Zahava Gopin

Web- 231

11/28/2022

Discussion Board 7.1 – Classes

A JavaScript class is the template for creating objects. Basically, classes are way to enclose data that can later be inputted into the objects. A class constructor is literally the word “constructor” that is automatically added inside of the class. The constructor sets the properties for the object. So, because the class is the template for the object, and an object has properties, the constructor is to properties what the class is to objects.

Some advantages to using classes are that they keep code organized, with every object getting defined properties. The code is compatible and portable. However, I found a lot of cons on the subject as well.

“Classes require you to do a classification: you have to look at all the objects in your application and determine what's in them (a useful thing to do), and then you have to make a taxonomy, figuring out how all those classes are going to be related to each other (what's going to inherit what, what's going to implement what, etc.), and this is usually done at a point in the project when you have the least understanding of these classes actually work, so you almost have to get it wrong. And once you get it wrong, the wrongness starts to propagate into all the upper layers, and you find yourself wishing you had multiple inheritance or other things to help you deal with the fact that your taxonomy is broken. It eventually gets so bad that you have to refactor, which is both labor-intensive and error-prone” (Wood, 2022).

Resources:

*JavaScript Classes*. JavaScript classes. (n.d.). Retrieved November 28, 2022, from https://www.w3schools.com/js/js\_classes.asp

Wood, A. (2022). *We are better off avoiding ES6 classes in JavaScript when possible*. Gist. Retrieved November 28, 2022, from https://gist.github.com/indiesquidge/f8c486795d7dd455c0327ce7e0aa8c16#file-objects-over-classes-md