CSS

* Cascading Style Sheets
* HTML describes content
* CSS formats content
* CSS selectors
  + How CSS identifies given elements to format
    - Class: <p class=”bobberts”>
      * Typically used for many elements
      * Selected by the ‘.’ Character
        + .bobberts{ color: green; }
    - ID: <p id=”bobbert”>
      * Typically used to identify one unique element
      * Selected by the ‘#’ character
        + #bobbert{ color: yellow; }
* Inline
  + <div style=”background-color: blue”>
* Internal stylesheet (embedded)
  + Uses the <style></style> to create a CSS block inside the given html page
    - <style>
      * H2{
        + Color: red;
      * }
    - </style>
* External style sheet
  + Saved as separate file
  + Brought in with <link>
    - <link rel=”stylesheet” type=”text/css” href=”[c:/location]”>

Document Object Model

* Html
  + Head
  + Body
    - Div
      * Nav
    - Div
      * Div
        + p
      * div
        + form

input

input

Content Delivery Network (CDN)

* A url that your website can hit to bring in specific files remotely. This would be opposed to downloading the file specifically and referencing it locally.

Bootstrap

* Collection of CSS and JavaScript files for designing fluid mobile friendly websites
* Utilizes a grid system for horizontally formatting CSS
* Container
  + CSS class to encapsulate and format a section of the page
    - Container- centers content more to middle
    - Container Fluid – uses more real estate

XML

* Extensible Markup Language
  + Used for distributing info over the internet
  + Designed to store and transport data
  + Self-descriptive
  + Doesn’t do anything
  + Carries data, html displays data
  + Syntax
    - Same structure as html (tags, elements, attributes, etc.)
      * Attributes are typically used for metadata
    - Must have one root tag
    - Optionally have a prologue
      * Must be first line
      * <?xml version=”1.0” encoding=”UTF-8”?>
    - All elements must have closing tags
    - Tags must match case
    - Attributes must be quoted
    - Reserved symbols require entity references
    - Note: white space is preserved
    - No overlapping tags
  + Follow all of these rules and you will have well formed XML

Namespace and prefixes

* To avoid classing words that are used for other reasons, we can us prefixes with namespaces
* <tables xmlns:xml=”custom tables”>
* <xml: table>
  + <xml: tr>

Document Type Definition (DTD)

* External or internal file that XML can use to enforce specific naming conventions and structure
* Example rule: root tag must be ‘A’
* ‘A’ must have ‘B,C,D’ and optionally ‘E’
* ‘C’ must have ‘F’
* XML Schema Definition (XSD)
  + Same goals as DTD
  + Stronger
    - More complex rule sets
      * Data type enforcement
  + Written in XML
* If your XML is well formed and perfectly conforms to XSD or DTD then you have valid XML

JavaScript

* Interpreted runtime language
* Supported by most browsers by default
* NOT Java
* Originally called mocha, then changed to livescript, shipped with Netscape as JavaScript
* Created in ten days
* As a result, it came out weakly typed, and is at it stands today
* Where to insert JavaScript
  + Internally useable within <script></script>
  + <script> can be used to import external javascript files (.js by convention)

Js Scope

* Global
  + Access anywhere
* Function
  + Within function
* Lexical (ECMA6+)
  + Block created in

ECMAScript6 (ES6)

* What JavaScript is built on
* Brought :
  + Let
  + Const
  + Arrow notation
  + Private data types
  + Classes

***More Requirements***

* ***At least one of each***
  + ***Inline CSS***
  + ***Internal CSS***
  + ***External CSS***
* ***All pages should have their own CSS file***
* ***Leverage bootstrap (optional)***
* ***At least three CSS selector types***
  + ***E1***
  + ***E1>e2***
  + ***E1~e3***