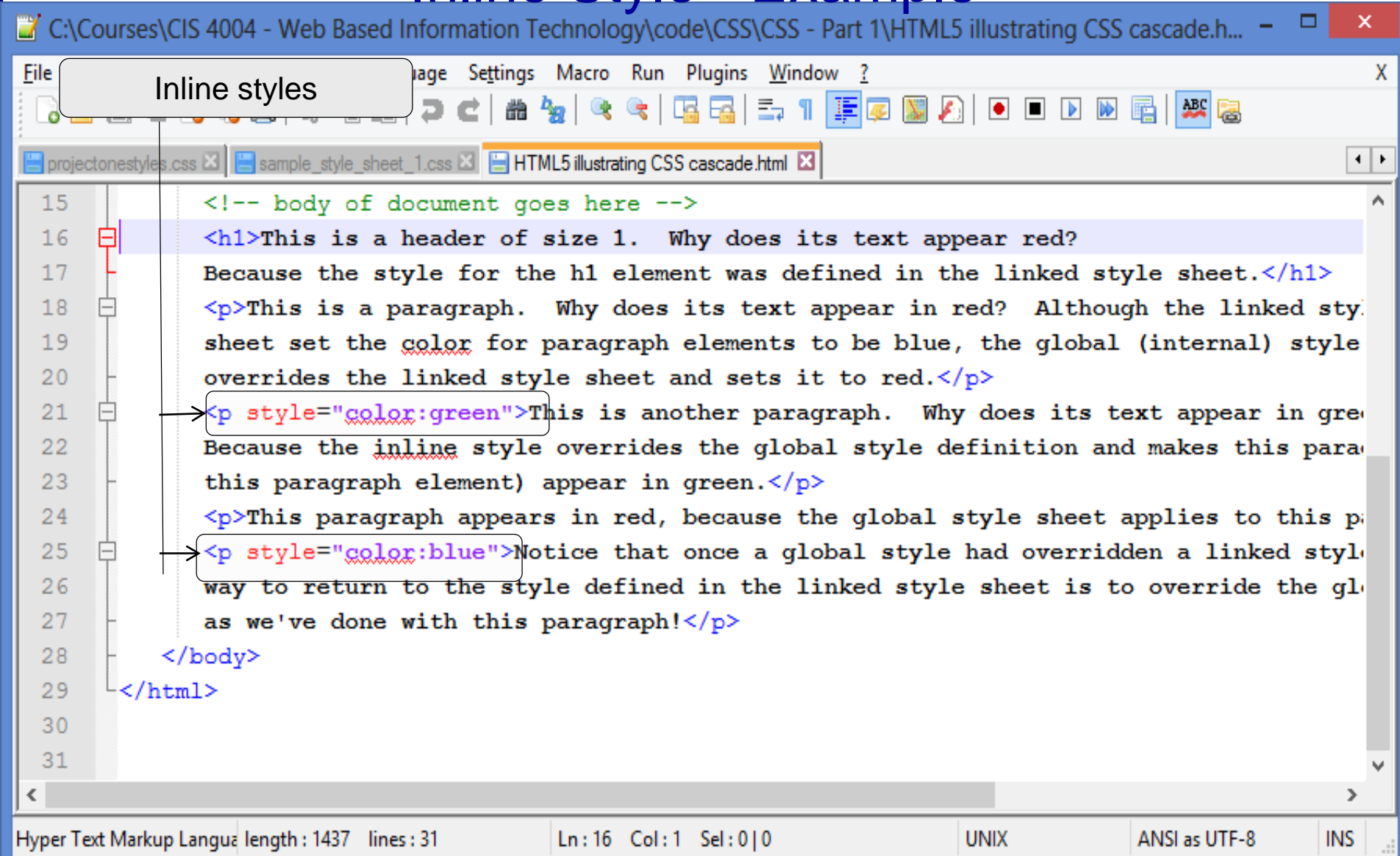
```
    paragraph text
```

```
</p>
```

- We've already used this style of CSS in a couple of our earlier examples.



# Inline Style - Example



The screenshot shows a web browser window with the following content:

```
15 <!-- body of document goes here -->
16 <h1>This is a header of size 1. Why does its text appear red?
17 Because the style for the h1 element was defined in the linked style sheet.</h1>
18 <p>This is a paragraph. Why does its text appear in red? Although the linked sty:
19 sheet set the color for paragraph elements to be blue, the global (internal) style
20 overrides the linked style sheet and sets it to red.</p>
21 <p style="color:green">This is another paragraph. Why does its text appear in gre
22 Because the inline style overrides the global style definition and makes this para
23 this paragraph element) appear in green.</p>
24 <p>This paragraph appears in red, because the global style sheet applies to this p:
25 <p style="color:blue">Notice that once a global style had overridden a linked styl
26 way to return to the style defined in the linked style sheet is to override the gl
27 as we've done with this paragraph!</p>
28 </body>
29 </html>
```

The browser's status bar at the bottom indicates: Hyper Text Markup Language length: 1437 lines: 31 Ln: 16 Col: 1 Sel: 0|0 UNIX ANSI as UTF-8 INS



# Style Precedence – The Cascade

- An HTML5 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 HTML5 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**.

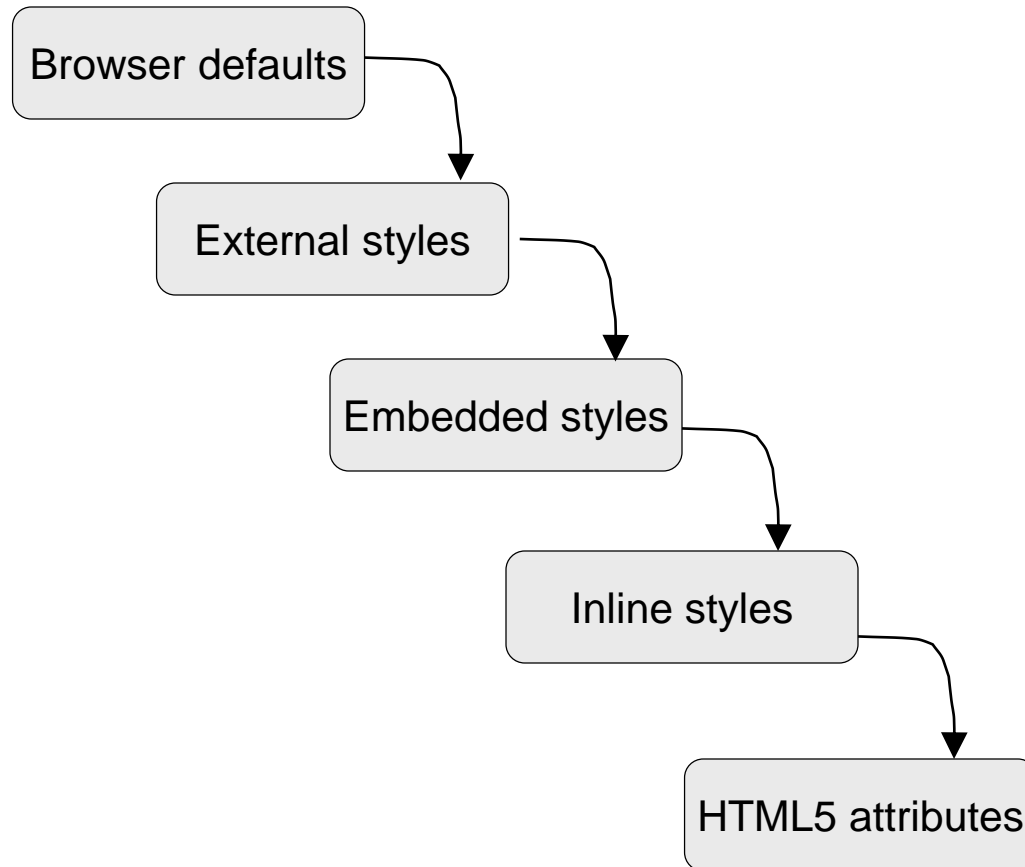


# Style Precedence – The Cascade

- 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 HTML document override style settings from linked CSS files.
- This is the “**cascade**” in CSS!

