Week Three: And we are off and meeting!

* What do you do this week?
* Weekly Self-Reports
* Team Meetings
* Team Schedules
* Teaching example
* Random but important notes
* Grading
* What is this weekly reflection thing?
* Observations

First of all – I do expect you to read my announcements/emails every week. Since we do not meet in a classroom twice a week this is a minimal way to stay in contact with each other and to inform you on what you should be thinking about this week. I will try not to bore you. I might have hidden ’Easter eggs’ in the middle of paragraphs that will ask you to respond to easy questions, just to see if you are paying attention. Not this week, but sometime soon.

# What do you do this week?

Start getting in the groove for the next 8 weeks. Things will be very repetitive.

* Learn your topic, prepare a presentation, present your material in the team meeting, have fun.
* Attend your team meeting, listen to a presentation, **ask** **questions**, look at the code, study your team mates’ code over the next week and make it do something more.
* As you gain understanding put together web pages that show you have begun to master a particular code topic.
* Submit a link to your demo page for review and feedback
* Wash, Rinse, Repeat for the next 8 weeks.

# Weekly Self-Reports

Sorry I do not extend the due dates on these

# Team meetings

First of all, if you are using a Google+ Community please add me to the community. I am **gtjames@gmail.com.** Do not use my BYUI.edu email address. Please invite me to all meetings! I like to attend as often as I can. I usually stay for only 5-10 minutes and only about every two to three weeks. Make me a part of all invites and I promise not to over stay my welcome. It helps me get to know you better and lets me answer questions in real time.

An hour should be long enough for the team meetings. Prepare well, create your presentation so it is nice and tight, demonstrate the principles of your chosen topic, show your code and answer questions. Distill your material down to the essentials.

An ideal meeting is 3 students teaching a different topic each for about 15 minutes. Next week different students, different topics and then eventually old topics revisited getting covered by someone else at a deeper level.

Make sure you are recording the team meetings. You will need to post them to your team community and direct my attention to where you teach and support the team. This is for the Professionalism part of the grade.

ATTENTION: I was made aware of a situation last semester, that I am sure is not unique and probably has happened a number of times but was not brought to my attention. It is this – we are all equal in this class. Women – men, on-campus – off campus, native English speakers and those who are not. Please treat each other with respect. Listen to each other. We have Freshmen to Seniors in this class, leave no programmer behind. If you know the topics already it is your duty to help the members of your team to gain from your experience. ‘nuff said.

# Team Schedules

Some of you are using Google docs to track your teaching schedules. That is wonderful. I love that. But you do need to submit that in Canvas for Professional Scheduling assignment. Canvas MUST have a submission to enable me to grade your work. Submit in Canvas->Assignments-> Scheduling->Personal and Team Plan and give me your spreadsheet or a link to your Google docs. Everyone needs to do this. Not just one member of the team.

Remember what I said about teaching 4 times. I have had teams in the past think they could just blast through them and everyone just teach once and be done. That is not what I want to see. It would be pretty hard to absorb everything that fast. So please don’t try to do it. So far this does not look like anyone has taken this route.

# Teaching Example

<https://www.youtube.com/watch?v=I9lw8sLAnpo&index=2&list=PLxe68xvCWR40cgkW_lIXX4wirx8o44CLJ>

This is a simple presentation I created for AJAX. Just a sample of what you can do. Here is the link to the working page.

<http://gtjames.github.io/AJAX/tinyAJAX.html>

Here is the link to the PowerPoint deck.

<https://drive.google.com/open?id=1JdiaOysjmkR3O_-xQjtuaIjIBurKCLCK>

This would NOT earn a SKILLED if submitted. It only shows how AJAX works in isolation. To get to Skilled you need to do additional work and use the topic to support a mini-application.

I shared this earlier. I will show something new next week. But don’t wait for me. Explore on your own!

