ITMD-461

Class 2
Sept 3, 2014

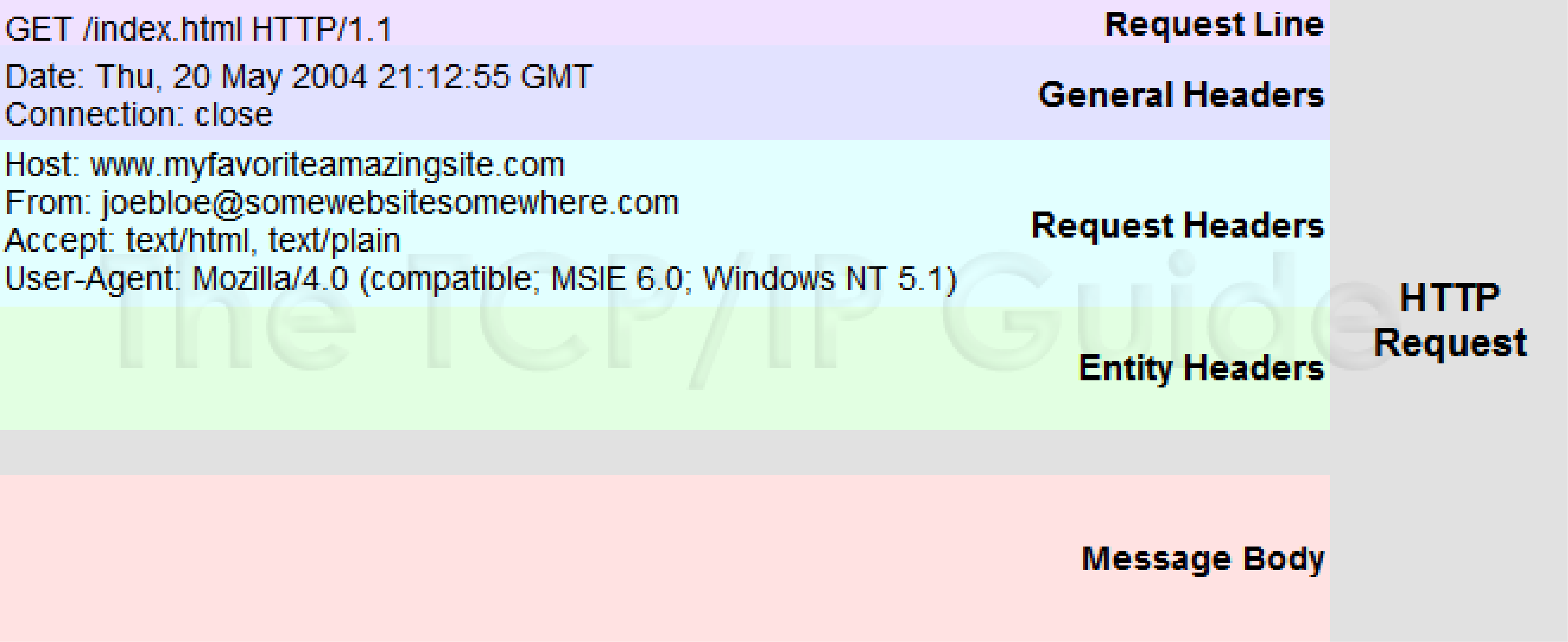
ITMD-461

- Tonight's Agenda
  - Review HTTP Request/Response / show examples
  - The Browser Components
  - Browser Developer Tools
  - Introduction to HTML, CSS, JavaScript Technologie
  - Basics of Tags and Page Structure
  - Build a basic page

HTTP Request/Response

- Client Parses the URL
  - protocol://server/request-URI
- Client sends request to Server
  - Usually GET or POST method, headers
- Server send response to Client
  - Includes status code, headers and content
- Process repeats for all assets on the page

