

# Tecnologias Web

---

7ª AULA, 1 de **abril**

**WWW**

Roberto Lam, Instituto Superior de Engenharia,  
Universidade do Algarve

[rlam@ualg.pt](mailto:rlam@ualg.pt)

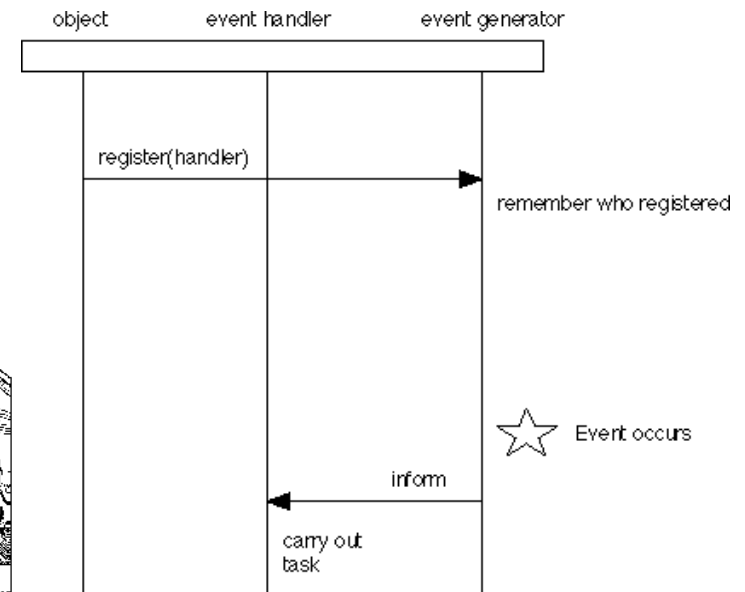
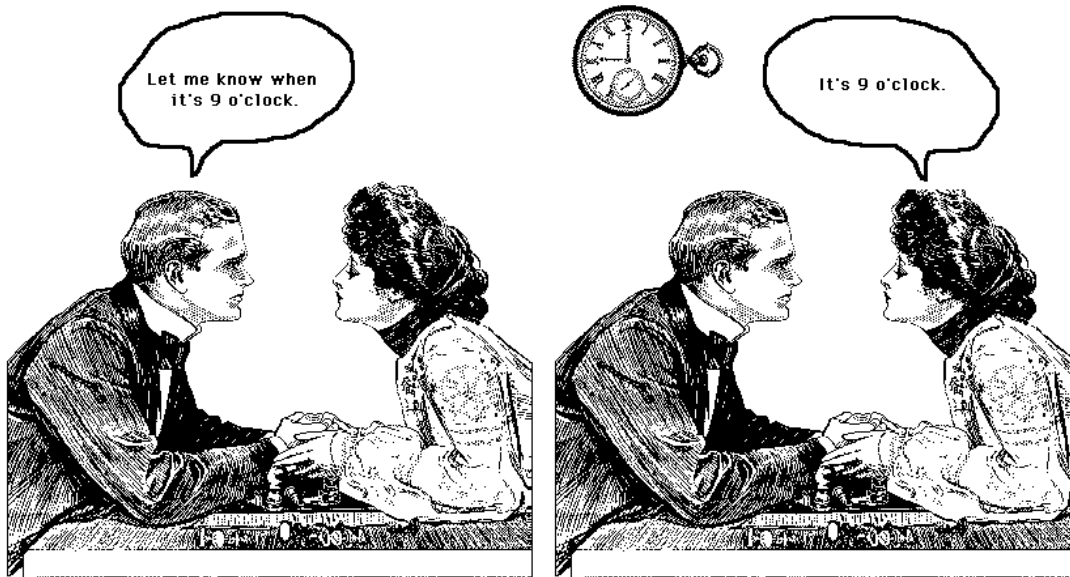
2023/24



