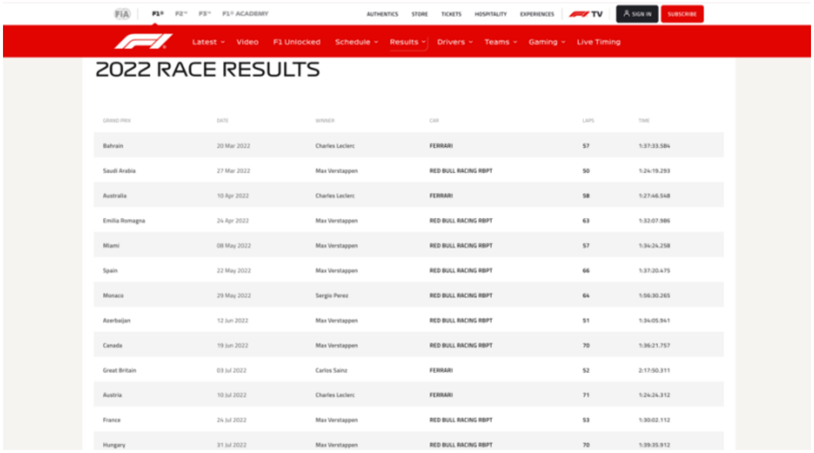
NM2207 Final Project Submission - Writeup
Archangel Quinterra (A0201173X)

Word Count: 1200 (including references)

Overview

Formula 1 (F1) is arguably world's most popular racing series, averaging over 70 million TV viewers per race in 2021 (Statista, 2022). The F1 grid comprises 10 teams (constructors) with 2 drivers per team, forming a line-up of 20 cars. F1 has two championships: World Drivers' Championship (WDC) and World Constructors' Championship (WCC), a tally of points every team/driver earns across races (F1Chronicle, 2020). In F1, the performance of individual drivers is contingent upon the team's performance, encompassing car setup and strategy, and every constructor's performance depends on the drivers' racing capabilities (Funnell, 2022). The bidirectional relationship between constructors and drivers makes F1 rather predictable.

F1's predictability is where data visualization comes to play. Understanding how the WCC progress throughout the season can give fans context to understand efforts exerted by the constructors to attain their position in the WCC standings. However, the official F1 website presents race results in manners that become difficult to derive meaningful insights. For instance, this is how the 2022 race results are presented:

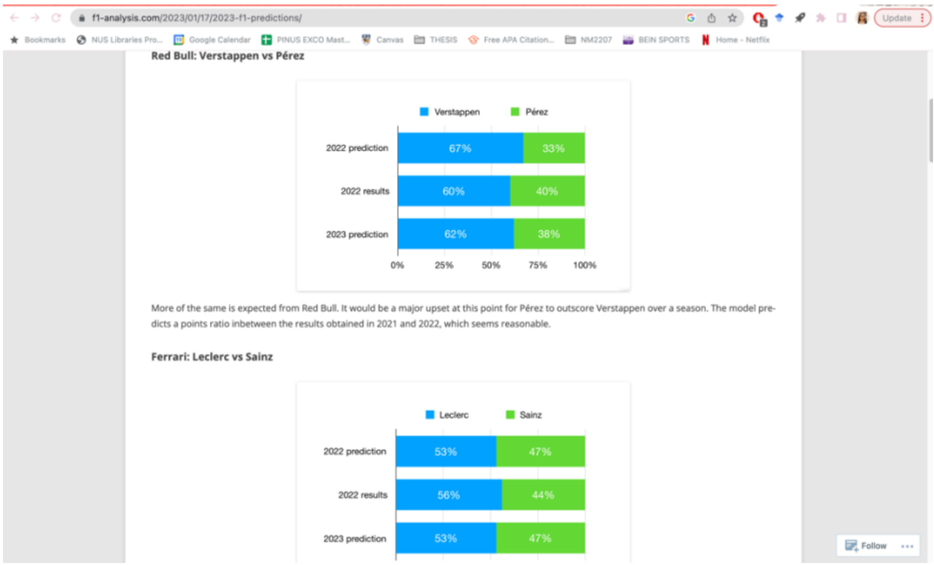


2022 RACE RESULTS

GRAND PRIX	DATE	WINNER	CAR	LAPS	TIME
Bahrain	20 Mar 2022	Charles Leclerc	FERRARI	57	1:37:33.584
Saudi Arabia	27 Mar 2022	Max Verstappen	RED BULL RACING RBPT	50	1:24:19.293
Australia	10 Apr 2022	Charles Leclerc	FERRARI	58	1:27:46.548
Emilia Romagna	24 Apr 2022	Max Verstappen	RED BULL RACING RBPT	63	1:32:07.986
Miami	08 May 2022	Max Verstappen	RED BULL RACING RBPT	57	1:34:24.258
Spain	22 May 2022	Max Verstappen	RED BULL RACING RBPT	66	1:37:20.475
Monaco	29 May 2022	Sergio Perez	RED BULL RACING RBPT	64	1:56:30.265
Azerbaijan	12 Jun 2022	Max Verstappen	RED BULL RACING RBPT	51	1:34:05.941
Canada	19 Jun 2022	Max Verstappen	RED BULL RACING RBPT	70	1:36:21.757
Great Britain	03 Jul 2022	Carlos Sainz	FERRARI	52	2:17:50.311
Austria	10 Jul 2022	Charles Leclerc	FERRARI	71	1:24:24.312
France	24 Jul 2022	Max Verstappen	RED BULL RACING RBPT	53	1:30:02.112
Hungary	31 Jul 2022	Max Verstappen	RED BULL RACING RBPT	70	1:39:35.912

(F1, n.d.).

As an avid F1 fan myself, I personally believe that F1's current visualization format is an opportunity wasted, for visualizing data could help make the sport more attractive, especially for new fans. F1 fans have attempted to utilize data visualization to explain and/or predict performance. For example, f1-analysis.com uses bar graphs to visualize the authors' 2023 predictions against 2022 results:



f1-analysis.com/2023/01/17/2023-f1-predictions/

Red Bull: Verstappen vs Pérez

More of the same is expected from Red Bull. It would be a major upset at this point for Pérez to outscore Verstappen over a season. The model predicts a points ratio inbetween the results obtained in 2021 and 2022, which seems reasonable.

Ferrari: Leclerc vs Sainz

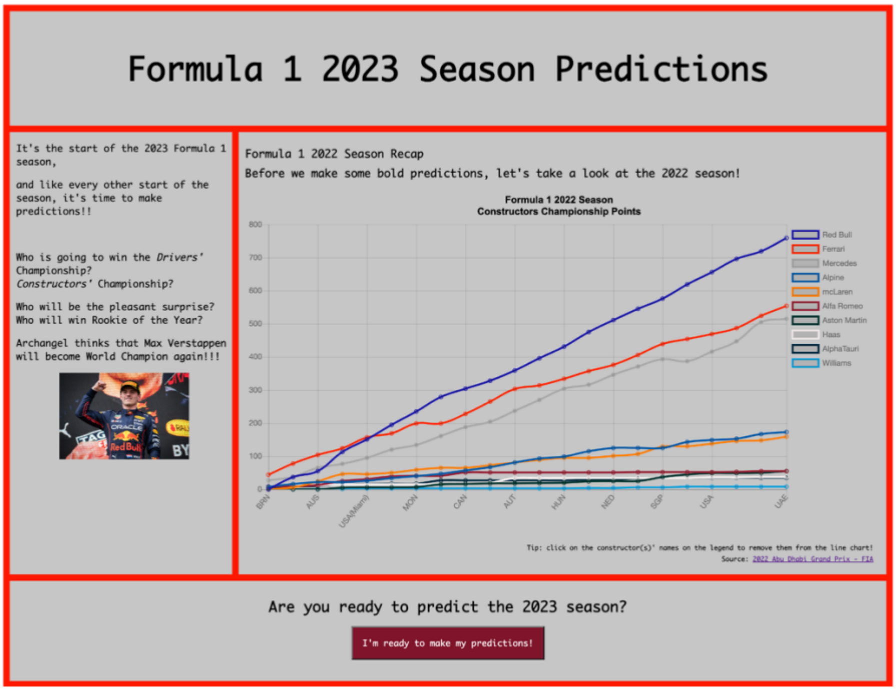
(F1 Analysis, 2023). However, such websites simply present data without allowing the readers to make their own predictions.

Thus, this project is an interactive website that provides data visualization of each constructors' growth of points through the 2022 season, where users can also input their predictions and receive a screenshot-friendly page of their predictions. It aims to help F1 fans make educated predictions of the 2023 season.

Design & Computational Thinking Process

The project comprises two HTML pages: *index.html* and *predictions.html.*

Index.html



Formula 1 2023 Season Predictions

It's the start of the 2023 Formula 1 season,

and like every other start of the season, it's time to make predictions!!

Who is going to win the *Drivers'* Championship?
Constructors' Championship?

Who will be the pleasant surprise?
Who will win Rookie of the Year?

Archangel thinks that Max Verstappen will become World Champion again!!!

