

Introducción a AJAX



AJAX no es un lenguaje exactamente.

Su nombre viene dado por el acrónimo de **Asynchronous JavaScript And XML** y es posiblemente la mayor novedad en cuanto a programación Web en estos últimos años.

El corazón de Ajax es el objeto **XMLHttpRequest** que nos permite realizar una conexión al servidor y al enviarle una petición y recibir la respuesta que procesaremos en nuestro código Javascript, estamos hablando del verdadero motor de Ajax; por ejemplo gracias a este objeto podemos desde una página HTML leer datos de una Web o enviar datos de un formulario sin necesidad de recargar la página.

10 razones para usar AJAX.

1. Basado en los estándares abiertos
2. Usabilidad
3. Válido en cualquier plataforma y navegador
4. Beneficia las aplicaciones Web
5. No es difícil su utilización
6. Compatible con Flash
7. Adoptado por los "gordos" de la tecnología Web
8. Web 2.0
9. Es independiente del tipo de tecnología de servidor que se utilice
10. Mejora la estética de la Web

La manera más fácil para comprender realmente la funcionalidad de Ajax es ver cómo funciona una aplicación Web con Ajax y cómo una sin Ajax.

Sin Ajax

Se crearía una página con un formulario, cuando el usuario envía los datos del formulario se produce una conexión a la base de datos y se muestra por pantalla la página que el servidor devuelve, todo esto hace que se recargue la página ya sea saltando a una diferente o a ella misma, el usuario debe esperar una nueva carga de página después de cada envío.

Es lento porque debe descargar la información HTML por duplicado.

Con Ajax

Utilizaríamos un código Javascript que crearía el mencionado objeto **XMLHttpRequest** al enviar el formulario, esta llamada se produce de forma **asíncrona** lo que significa que **se envían los datos y no se recarga la página**, una vez el servidor responde una función Javascript es la que valora la respuesta del servidor, si esta respuesta es la deseada imprimiremos el texto que indique al usuario que sus datos fueron enviados correctamente.

El navegador no recarga la página, la experiencia desde el punto de vista del usuario es muy satisfactoria puesto que se asemeja a la respuesta del típico software de escritorio, ya no te planteas enlazar páginas sino enviar y recibir datos en una misma página que mediante funciones evalúa las diferentes respuestas.

Es bastante más rápido puesto que no tiene que descargar de nuevo el código HTML de la página de confirmación del formulario.

--Rafael Veiga 12:51 15 jun 2011 (BST)