Week 07 – And the midterm is behind you!

* Office hours to walk through the To-do App if that will help you
* Design notes on the To-do project
* Filter Examples
* Final Project Examples
* (and somewhere a hidden question!)

First off some examples from your peers

|  |  |
| --- | --- |
| Cody Mickelsen | <https://cody-mick.github.io/frontend-dev-2/week5/index.html> |
| Daphne Museruka | https://daphneavril2604.github.io/WDD-330.github.io/week04/index.html |
| Eugene Olsen | https://eolsenbyui.github.io/wdd330/week04/ |
| Jesse Wilson | https://jesse-wilson88.github.io/wdd330/week05/ |

# Office hours to walk through the To-do App if that will help you

The To-do apps covered a lot of ground. There were some very creative ones and then on many others several features were not implemented. If you completed the app and are will to share PLEASE put a link in the response section with your TO-DO App. Some of you went above and beyond. Some will really benefit from your examples. Please share. Let’s learn together. I can’t tell you how many times I have seen some student’s code and have said ‘I have to have that in my portfolio’. I love seeing multiple ways of doing things. For those that did not get as far as you hoped I will be holding two office hours this week.

Tuesday 7:30 PM Mountain (8:30 PM Central)

Wednesday 10:00 AM Mountain (11:00 AM Central

<https://byui.zoom.us/my/bpjames>

I will walk through the code I developed. I have more comments than I do lines of code. Once I do the walkthrough, I will post one of the recorded sessions and share the repo with the class.

Below are a few design notes.

# Design notes on the To-do project:

* Q. I have My to-do list and I have my LI elements how do I keep them in sync?
  + Clicking on the to-do will toggle the CSS attribute to strikeout the screen element but you also need to toggle the done/completed state in the array of to-dos.
  + How do I keep them associated with each other?
  + You could:
    - Have a unique id that is added to the LI and an id attribute in the to-do array. When the LI is clicked you would have access to the ID of the target of the event!! and could use that to find the element in the array.
    - Use the current date time to set the ID of the to-do item and set the ID of the LI to be the same. Using row number will not work.
    - You might think to use a global variable and increment it each time a todo is added. BUT, the next time that app is loaded the global variable will start over again at 0 and your own localStorage list might already have a todo number 0. So then you have to read the LS find the highest numbered id and set your global id to one more than that. WHOOSH that is too much work! I think using a time stamp is the best approach. Also time stamps are very common in the industry to make something unique.
* Q. I am adding to my to-do list just fine, but I can’t seem to figure out how to show just the completed or just the active to-dos.
  + This is a great use of filtering. Some of you used for loops to find the active/inactive to-dos. That is fine. Since learning about filtering I have started using them whenever possible.
  + The filter functionality is a hidden for loop with an if statement. Any element of the array that matches (returns true) is then added to a new array

let activeToDos = allToDos.filter(t => t.active);

* + Translation

let activeToDos = [];

for (let todo of allToDos) {

// == true if you want to be verbose

if (todo.active) {

activeToDos.push(todo);

}

}

* + With the for-loop approach you might consider putting it in a function and passing the true or false to find active or inactive todos. But with the filter approach you just change the compare from t.active to !t.active. No need to encapsulate the code into a function.
  + Oh! what’s this? Just a random note to see if you are reading. Go put a post in this announcement and tell me your favorite movie quote (extra points if it comes from Casablanca! *Sotto voce:* Points not redeemable for WDD-330 points). “The Germans wore gray. You wore blue”. “As the leader of all illegal activities in Casablanca, I am an influential and respected man”. “I’m shocked, shocked to find that gambling is going on in here”! I could go on and on
* I saw several of you get to the point where you had a list of all to-dos, active to-dos or inactive to-dos. But didn’t display them, even though you had a perfectly good piece of code to show all to-dos. This is an opportunity for parameterization. Create a function that will have a single parameter – the list of to-dos to be displayed. Loop through the elements to create the LI screen elements and you are done. If you need to show all to-dos then you call the method with the allToDos list, if you need to show just the active then use the filter code to get just the actives and pass that list on to the function to show that list and then do the same for the inactive to-do lists.
* I see that some of you created an array of strings to track the to-dos. The big handicap with that approach is that you cannot match accurately. You need an id, status and the text. You need an Object! Then you also need to tag the <LI> with the ID of the element in the array. This ties the VIEW to the MODEL.  
   creating to do object  
   let id = new Date();  
   let todo = { id: id, status: ‘active’, value: document.getElementById(‘todoText’).value);  
  this todo object is them pushed onto your array of todos  
    
  creating your inner text you would have something like this to add in the event listener. Or as is more often seen, use addEventListener on the element. We don’t like to mix JS inside of HTML  
   `<input type=’checkbox’ id={id} onclick=’toggleActivity’>

