6 - Pasando datos al servidor por el método GET.

Para indicar cual es el método de envío de los datos al servidor lo hacemos en el primer parámetro del método open del objeto XMLHttpRequest:

```
conexion1.open('GET','pagina1.php?puntaje='+voto+'&nombre='+nom, true);
```

En este ejemplo vemos como indicamos que el envío de los datos se hace por el método GET. Si lo hacemos de esta forma tenemos que tener mucho cuidado en la codificación del segundo parámetro del método open donde indicamos el nombre de la página a pedir.

Seguido al nombre de la página debe ir el signo de interrogación, el nombre del parámetro, luego un igual y el valor del parámetro. En caso de haber más de un parámetro debemos separarlos mediante el caracter ampersand.

Por último el tercer parámetro del método open normalmente se pasa el valor true indicando que el requerimiento de la página es asíncrona (esto permite al visitante continuar interactuando con la página sin que se congele hasta llegar la solicitud)

Confeccionaremos un problema similar al anterior, es decir que nos permita calificar una foto con un valor de 1 al 10. La diferencia es que la calificación será por medio de una lista de hipervínculos del 1 al 10. Además haremos que la barra de selección de la calificación cambie de color cuando ingresamos con el mouse. Veremos que la mayor dificultad se encuentra en la codificación de esta última característica más que en la comunicación asincrónica.

El archivo HTML es (pagina1.html):

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Problema</title>
  <script src="funciones.js"></script>
  <link rel="StyleSheet" href="estilos.css" type="text/css">
</head>
<body>
  <h1>Vote esta foto</h1>
  <p>
  <img src="../foto1.jpg" alt="cuadro sobre geometria generativa">
  </p>
  <ul class="voto" id="votofoto1">
    <li><a href="pagina1.php?voto=1">1</a></li>
    <li><a href="pagina1.php?voto=2">2</a></li>
    <li><a href="pagina1.php?voto=3">3</a></li>
    <li><a href="pagina1.php?voto=4">4</a></li>
    <li><a href="pagina1.php?voto=5">5</a></li>
    <li><a href="pagina1.php?voto=6">6</a></li>
    <li><a href="pagina1.php?voto=7">7</a></li>
    <li><a href="pagina1.php?voto=8">8</a></li>
    <li><a href="pagina1.php?voto=9">9</a></li>
    <li><a href="pagina1.php?voto=10">10</a></li>
  </ul>
  <br>
  <div id="resultados"></div>
  <a href="votos.txt">Ver resultados</a>
</body>
</html>
```

La foto se encuentra en el directorio inmediatamente superior por eso indicamos el ../ :

```
<img src="../foto1.jpg" alt="cuadro sobre geometria generativa">
```

Para no complicar más el problema el archivo de resultados se almacena en un archivo de texto llamado votos.txt (al presionar el hipervínculo procedemos a mostrarlo en el navegador, esto nos podrá permitir verificar si realmente el voto se registró en el servidor):

```
<a href="votos.txt">Ver resultados</a>
```

Cada hipervínculo pasa como parámetro la calificación respectiva:

```
<li><a href="pagina1.php?voto=1">1</a></li>
<li><a href="pagina1.php?voto=2">2</a></li>
<li><a href="pagina1.php?voto=3">3</a></li>
```

La hoja de estilo queda definida como sigue (estilos.css):

```
.voto {
  padding:0px;
}
.voto a {
  float:left;
  width:15px;
  text-decoration:none;
  text-align:center;
  color:#f00;
  background-color:#f7f8e8;
  border-right:1px solid white;
  font-size:13px;
}

.voto li {
  display:inline;
}
```

Como la lista de hipervínculos los queremos todos en la misma línea inicializamos la propiedad display con el valor inline:

```
.voto li {
  display:inline;
}
```

Luego donde se encuentra la complejidad mayor es en funciones.js:

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

function inicializarEventos()
{
  var ref=document.getElementById('votofoto1');
  var vec= ref.getElementsByTagName('li');
  var vec2=ref.getElementsByTagName('a');
  for(var f=0;f<vec2.length;f++)
  {
    vec[f].addEventListener('mouseover',entrar,false);
    vec[f].addEventListener('mouseout',salir,false);
    vec2[f].addEventListener('click',presionBoton,false);
  }
}

function entrar(e)
{
  var ref=e.target;
  var ob=document.getElementById('votofoto1');
  var vec=ob.getElementsByTagName('li');
  for(var f=0;f<ref.firstChild.nodeValue;f++)
  {
    vec[f].firstChild.style.background='#f00';
    vec[f].firstChild.style.color='#fff';
  }
}
```

```
function salir(e)
{
  var ref=e.target;
  var ob=document.getElementById('votofoto1');
  var vec=ob.getElementsByTagName('li');
  for(var f=0;f<ref.firstChild.nodeValue;f++)
  {
    vec[f].firstChild.style.background='#f7f8e8';
    vec[f].firstChild.style.color='#f00';
  }
}


function presionBoton(e)
{
  e.preventDefault();
  var ref=e.target;
  cargarVoto(ref.firstChild.nodeValue);
}


var conexion1;
function cargarVoto(voto)
{
  conexion1=new XMLHttpRequest();
  conexion1.onreadystatechange = procesarEventos;
  var aleatorio=Math.random();
  conexion1.open('GET','pagina1.php?voto='+voto+"&aleatorio="+aleatorio, true);
  conexion1.send();
}

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

La primera función que se ejecuta cuando se termina de cargar por completo la página es inicializarEventos:

Obtenemos la referencia al div que contiene la foto:

```
var ref=document.getElementById('votofoto1');
```

Luego obtenemos a partir de la referencia de dicho div todos los elementos li (list item) y a (anchor):

