



CAPÍTULO

JAVASCRIPT

JPSantos
2023

8. JAVASCRIPT

A linguagem de programação JavaScript permite escrever scripts, ficheiros de texto, com programas que podem ser interpretados pelos Browsers WEB (Front-end) ou por outras aplicações como o NODE.JS que em “Back-end” podem interpretar programas escritos em Javascript.

O NODE.JS é um programa executável, desenvolvido em C, que implementa um “JavaScript engine” que ao interpretar os ficheiros em Javascript pode, de acordo com o programa em Javascript, fazer operações aritméticas, escrever em bases de dados, realizar comparações, operações booleanas, instruções de controlo de fluxo, ..., receber pedidos pela internet, tratá-los e responder a esses pedidos.

Quando os browsers WEB abrem ficheiros HTML com scripts em Javascript (Front-end), os seus “JavaScript Engines” interpretam esses scripts e de acordo com eles podem formatar, animar e tornar as páginas WEB interativas. Se esses scripts possuírem instruções Assíncronas em Javascript e XML (AJAX), o browser pode por exemplo estabelecer, em simultâneo, novas ligações a servidores HTTP/TCP/IP, e requererem novos dados em background, sem ser necessário recarregar toda a página WEB no browser.

Em Javascript podem ser desenvolvidas páginas e aplicações WEB (Web/Mobile Apps), programas para em tempo real trabalhar na internet (Real-time networking Apps), e Jogos.

<https://www.ecma-international.org/publications/standards/Ecma-262.htm>

JavaScript Tutorial for Beginners: Learn JavaScript in 1 Hour [2020]

<https://www.youtube.com/watch?v=W6NZfCO5SIk&t=6s>

8.1.Código javascript

8.1.1. Tipos de variáveis

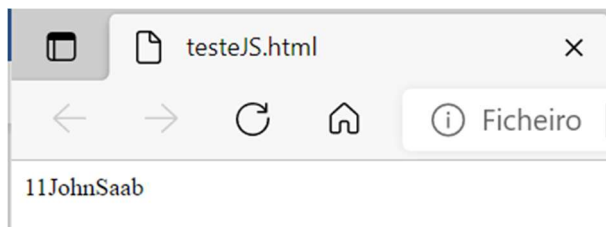
Experimente editar um ficheiro de texto com a extensão .html (por exemplo: testeJS.html) com o script seguinte:

```
<html>
<script>
// Comentario em javascript
var x = 5;
var y = 6;
var person = "John Doe", carName = "Volvo", price = 200;
var z = x + y;
var cars = ["Saab", "Volvo", "BMW"];
var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};

document.write(x + y);
document.write(person.firstName);
document.write(cars[0]);

</script>
</html>
```

Quando abrir este ficheiro com o Browser aparecerá:



https://www.w3schools.com/js/js_variables.asp

8.1.2. Operadores atribuição

Operator	Example	Same As
=	x = y	x = y
+=	x += y	x = x + y
-=	x -= y	x = x - y
*=	x *= y	x = x * y
/=	x /= y	x = x / y
%=	x %= y	x = x % y
<<=	x <<= y	x = x << y
>>=	x >>= y	x = x >> y
>>>=	x >>>= y	x = x >>> y
&=	x &= y	x = x & y
^=	x ^= y	x = x ^ y
=	x = y	x = x y
**=	x **= y	x = x ** y

8.1.3. Operadores aritméticos

Operadores aritméticos

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulus (Remainder)
++	Increment
--	Decrement

8.1.4. Operadores relacionais/comparação

Os operadores booleanos comparam dois valores e com base na comparação realizada retornam um valor booleano “Verdadeiro” ou “Falso”.

