The first week I felt so overwhelmed, I didn’t know what I was doing. I didn’t even get to my second class until Friday. Friday was so much better than Thursday.

**1/9/17-------------------------**

Today I felt pretty good about working on my school work. It then took 5 hours to finish my first program. It would have gone a lot better if Linux would have renamed my filename right off. I am just getting back to this class after 6:00pm.

CIT261Class

Stand19762

**1/10/17------------------------ *Loops and For Loops***

I am relearning about loops today using JavaScript. I use to know how to do this at Plexus but I and really rusty at it now. Both of these do the same thing. It takes a lot less space and easier to right especially if you have hundreds or thousands of entries. [Loops Example](http://www.w3schools.com/js/tryit.asp?filename=tryjs_loop_for)

|  |  |
| --- | --- |
| text += cars[0] + "<br>";  text += cars[1] + "<br>";  text += cars[2] + "<br>";  text += cars[3] + "<br>";  text += cars[4] + "<br>";  text += cars[5] + "<br>"; | for (i = 0; i < cars.length; i++) {      text += cars[i] + "<br>"; } |

I read to section 4 page 46 in Doing Stuff with WebThings. I learned how to save, edit and retrieve information that a user inputs. I’ve been using the Sublime Text program to write my code and I then test it in Google Chrome.

**1/11/17-------------------------- *Studying with Doing Stuff with WebThings***

Today, I am working on Doing Stuff with WebThings again. I had to troubleshoot some code that was in the textbook that wasn’t working. Using the Chrome Inspect option, I was able to figure out that the Input Box ID didn’t match what was in the function script.

I worked on my fluency letter which I think is coming along well. I got out group set up to use hangouts in out meeting. It is set for Friday at 2:00pm but that could change. Two have not accepted the time yet and one doesn’t appear to have hangouts.

**1/21/17 -------------------------- *YouTube Video Creation***

The 15th through the 21st was completely devoted to learning how to do videos on YouTube and trying to keep up in CS 124. These videos are taking about 1 full day to create. I am devoting every other day to CIT 261. Saturday, everyone showed for our group meeting. We had a good discussion. I attempted recording our meeting without being asked, but it did not record. I found we have to add people in YouTube not hangouts. During the group meeting we discussed we need to start doing it. I think it was Justin that said he would do it the next week.

**1/27/17 --------------------------**

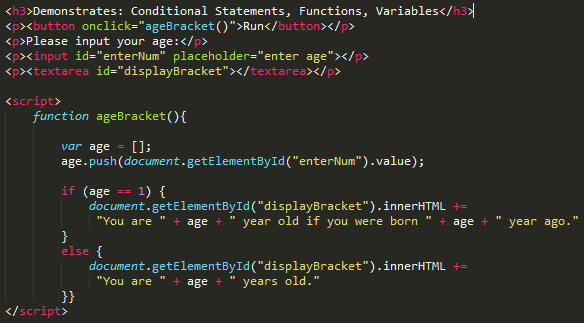
I finished my videos on Monday so I spent this entire week trying to catch up with CS 124. We have a Unit test that I am studying hard for and I am very worried about it. I have scheduled a meeting with a tutor and teacher. There are some very difficult things to understand. I have been keeping daily contact with my group. I really want to get back to coding in our class.

**1/30/17 -------------------------- *Using If Statements, Var, Arrays and Functions***

**Conditional IF Statement, Functions and Variables**

I was finally able to work on my age bracket code. I simplified it down and I was able to create it without any other help. I actually had some “document.get” statements that were sitting around unneeded. My code demonstrates the use of conditional if statements, variables, array and functions. I feel I am understanding if statements and functions, variables, if statements, arrays and parameters a lot better now. I am planning on requesting assessment tomorrow.

