CSCI335: Software Engineering via Web Applications

Professor Sprenkle
sprenkles@wlu.edu
Objectives

- Course overview
- Overview of web
- Introduction to HTML
Survey of Your Web Experience

• Uses the WWW?
  ➤ For what?

• Wrote Web pages?
  ➤ Using what tools?

• Wrote JavaScript?

• Wrote Web applications?
Discussion:
What are web applications?

• Give examples of static web pages and web applications
The Internet

- Connection of computer networks using the Internet Protocol (IP)
  - Allows network applications, e.g., email, file transfer, world wide web, remote login, ...
Creation of the World Wide Web

- Result of confluence of
  - Fast computers
  - Internet
  - Hypertext theory, e.g., links to other text

- Inventor: Tim Berners-Lee, a physicist
  - **Vision**: make all information available to all people at all times
  - 1989 at CERN

Web's historic logo designed by Robert Cailliau (Belgian CSist)
Creation of the World Wide Web

- Result of confluence of
  - Fast computers
  - Internet
  - Hypertext theory, e.g., links to other text
- Inventor: Tim Berners-Lee, a physicist
  - **Vision:** make all information available to all people at all times

Web's historic logo designed by Robert Cailliau (Belgian CSist)
Evolution of WWW

• ftp/email ...
• to gopher ...
• to simple html pages ...
• to web sites ...
• to dynamic html ...
• to web commerce, social media, desktop-like applications, ...
Most Popular Web Sites

• Guesses?
# Most Popular Web Sites

<table>
<thead>
<tr>
<th>Site</th>
<th>Domain</th>
<th>Alexa top 100 websites (As of March 23, 2016)[3]</th>
<th>SimilarWeb top 100 websites (As of April 4, 2016)[4]</th>
<th>Type</th>
<th>Principal country</th>
</tr>
</thead>
<tbody>
<tr>
<td>Google</td>
<td>google.com</td>
<td>1</td>
<td>2</td>
<td>Search engine</td>
<td>U.S.</td>
</tr>
<tr>
<td>YouTube</td>
<td>youtube.com</td>
<td>2</td>
<td>3</td>
<td>Video sharing</td>
<td>U.S.</td>
</tr>
<tr>
<td>Facebook</td>
<td>facebook.com</td>
<td>3</td>
<td>1</td>
<td>Social network</td>
<td>U.S.</td>
</tr>
<tr>
<td>Baidu</td>
<td>baidu.com</td>
<td>4</td>
<td>16</td>
<td>Search engine</td>
<td>China</td>
</tr>
<tr>
<td>Yahoo!</td>
<td>yahoo.com</td>
<td>5</td>
<td>5</td>
<td>Portal and media</td>
<td>U.S.</td>
</tr>
<tr>
<td>Amazon</td>
<td>amazon.com</td>
<td>6</td>
<td>14</td>
<td>E-commerce and cloud computing</td>
<td>U.S.</td>
</tr>
<tr>
<td>Wikipedia</td>
<td>wikipedia.org</td>
<td>7</td>
<td>9</td>
<td>Encyclopedia</td>
<td>U.S.</td>
</tr>
<tr>
<td>Tencent QQ</td>
<td>qq.com</td>
<td>8</td>
<td>42</td>
<td>Portal</td>
<td>China</td>
</tr>
<tr>
<td>Google India</td>
<td>google.co.in</td>
<td>9</td>
<td>17</td>
<td>Search engine</td>
<td>India</td>
</tr>
<tr>
<td>Twitter</td>
<td>twitter.com</td>
<td>10</td>
<td>11</td>
<td>Social network</td>
<td>U.S.</td>
</tr>
</tbody>
</table>

**Alexa**: 3-month average of page views, unique site users  
**SimilarWeb**: page views, panel, ISPs  

https://en.wikipedia.org/wiki/List_of_most_popular_websites
## Most Popular Web Sites

