



# Programação Web

## html/css/dom/ajax

J. Arnaldo Martins  
([jam@ua.pt](mailto:jam@ua.pt))



# História dos computadores

<http://www.computerhistory.org/>

**Até fim da década de 70**



**várias pessoas → 1 computador**



IBM S/360 (1964)

**Década de 80**

**1 pessoa → 1 computador**



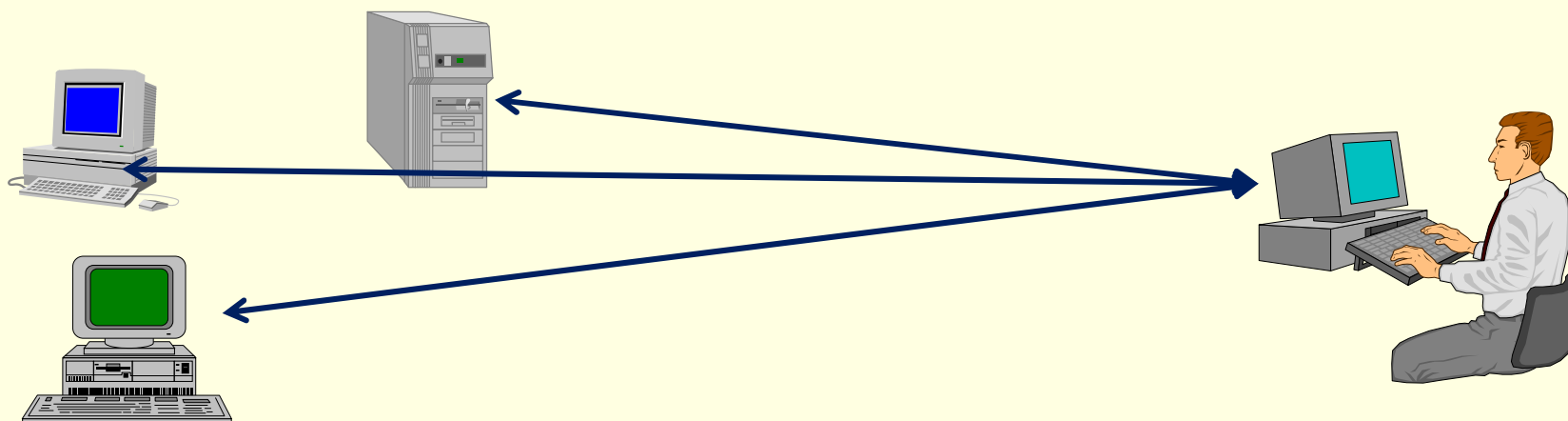


**Década de 90**

**vários computadores**



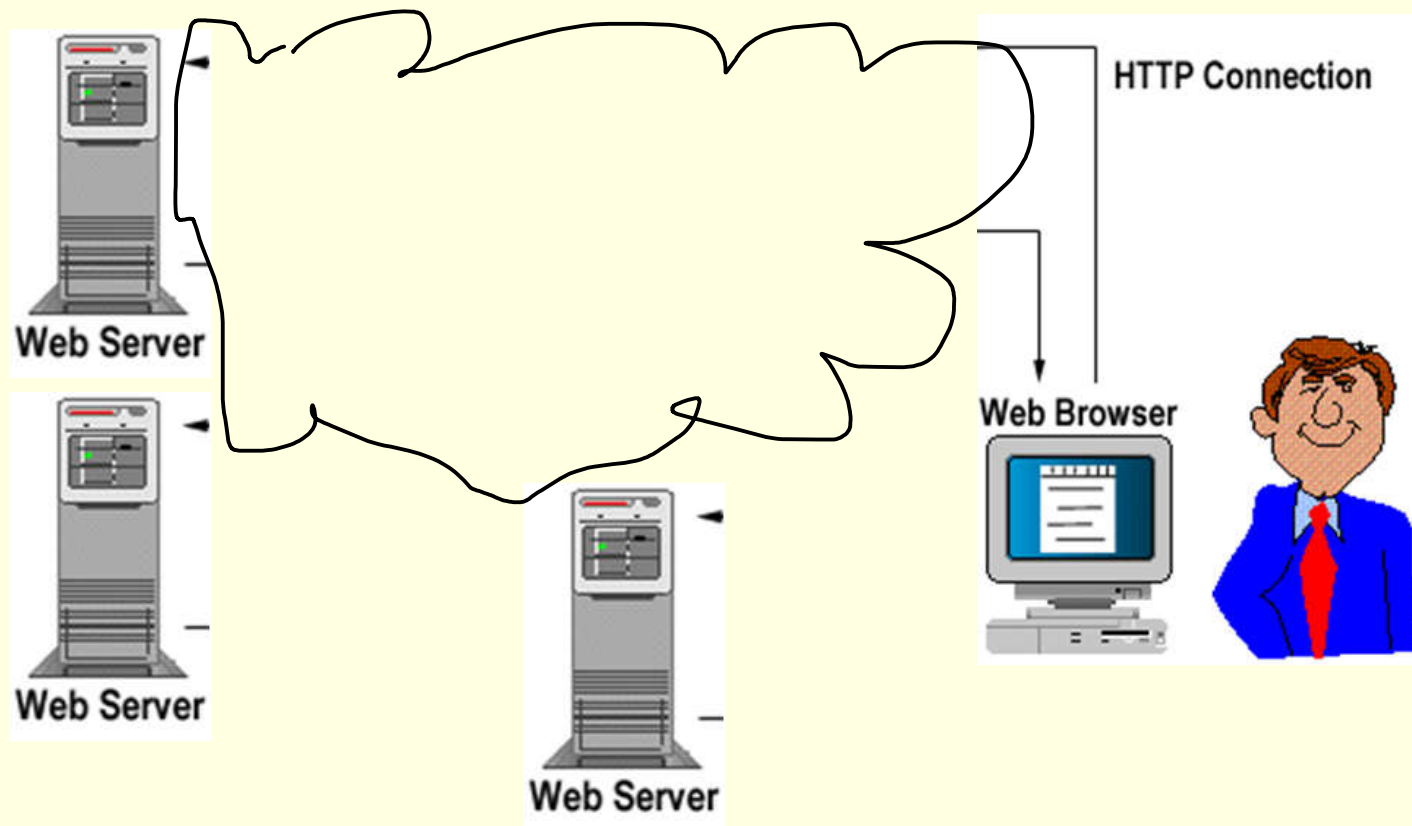
**1 pessoa**



**O avanço e estabilidade das TIT permite que os esforços de desenvolvimento se concentrem em novos serviços e aplicações**



