



Materia: Tecnologías para la Web.
Tema: Introducción a AJAX.

Introducción.

AJAX es significa Asíncrono JavaScript And XML. Es la técnica para crear aplicaciones web más rápidas e interactivas con XML, HTML, CSS y Java Script.

La aplicación web convencional transmite información hacia y desde el servidor usando solicitudes síncronas. Esto significa que se debe completar un formulario, presionar enviar y dirigirse a una nueva página con nueva información del servidor.

Con AJAX, cuando se presiona enviar, JavaScript hace una solicitud al servidor, interpreta los resultados y actualiza la pantalla actual.

Ejemplo PHP y AJAX

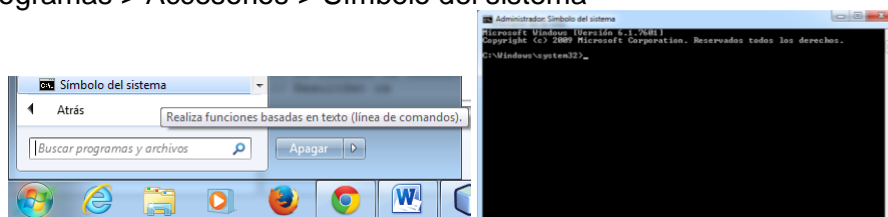
Para ilustrar claramente cuán fácil es acceder a la información desde una base de datos usando Ajax y PHP, vamos a construir consultas MySQL sobre la marcha y mostrar los resultados en "ajax.html". Pero antes de proceder, hagamos un trabajo de campo. Crea una tabla usando el siguiente comando.

PARTE I.

Creación de una base de datos.

Se puede trabajar con MySQL. Por ejemplo, en el menú Inicio se abre, como administrador, una ventana de comandos o Símbolo de sistema:

Inicio > Todos los programas > Accesorios > Símbolo del sistema



Para trabajar en MySQL en la ventana de comandos se ingresa la siguiente ruta:

```
C:\xampp\mysql\bin>
```

Para trabajar con una base de datos en la ruta indicada se ingresa la siguiente instrucción para MySQL:

```
mysql -h localhost -u root
```

Donde localhost puede ser 127.0.0.1, root es el usuario y no se posee contraseña en este caso. Como se indica en la figura:

```
C:\xampp\mysql\bin> mysql -h localhost -u root
```



```
ca. Administrador: Símbolo del sistema - mysql -h localhost -u root

C:\>cd xampp
C:\xampp>cd mysql
C:\xampp\mysql>cd bin
C:\xampp\mysql\bin>mysql -h localhost -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.6.16 MySQL Community Server (GPL)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql> _
```

En el cursor `mysql>_` se ingresa la instrucción `show` para mostrar las bases de datos disponibles, `create` para crear una nueva base de datos y `use` para utilizar esa base de datos. Cada instrucción termina con el carácter punto y coma, por ejemplo:

```
mysql> show databases;
mysql> create database prueba;
mysql> use prueba;
```

```
ca. Administrador: Símbolo del sistema - mysql -h localhost -u root

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| cdcol       |
| mysql       |
| performance_schema |
| phpmyadmin  |
| test        |
| webauth     |
+-----+
7 rows in set (0.00 sec)

mysql> create database prueba;
Query OK, 1 row affected (0.00 sec)

mysql> use prueba;
Database changed
```

Para crear una tabla **persona** en la base de datos **prueba** se utiliza la instrucción `create`, por ejemplo:

```
mysql> create table persona (
-> id smallint auto_increment,
-> primary key(id),
-> nombre varchar(60),
-> email varchar(60));
```