<table>
<thead>
<tr>
<th></th>
<th>Rank in Feb 2008</th>
<th>Rank in April 2010</th>
<th>Rank in April 2013</th>
<th>Rank in April 2016</th>
</tr>
</thead>
<tbody>
<tr>
<td>Google</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Facebook</td>
<td>9</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Yahoo! (mail)</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>MySpace</td>
<td>3</td>
<td>6</td>
<td>??</td>
<td>??</td>
</tr>
<tr>
<td>YouTube</td>
<td>10</td>
<td>5</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>eBay</td>
<td>6</td>
<td>12</td>
<td>8</td>
<td>24</td>
</tr>
<tr>
<td>Wikipedia</td>
<td>13</td>
<td>16</td>
<td>5</td>
<td>7</td>
</tr>
</tbody>
</table>

*Not quite a fair comparison – different sites/metrics used over time*
What This Course is About

• Web applications
  ➢ Distributed computing
  ➢ Web application technologies (server and client)
  ➢ How to develop high-quality Web applications
    • Software tools

• Software engineering
  ➢ Large development project
  ➢ More software, collaboration tools
  ➢ Emulate real-world experience with actual client

• Life-skills
  ➢ Reading, writing, discussion, presentation
My Responsibilities

• Prepare useful, interesting knowledge
• Come to class prepared, on time
• Interesting, relevant, and challenging assignments
• Prompt feedback on assignments
Your Responsibilities

• Come to class prepared, on time
• Turn in assignments on time  
  ➢ Don’t get behind when we only have 4 weeks!
• When you’re having trouble  
  ➢ Look for help on the Web  
    • Find, adapt solutions  
    • Give credit to where you found solution, if novel enough  
  ➢ Ask me for help
• Learn, absorb, synthesize  
  ➢ Extra Credit: take it to the next level
Approach to Learning

• Information: comes from lectures
• Knowledge: comes from readings and homework
• *Wisdom*: comes from experience
Evolution of WWW

• ftp/email ...
• to gopher ...
• to simple html pages ...
• to web sites ...
• to dynamic html ...
• to web commerce

Where we’re starting
World Wide Web

- Built on top of the Internet
- Web browsers and Web servers
- Communicate using HTTP (Over IP/TCP)

www.cnn.com

Web Server, e.g.,
Apache, Microsoft IIS

Web Browser,
e.g., Firefox, IE,
Safari, Opera, ...
How Does The Browser Get a Page?

• In Web browser, enter a URL
  ➢ URL: Uniform Resource Locator
  ➢ Protocol
  ➢ Host
  ➢ May not have explicitly typed in “http”
    • Default protocol
    • Other protocols: https, ftp
How Does The Browser Get a Page?

• Look up Host’s IP Address using DNS
  ➢ Need to be able to “find” host on the Internet
  ➢ Routing through Internet is by IP address

• Domain Name System (DNS)
  ➢ Set of servers that map domain name to IP Address(es) and vice versa

  www.espn.com ↔ 68.71.212.158

• Unix commands host and nslookup can lookup this information
How Does The Browser Get a Page?

• Browser now makes the request using HTTP
  ➢ HTTP: HyperText Transfer Protocol

• Common Types of HTTP Requests:
  ➢ GET: download a page
  ➢ POST: download a page
  ➢ HEAD: just get the “header” for a page

• For our example, browser makes request \texttt{GET /}

HTTP GET request

www.cnn.com

Web Browser
How Does the Web Server Serve a Web Page?

- Receives request for a resource on TCP port 80
- Looks for the resource in the Web Document directory
  - Not all files on a Web server are meant for others to see
  - Specific directory holds these files
- If the file is found, server sends an HTTP 200 response with the requested document
  - Otherwise, sends appropriate error response

HTTP GET request

HTTP Response
How Does the Browser Get a Page?

- Receives response from server
- Displays file in appropriate format
Evolution of CNN’s Site

2013 Look:

2010 Look:

2008 Look:

How did he die?

