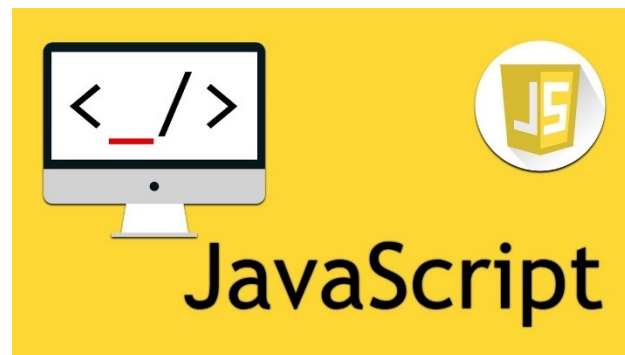




Programação Web





JS – JavaScript

JSON



JSON – JavaScript Object Notation



- O JSON é um formato de notação simples para troca de dados entre sistemas ou aplicações.
- Uma SPA, por exemplo, utiliza o JSON para atualizar a informação a mostrar ao utilizador, sem necessidade de recarregar outros recursos como HTML, CSS ou JS.
- A simplicidade do JSON tem resultado na sua disseminação e utilização como formato alternativo a outros formatos de troca de dados entre máquinas, como por exemplo o XML.
- Uma das vantagens reivindicadas do JSON sobre o XML é o fato de ser muito mais fácil de escrever um analisador (*parser*) de JSON. Isto foi importante para a sua aceitação.
- Atualmente é muito comum as aplicações fornecerem dados no formato JSON tanto quanto no formato XML.
 - Exemplos: o Google Maps, lastFM (<http://www.last.fm/api>)

JSON e XML - Semelhanças



- Ambos os formatos são escritos em texto simples (*plain text*).
- As estruturas de dados são auto-explicativas, o que ajuda na sua interpretação, quer por máquinas, quer por humanos.
- As estruturas de dados de ambas são hierárquicas (valores dentro de valores).
- Ambos podem ser carregadas para a aplicação através de AJAX ou de Fetch API

JSON e XML - Diferenças



- As estruturas de dados JSON:
 - não utilizam um marcador para sinalizar o fim
 - têm tamanho inferior às XML
 - podem utilizar conjuntos (*arrays*)
 - podem ser interpretadas por uma função JavaScript standard - `JSON.parse(texto)`
 - As estruturas XML têm de ser interpretadas por um interpretador (*parser*) XML

XML - Exemplo



- Dados sobre um conjunto de empregados em formato XML.

```
<employees>
  <employee>
    <firstName>John</firstName> <lastName>Doe</lastName>
  </employee>
  <employee>
    <firstName>Anna</firstName> <lastName>Smith</lastName>
  </employee>
  <employee>
    <firstName>Peter</firstName> <lastName>Jones</lastName>
  </employee>
</employees>
```

JSON - Exemplo



- Os dados anteriores em formato JSON.

```
{ "employees": [  
  { "firstName": "John", "lastName": "Doe"},  
  { "firstName": "Anna", "lastName": "Smith"},  
  { "firstName": "Peter", "lastName": "Jones"}  
]}
```

- Se não necessitarmos do nome do conjunto, podemos escrever:

```
[  
  { "firstName": "John", "lastName": "Doe"},  
  { "firstName": "Anna", "lastName": "Smith"},  
  { "firstName": "Peter", "lastName": "Jones"}  
]
```

JSON – Formato dos Dados



- Par Chave-Valor
 - Uma estrutura de dados em formato JSON é escrita em pares chave-valor.
 - Um par chave-valor consiste no nome de um campo (entre aspas), seguido pelo sinal “:” e terminado por um valor, entre aspas, caso seja uma string.
 - Exemplo:
 - `"firstName":"John"`

JSON – Tipos de Dados



- Os valores nas estruturas de dados JSON podem ser dos seguintes tipos:
 - número – inteiro ou real
 - string – escrita entre aspas
 - booleano – tomando os valores true ou false
 - conjunto - colocado dentro de parêntesis retos [...]
 - objeto – colocado dentro de chavetas { ... }
 - null – quando o valor for nulo.

