



APRENDERAPROGRAMAR.COM

¿QUÉ ES Y PARA QUÉ SIRVE AJAX? JAVASCRIPT ASÍNCRONO, XML Y JSON . VENTAJAS E INCONVENIENTES DE AJAX. XMLHTTPREQUEST . EJEMPLO. (CU01193E)

Sección: Cursos

Categoría: Tutorial básico del programador web: JavaScript desde cero

Fecha revisión: 2029

Resumen: Entrega nº93 del Tutorial básico "JavaScript desde cero".

Autor: César Krall

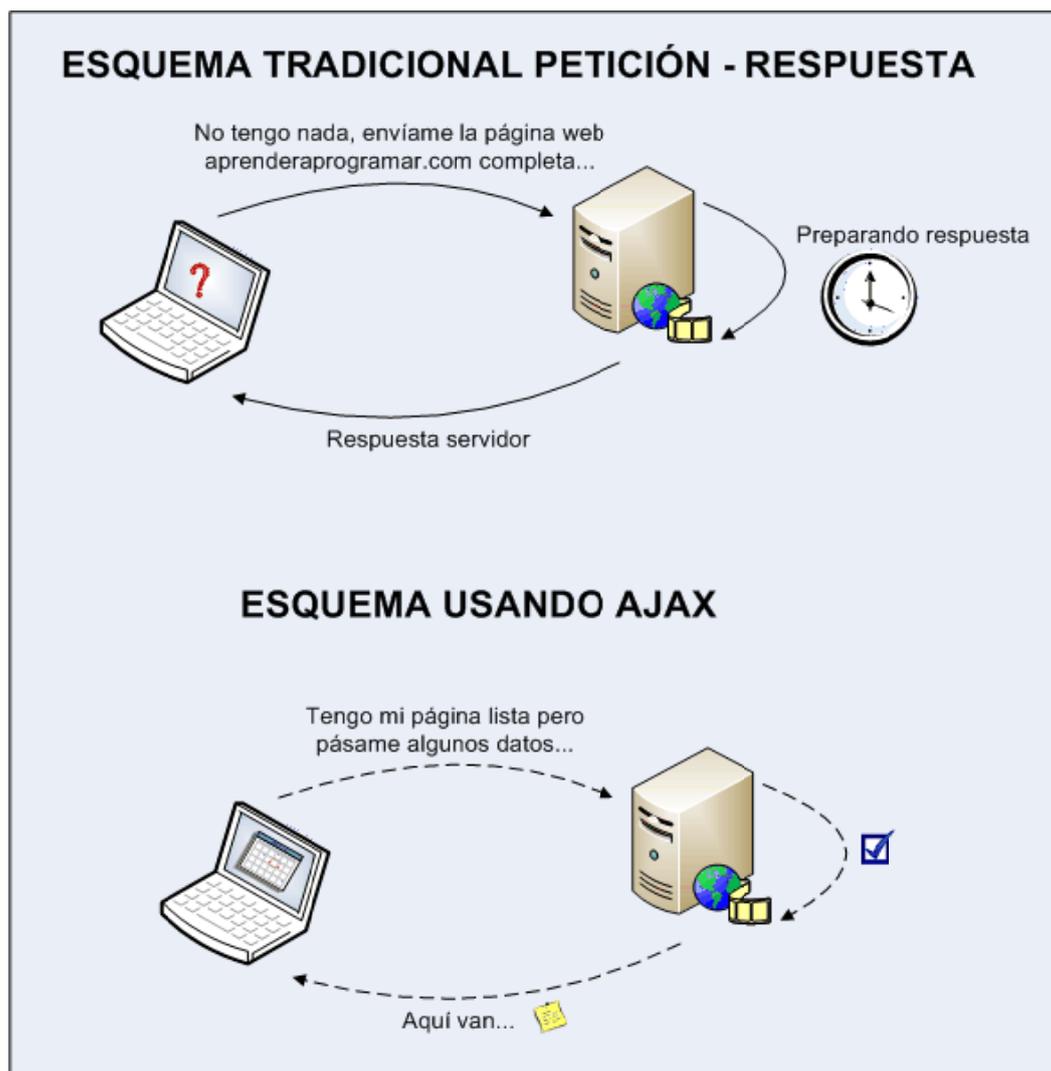
LADO DEL CLIENTE... ¿Y MÁS ALLÁ?