Formula 1 2022 Season Recap

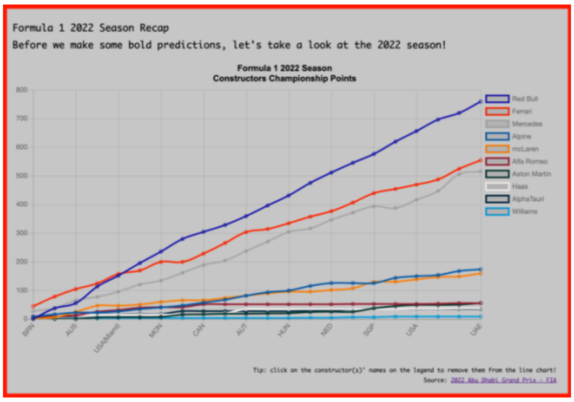
Before we make some bold predictions, let's take a look at the 2022 season!

Tip: click on the constructor(s)' names on the legend to remove them from the line chart!
Source: 2022 Abu Dhabi Grand Prix - FIA

Are you ready to predict the 2023 season?

Index.html is the project's main page, which comprises the title/header ("Formula 1 2023 Season Predictions"), some introductory texts (row 2, left column), a line chart (row 2, right column), and a footer with a "I'm ready!" button that links to *predictions.html*.

Line Chart & Main.js

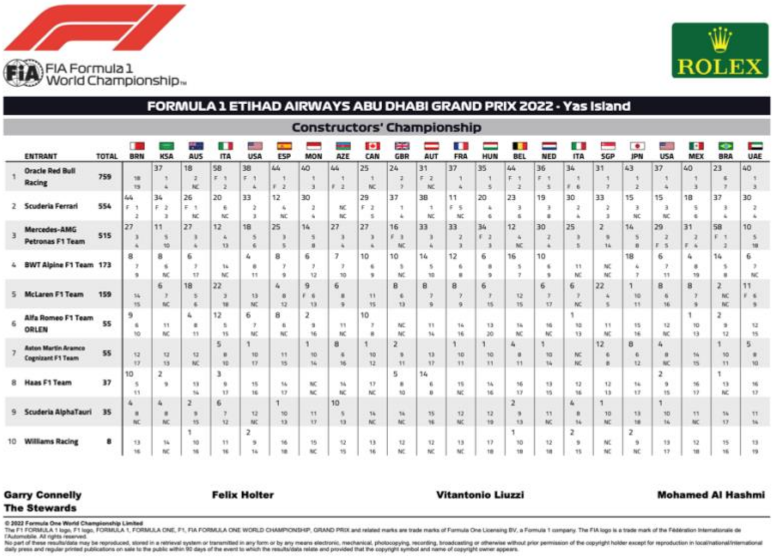


Formula 1 2022 Season Recap

Before we make some bold predictions, let's take a look at the 2022 season!

Tip: click on the constructor(s)' names on the legend to remove them from the line chart!
Source: 2022 Abu Dhabi Grand Prix - FIA

For the line chart, I obtained my data from the FIA's official document (page 4) outlining points collected by each constructor throughout the season.



FIA Formula 1 World Championship

FORMULA 1 ETIHAD AIRWAYS ABU DHABI GRAND PRIX 2022 - Yas Island

Constructors' Championship

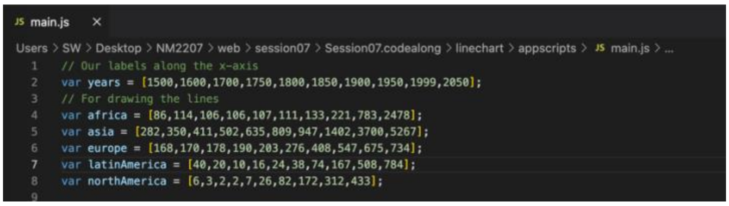
	ENTRANT	TOTAL	BRN	KSA	AUS	ITA	USA	ESP	MON	AZE	CAN	GBR	AUT	FRA	HUN	BEL	NED	ITA	SGP	JPN	USA	MEX	BRA	UAE
1	Oracle Red Bull Racing	759	18 19	37 1 4	18 2 NC	58 F 1 2	38 F 1 4	44 1 F 2	40 1 3	44 1 F 2	25 1 NC	24 2 7	31 F 2 NC	37 1 4	35 1 5	44 F 1 2	36 F 1 5	34 1 F 6	31 1 7	43 1 2	37 1 4	40 1 3	23 6 7	40 1 3
2	Scuderia Ferrari	554	44 F 1 2	34 F 2 3	26 F 1 NC	20 6 NC	33 2 3	12 4 NC	30 2 4	NC NC	29 F 2 5	37 1 4	38 1 NC	11 F 5 NC	20 4 6	23 3 6	19 3 8	30 2 4	33 2 3	15 3 NC	15 3 NC	18 5 6	37 3 4	30 2 4
3	Mercedes-AMG Petronas F1 Team	515	27 3 4	11 5 10	27 3 4	12 4 13	18 5 6	25 3 5	14 5 8	27 3 4	27 3 4	16 F 3 NC	33 3 4	33 2 3	34 F 2 3	12 4 NC	30 2 4	25 3 5	2 9 14	14 5 8	29 2 F 5	31 2 F 4	58 F 1 2	10 5 18
4	BWT Alpine F1 Team	173	8 7 9	8 6 NC	6 7 17	14 NC	4 8 11	8 7 9	6 7 12	7 7 10	10 6 9	10 5 NC	14 5 10	12 6 8	6 8 9	16 5 7	10 6 9	11 NC	NC NC	18 4 7	6 7 11	4 8 19	14 5 8	6 7 NC
5	McLaren F1 Team	159	14 15	6 7 NC	18 5 6	22 3 18	13 NC	4 8 12	9 F 6 13	6 8 9	11 15	8 6 13	8 7 9	8 7 9	6 7 15	12 15	6 7 17	6 7 NC	22 4 5	1 10 11	8 6 16	8 7 9	2 NC NC	11 F 6 9
6	Alfa Romeo F1 Team ORLEN	55	9 6 10	11 NC	4 8 11	12 5 15	6 7 NC	8 6 NC	2 9 16	11 NC	10 7 8	NC NC	11 14	14 16	13 20	14 NC	16 NC	1 10 13	11 NC	15 16	12 NC	1 10 13	2 9 12	12 15
7	Aston Martin Aramco Cognizant F1 Team	55	12 17	12 13	12 NC	5 8 10	1 10 17	11 15	1 10 14	8 6 16	1 10 12	2 9 11	13 17	1 10 11	1 10 11	4 8 11	1 10 14	NC NC	12 6 8	8 6 12	4 8 NC	14 15	1 10 11	5 8 10
8	Haas F1 Team	37	10 5 11	2 9	13 14	3 9 17	15 16	14 17	NC NC	14 NC	17 NC	5 8 10	14 6 8	15 NC	14 16	16 17	13 15	12 16	12 13	14 17	2 9 15	16 17	1 13 NC	16 17
9	Scuderia AlphaTauri	35	4 8 NC	4 8 NC	2 9 15	6 7 12	12 NC	1 10 13	11 17	10 5 13	14 NC	14 NC	15 16	12 NC	12 19	2 9 13	11 NC	4 8 14	1 10 NC	13 18	1 10 14	11 NC	14 17	11 14
10	Williams Racing	8	13 16	14 NC	1 10 16	11 16	2 9 14	16 18	15 NC	12 15	13 16	12 NC	12 NC	13 NC	17 18	1 10 18	12 18	2 9 15	NC NC	2 9 NC	13 17	12 18	15 16	13 19

Garry Connelly　　Felix Holter　　Vitantonio Liuzzi　　Mohamed Al Hashmi
The Stewards

(FIA, 2022).

This document provides the raw points obtained per race, and not a progression. Thus, I manually calculated the progression of points before inputting it to *main.js*, the JavaScript file.

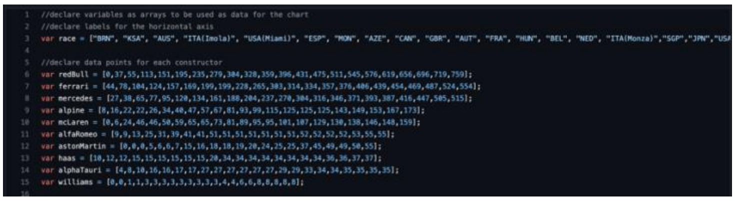
Main.js is a modification of the JavaScript file from session07.codealong:



main.js ×
Users > SW > Desktop > NM2207 > web > session07 > Session07.codealong > linechart > appscripts > JS main.js > ...

