

Web Development From the Ground Up, a Series for Novice Computer Users

Lecturer: Philip Matuskiewicz

Thursday: October 15, 2009

Email: phil@famousphil.com or pjm35@buffalo.edu

Lecture Website (created during 1st lecture):

<http://famousphil.com/09web>

My Background

- I've been a website developer for 7 years
 - Have developed in PHP / MySQL extensively for the past 3 years
 - I program most pages in XHTML Strict or Transitional
- Computer Science Grad Student
 - 4th year of the BS/MS combined degree program
- I've developed for both professional and personal websites
- I currently host 10 websites professionally on a private system.
- I maintain 4 servers
 - 1 Exchange 2007/Windows Server 2003 server
 - 2 CentOS (Redhat) Linux Servers running Cpanel (the hosting industry standard control panel)
 - 1 Ubuntu server responsible for handling all external mail and forwarding it onto the internal servers appropriately
- These servers (except Windows) average 200 days of uptime with minimal problems.

Why am I doing this Lecture?

- I enjoy sharing my knowledge with others
- I learn from this also! Preparing the slides forces me to look up terms that I usually assume “just work”
- I need experience in communicating my knowledge effectively
 - If you don't understand me, chances are no one else does, don't hesitate to stop me
 - I need feedback on how I'm doing! This lecture series is my way of gaining experience to become a better presenter and teaching assistant!

What is a Browser

- A browser is any application that displays websites from a remote server through the internet
- There are multiple browsers out there to choose from:
 - Firefox (Most popular)
 - Chrome
 - Safari
 - Opera
 - Internet Explorer (Microsoft's Browser)

Why do we need a browser?

- Browsers speak HTTP (HyperText Transfer Protocol) which is the language that web servers speak
- A web server will return a file with text in it (among other types of files; eg. Image files)
 - The main file returned is plain text and is written in HTML(HyperText Markup Language) which is the standard that all browsers should be able to understand
 - All files have extensions...
 - common extensions for HTML are .html, and .htm.
 - If the server is modifying the html of a page prior to sending it to the browser, the extension may be .php for PHP, .cgi for CGI, and .asp for ASP (these will be discussed in later lectures)

How does the browser communicate with the server?

- The internet is a huge computer network that stretches world wide
- When your browser requests a page, it uses the internet to find then connect to the server responsible for that page
 - This is basically done through a combination of DNS (Domain Name Service) lookups and routers that connect every computer together some way.
 - These can be compared to how the post office routes your mail through its system, they generally look at the to address and route your mail towards the large sorting facility which knows more about your location. Once it figures out what the sorting destination is (DNS), it will route the message there via planes and trucks (routers).

Focuses of this Lecture Series - XHTML

- Web Development in XHTML Transitional/Strict
 - XHTML is Extensible HyperText Markup Language, it is just an improved HTML
 - Most of my examples and pages are XHTML Strict compliant but it is easier to declare Transitional to make that one difficult part of the webpage line up correctly.
 - This is why many of the pages I develop are declared Transitional
 - The strict specification will make your XHTML look cleaner and guarantee that it will work better on most browsers as where transitional is generally easier to validate.

Sidetrack to focuses on XHTML

- All too often, when developing websites, all browsers will show your webpage differently. This is because they follow the same specifications but interpret them differently, or they may not follow the specifications at all.
 - Internet Explorers 6/7 are a huge problem because they didn't really follow the specifications. IE 8 has gotten a lot better! This is why on the IEEE website, I have separate code for CSS in IE vs Firefox and other browsers.
 - Chrome follows the standards almost exactly as does Internet Explorer 8 (Thank you Microsoft!). This is why sites may look fine in Firefox (which sometimes will loosely follow the standards and fill in the gaps) where IE8 / Chrome will make your page look ugly.
- Before standards were widely followed, browsers were really good at detecting errors and figuring out what they meant. This is why poorly coded WebPages still would look fine to all browsers that they were tested in.

Focuses on CSS

- CSS is “Cascading Style Sheet”
 - This is a plain text file that will be requested from the server if in the HTML header.
 - This file has formatting information on where to place text and how to make text look among a lot of other visibility features of a webpage.
 - The common file extension for CSS is .css

Focus on Design

- All WebPages nowadays look nice to the eye and generally are designed with several images
- I will use Adobe Photoshop CS4 at some point in the series (Tentatively Lecture 3) to display a template that serves one of the websites I develop for.
 - At this point, I plan on slightly modifying the template and then carry it to a website. I plan to do this live as a demonstration, explaining the tools that I use in Photoshop as I go along.

Focus on PHP / MySQL development

- The last 3-4 lectures of this series will focus heavily on PHP and MySQL development.
 - PHP stands for Hypertext Preprocessor
 - I've also seen it as: Pre Hypertext Processor
 - MySQL stands for My Structured Query language
- PHP and MySQL are combined to generate HTML pages with constantly changing content.
- I'm currently working on integrating MySQL into the IEEE website to make the site easier to modify without my assistance.

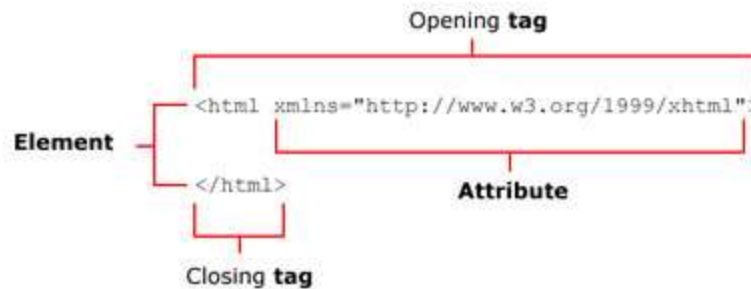
Enough generalized information...