Apr 25, 2016
# HTTP Status Codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>OK: Request succeeded</td>
</tr>
<tr>
<td>3xx</td>
<td>Redirection (temporary or permanent)</td>
</tr>
<tr>
<td>403</td>
<td>Error: No permission</td>
</tr>
<tr>
<td>404</td>
<td>Error: File not found</td>
</tr>
<tr>
<td>500</td>
<td>Internal server error</td>
</tr>
</tbody>
</table>
The Process Without GUI

sprenkle@machine courses$ telnet www.cs.wlu.edu 80
Trying 137.113.118.204...
Connected to terras.cs.wlu.edu.
Escape character is '^]'.
GET /
<html>
<head>
<meta http-equiv="REFRESH" content="0;url=http://www.wlu.edu/x18630.xml">
</head>
<BODY>
<!--You should be redirected to the correct page in a few seconds.-->
</BODY>
</HTML>
Connection closed by foreign host.
More on URLs

• Specifies the location of a resource

• Format: `<protocol>://<host>/*<path>`

  ➢ Examples:
  
  http://www.cs.wlu.edu/~sprenkle/cs335/

  http://www.cs.wlu.edu/~sprenkle/cs335/schedule.php
How Does Browser Get a Page?

- Receives response from server
- Displays file in **appropriate format**

What does that mean?
HTML: HYPERTEXT MARKUP LANGUAGE
HTML: HyperText Markup Language

• Describes the **content** and **structure** of information on a web page
  ➢ Not the same as the **presentation** (appearance on screen)

• **Markup** document with **elements**

• Written in plain text
HTML5

• HTML is an evolving standard

• Earlier versions of HTML
  ➢ “Loose” on standards for how to write/render HTML
  ➢ Benefit: allows for authoring “imperfect” HTML that still renders okay
  ➢ Limitation:
    • Page can look different in different browsers (e.g., mobile devices)
    • Lack of Multimedia, graphical content
    • Lack of semantic information
HTML5

• Goals
  ➢ Support multimedia (Web 2.0 applications)
  ➢ Better definitions for markup
  ➢ Consistent support by browsers
    • Including mobile devices
What does this all mean?

• We’re going to write in HTML5
  ➢ Tendency towards the XHTML syntax
    • Relatively easy, especially with right tools
    • Less ambiguous documents
  ➢ Can be validated

• HTML5 is an evolving standard but current version is a “stable target”
  ➢ With time, even more browser support
Element Syntax

• An element is made up of tags and content
  ➢ Syntax:     <tag>content</tag>
  ➢ Example:   <p>This is a paragraph</p>

• If an element contains no content, open and close a tag together:
  ➢ Syntax:   <tag/>
  ➢ Example:  <hr/>

• Most whitespace is insignificant in HTML
  ➢ It gets ignored or collapsed into a single space
HTML Tags

- Some tags can contain additional information called *attributes*
  - Syntax: `<tag attribute="value" attribute="value"> content </tag>`
  - Example: `<a href="page2.html">Next page</a>`

- Tags without content:
  - Syntax: `<tag attribute="value" attribute="value"/>`
  - Example: `<img src="bunny.jpg" alt="A bunny" />`
Structure of an HTML Page

- The **header** describes the page
- The **body** contains the page's contents
- An HTML page is saved into a file with a name ending in extension `.html` or `.htm`

```html
<html>
<head>
  information about the page
</head>
<body>
  page contents
</body>
</html>
```
Structure of an HTML5 Page

```html
<!DOCTYPE html>
<html>
<head>
  information about the page
</head>
<body>
  page contents
</body>
</html>
```

Says we’re using HTML5
Adding a Title to a Page

- **title** elements go inside of the **head** element
- Displayed in the web browser's title bar and the bookmark text

```
<title>CSCI335: Web Applications</title>
```

