



PROGRAMAÇÃO PARA INTERNET RICA

AJAX

Prof. Dr. Daniel Caetano

2013 - 1

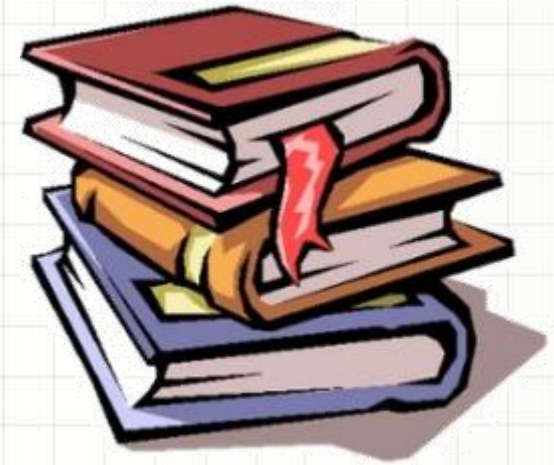
Objetivos

- Apresentar o conceito do AJAX
- Capacitar para a criação de sites / aplicativos dinâmicos com o uso de AJAX

- **Trabalho B!**



Material de Estudo



Material

Acesso ao Material

Apresentação

<http://www.caetano.eng.br/>

(Aula 11)

Material Didático

Ajax, Rich Internet Applications e Desenvolvimento Web para Programadores, páginas 312 a 341

Google

+AJAX



O QUE É AJAX?

Introdução

- Páginas Web Tradicionais
 - Requisições: links e forms
 - Respostas: uma nova página completa
- AJAX
 - Requisições: geradas no JavaScript
 - Respostas:
 - um trecho de página
 - um documento XML

Introdução

- AJAX significa
 - Asynchronous
 - JavaScript
 - And
 - XML



- É, basicamente, JavaScript com o uso do DOM... só que de maneira assíncrona
- O que é **assíncrono**?


Introdução

- Síncrono

x

Assíncrono





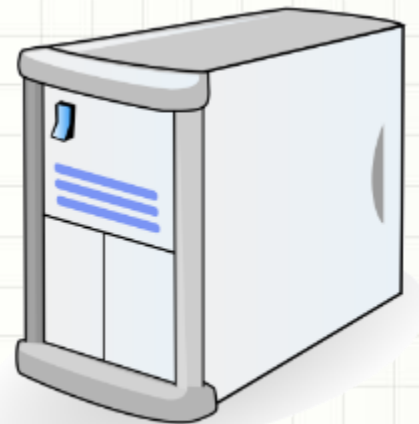
LÓGICA DE FUNCIONAMENTO DO AJAX

Lógica do AJAX

- Usualmente, quando uma requisição é enviada ao servidor:
 - 1) O navegador “trava” até a resposta chegar
 - 2) A resposta substitui a página atual



Cliente



Servidor

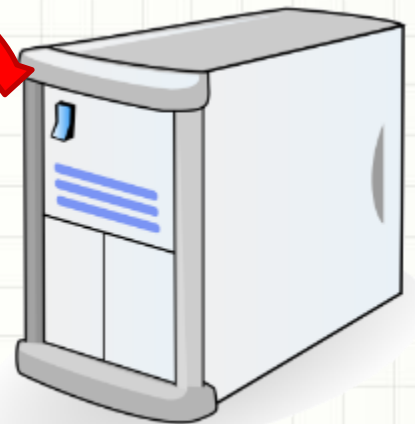
Lógica do AJAX

- Usualmente, quando uma requisição é enviada ao servidor:
 - 1) O navegador “trava” até a resposta chegar
 - 2) A resposta substitui a página atual



Cliente

REQUEST



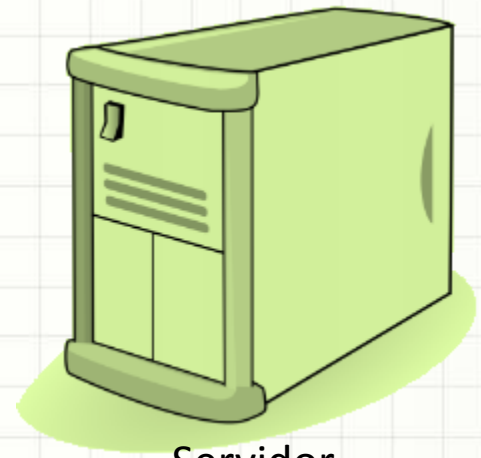
Servidor

Lógica do AJAX

- Usualmente, quando uma requisição é enviada ao servidor:
 - 1) O navegador “trava” até a resposta chegar
 - 2) A resposta substitui a página atual



