

# AJAX - Tutorial

Anselmo Luiz Éden Battisti, Christiano Julio Pilger de Brito

18 de agosto de 2006

# Agenda

1. Introdução
2. Sincrono X Assíncrono
3. O que é Ajax?
4. Para que Serve?
5. Exemplo 1 - Carregando um Arquivo
6. ajaxInit
7. Métodos e Atributos
8. Exibindo o Resultado
9. Exemplo 2 - Calculadora
10. Exemplo 3a - Alimentação de Select
11. Json
12. Exemplo 3b - Alimentação de Select com Json
13. Sugestão de Conteúdo
14. Conclusão
15. Referências

# Síncrono X Assíncrono!

## ▶ Síncrono

- ▶ Faz uma solicitação ao servidor;
- ▶ O usuário fica esperando a resposta do servidor;
- ▶ Quando vem a resposta toda a página é recarregada e o conteúdo exibido.

## ▶ Assíncrono

- ▶ Faz uma solicitação ao servidor de forma que o usuário do sistema não tome consciência disto;
- ▶ O usuário continua trabalhando com o sistema;
- ▶ Após o recebimento do conteúdo usa-se *Javascript* para processar e exibir o resultado na tela.

# O que é AJAX

- ▶ AJAX (acrónimo em língua inglesa de *Asynchronous Javascript And XML*);
- ▶ Um conjunto de Tecnologia
  - ▶ Objeto capaz de fazer requisições assíncronas;
  - ▶ HTML;
  - ▶ *JavaScript*;
  - ▶ XML *eXtensible Markup Language*;
  - ▶ *Json JavaScript Object Notation*;
  - ▶ DHTML;
  - ▶ CSS.
- ▶ AJAX não é somente um modelo, é também uma iniciativa na construção de aplicações web mais dinâmicas.

# Para que Serve?

- ▶ Criar WEB sites mais dinâmicos;
- ▶ Acelerar o acesso a informação;
- ▶ Diminuir o tráfego de informações entre cliente X servidor
- ▶ Produzir aplicações WEB mais semelhantes as Desktop.

# Tradicional X AJAX

- ▶ <http://adaptivepath.com/images/publications/essays/ajax-fig1.png>
- ▶ <http://adaptivepath.com/images/publications/essays/ajax-fig2.png>

## Código Fonte 1: 'ajaxInit.js'

---

```
1 function ajaxInit() {
2
3     var xmlhttp;
4
5     try {
6         xmlhttp = new XMLHttpRequest();
7     } catch (ee) {
8         try {
9             xmlhttp = new ActiveXObject("Msxml2.XMLHTTP");
10        } catch (e) {
11            try {
12                xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
13            } catch (E) {
14                xmlhttp = false;
15            }
16        }
17    }
18
19    return xmlhttp;
20 }
```

---

# Exemplo 1

- ▶ Carrega um arquivo chamado **texto.txt**
- ▶ Insere seu conteúdo no DIV cujo id é texto



## Código Fonte 2: 'Função do Exemplo 1'

---

```
1 function carregar() {
2     ajax = ajaxInit();
3     document.getElementById("texto").innerHTML = "
4         Carregando ...";
5     if (ajax) {
6         ajax.open('GET', 'http://200.201.81.38/anselmo/
7             ciclo/texto.txt', true);
8         ajax.onreadystatechange = function() {
9             if (ajax.readyState == 4) {
10                if (ajax.status == 200){
11                    document.getElementById("texto").
12                        innerHTML = ajax.responseText;
13                }
14            }
15        }
16    }
17    ajax.send(null);
18 }
```

---

# Métodos e Atributos

## ▶ Métodos

- ▶ **open(mode, url, boolean)**, mode é POST ou GET, url é o endereço, pode ser relativo, o último parâmetro é *TRUE* para conexão assíncrona e *FALSE* para conexão síncrona.
- ▶ **send()** É o método SEND que ativa a conexão e faz a requisição de informações ao documento aberto pelo método OPEN;

## ▶ Atributos

- ▶ **status** : Status do retorno do html, são os códigos padrões do html 200 ok, 400 no found;
- ▶ **responseText** : Retorna a cadeia de caracteres que o servidor enviou;
- ▶ **onreadystatechange** : Define qual função será chamada para fazer a manipulação dos dados assim que houver um retorno;
- ▶ **readyState** : Código que diz o status da solicitação
  - ▶ 0 (uninitialized);
  - ▶ 1 (a carregar);
  - ▶ 2 (carregado);
  - ▶ 3 (interactivo);
  - ▶ 4 (completo).