This text is the title

- Displayed in search engine results pages
Block-level vs. Inline Elements

Two different types of elements

Block-level Elements
- Contain an entire large region of text
- Examples: paragraphs, lists, table cells
- Browser displays a margin of vertical whitespace between block-level elements for separation

Inline Elements
- Affect a small amount of text
- Must be *nested* inside of a block-level element
- Examples: bold text, code fragments, images
- Browser allows many inline elements to appear on the same line or within the same block-level element
I have a dream that one day this nation will rise up and live out the true meaning of its creed: "We hold these truths to be self-evident: that all men are created equal."
Headings: `<h1>`, `<h2>`, ... `<h6>`

- Headings separate major areas of the page
  - Block-level

```html
<h1>Famous American Speeches</h1>
<h2>Civil Rights Leaders</h2>
<h3>Martin Luther King, Jr.</h3>
```

**Famous American Speeches**

**Civil Rights Leaders**

**Martin Luther King, Jr.**
Headings: `<h1>`, `<h2>`, ... `<h6>`

- Headings separate major areas of the page
  - *Block-level*

- Presentation:
  - 1, 2, and 3 use font sizes that are larger than the default font size
  - 4 uses the default size
  - 5 and 6 use smaller font sizes
Hypertext Links/Anchors: `<a>`

• Basis of Web
• Links to other pages or parts of a page
• `href` attribute specifies the destination URL
• Are *inline* elements
  ➢ Must be placed inside a block-level element, such as `<p>` or `<h1>`

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

Martin Luther King gave his most famous *speech* in 1963.

Apr 25, 2016
Sprenkle - CSCI335
More on Anchors: `<a>`

- **`href`** URLs can be
  - Relative: another page within this Web site
    - Just the resource path
    - Easier to maintain than absolute paths
  - Absolute: full URL

```html
<p>Today’s <a href="lab0.php">lab</a>.</p>
<p><a href="http://www.w3schools.com/tags">HTML Help</a></p>
```
Making links

• “Click here” is intrusive—the control interferes with the content

• Press the blue text is even worse

• A link traversal should be ancillary

• Good approaches:
  - Highlight glossary terms
  - Embed the links in footnotes
    • Who uses this approach a lot?
  - Allow the anchor to fit into the rest of the document naturally
## Optional Anchor Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Meaning</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>target</strong></td>
<td>Where to open the target URL</td>
<td><code>_self</code> = default, this window/frame, <code>_blank</code> → new window, <code>_top</code> → full window, <code>_parent</code> → parent frame</td>
</tr>
<tr>
<td><strong>title</strong></td>
<td>Text to display in a tool tip</td>
<td>The tool tip text</td>
</tr>
<tr>
<td><strong>name</strong></td>
<td>Names an anchor. Use to create a bookmark in a document.</td>
<td>The name</td>
</tr>
<tr>
<td>Or <strong>id</strong> (preferred)</td>
<td></td>
<td>Example in Wiki: CS335 Resources</td>
</tr>
</tbody>
</table>
Wikis

- Collaborative web sites
- Content generated by registered users or anyone
- Goal: easy development of content
  - Update content using a browser
- Our course Wiki:
  - http://cswiki.wlu.edu/dokuwiki/doku.php/courses/cs335/home
  - Powered by Dokuwiki, a PHP-based Web application
Nesting Tags

• Tags must be correctly nested
  ➢ A closing tag must *match the most recently opened* tag

• Browser may render it correctly anyway, but maybe not

• Bad HTML:

```html
<p><a href="00-intro.pdf">Lecture</a></p>
<p>This text also links to Lecture</p>
```
**Line Break: `<br/>`**

- Forces a line break in the middle of a block-level element (*inline*)

```html
<p>Today’s <a href="lab0.html">lab</a><br/>
The lab is due on Tuesday.</p>
<p>Tuesday’s lab coming soon...</p>
```

Today’s **lab**.
This lab is due on Tuesday.
Tuesday’s lab coming soon...
Horizontal Rule: `<hr/>`