Operator	Description
==	equal to

==	equal value and equal type
!=	not equal
!==	not equal value or not equal type
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to
?	ternary operator

8.1.5. Operadores lógicos

Operator	Description
&&	logical and
	logical or
!	logical not

8.1.6. Operadores bit a bit

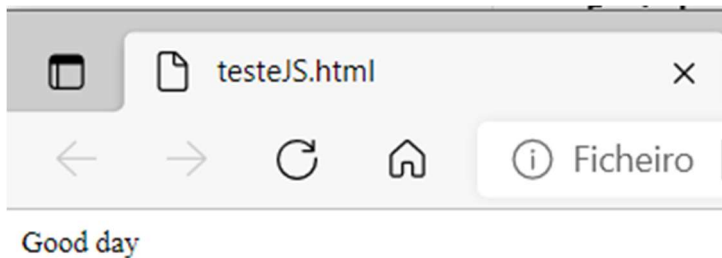
Operator	Description	Example	Same as	Result	Decimal
&	AND	5 & 1	0101 & 0001	0001	1
	OR	5 1	0101 0001	0101	5
~	NOT	~ 5	~0101	1010	10
^	XOR	5 ^ 1	0101 ^ 0001	0100	4
<<	Zero fill left shift	5 << 1	0101 << 1	1010	10
>>	Signed right shift	5 >> 1	0101 >> 1	0010	2
>>>	Zero fill right shift	5 >>> 1	0101 >>> 1	0010	2

8.1.7. IF

```
<html>
<script>
var greeting;
var hour=10;

if (hour < 18) {
    greeting = "Good day";
    document.write(greeting);
} else {
    greeting = "Good evening";
    document.write(greeting);
}

</script>
</html>
```

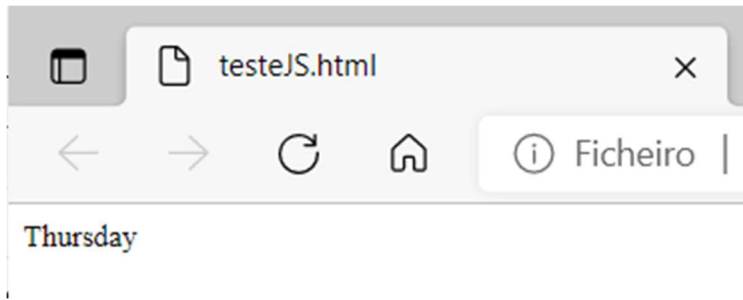


8.1.8. Switch

```
<html>
<script>
switch (new Date().getDay()) {
    case 0:
        day = "Sunday";
        break;
    case 1:
        day = "Monday";
        break;
    case 2:
        day = "Tuesday";
        break;
    case 3:
        day = "Wednesday";
        break;
    case 4:
        day = "Thursday";
        break;
    case 5:
        day = "Friday";
        break;
    case 6:
        day = "Saturday";
    }

    document.write(day);
</script>
```

</html>



8.1.9. For

```
for (i = 0; i < 5; i=i+1) {  
    text += "The number is " + i + "<br>";  
}
```

8.1.10. While

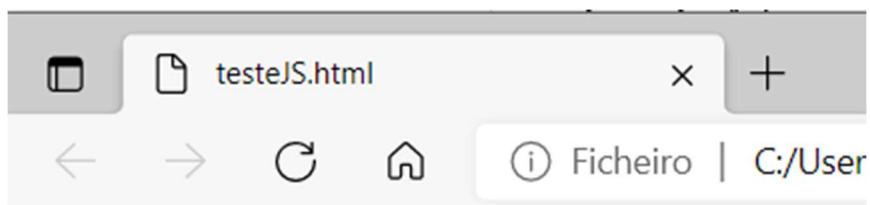
```
i=0;  
while (i < 10) {  
    text = text + "The number is " + i + "<BR>";  
    i++;  
}
```

8.1.11. Funções e procedimentos

As funções e os procedimentos permitem “agrupar” código, tornando mais fácil a sua organização e reutilização ao longo do programa, mas só as funções podem retornar valores.

https://www.w3schools.com/js/js_functions.asp

```
<!DOCTYPE html>  
<html>  
<body>  
<h2>JavaScript Functions</h2>  
<p>This example calls a function to convert from Fahrenheit to Celsius:</p>  
<p id="demo"></p>  
  
<script>  
function toCelsius(f) {  
    return (5/9) * (f-32);  
}  
document.getElementById("demo").innerHTML = toCelsius(77);  
</script>  
  
</body>  
</html>
```



