

CA1 – Building a Chart Classes in JavaScript

**Alice Corry**

**Your student number is N00211635**

Creative Coding 1

Year 2 2022-23

DL836 BSc (Hons) in Creative Computing

Link to resources created as part of the project.

| GitHub | <https://github.com/ac-png/Creative-Coding---CA1.git> |
| --- | --- |
| Report | <https://docs.google.com/document/d/1qRQzsab96zymYvsq4mtj1NYyehKxbSDYeGRY7HWktdk/edit?usp=sharing> |

Table of Contents

[1. Configurable User Options](#_eyytwyws84ag)

[2. Code Organisation](#_bgz1kq2lnmil)

[2.1. Indentation](#_zb2mg3hc5cge)

[2.2. Visual Studio Code Extensions](#_qasmejk45hek)

[3. Object-Oriented Programming](#_ogfp2qlf25ut)

[3.1. Classes](#_1q1zl71oxlzn)

[3.2. Objects](#_ovf8aop5s3fn)

[4. Use of Functions to Reduce Repetition and Improve Clarity](#_3bmsi374naqh)

[4.1. drawChart() Function](#_1uxey6hlj4g4)

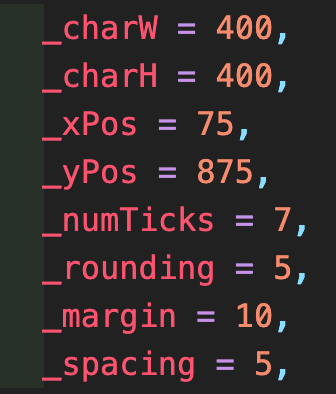
[4.2. Calculation Properties](#_pbvc98nqm6xg)

# 

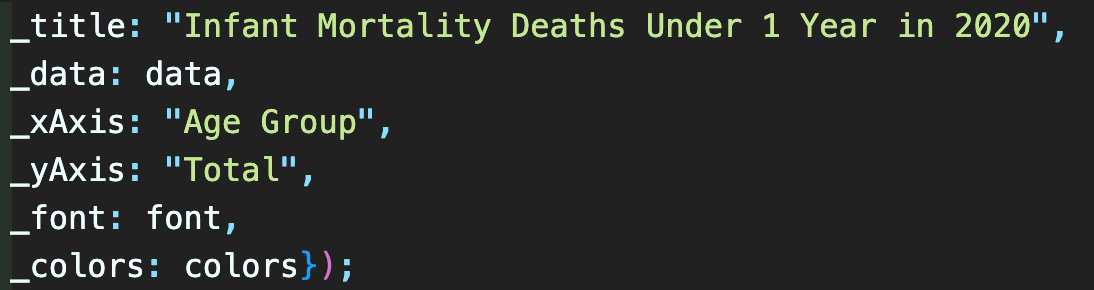
# Configurable User Options

I did my best throughout the project to ensure that almost any data was not hard-coded into the class. I ensured that any variable that included an integer or was taken from a CSV file was also a parameter so that the user could quickly change the values without going into the class.

When defining the parameters in the classes, I also added default values for most of the parameters, which means that the chart would still work even if the user did not enter particular values (such as chart height, number of tickets, etc.), but that they could also change the values if required.