## Década 2000 ... O COMPUTADOR É A REDE!



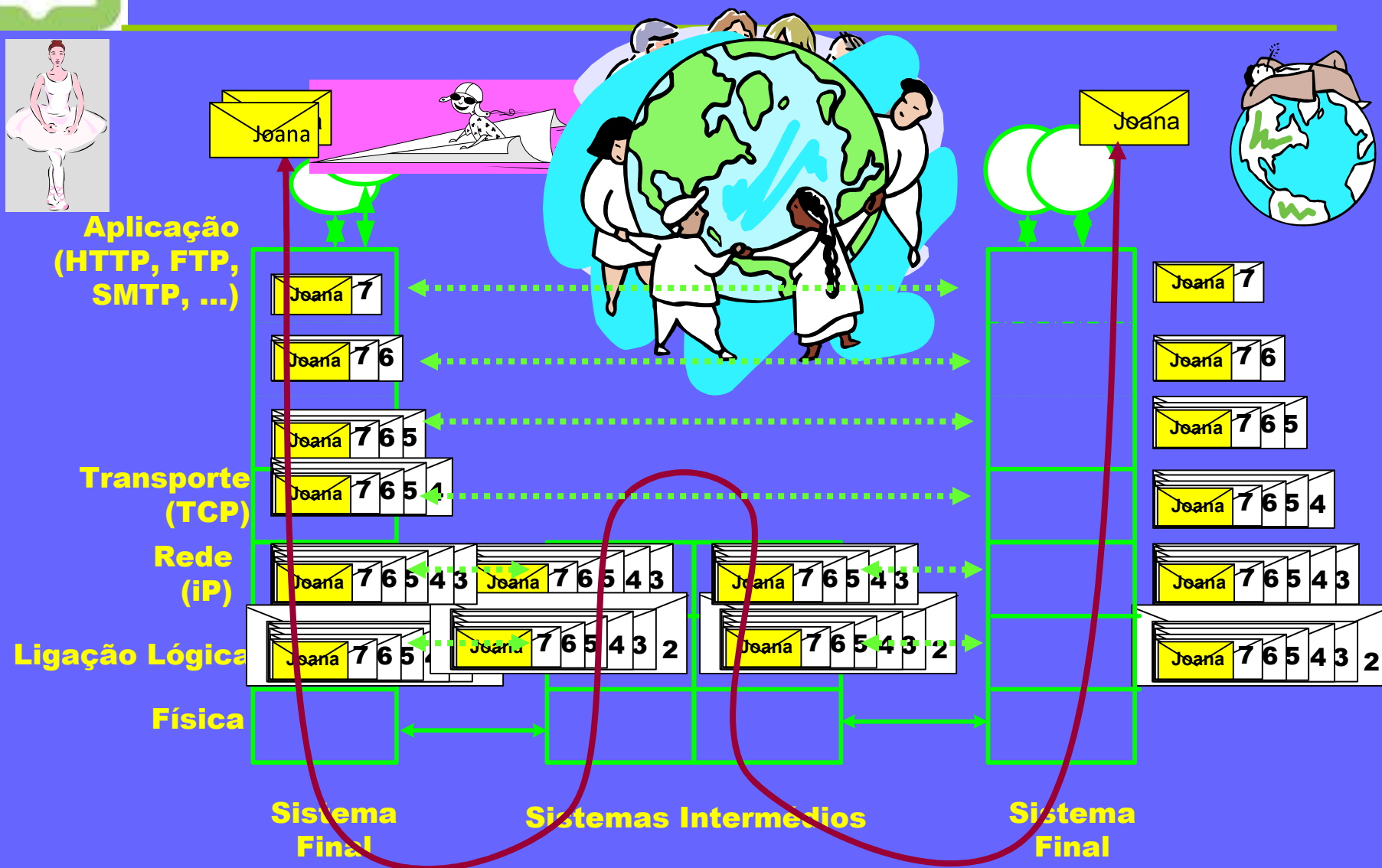


# História da Internet

- Pode-se dizer que a sua origem remonta a **1957**, ano em que USSR lança o Sputnik.
- Os USA respondem formando a ARPA (*Advanced Research Projects Agency*).
- Em **1969**, o Departamento de Defesa Americano cria a ARPANET para fazer investigação em redes sendo o primeiro nó criado na UCLA (é o nascimento do que hoje conhecemos por Internet);
- em **1971** Ray Tomlinson inventa a primeira forma de correio electrónico (E-Mail)
- em **1974** Vinton Cerf e Bob Kahn projectam o TCP (*Transmission Control Protocol*) um dos protocolos que, conjuntamente com o IP (Internet Protocol, criado em **1978**), são a base da Internet actual;
- no mesmo ano Robert Metcalfe projecta a rede local *Ethernet*, que é usada hoje em dia na maioria das redes locais, quer em nossas casas quer nas empresas. A rede contava então com cerca de 60 computadores;
- em **1990/91** a ARPANET acaba e a Internet começa a ser gerida comercialmente;
- em **1992** no CERN, Tim Berners-Lee lançou a *World Wide Web* (WWW), sendo uma data marcante pois foi a partir daí que o grande público passou a ter uma forma mais fácil de aceder e utilizar a rede através de uma forma gráfica mais amigável, usando os conhecidos *browsers*, permitindo assim a sua rápida massificação. A rede contava nessa altura cerca 1.000.000 de computadores.
- Em **1994** a Internet faz 25 anos e conta com 3.000.000 computadores;
- Em **2009** o número de computadores ligados à rede ultrapassa os 700.000.000. (<https://www.isc.org/solutions/survey/>)



# Modelo OSI/Internet (tcp/ip)





# HTTP - Hypertext Transfer Protocol

<http://www.w3.org/Protocols/>

## HTTP 1.0 (RFC 1945), HTTP 1.1 (RFC 2616, RFC 723x)

- <initial line, different for request vs. response>
- Header1: value1
- Header2: value2
- Header3: value3
- <optional message body goes here, like file contents or query data; it can be many lines long, or even binary data \$&\*%@!^\$@>



# HTTP - Hypertext Transfer Protocol

## Pedido:

- GET /path/file.html HTTP/1.0
- From: someuser@jmarshall.com
- User-Agent: HTTPTool/1.0
- [blank line here]

## Resposta:

- HTTP/1.0 200 OK
- Date: Fri, 31 Dec 1999 23:59:59 GMT
- Content-Type: text/html
- Content-Length: 1354
- <html>
- <body>
- <h1>Happy New Millennium!</h1>
- (more file contents)
- .
- .
- </body>
- </html>





# HTTP - Hypertext Transfer Protocol

- Comandos: **GET, POST, HEADER, .....**
- Respostas, erros:
  - **1xx** indicates an informational message only
  - **2xx** indicates success of some kind
  - **3xx** redirects the client to another URL
  - **4xx** indicates an error on the client's part
  - **5xx** indicates an error on the server's part

## Erros mais comuns:

- **200 OK** The request succeeded, and the resulting resource (e.g. file or script output) is returned in the message body.
- **404 Not Found**
  - The requested resource doesn't exist.
- **301 Moved Permanently**
- **302 Moved Temporarily**
- **303 See Other** (*HTTP 1.1 only*)
  - The resource has moved to another URL (given by the **Location:** response header), and should be automatically retrieved by the client. This is often used by a CGI script to redirect the browser to an existing file.
- **500 Server Error**



# HTML – Hyper Text Markup Language

<https://whatwg.org>

<http://www.w3schools.com/html/>

Tipo do documento – HTML5

elementos

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="utf-8">
```

```
<link rel="stylesheet" type="text/css" href="teste.css"/>
```

```
<title>TPW</title>
```

```
<!-- comentários -->
```

```
</head>
```

```
<body>
```

```
<h1>teste TPW</h1>
```

```
<div class="teste2" >
```

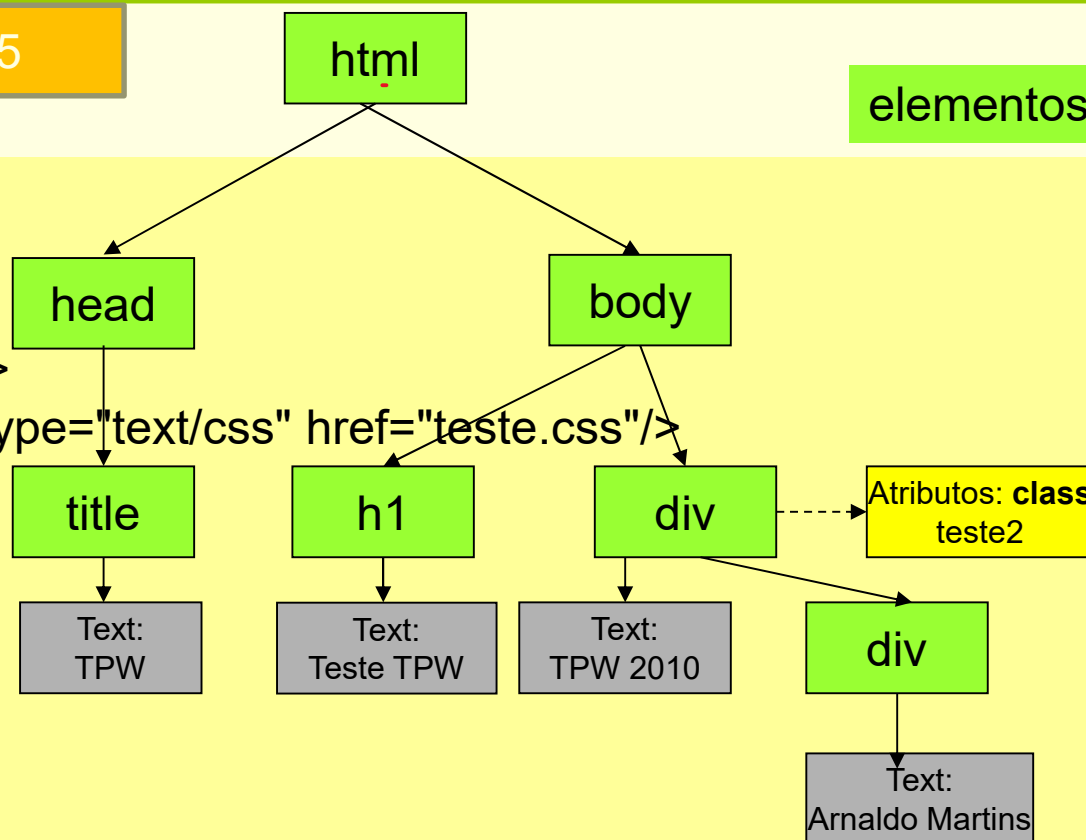
```
PW 2020
```

```
<div class='teste'>Arnaldo Martins</div>
```

```
</div>
```

```
<body>
```

```
</html>
```





# XML versus HTML

**Extensible Markup Language (XML)** <http://www.w3.org/XML/>

- O HTML é um caso particular de XML, que é genérico e aberto

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<books>
```

```
<book >
```

```
<title>The Da Vinci Code</title>
```

```
<author>Brown, Dan</author>
```

```
<publisher>Doubleday</publisher>
```

```
<price>$24.95</price>
```

```
<contentType>Fiction</contentType>
```

```
<format>Hardback</format>
```

```
<isbn>0385504209</isbn>
```

```
</book>
```

```
<book ninterno = 7832 >
```

```
<title>State of Fear</title>
```

```
<author>Crichton, Michael</author>
```

```
<publisher>HarperCollins</publisher>
```

```
<price>$27.95</price>
```

```
<contentType>Fiction</contentType>
```

```
<format>Hardback</format>
```

```
<isbn>0786868716</isbn>
```

```
</book>
```

```
</books>
```



# Modelo MVC

- Separar Lógica (javascript/HTML) da apresentação (CSS)

```
<html>Exemplos\tpw080925-css0.html
```

```
<title>TPW</title>
```

```
<body>
```

```
  <h1>teste TPW</h1>
```

```
  <div class="teste2" >
```

```
    TPW 2008
```

```
    <div class='teste'>Arnaldo Martins</div>
```

```
  </div>
```

```
</body>
```

```
</html>
```



```
<html_>Exemplos\tpw080925-css1.html
<link rel="stylesheet" type="text/css" href="teste.css"/>
<title>TPW</title>
<body>
  <h1>teste TPW</h1>
  <div class="teste2" >

    TPW 2008
    <div class='teste'>Arnaldo Martins</div>

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

```
h1,h2,h3 {
    color: blue;
}
.teste {
    color: red;
    position: absolute;
    top: 150;
    left: 400;
}
```

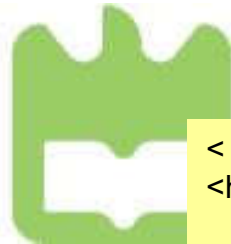


# DOM – acesso aos elementos

```
< html>Exemplos\tpw080925-tabela.htm
<head>
  <title>Untitled</title>
</head>
<body>

  <table summary="" border="1">
    <tr><td>graus</td><td>seno</td><td>coseno</td></tr>
    <script ty type="text/javascript">
      var anginicial=parseFloat(prompt("Angulo inicial=",0));
      var angfinal = parseFloat(prompt("Angulo final=",0));
      var delta = parseFloat(prompt("incremento=",0));
      for (ang=anginicial;ang <= angfinal; ang = ang + delta){
        document.write("<tr><td>" + ang + "</td><td>" + Math.sin(ang/180.*Math.PI) + "
      </td><td>" + Math.cos(ang/180.*Math.PI) + "</td></tr>")
      }
    </script>