What does XHTML look like?

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" dir="ltr" lang="en-us" xml:lang="en-us">
  <head>
    <!-- This is a comment tag, it will NOT be interpreted by HTML->
    <!-- title, meta, and other non-content tags go in the head ->
      <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
      <title>Web Dev Lecture Series :: Home</title>
    </head>
    <body>
      <!-- any page content will go inside the body tags->
        <p>
          Hello World! This lecture series is on its way!!!<br />
          <a href="http://famousphil.com/09web/lecture1.pdf">Lecture 1 Slides</a>
        </p>
      </body>
    </html>
```

HTML Tags

```
<html xmlns="http://www.w3.org/1999/xhtml" dir="ltr" lang="en-us" xml:lang="en-us">
</html>
```



- Looking at the image, in all html documents, we have an opening tag, a closing tag, and a list of attributes.
 - Attributes will always show up as name="value" in proper html and xhtml.
 - Attributes vary depending on what tag they are inside of
- Some tags are self closing, for example, the `
` tag has a `/>` at the end which self closes it
 - br means Break Line or goto the next line

Header of a Strict HTML Document

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

- !DOCTYPE tells the browser what language we are using
 - html will almost always follow this declaration!
- PUBLIC specifies this specification is accessible by anyone
- The owner (W3C) is specified next
- DTD is the Document Type Definition
- XHTML 1.0 Strict is the language used followed by //EN specifying English
- Finally in another set of quotes (“”) we place the URL (Universal Resource Locator) address of the page containing the specification requirements
- W3C is the organization that specifies HTML Standards

Continuing on the header:

- `<html xmlns="http://www.w3.org/1999/xhtml" dir="ltr" lang="en-us" xml:lang="en-us">`
- The `xmlns` attribute specifies the extended markup language namespace or standard.
 - XML is a more simplified version of HTML where you can specify your own tags and attributes. This is especially useful for defining databases and holding onto data in files without mixing data up.
- `dir` is the direction the text flows (ltr is Left to Right)
- `Lang` specifies the language – United States English
- `xml:lang` specifies the language of the tags being used

Continuing into the head tag

```
<head>
```

```
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

```
  <title>IEEE-UB :: Home</title>
```

```
  <link rel="stylesheet" type="text/css" href="style.css" />
```

```
</head>
```

- Between the `<head></head>` tags is where page information is contained
- The first meta defines most importantly what the document language is and what character set it is using (to prevent encoding errors)
- A title is specified between the `<title></title>` tags
- The `<link...>` tag is specifying where to get the style sheet for this page, we will come back to this in a later lecture

Body

```
<body>
```

```
  <p>Hello World! I made my first webpage!!!<br />
```

```
  <a href="http://famousphil.com/09web/lecture1.pdf">Lecture 1  
Slides</a>
```

```
  </p>
```

```
</body>
```

- Between the `<body></body>` tags is where you should put all the content that will show up on your webpage.
- All text should be surrounded by another tag. In my examples for today, I will use the `<p>` tag which is the paragraph tag.

The `<a>` Anchor Tag

- The href attribute defines the link URL or address.
 - This address can be relative to where the page is , or it can be a full path link including http://
 - For example, if we are at <http://famousphil.com/09web/index.html> and we specify href="lecture1.pdf", it will link to <http://famousphil.com/09web/lecture1.pdf> because we are in the same directory

Now that we've seen some XHTML...

Lets discuss XHTML editors

- Dreamweaver – Not Free!
 - WYSIWYG (what you see is what you get)
 - I never use WYSIWYG editors because they put unnecessary code on your page. Frontpage 2003 is a huge example, Dreamweaver CS4 is much better!
- PHP Designer 2008
 - Text only – recommended, not free! (Student discounts available)
- Scite- Free (not as nice as PHP Designer)
 - http://www.ebswift.com/Common/ASPCommon/Download/file_download.aspx?File=/OpenSource/SciTEInstaller/scite-win32-2.01.msi
- Notepad- included with windows
- Pico / Nano / Vim – Included with Linux – discussed later
 - Vim will highlight your code

Before Transferring Files...

- Be aware that Windows text editors (especially wordpad) will add unnecessary space characters to the end of each of your lines.
 - This can make your html files show up on a webpage the wrong way.
 - If you are finding your editor does this, switch editors!
 - PHP Designer has a handy function “remove empty lines” on its tools menu which I use frequently when copying files back and forth from a Linux web server.

With our HTML file created, lets get it on a server

- Filezilla is perhaps the easiest method
 - Use FTP (File Transfer Protocol)
 - or sftp (Secure file transfer protocol)
 - <http://filezilla-project.org/download.php>
 - Host is the server address, then Username/Password
- Live demo with filezilla –
 - Making this presentation and the website available at the lecture series website:
<http://famousphil.com/09web>
 - Note the www. is optional (exception: UB CSE department)
 - Using FTP
 - Using SFTP (uses SSH discussed later)

Editing files live on the server

- Using a program like sftp drive which maps a remote Linux drive to a Windows drive.
 - This is possible but not free
 - quick demo using sftpdribe
 - its now expandrive at <http://www.expandrive.com/windows>
- Live demo
 - Using pico, nano, vim via putty and ssh
 - SSH command list (I made it for CSE116) available at:
 - Slideshow (if we have time)
 - <http://www.cse.buffalo.edu/~pjm35/ta/116/linuxhelpV2.pdf>
 - Summary of Commands (Word / PDF)
 - <http://www.cse.buffalo.edu/~pjm35/ta/116/CSE116UnixCheatsheet.doc>
 - <http://www.cse.buffalo.edu/~pjm35/ta/116/CSE116UnixCheatsheet.pdf>