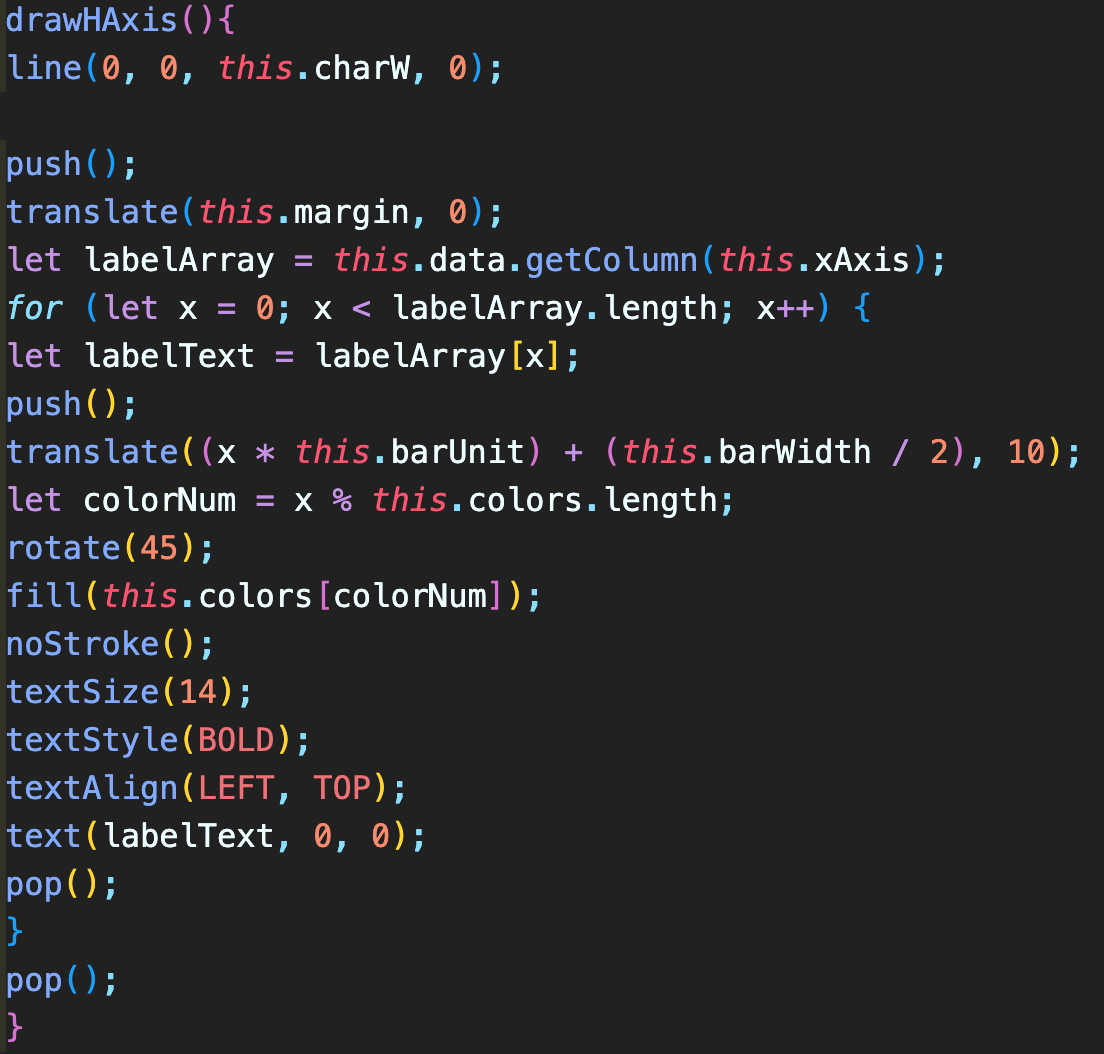
Although I added default values for most parameters, I decided some values did not need defaults. For example, I didn't want the classes to have default data because that would confuse the code when the user made a new chart using their own data. After all, most data has different headers and names.



