

# Como mostrar informações com fetch()

## A maioria dos problemas envolvendo fetch pode ser reduzida para:

Usuário coloca informação → site usa fetch para puxar dados da informação de alguma API → site mostra resultado ao usuário

Isso significa que podemos seguir uma certa fórmula ao usar o fetch para este propósito, já que vários exercícios ficam extremamente parecidos.

Esta fórmula pode ser algo como, no JavaScript:

```
function mostrar(){
  // Pegar informação que o usuário escreveu
  const nome = document.getElementById('idNome').value;

  // Acessar link da API junto com a informação
  fetch(`https://algumaAPI.com/${nome}`, {
    method: 'GET',
    //headers,
    headers: {
      "Content-Type": "application/json",
    },
  })
  .then(response => response.json())
  .then(data => { // Processar o que a API retornou em JSON, talvez mostrar para o
usuário via getElementById
    if (data !== null){
      document.getElementById("idAlgumaCoisa").value = data.algumaInformacao;
      // (lembrando que para colocar texto dentro de um div em JavaScript se acessa o
innerHTML, e para outros elementos se acessa o value)
    }
  })
  .catch(erro => {
    document.getElementById("erro").innerHTML = "Erro ao listar:" + erro;
  });
}
```

Daí, a parte do HTML em si ficaria semelhante a :

```
Nome: <input type="text" id="idNome"><br>
<input type="button" value="Mostrar" onclick="mostrar()"><br>
```

## Exemplos:

### Mostrar a bandeira de um país

HTML:

```
Nome do País: <input type="text" id="idNome"><br>
<input type="button" value="Mostrar Bandeira" onclick="mostrarBandeira()"><br>
<img id="bandeira" src="" alt="Bandeira aparecerá aqui" width="200"><br>
```

JavaScript:

```
function mostrarBandeira(){
  const nome = document.getElementById('idNome').value;
  fetch(`https://restcountries.com/v3.1/name/${nome}`, {
    method: 'GET',
    headers: { "Content-Type": "application/json" }
  })
  .then(response => response.json())
  .then(data => {
    document.getElementById("bandeira").src = data[0].flags.png;
  })
  .catch(erro => {
    document.getElementById("bandeira").alt = "Erro ao buscar bandeira: " + erro;
  });
}
```

## Mostrar significado de uma palavra (dicionário)

HTML:

```
Palavra: <input type="text" id="idNome"><br>
<input type="button" value="Mostrar Significado"
onclick="mostrarSignificado()"><br>
Significado: <span id="significado"></span>
```

JavaScript:

```
function mostrarSignificado(){
  const nome = document.getElementById('idNome').value;
  fetch(`https://api.dictionaryapi.dev/api/v2/entries/en/${nome}`, {
    method: 'GET',
    headers: { "Content-Type": "application/json" }
  })
  .then(response => response.json())
  .then(data => {
    const definicao = data[0].meanings[0].definitions[0].definition;
    document.getElementById("significado").innerHTML = definicao;
  })
  .catch(erro => {
    document.getElementById("significado").innerHTML = "Erro ao buscar significado:
" + erro;
  });
}
```

# Converter moeda

HTML:

```
Valor em USD: <input type="text" id="idNome"><br>
<input type="button" value="Converter para BRL" onclick="converterMoeda()"><br>
Valor em BRL: <span id="valorConvertido"></span>
```

JavaScript:

```
function converterMoeda(){
  const valor = document.getElementById('idNome').value;
  fetch(`https://api.frankfurter.app/latest?amount=${valor}&from=USD&to=BRL`, {
    method: 'GET',
    headers: {
      "Content-Type": "application/json"
    }
  })
  .then(response => response.json())
  .then(data => {
    document.getElementById("valorConvertido").innerHTML = "R$ " +
data.rates.BRL.toFixed(2);
  })
  .catch(erro => {
    document.getElementById("valorConvertido").innerHTML = "Erro: " + erro;
  });
}
```

No fim, a única coisa que realmente varia entre cada um é o formato do JSON que a API retorna, para entendê-lo melhor é uma boa ideia abrir o link da API direto no Google