# Exibindo os Resultados

- ▶ Para exibir o resultado da consulta pode-se usar dois métodos o `innerHTML` ou o `DOM`;
- ▶ `innerHTML` simplesmente interpreta o texto que vem do servidor como sendo HTML e joga dentro do objeto de destino;
- ▶ `DOM` (Modelo de Objeto de Documentos) É a definição da estrutura lógica dos documentos e o meio pelo qual um documento é acessado e manipulado

# Calculadora

- ▶ Nossa calculadora será composta de 2 caixas de texto para a entrada dos valores e uma label para exibição dos resultados, a única operação que ela realiza é a soma;
- ▶ O arquivo PHP que realiza o cálculo da soma é realmente simples, ele pega as duas variáveis **v1** e **v2** do *array* `$_GET` do PHP, soma seus valores e escreve o resultado;
- ▶ <http://200.201.81.38/anselmo/ciclo/exemplo2.html>.

## Código Fonte 3: 'Função do Exemplo 2'

```
1 function calcular() {
2     ajax = ajaxInit();
3     document.getElementById("resultado").innerHTML = "
4         Calculando ...";
5     if (ajax) {
6         ajax.onreadystatechange = resultado;
7         v1 = document.getElementById("v1").value;
8         v2 = document.getElementById("v2").value;
9         url = 'http://200.201.81.38/anselmo/ciclo/
10             calcular.php?v1='+v1+'&v2='+v2;
11         ajax.open('GET', url, true);
12         ajax.send(null);
13     }
14 }
15
16 function resultado() {
17     if (ajax.readyState == 4) {
18         if (ajax.status == 200){
19             document.getElementById("resultado").
20                 innerHTML = ajax.responseText;
21         }
22     }
23 }
```

## Código Fonte 4: 'Calculadora PHP'

---

```
1 <?  
2 echo $_GET['v1'] + $_GET['v2'];  
3 ?>
```

---

## Exemplo 3a - Alimentação de Select

- ▶ Um exemplo clássico do preenchimento de um *SELECT* a partir de outro é a escolha das cidades de um determinado estado. Este recurso é uma das maiores 'pedras no sapato' dos programadores de sistema WEB, pois quando o usuário seleciona o estado toda a página deve ser recarregada, inclusive os valores dos demais campos já preenchidos, para que o *SELECT* com as cidades seja exibido.

## Código Fonte 5: 'Cidades PHP'

---

```
1 <?
2 header("Content-Type: text/html; charset=iso-8859-1");
3
4 $estado = $_GET['estado'];
5 $cidades = "";
6 if ($estado=="pr"){
7     $cidades .= "<option value=\"1\">Tupãssi</option>";
8     $cidades .= "<option value=\"2\">Toledo</option>";
9     $cidades .= "<option value=\"3\">Cascavel</option>";
10    $cidades .= "<option value=\"4\">Pato Branco</option>";
11 }
12
13 if ($estado=="sp"){
14     $cidades .= "<option value=\"5\">Mogi</option>";
15     $cidades .= "<option value=\"6\">Palmeiras</option>";
16     $cidades .= "<option value=\"7\">Santos</option>";
17 }
18
19 echo $cidades;
20 ?>
```

---



## Código Fonte 6: 'Função do Exemplo 3'

```
1 function selecionaCidade(estado) {
2     ajax = ajaxInit();
3     if (ajax) {
4         ajax.onreadystatechange = escreveCidades;
5         url = 'http://200.201.81.38/anselmo/ciclo/
6             cidades.php?estado='+estado;
7         ajax.open('GET',url,true);
8         ajax.send(null);
9     }
10
11 function escreveCidades() {
12     if (ajax.readyState == 4) {
13         if(ajax.status == 200){
14             document.getElementById("cidade").innerHTML =
15                 ajax.responseText;
16         }
17     }
18
19 <select name="estado" id="estado" onChange="if (this.options
20     [this.selectedIndex].value){selecionaCidade(this.options
21     [this.selectedIndex].value)}">
```

# Json

- ▶ JSON *JavaScript Object Notation*
- ▶ <http://www.json.org/>

## Código Fonte 7: 'Cidades PHP'

---

```
1 <?
2 header("Content-Type: text/html; charset=iso-8859-1");
3
4 $estado = $_GET['estado'];
5 $cidades = "{ 'cidades':[";
6 if ($estado=="pr"){
7     $cidades .= "{ 'cdg':'1', 'nm':'Tupãssi' },
8                 { 'cdg':'2', 'nm':'Toledo' },
9                 { 'cdg':'3', 'nm':'Cascavel' },
10                { 'cdg':'4', 'nm':'Pato Branco' } ]}";
11 }
12
13 if ($estado=="sp"){
14     $cidades .= "{ 'cdg':'5', 'nm':'Mogi' },
15                 { 'cdg':'6', 'nm':'Palmeiras' },
16                 { 'cdg':'7', 'nm':'Santos' } ]}";
17 }
18
19 echo $cidades;
20 ?>
```

