

CS-081 Course Outline

1. The Internet
 - 1.1. Clients and Servers; Navigation and Negotiation
 - 1.1.1. Requests: messages from client to server
 - 1.1.2. Responses: messages from server to client
 - 1.2. Internet Protocol (IP) Packets
 - 1.2.1. Header Section
 - 1.2.1.1. Sender and Recipient IP Addresses and Ports
 - 1.2.1.1.1. Domain Name Service (DNS)
 - 1.2.2. Body (Data) Section
 - 1.3. Internet Protocols use IP Packet Bodies
 - 1.3.1. HTTP (HyperText Transfer Protocol)
 - 1.3.2. SMTP (Simple Mail Transfer Protocol)
 - 1.3.3. SFTP (SSH File Transfer Protocol: 2001)
 - 1.3.4. SFTP (Simple File Transfer Protocol: 1984)
2. HTTP: The World Wide Web
 - 2.1. Web Servers are HTTP Servers
 - 2.1.1. Web servers need an operating system
 - 2.1.1.1. Internet packet transmission and receipt
 - 2.1.1.2. File system management
 - 2.1.1.3. Database access
 - 2.1.2. Web servers must have static IP addresses
 - 2.1.3. HTTP server program “listens” for client requests
 - 2.1.3.1. Port number 80 by default
 - 2.1.4. Many HTTP server programs available
 - 2.1.4.1. Apache (any operating system)
 - 2.1.4.2. IIS (Microsoft only)
 - 2.1.4.3. Mongrel ...
 - 2.1.5. May use programs (scripts) to help generate responses
 - 2.1.5.1. PHP, Java, C, C++, C#, Ruby, Python, ...
 - 2.2. Web Browsers are HTTP Clients
 - 2.2.1. Possible to run on the same computer as a web server (our lab).
 - 2.2.2. Need an operating system
 - 2.2.2.1. Internet packet transmission and receipt
 - 2.2.2.2. Low-level user interface management
 - 2.2.2.2.1. Keyboard, Mouse, Screen
 - 2.2.2.2.2. Mobile device equivalents
 - 2.2.2.2.3. Assistive device equivalents
 - 2.2.2.3. Persistent storage of server data (cookies)
 - 2.2.3. May use dynamic IP addresses and arbitrary port numbers
 - 2.2.4. Browser generates HTTP requests in response to user actions
 - 2.2.5. Browser processes server response messages
 - 2.2.5.1. Generate additional requests if response contains links
 - 2.2.5.1.1. Stylesheets
 - 2.2.5.1.2. Script Files

- 2.2.5.1.3. Image Files
 - 2.2.5.2. Construct in-memory representation of entire document
 - 2.2.5.2.1. (See [The Document Object Model](#) below.)
 - 2.2.5.3. Render the web page on the user's screen
 - 2.2.5.3.1. *Render*: to convert text and graphics into a rectangular matrix of colored pixels. (See [The CSS Box Model](#) below.)
 - 2.2.5.3.2. *Viewport*: the portion of the browser window where the rendered page is displayed.
 - 2.2.5.4. Respond to user actions (keyboard/mouse)
- 2.3. Universal Resource Locators (URLs)
 - 2.3.1. Terminology: URIs, URLs, and URNs
 - 2.3.2. URI/URL components
 - 2.3.2.1. Protocol Name
 - 2.3.2.2. Host Name
 - 2.3.2.3. Port Number
 - 2.3.2.4. Path (See [Computer File Systems](#) below)
 - 2.3.2.5. Form Data (See [Forms and Form Processing](#) below)
- 2.4. HTTP Messages
 - 2.4.1. HTTP message headers and bodies are transmitted inside IP packet bodies
 - 2.4.2. HTTP Request Headers
 - 2.4.2.1. User-Agent
 - 2.4.2.2. Accept (MIME types)
 - 2.4.2.3. Accept-Charset
 - 2.4.2.4. Cache-Control
 - 2.4.3. HTTP Request Bodies
 - 2.4.3.1. Form Data (alternative to URL form data)
 - 2.4.3.2. File uploads
 - 2.4.4. HTTP Response Headers
 - 2.4.4.1. Response-Code
 - 2.4.4.2. Server
 - 2.4.4.3. Content-Type (MIME type)
 - 2.4.4.4. Content-Length
 - 2.4.4.5. Last-Modified
 - 2.4.4.6. Cache-Control
 - 2.4.5. HTTP Response Bodies ("Web Pages")
 - 2.4.5.1. XML declaration, if XHTML
 - 2.4.5.1.1. If present, tells the character encoding
 - 2.4.5.2. DOCTYPE: tells which version of (X)HTML is used
 - 2.4.5.3. HTML or XHTML Document
- 2.5. Content Markup Languages
 - 2.5.1. SGML
 - 2.5.2. HTML
 - 2.5.3. XML
 - 2.5.4. XHTML
- 2.6. Stylesheets

