CHAPTER 2: HTML

Note

□ All examples for this chapter are located at

https://swe.umbc.edu/~zzaidi1/is448/chap2-examples

□ You can look at the HTML code of the examples by doing a right click on the browser window displaying the page, and selecting 'View Source' option on the menu that pops up

How do I use the examples folder?

- When class starts, open the examples folder in the browser
  - When you see the note ‘See example:’ in the slides, it refers to an example program found in the above folder. Open it up and follow along.
- When working on a lab exercise
  - First, open the PDF slides of the lecture from Blackboard.
  - Then, find the slide that describes the lab exercise
  - Then, open the examples folder for that lecture in the browser. Usually, a lab exercise builds on examples shown in previous slides.
  - Then, start working on the lab exercise

HTML: Hypertext Markup Language

- Describes the content and structure of information on a web page
- Written in plain text
- Fundamental syntactic units: tags

Structure of an HTML page

- □ Header describes the page
- □ Body contains the page’s contents
- □ An HTML page is saved as a .html or .htm file

```
<html>

<head>
Info about the page
</head>

<body>
Page contents
</body>

</html>
```

First, some terminology…

HTML Elements and HTML Tags

- An HTML page is made up on HTML elements
- An element is made up of tags and content
  - Syntax: <tag>content</tag>
  - Example: **<p>** This is a paragraph**</p>**
- If element has no content, open and close tag are together
  - Syntax: <tag />
  - Example: **<br />**
- Most whitespace insignificant in HTML
  - Collapsed into a single space

HTML Attributes

- Some HTML tags can have attributes
  - Syntax:

    ```
    <tag attribute=“value” attribute=“value> content </tag>
    ```

  - Example:

    ```
    <a href=“page2.html”> Next </a>
    ```

- Tags without content
  - Syntax:

    ```
    <tag attribute=“value” attribute=“value />
    ```

  - Example:

    ```
    <img src=“rabbit.jpg" alt="A rabbit" />
    ```

Block vs. inline tags

- Two different types of tags
  - Block-level tags contain an entire large region of text
    - Examples: *paragraphs, lists, table cells*
    - A block tag breaks the current line so that its content appears on a new line
  - Inline tags affect a small amount of text and must be nested inside of a block-level tag
    - Examples: *bold text, code fragments, images*
    - The content of an inline tag appears on the current line
      - The inline tag does not implicitly include a line break
      - The browser allows many inline elements to appear on the same line or within the same block-level element
      - With the exception of the <br /> tag whose purpose is to insert newlines

Nesting of Tags

- Block tag cannot appear in content of inline tag
- Inline tag and text cannot directly be nested in body or form elements
- Only block tags can be nested directly in a body or a form element

Best coding practices to follow

- Tags **must** be lowercase
- Attributes **must** have values
- Each open tag must have a corresponding close tag
- Enforce correct nesting of elements

HTML5

□ Latest evolution of the standard that defines HTML

□ New elements, attributes and behaviors introduced

□ Allows you to describe your content precisely

□ audio and video elements to embed multimedia content

Difference in structure

HTML document

```
<html>

<head>
Info about the page
</head>

<body>
Page contents
</body>

</html>
```

HTML5 document

```
<!DOCTYPE html>
<html>

<head>
Info about the page
</head>

<body>
Page contents
</body>

</html>
```

Tags in HTML5 document

- □ Now, let us look at the tags within the <head> element
- □ Example: See alltags_lecture2.html

<head> element

- Typically, contains the
  - <title> element
  - <meta> element
- Example: See alltags_lecture2.html

Adding a title to page

- **title** element goes inside **head** element
- Displayed in browser’s title bar
- Is a required tag for XHTML validation
  - Must be present within the <head> tag in a page
- Syntax:

```
<title> This is the IS448 course </title>
```

- Example: See alltags_lecture2.html

The *meta* Element

- Used to provide additional info about document
- Place into page’s <head> element
- Two attributes
  - name (commonly chosen value for name=“keywords”)
  - content (corresponding words that characterize doc.)

```
<meta name=“keywords” content=“binary trees, linked lists, stacks” />
```

