CIS 4004: Web Based Information Technology
Spring 2013

Advanced CSS3 – Part 1

Instructor : Dr. Mark Llewellyn
markl@cs.ucf.edu
HEC 236, 407-823-2790
http://www.cs.ucf.edu/courses/cis4004/spr2013

Department of Electrical Engineering and Computer Science
University of Central Florida

Advanced CSS

- In the introductory notes on CSS, we covered basic style rules and the basic cascade in terms of linked style sheets, embedded style sheets and inline styles.
- This section of notes will cover more advanced CSS style rules and advanced selectors.
- Maybe too, you'll begin to see the differences between classes and ids in CSS.

Advanced CSS

- What happens when there is more than one style rule that applies to a given element?
- CSS uses the principle of the cascade to take into account such important characteristics as inheritance, specificity, and location in order to determine which of a group of conflicting rules should apply.
- Let's start by looking at *inheritance*. Many CSS properties not only affect the elements defined by the selector but are also inherited by the descendants of those elements.
- Look at the following markup and determine what style will be applied to the `<p>` elements inside the `<div>` element? Will they have a border around them? What color will be their text?

C:\Courses\CIS 4004 - Web-Based Info Tech\Spring 2013\code\Advanced CSS...

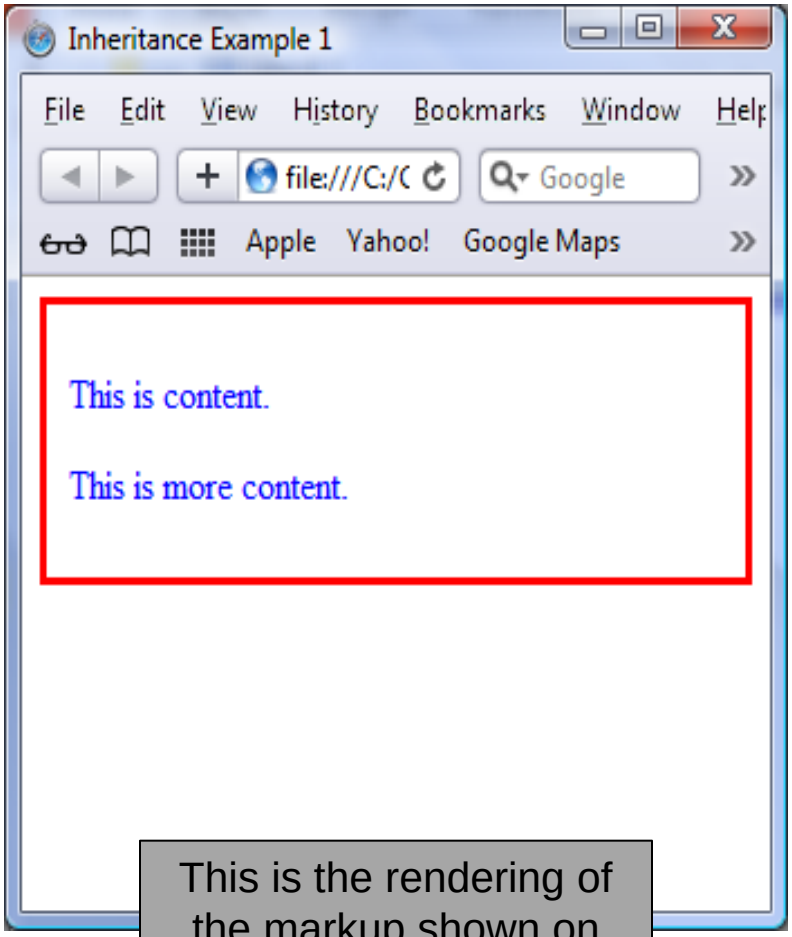
File Edit Search View Encoding Language Settings Macro Run TextFX Plugins Window ?

notes.html | bikedbscript.sql | inheritance.css | inheritance example 1.html

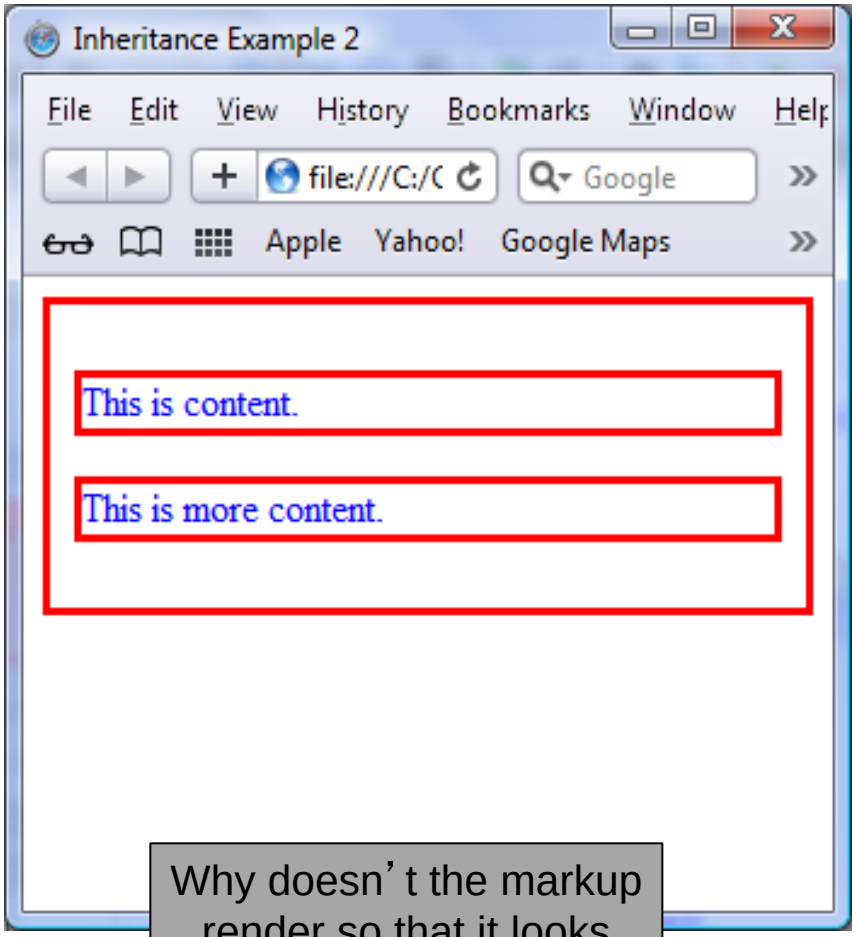
```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="utf-8" />
5      <title> Inheritance Example 1 </title>
6      <style>
7      <!--
8      div {color: blue; border: 3px red solid; padding:10px;}
9      -->
10     </style>
11 </head>
12 <body>
13     <div>
14         <p>This is content.</p>
15         <p>This is more content.</p>
16     </div>
17 </body>
18 </html>
```

leng Ln : 5 Col : 43 Sel : 0 | Dos\Windows | ANSI | INS

Advanced CSS



This is the rendering of the markup shown on the previous page.



Why doesn't the markup render so that it looks like this?

Advanced CSS

- The answer is that the color property of the `<div>` element is inherited by the `<p>` element that is nested inside the `<div>` element while the border property of the `<div>` element is not inherited by the `<p>` element.
- With most CSS properties you can use a value of `inherit` to force inheritance (which is what I did to produce the rendering on the right side of the previous page). See the markup on the next page.

C:\Courses\CIS 4004 - Web-Based Info Tech\Spring 2013\code\Advanced CSS\inher...

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

inheritance.css | inheritance example 1.html | inheritance example 2.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="utf-8" />
5      <title> Inheritance Example 2 </title>
6      <style>
7      <!--
8      div {color: blue; border: 3px red solid; padding:10px;}
9      p{ border:inherit; }
10     -->
11     </style>
12 </head>
13 <body>
14     <div>
15         <p>This is content.</p>
16         <p>This is more content.</p>
17     </div>
18 </body>
19 </html>
```

length : 33 Ln : 4 Col : 29 Sel : 0 | Dos\Windows | ANSI | INS

Advanced CSS

- While inheritance determines what happens if no style rule is applied to an element, *specificity* is the key when more than one rule is applied.
- The law of specificity states that: **the more specific the selector, the stronger the rule.**
- Thus, if one rule states that all `<h1>` elements should have blue text, but a second rule states that all `<h1>` elements with a `class` of `variant1` should be red, the second rule will override the first for all those `<h1>` elements whose `class` is `variant1`, because `h1.variant1` is a more specific selector than simply `h1`.
- The markup on the next page illustrates this concept.

C:\Courses\CIS 4004 - Web-Based Info Tech\Spring 2013\code\Advanced CSS\speci...

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

bikedbscript.sql | inheritance.css | specificity example 1.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="utf-8" />
5      <title> Specificity Example 1 </titl
6      <style>
7      <!--
8      h1 {color: blue;}
9      .variant1 {color:red;}
10     -->
11     </style>
12 </head>
13 <body>
14     <h1> This is a main header </h1>
15     <br />
16     <h1 class="variant1">This is another main header</h1>
17 </body>
18 </html>
```

length : 32 Ln : 1 Col : 1 Sel : 0 | Dos\Windows | ANSI | INS

Specificity Example 1

File Edit View History Bookmarks Window Help

file:///C:/Courses/C Google

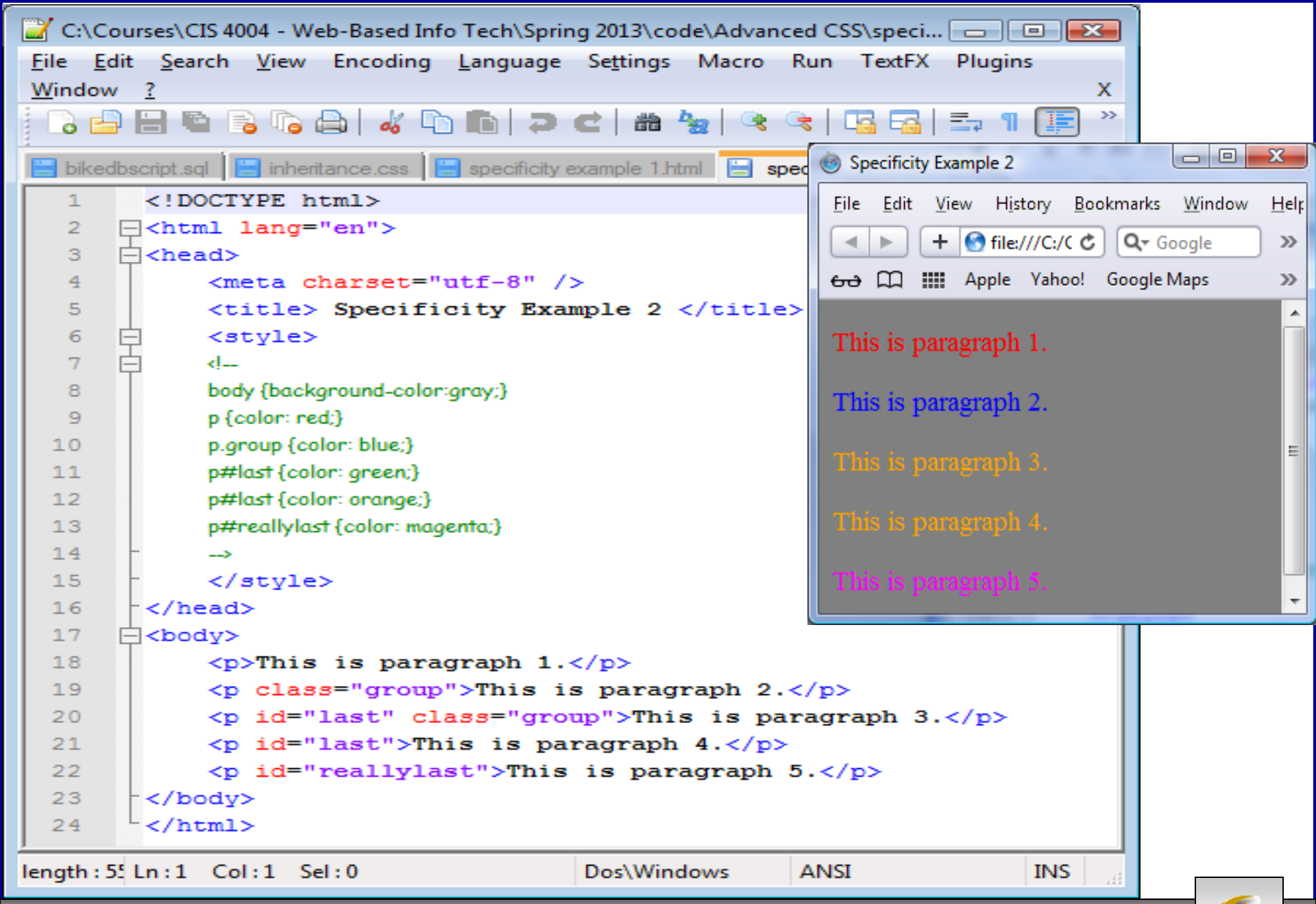
Apple Yahoo! Google Maps YouTube Wikipedia

This is a main header

This is another main header

Advanced CSS

- Note that `id` attributes are considered the most specific, since they must be unique within a document.
- The presence of a `class` attribute makes a selector more specific than a simple selector that has none.
- Further, a selector with more than one `class` is more specific than a selector with only one `class`.
- Selectors with only element names come next on the specificity scale.
- Inherited rules are considered to be the most general of all (least specific) and would be overruled by any other applicable rule.
- The markup on the next page illustrates specificity.