- 2.6.1. CSS 2.1
 - 2.6.2. CSS 3
 - 2.6.3. Browser Compatibility
- 2.7. Scripting
 - 2.7.1. Client Side
 - 2.7.1.1. JavaScript
 - 2.7.1.2. Flash, Silverlight ...
 - 2.7.2. Server Side
 - 2.7.2.1. PHP
 - 2.7.2.2. Ruby, Java, Python, C, C++, C#, Objective C ...
- 3. Computer File Systems
 - 3.1. Files and Documents
 - 3.2. Directories and Folders
 - 3.3. File and Directory Names
 - 3.3.1. Case Sensitivity
 - 3.3.1.1. Sensitive, Preserving, Insensitive
 - 3.3.2. Embedded Spaces
 - 3.3.3. Paths
 - 3.3.3.1. Absolute
 - 3.3.3.2. Relative
 - 3.3.3.3. Separators: forward and backward slashes, / and \
 - 3.3.3.4. Special Directory Names: . and ..
 - 3.4. Tree Terminology
 - 3.4.1. Root
 - 3.4.2. Parent
 - 3.4.3. Child
 - 3.4.4. Sibling
 - 3.4.5. Descendant
 - 3.4.6. Ancestor
- 4. XHTML: XML representation of HTML
 - 4.1. Syntax
 - 4.1.1. Case sensitivity
 - 4.1.2. Elements
 - 4.1.2.1. Attributes
 - 4.1.2.2. Contents
 - 4.1.3. See [DOM Tree Structure](#) below for required and optional elements
 - 4.2. Tags
 - 4.2.1. open, close
 - 4.2.2. self-closing
 - 4.3. Text
 - 4.3.1. Character Encoding
 - 4.3.1.1. ASCII, ISO-Latin-1, UTF-8
 - 4.3.2. Character Entities
 - 4.3.2.1. Named
 - 4.3.2.2. Numbered
- 5. The Document Object Model (DOM)

5.1. DOM Tree Structure

5.1.1. Same terminology as file systems, but totally different context

5.1.2. *html*: the root of the tree

5.1.3. *head* and *body*: required, and only, child nodes of *html*

5.1.4. *title*: required child node of *html*

5.1.5. Children of *head*

5.1.5.1. *title* (required)

5.1.5.2. *link* for stylesheets, shortcut icon (favicon), ...

5.1.5.3. *script* for JavaScript files

5.1.5.4. Legal, but avoid using:

5.1.5.4.1. *style* for embedded CSS rules

5.1.5.4.2. *script* for embedded JavaScript

5.1.6. Children of *body*

5.1.6.1. Text-related: headings and paragraphs

5.1.6.2. Hypertext anchors

5.1.6.3. List-related

5.1.6.4. Table-related

5.1.6.5. Images

5.1.6.6. Form-related

6. Text

6.1. Content Structure (XHTML Elements)

6.1.1. Headings: *h1*, *h2*, *h3*, *h4*, *h5*, *h6* elements

6.1.2. Paragraphs: the *p* element

6.1.3. Lists

6.1.3.1. Unordered, ordered, and definition

6.1.3.2. List items: the *li* element

6.1.3.3. Definition terms and definitions

6.1.4. Tables

6.2. Appearance (CSS Properties)

6.2.1. Fonts

6.2.1.1. Family, Style, Size

6.2.1.2. Weight, Variant

6.2.2. Line and character spacing

6.2.3. Spans

7. Cascading Style Sheets (CSS)

7.1.1. Rules

7.1.1.1. Selector

7.1.1.1.1. Which elements of DOM?

7.1.1.1.2. Selector syntax

7.1.1.1.2.1. tag name, id, class

7.1.1.1.2.2. position in DOM tree

7.1.1.1.3. Specificity

7.1.1.1.4. Dynamic conditions (hover, ...)

7.1.1.2. Property List Items

7.1.1.2.1. Property Name

7.1.1.2.2. Property Value

- 8. Colors
 - 8.1. The RGB model
 - 8.2. Representing colors in CSS
- 9. Image Files
 - 9.1. Managing File Size
 - 9.1.1. Compression
 - 9.1.2. Number of Colors
 - 9.2. Transparency
 - 9.3. Using Photoshop to prepare image files
- 10. The CSS Box Model
 - 10.1. Measurement Units
 - 10.1.1. EMs, percentages, ratios
 - 10.1.2. Pixels
 - 10.2. Background
 - 10.2.1. Color, Image, Image position, image repetition
 - 10.3. Areas: Margin, Border, Padding, Content
 - 10.3.1. Width and Height apply to Content
 - 10.3.2. Visible size includes content, padding, and border
 - 10.4. Visibility: visible, hidden, or none
- 11. Layout
 - 11.1. Relative, absolute, fixed positioning
 - 11.2. Float and Clear
 - 11.3. HTML 5 elements: header, navigation, section, article, figure, dialog, aside, footer
- 12. Forms and Form Processing
 - 12.1. Form Elements
 - 12.1.1. *method* and *action* attributes
 - 12.1.2. *fieldset* and *legend* elements
 - 12.1.3. *label* and its *for* attribute
 - 12.1.4. *input* elements
 - 12.1.4.1. *text*, *password*, *radio*, *checkbox*, *hidden*, *submit*, *reset*
 - 12.1.4.1.1. *name* attribute
 - 12.1.4.1.2. *value* attribute (not for *text* and *password*)
 - 12.1.4.1.3. radio groups
 - 12.1.5. *textarea* element
 - 12.1.6. *select* element
 - 12.1.6.1. *option*
 - 12.1.6.2. *optgroup*
 - 12.1.7. *button* element
 - 12.2. Form Processing
 - 12.2.1. Optional JavaScript form data validation
 - 12.2.2. Form data encoding
 - 12.2.3. Server script