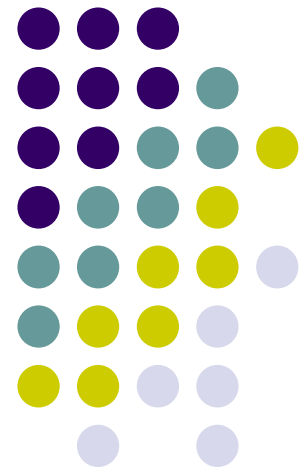


Ajax

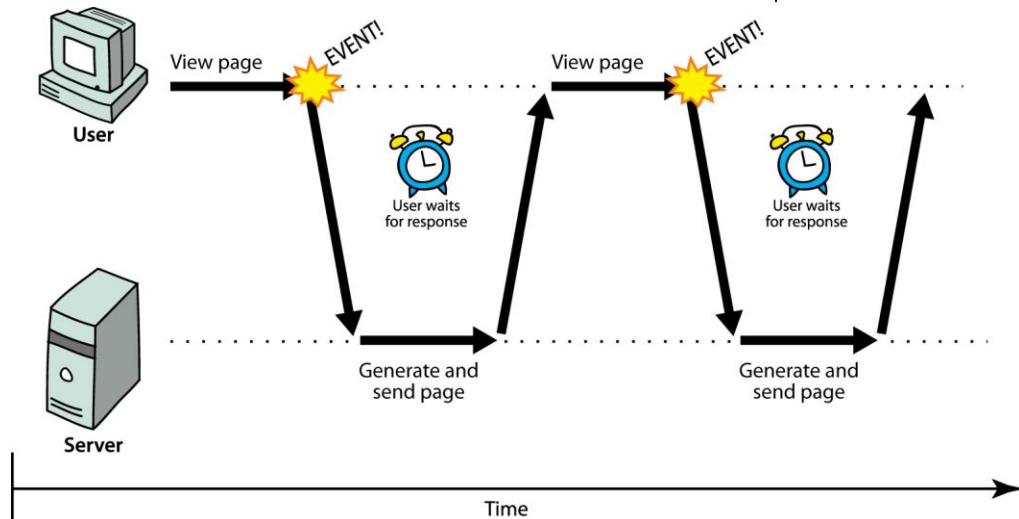




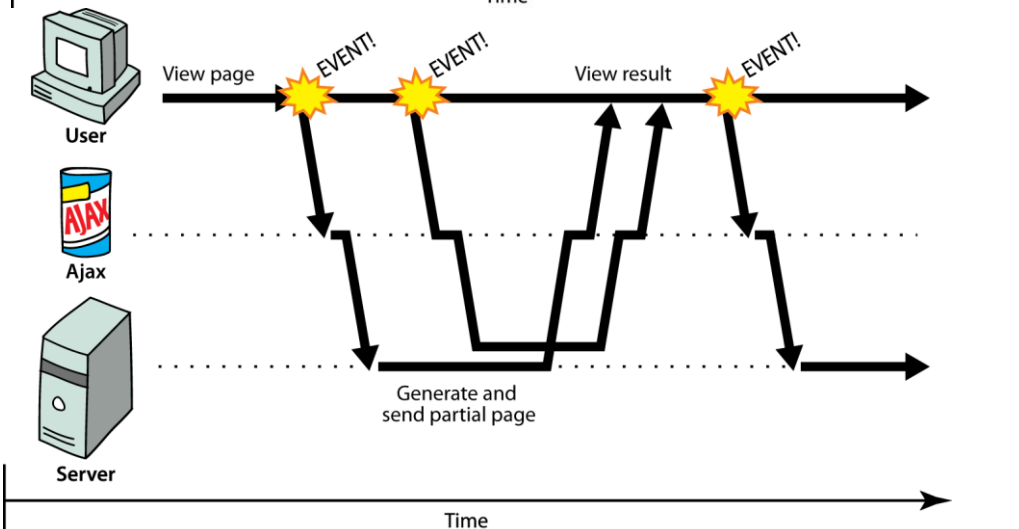
- aplicación web: un sitio web que imita una aplicación de desktop
 - Presenta una experiencia continua, y no la sensación de un conjunto de paginas separadas.
 - ejemplos: Gmail, Google Maps, Google Docs and Spreadsheets, Flickr
- Ajax: Asynchronous JavaScript and XML
 - No es un lenguaje de programación, es un modo particular de usar JavaScript
 - Descarga datos de un servidor en background
 - Permite actualizar dinámicamente una pagina sin que el usuario tenga que esperar