JavaScript Functions

This example calls a function to convert from Fahrenheit to Celsius:

25

8.2. Frontend (Browser)

8.2.1. Eventos no Browser que podem chamar funções Javascript

OnClick

Quando o botão presente na página HTML visualizada pelo Browser for selecionado pelo utilizador, a função em javascript será executada pelo Browser.

```
<!DOCTYPE html>
<html>
<body>

    <p>Click the button to display the date.</p>
    <p id="demo"></p>
    <button onclick="displayDate()" >The time is?</button>
<script>
    function displayDate() {
        document.getElementById("demo").innerHTML = Date();
    }
</script>
</body>
</html>
```

https://www.w3schools.com/js/js_events.asp

Event	Description
<i>onchange</i>	<i>An HTML element has been changed</i>
<i>onclick</i>	<i>The user clicks an HTML element</i>
<i>onmouseover</i>	<i>The user moves the mouse over an HTML element</i>
<i>onmouseout</i>	<i>The user moves the mouse away from an HTML element</i>
<i>onkeydown</i>	<i>The user pushes a keyboard key</i>
<i>onload</i>	<i>The browser has finished loading the page</i>

Quando o botão “submit” presente no browser for premido pelo utilizador, o Browser chama a função “validateForm()”

```
function validateForm() {  
    var x = document.forms["myForm"]["fname"].value;  
    if (x == "") {  
        alert("Name must be filled out");  
        return false;  
    }  
}  
  
<form name="myForm" action="/action_page.php" onsubmit="return validateForm()" method="post">  
    Name: <input type="text" name="fname">  
    <input type="submit" value="Submit">  
</form>
```

```
<input id="id1" type="number" min="100" max="300" required>  
<button onclick="myFunction()">OK</button>
```

```
<p id="demo"></p>
```

```
<script>  
function myFunction() {  
    var inpObj = document.getElementById("id1");  
    if (!inpObj.checkValidity()) {  
        document.getElementById("demo").innerHTML = inpObj.validationMessage;  
    }  
}  
</script>
```

8.2.2. Editar elementos HTML em javascript

```
document.createElement("article");  
  
document.getElementById("demo").style.fontSize = "25px";  
document.getElementById("demo").style.color = "red";  
document.getElementById("demo").style.backgroundColor = "yellow";  
document.getElementById("image").src = "picture.gif";  
document.getElementById("demo").innerHTML = "Hello JavaScript!"
```

8.2.3. Exemplo I - Hello

JavaScript Tutorial for Beginners: Learn JavaScript in 1 Hour [2019] ! Mosh (48 min Muito bom OK)
<https://www.youtube.com/watch?v=W6NZfCO5SIk>
(começa por usar browser Chrome para executar o javascript)
(instalar VStudio Code “code.visualstudio.com”, podia ser o sublime)
(começa pelo HTML, javascript, variáveis, funções, e como chamar o javascript no documento html)

---- Ficheiro HTML com javascript---

```
<!DOCTYPE html>
<html lang="en">
<head><title>Titulo</title></head>
<body>
  <script>
    var person = {name: 'jose',age: 55};
    function greet(name) {console.log('Hello ' + name);}
    function quadrado(num) {return num * num;}
    console.log(quadrado(2));
    document.getElementById(""); // completar o código
  ...
  </script>
</body>
</html>
```

8.2.4. Exemplo II

JavaScript: Understanding the Weird Parts - The First 3.5 Hours (OK)
https://www.youtube.com/watch?v=Bv_5Zv5c-Ts
(O vídeo explica as variáveis locais e globais, stack, e tratamento de eventos “click” e “HTTP Request”)
(variáveis, operadores, !==, ===, precedência, coercion, associativity, vários ficheiros js <script src=...js></script>)
(funções, objectos-literal syntax)

```
<!DOCTYPE html>
<html lang="en">
<head><title>Titulo</title></head>

<body>
  <script>
    function waitThreeSeconds() {
      var ms = 3000 + new Date().getTime();
      while (new Date() < ms) { }
      console.log('finished function 3 seg');
    }

    function clickHandler() {console.log('click event!');}

    document.addEventListener('click', clickHandler);
    waitThreeSeconds();
    console.log('finished execution');
  </script>
</body>
</html>
```

