CIS 4004: Web Based Information Technology Spring 2013

Inside HTML5 – Part 2

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

Inside HTML5 – Part 2

- In the first section of these notes, we looked at several new HTML5 elements that were not part of either XHTML or HTML4. Those new elements dealt with the main structure of the page.

- In this second section of notes, we'll look at some more new HTML5 elements as well as some elements that were part of XHTML and HTML4 that have been redefined in HTML5.

- The elements in this section of notes are known as *grouping* or *text-level* elements and deal with the content of the page.

Marking Up Figures And Captions

- The new HTML5 `figure` element allows you to wrap an image and give it a description.

- Previously, you would have had to use a `div` or something similar and then add the text to the page, and doing this meant that there was no semantic link between the image and the caption.

- The `figure` element does not always have to include an image; it can be used for sections of code, tabular data, audio, or video. Typically, though, figure would be used for an image.

- The markup on the following page illustrates an example.

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

float - example 1 - version 1.html | assignment2CSS.css | figure element example 1.html

```
 1    <!DOCTYPE html>
 2    <html lang="en">
 3    <head>
 4        <meta charset="utf-8" />
 5        <title> The figure element in HTML5 </title>
 6    </head>
 7    <body>
 8        <figure>
 9            <img alt="Tom Boonen in 2010" src="images/boonen.jpg" width="200" height="200" />
10            <figcaption>
11                Tom Boonen in 2010
12            </figcaption>
13        </figure>
14        <figure>
15            <img alt="Tom Boonen in 2012" src="images/boonen-2012.jpg" width="200" height="200" />
16            <figcaption>
17                Tom Boonen on September 20, 2012 preparing for World Championships in Holland
18            </figcaption>
19        </figure>
20    </body>
21    </html>
```

Hyper Text Markup Language file   length : 548   lines : 21   Ln : 7   Col : 7   Sel : 0   Dos\Windows   ANSI   INS

Marking Up Figures And Captions

- There has been some discussion on the W3C forums as to whether an `alt` text (the text alternative for browsers that does not support graphics) is still needed in a `figure` element.

- Outside of a `figure` element, an `img` element always needs an `alt` attribute. If the image is purely presentation and it does not need to be identified by assistive technology, the an empty `alt` attribute can be applied.

- With a `figure` element, if the caption is a suitable description, then no `alt` is needed. However, because of lack of browser and assistive technology support, this currently hinders accessibility.

- I would suggest erring on the side of caution here and always provide an `alt` attribute

Marking Up Figures And Captions

- You are not limited to a single image with a `figure` element; you can use the figure element to display multiple images.

- The markup on the next page illustrates such an example.

- The use of a `figcaption` element is not always required. If the image is for purely presentational reasons, then just use a normal `img` element. However, if it has additional information and is beneficial to the content, then it would likely require a description to go with it, so in these cases, use `figure` and `figcaption`.

- A `figure` element can have only one `figcaption` element.

- If you include a `figcaption`, it must be either the first or last element of the encompassing `figure` element.
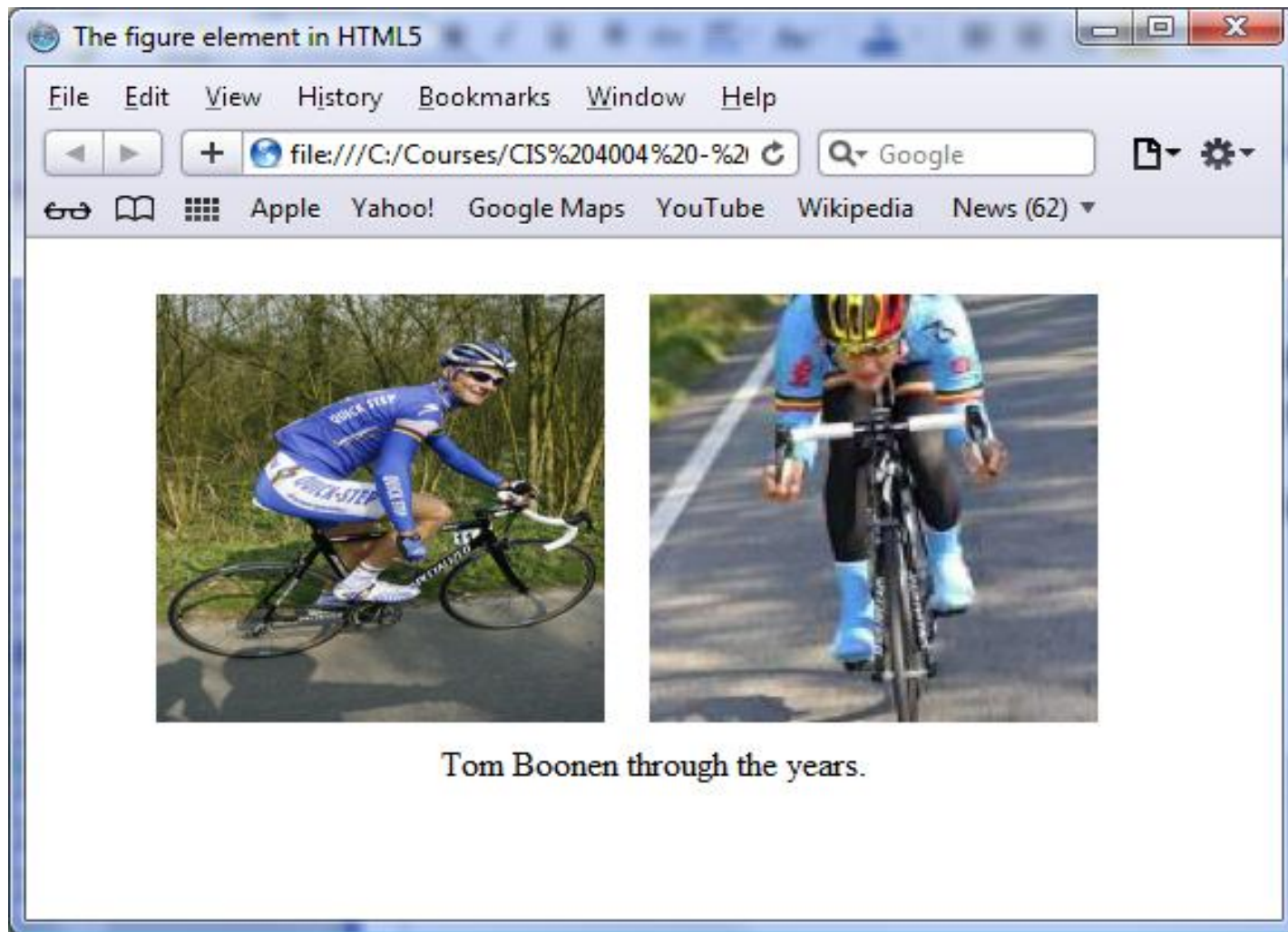
File   Edit   Search   View   Encoding   Language   Settings   Macro   Run   TextFX   Plugins   Window   ?

