* History of the Web
  + To put a website on the internet you need 3 thing
    - A domain name
    - A web server
    - Files
  + 2 main parts in the web industry
    - Front end
      * What you see on the web
      * Uses
        + HTML
        + CSS
        + JavaScript
    - Back end
      * What you do on the web
        + Posting a comment on the facebook
        + Making an order online
      * Uses
        + PHP
        + .net
        + Python
  + ICANN manages all the domain names for the web
  + W3c Manages the web standards
    - They control HTML and CSS
    - Latest Versions
      * HTML 5
      * CSS 3
  + W3c Does not make browsers, companies do
  + Web server
    - It’s a computer
    - Has some sort of connection to the internet
    - Runs some sort of web server software
  + The internet is not the web
  + FTP: File Transfer Protocol
    - One of the oldest protocols still used on the internet
    - Main use to is transfer files
      * Getting a putting
    - FTP Software required on both the client and server computers
  + Tim Burners Lee: Father of the WEB (not the internet)
    - HTML is his invension of a document sharing system
      * Started as SGML
    - Unique aspect: **Hyperlinks**
    - Ideas to call his invention
      * Information mesh
      * Mine of information
    - What he put together
      * Sharing documents on the internet
      * Hyperlinks
      * His own markup language
  + Mark Andreeson
    - Created the first web browser: Mosaic
    - Has inline objects
    - Evolved into what we know as Firefox
  + Why is internet got so popular
    - Removal of the ban on commercial activity
    - Graphic based web browsers
    - Cheap PCs
    - Dial-up ISP
    - The invention of the web
  + Network
    - Two or more computers connected together to share resources
    - Network model: Client-Server
      * We are familiar with the Client side of the web
  + HTTP: Hyper Text Transfer Protocol
    - Needed on both the client and Server
    - Most common server HTTP Software
  + TCP IP: Transfer Connection Protocol / Internet Protocol
    - TCP: The envelope for your data
    - IP: Address for your data
  + DNS: Domain Name System
    - A Server that runs software that changes domain names to the actual IP address: Name to number
  + URL:
    - HTTP part: The Protocol
    - Domain Part: [www.example.net](http://www.example.net)
      * Sub Domain/ Machine name
      * Main name
      * Extension
    - Virtual Path
  + Progressive enhancement
    - A strategy for structured layered design
    - Adds a layer that improves the previous layer
      * Layer 1: Structure
        + Uses HTML
      * Layer2: Presentation layer
        + Uses CSS
      * Layer 3: behavior layer
        + Uses JavaScript
* Html: Hypertext Markup Language
  + Descendent of SGML
  + XMHL. Ect.
  + Every markup language has a DTD: Document Text Definition
    - First line of every document has a doctype statement
    - In HTML 5, we use: <!doctype HTML>
    - Structural elements
      * HTML Tag, needs
        + A head

Meta data

* + - * + A Body

Gets displayed on the browers

In the View Port

* + - Heading Elements
      * h1, h2, h3 ect,
      * Used to give order
      * Lower level headings are meant to show subordination to headings above it
    - The purpose of HTML is intended to add value to content
      * Paragraphs have words
    - Tags
      * <p> paragraph
      * <ol>, <ul>: Ordered List, Unordered
        + Needs to be used appropriately
      * Nesting Elements
        + Opening and closing tags must be **Closed in the REVERSE order they have been opened**
      * <div> and <span>
        + can be used without sematic effect
        + have no meaning to the html
        + <div> is a block element
        + <span> is an inline element
        + BLOCK ELEMENTS Stack top to bottom
        + INLINE go side by side
      * Structural Elements
        + Use them to add semantic value to the purpose
        + Header <header>
        + Nav <nav>
        + Structure
        + Main
        + Article
        + Aside
        + Footer
      * <img> Image
        + <img src=”” alt=””>: two attributes

scr holds the path to the image

alt holds to text about the image

If the image doesn’t load for any reason, the alt text will be displayed in the browser