8.2.5. Exemplo III - AJAX e JSON

Asynchronous Javascript and XML (AJAX)

De acordo com as instruções AJAX na página web, o browser faz em background um XMLHttpRequest para pedir novos dados a um servidor WEB .

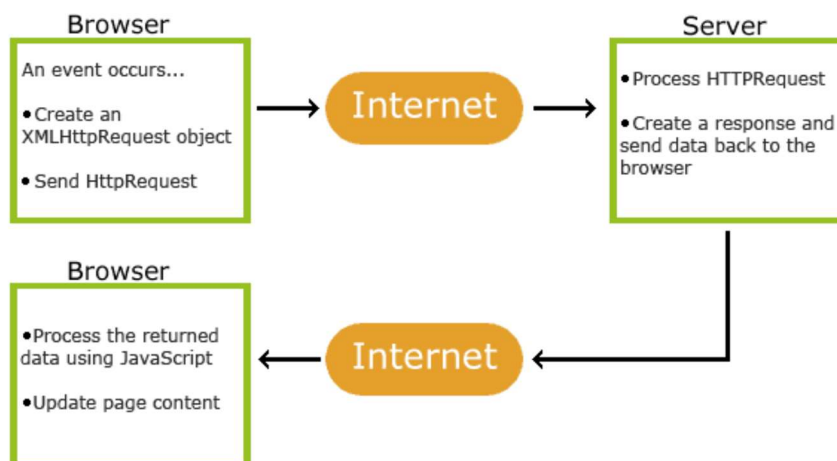
```
var XHTTP = new XMLHttpRequest();    // O Browser cria um objecto do tipo XMLHttpRequest

XHTTP.open("GET", "http://localhost/ajax_info.html", true); // o objecto retém a informação

XHTTP.send();                // O objecto estabelece uma ligação TCP/IP com o servidor e envia uma
                             // mensagem HTTP do tipo GET para o servidor "http://localhost"
                             // a pedir um documento, neste caso "ajax_info.html"

XHTTP.onreadystatechange = function() { } // contém a referencia à função que será chamada
                                           // pelo objecto XHTTP quando chegar ao browser
                                           // a resposta do servidor.
```

How AJAX Works



https://www.w3schools.com/js/js_ajax_intro.asp

Esta função Javascript, usa um objecto do tipo XMLHttpRequest para pedir ao servidor web um novo documento, neste caso o ficheiro “ajax_info.html”

Quando o servidor Web responde com o ficheiro pedido HTTP 200 Ok o objecto chama a função `function(){ ... }`

```
function loadDoc() {
    var XHTTP = new XMLHttpRequest();

    XHTTP.open("GET", "http://localhost/ajax_info.html", true);
    XHTTP.send();                // estabelece a ligação TCP/IP com o servidor e envia GET ...

    XHTTP.onreadystatechange = function() { // esta função é chamada pelo objecto XHTTP quando chega
                                           // a resposta HTTP vinda do servidor WEB
        if (this.readyState == 4 && this.status == 200) {
            document.getElementById("demo").innerHTML = this.responseText;
        } // if
    }; // function
}
```

----- HTML -----

```
<!DOCTYPE html>
<html>
<body>
  <div id="demo">
    <h2>Let AJAX change this text</h2>
    <button type="button" onclick="loadDoc()">Change Content</button>
  </div>
</body>
</html>
```

How to Get Started With AJAX | AJAX Tutorial For Beginners | Learn AJAX | PHP | JavaScript (4 min)