Para mostrar la creación de la tabla **persona** y su estructura, se ingresa la instrucción `describe`, por ejemplo:

```
mysql> describe persona;
```



```
ca. Administrador: Símbolo del sistema - mysql -h localhost -u root
mysql> create table persona<
-> id smallint auto_increment,
-> primary key(id),
-> nombre varchar(60),
-> email varchar(60);
Query OK, 0 rows affected (0.34 sec)

mysql> describe persona;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id    | smallint(6)   | NO   | PRI | NULL    | auto_increment |
| nombre | varchar(60)   | YES  |     | NULL    |                |
| email  | varchar(60)   | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)

mysql>
```

Para ingresar datos en la tabla **persona** se utiliza la instrucción **insert** y para ver el contenido de la tabla se utiliza la instrucción **select**, por ejemplo:

```
ca. Administrador: Símbolo del sistema - mysql -h localhost -u root
mysql> insert into persona (id, nombre, email)
-> values(1, 'Benito', 'bjuarez@hotmail.com');
Query OK, 1 row affected (0.02 sec)

mysql> insert into persona (id, nombre, email)
-> values(2, 'Pancho', 'pvilla@gmail.com');
Query OK, 1 row affected (0.01 sec)

mysql> insert into persona (id, nombre, email)
-> values(3, 'Emiliano', 'ezapata@yahoo.com.mx');
Query OK, 1 row affected (0.04 sec)

mysql> select * from persona;
+-----+-----+-----+
| id | nombre | email |
+-----+-----+-----+
| 1 | Benito | bjuarez@hotmail.com |
| 2 | Pancho | pvilla@gmail.com |
| 3 | Emiliano | ezapata@yahoo.com.mx |
+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

PARTE II.

a. Crear la siguiente tabla en la base de datos:

```
CREATE TABLE `ajax_example` (
  `nombre` varchar(50) NOT NULL,
  `edad` int(11) NOT NULL,
  `sexo` varchar(1) NOT NULL,
  `anio` int(11) NOT NULL,
  PRIMARY KEY (`nombre`)
)
```

b. Ingresar los siguientes datos en la tabla recién creada:

```
INSERT INTO `ajax_example` VALUES ('Benito', 120, 'm', 20);
INSERT INTO `ajax_example` VALUES ('Pancho', 75, 'm', 44);
INSERT INTO `ajax_example` VALUES ('Emiliano', 45, 'm', 87);
INSERT INTO `ajax_example` VALUES ('Asia', 22, 'f', 72);
INSERT INTO `ajax_example` VALUES ('Europa', 27, 'f', 0);
INSERT INTO `ajax_example` VALUES ('América', 35, 'f', 90);
```

c. Crear el archivo HTML:

```
<html>
  <body>
```



```
<script language = "javascript" type = "text/javascript">
  <!--
  function ajaxFunction(){
    var ajaxRequest;
    try {
      ajaxRequest = new XMLHttpRequest();
    }catch (e) {
      try {
        ajaxRequest = new ActiveXObject("Msxml2.XMLHTTP");
      }catch (e) {
        try{
          ajaxRequest = new ActiveXObject("Microsoft.XMLHTTP");
        }catch (e){
          alert("El navegador no funciona.");
          return false;
        }
      }
    }
    ajaxRequest.onreadystatechange = function(){
      if(ajaxRequest.readyState == 4){
        var ajaxDisplay = document.getElementById('ajaxDiv');
        ajaxDisplay.innerHTML = ajaxRequest.responseText;
      }
    }
    var edad = document.getElementById('edad').value;
    var anio = document.getElementById('anio').value;
    var sexo = document.getElementById('sexo').value;
    var queryString = "?edad=" + edad;
    queryString += "&anio=" + anio + "&sexo=" + sexo;
    ajaxRequest.open("GET", "ajax-ejemplo.php" + queryString, true);
    ajaxRequest.send(null);
  }
  <!-->
</script>
<form name = 'myForm'>
  Edad: <input type = 'text' id = 'age' /> <br />
  Año: <input type = 'text' id = 'edad' /> <br />
  Sexo: <select id = 'sexo'>
    <option value = "m">m</option>
    <option value = "f">f</option>
  </select>
  <input type = 'button' onclick = 'ajaxFunction()' value = 'Query MySQL' />
</form>
<div id = 'ajaxDiv'> Resultados: .....</div>
</body>
</html>
```

Lo anterior debe mostrar lo que se indica en la siguiente imagen:



Edad:

Año:

Sexo:

Resultados:

d. Crear el archivo PHP, ajax-ejemplo.php:

En el servidor se capturan los datos del navegador y se regresan a éste.

