Mark McCreery

Front-End Cohort 7.27.22

August 17, 2022

Week 3 Prompts

The concat() method (let array1 = [a,b]; let array2 = [c,d]; let bothArrays = array1.concat(array2)) concatenates two arrays and creates a new array from the elements of both arrays. This would be useful if you wanted an array that contains both sets of information, but you didn't want to call both functions each time to reference. The push() method (array1.push("example");) allows for a new element to be added to the end of the array which is useful if you need to include an additional element inside the array. Likewise, the pop() method (array1.pop("example");) does the opposite by removing the last element in an array. The splice() method allows for the addition and removal of elements in an array in regard to its placement. This method is useful because it allows you to skip extraneous additions and removals of data within the array. The signature would look like array.splice(what index to start from, how many elements to remove, elements to add); in example, array1.splice(0,1,d,c). Finally, the slice() method allows for the creation of a new array from elements of a preexisting array. For example, array3 = array1.slice(1) in which case array3 = [d] referencing the array mentioned above. This would useful if you only wanted certain parts of the array to use later.

How a closure works is that when you create a function within a function, it allows the inner function to access commands that are outside the of the scope of the function that it is contained within. Using an example from MDN, the closures would be add5 and add10 because

they retroactively work on the code that is outside of the scope of the function. It is also noteworthy with closures that they create their own pocket of code in which it doesn't affect the surrounding

```
function makeAdder(x) {
  return function (y) {
    return x + y;
  };
}

const add5 = makeAdder(5);
const add10 = makeAdder(10);

console.log(add5(2)); // 7
console.log(add10(2)); // 12
```

code.

Works Cited

"Closures - Javascript: MDN." *JavaScript | MDN*, https://developer.mozilla.org/en-US/docs/Web/JavaScript/Closures#practical_closures.

JavaScript Array Methods, https://www.w3schools.com/js/js_array_methods.asp.