Week 3 Prompts – Jakub

Select five methods that can be used on an Array and describe the following for each: 1) what the method signature is, 2) what the method does, and 3) why would this method be useful (how could you use it)?

What is the difference between == and ===?

What is a closure and how does it work? Provide an example.

What is your favorite thing you learned this week?

**5 array methods**

array.sort()-This will sort your array into alphabetical order. Useful if you have a string of names you would like to organize alphabetically.

array.reverse()-Like sort this will reverse an array. So if .sort does A-Z .reverse would return A-Z

array.push()- This is used to add something to the array so that you don’t have to continually rewrite the variable and array. You can push as many things as you want and this will add to the end of the array.

array.splice() – Related to .push, .splice allows to alter the contents of an array by adding or removing elements inside. To do this, within the paretheses of .splice() you specify two numbers. The first one is the index and specifies where to begin removal and the second number specifies how many elements to remove. So for example (1,1) would mean remove just the second element of an array. In addition to the removal you can also specify to add an element at the same time.

array.filter() – This allows you to run a boolean value on an array and returns an array that passes the value. For example logging on names that are above six letters or that contain an even or odd number of letters.

== is the equality operator while === is strict equality. Strict equality is, as the name implies a stricter version of equality. Not only does it consider value but it also considers type. So for example

“Jakub”== Jakub would return TRUE, while.

“Jakub”=== Jakub would return false, as one is a string and the other is not.

I am a little fuzzy on closure but as far as I understand closures refers to the ability for functions to reference information not defined within the scope of itself. Useful when having inner and outer functions. This can allow you to have something defined globally in JS and then locally within a particular function. Clunky example below:

function outer(){

    var num1=6;

    function inner(){

         console.log(num1+1)

    }

   inner()

}

outer()

Result should be 7.

I think the various array methods were fun to learn or to get introduced too as they allow not shortcuts per se, but allow to manipulate information in arrays in convenient ways. I.e. like not rewriting an array every time I want to add or remove an element. Learning more of these methods will be fun.