APRENDERAPROGRAMAR.COM

XMLHTTPREQUEST AJAX: MÉTODOS OPEN, SEND, GETRESPONSEHEADER, SETREQUESTHEADER (CU01209F)

Sección: Cursos
Categoría: Tutorial básico del programador web: Ajax desde cero

Fecha revisión: 2031

Resumen: Entrega nº9 del Tutorial básico "Ajax desde cero".

Autor: Alex Rodríguez

MÁS SOBRE EL OBJETO XMLHTTPREQUEST

En la anterior entrega del curso, hemos visto las propiedades del objeto predefinido XMLHttpRequest. Este objeto es fundamental en la forma de comunicación o tecnología que denominamos Ajax para intercambiar información en segundo plano con un servidor.

Revisa el código de la anterior entrega y mantenlo "delante de ti", pues va a resultar necesario para entender lo que vamos a explicar a continuación. Ahora vamos a centrarnos en completar nuestra comprensión sobre un fragmento concreto de código que ya hemos estudiado en parte:

```
<script>
function mostrarSugerencia(str) {
var xmlhttp;
var contenidosRecibidos = new Array();
var nodoMostrarResultados = document.getElementById('listaCiudades');
var contenidosAMostrar = '';

if (str.length==0) { document.getElementById("txtInformacion").innerHTML="";
nodoMostrarResultados.innerHTML = ''; return; }

xmlhttp=new XMLHttpRequest();
xmlhttp.onreadystatechange = function() {
 if (xmlhttp.readyState==4 && xmlhttp.status==200) {
   contenidosRecibidos = xmlhttp.responseText.split(",");
   document.getElementById("txtInformacion").innerHTML=contenidosRecibidos[0];
   for (var i=1; i<contenidosRecibidos.length;i++) {
            contenidosAMostrar = contenidosAMostrar+'<div id="ciudades'+i+'"> <a
href="http://aprenderaprogramar.com">' + contenidosRecibidos[i]+ '</a></div>';
            }
            nodoMostrarResultados.innerHTML = contenidosAMostrar;
            }
}

xmlhttp.open("GET"," datosCU01206F.php?pais="+str);
xmlhttp.send();
}

</script>
```

En la anterior entrega del curso hicimos un análisis preliminar de este código. Recordemos que en la línea xmlhttp=new XMLHttpRequest(); se instancia un objeto predefinido: el objeto XMLHttpRequest.

Este objeto al igual que otros objetos predefinidos tiene constructor para su inicialización, propiedades y métodos.

OBJETO XMLHTTPREQUEST: MÉTODOS

El objeto XMLHttpRequest es el objeto clave de Ajax, ya que nos proporciona la interfaz o herramientas necesarias para hacer las operaciones y comprobaciones necesarias para la comunicación en segundo plano con el servidor. XMLHttpRequest es un api definida por el World Wide Web Consortium (W3C), implementada por todos los navegadores modernos, y que nosotros estamos tratando como si fuera un objeto predefinido JavaScript, aunque en realidad no forma parte de la especificación JavaScript.

Los métodos para el objeto XMLHttpRequest son los siguientes:

Método	Descripción	Observaciones / Ejemplo
abort()	Cancela la petición en curso	Puede usarse para abortar el proceso que se está produciendo en segundo plano. Por ejemplo, si el proceso es la subida de un fichero al servidor y el usuario pulsa un botón de cancelar ejecutaríamos: xmlhttp.abort();
open (método, url)	Realiza una petición de apertura de comunicación con un método que puede ser principalmente GET o POST.	Con GET los parámetros de la petición van incluidos en la url. Con POST los parámetros de la petición van en las cabeceras de HTTP. url puede ser una ruta relativa o completa. xmlhttp.open("GET","datos.php?pais=spain");
open (método, url, async, userName, password)	Los parámetros async, userName y password son opcionales. El parámetro async indica si la petición debe ser gestionada asíncronamente. Un valor true implica que el procesamiento del script continuará después de invocarse send() sin bloquear la navegación aún no disponiendo de la respuesta completa del servidor.	El valor por defecto para async es true. Esto indica "procesar en segundo plano sin bloquear la navegación". Si se establece a false, se bloquearía la navegación hasta obtener respuesta. Esto en general es indeseable, pero puede ser necesario por ejemplo si se requiere usuario y contraseña que han de verificarse antes de proseguir. El userName y password serían nombre de usuario y password necesarios en cada caso para acceder al recurso solicitado. xmlhttp.open("GET","datos.php?pais=spain", false, "admin", "pswd");
send()	Envía la petición al servidor	xmlhttp.send();
send(cadena)	Envía la petición al servidor incluyendo datos en una cadena de texto, normalmente asociado al envío de datos mediante POST	xmlhttp.send("nombre=Juan&edad=33");
setRequestHeader (cabecera, valor)	Añade un par cabecera – valor a la cabecera HTTP. Necesario para pasar datos por POST.	xmlhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
getRespondeHeader (cabecera)	Devuelve una cadena con el contenido de la cabecera solicitada	var contType = xmlhttp.getResponseHeader ("Content-Type");
getAllResponseHeaders()	Devuelve una cadena de texto con todas las cabeceras de la respuesta del servidor	var headers = xmlhttp.getAllResponseHeaders ();
Otros métodos	Por ejemplo overrideMimeType	De momento no las estudiaremos.

