

5 - Objeto XMLHttpRequest

El objeto XMLHttpRequest es un elemento fundamental para la comunicación asincrónica con el servidor. Este objeto nos permite enviar y recibir información en formato XML y en general en cualquier formato (como vimos en el ejercicio anterior retornando un trozo de archivo HTML)

La creación de un objeto de esta clase se hace con la siguiente sintaxis (debemos definir una variable llamada conexion1):

```
conexion1=new XMLHttpRequest();
```

Las **propiedades** principales del objeto XMLHttpRequest son:

- *onreadystatechange* Almacena el nombre de la función que se ejecutará cuando el objeto XMLHttpRequest cambie de estado.
- *readyState* Almacena el estado del requerimiento hecho al servidor, pudiendo ser:
 - 0 No inicializado (el método open no a sido llamado)
 - 1 Cargando (se llamó al método open)
 - 2 Cargado (se llamó al método send y ya tenemos la cabecera de la petición HTTP y el status)
 - 3 Interactivo (la propiedad responseText tiene datos parciales)
 - 4 Completado (la propiedad responseText tiene todos los datos pedidos al servidor)
- *responseText* Almacena el string devuelto por el servidor, luego de haber hecho una petición.
- *responseXML* Similar a la anterior (responseText) con la diferencia que el string devuelto por el servidor se encuentra en formato XML.

Los **métodos** principales del objeto XMLHttpRequest son:

- *open* Abre un requerimiento HTTP al servidor.
- *send* Envía el requerimiento al servidor.

Confeccionaremos otro problema para fijar conceptos vistos hasta el momento.

Confeccionar una página que muestre una imagen y permita calificarla con un valor entre 1 y 10. Permitir ingresar el nombre del visitante. Disponer de un control de tipo select para seleccionar el valor. Luego al presionar un botón enviar el valor seleccionado utilizando el objeto XMLHttpRequest al servidor donde almacenaremos en un archivo de texto el nombre del visitante y el puntaje. Retornar luego todos los votos hasta el momento. Actualizaremos la página HTML con todos los nombres y votos hasta el momento.

El archivo HTML es:

pagina1.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Problema</title>
  <script src="funciones.js"></script>
</head>

<body>
  <h1>Vote esta foto</h1>
  <p></p>
  Nombre:<input type="text" id="nombre" size="20">
  <br>
  Voto:
  <select id="voto">
    <option value="0" selected>seleccione</option>
    <option value="1">1</option>
    <option value="2">2</option>
    <option value="3">3</option>
    <option value="4">4</option>
    <option value="5">5</option>
    <option value="6">6</option>
    <option value="7">7</option>
    <option value="8">8</option>
    <option value="9">9</option>
    <option value="10">10</option>
  </select>
  <br>
  <input type="button" id="boton1" value="votar">
  <div id="resultados"></div>
</body>
</html>
```

Lo primero que podemos observar es que no utilizaremos una hoja de estilo para reducir la complejidad del problema y concentrarnos en la lógica.

Si vemos utilizamos controles de tipo input, select y button, pero no disponemos ningún formulario. Esto se debe a que los datos ingresados se enviarán en forma asíncrona mediante el objeto XMLHttpRequest.

Otro punto a destacar que a cada control le definimos la propiedad id, esta es de suma importancia para poder accederla desde JavaScript. No definimos la propiedad name ya que no se enviarán los datos por medio de formulario.

Nuestro archivo con las funciones JavaScript es:

funciones.js

```

addEventListener('load',inicializarEventos,false);

function inicializarEventos()
{
    var ob=document.getElementById('boton1');
    ob.addEventListener('click',presionBoton,false);
}

function presionBoton(e)
{
    var ob1=document.getElementById('voto');
    var ob2=document.getElementById('nombre');
    cargarVoto(ob1.value,ob2.value);
}

var conexion1;
function cargarVoto(voto,nom)
{
    conexion1=new XMLHttpRequest();
    conexion1.onreadystatechange = procesarEventos;
    conexion1.open('GET','pagina1.php?puntaje='+voto+'&nombre='+nom, true);
    conexion1.send();
}

function procesarEventos()
{
    var resultados = document.getElementById("resultados");
    if(conexion1.readyState == 4)
    {
        resultados.innerHTML = conexion1.responseText;
    }
    else
    {
        resultados.innerHTML = 'Cargando...';
    }
}

```

Como podemos observar tiene mucho en común con el primer ejemplo de AJAX que habíamos desarrollado.

Lo primero que hacemos es inicializar el evento load con la función inicializarEventos, en esta inicializamos el evento click del único botón que contiene la página:

```

addEventListener('load',inicializarEventos,false);

function inicializarEventos()
{
    var ob=document.getElementById('boton1');
    ob.addEventListener('click',presionBoton,false);
}

```

Cuando se presiona el botón se dispara la función presionBoton, donde obtenemos la referencia a los dos controles (select y text) que tienen almacenados los valores. Llamamos finalmente a la función cargarVoto:

```

function presionBoton(e)
{
    var ob1=document.getElementById('voto');
    var ob2=document.getElementById('nombre');
    cargarVoto(ob1.value,ob2.value);
}

```

```
}
```

La función cargarVoto recibe como parámetro el valor del voto y el nombre del visitante, seguidamente creamos un objeto de la clase XMLHttpRequest.

Por último inicializamos la propiedad onreadystatechange y llamamos a los métodos open y send. En el método open pasamos los dos datos en la cabecera de la petición de página.

