**1/15**

Need 3 things to create a website:

-Domain name

-Web server

-Web pages (files and database)

**1/20**

Web != Internet

FTP = File Transfer Protocol

Tim Berners-Lee: created the world wide web

Html = hypertext markup language

Reasons for Internet growth in the 90’s:

-Lifted ban on commercial activity

-Tim Berners Lee created the www, html

-Creation of Mosaic, the first web browser

-PC’s got cheaper so more people bought them

-Low cost connections to the Internet

W3C

-Took over management of web specifications (html tags)

ICANN:

-Took over management of the Internet: servers, IP addresses, protocols

Client/Server model:

-The network model that the Web uses

-The client software is the browser

-Server is another computer running other software (could be your own computer or any computer)

-When you look at a web page, you aren’t look at the server’s file. A copy of the file has been sent to you on your web client.

Language between browser and server:

HTTP: hypertext transfer protocol

-IE is the worst browser: Microsoft tried to make changes, and they made mistakes and stuff. Web sites look different on different browsers.

Protocols:

-Language between clients and servers

-HTTP: Can I have a file? Here’s the file.

-FTP: Not only sends files, but can write files

URI: Uniform resource identifier:

-A URL is a type of URI

-A URI is a recipe, and the URL is the recipe for apple pie

TLD: top level domain:

-.com, .net, .org, .gov, .travel, .uk, .jp

**1/22**

-All file and folder names have to be lowercase with no spaces

-Some web servers are case sensitive and some are not, so we just agree not do this.

-Doc type tells the browser what version of html the file is using

-No error messages in the web page

**2/5**

float: left or right: gets text to wrap around pictures

good css website: css3generator.com

**2/11**

-Natural/normal document flow is a single column layout

**-5 techniques for layouts (other than the natural layout)**

*-Html Table* // don’t use

*-Float (css)* // Not preferred, but commonly used in old websites

-Both elements are floating left (stacked up against each other)

-Need to “clear” afterwards. Otherwise, the footer will wrap around

-syntax for clearing: footer{clear: left;} // clears whatever was floated left

*-Inline-block (CSS)*

-Properties of both block and inline

-Still a block, but only as wide as it needs to be

-Tell both elements to “display: inline-block;”

-Assign a width for both elements (need account for padding so it doesn’t exceed 100% width

-Vertically aligned to the bottom as a default

-To vertically align it to the top: “vertical-align: top;”

Problem with inline-block: can’t get background color of things to fill up all of the space.

-Also, inline-block accounts for whitespace in between the two elements

*-CSS Table-cell // works great, but scares people off because it’s called “table”*

-Backgrounds of different elements go all the way to the bottom

-Can’t use margins on table-cell

-Syntax: article, aside: {display: table-cell;}

-Can override default widths if you want

*-Flex // awesome*

-Big problem: It’s so new that older browsers do not recognize it. So it works great, but be aware it might not work in older browsers

-wraps around a bunch of elements: “section {display: flex;}”

-css-tricks.com // good website to learn about flex

Use website: caniuse.com // tells you what browsers can support which properties

**CSS Display:**

-Block, inline inline-block, table-cell, flex

**2/17**

-Border collapse: makes it so double bordering doesn’t happen

-Doesn’t work for flex

- Term of website: That term didn’t exist when all the HTML/CSS technology didn’t exist

- The whole idea of a unified design didn’t exist. There was no concept of that.

Nav Bar:

-use unordered list, and apply css to enhance it

1st thing you have to do: undo default styles for unordered lists

-4 main selector types

**2/19**

-Absolute: doesn’t do anything by itself: only see result other properties (top, right, left, bottom)

-4 positions to know:

-static (default position) (normal document flow)

-relative: changes locations of elements in relation to other elements, for where it would show up in the normal flow

-2 uses for relative positioning: 1: change location of something (and then nudge it with top,bottom, left, right)

2: don’t use nudging. Just use relative positioning, so that you can use absolute positioned items inside of it, and those will be constrained to that relative positioning.

-absolute: changes locations outside of the normal flow

-fixed: disregards scrolling. Nails itself down to viewport of browser. For example, it could stay at the top of the browser.

