7 - Pasando datos al servidor por el método POST.

Podemos enviar los datos por el método GET, como hemos visto hasta ahora, pero también podemos enviar los datos por el método POST.

El método POST se utiliza cuando hay que enviar mucha información al servidor.

Hay varios puntos a tener en cuenta para cuando codificamos los datos para el envío por el método POST:

1. Cuando llamamos al método open del objeto XMLHttpRequest como primer parámetro indicamos el string 'POST'

```
conexion1.open('POST','pagina1.php', true);
```

2. Llamamos al método setRequestHeader indicando que los datos a enviarse están codificados como un formulario.

```
conexion1.setRequestHeader("Content-Type","application/x-www-form-urlencoded");
```

3. Llamamos al método send del objeto XMLHttpRequest pasando los datos:

```
conexion1.send("nombre=juan&clave=z80");
```

Podemos concatenar datos recuperados de un formulario y enviarlos a través del método send.

Confeccionaremos un problema completo para probar la funcionalidad del envío de datos por el método POST.

Haremos una serie de páginas que nos permitan enviar los datos cargados en un formulario al servidor en forma asincrónica y por lo tanto no tendremos que refrescar la página luego de enviado los datos. El formulario solicitará que ingrese el nombre y comentarios del sitio.

El archivo HTML es el siguiente (pagina1.html):

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

<body>
  <form action="pagina1.php" method="post" id="formulario">
    Nombre:<input type="text" name="nombre" id="nombre" size="20"><br>
    Comentarios:<br>
    <textarea name="comentarios" id="comentarios" rows="10" cols="50"></textarea><br>
    <input type="submit" value="Enviar" id="enviar">
    <div id="resultados"></div>
    <a href="comentarios.txt">Ver resultados</a>
  </form>
</body>
</html>
```

Este archivo HTML no tiene nada de especial, básicamente es un formulario con un control de tipo input, otro de tipo textarea y finalmente un botón para el envío de los datos al servidor. Además contiene un div vacío para mostrar mensajes sobre el resultado del envío de los datos al servidor.

Disponemos un hipervínculo al archivo de texto que almacenará los datos cargados, esto con el objetivo de poder controlar si los datos realmente se cargaron en el servidor. En un problema más grande seguramente los datos los almacenaremos en una base de datos.

El archivo funciones.js es:

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

function inicializarEventos()
{
  var ref=document.getElementById('formulario');
  ref.addEventListener('submit',enviarDatos,false);
}

function enviarDatos(e)
{
  e.preventDefault();
  enviarFormulario();
}


function retornarDatos()
{
  var cad='';
  var nom=document.getElementById('nombre').value;
  var com=document.getElementById('comentarios').value;
  cad='nombre='+encodeURIComponent(nom)+'&comentarios='+encodeURIComponent(com);
  return cad;
}

var conexion1;
function enviarFormulario()
{
  conexion1=new XMLHttpRequest();
  conexion1.onreadystatechange = procesarEventos;
  conexion1.open('POST','pagina1.php', true);
  conexion1.setRequestHeader("Content-Type","application/x-www-form-urlencoded");
  conexion1.send(retornarDatos());
}

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

Lo primero que hacemos en la función inicializarEventos es:

```
var ref=document.getElementById('formulario');
ref.addEventListener('submit',enviarDatos,false);
```

Obtener la referencia del formulario e inicializar el evento submit para poder capturar el momento en que los datos se enviarán al servidor. Es decir la función enviarDatos se ejecutará cuando el operador presione el botón de tipo submit.

La función enviarFormulario:

```
function enviarFormulario()
{
  conexion1=new XMLHttpRequest();
  conexion1.onreadystatechange = procesarEventos;
  conexion1.open('POST','pagina1.php', true);
  conexion1.setRequestHeader("Content-Type","application/x-www-form-urlencoded");
  conexion1.send(retornarDatos());
}
```

Inicializa el objeto XMLHttpRequest indicando en el método open que los datos se enviarán mediante el comando POST.

Luego llamamos al método setRequestHeader indicando el tipo de contenido a enviar al servidor.

Finalmente llamamos al método send con los datos. Estos datos los recuperamos del formulario llamando a la función retornarDatos() que a continuación la explicaremos.

```
function retornarDatos()
{
  var cad='';
  var nom=document.getElementById('nombre').value;
  var com=document.getElementById('comentarios').value;
  cad='nombre='+encodeURIComponent(nom)+'&comentarios='+encodeURIComponent(com);
  return cad;
}
```

Obtenemos el valor contenido en cada control (input y textarea) y luego concatenamos nombre del dato y valor separándolos por el caracter ámpersand.

Otra cosa importante es llamar a la función JavaScript encodeURIComponent para codificar los datos ingresados por el visitante y se puedan enviar correctamente al servidor.

La función procesarEventos:

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

Dijimos ya que esta función se ejecuta cada vez que cambia el estado del objeto XMLHttpRequest. Cuando retorna un 4 significa que el envío de datos se efectuó en forma correcta, mostramos en ese momento el mensaje 'Gracias'.

El último archivo: pagina1.php

```
<?php
header('Content-Type: text/html; charset=utf-8');
$ar=fopen("comentarios.txt","a") or
  die("No se pudo abrir el archivo");
fputs($ar,"Nombre:".$_REQUEST['nombre']."\n");
fputs($ar,"Comentarios:".$_REQUEST['comentarios']."\n\n");
fclose($ar);
?>
```

Simplemente rescatamos los datos enviados desde el navegador y procedemos a grabarlos en el archivo de texto.

Problema resuelto.

Enviar los datos cargados en un formulario al servidor en forma asincrónica.
El formulario debe solicitar el ingreso del nombre y comentarios del sitio.

pagina1.html

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

<body>
  <form action="pagina1.php" method="post" id="formulario">
    Nombre:<input type="text" name="nombre" id="nombre" size="20"><br>
    Comentarios:<br>
    <textarea name="comentarios" id="comentarios" rows="10" cols="50"></textarea><br>
    <input type="submit" value="Enviar" id="enviar">
    <div id="resultados"></div>
```

Ejecutar ejemplo

funciones.js

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

function inicializarEventos()
{
  var ref=document.getElementById('formulario');
  ref.addEventListener('submit',enviarDatos,false);
}

function enviarDatos(e)
{
  e.preventDefault();
  enviarFormulario();
}
```

pagina1.php

```
<?php
header('Content-Type: text/html; charset=ISO-8859-1');
$ar=fopen("comentarios.txt","a") or
  die("No se pudo abrir el archivo");
fputs($ar,"Nombre:".$_REQUEST['nombre']."\n");
fputs($ar,"Comentarios:".$_REQUEST['comentarios']."\n\n");
fclose($ar);
?>
```

Retornar