Idea de Ajax

- Eventos síncronos



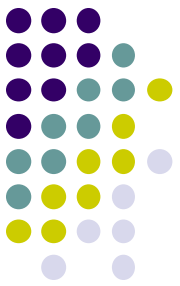
- Eventos asíncronos



XMLHttpRequest

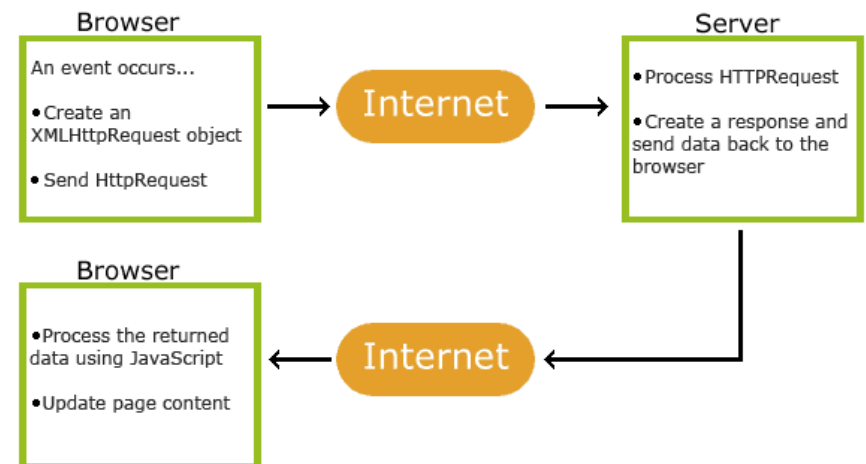
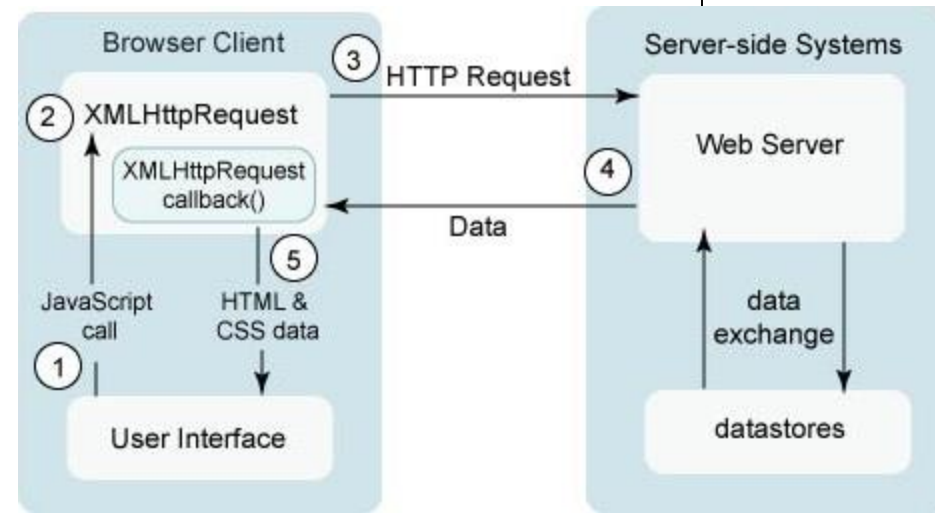


- JavaScript incluye un objeto **XMLHttpRequest** que puede ser recuperado desde el servidor
- Suportado por IE5+, Safari, Firefox, Opera, Chrome, etc.
- Puede ser usado de forma asíncrona (en background, de forma transparente al usuario)
- El contenido de la petición web puede ser insertado en la pagina web correrte usando los DOM
- Parece perfecto pero tiene varias incompatibilidades de browser
 - Se aconseja el uso de librerias (Prototype o JQuery para solucionar compatibilidad)

