

# コードで工夫したところ

- 1. 車両をプルダウンで選択したとき、関数showFrCaliperInfoFromSelection()を呼び出し。選択時にvalueの情報を渡す。Valueの値は、配列vehiclesのインデックス情報になるように設定。

```
<!-- 車両を選択 -->
<div class="vehicle-selection">
  <label for="vehicleSelect">車両選択:</label>
  <select id="vehicleSelect" onchange="showVehicleInfoFromSelection()">
    <option value="">車両を選択してください</option>
    <option value="0">ハイエースワゴン</option>
    <option value="1">クラウン</option>
    <option value="2">コスタ</option>
    <option value="3">シエンタ</option>
    <option value="4">ノア</option>
    <option value="5">アルファード</option>
  </select>
  <div id="vehicleInfo">
    <!-- ここに選択した車両情報を表示 -->
  </div>
</div>
```

```
let vehicle1 = {
  name: "ハイエースワゴン",
  emptyWeight: "2040",
  GVM: "2590",
  centerOfGravityHeight: "800", //仮値
  wheelBase: "2570",
  vehicleWidth: "1880",
  frontAxleWeight: "1590",
  rearAxleWeight: "1000",
  tireDynamicLoadRadius: "340" //仮値
};

...

let vehicle6 = {
  name: "アルファード",
  emptyWeight: "2230",
  GVM: "2615",
  centerOfGravityHeight: "750", //仮値
  wheelBase: "3000",
  vehicleWidth: "1850",
  frontAxleWeight: "1615",
  rearAxleWeight: "1000",
  tireDynamicLoadRadius: "370" //仮値
};

// 車両の配列を定義
const vehicles = [vehicle1, vehicle2, vehicle3, vehicle4, vehicle5, vehicle6];
```

- 2. valueの値を読みとり、vehicles[valueの値]とすることで、プルダウンメニューと車両情報を紐づけ

```
//車両をプルダウンで選択
function showVehicleInfoFromSelection() {
  let vehicleIndex = document.getElementById("vehicleSelect").value; //選択した車両の設定したvalueを返す。ここではvehiclesのインデックス番号を返す
  let vehicle = vehicles[vehicleIndex]; //選択した車両の情報をvehicleに代入
  let info = document.getElementById("vehicleInfo"); //id:vehicleInfoに情報表示させるためinfoに代入
  info.innerHTML = vehicle.name + "の重量: " + vehicle.emptyWeight + "kg, GVM: " + vehicle.GVM + "kg, Fr軸重: " + vehicle.frontAxleWeight + "kg, Rr軸重: " + vehicle.rearAxleWeight + "kg, 重心高: " + vehicle.centerOfGravityHeight + "mm, ホイールベース: " + vehicle.wheelBase + "mm, タイヤ動荷重半径: " + vehicle.tireDynamicLoadRadius + "mm";
}
```

- 3. .innerHTMLを使って、選択した車両情報が表示されるようにした。

車両選択: ハイエースワゴン ▼

車両名	空車重量[kg]	GVM[kg]	Fr軸重[kg]	Rr軸重[kg]	重心高[mm]	ホイールベース[mm]	タイヤ動荷重半径[mm]
ハイエースワゴン	2040	2590	1590	1000	800	2570	340

If文でvehicle1が選択されたとき、  
・・・vehicle2が選択されたとき、  
・・・など繰り返しをせずに  
コードを短縮化できた

○他にやりたかったこと

- ・実際に選択した車両がブレーキを踏んだとき理想制動力、実制動力を使うとどういう挙動になるか？何秒でとまるか？など車のアニメーションで描画。

○プレゼンテーションを作ってみて

自分の頭でやりたかったことが、実際にできるようになったことを実感できた。  
上記のようにまだうまく作れないものもあるので、  
今後もっと勉強して、業務に役立つツールの開発をしていきたいと思いました。

## グラフについて

参考記事：<https://qiita.com/Haruka-Ogawa/items/59facd24f2a8bdb6d369>

- ・ グラフ描画はcanvas要素を使用。

```
<canvas id="myChart"></canvas>
```

- ・ Chart.jsを使ってグラフを書くため、**CDN**に公開されているjsファイルを読み込み

```
<script>  
src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.7.2/Chart.bundle.js"  
</script>
```

- ・ canvas要素を取得し、変数 ctx に入力

```
let ctx = document.getElementById("myChart"); // キャンバス要素を取得
```

- ・ 描画するグラフを、new Chart()によって設定。

```
let myLineChart = new Chart(ctx,  
{ type: タイプ, data: データ, options: オプション }
```

設定項目	設定内容
type	描画するグラフの種類
data	ラベルとデータセット
options	オプション設定