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



# Aceder a um elemento e acrescentar +HTML

```
<html>Exemplos\tpw080925-tabela-modificarElemento.htm
<head>
  <title>Untitled</title>
  <script type="text/javascript">
    function tab(ang1, angf, adelta){
      var anginicial= parseInt(ang1);
      var angfinal = parseInt(angf);
      var delta = parseInt(adelta);
      var ang=0;

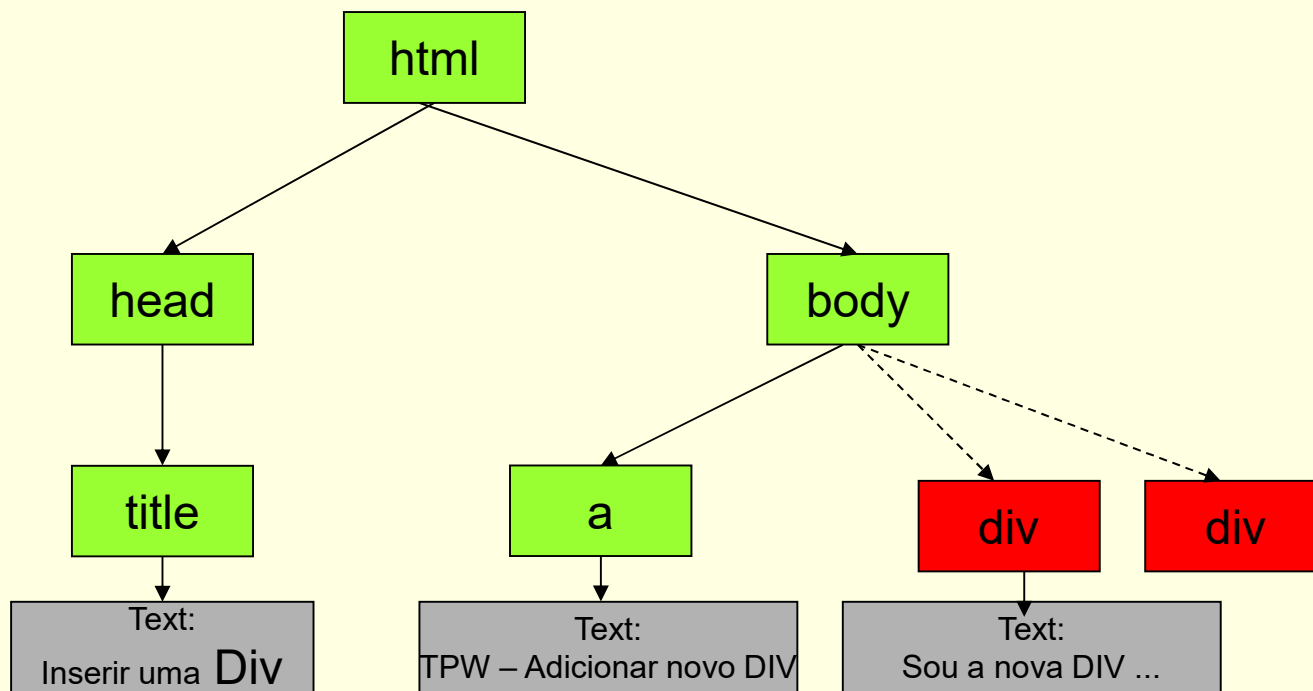
      var resultado = "<table border='1'><tr><td>graus</td><td>seno</td><td>coseno</td></tr>";
      for (ang=anginicial;ang <= angfinal; ang = ang + delta){
        resultado = resultado
        +"<tr><td>" +ang.toString()+"</td><td>" +Math.sin(ang/180.*Math.PI).toFixed(4)+"</td><td>" +Math.cos
        (ang/180.*Math.PI).toFixed(4)+"</td></tr>";
      }
      resultado = resultado + "</table>";
      return resultado;
    }
  </script>
</head>
<body>
  <input id="ai" type="text" value="0" size="4" />
  <input id="af" type="text" value="90" size="4"/>
  <input id="ainc" type="text" value="10" size="4" />
  <button onclick="document.getElementById('tabela').innerHTML=tab(ai.value, af.value,
  ainc.value);">calc</button>

  <div id="tabela"></div>

</body>
</html>
```



# Árvore dos nós HTML



O texto é  
sempre um nó!





# Criar um elemento novo (append)

```
<html><head><title> Exemplos\tpw080925-appendElemento.html - </title>
<script type = "text/javascript">
function addDiv() {
    var newDiv = document.createElement("div");
    newDiv.setAttribute("id","newDiv");
    var bodyTags = document.getElementsByTagName("body");
    var thisBody = bodyTags[0];
    thisBody.appendChild(newDiv);
    var textNode = document.createTextNode("Sou a nova DIV...");
    newDiv.appendChild(textNode);
}
</script>
</head>
<body>
<a href = "#" onClick = "addDiv(); return false;">TPW - Adicionar novo div.</a>
</body>
</html>
```

o **return false** no fim do evento faz com que a tag HTML a não seja executada

# Inserção e Remoção de um Elemento

```
<html><head><title> Exemplos\tpw080925-InserirElemento.html - </title>
```

```
<script type = "text/javascript">
```

```
function addSpouse(theRadio) {  
  if (document.getElementById("newDiv") == null) {  
    var newDiv = document.createElement("div");  
    newDiv.setAttribute("id", "newDiv");  
    var newInput = document.createElement("input");  
    newInput.setAttribute("type", "text");  
    newInput.setAttribute("id", "spouseText");  
    var nameText = document.createTextNode(" Nome da(o) Esposa(o)");  
    newDiv.appendChild(newInput);  
    newDiv.appendChild(nameText);  
    newDiv.appendChild(document.createElement("br"));  
    var theForm = document.getElementById("myForm");  
    var theAddress = document.getElementById("address");  
    theForm.insertBefore(newDiv, theAddress);  
  }  
}
```

```
function removeSpouse....  
</script>  
</head>
```

```
function removeSpouse(theRadio) {  
  if (document.getElementById("newDiv") != null) {  
    var newDiv =  
    document.getElementById("newDiv");  
    var theForm =  
    document.getElementById("myForm");  
    myForm.removeChild(newDiv);  
  }  
}
```

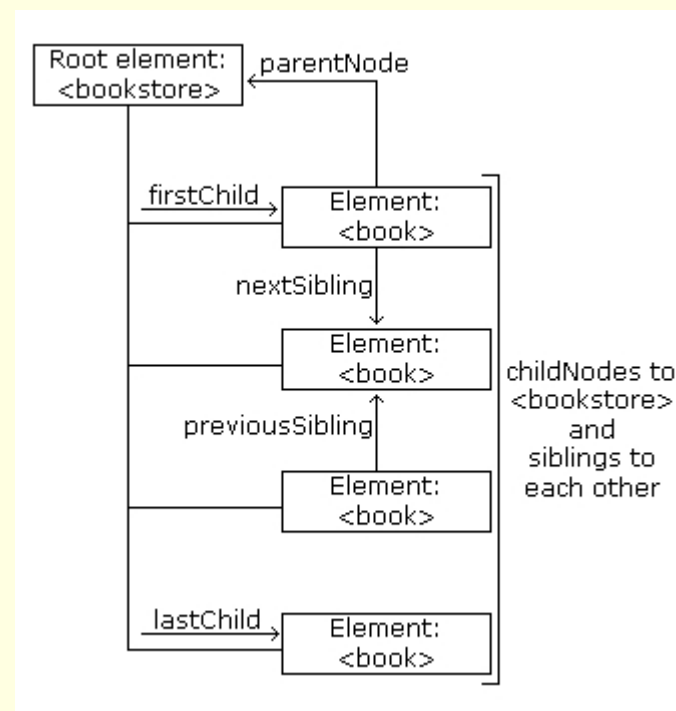
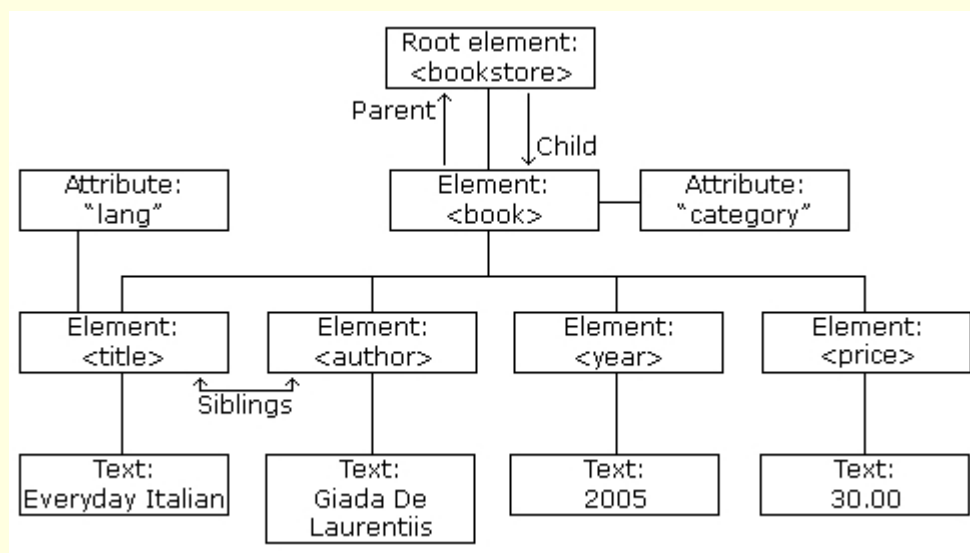
```
<body>  
<form id = "myForm">  
<input type = "text" id = "name"> Nome<br>  
Estado Civil:<br>  
  <input type = "radio" onClick = "removeSpouse(this);" name = "ms">Solteiro(a)  
<br>  
  <input type = "radio" onClick = "addSpouse(this);" name = "ms"> Casado(a)<br>  
<input type = "text" id = "address"> Endereco <br>  
</form>  
</body>  
</html>
```



# DOM

- Exemplo de um ficheiro XML de uma livraria

Exemplos\books-dom.xml





## Node Object Methods (W3C DOM Level 2)

Method	Description	IE5+	Moz1	Safari 1
<code>appendChild(newChild)</code>	Adds child node to end of current node	Yes	Yes	Yes
<code>cloneNode(deep)</code>	Grabs a copy of the current node (optionally with children)	Yes	Yes	Yes
<code>hasChildNodes()</code>	Determines whether current node has children (Boolean)	Yes	Yes	Yes
<code>insertBefore(new, ref)</code>	Inserts new child in front of another child	Yes	Yes	Yes
<code>removeChild(old)</code>	Deletes one child	Yes	Yes	Yes
<code>replaceChild(new, old)</code>	Replaces an old child with a new one	Yes	Yes	Yes
<code>isSupported(feature, version)</code>	Determines whether the node supports a particular feature	No	Yes	Yes