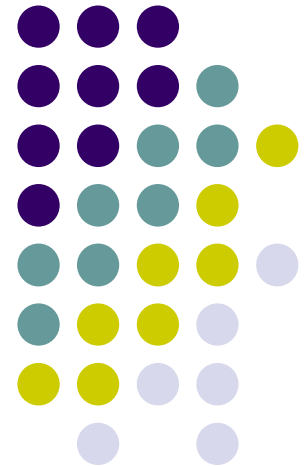


Una típica operación ajax

- El usuario clic, pidiendo un handler de evento
- El código del handler crea un objeto XMLHttpRequest
- El objeto XMLHttpRequest pide una página al servidor
- El servidor recupera los datos correspondientes y los envía al cliente.
- XMLHttpRequest activa un evento (callback) cuando los datos llegan
- Se puede juntar un handler al evento callback
- El handler del evento callback procesa los datos y los enseña



Ejemplos Ajax





Ejemplo ajax (auto-complete)

- http://www.w3schools.com/Php/php_ajax_php.asp

Example

Start typing a name in the input field below:

First name:

Suggestions: Anna, Amanda

CLIENTE

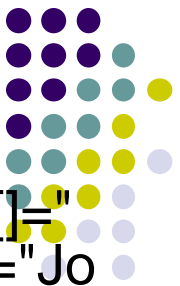


2 {

```
<html> <head><script>
function showHint(str)
{if (str.length==0)
{
    document.getElementById("txtHint").innerHTML="";
    return;
}
var xmlhttp=new XMLHttpRequest();
xmlhttp.onreadystatechange=function()
{
    if (xmlhttp.readyState==4 && xmlhttp.status==200)
    {
        document.getElementById("txtHint").innerHTML=xmlhttp.responseText;
    }
}
xmlhttp.open("GET","gethint.php?q="+str,true);
xmlhttp.send();
}
</script></head>
```

1 {

```
<body>
<p><b>Start typing a name in the input field below:</b></p>
<form>
First name: <input type="text" onkeyup="showHint(this.value)">
</form>
<p>Suggestions: <span id="txtHint"></span></p>
</body></html>
```

```
<?php
// Fill up array with names
$a[]="Anna";$a[]="Brittany";$a[]="Cinderella";$a[]="Diana";$a[]="
Eva";$a[]="Fiona";$a[]="Gunda";$a[]="Hege";$a[]="Inga";$a[]="Jo
hanna";$a[]="Kitty";$a[]="Linda";$a[]="Nina";$a[]="Ophelia";$a[]="
Petunia";$a[]="Amanda";$a[]="Raquel";$a[]="Cindy";$a[]="Doris";

// get the q parameter from URL
$q=$_REQUEST["q"]; $hint="";

// lookup all hints from array if $q is different from ""
if ($q != "")
{ $q=strtolower($q); $len=strlen($q);
  foreach($a as $name)
  { if (stristr($q, substr($name,0,$len)))
    { if ($hint=="")
      { $hint=$name; }
      else
      { $hint .= ", $name"; }
    }
  }
}

// Output "no suggestion" if no hint were found
// or output the correct values
echo $hint==" " ? "no suggestion" : $hint;

?>
```



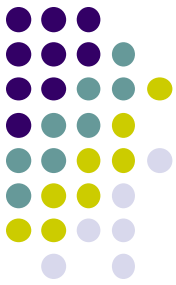
Ejemplo ajax (connexion DB)

- http://www.w3schools.com/Php/php_ajax_database.asp

Example

Lois Griffin	▼
Select a person:	
Peter Griffin	
Lois Griffin	
Joseph Swanson	
Glenn Quagmire	

Firstname	Age	Hometown	Job
Griffin	40	Newport	Piano Teacher

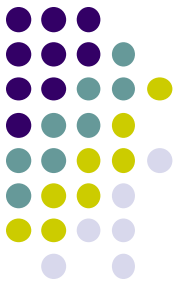


```
<html> <head> <script>
function showUser(str)
{if (str=="")
{
document.getElementById("txtHint").innerHTML="";
return; }
}
```

```
xmlhttp=new XMLHttpRequest();
xmlhttp.onreadystatechange=function()
{
if (xmlhttp.readyState==4 && xmlhttp.status==200)
{
document.getElementById("txtHint").innerHTML=xmlhttp.responseText;
}
}
xmlhttp.open("GET","getuser.php?q="+str,true);
xmlhttp.send();
}
</script> </head> <body>
```

```
<form>
<select name="users" onchange="showUser(this.value)">
<option value="">Select a person:</option>
<option value="1">Peter Griffin</option> <option value="2">Lois Griffin</option>
<option value="3">Glenn Quagmire</option> <option value="4">Joseph Swanson</option>
</select> </form>
<br> <div id="txtHint"><b>Person info will be listed here.</b></div>
</body></html>
```