- Can also use

  <meta name = “descriptions” content=“a description here” >
- Some web search engines use info provided within the *meta* element to categorize and index web pages

Tags in HTML5 document

- Now, let us look at the tags within the <body> element

<body> element

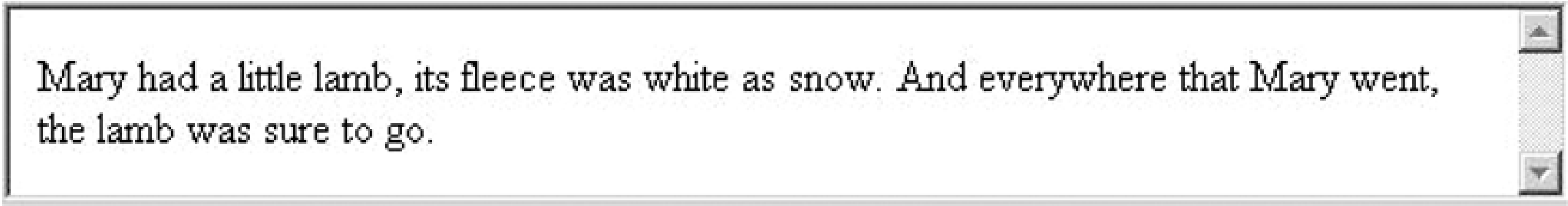
- Contains all other elements, such as
  - Paragraph
  - Line break
  - Headings
  - Image
  - Form
  - Audio
  - Video
  - etc.

Paragraphs

- Paragraph tag: <p> text here </p>
- Block-level tags
- Example: See alltags_lecture2.html

```
<p> Mary had
a
little lamb, its fleece as white as snow. And
everywhere that
Mary went, the lamb was sure to go </p>
```

HTML code



Mary had a little lamb, its fleece was white as snow. And everywhere that Mary went,
the lamb was sure to go.

Displayed in browser without white spaces

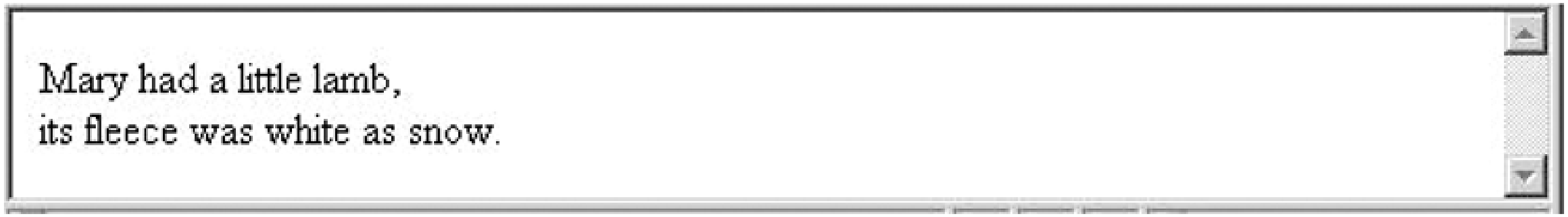
Line breaks

- □ Break tag: <br />
- □ Inline tag. must be nested inside block tag
- □ Example: See alltags_lecture2.html

```
<p> Mary had  a
little lamb, <br />
its fleece as white as snow. </p>
```

HTML code

Browser display



Mary had a little lamb,
its fleece was white as snow.

HTML Comments

- Use comments to increase readability of programs
- Comments should describe what the program does, when it was created, who created it etc. in English
- Syntax:
- Example:

```
<!–- comment text here -->
```

- Example: See alltags_lecture2.html

```
<!-- This document describes me -->
```

Headings

- Used to separate major areas of text
- Block tag
- Types of headings
  - <h1>, <h2>, <h3>, <h4>, <h5>, <h6>
- Presentation differences
  - 1, 2, 3, use font sizes larger than default font size
  - 4 uses the default font size
  - 5 and 6 use smaller font sizes
- Example: See alltags_lecture2.html

Images

- Most common formats: GIF, JPEG
- More recent: PNG
- HTML tag for images
  - Inline tag, must be nested inside block tag
  - Syntax:

    ```
    <img />
    ```

  - Simplest form: Two attributes: **src** and **alt**
  - Example:

```
<img src=“images/c210.jpg” alt=“Picture of Cessna 210” />
```

  - Optional attributes: width and height

Example HTML page with Images

Example: See image.html, alltags_lecture2.html

Links – Anchor tags

- Acts as a pointer to some resource
- Resource:
  - another XHTML document anywhere on the web, or some other place in document displayed currently
  - Specified as attributes in anchor tag
- Links are specified using the *anchor* tag
- Syntax:

```
<a href="some url"> text that will be linked </a>
```

- Are inline tags, must be nested inside block tag : must be within <p> or <h1> tags

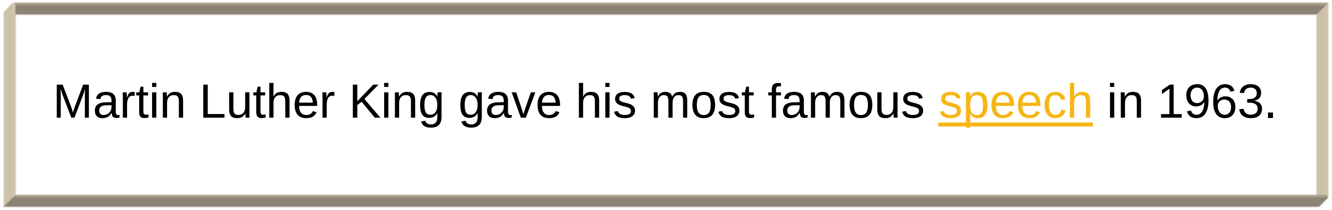
Links - Anchor Tags

- Syntax of <a> tag
  - href attribute specifies destination URL

HTML code

```
<p>Martin Luther King’s gave his most famous
<a href=“http://www.americanrhetoric.com/speeches/
mlkihaveadream.htm”>speech</a> in 1963.</p>
```

Browser display



Martin Luther King gave his most famous speech in 1963.

Example: See alltags_lecture2.html

Links - Anchor Tags

- Content of **anchor** tag becomes clickable link
- Links usually implicitly displayed in a different color than other text
- Links can point to other *local* documents
  - relative paths (e.g., /images/test.html)

    vs.
  - absolute paths (e.g., C:/IS448/images/test.html or http://userpage.umbc.edu/~*username*/images/test.html)
  - Example: See image.html

Using Images as Links

HTML code

```
<p>
  <a href=“http://www.shortorderdogs.com”>
  <img src = "littleDogsRock.jpg"  alt = "A little rocking dog" />
   </a>
   <br />
   This is an example of an image as a link!
</p>
```

