**RampUP Notes:**

<h1> = header 1

<h1>AJ</h1>

image tags: <img src=”file path”/>

* 3 pieces of web:

1. HTML – displays content/ provides structure. Always read top to bottom.
2. CSS – Styling
3. Javascript – handles events

* J-Query – a library of javascript codes.
* Attributes:

All tags have a style attribute, containing styles that can be applied (buttons, headers, colors, etc.).

Attributes like width and height are written in pixels (px)

Width= 500px

Height= 75%

<div> (stands for divisions)

* tags can be “nested” , example = <div>

<h1>Hello</h1>

<img ……/>

<div>

**6/12/14**

* **Lists in HTML**, two kinds
  + Ordered lists <ul> default is a bullet point
  + Un-ordered listed <ol> is a number
  + List items <li>hello</li>
* **Blocked tag** – everything gets its own line in a blocked tag, no matter where you put it.
* A tags <a> holds links. Inside them are tags called “href”
* **File paths** (switching between web pages) start from top, work your way down.
  + To go down the file path, it is a /folder names: To go back up, you use ../folder names.
* **Inputs -** <input type=”text”/>a generic term for anything you are going to interact with on the screen.
  + you can have as many attributes as you want. Attribute=”whatever”/
* Nav tags

**6/26/14**

* All CSS classes are noted by a “.”
  + ID = #
* Classes are ranked by specificity
* The difference between an ID and class is that you only want one ID that is unique, those ID’s should not be elsewhere.
* Pseudo selectors: style interactions when you click or hover on something. Responds to an action.

**7/3/14**

* 3 different kinds of images
  + .gif – few colors, keeps the file size of your image down
  + .jpeg – complex colors, patterns
  + .png – always use if you can, newer, allows transparency
* Fonts
  + In css 🡪 @font-face{ font-family:”nameOfFont”, src:url(../fonts/filename.ttf);

Start thinking about a portfolio website.

Use a site called Dribbble.com

Hostgator.com

1&1.com

Use craigslist for freelance work.

* Popular back end languages: Ruby, Javascript, C-sharp
  + Javascript – front end web based applications.
  + C-Sharp – can make mobile apps, back-end web based applications.

**N-tier Architecture**:

* Web application development – 3 Tiers
  + Front-end – Website or mobile application
  + Back-end – Web services, allows a user to interact with the database
    - Create
      * Put
    - Read
      * Get
    - Update
      * Post
    - Delete
      * Delete
  + Database – stores the user data

J-Query – “query the DOM” or query your HTML

* Jquery is always denoted with a $
  + $(“#username”).text();