# JavaScript



## Eventos, programação por eventos



<http://csis.pace.edu/~bergin/Java/eventfundamentals.html>

# JavaScript



## Eventos

**Click do rato**

**Carga da página**

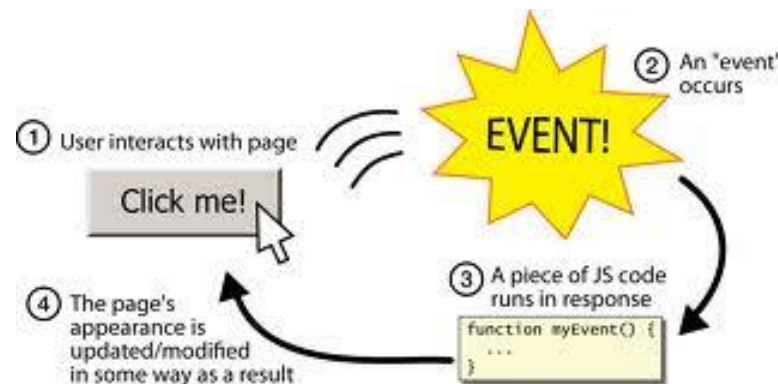
**Carga de uma imagem**

**Mov. Rato sobre elemento**

**Campo de entrada é alterado**

**Submissão de um formulário**

**Pressão sobre uma tecla**



**JQuery** **Prototype**  
**YUI** **Yahoo** **Dojo** **Mochikit**

consultar: [http://www.w3schools.com/jsref/obj\\_window.asp](http://www.w3schools.com/jsref/obj_window.asp)

# JavaScript

---



## *Atributos evento*

```
<a href="http://www.ualg.pt" onclick="alert(this.href);  
    return false; ">Click Here</a>
```

```
var link = document.getElementsByTagName("a")[0];
```

```
link.onclick = function clickHandler() {  
    alert(this.href);  
    return false;  
};
```

A blue speech bubble with a black outline. It has a long tail pointing towards the 'return false;' line in the code block above it. Inside the bubble, the word 'False!' is written in white, bold, sans-serif font.

False!



## *Contexto do atributo evento*

### *Ponteiro **this***

```
function whatDidIClickOn() {  
    console.log("You clicked on ", this);  
    return false;  
}  
  
var links = document.getElementsByTagName("a");  
  
for (var i = 0, link; link = links[i];i++) {  
    link.onclick = whatDidIClickOn;  
}
```

```
Console (ferramentas do programador)  
You clicked on <a href="http://www.ualg.pt">
```



## Métodos dos eventos

`addEventListener(eventType, handler, capture)`

Alguns *eventType*: **submit, mouseover, keypressed, click, load**

```
<p id="para">Click here</p>
```

```
var para = document.getElementById("para");  
para.addEventListener("click", eventHandler, false);  
para.addEventListener("click", otherEventHandler, false);
```

```
function eventHandler(event) {  
    alert("P clicked " + this.nodeName);  
}
```

```
function otherEventHandler(event) {  
    alert("Evento " + event.type + ".");  
}
```

“P clicked” e “Evento click.”



## *Objecto evento*

### *Propriedades (atributos)*

- **type** Tipo de evento
- **target** Elemento que emitiu o evento. Pode não ser o mesmo elemento ao que o manipulador de eventos está associado.
- **currentTarget** Elemento ao qual está associado ao evento.
- **eventPhase** Numero que indica se o evento está na fase de captura (1), bubbling (3), ou se atingiu o elemento (2).
- **timeStamp** O numero de segundos em que o evento ocorreu.

### *Metodos*

**preventDefault()** Não permite a execução definida por defeito.  
Por exemplo: links e submissão de forms.

# JavaScript

---



```
<a id="link" href="http://www.google.com">Don't go to  
Google</a>
```

```
var link = document.getElementById("link");  
link.addEventListener("click", dontFollowLink, false);
```

```
function dontFollowLink(event) {  
  alert("Not going to " + this.href);  
  event.preventDefault();  
}
```