JavaScript:

-smooth scrolling between links in a website takes the same 3 lines

**2/19**

-Having link open in same browser: // for your own pages

<a href = “whatever.html”> Opens Same Page</a>

-Having link open in new tab: // for a separate website

<a href = “google.com” target=”\_blank”>Opens in a new Page</a>

How to create consistent header and footer on large website:

-W3c gave us nothing for this

-Other people came up with “server side includes” (SSI) which allows us to do it

-Can do SSI with many languages: example: PHP (use .php extension)

**Midterm Notes:**

Sections:

-History of the internet and the web

-HTML

-CSS Formating

-CSS layout

-CSS positioning

-CSS navigation

-Server side includes

2 types of development:

-Front end: html, css, javascript

-Back end: what you do on the web: application programming, databases on the server: php, asp, ruby, python

ICANN:

-Manage domain names and numbers

-Normal people work through third parties, who work with ICAN to register a domain name for us

W3C

-Worldwide web consortium

-Organization for web standards, took over that job from Tim Burners Lee

-Writes specifications for HTML, CSS

-Their latest standards: HTML5, CSS3

Web browsers:

-Can vary, can have bugs: some websites look different on different browsers

Web servers:  
-Computer with Internet connection, and running web server software

Internet != web:

-Internet: worldwide network of interconnected computers and related equipment (routers, wiring)

-WWW: Internet application

Other Internet applications: email, FTP, gofer

Gofer: similar to the web except it didn’t have hyper text

Tim burners Lee: Invented the Web

-Added hyperlinks

-Reason for invented the web: help scientists exchange data and research, and link them together

-Brought these together: Internet, HTML (based on SGML), hypertext

Mark Andreeson

-Invented first web browser: mosaic, later became Firefox

Growth of Internet:

-Removal of commercial ban

-Cheap PC’s

-Web

Network:

-2 or more computers connected for the purpose of sharing resources

-Internet is a network

Network model of the Internet:

-Client server

-Client server software: web browser

Protocols:

-Methods for clients and servers to communicate over the network

-FTP, HTTP, TCPIP

TCPIP: TCP is envelop to put data, IP is address on envelop of where it goes

Domain name:

-Name converted to IP address by DNS server

URL

-Type of URI

Progressive enhancement:

-Each layer adds value

-Structure, presentation, behavior (JavaScript or PHP)

HTML:  
-Descendent of SGML

XML:

-Extensible markup language

-Intended to transport structured data over the web

XHTML:

-Extensible hypertext markup language

-Was intended to combine the strengths of xml and html

HTML5:  
-Combination of HTML4 and XHTML1.1

-Got rid of craziness

-Adds structure and meaning (Not format or position)

DTD

-Doctype statement

-Identifies HTML version to the web browser

-EX for HTML5L <!doctype html>

Requirements for HTML5:

-<html>, <head>, <body>

<head>

-Meta information: information about document

<body>

-Displayed in browsers viewport

Heading elements:

-Intended to create document outline

-Must be in order

-Lower level headings are used to indicate a subordinate to a higher level heading above it

Semantic HTML:

-The use of HTML markup to reinforce the meaning of the content of the document

-Correct use

<br>

-Street address

-Poetry

-NOT to push content down the webpage (use CSS)

<p>

-Only use it for a paragraph

<ol>

-Only for ordered data

-Only <li> allowed inside

-Can do anything you want within <li>

Nesting

-Reverse order when closing tags

Two kinds of HTML elements:

-Normal tags, and standalone tags that you don’t close (such as <img>)

<div>, <span>

-Doesn’t add or diminish to the semantics of the document

-Difference: <div> is block element, <span> is an inline element

Block element:

-Displays in web browser as wide as it can be.

-Stack up top over bottom

Inline element:

-Only as wide as element needs it to be

-Stacks up side by side

HTML structural elements:

-Header, nav, section, main, article, aside, footer

-Identify groups of content, have the purpose of their names

<article>

-A complete self contained composition of content

-Can pull out of HTML page and would still make sense by itself

<aside>

-Represents content that is tangentially related to the content around or near it

-Supporting information

-If you pulled aside out of HTML document, it would look random and incomplete

Attributes:

-Inserted into the opening part of the tag

-Some attributes are mandatory (img requires src and alt)(a requires href)

Hypertext in HTML:

-<a href=”yo”>stuff</a>

Parent link:

../filename.html

Child link:

childFolder/filename.html

Sibling link:

../siblingFolder/filename.html

Syntax Errors in HTML

-W3C’s validator

-Work on first error (can fix other errors)

<b> tags: bad (it’s old and replaced)

**CSS**

Style sheet:

-W3C borrowed the idea from media industry

External style sheet:

-Can link to multiple HTML files

Declarations

-Property: what is to be changed (color)

- Colon, and then the value (what you want it to be changed to (red))

Common properties:

-font-weight, font-style, font-size, font-family, color

Font-stack:

-Browser will try the list in order until it finds one that works

Colors:

-Can use keywords (red, blue)

-RGB hex number (FFF000) (16 million options)

-RGB (255,255,255)

3 Places to write CSS:

External style sheets: separate text file, use link tag to connect it to the file

-Can link same style sheet to multiple documents

-Syntax: <link rel=”sylesheet” href=”whatever.css”>

Embedded styles

-CSS files nested in the head of the HTML document

-Only works on that single HTML document (not multiple)

