Nathan Birch

CIT 261

Understandings Portfolio

JavaScript Objects

1/9/2015

I used the same sharing video for each of the elements. That video can be found below [here](https://www.youtube.com/watch?v=yW4cK2-y9pQ&feature=youtu.be).

**Using the JavaScript call method (can mimic inheritance in JavaScript**

This week I was able to study the JavaScript call method. I have seen this method in several places, but I had never used it until my learning this week. I was amazed at how simple it was so trigger inherited attributes by using the call method. There are other ways to mimic inheritance as well, but this was extremely convenient. I loved learning about this. I have put links to several of my pieces of code that I developed while learning about inheritance…and one example from the web.

[JS Objects - Call method 1](https://github.com/nathanbirch/CIT-261-Winter-2016/blob/master/JS%20Objects%20-%20Call%20method%201)

Example from the web: [JS Objects - Call method WEB example](https://github.com/nathanbirch/CIT-261-Winter-2016/blob/master/JS%20Objects%20-%20Call%20method%20WEB%20example)

**Wikipedia Definition of Inheritance**

This week, I was able to study about inheritance. From my understanding, inheritance is only in Object Oriented Programming, and it is when either an object or a class is based on another object or class, meaning it has the same attributes. According to Wikipedia, “Inheritance (OOP) is when an object or class is based on another object (prototypal inheritance) or class (class-based inheritance), using the same implementation (inheriting from an object or class) specifying implementation to maintain the same behavior (realizing an interface; inheriting behavior).”

This can be extremely useful in minimizing the length of code in an application, and in providing a more robust product because you can predict the behavior of inherited objects or classes. It was fun to learn about inheritance this week. I have put links to several examples of my code and learning of inheritance below.

* [JS Objects - Inheritance 1 - typical](https://github.com/nathanbirch/CIT-261-Winter-2016/blob/master/JS%20Objects%20-%20Inheritance%201%20-%20typical)
* [JS Objects - Inheritance 2 - JavaScript](https://github.com/nathanbirch/CIT-261-Winter-2016/blob/master/JS%20Objects%20-%20Inheritance%202%20-%20JavaScript)

Example from the web: [JS Objects - Inheritance WEB example](https://github.com/nathanbirch/CIT-261-Winter-2016/blob/master/JS%20Objects%20-%20Inheritance%20WEB%20example)

**A list of some existing JavaScript objects**

This week I was able to study a lot about various JavaScript objects. I really enjoyed learning about them simply because JavaScript is a very diverse programming language with many functionalities that I am not yet familiar with. As I studied this however, I was exposed to new built in features in JavaScript and helpful objects and functions or methods that are present in the programming language. I developed several pieces of code for this understandings portfolio. They are listed below:

* [JS Objects - Existing Objects - Date](https://github.com/nathanbirch/CIT-261-Winter-2016/blob/master/JS%20Objects%20-%20Existings%20Objects%20-%20Date)
* [JS Objects - Existing Objects - Push](https://github.com/nathanbirch/CIT-261-Winter-2016/blob/master/JS%20Objects%20-%20Existing%20Objects%20-%20Push)
* [JS Objects - Existing Objects - Substr](https://github.com/nathanbirch/CIT-261-Winter-2016/blob/master/JS%20Objects%20-%20Existing%20Objects%20-%20Substr)

Example from the web: [JS Objects - Existing Objects - WEB example](https://github.com/nathanbirch/CIT-261-Winter-2016/blob/master/JS%20Objects%20-%20Existing%20Objects%20-%20WEB%20example)

**JavaScript Objects**

This week I was able to study quite a bit about JavaScript Objects as a whole. I learned about their benefit and how to work with JavaScript Objects in a convenient way. I read about how they came to be, and why they are so good. In the website referred to me by my instructor I read, “Objects work so well because they act just like real life objects- objects have properties and methods. So if we were talking about a lamp, a property of it may be its height or width, say 12cm. A method of it may be to shine (an action). And when it's shining, its brightness property would be of a greater value than when it wasn't.” And this is completely true. In my work, I deal with all sorts of objects on a daily basis, and they are truly the only efficient way to go. I have included several examples of my code and my learning below. In the first example, it is an object of an employee’s timesheet before they have entered any time. After they enter time, most of those fields will be populated. In the second example, it is also an empty object, with many attributes, and the third example uses it to edit a user’s mission that they have already entered into the system.

