**Callbacks**

**Working with Callbacks**

**Learning Outcome Addressed**

1. Use callback functions to filter, sort and map arrays and objects in JavaScript

To fully grasp the purpose of callback functions, you have to first learn about asynchronous programming. In JavaScript, a code is executed sequentially from top to bottom. However, in some cases, a code needs to run **only** after something else has happened (not sequentially). This is called asynchronous programming.

Callbacks make sure a function does not run until a specific task has been completed. For example, you can have a loading icon displayed on your page while everything loads up, then have a callback function that hides the loading icon once the page is ready.

**Creating a callback function**

In JavaScript, a callback function is passed into another function as a parameter. The call back function is later called inside of the function it was passed to, once that function is fired.

For example:

var callMeBack = function() {  
    console.log("I'm a callback!");  
}  
   
setTimeout(callMeBack, 5000);

In this example, the function callMeBack is called after five seconds have elapsed. In other words, the function is only invoked after something has happened.

**Let's practice callbacks**

Your task in this activity is to create a function called filterArray that take two arguments:

* myArray: An array of numbers.
* isEvenCallBack: A function that returns the Boolean true if a number is even, and returns false if it's odd.

Your task is to implement the isEvenCallBack function and then pass it as a callback function into filterArray. The function filterArray should then return an array containing only even numbers.

*Hint: There are comments in the callbacks.js file to help you through this task*

*Check the results in your console. If your code works, it should print [2, 6, 12]*

Task

create a function called filterArray that take 2 arguments:

* myArray: this is an array of numbers.
* isEvenCallBack: this is a function that returns the boolean true if a number is even, and returns false if it's an odd number.
* // This is the array that contains numbers for you to work with
* var myArray = [1, 2, 5, 6, 12, 23, 15, 31];
* // This array should only contain even numbers
* var evenArray = [];
* // This code will call your function when the page loads up
* // Don't edit this function!
* window.onload = () => {
* console.log(filterArray(myArray, isEvenCallBack));
* };
* function filterArray(myArray, isEvenCallBack) {
* // TODO: use filterArray to determine if a number is even or odd.
* //If the number is even, add it to the array 'evenArray'
* return evenArray;
* }
* // This function should return 'true' if a number is even and 'false' if a number is odd
* function isEvenCallBack(number) {
* // TODO: use the mod operator (%) to determine if number is even or odd
* }
* // Do not edit code past this point
* if (typeof module !== 'undefined') {
* module.exports = { filterArray, isEvenCallBack };
* }