**JavaScript Callback**

In JavaScript, functions are first-class objects; that is, functions are of the type Object and they can be used in a first-class manner like any other object (String, Array, Number, etc.) since they are in fact objects themselves. They can be “stored in variables, passed as arguments to functions, created within functions, and returned from functions”

JavaScript Higher-order Functions, aka Callback Functions

Because functions are first-class objects, we can pass a function as an argument in another function and later execute that passed-in function or even return it to be executed later. This is the essence of using callback functions in JavaScript.

In JavaScript, functions are objects. Because of this, functions can take functions as arguments, and can be returned by other functions. Functions that do this are called higher-order functions. Any function that is passed as an argument and subsequently called by the function that receives it, is called a callback function.

Example:

let x = function () {

console.log (“I am called from inside a function”);

}

let y= function (callback) {

Console.log (“Do something”);

}

y (x);