Servidor



```
<?php
$q = intval($_GET['q']);

$con = mysqli_connect('localhost','peter','abc123','my_db');
if (!$con)
{ die('Could not connect: ' . mysqli_error($con));}
mysqli_select_db($con,"ajax_demo");
$sql="SELECT * FROM user WHERE id = '". $q ."'";
```

```
echo "<table border='1'>
<tr>
<th>Firstname</th>                <th>Lastname</th>
<th>Age</th>                      <th>Hometown</th>
        <th>Job</th>
</tr>";
```

```
$result = mysqli_query($con,$sql);
while($row = mysqli_fetch_array($result))
{
    echo "<tr>";
    echo "<td>" . $row['FirstName'] . "</td>";    echo "<td>" . $row['LastName'] . "</td>";
    echo "<td>" . $row['Age'] . "</td>";          echo "<td>" . $row['Hometown'] . "</td>";
    echo "<td>" . $row['Job'] . "</td>";
    echo "</tr>";
}
```

```
echo "</table>";
mysqli_close($con);
```

id	FirstName	LastName	Age	Hometown	Job
1	Peter	Griffin	41	Quahog	Brewery
2	Lois	Griffin	40	Newport	Piano Teacher
3	Joseph	Swanson	39	Quahog	Police Officer
4	Glenn	Quagmire	41	Quahog	Pilot

?>

Los métodos de XMLHttpRequest



Método	Descripción
<code>open(<i>method</i>, <i>url</i>, <i>async</i>)</code>	Especifica el URL y el método HTTP
<code>send()</code> <code>send(<i>postData</i>)</code>	Envía la petición HTTP al servidor, usando parámetros opcionales POST
<code>abort()</code>	Interrumpe la petición
<code>getAllResponseHeaders()</code> <code>getResponseHeader(<i>name</i>)</code> <code>setRequestHeader(<i>name</i>, <i>value</i>)</code>	Sirve para recuperar los datos HTTP

Propiedades de XMLHttpRequest



Propiedad	Descripción
responseText	El texto de las informaciones recuperadas , con formato string
responseXML	El contenido de las informaciones recuperadas, en formato XML
status	El código de la petición HTTP (200 = OK, etc.)
statusText	El texto del código de la petición (e.g. “Bad Request” para 400)
timeout	Cuantos mili-segundos se espera antes de interrumpir la petición (default 0 = espera para siempre)
readyState	El estado de la petición actual (0 = non inicializada, 1 = lista, 2 = enviada, 3 = en progreso, 4 = completada)



Eventos XMLHttpRequest

Evento	Descripción
load	Ocurre cuando la petición es completada
error	Ocurre cuando la petición falla
timeout	ocurre cuando la petición pasa el limite de tiempo indicado
abort	Ocurre cuando la petición es interrumpida llamando abort()
loadstart, loadend, progress, readystatechange	Eventos de progresión para seguir una petición en curso



Ajax ideas básicas

```
var ajax = new XMLHttpRequest();  
ajax.onload = functionName;  
ajax.open("GET", url, true);  
ajax.send();  
...  
function functionName() {  
    do something with this.responseText;  
}
```

- **Adjuntamos un event handler al evento onload**
- El handler será llamado cuando el estado de la petición cambia (por ejemplo cuando acaba)
- La función contiene el código que se ejecuta cuando la petición es completa
 - **this.*** Se refiere al objeto Ajax
 - Permite acceder al texto de la respuesta y a otras propiedades



Ajax ideas básicas

```
var ajax = new XMLHttpRequest();
ajax.onload = functionName;
ajax.open("GET", url, true);
ajax.send();

...
function functionName() {
    if (this.status == 200) {
        do something with this.responseText;
    } else {
        code to handler the error;
    }
}
```

- El servidor web devuelve un código de estado (por ejemplo 200 significa suceso)
- Se puede escribir un código en el caso que haya un fallo



Gestionar el error de evento

```
var ajax = new XMLHttpRequest();  
ajax.onload = functionName;  
ajax.onerror = errorFunctionName;  
ajax.open("GET", url, true);  
ajax.send();  
...  
function errorFunctionName(e) {  
    do something with e, this.status, this.statusText, ...  
}
```

- Podemos adjuntar un **listener** al evento “error”
- De esta forma se puede examinar el error, como el estado de la petición, para saber que ha pasado.