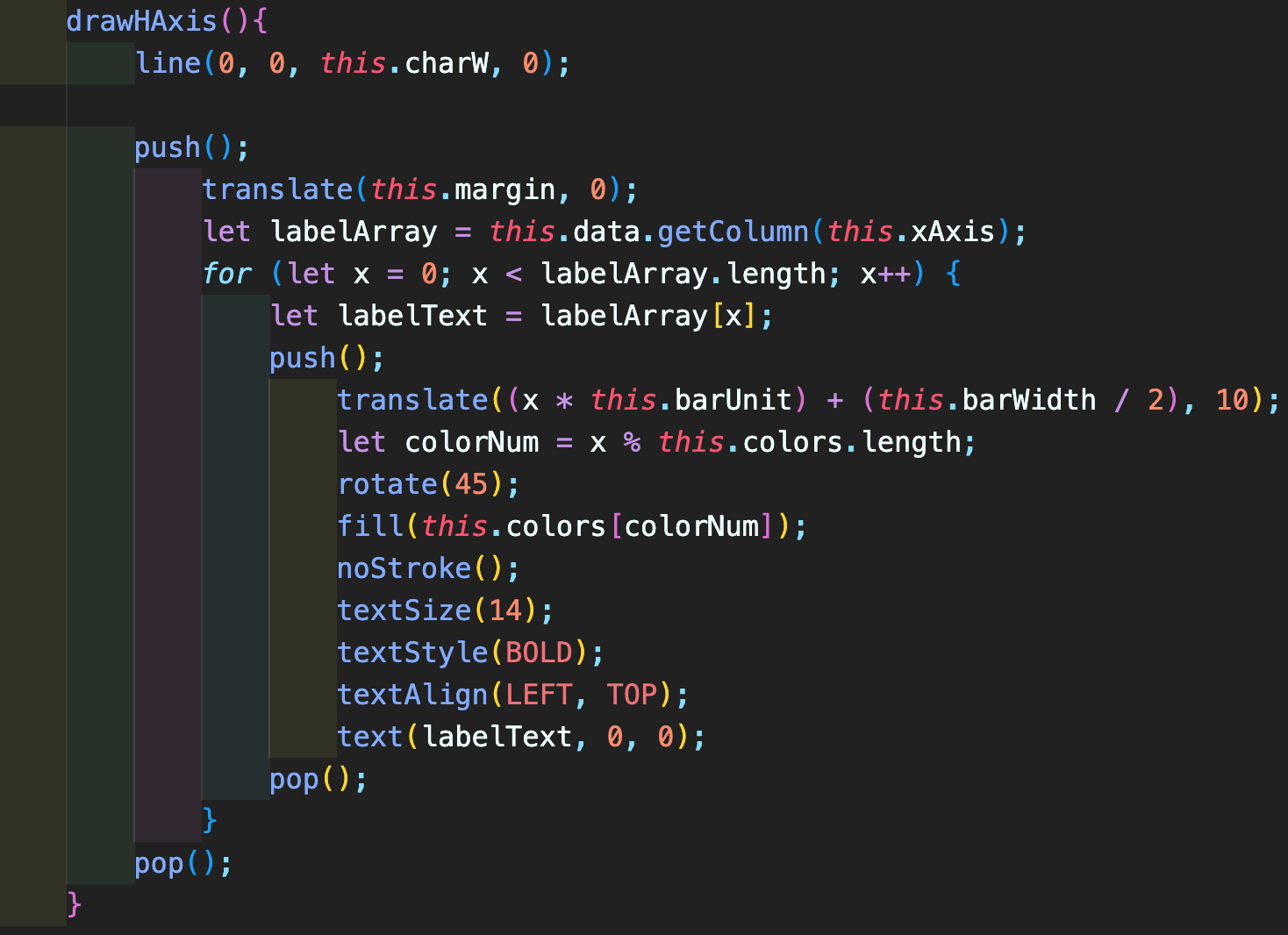
# Code Organisation

## Indentation

I made sure to use a lot of indentation in my code. Anyone looking at the code can see which code belongs to each loop or function since I indented all of the code within them when I added, for example, a loop or function.



^ It is much more difficult to see what each piece of code tries to do and where it goes before adding the indents.



^ As you can see, it is now much simpler to identify which lines of code belong to each function or loop now that there are more indents.

## Visual Studio Code Extensions

Although now directly helping me understand my code, I used many extensions within Visual Studio Code that helped me navigate and see my code better.

* + 1. **Rainbow CSV**

This extension uses different colours to highlight columns in CSV files, making it easier to see the data.

* + 1. **P5 Snippets and Tools**

This extension allows users to connect built-in autofill functionality to the P5 library of functions, allowing them to use standard P5 and VSC functions.

* + 1. **Better Comments**

Visual Studio Code automatically makes comments the same colour; this extension allows users to change the colour by adding symbols at the beginning.