**stopPropagation()** Para, suspende a propagação do evento pela estrutura DOM. Mecanismo de captura e *bubbling*



# JavaScript

---



removeEventListener(eventType, handler, capture)

```
<h1 id="header">Click this multiple times</h1>
```

```
var header = document.getElementById("header");
```

```
header.addEventListener("click", selfRemovingEvent, false);
```

```
function selfRemovingEvent(event) {  
    alert("Listening for the event and removing myself");  
    this.removeEventListener(event.type, selfRemovingEvent,  
false);  
}
```

consultar: [http://docstore.mik.ua/oreilly/webprog/jscript/ch12\\_01.htm](http://docstore.mik.ua/oreilly/webprog/jscript/ch12_01.htm)  
[http://docstore.mik.ua/oreilly/webprog/jscript/ch19\\_01.htm](http://docstore.mik.ua/oreilly/webprog/jscript/ch19_01.htm)



## *Captura e Bubbling do objecto evento*

### *Bubbling*

```
<html>
<head>  <title> exemplo </title>
</head>
<body>
  <div>
    <p>
      <a href="http://www.google.com">Click</a>
    </p>
  </div>
  <script type="text/javascript">
    // codigo javascript

  </script>
</body>
```

# JavaScript

---



## Código javascript

```
var div = document.getElementsByTagName("div")[0];
var p = document.getElementsByTagName("p")[0];
var a = document.getElementsByTagName("a")[0];

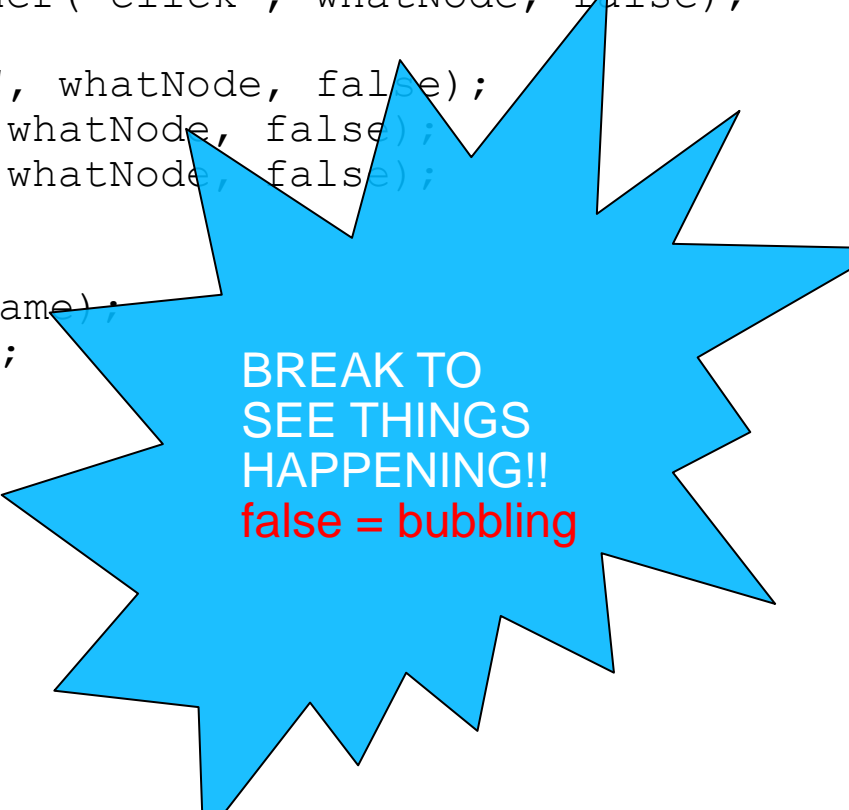
document.addEventListener("click", whatNode, false);
document.body.addEventListener("click", whatNode, false);

div.addEventListener("click", whatNode, false);
p.addEventListener("click", whatNode, false);
a.addEventListener("click", whatNode, false);

function whatNode(event) {
    console.log(this.nodeName);
    event.preventDefault();
}
```

