Web Architecture

Slides adopted from Marty Stepp
http://www.cs.washington.edu/education/courses/
cse190m/07sp/lectures/slides/02-html.html
- a simple protocol for attempting to send data between two computers
  - each device has a 32-bit IP address (64 bits in IPv6)
  - addresses written as four 8-bit numbers (between 0 and 255)
    e.g. 145.10.34.3

- find out your internet IP address: whatismyip.com
- find out your local IP address in a terminal, type: `ipconfig` (Windows) or `ifconfig` (Mac/Linux)
Transmission Control Protocol (TCP)

- adds multiplexing, guaranteed message delivery on top of IP
- multiplexing: multiple programs using the same IP address
  - port: a number given to each program or service
  - port 80: web browser
  - port 25: email
  - port 22: ssh

- some programs (games, streaming media programs) use UDP protocol instead of TCP
Domain Name System (DNS)

- a set of servers that map written names to IP addresses
- `nslookup west.uni-koblenz.de`
  - 141.26.64.113

- many systems maintain a local cache called a hosts file
  - Windows: C:\Windows\system32\drivers\etc\hosts
  - Mac: /private/etc/hosts
  - Linux: /etc/hosts
Organizations

- World Wide Web Consortium (W3C): web standards
  - Working Draft (WD)
  - Candidate Recommendation (CR)
  - Proposed Recommendation (PR)
  - W3C Recommendation (REC)

- Internet Engineering Task Force (IETF): internet protocol standards

- Internet Corporation for Assigned Names and Numbers (ICANN): decides top-level domain names
Web Server

- a computer running web server software that listens for web page requests on TCP port 80

- popular web server software:
  - Apache: www.apache.org
  - Microsoft Internet Information Server (IIS)
    - available as part of Windows XP (directions)
Web Browser

- a software application that displays web pages

- popular web browser software:
  - Mozilla Firefox: getfirefox.com
  - Microsoft Internet Explorer (IE): part of Windows
  - Apple Safari: part of Mac OS X
  - Opera: opera.com
  - Google Chrome
Hypertext Transport Protocol (HTTP)

- the set of commands understood by a web server and sent from a browser
- some HTTP commands (your browser sends these internally):
  - GET *filename* : download
  - POST *filename* : send a web form response
  - PUT *filename* : upload
- simulating a browser with a terminal window:
  
  ```
  $ telnet www.cs.washington.edu 80
  Trying 128.208.3.88...
  Connected to 128.208.3.88 (128.208.3.88).
  Escape character is '^]'.
  
  GET /index.html
  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 ...">
  <html>
  ```
the web server returns a special "error code" number to the browser, possibly followed by an HTML document

common error codes:

<table>
<thead>
<tr>
<th>Number</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>OK</td>
</tr>
<tr>
<td>301-303</td>
<td>page has moved (permanently or temporarily)</td>
</tr>
<tr>
<td>403</td>
<td>you are forbidden to access this page</td>
</tr>
<tr>
<td>404</td>
<td>page not found</td>
</tr>
<tr>
<td>500</td>
<td>internal server error</td>
</tr>
</tbody>
</table>

complete list of error codes:
http://en.wikipedia.org/wiki/Http_error_codes
Uniform Resource Locator (URL)

- an identifier for the location of a document on a web site

- a basic URL:
  
  http://www.aw-bc.com/info/regesstepp/index.html

- upon entering this URL into the browser, it would:
  1. ask the DNS server for the IP address of www.aw-bc.com
  2. connect to that IP address at port 80
  3. ask the server to GET /info/regesstepp/index.html
  4. display the resulting page on the screen
More advanced URLs

- anchor: jumps to a given section of a web page
  
  http://www.textpad.com/download/index.html#downloads
  
  - the above URL fetches index.html and then jumps downward to a part of the page labeled downloads

- port: for web servers on ports other than the default 80
  
  http://www.cs.washington.edu:8080/secret/money.txt

- query string: a set of parameters passed to a web program
  
  http://www.google.com/search?q=miserable+failure&start=10
  
  - the above URL asks the server at www.google.com to run the program named search and pass it two parameters:
    - q (set to "miserable+failure")
    - start (set to 10)
Some Web programming technologies