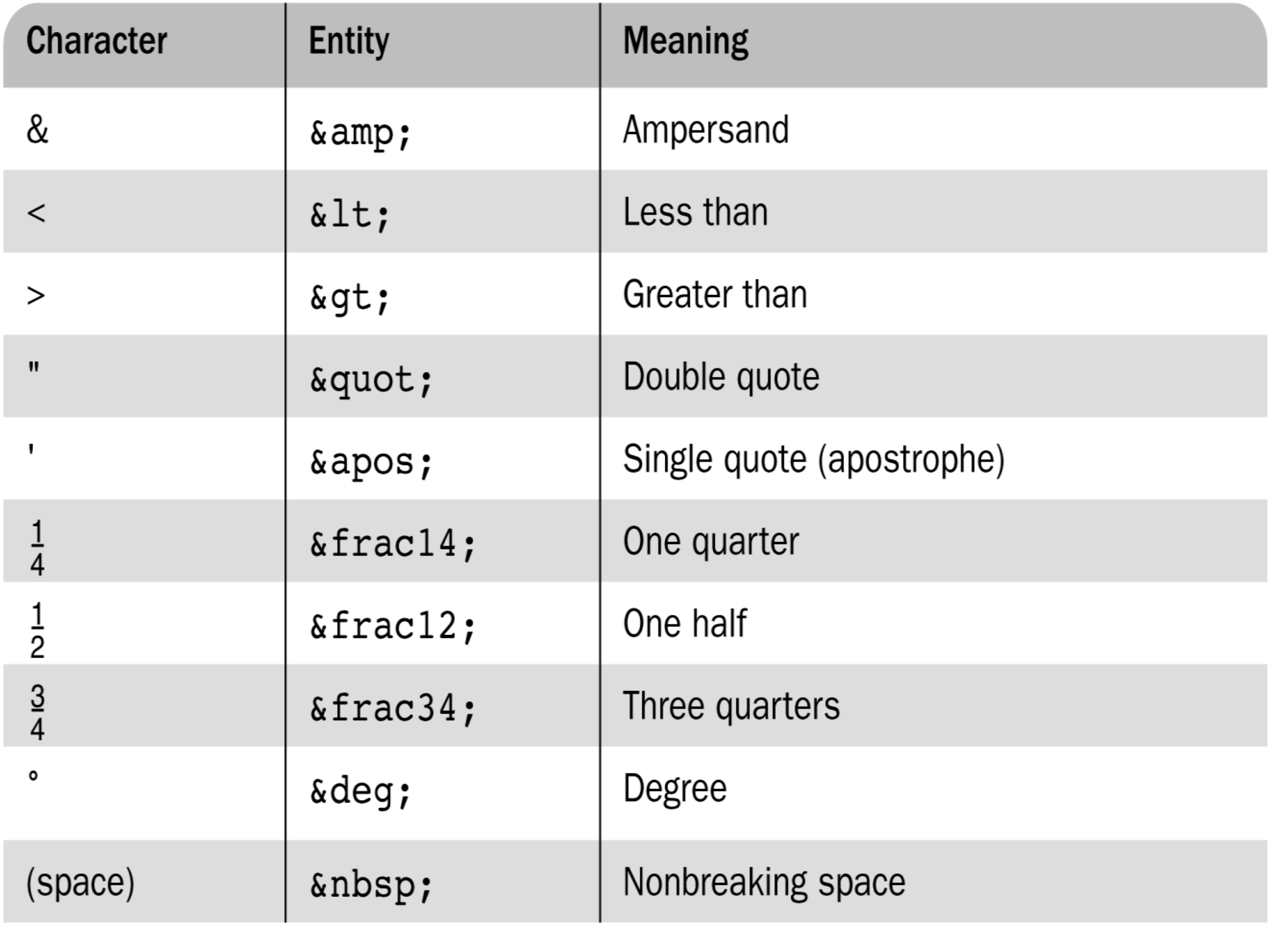
Browser display Example: see image.html

Overview of HTML tags discussed in this lecture

- □ Title tag <title> Page about IS448 </title>
- □ Meta tag <meta name=“keywords” content=“umbc,course” />
- □ Paragraph tag <p> This is some text </p>
- □ Break line tag <br />
- □ Comments <!-- this is a comment -->
- □ Heading tag <h1> IS448 </h1>
- □ Image tag <img src = “plane.jpg” alt=“plane” />
- □ Anchor tag <a href=“test.html”> This is a test </a>

Character Entities

- Character entities are used to specify characters that are needed but cannot be typed directly



Character	Entity	Meaning
&	`&amp;`	Ampersand
<	`&lt;`	Less than
>	`&gt;`	Greater than
"	`&quot;`	Double quote
'	`&apos;`	Single quote (apostrophe)
$\frac{1}{4}$	`&frac14;`	One quarter
$\frac{1}{2}$	`&frac12;`	One half
$\frac{3}{4}$	`&frac34;`	Three quarters
°	`&deg;`	Degree
(space)	`&nbsp;`	Nonbreaking space

Character Entities

- Example

```
<p>
For the ampersand character, the character entity
&amp; 4 &gt; 1 should be used. </p>
```

Example: See alltags_lecture2.html

Editors for programming

- Notepad++ is a great editor on Windows
  - Free software: installed on school computers
  - Google for it and download if you want a copy for your local computer
- Can also use Aptana Studio
  - Open source software: installed on school computers
- Other editors
  - General purpose editors: notepad or jedit
  - HTML editors: Microsoft Frontpage, Macromedia Dreamweaver, Adobe PageMill

Lab 1 (part 1): Labs in this course

- Save all the files you create in this class in your ‘Documents’ folder while working in class on them. Then, either move it to Box or put it on a jump-drive.