```
  var vec= ref.getElementsByTagName('li');
  var vec2=ref.getElementsByTagName('a');
```

Pasamos a la asignación de eventos a estos elementos dentro de un for, a los elementos de tipo li le definimos los eventos mouseover y mouseout, luego a los elementos a le definimos el evento click:

```
  for(var f=0;f<vec2.length;f++)
  {
    vec[f].addEventListener('mouseover',entrar,false);
    vec[f].addEventListener('mouseout',salir,false);
    vec2[f].addEventListener('click',presionBoton,false);
  }
```

La función entrar se ejecuta cuando la flecha del mouse se dispone dentro de algún elemento de la lista. Luego dentro de un for cambiamos el color de todos los elementos menores e incluyendo al que emitió el evento:

```
function entrar(e)
{
  var ref=e.target;
  var ob=document.getElementById('votofoto1');
  var vec=ob.getElementsByTagName('li');
  for(var f=0;f<ref.firstChild.nodeValue;f++)
  {
    vec[f].firstChild.style.background='#f00';
    vec[f].firstChild.style.color='#fff';
  }
}
```

El algoritmo es similar cuando sacamos la flecha del mouse de un elemento de tipo li:

```
function salir(e)
{
  var ref=e.target;
  var ob=document.getElementById('votofoto1');
  var vec=ob.getElementsByTagName('li');
  for(var f=0;f<ref.firstChild.nodeValue;f++)
  {
    vec[f].firstChild.style.background='#f7f8e8';
    vec[f].firstChild.style.color='#f00';
  }
}
```

La función presionBoton se ejecuta cuando alguno de los elementos a (anchor) es presionado:

```
function presionBoton(e)
{
  e.preventDefault();
  var ref=e.target;
  cargarVoto(ref.firstChild.nodeValue);
}
```

En la función presionBoton desactivamos la acción por defecto de un elemento "a" llamando a la función e.preventDefault().

Procedemos luego a llamar a la función cargarVoto pasando como referencia el valor contenido en el elemento "a".

Veamos ahora la función cargarVoto:

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

Creamos un objeto de la clase XMLHttpRequest. Inicializamos la propiedad onreadystatechage con el nombre de la función que se disparará cada vez que el objeto XMLHttpRequest cambie de estado.

Ahora llamamos a la función open del objeto XMLHttpRequest indicando:

```
  var aleatorio=Math.random();
  conexion1.open('GET','pagina1.php?voto='+voto+"&aleatorio="+aleatorio, true);
```

El método de conexión utilizado, en este caso es 'GET', como segundo parámetro la página que procesará el dato enviado al servidor (en nuestro caso enviamos el valor entero comprendido entre 1 y 10, más precisamente el que seleccionó el operador) El dato a enviar se llama voto y almacena el contenido de la variable voto que llega como parámetro a esta función.

Otra cosa que vamos a incorporar es el envío de un valor aleatorio. Esto es necesario si el navegador está configurado que recupere las páginas del cache. Imaginemos que votamos y asignamos el valor 10, luego si volvemos a seleccionar el 10 puede recuperar la pagina1.php?voto=10 del cache y no actualizar el archivo de texto con el voto seleccionado. La solución mas sencilla es enviar un parámetro con un valor aleatorio, lo cual el navegador interpretará que se trata de otra página.

Recordemos que el tercer parámetro de la función open indica que el proceso se efectúe en forma asincrónica (esto si le pasamos el valor true)

La función procesarEventos obtiene una referencia al div llamado detalles y mustra inicialmente el texto: 'Cargando...', luego cuando el servidor informa que los datos se registraron pasa a mostrar el texto: 'Gracias'.

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

Por último la página que se ejecuta en el servidor (pagina1.php) tiene por objetivo registrar en un archivo de texto el valor seleccionado por el visitante:

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

Problema resuelto.

Confeccionar un problema que nos permita calificar una foto con un valor de 1 al 10. La calificación será por medio de una lista de hipervínculos del 1 al 10. Además hacer que la barra de selección de la calificación cambie de color cuando ingresamos con el mouse.

pagina1.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Problema</title>
  <script src="funciones.js"></script>
  <link rel="StyleSheet" href="estilos.css" type="text/css">
</head>
<body>
  <h1>Vote esta foto</h1>
  <p>
  <img src="../foto1.jpg" alt="cuadro sobre geometria generativa">
  </p>
  <ul class="voto" id="votofoto1">
    <li><a href="pagina1.php?voto=1">1</a></li>
```

Ejecutar ejemplo

estilos.css

```
.voto {
  padding:0px;
}
.voto a {
  float:left;
  width:15px;
  text-decoration:none;
  text-align:center;
  color:#f00;
  background-color:#f7f8e8;
  border-right:1px solid white;
  font-size:13px;
}

.voto li {
```

funciones.js

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

function inicializarEventos()
{
  var ref=document.getElementById('votofoto1');
  var vec= ref.getElementsByTagName('li');
  var vec2=ref.getElementsByTagName('a');
  for(var f=0;f<vec2.length;f++)
  {
    vec[f].addEventListener('mouseover',entrar,false);
    vec[f].addEventListener('mouseout',salir,false);
    vec2[f].addEventListener('click',presionBoton,false);
  }
}
```

pagina1.php

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

Problema propuesto.

Confeccionar el mismo problema de calificar una foto con un valor de 1 al 10 utilizando una lista de hipervínculos del 1 al 10. Además hacer que la barra de selección de la calificación cambie de color cuando ingresamos con el mouse.

En vez de utilizar un valor aleatorio emplear la hora actual para que el navegador siempre pida la página al servidor y no rescate la misma de la cache.

Ver solución

Retornar