# Object-Oriented Programming

## Classes

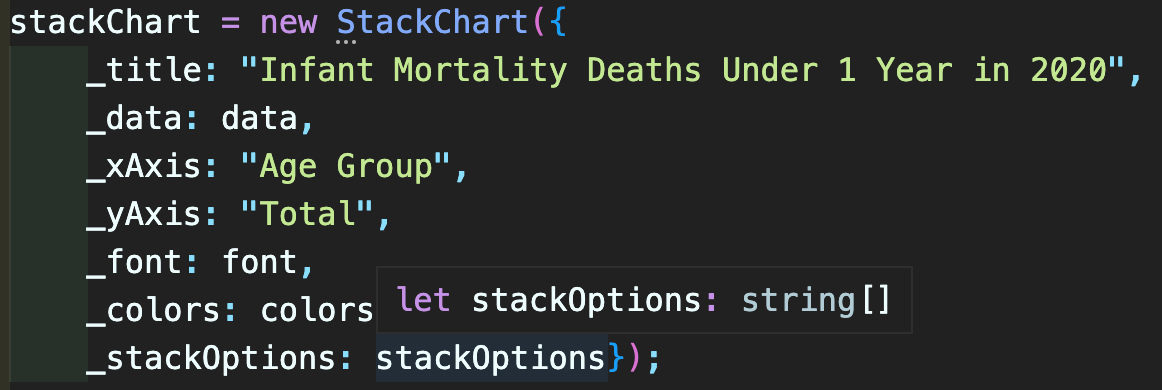
I used a class for each chart to organise my code and avoid having a million lines of it. As a result, instead of coding the whole chart each time, I just needed to create a new one from the class with a few parameters.

For example, I created a class that creates a bar chart (including the axes and bars). This means that instead of writing the full class into sketch.js every time I want to create a new bar chart, I can just say barChart = new BarChart(parameters).

## Objects

I did use a few Objects in my project while using fewer than I do in Classes. I've made it so I can now give the parameters as properties to objects when building a new chart in sketch.js rather than just sending in the values.

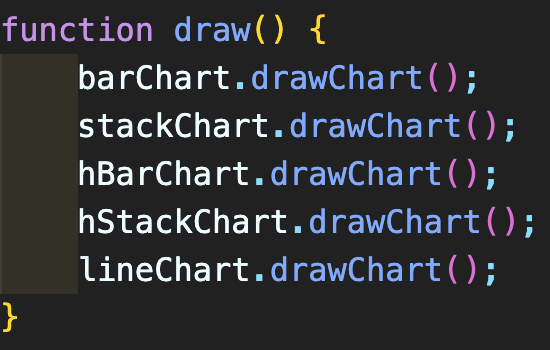
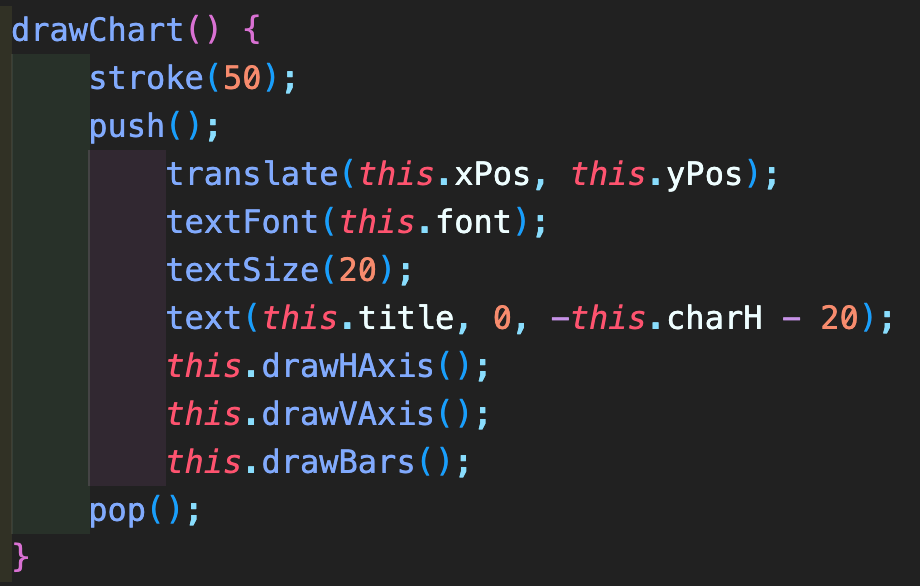
For example, I can create a new stacked chart by stating new StackChart( charW: 100, \_charH: 800, etc.) instead of new StackChart(100, 800, etc.). Users can more easily see what they are giving the object by doing this.



# Use of Functions to Reduce Repetition and Improve Clarity

## drawChart() Function

This function reduces the need to repeatedly draw the axes, bars and titles in sketch.js. Instead, I can draw the axes, bars, and titles in sketch.js by referencing the drawChart() function found at the beginning of each class.



## Calculation Properties

These properties are calculations that need to be done to calculate a specific number. These are calculated using the normal properties.

These are put at the top of the class, so we do not need to keep writing them in the individual functions when we need these values.

