CTeSP PSI 2020/21

Introdução às Linguagens e Tecnologias Web

AJAX

Pedro Colarejo
pcolarejo@ua.pt
Escola Superior de Tecnologia e Gestão de Águeda
Universidade de Aveiro
2020/21

AJAX

Asynchronous *JavaScript and XML*

http://www.w3schools.com/ajax/

AJAX

AJAX stands for “Asynchronous JavaScript and XML“, and is a way that a webpage can use JavaScript to send and receive data from a server without refreshing a webpage. XML is a kind of markup language – like HTML, which people sometimes use for sending data across the internet

AJAX: *Asynchronous JavaScript and XML.* A system for sending and receiving data from a server without a page refresh. (example below)

XML: *eXtensible Markup Language*. A language for organizing arbitrary data. Uses lots of angle brackets “<>”.

HTML: *HyperText Markup Language*. A subset of XML specifically for describing and organizing web pages.

JSON: *JavaScript Object Notation*. A more modern way of packaging data that’s often used with AJAX. Can be natively read by JavaScript.

In “html_xhtml_and_css_all-in-one_for_dummies”, Andy Harris, Whiley Publishing, Inc

AJAX

Direct control of client-server communication: Rather than the automatic communication between client and server that happens with Web forms and server-side programs, AJAX is about managing this relationship more directly:

Use of the XMLHttpRequest object: This is a special object that's been built into the DOM of all major browsers for some time, but it wasn't used heavily. The real innovation of AJAX was finding creative (and perhaps unintentional) uses for this heretofore virtually unknown utility.

In "html_xhtml_and_css_all-in-one_for_dummies", Andy Harris, Whiley Publishing, Inc

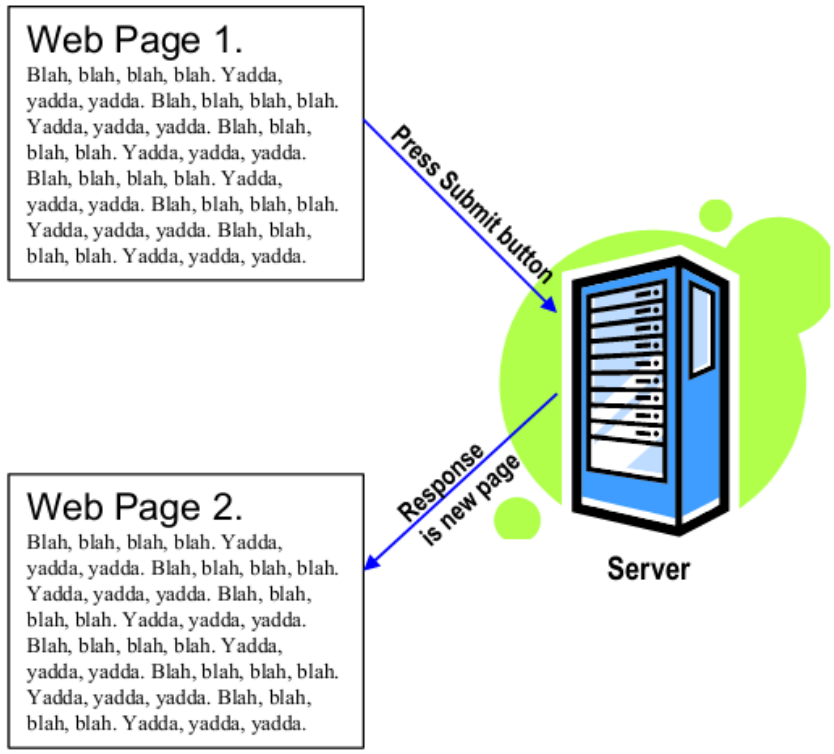
AJAX

A closer relationship between client-side and server-side programming:
Up to now, client-side programs (usually JavaScript) did their own thing, and server-side programs (PHP) operated without too much knowledge of each other. AJAX helps these two types of programming work together better.

