**First off, let me apologize** to those who attended my office hours this last week. My body changed time zones, but my brain did not. Which explains why I was online at a different time than everyone else. Again, my apologies.

Office hours this week will be:

Tuesday 7:00 PM Mountain (Bro James, that means 8:00 PM for you!)

Wednesday 7:15 PM Mountain

# Week 3 – This is the Way

* Weekly pattern
* How should I turn in my code?
* How is everyone else doing it?
* Team Activity
* The weekly activity
* Weekly Report
* Your Questions and Discussions
* This is the BYU-I Way
* A few fun examples using the High Order Functions for JavaScript arrays

# Weekly Pattern

1. Read the assigned sections
2. Take notes
3. Do the sand box exercises and experiments
4. Turn in your notes – These notes are more for YOU than they are for me. You will be glad you saved some good notes in the project phase when you need to be reminded of something.
5. Finally submit your Code link AND/OR your notes
   1. Make sure you have a website.

The first two weeks you might not have had much to submit. Now you do. I read your notes and review your code examples to see how you are progressing. Without this I can’t assess your progress. Many of you just submit the URL and that works great. Canvas takes a screen shot of your page and gives me a clickable link to jump right to your site.

BTW I let you turn in late work on Week 1 because we are all just getting started. I cannot accept late work going forward.

# How should I turn in my code?

1. Best: An actual web site works the best
2. Good: GitHub pages work well. If you do not have a github.io repo they are **very** easy to create. If your GitHub ID is ClarkKent, you would create a repo call ClarkKent.github.io. Put an index.html page in there with references to your individual assignments and off your go! Test it before you submit. Make sure that link is working. I see a lot of 404 errors. Have someone else test it to make sure it works.
3. For example:
   1. My GitHub identity is gtjames.
   2. This is the link to ALL of my repos [www.github.com/gtjames](http://www.github.com/gtjames) (if you see something you like clone it)
   3. This is the link to my GitHub pages repo [www.github.com/gtjames/gtjames.github.io](http://www.github.com/gtjames/gtjames.github.io)
   4. And this is the link to run my web pages <gtjames.github.io>. By designed it is not pretty. It is purely to show you a sample of how a GitHub pages site works.
   5. Anyone can do this. If you need a website, take 15 minutes and rename your current repo. YourUserName.github.io. No excuse to not have a simple working site.
4. Because creating a github.io site is so simple I will not accept just your github repo url or zip files.

# How is everyone else doing it?

I have seen several very good and/or simple web sites from your peers. Please, look at what they have done (100% of them are github.io sites). Some have nice CSS going on and others are very simple. It makes no difference to me. However, in the weeks to come you should consider adding some nice styling to your pages. Notice how they combine the code and the notes for one stop grading. Notice the details of the notes. These notes benefit you, be detailed.

|  |  |
| --- | --- |
| Daphne Musenuka | <https://daphneavril2604.github.io/WDD-330.github.io/> |
| Lisa Ward | https://war21001.github.io/WDD-330/ |
| Nicolas Cable | <https://watercable76.github.io/Personal-Portfolio/> |
| Brennen Grimshaw | <https://begrimshaw.github.io/WDD330/> |
| Emily Wall | [http://wallylime.github.io](http://wallylime.github.io/) |
| Jesse Wilson | <https://jesse-wilson88.github.io/wdd330/> |
| Ashley Zufelt | <http://azufelt.github.io/wdd330> |
| Alesha McCown | <https://aleshana.github.io/WDD-330-Portfolio/index.html> |
| Eugene Olsen | <https://eolsenbyui.github.io/wdd330/> |

# Team Activity

Answering the questions will get you your points but it will not get you understanding. Make sure you invest the time on the assignment. Work together. Do some prep work before your meeting, come ready to contribute. Wait to the last step to view the instructor’s example. If you look at it right off the bat it will short circuit your learning. Pondering is a necessary step to deeper understanding. If the answer is just given to you, your understanding will be more limited. These points are practically free. There is nothing for me to observe or grade. So just the act of filling out the form gives you your points but like I said at the start it won’t add to your understanding unless you work at it.

# Weekly Activity – Arrays and Array functions

See the end of this announcement for some examples. Including links to code I have on GitHub.

# Weekly Report

1. What did I do since the last meeting?
2. What am I going to do next?
3. What are my blockers?

While there is nothing for me to really assess with this submission, take the time think about the last week and consider needs to be overcome for this week. Do not race past this seemly inconsequential assignment. I see way too many responses like this.

