Introduction to HTML5

23rd September

Introduction

- Web developers views a web page as a document that must be created according to authoring and development guidelines
- Web developers use HTML or XHTML to write code of a web page
- Web browsers have a built-in interpreter to render the results of a code in its window

The World Wide Web

- Definitions
  - **HTML**
    - The HyperText Markup Language
    - The language used to design web pages
  - **XHTML**
    - The eXtensible HyperText Markup Language
    - Intended to enhance HTML, not replace it
- World Wide Consortium (W3C) organisation is responsible for developing web standards

Content of Web Pages

- The content of a web site can be classified as:
  - Static content – does not change regularly e.g. personal and professional web sites
  - Dynamic content – changes regularly e.g., banking web sites, weather reports…

Authoring of Web Pages

- After a web page is designed one needs to use HTML/XHTML code to author a web page

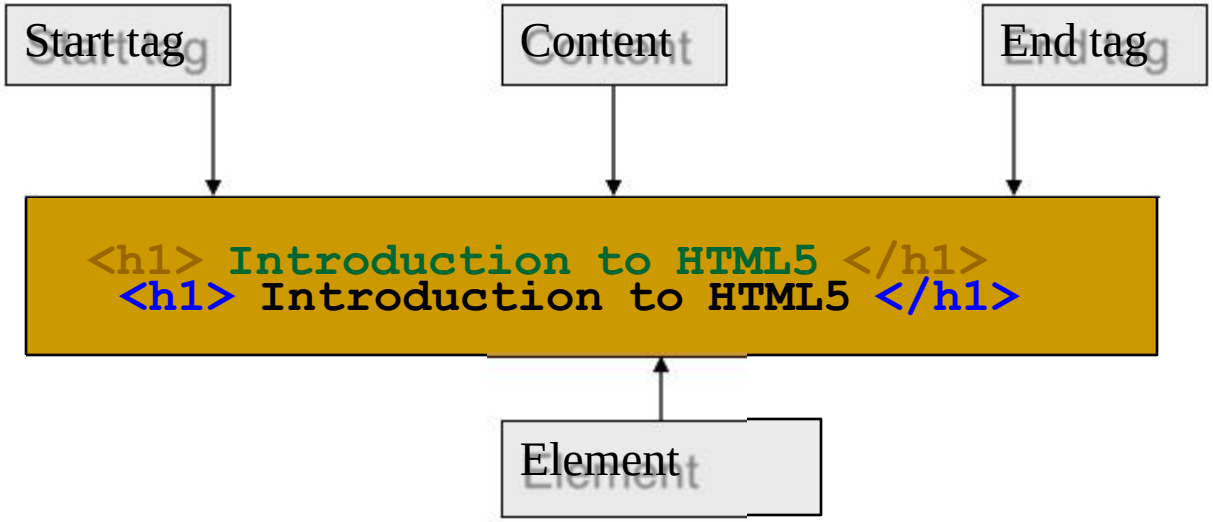
XHTML Document Building Blocks

- Like any language, HTML has a number of building blocks that are used to create complete documents.
  - The English language, for example, is made up of nouns, verbs, adjectives, adverbs, prepositions, and so on, which are used in conjunction with each other to form sentences and paragraphs.
- Both HTML and XHTML provide language building blocks that can be added to any text document. Web browsers know how to interpret these elements in order to present the document based on formatting rules.

XHTML Elements

- XHTML elements are the core components of XHTML documents and are used to describe the data in a document.
  - Elements are like nouns in the English language.
- Elements are the markup, or formatting instructions, of the XHTML document.
- Elements define the text styles, formatting links, and other pieces of the document.
- Elements and tags are sometimes used interchangeably, but strictly speaking, a tag is a piece of an element.

HTML5 Elements



- All elements, except for empty elements, consist of three pieces: a start tag, content, and an end tag.
- HTML5 element names by convention are written in lowercase letters.

HTML5 Elements

- Empty elements are used primarily to describe pieces of data that don't contain any content.
- For example, some common empty elements in HTML5 are:

  `<br>` for line break

  `<img>` for image

  `<p>` for paragraph

- In HTML5, all elements must have a start tag and end tag.

Attributes

- HTML5 attributes are pieces of information that help to describe elements. Some elements have required attributes, others are optional and depend on the content that is being marked up.
- Attributes are referred to as name-value pairs and have the following syntax: The name of the attribute is on the left, followed by an equal sign, then the value.

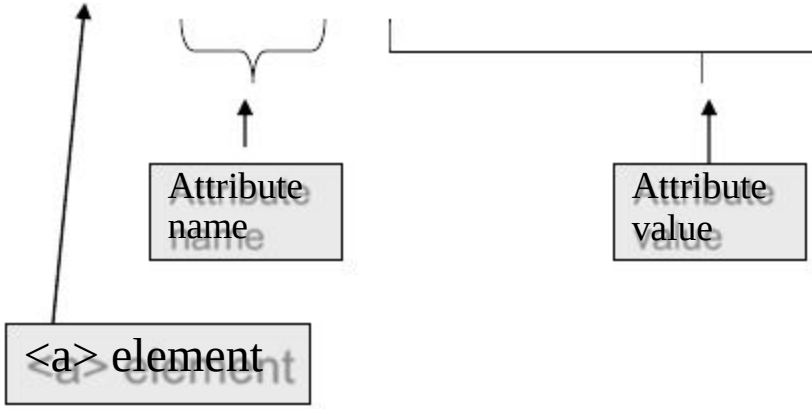
```
name = "value"
```

- Here are a few examples:

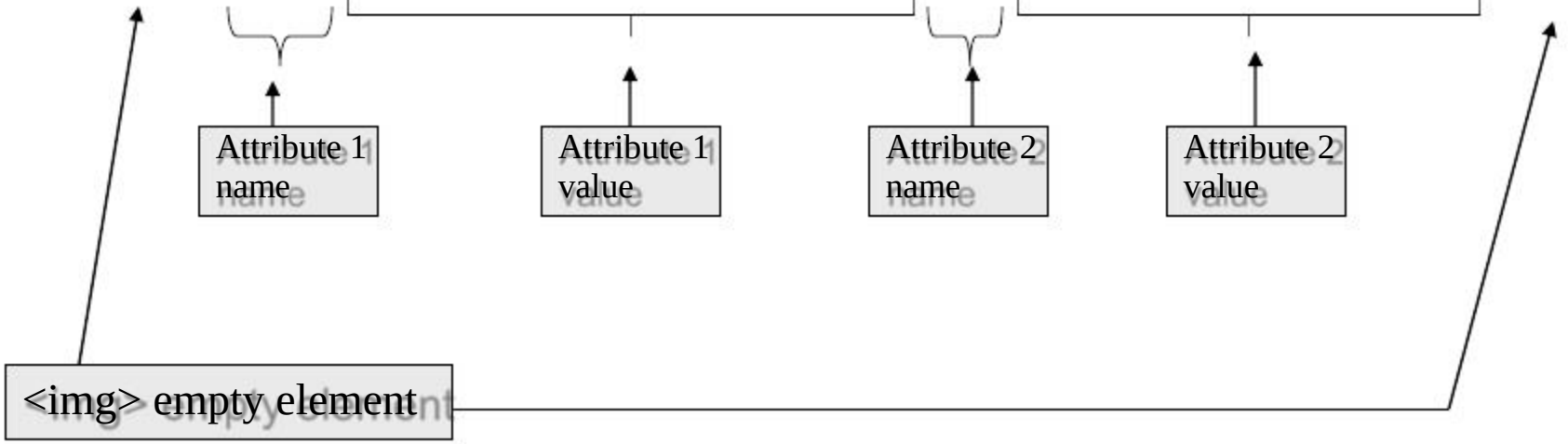
```
<a href="http://mark.com">Click here</a>

<img src="/images/pictures.gif" id="Picture of House" />
```

Attributes

```
<a href="http://mark.com">Click here</a>
```




```
<img src="/images/pictures.gif" id="Picture of House"   />
```



Attribute Rules

- We'll see more rules later, but here are a few rules about HTML5 attributes:

1. Attributes are always contained within the start tag of an element.
2. Names must be in lowercase letters.
3. Attributes must have a value that is surrounded by quotes.

HTML Core Attributes

- In the HTML5 specification, there are a set of core attributes that can be used with most XHTML elements.
- They cannot be used in `<base>`, `<head>`, `<html>`, `<meta>`, `<param>`, `<script>`, `<style>`, and `<title>` elements.
- The core attributes are:

1. **`id`** – document wide unique id.
2. **`class`** – list of classes of the element.
3. **`style`** – associated style information.
4. **`title`** – advisory title amplification.

