1. Introduction

function greet() {

console.log('Functions are cool!');

}

1. First-Class Functions

Functions are First-Class Functions

In JavaScript, functions are first-class functions. This means that you can do with a function just about anything that you can do with other elements, such as numbers, strings, objects, arrays, etc. JavaScript functions can:

Be stored in variables

Be returned from a function.

Be passed as arguments into another function.

1. Callbacks

A function that is passed as an argument into another function is called a callback function.

So nameLengths will be a new array: [5, 7, 8]. Again, it is important to understand that the map() method returns a new array; it does not modify the original array.

const musicData = [

{ artist: 'Adele', name: '25', sales: 1731000 },

{ artist: 'Pentatonix', name: 'A Pentatonix Christmas', sales: 904000 }];

const albumSalesStrings = musicData.map(function(mData) {

return mData.name + ' by ' + mData.artist + ' sold ' + mData.sales + ' copies';

});

console.log(albumSalesStrings);

const results = musicData.filter(function(mData) {

return mData.name.length >= 10 && mData.name.length <= 25;

});

Array methods, such as forEach(), map(), and filter(), take advantage of callbacks to execute functions onto a given array's elements.

1. Scope

A function's runtime scope describes the variables available for use inside a given function. The code inside a function has access to:

1. The function's arguments.
2. Local variables declared within the function.
3. Variables from its parent function's scope.
4. Global variables.

JavaScript is Function-Scoped

1. Closures

Functions Retain Their Scope

A closure is a combination of a function and it’s lexical environment.

function expandArray() {

let myArray = [1, 1, 1];

return function() {

myArray.push(1);

return myArray; }}

1. Immediately-Invoked Function Expressions (IIFE)

An immediately-invoked function expression, or IIFE (pronounced iffy), is a function that is called immediately after it is defined.

(function sayHi(){

alert('Hi there!');

}

)();

// alerts 'Hi there!'

One of the primary uses for IIFE's is to create private scope (i.e., private state). Recall that variables in JavaScript are traditionally scoped to a function. Knowing this, we can leverage the behavior of closures to protect variables or methods from being accessed!

1. Lesson Summary

An immediately-invoked function expression (IIFE) is a function that is called immediately after it is defined. Utilizing an IIFE alongside closures allows for a private scope, which maintains privacy for variables defined within them. And since less variables are created, an IIFE will help to minimize pollution of the global environment, hindering the chances of variable name collisions.