float - example 1 - version 1.html   assignment2CSS.css   figure element example 2.html

```
 1  <!DOCTYPE html>
 2  <html lang="en">
 3  <head>
 4      <meta charset="utf-8" />
 5      <title> The figure element in HTML5 </title>
 6      <style>
 7      <!--
 8          img { position:relative; float:left; margin:10px;}
 9          figcaption {text-align:center; }
10      -->
11      </style>
12  </head>
13  <body>
14      <figure>
15          <img alt="Tom Boonen in 2010" src="images/boonen.jpg" width="200" height="200" />
16          <img alt="Tom Boonen in 2012" src="images/boonen-2012.jpg" width="200" height="200" />
17          <figcaption>
18              Tom Boonen through the years.
19          </figcaption>
20      </figure>
21  </body>
22  </html>
```

Hyper Text Markup Language file   length : 541   lines : 22   Ln : 21   Col : 1   Sel : 0   Dos\Windows   ANSI   INS

Marking Up Figures And Captions

The `time` Element

- The `time` element allows you to code dates and times that are readable by machines but are displayed to users in a readable fashion.

- You can timestamp things such as publishing dates or events that can populate other technologies, such as a calendar.

- The `time` element is intended to show precise dates, such as "25th September 2012," not vague dates such as "some point in 2012."

- The `datetime` attribute (see next page) must be in the format of the Gregorian calendar: YYYY-MM-DD, with time coded as T00:00.

The `time` Element

- The `time` element has two optional attributes:

  - `datetime`: The end user will see the content inside the `time` element, but a machine will be able to read the `datetime` value: `dateime=`"2012-09-25T13:00Z". The time part of this value (T13:00) is optional. You can also add a time zone offset: T13:00+04:00. The "Z" represents Universal Coordinated Time (UTC), which is the same as adding a time zone offset of +00:00.

  - `pubdate`: This is a boolean attribute. It indicates the date, and possible time, of the publication of its nearest parent `article` element. If there is no parent `article` element, then the `pubdate` refers to the whole document. Each `article` must have only one `time` element with a `pubdate`.

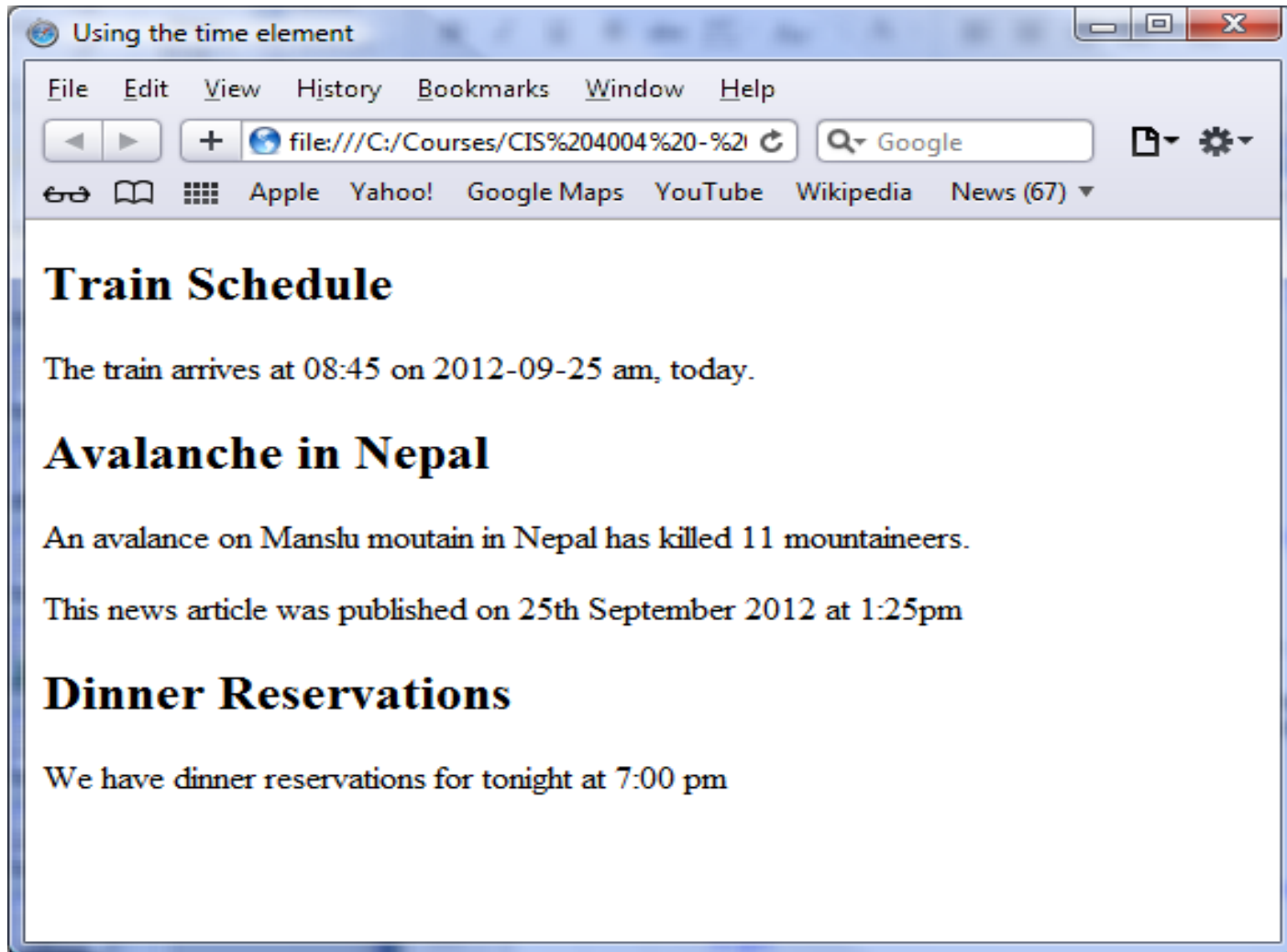- See the markup on the next page for some examples.