HTML5 Comments

- Comments in HTML5 are notations that are ignored by programs and parsers.
- The syntax of an HTML5 comment is identical to XHTML.
- You can use comments to document your code, add additional information about a piece of data, add visual breaks, or add information that other people working on or using your document would find useful.
- The following is an HTML5 comment:

```
<!--  This is a comment  -->
```

Creating Your First HTML5 Document

- We'll begin creating our first HTML5 document by examining a plain text document that we would like to markup with formatting elements.
- Using NotePad++ or some similar text editor, create a file with the name "`unformatted.html`", that includes the text shown below (type it exactly as it appears below):

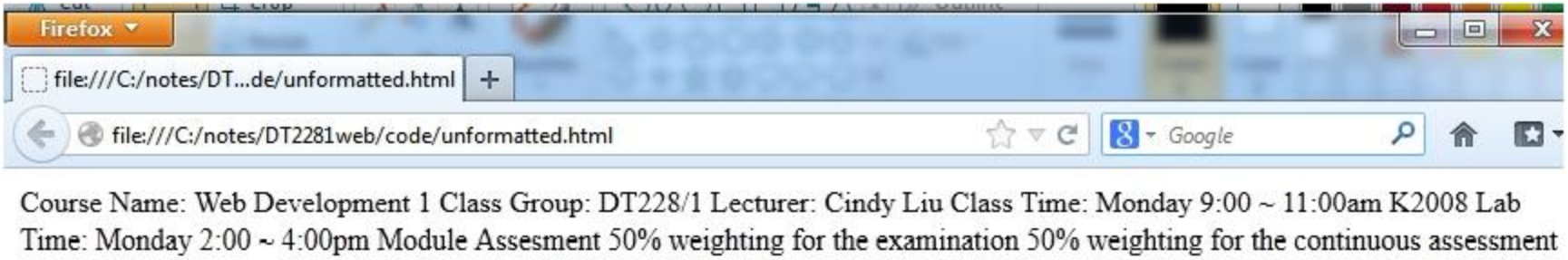
Course Name: Web Development 1
Class Group: DT228/1
Lecturer: Cindy Liu
Class Time: Monday 9:00 ~ 11:00am K2008
Lab Time: Monday 2:00 ~ 4:00pm

Module Assessment
50% weighting for the examination
50% weighting for the continuous assessment

Unformatted Text Document – name it "unformatted.html"

Creating Your First XHTML Document

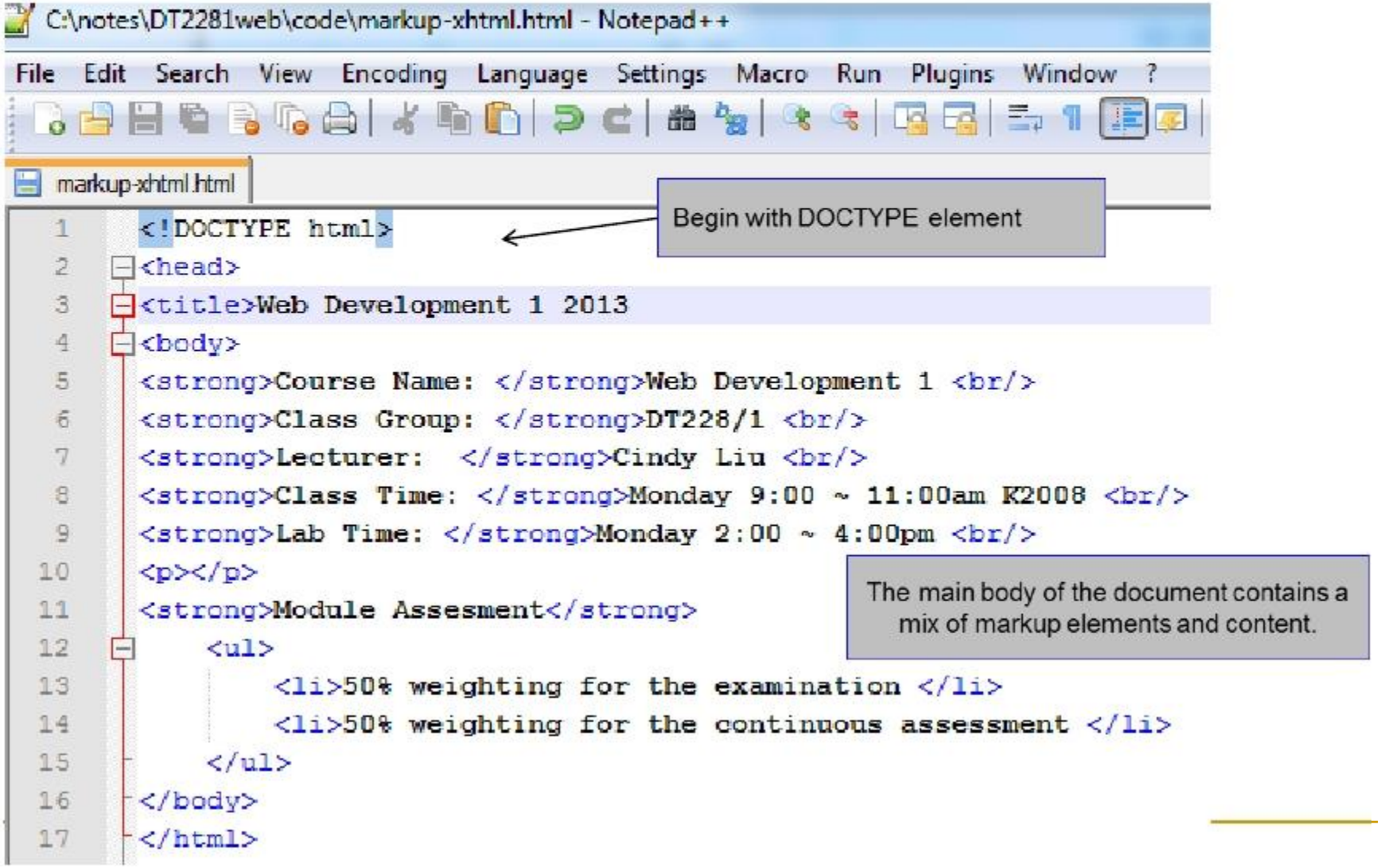
- Next, open your Web browser and load the file "unformatted.html" that you just created. The screen shot below, shows what the file looks like in Firefox.



Creating Your First HTML5 Document

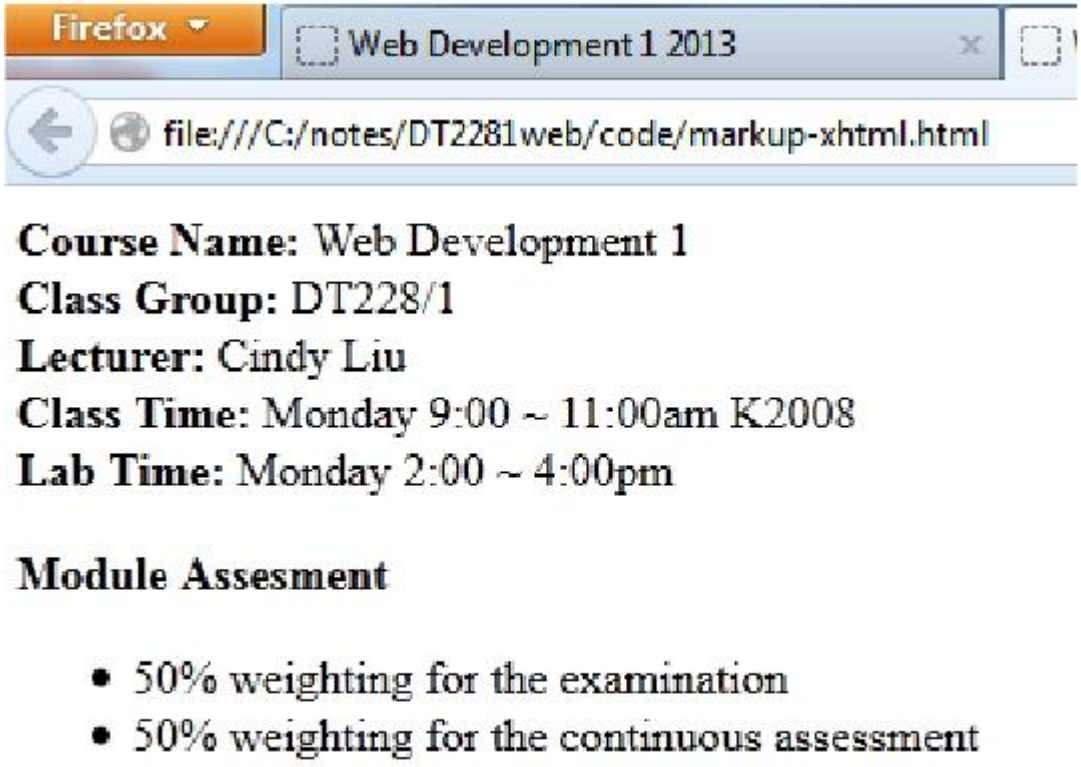
- The screen shot on the previous page illustrates how the Web browser can open the document, but without formatting instructions, it does not know how to correctly format the document.
- Web browsers ignore all whitespace characters, including line breaks, so without the proper markup, our document displays as just a block of text.
- Next, we'll add some formatting elements (markup) to our document.
- **NOTE:** Do not type the line numbers in your document, they are for only used in these notes to make references to specific lines.

