Discussion Board 8.1 – Dealing with Arrays

Eric McCool

Bellevue University - College of Science and Technology

Web231-339A Enterprise JavaScript I

Professor Krasso

December 7, 2022

This week we look at arrays and ways to iterate through them using loops. We look at for, for/in, and for/of, as well as the while and do-while loops. Of them all, I like the simple for-loop for its utility.

There are many ways to find what you are looking for in an array. The for/in loop is designed to read values from an object. The for/of loops through the values of an iterable object. That almost sounds the same. While loops are great, especially when you need to absolutely execute some code at least once before it is compared (W3S, 2022). It turns out there are numerous built-in methods for working with arrays, such as forEach() and indexOf(). These are all prepackaged, and while they work great when needed, sometimes they are not 'exactly.' This is the advantage of the for-loop. It is the bare-bones loop. All of the others can be built from this one loop. It is highly customizable. The drawback, it is not prepackaged. It involves customizing it oneself. This can create more code, reduce readability, and open the potential for more errors. Performance can be affected in very long load-times, and reducing code is a prime goal. Readability also suffers if a complex for-loop takes too much time to muddle through. The following is a simple for loop and one of it's compatriots for comparison.

```
Let txt = "";

const states = [imagine an array of 50 state names];


for (let x = 0, x < 51, x++) {

        txt += states[x];      => prints out the states by index

or,


states.forEach(someFunction);

function someFunction(value) {

        txt = += value          => prints out the states by value
```

One is easier to read, but the other is more recognizable. One offers modularity, the other is scratch-built every time. I encourage using the built-in functions. I just like the for loop because it is the granddaddy, and when worse comes to worse, it will be the fallback when the other options aren't working.

References

W3S, 2022. *"JS Arrays, JS Array Methods, JS Array Iteration"* JS Tutorial – Arrays.

W3School.com. W3Schools, Larsamyraa 18, 4313 SANDNES, Norway.

https://www.w3schools.com/js/js_arrays.asp