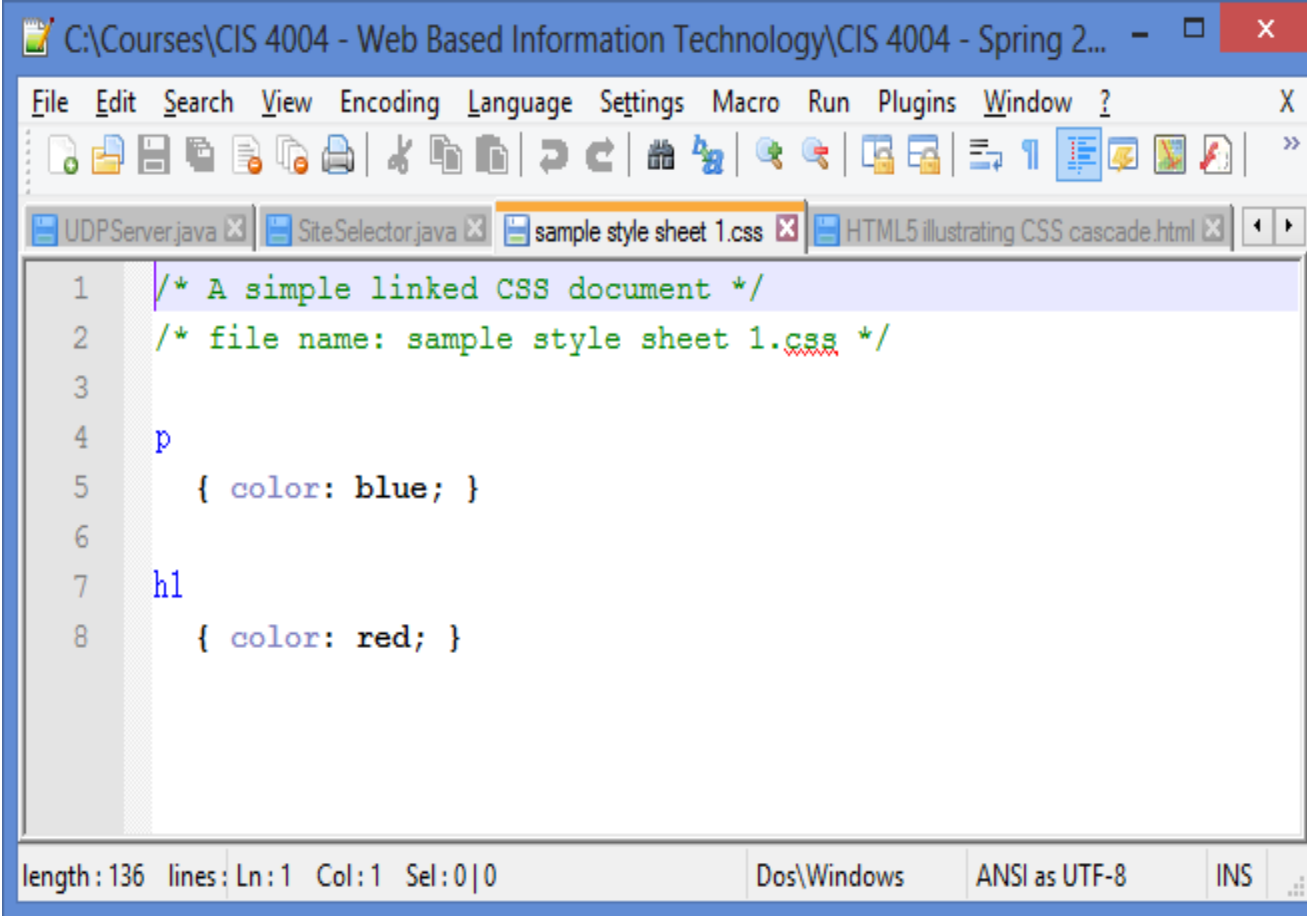


# Style Precedence – The Cascade





# Style Precedence – Illustration of the Cascade



```
1  /* A simple linked CSS document */
2  /* file name: sample style sheet 1.css */
3
4  p
5    { color: blue; }
6
7  h1
8    { color: red; }
```

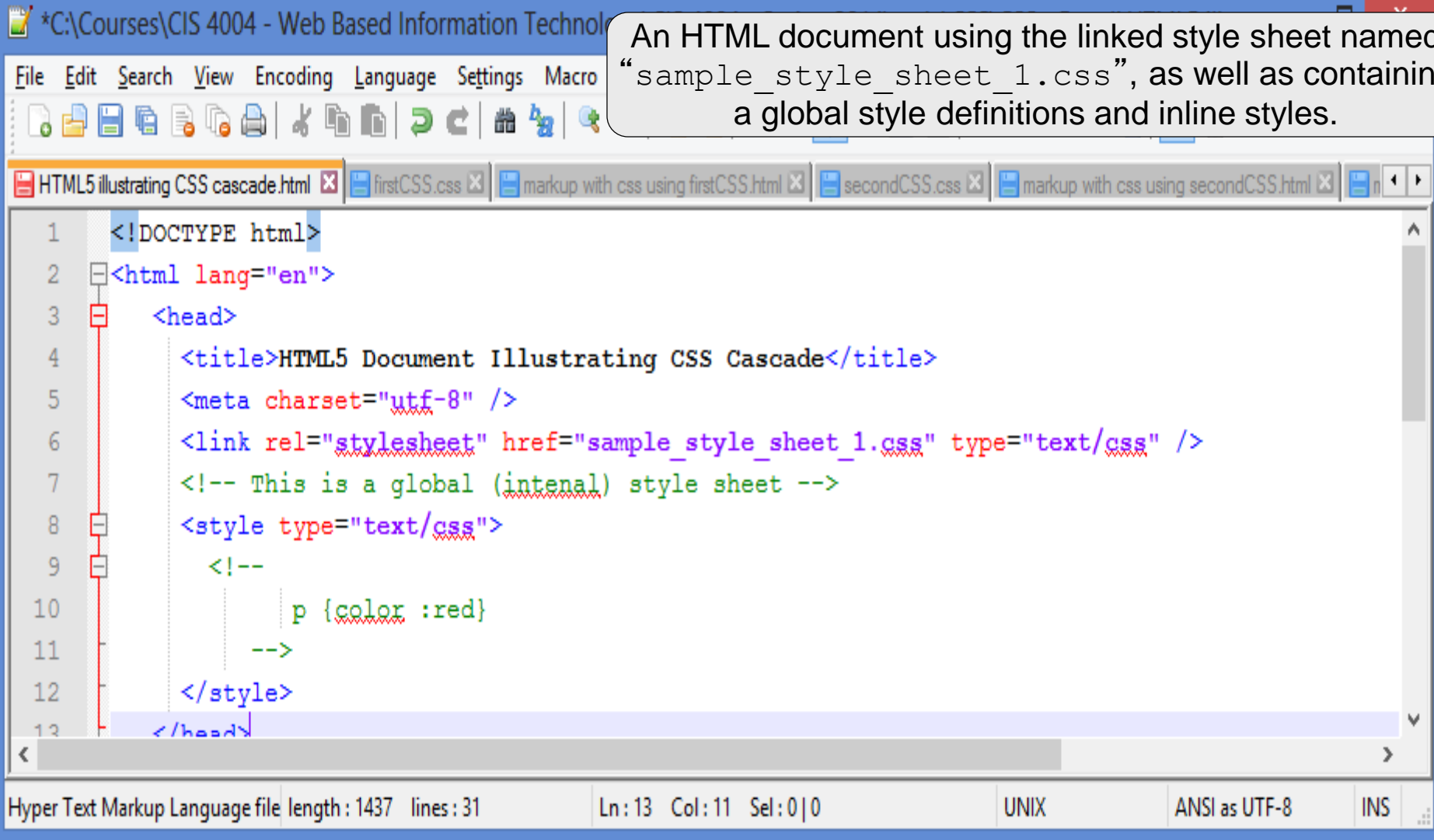
length: 136 lines: Ln: 1 Col: 1 Sel: 0|0    Dos\Windows    ANSI as UTF-8    INS