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

assignment2CSS.css | notes.html | notes.html | time element example.html

```html
 1  <!DOCTYPE html>
 2  <html lang="en">
 3  <head>
 4      <meta charset="utf-8" />
 5      <title> Using the time element </title>
 6  </head>
 7  <body>
 8      <article>
 9          <h2>Train Schedule</h2>
10              <p>The train arrives at <time>08:45</time> on <time>2012-09-25</time></p>
11      </article>
12      <article>
13          <h2>Avalanche in Nepal</h2>
14              <p> An avalance on Manslu moutain in Nepal has killed 11 mountaineers.</p>
15          <footer>
16              <p>This news article was published on <time datetime="2012-09-24T13:25">
17                  25th September 2012 at 1:25pm</time>
18              </p>
19          </footer>
20      </article>
21      <article>
22          <h2>Dinner Reservations</h2>
23              <p>We have dinner reservations for <time datetime="2012-09-25T23:00-04:00">
24                  tonight at 7:00 pm</time>
25              </p>
26      </article>
27  </body>
28  </html>
```

Hyper Text Markup Language file   length : 749   lines : 28   Ln : 1   Col : 16   Sel : 0   Dos\Windows   ANSI   INS

The `time` Element

The `time` Element

- The `time` element's time is based on a 24-hour clock with an optional time-zone offset from UTC.

- Times may be stated as: YYYY-MM-DDThh:mm:ss or YYY-MM-DDThh:mm.sss. The second form allowing for millisecond precision. Note the period before the milliseconds not a colon as with seconds.

- The time-zone offset can be either positive or negative from UTC. See the example on the previous page which uses current DST in eastern United States as being UTC-04:00.

- If you use `time` with `pubdate` to indicate an article's publication date, its common, but not mandatory to place it in either a `header` or `footer` element of the `article` element.

- You may not nest a `time` element inside another `time` element.

Marking Important And Emphasized Text

- The HTML5 elements `strong`, `em`, `b`, and `i` all appeared in XHTML and earlier versions of HTML, but the `b` and `i` elements have been redefined in HTML5.

- Semantically, the `strong` element denotes important text, while `em` conveys emphasis. They can be used individually or together as your content requires.

- You may nest `strong` text inside a phrase that is also marked with `strong`. If you do, the importance (semantically) of `strong` text increases each time it's a child of another `strong`. The same is true for `em`. See the example on the next page.

- DO NOT use the `b` and `i` elements as replacements for `strong` and `em`. Although they may look similar in a browser, their meanings are very different.

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

assignment2CSS.css | notes.html | notes.html | strong and emphasis example.html

```
 1   <!DOCTYPE html>
 2   <html lang="en">
 3   <head>
 4       <meta charset="utf-8" />
 5       <title> Using the strong and em elements </title>
 6   </head>
 7   <body>
 8       <article>
 9           <p><strong>Warning!! Do not approach a zombie <em>under any circumstances</em>.
10               They may <em>look</em> friendly, but that's just because they want
11               to <strong>eat</strong> your arm!
12           </strong>
13           </p>
14       </article>
15   </body>
16   </html>
```

Using the strong and em elements

File   Edit   View   History   Bookmarks   Window   Help

◄   ►   +   file:///C:/Courses/CIS%204004%20-%2   ⟳   Q▾ Google

🔖   ▦   Apple   Yahoo!   Google Maps   YouTube   Wikipedia   News (67) ▾

Warning!! Do not approach a zombie *under any circumstances*. **They may** *look* **friendly, but that's just because they want to** **eat** **your arm!**

Hyper Text Markup Language file | length : 401   lines : 16   Ln : 16   Col : 8   Sel : 0   Dos\Windows   ANSI   INS

Marking Important And Emphasized Text

- The `b`, and `i` elements have been redefined in HTML5 since `strong` and `em` are not always semantically correct.

- Some typographic conventions in traditional publishing fall through the cracks of available HTML semantics. Among these are italicizing certain scientific names, named vehicles (e.g. "We rode the *Orient Express*."), and foreign (to English) language phrases (e.g., "they exhibited a *joie de vivre* that was infectious."). These terms aren't italicized for emphasis, just stylized per convention.

- Rather than create several new semantic elements to address cases like these, HTML5 took the practical approach and redefined the `b` and `i` elements to handle this "through the cracks" cases.

Marking Important And Emphasized Text

- The notion is that although `b`, and `i` don't carry explicit semantics, the reader will recognize that a difference is implied because they differ from the surrounding text.

- You can still change their appearance from bold and italics with CSS.

- HTML5 emphasizes that you use `b` and `i` only as a last resort when another element (such as `strong`, `em`, `cite`, and others) won't do.

- The "official" use of the `b` and `i` elements in HTML5 are shown on the next page.

Marking Important And Emphasized Text

The **b** element represents a span of text to which attention is being drawn for utilitarian purposes without conveying any extra importance and with no implication of an alternate voice or mood, such as key words in a document abstract, product names in a review, actionable words in interactive text-driven software, or an article lede.

The **i** element represents a span of text in an alternative voice or mood, or otherwise offset from the normal prose in a manner indicating a different quality of text, such as a taxonomic designation, a technical term, an idiomatic phrase from another language, a thought, or a ship name in western texts.

Highlighting Text With The `mark` Element

- Everyone has probably used a highlighter pen at some point. You used it to mark key words or phrases that were relevant, perhaps when you were studying for an exam.

- HTML5 replicates this with the new `mark` element. Think of `mark` as a semantic version of a highlighter pen. In other words, what's important is that you're noting certain words; how they appear is irrelevant.

- Style the highlighted text as you please with CSS (or not at all), but use `mark` only when its pertinent to do so.

- Use the `mark` element when you want to draw a reader's attention to a particular text segment.