<https://www.youtube.com/watch?v=tNKD0kfel6o>

JSON and AJAX Tutorial: With Real Examples (40 min) (ok)

https://www.youtube.com/watch?v=rJesac0_Ftw

(apresenta a estrutura/formatação Json, os request em Ajax de documentos Json e escreve na página html a resposta)

-----Ficheiro HTML -----

```
<!DOCTYPE html>
<html lang="en">
<head>
</head>
<body>
  <header>
    <h1>JSON and AJAX</h1>
    <button id="btn">Fetch info for 3 new animals</button>
  </header>
  <div id="animal-info"></div>
  <script src="main2.js"> </script>
</body>
</html>
```

-----Ficheiro em Javascript-----

```
var animalcontainer=document.getElementById("animal-info");

var btn= document.getElementById("btn");

btn.addEventListener("click",function(){
  var ourRequest = new XMLHttpRequest();
  ourRequest.open('GET','t.json');
  ourRequest.onload= function(){
    var ourData= JSON.parse( ourRequest.responseText);
    renderHtml(ourData);
  }; //function
  ourRequest.send();
}); //function

function renderHtml(data){
  animalcontainer.insertAdjacentHTML('beforeend', data[0].name);
}
```

-----Ficheiro em JSON-----

```
[
{
  "name": "miau",
  "especie": "gato",
  "comida": {
    "gosta": ["atum", "catnip"],
    "naogosta": ["presunto", "zucc"]
  }
},
{
  "name": "Auau",
  "especie": "cao",
  "comida": {
    "gosta": ["ossos", "carne"],
    "naogosta": ["atum"]
  }
}
]
```

Exemplo/Trabalho de casa:

ESP8266 Web Server and Websockets Using Arduino IDE (Mac OSX and Windows) | vs. AJAX and HTTP
<https://www.youtube.com/watch?v=ROeT-gyYZfw>

8.2.6. Exemplo IV - Gráficos HighCharts

Para os exemplos em Javascript “HighCharts”

1- Instalar o XAMP, e no painel de controlo do XAMPP ativar o servidor WEB (Apache) e o servidor de base de dados (MySQL server)

2- Descarregar o zip [Highcharts 6.0.4](https://www.highcharts.com/download) (<https://www.highcharts.com/download>)

3- Instalar em c:\xampp\htdocs\highcharts\

4- testar um exemplo com um browser fazendo:

<http://localhost/highcharts/examples/line-basic/index.htm>



O nome da página é `index.htm` e contém o código Javascript

```
<!DOCTYPE HTML>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <title>Highcharts Example</title>

  <script type="text/javascript" src="https://ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery.min.js"></script>
  <style type="text/css">
    #container {
      min-width: 310px;
      max-width: 800px;
      height: 400px;
      margin: 0 auto
    }
  </style>
</head>
```

```

<body>

<script src="c:\xampp\htdocs\highcharts5/code/highcharts.js"></script>
<script src="c:\xampp\htdocs\highcharts5/code/modules/exporting.js"></script>
<div id="container"></div>

<script type="text/javascript">
Highcharts.chart('container', {

  title: { text: 'Solar Employment Growth by Sector, 2010-2016' },
  subtitle: {text: 'Source: thesolarfoundation.com'},
  yAxis: {title: { text: 'Number of Employees'}},
  legend: {layout: 'vertical', align: 'right', verticalAlign: 'middle'},
  plotOptions: { series: { pointStart: 201 } },
  series: [{

    name: 'Installation',
    data: [43934, 52503, 57177, 69658, 97031, 119931, 137133, 154175]
  }, {

    name: 'Manufacturing',
    data: <?PHP echo "[24916, 24064, 29742, 29851, 32490, 30282, 38121, 40434]"?>
  }, {

    name: 'Sales & Distribution',
    data: [11744, 17722, 16005, 19771, 20185, 24377, 32147, 39387]
  }, {

    name: 'Project Development',
    data: [null, null, 7988, 12169, 15112, 22452, 34400, 34227]
  }, {

    name: 'Other',
    data: [12908, 5948, 8105, 11248, 8989, 11816, 18274, 18111]
  }]
});
</script>

</body>
</html>

```

8.2.7. Exemplo V - Visualização num browser WEB a tabela “Histórico” (HighCharts)

Considere que uma base de dados “Reservatorio”, contém uma tabela “HistoricoReservatorio” com o valor das entradas digitais do PLC (X0,X1,X2,X3) , com as saídas (Y0,Y1,Y2) e com o nível de água (0 a 100%)
Pretende-se visualizar num browser WEB esses valores.