## CSS (??)

```
body, div, p {
    padding: 10px;
}
```



BREAK TO  
SEE THINGS  
HAPPENING!!  
**false = bubbling**

# JavaScript



## Captura

```
document.addEventListener("click", whatNode, true);
document.body.addEventListener("click", whatNode, true);

div.addEventListener("click", whatNode, true);
p.addEventListener("click", whatNode, true);
a.addEventListener("click", whatNode, true);

function whatNode(event) {
  console.log(this.nodeName, "phase: " + event.eventPhase);
  event.preventDefault();
}
```

Output:

#document phase: 1
BODY phase: 1
DIV phase: 1
P phase: 1
A phase: 2
A phase: 2
P phase: 3
DIV phase: 3
BODY phase: 3
#document phase: 3

BREAK TO  
SEE THINGS  
HAPPENING!!  
**true = captura**

# JavaScript

---



## *Suspensão propagação evento (na captura)*

```
function whatNode(event) {  
    if (this.nodeName === "DIV") {  
        event.stopPropagation();  
    }  
    console.log(this.nodeName, "phase: " + event.eventPhase);  
    event.preventDefault();  
}
```

## Eventos associados ao BOM

**onload** window.onload, também para <img>

**unload** window.unload

**beforunload** window.beforunload

**resize** window.resize



## Eventos associados ao rato

**clientX, clientY** posição do rato no *viewport* da janela. Origem topo esquerdo da janela.

**pageX, pageY** posição do rato na página. Com possível deslocamento da página. Origem topo esquerdo da janela.

**screenX, screenY** posição da janela browser. Origem topo esquerdo da janela do desktop

**offsetX, offsetY** posição do rato no elemento de accionou o evento. Origem topo esquerdo da área do elemento.

**altkey, shiftkey, ctrlkey** verifica se alguma das teclas está pressionada

**button, detail, relatedTarget, mousedown, mouseup, dblclick, mouseover, mouseout**

**click**, accionado depois de um mousedown e mouseup. Também poder ser acc. com tecla *enter* (usando tab para chegar ao elemento)



## Eventos associados ao teclado

**keydown** qualquer tecla faz disparar este evento.

**keypress** só teclas passíveis de impressão. Alt, shit, ctrl e enter não accionam este evento.

**keyup** evento accionado quando o utilizador liberta a tecla e depois do keypress e keypress.

## Eventos associados ao <form>

**change** o evento é accionado quando o utilizador altera o conteúdo de uma das entradas do <form>

**submit** o evento só é accionado ao submeter-se o conteúdo do <form>. É muito utilizado para validação de entradas

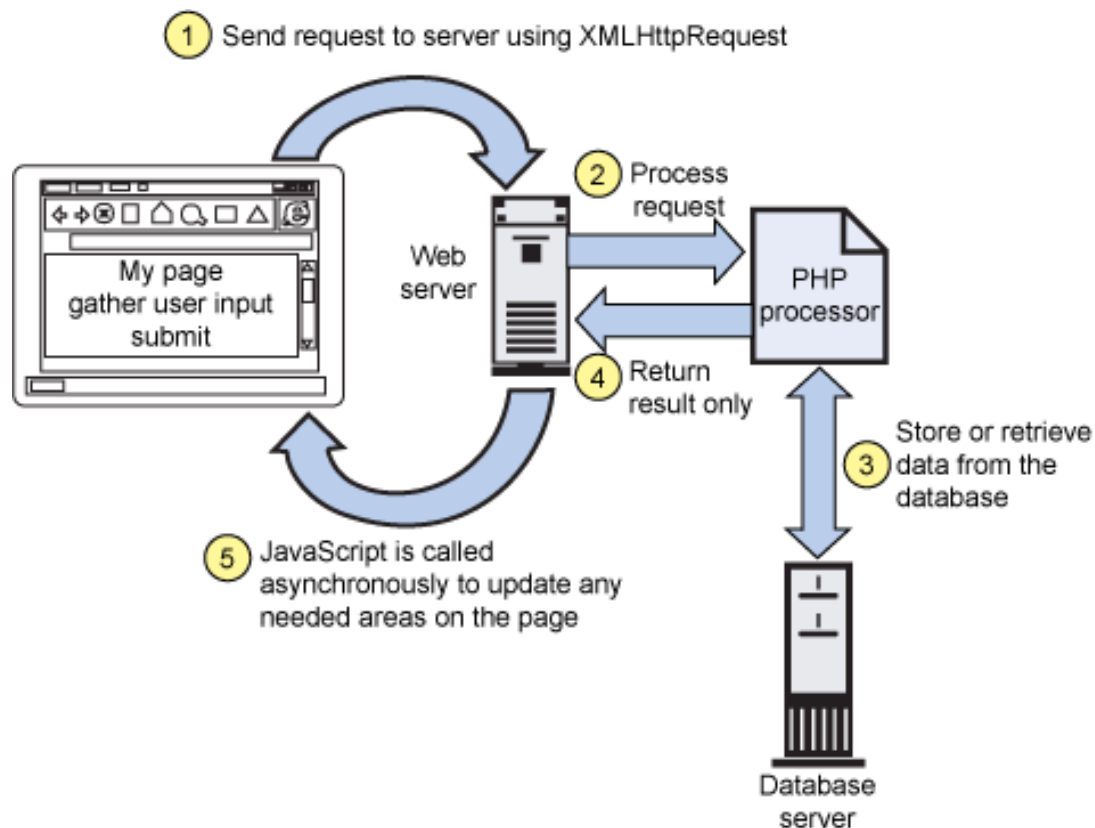
Outros eventos: **focus** e **blur** (associados ao rato e teclado, nas entradas do elemento <form>)