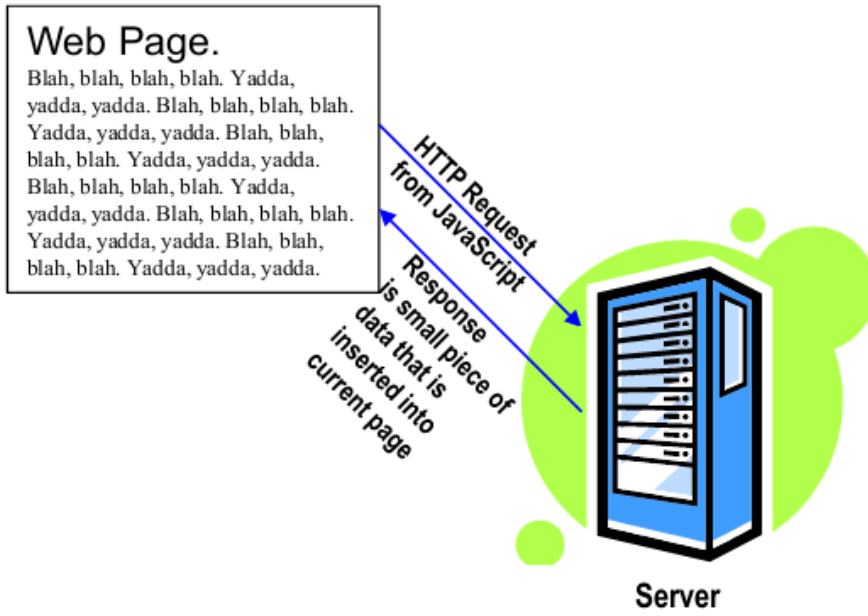
A series of libraries that facilitate this communication: AJAX isn't that hard, but it does have a lot of details. Several great libraries have sprung up to simplify using AJAX technologies. You can find AJAX libraries for both client-side languages, like JavaScript, and server-side languages, like PHP.

In "html_xhtml_and_css_all-in-one_for_dummies", Andy Harris, Whiley Publishing, Inc

Web Apps vs. AJAX Apps



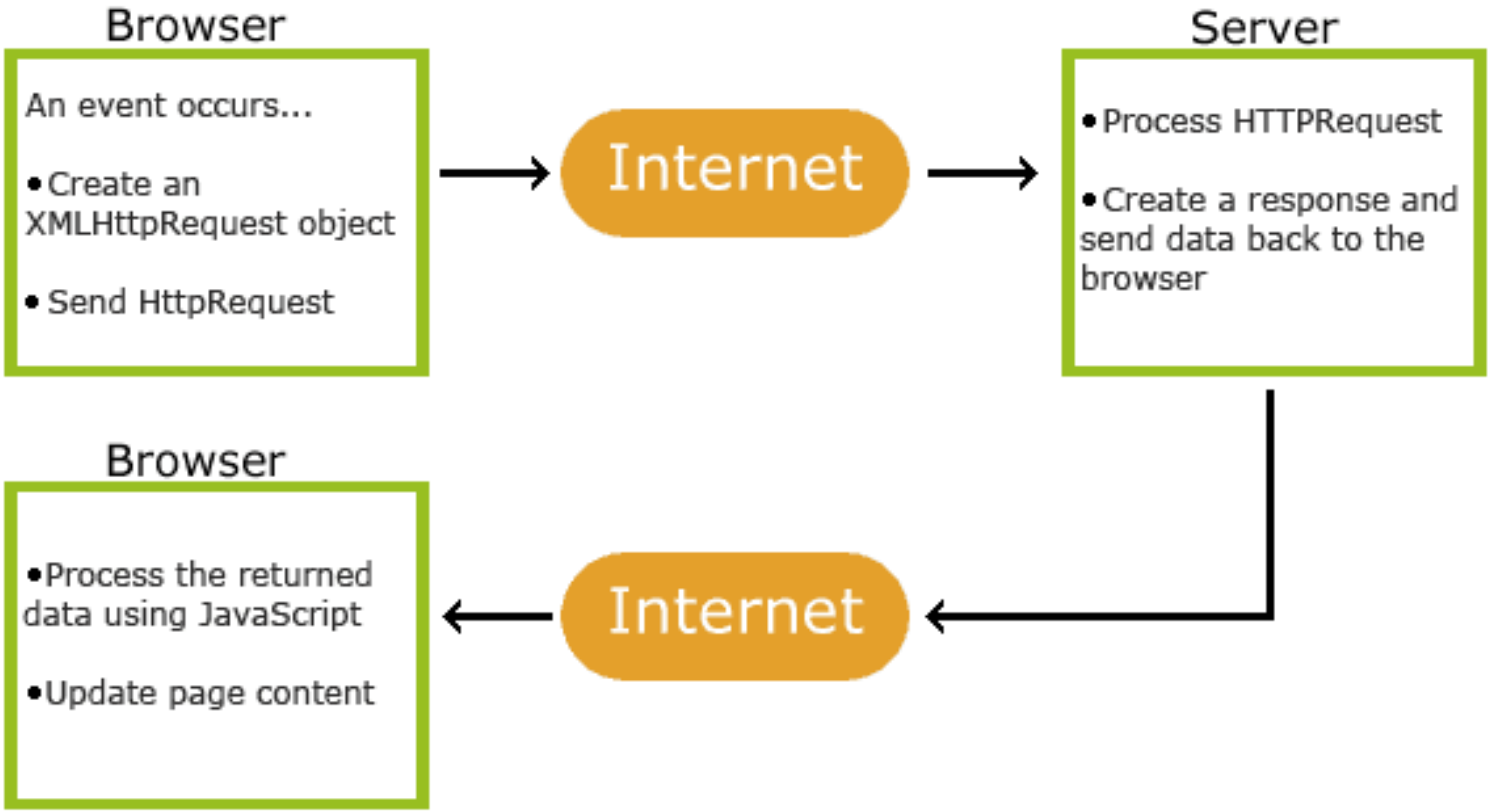
Web Apps tradicionais
atualizações alargadas e pouco frequentes