• A horizontal line to separate sections of a page visually (*block*-level)

```html
<p>First paragraph</p>
<hr/>
<p>Second paragraph</p>
```
Images: `<img>`

- Inline element
- Inserts a graphical image into the page
  - `src` attribute specifies image URL
  - XHTML requires an `alt` attribute that describes the image

```html
<p><img src="lol_cat.jpg" alt="LOL Cat"/></p>
```

`alt`: Used by search engines, screen readers, browser if image can’t be displayed
Images as Links

- Nest inside an `<a>` tag
- `<title>` attribute for tool tip

```html
<p><a href="http://icanhascheezburger.com"> <img src="f10_cat.jpg" alt="F10 Cat" title="LOL Cat" /></a></p>
```
Practice

• You want your page to look like:

Favorite LOL Cats

![Blah, Blah, Blah](image1)

![And den she comes outta da tv like dis...](image2)
Comments <!-- -->

- Document your HTML file
- Comment out text

<!-- Lab Web Page, By Sara Sprenkle -->
<p> Lab is due <!-- tomorrow --> today. </p>

Lab is due today.
Three Types of Lists

• Unordered Lists `<ul>`
• Ordered Lists `<ol>`
• Definition Lists `<dl>`
Unordered List: `<ul>`, `<li>`

- `<ul>` represents a bulleted list (block-level)
- `<li>` represents a single item within the list (block-level)

```html
<ul>
  <li>6-cylinder engine</li>
  <li>Cruise control</li>
  <li>Automatic locks</li>
</ul>
```

- 6-cylinder engine
- Cruise control
- Automatic locks
Nested Unordered Lists

• a list can contain other lists

<ul>
  <li>Standard Features:
    <ul>
      <li>4-cylinder engine</li>
      <li>CD player</li>
    </ul>
  </li>
  <li>Optional Features:
    <ul>
      <li>Cruise control</li>
      <li>Automatic locks</li>
    </ul>
  </li>
</ul>

• Standard Features:
  ▪ 4-cylinder engine
  ▪ CD player
• Optional Features:
  ▪ Cruise control
  ▪ Automatic locks
Ordered List `<ol>`

- `<ol>` represents an ordered (by default, numbered) list of items (block-level)

```html
<p>Underpants Gnomes’ Three-Phase Business Plan</p>
<ol>
  <li>Collect underpants</li>
  <li>?</li>
  <li>Profit</li>
</ol>
```

Underpants Gnomes’ Three-Phase Business Plan

1. Collect underpants
2. ?
3. Profit
Common Error: Not Closing a List

Underpants Gnomes’ Three-Phase Business Plan
1. Collect underpants
2. ?
3. Profit

More information
Definition List: `<dl>`, `<dt>`, `<dd>`

- `<dl>` represents a list of definitions of terms (block-level)
- `<dt>` represents each term
- `<dd>` represents the term’s definition

```html
<dl>
  <dt>&lt;dl&gt;</dt>
  <dd>represents a list of definitions of terms</dd>
</dl>
```

```html
<dl>
  represents a list of definitions of terms
</dl>
```
HTML Character Entities

- A way of representing any Unicode character within a web page

<table>
<thead>
<tr>
<th>Character(s)</th>
<th>Entity</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt; &gt;</code></td>
<td><code>&amp;lt; &amp;gt;</code></td>
</tr>
<tr>
<td>“ &amp;”</td>
<td><code>&amp;quote; &amp;amp;</code></td>
</tr>
<tr>
<td>™ ©</td>
<td><code>&amp;trade; &amp;copy;</code></td>
</tr>
<tr>
<td>π δ Δ</td>
<td><code>&amp;pi; &amp;delta; &amp;Delta;</code></td>
</tr>
<tr>
<td>é è ñ</td>
<td><code>&amp;eacute; &amp;egrave; &amp;ntilde;</code></td>
</tr>
</tbody>
</table>

How would you display `&amp;` on a web page?
Phrase Elements: `<em>`, `<strong>`, `<code>`

- **em**: emphasized text, usually rendered in italic
- **strong**: strongly emphasized text, usually rendered in bold
- **code**: a short section of computer code, usually rendered in a fixed-width font

```html
<p>The `<code>ul</code>` and `<code>ol</code>` tags make lists.</p>
<p>HTML is `<em>really</em>`, `<strong>REALLY</strong> fun!</p>
```

The ul and ol tags make lists.

