

CMPS 2680 Homework 7

Spring 2024

Javascript Arrays

- [Introduction](#)
- [Array Methods \(Functions\)](#)
- [Using const](#)

More About Arrays

- It is worth mentioning that arrays in Javascript are dynamically sized by default. It is not necessary to initialize an array with a specific size. You can create an empty array like this:

```
const values = [];
```

However, this will make the "length" of the array 0, so to add elements you should use the push method.

- Since Javascript is loosely typed, one array can hold values of different types:

```
const stuff = [ 43, "Bob", 1.337, false ];
```

Javascript for loop

- [Introduction](#)
- As a general rule, for loops should be used any time you know the exact number of times you want to loop. For example, to print "hello" to the console three times:

```
for (let i = 0; i < 3; i++){  
  console.log("hello");  
}
```

- The variable i can be accessed inside of the loop. For example, this will print the numbers 1 through 10 to the console:

```
for (let i = 1; i <= 10; i++){  
  console.log(i);  
}
```

Looping Through Arrays

- To loop through each item in an array, you can use the `array.length` property, just remember that the index for the last array value is the array length minus 1. For example, to iterate through an array called "names" and print all of the values to the console:

```
const names = [ "Bob", "Sue", "Angela", "Frank" ];  
  
for (let i = 0; i < names.length; i++){  
  console.log(names[i]);  
}
```

- You can also iterate through an array in reverse:

```
const names = [ "Bob", "Sue", "Angela", "Frank" ];  
  
for (let i = names.length - 1; i >= 0; i--){  
  console.log(names[i]);  
}
```