C:\Courses\CIS 4004 - Web-Based Info Tech\Spring 2013\code\Advanced CSS\speci...
File Edit Search View Encoding Language Settings Macro Run TextFX Plugins
Window ?
bikedbscript.sql
inheritance.css
specificity example 1.html
spec
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="utf-8" />
5 <title> Specificity Example 2 </title>
6 <style>
7 <!--
8 body {background-color:gray;}
9 p {color: red;}
10 p.group {color: blue;}
11 p#last {color: green;}
12 p#last {color: orange;}
13 p#reallylast {color: magenta;}
14 -->
15 </style>
16 </head>
17 <body>
18 This is paragraph 1.
19 This is paragraph 2.
20 This is paragraph 3.
21 This is paragraph 4.
22 This is paragraph 5.
23 </body>
24 </html>
length : 5 Ln : 1 Col : 1 Sel : 0
Dos\Windows
ANSI
INS
Specificity Example 2
File Edit View History Bookmarks Window Help
file:///C:/C
Google
Apple Yahoo! Google Maps
This is paragraph 1.
This is paragraph 2.
This is paragraph 3.
This is paragraph 4.
This is paragraph 5.

Advanced CSS

- Sometimes specificity is not enough to determine which style rule will apply when conflicting rules are present. In these cases, the location of the rule resolves the conflict: Rules that appear later have more weight.

- This means that inline styles (not recommended except in rare cases) are considered to appear after (and thus have more weight than) equally specific rules applied either in an embedded or external style sheet.

You can override the entire cascade system by declaring that a particular rule be more important than the others by adding `!important` at the end of the rule. This is also not a recommended practice except in very rare cases. The example on the next page illustrates this, but do not get in the habit of doing something like this. It is considered very bad practice because it makes the declaration too strong and you CSS will get bogged down with longer rules if you need to override it.

C:\Courses\CIS 4004 - Web-Based Info Tech\Spring 2013\code\Advanced CSS\speci...

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

specificity example 1.html | specificity example 2.html | specific

```
1   <!DOCTYPE html>
2   <html lang="en">
3   <head>
4       <meta charset="utf-8" />
5       <title> Specificity Example 3 </t
6       <style>
7       <!--
8           body {background-color:gray;}
9           p {color: red;} !important
10          p.group {color: blue;}
11          p#last {color:green;} !important
12          p#last {color:orange;}
13          p#reallylast {color:magenta;}
14      -->
15      </style>
16  </head>
17  <body>
18      <p>This is paragraph 1.</p>
19      <p class="group">This is paragraph 2.</p>
20      <p id="last" class="group">This is paragraph 3.</p>
21      <p id="last">This is paragraph 4.</p>
22      <p id="reallylast">This is paragraph 5.</p>
23  </body>
24  </html>
```

length : 57 Ln : 1 Col : 1 Sel : 0 | Dos\Windows | ANSI | INS

Specificity Example 3

File Edit View History Bookmarks Window Help

file:///C:/C | Google

Apple Yahoo! Google Maps

This is paragraph 1.

This is paragraph 2.

This is paragraph 3.

This is paragraph 4.

This is paragraph 5.

Advanced CSS

- The relationship between an embedded style sheet and any linked external style sheets depends on their relative positions. So too does the relationship between multiple linked external style sheets.
- If the `<link>` element comes later in the `<head>` element than the `<style>` element, then it will override the rules in the `<style>` element which preceded it in the markup.
- If the `<link>` element comes earlier in the markup than the `<style>` element, then the rules in the `<style>` element will override the rules in the `<link>` element.
- The examples on the following few pages should illustrate this concept.

Linked stylesheet before embedded stylesheet

C:\Courses\CIS 4004 - Web-Based Info Tech\Spring 2013\code\Advanced CSS\spe...

inheritance.css | specificity example 4.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="utf-8" />
5      <title> Specificity Example 4 </title>
6      <link rel="stylesheet" href="stylesheet1.css" />
7      <style>
8      <!--
9          body {background-color:gray;}
10         p {color: magenta;}
11         p.group {color: yellow;}
12     -->
13     </style>
14 </head>
15 <body>
16     <p>This is paragraph 1.</p>
17     <p class="group">This is paragraph 2.</p>
18 </body>
19 </html>
```

length : 38 Ln : 1 Col : 1 Sel : 0 | Dos\Windows | ANSI | INS

Link element first

C:\Courses\CIS 4004 - Web-Base...

File Edit Search View Encoding Language Settings Window

External stylesheet

inheritance.css | specificity example 4.html

```
1  p {color: blue; }
2
3  p.group {color: red; }
```

Ln | Dos\Windows | ANSI | INS

Specificity Example 4

File Edit View History Bookmarks Window Help

file:///C:/C Google

Apple Yahoo! Google Maps

This is paragraph 1.

This is paragraph 2.

Linked stylesheet after embedded stylesheet

C:\Courses\CIS 4004 - Web-Based Info Tech\Spring 2013\code\Advanc

stylesheet1.css | specificity example 5.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="utf-8" />
5      <title> Specificity Example 5 </title
6      <style>
7      <!--
8          body {background-color:gray;}
9          p {color: magenta;}
10         p.group {color: yellow;}
11     -->
12     </style>
13     <link rel="stylesheet" href="stylesheet1.css
14 </head>
15 <body>
16     <p>This is paragraph 1.</p>
17     <p class="group">This is paragraph 2.</p>
18 </body>
19 </html>
```

lengtl Ln:1 Col:1 Sel:0 | Dos\Windows | ANSI | INS

Link element second

C:\Courses\CIS 4004 - Web-Based Info Tech\Fa...

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

External stylesheet

specificity example 2.html | spe | Cut | example 4.html | sty

```
1  p {color: blue; }
2
3  p.group {color: red; }
```

Ln:1 Col:1 S Dos\Windows ANSI INS

Specificity Example 5

File Edit View History Bookmarks Window Help

file:///C:/C Google

Apple Yahoo! Google Maps

This is paragraph 1.

This is paragraph 2.

Two linked stylesheets

C:\Courses\CIS 4004 - Web-Based Info Tech\Spring 2013\code\Advanced CSS\...

stylesheet1.css | specificity example 5.html | specificity example 6.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="utf-8" />
5      <title> Specificity Example 6 </title>
6      <link rel="stylesheet" href="stylesheet1.css" />
7      <link rel="stylesheet" href="stylesheet2.css" />
8  </head>
9  <body>
10     <p>This is paragraph 1.</p>
11     <p class="group">This is paragraph 2.</p>
12 </body>
13 </html>
```

Ln:1 Col:1 Sel:0 | Dos\Windows | ANSI

External stylesheet1

C:\Courses\CIS 4004 - Web-Based Inf...

specificity example 2.html | specificity example 6.htn

```
1  p {color: blue; }
2
3  p.group {color: red; }
```

Ln:1 | Dos\Windows | ANSI | INS

External stylesheet2

stylesheet1.css | stylesheet2.css

```
1  p {color: navy; }
2
3  p.group {color: lime; }
```

Ln:1 | Dos\Windows | ANSI | INS

Specificity Example 6

This is paragraph 1.

This is paragraph 2.

Two linked stylesheets – ordered reversed

C:\Courses\CIS 4004 - Web-Based Info Tech\Spring 2013\code\Advanced CSS\...

specificity example 5.html | specificity example 6.html | specificity example 7.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="utf-8" />
5      <title> Specificity Example 7 </title>
6      <link rel="stylesheet" href="stylesheet2.css" />
7      <link rel="stylesheet" href="stylesheet1.css" />
8  </head>
9  <body>
10     <p>This is paragraph 1.</p>
11     <p class="group">This is paragraph 2.</p>
12 </body>
13 </html>
```

lengtl Ln:1 Col:1 Sel:0 | Dos\Windows | ANSI

External stylesheet1

C:\Courses\CIS 4004 - Web-Based Inf...

specificity example 2.html | specificity example 6.htn

```
1  p {color: blue; }
2
3  p.group {color: red; }
```

Ln:1 | Dos\Windows | ANSI | INS

External stylesheet2

stylesheet1.css | stylesheet2.css

```
1  p {color: navy; }
2
3  p.group {color: lime; }
```

Specificity Example 7

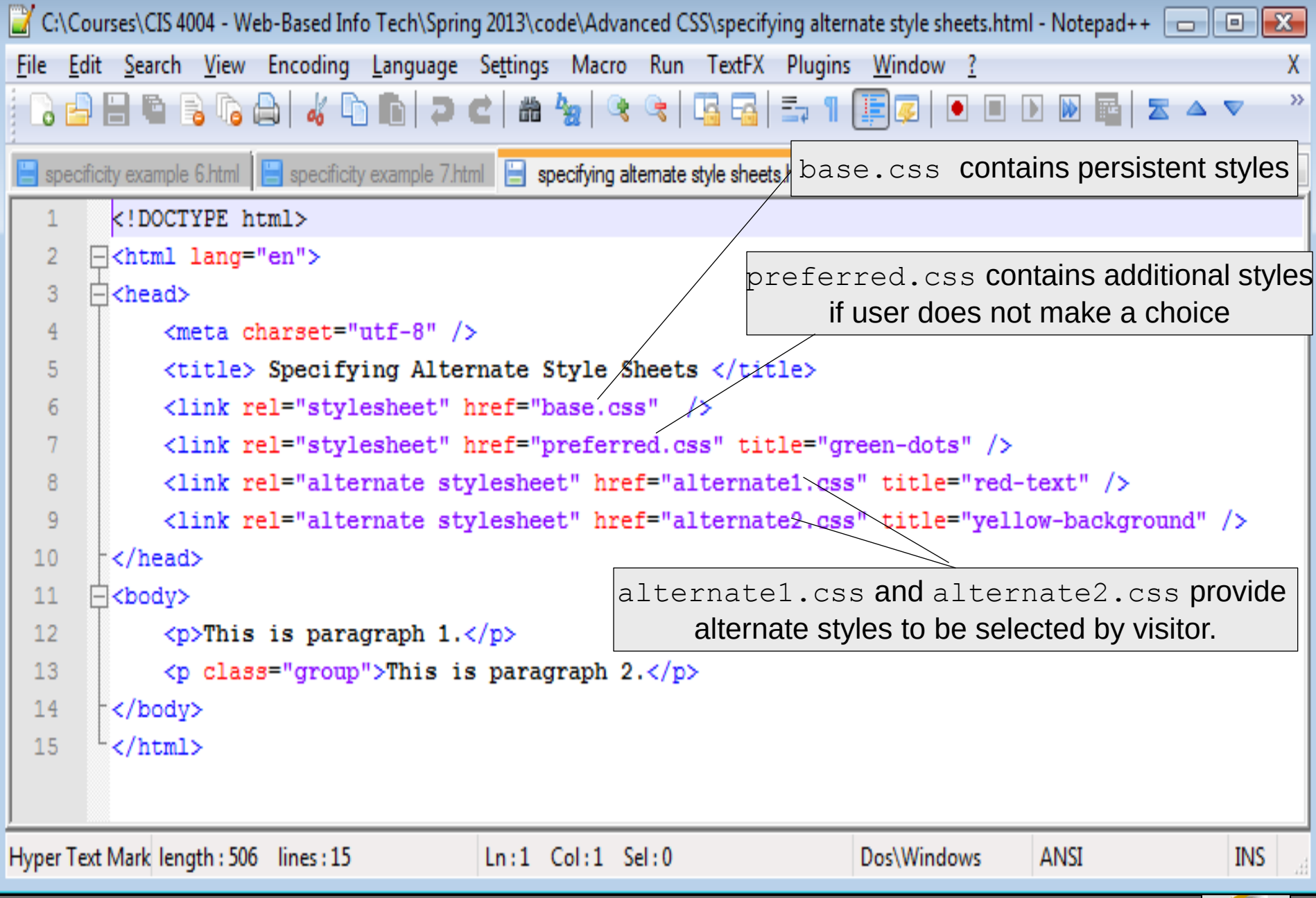
file:///C:/C

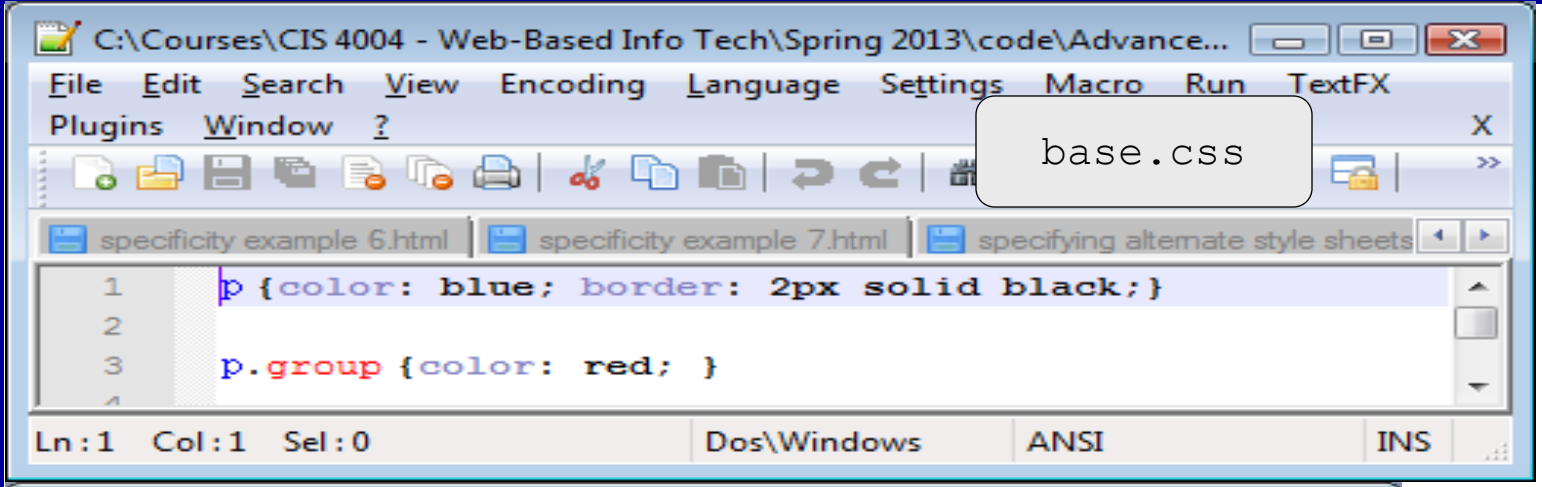
This is paragraph 1.

