

PRÁTICA LABORATORIAL 09

Objetivos:

- Construção de ficheiros JSON
- Manipulação de dados em JSON (stringify e parse)
- Ligação a API

Quando uma página é carregada o navegador cria um DOM (Document Object Model) da página. O HTML DOM é construído como sendo uma árvore de objetos

PARTE I – STRINGIFY E PARSE

Num script adicione um objeto com o nome pessoa e os atributos nome e idade (os valores dos atributos ficam ao seu critério).

1. Realize um “console.log” para apresentar todos os dados da pessoa;
2. Realize um “console.log” para apresentar apenas o nome da pessoa.

Aplique agora o comando “pessoa = JSON.stringify(pessoa)” e verifique o resultado. NOTA: substitua “pessoa” pelo nome que atribuiu à variável anterior.

1. Realize um “console.log” para apresentar todos os dados da pessoa;
2. Realize um “console.log” para apresentar apenas o nome da pessoa.
3. Compare o resultado destes dois console.log com os realizados anteriormente.

De notar que o comando “stringify” converte um objeto numa string, daí a impossibilidade de aceder ao atributo nome.

Aplique agora o comando “pessoa = JSON.parse(pessoa)” e verifique o resultado. NOTA: substitua “pessoa” pelo nome que atribuiu à variável anterior.

1. Realize um “console.log” para apresentar todos os dados da pessoa;
2. Realize um “console.log” para apresentar apenas o nome da pessoa.

O método parse permite converter um código JSON válido gravado em texto num objeto.

PARTE II – CONSTRUÇÃO DE FICHEIROS JSON

Crie um novo ficheiro com o nome `personas.json` e atribua-lhe o seguinte conteúdo:

```
{
  "personas": [
    {
      "nome": "Ana",
      "idade": 20
    },
    {
      "nome": "Miguel",
      "idade": 30
    },
    {
      "nome": "Carlos",
      "idade": 26
    }
  ]
}
```

O ficheiro identifica um primeiro array de pessoas em que cada elemento será um objeto com as propriedades `nome` e `idade` e os seus valores. Desta forma é possível criar uma base de dados de elementos para ser usada e distribuída.

PARTE III – LEITURA DE JSON LOCAL OU REMOTO

Cole e teste o código seguinte:

```
var xmlhttp = new XMLHttpRequest();
var url = "pessoas.json";

xmlhttp.onreadystatechange = function() {
  if (this.readyState == 4 && this.status == 200) {
    var myArr = JSON.parse(this.responseText);
    console.log(myArr);
  }
};
xmlhttp.open("GET", url, true);
xmlhttp.send();
```

Disponível em: https://www.w3schools.com/js/js_json_http.asp

Verifique que com a execução (NOTA: tem obrigatoriamente de usar o Live Server porque o método XMLHttpRequest assim o obriga), o código JSON do documento pessoas.json é transportado para a variável myArr e assim podemos utilizá-lo como entendermos.

EXERCÍCIOS

1. Crie um ficheiro JSON com a identificação seguinte:
 - a. Nome: João Silva; Idade: 25; Morada → Rua: Rua do Sobe e Desce; Morada → Cidade: Porto; Filhos (array): “Joana” e “Martim”.
 - b. Nome: António Cardoso; Idade: 33; Morada → Rua: Rua do Desce e Sobe; Morada → Cidade: Porto; Filhos (array): “Cláudio”.
 - c. Nome: Sandra Peixoto; Idade: 29; Morada → Rua: Rua do nem Sobe nem Desce; Morada → Cidade: Porto; Filhos (array): “Sofia”.
2. Valide a construção do ficheiro anterior no site JSONLint (<https://jsonlint.com/>) ou similar.
3. Crie uma página que apresente os dados constantes no ficheiro JSON lido num formato semelhante ao apresentado abaixo.

- João Silva (25)
- António Cardoso (33)
- Sandra Peixoto (29)

4. Altere o script anterior para que sejam associados também os filhos de cada pessoa, semelhante ao apresentado de seguida:

- João Silva (25)
 - Joana
 - Martim
- António Cardoso (33)
 - Cláudio
- Sandra Peixoto (29)
 - Sofia

SABER MAIS – TRABALHO AUTÓNOMO

1. Utilize um ficheiro JSON disponível online e apresente a sua informação da forma que achar mais conveniente. Pode usar como exemplo um dos seguintes:
 - <https://rapidapi.com/skyscanner/api/skyscanner-flight-search>
 - <https://rapidapi.com/community/api/open-weather-map>
 - <https://rapidapi.com/api-sports/api/api-football>
 - <https://rapidapi.com/theapiguy/api/the-cocktail-db>
 - <https://rapidapi.com/apilayernet/api/rest-countries-v1>
 - <https://rapidapi.com/apidojo/api/yahoo-finance1>
 - <https://rapidapi.com/ajith/api/love-calculator>
 - <https://rapidapi.com/BigLobster/api/url-shortener-service>

Bom trabalho! ☺