The HTML5 markup



Rendering/Viewing The HTML5 Markup

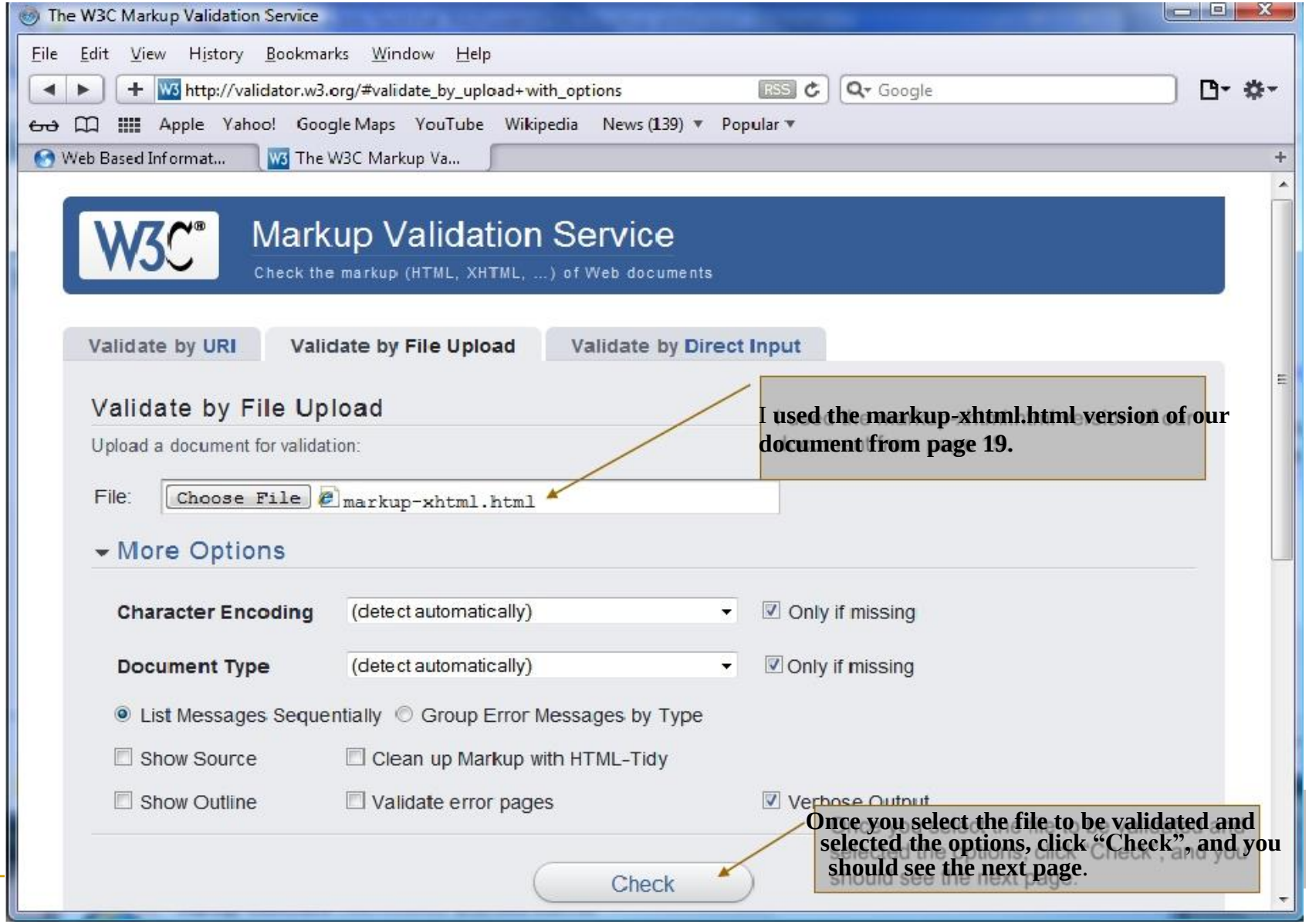
- Create a new file that contains the text from the previous page (remember – do not type in the line numbers). I called my version of this file: "`markup-xhtml.html`", but you can call it whatever you would like. Next, open your Web browser and load the file "`markup-xhtml.html`" that you just created. The screen shot below, shows what the file looks like in Firefox.



Validating An HTML5 Document

- While there are many free and commercial programs available, one of the simplest ways to check if your XHTML documents are well-formed and valid is to use the free on-line validating tool from the W3C Web site. It can check your document either from your computer or from a website.
- The W3C markup validation service is available at: http://validator.w3.org.
- The next page shows the first step in using the file upload version of the validator.

Validating An HTML5 Document



HTML5 Code With Errors – Can You Find Them?

C:\notes\DT2281web\code\markup-xhtml.html - Notepad++

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

markup-xhtml.html

```
1   <!DOCTYPE html>
2   <head>
3   <title>Web Development 1 2013
4   <body>
5   <strong>Course Name: </strong>Web Development 1 <br/>
6   <strong>Class Group: </strong>DT228/1 <br/>
7   <strong>Lecturer:  </strong>Cindy Liu <br/>
8   <strong>Class Time: </strong>Monday 9:00 ~ 11:00am K2008 <br/>
9   <strong>Lab Time: </strong>Monday 2:00 ~ 4:00pm <br/>
10  <p></p>
11  <strong>Module Assesment</strong>
12      <ul>
13          <li>50% weighting for the examination </li>
14          <li>50% weighting for the continuous assessment </li>
15      </ul>
16  </body>
17  </html>
```

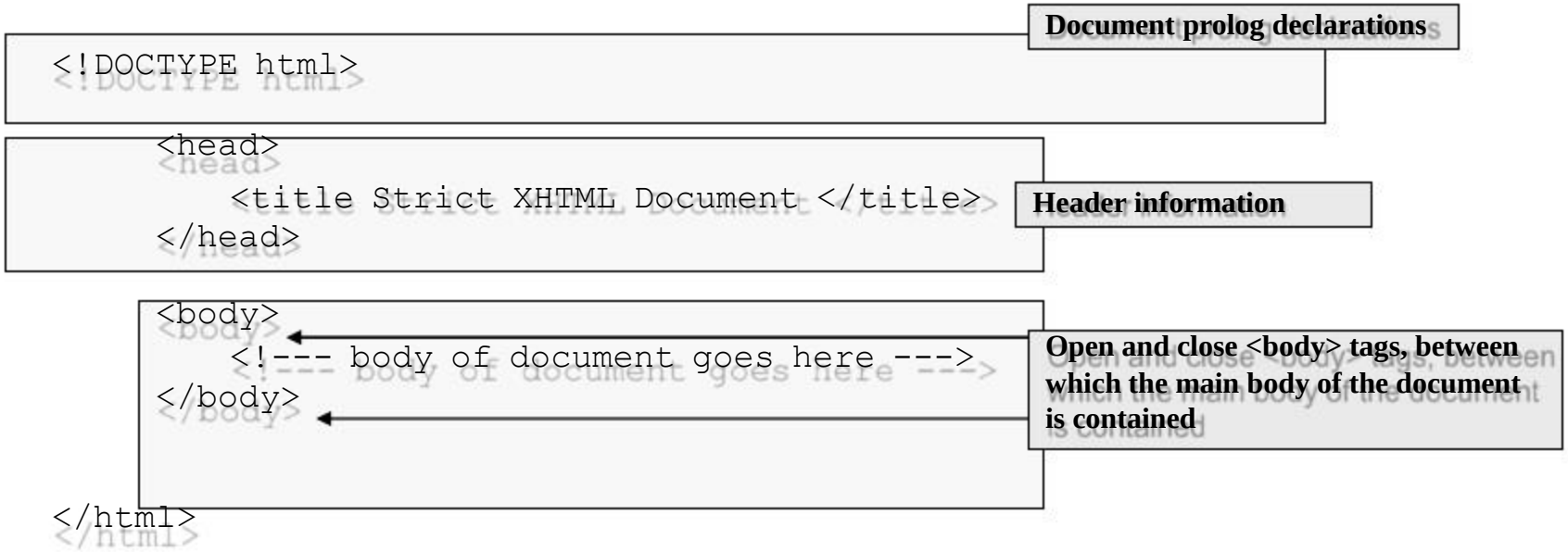
Missing end tag </title>

A Few Words On Text Editor

- Creating an XHTML document can be done in any text editor, such as Notepad.
- If you must use a word processing program, be sure that it allows you to enter text only documents and be sure that you set the program for this mode when entering XHTML documents. You do not want hidden word processing characters to appear in your XHTML documents because most browsers will not be able to correctly interpret these codes. This will lead to questionable display characteristics.
- Notepad++ is a good text editor that was developed to be syntax friendly for many different programming/markup languages including XHTML. You can download Notepad++ at: http://notepad-plus.sourceforge.net/uk/site.htm.

How HTML5 Documents Are Structured

- XHTML documents are comprised of a simple three-part framework:
  1. Document prolog
  2. Header section
  3. Body of document.



Document Framework Elements

- The elements that make up the framework of HTML5 documents do not produce any output in a browser window. Instead, they provide information to the program about the document.
- The elements that makeup the framework are: <html>, <head>, <title>, and <body>. We'll look at each of these more closely.

The <html> Element

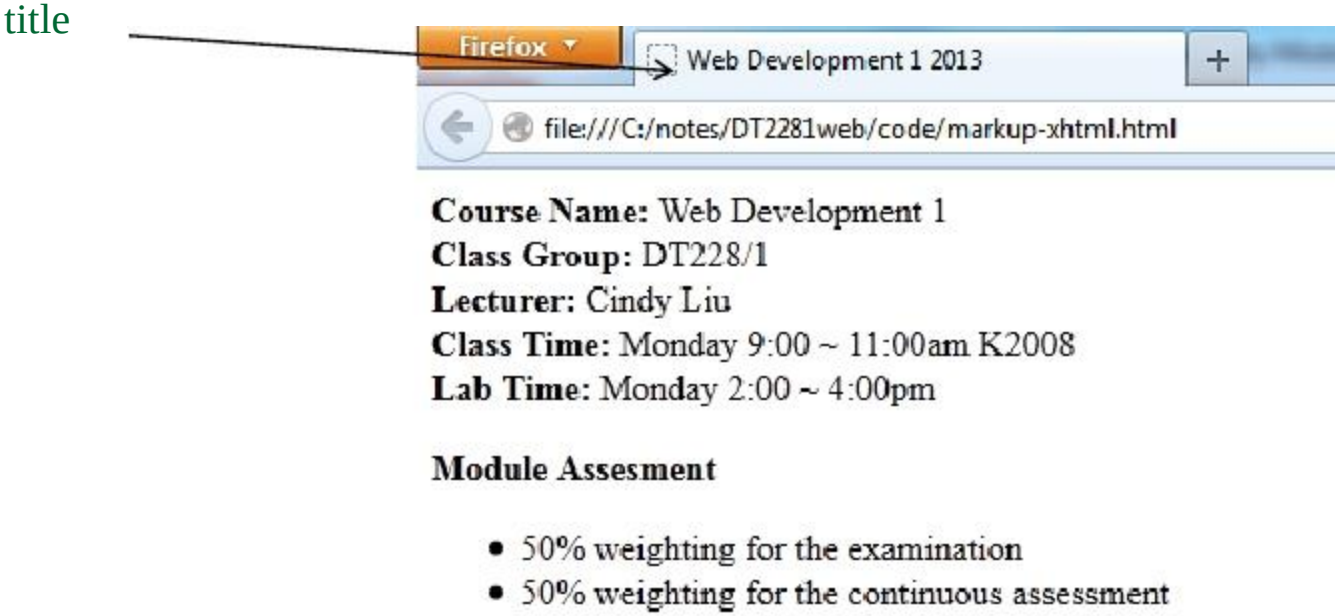
- The <html> element is the root element of an HTML5 document, within which every other element in the document is contained (recall our discussion on nesting of elements).
- The document begins with the <html> start tag and ends with the </html> end tag. The header and body information of the document are contained in the root element.