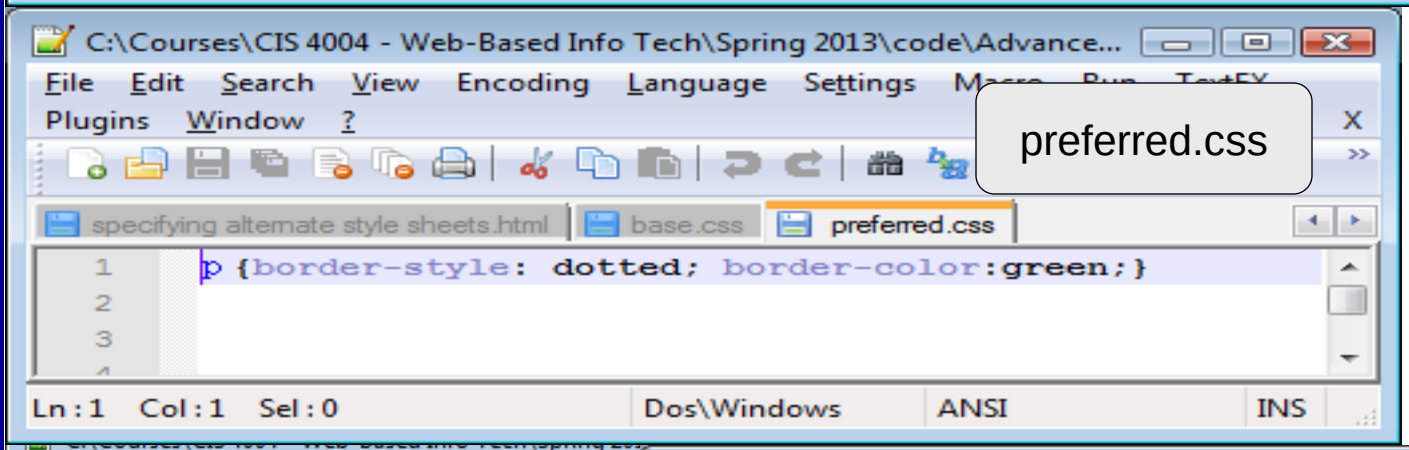
This is paragraph 2.

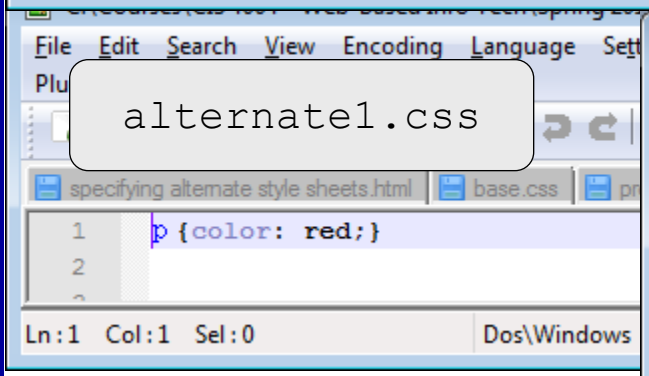
Specifying Alternate Style Sheets

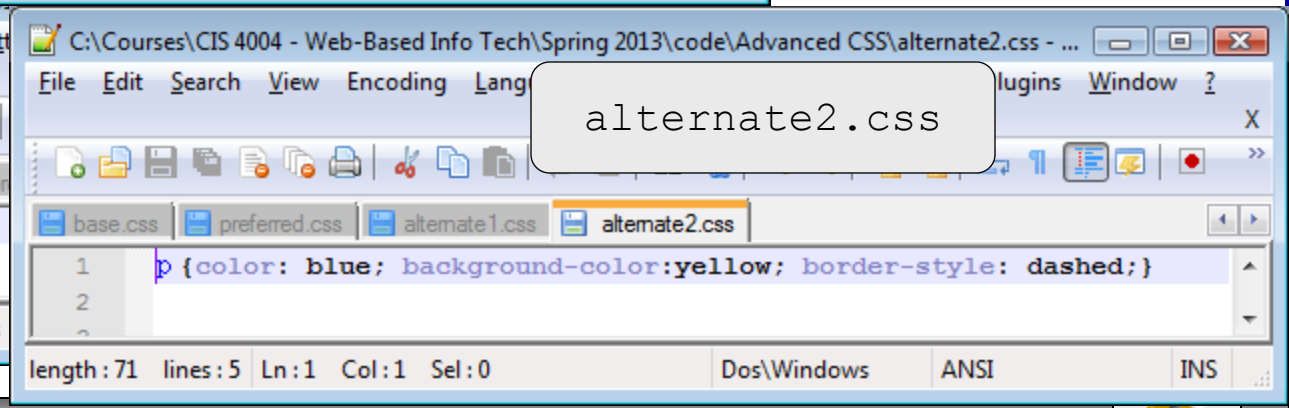
- You can link to more than one style sheet and let visitors choose the styles they like best.
- The HTML5 specifications allow for a base set of persistent styles that are applied regardless of the visitor's preference, a default or preferred set of additional styles that are applied if the visitor makes no choice, and one or more alternate style sheets that the visitor can choose, at which point the preferred set (though not the persistent set) is deactivated and ignored.
- Alternate style sheets allow you to provide different themes for your website.
- The example on the following few pages illustrates using alternate style sheets.

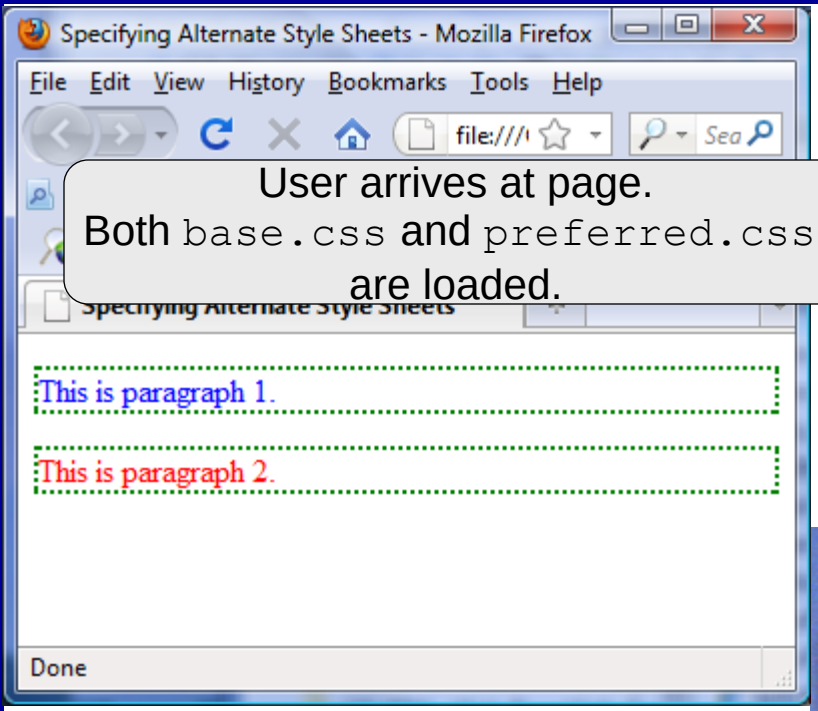

C:\Courses\CIS 4004 - Web-Based Info Tech\Spring 2013\code\Advanced CSS\specifying alternate style sheets.html - Notepad++
File Edit Search View Encoding Language Settings Macro Run TextFX Plugins Window ?
specificity example 6.html
specificity example 7.html
specifying alternate style sheets.h
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="utf-8" />
5 <title> Specifying Alternate Style Sheets </title>
6 <link rel="stylesheet" href="base.css" />
7 <link rel="stylesheet" href="preferred.css" title="green-dots" />
8 <link rel="alternate stylesheet" href="alternate1.css" title="red-text" />
9 <link rel="alternate stylesheet" href="alternate2.css" title="yellow-background" />
10 </head>
11 <body>
12 This is paragraph 1.
13 This is paragraph 2.
14 </body>
15 </html>
base.css contains persistent styles
preferred.css contains additional styles if user does not make a choice
alternate1.css and alternate2.css provide alternate styles to be selected by visitor.
Hyper Text Mark length : 506 lines : 15
Ln : 1 Col : 1 Sel : 0
Dos\Windows
ANSI
INS


C:\Courses\CIS 4004 - Web-Based Info Tech\Spring 2013\code\Advance...
File Edit Search View Encoding Language Settings Macro Run TextFX
Plugins Window ?
base.css
specificity example 6.html
specificity example 7.html
specifying alternate style sheets
1 p {color: blue; border: 2px solid black;}
2
3 p.group {color: red; }
Ln:1 Col:1 Sel:0
Dos\Windows
ANSI
INS


C:\Courses\CIS 4004 - Web-Based Info Tech\Spring 2013\code\Advance...
File Edit Search View Encoding Language Settings
Plugins Window ?
preferred.css
specifying alternate style sheets.html
base.css
preferred.css
1 p {border-style: dotted; border-color:green;}
2
3
Ln:1 Col:1 Sel:0
Dos\Windows
ANSI
INS

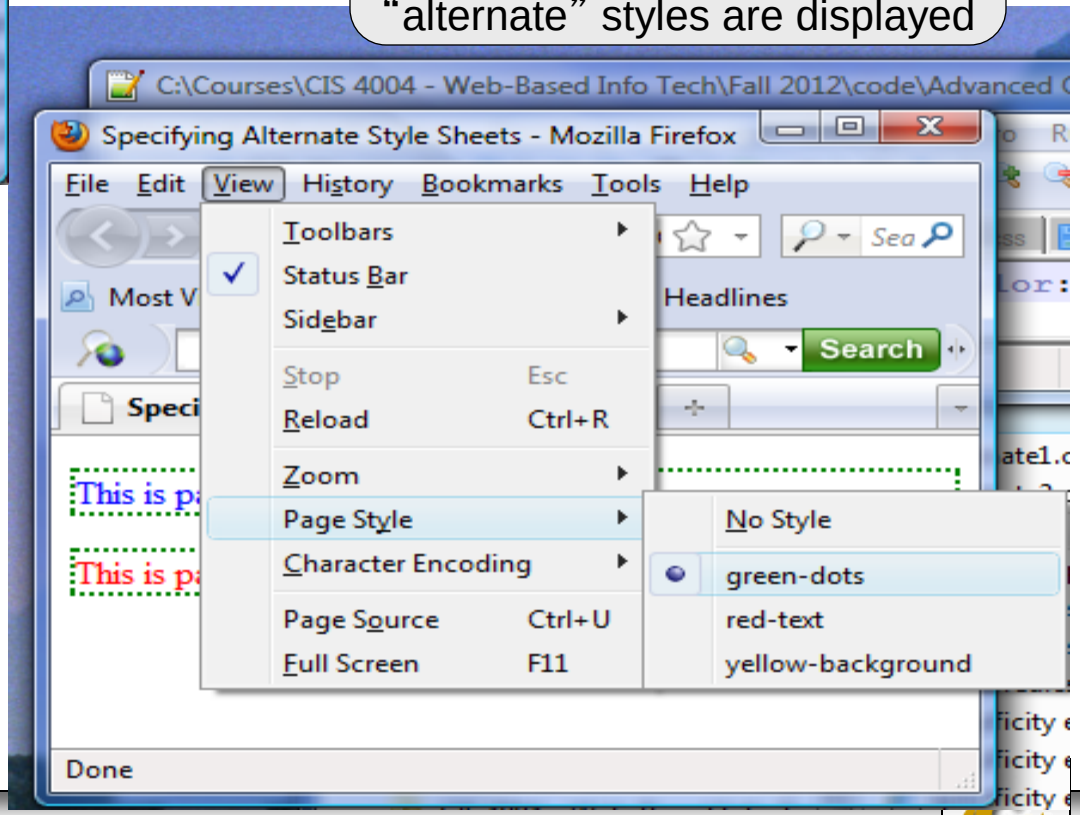

File Edit Search View Encoding Language
alternate1.css
specifying alternate style sheets.html
base.css
1 p {color: red;}
2
Ln:1 Col:1 Sel:0
Dos\Windows