AJAX Apps
pequenas e frequentes atualizações

(adaptado de http://courses.coreservlets.com)

Web Apps vs. AJAX Apps



(adaptado de https://www.w3schools.com/xml/ajax_intro.asp)

Processo de funcionamento

Javascript

- Definir objeto para envio do HTTP Request
- Iniciar o HTTP Request
  - Obter o objeto para o HTTP Request
  - Definir uma função para tratamento do HTTP Response
  - Iniciar um HTTP Request do tipo GET ou POST
  - Enviar os dados para o servidor
- Processar a resposta do servidor
  - Esperar pelos seguintes estados:

    readState==4 e HTTP Status==200
  - Extrair o texto retornado com responseText ou responseXML
  - Efetuar o processamento da informação recebida

HTML

- Carregar o código Javascript
- Definir o controlo (evento) que iniciará o HTTP Request
- Definir o container HTML que receberá o resultado

Estados da resposta

readyState

- **0**: request not initialized
- **1**: server connection established
- **2**: request received
- **3**: processing request
- **4**: request finished and response is ready
  - status 200: "OK"
  - Status 404: Page not found

Status

- **500 - 599:** the server had an error
- **400 - 499:** this is a client error (Ex: 404 page not found)
- **300 - 399:** then exists a redirect
- **200 - 299:** then it is correct and
- **100 - 199:** means information message