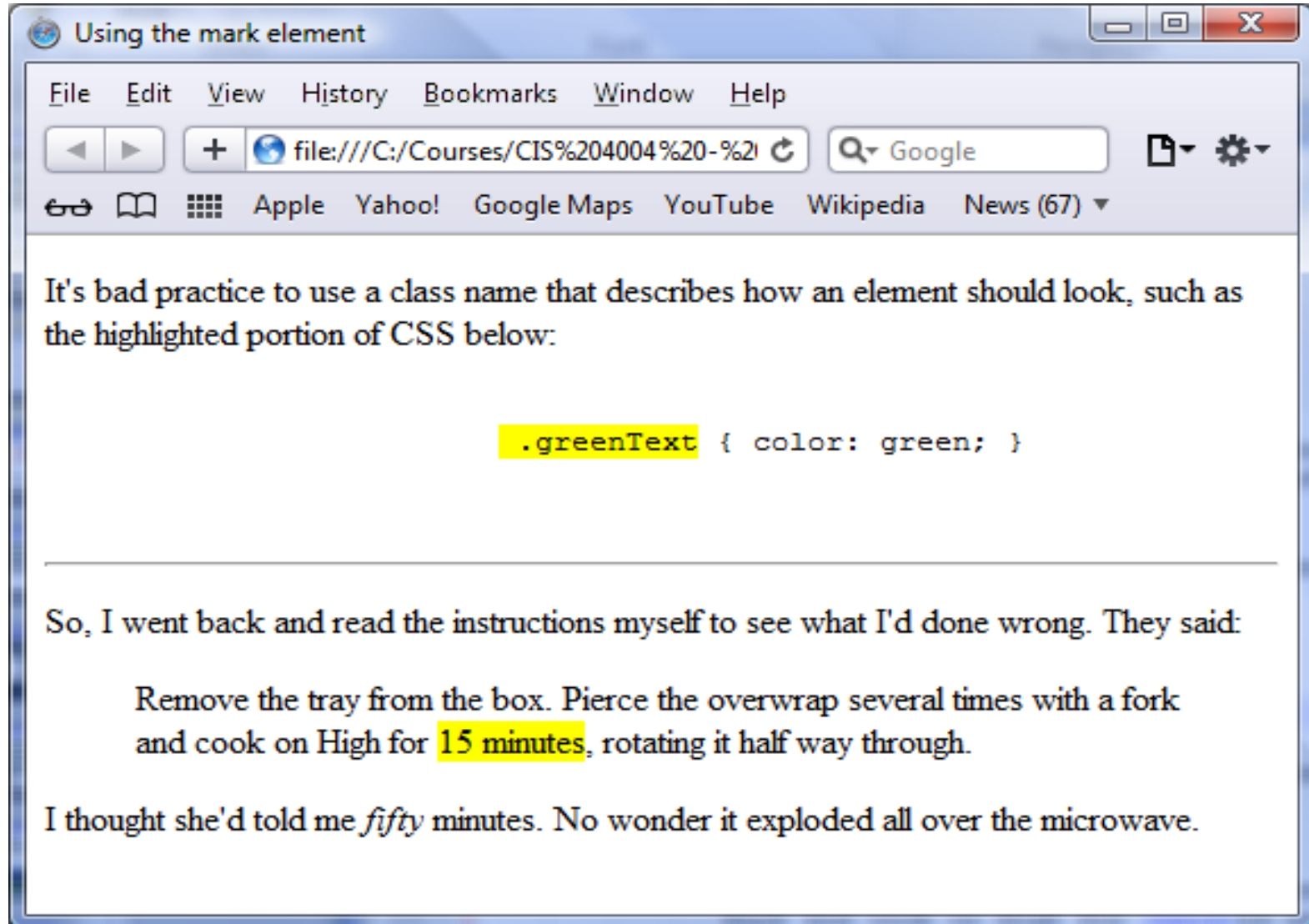- The example on the next page illustrates the `mark` element.

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

assignment2CSS.css   notes.html   notes.html   mark example.html

```
 3    <head>
 4        <meta charset="utf-8" />
 5        <title> Using the mark element </title>
 6    </head>
 7    <body>
 8        <article>
 9            <p>It's bad practice to use a class name that describes how an element should
10                look, such as the highlighted portion of CSS below:
11            <pre>
12                <code>
13                 <mark> .greenText</mark> { color: green; }
14                </code>
15            </pre>
16            </p>
17        </article>
18        <hr />
19        <article>
20            <p>So, I went back and read the instructions myself to see what I'd done wrong.
21                They said:
22            </p>
23            <blockquote>
24                <p>Remove the tray from the box.  Pierce the overwrap several times with a
25                    fork and cook on High for <mark>15 minutes</mark>, rotating it half way
26                    through.
27                </p>
28            </blockquote>
29            <p>I thought she'd told me <em>fifty</em> minutes.  No wonder it exploded all
30                over the microwave.
31            </p>
```

Hyper Text Markup Language file   length : 893   lines : 34   Ln : 18   Col : 11   Sel : 0   Dos\Windows   ANSI   INS

Highlighting Text With The `mark` Element

Using the mark element

It's bad practice to use a class name that describes how an element should look, such as the highlighted portion of CSS below:

==.greenText== { color: green; }

So, I went back and read the instructions myself to see what I'd done wrong. They said:

Remove the tray from the box. Pierce the overwrap several times with a fork and cook on High for ==15 minutes==, rotating it half way through.

I thought she'd told me *fifty* minutes. No wonder it exploded all over the microwave.

Highlighting Text With The `mark` Element

- The second `article` element in the markup on the previous page illustrates a less common usage of the mark element. In this case, the phrase "15 minutes" was not highlighted in the instructions on the original packaging. Instead, the author uses the `mark` element to call out the phrase as part of the story.

- Since the `mark` element is a new HTML5 element, older browsers don't render a background color by default. You can instruct them to do so via CSS by adding: `mark {background-color: yellow;}` to your stylesheet.

- Be sure not to use `mark` simply to give text a background color or other visual treatment. It all you're looking for is a means to style text and there is no proper semantic HTML5 element with which to wrap it, use the `span` element and style it with CSS.

Noting Edits and Inaccurate Text

- Sometimes you might want to indicate content edits that have occurred since the previous version of your page, or mark up text that is no longer accurate or relevant.

- There are two new elements in HTML5 for denoting edited content: `ins` and `del`, and one element denoting content that is no longer accurate or relevant: `s`.

- The `ins` element represents content that has been added to a page, and the `del` element represents content that has been removed from a page. The `s` element represents content that is no longer accurate or relevant. Think `s` for "strike".

- Before we explore the semantic differences between `del` and `s`, let's look at a couple of examples.