# How about some examples of filters?

(I think I shared this a week or two ago but some new questions came up this week)

Several of you said you wanted some discussion on filter. Here is some code I wrote to play with a fun JSON object I found: 804 volcanoes. The JSON file contains Year, Deaths, Country, Name, Latitude, Longitude, Elevation. Enough attributes to have some fun with. This is some code I developed for a lesson on High Order JavaScript Functions. I use **reduce** to calculate the total number of deaths and the total height of all volcanoes stacked on top of each other, plus the average deaths and elevations. There are several filters to find

* Volcanoes since 1980
* Deaths > 100
* Volcanoes in Indonesia since 2014

Here is the link to the volcanoes.json file in GitHub

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

*// Instead of using the fs package we will take the easy way out to load the Volcano JSON file*let volcanoes =require('./volcanoes.json');  
  
*console*.log(`There are ${volcanoes.length} volcanoes in our list`);  
  
*// this will tidy up our data. The Deaths attribute is inside of quotes  
// so JS sees it as a string. This little tweak will convert Deaths to a number*volcanoes.forEach(v => v.Deaths = +v.Deaths);  
  
*// Volcanoes since 1980*let volcanoesSince1980 = volcanoes.filter(v => v.Year >= 1980);  
  
*// Volcanoes with more than 100 deaths*let deathsGT100 = volcanoesSince1980.filter(v => v.Deaths > 100);  
  
*// Volcanoes since 1970 and with more than 100 deaths*deathsGT100 = volcanoes.filter(v => v.Year >= 1970 && v.Deaths > 100);  
*console*.table(deathsGT100);

*// get a list of all Volcano names in Indonesia since 2014*let names = volcanoes.filter(v => v.Country = "Indonesia" && v.Year > 2014).map(v => v.Name);  
*console*.table(names);

*// count total deaths from volcanoes*let deaths = volcanoes.reduce((total, v) => total + v.Deaths, 0);  
*console*.table(`number of deaths ${deaths}`);  
  
let aveDeaths = deaths / volcanoes.length;  
*console*.table(`average # of deaths ${aveDeaths.toFixed(0)}`);  
  
*// what if we stacked ALL volcanoes on top of each other. How high would they reach?*let elevationTot = volcanoes.reduce((total, v) => total + v.Elevation, 0);  
let aveElevation = elevationTot / volcanoes.length;  
*console*.table(`Total Elevation of all volcanoes ${aveElevation.toFixed(0)}`);

Did this help? What else is causing some confusion? Let me know. I want to have a mini tutorial here each week on the topics you have trouble with.

Your anxiety may be starting to climb at this point. You are halfway through the semester, and you see more mountains and volcanoes ahead of you. Everything you have learned will be with you for the slopes ahead. Look behind you and see what you have accomplished so far in the semester. I see great examples of your code. Take a half hour and look at some of the examples from your team members and the links I have shared from others in the class. There isn’t one way to do anything, especially in programming! The simplest line of code in JavaScript can be done in 4 ways (x = x + 1; x += 1; x++; ++x;). There are arguments for using each one. (BTW go find some articles on x++, some interesting opinions on that simple line of code. Because of that I avoid it more and more). So, take some time to look at alternatives from your peers. I love the perspective I get from looking at other’s code. I use Edabit.com as an assignment in another class I teach. Once you complete a problem you can look at how others have solved it. Sometimes I think ‘well their way is wrong, mine is elegant’, other times I think ‘wow! What a cool approach’. Be open to new ideas which means you need to be out there looking at those new ideas.

# Perspective

Let’s close with an eternal perspective. One of the underlying principles in the paragraph above may not be obvious. It is revelation. Revelation usually comes after lots of hard work and pondering.