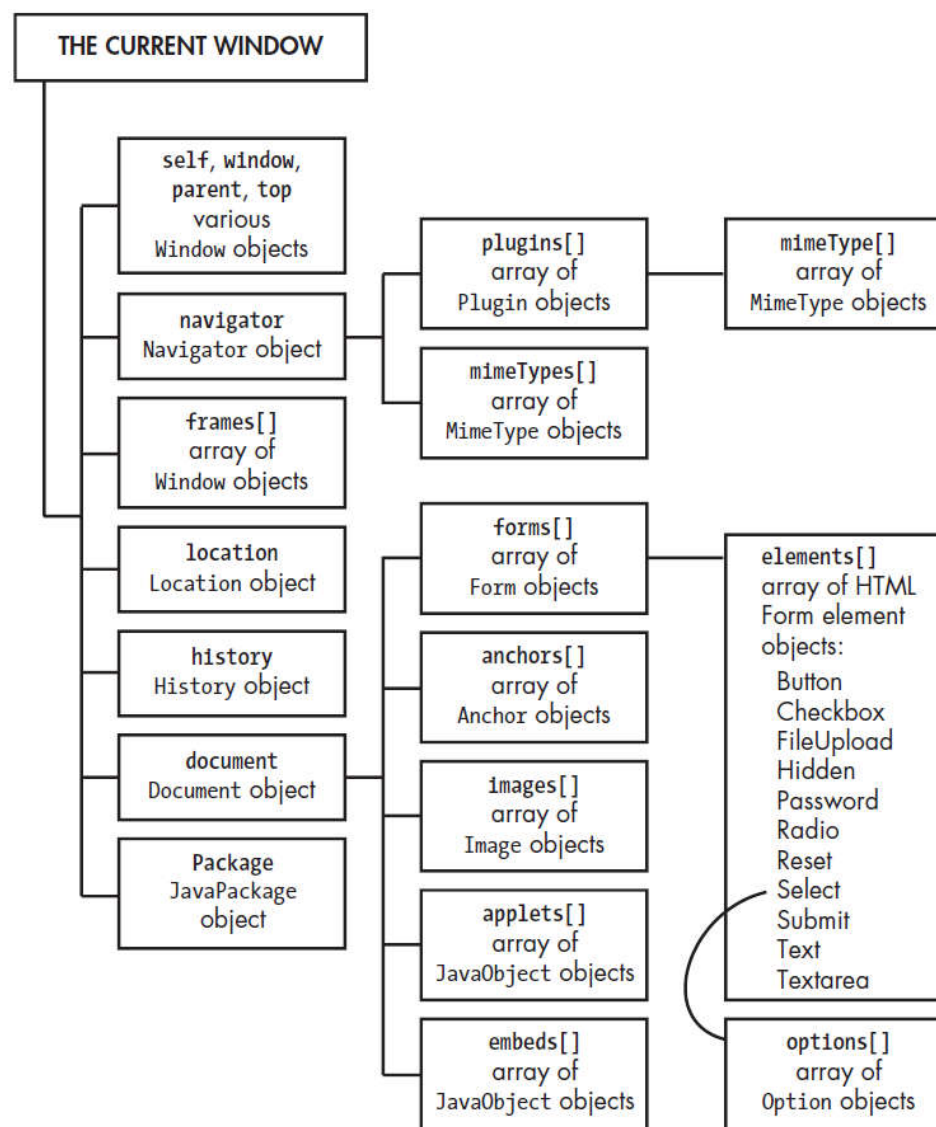


# DOM – propriedades dos nós

Node	Description
parentNode	Parent node of the node
childNodes	List of the children of the node
firstChild	First child of the node
lastChild	Last child of the node
nextSibling	Next node, which is a child of this node's parent
previousSibling	Previous node, which is a child of this node's parent
nodeValue	Text of the node if it's a text node, null otherwise
nodeType	Elements are type 1, attributes are type 2, text nodes are type 3
nodeName	Name of the attribute or node (h1, br, or form, for example); some browsers capitalize these names (BR, H1)
attributes	Array of attributes of this node



# DOM – Objectos do browser





## Objecto Style – altera estilo de um elemento

```
<html>Exemplos\Objecto_style.html -
```

```
<head>
```

```
<script type="text/javascript">
```

```
function ChangeText() {
```

```
    var p1 = document.getElementById("p1")
```

```
    var display = p1.style.display ? " : 'none';
```

```
    p1.style.color="blue";
```

```
    p1.style.fontFamily="Arial";
```

```
    p1.style.display= display
```

```
}
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<p id="p1">Hello world!</p>
```

```
<input type="button" onclick="ChangeText()" value="Click me to change text  
above">
```

```
</body>
```

```
</html>
```



# XML – exemplos de algumas normas

- Books: XML, Representação com CSS

Exemplos\books0.xml

..\books.xml

..\books.css

- Movie: XML, Representação com XSLT

Exemplos\movie0.xml

..\movie.xml

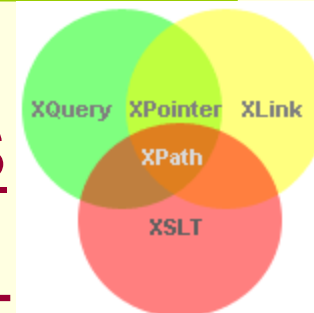
..\movie.xsl

- XHTML, XForms, SVG, .....

Exemplos\svg1.xml

- XPath

- é uma linguagem para encontrar informação num documento XML







# XPath - exemplo

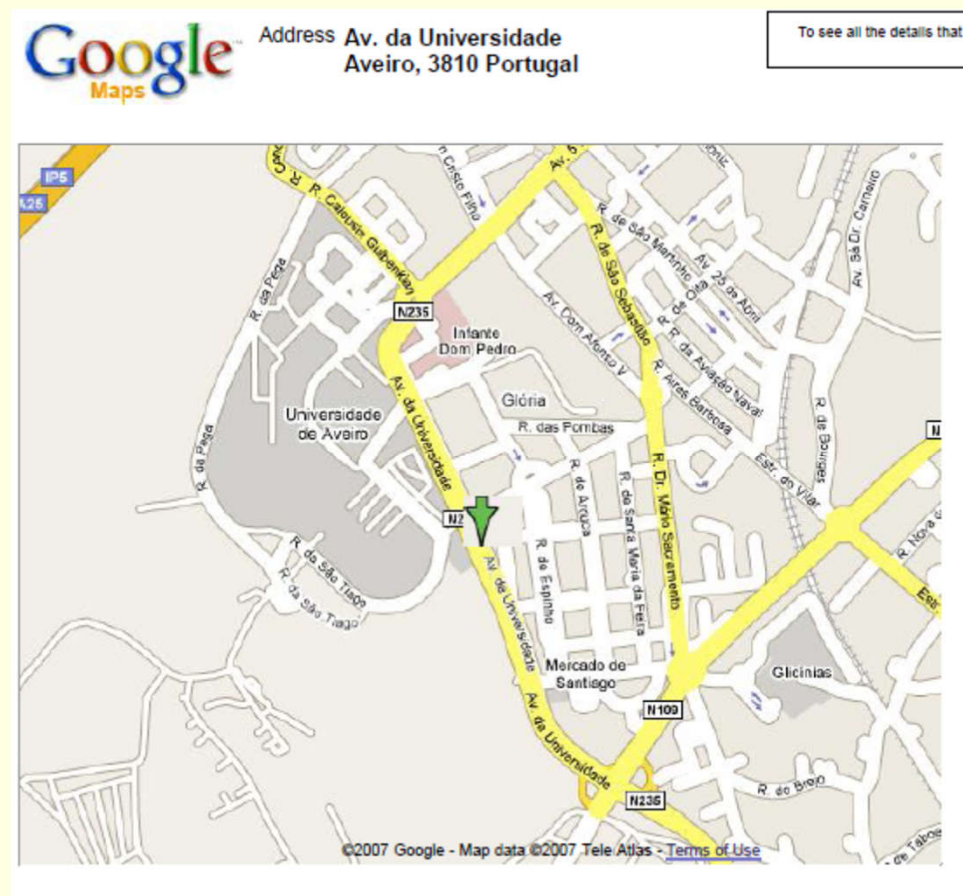
```
<Task>
  <TaskItem id = "123"
    value="Status Report"/>
  <TaskItem id = "124"
    value="Writing Code"/>
  <TaskItem value="Idle Chat"/>
  <Meeting id="125"
    value="Daily Briefing"/>
</Task>
```

XPath Expression	What the Expression Means	Return Value
<code>//TaskItem[@id]</code>	"Give me all TaskItem elements that have ID attributes"	<pre>&lt;TaskItem id = "123" value="Status Report"/&gt; &lt;TaskItem id = "124" value="Writing Code"/&gt;</pre>
<code>//*[@id]</code>	"Give me all ID attributes"	<pre>id = "123" id = "124" id = "125"</pre>
<code>/Task/Meeting</code>	"Select all elements named 'Meeting' that are children of the root element 'Task'"	<pre>&lt;Meeting id="125" value="Daily Briefing"/&gt;</pre>