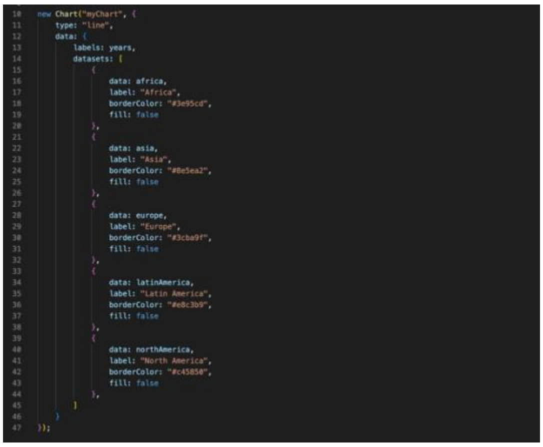
```
// Our labels along the x-axis
var years = [1500,1600,1700,1750,1800,1850,1900,1950,1999,2050];
// For drawing the lines
var africa = [86,114,106,106,107,111,133,221,783,2478];
var asia = [282,350,411,502,635,809,947,1402,3700,5267];
var europe = [168,170,178,190,203,276,408,547,675,734];
var latinAmerica = [40,20,10,16,24,38,74,167,508,784];
var northAmerica = [6,3,2,2,7,26,82,172,312,433];
```

(Ahlin, 2017).



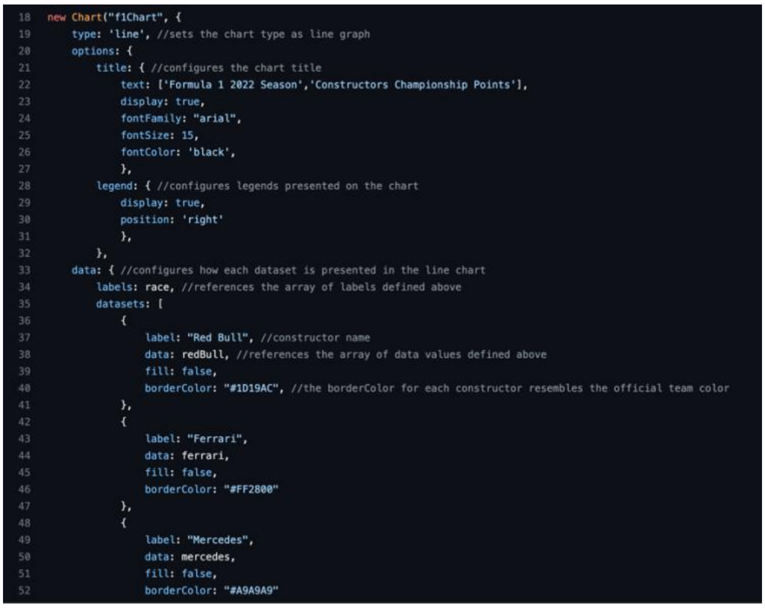
```
//declare variables as arrays to be used as data for the chart
//declare labels for the horizontal axis
var race = ["BRN", "KSA", "AUS", "ITA(Imola)", "USA(Miami)", "ESP", "MON", "AZE", "CAN", "GBR", "AUT", "FRA", "HUN", "BEL", "NED", "ITA(Monza)","SGP","JPN","USA

//declare data points for each constructor
var redBull = [0,37,55,113,151,195,235,279,304,328,359,396,431,475,511,545,576,619,656,696,719,759];
var ferrari = [44,78,104,124,157,169,199,199,228,265,303,314,334,357,376,406,439,454,469,487,524,554];
var mercedes = [27,38,65,77,95,120,134,161,188,204,237,270,304,316,346,371,393,387,416,447,505,515];
var alpine = [8,16,22,22,26,34,40,47,57,67,81,93,99,115,125,125,125,143,149,153,167,173];
var mcLaren = [0,6,24,46,46,50,59,65,65,73,81,89,95,95,101,107,129,130,138,146,148,159];
var alfaRomeo = [9,9,13,25,31,39,41,41,51,51,51,51,51,51,51,52,52,52,52,53,55,55];
var astonMartin = [0,0,0,5,6,6,7,15,16,18,18,19,20,24,25,25,37,45,49,49,50,55];
var haas = [10,12,12,15,15,15,15,15,15,20,34,34,34,34,34,34,34,34,36,36,37,37];
var alphaTauri = [4,8,10,16,16,17,17,27,27,27,27,27,27,29,29,33,34,34,35,35,35,35];
var williams = [0,0,1,1,3,3,3,3,3,3,3,3,3,4,4,6,6,8,8,8,8,8];
```

I decided to follow the codealong's method of declaring all raw data first before referring to them in the chart instead of putting the raw data in the data variable of the chart. The chart configuration feels neater:



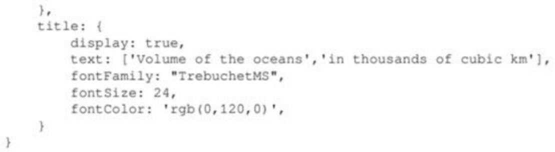
```
new Chart("myChart", {
    type: "line",
    data: {
        labels: years,
        datasets: [
            {
                data: africa,
                label: "Africa",
                borderColor: "#3e95cd",
                fill: false
            },
            {
                data: asia,
                label: "Asia",
                borderColor: "#8e5ea2",
                fill: false
            },
            {
                data: europe,
                label: "Europe",
                borderColor: "#3cba9f",
                fill: false
            },
            {
                data: latinAmerica,
                label: "Latin America",
                borderColor: "#e8c3b9",
                fill: false
            },
            {
                data: northAmerica,
                label: "North America",
                borderColor: "#c45850",
                fill: false
            },
        ]
    }
});
```

(Ahlin, 2017).

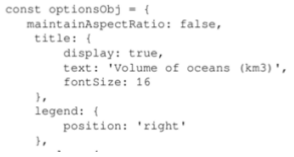


```
new Chart("f1Chart", {
    type: 'line', //sets the chart type as line graph
    options: {
        title: { //configures the chart title
            text: ['Formula 1 2022 Season','Constructors Championship Points'],
            display: true,
            fontFamily: "arial",
            fontSize: 15,
            fontColor: 'black',
            },
        legend: { //configures legends presented on the chart
            display: true,
            position: 'right'
            },
        },
    data: { //configures how each dataset is presented in the line chart
        labels: race, //references the array of labels defined above
        datasets: [
            {
                label: "Red Bull", //constructor name
                data: redBull, //references the array of data values defined above
                fill: false,
                borderColor: "#1D19AC", //the borderColor for each constructor resembles the official team color
            },
            {
                label: "Ferrari",
                data: ferrari,
                fill: false,
                borderColor: "#FF2800"
            },
            {
                label: "Mercedes",
                data: mercedes,
                fill: false,
                borderColor: "#A9A9A9"
```

For the title and legend options, I modified code from *Learn ChartJS* and modified the text, fontFamily, fontSize, fontColor, and legend respectively:

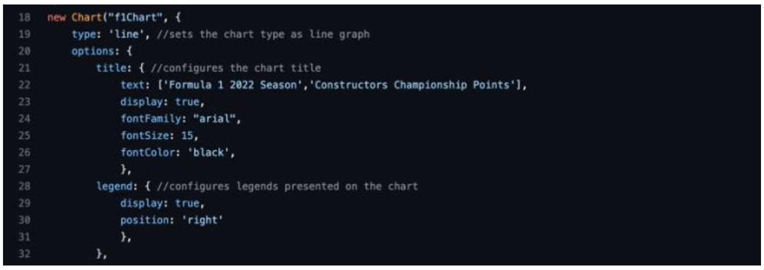


```
        },
        title: {
            display: true,
            text: ['Volume of the oceans','in thousands of cubic km'],
            fontFamily: "TrebuchetMS",
            fontSize: 24,
            fontColor: 'rgb(0,120,0)',
        }
    }
```



```
const optionsObj = {
    maintainAspectRatio: false,
     title: {
         display: true,
         text: 'Volume of oceans (km3)',
         fontSize: 16
     },
     legend: {
         position: 'right'
     },
```

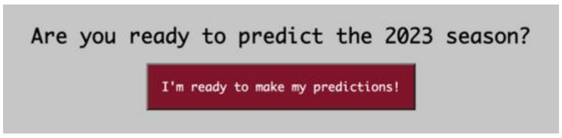
(Rocha, 2019).



```
new Chart("f1Chart", {
    type: 'line', //sets the chart type as line graph
    options: {
        title: { //configures the chart title
            text: ['Formula 1 2022 Season','Constructors Championship Points'],
            display: true,
            fontFamily: "arial",
            fontSize: 15,
            fontColor: 'black',
            },
        legend: { //configures legends presented on the chart
            display: true,
            position: 'right'
            },
        },
```

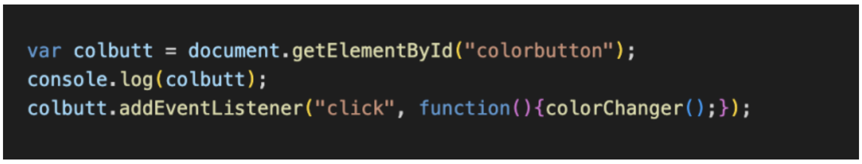
For the borderColor, I followed the HEX code of every constructor's signature color.

Predictions button



Are you ready to predict the 2023 season?

