

# CSCE 120: Learning To Code

## Manipulating Data I Hacktivity 7.3

### Introduction

Prior to engaging in this hacktivity, you should have completed all of the pre-class activities as outlined in this module. At the start of class, you will be randomly assigned a partner to work with on the entirety of this hacktivity as a *peer programming* activity. Your instructor will inform you of your partner for this class.

One of you will be the driver and the other will take on the navigator role. Recall that a driver is in charge of the keyboard and computer while the navigator is in charge of the handout and directing the activity. However, you are *both* responsible for contributing and discussing solutions. If you were a driver/navigator in the prior activity, switch roles for this activity.

You will use the same project from the previous Hacktivity. If you need to, you can re-download it from GitHub using the URL, <https://github.com/cbourne/jqueryProject>.

### Making an App

We have provided a simple HTML document in `AlbumApp/index.html` that contains a simple web form and a table containing information about your favorite albums. You will complete this application by adding functionality that will allow a user to enter information about their favorite albums, click the “Add Album” button and it will add it to the end of the table.

Add code to the `addAlbum()` function in the JavaScript file, `AlbumApp/album.js` to achieve the following:

1. Pull the data values (album title, band, release year) from the web form and store them into variables.

2. Add code to *validate* the data: think of any invalid input values and give the user an indication of an error. Hints:
  - We’ve provided an `addErrorBox()` function to help you (figure out how you can use it).
  - If the input is an error, the function should `return` without doing any other action (aside from raising an error).
3. Add code that determines the number of rows already in the table so that the next row’s first column value will be the next available number.
4. Add code that creates a new HTML row element with the album’s number, its data (title, band, year) and a “delete” button in the last column (use the hard-coded HTML as a guide).
5. Add code to append the new row to the table in the HTML document.
6. Add code to animate the new row so that it fades in (or some other effect); full documentation on all effects can be found here: <http://api.jquery.com/category/effects/>
7. Add some additional code to clear out the form data after each entry (error or not) so that the user doesn’t have to erase it to enter another album.
8. Make the delete buttons functional. If someone clicks them, it calls a function that passes the button element to the function. Use the `parent()` function to traverse up the DOM to the row element. Fade the row out and remove it.