A linked style sheet named  
“sample style sheet 1.css”



# Style Precedence – Illustration of the Cascade

An HTML document using the linked style sheet named "sample\_style\_sheet\_1.css", as well as containing a global style definitions and inline styles.



The screenshot shows a Notepad++ window with the following content:

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <title>HTML5 Document Illustrating CSS Cascade</title>
5     <meta charset="utf-8" />
6     <link rel="stylesheet" href="sample_style_sheet_1.css" type="text/css" />
7     <!-- This is a global (intenal) style sheet -->
8     <style type="text/css">
9       <!--
10          p {color :red}
11        -->
12     </style>
13 </head>
```

The status bar at the bottom indicates: Hyper Text Markup Language file length : 1437 lines : 31 Ln: 13 Col: 11 Sel: 0|0 UNIX ANSI as UTF-8 INS



## Style Precedence – Illustration of the Cascade

**This is a header of size 1. Why does its text appear red? Because the style for the h1 element was defined in the linked style sheet.**

This is a paragraph. Why does its text appear in red? Although the linked style sheet set the color for paragraph elements to be blue, the global (internal) style sheet overrides the linked style sheet and sets it to red.

This is another paragraph. Why does its text appear in green? Because the inline style overrides the global style definition and makes this paragraph element (and only this paragraph element) appear in green.

This paragraph appears in red, because the global style sheet applies to this paragraph.

Notice that once a global style had overridden a linked style for a given element the only way to return to the style defined in the linked style sheet is to override the global style using an inline style as we've done with this paragraph!



# 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 (i.e. 10 px), percentage (i.e. 10%)
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)
<code>color</code>	Text color	<code>red, blue, color code</code>
<code>background</code>	Global background declaration. Can define all background properties in one property.	<code>background-color, background-image, background-position, background-repeat, background-attachment</code>
<code>background-color</code>	Color of element's background	<code>color name, transparent, inherit</code>
<code>background-image</code>	Image to be used as a background.	URL, name of local file.
<code>background-attachment</code>	Scrolling of background image with the element.	<code>scroll, fixed, inherit</code>
<code>background-position</code>	Position of element's background.	<code>top, center, bottom, left, right, percentage, number of pixels</code>
<code>background-repeat</code>	Repeat pattern for background image (tiling).	<code>repeat, repeat-x, repeat-y, no-repeat</code>





# 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)
<code>padding-bottom</code>	Amount of padding at bottom of element.	0, number of pixels, percentage.
<code>padding-left</code>	Amount of padding on left side of element.	0, number of pixels, percentage.
<code>padding-right</code>	Amount of padding on right side of element.	0, number of pixels, percentage.
<code>clear</code>	Whether an element permits other elements on its sides.	<code>none</code> , <code>left</code> , <code>right</code>
<code>float</code>	Floating element.	<code>none</code> , <code>left</code> , <code>right</code>
<code>height</code>	Height of an element.	<code>auto</code> , number of pixels, percentage
<code>width</code>	Width of section	<code>auto</code> , number of pixels, percentage



# Display Properties

Property	Description	Value Example(s)
<code>display</code>	Controls display of an element.	<code>block, inline, list-item</code>
<code>white-space</code>	Whitespace formatting.	<code>normal, pre, nowrap</code>
<code>visibility</code>	Controls visibility of element.	<code>inherit, visible, 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 HTML5 documents (recall HTML5 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 HTML5 documents, you should comment your CSS files.

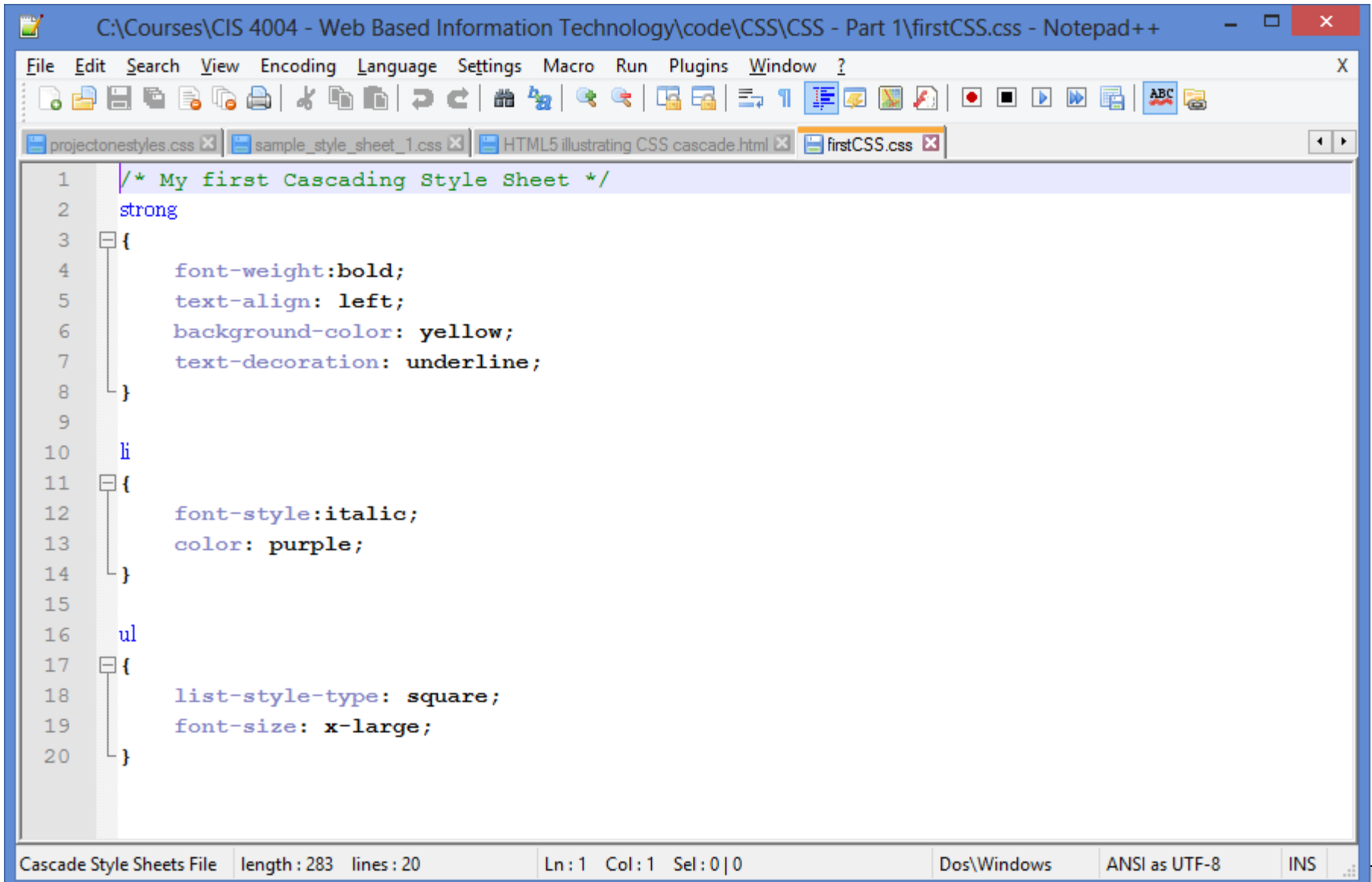


# CSS Example

- Before we go too much further in examining CSS, let's include some CSS into a document.
- For this 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 HTML document. So let's create the CSS file and name it `“firstCSS.css”`.



# firstCSS.css