* [JS Objects - JavaScript Objects 1](https://github.com/nathanbirch/CIT-261-Winter-2016/blob/master/JS%20Objects%20-%20JavaScript%20Objects%201)
* [JS Objects - JavaScript Objects 2 - add mission empty object](https://github.com/nathanbirch/CIT-261-Winter-2016/blob/master/JS%20Objects%20-%20JavaScript%20Objects%202%20-%20add%20mission%20empty%20object)
* [JS Objects - JavaScript Objects 3 edit mission example](https://github.com/nathanbirch/CIT-261-Winter-2016/blob/master/JS%20Objects%20-%20JavaScript%20Objects%203%20edit%20mission%20example)

Example from the web: [JS Objects - JavaScript Objects WEB example](https://github.com/nathanbirch/CIT-261-Winter-2016/blob/master/JS%20Objects%20-%20JavaScript%20Objects%20WEB%20example)

**What is Object Oriented Programming?**

This week I was able to study more about Object Oriented Programming. I have done quite a bit of Object Oriented Programming between my work and school, but I was still able to learn quite a bit and acknowledge that I still have much to learn. I was able to learn several things that I wanted to share. The Lynda video provided to me by my instructor was most helpful in comparing what life would be like if it weren’t for object oriented programming. I imagined not having modularized code, and all of it being in the exact same place…no organization except by sequence, no reusable code…having to copy and paste functionality that I needed to duplicate elsewhere. Thank goodness for object oriented programming. ☺ I also developed several pieces of code that show the benefits of modularizing code, and using objects in an application. My pieces of code are below:

* [JS Objects - OOP 1 - Save Address Example](https://github.com/nathanbirch/CIT-261-Winter-2016/blob/master/JS%20Objects%20-%20OOP%201%20-%20Save%20Address%20Example)
* [JS Objects - OOP 2 - Post Address](https://github.com/nathanbirch/CIT-261-Winter-2016/blob/master/JS%20Objects%20-%20OOP%202%20-%20Post%20Address)
* [JS Objects - OOP 3 - Post Address Service](https://github.com/nathanbirch/CIT-261-Winter-2016/blob/master/JS%20Objects%20-%20OOP%203%20-%20Post%20Address%20Service)
* [JS Objects - OOP 4 - Delete Address Example](https://github.com/nathanbirch/CIT-261-Winter-2016/blob/master/JS%20Objects%20-%20OOP%204%20-%20Delete%20Address%20Example)

Example from the web: [JS Objects - OOP WEB example](https://github.com/nathanbirch/CIT-261-Winter-2016/blob/master/JS%20Objects%20-%20OOP%20WEB%20example)

**JavaScript objects as associative arrays**

I have been studying JavaScript objects as associative arrays. As I studied associative arrays in and of themselves I came to the understanding that they are very broad elements and are everywhere in JavaScript code. According to my understanding, an associative array is just a set of key value pairs. Each ‘key’ is stored with a value (or is ready for a value to be entered by the user in some cases). Associative arrays allow you to access these values through the keys using either numbers or strings by association. I think these are really cool because you can pick out specific elements in an array based on string matching. It can be very convenient. I have developed several pieces of code that I wanted to share in conjunction with associative arrays. The first is:

* [JS Objects – Associative Arrays](https://github.com/nathanbirch/CIT-261-Winter-2016/blob/master/JS%20Objects%20-%20Associative%20Arrays)
* [JS Objects – Associative Arrays – Months Example](https://github.com/nathanbirch/CIT-261-Winter-2016/blob/master/JS%20Objects%20-%20Associative%20Arrays%20-%20Months%20Example)

Example from the web: [JS Objects - Associative Arrays WEB example](https://github.com/nathanbirch/CIT-261-Winter-2016/blob/master/JS%20Objects%20-%20Associative%20Arrays%20WEB%20example)