

Linguagem de Programação para Web

XMLHttpRequest

Prof. Tales Bitelo Viegas

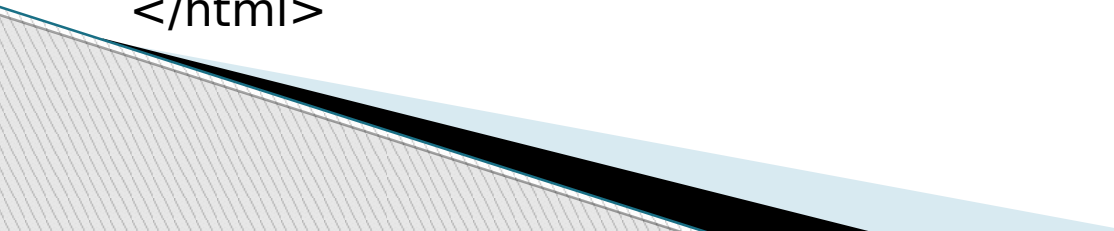
The bottom of the slide features a decorative graphic consisting of several overlapping, wavy lines in shades of gray and light blue, creating a modern, abstract background.

XMLHttpRequest

- ▶ Permite requisição a páginas/arquivos/scripts dinamicamente pelo cliente Web
- ▶ Criado pela Microsoft no IE4
 - No IE, é um objeto ActiveX
 - Nos demais é um objeto nativo

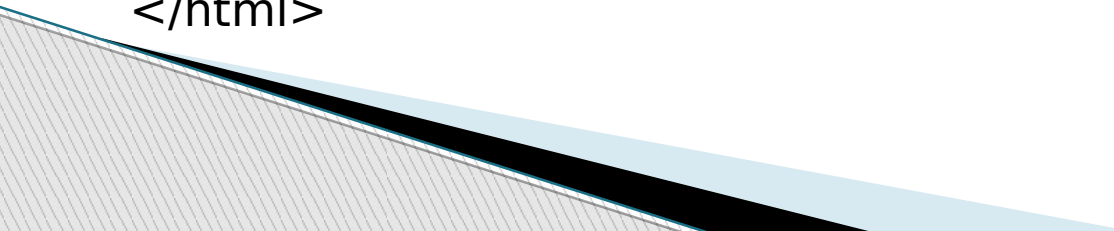
Exemplo

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
<meta charset="UTF-8"/>
<title>XMLHttpRequest</title>
</head>
<body>
<div id="carregando"></div>
<div id="conteudo">
<a href="#" onclick="alteraURL('pagina.html');">
Exemplo
</a>
</div>
</body>
</html>
```



Exemplo

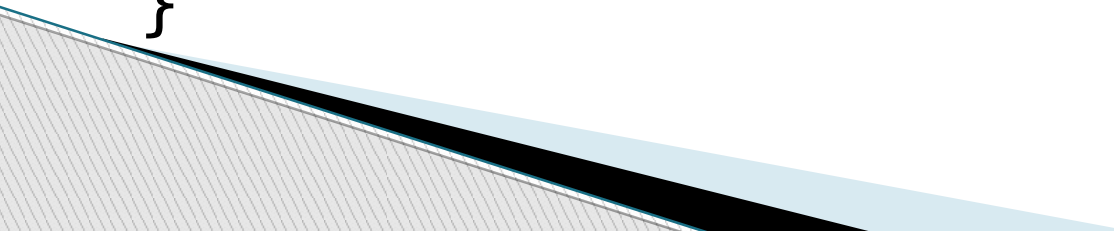
```
<!DOCTYPE html>
<html lang="pt-br">
<head>
<meta charset="UTF-8"/>
<title>XMLHttpRequest</title>
</head>
<body>
<div id="carregando"></div>
<div id="conteudo">
<a href="#" onclick="alteraURL('pagina.html');">
Exemplo
</a>
</div>
</body>
</html>
```



Exemplo

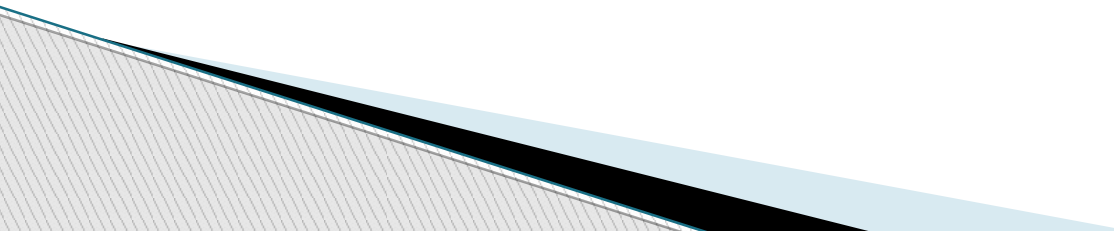
```
var xmlhttp = getXmlHttpRequest();
```

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



Exemplo