- Hypertext Markup Language (HTML/XHTML): used for writing web pages
- Cascading Style Sheets (CSS): supplies stylistic info to web pages
- Javascript: allows interactive and programmable web pages
- Asynchronous Javascript and XML (AJAX): allows fetching of web documents in the background for enhanced web interaction
- PHP Hypertext Processor (PHP): allows the web server to create pages dynamically
- Structured Query Language (SQL): interaction with databases
BRIEF OVERVIEW OF HTML
Hypertext Markup Language (HTML)

- describes the content and structure of information on a web page
  - not the same as the presentation (appearance on screen)
- surrounds text content with opening and closing tags
- each tag's name is called an element
  - syntax: `<element> content </element>`
  - example: `<p>This is a paragraph</p>`

- most whitespace is insignificant in HTML
  (it gets ignored or collapsed into a single space)
More about HTML tags

- some tags can contain additional information called attributes
  - syntax:
    
    ```html
    <element attribute="value" attribute="value"> content </element>
    ```
  - example:
    
    ```html
    <a href="page2.html">Next page</a>
    ```

- some tags don't contain content; can be opened and closed in one tag
  - syntax:
    
    ```html
    <element attribute="value" attribute="value" />
    ```
  - example:
    
    ```html
    <img src="bunny.jpg" alt="A bunny" />
    ```
  - example: `<hr />`
Structure of an HTML page

- a header describes the page and a body contains the page's contents
- an HTML page is usually saved into a file ending with extension .html
Welcome to my first web page!

This is a paragraph of text. I am very proud of it.
<html>
<head>
<title>My first web page</title>
</head>
<body>
<h1>Welcome to my first web page!</h1>
<p>This is a paragraph of text. I am very proud of it.</p>
</body>
</html>
SGML, XML, HTML, XHTML

SGML

„subset of“

XML

application of

HTML 4

same expressiveness as

XHTML 1.0

application of
XHTML 1.0

- A reformulation of HTML 4 in XML 1.0
  - [http://www.w3.org/TR/xhtml1/](http://www.w3.org/TR/xhtml1/)
  - [http://www.w3.org/XML/](http://www.w3.org/XML/)

- why use XHTML and web standards?
  - more rigid and structured language
  - more interoperable across different web browsers
  - more likely that our pages will display correctly in the future
  - can be interchanged with other XML data: SVG (graphics), MathML, MusicML, etc.

- a strict XHTML page uses some slightly different syntax and tags
Structure of an XHTML page

```xml
<!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">

<head>
<title>information about the page</title>
</head>

<body>
<p>some paragraph content</p>
</body>

</html>
```

XML (and therefore XHTML) requires
• nesting of tags
• Every opening tag must be closed
<h1>University of Whoville</h1>
<h2>Department of Computer Science</h2>
<h3>Sponsored by Microsoft</h3>
<p>Here comes some paragraph text...</p>
How headings and paragraphs are displayed is influenced by:

- style sheets
- preferences indicated in the browser
- display size
- fonts available on the computer
**Linking with anchors**

- types of URLs that can appear in anchors:
  - absolute (to another web site)
  - relative (to another page on this web site)
- specify a tooltip with the title attribute
Images

- the src attribute specifies the image URL
- XHTML also requires an alt attribute describing the image
- title specifies tooltip
- anchor makes the image become a link

```html
<p><a href="http://thereyougo.com">!
<img src="gollum.jpg" alt="Gollum from LOTR" title="Beware!"/></a></p>
```
<ul>
<li>No shoes</li>
<li>No shirt</li>
<li>No problem!</li>
</ul>

- No shoes
- No shirt
- No problem!

ul stands for unordered (bulleted) list
li stands for list item
Nesting lists, ordered lists

<ol>
  <li>Simpsons:
      <ul>
        <li>Bart</li>
        <li>Lisa</li>
      </ul>
  </li>
  <li>Family Guy:
      <ul>
        <li>Peter</li>
        <li>Lois</li>
      </ul>
  </li>
</ol>

1. Simpsons:
   - Bart
   - Lisa

2. Family Guy:
   - Peter
   - Lois

ol stands for ordered (numbered) list
Lists can be nested
Further block elements

- definition lists (dl, dt, dd)
- Quotes (q), blockquotes (blockquote)
- preformatted text, keeping whitespace (pre)

- tables
Phrase elements

**em**: emphasized text (usually rendered in italic)

**strong**: strongly emphasized text (usually rendered in bold)

**code**: a short section of computer code (usually rendered in a fixed-width font)

```html
<p>The <code>ul</code> and <code>ol</code> tags make lists.</p>

<p>HTML is <em>really</em>, <strong>REALLY</strong> fun!</p>
```

The **ul** and **ol** tags make lists.