Cliente



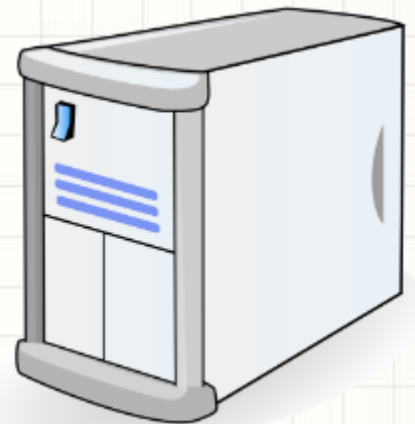
Servidor

Lógica do AJAX

- Usualmente, quando uma requisição é enviada ao servidor:
 - 1) O navegador “trava” até a resposta chegar
 - 2) A resposta substitui a página atual



Cliente



Servidor

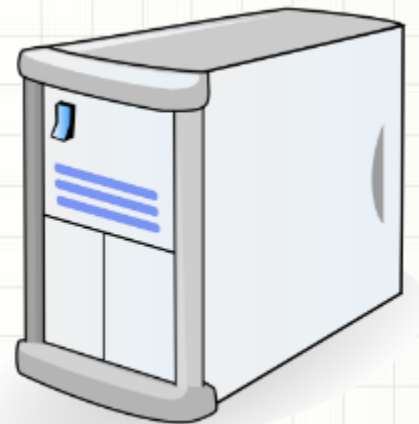
RESPONSE

Lógica do AJAX

- Usualmente, quando uma requisição é enviada ao servidor:
 - 1) O navegador “trava” até a resposta chegar
 - 2) A resposta substitui a página atual



Cliente



Servidor

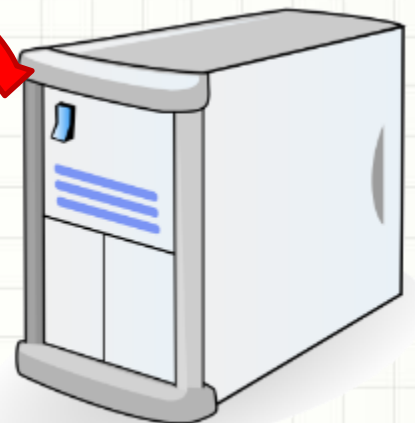
Lógica do AJAX

- No AJAX:
 - 1) O navegador continua funcionando
 - 2) O programador é que terá de dizer o que será feito com a resposta



Cliente

REQUEST



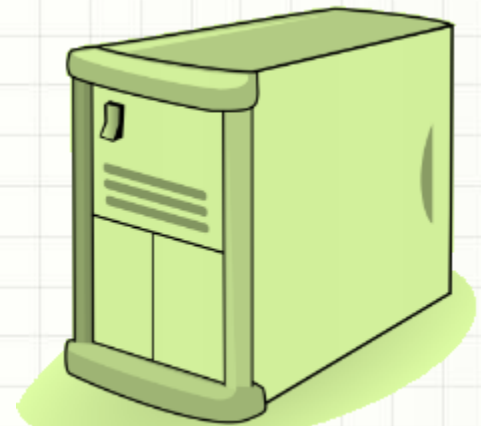
Servidor

Lógica do AJAX

- No AJAX:
 - 1) O navegador continua funcionando
 - 2) O programador é que terá de dizer o que será feito com a resposta



Cliente



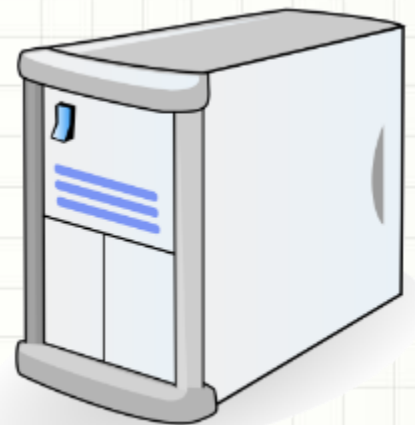
Servidor

Lógica do AJAX

- No AJAX:
 - 1) O navegador continua funcionando
 - 2) O programador é que terá de dizer o que será feito com a resposta