Figure : Conditional IF, Function, Variables



**2/1/17 --------------------------- *Saving and Loading data***

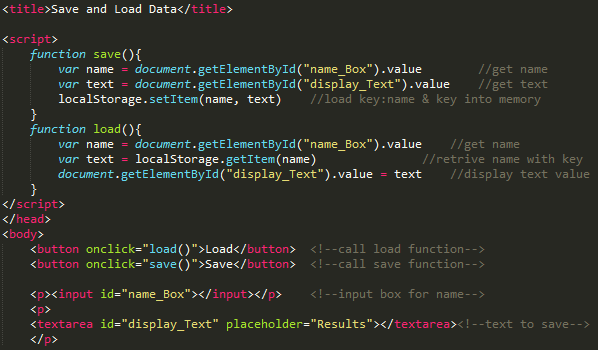
In learning about saving and loading data from memory and attempting to write it myself, I could not get the code to work I was writing. After checking one thing at a time, I discovered that the “d” was in caps in “getElementByI**D**”. It really counts to write things correctly the first time, because errors are very hard to find and takes time to find.

Note: I want to point out that it is important to remember that *name* is actually a key to find the text you will be saving and loading.

This program entitled “Save and Load Data” contains two functions which are initiated by two buttons “Load” and “Save”. The following is an explanation on the workings of both functions.

1. **Save Function:**
   1. It takes the value of the input *name\_Box* and adds it to the *name* variable.
   2. It then gets the value of textarea *display\_Text* and adds it to the *text* variable.
   3. localStorage.setItem then loads the key *name* and *text* value into memory.
2. **Load Function:**
   1. Takes the value of the input *name\_Box* and adds it to the *name* variable.
   2. localStorage.getItem uses the key *name* and put its associated text in variable *text.*
   3. It then displays the value of the *text* variable in textarea *display\_Text*.

Figure : localStorage



**2/8/17 ------------------------- *Created InsertBefore and Remove Child – Updated Add Child***

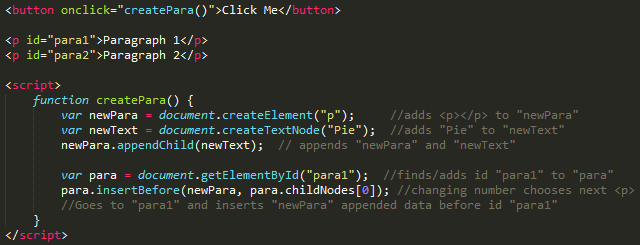
Today, I created and learned about InsertBefore and Remove Child and added them to GitHub and Webpage.

*Plan:* I am wanting to create a web Journal if I ever get enough time to do so. Creating these codes and understanding them takes a lot of time. I also want to play around with the code but not much time for that either.

**insertBefore**

In figure 3 below we first create a new <p> tag and add it to variable “newPara”. Next, we create text “Pie” and add that to the variable “newText”. We will then need to append or combine the data together into “newPara”. This data now consists of the <p> tag and the text. We then locate the ID “para1” and insert “newPara” before the (para.childnode) number 0. We can increment this number and so doing choose which <p> tag to enter the new tag before.

Figure 3: iInsertBefore

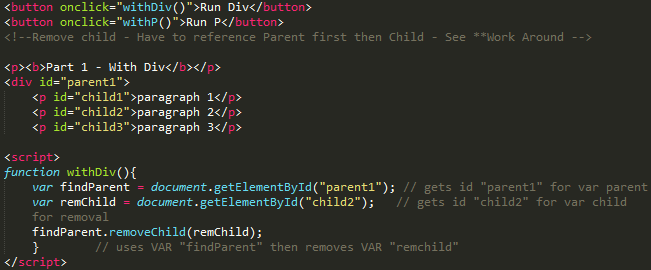


**Remove Child**

I have included two parts for the Remove Child code. In order to find the child <p> for removal it is pertinent to reference the parent <div> first. The 2nd part is a work around for this.

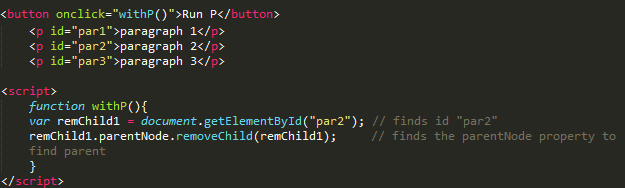
**Part 1 –** The code first of all locates the ID “parent1” which is a <div> tag and adds it to the variable “findParent”. Next, we find the ID “child2” which is a <p> tag and adds it to variable “remChild”. It then uses the findParent variable to locate the child within which is defined in the “remChild” variable. It then initiates the “removeChild” directive, removing “child2”.

