

PROG 2700 D3/C3

Client-Side Programming

ASSIGNMENT 5 (PARTS A, B, & C) – SASS/SCSS,

Part A: FreeCodeCamp

<https://learn.freecodecamp.org/>

Sign up with your Github account.

Complete the Tutorial Steps in the '*Bootstrap*', '*jQuery*' and '*Sass*' Portions under '*Front End Libraries Certification*'.

Part B: FreeCodeCamp & C3/D3 Charts of the Friends TV Show

PREREQUISITE (More FreeCodeCamp):

Complete the Tutorial Steps in the '*Data Visualization with D3*' section under '*Data Visualization Certification*'.

INSTRUCTIONS

We will be revisiting the Friends API from Assignment 3B. You can use your code from Assignment 3B as a starting point or use the Assignment 3B starting files included with Assignment 3B in Brightspace, if you prefer. As a reminder the Friends API data can be accessed from the following link:

[http://api.tvmaze.com/shows/431?embed\[\]=episodes&embed\[\]=cast](http://api.tvmaze.com/shows/431?embed[]=episodes&embed[]=cast)

There will be an explanatory video next to this assignment in BrightSpace. **Please watch that video.** The goal will be to use the D3/C3 JavaScript Data Manipulation/Visualization libraries and Map, Filter, and Reduce array functions (so avoid loops) to manipulate and visualize the Friends API data.

You should try to produce the first two graphs/charts demonstrated in the video as closely as possible and then either a reproduction or equivalent Time Series chart with delay loaded data, as the third one demonstrated. Then you will add two more charts/graphs based on Friends data of your choosing from the API for a **total of 5 charts/graphs**.

Part C: Chosen API + C3/D3

Summary

- Your application will retrieve its data from the JSON API of your choice. Please refer to Todd Motto's list of a publicly available APIs (<https://github.com/toddmotto/public-apis>) for some inspiration. You will want to pick an API with “dense” enough data to make several charts and/or tables interesting. **It should not be an API used for another assignment in this course (i.e. no Deck of Cards, Flight/Bus api) but could be the same original API you picked yourself back in Assignment 2B.**
- You will manipulate the data via appropriate Array Functions and limit the usage of loops
- You will create **at least 3** interesting C3/D3 charts and tables on top of the data
- **At least one of the charts will contain delay-loaded data.**

Notes

- Restrict your API choice to ones whose Auth type is listed as either 'No' or 'apiKey'. Refrain from selecting an API requiring an OAuth authentication.
 - If you choose one that uses an api Key, it is likely that you'll have to register with the api provider in order to get a key to use. If this proves difficult to figure out, choose an API with 'No' as the Auth Type value.
- Restrict your API choice to ones whose CORS value is **not** 'No'.

General Requirements (36 points)

REQ-001 RETRIEVAL OF THE JSON API DATA (9 PTS.)

The application is retrieving all of its data from a JSON API of the student's choice.

REQ-002 USE OF ARRAY FUNCTIONS TO MANIPULATE DATA (9 PTS.)

The application will feature minimal usage of loops, but will rather utilize **.map**, **.filter**, and **.reduce** array functions to manipulate data retrieved from the API.

REQ-003 THREE OR MORE C3/D3 CHARTS AND/OR TABLES PRESENT (9 PTS.)

Three or more distinct and unique C3/D3 charts/tables will be generated within the website on top of the data retrieved from the chosen API.

REQ-004 “DELAY-LOADED” DATA IN CHARTS (9 PTS.)

At least one chart will have data that loads after a delay or the chart itself will change after a set interval of time, without needing user interaction,.

Architecture Requirements (6 points)**REQ-005 WRAP THE ENTIRE APPLICATION IN AN INDEPENDENTLY INVOKED FUNCTION EXPRESSION (IIFE) (OR EQUIVALENT) (3 PTS)**

In order for the entire application to be contained within its own scope and to not pollute the global scope, wrap the entire contents of the file in an Independently Invoked Function Expression (IIFE) or Equivalent routine and be prepared to demonstrate how your script's data is contained within its own local scope and not within the browser's global scope (window).

REQ-006 UNOBTRUSIVE JAVASCRIPT. (3 PTS.)

The JavaScript will all be unobtrusive (i.e. little or no JavaScript present in the HTML page).

Instructions

1. **Don't forget that a Code Review demonstration of your code is a necessary part of this assignment. You MUST complete the code explanation/code review part of the video submission checklist to get credit for the assignment. Part of the assessment will include your ability to speak about the code you wrote, even if it doesn't completely work or do what you expect. You do not need audio or to speak during the rest of the video, but it is required for the code review section as indicated in the checklist.**
2. **Late submissions will be subject to the late penalties laid out in the course outline.**

Academic Integrity and Plagiarism

Code sharing by any means is considered plagiarism and is strictly forbidden under the NSCC Academic Integrity policy.

[NSCC ACADEMIC INTEGRITY GUIDELINES](#)
[NSCC ACADEMIC INTEGRITY REPORTING POLICY](#)

PROG2700: Assignment Five - Video Submission Checklist

Part A: FreeCodeCamp	<p>Show the following, <u>if not already given credit for Part A</u>:</p> <ul style="list-style-type: none"> <input type="checkbox"/> in the Browser log into Code Academy and show completion of tasks
Part B: FreeCodeCamp & C3/D3 Charts of the Friends TV Show	<p>Show the following:</p> <ul style="list-style-type: none"> <input type="checkbox"/> in the Browser log into Code Academy and show completion of tasks <input type="checkbox"/> the program running and displays <u>requested</u> Chart #1 in browser <input type="checkbox"/> the program running and displays <u>requested</u> Chart #2 in browser <input type="checkbox"/> the program running and displays <u>requested</u> Chart #3 in browser <input type="checkbox"/> the program running and displaying <u>chosen</u> Chart #4 in the browser <input type="checkbox"/> the program running and displaying <u>chosen</u> Chart #5 in the browser <input type="checkbox"/> the program displays at least one chart with delay-loaded data <input type="checkbox"/> show that the code uses no loop structures of any kind
Part C: Chosen API + C3/D3	<p>Show the following:</p> <ul style="list-style-type: none"> <input type="checkbox"/> the program running and displays <u>chosen</u> Chart #1 in browser <input type="checkbox"/> the program running and displays <u>chosen</u> Chart #2 in browser <input type="checkbox"/> the program running and displays <u>chosen</u> Chart #3 in browser <input type="checkbox"/> the program displays at least one chart with delay-loaded data <input type="checkbox"/> show that the code is not in global scope (e.g. IIFE usage) <input type="checkbox"/> show the code to retrieve data from the chosen API <input type="checkbox"/> show that the code uses no loop structures of any kind <input type="checkbox"/> citations for any code samples used
Code Review: <u>Mandatory</u>	<p>Show the following:</p> <ul style="list-style-type: none"> <input type="checkbox"/> explain the code in detail for <u>the entire application for 5C</u> (i.e. querying the API, filtering and preparing the data, charting it using C3/D3) (needs audio)