HTTP Request



```
GET /index.html HTTP/1.1
Date: Thu, 20 May 2004 21:12:55 GMT
Connection: close
Host: www.myfavoriteamazingsite.com
From: joebloe@somewebsitesomewhere.com
Accept: text/html, text/plain
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
```

Request Line

General Headers

Request Headers

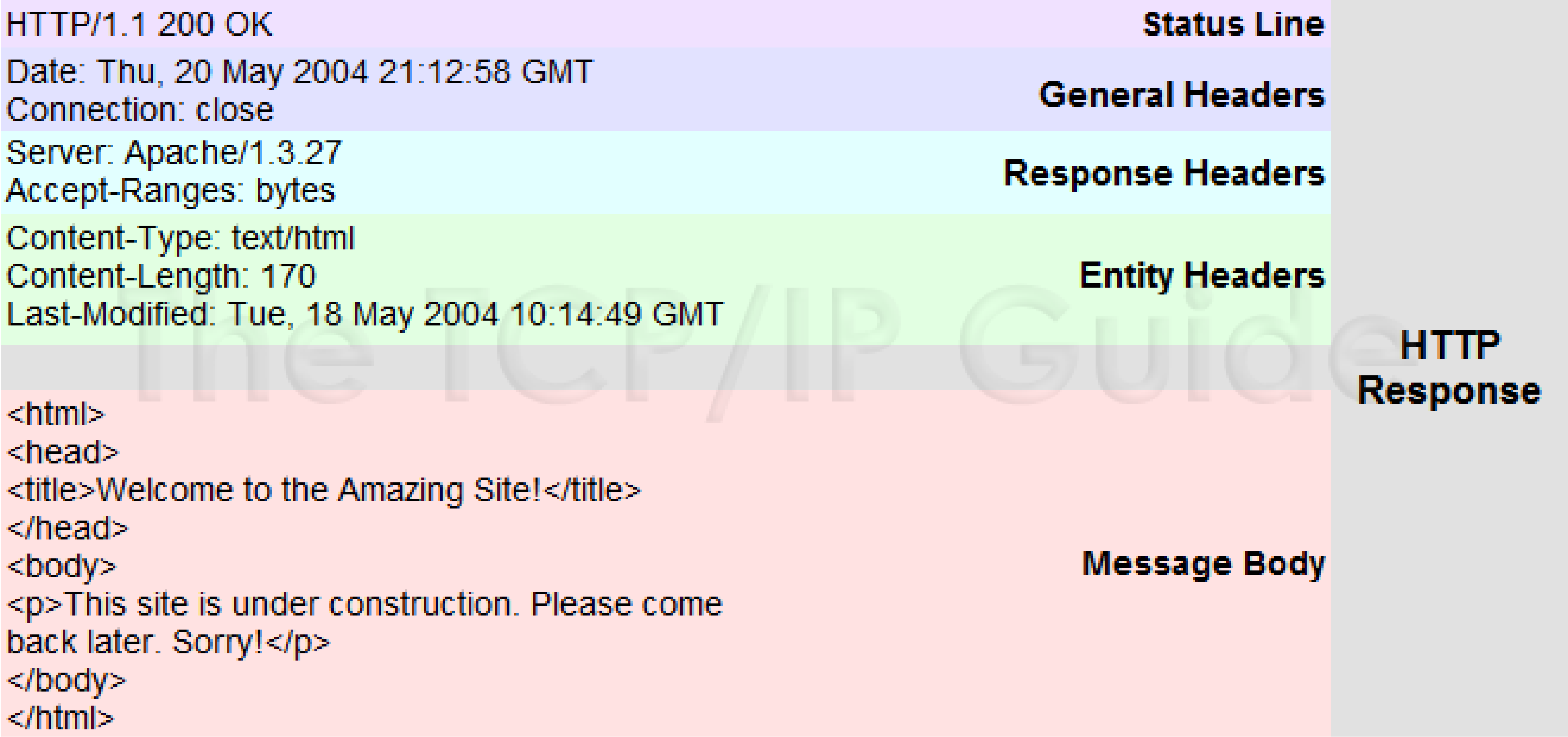
Entity Headers

Message Body

HTTP Request

http://www.tcpipguide.com/free/t_HTTPRequestMessageFormat.htm

HTTP Response



```
HTTP/1.1 200 OK
Date: Thu, 20 May 2004 21:12:58 GMT
Connection: close
Server: Apache/1.3.27
Accept-Ranges: bytes
Content-Type: text/html
Content-Length: 170
Last-Modified: Tue, 18 May 2004 10:14:49 GMT

<html>
<head>
<title>Welcome to the Amazing Site!</title>
</head>
<body>
<p>This site is under construction. Please come
back later. Sorry!</p>
</body>
</html>
```

http://www.tcpipguide.com/free/t_HTTPResponseMessageFormat.htm

HTTP Request/Response

- Demo using Graphical HTTP Client (mac)
- http://web-sniffer.net/
- http://www.rexswain.com/httpview.html

The Browser as the Platform

- What is a browser?
  - How do you build a browser?
  - What are the functional requirements of a browser
  - What needs to be implemented to build a browser
  - What are the engines in a browser?
  - What is your competition?

The Browser

- Browsers are pieces of SOFTWARE written for an operating system. Browsers are compiled C / C++ / Objective-C or Java code.
- The browser is responsible for the following
  - Communicating with the computer's networking card to package up, send and receive HTTP commands (remember HTTP stands for hyper text transfer protocol)
  - Reading and rendering HTML markup
  - Reading and rendering CSS styles
  - Reading and executing Javascript code
  - Providing the user with an interface to use the HTTP protocol and domain name services – typing in http://www.google.com to initiate HTTP GET requests