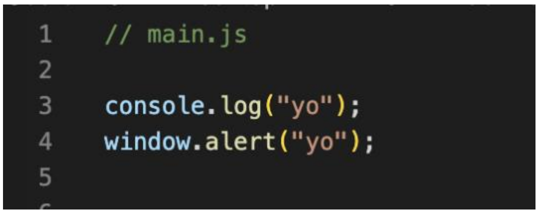
When a user clicks the ready button, two things happen: show an alert box, and open predictions.html. I modified addEventListener from session05.codealong:



```
var colbutt = document.getElementById("colorbutton");
console.log(colbutt);
colbutt.addEventListener("click", function(){colorChanger();});
```

(Jaidka, 2022a).

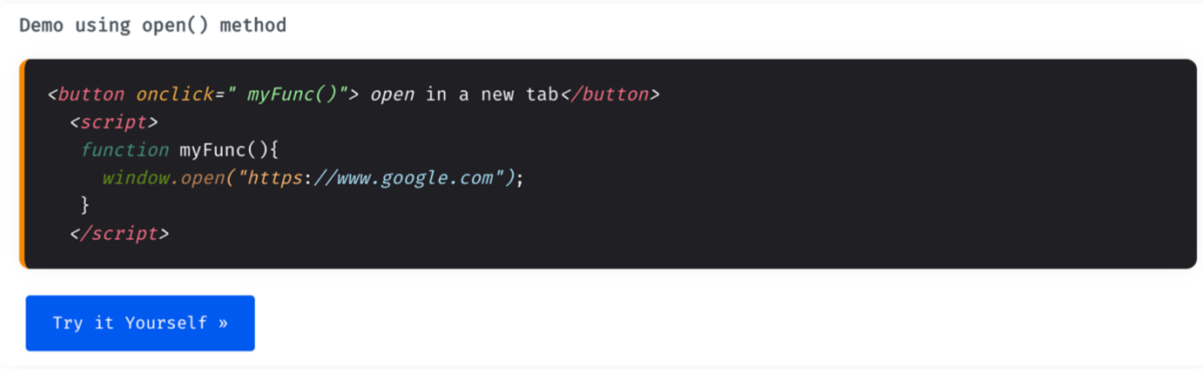
and window.alert from session04.codealong:



```
// main.js

console.log("yo");
window.alert("yo");

```

(Jaidka, 2022b).

I referred to 3schools's example to open a new page using window.open()

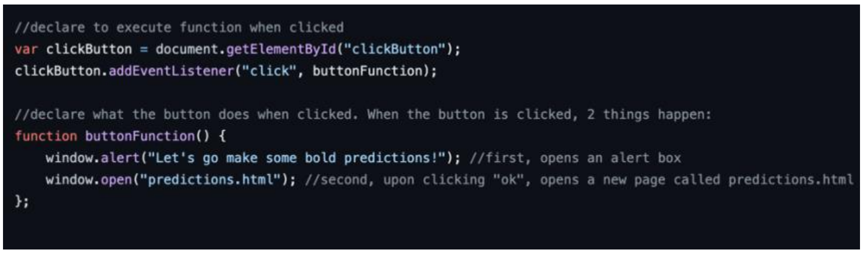


Demo using open() method

```
<button onclick=" myFunc()"> open in a new tab</button>
  <script>
   function myFunc(){
     window.open("https://www.google.com");
   }
  </script>
```

(Apu, 2022).

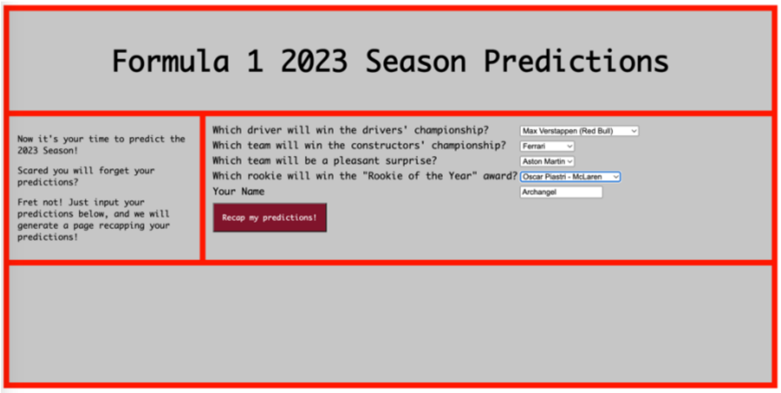
Thus, my code for the button:



```
//declare to execute function when clicked
var clickButton = document.getElementById("clickButton");
clickButton.addEventListener("click", buttonFunction);

//declare what the button does when clicked. When the button is clicked, 2 things happen:
function buttonFunction() {
    window.alert("Let's go make some bold predictions!"); //first, opens an alert box
    window.open("predictions.html"); //second, upon clicking "ok", opens a new page called predictions.html
};
```

Predictions.html

Predictions.html is basically a form where the users can input their predictions:



Formula 1 2023 Season Predictions

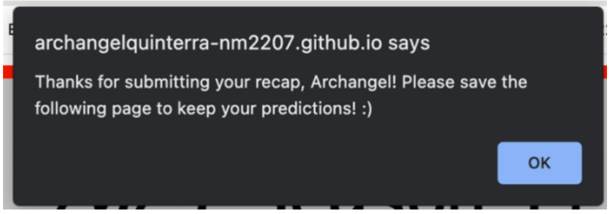
Now it's your time to predict the 2023 Season!

Scared you will forget your predictions?

Fret not! Just input your predictions below, and we will generate a page recapping your predictions!

Which driver will win the drivers' championship? Max Verstappen (Red Bull)
Which team will win the constructors' championship? Ferrari
Which team will be a pleasant surprise? Aston Martin
Which rookie will win the "Rookie of the Year" award? Oscar Piastri - McLaren
Your Name Archangel

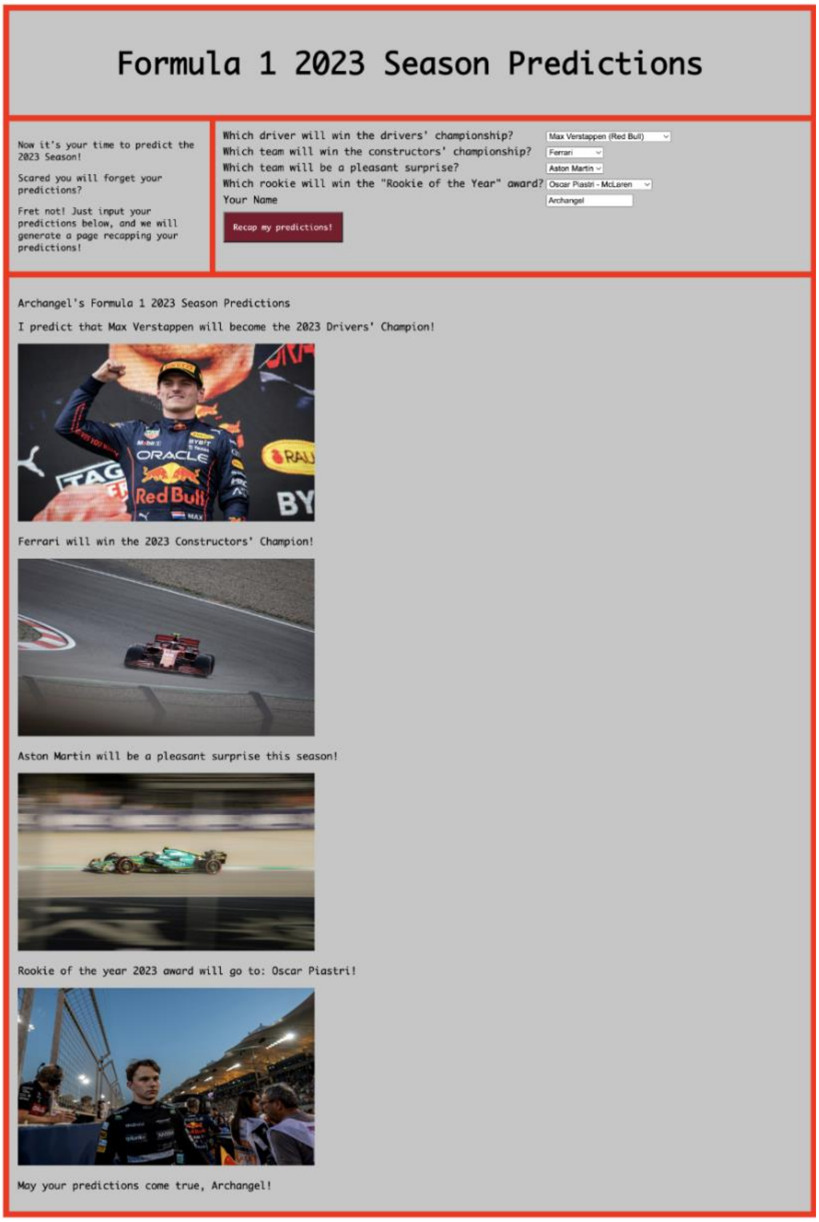
And when they click "Recap my predictions!", a dialog box will appear:



archangelquinterra-nm2207.github.io says

Thanks for submitting your recap, Archangel! Please save the following page to keep your predictions! :)

And the page will generate statements and images based on the user's predictions:



Formula 1 2023 Season Predictions

Now it's your time to predict the 2023 Season!