What did I do since the last meeting? I read and did the assignments

What am I going to do next? Work on the next assignments

What are my blockers? I really don’t have the time for everything

That is just what I do **not** want to read. These notes are not for me they are to hold you accountable for last week and give you goals for the next week. Report on the specific tasks you accomplished last week and commit to the specific activities you will work on this next week. If time is a blocker; do not tell me that time is your blocker. Report on what you will do next week to carve out the time needed for the work. And if you have a real blocker like callback functions then that lets me provide some examples to help everyone.

# Your Questions and Discussions

I regularly see questions and it is not always obvious if you are just posing that question to yourself for future research or if it is something you have discussed in your team and still need some help from me. If you have researched a topic and are still stuck on it, please ask me. Either in an email or your weekly report. Make it clear that you are looking for some additional information and I will provide some commentary in my response to your submission.

Discussions – you might notice I have changed the modules so that each week has a Discussion board for that week’s topics. Post your questions and insights and answers there. This can be a great resource to the class

# ‘This is the (BYU-I) Way’

Your teaming is critical to your progress. You will gain so much more my attending your team meetings. Please do not slack off and ignore that experience. As the Mandalorian would say ‘This is the way’. Maybe we would say ‘This is the BYU-Idaho way’. We learn together and share what we have learned. If you know it all, please share that with the team. If you don’t know anything, PLEASE share your questions because I can guarantee you, there is someone on the team meeting with the same question.

I think another part of the BYU-I Way is to have fun along the way. After the first 4 weeks or so you have a lot of tools you can start to use. Use them to create something silly, explore, find a new API to tinker with!

Next week I will start sharing some work from students in the class.

# Week 3 you will experiment with arrays and the array functions

How is you experimenting with arrays going? JSON has become a common data format to use to pass between servers. Most AJAX APIs use JSON for their data format. Inevitably that data includes arrays. So, these exercises were great for developing familiarity with them.

If you would like some more practice, I have a link here for about 800 volcanoes.

## Volcanoes

<https://github.com/gtjames/justJS/blob/master/volcanoes.json>

It includes the year, country, elevation, and number of deaths.

Here is a link to some starter code to play with the data

<https://github.com/gtjames/justJS/blob/master/volcanoes.js>

I use a node package ‘fs’ to read in the volcanoes.json file.

## People on the Titanic

I love this one. This is a GitHub gist set. I have three files here

<https://gist.github.com/gtjames/717c21399cb8198348b7fcc138bc2057>

Titanic.js

Titanic.json these two files contain the same info. Name, age, passenger/crew class,

The code is set up to read the json file. I have provided numerous examples using filter, map and reduce.

## Boulevards de Paris

Did you find all of the boulevards in Paris which include ‘de’

*const category = document.querySelector('.mw-category');  
const links = Array.from(category.querySelectorAll('a'));  
links.filter(a => a.title.indexOf("de") >= 0)*

Your solution could have been as simple as the above. Of maybe you made yours a little smarter to only look for “ de “. That way boulevards with des would be excluded.

I believe some of you may have been ‘fooled’ by the browser in to thinking the querySelctor and querySelctorAll did not work because they returned ‘undefined’. That is an oddity of the console window. It simply means that this is the first time that the object **category** or **links** was created and the initial value was **undefined**. But then it immediately was assigned a value. I do not know why the console does that. If you enter **category** or **links** after the query, you will see that they do contain an array of information.

While we are on the topic of the category and links arrays. What kind of data do they contain? It is DOM information. Therefore, we will treat them like DOM objects. You can see that I use **a.title**, in the predicate of my filter operation. **a** is the anchor tag from the **querySelectorAll** and **.title** is the title of the **anchor** tag. There are other ways to accomplish this. My example is just one of many.

## One more example from the Church’s website

Just for fun I went to the photo approval page for my ward and opened up the inspect window

I Looked at the html and saw that all of the members had a class of “**manage-photo-name ng-binding ng-scope”**

I put that in the getElementsBy…. API and BOOM! I had the DOM for all members

Converted the result to an array, used the map function to strip out the names and BAM! I had a list of all members

ele = document.getElementsByClassName('manage-photo-name ng-binding ng-scope')

let names = Array.from(ele).map(e => e.innerText)

I tried to use it on the Directory page for the ward but there were problems from the server. So, I thought ‘I wonder what else I could do it on?’

I will play around more this week and find some other clever places to extract data from a web page. When we get to the API weeks of class it becomes more fun!