

Manipulação de API

em arquivos separados

```
index.html > JS main.js JS poke-api.js
index.html > html > body > script
22      <span class="name">Bulbasaur</span>
23
24      <div class="detail">
25
26          <ol class="types">
27              <li class="type">Grass</li>
28              <li class="type">Poison</li>
29          </ol>
30
31          
32      </div>
33  </li>
34 </ol>
35
36 </section>
37 <script src="./assets/js/poke-api.js"></script>
38 <script src="./assets/js/main.js"></script>
39 </body>
40 </html>
41
```

Importamos os arquivos no HTML, lembrando que o arquivo da **Api** deve vir primeiro, pois tem que ser carregado para que o próximo elemento que chamara receba os parâmetros.

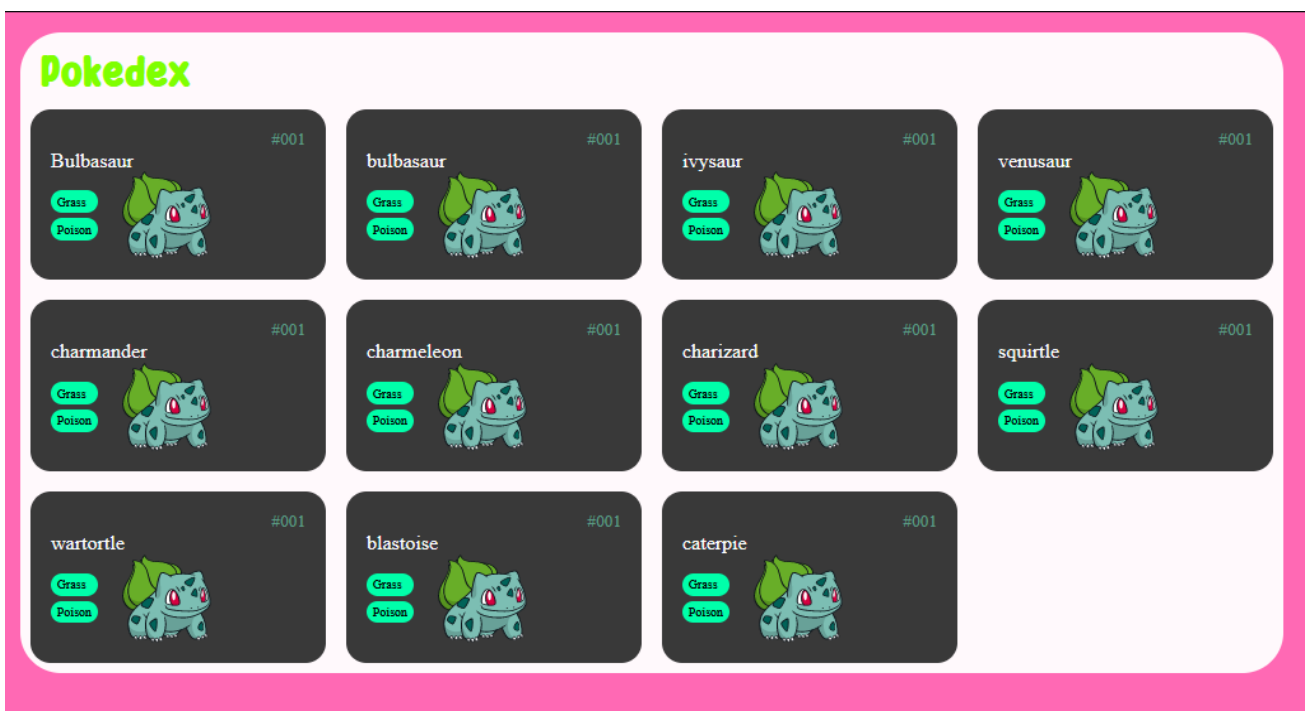
```
index.html JS main.js JS poke-api.js X
assets > js > JS poke-api.js > getPokemons
1  const pokeApi = {}
2
3  pokeApi.getPokemons = (offset = 0, limit = 10) => {
4      const url = `https://pokeapi.co/api/v2/pokemon?offset=${offset}&limit=${limit}`
5
6      return fetch(url)
7          .then(result => result.json())
8          .then(result => result.results)
9          .catch(error => console.log(error))
10 }
```

Como podemos ver aqui estamos criando um objeto que terá um parâmetro como função, sendo o comando principal para executar API, tendo como padrão elementos de busca. Além disso, está retornando a busca dos dados da Api, basta solicitar o método para acessar os dados.

```
index.html • JS main.js × JS poke-api.js
assets > js > JS main.js > then() callback

17         1.svg" alt="${pokemons.name}">
18     </div>
19 </li>
20
21 }
22
23 const elementoHtmlListPokemon = document.getElementById('pokemonList')
24
25 pokeApi.getPokemons().then((repostt) => {
26     let pkemonlist = []
27     repostt.forEach(pokemon => {
28         console.log(pokemon.name)
29         pkemonlist.push(pokemonsHtml(pokemon))
30     });
31
32     pkemonlist.map(list => {
33         elementoHtmlListPokemon.innerHTML += list
34     })
35
36 })
37
38
39 console.log(dados)
```

Aqui temos o acesso Api, criamos uma then para obter os dados e para acessar cada um individualmente utilizamos o forEach. Depois de termos acessos aos dados, colocamos dentro de uma lista(array) ordenada. Logo a pois isso acessamos individualmente novamente cada dado através da função map.