```
1  /* My first Cascading Style Sheet */
2  strong
3  {
4      font-weight:bold;
5      text-align: left;
6      background-color: yellow;
7      text-decoration: underline;
8  }
9
10 li
11 {
12     font-style:italic;
13     color: purple;
14 }
15
16 ul
17 {
18     list-style-type: square;
19     font-size: x-large;
20 }
```

Cascade Style Sheets File | length : 283 | lines : 20 | Ln : 1 | Col : 1 | Sel : 0 | 0 | Dos\Windows | ANSI as UTF-8 | INS





# Validating Cascading Style Sheets

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

[http://.www.jigsaw.w3.org/css-validator.](http://.www.jigsaw.w3.org/css-validator)





## The W3C CSS Validation Service

W3C CSS Validator results for firstCSS.css (CSS level 3)

Jump to: [Validated CSS](#)

### W3C CSS Validator results for firstCSS.css (CSS level 3)

#### Congratulations! No Error Found.

This document validates as [CSS level 3](#) !

To show your readers that you've taken the care to create an interoperable Web page, you may display this icon on any page that validates. Here is the XHTML you could use to add this icon to your Web page:



```
<p>
  <a href="http://jigsaw.w3.org/css-validator/check/referer">
    
  </a>
</p>
```



```
<p>
  <a href="http://jigsaw.w3.org/css-validator/check/referer">
    
  </a>
</p>
```



# markup with css.html

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

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <title>Web Based Information Technology Summer 2014</title>
5     <meta charset="utf-8" />
6     <link rel="stylesheet" href="firstCSS.css" type="text/css" />
7   </head>
8   <body>
9     <div>
10      <div>
11        <strong>Course Name: </strong> Web Based Information Technology<br />
12        <strong>Course Number: </strong> CIS 4004 <br />
13        <strong>Instructor: </strong> Dr. Mark Llewellyn <br />
14        <strong>Class Meets: </strong> Monday and Wednesday, 6:00-7:50pm, HEC 110 <br />
15      </div>
16      <strong>Course Description: </strong> HTML5, CSS, JavaScript, AJAX, Digital libr:
17      Mobile Internet and WML, Emerging technologies. Capacity planning for web servi
18    </p>
19    <p>
20      <strong>Prerequisites: </strong>
21    </p>
22    <ul>
23      <li> CNT 3004 and,</li>
24      <li> COP 3330</li>
25    </ul>
26  </div>
27 </body>
28 </html>
```



# Viewing the HTML With CSS

The screenshot shows a web browser window with the title "Web Based Information Technology Summer 2014". The address bar shows the file path: `file:///C:/COURSES/CIS%204004%20-%20Web%20Based%20Information%20Technology/`. The page content is as follows:

**Course Name:** Web Based Information Technology  
**Course Number:** CIS 4004  
**Instructor:** Dr. Mark Llewellyn  
**Class Meets:** Monday and Wednesday, 6:00-7:50pm, HEC 110

**Course Description:** HTML5, CSS, JavaScript, AJAX, Digital libraries, Media formats, Compression, Streaming Media, Mobile Internet and WML, Emerging technologies. Capacity planning for web services.

**Prerequisites:**

- *CNT 3004 and,*
- *COP 3330*

A callout box titled "Original version without CSS" shows the same content but with a different layout and styling, including a different "Class Meets" time and location: "Tuesday and Thursday, 3:00-4:15pm, CB2 206".

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

- Before we go any further with learning CSS. I want to illustrate how CSS can be used to modify the presentation of an HTML5 document without changing the HTML5 markup. Using the same HTML5 file from Example 1, we'll make another linked style sheet, this time called `secondCSS.css`. We'll change the way the various HTML5 elements that appear in our document are presented by using different CSS rules to style the same document.



```
1  /* My second Cascading Style Sheet */
2  /* file name:  secondCSS.css */
3  strong
4  {
5      font-weight:bold;
6      text-align: right;
7      text-decoration: underline;
8      color: blue;
9  }
10
11  li
12  {
13      font-style:italic;
14      font-size: 14pt;
15      color: green;
16  }
17
18  ul
19  {
20      list-style-type: circle;
21      font-family: arial;
22      font-variant: normal;
23      text-transform: lowercase;
24  }
```