<http://localhost/Reservatorio.php>

Ler



8.2.8. Exemplo VI - Visualizar gráficos no Browser usando a biblioteca HighCharts, PHP e a base de dados MySQL

id	Y0	Y1	Y2	X0	X1	X2	X3	Nivel
1	0	0	0	0	0	0	0	0
2	1	1	1	1	0	0	0	20
3	1	0	1	1	1	0	0	30
4	1	1	1	1	1	0	0	40
5	1	1	1	1	1	0	0	50
6	1	1	1	1	1	0	0	60
7	1	1	1	1	1	0	0	70
8	1	1	1	1	1	1	0	80
9	0	1	1	1	1	1	1	90
10	0	1	1	1	1	1	0	80
11	0	1	1	1	1	0	0	70
12	0	1	1	1	1	0	0	60
13	0	1	1	1	1	0	0	50
14	0	1	1	1	1	0	0	40
15	0	1	1	1	1	0	0	30
16	1	1	0	1	0	0	0	20
17	1	1	0	0	0	0	0	0

Página HTML & PHP & JAVASCRIPT para visualizar a tabela “Histórico”

O nome da página é Reservatorio.php

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Highcharts Example</title>
<script type="text/javascript" src="https://ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery.min.js"></script>
<style type="text/css">
    #container {
        min-width: 310px;
        max-width: 800px;
        height: 400px;
        margin: 0 auto
    }
</style>
</head>

<body>

<?php

$HistNivel="";
$HistX0="";

/* Ligacao a base de dados “Reservatorio” */
$link = mysqli_connect("localhost", "root", "") or die("N,,o pude conectar: " . mysqli_error());
mysqli_select_db($link,"Reservatorio") or die("N,,o pude selecionar o banco de dados");

/* Ler tabela HistoricoReservatorio */
$query = "SELECT * FROM HistoricoReservatorio";
$result = mysqli_query($link,$query) or die("A query falhou: " . mysqli_error());
while ($row=mysqli_fetch_array($result,MYSQLI_ASSOC)){
    $HistNivel= $HistNivel.$row['Nivel'].",";
    $HistX0= $HistX0.$row['X0']. "0,";
}

/* Liberta o resultado */
mysqli_free_result($result);

/* Fechando a conex.,o */
mysqli_close($link);
?>

<form action="Reservatorio.php" method="POST">
<input type="submit" value="Ler"><br><br>
</form>

<script src="..\code\highcharts.js"></script>
<script src="..\code\modules\exporting.js"></script>

<div id="container"></div>
<script type="text/javascript">
Highcharts.chart('container', {
    title: {text: 'Reservatorio, 2017-2018'},
    subtitle: {text: 'Source: Informatica Industrial'},
    yAxis: { title: {text: 'HistoricoReservatorio'}},
    legend: {layout: 'vertical',align: 'right',verticalAlign: 'middle'},
    plotOptions: { series: {pointStart: 0}},
    series: [{
```

```
name: 'Nivel',
data: [ <?PHP echo $HistNivel;?>]
}, {
name: 'X0',
data: [ <?PHP echo $HistX0;?>]
}]
});
</script>

</body>
</html>
```

8.2.9. Exemplo VII - Gráficos HighCharts dinâmicos

Para compreender este exemplo é necessário saber:

- usar arrays em PHP e a função `json_encode($array3)`
- a notação JSON
- fazer pedidos AJAX em Javascript

No browser web deve fazer

http://localhost/highchart_test.php

No documento PHP “[highchart_test.php](#)” pedido pelo Browser, existe uma parte em PHP para ler a base de dados e atualizar a a string “\$conj” e uma parte para criar um gráfico “highchart” **chart1**

Para que o browser apresente o gráfico highchart use este script em “javascript” :