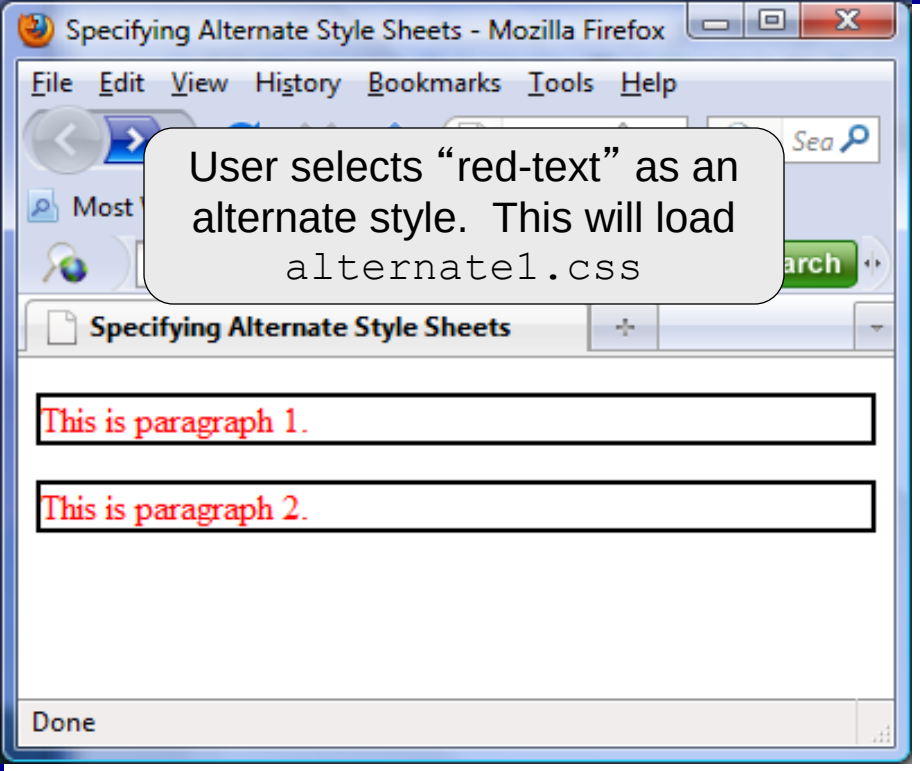

C:\Courses\CIS 4004 - Web-Based Info Tech\Spring 2013\code\Advanced CSS\alternate2.css - ...
File Edit Search View Encoding
alternate2.css
base.css
preferred.css
alternate1.css
alternate2.css
1 p {color: blue; background-color:yellow; border-style: dashed;}
2
length:71 lines:5 Ln:1 Col:1 Sel:0
Dos\Windows
ANSI
INS

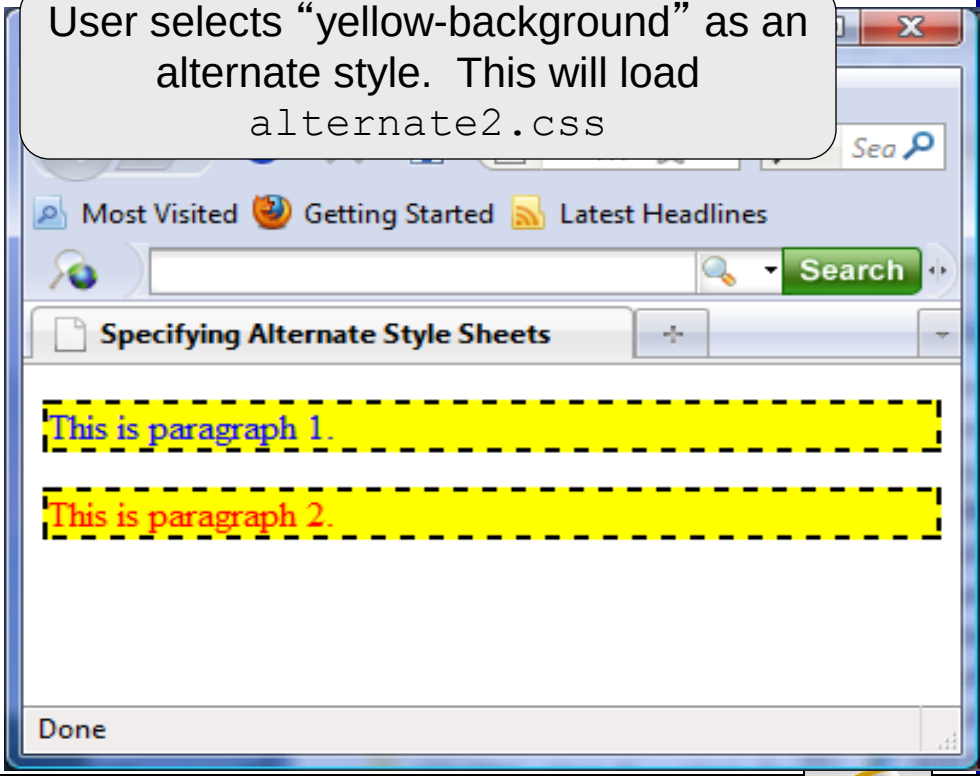


User arrives at page.
Both `base.css` and `preferred.css` are loaded.

User selects “View”
User selects “Page Style”
The “preferred” and various “alternate” styles are displayed




Specifying Alternate Style Sheets - Mozilla Firefox
File Edit View History Bookmarks Tools Help
User selects "red-text" as an alternate style. This will load alternate1.css
Specifying Alternate Style Sheets
This is paragraph 1.
This is paragraph 2.
Done


User selects "yellow-background" as an alternate style. This will load alternate2.css
Most Visited
Getting Started
Latest Headlines
Search
Specifying Alternate Style Sheets
This is paragraph 1.
This is paragraph 2.
Done

Specifying Alternate Style Sheets

- You do not need to specify a preferred style sheet in order to provide alternative style sheets.
- Note that the style sheet which is to be designated as the preferred style sheet must have: `rel = "stylesheet"` and `title="label"` where "`label`" identifies the preferred style sheet.
- Note that the style sheet(s) which is(are) offered as an alternate choice must use `rel="alternate stylesheet"` and `title="label"` where "`label`" identifies the alternate style sheet(s).
- Firefox and Opera offer the easiest way to switch from one style sheet to another. However, there are JavaScript solutions for other browsers. Do a Google search for "style sheet switcher" to find code you can use.

Advanced Selector Construction

- You've already seen many different examples in the markup of some of the basic types of CSS selectors.
- We've been using the most common forms of selectors, which are named based as well as those that use classes and ids. Now we'll explore some more advance selector constructions.
- A selector can define up to five different criteria for choosing the elements that should be formated:
  - The type or name of the element.
  - The context in which the element is found.
  - The class or id of an element.
  - The pseudo-class of an element or a pseudo-element.
  - Whether or not an element has certain attributes and values.

Advanced Selector Construction

Name/type Based Selector

Name of desired element

```
h1 { color: red; }
```

The simplest type of selector is simply the name of the type of element that should be formatted – in this case the <h1> element.

Context Based Selector

Context

Name of desired element

```
h1 em { color: red; }
```

The context selector will apply formatting to the specified named element only when it is found in the specified context. In this case the elements that appear inside <h1> elements will be formatted. Any element found elsewhere will not be formatted.

Advanced Selector Construction

Class Based Selector

Class Name

```
.level1 { color: red; }
```

The `class` selector chooses all elements that belong the class. In this case a class named "`level1`" is defined. In the markup any element using this class will have red text, for example, `<p class="level1"> . . . </p>`. A `class` selector can appear any number of times in a page of markup.

ID Based Selector

ID Name

```
#level1 { color: red; }
```

The `id` selector chooses the one element with the specified id. In this case an `id` named "`level1`" is defined. In the markup there can be at most one element using this `id` and it will have red text, for example, `<p id="level1"> . . . </p>`. An `id` selector can appear only once in each page of markup.

Advanced Selector Construction

More Specific Class Based Selector

Class Name

```
em.level1 { color: red; }
```

You can be more specific by prefixing a `class` selector with an element name to target. In this case the selector chooses only the `<em>` elements with the "`level1`" class rather than every element with the "`level1`" class.

More Specific ID Based Selector

ID Name

```
em#level1 { color: red; }
```

You can be more specific by prefixing an `id` selector with an element name to target. In this case the selector chooses only the one `<em>` element with the "`level1`" `id`.

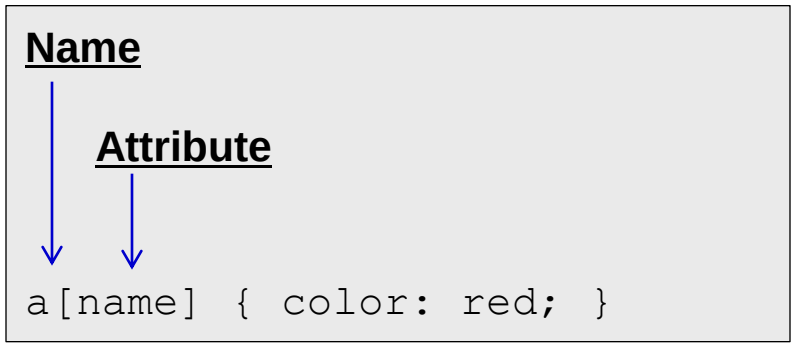
WARNING: In general, do not use this approach unless you have to. The less specific `class` and `id` selectors on the previous page are the preferred methods for using classes and ids.

Advanced Selector Construction

Pseudo-Class Based Selector

The pseudo-class based selector chooses elements that belong to the pseudo-class. In this case, the <a> elements that have not yet been visited.
More on pseudo-classes later.

Attribute/Value Based Selector



The attribute/value selector allows you to specify in square brackets additional information about the desired elements attributes, values, or both.
More later on the specific cases that can appear in this type of selector.

Advanced Selector Construction

- Selectors can include any combination of the various types of selectors shown on the previous four pages in order to pinpoint the desired elements for your styles.
- Mostly, you will use one or two at a time.
- You can apply the same declarations to several selectors at once if you need to apply the same style rules to different groups of elements. This is done by separating the various selectors with commas as shown below:

Group selector

```
h1,
p,
ul   { color: red; }
```

Separating selectors with commas allows you to apply the same styles to many different elements, classes, ids, etc. Convention is to separate each member of the group on a separate line in your style sheet.

An Aside on Naming Classes and Ids

- As a general rule, you should not name classes nor ids using a name that describes how something looks. For example, don't define a class such as: `.greentext {color: green;}`.
- Rather you should name the `class` (or `id`) using a name that conveys some meaning about the content to which it is being applied. This is especially true when you consider that you might change the style later.
- Remember that classes and ids add semantic value to your HTML5 markup just as the elements do, so be sure to use semantically meaningful `class` and `id` names as well.

