AJAX em 30 segundos

Domingo, 01 de Janeiro de 2006
Tags: ajax, javascript, php, tutorial

Não, você não leu errado. São 30 segundos mesmo para entender o código e começar a pegar o jeito do tal do AJAX, segundo o blog lunatechian.

Sigla de **Asynchronous JavaScript and XML**, AJAX é uma técnica antiga e era pouco usada, serve para sincronizar as informações do servidor com a página aberta no computador do cliente.

Navegando pelo diggdot.us, esse link me chamou muita atenção com seu título "Learn AJAX in 30 seconds". Baseado nesse artigo, eu resolvi fazer uma versão brasileira para o tutorial, com um código um pouco mais complexo e com uns erros corrigidos.

Primeiramente vamos ao script. Esse é o "motor" do AJAX, o javascript é o código executado no computador do usuário. Considerando que o usuário pode estar usando tanto o Internet Explorer, quanto um browser, e nós queremos que o código funcione em ambos os casos, a primeira função que o cara escreveu no javascript identifica o navegador usado e define o código apropriado para ele. As outras duas funções servem para enviar o *request* para a base de dados (que na verdade é um arquivo PHP) e exibir a resposta gerada pelo programa, que nós vamos ver mais a frente.

Esse é o código do javascript:

```
function createXMLHttpRequest() {
var bc;
if(window.XMLHttpRequest) {
try {
bc = new XMLHttpRequest();
} catch(e) {
bc = false;
}
} else if(window.ActiveXObject) {
try {
bc = new ActiveXObject("Microsoft.XMLHTTP");
} catch(e) {
bc = false;
}
}
return bc;
}
var req = createXMLHttpRequest();
function sendRequest(id,item) {
req.open('get', 'base.php?id=' + id +'&item=' + item);
req.onreadystatechange = handleResponse;
req.send(null);
}
function handleResponse() {
if(req.readyState == 4){
var response = req.responseText;
var update = new Array();
if(response.indexOf('|' != -1)) {
update = response.split('|');
```

```
document.getElementById(update[0]).innerHTML = update[1];
}
}
}
```

Na última parte do código o javascript atualiza um elemento com id igual a “ajaxbox”. Para não ocorrer um erro, é preciso que exista esse elemento na mesma página que contém o script. Eu prefiro usar uma div com o código * *, mas é possível usar outros elementos, como uma lista (*
<li id=”ajaxbox” >
*) ou um *span* (**).

Depois disso, é preciso ter uma forma de o usuário ativar o script. Isso pode ser feito com um formulário, por exemplo. Mas no nosso caso será usado um link.

```
<a href="javascript:sendRequest('ajaxbox', '1');">Exibir o ítem
1</a><br/>
<a href="javascript:sendRequest('ajaxbox', '2');">Exibir o ítem 2</a>
```

1 e 2 são os números dos ítens a serem exibidos, isso ficará mais claro no próximo código.

Por fim, temos o arquivo base.php. Nesse arquivo é que ficam armazenadas as informações que serão exibidas na página quando o internauta executar o javascript acima. O programa será rodado no servidor e seu resultado será a informação pedida pelo usuário, que voltará ao javascript para ser exibida pelo processo já explicado acima

Código do arquivo base.php:

```
<?php
$id = $_REQUEST['id'];
$frase[1] = "Texto qualquer da primeira frase";
$frase[2] = "Texto qualquer da segunda frase";
$exfrase = $frase[$_GET['item']];
echo "$id|$exfrase";
?>
```

Tudo certo? Ok. Por hoje é só isso.