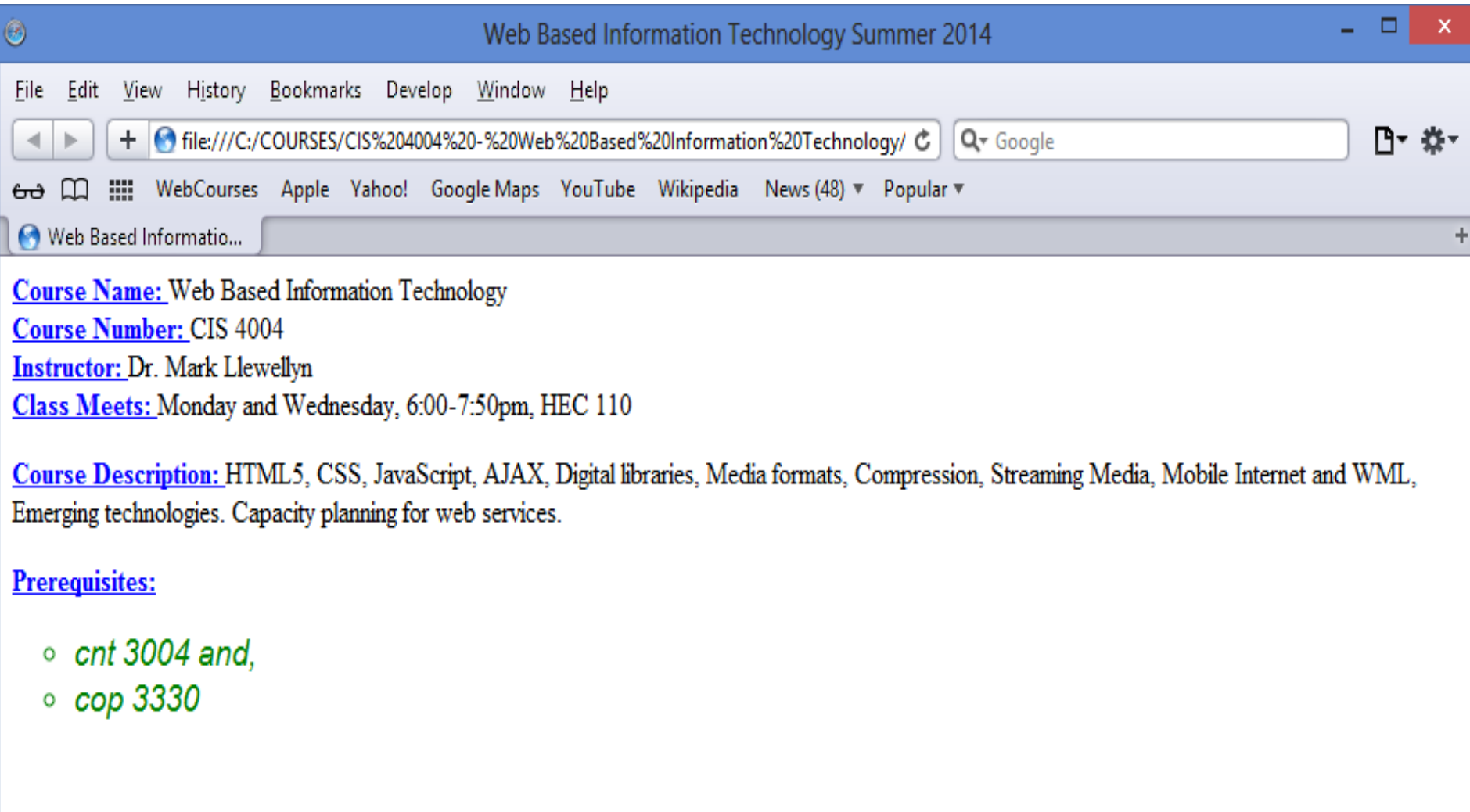
# Modified markup with css.html

This is the same file as before with the exception of the highlighted line that must be changed to link the new stylesheet.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <title>Web Based Information Technology Summer 2014</title>
5 <meta charset="utf-8" />
6 <link rel="stylesheet" href="secondCSS.css" type="text/css" />
7 </head>
8 <body>
9 <div>
10 <strong>Course Name: </strong> Web Based Information Technology<br />
11 <strong>Course Number: </strong> CIS 4004 <br />
12 <strong>Instructor: </strong> Dr. Mark Llewellyn <br />
13 <strong>Class Meets: </strong> Monday and Wednesday, 6:00-7:50pm, HEC 110 <br />
14 <p>
15 <strong>Course Description: </strong> HTML5, CSS, JavaScript, AJAX, Digital libr:
16 Mobile Internet and WML, Emerging technologies. Capacity planning for web servi:
17 </p>
18 <p>
19 <strong>Prerequisites: </strong>
20 </p>
21 <ul>
22 <li> CNT 3004 and,</li>
23 <li> COP 3330</li>
24 </ul>
25 </div>
```



# Rendering of modified version of HTML using `secondCSS.css` as the style sheet



The screenshot shows a web browser window titled "Web Based Information Technology Summer 2014". The address bar contains the file path: `file:///C:/COURSES/CIS%204004%20-%20Web%20Based%20Information%20Technology/`. The browser interface includes a menu bar (File, Edit, View, History, Bookmarks, Develop, Window, Help), a search bar with "Google", and a toolbar with various icons. The main content area displays the following course information:

**Course Name:** Web Based Information Technology  
**Course Number:** CIS 4004  
**Instructor:** Dr. Mark Llewellyn  
**Class Meets:** Monday and Wednesday, 6:00-7:50pm, HEC 110

**Course Description:** HTML5, CSS, JavaScript, AJAX, Digital libraries, Media formats, Compression, Streaming Media, Mobile Internet and WML, Emerging technologies. Capacity planning for web services.

**Prerequisites:**

- *cnt 3004 and,*
- *cop 3330*





## CSS Example 3

- Using the same HTML5 file from Example 2, this time we'll use the global style sheet form, in which the style definitions are contained within the HTML5 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

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <title>Web Based Information Technology Summer 2014</title>
5 <style type="text/css">
6 <!--
7     strong {
8         font-weight:bold;
9         text-align: right;
10        text-decoration: underline;
11        color: blue;
12    }
13    li {
14        font-style:italic;
15        font-size: 14pt;
16        color: green;
17    }
18    ul {
19        list-style-type: circle;
20        font-family: arial;
21        font-variant: normal;
22        text-transform: lowercase;
23    }
24    -->
25 </style>
26 </head>
27 <body>
28 <div>
```