JSON – Objectos



- Tal como referido, os objetos JSON são descritos dentro de chavetas.
- Tal como nos objetos JavaScript, os objetos JSON podem conter múltiplos pares chave-valor.
- Exemplos:

```
{ "name": "Joao Manuel Azevedo",  
  "street": "Avenida Calouste Gulbenkian, 10",  
  "phone": "+351 234 372 956"  
}
```

OU

```
{ "name": "Joao Manuel Azevedo",  
  "home": { "street": "Avenida Calouste Gulbenkian",  
            "number": 10 },  
  "phone": "+351 234 372 956"  
}
```

JSON – Objectos



- Exemplo - HTML

```
<body>
  <div class="container mt-3">
    <h2>Person Information</h2>
    <div class="mb-3 mt-3">
      <label for="name">Name:</label>
      <input id="name" type="text" class="form-control" placeholder="Enter name">
    </div>
    <div class="mb-3">
      <label for="home">Address:</label>
      <input id="home" type="text" class="form-control" placeholder="Enter home address">
    </div>
    <div class="mb-3">
      <label for="phone">Phone:</label>
      <input id="phone" type="text" class="form-control" placeholder="Enter phone number">
    </div>
    <button class="btn btn-primary" onClick="fill()">Fill</button>
  </div>

  <!-- My JavaScript -->
  <script src="ex01.js"></script>
</body>
```

JSON – Objectos



- Exemplo - JS

```
var person = '{ \
    "name": "Joao Manuel Azevedo", \
    "home": {"street":"Avenida Calouste Gulbenkian", "number": 10}, \
    "phone": "+351 234 372 956" \
}';

function fill() {
    let data = JSON.parse(person);
    document.getElementById("name").value = data.name;
    document.getElementById("home").value = data.home.street + ", " + data.home.number;
    document.getElementById("phone").value = data.phone;
}
```

JSON – Conjuntos de Objectos



- Os conjuntos (*arrays*) de objetos JSON são descritos dentro de parênteses retos.

- Exemplo:

```
[  
  {"firstName":"John", "lastName":"Doe"},  
  {"firstName":"Anna", "lastName":"Smith"},  
  {"firstName":"Peter", "lastName":"Jones"}  
]
```

- Neste exemplo, o conjunto possui 3 objetos JSON, contendo informação sobre os primeiro e último nomes de pessoas.

JSON – Conjuntos de Objectos



- Exemplo - HTML

```
<body>
  <div class="container mt-3">
    <h2>Persons Information</h2>
    <table class="table table-striped">
      <thead>
        <tr>
          <th>Firstname</th>
          <th>Lastname</th>
          <th>Email</th>
        </tr>
      </thead>
      <tbody id="pessoal">
      </tbody>
    </table>
    <button class="btn btn-primary" onClick="fill()">Fill</button>
  </div>
  <!-- My JavaScript -->
  <script src="ex02.js"></script>
</body>
```

JSON – Conjuntos de Objectos



- Exemplo - JS

```
var persons = '[ \n
  { "fname": "Joao", "lname": "Azevedo", "email": "jazevedo@mail.pt" }, \n
  { "fname": "Manuel", "lname": "Pinto", "email": "mpinto@mail.pt" }, \n
  { "fname": "Gustavo", "lname": "Soares", "email": "gsoares@mail.pt" } \n
]';

function fill() {
  let data = JSON.parse(persons);
  let htm = "";
  for (i=0; i < data.length; i++)
  {
    htm += "<tr>";
    htm += "<td>" + data[i].fname + "</td>";
    htm += "<td>" + data[i].lname + "</td>";
    htm += "<td>" + data[i].email + "</td>";
    htm += "</tr>";
  }
  document.getElementById("pessoal").innerHTML = htm;
}
```

Fetch API – Carregar JSON



- Exemplo de uma função para carregar JSON a partir de um ficheiro, para uso posterior.

```
// Return JSON from file
// @file - data filename
async function LoadData(file) {
  let url = "data/" + file;
  let obj = await fetch(url);
  let data = await obj.json();
  return data;
}
```