The <head> Element

- The start <head> tag comes directly after the <html> start tag in an HTML5 document. This element must be placed inside the <html> element.
- It contains information about the document that is mainly used by programs, such as keywords for search engines and link information that defines the relationship this document has to other documents.
- The <head> tag also contains the required <title> element. The following is a list of elements that can be contained in the <head> element – all but the <title> element are optional.
  - <title> - defines the title of the document.
  - <base> - defines the document base URL, which is used for relative links in the document.
  - <link> - defines the relationship of this document to other documents.
  - <meta> - defines additional information about the document, including the document's content type and special instructions for browsers and search engines.
  - <script> - defines links to scripts used within the document, such as Javascript.
  - <style> - defines links to style sheets to be used within the document, such as CSS.

The <title> Element

- The <title> element is the only required element within the <head> element. It must be contained within the open and close tags of the <head> element.
- There can only be one <title> element per document. It defines the title of the document that is displayed in the title bar of the browser window as well as the name of bookmarks to that page.
- Most search engines use the content of the <title> element as the text to display on their results pages as well.



The <body> Element

- The <body> element contains the content and all of the markup elements of the document.
- The body of the document is contained between the open <body> tag and the ending </body> tag.
- All of the other elements we will cover are contained within the <body> element.

Basic Formatting Elements In HTML5

- Now that you understand how HTML5 documents are structured, let's start building some Web pages. We'll start with basic formatting elements, show some examples of how to use each of the elements, and create a few documents to illustrate how they look in a Web browser.
- We'll start by looking at block-level formatting elements.

Block-level Formatting Elements – Summary Chart

Element Name	Formatting Style
`<p>...</p>`	Paragraph element
` `	Line break (empty element)
`<h1>...</h1>` to `<h6>...</h6>`	Heading elements (1 is largest, 6 is smallest)
`<hr />`	Horizontal rule (empty element)
`<div>...</div>`	Section divider

Block-level Formatting Elements

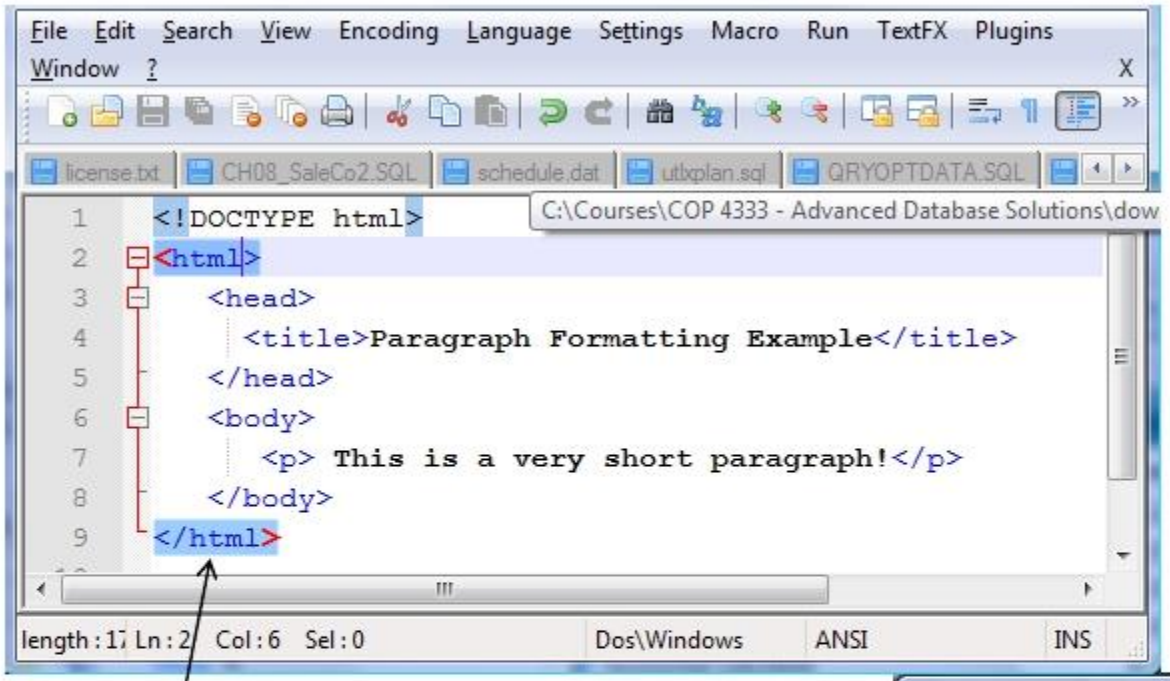
- Documents are broken into logical sections based on the document content to make it easer for users to read.
- The elements shown in the table on the previous page are used to break documents into logical chunks and to label the main content headings.
- These elements are referred to as block-level elements because they describe blocks of content.
- We'll examine each of these elements separately.

The Element

- The element divides content into paragraphs.
- The tag designates the beginning of a paragraph, and the tag ends the paragraph.
- Most browsers will automatically insert a double line return (carriage return) around the paragraph element.
- Example:

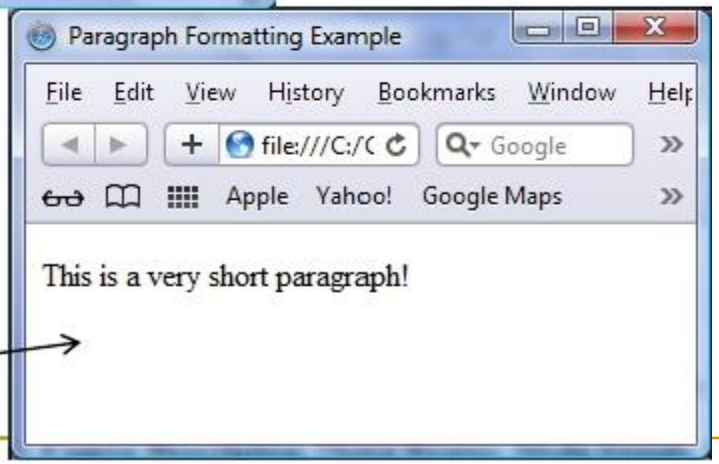
```
<p>This is a very short paragraph.</p>
```

