Converting the API calls to fetch was a lot easier than I thought it was going to be. In my head, transpiling would be a mess of syntax , “thens”, throws, and catches, but instead, I used the structure that you gave in class and completed the first part of the homework in less than 20 minutes. Through it, I learned what the “resolve” and “reject” callbacks do because if I am honest, I had absolutely no clue when we talked about it. By messing around and printing variables to the console while I was trying to get my function to load the variables. properly, I realized that when I called those functions—I was changing the state of my promise from pending to “rejected” or “fulfilled.” I also learned that I have to be careful with my parentheses as I had to parse the return text of the chapter text API with a .text() and not just a .text. Both in class this morning and in the afternoon when I was working through projects, I made mistakes when forgetting that symbol!

Looking through your code, I found out a lot of things. First off, you declare a lot more variables than I do, which made me reflect on how many dang literals I was using in my code! I think I will likely go through and clean that up for my project 2 as I can imagine a lot of cases where there could be a disaster. Where I had trouble passing some things into their own functions instead of cluttering one function, you split each task cleanly and let each function speak for itself in its declaration. This was an especially stark contrast with my code in the creation of different elements in the html page. Finally, I realized that you only ended up with three global variables, and I had left so many of my functions global from when I was debugging! Scoping was a difficult concept to grasp, so I appreciated the inside look at your code to understand how you placed and accessed each of your variables.