Noting Edits and Inaccurate Text

```
<!DOCTYPE html>
<html lang="en">
<head>
        <meta charset="utf-8" />
        <title> Using del, ins, and s elements - Before Changes </title>
</head>
<body>
  <article>
    <h2> Charitable Gifts Wishlist</h2>
    <p>  Please consider one or more of the following items to our community center</p>
    <ul>
        <li> 2 desks </li>
        <li> whiteboard with markers </li>
        <li> 3 laptop computers</li>
        <li> bicycle</li>
        <li> coffee maker</li>
    </ul>
  </article>
</body>
</html>
```
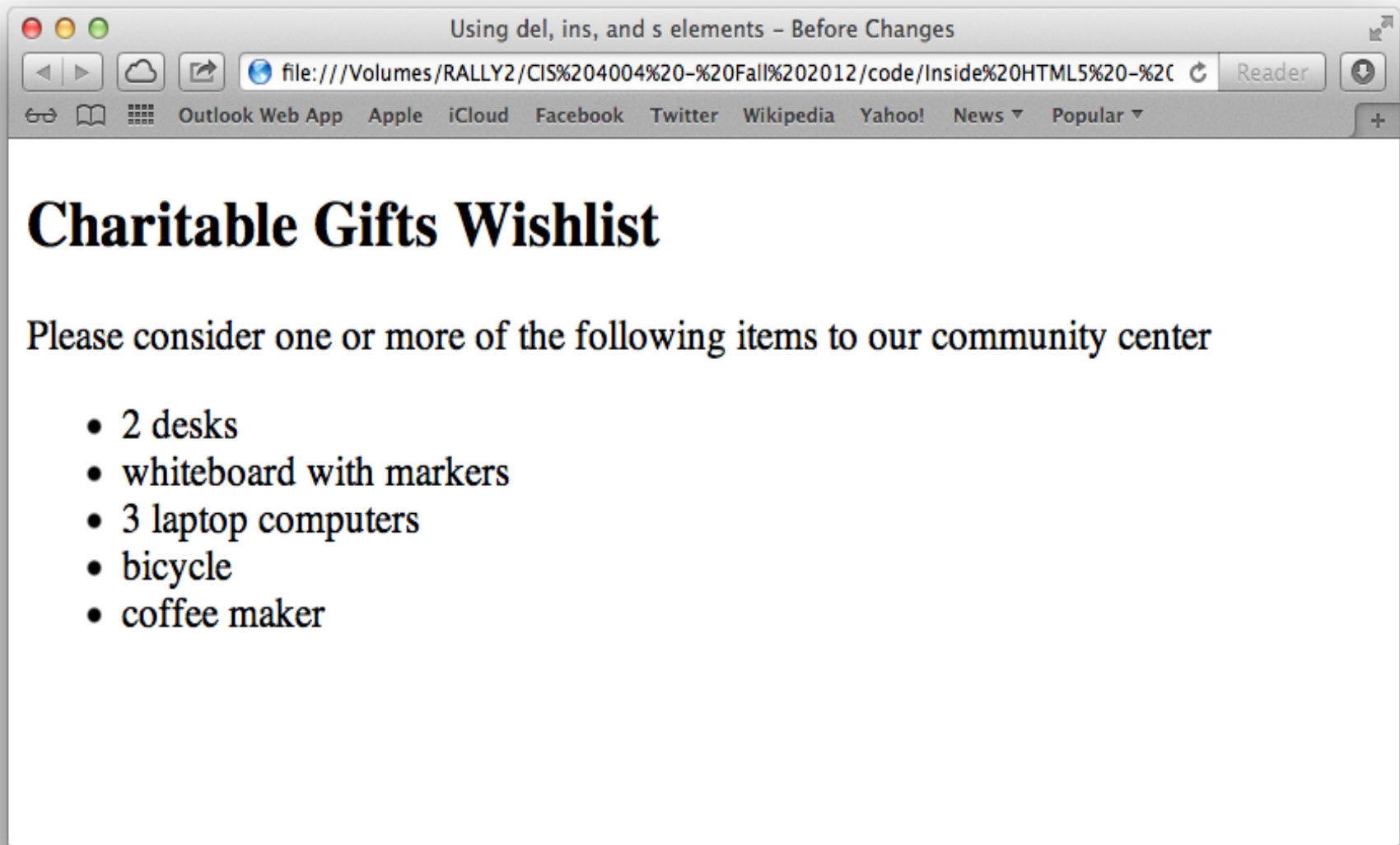
Noting Edits and Inaccurate Text

Noting Edits and Inaccurate Text

```
<!DOCTYPE html>
<html lang="en">
<head>
        <meta charset="utf-8" />
        <title> Using del, ins, and s elements - After Changes to markup version 1 </
title>
</head>
<body>
  <article>
    <h2> Charitable Gifts Wishlist</h2>
    <p>  Please consider one or more of the following items to our community center</p>
    <ul>
        <li> <del>2 desks</del> </li>
        <li> whiteboard with markers </li>
        <li> <del>3 laptop computers<del></li>
        <li> bicycle</li>
        <li> coffee maker</li>
        <li><ins>coffee mugs</ins></li>
    </ul>
  </article>
</body>
</html>
```

Noting Edits and Inaccurate Text

Charitable Gifts Wishlist

Please consider one or more of the following items to our community center

- ~~2 desks~~
- whiteboard with markers
- ~~3 laptop computers~~
- bicycle
- coffee maker
- <u>coffee mugs</u>
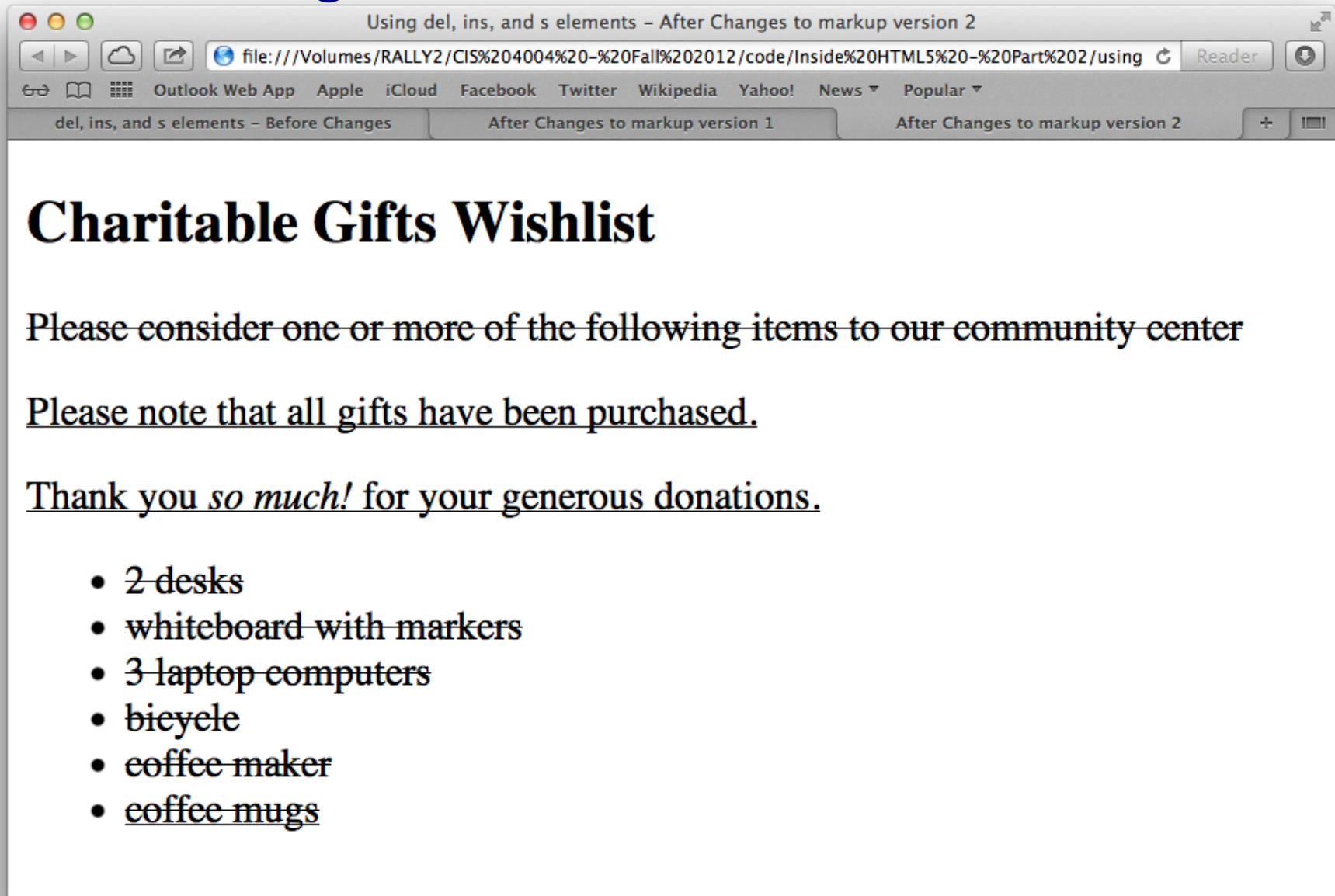
Deleted content

