

Introduction to Web Development



CS/IT 490 WD Fall 2013

Last update 2013-08-16

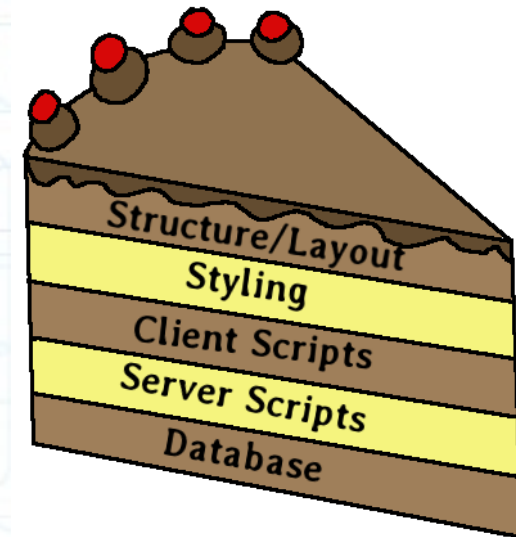
Written by Rachel J. Morris
Creative Commons Attribution-NonCommercial-ShareAlike
3.0 Unported License

Breakdown

- What goes into web development?
- What are all these different languages?!
- What we will learn in this class
- Why Open Source?
- Goals

What goes into web development?

- There are many layers when it comes to web development (and, with software development, too!)
- It is good practice to separate out things like the page content from the styling, and those things from the behind-the-scenes logic.



What goes into web development?

- User Experience (UX) is important to make sure users can easily navigate the webpage and things look nice. This includes HTML & CSS.
- Data calculated, procured, or otherwise obtained by the server-side scripts need to be plugged into the HTML pages.
- Server-side scripts will be passed in data and return data – a web request that returns a database query?
- Data may need to be stored and retrieved from a database.

What are all these different languages?!

- XHTML: eXtensible Hypertext Markup Language
 - The page structure
 - Not a programming language. Don't call it a programming language. It's markup.
 - Early versions of HTML had tags for styling, such as `<color="#FF0000">` or `<center>`
 - These tags are **deprecated**, and styling should **be handled by CSS now**.
 - XHTML is for **creating the elements of a webpage**.

What are all these different languages?!

- HTML vs. XHTML:
 - XHTML is a derivative of HTML
 - It is based on XML
 - Makes HTML more consistent, obvious if it is wrong (uneven tag opening/closing, etc.)
 - It is very close to HTML 4.01
 - Mandatory tags added
 - Formatting style required

What are all these different languages?!

- XHTML Mandatory tags:
 - DOCTYPE
 - `<html>`, `<head>`, `<title>`, `<body>`
- XHTML Formatting:
 - All elements & attributes are lower-case
 - All attribute values enclosed within quotes
 - All elements must be closed
`<body> ... </body>`

Source: http://www.w3schools.com/html/html_xhtml.asp

Last update 2013-08-16

Written by Rachel J. Morris
Creative Commons Attribution-NonCommercial-ShareAlike
3.0 Unported License

What we will learn in this class

- CSS – Cascading Style Sheets
 - Also use CSS for specifying positions, alignments, dimensions, etc. of elements on a webpage.
 - CSS can be frustrating at first but it's a great tool.
 - You CAN inline CSS into your HTML code -- *but don't*.

What we will learn in this class

- PHP – PHP: Hypertext Preprocessor
 - Used for server-side scripting
 - Similar to C++ syntax
 - Many PHP libraries out there
 - Can add two numbers together, or query a database, or interact with a third-party API (Application Programming Interface)

What we will learn in this class

- PHP – PHP: Hypertext Preprocessor
 - Very very easy to write messy code
 - Web frameworks exist to lessen writing of “boilerplate” code.
 - CakePHP
 - Symfony
 - CodeIgniter

What we will learn in this class

- PHP – PHP: Hypertext Preprocessor
 - Many “ready-to-go” web solutions also are written in PHP
 - Wordpress
 - Drupal
 - Joomla
 - PhpBB

What we will learn in this class

- MySQL
 - RDBMS (Relational Database Management System)
 - Store data in a database with **tables**
 - **Tables** can have relationships to each other (therefore, *Relational* Database)
 - We will focus on very simple database schemas; this is not a formal RDBMS Design class.

What we will learn in this class

- JavaScript, JQuery, AJAX
 - JavaScript is a client-side scripting language
 - JQuery is a library to extend JavaScript and helps you write easy, unobtrusive, clean code
 - AJAX allows you to send/receive data to a server-side call from the client-side (obvious symptom: No refreshing!)

What we will learn in this class

- JavaScript, JQuery, AJAX
 - Client-side JavaScript can be used to *enhance* a webpage.
 - Thumbnails that blow up when clicked
 - Animated drop-down boxes
 - Pop-up modal windows
 - It is good practice to make sure your webpage can be navigated *without* JavaScript enabled, however.

Why Open Source?

- It's (usually) free: No cost
- You can view the source code!
 - You could edit it to fit your needs, if desired
- Lots of people use it because it's Open Source
 - More help, documentation, plugins, etc.
 - Plugins for proprietary systems may end up costing more

Why Open Source?

- It's (usually) free: No cost
- You can view the source code!
 - You could edit it to fit your needs, if desired
- Lots of people use it because it's Open Source
 - More help, documentation, plugins, etc.
 - Plugins for proprietary systems may end up costing more

Why Open Source?

- http://www.youtube.com/watch?v=7NShpY_oSs4
- Everybody can peer-review Open Source code. Anyone can contribute to Open Source code. Anyone can fix Open Source code.
- Proprietary solutions can cost \$\$\$
 - Sometimes forbidding for small companies & individuals

Goals

1. Be able to build a competent & functional dynamic webpages with Open Source technology like PHP and MySQL.
2. Be able to design and build websites with good UX, with HTML, CSS, and JavaScript.
3. Be able to design & implement basic database schemas.
4. Set up a web server with Apache, and/or be able to set up websites on a hosted service.

Goals

Any goals you want to add?

References

Cover photo found at <http://imgur.com/a/Qlh7Y>

Introduction to XHTML, w3schools, http://www.w3schools.com/html/html_xhtml.asp