Announcements

- Instructor: Nelson Padua-Perez
  - Web Site→ http://www.cs.umd.edu/~nelson
  - Email →nelson@cs.umd.edu
  - About E-mail

    http://www.cs.umd.edu/~nelson/classes/utilities/emailingNelson.html
- Class Web Site:

  http://www.cs.umd.edu/class/summer2015/cmsc389N/

- Password protected sections
- Slides updated often; always go to the class web site to retrieve latest version
- Regarding laptop/mobile device used in class
- Check class announcements daily
- Let's go over the syllabus

Fundamentals: Client/Server

- Client and server are two terms frequently used
- Client/Server Model
- Client/Server Model when talking about software
- Client/Server Model when talking about hardware

Fundamentals: IP Addresses

- Unique address for machine on internet
  - Get from ISP when connecting to internet
  - Allows network to find your machine
- Internet Protocols IPV4, IPV6
  - Define how data is sent between computes over packet-switched network
- (IPV4) Internet Protocol Version 4
  - 32-bit unsigned integer → 128.8.128.8
  - Domain name → cs.umd.edu
  - localhost → 127.0.0.1
- (IPV6) Internet Protocol Version 6
  - 128-bit address
  - Designed to replace IPV4
  - Addresses exhaustion of addresses associated with IPV4 (now we have 2^128)
  - Format: http://msdn.microsoft.com/en-us/library/aa921042.aspx

Fundamentals: Web Server

- Web Server
  - Computer program that delivers (serves up) web pages
  - It is like a person that is in charge of a warehouse
- Popular Web Server Programs
  - Apache → http://www.apache.org/ Free!!!
  - IIS → Internet Information Services
  - Sun Java System Web Server
- You can install and run a web server in your computer
  - https://www.apachefriends.org/
- Local address: http://localhost or http://127.0.0.1/
  - Let's access the local web server and create a user account
- If you use a port different from default (80) you must specify it (e.g., http://127.0.0.1:8080/)
- Web server statistics
  - http://news.netcraft.com/archives/category/web-server-survey/

Fundamentals: DNS

- DNS → Domain Name Systems
- Protocol for translating domain names to IP addresses
    - Example: cs.umd.edu → 128.8.128.44
- Multiple DNS servers on internet
- DNS server may need to query other DNS servers
    - edu DNS server queries umd.edu server to find cs.umd.edu

Fundamentals: URLs

- URL – Uniform Resource Locator
- Represents web resource
  - Arbitrary files
  - Web pages
- Examples
  - http://www.cs.umd.edu/index.html
  - ftp://www.cs.umd.edu/pub/doc/policies.pdf
  - https://login.yahoo.com/
  - file://dir/my.txt

Fundamentals: URL Structure

URL consists of:

- Protocol
  - http
  - ftp
  - https (secure http)
  - file
  - …
- IP address (or domain name)
- Port (optional most of the time)
  - http://www.cs.umd.edu:80/
- Path

Fundamentals: Chrome

- Browser we will use
  - https://www.google.com/chrome
  - We will grade your project using that browser
- Some nice free apps from the chrome web store
  - https://chrome.google.com/webstore
  - 20 Things I learned About Browsers and The Web
  - PHP Code Editor (you don't need to use it)

Database Management Systems

- Database System
  - Allows you to store data and process data (e.g. submit queries, create indices, etc.)
- Examples:
  - Records of students at UMCP
  - Information about telephone numbers
- Software Systems
  - Oracle
  - MySQL (the one we will use ☺)
  - Postgres
- You can install MySQL in your computer

HTML

- Language used to define web pages
- What the server sends to the browser
- Browser reads HTML and renders the page
  - **May require downloading data from server (e.g., images)**

HTTP

- Hypertext Transfer Protocol (HTTP) → protocol that defines how user agents (e.g., browser) and web server can communicate
- HTTP is a request/response protocol between clients and servers
- Some methods (operations) defined as part of the protocol
  - GET → Use to download a resource (e.g., image, web page). Most common method used
  - HEAD → Returns only the header
  - POST → Submits data (e.g., form data) to the server
- Do not confuse with HTML

PHP

- What is PHP?
  - Server-side, cross-platform, HTML Embedded scripting language
  - Text files with .php extension
- What does it allow us to do?
  - To dynamically generate HTML
  - To interact with other systems (e.g., DB Systems, File Systems)
- Examples
  - Flight Information
  - Application System

Software Installation

- Alternatives for Apache, MySQL, and PHP
  - Individual installation of packages (See Resources section, web page)
  - Bundles (include Apace, MySQL, PHP, OpenSSL, etc.)
  - XAMPP (Linux, Windows)
    - http://www.apachefriends.org
    - Mac is also supported although you may want to try MAMP(see below)
  - MAMP →Macintosh, Apache, MySQL, and PHP
    - https://www.mamp.info/en/
    - We are using free version (Not MAMP PRO)

Software Installation

- Alternatives for Editors
  - Notepad/WordPad or any other text editor ☹
  - Komodo Edit
    - http://www.activestate.com/komodo_edit/
    - Do not confuse with Komodo IDE (Not free)
    - Works in Windows, Max, Linux
  - Eclipse
    - http://www.eclipse.org/pdt/index.html#download
    - http://www.phpeclipse.com/
- **Your assignment for tomorrow**
  - Install Apache and PHP