# JavaScript com AJAX



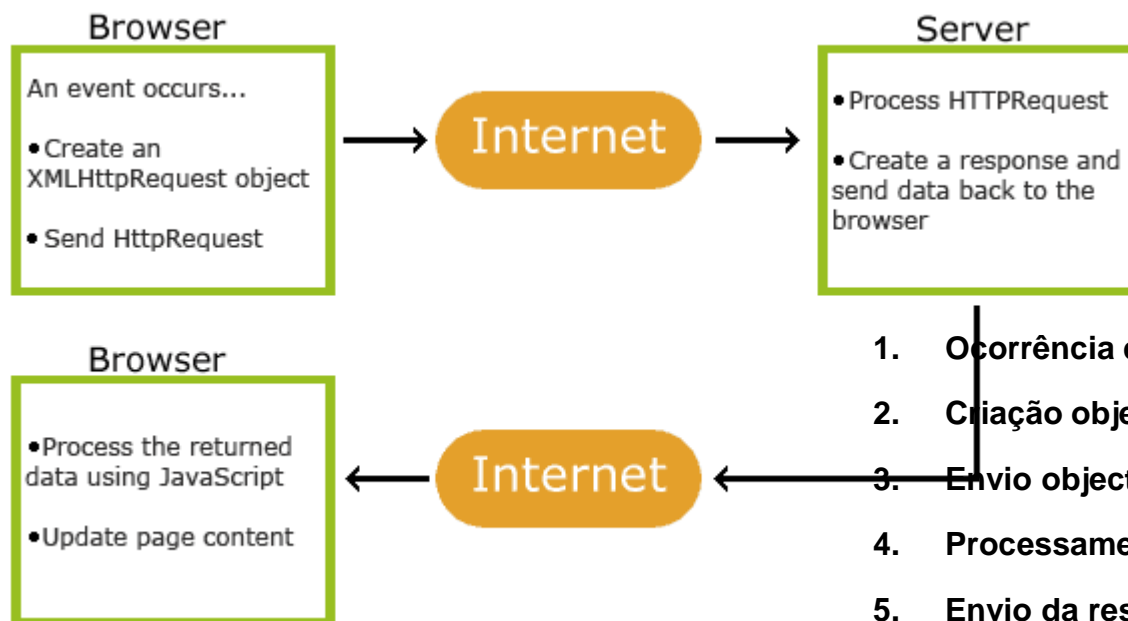
AJAX?