The <p> Element



HTML5

Rendering

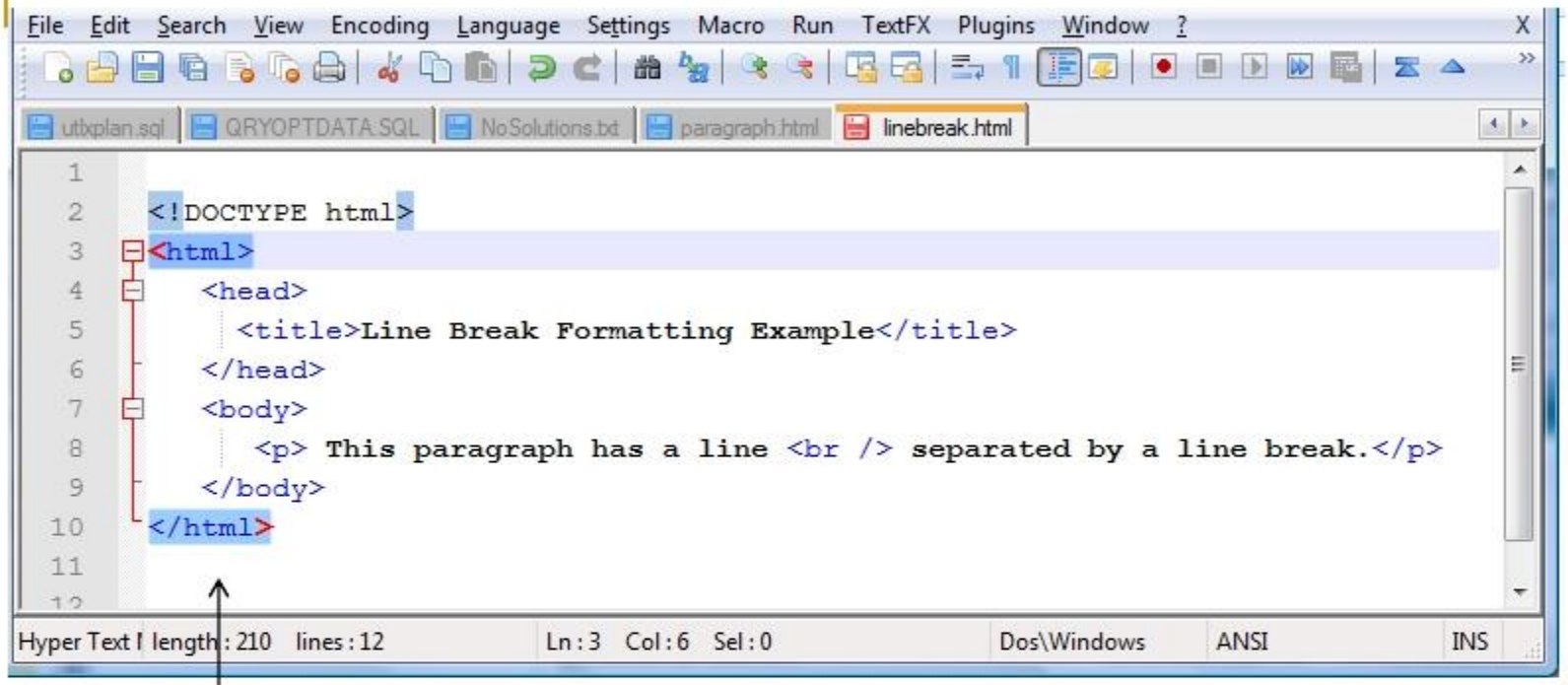


The
 Element

- The <br /> element is the line break element. Similar to the <p> element, it is used to break up sections of text.
- The <br /> element causes the browser to create a single line return (single carriage return).
- The <br /> element is an empty element and must end with /> in order to conform to the rules of a well-formed document.
- Example:

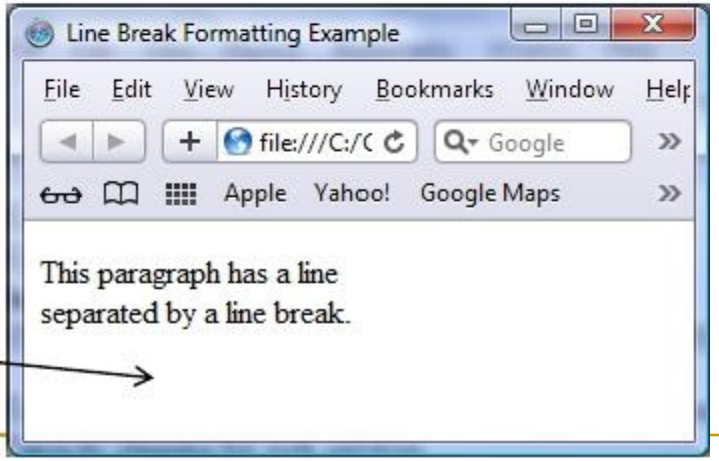
```
<p> This paragraph has a line <br />

    separated by a line break.</p>
```

The
 Element



HTML5

Rendering

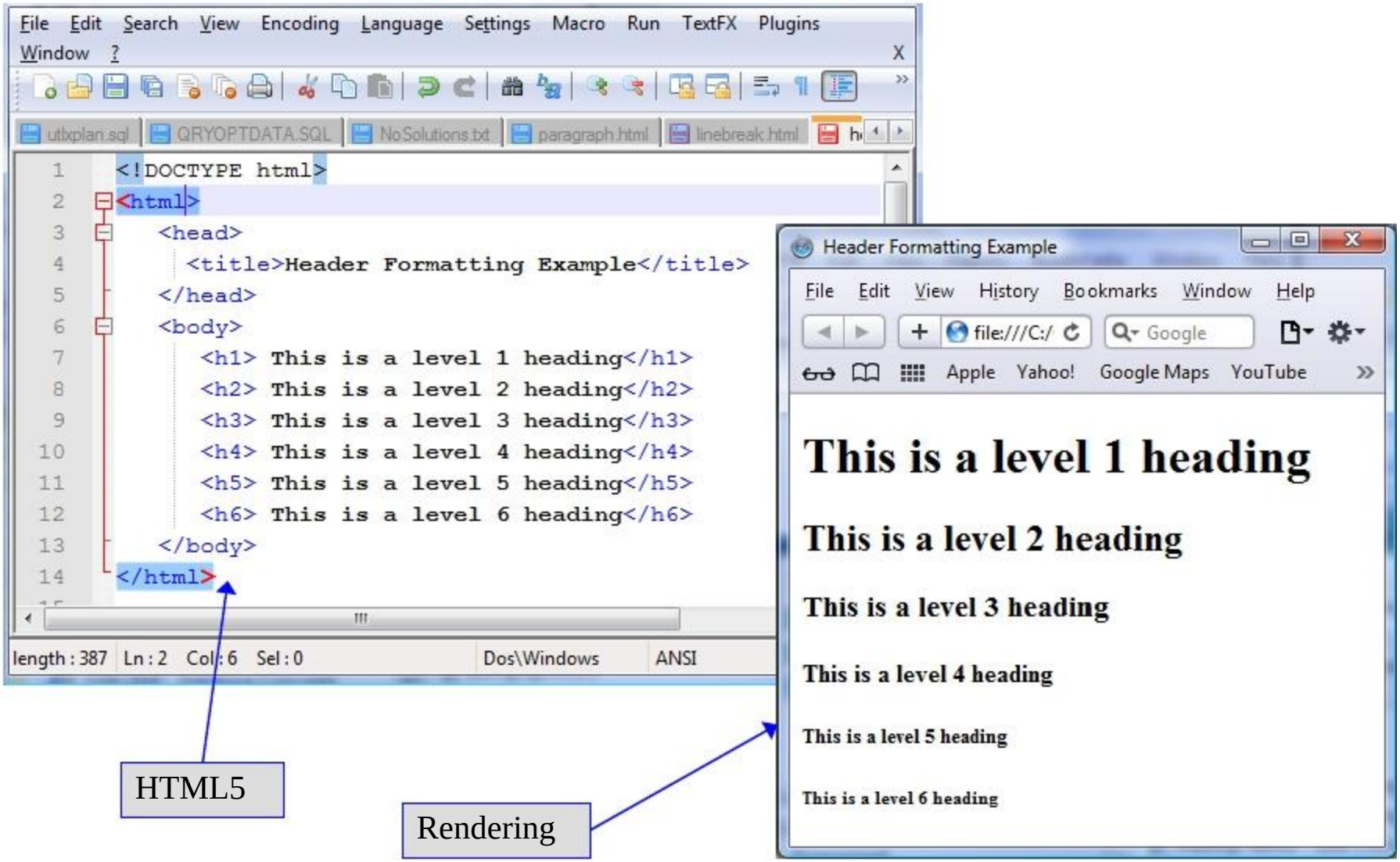


The <h1> . . . <h6> Elements

- These elements are the heading elements. They are used to label section headings of a document.
- There are six heading levels: <h1>, <h2>, <h3>, <h4>, <h5>, and <h6>.
- The <h1> head element should be used to label the top-most heading, and the rest of the elements should be used for subheads, much like a table of contents hierarchy.
- The browser will display the font for each of these levels differently, starting with a larger font for <h1> and progressively getting smaller as the heading number increases.
- Example:

```
<h1> This is a level 1 heading</h1>
<h2> This is a level 2 heading</h2>
<h3> This is a level 3 heading</h3>
<h4> This is a level 4 heading</h4>
<h5> This is a level 5 heading</h5>
<h6> This is a level 6 heading</h6>
```

The <h1> . . . <h6> Elements

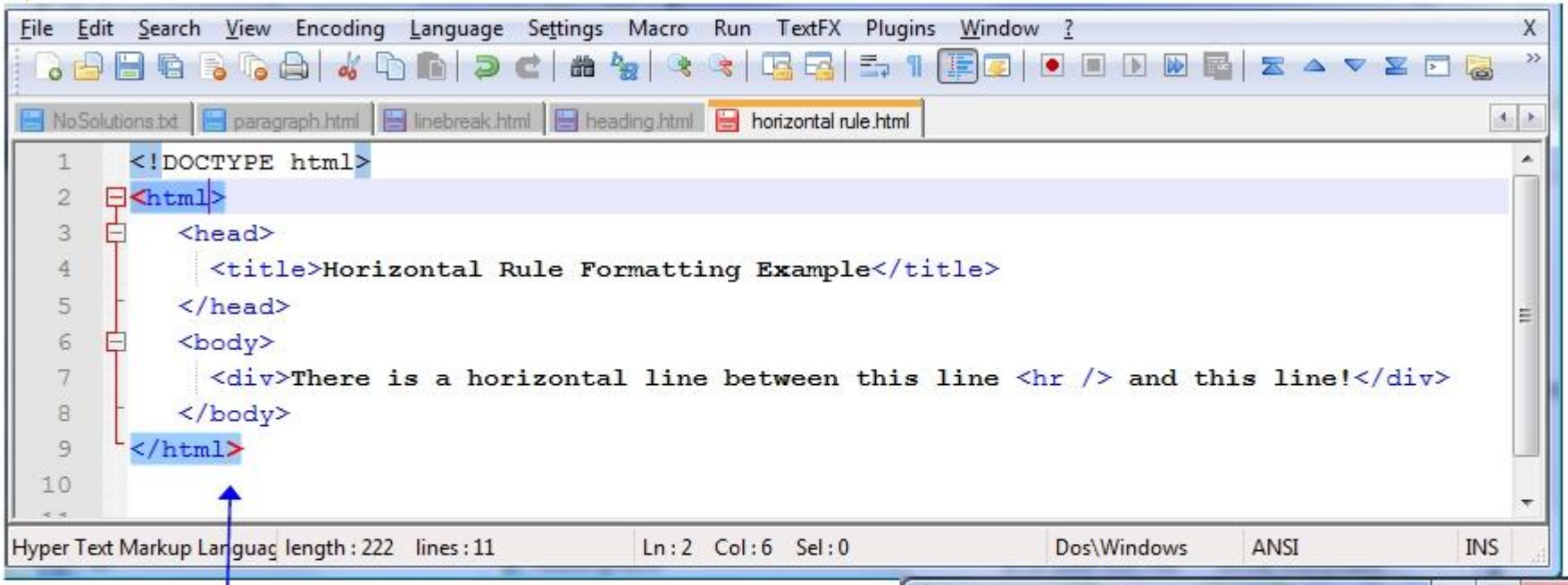


The <hr/> Element

- The <hr /> element is the horizontal rule element, used to create a visible horizontal line on the Web page to indicate a section break or change in content.
- XHTML Transitional and Frameset provide a set of attributes that can be used with this element to customize the rule. In XHTML Strict customization of the line is done via CSS.
- Example:

  There is a horizontal line between this line <hr /> and this line.