Ejemplo: Gestión del error asociado al evento



```
var ajax = new XMLHttpRequest();  
...  
ajax.onerror = ajaxFailure;  
...  
function ajaxFailure(exception) {  
    alert("Error making Ajax request:" +  
        "\n\nServer status:\n" + this.status + " " +  
        this.statusText + "\n\nServer response text:\n" +  
        this.responseText);  
    if (exception) {  
        throw exception;  
    }  
}
```

Pasar distintos parámetros a una petición Ajax (usando get)



```
var ajax = new XMLHttpRequest();  
ajax.onload = functionName;  
ajax.open("GET", "url?name1=value1&name2=value2&...",  
true);  
ajax.send();
```

- Para pasar parámetros se pueden concatenarlos al URL

Pasar parámetros a una petición Ajax (usando post)

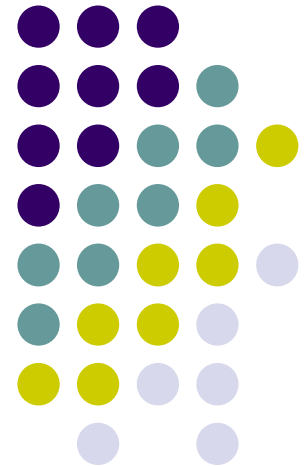


```
var params = new FormData();  
params.append("name", value);
```

```
var ajax = new XMLHttpRequest();  
ajax.onload = functionName;  
ajax.open("POST", "url", true);  
ajax.send(params);
```

- Se puede usar un objeto Formdata para recuperar los parametros query
- Los parametros (params) se envian por medio del metodo send

Ajax usando prototype





Ajax usando Prototype

```
new Ajax.Request("url", {  
    option : value,  
    option : value,  
    ...  
    option : value  
});
```

- Se construye un Prototype Ajax. Pide un objeto desde una pagina usando Ajax
- El constructor acepta dos parametros:
 - El **URL** adonde conectarse, de tipo String,
 - Un set de opciones, como un conjuntos de parejas *key : value* contenidas en {}
- Esta forma remplaza el XMLHttpRequest; y funciona bien en todos los browsers

Opciones de Ajax con prototype



1. method
 2. asynchronous
 3. parameters
 4. onSuccess
 5. onFailure
 6. onException
 - otros:
 - contentType, encoding, requestHeaders; events: onCreate, onComplete, on#### (for HTTP error code ####)
1. Como recuperar una petición desde el server
 2. Si la petición debe ser enviada en modo asíncrono en background? (por defecto true)
 3. Parámetros de una query que se tienen que pasar al servidor, (como string o object)
 4. *event*: petición completada con successo
 5. *event*: la petición fue sin sucesso
 6. *event*: la petición tiene un error de sintaxis, o un error de seguridad

Ejemplo de ajax con prototype



```
new Ajax.Request("foo/bar/mydata.txt", {  
    method: "get",  
    onSuccess: myAjaxSuccessFunction  
});  
...
```

```
function myAjaxSuccessFunction ajax) {  
    hacer algo con el "ajax.responseText";  
}
```

- Junta una función event handler al evento onSuccess
- El handler recibe un **Ajax response object** como un parámetro.

Propiedades de un Ajax response object



- | | |
|---|---|
| <ol style="list-style-type: none">1. status2. statusText3. <i>responseText</i>4. responseXML,
responseJSON | <ol style="list-style-type: none">1. el codigo HTTP de la peticion (200 = OK, etc.)2. El texto HTTP del codigo de status3. El texto completo del fichero pedido, de tipo string4. El contenido del fichero pedido, en otros formatos |
|---|---|

```
function myAjaxSuccessFunction ajax {  
    alert(ajax.responseText);  
}
```

Como gestionar errores de Ajax (ejemplo)



```
new Ajax.Request("url", {
    method: "get",
    onSuccess: functionName,
    onFailure: ajaxFailure,
    onException: ajaxFailure
});
...
function ajaxFailure ajaxFailure (ajax, exception) {
    alert("Error recibido en una peticion de Ajax:" +
        "\n\nestado del servidor :\n" + ajax.status + " " +
        ajax.statusText +
        "\n\ntexto de respuesta del Server :\n" + ajax.responseText);
    if (exception) {
        throw exception;
    }
}
```