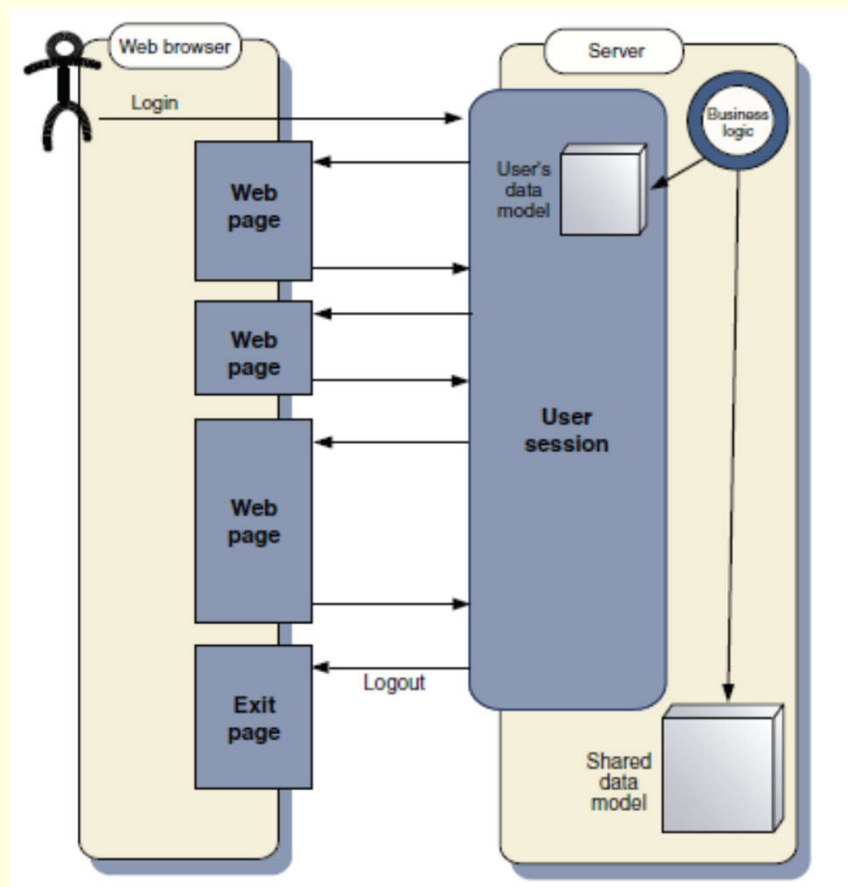
# AJAX

- Asynchronous JavaScript and XML
  - Exemplos: Google maps, mail, office.....





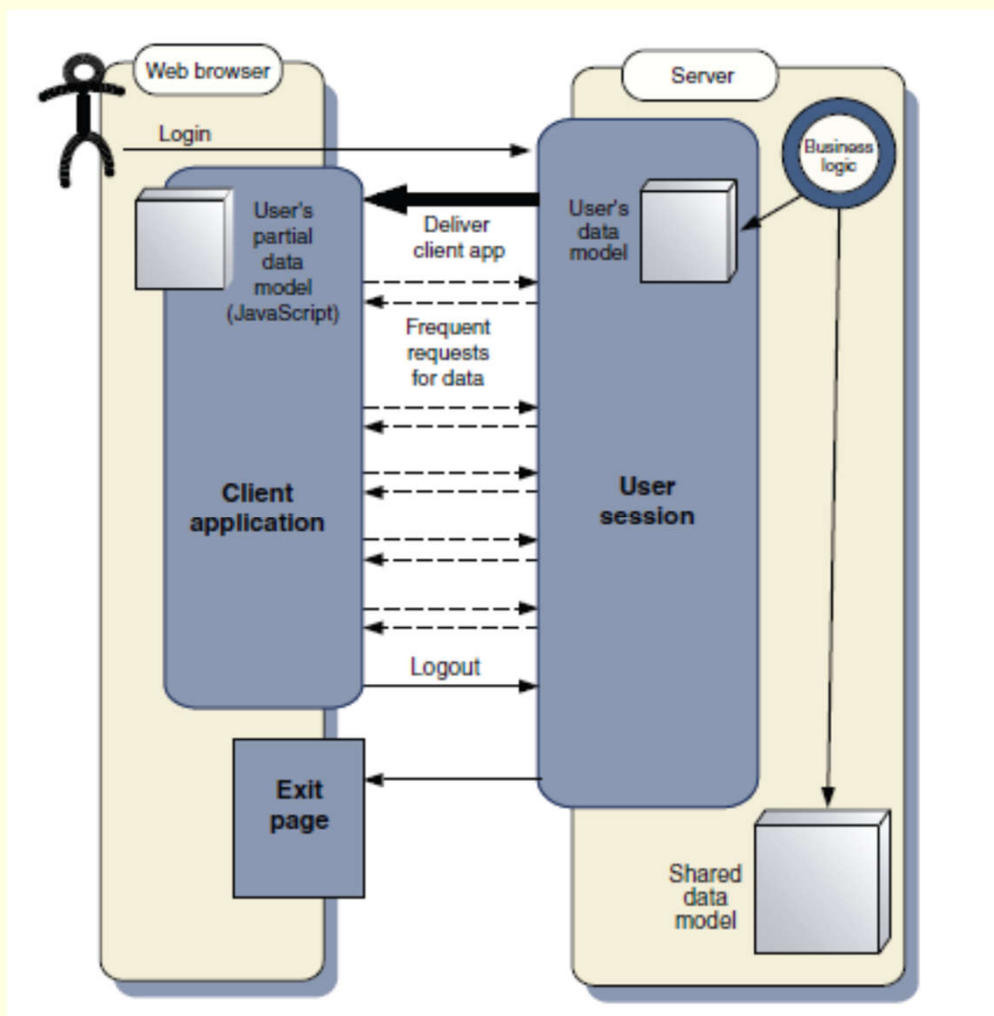
# AJAX



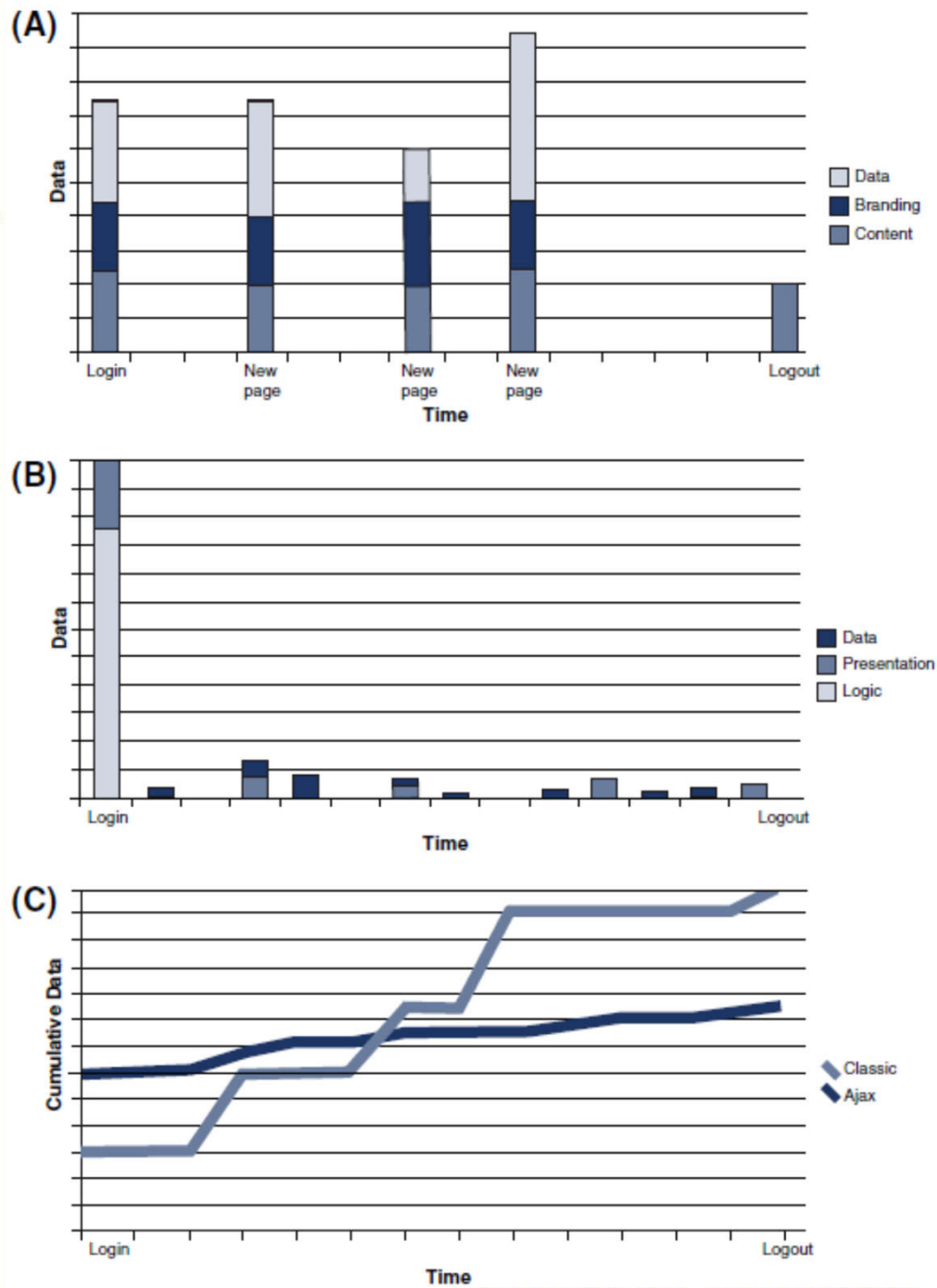
Ciclo de vida de uma aplicação Web Clássica



# AJAX



Ciclo de vida de uma aplicação AJAX



# AJAX

- (A) – Clássica
- (B) - AJAX



# AJAX

- Passos para criar uma aplicação AJAX
  - Criar um objecto para o pedido
    - `var request = new XMLHttpRequest("Microsoft.XMLHTTP");` (IE)
    - `var request = new XMLHttpRequest();` (Firefox)
  - Dizer ao objecto para onde enviar o pedido
    - `request.open("GET", "my_file.txt");`
  - Dizer ao objecto o que fazer quando chegar a resposta
    - `request.onreadystatechange = function() {  
    if (request.readyState == 4) {  
        alert("Download complete! ");  
    }}`
- Dizer ao objecto para fazer o pedido
  - `request.send(null);`



# AJAX

**Table 14-1:** Values of a Request Object's `readyState` Property

Property Value	State Name	Description
0	Uninitialized	The object has been created but not told about the request: <code>open()</code> has not been called.
1	Loading	The object knows about the request but has not sent it yet: <code>send()</code> has not been called.
2	Loaded	The request has been sent, and basic information about the response is available.
3	Interactive	The response is being loaded into the request object.
4	Completed	The entire response has been loaded into the request object and is now available.