RESUMEN

El objeto XMLHttpRequest es un objeto predefinido que nos provee de propiedades y métodos para manejar el intercambio de información en segundo plano con el servidor.

Con lo que hemos visto hasta ahora ya somos capaces de comprender todo el código del ejemplo que estamos estudiando:

```
xmlhttp=new XMLHttpRequest();
xmlhttp.onreadystatechange = function() {
 if (xmlhttp.readyState==4 && xmlhttp.status==200) {
   contenidosRecibidos = xmlhttp.responseText.split(",");
   document.getElementById("txtInformacion").innerHTML=contenidosRecibidos[0];
   for (var i=1; i<contenidosRecibidos.length;i++) {
            contenidosAMostrar = contenidosAMostrar+'<div id="ciudades'+i+'"> <a
href="http://aprenderaprogramar.com">' + contenidosRecibidos[i]+ '</a></div>';
            }
            nodoMostrarResultados.innerHTML = contenidosAMostrar;
            }
}

xmlhttp.open("GET"," datosCU01206F.php?pais="+str);
xmlhttp.send();
```

Este código lo interpretamos de la siguiente manera: se crea un objeto XMLHttpRequest. Al indicar que la propiedad onreadystatechange es una función anónima, dicha función se convierte en la manejadora del evento de cambio en la propiedad readyState. Es decir, cuando la situación de la propiedad indicadora de la situación de una petición hecha a través de este objeto evolucione, se ejecutará esta función.

Seguidamente se invoca una url a la que se accede en segundo plano mediante get (de este modo la url lleva implícitos algunos datos). Al invocar open se define la petición.

Finalmente se invoca send, con lo cual se abre la url indicada y se espera una respuesta. La propiedad readyState irá evolucionando desde 0 (sin enviar), a 1 (se ha llamado a send), 2 (recibidas las cabeceras), 3 (descargando la respuesta) y 4 (operación completada). Esta evolución es observada por la función manejadora definida para el objeto, dentro de la cual se especifican las instrucciones a ejecutarse cuando la propiedad readyState valga 4 y la propiedad status valga 200, lo que indica que la transferencia de datos ha tenido lugar correctamente. Una vez recibida la respuesta, ya operamos con xmlhttp.responseText como si de una cadena normal se tratara. En este caso, lo que recibimos es una lista de ciudades separadas por comas, por lo que usamos la función JavaScript split para traspasar esos datos a un array y mostrar las ciudades por pantalla.

En las próximas entregas profundizaremos en el objeto XMLHttpRequest y la operación de Ajax.

EJERCICIO Nº1

Crea un archivo denominado respuesta.php que contenga el código necesario para devolver lo siguiente:

a) Si se recibe el parámetro nombre con valor alberto, la frase "Alberto está realizando un curso de programación".

b) Si se recibe el parámetro nombre con valor juan, la frase "Juan está realizando prácticas de ingeniería de sistemas"

c) Si el parámetro viene vacío o no existe, la frase "No hay información. Revise la petición".

En un archivo denominado peticion.html crea una casilla de texto que pida un nombre y un botón chequear, de modo que al pulsarse el botón chequear se produzca una petición vía Ajax para obtener los resultados contenidos en el archivo respuesta.php. Dichos resultados deberán mostrarse en una ventana de alerta JavaScript cada vez que se pulse el botón chequear.

Para comprobar si tus respuestas son correctas puedes consultar en los foros aprenderaprogramar.com.

EJERCICIO Nº2

Crea un archivo denominado respuesta1.txt que contenga varias líneas de texto y súbelo al servidor. Haz lo mismo para los archivos respuesta2.txt y respuesta3.txt. A continuación crea un archivo html donde usando Ajax y tres botones denominados "Ver archivo 1", "Ver archivo 2" y "Ver archivo 3" puedas visualizar el contenido de cada archivo en un div preparado para ello encima del cual aparecerá "Contenido del archivo solicitado es:".

Para comprobar si tus respuestas son correctas puedes consultar en los foros aprenderaprogramar.com.

Próxima entrega: CU01210F

Acceso al curso completo en aprenderaprogramar.com -- > Cursos, o en la dirección siguiente: http://aprenderaprogramar.com/index.php?option=com_content&view=category&id=83&Itemid=212