Web Hosting

- As a student of the university you have a directory (folder) where you can place your html documents so they can be accessed on the web
- **Location of your folder**
  - Machine: terpconnect.umd.edu
  - Login by using your directoryID/Password combination
  - Folder location: /pub/YOURDIRECTORYID
- **To activate:**
  - Visit **http://www.it.umd.edu/new/**
- **Web Page Address**: http://www.terpconnect.umd.edu/~YOURDIRECTORYID
  - Notice there is a ~ before your directory id
- **Software/Instructions to Transfer Data**
  - http://www.cs.umd.edu/~nelson/classes/utilities/TransferingDataToTerpconnect/
  - https://dav.terpconnect.umd.edu
- **Free/Low Cost web hosting**
  - http://www.cs.umd.edu/~nelson/nicesites/SmallBusinessWebsites.html

Creating Web Pages

- HTML - HyperText Markup Language
- HTML Standard
  - Developed by the World Wide Web Consortium (W3C)
  - http://www.w3.org
- Document is described through a series of commands and directives present in a text file
- HTML goal is to describe structure only. Presentation should be left to cascading style sheets
- When interpreted by an HTML **viewer** those commands determine the appearance of the page
- HTML documents are entirely ASCII text
- Commands are explicitly inserted
- HTML/CSS tutorial site
  - http://www.w3schools.com/

HTML Editors

- Text Editor
  - Any text editor (e.g., wordpad, notepad, pico, etc.)
- Authoring tools
  - Dreamweaver →Fairly complex but powerful
- Recommended:
  - Komodo Edit → http://www.activestate.com/komodo_edit/

HTML

- We will use HTML5
  - Nice reference for HTML5
  - http://slides.html5rocks.com
- HTML Validation
  - http://validator.w3.org/

HTML Fundamentals

- **Example:** BasicHTMLSkeleton.html
- Html documents rely on tags
  - Specifies a command or directive
  - It surrounds content and apply meaning to that content
  - <title>This is the document title</title>
  - General format<elementName attributes>
  - Most HTML elements have two tags (start tag and end tag)
  - Example: <h1> text </h1>
- An html document has two main parts
  - **Header** → provides information about the document
  - It does not generate displayed contents
  - **Body** → contents of the page
- Let's validate our document
- Let's introduce an error and validate

Title Tag

- <title> </title>
  - Part of the header
  - It is required
  - Search engines depend on it
  - Provide a meaningful name as it is bookmarked

Attributes

- An attribute extends or modifies a tag
- Attributes
  - Only appear in the start tag
  - You can have several attributes in one tag each separated by spaces
  - Order is immaterial
  - Some take values which are specified after an =
- General format
  - <ELEM ATTR="attrValue">Displayed Text</ELEM>
- Example

```
<img src="bear.gif" width="100" height="75" alt="bear image"  />
```

Nested Tags and Spaces

- Nested tags are possible but don't overlap sets of them
- Incorrect → Message
- Correct → Message
- Browser Processing
  - Multiple spaces are converted to one space.
  - John Mary Peter
  - John Mary Peter
- Line returns are ignored

Comments

- Comments
  - Represented by <!-- --> Note: (two sets of double -)
  - We can use comments to disable HTML

Frequently Used Tags

- Emphasis
  - text here **Text usually rendered in italics**
  - text here **Text usually rendered in bold**
- Super/Sub script
  - _{text here}
  - ^{text here}
- Quotations
  - <q> quote here </q>
- Line Breaks
  -
- Verbatim (text displayed exactly as it appears)
  - <pre> text here </pre>
  - You can have tags inside them
- **Example:** CommonTags.html

HTML Lists

- Unordered lists
  -   identify the list
  -   each list item
  - **Example:** Lists.html
- Ordered lists
  - identify the list
  -   each list item
  - **Example:** Lists.html
- Lists can be nested
  - Notice where the text inside of main list appears
  - **Example:** Lists.html

HTML Image Inclusion

- <img> tag used for image inclusion
- Attribute
  - Modify/extends a tag
  - Only appear in the start tag
  - Order is immaterial
  - General Format
    - <ELEM ATTR="attrValue">
- <img> attributes
  - src
  - width
  - height
  - alt
- **Example:** Image.html

HTML Links

- Link → Connection between web resources
- Links are created using the<a> tag
  - CNN Page
  - Need to specify http://
  - **Example:** Links.html
- You can also link a page with another as follows:
  - <a href="page.html">
- We can replace the text with an image
  - **Example:** Links.html

CHARACTER ENTITY REFERENCES

- Special Characters (Character Entity References)
  - Copyright → ©
  - Registered Trademark →®
  - & → &
  - < → <
  - > → >
  - Non break space →
    - Allow us to add single spaces
- List:
  - http://dev.w3.org/html5/html-author/charref
- **Example:** CharacterReferences.html

Suggestions for Writing HTML Code

- Add the corresponding end tag immediately
- Use indentation
- Have a consistent style
- Use comments to separate sections of your code
- Validate your code as you develop it (not at the end)