# AJAX – exemplo fich texto local

```
<_html>Exemplos\ajax1local-filetxt.htm -  
<head>  
  <title>AJAX - teste</title>  
  <script type="text/javascript">  
    function doAjaxCall(the_request) {  
      .....  
    }  
</script>  
</head>  
<body>  
  <form>  
    File de texto no disco local:  
    <input type="button" value="Fazer Pedido Ajax"  
      onclick="doAjaxCall('sample.txt'); return true;">  
  </form>  
  Resultado:<br>  
  <div id="resultDiv"></div>  
</body>  
</html>
```

Ficheiro de texto local

'sample.txt?nocache='+Math.random()  
*Permite ultrapassar a cache criando URLs  
sempre diferentes!*





# AJAX – criação, pedido, resposta

```
function doAjaxCall(the_request) {  
    var request = null;                                // Cria objecto AJAX  
    if (window.XMLHttpRequest) {                       // FireFox  
        request = new XMLHttpRequest();  
    } else if (window.ActiveXObject) {                 // IE  
        request = new ActiveXObject("Microsoft.XMLHTTP");  
    }  
    if (request) {  
        request.open("GET", the_request);  
        request.onreadystatechange = function() {  
            if (request.readyState == 4) {  
                var texto = request.responseText;  
                document.getElementById("resultDiv").innerHTML = texto  
            }  
        }  
        request.send(null);  
    } else {  
        alert("Desculpe, mas deve actualizar o browser antes de ver o Ajax  
em accao.");  
    }  
}
```



# AJAX – Exemplo fich XML local

```
<html>Exemplos\ajax1localfilexml.htm -
```

```
<head>
```

```
<title>AJAX - teste</title>
```

```
<script type="text/javascript">
```

```
function doAjaxCall(the_request) {
```

```
.....
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<form>
```

File xml no disco local:

```
<input type="button" value="Fazer Pedido Ajax"
```

```
onclick="doAjaxCall('sample.xml');return true;">
```

```
</form>
```

```
Resultado:<br>
```

```
<div id="resultDiv">
```

```
</div>
```

```
</body>
```

```
</html>
```

↓

```
<cinema>
```

```
<filme>Voando sobre um ninho de Cucos</filme>
```

```
<filme>E tudo o vento levou</filme>
```

```
<filme>Crash</filme>
```

```
</cinema>
```



# AJAX – criação, pedido, resposta

```
function doAjaxCall(the_request) {  
    var request = null;                                // Cria objecto AJAX  
    if (window.XMLHttpRequest) {                       // FireFox  
        request = new XMLHttpRequest();  
    } else if (window.ActiveXObject) {                 // IE  
        request = new ActiveXObject("Microsoft.XMLHTTP");  
    }  
    if (request) {  
        request.open("GET", the_request);  
        request.onreadystatechange = function() {  
            if (request.readyState == 4) {  
                var xmldoc = request.responseXML;  
                var filmes = xmldoc.getElementsByTagName("filme")  
                document.getElementById("resultDiv").innerHTML = "Titulo do filme  
= " + filmes[1].firstChild.nodeValue;  
            }  
            request.send(null);  
        } else {  
            alert("Desculpe, mas deve actualizar o browser .....");  
        }  
    }  
}
```

JAM



# AJAX – exemplo fich XML remoto

```
<html>http://localhost:8080/TPW/ajax1serverfilexml.html
```

```
<head>
```

```
<title>AJAX - teste</title>
```

```
<script type="text/javascript">
```

```
function doAjaxCall(the_request) {
```

```
.....
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<form>
```

File xml no servidor:

```
<input type="button" value="Fazer Pedido Ajax"
```

```
onclick="doAjaxCall('http://localhost:8080/TPW/sample.xml'); return true;">
```

```
</form>
```

Resultado:<br>

```
<div id="resultDiv">
```

```
</div>
```

```
</body>
```

```
</html>
```

```
<cinema>
```

```
<filme>Voando sobre um ninho de Cucos</filme>
```

```
<filme>Atracao Fatal</filme>
```

```
<filme>Zorro</filme>
```

```
</cinema>
```



# AJAX – exemplo de XML gerado no servidor

```
<html>http://localhost:8080/TPW/ajax1servergxml.html -
```

```
.....
```

```
<body>
```

```
  <form>
```

```
    Servidor Groovy:
```

```
    <input type="button" value="Fazer Pedido Ajax"
```

```
    onclick="doAjaxCall('http://localhost:8080/TPW/xml.gsp'); return  
    true;">
```

```
  </form>
```

```
  Resultado:<br>
```

```
  <div id="resultDiv">
```

```
  </div>
```

```
</body>
```

```
.....
```

```
<%  
// gerar xml  
response.contentType = 'text/xml'  
out << '<?xml version="1.0" encoding="ISO-8859-1"?>'  
out << '<titl>groovy ajax-xml-teste</titl>'  
  
%>
```



# AJAX – modo assíncrono

## Exemplo do funcionamento ASSINCRONO

<http://localhost:8080/TPW/ajax2-modoassincrono.html>

.....  
<script type="text/javascript">

```
function demoAsync() {  
  var now = new Date();  
  downloadFile("longWait", now);  
  downloadFile("shortWait", now);  
}
```

```
function downloadFile(the_request, start_time) {
```

```
  var request = null;
```

```
  if (window.XMLHttpRequest) {
```

```
    request = new XMLHttpRequest();
```

```
  } else if (window.ActiveXObject) {
```

```
    request = new ActiveXObject("Microsoft.XMLHTTP");
```

```
  }
```

```
  if (request) {
```

```
    request.open("GET", "http://localhost:8080/TPW/" + the_request + ".gsp" );
```

```
    request.onreadystatechange = function() {
```

```
      if (request.readyState == 4) {
```

```
        document.getElementById("resultDiv").innerHTML +=
```

```
          "O Ficheiro " + the_request + " fez o download em " +  
          getExpiredTime(start_time) + " segundos <br>";
```

```
      }
```

```
    }
```

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

JAM

.....

```
<%  
sleep 10000  
println "10 segundos"  
%>
```

```
<%  
sleep 3000  
print "3 segundos"  
%>
```

A resposta ao primeiro pedido  
(longWait)

Aparece depois da resposta do  
segundo pedido (shortWait)

*Em modo síncrono o segundo pedido  
só seria satisfeito depois do primeiro.*



# AJAX - GET

<http://localhost:8080/TPW/ajax-get.html>

```
function submitMe(my_form) {  
  var request = null;  
  var message = "Nome=" + my_form.elements["personName"].value + "&anuncio=" +  
    my_form.elements["board"].value;  
  request = new XMLHttpRequest();  
  var the_URL = "http://localhost:8080/TPW/ajax1.gsp";  
  if (request) {  
    request.open("GET", the_URL + "?" + message, true);  
    request.setRequestHeader("Content-Type", "application/x-www-form-urlencoded; charset=UTF-8");  
    request.onreadystatechange = function() {  
      if (request.readyState == 4 && request.status == 200) {  
        document.getElementById("display").innerHTML = request.responseText;  
      }  
    }  
    request.send(null);  
  } else { alert("Sorry, you must update your browser ..."); }  
}
```

```
.....  
<body>  
<h1>Anúncios</h1>  
<div id = "display">  
  <form>  
    O seu Nome: <input name = "personName" type = "text"><br>  
    O seu Anúncio: <br>  
    <textarea name = "board" rows = "10" cols = "40"></textarea><br>  
    <input type = "button" value = "Colocar Anúncio" onmouseup="submitMe(this.form);return false">  
  </form>  
</div>  
</body>
```

→

```
<%  
print "AJAX GET/POST</br>"  
print request.toString() + "</br>"  
print params.anuncio.toUpperCase()  
+ "</br>" + params.Nome  
%>
```



# AJAX - POST

<http://localhost:8080/TPW/ajax-post.html>

```
function submitMe(my_form) {  
    var request = null;  
    var message = "Nome=" + my_form.elements["personName"].value + "&anuncio=" +  
        my_form.elements["board"].value;  
    request = new XMLHttpRequest();  
    var the_URL = "http://localhost:8080/TPW/ajax1.gsp";  
    if (request) {  
        request.open("POST", the_URL, true);  
        request.setRequestHeader("Content-Type", "application/x-www-form-urlencoded; charset=UTF-8");  
        request.onreadystatechange = function() {  
            if (request.readyState == 4 && request.status == 200) {  
                document.getElementById("display").innerHTML = request.responseText;  
            }  
        }  
        request.send(message);  
    } else { alert("Sorry, you must update your browser ..."); }  
}
```

```
.....  
<body>  
<h1>Anúncios</h1>  
<div id = "display">  
    <form>  
        O seu Nome: <input name = "personName" type = "text"><br>  
        O seu Anúncio: <br>  
        <textarea name = "board" rows = "10" cols = "40"></textarea><br>  
        <input type = "button" value = "Colocar Anúncio" onmouseup="submitMe(this.form);return false">  
    </form>  
</div>  
</body>
```

```
<%  
print "AJAX GET/POST</br>"  
print request.toString() + "</br>"  
print params.anuncio.toUpperCase()  
+ "</br>" + params.Nome  
%>
```





# AJAX – enviar dados XML para servidor – grava ficheiro (1)

```
<html><head><title>http://localhost:8080/TPW/ajax-XMLsave.html</title>
<script type = "text/javascript">.....
.....
</script>
</head>
<body>
    <h1>Sending XML to a file in the server...</h1>

    <form id='teste'>
        Your XML: <br>
        <textarea name = "xml" rows = "10" cols = "40"></textarea><br>
        <input type = "button" value = "Save Your XML"
            onmouseup="submitMe(this.form);return false">
    </form>
    <div id = "display"></div>
</body>
</html>
```



# AJAX – enviar dados XML para servidor - grava ficheiro (2)

```
function submitMe(my_form) {  
  var message = "<?xml version='1.0' ?>" + my_form.elements["xml"].value;  
  var the_URL = "http://localhost:8080/TPW/ajax2saveXML.gsp";  
  var request = null;  
  request = new XMLHttpRequest();  
  if (request) {  
    request.open("POST", the_URL, true);  
    request.setRequestHeader("Content-Type", "text/xml; charset=UTF-8");  
    request.onreadystatechange = function() {  
      if (request.readyState == 4 ) {  
        var texto = request.responseText;  
        document.getElementById("display").innerHTML = texto  
      }  
    }  
    request.send(message);  
  } else {  
    alert("Sorry, you must update your browser");  
  }  
}
```

```
<%  
def Fich = new File('xmltest.txt')  
Fich << request.getInputStream()  
  
print "<h2>ok</h2>"  
%>
```



# Ajax – exemplo dicionário

```
<?xml version = "1.0" ?>
```

```
<dictionary>
```

```
<word>
```

```
<english>blue</english>
```

```
<translation>azul, dragões</translation>
```

```
</word>
```

```
<word>
```

```
<english>red</english>
```

```
<translation>vermelho, encarnado</translation>
```

```
</word>
```

```
<word>
```

```
<english>green</english>
```

```
<translation>verde, lagartos</translation>
```

```
</word>
```

```
</dictionary>
```

Problemas no parsing do XML!!