```
function alteraURL(url){
    document.getElementById("carregando").innerHTML =
        "<p>Carregando...</p>";
    xmlhttp.open("GET", url, true);
    xmlhttp.onreadystatechange = function(){
        if (xmlhttp.readyState == 4 ){
            document.getElementById("conteudo").innerHTML =
                xmlhttp.responseText;
            document.getElementById("carregando").innerHTML = "";
        }
    }
    xmlhttp.send();
}
```



Olhando com calma

- ▶ A função `getXmlHttpRequest` retorna um objeto `XMLHttpRequest`.
- ▶ Ela verifica se existe o objeto nativamente ou se o objeto deve ser instanciado via `ActiveXObject`, para compatibilidade com o Internet Explorer

Olhando com calma

- ▶ A função alteraURL primeiramente coloca um valor “Carregando”, para mostrar que está carregando um documento.
- ▶ Após, é utilizado o método open() do XMLHttpRequest, passando o método HTTP desejado, a URL a ser requisitada e se a chamada é assíncrona ou não.

Olhando com calma

- ▶ Como a chamada é assíncrona, é necessário utilizar uma função “callback” através do atributo `onreadystatechange`, que irá executar a função quando a requisição estiver completa.

Olhando com calma

- ▶ A variável `onreadystatechange` pode ter 5 valores:
 - 0 – Não enviado (UNSENT)
 - ▢ `open()` ainda não foi chamado
 - 1 – Aberto (OPENED)
 - ▢ `send()` ainda não foi chamado
 - 2 – Cabeçalhos recebidos (HEADERS_RECEIVED)
 - ▢ `send()` foi chamado e os cabeçalhos de resposta retornados
 - 3 – Carregando (LOADING)
 - ▢ Fazendo o download, `responseText` com resposta parcial
 - 4 – Concluído (DONE)
 - ▢ Operação completa, `responseText` com resposta total

Alguns atributos do objeto

- ▶ `responseType`
 - O tipo da resposta desejada
 - ▢ `json`, `text`, `document`, `xml`
- ▶ `response`
 - O corpo da resposta, de acordo com o `responseType`
- ▶ `responseText`
 - O corpo da resposta em modo texto
- ▶ `responseXML`
 - O corpo da resposta em modo XML

Alguns atributos do objeto

- ▶ status
 - Código HTTP da resposta
- ▶ timeout
 - Tempo (em ms) para esperar a resposta

Alguns métodos do objeto

- ▶ `getAllResponseHeaders()`
 - Retorna todos os cabeçalhos HTTP da resposta
- ▶ `setRequestHeader(header, value)`
 - Define um cabeçalho de requisição

Exercício

- ▶ Escreva um HTML onde:
 - O lado esquerdo mostre uma lista de 10 nomes de clientes
 - O lado direito mostre um formulário para edição, com os campos nome, email e telefone
 - Ao clicar no nome do cliente, as informações devem ser exibidas
 - As informações dos clientes devem estar em um arquivo XML
 - As atualizações devem ser feitas via AJAX

Exercício

- ▶ Altere o exercício anterior para que seja possível adicionar novos clientes na lista
- ▶ Ponto Extra: altere para que a adição na lista seja permanente, ou seja, ao recarregar a página o cliente mantenha-se cadastrado
 - Utilize apenas front-end
 - Verifique o funcionamento do Local Storage
 - ▮ <http://zenorocha.com/html5-local-storage/>