```
<?php
$dbhost = "localhost";
$dbuser = "dbusername";
$dbpass = "dbpassword";
$dbname = "dbname";
mysql_connect($dbhost, $dbuser, $dbpass);
mysql_select_db($dbname) or die(mysql_error());
$edad = $_GET['edad'];
$sexo = $_GET['sexo'];
$anio = $_GET['anio'];
$edad = mysql_real_escape_string($edad);
$sexo = mysql_real_escape_string($sexo);
$anio = mysql_real_escape_string($anio);
$query = "SELECT * FROM ajax_example WHERE sexo = '$sexo'";
if(is_numeric($edad))
    $query .= " AND edad <= $edad";
if(is_numeric($anio))
    $query .= " AND anio <= $anio";
$qry_result = mysql_query($query) or die(mysql_error());
$display_string = "<table>";
$display_string .= "<tr>";
$display_string .= "<th>Nombre</th>";
$display_string .= "<th>Edad</th>";
$display_string .= "<th>Sexo</th>";
$display_string .= "<th>Anio</th>";
$display_string .= "</tr>";
while($row = mysql_fetch_array($qry_result)) {
    $display_string .= "<tr>";
    $display_string .= "<td>$row[nombre]</td>";
    $display_string .= "<td>$row[edad]</td>";
    $display_string .= "<td>$row[sexo]</td>";
    $display_string .= "<td>$row[anio]</td>";
    $display_string .= "</tr>";
}
echo "Consulta: " . $query . "<br />";
$display_string .= "</table>";
echo $display_string;
?>
```

Probar una vez más la aplicación. Comentar los resultados mostrados cuando se ingresan los datos.



Edad:

Año:

Sexo:

Resultados:

PARTE III.

Análisis de un documento XML.

El módulo SimpleXML de PHP convierte un documento XML en un objeto que proporciona acceso estructurado al XML. Para crear un objeto SimpleXML a partir de un documento XML almacenado en una cadena, se pasa la cadena al método `simplexml_load_string()` que regresa un objeto SimpleXML.

Por ejemplo:

```
<html>
  <body>
    <?php
      $note=<<<XML
      <note>
        <to>Gopal K Verma</to>
        <from>Sairamkrishna</from>
        <heading>Project submission</heading>
        <body>
          Lea cuidadosamente.
        </body>
      </note>
      XML;
      $xml=simplexml_load_string($note);
      print_r($xml);
    ?>
  </body>
</html>
```

Probar la ejecución. Se muestra el siguiente resultado:

```
SimpleXMLElement Object([to]=>Gopal K Verma[form]=>Sairamkrishna[heading]
=>Project submission[body]=>Lea cuidadosamente)
```

Otro ejemplo:

```
<?php
  $channel = array('title' => "Hola!", 'link' => 'http://saludo.ejemplo.com/',
                  'description' => 'Buenos días.');
```

```
print "<channel>\n";
foreach ($channel as $element => $content) {
  print " <$element>";
  print htmlentities($content);
  print "</$element>\n";
}
print "</channel>";
?>
```

Se producirá el siguiente resultado:



```
<channel>
  <title>Hola!</title>
  <link>http://saludo.ejemplo.com/</link>
  <description>Buenos días.</description>
</channel>
```

NOTA: Generar un documento con el desarrollo de los ejercicios anteriores. Anexar el documento a la carpeta del proyecto con todos los archivos obtenidos de la aplicación en un archivo **NombreAJAX.zip**.