Definir o objeto para envio do HTTP Request

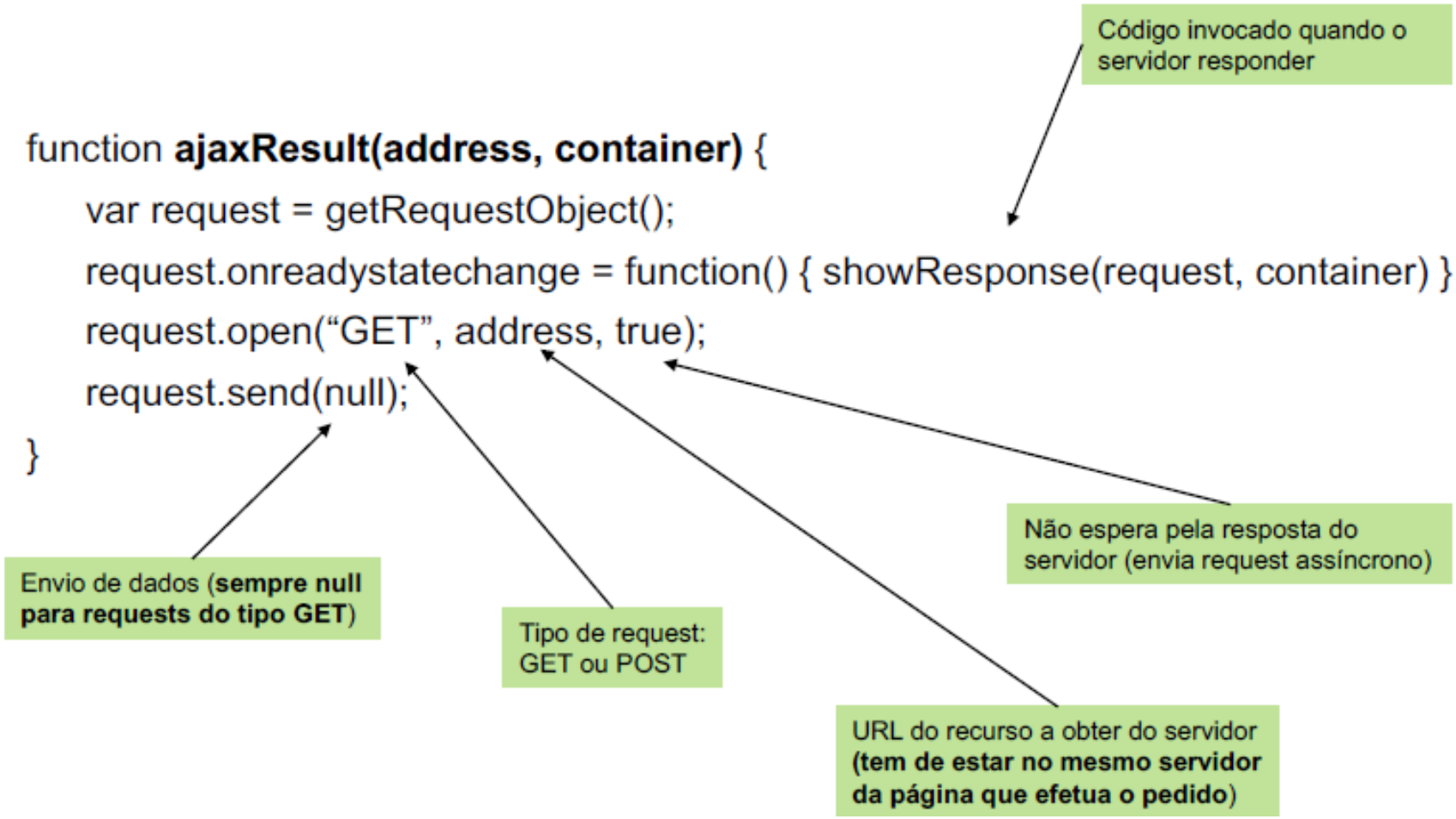
```
function getRequestObject() {
    if (window.XMLHttpRequest) {
        return(new XMLHttpRequest());
    } else if (window.ActiveXObject) {
        return(new ActiveXObject("Microsoft.XMLHTTP"));
    } else  {
        return(null);
    }
}
```

Versão para:
Firefox, Netscape 5+, Opera, Safari, Mozilla, Chrome e IE a partir do 7

Versão para:
IE 5.5 e 6

Falha a criação do objeto para antigos browsers não standard

Iniciar o HTTP Request



Processar a resposta do servidor

```
function showResponse(request, container) {
    if  ( (request.readyState == 4) && (request.status == 200) ) {
        document.getElementById(container).innerHTML= request.responseText;
    }
}
```

O Valor 4 significa que a resposta do servidor está concluída. O valor 200 significa que a resposta do servidor foi obtida sem erros

Código para processar a informação enviada pelo servidor. Neste exemplo coloca o texto num dado container HTML

Container HTML onde colocar o resultado enviado pelo servidor

Texto proveniente da resposta do servidor (HTTP Response)

Código HTML

- Utilizar HTML5
- Carregar o ficheiro de Javascript
- Definir o controlo (evento) que iniciará o HTTP Request
- Definir o container HTML que receberá o resultado

Exemplo (show-message.js)

```
function getRequestObject() {
        if (window.XMLHttpRequest) {
                return(new XMLHttpRequest());
        } else {
                return(null);
        }
}
function ajaxResult(address, container) {
        var request = getRequestObject();
        request.onreadystatechange = function() {
                                        showResponse(request, container) };
        request.open("GET", address, true);
        request.send(null);
}
function showResponse(request, container) {
        if ( (request.readyState == 4) && (request.status == 200) ) {
                document.getElementById(container).innerHTML=
                        request.responseText;
        }
}
```

Exemplo (index.html)

```
<!DOCTYPE html>
<html>
<head>
        <title>Exemplo AJAX</title>
        <script type="text/javascript" src="show-message.js"></script>
</head>
<body>
        <h1>Exemplo AJAX</h1>
        <div>
                <button value="Show Message"
                        onclick="ajaxResult('message.html','msg')">
                        Show Message
                </button>
                <br /><br />
        </div>
        <div id="msg"></div>
</body>
</html>
```

message.html => “Texto que se pretende apresentar”