The <hr/> Element



HTML5

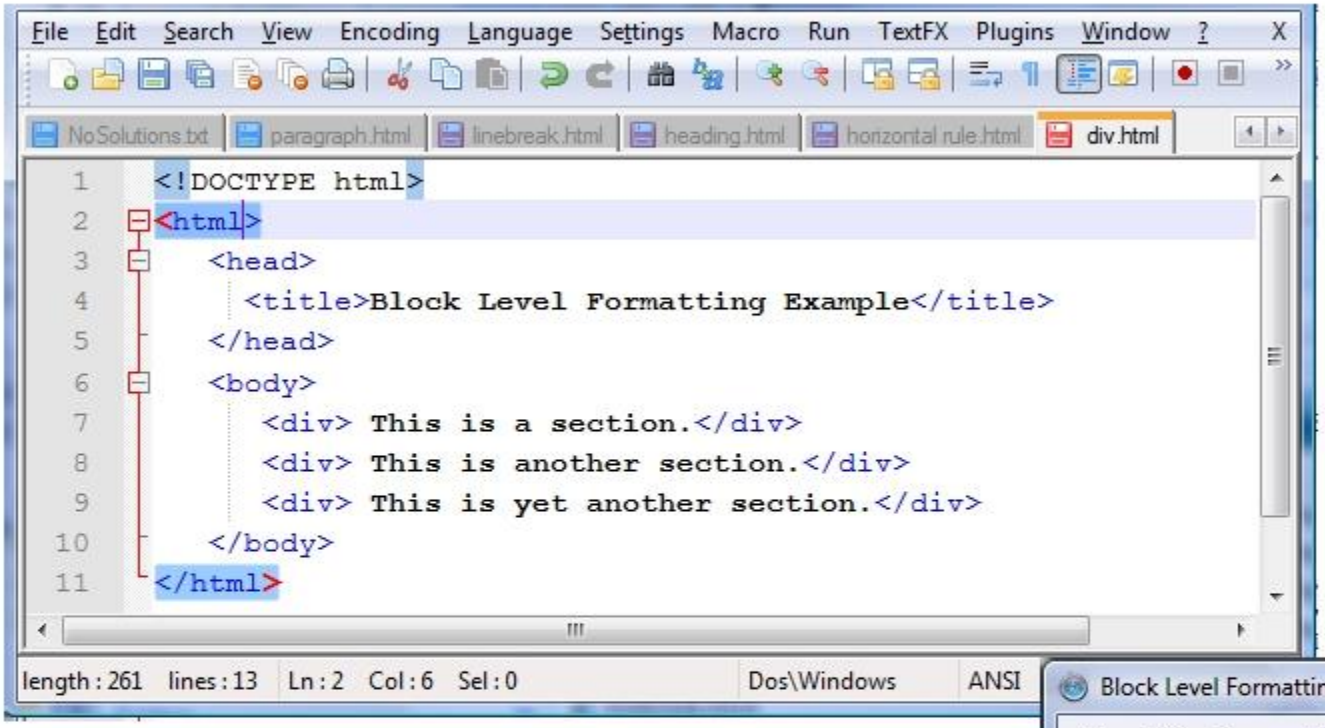
Rendering

The Element

- The element is used to divide sections of content. This element is used to label sections of the document, and can contain any number of other elements.
- This element can use the id and class core XHTML attributes to identify the various sections of the document to be used with parser programs.
- Example:

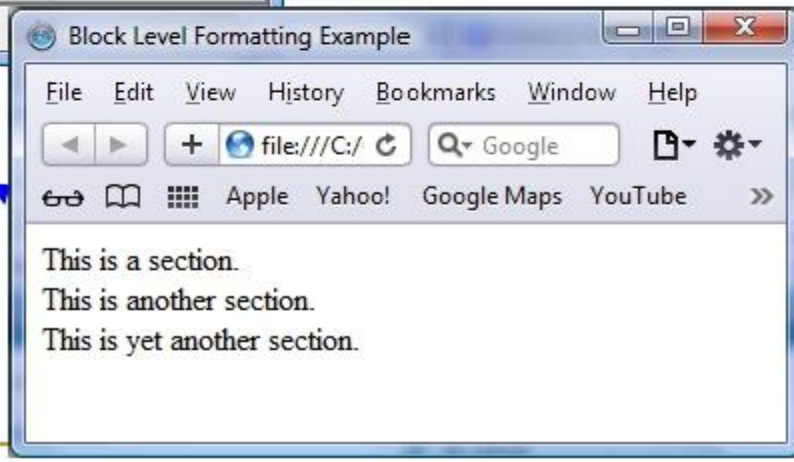
```
<div> This is a section.</div>

<div> This is another section.</div>
```

The Element



HTML5

Rendering



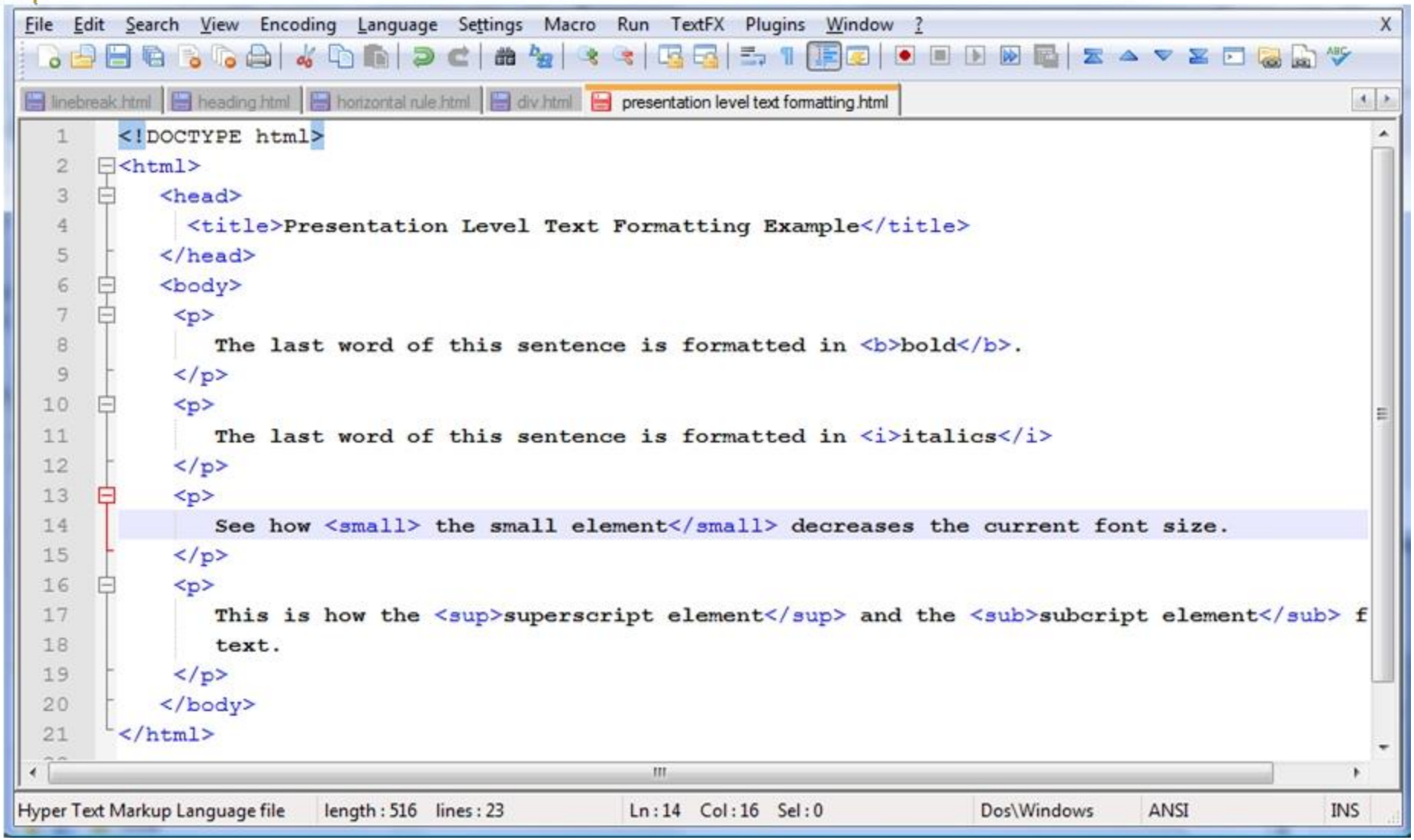
Text Formatting Elements

- Text formatting elements are referred to as character-level elements because, unlike the block-level elements, which describe blocks of content, these elements describe the text itself.
- Character-level elements describe the formatting of words or phrases as opposed to sections or paragraphs.
- There are two basic groups of text formatting elements: presentation styles, and logical styles.
  - Presentational styles describe how the text should be displayed, in bold type or italics, for example.
  - Logical styles describe the meaning of the style more than the actual format.