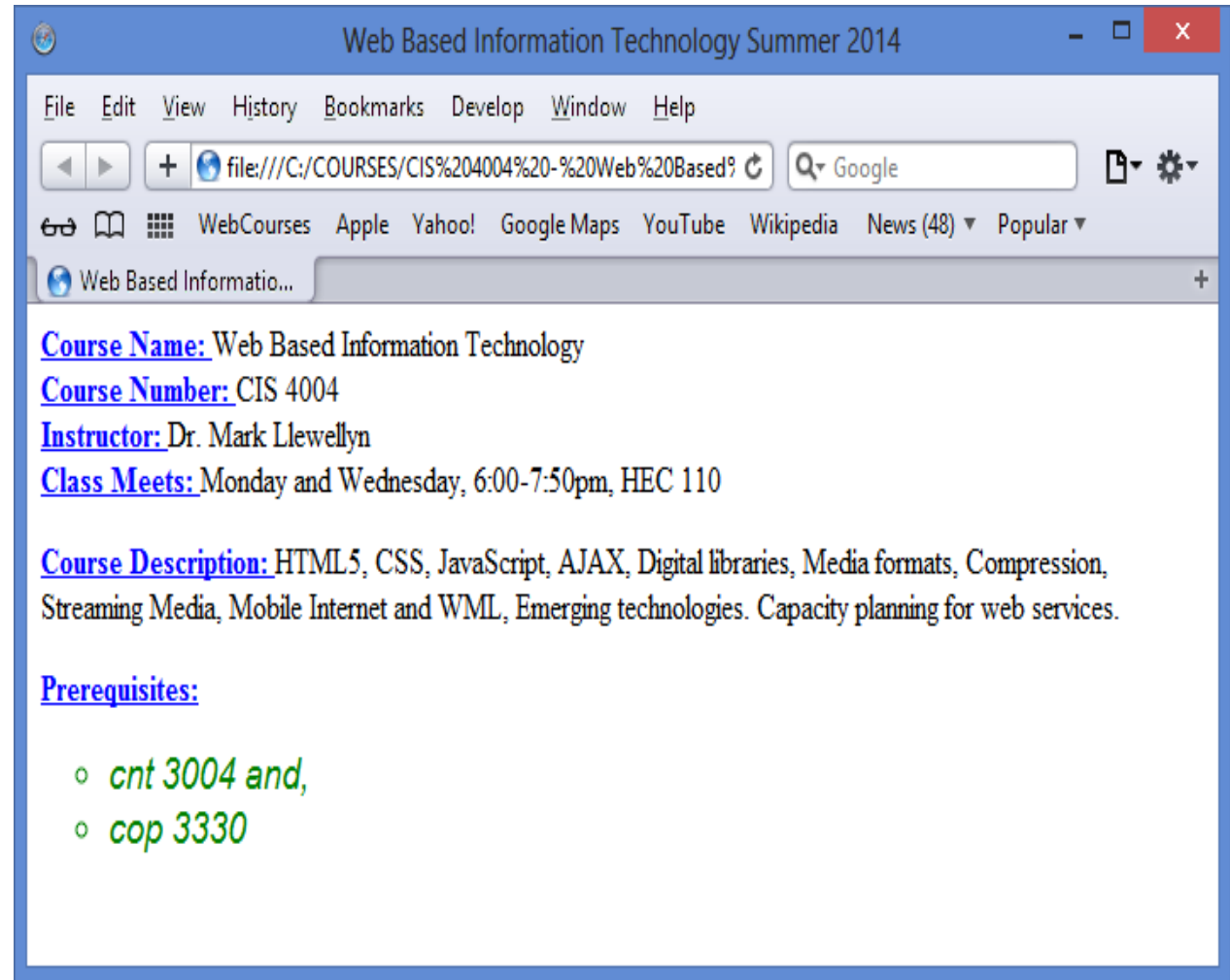
Global style sheet format – embedded in HTML5 document.



# Viewing the HTML5 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.



## Using A Universal Value - thirdCSS.css

```
1  /* My third Cascading Style Sheet */
2  /* Illustrates a universal value */
3  /* file name: thirdCSS.css */
4
5  strong {
6      font-weight:bold;
7      color: red;
8  }
9
10 li {
11     font-style:normal;
12     color: purple;
13 }
14 /* Note: * is a universal value and applies to all elements in a document
15 unless it is specifically overridden by an element. In this case,
16 everything will be green. Note, however, that the <li> element will
17 override this case by making list element text appear in purple. */
18
19 * {
20     color: green;
21 }
```



