pra comeco de historia, o Ajax na verdade, eh um treco do JavaScript, o qual permite puxar dados de arquivos, e permite tambem, enviar dados pra esses arquivos (nossos queridos GET e POST), pra fazer uma cosulta ao arquivo (parece o LoadVars do Flash)

quem quizer aprender mesmo isso, alem de conhecimento PHP, eh bom saber um Javascript legal, pois ele é o mais usado nesse caso

ao codigo!

precisamos primeiro inicializar o Ajax, seria muito facil, se todo mundo usasse o mesmo browser... mas como isso não é real, entao vamos criar uma funcao que retornao Ajax do browser em uso:

ajaxInit.js

CODE

```
function ajaxInit() {
      var req;

       try {
req = new ActiveXObject("Microsoft.XMLHTTP");
      } catch(e) {
       try {
  req = new ActiveXObject("Msxml2.XMLHTTP");
      } catch(ex) {
        try {
    req = new XMLHttpRequest();
      } catch(exc) {
alert("Esse browser não tem recursos para uso do Ajax");
      req = null;
       }
      }
     }

    return req;
     }
```

oque esse script faz é ir tentando carregar o Ajax até conseguir, usando try / catch

agora ja temos a funcao de inicializacao (oia q legal!)

agora vamos criar um script PHP para ser usado nesse exemplo
obs: esse script poderia ser feito apenas em javascript, mas a intencao é apenas demonstrar o uso do Ajax

scriptAjax.php

PHP

```
1.  <?php
2.  $gmtDate = gmdate("D, d M Y H:i:s");
```

```
3.
4.    header("Expires: {$gmtDate} GMT");
5.    header("Last-Modified: {$gmtDate} GMT");
6.    header("Cache-Control: no-cache, must-revalidate");
7.    header("Pragma: no-cache");
8.
9.    //os readers acima serao explicados apos o script
10.
11.  $n = $_GET["n"]; //pegar a variavei enviada
12.
13.  //vamos multiplicar essa variavel por 50
14.  $n *= 50;
15.
16.  echo $n; //agora vamos "retornar" o valor, para isso escrevemos ele
17.
18.  ?>
```

como vimos, o script recebe um numero, e multiplica ele por 50, e depois retorna o novo valor, para um uso pratico, poderiamos puxar dados no banco de dados e retornar, oq seria um uso real disso

os headers iniciais, em base querem dizer: "NAO USE O CACHE DO BROWSER!!!"
ou seja, sevem pra vc n correr o risco de tentar baixar coisas do cache, oq seria inutil para uma linguagem server-side

agora vamos criar o script final, que usa nossos scripts pre-criados

calcula.php

CODE

```
<html>
<head>
<title> Uso do Ajax </title>
<script src="ajaxInit.js" language="javascript" type="text/javascript"></script>
<script>

function calcula() {
valIni = document.getElementById("valorIni").value;
ajax = ajaxInit();
if(ajax) {
ajax.open("GET", "scriptAjax.php?n=" + valIni, true);
ajax.onreadystatechange = function() {
if(ajax.readyState == 4) {
if(ajax.status == 200) {
document.getElementById("resultado").value = ajax.responseText;
} else {
alert(ajax.statusText);
}
}
}
ajax.send(null);
```

```
                                        }
                                        }

                                    </script>
                                     </head>
                                      <body>
  <input type="text" id="valorIni"> * 50 = <input type="text" id="resultado"
readonly="true"> <button type="button" onclick="calcula();">Calcular</button>
                                     </body>
                                     </html>
```

bom, acho que deu pra entender neh galera, o script manda o valor de N por get, pelo proprio endereco (scriptAjax.php?n=xxx), pra completar, eu vo ensina abaixo como seria pra enviar esse mesmo dado por POST:

CODE

```
                                        ...
                            ajax.open("POST", "scriptAjax.php", true);
                ajax.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
                                        ...
                                ajax.send("n=" + valIni);
```

escrevi os ... pra dizer q a parte do codigo q falta, tanto por get como por post, na hora de enviar variaveis, vc deve as separar por &, td numa string soh (mesmo no post, n pense q ele aceita varios argumentos pois é um só)

vo dexa a descricao dos objetos abaixo (peguei no PHPBrasil):

abort():
Pára a requisição atual.

getAllResponseHeaders():
Retorna todos os cabeçalhos (labels e valores) como uma string.

getResponseHeader("headerLabel"):
Retorna o valor de um único label do cabeçalho.

open("metodo", "URL"[, asyncFlag[, "userName"[, "password"]]]):
Define a página a ser aberta, o método (GET | POST), a URL, o marcador de "assíncrono", e, se a página requerer login, o nome de usuário e a senha.

send(content):
Envia a requisição, opcionalmente pode ser uma string ou até um objeto DOM.

setRequestHeader("label", "valor"):
Define um par label + valor para ser enviado com o cabeçalho da requisição atual.