Asynchronous Javascript and XML





# JavaScript com AJAX



1. Ocorrência de evento
2. Criação objecto XMLHttpRequest
3. Envio object XMLHttpRequest
4. Processamento da solicitação HTTP
5. Envio da resposta ao cliente
6. Recepção e actualização da página por javascript

**Servidor  
diferente**



-WebSockets-



<http://www.youtube.com/watch?v=FQ5nj1u2e4c>

<http://www.html5rocks.com/pt/tutorials/websockets/basics/>

# JavaScript com AJAX

---



## Criação, preenchimento do objecto AJAX

```
var request = new XMLHttpRequest();  
request.open("GET", "escalões.txt", false);
```

## Envio do objecto AJAX

```
request.send(null);
```

true para  
solicitação  
síncrona

## Receção processamento objecto AJAX

```
if (request.status == 200){  
    //we got a response  
    alert(request.responseText);  
} else {  
    //algum erro  
    alert(«Error- « + request.status + «: « + request.statusText);  
} // end if
```

# JavaScript com AJAX



**Table 1-1 Useful Members of the XMLHttpRequest Object**

<i><b>Member</b></i>	<i><b>Description</b></i>	<i><b>Basket analogy</b></i>
<code>open(protocol, URL, synchronization)</code>	Opens up a connection to the indicated file on the server.	Stands under a particular window.
<code>send(parameters)</code>	Initiates the transaction with given parameters (or null).	Releases the basket but hangs on to the string.
<code>status</code>	Returns the HTTP status code returned by the server (200 is success).	Checks for error codes ("window closed," "balloon popped," "string broken," or "everything's great").
<code>statusText</code>	Text form of HTTP status.	Text form of status code, a text translation of the numeric error code returned by status.
<code>responseText</code>	Text of the transaction's response.	Gets the contents of the basket.
<code>readyState</code>	Describes current status of the transaction (4 is complete).	Is the basket empty, going up, coming down, or here and ready to get contents?
<code>onReadyStateChange</code>	Event handler. Attach a function to this parameter, and when the <code>readyState</code> changes, the function will be called automatically.	What should I do when the state of the basket changes? For example, should I do something when I get the basket back?

# JavaScript com AJAX

---



```
<html>  
<head>  
<script>
```

```
    // código javascript + AJAX a ser colocado
```

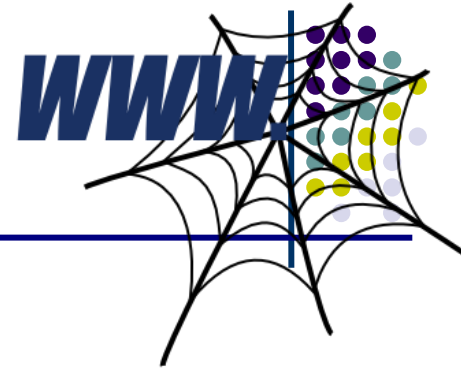
```
</script>  
</head>  
<body>
```

```
<div id="myDiv"><h2>Escalões: </h2></div>  
<button type="button" onclick="loadStrDoc()">Change Content</button>
```

```
</body>  
</html>
```

# JavaScript com AJAX

---



## Código JavaScript com AJAX

```
function loadStrDoc()
{
    var xmlhttp;
    if (window.XMLHttpRequest)
    {
        // code for IE7+, Firefox, Chrome, Opera, Safari
        xmlhttp=new XMLHttpRequest();
    }

    xmlhttp.onreadystatechange=function()
    {
        if (xmlhttp.readyState==4 && xmlhttp.status==200)
        {
            document.getElementById("myDiv").innerHTML=
                xmlhttp.responseText;
        }
    }
    xmlhttp.open("GET","escaloes.txt",true);
    xmlhttp.send(null);
}
```

Pode ser  
função  
explícita!!

