Dec 06, 2024

Plagiarism Scan Report

13% Plagiarized

87% Unique

Characters:11539	Words:983
Sentences:30	Speak Time: 8 Min

Excluded URL	None

Content Checked for Plagiarism

// Select the root element in the HTML where we will append our content const app = document.getElementById('root'); const PNG = document.getElementById('image') // Bool variable for the toggle function let isDataVisible = false; let isDataVisible2 = false; // Create an element segment for the F1 logo const logo = document.createElement('img'); logo.src = 'FormulaOne.png'; // Creates a div for the toggle buttons const buttons = document.createElement('div'); buttons.setAttribute('class', 'container'); // Container 1 - 4 are all created here and use the same container class CSS const container = document.createElement('div'); container.setAttribute('class', 'container'); const container2 = document.createElement('div'); container2.setAttribute('class', 'container'); const container3 = document.createElement('div'); container3.setAttribute('class', 'container'); const container4 = document.createElement('div'); container4.setAttribute('class', 'container'); // Appends all the containers to the root div PNG.appendChild(logo); app.appendChild(buttons) app.appendChild(container); app.appendChild(container2); app.appendChild(container3); app.appendChild(container4) //The Json API used does not have images, the following 3 arrays of images are used in specific container. //Drivers Images let images= ['1alealb01.png','2feralo01.png','3olibea01.png','4valbot01.png', '5fracol01.png','6piegas01.png','7lewham01.png','8nichul01.png','9lialaw01.png', '10chalec01.png','11kevmag01.png','12lannor01.png','13estoco01.png','14serper01.png', '15oscpia01.png','16danric01.png','17georus01.png','18carsai01.png','19logsar01.png', '20lanstr01.png','21yuktsu01.png','22maxver01.png','23guazho01.png'] //Constructors Images let ConImages = ['alpine.png','aston-martin.png','ferrari.png','haas.png','mclaren.png', 'mercedes.png','rb.png','red-bull-racing.png','sauber.png','williams.png'] //Track Images let TrackImages = ['1Bahrain_Circuit.png','2Saudi_Arabia_Circuit.png','3Australia_Circuit.png', '4Japan_Circuit.png','5China_Circuit.png','6Miami_Circuit.png','7Emilia_Romagna_Circuit.png', '8Monaco_Circuit.png','9Canada_Circuit.png','10Spain_Circuit.png','11Austria_Circuit.png', '12Great_Britain_Circuit.png','13Hungary_Circuit.png','14Belgium_Circuit.png','15Netherlands_Circuit.png', '16Italy_Circuit.png','17Baku_Circuit.png','18Singapore_Circuit.png','19USA_Circuit.png', '20Mexico_Circuit.png','21Brazil_Circuit.png','22Las_Vegas_Circuit.png','23Qatar_Circuit.png', '24Abu_Dhabi_Circuit.png'] //FetchDrivers function fetches the data from the URL.json async function fetchDrivers() { const DriversTitle = document.createElement('img'); //the image element for Drivers Title

```
DriversTitle.src = 'F1 Drivers 2024.png'; container.appendChild(DriversTitle);
//counter for the images let counter = 0 try { const response = await
fetch('https://api.jolpi.ca/ergast/f1/2024/drivers/'); const data = await
response.json(); // Parses the response as JSON const drivers =
data.MRData.DriverTable.Drivers; drivers.forEach(driver => {
createDriverCards(driver, counter) //calls the function to sort and print the
Data counter++ }); } catch (error) { // Display an error message if the fetch fails
const errorMessage = document.createElement('div');
errorMessage.textContent = `Gah, it's not working! Error: ${error.message}`;
app.appendChild(errorMessage); } } //Function to sort and print the Data
function createDriverCards(driver, counter) { const card =
document.createElement('div'); card.setAttribute('class', 'card'); // Styling class
for each card const driverName = document.createElement('h1');
driverName.textContent = `${driver.givenName} ${driver.familyName}`; // Set
the title for each card const driverimage = document.createElement('img')
//image for the cards driverimage.src = "photos/"+images[counter]; //Data
form the json to be sorted and printed const driverNumber =
document.createElement('h2') driverNumber.textContent = `Driver Number :
${driver.permanentNumber}`; const driverCode =
document.createElement('h3') driverCode.textContent = `Driver initials :
${driver.code}` const driverNationality = document.createElement('h3')
driverNationality.textContent = `Nationality : ${driver.nationality}`; const
driverDOB = document.createElement('h3') driverDOB.textContent = `Date of
Birth : ${driver.dateOfBirth}`; container.appendChild(card); // Append card to
container card.appendChild(driverName); //Appends the data to each card
card.appendChild(driverimage) card.appendChild(driverNumber);
card.appendChild(driverCode); card.appendChild(driverNationality);
card.appendChild(driverDOB); }; //FetchConstructors function fetches the data
from the URL.json async function fetchConstructors() { let counter2 = 0 const
constructorTitle = document.createElement('img'); //the image element for
Constructors Title constructorTitle.src = 'F1 Teams 2024.png';
container2.appendChild(constructorTitle); try { const response = await
fetch('https://api.jolpi.ca/ergast/f1/2024/constructors/'); const data = await
response.json(); // Parses the response as JSON const constructors =
data.MRData.ConstructorTable.Constructors; //access the specific data
constructors.forEach(teams => { Constructor(teams, counter2) counter2++ }); }
catch (error) { // Display an error message if the fetch fails const
errorMessage = document.createElement('div'); errorMessage.textContent =
`Gah, it's not working! Error: ${error.message}`;