E as propriedades:

onreadystatechange:
(!) Método a ser invocado quando o estado (definido em readyState) mudar. (!)

readyState:
Inteiro representando o estado da requisição:
0 = não inicializado
1 = carregando
2 = carregado
3 = modo interativo
4 = completado

responseText:
Versão em string dos dados retornados pela requisição.

responseXML:
Objeto DOM retornado pela requisição

status:
Código numérico do estado da requisição retornado pelo servidor. O conhecido 404 "Not Found" e o menos conhecido, mas mais importante 200 "Ok"

statusText:
A string que acompanha o código de estado acima ("Not Found", "Ok", etc)

bom galera, pra completar aqui, vou mostrar um exemplo pratico do uso do Ajax

vamos aqui aprender a criar um sistema de contagem de usuarios online, a diferenca desse sistema para os demais da net, eh q ele funciona em tempo real, ou seja, c alguem entrar no site, ou sair (o prazo de tempo expirar), o contador eh atualizado na mesma hora, sem precisar trocar de pagina

pra nao complicar, usaremos um arquivo de texto simples no lugar de um banco de dados

quanto a explicacao, agora eu vou me focar mais no javascript do que no PHP, pois eu nao expliquei bem isso no tutorial, e podem ter ficado duvidas, vamos agora criar o script PHP que faz a contagem e atualizacao dos usuarios online

usersCount.php

PHP

```php
<?php

$gmtDate = gmdate("D, d M Y H:i:s");
header("Expires: {$gmtDate} GMT");
header("Last-Modified: {$gmtDate} GMT");
header("Cache-Control: no-cache, must-revalidate");
header("Pragma: no-cache");

//os headers acima ja foram explicados no tutorial

$timeExpire = 30; //tempo em segundos para expirar usuario
$fileName = "online.txt"; //nome do arquivo a ser usado

//criar o arquivo se ele não existir
if(!file_exists($fileName)) {
    $f = fopen($fileName, "w");
    fclose($f);
}

$ip = $_SERVER['REMOTE_ADDR']; //pegar ip do usuario
$tempo = time(); //pegar o timespan

$stringUser = $ip . ":" . $tempo; //concatenamos usando o caractere : como divisor

$onlineNow = file_get_contents($fileName); //arquivo txt com dados de usuarios atuais
```

```
26.  $arrayNow = explode("|", $onlineNow); //o divisor usado para separar valores é |
27.  $newUsers = array(); //essa array irá gravar os usuarios ainda online
28.
29.  //agora vamos filtrar os usuarios com timespan vencido
30.  foreach($arrayNow as $an) {
31.      list($tIP, $tTime) = explode(":", $an); //separamos usando o separador previamente definido
32.
33.      if($tIP != NULL && $tTime != NULL && $tIP != $ip && $tTime > $tempo - $timeExpire) {
34.          $newUsers[] = $tIP . ":" . $tTime; //caso o usuario passe no teste, ele é adicionado
35.      }
36.  }
37.
38.  $newUsers[] = $stringUser; //adicionando o usuario atual
39.
40.  file_put_contents($fileName, implode("|", $newUsers)); //atualiza o arquivo
41.
42.  echo count($newUsers); //escreve a quantidade de usuarios atual
43.
44.  ?>
```

dexei comentarios expalhados, entao esse script é auto-explicativo, postem em caso de duvidas

vamos usar o arquivo ajaxInit.js que criamos no tutorial, e agora vamos ao script js que vai ler esses dados:

usersCount.js

CODE

```
function contaUsuarios(spanID) {
var usersCountTmpVar = document.getElementById(spanID);

if(!usersCountTmpVar) {
alert("Campo não encontrado");
return; //se ele não encontrar o campo, da um alerta e escapa a funcao
}

var ajaxUC = ajaxInit(); //inicia a variavel ajax para uso e UserCount
ajaxUC.open("GET", "usersCount.php", true);
ajaxUC.onreadystatechange = function() { //funcao executada ao trocar de stado
if(ajaxUC.readyState == 4) { //verifica se o estado atual é "concluido"
if(ajaxUC.status == 200) { //verifica se o arquivo foi lido corretamente
usersCountTmpVar.innerHTML = ajaxUC.responseText + " usuários online"; //define o texto do span
}

setTimeout("contaUsuarios('" + spanID + "')", 2000); //reexecutar a funcao apos 2 segundos
}
}
ajaxUC.send(null); //enviar dados para poder receber resposta
}
```

bom, nem tem mto oq explicar ai, foi passado um parametro que indica o objeto aonde o texto informativo vai ser colocado, o resto ta auto-explicativo, consultem o tutorial para ver como funcionam os metodos do ajax

agora que temos os scripts prontos, vamos usa-los

pagina.php

CODE

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
      <html xmlns="http://www.w3.org/1999/xhtml">
                          <head>
    <title> :: Ajax - Contador de Usuarios :: </title>
 <script src="ajaxInit.js" type="text/javascript"></script>
<script src="usersCount.js" type="text/javascript"></script>
                          </head>
      <body onload="contaUsuarios('usuariosOnline');">
             <span id="usuariosOnline"></span>
                          </body>
                          </html>
```

como podem ver, incluimos os arquivos js que serao usados, definimos uma tag (tambem poderia ser), e colocamos um identificador (id) nela, e na tag body, dizemos que ao ler o documento, ele deve executar a funcao de contar usuarios, passando o retorno para o identificador 'usuariosOnline', nao podem esquecer de usar aspas, senao vai dar erro

pra quem nao sabe, a diferenca principal entre e é que o assim que usado, ele preenche o maximo de margem horizontal que ele puder, ou seja, apos um div, qualquer coisa no HTML vai vir numa linha inferior, enquanto o soh ocupa o espaco usado pelo seu HTML interno, c vc digitar algo apos o span, esse algo vai aparecer ao lado, e nao em baixo