# markup with css.html

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

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <title>Web Based Information Technology Summer 2014</title>
5     <meta charset="utf-8" />
6     <link rel="stylesheet" href="thirdCSS.css" type="text/css" />
7   </head>
8   <body>
9     <div>
10      <strong>Course Name: </strong> Web Based Information Technology<br />
11      <strong>Course Number: </strong> CIS 4004 <br />
12      <strong>Instructor: </strong> Dr. Mark Llewellyn <br />
13      <strong>Class Meets: </strong> Monday and Wednesday, 6:00-7:50pm, HEC 110 <br />
14      <p>
15        <strong>Course Description: </strong> HTML5, CSS, JavaScript, AJAX, Digital libr:
16        Mobile Internet and WML, Emerging technologies. Capacity planning for web servi.
17      </p>
18      <p>
19        <strong>Prerequisites: </strong>
20      </p>
21      <ul>
22        <li> CNT 3004 and,</li>
23        <li> COP 3330</li>
24      </ul>
25    </div>
26  </body>
27 </html>
```

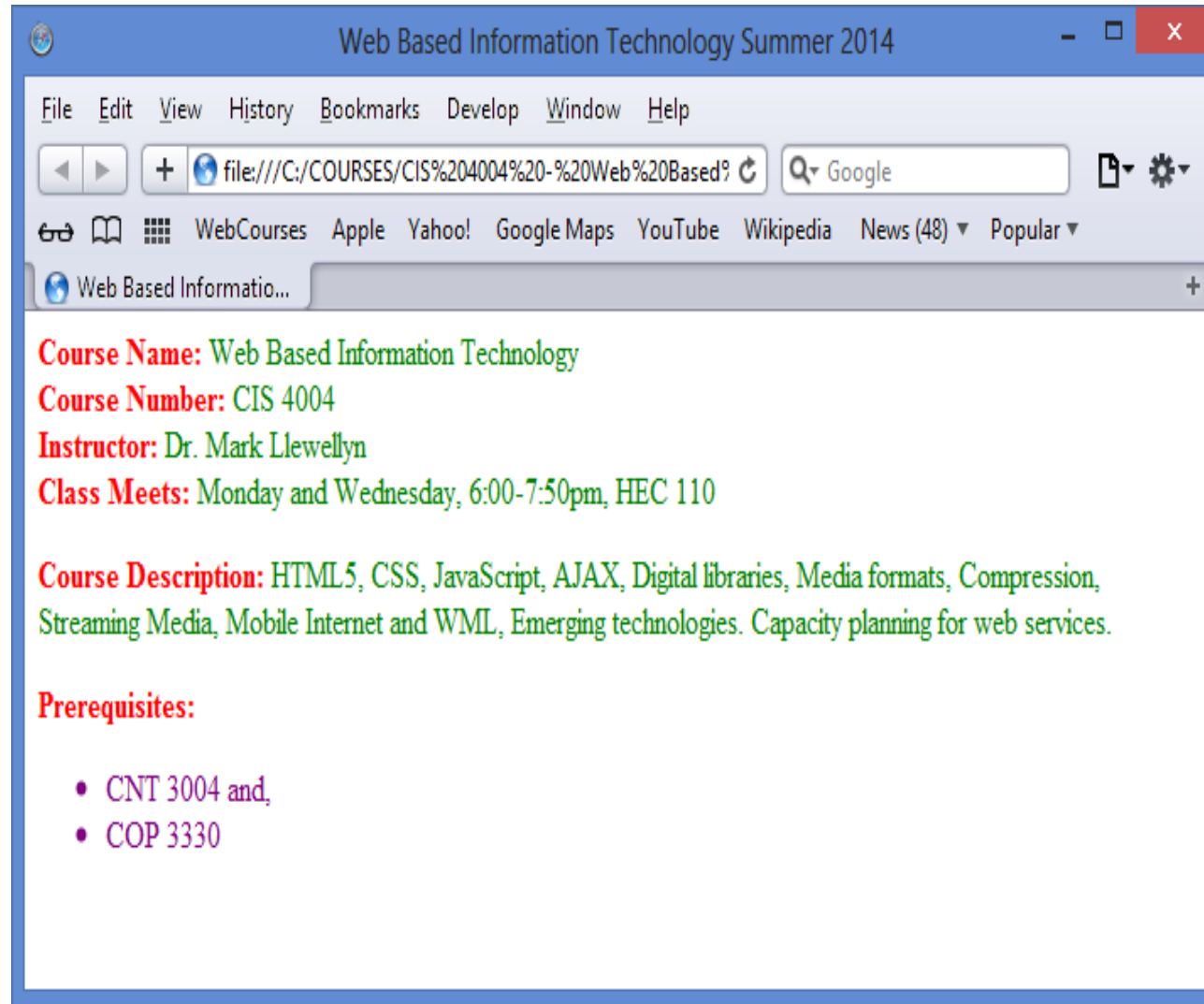


# Viewing the HTML5 With CSS

Just as before except using the style sheet “thirdCSS.css:.

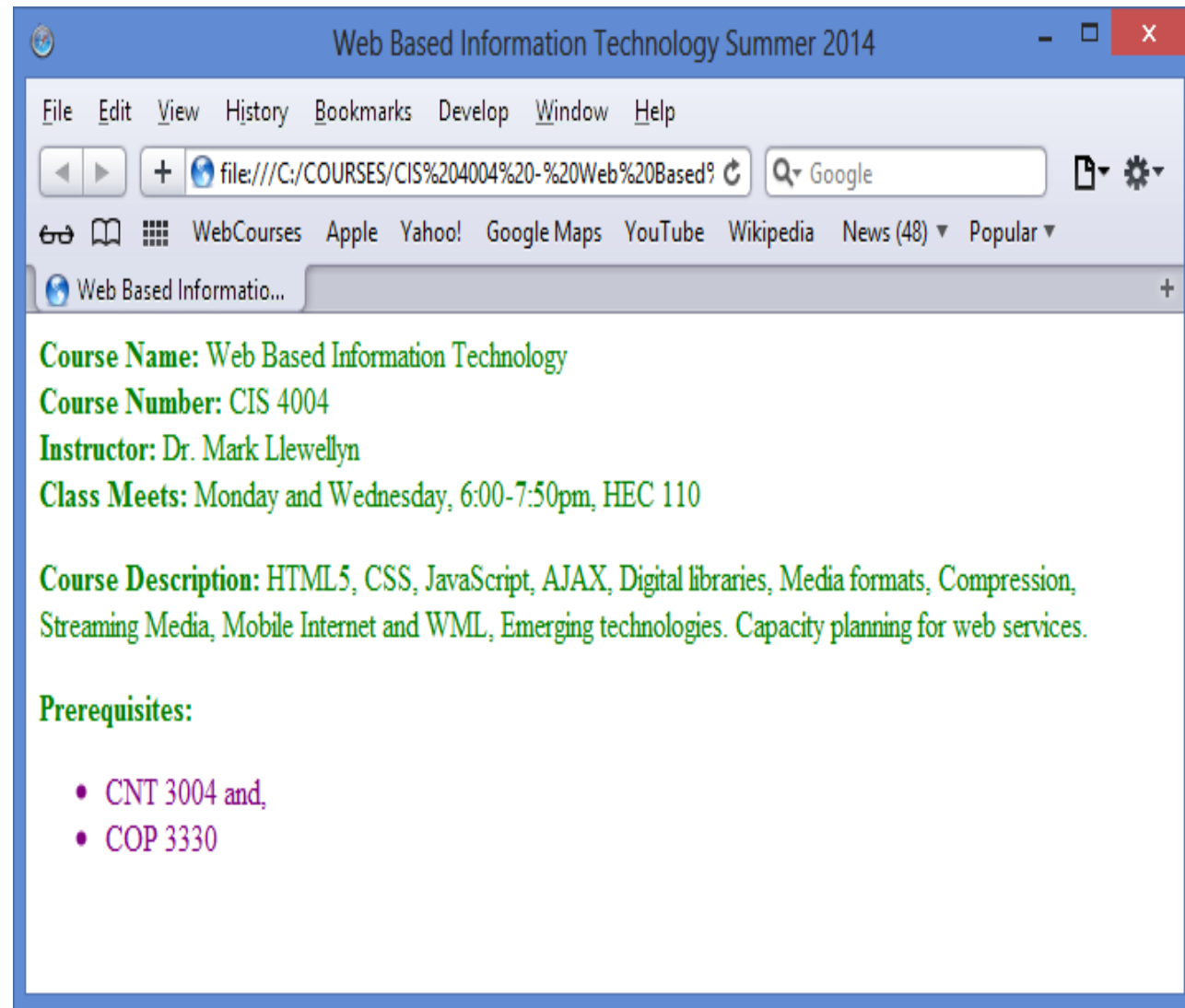
Note that all the text is green except for that in the list elements which was overridden by the `<li>` declarations and the strong elements which were overridden by the `<strong>` declarations in `thirdCSS.css`.

What would happen if you removed the color: red rule from `thirdCSS.css`? (see next page for answer.)



# Viewing the HTML5 With CSS

Since the strong element no longer styles the color to be red, it no longer overrides the universal value and thus the color of all strong elements will be the same as specified by the universal element – green! Only the list element which still styles the color to be purple will override the universal value.



# CSS – Classes and IDs

- You might be wondering after reading the first part of the CSS notes and creating your first style sheets what to do if you need to assign more than one style to the same element.
- For example, suppose that you define the following style for the `<p>` element:

```
p { color: red }
```

- If you define the `<p>` element like this, all paragraphs in your document will be formatted with red text. If you want some paragraphs to have black text, you'll need to override the global styles with an inline style. This however can become quite tedious if many style changes are needed in a single document.
- The better solution is to use a **class** or **id selector**.





# CSS – Classes and IDs

- **Class** and **id selectors** are used to define styles in your document that are independent of elements.
- Classes can be used to define styles for any number of elements and for any number of occurrences of elements in a document.
- The **id** attribute occurs only once in a document, so it should not be used to declare styles for a number of elements.
- For example, if you know that a certain element in a document will be used to uniquely identify the document, but you are not sure which element it will be in each document, you can use the **id** selector to generically reference the unique element independent of the document itself.
- The syntax for the class and id attributes are shown on the next two pages.



# Class Selector Syntax

## Style Sheet

```
.class_example {  
    color : red  
}
```

## HTML document reference

```
<p class="class_example">
```



# ID Selector Syntax

## Style Sheet

```
#id_example {  
    color : black  
}
```

## HTML document reference

```
<p id="id_example">
```



# Using class and id Selectors

- Using class and id selectors for style formatting requires certain changes to the HTML5 document because the appropriate attributes must be defined for each element to be formatted.
- Classes can also be assigned to individual elements to allow more control over formatting. This is done by placing the name of an element in front of the period in a class style declaration. For example, the following defines class formatting styles that apply only to the <p> element:

```
p          { color: black }  
p.red_text { color: red   }  
p.cyan_text { color: cyan }
```

- These declarations set font colors for the <p> element depending on which class is defined in the element. If no class attribute is specified, then the declaration for the <p> element of black is applied.



# Using `class` and `id` Selectors

- Let's look at a slightly more complex example, using our original `markup.html` file for our course description.
- We'll modify this original example using `class` and `id` attributes. We'll also define a new linked style sheet (external to the document) as well as using a global style sheet (internal to the document) and also include a few inline styles as well.
- Pages 54 & 55 show the linked style sheet we'll call `fourthCSS.css`.
- Pages 56-57 show the modified HTML document with global and inline styles.
- Finally page 58 illustrates how the document is rendered in a browser.





fourthCSS.css (first part)