*Espaços, linhas, ... Podem ser erradamente contabilizados como nós ....*

```
<?xml version = "1.0" ?><dictionary><word><english>blue</english><translation>blu, azzurro</translation></word><word><english>red</english><translation>rosso</translation></word><word><english>green</english><translation>verde</translation></word></dictionary>
```



# Ajax – exemplo dicionário(2)

```
<html><head><title>http://localhost:8080/TPW/ajaxTraduzir1.html</title>
<script type = "text/javascript">
```

```
function getTranslations(the_word) {
  var languages = new Array("fr","it","pt");
  for (var loop = 0; loop < languages.length; loop++) {
    getTranslationFromFile(languages[loop], the_word);
  }
}
```

.....

```
</script>
```

```
</head>
```

```
<body>
```

```
<form>
```

```
  Inglês:<input type = "text" id = "theText">
```

```
  <input type = "button" onClick = "getTranslations(document.getElementById('theText').value);
    return false;" value = "Traduzir!">
```

```
</form>
```

```
<b>Português:</b> <span id = "pt"></span><br>
```

```
<b>Italiano:</b> <span id = "it"></span> <br>
```

```
<b>Francês:</b> <span id = "fr"></span> <br>
```

```
</body>
```

```
</html>
```



# Ajax – exemplo dicionário(3)

```
function getTranslationFromFile(the_file, the_word) {  
    var request = null;  
    request = new XMLHttpRequest();  
    if (request) {  
        request.open("GET", the_file + ".xml");  
        request.onreadystatechange = function() {  
            if (request.readyState == 4) {  
                xml_response = request.responseXML;  
                document.getElementById(the_file).innerHTML =  
                    findTranslation(xml_response, the_word);  
            } else {  
                document.getElementById(the_file).innerHTML = "SEARCHING...";  
            }  
        }  
        request.send(null);  
    } else {  
        alert("Sorry, you must update your browser ...");  
    }  
}
```



# Ajax – exemplo dicionário(4)

```
function findTranslation(xml_doc, the_word) {  
    var the_translation = "unknown";  
    var this_word = "";  
    var this_english_element = null;  
    var english_word_elements = xml_doc.getElementsByTagName("english");  
    for (var loop = 0; loop < english_word_elements.length; loop++) {  
        this_english_element = english_word_elements[loop];  
        this_word = this_english_element.firstChild.nodeValue;  
        if (this_word == the_word) {  
            the_translation = this_english_element.nextSibling.firstChild.nodeValue;  
        }  
    }  
    return the_translation;  
}
```



# Ajax – Exemplo1 Juízes(1)

```
<juizes>
  <juiz alcunda="O justiceiro" datanasc="1-1-1947">
    <jid>1</jid>
    <nome> Jose Lima</nome>
    <res>Ovar</res>
    <email>zlima@trib.pt</email>
  </juiz>
  <juiz>
    <jid>2</jid>
    <nome> Julio Santos</nome>
    <res>Aveiro</res>
    <email>jsantos@trib2.pt</email>
  </juiz>
  <juiz>
    <jid>3</jid>
    <nome> Joana Lisboa</nome>
    <res> Aveiro</res>
    <email>jlisboa@trib2.pt</email>
  </juiz>
</juizes>
```

Juiz
jid
nome
res
email



# Ajax –Ex1 juizes(2)

*lista todos juizes*

<http://localhost:8080/TPW/ajax-juizes1.html> -

```
function doAjaxCall(the_request) {
    var request = null;
    request = new XMLHttpRequest();
    if (request) {
        request.open("GET", the_request);
        request.onreadystatechange = function() {
            if (request.readyState == 4) {
                var xmlDoc = request.responseXML;
                exemplo1(xmlDoc);
            }
        }
        request.send(null);
    } else { alert(...) }
}

function exemplo1(docxml){
    var juizes = docxml.getElementsByTagName("juiz")
    var resultado = document.getElementById("resultDiv")
    var i;
    var reshtml="JUIZ  <br>";
    for (i=0; i < juizes.length; i++){
        var juiz = juizes[i].childNodes; // jid, nome, res, email
        reshtml = reshtml +juiz[1].firstChild.nodeValue +" "+ juiz[3].firstChild.nodeValue + "<br>";
    }
    resultado.innerHTML = reshtml
}
```





# Ajax –Exemplo 2 Juízes(1)

*selecionar um juiz da lista* <http://localhost:8080/TPW/ajax-juizes2select.html> –

```
var juizes_db = new Array();    // variável global com lista de juízes
```

```
function doAjaxCall(the_request) {  
    var request = null;  
    request = new XMLHttpRequest();  
    if (request) {  
        request.open("GET", the_request);  
        request.onreadystatechange = function() {  
            if (request.readyState == 4) {  
                var xmldoc = request.responseXML;  
                juizes_db = exemplo2(xmldoc);  
            }  
        }  
        request.send(null);  
    } else { alert("deve actualizar o browser ...."); }  
}
```

```
<body>  
    <form>  
        Lista de Juizes:<input type="button" value="Fazer Pedido Ajax"  
        onclick="doAjaxCall('juizes.xml');return true;">  
    </form>  
    Resultado:<br>  
    <div id="resultDiv"></div>  
  
</body>
```



# Ajax – Exemplo2 Juízes(2)

*selecionar um juiz da lista*

```
function exemplo2(docxml) {
  var juizes = docxml.getElementsByTagName("juiz")
  var resultado = document.getElementById("resultDiv");
  var i;
  var reshtml='<select onmouseup="MostraJuiz(this.value, juizes_db)">';
  for (i=0; i < juizes.length; i++){
    var juiz = juizes[i].childNodes; // jid, nome, res, email
    reshtml += '<option value="'+i+'">'+juiz[1].firstChild.nodeValue+' </option>'
  }
  resultado.innerHTML = reshtml + "</select><br>"
  return juizes;
}

function MostraJuiz(n, juizes) {
  var resultado = document.getElementById("resultDiv")
  var j = juizes[n].childNodes; // jid, nome, res, email
  reshtml = j[1].firstChild.nodeValue +"; " + j[2].firstChild.nodeValue +"; " + j[3].firstChild.nodeValue
  + "<br>";
  resultado.innerHTML += reshtml
}
```



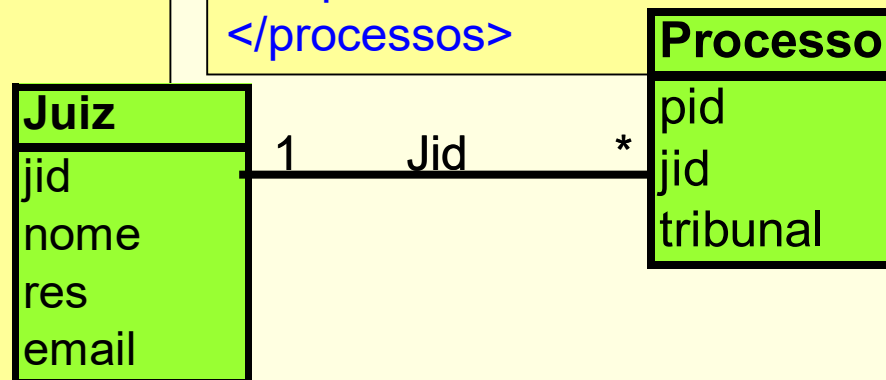
# Ajax –Exemplo 3 Juízes(1)

*listar juiz e respectivos processos* [http://localhost:8080/TPW/ajax-](http://localhost:8080/TPW/ajax-juizes3proc.html)

[juizes3proc.html](#) —

```
<juizes>
  <juiz alcunda="O justiceiro" datanasc="1-1-1947">
    <jid>1</jid>
    <nome> Jose Lima</nome>
    <res>Ovar</res>
    <email>zlima@trib.pt</email>
  </juiz>
  <juiz>
    <jid>2</jid>
    <nome> Julio Santos</nome>
    <res>Aveiro</res>
    <email>jsantos@trib2.pt</email>
  </juiz>
  <juiz>
    <jid>3</jid>
    <nome> Joana Lisboa</nome>
    <res> Aveiro</res>
    <email>jlisboa@trib2.pt</email>
  </juiz>
</juizes>
```

```
<processos>
  <processo>
    <pid>3</pid>
    <jid>1</jid>
    <tribunal>aveiro</tribunal>
  </processo>
  <processo>
    <pid>11</pid>
    <jid>3</jid>
    <tribunal>ovar</tribunal>
  </processo>
</processos>
```





# Ajax –Exemplo 3 Juízes(2)

*listar juiz e respectivos processos* <http://localhost:8080/TPW/ajax-juizes3proc.html> -

// **variáveis globais**

```
var juizes_db = new Array();  
var procs_db = new Array();  
var juizesxml=null;  
var procxml = null;
```

```
doAjaxCall("proctrib.xml");
```

```
while(procxml == null){procxml=juizesxml;}; // espera que doc xml carregue  
procs_db = procxml.getElementsByTagName("processo");
```

```
function doAjaxCall(the_request) {  
  var request = null;  
  request = new XMLHttpRequest();  
  if (request) {  
    request.open("GET", the_request);  
    request.onreadystatechange = function() {  
      if (request.readyState == 4) {  
        juizesxml = request.responseXML;  
      }  
    }  
    request.send(null);  
  } else { alert("deve actualizar o browser ...."); }  
}
```