Exercício

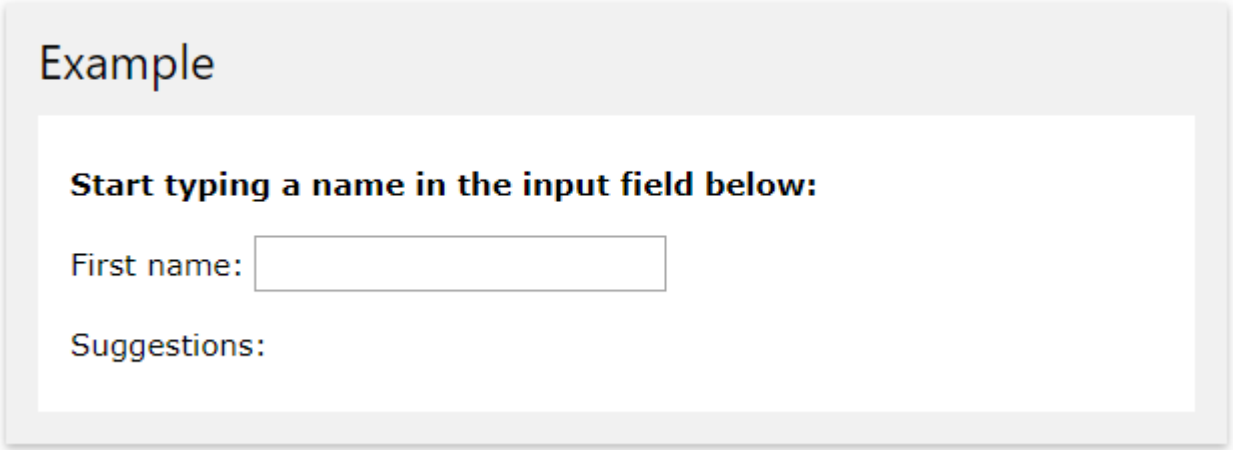
1. Altere o exemplo anterior de forma a que:

- Ao clicar sobre o botão “Show Message”, apareça o conteúdo da página “message.html” e o texto do botão passe para “Hide Message”.
- Ao clicar sobre o botão “Hide Message”, desapareça o conteúdo da página “message.html” e o texto do botão passe para “Show Message”.

Segurança:
Both the web page and the XML file it tries to load, must be located on the same server.
Same Origin Policy (http://en.wikipedia.org/wiki/Same_origin_policy)”

Acesso a uma API

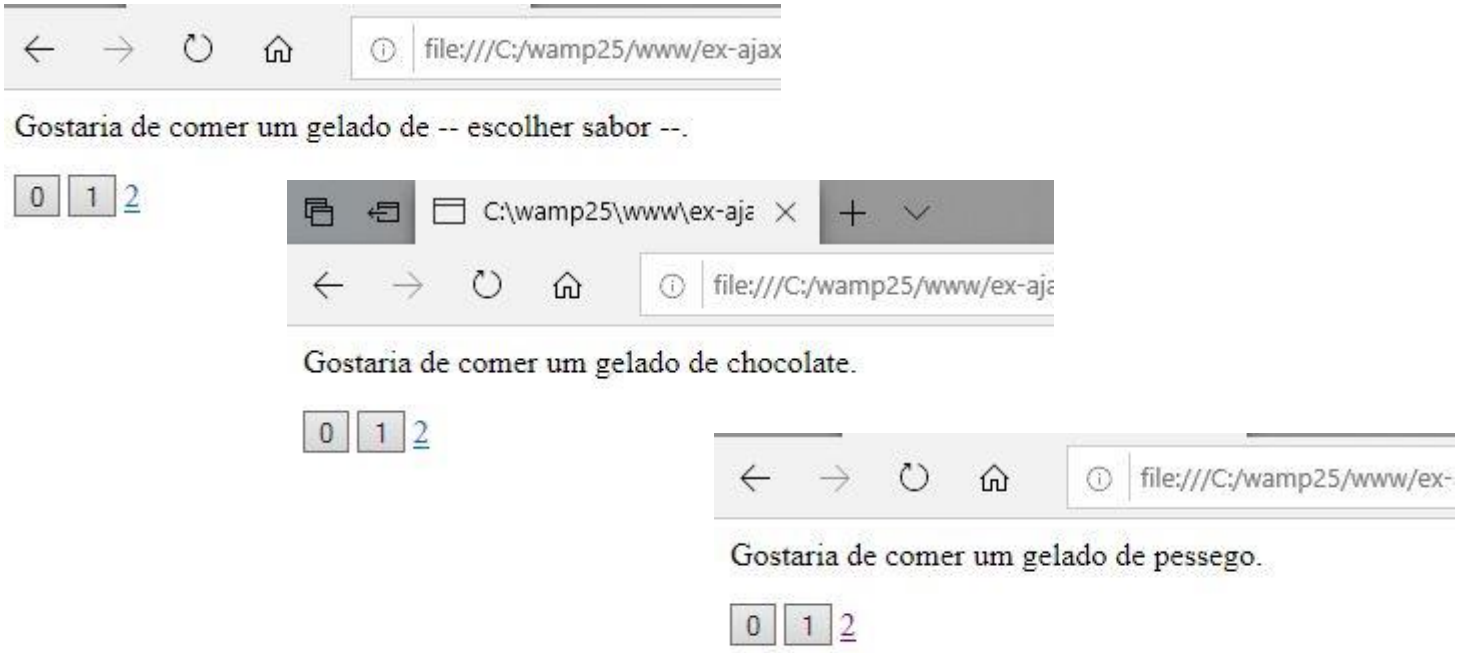
Aceder à API disponibilizada em
http://estga-dev.clients.ua.pt/~asw/gethint/gethint.php?q=
em que o parâmetro "q" define o valor retornado pelo teclado