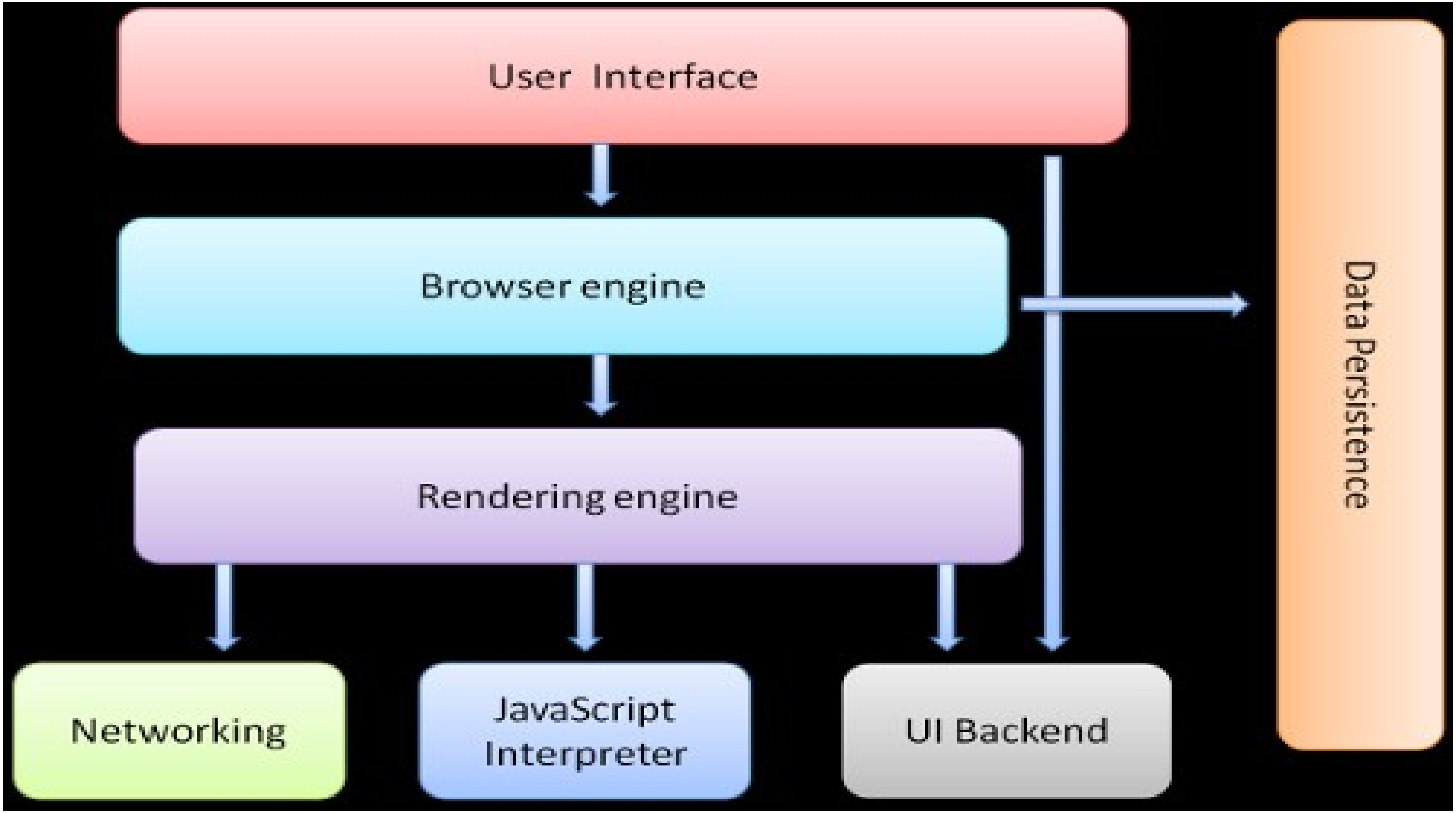
The Browser

- The browser implements the following protocols and technologies
  - HTTP, Hyper text transfer protocol
  - FTP, File transfer protocol
  - HTTPS, Secure Hyper text transfer protocol
  - DNS – Domain Name services to translate IP addresses to URLs (Uniform Resource Locator), also called URIs (Uniform resource identifier)
  - The HTML (Hyper text markup language) specification as outlined by W3C
  - The Javascript specification as outlined by the W3C
  - The CSS (cascading style sheet) specification as outlined by the W3C

The Browser Components

- In most cases, a modern browser is designed as a combination of multiple components.
- The User Interface
- The Browser Engine
- The Rendering Engine
- Networking
- UI Backend
- JavaScript Interpreter
- Data Storage

The Browser Components



http://www.html5rocks.com/en/tutorials/internals/howbrowserswork/

The Browser Components

- The user interface
  - Address bar, back/forward buttons and all other parts of the browser display except the main content area.
- The browser engine
  - Controls the interaction of the UI and rendering engine
- The rendering (layout) engine
  - Responsible for parsing and displaying the content
- Networking
  - Used for networking calls like HTTP. Uses a platform independent interface with a platform specific implementation underneath.

The Browser Components

- UI Backend
  - Used for drawing basic widgets. Platform independent interface. Underneath uses the OS UI methods.
- JavaScript Interpreter
  - Parses and executes JavaScript
- Data Storage
  - A persistence layer. The browser needs to save various things to the system hard disk, some examples are cookies, cache items, bookmarks. HTML5 spec defines a 'Indexed Database' which is a lightweight database in the browser and 'Web Storage' which is a simple key value store.
  - http://www.html5rocks.com/en/features/storage

The Browser Rendering Engine

- The rendering (layout) engine parses the marked up content and formatting information and displays it on the screen.
- Each browser has a different rendering engine
  - Firefox – Gecko
  - Safari – Webkit
  - Chrome – Webkit, Blink (Version 28+)
  - Internet Explorer – Trident
  - Opera – Presto (up to version 15), Blink (version 15+)
  - Konqueror – KHTML
    - Developed by KDE project. Webkit is a forked version.
  - http://en.wikipedia.org/wiki/List_of_layout_engines
- A description of browser parsing and rendering of content is available at this link.
  - http://www.html5rocks.com/en/tutorials/internals/howbrowserswork/