```
var conexion1;  
function cargarVoto(voto,nom)  
{  
    conexion1=new XMLHttpRequest();  
    conexion1.onreadystatechange = procesarEventos;  
    conexion1.open('GET','pagina1.php?puntaje='+voto+'&nombre='+nom, true);  
    conexion1.send();  
}
```

Nos queda la función procesarEventos, que cuando la propiedad readyState del objeto XMLHttpRequest tiene un valor 4 (proceso completado) recupera el valor de la propiedad.responseText con la información que se retornó desde el servidor:

```
function procesarEventos()  
{  
    var resultados = document.getElementById("resultados");  
    if(conexion1.readyState == 4)  
    {  
        resultados.innerHTML = conexion1.responseText;  
    }  
    else  
    {  
        resultados.innerHTML = 'Cargando...';  
    }  
}
```

El archivo PHP que se ejecuta en el servidor es el siguiente:

pagina1.php

```
<?php  
header('Content-Type: text/html; charset=utf-8');  
$ar=fopen("puntaje.txt","a") or  
    die("No se pudo abrir el archivo");  
fputs($ar,"Nombre:".$_REQUEST['nombre']."<br>");  
fputs($ar,"Voto:".$_REQUEST['puntaje']."<br><br>");  
fclose($ar);  
$ar=fopen("puntaje.txt","r") or  
    die("No se pudo abrir el archivo");  
while (!feof($ar))  
{  
    $linea=fgets($ar);  
    echo $linea;  
}  
fclose($ar);  
?>?>
```

Lo primero que hacemos es abrir el archivo para agregar datos, es decir no borramos los votos existentes (puede probar de cambiar "a" de append por "w" que crea el archivo):

```
$ar=fopen("puntaje.txt","a") or
```

```
die("No se pudo abrir el archivo");
```

Luego recuperamos los parámetros que llegan a la página y los grabamos:

```
fputs($ar,"Nombre:".$_REQUEST['nombre']."<br>");  
fputs($ar,"Voto:".$_REQUEST['puntaje']."<br><br>");
```

Cerramos y abrimos nuevamente el archivo, pero ahora con el objetivo de leerlo:

```
fclose($ar);  
$ar=fopen("puntaje.txt","r") or  
die("No se pudo abrir el archivo");
```

Por último generamos el archivo a retornar al navegador:

```
while (!feof($ar))  
{  
    $linea=fgets($ar);  
    echo $linea;  
}
```

Con este segundo ejemplo debemos poder identificar que partes son comunes al problema anterior.

Problema resuelto.

Confeccionar una página que muestre una imagen y permita calificarla con un valor entre 1 y 10. Permitir ingresar el nombre del visitante. Disponer de un control de tipo select para seleccionar el valor. Luego al presionar un botón enviar el valor seleccionado utilizando el objeto XMLHttpRequest al servidor donde almacenaremos en un archivo de texto el nombre del visitante y el puntaje. Retornar luego todos los votos hasta el momento. Actualizaremos la página HTML con todos los nombres y votos hasta el momento.

pagina.html

```
<!DOCTYPE html>  
<html>  
<head>  
    <meta charset="utf-8">  
    <title>Problema</title>  
    <script src="funciones.js"></script>  
</head>  
  
<body>  
    <h1>Vote esta foto</h1>  
    <p></p>  
    Nombre:<input type="text" id="nombre" size="20">  
    <br>  
    Voto:  
    <select id="voto">
```

[Ejecutar ejemplo](#)

funciones.js

```
addEventListener('load',inicializarEventos,false);

function inicializarEventos()
{
    var ob=document.getElementById('boton1');
    ob.addEventListener('click',presionBoton,false);
}

function presionBoton(e)
{
    var ob1=document.getElementById('voto');
    var ob2=document.getElementById('nombre');
    cargarVoto(ob1.value,ob2.value);
}
```

pagina1.php

```
<?php
header('Content-Type: text/html; charset=utf-8');
$ar=fopen("puntaje.txt","a") or
    die("No se pudo abrir el archivo");
fputs($ar,"Nombre:".$_REQUEST['nombre']."<br>");
fputs($ar,"Voto:".$_REQUEST['puntaje']."<br><br>");
fclose($ar);
$ar=fopen("puntaje.txt","r") or
    die("No se pudo abrir el archivo");
while (!feof($ar))
{
    $linea=fgets($ar);
    echo $linea;
}
fclose($ar);
```

Problema propuesto.

Confeccionar el mismo problema es decir una página que muestre una imagen y permita calificarla con un valor entre 1 y 10. Permitir ingresar el nombre del visitante. Disponer de un control de tipo select para seleccionar el valor. Resolver el problema utilizando un formulario tradicional sin comunicación asincrónica. Observar las diferencias con la metodología de utilizar AJAX.

[Ver solución](#)

[Retornar](#)