Hasta ahora nos hemos centrado en JavaScript del lado del cliente. Pero las aplicaciones web o apps para smartphones tienen que conectar con servidores (donde normalmente residen los datos en bases de datos). Las peticiones a servidores se suelen hacer con lenguajes como PHP y el inconveniente que tienen son que ralentizan las páginas web.



AJAX ¿QUÉ ES Y PARA QUÉ SIRVE?

Veamos un esquema para comprender mejor la idea que hay detrás del uso de Ajax. Este esquema lo iremos comentando y comprendiendo a medida que avancemos en la explicación.



Realizar peticiones al servidor y esperar respuesta puede consumir tiempo (el tiempo necesario para recargar una página completa). Para agilizar los desarrollos web surgió Ajax (inicialmente Asynchronous JavaScript And XML, aunque hoy día ya no es una tecnología ligada a XML con lo cual no pueden asociarse las siglas a estos términos), una tecnología que busca evitar las demoras propias de las peticiones y respuestas del servidor mediante la transmisión de datos en segundo plano usando un protocolo específicamente diseñado para la transmisión rápida de pequeños paquetes de datos.

Con Ajax, se hace posible realizar peticiones al servidor y obtener respuesta de este en segundo plano (sin necesidad de recargar la página web completa) y usar esos datos para, a través de JavaScript, modificar los contenidos de la página creando efectos dinámicos y rápidos.

En el esquema anterior vemos las ideas en torno a Ajax de forma gráfica. En la parte superior hemos representado lo que sería un esquema de comunicación tradicional: el cliente solicita una página web completa al servidor. El servidor recibe la petición, se toma su tiempo para preparar la respuesta y la envía. El resultado, una pequeña demora debido al tiempo que tarda en llegar la petición al servidor, el tiempo que éste tarda en preparar la respuesta, y el tiempo que tarda en llegar la respuesta más recargarse en el navegador.

En la parte inferior vemos lo que sería un esquema de comunicación usando Ajax: el cliente tiene una página web cargada (puede ser una página web completa, o sólo el esqueleto de una página web). El cliente sigue trabajando y en segundo plano (de ahí que hayamos dibujado con líneas punteadas las comunicaciones) le dice al servidor que le envíe un paquete de datos que le hacen falta. El servidor procesa la petición. Ahora la respuesta es mucho más rápida: no tiene que elaborar una página web completa, sino sólo preparar un paquete de datos. Por tanto el tiempo de respuesta es más rápido. El servidor envía el paquete de datos al cliente y el cliente los usa para cambiar los contenidos que se estaban mostrando en la página web.

VENTAJAS E INCONVENIENTES DE AJAX

Las ventajas que proporciona Ajax son varias:

- a) No es necesario recargar y redibujar la página web completa, con lo que todo es más rápido.
- b) El usuario no percibe que haya demoras: está trabajando y al ser las comunicaciones en segundo plano no hay interrupciones.
- c) Los pasos que antes podía ser necesario dar cargando varias páginas web pueden quedar condensados en una sola página que va cambiando gracias a Ajax y a la información recibida del servidor.

Como todo en la vida, Ajax también tiene inconvenientes:

- a) El usuario puede perder la capacidad para hacer cosas que hacía con webs tradicionales puesto que no hay cambio de página web. Por ejemplo usar los botones de avance y retroceso del navegador o añadir una página a favoritos puede dejar de ser posible. Esto en algunos casos no es deseable.
- b) El desarrollo de aplicaciones web se puede volver más complejo. Supongamos que antes tuviéramos un proceso en el que avanzábamos a través de varias páginas web como 1, 2, 3. De este modo la organización resulta sencilla. Si condensamos todo en una sola página web: 1, escribir y depurar el código puede volverse más complicado. En sitios complejos, puede ser muy difícil depurar errores.
- c) Existen problemas y restricciones de seguridad relacionados con el uso de Ajax. Hay que tener en cuenta que por motivos de seguridad no todos los procesos se pueden realizar del lado del cliente (que por su propia naturaleza es "manipulable"). También existen restricciones de seguridad para impedir la carga de contenidos mediante Ajax desde sitios de terceras partes.
- d) La indexación para los motores de búsqueda se ve dificultada, con lo cual nuestros sitios web pueden perder visibilidad en los buscadores. No es lo mismo un contenido "constante" o aproximadamente estático, fácilmente rastreable para un buscador, que un contenido "cambiante" en función de la ejecución de JavaScript, difícilmente rastreable para un buscador.