Inserted content

Noting Edits and Inaccurate Text

```html
<!DOCTYPE html>
<html lang="en">
<head>
        <meta charset="utf-8" />
        <title> Using del, ins, and s elements - After Changes to markup version 2  </title>
</head>
<body>
  <article>
    <h2> Charitable Gifts Wishlist</h2>
    <del>
       <p>Please consider one or more of the following items to our community center</p>
    </del>
    <ins>
       <p>Please note that all gifts have been purchased.</p>
       <p>Thank you <em> so much!</em> for your generous donations.</p>
    </ins>
    <del>
      <ul>
        <li> <del>2 desks</del> </li>
        <li> whiteboard with markers </li>
        <li> <del>3 laptop computers</del></li>
        <li> bicycle</li>
        <li> coffee maker</li>
        <li> <ins>coffee mugs</ins></li>
      </ul>
    </del>
  </article>
</body>
</html>
```

Noting Edits and Inaccurate Text

file:///Volumes/RALLY2/CIS%204004%20-%20Fall%202012/code/Inside%20HTML5%20-%20Part%202/using

Outlook Web App   Apple   iCloud   Facebook   Twitter   Wikipedia   Yahoo!   News ▼   Popular ▼

del, ins, and s elements – Before Changes | After Changes to markup version 1 | After Changes to markup version 2

Charitable Gifts Wishlist

~~Please consider one or more of the following items to our community center~~

Please note that all gifts have been purchased.

Thank you *so much!* for your generous donations.

- ~~2 desks~~
- ~~whiteboard with markers~~
- ~~3 laptop computers~~
- ~~bicycle~~
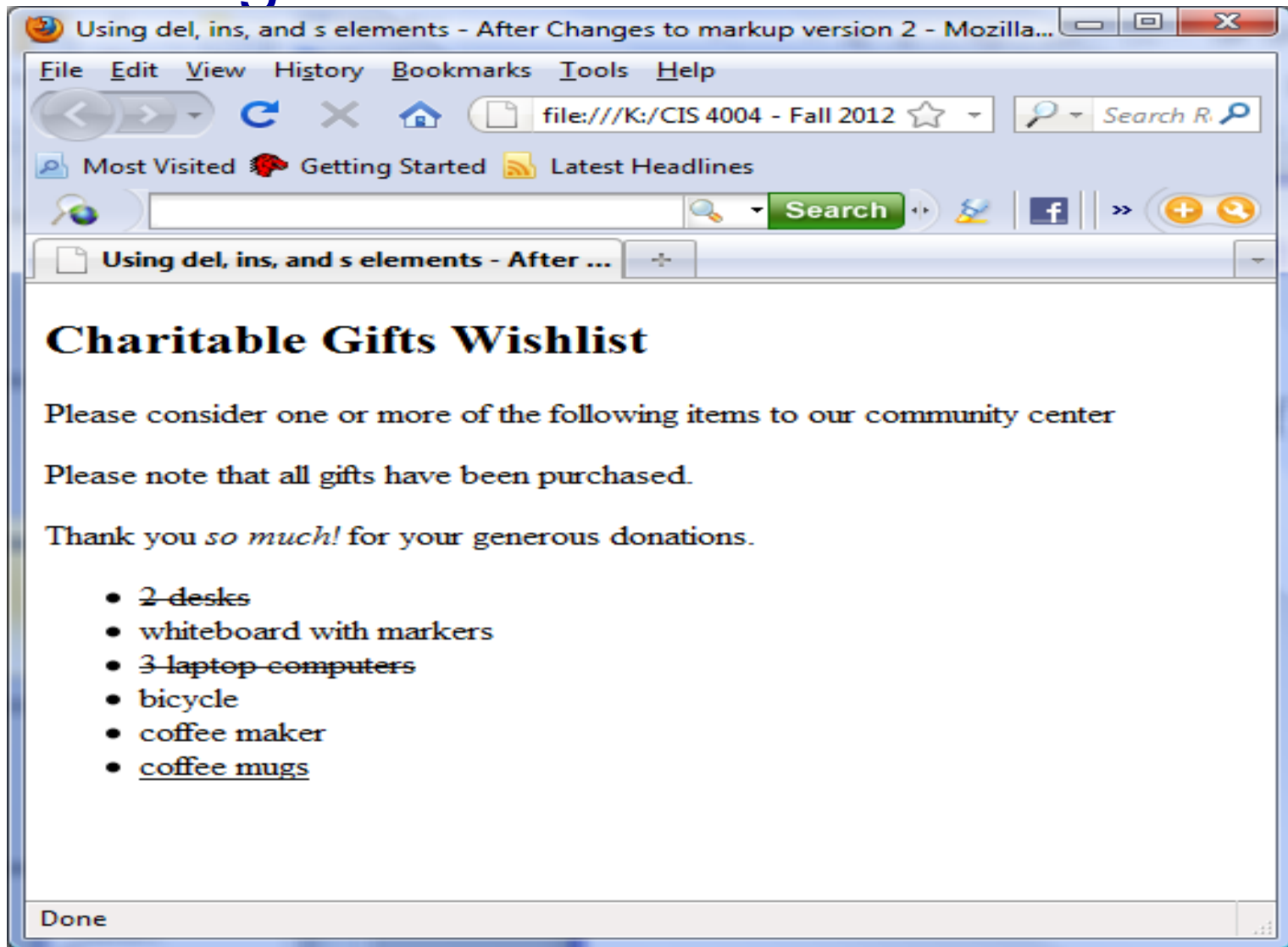- ~~coffee maker~~
- coffee mugs

Noting Edits and Inaccurate Text

- Browsers typically display a line through deleted text and underline inserted text. You can of course, override this default effect via CSS.