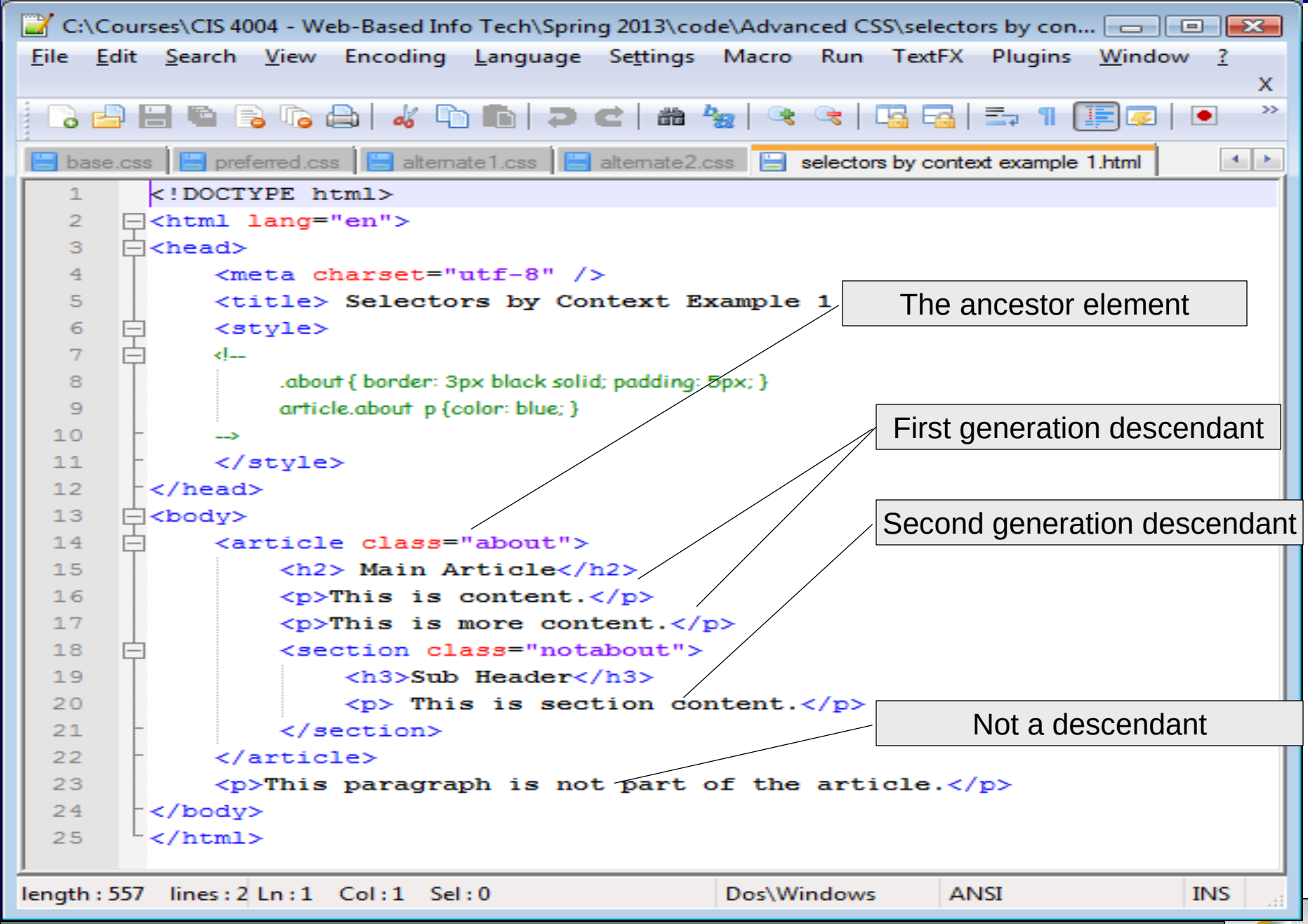
An Aside on Classes Vs. Ids

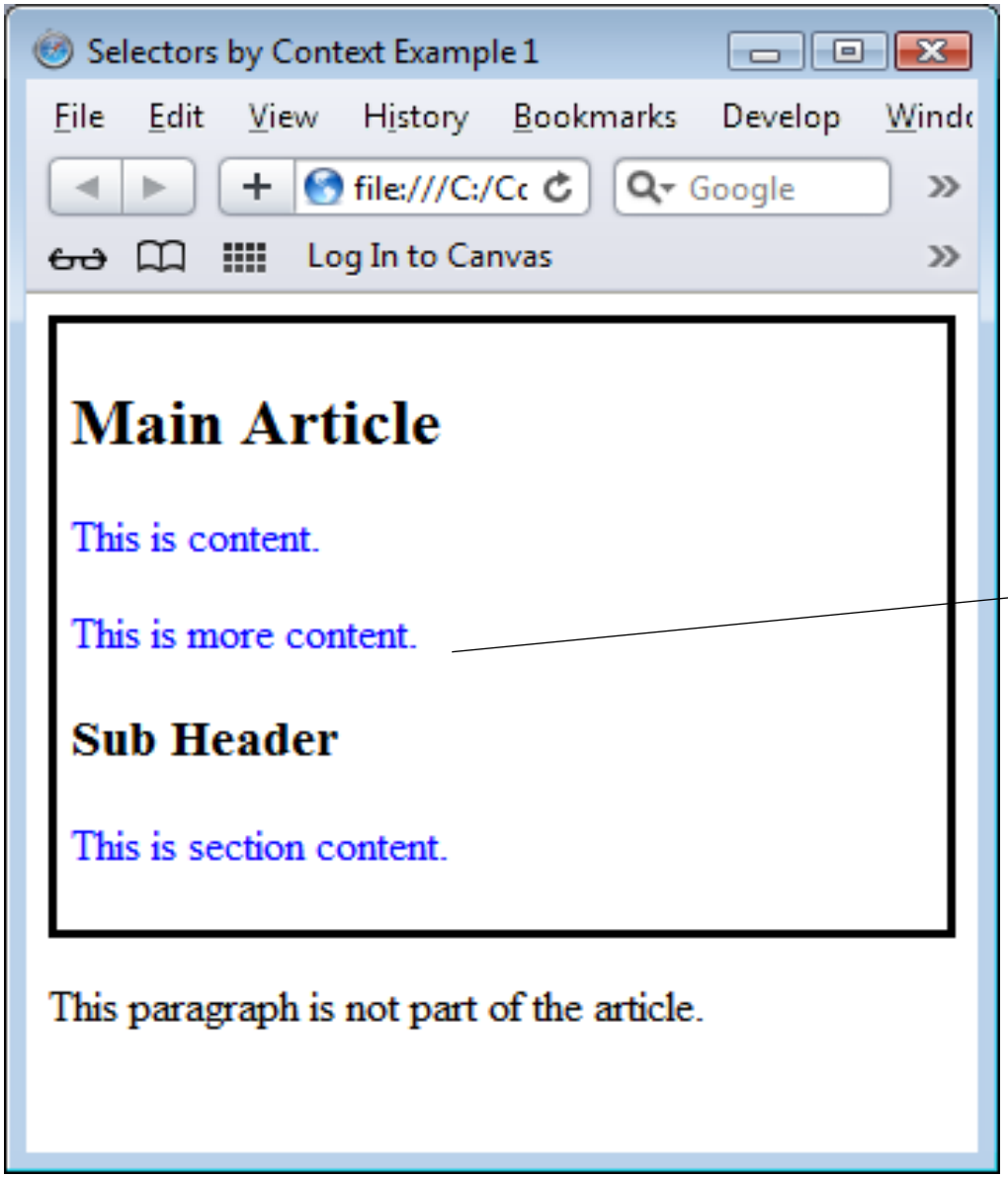
- When deciding between `class` selectors and `id` selectors, I suggest using classes whenever possible, in large part because you can reuse them. Some web designers advocate not using ids at all, an argument I understand, though ultimately the choice is yours. It's a subject that has been strongly debated on various web development forums. In any case, here are two of the issues that id selectors introduce:
  - Their associated styles can't be reused on other elements, since an `id` can appear on only one element per page. This can lead to repeating styles on other elements, rather than sharing them as a `class`.
  - They are far more specific than `class` selectors. This means that if you ever need to override styling that was defined with an `id` selector, you'll need to write a CSS rule that's even more specific. A few of these might not be too hard to manage, but if you're working on a site of decent size, your CSS can get longer and more complicated than necessary.

Selecting Elements By Context

- In CSS, you can pinpoint elements depending on their ancestors, their parent, or their siblings.
- An ancestor is any element that contains the desired element (the descendant), regardless of the number of generations that separate them.
- To make use of selecting elements by context, it is imperative that you properly indent your markup or you will be hopelessly lost when it comes to applying the correct style.

NOTE: A selector based on an element's ancestor has traditionally been known (in CSS1 and CSS2) as a `descendant selector`, but CSS3 has renamed it a `descendant combinator`. Many people (including me) still refer to it as a selector.

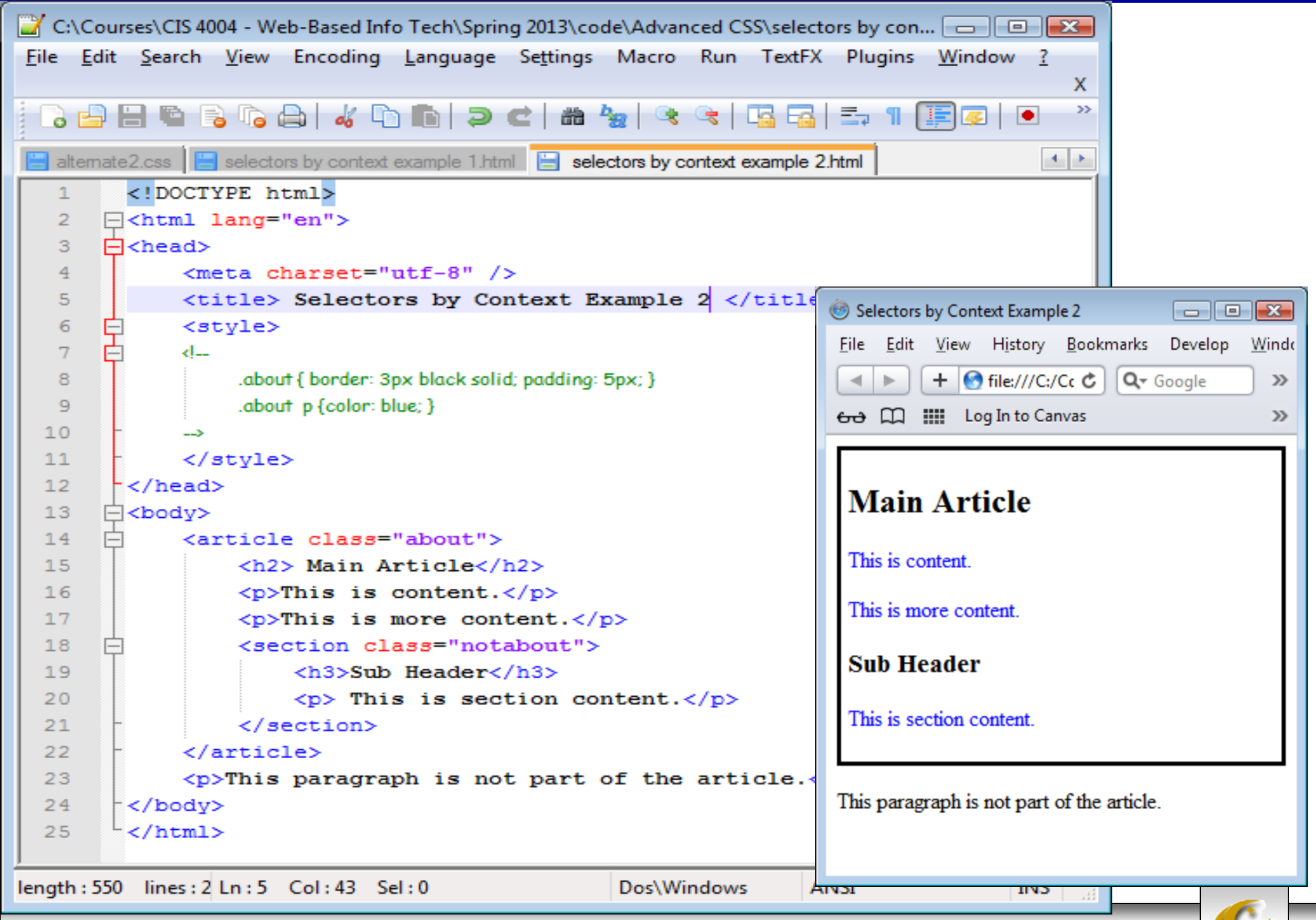

C:\Courses\CIS 4004 - Web-Based Info Tech\Spring 2013\code\Advanced CSS\selectors by con...
File Edit Search View Encoding Language Settings Macro Run TextFX Plugins Window ?
base.css
preferred.css
alternate1.css
alternate2.css
selectors by context example 1.html
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8" />
<title> Selectors by Context Example 1</title>
<style>
<!--
.about { border: 3px black solid; padding: 5px; }
article.about p {color: blue; }
-->
</style>
</head>
<body>
<article class="about">
<h2> Main Article</h2>
This is content.
This is more content.
<section class="notabout">
<h3>Sub Header</h3>
 This is section content.
</section>
</article>
This paragraph is not part of the article.
</body>
</html>
The ancestor element
First generation descendant
Second generation descendant
Not a descendant
length : 557 lines : 2 Ln : 1 Col : 1 Sel : 0
Dos\Windows
ANSI
INS



The `<article>` element is a member of the class "`about`" and is styled with only padding in that element. The contextual selector finds 3 separate paragraphs as descendants inside the `<article>` element and thus they are all styled with blue text. Note that the last paragraph in the markup is not a descendant of the `<article>` element and is thus not styled by the contextual selector.

Selecting Elements By Context

- There is often more than one way to craft your selectors to get the desired effect.
- It often comes down to how specific you need to be.
- The previous example, uses a somewhat verbose and more specific form for contextual selection than is actually required in this case.
- The following example combines a `class` selector with a descendant selector (you could combine with an `id` selector as well) to achieve exactly the same effect. Notice that in this second version the first selector (the `article p { }`) is less specific than both the one that follows it (the `.about p { }`) and the one from the first version.

