

PHP - AJAX and PHP

[« Previous](#)

[Next Chapter »](#)

AJAX is used to create more interactive applications.

AJAX PHP Example

The following example will demonstrate how a web page can communicate with a web server while a user type characters in an input field:

Example

Start typing a name in the input field below:

First name:

Suggestions:

Example Explained - The HTML Page

When a user types a character in the input field above, the function "showHint()" is executed. The function is triggered by the "onkeyup" event:

```
<html>
<head>
<script>
function showHint(str)
{
if (str.length==0)
{
document.getElementById("txtHint").innerHTML="";
return;
}
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","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>

```

Source code explanation:

If the input field is empty (str.length==0), the function clears the content of the txtHint placeholder and exits the function.

If the input field is not empty, the showHint() function executes the following:

- Create an XMLHttpRequest object
- Create the function to be executed when the server response is ready
- Send the request off to a file on the server
- Notice that a parameter (q) is added to the URL (with the content of the input field)

The PHP File

The page on the server called by the JavaScript above is a PHP file called "gethint.php".

The source code in "gethint.php" checks an array of names, and returns the corresponding name(s) to the browser:

```

<?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[]="Johanna";
$a[]="Kitty";
$a[]="Linda";
$a[]="Nina";
$a[]="Ophelia";
$a[]="Petunia";
$a[]="Amanda";
$a[]="Raquel";
$a[]="Cindy";
$a[]="Doris";
$a[]="Eve";
$a[]="Evita";
$a[]="Sunniva";
$a[]="Tove";
$a[]="Unni";
$a[]="Violet";
$a[]="Liza";
$a[]="Elizabeth";
$a[]="Ellen";
$a[]="Wenche";
$a[]="Vicky";
```

```
//get the q parameter from URL
```

```
$q=$_GET["q"];
```

```
//lookup all hints from array if length of q>0
```

```
if (strlen($q) > 0)
```

```
{
```

```
  $hint="";
```

```
  for($i=0; $i<count($a); $i++)
```

```
  {
```

```
    if (strtolower($q)==strtolower(substr($a[$i],0,strlen($q))))
```

```
    {
```

```
      if ($hint=="")
```

```
      {
```

```
        $hint=$a[$i];
```

```
      }
```

```
    else
```

```
    {
```

```
      $hint=$hint." , ".$a[$i];
```

```
    }
```

```
  }
```

```
}
```

```
}
```

```
// Set output to "no suggestion" if no hint were found
// or to the correct values
if ($hint == "")
{
    $response="no suggestion";
}
else
{
    $response=$hint;
}

//output the response
echo $response;
?>
```

Explanation: If there is any text sent from the JavaScript (`strlen($q) > 0`), the following happens:

1. Find a name matching the characters sent from the JavaScript
2. If no match were found, set the response string to "no suggestion"
3. If one or more matching names were found, set the response string to all these names
4. The response is sent to the "txtHint" placeholder

PHP - AJAX and MySQL

[<< Previous](#)
[Next Chapter >>](#)

AJAX can be used for interactive communication with a database.

AJAX Database Example

The following example will demonstrate how a web page can fetch information from a database with AJAX:

Example

Person info will be listed here...

Example Explained - The MySQL Database

The database table we use in the example above looks like this:

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

Example Explained - The HTML Page

When a user selects a user in the dropdown list above, a function called "showUser()" is executed. The function is triggered by the "onchange" event:

```

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

```

The showUser() function does the following:

- Check if a person is selected
 - Create an XMLHttpRequest object
 - Create the function to be executed when the server response is ready
 - Send the request off to a file on the server
 - Notice that a parameter (q) is added to the URL (with the content of the dropdown list)
-

The PHP File

The page on the server called by the JavaScript above is a PHP file called "getuser.php".

The source code in "getuser.php" runs a query against a MySQL database, and returns the result in an HTML table:

```

<?php
$q=$_GET["q"];

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

mysql_select_db("ajax_demo", $con);

$sql="SELECT * FROM user WHERE id = '".$q."'";

$result = mysql_query($sql);

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

```

```
while($row = mysql_fetch_array($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>";

mysql_close($con);
?>
```

Explanation: When the query is sent from the JavaScript to the PHP file, the following happens:

1. PHP opens a connection to a MySQL server
2. The correct person is found
3. An HTML table is created, filled with data, and sent back to the "txtHint" placeholder

PHP Example - AJAX and XML

[« Previous](#)
[Next Chapter »](#)

AJAX can be used for interactive communication with an XML file.

AJAX XML Example

The following example will demonstrate how a web page can fetch information from an XML file with AJAX:

Example

CD info will be listed here...

Example Explained - The HTML Page

When a user selects a CD in the dropdown list above, a function called "showCD()" is executed. The

function is triggered by the "onchange" event:

```
<html>
<head>
<script>
function showCD(str)
{
if (str=="")
{
document.getElementById("txtHint").innerHTML="";
return;
}
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","getcd.php?q="+str,true);
xmlhttp.send();
}
</script>
</head>
<body>

<form>
Select a CD:
<select name="cds" onchange="showCD(this.value)">
<option value="">Select a CD:</option>
<option value="Bob Dylan">Bob Dylan</option>
<option value="Bonnie Tyler">Bonnie Tyler</option>
<option value="Dolly Parton">Dolly Parton</option>
</select>
</form>
<div id="txtHint"><b>CD info will be listed here...</b></div>

</body>
</html>
```

The showCD() function does the following:

- Check if a CD is selected
 - Create an XMLHttpRequest object
 - Create the function to be executed when the server response is ready
 - Send the request off to a file on the server
 - Notice that a parameter (q) is added to the URL (with the content of the dropdown list)
-

The PHP File

The page on the server called by the JavaScript above is a PHP file called "getcd.php".

The PHP script loads an XML document, "[cd_catalog.xml](#)", runs a query against the XML file, and returns the result as HTML:

```
<?php
$q=$_GET["q"];

$xmlDoc = new DOMDocument();
$xmlDoc->load("cd_catalog.xml");

$x=$xmlDoc->getElementsByTagName('ARTIST');

for ($i=0; $i<=$x->length-1; $i++)
{
    //Process only element nodes
    if ($x->item($i)->nodeType==1)
    {
        if ($x->item($i)->childNodes->item(0)->nodeValue == $q)
        {
            $y=($x->item($i)->parentNode);
        }
    }
}

$cd=($y->childNodes);

for ($i=0;$i<$cd->length;$i++)
{
    //Process only element nodes
    if ($cd->item($i)->nodeType==1)
    {
        echo("<b>" . $cd->item($i)->nodeName . "</b> ");
        echo($cd->item($i)->childNodes->item(0)->nodeValue);
        echo("<br>");
    }
}
?>
```

When the CD query is sent from the JavaScript to the PHP page, the following happens:

1. PHP creates an XML DOM object
2. Find all <artist> elements that matches the name sent from the JavaScript
3. Output the album information (send to the "txtHint" placeholder)