- As illustrated in version 2 of the previous example, del and ins elements are rare in HTML5 in that they can surround both phrasing ("inline" in XHTML parlance) content and blocks of content. However, there is a difference in how the various browsers will interpret this. For example, Chrome and Safari will render this as shown on the previous page – correctly interpreted for both phrases and blocks. Firefox on the other hand will render the lines for `del` and `ins` text phrases only within encompassing elements. See next page for illustration of Firefox's version of the previous example.

Noting Edits and Inaccurate Text

File   Edit   View   History   Bookmarks   Tools   Help

file:///K:/CIS 4004 - Fall 2012

Search R.

Most Visited   Getting Started   Latest Headlines

Search

Using del, ins, and s elements - After ...

Charitable Gifts Wishlist

Please consider one or more of the following items to our community center

Please note that all gifts have been purchased.

Thank you *so much!* for your generous donations.

- ~~2 desks~~
- whiteboard with markers
- ~~3 laptop computers~~
- bicycle
- coffee maker
- <u>coffee mugs</u>

Done

Noting Edits and Inaccurate Text

- For marking content that is no longer relevant or accurate, the `s` element should be used in this context.

- The HTML5 documentation states that "The `s` element is not appropriate when indicating document edits; to mark a span of text as having been removed from a document, use the `del` element."

- The example on the following page might help to clarify this distinction between deleted content and no longer relevant content.

Noting Edits and Inaccurate Text

```
<!DOCTYPE html>
<html lang="en">
<head>

        <meta charset="utf-8" />
        <title> Using the s element </title>
</head>
<body>
   <article>
     <h2>Today's Showtimes</h2>
     <p>Tickets are available for the following times today:</p>
     <ol>
         <li><ins>2:00 pm - (This show just added at <time datetime="2012-09-25T10:45">
                    at 10:45 am)| </time> </li>
         <li><s>5:00 pm</s> SOLD OUT </li>
         <li><s>8:30 pm</s> SOLD OUT </li>
     </ol>
   </article>
</body>
</html>
```
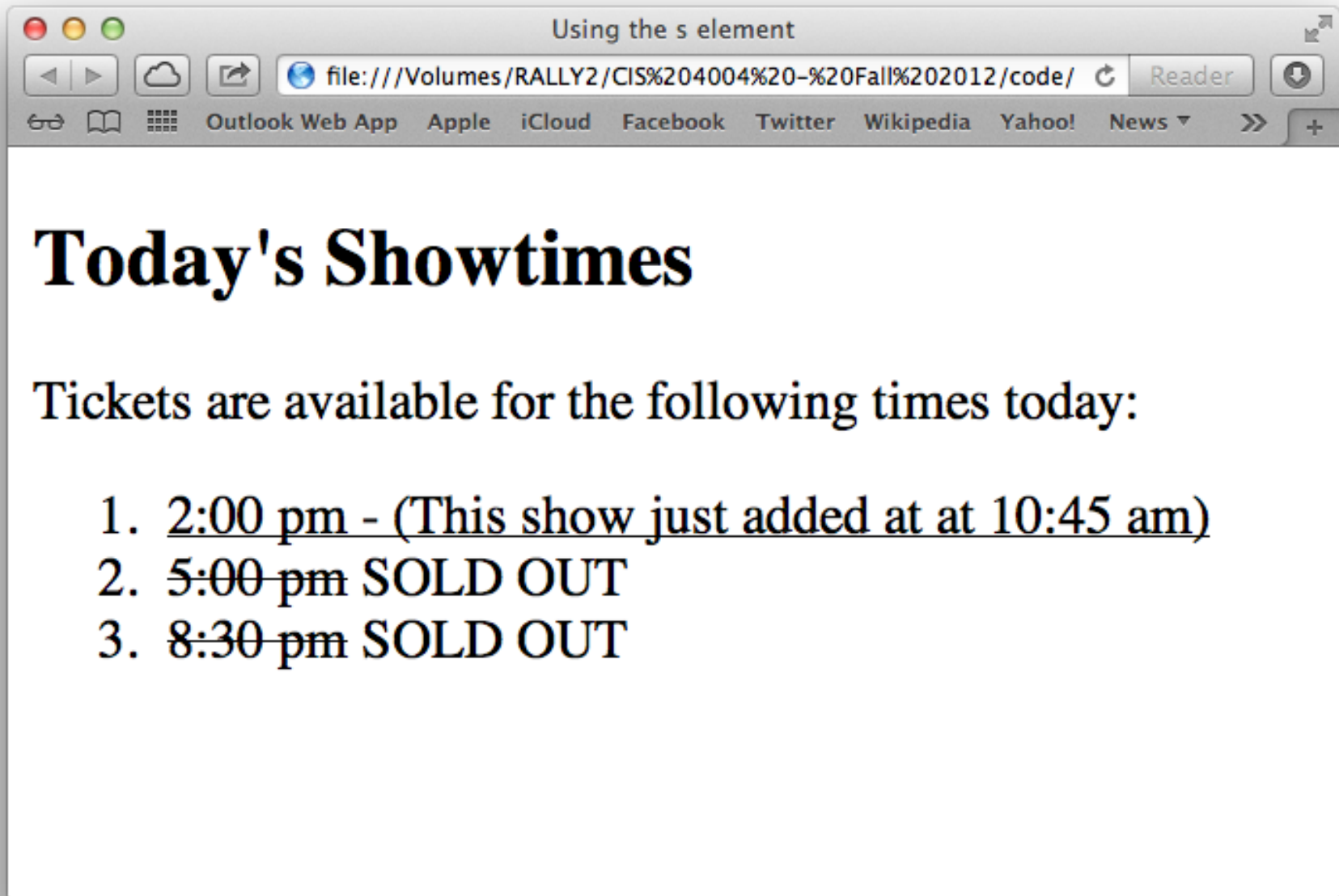
Noting Edits and Inaccurate Text

Noting Edits and Inaccurate Text

- The `del` and `ins` elements both support two attributes: `cite` and `datetime`.

- The `cite` attribute (not the same as the `cite` element) us for providing a URL to a source that explains why and edit was made.

- Use the `datetime` attribute to indicate the time of the edit.

- Browsers do not display the values you assign to either of these attributes, so their use will not be widespread, but feel free to include them as they add context to your content. The values could be extracted via JavaScript or a program that parses through your page.

Noting Edits and Inaccurate Text

- Use the `del` and `ins` elements anytime that you want to inform your visitors of your content's evolution.

- For instance, you'll often see them uses in a Web development or design tutorial to indicate information learned since it was initially posted, while maintaining the copy as it originally stood for completeness. The same will be true for blogs, news sites, and so on.

- Text marked with a del element is generally struck out. You might ask why not just delete that content. The answer is that it depends on the context of your content. Striking it out makes it easy for sighted users to know what has changed. Also screen readers could announce the content as having been removed, unfortunately, their support for doing so has been lacking.