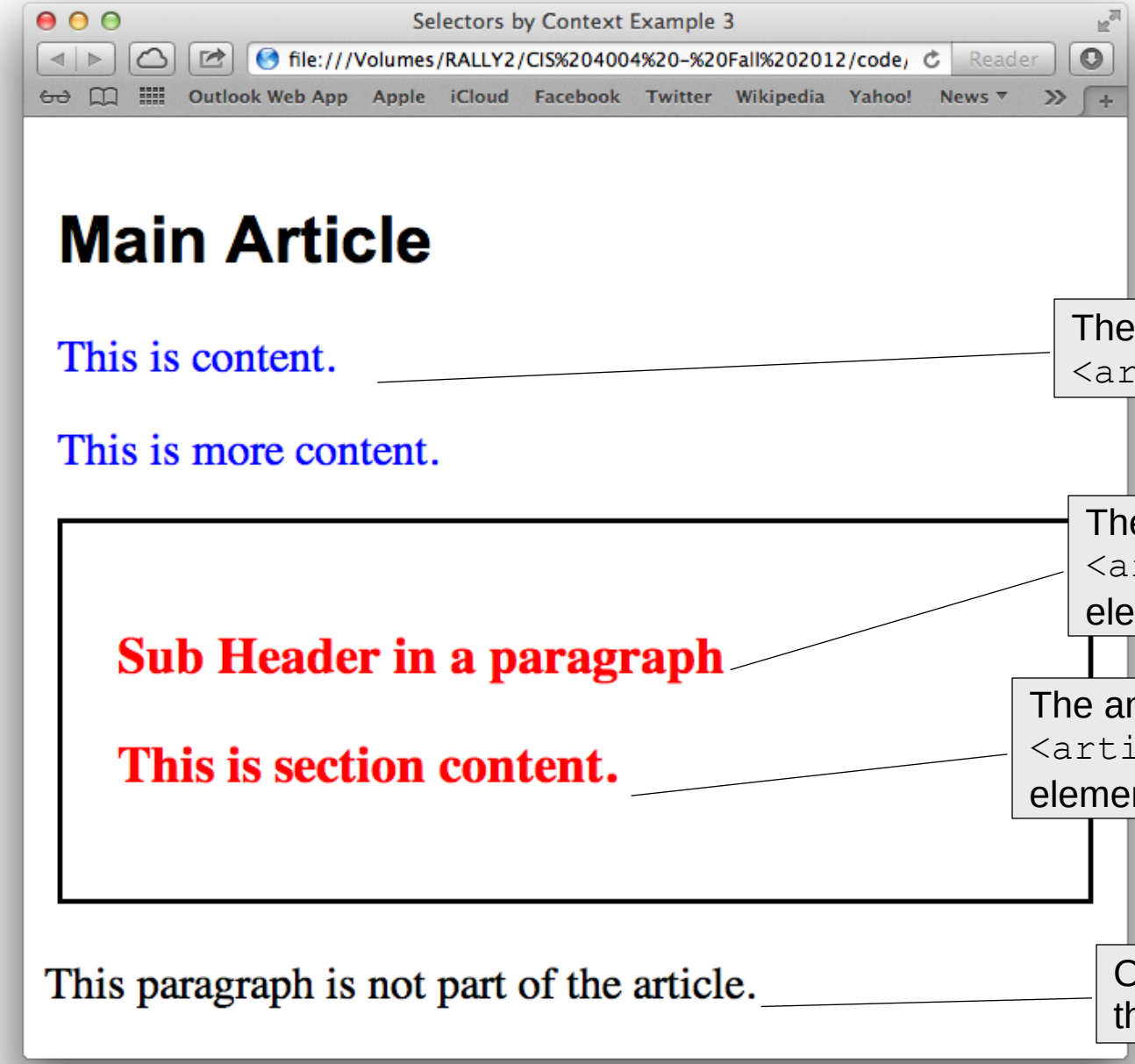

C:\Courses\CIS 4004 - Web-Based Info Tech\Spring 2013\code\Advanced CSS\selectors by con...
File Edit Search View Encoding Language Settings Macro Run TextFX Plugins Window ?
alternate2.css
selectors by context example 1.html
selectors by context example 2.html
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8" />
<title> Selectors by Context Example 2 </title
<style>
<!--
.about { border: 3px black solid; padding: 5px; }
.about p {color: blue; }
-->
</style>
</head>
<body>
<article class="about">
<h2> Main Article</h2>
This is content.
This is more content.
<section class="notabout">
<h3>Sub Header</h3>
 This is section content.
</section>
</article>
This paragraph is not part of the article.
</body>
</html>
length : 550 lines : 2 Ln : 5 Col : 43 Sel : 0
Dos\Windows
Selectors by Context Example 2
File Edit View History Bookmarks Develop
file:///C:/C
Google
Log In to Canvas
Main Article
This is content.
This is more content.
Sub Header
This is section content.
This paragraph is not part of the article.

Selecting Elements By Context

- There is no limit to how deep you can extend the descendant combinator. From a practical point of view more than 3 levels is rarely seen.
- The example on the next page illustrates more depth.
- Be sure you understand why both the `<h3>` element and the `<p>` element inside the `<section>` element have red text.

selectors by context example 3.html

```
<!DOCTYPE html>
<html lang="en">
<head>
        <meta charset="utf-8" />
        <title> Selectors by Context Example 3 </title>
        <style>
        <!--
                article { border: none; padding: 5px; }
                section { border: 2px solid black; padding:20px; }
                h2 { font-family: arial; }
                article section h3 p { color: red; }
                article p {color: blue; }
        -->
        </style>
</head>
<body>
        <article class="about">
                <h2> Main Article</h2>
                <p>This is content.</p>
                <p>This is more content.</p>
                <section class="notabout">
                        <h3><p>Sub Header in a paragraph</p</h3>
                        <p> This is section content.</p>
                </section>
        </article>
        <p>This paragraph is not part of the article.</p>
</body>
</html>
```

The ancestor element is the `<article>` element

The ancestor element is the `<article><section><h3>` elements

The ancestor element is the `<article><section>` elements

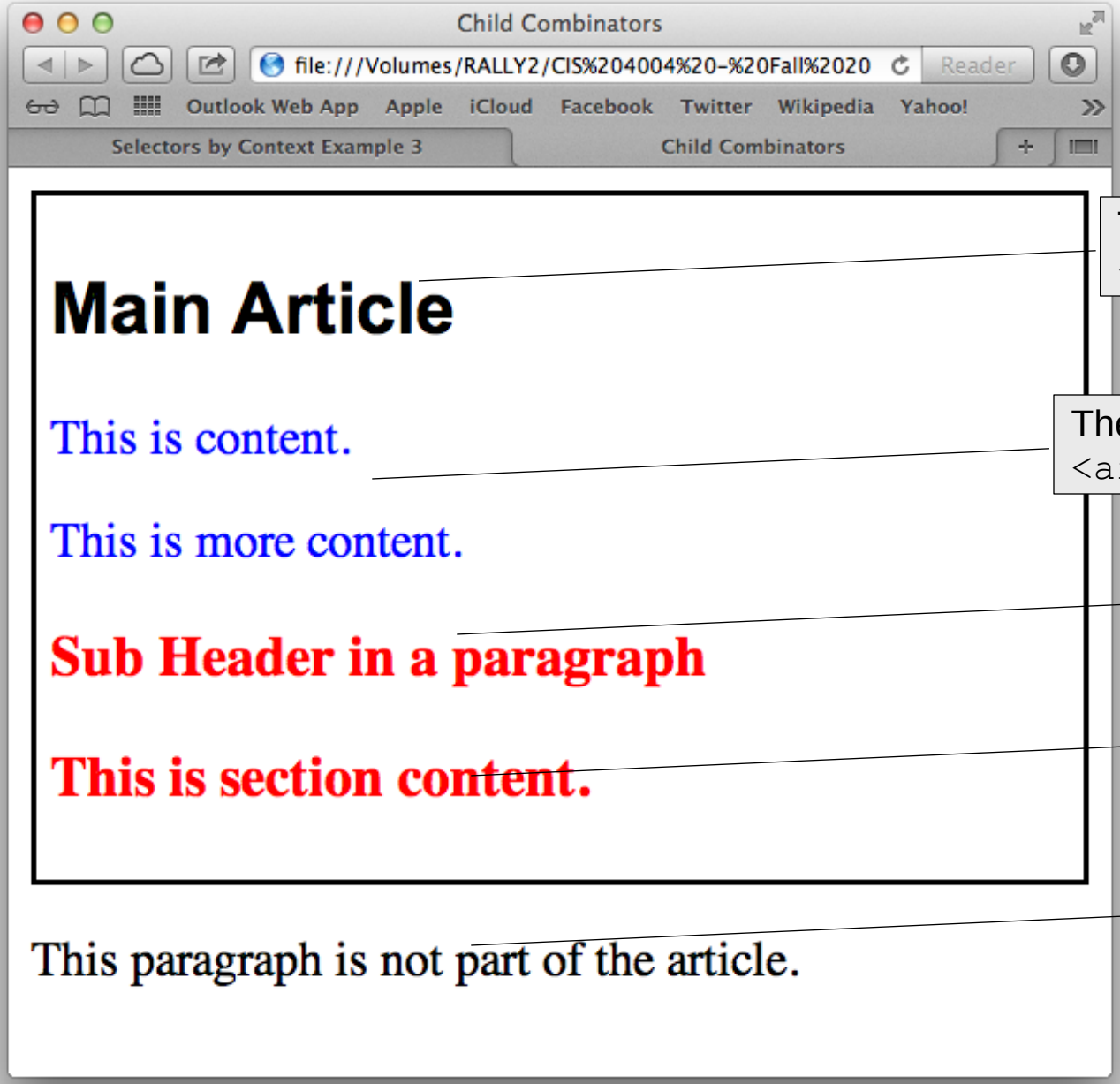
Only ancestor element is the `<body>` element

Selecting Elements By Context

- CSS3 also defines a `child combinator`, which allows you to define a rule for an immediate descendant (in other words, a child) of a parent element. These were called `child selectors` before CSS3.
- A parent is an element that directly contains another element (a child), meaning that they are only one generation away.
- A child combinator is defined as follows: `parent > child { . . . }`
- The example on the next page illustrates defining and using a child combinator.

child combinators.html — Edited

```
<!DOCTYPE html>
<html lang="en">
<head>
        <meta charset="utf-8" />
        <title> Child Combinators </title>
        <style>
        <!--
                article { border: 2px solid black; padding: 5px; }
                article > h2 { font-family: arial; }
                section > h3 > p { color: red; }
                article > p {color: blue; }
        -->
        </style>
</head>
<body>
        <article class="about">
                <h2> Main Article</h2>
                <p>This is content.</p>
                <p>This is more content.</p>
                <section class="notabout">
                        <h3><p>Sub Header in a paragraph</p</h3>
                        <p> This is section content.</p>
                </section>
        </article>
        <p>This paragraph is not part of the article.</p>
</body>
</html>
```

The parent element is the `<article>` element

The parent element is the `<article>` element

The parent element is the `<section><h3>` element

The parent element is the `<section>` element

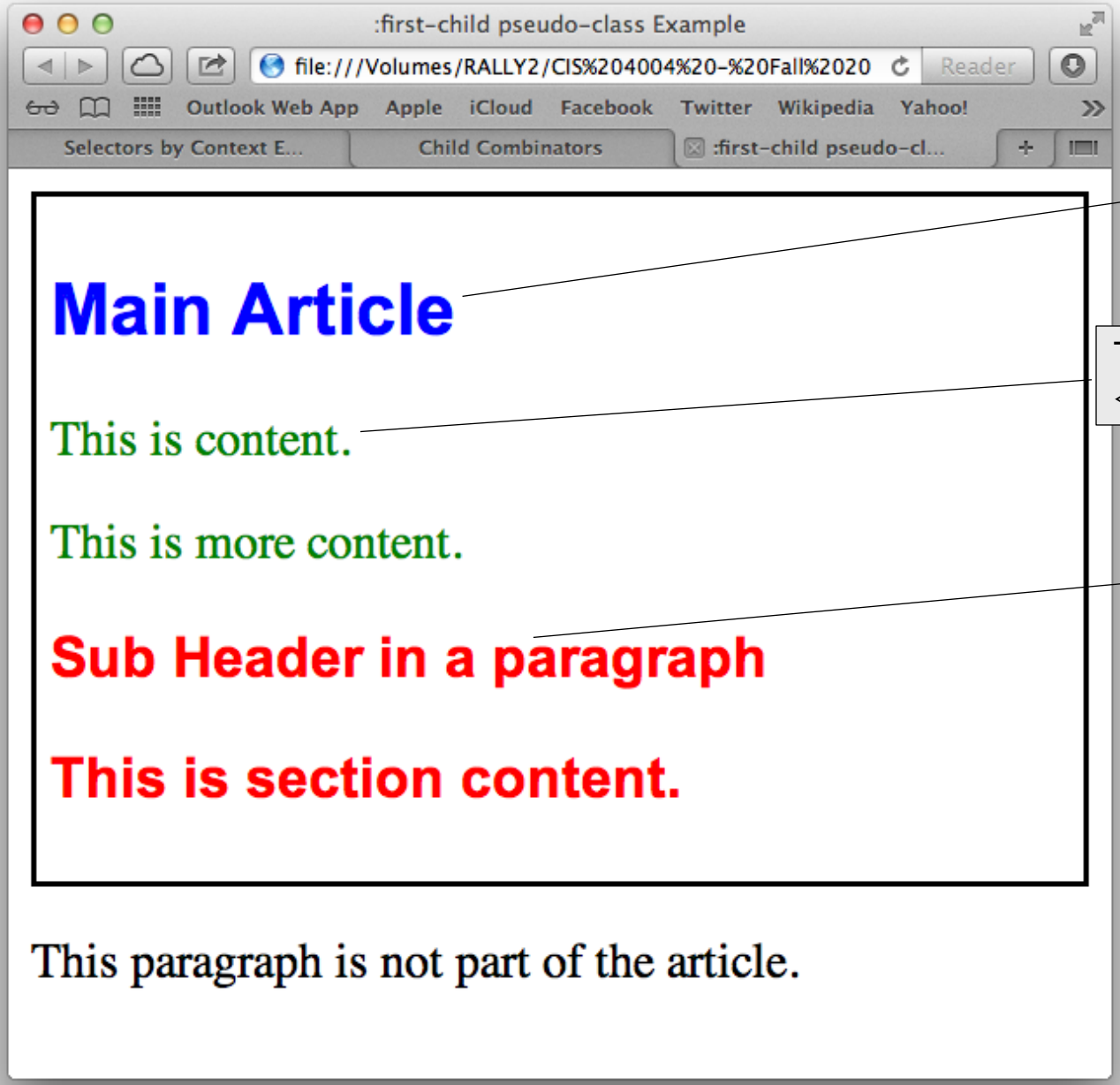
The parent element is the `<body>` element

Selecting Elements By Context

- Its sometimes useful to be able to select only the first child of an element, as opposed to all the children of an element.
- To do this use the `:first-child` pseudo-class. The first-child part of the selector is called a pseudo-class because it identifies a group of elements without you (the designer) having to mark them in the HTML5 markup.
- For example, you might want to style the first paragraph of an article differently than all of the other paragraphs in the article, so this would come in handy.
- The syntax for this is: `element:first-child { . . . }`
- The example on the next page illustrates the `:first-child` pseudo-class.

first-child pseudo-class.html

```
<!DOCTYPE html>
<html lang="en">
<head>
        <meta charset="utf-8" />
        <title> :first-child pseudo-class Example </title>
        <style>
        <!--
                article { border: 2px solid black; padding: 5px; }
                article > h2:first-child { font-family: arial; color: blue; }
                section > h3:first-child { font-family: arial; color: red; }
                article > p { color: green; }
        -->
        </style>
</head>
<body>
        <article class="about">
                <h2> Main Article</h2>
                <p>This is content.</p>
                <p>This is more content.</p>
                <section class="notabout">
                        <h3><p>Sub Header in a paragraph</p></h3>
                        <p> This is section content.</p>
                </section>
        </article>
        <p>This paragraph is not part of the article.</p>
</body>
</html>
```

