

Scope Lab

- Due No Due Date
- Points 1
- Submitting a website url



[\(https://github.com/learn-co-curriculum/phase-1-scope-lab\)](https://github.com/learn-co-curriculum/phase-1-scope-lab)



[\(https://github.com/learn-co-curriculum/phase-1-scope-lab/issues/new\)](https://github.com/learn-co-curriculum/phase-1-scope-lab/issues/new)

Learning Goals

- Understand how functions determine scope in JavaScript

Introduction

Scuber's drivers are beginning to take down some notes about their customers. Use your knowledge of scope to access and change some of these variables.

Instructions

A word of warning. In this lab, you will practice doing some bad things. For example, we'll ask you to declare variables in global scope, and write functions that break down. You'll also need to use `var` to declare all variables except when specifically instructed otherwise. We do this because, by testing the limits and exposing the underbelly of a language, you can better understand it and know what to avoid in the future.

Fork and clone this lab into your local environment. Navigate into its directory in the terminal, then run `code .` to open the files in Visual Studio Code.

Ok, these are your tasks:

- `customerName` : Declare a variable in global scope called `customerName` using the `var` keyword and assign it the value `'bob'` .
- `upperCaseCustomerName()` : Write a function that accesses that global `customerName` variable, and uppercases it.
- `setBestCustomer()` : Write a function that when called, declares a variable called `bestCustomer` in global scope and assigns it to be `'not bob'` . (Poor Bob.) Also, poor us, declaring a global variable from inside a function is one of those things we would never want to do normally, but is good for us to explore right now.

- `overwriteBestCustomer()` : See the consequences of declaring a variable in global scope, by writing a new function called `overwriteBestCustomer()` that changes that `bestCustomer` variable to '`'maybe bob'`' .
- `leastFavoriteCustomer` and `changeLeastFavoriteCustomer()` : Now declare a constant (using the `const` keyword) in global scope called `leastFavoriteCustomer` ; be sure to assign it some initial value. Finally, write a function called `changeLeastFavoriteCustomer()` that attempts to change that constant - notice what JavaScript does when you try to change the constant.

When you're done, remember to commit and push your changes up to GitHub, then submit your work to Canvas using CodeGrade.