- Create sub-folders in ‘Documents’, Box or your thumb-drive, such as **is448/chapter2**

- **Avoid** using **upper case letters** (unless using camel casing) and **spaces** in folder names and file names that you create in this class

Lab 1 (part 2): First web page

Now, let's create our first web page as follows:

1. Download the file mypage.html from the examples folder
   - It is a file with some basic tags but otherwise empty of textual content
2. Open it in Notepad++. Then, do the following
   - Add a title to the page within the <head> tag
   - Within the <body> tag add a comment that describes the content of the page
   - In the body of the page,
     - In a paragraph tag, add the text "My name is *your-name-here*. I am in year _____ of the undergraduate program at UMBC"
     - Add another paragraph of text listing your hobbies
   - Make sure you use the break tag somewhere on this page
3. Save the file as mypage.html
4. Double-click on the file to view the page in the web browser
5. Move on to next part of the lab (next slide)

Lab 1 (part 3): Creating your Internet visible folder

- Go to the handout on Blackboard under CourseDocuments -> Handouts and open the handout titled “InternetVisible.pdf”
- Follow the instructions in this handout
- Once you have completed Section 2 on the handout, come back here and follow the instructions for the next part of the lab in the next slide

Opening a terminal software

- Desktop machine users:
  - If GL doesn't work
    - search for 'Putty' on your machine
    - Type server name as gl.umbc.edu
    - Then give your UMBC username and password when asked
- Personal Windows machine users
  - Download SSH terminal as described in handout or download Putty
- Personal Mac machine users
  - Use Terminal on your software
  - If you are getting a remote server changed error, type: ssh-keygen –R gl.umbc.edu at the terminal prompt and that should take care of the error

Lab 1 (part 4): HTML5 Validator

- The HTML5 validator can be found at http://validator.w3.org/
- Pass your mypage.html through the validator and check if your page is validated correctly
  - Go to the 'Validate by File Upload Tab' and upload your file to validate it
  - Or, submit the URL of your page. If you followed instructions from two slides ago precisely, your page should be at the URL:

https://swe.umbc.edu/~*your_umbc_user_name*/is448/chapter2/mypage.html

Lab 1 (part 5): headings, links, images

- Open mypage.html
- Put the text "My name is *yourname*" in a <h1> tag
- Download an image of your hobby from the web and add text to go with the image like "This is an image of my favorite hobby cooking" **(be wary of copyrighted images)**
- Add two links
  - link to a page on the web that contains more information about your hobby
  - Make your image a link pointing to the original source of the image
- Validate your page

Lab 1(part 6): Securing your pages

Your final tasks for this lab are to:

- □ Create a file called index.html within your sweLatest folder. This should be an empty file with no content. Remember, from Chapter 1, that the web server will display contents of index.html when no specific page is given in the URL.
  - If you do not have a blank index.html, anyone who visits your website will be able to see all your files
  - By having this blank file in your www folder, you are preventing other Internet users from reading/knowing the contents of your directory/folders.
- □ Also, for *each chapter and homework subfolder*, you should create a new blank index.html file and place it in the chapter/homework subfolder

Note about Homework files

- Your homework programs/files should be inside *sweLatest* folder when you submit them
- Typically, you will create a sub-folder inside *sweLatest/is448*, such as *hw1* or *hw2* and place your files inside this folder. To move files from your local machine to the web server, follow the instructions in Section 2 of the handout InternetVisible.pdf available under Course Documents -> Handouts, only your folder name is not *chapter2* but *hw1* or *hw2* etc..
- YOU MUST CREATE a blank *index.html* file in the *hw1*, *hw2*, etc. folders, failing which you will lose points on the homework.
- The index.html is just a blank file which is named index.html. Keeping such a file in a web-visible folder will prevent others from seeing the contents of your folders