

Explicação do código

A explicação vai ocorrer por função, mas antes vou explicar a base que utilizei para resolver a maior parte do problema passado.

Para melhor visualização eu decidi utilizar um HTML como base para imprimir os dados que estavam sendo manipulados.

```
<!DOCTYPE html>
<html Lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Desafio</title>
</head>
<body>
  <p>v1.0.4</p>
  <div id="dados"></div>
  <h2>Marcas</h2>
  <div id="marca"></div>

  <script src="desafio.js"></script>
</body>
</html>
```

Usei no código base 2 divs para impressão dos dados retirados do Json.

```
function corrigirCaracteres(json) {
  // Substitui os caracteres "æ" por "a" e "ø" por "o" na propriedade "nome"
  json.forEach(veiculo => {
    veiculo.nome = veiculo.nome.replace(/æ/g, 'a').replace(/ø/g, 'o');
  });
  json.forEach(veiculo => {
    veiculo.vendas = parseInt(veiculo.vendas);
  });

  return json;
}
```

Como o próprio nome diz, aqui temos a função para corrigir os caracteres:

1. Na primeira linha temos o `forEach` que passará para cada elemento dentro do Array a instrução de substituir em nome tudo o que tiver os caracteres “æ” e “ø” por, respectivamente, “a” e “o”.
2. No segundo `forEach`, é usado para transformar em `Int` todas as vendas que estão presentes no segundo Json.

```
function corrigirMarca(json) {
  // Substituí os caracteres "æ" por "a" e "ø" por "o" na propriedade "marca"
  json.forEach(veiculo => {
    veiculo.marca = veiculo.marca.replace(/æ/g, 'a').replace(/ø/g, 'o');
  });

  return json;
}
```

Mesma coisa que o código acima, porém substitui em marca e não tem função para alterar os dados da venda em Int.

```
document.addEventListener("DOMContentLoaded", function () {
  let divDados = document.querySelector("#dados");

  fetch('broken_database_1.json')
    .then(response => response.json())
    .then(data => {
      const json = corrigirCaracteres(data);
      let jsonDados = JSON.stringify(json);
      divDados.innerHTML = jsonDados;
    })
    .catch(error => console.error(error));

  let divMarca = document.querySelector("#marca");

  fetch('broken_database_2.json')
    .then(response => response.json())
    .then(data => {
      console.log(data);
      const json = corrigirMarca(data);
      let jsonMarca = JSON.stringify(json);
      divMarca.innerHTML = jsonMarca;
    })
    .catch(error => console.error(error));
});
```

Aqui nós temos a função principal e que faz tudo acontecer no código:

1. Na primeira linha temos a variável `divDados` que serve para selecionar a div do HTML base para impressão através do Id (`#dados`).
2. Abaixo temos o `fetch` que tem a função de buscar o arquivo `json` para ser utilizado a seguir.

3. a função passada para o primeiro then recebe a resposta da requisição e a transforma em um objeto JSON usando o método `json()`. Esse objeto JSON é então passado para a função `corrigirCaracteres`, que provavelmente faz algum tipo de correção ou tratamento no JSON retornado. Em seguida, o JSON é transformado em uma string usando o método `JSON.stringify` e é inserido no elemento HTML `divDados` usando `innerHTML`.
4. Se ocorrer um erro durante a requisição, o catch é acionado e a mensagem de erro é exibida no console usando o método `console.error`.

Fiz algumas tentativas para exportação dos dados para Json novamente e infelizmente não consegui