# JavaScript com AJAX

---



Código JavaScript com AJAX (solução do ex. 3 da aula prática 6)

```
<html>
  <head>
    <title>Exemplo 1 AJAX simples </title>
    <script type="text/javascript">

      //JavaScript AJAX code here
      //Prepare the global variable for the request

      var myRequest;
      var n;

      function getText(url){

        //check support for the XMLHttpRequest object
        if (window.XMLHttpRequest)
        {
          myRequest = new XMLHttpRequest();
        }
      }
    </script>
  </head>
</html>
```

# JavaScript com AJAX

---



```
//metodo open() com protocolo
myRequest.open("GET", url, true);

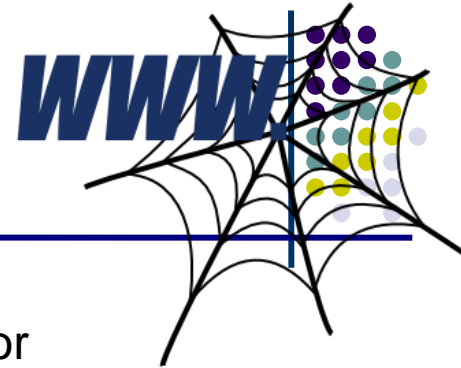
// envio da solicitação
myRequest.send(null);

//Atribuir a função getData() à propriedade
//onreadystatechange para processar a resposta do servidor
myRequest.onreadystatechange = getData;

} // fim da funcao getText
```

# JavaScript com AJAX

---



//Função que vai processar a resposta do servidor

```
function getData()  
{
```

```
//Get a reference to the header element where  
//the returned result will be displayed
```

```
var myHeader = document.getElementById("myHeader");  
var mySelection = document.getElementById("selection");
```

```
//Check the response is complete  
if (myRequest.readyState ===4)
```

```
{  
  //Check the status code of the response is successful  
  if (myRequest.status === 200){
```

```
    //Store the response  
    var text = myRequest.responseText;
```



# JavaScript com AJAX

---



```
var stringAux=text.split(";");
for(var i=0; i<stringAux.length; i++){
    //alert(stringAux[i]);
    var option=document.createElement("option");
    option.text=stringAux[i];
    try
    {
        // for IE earlier than version 8
        mySelection.add(option,x.options[null]);
    }
    catch (e)
    {
        mySelection.add(option,null);
    }
}
mySelection.remove(0);
}
}
}
}
</script>
</head>
```