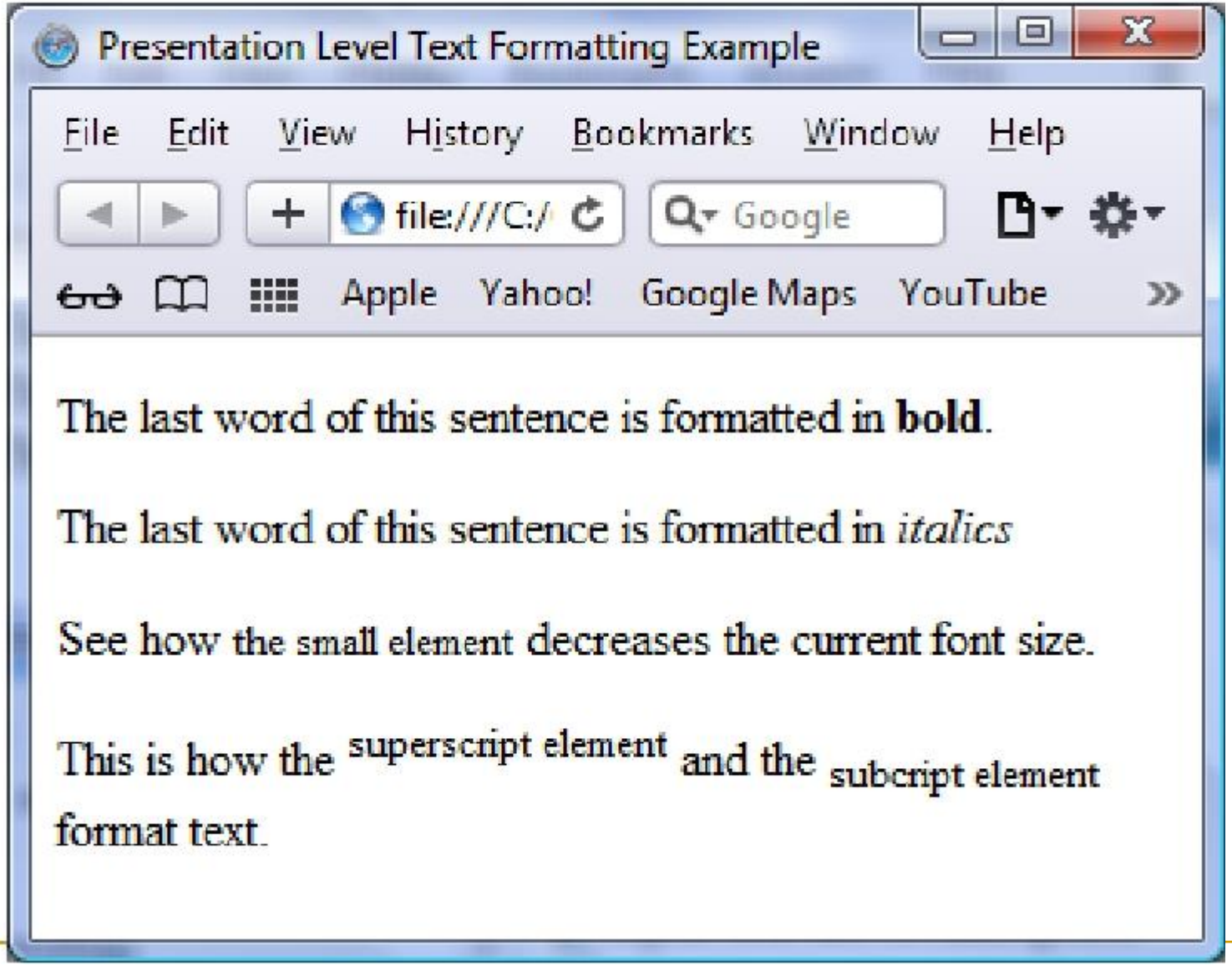
Presentational Text Formatting – Summary Chart

Element Name	Formatting Style
`<b>...</b>`	Bold font style
`<i>...</i>`	Italic font style
`<small>...</small>`	Decrease current font size
`<sub>...</sub>`	Subscripted text
`<sup>...</sup>`	Superscripted text

Presentational Text Formatting – Example



Presentational Text Formatting – Example



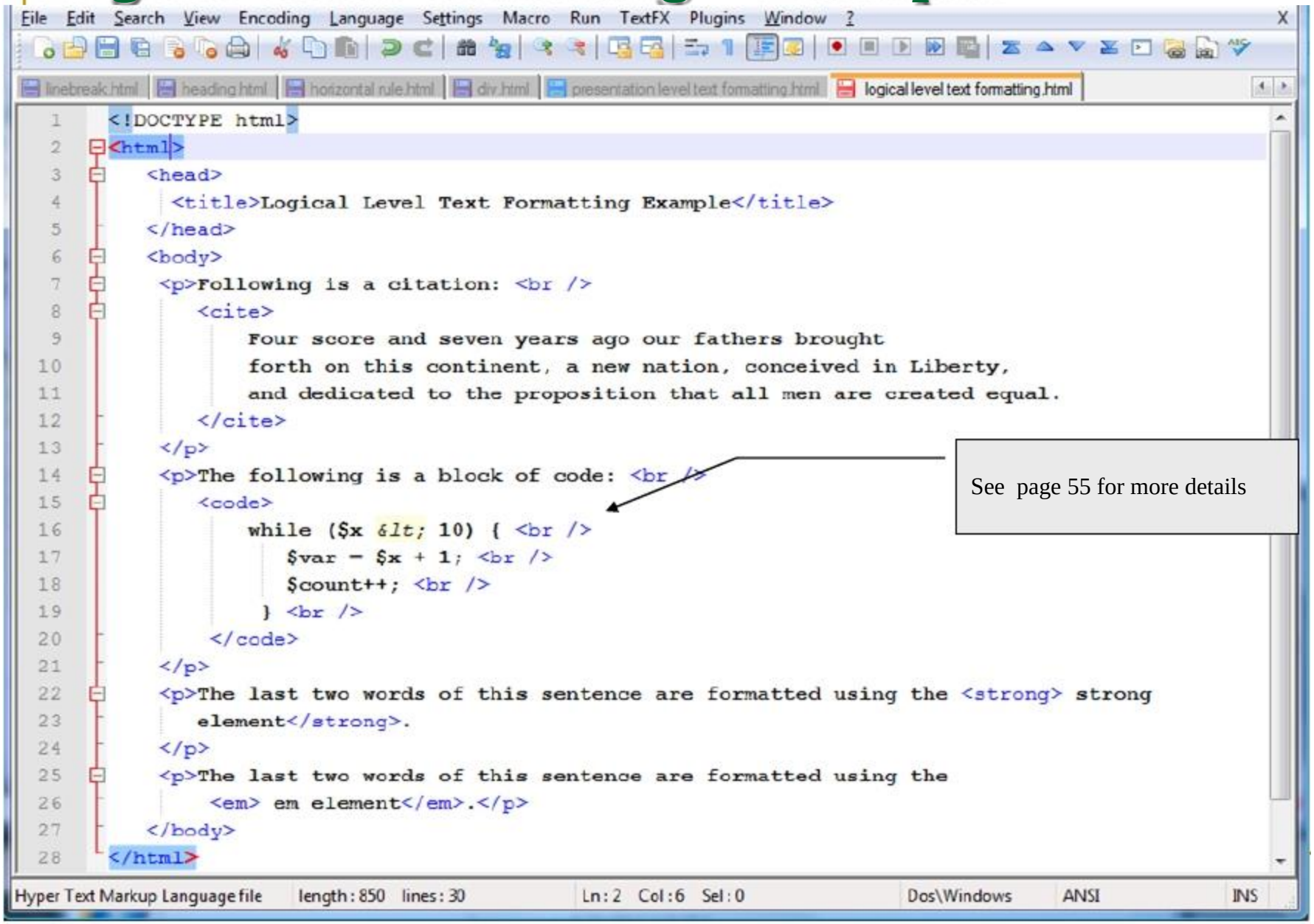
Logical Text Formatting Elements

- Logical text formatting describe the meaning of the style more than the actual format.
- Initially, browsers were left to determine the presentation of these tags as they saw fit, but over time certain standards were developed, and these are unlikely to change in the foreseeable future.
- For example, if you want a certain type, like bold, you should use the `<b>` elements, but the `<strong>` element will give you the same effect, because most browsers will interpret `<strong>` as `<b>`.
- The table on the next page lists the most commonly used logical elements.