The Browser JavaScript Engine

- JavaScript engine interprets and executes JavaScript (or ECMAScript).
- There has been a race for who can develop the fastest JavaScript engine.
- New generation engines implement just-in-time compilation (JIT)
  - Program code is stored in memory
  - Code is compiled to native machine code just before it is executed
  - Provides high-speed execution of interpreted or byte code

JavaScript Engines

- JavaScript Engines implementing JIT variations
  - Firefox – SpiderMonkey → IonMonkey
  - Chrome, node.js – V8
  - Safari – SquirrelFish known as Nitro
  - Opera – Carakan (version 10.5 – 13), V8 (version 14+)
  - IE >= 9 – Chakra
- Before IE 9 was just called JScript and was part of Trident rendering engine.
- http://en.wikipedia.org/wiki/List_of_ECMAScript_engines

Development Browsers

- For class please use one of the following browsers for you primary development work.
  - Firefox
  - Chrome
  - Safari
- These browsers have many add-on tools that help with web development.
- They are considered the most “developer friendly” browsers
- IE can be used to check your layouts for IE specific bugs and rendering problems. Varies by version.

Browser Tools

- Firefox
  - Firebug, html, css, javascript debug tool
  - Native element inspector, added tilt recently
  - Web developer toolbar
- Chrome
  - Native element inspector
  - Firebug lite
- Safari
  - Native element inspector
  - Develop menu option in preferences

Introduction to the Technologies

- Meet the technologies
  - HTML
  - CSS
  - Javascript
  - These are called “client side technologies” and the browser is referred to as a client, also called a User Agent
  - What about Server and backend code? - coming soon.

Introduction to the Technologies

- HTML – HyperText Markup Language
- HTML was invented in the early 90s - see the “dive into html 5” on blackboard for more details – created by Tim Berners-Lee
- HTML is the **CONTENT** layer of your webpage / website. HTML provides the **STRUCTURE** and **SEMANTIC** meaning of your content
- Your HTML is how google, google spiders, and other search bots, see your page, so make it good.
- We are on HTML version 5

Introduction to the Technologies

- CSS – Cascading Style Sheets
- CSS provides a mechanism to **DECORATE** your html tags.
- CSS is the **PRESENTATION** layer of your website / web page
- CSS allows you to make style rules to govern position, colors, fonts, images, and other visual formatting to your HTML tags
- CSS is directly integrated with HTML through the selectors you use to style content

Introduction to the Technologies

- JavaScript – ECMA compliant Object Oriented language
- JavaScript or JS is the **BEHAVIOR** layer of your web page or website. Javascript is a programming language. It is capable of the following
  - Moving things around the screen
  - Manipulating HTML tags and CSS rules
  - Implementing cool browser APIs like location and local storage.
  - And more...

Introduction to the Technologies

- Server
  - The server is just a powerful computer with a lot of processors and a lot of RAM
  - The server listens for HTTP requests and serves responses to clients when requested
  - To set up a server, we must have 3 things
    - Static IP Address / constant internet connection
    - Domain Name mapped to the static IP address
    - Apache, IIS or other server software running and configured with the IP address and Domain name information

Introduction to the Technologies

- Backend technologies / code
  - Backend code allows you to use various programming languages on the server. Server side code is processed by code execution processes on the server, and may integrate with a database and the filesystem of the server
  - Server side code can react to HTTP POST information as a result of a client filling out a form, or pushing a button or any of the other HTTP methods
  - Server side code reacts to method requests, runs some code and talks to databases, and returns HTML, CSS and JS

Intro to HTML

- HTML is a Markup Language, it is not a programming language
- HTML is PARSED
- HTML needs valid structure to be parsed accurately
  - if you loose the formatting in Microsoft word doc it does not know how to display the page. If your markup is invalid the browser does not know how to display the page.
- HTML is used to structure content on a webpage and provide **SEMANTIC** meaning of your content