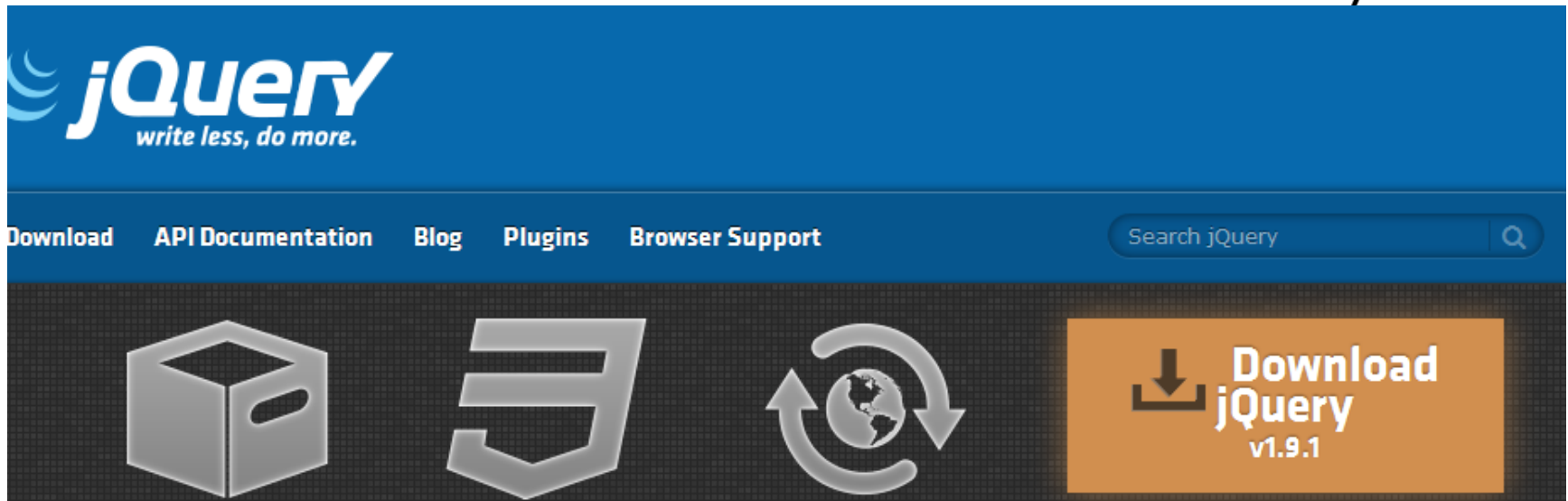
# JavaScript com AJAX

---



```
<body onload="getText('/~rlam/escaloes.txt')">  
  <h1>Obtenção das opções (<select>) via AJAX </h1>  
  <div>  
    <h2 id="myHeader">Este texto pode ser alterado </h2>  
    <select id="selection">  
      <option></option>  
    </select>  
    <input type="button" value="Click Me!" />  
  </div>  
</body>  
</html>
```

# JQuery



<http://jquery.com/download/>

Ver capitolo 10

JavaScript and AJAX for Dummies, Andy Harris, Wiley Publishing, Inc.



## Inclusão da JQuery (no servidor)

```
<script type = "text/javascript" src = "jquery-1.3.2.min.js"></script>
```

## Inclusão da JQuery (nos servidores google)

```
<<script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js">  
</script>
```

```
<script type = "text/javascript">  
//your code here
```

```
</script>
```

# JQuery – Exemplo 1



```
<head>
<meta http-equiv="content-type" content="text/xml;
charset=utf-8" />
<title>change.html</title>
<script type = "text/javascript" src = "jquery-1.3.2.min.js"></script>
<script type = "text/javascript">
```

```
function changeMe(){
    $("#output").html("I changed");
}
```

?????

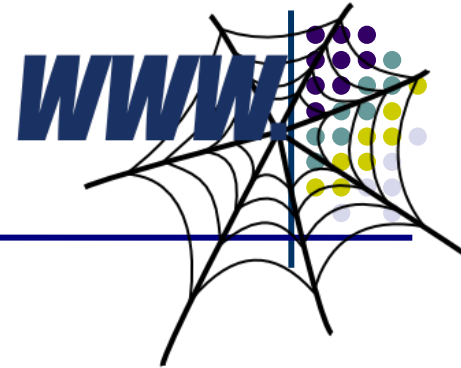
```
</script>
</head>
<body onload = "changeMe()">
<h1>Basic jQuery demo</h1>
<div id = "output">
Did this change?
</div>
</body>
</html>
```

```
var jqOutput = $("#output");
var DOMoutput =
document.getElementById("output");
```

```
jqOutput.html("I've changed"); //jQuery
DOMoutput.innerHTML = "I've changed"; //Javascript
```

# JQuery – Exemplo 2

---



```
<html>
<head>
  <meta http-equiv="content-type" content="text/xml; charset=utf-8" />
  <script type = "text/javascript" src = "jquery-1.3.2.min.js"></script>
  <script type = "text/javascript">
    $(init);

    function init(){
      $("li").hover(border, noBorder);
    } // end init

    function border(){
      $(this).css("border", "1px solid black");
    }

    function noBorder(){
      $(this).css("border", "0px none black");
    }
  </script>
</head>
<body>
  <ul>
    <li>Item 1</li>
    <li>Item 2</li>
    <li>Item 3</li>
    <li>Item 4</li>
    <li>Item 5</li>
  </ul>
</body>
</html>
```

# JQuery – Exemplo 2

---



```
</script>
<title>hover.html</title>
</head>
<body>
  <h1>Hover Demo</h1>
  <ul>
    <li>alpha</li>
    <li>beta</li>
    <li>gamma</li>
    <li>delta</li>
  </ul>
</body>
</html>
```



---

***Perguntas?***

***Quem viu JavaScript JSON?***  
***Goodbye JavaScript!***