# Random but important notes:

* **No frameworks**. Sorry no Angular. What about JQuery or Bootstrap? Also, No and No. No dollar signs in your code. Just the basics for this course. We are focusing on the foundational principles. I want you to see the basics of AJAX, Local Storage, DOM and all the rest that gets hidden by the very nice and powerful frameworks. We are going to understand the first principles. Also, I don’t want students who are not familiar with the frameworks to get lost in yet another new way to do things.
* **What is a ‘single page’ application (SPA)?** An SPA creates a flowing user experience by using DOM to manipulate what is visible to the user, local storage to save the ‘state’ of the user and AJAX to dynamically pull new data or screen panels to change the UI. We do not want the user to feel like they are going from page to page to page. This is a key part of responsive design. Minimal lag time between the user’s actions and the application’s reaction. AJAX and DOM make that happen.

# Grading

As I see your code topics examples I will enter scores and feedback in Canvas Assignments. I will enter in feedback every time you submit code. Make sure you look for my comments.

**Proficiency with the Code Topics**

The best way (**actually the ONLY WAY**) to show your code and get it graded is in a working web site.Create an index page that lists each of the code topics with links to a page where your code is demonstrated. If you have a web site great, if not you might consider using **github pages** which is just a repo on github called <yourName>.github.io. For instance, I have <http://gtjames.github.io>. (It is extremely ugly and I do not recommend the style of layout (or lack thereof) to anyone. Many links are not functioning. It is purely there to give you a basic idea). But it shows how to use **github pages** to demo your code. I have several links that show some pages on AJAX, DOM and LocalStorage. The AJAX page is associated with a training video I have included in this announcement. I may create other videos to share just as examples, but don’t count on them. This class is about learning to identify and use resources on your own and with your team to gain understanding of new topics.

# Weekly Reflection

Some have asked about the weekly reflections. They are NOT submitted at ANY time. They are for you to keep a record of your learning process. What troubles you had and overcame. How you resolved the problem you faced. What types of resources you found that were the most help. THEN on week 14 you will create a document (more than one page!!!!!) wherein you document what you learned about the learning process. So, if you do not keep a record in these weeks your Final Reflection will be rather shallow.

# Observations from office hours, emails and Canvas Code Topic Submissions

* What about Personal Projects?

Start thinking about a personal project. DO NOT start it yet. You need me to OK it. Submit ideas to me as soon as you like. Just email me.

* Submitting code for Code Topics reviews
  + **Make sure the link you send me works.**   
    If I get a Page Not Found error I may try and figure out your typo for a while, but I won’t spend much time. Double check the URL before you give it to me.
  + **Make sure the code on your page works.** I will NOT debug your code. If you have questions, keep doing your homework, if you are still stuck talk to your team, after that I will be happy to take a look. I have made so many programming mistakes in my life that I can find errors fairly quickly. You get that skill from fixing your own mistakes. I do not want to deny you that gift.
  + If your code is too simple I will only comment on it and not grade it. You will not get a grade for code you simply found on W3Schools.
  + A few simple examples may get you to **Basic** on a Code Topic. To get to **Working** I expect variety and diverse examples of the Topic. To get to **Skilled** it must work with other topics in an application.
  + Some have asked that I give you a check list of things to do for the Topics. That is a Mosaic law kind of approach. Kind of like Home Teaching used to me. Just a check box. So together in your teams and explore the possibilities of each Topic and find out what you can do with it.

If you made it this far your anxiety is probably still pretty high. Don’t worry you will be comfortable with this soon enough.

If you learned a foreign language for your mission just remember how it sounded the first few weeks. It sounded like gibberish and least Dutch did to my ears. Then in a month it was easy to pick out words and make sense of the conversation and then finally you became fluent. The Holy Ghost helped you there, He will help you here. Hang in there. Let me promise you again the Holy Ghost is as interested in helping you find the truth here as in any other worthy topic.

May the good code me yours,

Bro. James