¿MEJOR USAR O NO USAR AJAX?

Como todo, hay que usar las cosas en su justa medida. Ajax bien usado puede ser muy útil para una página web. Ajax mal usado puede ser un desastre para una página web.

¿ES AJAX UN LENGUAJE DE PROGRAMACIÓN?

No, Ajax es un conjunto de técnicas que se usan para lograr un objetivo y se basa en lenguajes ya existentes como JavaScript.

Podríamos dar esta definición de Ajax: "Ajax es un conjunto de métodos y técnicas que permiten intercambiar datos con un servidor y actualizar partes de páginas web sin necesidad de recargar la página completamente".

Aunque Ajax se pensó inicialmente para transferir datos en un solo formato (XML), actualmente Ajax permite la transmisión de datos en múltiples formatos: XML, JSON, EBML, texto plano, HTML, etc.

¿QUÉ ES XML Y QUÉ ES JSON?

XML (extensible markup language) es un lenguaje de etiquetas que se usa para almacenar y enviar información. No vamos a estudiar XML, simplemente pondremos un ejemplo para hacernos una idea de qué son datos en formato XML. XML se usa para múltiples aplicaciones.

JSON (JavaScript Object Notation) es un formato para el intercambio de datos que se usa para almacenar y enviar información, basado en la notación literal de objetos de JavaScript. No vamos a

estudiar JSON, simplemente pondremos un ejemplo para hacernos una idea de qué son datos en formato JSON. JSON se usa como alternativa al XML en AJAX.

Ejemplo datos en formato XML	Ejemplo datos en formato JSON
<pre> <Curso> <Titulo>JavaScript desde cero</Titulo> <Autores> <Nombre>César Krall</Nombre> <Nombre>Bill Clinton</Nombre> </Autores> <WebPublicacion> <Url>aprenderaprogramar.com</Url> <PageRank>10</PageRank> </WebPublicacion> </Curso> </pre>	<pre> { "Curso": { "Titulo": "JavaScript desde cero", "Autores": { "Nombre": ["César Krall", "Bill Clinton"] }, "WebPublicacion": { "Url": "aprenderaprogramar.com", "PageRank": "10" } } } </pre>

Esto nos sirve para hacernos una idea de qué es un paquete de datos que se intercambia entre el cliente y el servidor a través de Ajax.

Como curiosidad, tener en cuenta que el término XHTML hacer referencia a la combinación de HTML con XML en el sentido de escribir HTML siguiendo las reglas de sintaxis propias de XML. Bajo XHTML por ejemplo, no pueden existir etiquetas que no se cierren, ya que XML impone la obligación de que todas las etiquetas se cierren.

UN EJEMPLO SIMPLE DE USO DE AJAX

En este curso nos centramos en JavaScript del lado del cliente y no vamos a estudiar Ajax. Hemos hecho una breve introducción a Ajax porque es una tecnología interesante para aquellas personas que tras terminar el curso quieran seguir profundizando en programación web y por ello hemos considerado conveniente hablar brevemente sobre Ajax. El ejercicio-ejemplo que se plantea es un ejercicio opcional, por tanto puedes saltarlo y pasar a la siguiente entrega del curso.

EJERCICIO OPCIONAL

Este ejercicio no forma parte del curso, por eso se califica como opcional. El ejercicio consiste en ver un breve ejemplo de uso de Ajax para realizar sugerencias de búsqueda (algo parecido a lo que hacen los buscadores cuando muestran sugerencias una vez hemos empezado a introducir una palabra).