```
<script src="https://code.jquery.com/jquery-3.1.1.min.js"></script>
<script src="https://code.highcharts.com/highcharts.js"></script>
<script src="https://code.highcharts.com/modules/exporting.js"></script>
<script src="https://code.highcharts.com/highcharts-more.js"></script>

<script type="text/javascript">
var chart1
chart1=Highcharts.chart('container1',
{
    chart: { type: 'spline',
              events: { load: requestData() } // A função javascript requestData() será chamada pelo Browser, ..
            },

    title: { text: 'Fluxo Luminoso, 2018-2019' },

    subtitle: { text: 'Source: Sistemas Flexíveis de Produção' },

    xAxis: { type: 'datetime',
              labels: { rotation: -45, align: 'right',//format: '{value:%Y-%b-%e %H:%m}' }
            },

    legend: { layout: 'vertical',align: 'right',verticalAlign: 'middle' },

    plotOptions: {
      series: {pointStart: 0},
      marker: {enabled: true}
    },

    series: [
      { name: 'Fluxo1', data: [ <?PHP echo $conj; ?>]}, // A string $conj foi preenchida com PHP
      { name: 'Fluxo2', data: [ <?PHP echo $conj; ?>]}
    ]
};
```

```
function requestData() {
$.ajax({
  url: "high00.php?callback=?", // retorna uma só linha, ex: ?([[[1535476295000,96]],[[1535476295000,96]]])
  type: "GET",
  dataType: "json",
  success: function (point) { // o objecto "point" tem os valores retornados pelo echo do ficheiro high00.php
    // Se o texto retornado (echo) pelo fich php for ?([[[1535476295000,96]],[[1535476295000,96]]])
    // point[0][0] contém [1535027017000, 55 ]
    // point[0][0][0] contém 1535027017000 e point[0][0][1] contém 55
    chart1.series[0].addPoint(point[0][0], true, false); // line
    chart1.series[1].addPoint(point[1][0], true, false);

    // call it again after two second
    setTimeout(requestData, 2000);
  } //function(point

}); // ajax({
} // function request()

</script>
```

O ficheiro seguinte, High00.php , retorna dados no formato JSON

```
<?php
header("content-type: application/json");

$the_date=0;
$conj="[";

/* Conectando, escolhendo o banco de dados */
$link = mysqli_connect("localhost", "user", "pass") or die("Nao pude conectar: " . mysqli_error());
mysqli_select_db($link,"LUX") or die("Nao pude seleccionar o banco de dados");

/* Fazendo a query SQL DE LEITURA DA BASE DE DADOS*/
// o registo/linha mais recente, com o id mais alto
$query = "SELECT * FROM HistoricoLuminosidadeSetor1 ORDER BY id DESC LIMIT 0,1";
$result = mysqli_query($link,$query) or die("A query falhou: " . mysqli_error());
while ($row=mysqli_fetch_array($result,MYSQLI_ASSOC)){
  $the_date = 1000 * strtotime($row['DataHora']);
  $conj=$conj.$the_date.", ".$row['Fluxo'];
  $ii=intval($row['Fluxo']);
  $array1[]=array($the_date,$ii);
  $array2[]=array($the_date,$ii);
}
$conj=$conj."]";

/* Liberta o resultado */
mysqli_free_result($result);

/* Fechando a conexão */
mysqli_close($link);

$array3=array($array1,$array2);

// O echo seguinte retorna ?([[[ 1535472342000, 240 ]],[ 1535472342000 , 240 ]]])
// contém dois pontos. Cada ponto tem o timeStamp da base de dados e o fluxo luminoso
echo $_GET['callback']. '('. json_encode($array3) . ')';
?>
```

Trab nº 6 – Crie páginas em javascript/highchart que permita ao Browser aceder a base de dados e criar gráficos

Veja o exemplo 8.2.8. Exemplo VI - Gráficos HighCharts