HTML is *really*, **REALLY** fun!
Phrase Elements: `<sub>`, `<sup>`

- **sub**: subscript characters
- **sup**: superscript characters

```html
<p>x <sub>2</sub><sup>3</sup> + y <sub>1</sub><sup>2</sup></p>
```

$x_2^3 + y_1^2$
Bad HTML

\[
\text{\textless p} \text{This is \textless strong>the \textless em>best</strong> time \textless /em> to invest!</textless p>
\]

• What is wrong?

• How did the author want this to display?

• How can we write it correctly?
Bad HTML

```html
<p>This is <strong>the <em>best</em></strong> time <em>to invest!</em></p>
```

- What is wrong?
  - Open/close tags aren’t matching

- How did the user want this to display?
  - This is the <strong>best time</strong> to invest!

- How can we write it correctly?

```html
<p>This is <strong>the <em>best</em></strong> time <em>to invest!</em></p>
```
Practice Problem

• We want the page to look like:

The Simpsons Fun Facts

• Longest-running scripted series
• Started on The Tracy Ullman Show
Preformatted Text: `<pre>`

- A large block of code text in a fixed-width font (block-level)
- `<pre>` blocks are displayed with exactly the whitespace and line breaks given between the tags

```java
public static void main(String[] argv) {
    System.out.println("Hello, world");
}
```

```java
public static void main(String[] argv) {
    System.out.println("Hello, world");
}
```
Preformatted Text: `<pre>`

- A large block of code text in a fixed-width font (block-level)
- `<pre>` blocks are displayed with exactly the whitespace and line breaks given between the tags

```xml
<pre>
    public static void main(String[] argv) {
        System.out.println("Hello, world");
    }
</pre>
```

How would it look if we had instead enclosed it in `<code>` tags?
Tags for Quotations

• `<q>`: a short, *inline* quotation
  - Typically displayed with quotation marks, italics

  ```html
  <p>Quothe the Raven <q>Nevermore.</q></p>
  ```

  Quothe the Raven, “Nevermore.”

• `<blockquote>`: a lengthy, *block-level* quote

  ```html
  <p>Lincoln, in his Gettysburg Address:</p>
  <blockquote><p>Four score and seven years ago, our fathers ...</p></blockquote>
  ```

  Lincoln, in his Gettysburg Address:
  
  *Four score and seven years ago, our fathers ...*
Tables `<table>`

- **Used to layout tables**
  - Sometimes used for layout but better techniques (like CSS), which we’ll cover later

- **Matrix of rows and columns**

- **Cell** is an intersection of row and column
  - Contains almost any document element, e.g., text, headings, images, tables, ...

- **Table caption**: `<caption>`
  - Inside of a `<table>` element
  - Defines the title that proceeds a table
Table Row: `<tr>`, Label: `<th>`, Data: `<td>`

- **tr**: defines a row in the table
- **th**: defines a row or column’s label (heading)
- **td**: defines a table’s data cell

```html
<table border="2">
  <caption>Baseball Statistics</caption>
  <tr>
    <th>Name</th>
    <th>Batting Average</th>
  </tr>
  <tr>
    <td>Cal Ripken, Jr.</td>
    <td>.276</td>
  </tr>
</table>
```
Table Row: `<tr>`, Label: `<th>`, Data: `<td>`

```html
<table border="2">
  <caption>BaseballStatistics</caption>
  <tr>
    <th>Name</th>
    <th>Batting Average</th>
  </tr>
  <tr>
    <td>Cal Ripken, Jr.</td>
    <td>.276</td>
  </tr>
</table>
```