Exercício

Acesso a uma API

Produza o código necessário, usando AJAX, para aceder à API disponibilizada em
http://estga-dev.clients.ua.pt/~asw/flavour/ajaxinput.php?q=
em que o parâmetro "q" define o valor retornado pelos botões 0,1,2

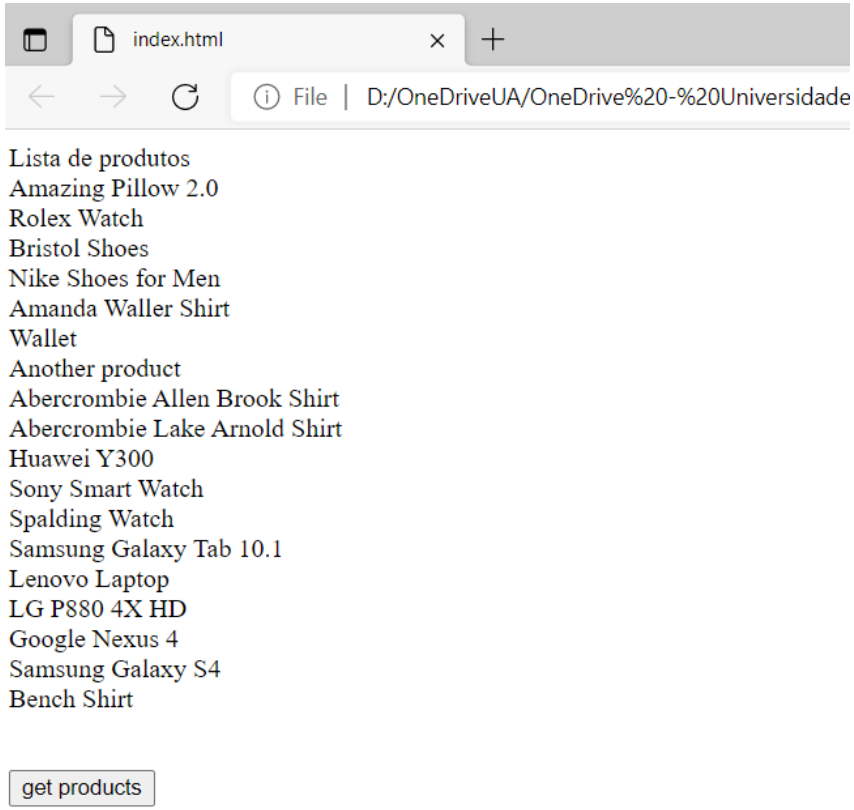


Exercício

Acesso a uma API

Produza o código necessário, usando AJAX, para aceder à API disponibilizada em http://estga-dev.clients.ua.pt/~asw/api/product/read.php

E apresentar os resultados de uma forma similar a



Exercício

Acesso a uma API

Produza o código necessário, usando AJAX, para aceder à API disponibilizada em https://epistat.wiv-isp.be/covid/

Esta API devolve alguns json com informação sobre a evolução do COVID na Bélgica.

Utilize a função de javascript JSON.parse para converter o json num array e poder apresentar os resultados à sua escolha.

Sugestão: apresente uma lista com os casos de COVID com a informação de data, província e número de casos