The `<h2>` element is the first-child of its parent element `<article>` element

The parent element is the `<article>` element

The `<h3>` element is the first-child of its parent element `<section>` element

Selecting Elements By Context

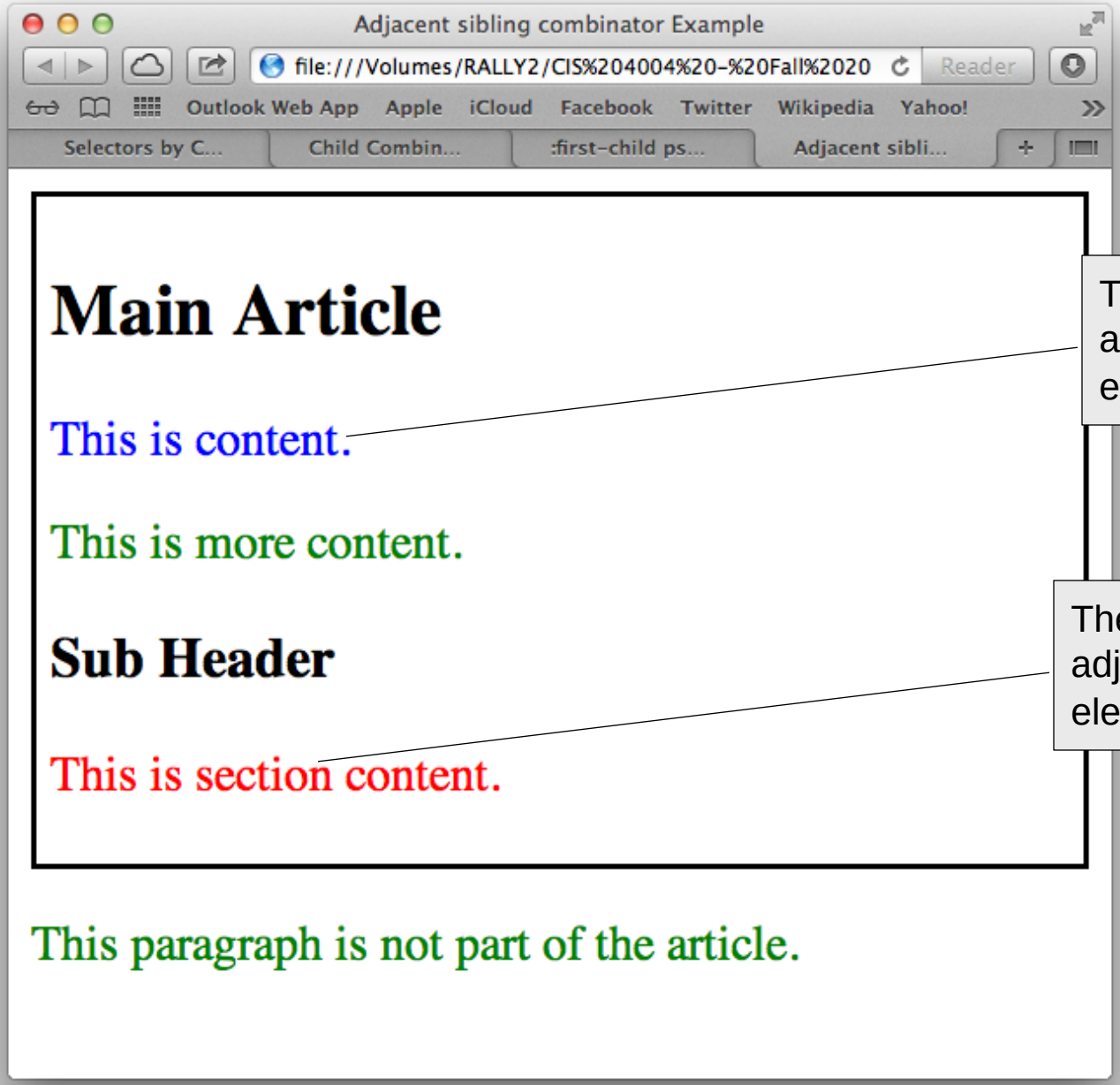
- Continuing with the familial theme, sibling elements are elements of any kind that are children of the same parent.
- Adjacent siblings are elements that are next to each other directly, meaning that no other sibling sits between them.
- The CSS3 `adjacent sibling combinator` allows you to target a sibling element that is preceded immediately by a sibling that you specify.

In the crude markup example shown on the right, the `<h1>` element and the `<p>` element are adjacent siblings, and the `<p>` element and the `<h2>` element are adjacent siblings, but the `<h1>` element and the `<h2>` element are not. However, they are all siblings (and children) of the `<body>` element.

```
<body>
        <h1>. . . </h1>
        <p> . . . </p>
        <h2>. . . </h2>
</body>
```


adjacent sibling combinator – example.html

```
<!DOCTYPE html>
<html lang="en">
<head>
        <meta charset="utf-8" />
        <title> Adjacent sibling combinator Example </title>
        <style>
        <!--
                article { border: 2px solid black; padding: 5px; }
                h2+p {color:blue; }
                h3+p {color:red; }
                p { color: green; }
        -->
        </style>
</head>
<body>
        <article class="about">
                <h2> Main Article</h2>
                <p>This is content.</p>
                <p>This is more content.</p>
                <section class="notabout">
                        <h3>Sub Header</h3>
                        <p> This is section content.</p>
                </section>
        </article>
        <p>This paragraph is not part of the article.</p>
</body>
</html>
```

The <p> element is the adjacent sibling of the <h2> element

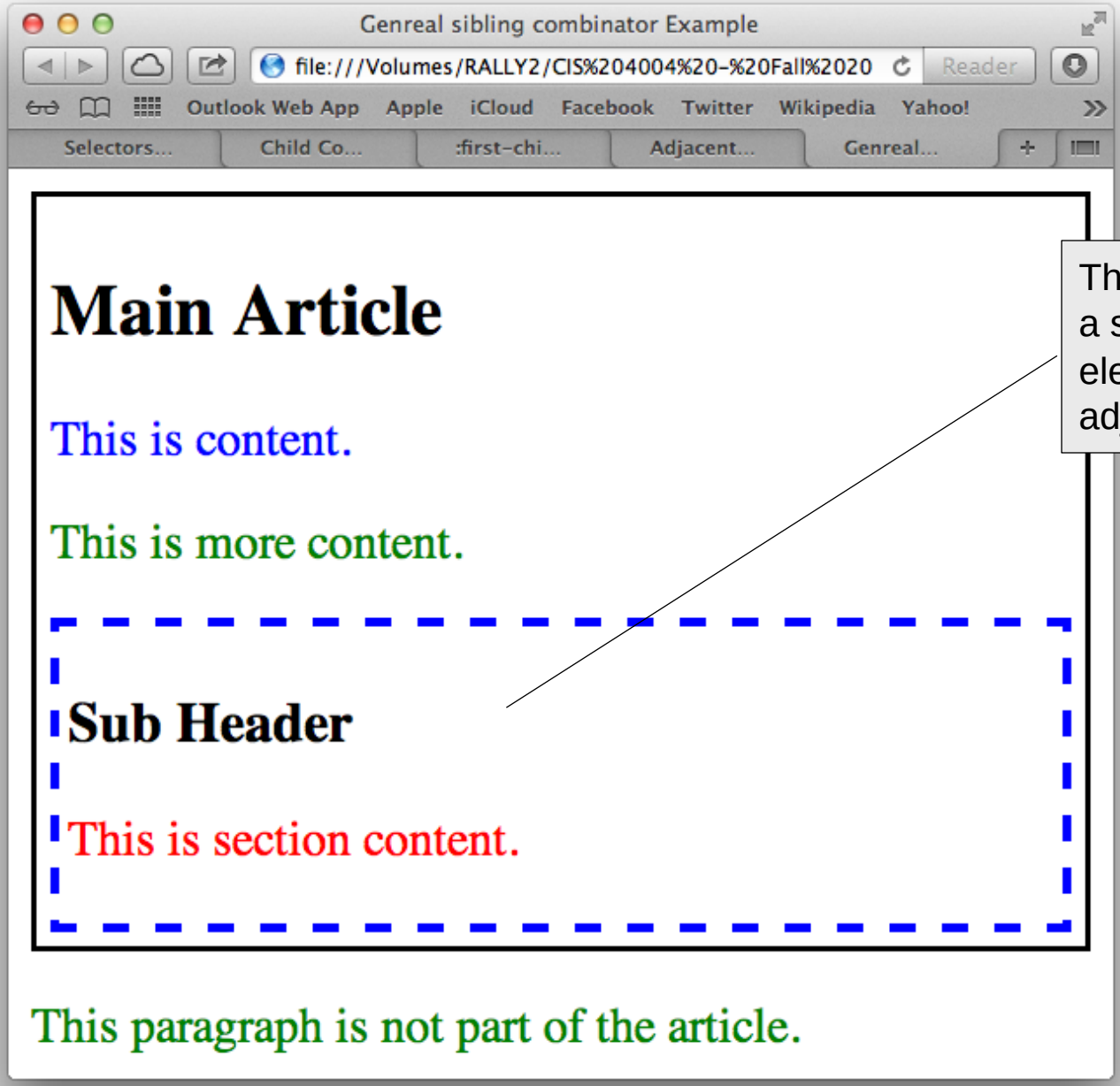
The <p> element is the adjacent sibling of the <h3> element

Selecting Elements By Context

- CSS3 introduces the `general sibling combinator`, which allows you to select a sibling that is not necessarily immediately preceded by another sibling.
- The only difference in the syntax is that the `+` sign of the `adjacent sibling combinator` is replaced by the tilde (~) sign in the `general sibling combinator`.
- For example: the rule `h1~h2 { color: blue;}` would make any `<h2>` element blue as long as it was preceded by a sibling `<h1>` element somewhere within the parent element. It could be immediately adjacent, but it does not need to be. The following example illustrates the `general sibling combinator`.

general sibling combinator example.html

```
<!DOCTYPE html>
<html lang="en">
<head>
        <meta charset="utf-8" />
        <title> Genreal sibling combinator Example </title>
        <style>
        <!--
                article { border: 2px solid black; padding: 5px; }
                h2+p {color:blue; }
                h3+p {color:red; }
                p { color: green; }
                h2~section {border: 3px dashed blue; padding: 3px;}
        -->
        </style>
</head>
<body>
        <article class="about">
                <h2> Main Article</h2>
                <p>This is content.</p>
                <p>This is more content.</p>
                <section class="notabout">
                        <h3>Sub Header</h3>
                        <p> This is section content.</p>
                </section>
        </article>
        <p>This paragraph is not part of the article.</p>
</body>
</html>
```

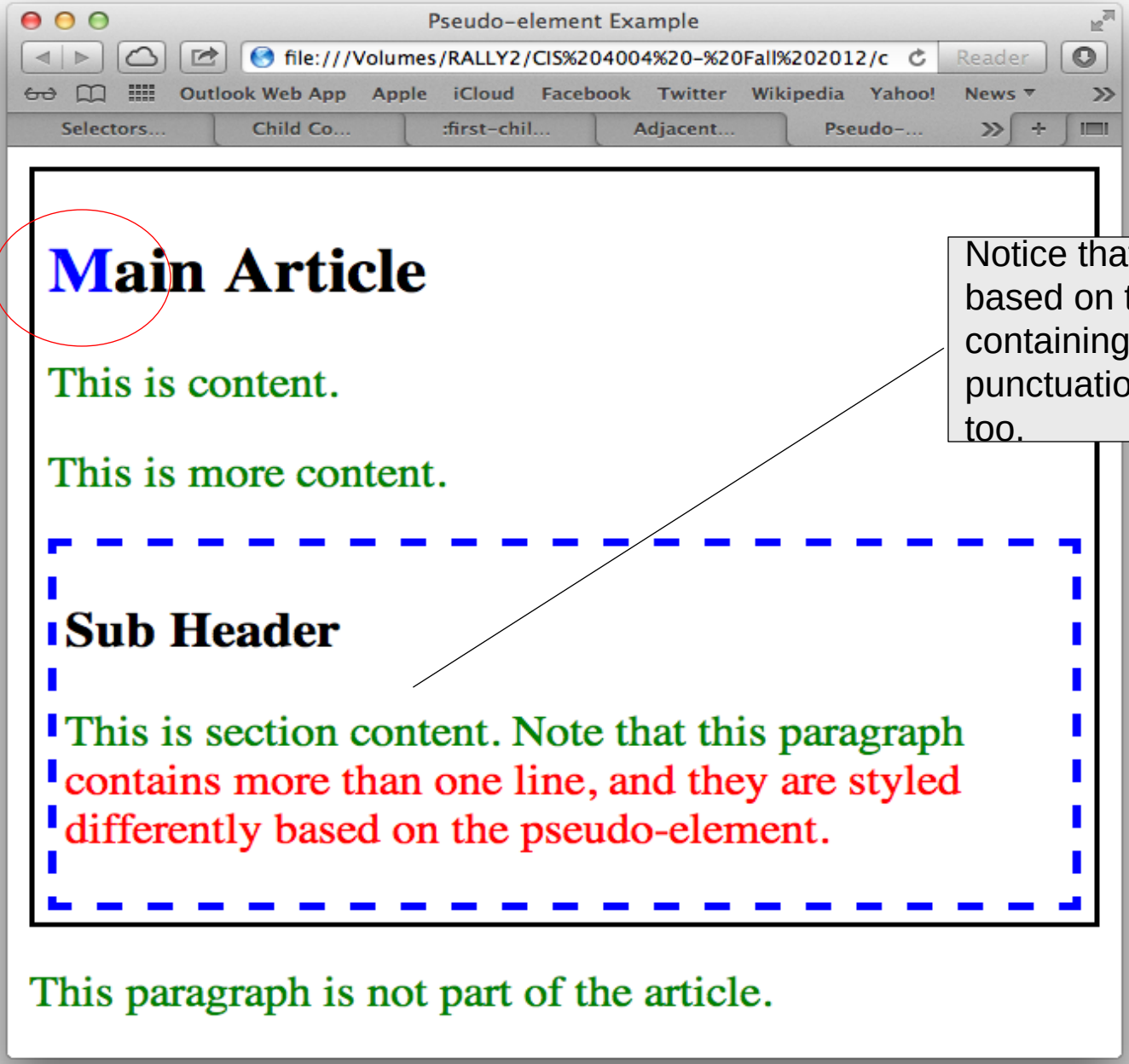
The `<section>` element is a sibling of the `<h2>` element, just not an adjacent one.