---

## Código Fonte 8: 'Função que Chama as Cidades'

---

```
1 function selecionaCidadeJson(estado) {
2     ajax = ajaxInit();
3     if (ajax) {
4         ajax.onreadystatechange = escreveCidadesJson;
5         url = 'http://200.201.81.38/anselmo/ciclo/
           cidadesJson.php?estado='+estado;
6         ajax.open('GET',url,true);
7         ajax.send(null);
8     }
9 }
```

---

## Código Fonte 9: 'Função que Exibe o Resultado'

---

```
1 function escreveCidadesJson() {
2     if (ajax.readyState == 4) {
3         if(ajax.status == 200){
4             var x = 0;
5             var json = eval("(" + ajax.responseText + ")");
6             limparDestino("cidade2");
7             for(x=0;x<=json.cidades.length;x++){
8                 option = document.createElement("option");
9                 option.setAttribute("value", json.cidades[x].
10                     cdg);
11                 option.appendChild(document.createTextNode(
12                     json.cidades[x].nm));
13                 document.getElementById(document.
14                     createTextNode(json.cidades[x].nm));
15                 document.getElementById("cidade2").
16                     appendChild(option);
17             }
18         }
19     }
20 }
```

---

## Código Fonte 10: 'Função que Remove Tods os elementos'

---

```
1  /*
2  Remove todos os elementos filhos de um elemento
3  */
4  function limparDestino(destino) {
5      obj = document.getElementById(destino);
6      while(obj.firstChild)
7          obj.removeChild(obj.firstChild);
8  }
```

---

## Código Fonte 11: 'Função para conectar ao banco de dados'

---

```
1 <?php
2 # FileName="Connection_php_mysql.htm"
3 # Type="MYSQL"
4 # HTTP="true"
5 $hostname_conexao = "localhost";
6 $database_conexao = "test";
7 $username_conexao = "aluno";
8 $password_conexao = "aluno";
9 $conexao = mysql_pconnect($hostname_conexao ,
    $username_conexao , $password_conexao) or die(mysql_error
    ());
10 mysql_select_db($database_conexao , $conexao);
11 ?>
```

---

## Código Fonte 12: 'Auto Completar Usando Banco de Dados'

---

```
1 <?
2     include("conexao.php");
3
4     $search = $_GET['s'];
5
6     $sql= "SELECT * FROM municipio WHERE mnc_descricao like '
7           ".$search."%'";
8
9     $consulta = mysql_query($sql,$conexao);
10    $linhas = mysql_fetch_array($consulta);
11
12    $res = "{ 'pal':[";
13    do {
14        $res .= "'".str_replace("'", " ", $linhas[ '
15            mnc_descricao' ])."',";
16    } while ($linhas = mysql_fetch_array($consulta));
17
18    $res = substr($res,0,(strlen($res)-1));
19    $res .= "]}";
20    echo $res;
21 ?>
```

---



# Vantagens

- ▶ Como a modificação das informações da tela são parciais uma grande quantidade de informações deixa de trafegar inutilmente pela rede;
- ▶ O servidor que roda a aplicação fica menor carregado pois existe uma divisão de tarefas com o cliente;
- ▶ AJAX não é uma tecnologia por isto não é necessário pagar para a utilizar.

# Problemas

- ▶ AJAX não é a solução milagrosa para todos os males e se usado de forma inadequada pode piorar uma situação que já era feia;
- ▶ Os Navegadores usam diferentes métodos por isso temos que estar atentos
- ▶ Os Navegador em geral não suporta uma carga muito pesada de *scripts*;
- ▶ Os botão voltar, avança e histórico não funcionam muito bem com AJAX.

# Referências

- ▶ Artigo muito bom sobre AJAX  
<http://adaptivepath.com/publications/essays/archives/000385.php>
- ▶ Não use innerHTML  
[http://slayeroffice.com/articles/innerHTML\\_alternatives/#intro](http://slayeroffice.com/articles/innerHTML_alternatives/#intro)
- ▶ Página oficial do Json  
<http://www.json.org/>
- ▶ DHTML  
[http://www.quirksmode.org/js/cross\\_dhtml.html](http://www.quirksmode.org/js/cross_dhtml.html)
- ▶ Material sobre Css  
<http://www.maujor.com>
- ▶ DOM XHTML e HTML  
<http://www.amtechs.com/w3c/introduction.html>