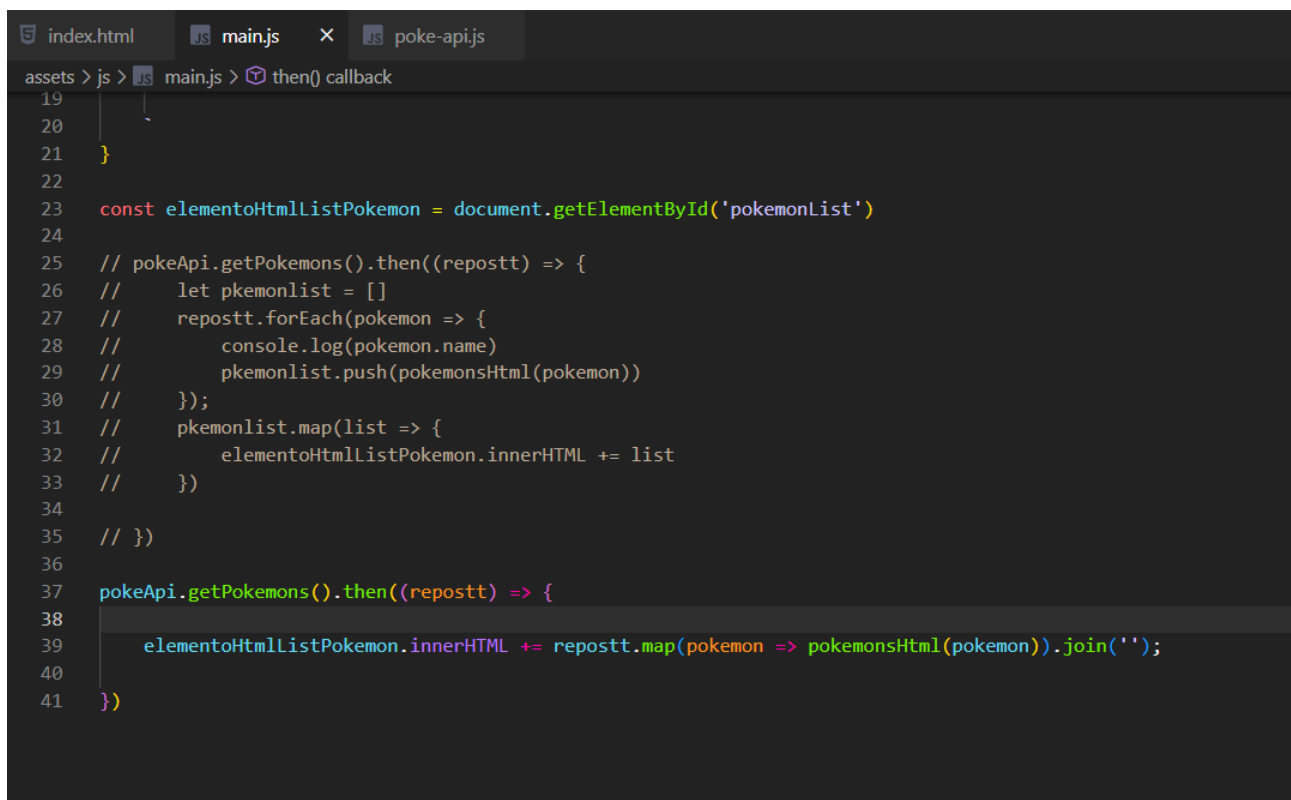
Obtemos esse seguinte resultado, foram criados cada pokemon com o nome diferente.

Obs: Não foi mostrado a função que cria elementos HTML para mostrar.

Real objeto de aguardar os dados obtido da Api em outra lista, foi devido à necessidade de não ter que realizar diversa vez a requisição dos dados para ser mostrado. Foi apenas solicitado 1 vez os dados e salvo em uma lista os dados para assim ser acesso através do map.

Não foi criado a função da troca de imagem dos elementos HTML, apenas a troca do nome.

Maneira de encurtar o código e ter o mesmo funcionamento.

A screenshot of a code editor with three tabs: 'index.html', 'main.js', and 'poke-api.js'. The 'main.js' tab is active, showing a JavaScript file. The code in the editor is as follows:

```
assets > js > main.js > then() callback
19
20
21 }
22
23 const elementoHtmlListPokemon = document.getElementById('pokemonList')
24
25 // pokeApi.getPokemons().then((repostt) => {
26 //   let pkemonlist = []
27 //   repostt.forEach(pokemon => {
28 //     console.log(pokemon.name)
29 //     pkemonlist.push(pokemonsHtml(pokemon))
30 //   });
31 //   pkemonlist.map(list => {
32 //     elementoHtmlListPokemon.innerHTML += list
33 //   })
34 // })
35
36
37 pokeApi.getPokemons().then((repostt) => {
38
39   elementoHtmlListPokemon.innerHTML += repostt.map(pokemon => pokemonsHtml(pokemon)).join('');
40
41 })
```

Resumidamente, foi utilizado uma função para mapear os objetos que recebemos do JavaScript e realizamos a conversão dos objetos com uma função que transforma o objeto em elementos essenciais para o HTML. Depois utilizamos uma função chamada `join("")` para transformar em uma lista enorme em HTML, e através do método `elementoHtmlListPokemon` o passamos para o HTML que renderiza o conteúdo.