The `meter` Element

- HTML5 has a lot of new and neat features. Some of which we'll see later in the course. For now here is one that you might find quite useful.

- The `meter` element is a new element that can be used to indicate a fractional value or a measurement within a known range. Basically, its like a gauge that you could use for things like voting ranges, project completion reports, tickets sold, etc..

- HTML5 suggests that browsers render a `meter` "not unlike a thermometer laying on its side – a horizontal bar with the measured value colored differently than the maximum value unless they are the same."

- The example on the next page illustrates the `meter` element.

The `meter` Element

```
<!DOCTYPE html>
<html lang="en">
<head>

        <meta charset="utf-8" />
        <title> The meter element </title>
</head>
<body>
   <p> Project completion status: <meter value="0.80"> 80% completed</meter></p>
   <hr />
   <h2>Results of our color survey</h2>
   <pre>
   <ul>
     <li>blue  <meter value="0.60">60% rated as #1</meter></li>
     <li>red   <meter value="0.30">30% rated as #1</meter></li>
     <li>green <meter value="0.10">10% rated as #1</meter></li>
   </ul>
   </pre>
</body>
</html>
```
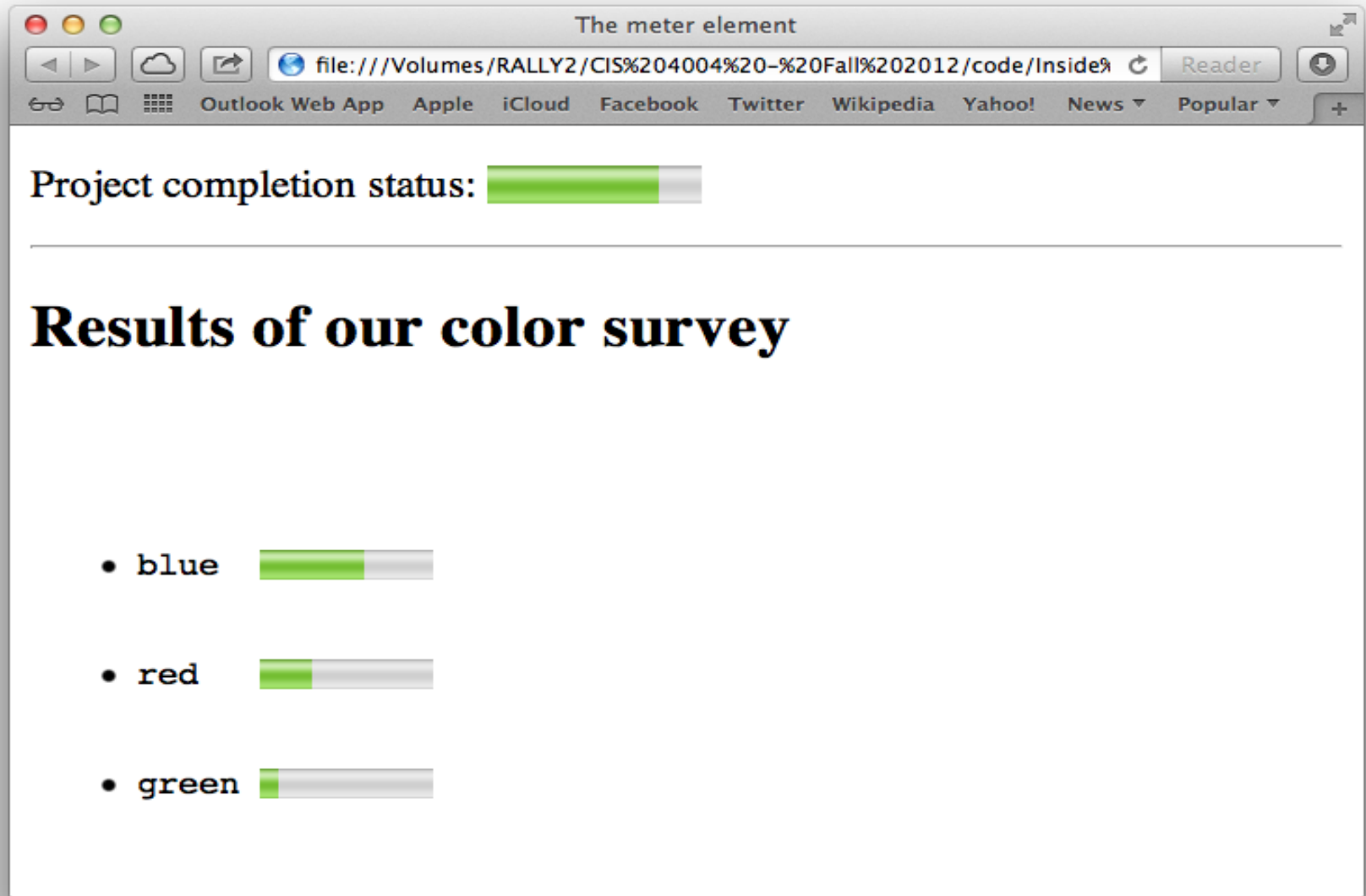
The `meter` Element

The `meter` Element

- The `meter` element supports several attributes:

  - The `value` attribute is the only one that is required, and specifies the value of the meter display.

  - The `min` and `max` attributes, default to 0 and 1.0 respectively, if omitted.

  - The `low`, `high`, and `optimum` attributes work together to split the range into low, medium and high segments. `Optimum` specifies the optimum position within the range, such as "0 brake pad wear" in the following example (see next page). Set `optimum` in between if neither a low nor high value is optimal.

  - The example on the next page illustrates the use of the attributes for the `meter` element.

The `meter` Element Using Attributes

```html
<!DOCTYPE html>
<html lang="en">
<head>
        <meta charset="utf-8" />
        <title> The meter element using attributes </title>
</head>
<body>
    <p> Weekly Riding Mileage Goal Status – Week of September 23rd as of Tuesday:
      <meter min="0" max="260" value="110">110 miles of 260 completed</meter></p>
    <hr />
    <p>Car brake pad wear: <meter low="0.25" high="0.75" optimum="0" value="0.22">
        22% worn as of now</meter></p>
</body>
</html>
```

the meter element with attributes.html

The meter element using attributes

file:///Volumes/RALLY2/CIS%204004%20-%20Fall%202012/code/Inside%20HTML5%20-%20Part%202/the%20meter%20element%20with%20attributes.htm

Outlook Web App   Apple   iCloud   Facebook   Twitter   Wikipedia   Yahoo!   News ▾   Popular ▾

Weekly Riding Mileage Goal Status - Week of September 23rd as of Tuesday:

Car brake pad wear: