10 - Propiedades onreadystatechange y readystate.

Dos propiedades fundamentales del objeto XMLHttpRequest son onreadystatechange y readyState.

El objetivo de cada una es:

- *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)

Normalmente cuando creamos un objeto de la clase XMLHttpRequest inicializamos la propiedad onreadystatechange con el nombre de la función que procesará los datos enviados por el servidor:

```
function cargarHoroscopo(url)
{
 if(url=='')
 {
  return;
 }
 conexion1=crearXMLHttpRequest();
 conexion1.onreadystatechange = procesarEventos;
 conexion1.open("GET", url, true);
 conexion1.send(null);
}
```

Por otro lado dentro de la función que previamente fue asignada a la propiedad onreadystatechange verificamos el estado de la propiedad readyState:

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

Para ver el paso de estados del objeto XMLHttpRequest implementaremos una aplicación que calcule el cuadrado de un número que ingresamos por teclado (el cálculo se efectuará en el servidor) Y además mostraremos mediante un alert el estado actual de la propiedad readyState.

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">
    Ingrese nro:<input type="text" name="nro" id="nro" size="10"><br>
    <input type="submit" value="Calcular el cuadrado" id="enviar">
  </form>
  <div id="resultados"></div>
</body>
</html>
```

funciones.js

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

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

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

var conexion1;
function enviarFormulario()
{
  conexion1=new XMLHttpRequest();
  conexion1.onreadystatechange = procesarEventos;
  var num=document.getElementById('nro').value;
  alert('Valor de la propiedad readyState:'+conexion1.readyState);
  conexion1.open('GET','pagina1.php?numero='+num, true);
  conexion1.send();
}

function procesarEventos()
{
  alert('Valor de la propiedad readyState:'+conexion1.readyState);
  var resultados = document.getElementById("resultados");
  if(conexion1.readyState == 4)
  {
    resultados.innerHTML = conexion1.responseText;
  }
  else
    if (conexion1.readyState==1 || conexion1.readyState==2 || conexion1.readyState==3)
    {
      resultados.innerHTML = 'Procesando...';
    }
}
```

Como podemos observar si accedemos a la propiedad readyState antes de llamar a los métodos open y send la misma almacena el valor cero:

```
conexion1=new XMLHttpRequest();
conexion1.onreadystatechange = procesarEventos;
var num=document.getElementById('nro').value;
alert('Valor de la propiedad readyState:'+conexion1.readyState); //Muestra cero
```

Cuando llamamos al método open se ejecuta por primera vez la función que inicializamos en la propiedad onreadystatechange, al mostrar la propiedad readyState veremos que almacena un uno:

```
conexion1.open('GET','pagina1.php?numero='+num, true);
```

Cuando llamamos al método send se ejecuta por segunda vez la función que inicializamos en la propiedad

onreadystatechange, al mostrar la propiedad readyState veremos que continua con el valor uno.

Luego la función:

```
function procesarEventos()
{
  alert('Valor de la propiedad readyState:'+conexion1.readyState);
  var resultados = document.getElementById("resultados");
  if(conexion1.readyState == 4)
  {
    resultados.innerHTML = conexion1.responseText;
  }
  else
    if (conexion1.readyState==1 || conexion1.readyState==2 || conexion1.readyState==3)
    {
      resultados.innerHTML = 'Procesando...';
    }
}
```

mostrará los valores 2,3 y 4.

Recordemos:

- 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)

pagina1.php

```
<?php
header('Content-Type: text/html; charset=utf-8');
$cuadrado=$_REQUEST['numero']*$_REQUEST['numero'];
echo $cuadrado;
?>
```

Problema resuelto.

Implementar una aplicación que calcule el cuadrado de un número que ingresamos por teclado. Y además mostrar mediante un alert el estado actual de la propiedad readyState del objeto XMLHttpRequest.
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">
   Ingrese nro:<input type="text" name="nro" id="nro" size="10"><br>
   <input type="submit" value="Calcular el cuadrado" id="enviar">
 </form>
 <div id="resultados"></div>
</body>
</html>
```

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();
}

var conexion1;
```

pagina1.php

```
<?php
header('Content-Type: text/html; charset=utf-8');
$cuadrado=$_REQUEST['numero']*$_REQUEST['numero'];
echo $cuadrado;
?>
```

Retornar