Inline styles

-Use attribute on an HTML document

-Syntax: <p style=”color: red;”>stuff</p>

CSS Overrides: Which type of CSS holds presence

1. HTML attributes styles

2. Inline

3. Embedded styles

4. Embedded styles

5. Built in browser styles

Shortcuts:

-margin: 10px 0 0 0; Top, right bottom, left

-margin: 30px 40px; Top and bottom, right and left

-border: 1px solid red; width, style (could be dotted, dashed), color

Box Model:

-For block element (such as p)

-How we think of these properties working together: margin, padding, border

Float:

-If float element the right, the surrounding content will go the left

-Clear a float: Too much content wraps around an element: 2 ways to do it

-Apply the clear property to the next element after the content

-Apply overflow property to wrapped elements

-There’s a float right, left but NOT center

To create an artificial float center:

-margin: 0 auto;

-width: 80%

CSS Selectors (6 types):

Element selector:

-body{}

-heavy handed, does it to all of them

Class selector:

.class{}

-Can use multiple times in document

-Use classes most often

ID selector:  
 -#id{}

-Can only use once in document

Descendent selector:

-nav li{}

-Only apply li elements that are in the nav

Pseudo selectors:  
 -a:hover{}

Grouped selectors:

-article, aside {}

-Apply selectors to multiple elements at the same time

Normal document flow: (normal document flow)

-However you type it in HTML, that’s how it appears

display (5 options)

-inline, block, inline-block, table-cell, flex

Inline vs block

-Each element is default inline or block

-Headers are blocks

-a, strong, img are inline

display: nline-block:

-Box model attributes, but doesn’t extend 100% of width like a block would do

display: table-cell:

-Neighboring block elements line up side by side with equal height

display: flex

-Neighboring block elements line up side by side with equal height

-Older browsers don’t recognize it

5 Techniques of layout: In order of age

-HTML Table

-Float layout

-Inline-block, table-cell, flex

CSS Positioning:

-Syntax: position: static, fixed, relative, absolute

static: default for all elements

-fixed: element will appear with no regards to scrolling

-relative: relative to normal document floor based on location properties (top, right, bottom, left) in relation to normal document flow

-absolute: same as relative but outside the normal document flow

Server Side includes:

-Reduce redundant code in HTML documents

-Can do this through PHP

-PHP runs on server, not client

-When testing, it won’t work until you put it on the web server

**3/5**

2 Types of image files:

-Bit map type: grid of small dots

-Vector type: (not going into vector types in this course)

SVG:

-Vector type

-Not photographs

-Code written to make picture

-Can generate this code using plugins

-Can zoom in and out and it maintains high quality (good for mobile design)

3 main file types: all bit map type files

-Gif, jpg, png

JPEG (or jpg):

-Can display 16 million colors

-Lossy compression: Can get pixelated after compressing

-No transparency

GIF:  
-256 Total colors

-Use for logo, not photograph

-Image is always a rectangle, even if it doesn’t look like it (can make background-color transparent)

-Lossless compression

-Can do animation

PNG:

-Everything a GIF can do, a PNG can do better except for animation

-16 million colors

-Lossless compression

-Not as compressed as JPG

-Can do transparency

IMG element:

-Src and alt attributes, but not height and width attributes

-Why shouldn’t you include height and width attributes?

-Large picture will still download on client, and then compress it. You want to compress it first and then download it to the client.

-Can also distort the picture depending on the browser

-Not mobile friendly. The size of the picture is locked to that size.

Using CSS for adding images:

Any images used to style the image should be in the CSS not HTML (for example used in the background)

To get to image from CSS file:

-body{background-image: url(../images/clouds.jpg);}

// This image will copy itself as many times as it needs to take up the space of the viewport

If you don’t want it to repeat: body {background-repeat: no-repeat;}

-Can also do repeat it y or x: background-repeat:repeat-y;

-background-size: cover; // distorts image to cover background

-background-size: contain// replicates itself rather than distorts image

Inspecting element on browser: (prof uses firebug)

-Hover on html, will show you the CSS for that element

-Can “change” the CSS to see what it does

-Not actually changing anything, just changing the downloaded file

-Can also debug JavaScript

Opacity

-Can give it opacity and it looks cool

-a:hover{opacity:.7;}

-background-color:rgba(255,255,255,.5) // white background with half opacity

**3/19/15**

-Good source for helping making web forms: wufoo

for HTML Forms:

-In real world would need a lot of CSS (won’t need for this class) to style the forms

radio:

-For when you’re looking for one answer

-ID’s have to be different for each option, but the “name” must be the same so that they all work together

**3/26/15**

-Can write JS in 3 places: embedded, events, separate file

-Order does matter for JS (read from top down)

**4/16/15**

**Search Engine Optimization**

-Search engine: software organizes it

-Search directory: organized by humans

Google (1998) was the first good search engine

-Changed the algorithm of previous search engines

-Made it actually good and fast