Week One

### Numbers

* Are ALWAYS floats
* Exponents are notated as: Math.pow(2, 4) === 16
* Square Root: Math.sqrt(81) === 9
* Random: Math.random()\*10 === (num between 1…10)
* Rounding down: Math.floor(9.9999) === 9
* parseInt: .parseInt(string) -string to Integer
* parseFloat: .parseFloat(string) -string to float

1. Array Methods

* .sort(): sort(function(a, b) {return a - b; }) -for numbers
  + .sort() -for strings
* .concat: myFriends.concat(yourFriends)
  + combines two arrays
* .indexOf: foods.indexOf(“Pho") === 9
  + finds the “index of”
* .splice: foods.splice(13, 1) -goes to index 13 and removes 1
  + to add to array at that index: splice: foods.splice(13, 0, “Ice Cream”)
* .slice: foods.slice(4, 10) -finds array items starting at 4 not including 10
* .split: string.split(“, ”)
* .join: array.join(“, “)
* .push: array.push(“New Stuff”) -puts it at the end of array
* .pop: array.pop() -removes last idem from array
* .remove: array.remove(1, -2) -removes the 2nd and 2nd from the last items
* .shift: array.shift() -like pop, but retrieves and removes 1st array item
* Copying Arrays -array2 = array.splice[0]

1. String Methods

* .toString: (20).toString() === “20” (or 1..toString())
* .replace: string.replace(/game/g, “book”) -replace “game” with “book” everywhere in a string
  + string.replace(“b”, “a”) -replaces first instance of “a”

1. Object Rules

* Associative Arrays
  + Ruby calls them hashes
  + keys will ALWAYS be strings
* Functions are Methods, not Objects
* Arrays are Objects
* Method: person.greet.name
* JavaScript objects are “pass by value” [not by variable]
  + passing values into objects
  + functions & objects are bubbles with their own set of values
    - in other words, they have limited scope
    - var variable = local to function
    - variable same as passed parameter = local
    - variable [without var] becomes global
    - Hoisting = function doesn’t search outside of itself for a definition of a variable if it is defined later in the function
    - Closure = function looks outside of itself to define (close) an undefined variable
    - Lexical (scope) = variables staying within their functions/objects unless sent out intentionally. Variables stay within the “scope” that they are used
    - Shadowing = overriding the definition of an outside variable with an inside(function) variable
    - Procedural Abstraction = naming an object to reduce complexity
    - First Class Treatment of Functions = functions can be passed as values
    - variable( ) -parenthesis expects that the variable is a function
    - Inversion of Control: -objects within objects within methods…
    - In Place -working with arrays without modifying them
    - Necessary Eval - NEVER use eval( )!

1. HTML/CSS

* Semantic Tags -such as <em></em> help google results and page readers. Headings are also treated this way.

1. Document Object Manager (DOM)

* DOM -Shows structured elements on a page. It is a representation of your code on a web page.
  + DOM Selection
    - You can call things from the ‘document’: document.getElementById(“greet”) finds the element id=“greet” and pulls it up.
    - document.getElementsByTagName(“div”). Or (“input”), (“ul”)
  + DOM Manipulation
    - element.style -lists all CSS styles
    - “element”.style.color = “red” -changes the element to red
      * “element” is set by getElement… and setting it to a variable called: element.
    - document.querySelector(“div”) -grabs first div on the page
      * document.querySelector(“.greet”) - grabs first class of “greet”
      * document.querySelectorAll(“div”) -grabs all divs on a page
    - You can use for loops to iterate through elements
  + DOM Events
    - document.getElementById("greeting").onmouseover = function(event){
    - alert(“Hovering!!");
      * pops up an alert every time I hover on “greeting”
    - window.onload = function () {

do something that you want to load on window load }

* + DOM Keywords
    - classList -
    - HTML data attribute: data-name=“whatever”

1. Outcomes Team

* In week 15, we will present at a job fair with 15-100 recruiters
  + October 13! Mock Meet & Greet
  + October 15! Meet & Greet 5pm - 9:30pm
  + Use GlassDoor to look up company culture

1. Best Practices

* single ‘ ‘ in JavaScript
* double “ “ in HTML
* Test your code

Week 2

1. Test Driven Development - Week 2

* RSpec -Ruby and Rails
* Mocha & Jasmine for JavaScript
* Node Package Manager (NPM)
  + install packages
  + version control
  + npm install -g
    - -g means global
    - npm init
    - npm install —safe chai
      * will install
* Why Test?
  + If app breaks, the test will help
  + Collaboration
  + Helps browser companies with bug reports
* JSON
  + JavaScript Object Notation
  + Has very little to do with JavaScript except having objects
  + just a list of Objects
  + description of a new project you create
* Mocha
  + console.assert(true, “message”) -if true, print error with message
  + setup(function() { var setVal { }} it( )…
    - begins each it check
  + it(‘message’, function( )) -
    - contains assert and/or assert.equal
  + assert(true, “message”) -if true, print error with message
  + assert.equal(that, ‘this is’, ‘message’) -if “this is” that…
    - (true, 0===0), (friend.name, ‘john’)
  + teardown(function( ) { }
    - finishes the test

Subject