# Index.html

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>Module 4 Solution Starter</title>

<script src="SpeakHello.js"></script>

<script src="SpeakGoodBye.js"></script>

<script src="script.js"></script>

</head>

<body>

<h1>Module 4 Solution Starter</h1>

</body>

</html>

# Script.js

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// START HERE IF YOU WANT AN EASIER STARTING POINT FOR THIS ASSIGNMENT

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//

// Module 4 Assignment Instructions.

//

// The idea of this assignment is to take an existing array of names

// and then output either Hello 'Name' or Good Bye 'Name' to the console.

// The program should say "Hello" to any name except names that start with a "J"

// or "j", otherwise, the program should say "Good Bye". So, the final output

// on the console should look like this:

/\*

Hello Yaakov

Good Bye John

Good Bye Jen

Good Bye Jason

Hello Paul

Hello Frank

Hello Larry

Hello Paula

Hello Laura

Good Bye Jim

WARNING!!! WARNING!!!

The code does NOT currently work! It is YOUR job to make it work

as described in the requirements and the steps in order to complete this

assignment.

WARNING!!! WARNING!!!

\*/

// STEP 1: (NOTHING TO DO. ALREADY DONE FOR YOU)

// Wrap the entire contents of script.js inside of an IIFE

// See Lecture 52, part 2

// (Note, Step 2 will be done in the SpeakHello.js file.)

(function () {

var names = ["Yaakov", "John", "Jen", "Jason", "Paul", "Frank", "Larry", "Paula", "Laura", "Jim"];

// STEP 10: (NOTHING TO DO. ALREADY DONE FOR YOU)

// Loop over the names array and say either 'Hello' or "Good Bye"

// using the 'speak' method or either helloSpeaker's or byeSpeaker's

// 'speak' method.

// See Lecture 50, part 1

for (var i = 0; i < names.length; i++) {

// STEP 11: (NOTHING TO DO. ALREADY DONE FOR YOU)

// Retrieve the first letter of the current name in the loop.

// Use the string object's 'charAt' function. Since we are looking for

// names that start with either upper case or lower case 'J'/'j', call

// string object's 'toLowerCase' method on the result so we can compare

// to lower case character 'j' afterwards.

// Look up these methods on Mozilla Developer Network web site if needed.

var firstLetter = names[i].charAt(0).toLowerCase();

// STEP 12: (NOTHING TO DO. ALREADY DONE FOR YOU)

// Compare the 'firstLetter' retrieved in STEP 11 to lower case

// 'j'. If the same, call byeSpeaker's 'speak' method with the current name

// in the loop. Otherwise, call helloSpeaker's 'speak' method with the current

// name in the loop.

if (firstLetter === 'j') {

byeSpeaker.speak(names[i]);

} else {

helloSpeaker.speak(names[i]);

}

}

})();

# SpeakGoodBye.js

// NOTE! The steps in this file are basically identical to the ones you

// performed in the SpeakHello.js file.

// STEP 6: Wrap the entire contents of SpeakGoodBye.js inside of an IIFE

// See Lecture 52, part 2

// STEP 7: Create an object, called 'byeSpeaker' to which you will attach

// the "speak" method and which you will expose to the global context

// See Lecture 52, part 1

// var byeSpeaker =

// DO NOT attach the speakWord variable to the 'byeSpeaker' object.

var speakWord = "Good Bye";

// STEP 8: Rewrite the 'speak' function such that it is attached to the

// byeSpeaker object instead of being a standalone function.

// See Lecture 52, part 2

function speak(name) {

console.log(speakWord + " " + name);

}

// STEP 9: Expose the 'byeSpeaker' object to the global scope. Name it

// 'byeSpeaker' on the global scope as well.

// xxxx.xxxx = byeSpeaker;

# SpeakHello.js

// STEP 2: Wrap the entire contents of SpeakHello.js inside of an IIFE

// See Lecture 52, part 2

// STEP 3: Create an object, called 'helloSpeaker' to which you will attach

// the "speak" method and which you will expose to the global context

// See Lecture 52, part 1

// var helloSpeaker =

// DO NOT attach the speakWord variable to the 'helloSpeaker' object.

var speakWord = "Hello";

// STEP 4: Rewrite the 'speak' function such that it is attached to the

// helloSpeaker object instead of being a standalone function.

// See Lecture 52, part 2

function speak(name) {

console.log(speakWord + " " + name);

}

// STEP 5: Expose the 'helloSpeaker' object to the global scope. Name it

// 'helloSpeaker' on the global scope as well.

// See Lecture 52, part 2

// (Note, Step 6 will be done in the SpeakGoodBye.js file.)

// xxxx.xxxx = helloSpeaker;

## CONTENIDO

[Index.html 1](#_Toc45626363)

[Script.js 2](#_Toc45626364)

[SpeakGoodBye.js 4](#_Toc45626365)

[SpeakHello.js 5](#_Toc45626366)