app.appendChild(errorMessage); } }; function Constructor(teams, counter2) {
const card = document.createElement('div'); card.setAttribute('class', 'card');
const constructorsName = document.createElement('h1');
constructorsName.textContent = `${teams.name}`; const constructorsimage
= document.createElement('img') constructorsimage.src =
"teamphotos/"+ConImages[counter2]; const teamNationality =
document.createElement('h2') teamNationality.textContent = `Nationality:
${teams.nationality}`; container2.appendChild(card); // Append card to
```

```
container card.appendChild(constructorsName); //Append data to the card
card.appendChild(constructorsimage); card.appendChild(teamNationality); }
//FetchTrackData function fetches the data from the URL.json async function
fetchTracks() { let counter3 = 0 const ScheduleTitle =
document.createElement('img'); ScheduleTitle.src = 'F1 Schedule 2024.png';
container3.appendChild(ScheduleTitle); try { const response = await
fetch('https://api.jolpi.ca/ergast/f1/2024/'); const data = await response.json();
const Track = data.MRData.RaceTable.Races; Track.forEach(tracks => {
Tracks(tracks, counter3) counter3++ }); } catch (error) { // Display an error
message if the fetch fails const errorMessage =
document.createElement('div'); errorMessage.textContent = `Gah, it's not
working! Error: ${error.message}`; app.appendChild(errorMessage); } };
function Tracks(tracks, counter3) { let FP2 = ('') let FP3 = ('') let Quli = ('') let
Sprint =('') let SQ1 = ('') const card = document.createElement('div');
card.setAttribute('class', 'card'); // Styling class for each movie card const
GPName = document.createElement('h1'); GPName.textContent =
`${tracks.raceName}`; const trackimage = document.createElement('img')
trackimage.src = "trackphotos/"+TrackImages[counter3]; const TrackName =
document.createElement('h2') TrackName.textContent = `Circuit:
${tracks.Circuit.circuitName}`; const GPRace = document.createElement('h3')
GPRace.textContent = `Race Date & Time: ${tracks.date} -- ${tracks.time}`;
const FP1 = document.createElement('h3') FP1.textContent = `FP1 Date &
Time: ${tracks.FirstPractice.date} -- ${tracks.FirstPractice.time}`;
console.log(tracks.Qualifying) container3.appendChild(card); // Append card
to container card.appendChild(GPName); card.appendChild(trackimage)
card.appendChild(TrackName); card.appendChild(GPRace);
card.appendChild(FP1) if (tracks.Sprint) //If statement to check the different
race formats between weekends.. Sprint race weekend and a typical Race
Weekend { Quli = document.createElement('h3') Quli.textContent = `Qulifying
Date & Time: ${tracks.Qualifying.date} -- ${tracks.Qualifying.time}`; Sprint =
document.createElement('h3') Sprint.textContent = `Sprint Date & Time:
${tracks.Sprint.date} -- ${tracks.Sprint.time}`; SQ1 =
document.createElement('h3') SQ1.textContent = `Sprint Date & Time:
${tracks.SprintQualifying.date} -- ${tracks.SprintQualifying.time}`;
card.appendChild(SQ1); card.appendChild(Sprint); //and appends the correct
race format card.appendChild(Quli); } else { FP2 =
document.createElement('h3') FP2.textContent = `FP2 Date & Time:
${tracks.SecondPractice.date} -- ${tracks.SecondPractice.time}`; FP3 =
document.createElement('h3') FP3.textContent = `FP3 Date & Time:
${tracks.ThirdPractice.date} -- ${tracks.ThirdPractice.time}`; Quli =
document.createElement('h3') Quli.textContent = `Qulifying Date & Time:
${tracks.Qualifying.date} -- ${tracks.Qualifying.time}`; card.appendChild(FP2);
card.appendChild(FP3); //appends the correct race format
card.appendChild(Quli); } } //FetchStandings function fetches the data from
the URL.json async function fetchStandings() { const StandingsTitle =
document.createElement('img'); StandingsTitle.src = '2024 Drivers’
Standings.png'; container4.appendChild(StandingsTitle); let count = 1 try {
```

```
const response = await
fetch('https://api.jolpi.ca/ergast/f1/2024/driverstandings/'); const data = await
response.json(); // Parses the response as JSON console.log(data) const
standings = data.MRData.StandingsTable.StandingsLists[0].DriverStandings //
access the correct data for sorting and display console.log(standings)
standings.forEach(driver => { console.log(`Name: ${driver.Driver.givenName}
${driver.Driver.familyName}`); //writes the data to the console
console.log(`Ponits: ${driver.points}`); console.log(`Wins: ${driver.wins}`);
console.log('---'); createStandings(driver, count) count++ }); } catch (error) { //
Display an error message if the fetch fails const errorMessage =
document.createElement('div'); errorMessage.textContent = `Gah, it's not
working! Error: ${error.message}`; app.appendChild(errorMessage); } }
function createStandings(driver, count) { const card =
document.createElement('div'); card.setAttribute('class', 'card'); const
driverName = document.createElement('h2'); driverName.textContent =
`${count} : ${driver.Driver.givenName} ${driver.Driver.familyName}` // Set the
title for each card const Points = document.createElement('h3')
Points.textContent = `Points : ${driver.points}`; const Wins =
document.createElement('h3') Wins.textContent = `Wins : ${driver.wins}`;
container4.appendChild(card); // Append card to container
card.appendChild(driverName); card.appendChild(Points);
card.appendChild(Wins); };
```

Sources

100% Plagiarized

Display a loading message while fetching data. Display an error message if the fetch fails. [object Object],[object Object],[object Object], ...

https://webpilot.ai/writeDetail/39ccc98b-f755-4f13-bd17-ede0c05befaf

Home Blog Testimonials About Us Privacy Policy