Para desarrollar este ejercicio necesitas tener conocimientos básicos de PHP y disponer de un servidor (puede ser un servidor remoto o un servidor local como WAMP ó XAMPP o similar).

Escribe este código y guárdalo **en el servidor** con el nombre de archivo pruebaAjax.html:

```
<!DOCTYPE html><html><head><title>Cursos aprende a programar</title><meta charset="utf-8">
<script>
function mostrarSugerencia(str) {
var xmlhttp;
if (str.length==0) { document.getElementById("txtSugerencia").innerHTML=""; return; }
xmlhttp=new XMLHttpRequest();
xmlhttp.onreadystatechange = function() {
if (xmlhttp.readyState==4 && xmlhttp.status==200) {
document.getElementById("txtSugerencia").innerHTML=xmlhttp.responseText;
}
}
xmlhttp.open("GET", "listadoDatos.php?q="+str,true);
xmlhttp.send();
}
</script></head>
<body><h3>Empieza a escribir un nombre en la casilla para obtener una sugerencia:</h3>
<form action="">
Nombre: <input type="text" id="txt1" onkeyup="mostrarSugerencia(this.value)" />
</form>
<p>Sugerencias: <span id="txtSugerencia"></span></p>
</body></html>
```

Escribe este código y guárdalo **en el servidor** con el nombre de archivo listadoDatos.php:

```
<?php // Rellenamos un array con nombres
$a[]="Ana"; $a[]="Belinda"; $a[]="Cinderella"; $a[]="Diana"; $a[]="Eva"; $a[]="Fiona"; $a[]="Gabriela"; $a[]="Hilda";
$a[]="Idaira"; $a[]="Johanna"; $a[]="Kitty"; $a[]="Linda"; $a[]="Nina"; $a[]="Ofelia"; $a[]="Petunia"; $a[]="Amanda";
$a[]="Raquel"; $a[]="Cindy"; $a[]="Doris"; $a[]="Eve"; $a[]="Evita"; $a[]="Sunniva"; $a[]="Tania"; $a[]="Ursula";
$a[]="Violeta"; $a[]="Liza"; $a[]="Elizabeth"; $a[]="Ellen"; $a[]="Walda"; $a[]="Vicky";

// Rescatamos el parámetro q que nos llega mediante la url que invoca xmlhttp
$q=$_REQUEST["q"]; $hint="";

// Buscamos la coincidencia en el primer caracter entre nombres del array con el primer caracter de q
if ($q != ""){
    $q=strtolower($q); $len=strlen($q);
    foreach($a as $name) {
        if (striestr($q, substr($name,0,$len)) {
            if ($hint=="") { $hint=$name; }
            else { $hint .= ", $name"; }
        }
    }
}
// Devolvemos "no hay sugerencias" si no se encuentran coincidencias
// o los datos en $hint (nombres separados por comas) si se encontraron coincidencias
echo $hint==" " ? "No hay sugerencias" : $hint;
?>
```

Accede al archivo html invocando la ruta correcta (que dependerá de la ruta dentro de la que esté alojado en el servidor. La ruta será por ejemplo <http://aprenderaprogramar.com/pruebaAjax.html>).

Fijate en que lo interesante de este ejemplo es que se recuperan datos que se encuentran en el servidor.

Responde a estas preguntas:

- a) ¿Qué sugerencias se muestran si escribes la letra L en el cuadro de texto donde se pide el nombre?
- b) ¿Qué función JavaScript se dispara cada vez que se pulsa una tecla?
- c) ¿Se usa XML en este código?
- d) ¿Se usa JSON en este código?
- e) Busca información en internet sobre las instrucciones JavaScript / Ajax que se usan en el código y explica cada una de ellas.
- f) Reescribe el código para que todo el proceso tenga lugar del lado del cliente. Es decir, traslada los datos al lado del cliente y hazlo todo con código JavaScript sin necesidad de utilizar Ajax.

Para comprobar si tus respuestas y código son correctos puedes consultar en los foros aprenderaprogramar.com.

Próxima entrega: CU01194E

Acceso al curso completo en aprenderaprogramar.com -- > Cursos, o en la dirección siguiente:
http://aprenderaprogramar.com/index.php?option=com_content&view=category&id=78&Itemid=206