Figure 4: Part1 - Remove Child



**Part 2 –** This code doesn’t need a <div> or parent to function. We still have to reference a parent though for it to work. We first need to locate “par2” and add it to a variable “remChild”. We then reference that same location with the “remChild1” variable, and reference the parentNode. We can then remove “par2” within the “remChild1” variable.

Figure 5: Part 2 – Remove Child \*Work Around



Other Tasks:

I also added some functions and buttons to several of my different codes to make them a little more interactive for the user.

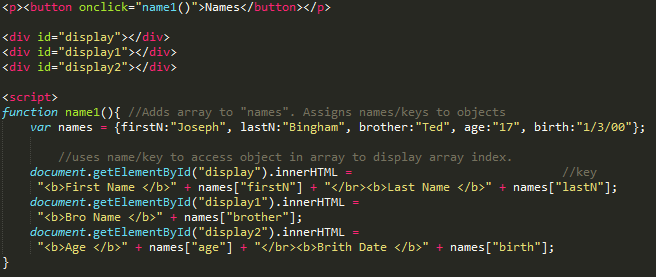
**2/8/17 ------------------------- Arrays & Objects and JSON**

**Arrays & Objects**

In the following code, we can use names to access each index instead of numbers.

The code reads as follows. We first have our Array “Name1”. It consists of 5 indexes in which each index has a name associated with it (e.g. firstN:”Joseph”). We can then access each index by name/key word. This is very similar to an associative array in which we are required to use the “document.write” element. Here we can use our “document.getElementById” to output our code to any tag of our choosing. We access an index by stating the array[“indexname”].

Figure 6 – Array & Object



**JSON**

JSON stringify is used to stringify or convert all data sent through it into a string. A string is just as it sounds. It is a string of numbers sent to storage for later access. When we do want to use the data again we will need to parse it. Parsing using JSON turns it back into the format before it was stringified.

Figure 7 - JSON



Referring to Figure 7 above, I am using my save and load code which uses the localStorage property. In our “function store”, we will use “JSON.stringify” to turn our array into a string and save it to our local storage in memory. In our “function load”, we will retrieve the data from storage and use “JSON.parse” to format the data back into a form we can use. We then display it to the screen.

**2/11/17 ------------------------- Group Meeting & Stylesheets**

Today, we had our group meeting. Everyone showed up which was nice. I re-learned something that I had forgotten when I worked for Plexus. We were talking about how to use a stylesheet located in a file. I worked a lot with these working with Plexus but it has been a long time. I instituted this into my webpage. Justin also showed us how to do it with PHP but it looks a bit foreign to me right now. The only thing I have done with PHP up to this point is submitting form data to the server.

**2/16/17 ------------------------- Object Creation containing properties and Methods**

This day, I watched Justin’s video about Object creation and assigning properties to that object using “this.” property keys. The Object itself is created with a “function name ()”. Within this same function () we create another function and create a variable to combine all of the properties.

To use this information, we will need to initialize home () by assigning it to a variable “johnsHome” We can then run the function by calling that function with the new variable “johnsHome” as demonstrated in Figure 8 below.

Figure 8 – Object Creation, Properties and Methods



**2/16/17 ------------------------- Object Creation – Inheriting Objects and Properties.**

I again watched another one of Justin’s video’s, this time learning about Object inheritance. Everything thing is the same as in the previous code except we are adding another object for an Agency.

We will demonstrate this by creating a function called agency () and give it properties just like we did for the home (). Let’s say that this Agency acquired the home. It is now part of the agency’s assets. In order to copy/use the data from the first function home (). We will use a prototype. The agency can then access the data of the home ().

Figure 9 – Object Inheritance