For people who are visually impaired who used a screen reader, it will read what is in the alt attribute for the user

Search Engine Optimization

* + - * <article>
      * Type of links
        + Src=”this.html’: Link in same folder
        + Src=”folder/this.html”: child link
        + Src=”../this.html”: parent folder link
        + Src=”../folder/this.html”: Sibling link
      * When using the HTML w3c validator
        + Work on the first error only, then move to the next
* CSS: Cascading Style Sheet
  + General CSS questions
    - Predates the web
    - W3c Mantains the CSS Standards
    - Done in a separate file
  + Syntax
    - h1 (color: red;) : selector ( property: value; }
      * selector targets HTML
        + .**class**: can target classes too
        + can use multiple classes multiple times
        + **#id**: can target id’s

can only use id’s once in an html doc

* + - * declaration
        + property
        + value
        + ends with semicolon
      * Multiple declarations for a selector is possible
    - <link rel=”stylesheet” href=”mycss.css”>
      * In the header
    - This can be done in multiple places
      * Using <link> to a separate file
        + LETS YOU CASCADE
      * Using the HTML <style> tag
      * You can do it inline
        + Issue: only works in one file, no cascade
    - shortCuts
      * p {margin-top: 10px;  
        margin-right: 20px;  
        margin-bottom: 30px;
      * margin-left: 40px;}
      * OR
      * p {margin: 10px, 20px, 30px, 40px }
      * p (border: 1px solid blue;}
    - .stuff .next article p { color: green}
    - color the paragraphs in articles part of the next class under the stuff class, green
  + Layout Techniques
    - Table Layout: Do not use
      * Hard to maintain
      * Sematically void, Goggle will not line
    - Float: Do not use
      * Hard to maintain
    - Inline-block elements
      * All elements will lineup side by side
      * Comment out trick
    - Table –cell
      * Underutilized
      * Works with any neighbor elements
      * Problem: Doesn’t respect some of the aspects of the box model
    - Flex
      * Too new~?
      * Ignored by some old web browsers
      * Does have a lot of opportunity
  + Positioning
    - Selector { **position**: my thing; }
    - 4 types
      * static
      * default for all elements
      * fixed
        + top: #px; :will go to top and stay there
        + WILL NOT scroll with the rest of the doc
      * absolute
        + will not follow the natural document flow and will just do it’s own thing.
        + Ignores all other elements
        + WILL scroll with the document, can disappear away
      * relative
        + element will go wherever it’s supposed to go within the natural document flow, but will nudge something depending on what properties you give it
  + Navigation
    - Two questions
      * Where am I now?
      * Where can I go?
      * Bad navigation causes
        + High bounce rate: number 1 reason
      * Semantically uses an unordered list
      * There is now a <nav> element in HTML 5
        + Adds good semantic value to the system
        + Helps google
      * Can code an internal Jump List
        + Jumps to different parts of the page
        + Can be done using the anchor tag and an id

<a href=”#id”>My link</a>

* + SSI: Server side includes
    - Purpose: to reduce redundant code in the HTML docs
    - PHP is one way, what we will use
      * Runs on the web server, not on the client
      * The server intercepts the usual request and response process to search the page for PHP instructions
      * To trigger PHP, change the file extension from .html to .php

2. A web server is a computer with a connection to the internet and has the software to host the web server.

3. The web was created in a way that is meant to be able to naturally connect similar ideas, like the human brain.

4. FTP: File Transfer Protocol

5. ICANN deal with address registration

6. W3C deals with the Web and the standards and such

7. URL is a type of URI, not the other way around

8. Whenever you have nested tags, you close the tags in reverse order; i.e.

<strong><em>text</em></strong>

9. when is comes to tags that display content on a page, there are two kinds. Block and inline tags.

10. Tags are not for APPEARANCE, they are for STRUCTURE

11.Normal flow: Browsers read our code in order, from top to bottom

12. w3c is in charge of keeping the standard