Cliente



Servidor

RESPONSE

Lógica de Funcionamento

- Criar uma requisição
- Configurar a Requisição
 - Função “callback”
 - Função que é executada quando a resposta chega
- Enviar a Requisição
- ...
- Ao chegar a resposta, função *CallBack* acionada
 - Função **callback**: coloca resposta na página
 - Por exemplo, usando innerHTML



CÓDIGO DE CHAMADA

Criando uma Requisição

- Não é padrão... então, criamos uma função!

```
function createRequest() {
    try {
        request = new XMLHttpRequest();
    }
    catch (tryMS) {
        try {
            request = new ActiveXObject("Msxml2.XMLHTTP");
        }
        catch (otherMS) {
            try {
                request = new ActiveXObject("Microsoft.XMLHTTP");
            }
            catch (failed) {
                request = null;
            }
        }
    }
    return request;
}
```

Criando uma Requisição

- Gerando uma Request

```
request = createRequest();  
if (request == null) {  
    window.alert("Impossível criar requisição AJAX");  
    return;  
}  
var url="http://www.caetano.eng.br/xml/xml.php";  
request.open("GET", url, true);  
request.onreadystatechange = processResponse;  
request.send(null);
```

- Endereço da requisição (mesmo servidor!)
- **request.open(*tipo, url, assinc*)**
- Função Callback



FUNÇÃO CALLBACK

Tratando uma Resposta

- Recebendo uma Resposta

```
function processResponse() {  
    if (request.readyState == 4) {  
        if (request.status == 200) {  
            pintaTabuleiro(request.responseText);  
        }  
    }  
}
```

- Verificar se resposta já chegou correta...
- E realizar algum trabalho com a resposta

Tratando uma Resposta

- Recebendo uma Resposta

```
function processResponse() {  
    if (request.readyState == 4) {  
        if (request.status == 200) {  
            pintaTabuleiro(request.responseText);  
        }  
    }  
}
```

- 0: Requisição não iniciada (*antes de chamar o open()*).
- 1: Requisição configurada, mas não enviada (*antes de chamar o send()*).
- 2: Requisição enviada e sendo processada (*é possível obter o cabeçalho da resposta*).
- 3: Requisição sendo processada (*é possível obter dados parciais, mas o servidor ainda não finalizou a resposta*).
- 4: Resposta completa (*todos os dados da resposta estão disponíveis*).

Tratando uma Resposta

- Recebendo uma Resposta

```
function processResponse() {  
    if (request.readyState == 4) {  
        if (request.status == 200) {  
            pintaTabuleiro(request.responseText);  
        }  
    }  
}
```

1xx: Informações.

2xx: Sucesso (em especial, 200).

3xx: Página movida para outro endereço.

4xx: Erro no cliente.

5xx: Erro no servidor.