Doctrine and Covenants 9:7,8 Behold, you have not understood; you have supposed that I would give it unto you, when you took no thought save it was to ask me. But, behold, I say unto you, that you must study it out in your mind; then you must ask me if it be right, and if it is right I will cause that your bosom shall burn within you; therefore, you shall feel that it is right.

Oliver’s desire was the same as yours – to understand an unknown language, not just the words but the meaning as well. Invest the time, study it out and you’ll be successful. I have put this to the test many times with Pascal, APL, Fortran, COBOL, Jovial, Assembler, SNOBOL, C, Java, Python, JavaScript, C#, SQL (as far as I can tell revelation does not work with C++, but that just might be me).

God bless you. Have faith in yourself. Your Father in Heaven believes in you. You should as well

Bro James

## Project Examples

It is time to consider your final project. Here are a few samples you can explore. I have NOT validated all of the links. Some may be stale. BTW While you are reviewing the projects you can also get some ideas from their assignments (if the project is for CIT 261 their assignments are different than yours). I’ll share more project ideas next week, but I will drop this list here again for convenience. Also look at last week for the rubric to be used for the final project.

|  |  |
| --- | --- |
| Student | Project URL |
| A Jester | https://jester-andrew.github.io/com |
| Aaron Eicke | https://aaroneiche.github.io/cit261/rover.html |
| Amy Baker | https://shakerbaker78.github.io/amy\_baker\_portfolio/block2/monster\_faces/index.html |
| Ann Halgren | https://homework-website.herokuapp.com/CIT261/ShoppingCart/index.php |
| Anthony DeVito | https://triatomicxeon.github.io./CSS3Animations/solarsystem.html |
| Anthony Wagner | https://mileagetracker-c9240.web.app/mileage-tracker |
| Benjamin Greenway | https://greenway-benjamin.github.io/flashcards/flashcards.html |
| C David | https://desolate-plateau-12188.herokuapp.com/ |
| Cristina Irwin | https://irwin-cristina.github.io/byucit261.github.io/assignments/CristinaIrwinfinal/contact.html |
| D Cenatiempo | https://devins-weather-app.herokuapp.com/ |
| Darlene Chapman | http://webworkshop.site/nasa\_archive.html |
| David Lambert | https://immense-head |
| Elden Chandler | https://ultra-anger.glitch.me/stock.html |
| Elisangela Alves | https://lisbyui.github.io/WDD-330/books/index.html (Links to an external site.) |
| Evan Harrison | https://harrevan.github.io/wdd330/challenge2/index.html (Links to an external site.) |
| Heather | http://heather-marie.com/WeatherNow/ |
| J Froelich | https://knightfalkon.github.io/final/final.html |
| Jake Bassett | http://cit261-assignments.herokuapp.com/assignments/final\_project/app.html |
| Jamie Pascual | https://pascual-jamie.github.io/PascualJamie.github.io/solarsystem.html#display-response |
| Jared Kelley | https://j-a-r-e-d.github.io/wdd330/index.html |
| Jason Reed | https://reed-jason.github.io/projects/tags.html |
| Joshua Nilsson | http://cit261.joshuanilsson.com/topics/project-basics.html |
| Kimi Weldon | https://kiadawel.github.io/WDD330KW/poke\_breeder/ |
| Kristub Plauche | https://kristinaplauche.github.io/cs261/project.html?# |
| Luiz Zonatto | https://zonatto-luiz.github.io/local-storage.html |
| Marvin Enamorado | http://final-project.jma123ssl.com |
| NAI SHIH CHEN | https://naishihchen.github.io/WDD330Portfolio/FinalProject/APODCards.html |
| Nanci Newell | https://nancinewell.github.io/wdd330/ (Links to an external site.) |
| Philip Bowler | https://prbowler.github.io/Readings/Week03/week03.html |
| Ramil Capangpangan | https://ramingsky.github.io/wdd330/project/index.html |
| Roger Pollard | https://rpollard.com/cit261/space.html |
| Steven Milton | https://stevendmilton.000webhostapp.com/Final%20Project/index.html |
| Sunday Onwuchekwa | https://onwuchekwa.github.io/onwuchekwa\_wdd330.github.io/challenge\_02/index.html (Links to an exte) |
| Timothy Bohman | https://shrouded-lowlands-62979.herokuapp.com/movie-db.html |