```
1  /* My fourth Cascading Style Sheet */
2  /* file name: fourthCSS.css */
3
4  /* define a class called box */
5  div.box {
6      margin-top:50px;
7      background-color: yellow;
8      color: #000090;
9      border-style: double;
10     padding: 10px;
11     border-color: #000090;
12 }
13 /* define styles for <p> element */
14 p {
15     font-size:16pt;
16 }
17 /* define specific properties for the <p> element with the class name of description */
18 p.description {
19     color: #000099;
20     background-color: #cccccc;
21     font-style: italic;
22 }
```



C:\Courses\CIS 4004 - Web Based Information Technology\code\CSS\CSS - Part 2\fourthCSS.css - Notepad++

File Edit Search View Encoding Language Settings Macro Run Plugins Window ?

simpleXMLxhr.js markup with css using secondCSS.html markup using global css.html fourthCSS.css (second part)

```
23
24  /* define a unique id selector that will be applied to one element within the document */
25  #identifier {
26      color: red;
27  }
28
29  /* define a class to align text to the right */
30  .right {
31      text-align: right;
32  }
33
34  /* define universal element formatting styles */
35  * {
36      color: #333333;
37      font-family: arial;
38      font-size: 10pt;
39  }
40
```

Cascade Style Sheets File length : 848 lines : 40 Ln : 8 Col : 1 Sel : 0 | 0 Dos\Windows ANSI as UTF-8 INS



```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <title>Web Based Information Technology Summer 2014</title>
5     <link rel="stylesheet" href="fourthCSS.css" type="text/css" />
6     <style type="text/css">
7       <!--
8         h1 {
9           font-size:30pt;
10          text-align:center;
11          font-style:italic;
12        }
13        h2 {
14          font-size:18pt;
15          text-align: left;
16          font-style: italic;
17        }
18        h3 {
19          font-size:16pt;
20          text-align: left;
21          font-style: italic;
22        }
23      -->
24    </style>
25  </head>
26  <body>
27    <h1>Course Description</h1>
```

markup with fourthCSS.html  
(first part)





```

File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
markup using global css.html x markup with css using thirdCSS.html x fourthCSS.css x markup with fourthCSS.html x
20     text-align: left;
21     font-style: italic;
22     }
23 -->
24 </style>
25 </head>
26 <body>
27     <h1>Course Description</h1>
28     <div class="box">
29 <div class="right"><strong>Course Name: </strong> <span style="color:red">Web Based
30     <strong>Course Number: </strong> CIS 4004 <br />
31     <strong>Instructor: </strong> Dr. Mark Llewellyn <br />
32     <strong>Class Meets: </strong>Monday and Wednesday, 6:00-7:50pm, HEC 110<br />
33 </div>
34 <h2 id="identifier">Course Description: </h2>
35 <p class="description">HTML5, CSS, JavaScript, AJAX, Digital libraries, Media form
36     Mobile Internet and WML, Emerging technologies. Capacity planning for web servi
37 </p>
38 <h3>Prerequisites: </h3>
39 <ul>
40     <li> CNT 3004 and,</li>
41     <li> COP 3330</li>
42 </ul>
43 </body>
44 </html>
45
46

```

markup with fourthCSS.html  
(second part)



# Course Description

Course Name: **Web Based** Information Technology

Course Number: CIS 4004

Instructor: Dr. Mark Llewellyn

Class Meets: Monday and Wednesday, 6:00-7:50pm, HEC 110

## Course Description:

*HTML5, CSS, JavaScript, AJAX, Digital libraries, Media formats, Compression, Streaming Media, Mobile Internet and WML, Emerging technologies. Capacity planning for web services.*

## Prerequisites:

- CNT 3004 and,
- COP 3330



# Property Inheritance

- We looked at nesting elements in the HTML5 notes.
- Elements that are contained within other elements are said to be **children** of the outer elements, and the outer elements are referred to as **parents** of the nested elements.
- This hierarchy of elements is applied to CSS in the form of property inheritance.
- **Property inheritance** means the properties that are defined for parent elements are passed along to child elements, unless the child element overrides the property.



# Property Inheritance

- For example, if the parent of an element sets its font to be 18 points, the child elements will also have a font size of 18 points unless they declare their own rules to override the rules defined by the parent.
- Using the course description example, let's create a new style sheet called `inheritance.css` to demonstrate property inheritance for this HTML5 document.



```
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
markup using global css.html x markup with css using thirdCSS.html x fourthCSS.css x markup with fourthCSS.html x inheritance.css x
1 /* Cascading Style Sheet Showing Inheritance*/
2 /* file name: inheritance.css */
3
4 /* define the body style */
5 body {
6     color: green;
7     font-style: normal;
8     font-size:12pt;
9 }
10 /* define strong style */
11 strong {
12     color:white;
13     background-color:blue;
14 }
15 /* define paragraph style */
16 p {
17     margin-top:50px;
18     color: blue;
19     background-color:yellow;
20     font-size:20pt;
21     font-style:italic;
22     border-style: double;
23 }
24
```

inheritance.css

Cascade Style Sheets File length: 447 lines: 24 Ln: 1 Col: 1 Sel: 0 | 0 Dos\Windows ANSI as UTF-8 INS



```
C:\Courses\CIS 4004 - Web Based Information Technology\code\CSS\CSS - Part 2\markup with inheritance.html - ...
File Edit Search View Encoding Language Settings Macro Run Plugins markup with inheritance.html X
markup with css using thirdCSS.html fourthCSS.css markup with fourthCSS.html inheritance.css markup with inheritance.html X
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <title>Web Based Information Technology Summer 2014</title>
5     <meta charset="utf-8" />
6     <link rel="stylesheet" href="inheritance.css" type="text/css" />
7   </head>
8   <body>
9     <div>
10      <strong>Course Name: </strong> Web Based Information Technology<br />
11      <strong>Course Number: </strong> CIS 4004 <br />
12      <strong>Instructor: </strong> Dr. Mark Llewellyn <br />
13      <strong>Class Meets: </strong> Monday and Wednesday, 6:00-7:50pm, HEC 110 <br />
14      <p>
15        <strong>Course Description: </strong> HTML5, CSS, JavaScript, AJAX, Digital libr.
16        Mobile Internet and WML, Emerging technologies. Capacity planning for web servi.
17      </p>
18      <p>
19        <strong>Prerequisites: </strong>
20      </p>
21      <ul>
22        <li> CNT 3004 and,</li>
23        <li> COP 3330</li>
24      </ul>

```

Hyper Text Markup Language length: 920 lines: 29

Ln: 13 Col: 80 Sel: 0 | 0

UNIX

ANSI as UTF-8

INS



**Course Name:** Web Based Information Technology  
**Course Number:** CIS 4004  
**Instructor:** Dr. Mark Llewellyn  
**Class Meets:** Monday and Wednesday, 6:00-7:50pm, HEC 110

Notice that all text within the `<strong>` elements is white which overrides the settings of the `<strong>` element's parent, which is `<body>`, since the `<body>` element sets the text color to green.

Since the Course Description is contained within the `<p>` element, this overrides the style defined in the parent element `<body>`.

**Course Description:** *HTML5, CSS, JavaScript, Ajax, Digital Portals, Media formats, Compression, Streaming Media, Mobile Internet and WML, Emerging technologies. Capacity planning for web services.*

**Prerequisites:**

Here `<strong>` overrides `<p>` which overrides `<body>`.

- CNT 3004 and,
- COP 3330

The list elements are not overridden and appear in the green color as specified by the `<body>` element.