Baseball Statistics

<table>
<thead>
<tr>
<th>Name</th>
<th>Batting Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cal Ripken, Jr.</td>
<td>.276</td>
</tr>
</tbody>
</table>

Notice formatting
rowspan, colspan Attributes

• Some columns need to span multiple columns
  ➢ Similarly with rows
• Use the colspan attribute to specify how many columns to span
  ➢ Use in <th> or <td> tags

```html
<tr>
  <td></td>
  <th colspan="3">Career Statistics</th>
</tr>
```

An empty data cell

Spans 3 columns
Table Sections

• Can separate the table into three parts: header, body, and footer
  ➢ Denoted with \texttt{thead}, \texttt{tbody}, and \texttt{tfoot}
  ➢ Useful for presentation (Tues)

• Header includes column labels

• Body includes table data, including row labels
  ➢ If multiple body sections, browser marks with thicker horizontal lines

• Footer sometimes includes column labels again and totals for columns
W3C HTML Validator

http://validator.w3.org

- Checks your HTML code to make sure it meets the HTML specifications
- More picky than the Web browser, which may render malformed HTML correctly
- Link included as part of Firefox plug-in WebDeveloper
Web Page Metadata: `<meta>`

- Represents information about your page
  - For Web browser, search engine, etc.
- Placed into a page’s `head` element
- Not displayed in the page itself
- `meta` tags have
  - `content` attribute and
  - `name` or `http-equiv` (for HTTP headers) attribute

```html
<meta name="description" content="HTML help page" />
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"/>
```
Web Page Metadata: `<meta>`

- Describe the page

```html
<head>
<meta name="author" content="author’s name"/>
<meta name="revised" content="web page version and/or last-modified date" />
<meta name="generator" content="software used to create the page"/>
</head>
```
Web Page Metadata: `<meta>`

- Suggestions to search engines about how to index your page
- Search engine may choose to ignore them

```html
<head>
  <meta name="description" content="for search engine" />
  <meta name="keywords" content="words to associate with your page, comma-delimited" />
</head>
```
Web Page Metadata: `<meta>`

- Aid browser or Web server

```html
<head>
<meta http-equiv="refresh" content="how often to refresh the page in seconds" />
<meta http-equiv="Content-Type" content="what type of document this is (may include internationalization info such as character encoding)" />
</head>
```

- Can also use refresh tag to redirect, e.g.,
  ```html
  content="5;url=http://www.cs.wlu.edu"
  ```
Favorites Icon (favicon)

- **link** tag - placed in **head** element
  - Can specify an icon for a web page
  - Icon is placed in browser title bar and bookmark/favorite
- **Syntax:**
  ```html
  <link rel="shortcut icon" type="MIME type" href="filename" />
  ```
  - Default type is **ico**
Favicon Example: W&L Web Site

```
<link rel="shortcut icon" href="prebuilt/images/favicon.ico?v=1"/>
```

16x16 pixel image
Internet Media ("MIME") Types

- **Multipurpose Internet Mail Extensions**
  - Classifications of data that travel over the internet
  - Originally developed for email

- Used to tell the browser the form (**type**) of a file returned by the server

- Type specifications
  - Form: **type/subtype**
Internet Media ("MIME") Types

- **Multipurpose Internet Mail Extensions**
  - Sometimes used to specify something’s type when including resources on a page

- **Form:** type/subtype

<table>
<thead>
<tr>
<th>MIME Type</th>
<th>File Extension</th>
</tr>
</thead>
<tbody>
<tr>
<td>text/html</td>
<td>.html</td>
</tr>
<tr>
<td>text/plain</td>
<td>.txt</td>
</tr>
<tr>
<td>image/gif</td>
<td>.gif</td>
</tr>
<tr>
<td>image/jpeg</td>
<td>.jpg</td>
</tr>
<tr>
<td>video/quicktime</td>
<td>.mov</td>
</tr>
</tbody>
</table>
Presentation of Web Pages

