Client Side Tech Review

* The three pillars of client side technology are
  + HTML
    - Structure and content
  + CSS
    - Styling
  + JS
    - Logic/ dynamic web pages

HTML

* Hyper Text markup Language.
* Based on tags/elements.
* Elements can be nested.
* Elements have attributes.
* NOT A PROGRAMMING LANGUAGE.

CSS

* Cascading Style sheets
* The cascading algorithm is what determines the ultimate CSS look for an element.
  + Closest most specific CSS wins.
* Bootstrap is a popular CSS library.
  + Created by Twitter.

JavaScript

* The only programming language on the internet essentially.
  + All browsers run JavaScript.
* Has almost nothing in common with Java.
* Scopes in JS
  + Global (put nothing in front of the variable) AVOID
  + Function (put var in front of the variable) AVOID ALWAYS
  + Block
    - Let
      * the variable can be reassigned.
    - const
      * the variable cannot be reassigned.
* Objects
  + Objects are just key value pairs.
  + Objects are not class based.
  + Object literal syntax
    - {name:”Adam”,age:19}
  + Prototypal inheritance
* Truthy falsy
  + JS Has very aggressive type coercion.
    - “JS will compare apples to oranges even if they are both bananas.”
  + Any value in JS can be coerced into true or false
  + Every value is true EXCEPT the falsy values
    - 0
    - “”
    - Null
    - Undefined
    - NaN
    - False
  + == abstract equality
    - JS will perform type coercion then compare values
  + === (strict equality)
    - JS will compare value and type
* Primitive Types in JS
  + String
  + Number
  + Boolean
  + Undefined
  + Null
  + Symbol (I have never used)
* Arrays
  + Can hold anything in JS.
  + Dynamically resize themselves.
* Functions in JS
  + Functions are OBJECTS!!!!
  + JS is a fairly functional language.
  + Types of functions
    - Method
      * Function that is a property of an object
    - Callback function
      * Function passed in as a argument to another function
      * .addEventListener(“click”,func)
    - Arrow function
      * ()=>{}
        + The binding of the this keyword is constant in an arrow function.
    - Higher order function
      * A function that takes in a callback function
      * Myarray.map()
    - Anonymous function
      * A function that has no name and defined on the spot.
* AJAX
  + Asynchronous JavaScript and XML.
  + We can issue http requests in JS without having to reload the entire page.
    - HTML forms reload the entire page by default.
    - The web used to require people to reload the entire page whenever you sent an http request.
* Async await and promises
  + Async is a keyword that marks a function as asynchronous
    - It allows you to use the await keyword within that function.
    - Any async function always returns a promise.
  + Promise
    - An empty object that will eventually hold a value.
    - You can await a promise which means JS will hold off on executing any more code in that function until that promise is fulfilled.
* Closures
  + Function within a function
  + The inner function can use variables defined in the outer function.
  + Even after the outer function has ended