

The word "AJAX" is written in a bold, orange, sans-serif font. It is positioned in the bottom right corner of a large, light gray rectangular area that occupies the top half of the slide. The letters have a slight 3D effect with a darker orange shadow.

Aula 01 - Introdução

AJAX é acrônimo em língua inglesa de "*Asynchronous Javascript and XML*", que em português significa "**Javascript e XML Assíncronos**"

Introdução

- O Ajax é baseado nos seguintes padrões de Web Standards:
 - Javascript
 - XML
 - HTML
 - CSS

Introdução

- O IE7, Firefox, Chrome, Safari e Opera implementaram a classe XMLHttpRequest que suporta os métodos e propriedades do Object ActiveX original da Microsoft.

XMLHTTP e XMLHttpRequest

- 1ª Instância multiplataforma para o XMLHttpRequest:

```
<script type="text/javascript">
if (window.XMLHttpRequest) { // IE7+, Mozilla, Safari, Chrome...
xmlhttp = new XMLHttpRequest();
} else if (window.ActiveXObject) { // IE5 ou IE6
xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
} else {
alert("Seu navegador não tem suporte para AJAX");
}
</script>
```

XMLHTTP e XMLHttpRequest

Algumas versões de navegador Mozilla necessitam desta linha:

```
xmlhttp.overrideMimeType('text/xml');
```

Próxima coisa a ser feita é decidir o que fazer após receber a resposta do servidor ao seu pedido:

```
xmlhttp.onreadystatechange = function(){  
  // código a executar aqui dentro, exemplo:  
  var valor_nome = document.getElementById("nome").value;  
  xmlhttp.open("GET", "localhost/t.php?nome="+valor_nome, true);  
  xmlhttp.send();  
};
```

open e send

Se for utilizar método POST você deve alterar o tipo MIME do pedido:

```
var nome = document.getElementById("nome").value;
var sobrenome = document.getElementById("sobrenome").value;
xmlhttp.open("POST", "localhost/PI/recebe.php", true);
xmlhttp.setRequestHeader('Content-Type','application/x-www-form-urlencoded'); /* linha para alterar o tipo MIME */
xmlhttp.send("nome="+valor_nome+"sobrenome="+sobrenome);
```

Se utilizar método POST

```
if(xmlhttp.readyState == 4) {  
    /* resposta recebida com sucesso */  
    if(xmlhttp.status == 200) {  
        // resposta ok!  
    } else {  
        // houve um problema com o pedido.  
    }  
} else {  
    /* ainda não está pronto, resposta ainda não foi recebida */  
}
```

Lista de valores do **readyState**:

- 0 – não iniciada;
- 1 – iniciando conexão;
- 2 – conexão estabelecida;
- 3 – em atividade (algum dado foi recebido);
- 4 – completa

Todos códigos de **status** estão listados na página W3C. 200 é igual a "OK".

readyState e status

- `xmlhttp.responseText` = irá devolver a resposta do servidor como uma linha de texto (string)
- `xmlhttp.responseXML` = irá devolver a resposta do servidor como um objeto `XMLDocument` que pode ser percorrido utilizando as funções DOM do JavaScript.

responseText e responseXML

- Recuperando uma informação com `responseText`:

```
<script type="text/javascript">  
document.getElementById("div_show").innerHTML = xmlhttp.responseText;  
</script>
```

responseText

- Documento XML?

```
<?xml version='1.0' encoding='UTF-8'?>
<catalogo>
  <cd>
    <titulo>Entrada para Raros</titulo>
    <artista>O Teatro Mágico</artista>
    <localidade>Osasco</localidade>
    <companhia>Independente</companhia>
    <preco>5.00</preco>
    <ano>2003</ano>
  </cd>
  <cd>
    <titulo>Grão do Corpo</titulo>
    <artista>O Teatro Mágico</artista>
    <localidade>Osasco</localidade>
    <companhia>Independente</companhia>
    <preco>20.00</preco>
    <ano>2014</ano>
  </cd>
</catalogo>
```

responseXML

- No ajax:

```
<script type="text/javascript">
documentoXML = xmlhttp.responseXML;
texto = "";
artistas = documentoXML.getElementsByTagName("ARTISTA");
for (i=0;i<artistas.length;i++)
{
    texto = texto + artistas[i].childNodes[0].nodeValue + "<br/>";
}
document.getElementById("div_show").innerHTML = texto;
</script>
```

responseXML

- Crie dois documentos XML a partir das seguintes tabelas:

IMÓVEIS

Tipo	Preço	Bairro
Casa	R\$ 234.000,00	Granja Olga
Casa	R\$ 126.000,00	Pq. Campolim
Apartamento	R\$ 93.000,00	Jd. Novo Eldorado

ALUNOS

Nome	Curso	Nota
Lucia	Word	9.5
Marcos	Excel	89
Luiz	Internet	100

Atividade – Documento XML

Método	Descrição
open("método", "URL", assíncrono)	Especifica o método, a URL e outros atributos opcionais de uma requisição. O parâmetro "método" pode ser GET, POST ou PUT. O parâmetro URL pode ser uma url relativa ou completa. O parâmetro assíncrono pode ser true (script continuará após a execução do método send) ou false (script ficará esperando a resposta do servidor no método send).
setRequestHeader("campo", "valor")	Adiciona um par de (nome_campo/valor_campo) ao cabeçalho http a ser enviado
send(postdata)	Executa a requisição HTTP
getAllResponseHeaders()	Retorna a coleção de cabeçalhos do http como uma String
getResponseHeader(label)	Retorna o cabeçalho HTTP associado ao label
abort()	Interrompe uma requisição que está sendo processada
send(conteudo)	Envia a requisição

Métodos do XMLHttpRequest

Propriedade	Descrição
onreadystatechange	Informa a função JavaScript que deve ser chamada a cada mudança de estado
readyState	Estado atual da requisição pode ser: 0 – não iniciada; 1 – iniciando conexão; 2 – conexão estabelecida; 3 – em atividade (algum dado foi recebido); 4 – completa
status	Retorna o estado HTTP da requisição como um número. Ex: 200 = OK; 403 = Forbidden; 404 = Not Found; 500 = Internal Error;
responseText	Retorna a resposta como uma String
responseXML	Retorna a resposta como um XML (que pode ser tratada como um W3C DOM)
statusText	Representação textual (string) do status HTTP recebido do servidor (OK, Forbidden, etc)

Propriedades do XMLHttpRequest

Próxima Aula

Aula 02 – Prática: Método GET