JAM

```
<body>  
  <form>  
    Lista de Juizes:<input type="button" value="Fazer Pedido Ajax"  
    onclick="doAjaxCall('juizes.xml');alert('espera que carregue  
    ...');juizes_db=exemplo2(juizesxml);return true;">  
  </form>  
  JUIZ:<br>  
  <div id="resultDiv"></div>  
  <div id="resultDiv2"></div>  
</body>
```

**Em modo assíncrono, quando os pedidos estão relacionados é preciso esperar que os 2 (ou mais) pedidos estejam satisfeitos, de forma a continuar com a informação num estado consistente!**



# Ajax – Exemplo3 Juízes(3)

*listar juiz e respectivos processos*

```
function exemplo2(docxml) {
    var juizes = docxml.getElementsByTagName("juiz")
    var resultado = document.getElementById("resultDiv");
    var i;
    var reshtml='<select onmouseup="MostraJuiz(this.value, juizes_db, procs_db)">';
    for (i=0; i < juizes.length; i++){
        var juiz = juizes[i].childNodes;           // jid, nome, res, email,
        reshtml += '<option value =""+i+">'+juiz[1].firstChild.nodeValue+' </option>'
    }
    resultado.innerHTML = reshtml + "</select><br>"
    return juizes;
}

function MostraJuiz(n, juizes, procs) {
    var resultado = document.getElementById("resultDiv2");
    var j = juizes[n].childNodes;                 // jid, nome, res, email
    var i, processo = "<br>PROCESSOS:";
    for (i=0; i < procs.length; i++){             // procura processos com id deste juiz
        var p = procs[i].childNodes;              // pid, jid, tribunal
        if (j[0].firstChild.nodeValue == p[1].firstChild.nodeValue){
            processo += "<br>"+p[0].firstChild.nodeValue + " " + p[2].firstChild.nodeValue;
        }
    }
    reshtml = "<br>"+j[1].firstChild.nodeValue + "; " + j[2].firstChild.nodeValue + "; " + j[3].firstChild.nodeValue
    + processo + "<br>";
    resultado.innerHTML += reshtml
}
```



# Ajax –exemplo 4 -Juizes(1)

*listar juiz e respectivos processos–versão2 (c/ sincronismo)*

<http://localhost:8080/TPW/juizes3proc2melhor.html> -

```
<html>
<head>
<title>AJAX - teste</title>
<script type="text/javascript">
  var juizes = new Array();
  var procs = new Array();
```

```
function doAjaxCall1(the_request) {
  var request = null;
  request = new XMLHttpRequest();
  if (request) {
    request.open("GET", the_request, true);
    request.onreadystatechange = function () {
      if (request.readyState == 4) {
        var oxml = request.responseXML;
        doAjaxCall2("juizes.xml", oxml);
      }
    }
    request.send(null);
  } else {
    alert("actualizar o browser ....");
  }
}
```

```
function doAjaxCall2(the_request, oxml) {
  var request = null;
  request = new XMLHttpRequest();
  if (request) {
    request.open("GET", the_request, true )
    request.onreadystatechange = function () {
      if (request.readyState == 4) {
        var jxml = request.responseXML; alert(jxml);
        exemplo2(jxml,oxml);
      }
    }
    request.send(null);
  } else {
    alert("deve actualizar o browser ....");
  }
}
```

```
<body>
  <form>
    Lista de Juizes:<input type="button" value="Fazer Pedido Ajax"
    onclick="doAjaxCall1('proctrib.xml' );return true;">
  </form>
  JUIZ:<br>
  <div id="resultDiv"></div>
  <div id="resultDiv2"></div>
</body>
```



# Ajax – Exemplo4 Juízes(2)

*listar juiz e respectivos processos–versão2 (c/ sincronismo)*

```
function exemplo2(docxml,pxml) {  
    juizes = docxml.getElementsByTagName("juiz");  
    procs = pxml.getElementsByTagName("processo");  
    var resultado = document.getElementById("resultDiv");  
    var i;  
    var reshtml='<select onmouseup="MostraJuiz(this.value, juizes, procs)">';  
    for (i=0; i < juizes.length; i++){  
        var juiz = juizes[i].childNodes; // jid, nome, res, email,  
        reshtml += '<option value =""+i+">'+juiz[1].firstChild.nodeValue+' </option>'  
    }  
    resultado.innerHTML = reshtml + "</select><br>"  
    return;  
}
```

```
function MostraJuiz(n, juizes, procs) {  
    var resultado = document.getElementById("resultDiv2");  
    var j = juizes[n].childNodes; // jid, nome, res, email  
    var i, processo = "<br>PROCESSOS:";  
    for (i=0;i < procs.length; i++){ // procura processos com id deste juiz  
        var p = procs[i].childNodes; // pid, jid, tribunal  
        if (j[0].firstChild.nodeValue == p[1].firstChild.nodeValue){  
            processo += "<br>" + p[0].firstChild.nodeValue + " " + p[2].firstChild.nodeValue;  
        }  
    }  
    reshtml = "<br>" + j[1].firstChild.nodeValue + "; " + j[2].firstChild.nodeValue + "; " +  
    j[3].firstChild.nodeValue + processo + "<br>";  
    resultado.innerHTML += reshtml  
}  
</script>
```



# AJAX – Processamento do XML não linearizado

<http://localhost:8080/TPW/ajax-juizes1XMLlinear.html>

```
<juizes>
  <juiz alcunda="O justiceiro" datanasc="1-1-1947">
    <jid>1</jid>
    <nome>Jose Lima</nome>
    <res>Ovar</res>
    <email>zlima@trib.pt</email>
  </juiz>
  <juiz>
    <jid>2</jid>
    <nome>Julio Santos</nome>
    <res>Aveiro</res>
    <email>jsantos@trib2.pt</email>
  </juiz>
  .....
  <processo>
    <pid>2</pid>
    <jid>1</jid>
    <tribunal>aveiro</tribunal>
  </processo>
  <processo>
    <pid>3</pid>
    <jid>1</jid>
    <tribunal>aveiro</tribunal>
  </processo>
  .....
</juizes>
```

```
function exemplo1(docxml){
  var juizes = docxml.getElementsByTagName("juiz");
  var resultado = document.getElementById("resultDiv")
  var i,k;
  // Exemplo1- lista juizes, desde que os childnodes estejam certos
  // (XML linearizado)!
  var reshtml="JUIZES<br>versão1:<br>";
  for (i=0; i < juizes.length; i++){
    var juiz = juizes[i].childNodes; // jid, nome, res, email
    if (juiz.length == 4){
      reshtml += juiz[1].firstChild.nodeValue +"; "
                + juiz[2].firstChild.nodeValue + "<br>";
    }else {
      reshtml += "ERRO Parsing do XML<br>";
    }
  }
  resultado.innerHTML = reshtml+"<br>";
}
```

Usa tabela de juizes e processos no mesmo ficheiro XML





# AJAX – Processamento do XML não linearizado(2).

```
// Exemplo2 - cria array das tags a tratar (evita problemas de nós extra,  
//          mas não da falta da mesma tag ou tags a mais!  
var nomes = docxml.getElementsByTagName("nome");  
var ress = docxml.getElementsByTagName("res");  
reshtml = "versão 2:<br>";  
for (i=0; i < juizes.length; i++){  
    reshtml += nomes[i].firstChild.nodeValue + "; " + ress[i].firstChild.nodeValue+"<br>";  
}  
resultado.innerHTML += reshtml + "<br>";  
// Exemplo3- lista nós filhos de um processo, percorrendo a lista de nós selecionando só os válidos  
// Evita problema da introdução de nós extra (ou falta de nós) pelo parsing do XML!  
var procs = docxml.getElementsByTagName("processo");  
reshtml="PROCESSOS<br>";  
for (i=0; i < procs.length; i++){  
    var proc = procs[i].childNodes;    // pid, jid, tribunal  
    for (k=0;k<proc.length;k++){      // percorre os nós e ignora os inválidos. Adapta-se tb  
dinamicamente ao nº de nós  
        if (proc[k].tagName != undefined )reshtml += proc[k].firstChild.nodeValue + "; ";  
    }  
    reshtml += "<br>";  
}  
resultado.innerHTML += reshtml;  
}
```



# AJAX – Uma Pág, 2 exemplos assíncronos

*a mesma página é usada como base para 2 pedidos independentes*

*-- mostrando todo o potencial do ajax!*

```
<html>
```

```
<link rel="stylesheet" type="text/css" href="ajax1.css"/>
```

```
<script src="dicionario.js" type="text/javascript"></script>
```

```
<head>
```

```
<title>AJAX - teste</title>
```

```
<script type="text/javascript">
```

```
var juizes_db = new Array();
```

```
function doAjaxCall(the_request) {
```

```
.....  
</script>
```

```
</head>
```

```
<body>
```

```
<div class="j" >
```

```
<form>
```

```
Lista de Juizes:<input type="button" value="Pedido.." onclick="doAjaxCall('juizes.xml');return true;">
```

```
</form>
```

```
Resultado:<br>
```

```
<div class="juizes" id="resultDiv"></div>
```

```
</div>
```

```
<form class="dici" >
```

```
<input type = "text" size = "55" id = "theText" onKeyUp = "getTranslations('italian', this.value) ;">
```

```
<div id = "theResults" style = "width:22em; border:1px black solid;
```

```
padding-left:2px;padding-right:2px">
```

```
</div>
```

```
</form>
```

```
</body>
```

```
</html>
```

JAM

JavaScript do exemplo do dicionário

JavaScript do exemplo dos juizes – lista de processos

```
.dici {  
  color: red;  
  position: absolute;  
  top: 150px;  
  left: 400px;  
}  
.juizes {  
  background-color: yellow;  
  width: 300px;  
}  
.j {  
  background-color: aqua;  
  width: 350px;  
  border: solid red;  
}
```

<http://localhost:8080/TPW/ajax-2AP-juizesDic.html> -