Scared you will forget your predictions?

Fret not! Just input your predictions below, and we will generate a page recapping your predictions!

Which driver will win the drivers' championship?	Max Verstappen (Red Bull)
Which team will win the constructors' championship?	Ferrari
Which team will be a pleasant surprise?	Aston Martin
Which rookie will win the "Rookie of the Year" award?	Oscar Piastri - McLaren
Your Name	Archangel

Archangel's Formula 1 2023 Season Predictions

I predict that Max Verstappen will become the 2023 Drivers' Champion!

Ferrari will win the 2023 Constructors' Champion!

Aston Martin will be a pleasant surprise this season!

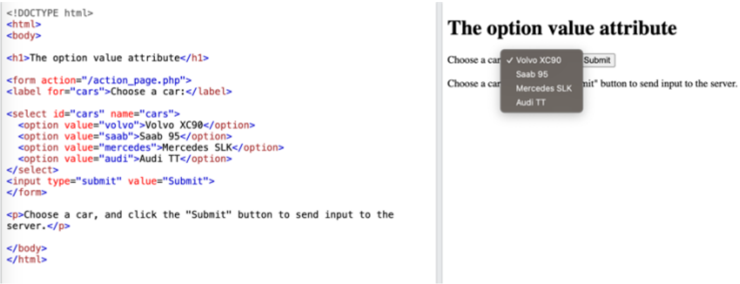
Rookie of the year 2023 award will go to: Oscar Piastri!

May your predictions come true, Archangel!

The images used are royalty-free, downloaded from Shutterstock (free trial), Pexels, Unslpash, and Pixabay.

Predictions.html

The form on *predictions.html* is a modification of w3 school's option value attribute:



```
<!DOCTYPE html>
<html>
<body>

<h1>The option value attribute</h1>

<form action="/action_page.php">
<label for="cars">Choose a car:</label>

<select id="cars" name="cars">
  <option value="volvo">Volvo XC90</option>
  <option value="saab">Saab 95</option>
  <option value="mercedes">Mercedes SLK</option>
  <option value="audi">Audi TT</option>
</select>
<input type="submit" value="Submit">
</form>

<p>Choose a car, and click the "Submit" button to send input to the
server.</p>

</body>
</html>
```

The option value attribute

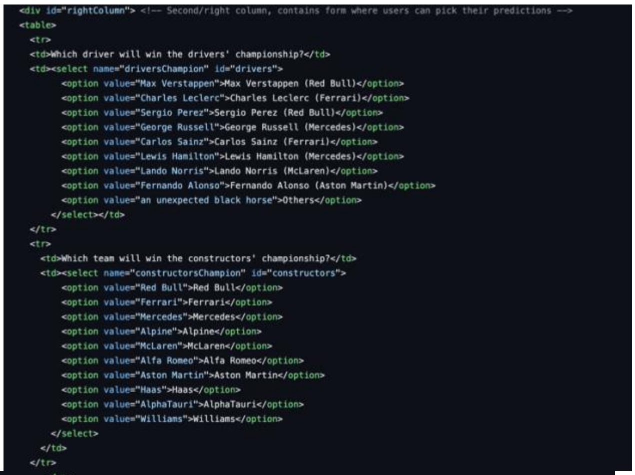
Choose a car

Choose a car   nit" button to send input to the server.

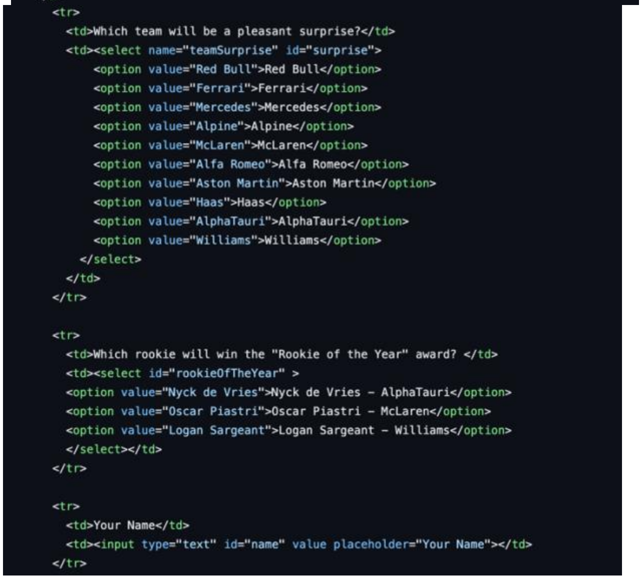
(w3schools, n.d.a)

(Smith, 2015).

Modified into:



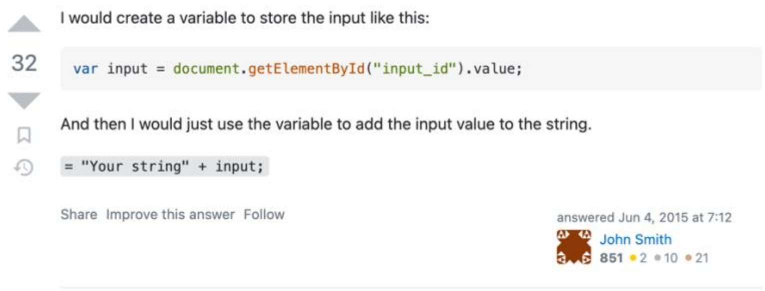
```
<div id="rightColumn"> <!-- Second/right column, contains form where users can pick their predictions -->
<table>
  <tr>
  <td>Which driver will win the drivers' championship?</td>
  <td><select name="driversChampion" id="drivers">
        <option value="Max Verstappen">Max Verstappen (Red Bull)</option>
        <option value="Charles Leclerc">Charles Leclerc (Ferrari)</option>
        <option value="Sergio Perez">Sergio Perez (Red Bull)</option>
        <option value="George Russell">George Russell (Mercedes)</option>
        <option value="Carlos Sainz">Carlos Sainz (Ferrari)</option>
        <option value="Lewis Hamilton">Lewis Hamilton (Mercedes)</option>
        <option value="Lando Norris">Lando Norris (McLaren)</option>
        <option value="Fernando Alonso">Fernando Alonso (Aston Martin)</option>
        <option value="an unexpected black horse">Others</option>
      </select></td>
  </tr>
  <tr>
    <td>Which team will win the constructors' championship?</td>
    <td><select name="constructorsChampion" id="constructors">
        <option value="Red Bull">Red Bull</option>
        <option value="Ferrari">Ferrari</option>
        <option value="Mercedes">Mercedes</option>
        <option value="Alpine">Alpine</option>
        <option value="McLaren">McLaren</option>
        <option value="Alfa Romeo">Alfa Romeo</option>
        <option value="Aston Martin">Aston Martin</option>
        <option value="Haas">Haas</option>
        <option value="AlphaTauri">AlphaTauri</option>
        <option value="Williams">Williams</option>
      </select>
    </td>
  </tr>
```



```
    <tr>
      <td>Which team will be a pleasant surprise?</td>
      <td><select name="teamSurprise" id="surprise">
          <option value="Red Bull">Red Bull</option>
          <option value="Ferrari">Ferrari</option>
          <option value="Mercedes">Mercedes</option>
          <option value="Alpine">Alpine</option>
          <option value="McLaren">McLaren</option>
          <option value="Alfa Romeo">Alfa Romeo</option>
          <option value="Aston Martin">Aston Martin</option>
          <option value="Haas">Haas</option>
          <option value="AlphaTauri">AlphaTauri</option>
          <option value="Williams">Williams</option>
        </select>
      </td>
    </tr>

    <tr>
      <td>Which rookie will win the "Rookie of the Year" award? </td>
      <td><select id="rookieOfTheYear" >
      <option value="Nyck de Vries">Nyck de Vries - AlphaTauri</option>
      <option value="Oscar Piastri">Oscar Piastri - McLaren</option>
      <option value="Logan Sargeant">Logan Sargeant - Williams</option>
      </select></td>
    </tr>

    <tr>
      <td>Your Name</td>
      <td><input type="text" id="name" value placeholder="Your Name"></td>
    </tr>
```

Predictions.js

Predictions.js is all about extracting the user's choice (option value) from *predictions.html*. I followed this suggestion on StackOverflow:



I would create a variable to store the input like this:

```
var input = document.getElementById("input_id").value;
```

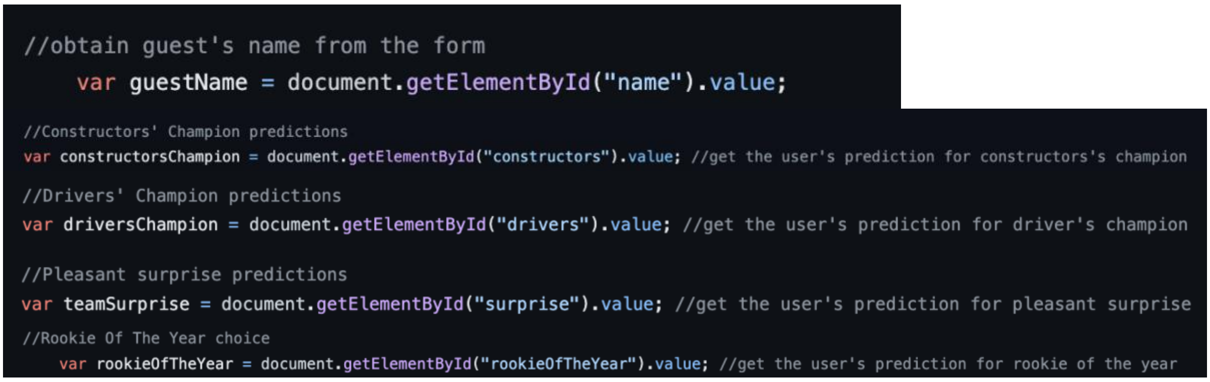
And then I would just use the variable to add the input value to the string.

```
= "Your string" + input;
```

32
Share Improve this answer Follow
answered Jun 4, 2015 at 7:12
John Smith
851 ●2 ●10 ●21

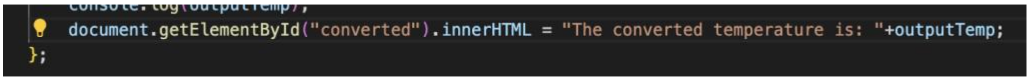
(Smith, 2015).

