

# AJAX Introduction

AJAX is about updating parts of a web page, without reloading the whole page.

## What is AJAX?

AJAX = Asynchronous JavaScript and XML.

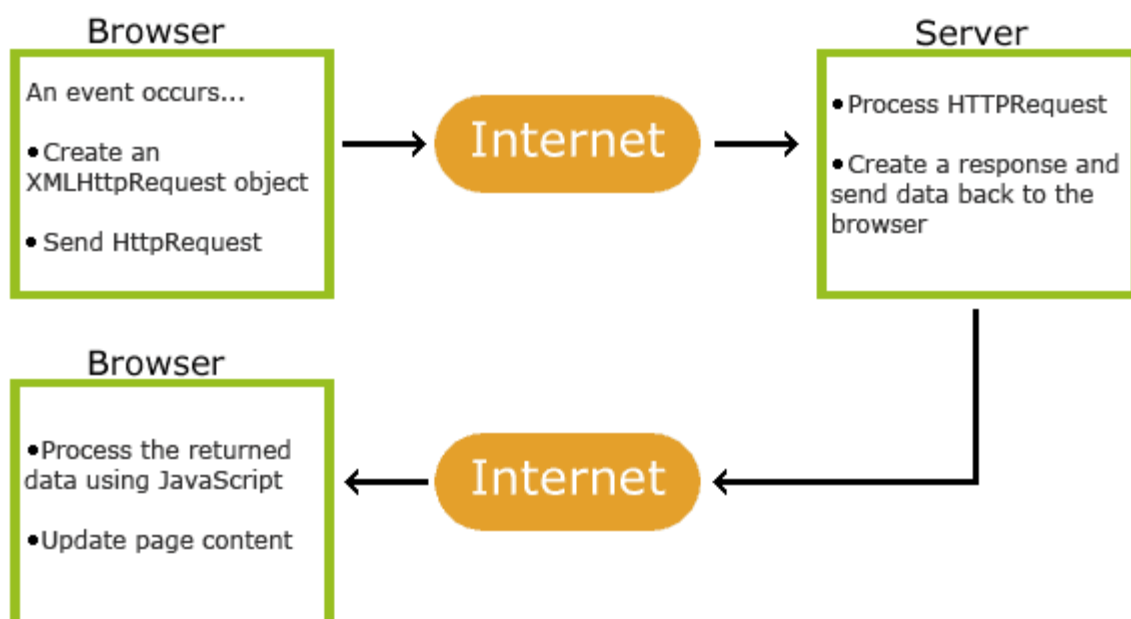
AJAX is a technique for creating fast and dynamic web pages.

AJAX allows web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.

Classic web pages, (which do not use AJAX) must reload the entire page if the content should change.

Examples of applications using AJAX: Google Maps, Gmail, Youtube, and Facebook tabs.

## How AJAX Works



# AJAX is Based on Internet Standards

AJAX is based on internet standards, and uses a combination of:

- XMLHttpRequest object (to exchange data asynchronously with a server)
- JavaScript/DOM (to display/interact with the information)
- CSS (to style the data)
- XML (often used as the format for transferring data)

AJAX applications are browser- and platform-independent!

## PHP - AJAX and PHP

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

In the example above, when a user types a character in the input field, a function called "showHint()" is executed.

The function is triggered by the onkeyup event.

Here is the HTML code:

## Example

```
<html>
<head>
<script>
function showHint(str) {
    if (str.length == 0) {
        document.getElementById("txtHint").innerHTML = "";
        return;
    } else {
        var xmlhttp = new XMLHttpRequest();
        xmlhttp.onreadystatechange = function() {
            if (this.readyState == 4 && this.status == 200) {
document.getElementById("txtHint").innerHTML = this.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>
```

Code explanation:

First, check if the input field is empty (`str.length == 0`). If it is, clear the content of the `txtHint` placeholder and exit the function.

However, if the input field is not empty, do the following:

- Create an `XMLHttpRequest` object
- Create the function to be executed when the server response is ready
- Send the request off to a PHP file (`gethint.php`) on the server
- Notice that `q` parameter is added to the url (`gethint.php?q="+str`)
- And the `str` variable holds the content of the input field

## PHP Example - AJAX and XML

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 (this.readyState==4 && this.status==200) {

document.getElementById("txtHint").innerHTML=this.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="Bee Gees">Bee Gees</option>
<option value="Cat Stevens">Cat Stevens</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)

## PHP - AJAX and MySQL

---

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	Businessman

2	Lois	Griffin	40	Newport	P
3	Joseph	Swanson	39	Quahog	P
4	Glenn	Quagmire	41	Quahog	P

---

## Example Explained

In the example above, when a user selects a person in the dropdown list above, a function called "showUser()" is executed.

The function is triggered by the onchange event.

Here is the HTML code:

### Example

```
<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 (this.readyState == 4 && this.status == 200) {
```



```

document.getElementById("txtHint").innerHTML = this.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>

```

Code explanation:

First, check if person is selected. If no person is selected (`str == ""`), clear the content of `txtHint` and exit the function. If a person is selected, do 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 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:

```
<!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>

```

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