- Talked mostly about structure and content of HTML pages
- Want presentation to be separate
  - Don’t encode style into the HTML page itself
  - Easier to apply different styles to a set of web pages or a whole web site
Cascading Style Sheets (CSS)

• Describe the appearance, layout, and presentation of information on a web page
  ➢ As opposed to HTML, which describes the content, structure of the page

• Describe *how* information is to be displayed, not what is being displayed

• Can be embedded in HTML document or placed into separate `.css` file
  ➢ Advantage of separate `.css` file: one style sheet can be shared across many HTML documents
Attaching a CSS File: `<link>`

- **link** appears in **head** element
- Can link to multiple style sheet files
- Syntax:
  ```html
  <link rel="stylesheet" type="text/css" href="filename"/>
  ```
- Example from W&L site:
  ```html
  <link rel="stylesheet" type="text/css" href="http://www.wlu.edu/prebuilt/assets/css/global.css" media="only screen" />
  ```
My First Web Page

- Circa 1995

http://www.cs.wlu.edu/~sprenkle/personal/China.html
Outline for the Course

- Static Web Pages
  - HTML5, Presentation (CSS)
- Usability
- Dynamic Web Pages, Applications
  - Server-side (Servlets, JSPs, …)
- Testing Web Applications
- JDBC
- Other Java frameworks/tools
- Java Script, AJAX
- JSTL
- Security

Project Development

Apr 25, 2016
Sprenkle - CSCI335
Introduction to Group Projects

- 2 digital humanities projects
- Teams of 3 or 4 people
- Worth 38% of your course grade
- Different projects, different implementation, and development opportunities
  - All will be publicly accessible!
  - Promote on your resume!
Ancient Graffiti Project

- Client: Rebecca Benefiel, associate professor of classics at W&L
- 3-4 team members
- Final Deliverable: An interface for adding and updating inscriptions found in Pompeii and Herculaneum
  - Goal: Look at the inscriptions in context of the location
  - Reports of inscriptions
- Expected skills/technology leveraged during project: database access, manipulation, updates; remote database access; user interaction; usability; digital humanities
- Extensions: Dynamic searching features, mapping, mining data for new information
- Expected audience: scholars and students of Pompeii and Herculaneum can use the tool to learn more about the culture at the time of Vesuvius's eruption.

http://agp.wlu.edu/
Symbolic Logic Tutorial

- Client: Dr. Paul Gregory, associate professor of philosophy
- 3-4 team members
- Final Deliverable: An interface to make symbolic logic tests easier on both students and faculty
  - Take quizzes
  - Graded quizzes
- Expected skills/technology leveraged during project: database access, manipulation, and updates; user interaction; access restrictions; usability; digital humanities
- Extensions: dynamic interaction
- Expected audience: philosophy students and faculty
  - Beyond just W&L

http://logic.wlu.edu/
Planning for the Semester

• Group Project
  ➢ Tomorrow a.m.: choose teams; discuss project development
    • (Also CSS)
    • In Parmly 404 (seminar room)
  ➢ Tomorrow p.m.: meet with me about projects

• Today: typical of semester
  ➢ Part lecture, part discussion, part lab
Lab 0

- Intro to the wiki
- Tools set up
- Create web pages in HTML5
  - Some things you need to have
  - Feel free to be creative and do other things
TODO

• Lab 0
• Create your project preference list
• Explore Course Web Page
  ➢ Particular attention to the schedule, examples
  ➢ Read over the project page on the course web site
• Read about how Google search works
  ➢ Write summary in Sakai forum by Wednesday midnight
• Read on what images can use
  ➢ Write summary in Sakai forum by Wednesday at midnight