And modified it:



```
//obtain guest's name from the form
    var guestName = document.getElementById("name").value;

//Constructors' Champion predictions
var constructorsChampion = document.getElementById("constructors").value; //get the user's prediction for constructors's champion

//Drivers' Champion predictions
var driversChampion = document.getElementById("drivers").value; //get the user's prediction for driver's champion

//Pleasant surprise predictions
var teamSurprise = document.getElementById("surprise").value; //get the user's prediction for pleasant surprise
//Rookie Of The Year choice
    var rookieOfTheYear = document.getElementById("rookieOfTheYear").value; //get the user's prediction for rookie of the year
```

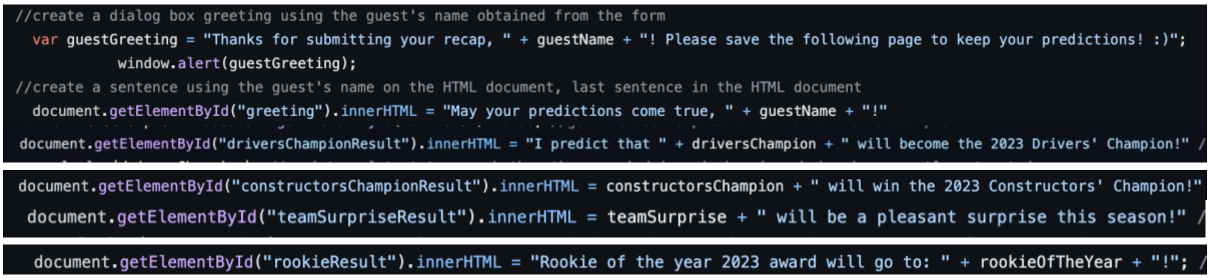
For the statements (strings), I modified code from session03.codealong:



```
    console.log(outputTemp);
    document.getElementById("converted").innerHTML = "The converted temperature is: "+outputTemp;
};
```

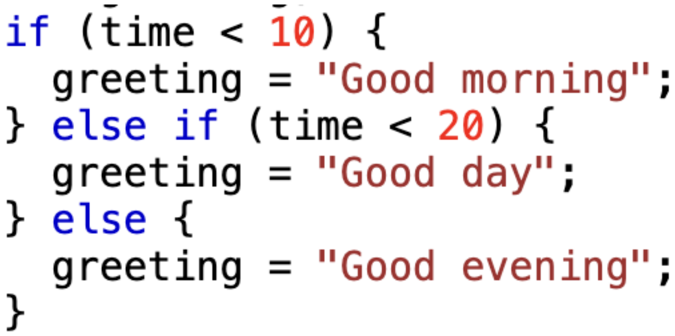
(Jaidka, 2022c).

It becomes:



```
//create a dialog box greeting using the guest's name obtained from the form
  var guestGreeting = "Thanks for submitting your recap, " + guestName + "! Please save the following page to keep your predictions! :)";
          window.alert(guestGreeting);
//create a sentence using the guest's name on the HTML document, last sentence in the HTML document
  document.getElementById("greeting").innerHTML = "May your predictions come true, " + guestName + "!"
 document.getElementById("driversChampionResult").innerHTML = "I predict that " + driversChampion + " will become the 2023 Drivers' Champion!" /
 document.getElementById("constructorsChampionResult").innerHTML = constructorsChampion + " will win the 2023 Constructors' Champion!"
  document.getElementById("teamSurpriseResult").innerHTML = teamSurprise + " will be a pleasant surprise this season!" /
   document.getElementById("rookieResult").innerHTML = "Rookie of the year 2023 award will go to: " + rookieOfTheYear + "!"; /
```

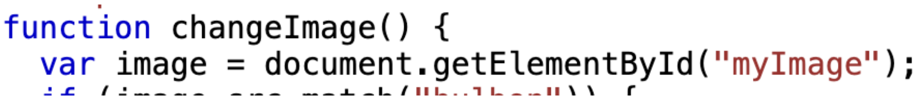
For the if/else/elseif statement, I followed w3schools' example as structure to set the conditions:



```
if (time < 10) {
  greeting = "Good morning";
} else if (time < 20) {
  greeting = "Good day";
} else {
  greeting = "Good evening";
}
```

(w3schools, n.d.b)

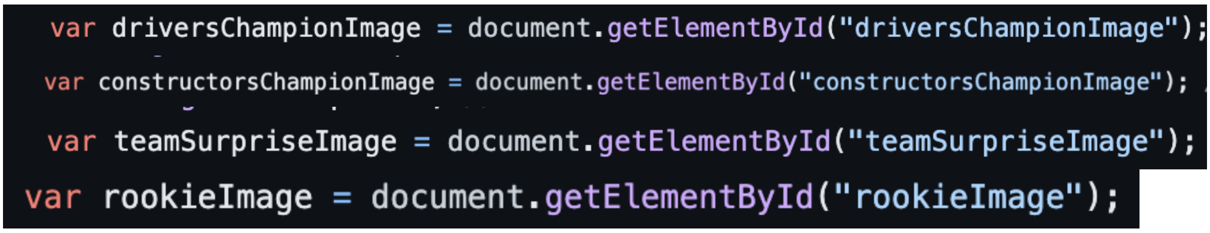
And this var image = document.getElementById("myImage") to place the image on its respective container:



```
function changeImage() {
  var image = document.getElementById("myImage");
  if (image.src.match("bulbon")) {
```

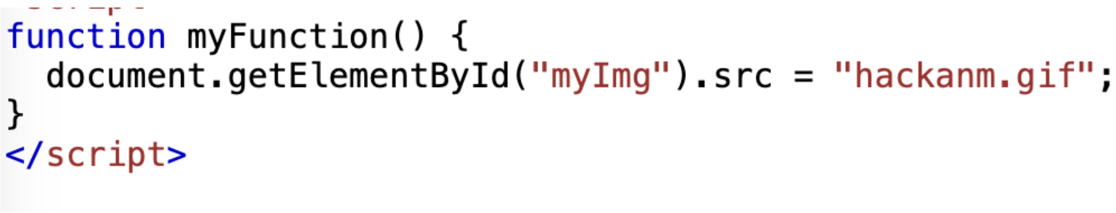
(w3schools, n.d.c)

Modified into:



```
var driversChampionImage = document.getElementById("driversChampionImage");
var constructorsChampionImage = document.getElementById("constructorsChampionImage");
var teamSurpriseImage = document.getElementById("teamSurpriseImage");
var rookieImage = document.getElementById("rookieImage");
```

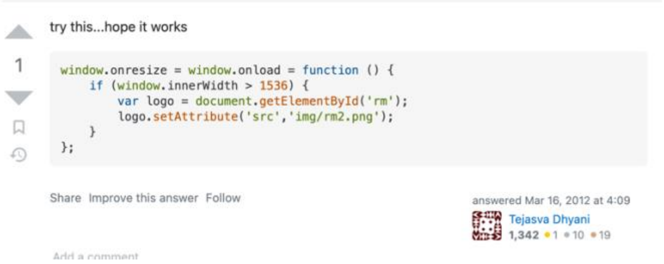
Once the images have its respective containers, I need to add respective images that meets the conditions. I first tried this example by w3schools:



```
function myFunction() {
  document.getElementById("myImg").src = "hackanm.gif";
}
</script>
```

(w3schools, n.d.d)

But because I have declared document.getElementById as a varible, I tried driversChampionImage.src = “resources/MaxVerstappen.jpg” but this did not work. I found this code on StackOverflow:



1

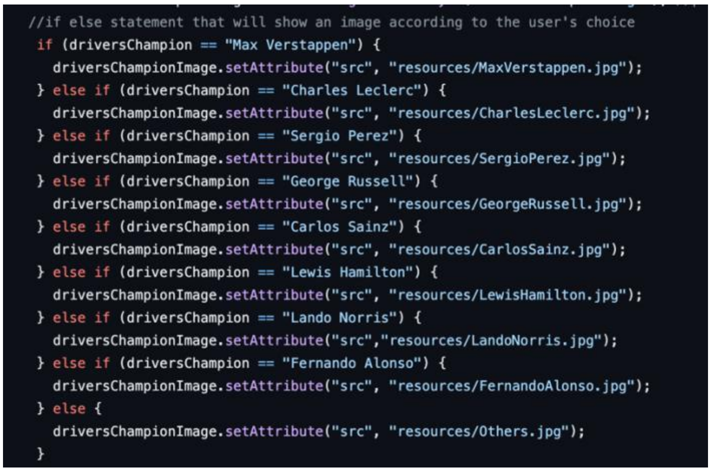
try this...hope it works

```
window.onresize = window.onload = function () {
    if (window.innerWidth > 1536) {
        var logo = document.getElementById('rm');
        logo.setAttribute('src','img/rm2.png');
    }
};
```

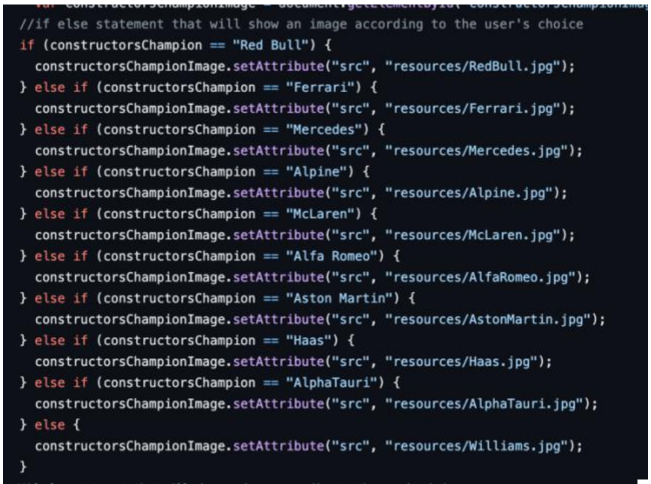
answered Mar 16, 2012 at 4:09
Tejasva Dhyani
1,342 ●1 ●10 ●19

(Dhyani, 2012).

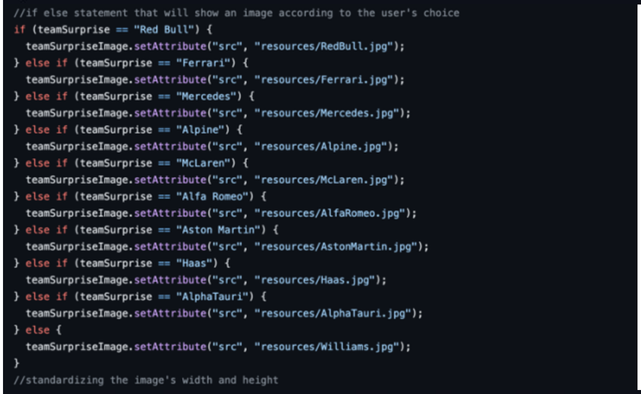
And my modification worked:



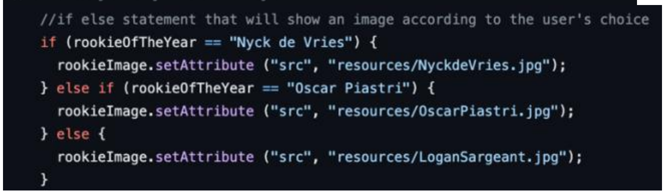
```
//if else statement that will show an image according to the user's choice
 if (driversChampion == "Max Verstappen") {
   driversChampionImage.setAttribute("src", "resources/MaxVerstappen.jpg");
 } else if (driversChampion == "Charles Leclerc") {
   driversChampionImage.setAttribute("src", "resources/CharlesLeclerc.jpg");
 } else if (driversChampion == "Sergio Perez") {
   driversChampionImage.setAttribute("src", "resources/SergioPerez.jpg");
 } else if (driversChampion == "George Russell") {
   driversChampionImage.setAttribute("src", "resources/GeorgeRussell.jpg");
 } else if (driversChampion == "Carlos Sainz") {
   driversChampionImage.setAttribute("src", "resources/CarlosSainz.jpg");
 } else if (driversChampion == "Lewis Hamilton") {
   driversChampionImage.setAttribute("src", "resources/LewisHamilton.jpg");
 } else if (driversChampion == "Lando Norris") {
   driversChampionImage.setAttribute("src","resources/LandoNorris.jpg");
 } else if (driversChampion == "Fernando Alonso") {
   driversChampionImage.setAttribute("src", "resources/FernandoAlonso.jpg");
 } else {
   driversChampionImage.setAttribute("src", "resources/Others.jpg");
 }
```



```
//if else statement that will show an image according to the user's choice
if (constructorsChampion == "Red Bull") {
  constructorsChampionImage.setAttribute("src", "resources/RedBull.jpg");
} else if (constructorsChampion == "Ferrari") {
  constructorsChampionImage.setAttribute("src", "resources/Ferrari.jpg");
} else if (constructorsChampion == "Mercedes") {
  constructorsChampionImage.setAttribute("src", "resources/Mercedes.jpg");
} else if (constructorsChampion == "Alpine") {
  constructorsChampionImage.setAttribute("src", "resources/Alpine.jpg");
} else if (constructorsChampion == "McLaren") {
  constructorsChampionImage.setAttribute("src", "resources/McLaren.jpg");
} else if (constructorsChampion == "Alfa Romeo") {
  constructorsChampionImage.setAttribute("src", "resources/AlfaRomeo.jpg");
} else if (constructorsChampion == "Aston Martin") {
  constructorsChampionImage.setAttribute("src", "resources/AstonMartin.jpg");
} else if (constructorsChampion == "Haas") {
  constructorsChampionImage.setAttribute("src", "resources/Haas.jpg");
} else if (constructorsChampion == "AlphaTauri") {
  constructorsChampionImage.setAttribute("src", "resources/AlphaTauri.jpg");
} else {
  constructorsChampionImage.setAttribute("src", "resources/Williams.jpg");
}
```

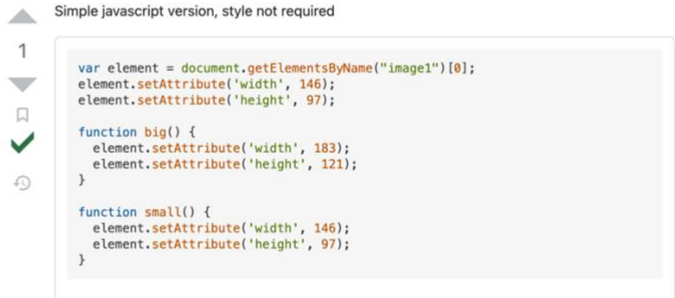


```
//if else statement that will show an image according to the user's choice
if (teamSurprise == "Red Bull") {
  teamSurpriseImage.setAttribute("src", "resources/RedBull.jpg");
} else if (teamSurprise == "Ferrari") {
  teamSurpriseImage.setAttribute("src", "resources/Ferrari.jpg");
} else if (teamSurprise == "Mercedes") {
  teamSurpriseImage.setAttribute("src", "resources/Mercedes.jpg");
} else if (teamSurprise == "Alpine") {
  teamSurpriseImage.setAttribute("src", "resources/Alpine.jpg");
} else if (teamSurprise == "McLaren") {
  teamSurpriseImage.setAttribute("src", "resources/McLaren.jpg");
} else if (teamSurprise == "Alfa Romeo") {
  teamSurpriseImage.setAttribute("src", "resources/AlfaRomeo.jpg");
} else if (teamSurprise == "Aston Martin") {
  teamSurpriseImage.setAttribute("src", "resources/AstonMartin.jpg");
} else if (teamSurprise == "Haas") {
  teamSurpriseImage.setAttribute("src", "resources/Haas.jpg");
} else if (teamSurprise == "AlphaTauri") {
  teamSurpriseImage.setAttribute("src", "resources/AlphaTauri.jpg");
} else {
  teamSurpriseImage.setAttribute("src", "resources/Williams.jpg");
}
//standardizing the image's width and height
```



```
//if else statement that will show an image according to the user's choice
if (rookieOfTheYear == "Nyck de Vries") {
  rookieImage.setAttribute ("src", "resources/NyckdeVries.jpg");
} else if (rookieOfTheYear == "Oscar Piastri") {
  rookieImage.setAttribute ("src", "resources/OscarPiastri.jpg");
} else {
  rookieImage.setAttribute ("src", "resources/LoganSargeant.jpg");
}
```

I followed this suggestion on StackOverflow to standardize the size of images:

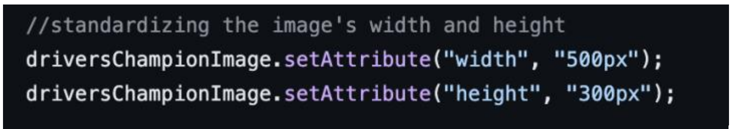


Simple javascript version, style not required

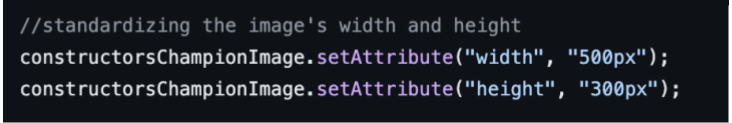
```
var element = document.getElementsByName("image1")[0];
element.setAttribute('width', 146);
element.setAttribute('height', 97);

function big() {
  element.setAttribute('width', 183);
  element.setAttribute('height', 121);
}

function small() {
  element.setAttribute('width', 146);
  element.setAttribute('height', 97);
}
```

(Tleukabiluly, 2016),

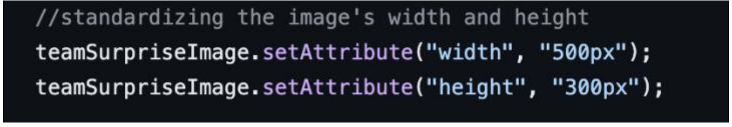
I modified the code into:



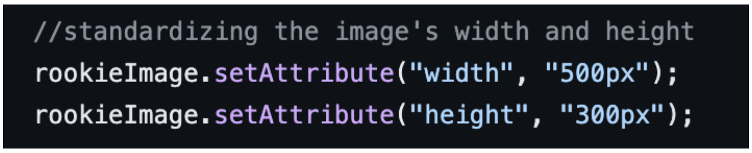
```
//standardizing the image's width and height
driversChampionImage.setAttribute("width", "500px");
driversChampionImage.setAttribute("height", "300px");
```



```
//standardizing the image's width and height
constructorsChampionImage.setAttribute("width", "500px");
constructorsChampionImage.setAttribute("height", "300px");
```



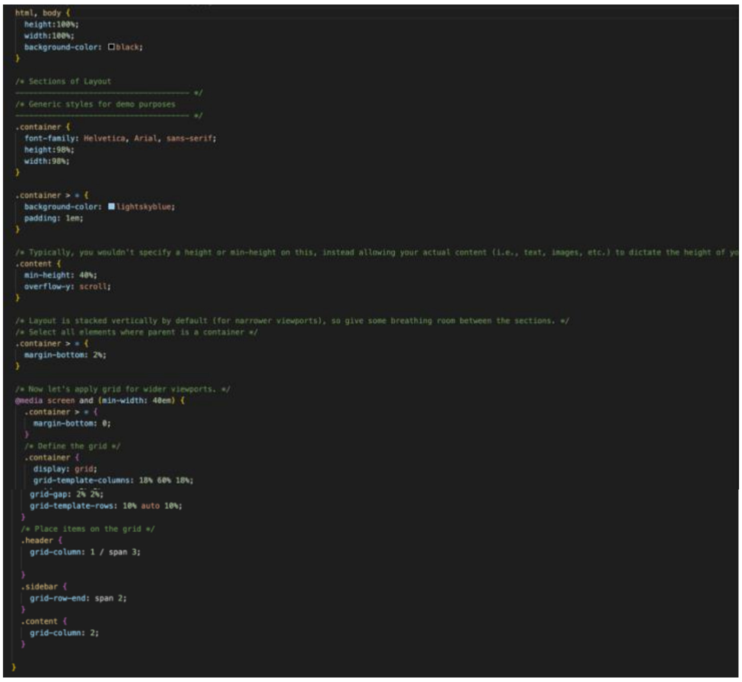
```
//standardizing the image's width and height
teamSurpriseImage.setAttribute("width", "500px");
teamSurpriseImage.setAttribute("height", "300px");
```



```
//standardizing the image's width and height
rookieImage.setAttribute("width", "500px");
rookieImage.setAttribute("height", "300px");
```

CSS File

Both CCS files, *appstyle.css* and *predictions.css* are adapted from the index.html file provided in class for our nm2207.org website:



```
html, body {
  height:100%;
  width:100%;
  background-color: □black;
}

/* Sections of Layout
------------------------------------------ */
/* Generic styles for demo purposes
------------------------------------------ */
.container {
  font-family: Helvetica, Arial, sans-serif;
  height:98%;
  width:98%;
}

.container > * {
  background-color: ■lightskyblue;
  padding: 1em;
}

/* Typically, you wouldn't specify a height or min-height on this, instead allowing your actual content (i.e., text, images, etc.) to dictate the height of yo
.content {
  min-height: 40%;
  overflow-y: scroll;
}

/* Layout is stacked vertically by default (for narrower viewports), so give some breathing room between the sections. */
/* Select all elements where parent is a container */
.container > * {
  margin-bottom: 2%;
}

/* Now let's apply grid for wider viewports. */
@media screen and (min-width: 40em) {
  .container > * {
    margin-bottom: 0;
  }
  /* Define the grid */
  .container {
    display: grid;
    grid-template-columns: 18% 60% 18%;
   grid-gap: 2% 2%;
   grid-template-rows: 10% auto 10%;
  }
  /* Place items on the grid */
  .header {
   grid-column: 1 / span 3;

  }
  .sidebar {
   grid-row-end: span 2;
  }
  .content {
   grid-column: 2;
  }

}
```

(Jaidka, 2022d).

The modifications are done through trial and error.

Self-Critique

Some aspects I could improve on include:

- If we learnt about Chart Events earlier, I could have used events to add more lines (constructors' data) upon the user's request (clicking button) instead of displaying everything. It is a better idea, as the current display of 10 lines is rather messy, and some people prefer seeing data of only the top 3-5 teams.
- I could have used a single CSS for both HTML files (instead of separate CSS files), as that would make more sense for projects with more HTML files involved.

- Given more time, I could have challenged myself to use fetch API to obtain my datasets instead of typing everything manually.

References

Ahlin, T. (2017, March 5). *Data visualization with Chart.js: An introduction.* https://tobiasahlin.com/blog/introduction-to-chartjs/.

Apu (2022, November 13). *How to open new html page on button click in javascript*. 3schools. https://www.3schools.in/2021/09/how-to-open-new-html-page-button-javascript.html.

Da Rocha, H. (2019). Learn Chart.js: Create interactive visualizations for the Web with Chart.js 2. Packt Publishing Ltd.

Dhyani, T. (2012, March 16). *try this...hope it works* [Comment on online forum post *JavaScript change img src attribute without jQuery*]. StackOverflow. https://stackoverflow.com/questions/9731728/javascript-change-img-src-attribute-without-jquery.

F1. (n.d.). 2022 RACE RESULTS. https://www.formula1.com/en/results.html/2022/races.html.

F1 Analysis. (2023, January 17). *2023 F1 Driver Predictions*. https://f1-analysis.com/2023/01/17/2023-f1-predictions/.

F1Chronicle (2020, July 30). *The Complete Beginners Guide to Formula 1*. https://f1chronicle.com/a-beginners-guide-to-formula-1/.

FIA. (2022, November 20). 2022 Abu Dhabi Grand Prix – Championship Points. https://www.fia.com/sites/default/files/decision-document/2022%20Abu%20Dhabi%20Grand%20Prix%20-%20Championship%20Points.pdf.

Funnell, R. (2022, July 7). *Is Formula 1 Racing Success Down To The Driver, Their Team, Or Their Car?*. IFL Science. https://www.iflscience.com/is-formula-1-racing-success-down-to-the-driver-their-team-or-their-car-

64326#:~:text=The%20F1%20race%20team%20and,around%2030%20to%2040%20percent.

Jaidka, K. (2022a). session05.codealong. In *NM2207: Computational Media Literacy*. National University of Singapore.

Jaidka, K. (2022b). session04.codealong. In *NM2207: Computational Media Literacy*. National University of Singapore.

Jaidka, K. (2022c). session03.codealong. In *NM2207: Computational Media Literacy*. National University of Singapore.

Jaidka, K. (2022d). appstyles.css. In *NM2207: Computational Media Literacy*. National University of Singapore.

Smith, J. (2015, Jun 4). *I would create a variable to store the input like this:* [Comment on online forum post *How do I get the value of text input field using JavaScript*]. StackOverflow. https://stackoverflow.com/questions/11563638/how-do-i-get-the-value-of-text-input-field-using-javascript.

Statista (2022, February). Average TV viewers of Formula One (F1) racing per Grand Prix from 2014 to 2021. https://www.statista.com/statistics/992675/f1-audience-countries/#:~:text=The%20average%20number%20of%20Formula,per%20Grand%20Prix%20in%202020.

Tleukabiluly, M. (2016, Jan 17). *Simple javascript version, style not required*. [Comment on online forum post *Change image width/height with Javascript]*. StackOverflow. https://stackoverflow.com/questions/34838954/change-image-width-height-with-javascript.

W3schools (n.d.a). *The option value attribute*. https://www.w3schools.com/tags/tryit.asp?filename=tryhtml_option_value.

W3schools (n.d.b). *JavaScript if.. else.* https://www.w3schools.com/js/tryit.asp?filename=tryjs_elseif.

W3schools (n.d.c). *JavaScript Statements.* https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_state_if_else2.

W3schools (n.d.d). *Image src Property*. https://www.w3schools.com/jsref/prop_img_src.asp.