<http://www.iana.org/assignments/http-status-codes/http-status-codes.xml>

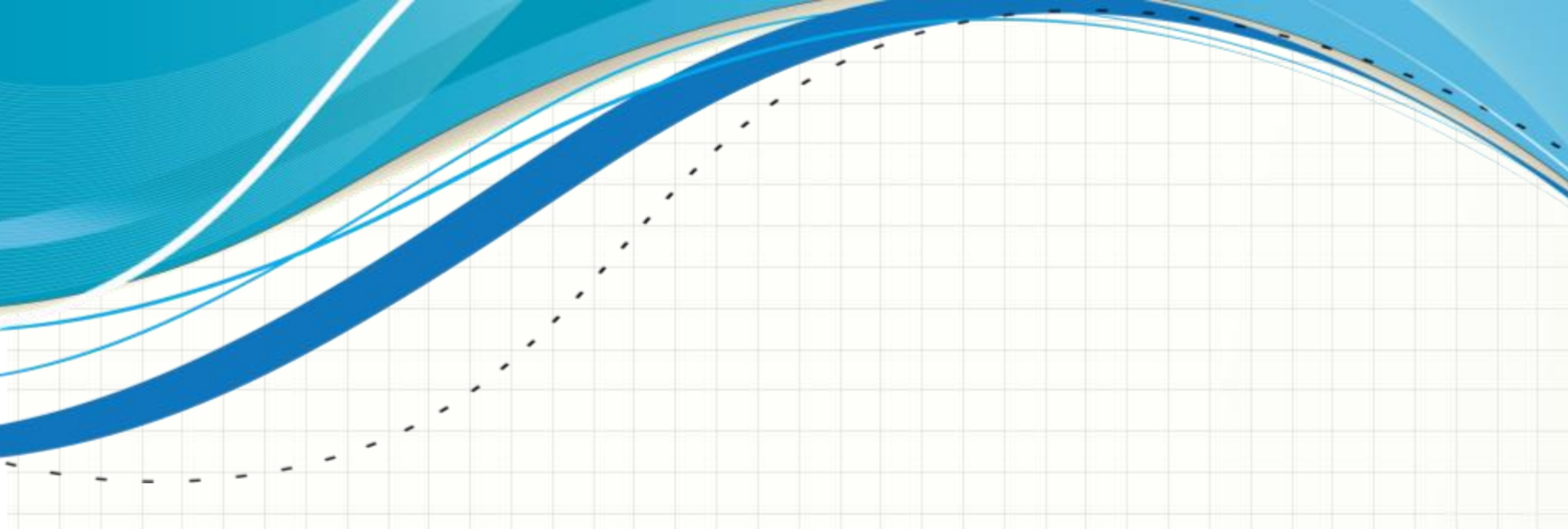


RESPOSTAS TEXTO NA PRÁTICA

Na Prática

- Acompanhe o professor!
- Testando o servidor Apache
- Acrescentando AJAX no jogo da velha

ATENÇÃO! AJAX NÃO FUNCIONA “OFFLINE”



TRATANDO RESPOSTAS XML

Tratando uma Resposta XML

- Respostas em XML

O documento XML não está associado a estilos. A estrutura do documento é representada abaixo.

```
-<biblioteca>
  -<livro id="1">
    <nome>Duna</nome>
    <paginas>652</paginas>
    <autor>Frank Herbert</autor>
  </livro>
  -<livro id="2">
    <nome>Senhor dos Anéis</nome>
    <paginas>978</paginas>
    <autor>J.R.R. Tolkien</autor>
  </livro>
</biblioteca>
```

Tratando uma Resposta XML

- Respostas em XML
- `<livro id="xx"><nome>ABC</nome></livro>`

```
function processResponse() {
  if (request.readyState == 4) {
    if (request.status == 200) {
      var doc = request.responseXML.documentElement;
      var livro = doc.getElementsByTagName("livro")[0];

      id = livro.getAttribute("id");
      nomeTag = livro.getElementsByTagName("nome")[0];
      nome = nomeTag.firstChild.nodeValue;

      var par = document.getElementById("par");
      par.innerHTML = "ID: " + id + " Nome: " + nome;
    }
  }
}
```

Tratando uma Resposta XML

- Respostas em XML – Usando jQuery
- <livro id="xx"><nome>ABC</nome></livro>

```
window.onload = init;
function init() {
    $("#b1").click(function() {
        $.get( "/xml.php", processResponse, "xml");
    });
}

function processResponse(data) {
    var texto = "";
    $(data).find("livro").each(function() {
        var id = $(this).attr("id");
        var nome = $(this).find("nome").text();

        var par = document.getElementById("par");
        par.innerHTML = "ID: " + id + " - Nome: " + nome;

    });
}
```



NA PRÁTICA

Na Prática

- Acompanhe o professor!
- Criando AJAX com XML
- Usando AJAX com XML pelo jQuery

ATENÇÃO! AJAX NÃO FUNCIONA “OFFLINE”



CONCLUSÕES

Resumo

- AJAX: requisições assíncronas
- Apesar de ser um recurso simples, ele permite grande flexibilidade e resultados muito interessantes
- A biblioteca jQuery facilita o uso do AJAX
- **TAREFA**
 - Trabalho B!



PERGUNTAS?

Próxima Aula



- AJAX é legal...
 - Mas não tem um jeito mais simples de fazer “coisas animadas”?
 - Tem... Action Script!
 - Flash!



**BOM DESCANSO
A TODOS!**