Pasar parámetros queries a una petición Ajax



```
new Ajax.Request("lookup_account.php", {  
    method: "get",  
    parameters: {name: "Ed Smith", age: 29, password: "abcdef"},  
    onFailure: ajaxFailure,  
    onException: ajaxFailure  
});
```

- Los parámetros de query se pasan como objeto de parámetros, entre paréntesis {} y con sintaxis “name : value”
 - (la petición arriba es equivalente a escribir:
"name=Ed+Smith&age=29&password=abcdef")

Creación de una petición post



```
new Ajax.Request("url", {  
    method: "post",  
    parameters: {name: value, name: value, ...,  
                name: value},  
    onSuccess: functionName,  
    onFailure: functionName,  
    onException: functionName  
});
```

Ejemplo autocomplete scriptaculus



```
<html> <head>
  <title>Simple Ajax Auto-completer Example</title>
  <script type = "text/javascript" src = "/javascript/prototype.js"></script>
  <script type = "text/javascript" src = "/javascript/scriptaculous.js?load =
effects,controls"></script>
  <script type = "text/javascript">
    window.onload = function() {
      new Ajax.AutoCompleteer(
        'autoCompleteTextField',
        'autoCompleteMenu',
        '/script.aculo.us/serverSideScript.php', {}
      );
    }
  </script> </head>
```

```
serverSideScript.php
<ul>
  <li>One</li>
  <li>Two</li>
  <li>Three</li>
  <li>Four</li>
  <li>Five</li>
  <li>Six</li>
</ul>
```

Type something in this box and then select suggested option from the list

Text field:

- One
- Two
- Three
- Four
- Five

```
<body>
  <p>Type something in this box and then select suggested option from the list </p>
  <div>
    <label>Text field:</label>
    <input type = "text" id = "autoCompleteTextField"/>
    <div id = "autoCompleteMenu"></div>
  </div> </body> </html>
```

Ejemplo ajax scriptaculus



```
<html>  <head>    <title>Prototype examples</title>
  <script type = "text/javascript" src = "/javascript/prototype.js"></script>
  <script>
    function SubmitRequest() {
      new Ajax.Request('/cgi-bin/ajax.cgi', {
        method: 'get',
        onSuccess: successFunc,
        onFailure: failureFunc
      });
    }
    function successFunc(response) {
      if (200 == response.status) {
        alert("Call is success");
      }
      var container = $('notice');
      var content = response.responseText;
      container.update(content);
    }
    function failureFunc(response) {
      alert("Call is failed" );
    }
  </script>  </head>
```

ajax.cgi

```
#!/usr/bin/perl
print "Content-type: text/html\n\n";
print "This content is returned by AJAX cgi <br />";
print "Current Time " . localtime;
```

```
<body>  <p>Click submit button see how current notice changes.</p>  <br />
<div id = "notice">Current Notice</div>  <br />  <br />
  <input type = "button" value = "Submit" onclick = "SubmitRequest();"/>
</body> </html>
```

SRC=https://www.tutorialspoint.com/prototype/prototype_ajax_request.htm



Ejercicios colaborativos

- <https://codeshare.io/COW06>
- Realizar una conexión Ajax entre dos ficheros, cliente.html y servidor.php,
 - El fichero servidor.php visualiza las primeras 5 ciudades de la tabla world (usar el fichero servidor.php PDO creado en la unidad anterior)
 - Implementar el cliente.

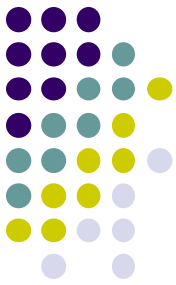
Appendix

Códigos ajax



Codigo ejemplo w3school

http://www.w3schools.com/Php/php_ajax_php.asp



Emaple.html

```
<html>
<head>
<script>
function showHint(str) {
    if (str.length == 0) {
        document.getElementById("txtHint").innerHTML = "";
        return;
    } else {
        var xmlhttp = new XMLHttpRequest();
        xmlhttp.onreadystatechange = function() {
            if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
                document.getElementById("txtHint").innerHTML =
xmlhttp.responseText;
            }
        };
        xmlhttp.open("GET", "gethint.php?q=" + str, true);
        xmlhttp.send();
    }
}
</script>
</head>
<body>

<p><b>Start typing a name in the input field below:</b></p>
<form>
First name: <input type="text" onkeyup="showHint(this.value)">
</form>
<p>Suggestions: <span id="txtHint"></span></p>
</body>
</html>
```