Intro to HTML

- Files should end with .html extension
- **Do not use spaces in a file name or special chars**
  - Common to use _ or – instead of space
  - Limit filename to letters, numbers, underscores, hyphens and periods.
- **Filenames may be case-sensitive on some OS's**
- Shorter filenames are better
- All lower case letters with hyphens separating words is a common convention
- The browser will ignore extra whitespace, line breaks, unrecognized markup and comments

Intro to HTML

- HTML is comprised of text and tags. Tags follov specific pattern and must be formatted correctl
- HTML vs XHTML
  - Previous versions were HTML 4.01 and XHTML 1
  - Now we will be using HTML5 influenced by some XHTML syntax
    - All tags must be lowercase
    - All tags must be opened and closed properly
    - All attributes in tags must be properly quoted
    - All tags must be properly nested

Anatomy of an HTML tag

- <a href="http://www.google.com" title="A Link">Google</a>
  - The < is the **OPENING** angle bracket. The angle bracket tells the browser that markup is starting
  - “a” is the **ELEMENT**. All HTML tags are comprised of elements and attributes.
  - href=”” is an **ATTRIBUTE**. HTML tags can have one or more attributes applied to them. Attributes follow name/value pair convention. href is the name, http://www.google.com is the value
  - “Google” is the **TEXT NODE VALUE** of the html tag, this is the text a user will see
  - </a> is the **CLOSING TAG**. It repeats the element with a forward slash before it
  - Do not use the typographic curly quotes “ ”, use straight " " quotes instead. Curly quotes will break your markup. Biggest culprit is not using a plain text editor or coping and pasting carelessly.

Basic HTML Page Structure

- There are 4 main components to a proper HTML5 skeleton structure.
  - A doctype declaration – to let the browser know the mime type of this document
  - A html root element
  - A head section with information describing the page
  - A body section for the part of the web page the user will see
- All pages should have this basic structure
- Let's look at what that basic page structure looks like. Also on page 56 in the book.

Basic HTML5 Page Structure

```
<!DOCTYPE html>
<html>

    <head>
        <meta charset="utf-8">
        <title>Title Here</title>
    </head>

    <body>
    Page content here
    </body>

</html>
```

Basic Tags Overview

- <h1>Head</h1> thru <h6>Head</h6>
- <img src=”imagename.jpg” >
- <a href=”page.html”>contact us</a>
- <p>Here is a paragraph</p>
- <div>Here is a division or container</div>
- This is <span>a span</span> in a sentence.
- <strong>Strong Importance</strong>
- <em>Emphasized text</em>
- <br> or <br />

Build a Basic Page

- Let's go through the sample page in chapter 4.

Software

- Text Editor
  - Notepad++ (windows)
  - Textpad (windows)
  - TextWrangler (mac)
  - Sublime Text 2 (multi-platform)
- SFTP
  - WinSCP (windows)
  - Filezilla (multi-platform)
  - Cyberduck (mac)
  - Transmit (mac)
- SSH
  - Terminal (mac & linux)
  - PuTTY (windows)

Readings

- Have read Part I and start Part II in our book
- Make sure you understand the HTML5 Skeleto structure we talked about in class and is discussed in the readings.
- Read through Chapter 1 of the “Dive into HTML book I put on blackboard (up to p 34)

Assignment

- Do the survey in blackboard (Due Sept 10 by 11:59pm Chicago Time – **not accepted late** – will get 5 points)
- **Assignment 1 (10 points):**
  - Go to Dice.com or any other job posting site
  - Search for web development jobs
  - Keyword search, HTML, CSS, JavaScript, Jquery, Front-end, UX design, UI design
  - Create a pdf or jpeg screenshot out of **one** job posting details not the list of postings
  - Upload to Blackboard under assignment 1
- **Due Sept 12 – 23:59 Chicago Time**
  - **No late work accepted**