**HTML** is *really*, **REALLY** fun!
Further phrase elements

- Text deletions
- Text insertions
- Abbreviations
W3C XHTML Validator

- validator.w3.org
- checks your XHTML code to make sure it meets the official strict XHTML specifications
- more picky than the web browser, which may render malformed XHTML correctly
Web page metadata

```html
<meta name="description" content="Authors' web site for Building Java Programs." />

<meta name="keywords" content="java, textbook" />

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
```

- placed in the head of your HTML page
- meta tags often have both the name and content attributes
Web page metadata

```html
<head>
<meta name="author" content="web page's author" />
<meta name="revised" content="web page version and/or last modification date" />
<meta name="generator" content="the software used to create the page" />
</head>
```
Web page metadata

```html
<head>
<meta http-equiv="refresh" content="30"/>
<meta http-equiv="content-type" content="text/html; charset=UTF-8"/>
<meta http-equiv="default-style" content="id_of_link_element"/>
</head>
```

- Preferred style sheet
- Character encoding
- Avoid "refresh"
  - takes away control from the user,
  - Against W3C web content accessibility guidelines
    - http://www.w3.org/WAI/intro/wcag.php
Some rules for HTML5 were established:

- New features should be based on HTML, CSS, DOM, and JavaScript
- Reduce the need for external plugins (like Flash)
- Better error handling
- More markup to replace scripting
- HTML5 should be device independent

Current status:

- Working draft: http://www.w3.org/TR/2011/WD-html5-20110525/
Some of the most interesting new features in HTML5:

- Drag and drop
- The `<canvas>` element for 2D drawing
  - Inline SVG (scalable vector graphics)
- The `<video>` and `<audio>` elements for media playback
- Support for local storage at Web browser (replacing cookies)
- New content-specific elements, like `<article>`, `<footer>`, `<header>`, `<nav>`, `<section>`
- New input types for forms like calendar, date, time, email, url, search
- Support for geolocation of user
Misnomer: „Semantic tags“

- Not about „meaning“ but about structure
- They generally have no default outward appearance on the page, instead they give insight into the structure of the page.
  - section
  - header
  - footer
  - nav
  - aside
  - article
  - More.....
STYLE SHEETS
The bad way to produce styles

Welcome to Greasy Joe's. You will never, ever, EVER beat OUR prices!

tag such as b, i, u, and font are discouraged in strict HTML

Why is this bad?
Cascading style sheets

- **CSS** describes the appearance and layout of information on a web page
  - (as opposed to HTML, which describes the content of the page)
- can be embedded in HTML or placed into separate .css file (preferred)

```html
<head>
  ...
  <link href="style.css" type="text/css" rel="stylesheet" />
  ...
</head>
```
Basic CSS rule syntax

```css
selector {
    property: value;
    property: value;
    ...
    property: value;
}
```

```css
p {
    font-family: sans-serif;
    color: red;
}
```

- a CSS file consists of one or more **rules**
- a rule's **selector** specifies HTML element(s) and applies style **properties**
  - a selector of * selects all elements
CSS properties for colors

```css
p {
    color: red;
    background-color: yellow;
}
```

This paragraph uses the style above.

<table>
<thead>
<tr>
<th>property</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>color</td>
<td>color of the element’s text</td>
</tr>
<tr>
<td>background-color</td>
<td>color that will appear behind the element</td>
</tr>
</tbody>
</table>
Cascading Style Sheets (CSS)

- Affecting
  - Fonts
  - Font size (absolute, relative)
  - Font weight
  - Font style
  - Colors
  - Background
  - Text alignment
  - Text decoration
  - List styles
  - Tables

- Classes
Cascading style sheets

- It's called Cascading Style Sheets because the properties of an element cascade together in this order:
  - Browser's default styles (reference)
  - External style sheet files (in a `<link>` tag)
  - Internal style sheets (in a `<style>` tag in the page header)
  - Inline style (the style attribute of an HTML element)
Content vs. presentation

- HTML is for content; what is on the page (heading; list; code; etc.)
- CSS is for presentation; how to display the page (bold; centered; 20px margin; etc.)
- Keeping content separate from presentation is a very important web design principle
  - Actually also for writing books, theses
- If the HTML contains no styles, its entire appearance can be changed by swapping .css files

What is the disadvantage of style sheets?