gethint.php

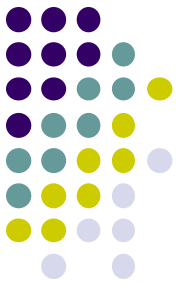
```
<?php
// Array with names
$a[] = "Anna";
$a[] = "Brittany";
$a[] = "Cindrella";
$a[] = "Diana";
$a[] = "Eva";
$a[] = "Fiona";
$a[] = "Gunda";
$a[] = "Haga";
$a[] = "Inga";
$a[] = "Johanna";
$a[] = "Kitty";
$a[] = "Linda";
$a[] = "Mica";
$a[] = "Ophelia";
$a[] = "Petunia";
$a[] = "Amanda";
$a[] = "Ragini";
$a[] = "Cindy";
$a[] = "Daria";
$a[] = "Eva";
$a[] = "Britta";
$a[] = "Bianca";
$a[] = "Emma";
$a[] = "Domi";
$a[] = "Violet";
$a[] = "Lisa";
$a[] = "Elizabeth";
$a[] = "Ellen";
$a[] = "Mascha";
$a[] = "Vicky";

// get the q parameter from URL
$q = $_REQUEST["q"];

$hint = "";

// lookup all hints from array if $q is different from ""
if ($q != "") {
    $q = strtolower($q);
    $len=strlen($q);
    foreach($a as $name) {
        if (stripos($q, substr($name, 0, $len)) {
            if ($hint == "") {
                $hint = $name;
            } else {
                $hint .= ", $name";
            }
        }
    }
}

// Output "no suggestion" if no hint was found or output correct values
echo $hint == "" ? "no suggestion" : $hint;
>
```



Codigo ejemplo w3school

http://www.w3schools.com/Php/php_ajax_database.asp

Example.html

```
<html>
<head>
<script>
function showUser(str) {
    if (str == "") {
        document.getElementById("txtHint").innerHTML = "";
        return;
    } else {
        if (window.XMLHttpRequest) {
            // code for IE7+, Firefox, Chrome, Opera, Safari
            xmlhttp = new XMLHttpRequest();
        } else {
            // code for IE6, IE5
            xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
        }
        xmlhttp.onreadystatechange = function() {
            if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
                document.getElementById("txtHint").innerHTML = xmlhttp.responseText;
            }
        };
        xmlhttp.open("GET","getuser.php?q="+str,true);
        xmlhttp.send();
    }
}
</script>
</head>
<body>

<form>
<select name="users" onchange="showUser(this.value)">
  <option value="">Select a person:</option>
  <option value="1">Peter Griffin</option>
  <option value="2">Lois Griffin</option>
  <option value="3">Joseph Swanson</option>
  <option value="4">Glenn Quagmire</option>
</select>
</form>
<br>
<div id="txtHint"><b>Person info will be listed here...</b></div>

</body>
</html>
```

getuser.php

```
<!DOCTYPE html>
<html>
<head>
<style>
table {
    width: 100%;
    border-collapse: collapse;
}

table, td, th {
    border: 1px solid black;
    padding: 5px;
}

th {text-align: left;}
</style>
</head>
<body>

<?php
$q = intval($_GET['q']);

$con = mysqli_connect('localhost','peter','abc123','my_db');
if (!$con) {
    die('Could not connect: ' . mysqli_error($con));
}

mysqli_select_db($con,"ajax_demo");
$sql="SELECT * FROM user WHERE id = ".$q."";
$result = mysqli_query($con,$sql);

echo "<table>
<tr>
<th>Firstname</th>
<th>Lastname</th>
<th>Age</th>
<th>Hometown</th>
<th>Job</th>
</tr>";
while($row = mysqli_fetch_array($result)) {
    echo "<tr>";
    echo "<td>" . $row['FirstName'] . "</td>";
    echo "<td>" . $row['LastName'] . "</td>";
    echo "<td>" . $row['Age'] . "</td>";
    echo "<td>" . $row['Hometown'] . "</td>";
    echo "<td>" . $row['Job'] . "</td>";
    echo "</tr>";
}
echo "</table>";
mysqli_close($con);
?>
</body>
</html>
```