Selecting Elements By Context

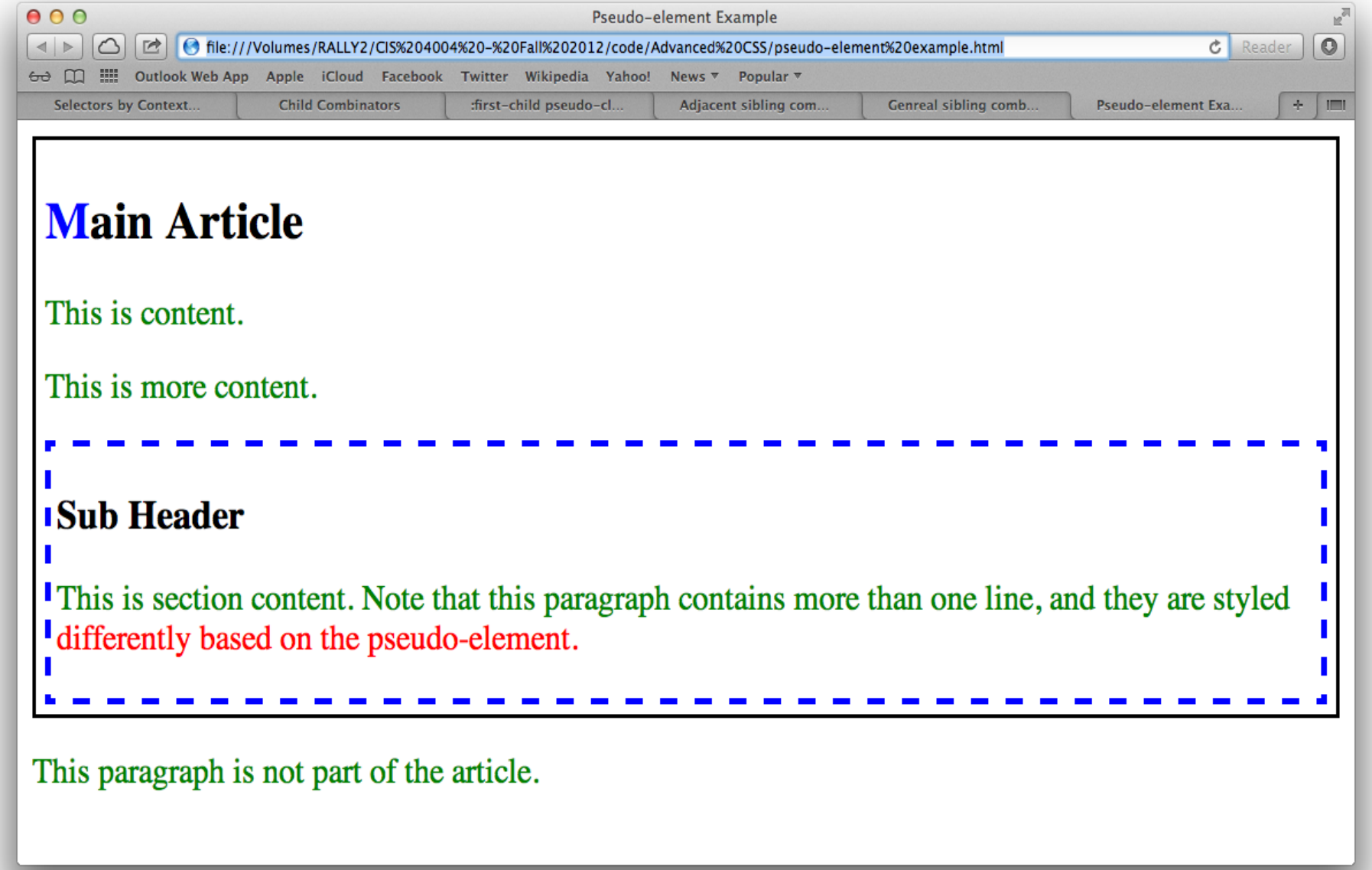
- CSS3 also allows you to select only part of an element, rather than include the entire element in the styling.
- This is done using new pseudo-elements of `::first-word`, `::first-line`, `::before`, and `::after`. Note the double colon which is used to distinguish the new pseudo-elements from the existing pseudo-classes such as `:link`, `:hover`, etc., which use a single colon syntax.
- The `::first-line` pseudo-element selects the first line of the selected element. The `::first-letter` pseudo-element selects the first letter of the selected element. We'll look only at these two pseudo-elements for the time being. The example on the following page illustrates.

pseudo-element example.html

```
<!DOCTYPE html>
<html lang="en">
<head>
        <meta charset="utf-8" />
        <title> Pseudo-element Example </title>
        <style>
        <!--
                article { border: 2px solid black; padding: 5px; }
                h2::first-letter {color:blue; }
                h3+p {color:red; }
                p::first-line { color: green; }
                h2~section {border: 3px dashed blue; padding: 3px;}
        -->
        </style>
</head>
<body>
        <article class="about">
                <h2> Main Article</h2>
                <p>This is content.</p>
                <p>This is more content.</p>
                <section class="notabout">
                        <h3>Sub Header</h3>
                        <p> This is section content.  Note that this paragraph
                        contains more than one line, and they are styled differently
                        based on the pseudo-element.</p>
                </section>
        </article>
        <p>This paragraph is not part of the article.</p>
</body>
</html>
```

Notice that the "first-line" is based on the size of the containing element and not punctuation. See next page too.


Pseudo-element Example
file:///Volumes/RALLY2/CIS%204004%20-%20Fall%202012/code/Advanced%20CSS/pseudo-element%20example.html
Reader
Outlook Web App
Apple
iCloud
Facebook
Twitter
Wikipedia
Yahoo!
News
Popular
Selectors by Context...
Child Combinators
:first-child pseudo-cl...
Adjacent sibling com...
Genreal sibling comb...
Pseudo-element Exa...
Main Article
This is content.
This is more content.
Sub Header
This is section content. Note that this paragraph contains more than one line, and they are styled differently based on the pseudo-element.
This paragraph is not part of the article.

Defining Styles Based on Tag Attributes

- Although style attributes should all be handled by CSS, many HTML5 elements still have attributes that define how they behave. For example, the image tag, `img`, always includes the `src` attribute to define the source for the image file to be located.
- Styles can also be applied to an HTML element based on an attribute or an attribute value, allowing you to set styles if the attribute has been set, is or is not a specific value, or contains a specific value.
- A table of the possible attribute selector variations is shown on the next page.
- An example using attribute selectors appears on the page after the table.

Defining Styles Based on Tag Attributes

Format	Name	Elements that are styled if that element
`[attr]`	Attribute	Has specified attribute
`[attr="value"]`	Exact value	Has specified attribute with exact value
`[attr~="value"]`	Spaced list	Has specified attribute equal to exact value within space-separated list
`[attr|="value"]`	Hyphenated list	Has specified attribute equal to exact value within hyphen-separated list
`[attr^="value"]`	Begins with	Has specified attribute equal to exact value at beginning
`[attr$="value"]`	Ends with	Has specified attribute equal to exact value at end
`[attr*="value"]`	Contains	Has specified attribute equal to exact value anywhere

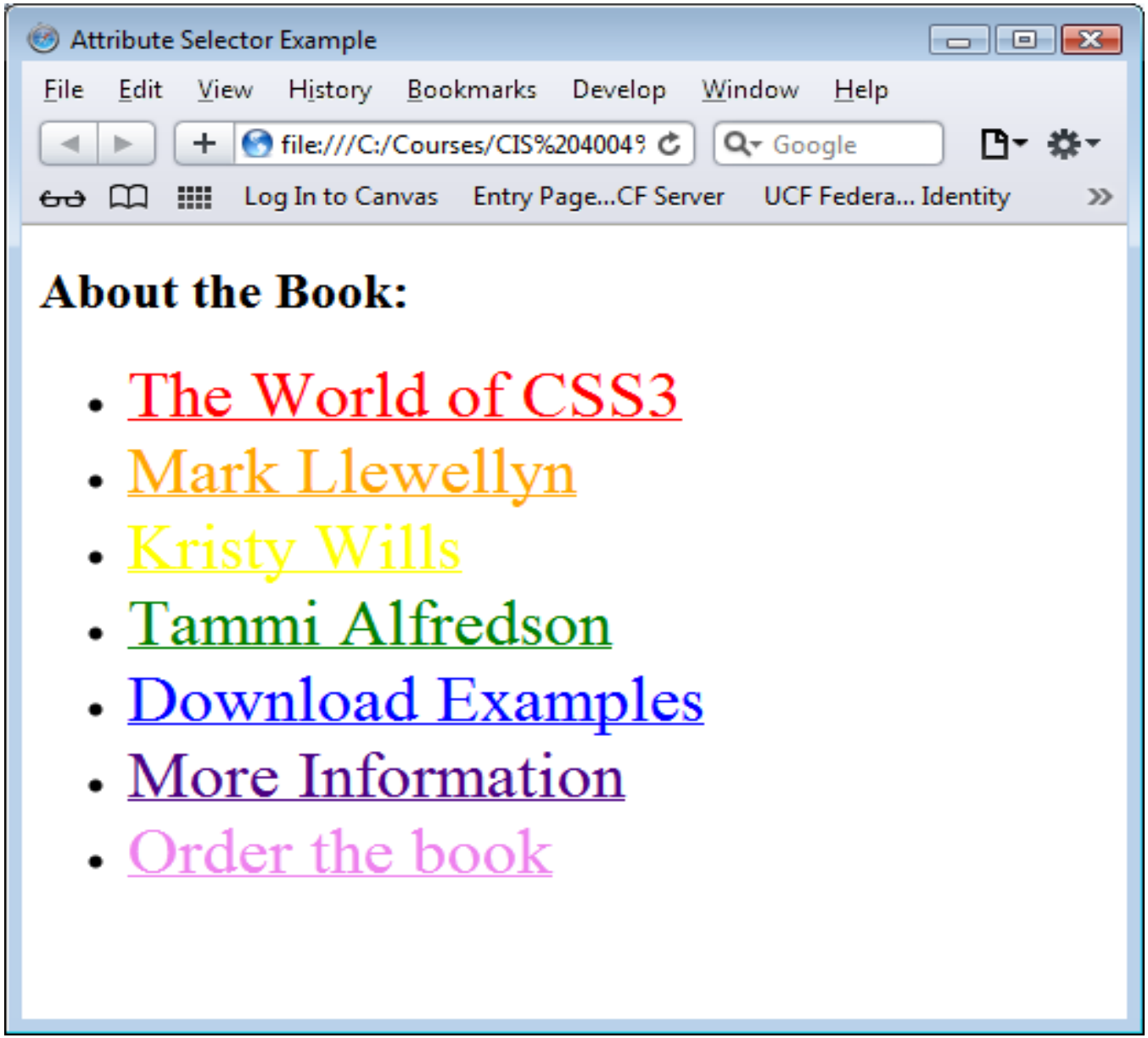
C:\Courses\CIS 4004 - Web-Based Info Tech\Spring 2013\code\Advanced CSS\attribute selector example.html - Notepad++

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

project2css.css | project2.htm | attribute selector example.html

```
7       <!--
8           a { display:block; font-size:2em; }
9           a[title] { color:red; }
10          a[title="Author"] {color: orange; }
11          a[title~="white"] {color:yellow; }
12          a[title|="illustrations"] {color:green; }
13          a[href^="http://"] {color:blue; }
14          a[href*="order"]{color:indigo; }
15          a[href$="css3-now"]{color:violet;}
16      -->
17      </style>
18  </head>
19  <body>
20      <article class="chaptertext">
21          <h1>About the Book:</h1>
22          <ul>
23              <li><a href="index.html" title="The World of CSS3"> The World of CSS3</a></li>
24              <li><a href="index.html" title="Author">Mark Llewellyn</a></li>
25              <li><a href="index.html" title="Illustrations black and white">Kristy Wills</a></li>
26              <li><a href="index.html" title="illustrations-in-color">Tammi Alfredson</a></li>
27              <li><a href="http://www.markllewellyn.com">Download Examples</a></li>
28              <li><a href="http://www.markllewellyn.com/css3-now/order">More Information</a></li>
29              <li><a href="http://www.markllewellyn.com/css3-now">Order the book</a></li>
30          </ul>
31      </article>
32  </body>
33  </html>
```

Hyper Text Markup Language fil | length : 1139 lines : 33 | Ln : 33 Col : 8 Sel : 0 | Dos\Windows | ANSI | INS


Attribute Selector Example
File Edit View History Bookmarks Develop Window Help
file:///C:/Courses/CIS%204004?
Google
Log In to Canvas
Entry Page...CF Server
UCF Federa... Identity
About the Book:
The World of CSS3
Mark Llewellyn
Kristy Wills
Tammi Alfredson
Download Examples
More Information
Order the book