Logical Text Formatting – Summary Chart

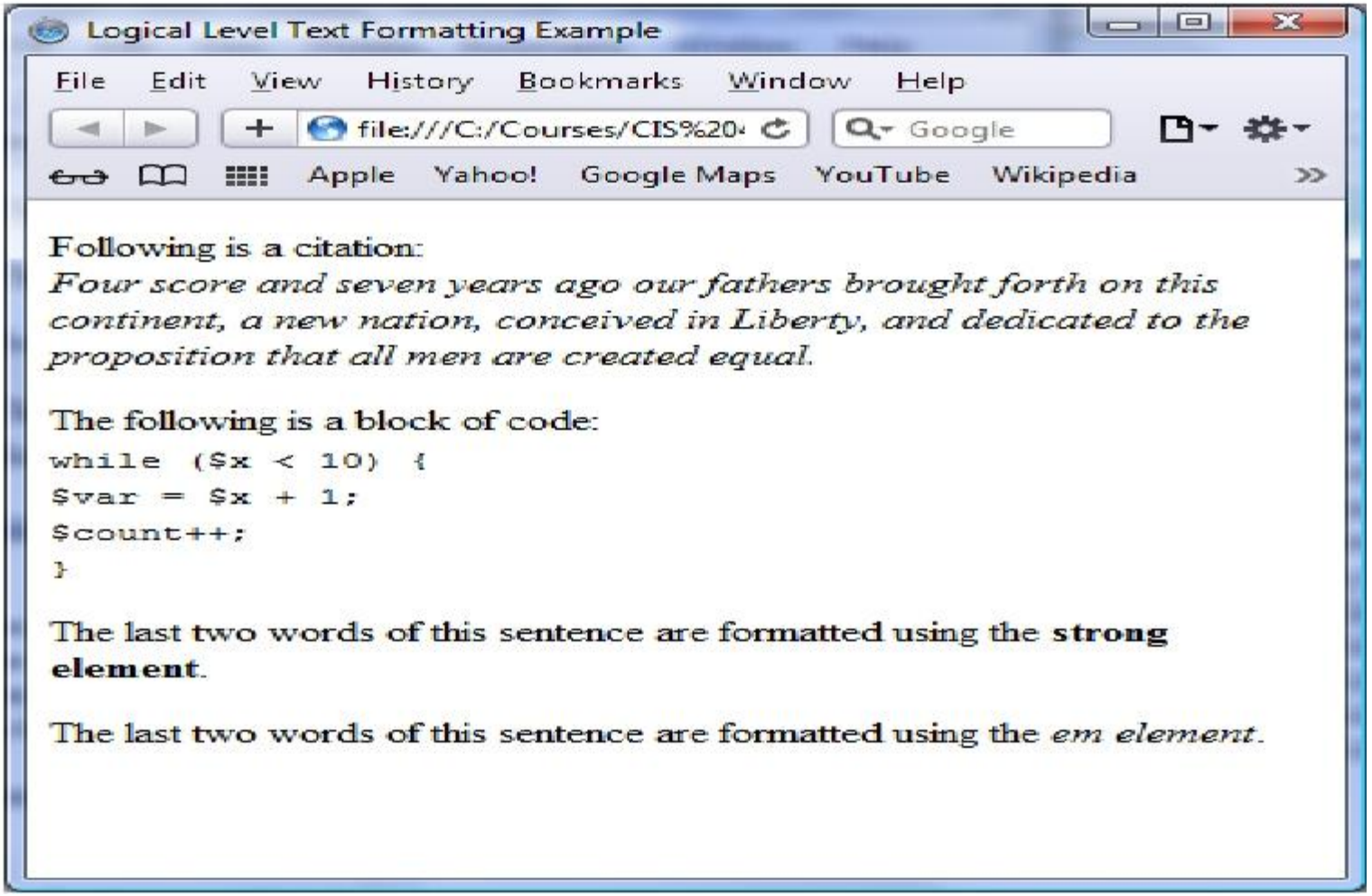
Element Name	Formatting Style
`<cite>...</cite>`	Defines a citation
`<code>...</code>`	Presents computer code examples
`<em>...</em>`	Emphasis. In most browsers, this is italics
`<strong>...</strong>`	Emphasis. In most browsers, this is bold
`<span>...</span>`	Provides a logical inline grouping with no predefined look.

Logical Text Formatting – Example



See page 55 for more details

Logical Text Formatting – Example



A Bit More On and

- The and elements are most often used in conjunction with style sheets (as we will see later) to avoid the default rendering of elements.
- is a block element whereas is an inline element. Neither of these elements have a default rendering, which in a sense makes them generic tags. Since they have no default rendering, they are very useful for arbitrary style duties.
- A element induces a hard return while the inline element does not.
- The following example will more clearly illustrate the differences between these two elements.

A Bit More On and

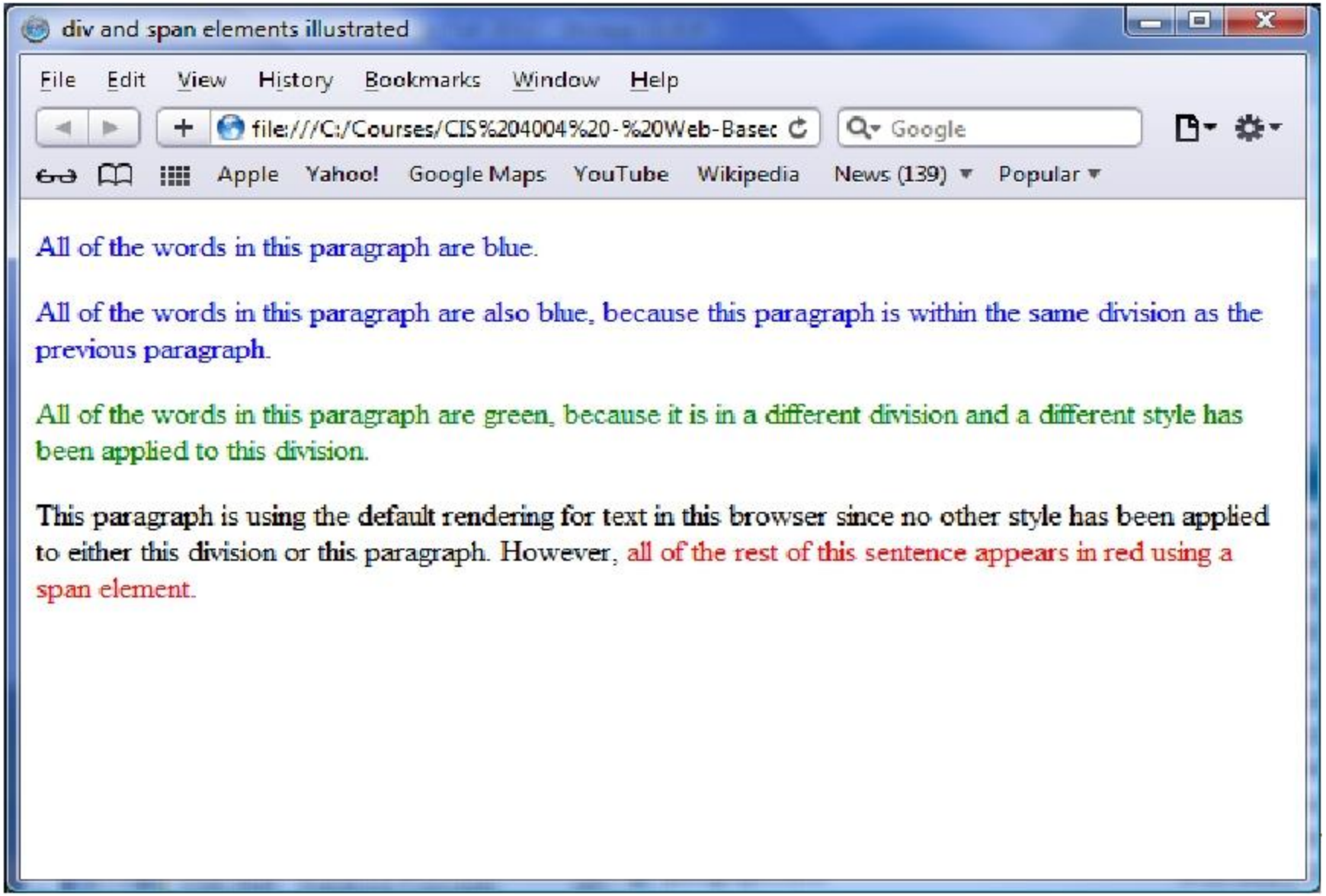
File Edit Search View Encoding Language Settings Macro Run TextFX Plugins Window ?

heading.html | horizontal rule.html | div.html | presentation level text formatting.html | logical level text formatting.html | div and span illustrated.html

```
1
2   <!DOCTYPE html>
3   <html>
4   <head>
5   <title>div and span elements illustrated  </title>
6   </head>
7   <body>
8     <div style="color:blue">
9         <p>All of the words in this paragraph are blue.  </p>
10        <p>All of the words in this paragraph are also blue, because this paragraph is
11           within the same division as the previous paragraph.  </p>
12    </div>
13    <div style="color:green">
14        <p>All of the words in this paragraph are green, because it is in a different division and
15           a different style has been applied to this division.</p>
16    </div>
17    <div>
18      <p> This paragraph is using the default rendering for text in this browser since no other
19          style has been applied to either this division or this paragraph.  However, <span style="color:red">
20          all of the rest of this sentence appears in red using a span element.</span>
21      </p>
22    </div>
23  </body>
24  </html>
25
```

This is an internal style (i.e., CSS)

A Bit More On and



Character Entities In XHTML

- The character of the less than symbol shown in the example on page 50 is written as <.
- Certain characters in XHTML have special meaning to parser, like less than < and greater than >. These characters identify the beginning and ending of a tag, so if you want to add these characters as literal values, you must use the character entity code for them.
- A character entity is written in the following syntax: &code;. It begins with an ampersand (&) character, then the code for the entity, then a semicolon (;).
- Hundreds of symbols can be referenced and included on Web pages using entities. Some of the more popular symbols can be referenced by their abbreviations, like less than (lt) and greater than (gt), but they can also be referenced using their decimal value in the ASCII Table.
- Some of the most common character entity codes are shown in the table on the next page.

Some Common Character Entity Codes

Symbol	Description	XHTML Code
>	Greater than	`>  or  &62;`
<	Less than	`<  or  &60;`
®	Trademark	`&trade;  or  &174;`
©	Copyright	`&copy;  or  &169;`
¢	Cent sign	`&cent;  or  &162;`
	Non-breaking space	`   or  &160`