



Práctica 14 – AJAX: HTTP y el Objeto XMLHttpRequest

INSTRUCCIONES: Apóyate del documento adjunto que lleva por título “*HTTP y el Objeto XMLHttpRequest*” verificando la información que se te pide para entender mejor el siguiente código.

Prueba en *Firefox* y *Chrome*.

```
1. <!DOCTYPE html>
2. <html>
3.   <head>
4.     <meta charset="UTF-8"/>
5.     <title>Práctica I - AJAX </title>
6.   </head>
7.   <script type="text/javascript">
8.     function loadDoc(opc) {
9.       var xhttp = new XMLHttpRequest();
10.      xhttp.onreadystatechange = function() {
11.        if (this.readyState == 4 && this.status == 200) {
12.          switch(opc) {
13.            case 1: document.getElementById("demo1").innerHTML =
14.              "<p>He AQUÍ el Primer NUEVO contenido</p>";
15.              break;
16.            case 2: document.getElementById("demo2").innerHTML =
17.              "<p>He AQUÍ el Segundo NUEVO contenido --> "+
18.              document.forms[0]["nombre"].value+"</p>";
19.              break;
20.          }
21.        }
22.      };
23.      xhttp.open("GET", " ", true);
24.      xhttp.send(null);
25.    }
26.  </script>
27.  <body>
28.    <h3>Ejemplo 1 del uso del Objeto XMLHttpRequest</h3>
29.    <button type="button" onclick="loadDoc(1)">Resultado</button>
30.    <div id="demo1">
31.      <p>Mi informacion cambiará sin recargar la página completa...</p>
32.    </div>
33.    <h3>Ejemplo 2 del uso del Objeto XMLHttpRequest</h3>
34.    <form>
35.      <label>Entrada: </label><input type="text" name="nombre"/>
36.      <button type="button" onclick="loadDoc(2)">Resultado</button>
37.    </form>
38.    <div id="demo2">
39.      <p>Mi informacion cambiará sin recargar la página completa...</p>
40.    </div>
41.  </body>
42. </html>
```

Revisa la descripción de los siguientes elementos en el documento adjunto:

- Instanciación de un objeto XMLHttpRequest según el navegador y su versión (*Tabla 3*)
- Propiedades del objeto XMLHttpRequest (*Tabla 4*)
- Métodos del objeto XMLHttpRequest (*Tabla 5*)



1. Implementa una función en JavaScript de nombre *getXMLHttpRequestOBJ()* que permita crear un objeto XMLHttpRequest para cualquier navegador (compatible), y que devuelva dicho objeto.
 - Utiliza esta función en lugar de la sentencia “new XMLHttpRequest()”.
 - Verifica que funcione correctamente.

NOTA: este será el motor que uses de aquí en adelante siempre que programes aplicaciones con AJAX puro.

Referencias de apoyo:

A. Métodos, propiedades y su respectiva descripción de HTML DOM en:
https://www.w3schools.com/jsref/dom_obj_document.asp

B. Métodos, propiedades y ejemplos del Objeto XMLHttpRequest en:
https://www.w3schools.com/xml/ajax_xmlhttprequest_create.asp

2. Revisa y prueba el código adjunto, notarás que el funcionamiento es similar al código ejemplo de la *Práctica 4*, pero con la diferencia de que esta vez se utiliza AJAX.

Los archivos adjuntos son:

- **consultarBD.php** – se encarga de realizar la conexión a base de datos, hacer la consulta y devolver un *string* en JSON que contiene la información de la consulta.
- **busquedaGET.php** – formulario simple que utiliza el objeto XMLHttpRequest para establecer una comunicación asíncrona con el servidor y así obtener, mediante el método GET, los datos de la persona especificada por nombre.
- **busquedaPOST.php** – formulario simple que utiliza el objeto XMLHttpRequest para establecer una comunicación asíncrona con el servidor y así obtener, mediante el método POST, los datos de la persona especificada por nombre.

Simplemente configura la conexión para que funcione con la base de datos de la *Práctica 4* (“*tecnologiasweb_p04*”).