

Desarrollo Web en entorno Cliente: JAVASCRIPT

Introducción

Lo primero que deberíamos de saber es que Javascript no es java.

Javascript no es un lenguaje compilado, lo que lo hace parecer un lenguaje no muy sólido. Los programadores de Internet más adelantados de finales de los años 90 inicialmente descartaron Javascript, aunque pronto se dieron cuenta de su utilidad y solidez, tanto para estimular como para crear interactividad en la Word Wide Web.

Javascript se concentró en la validación de formularios y en añadir a las imágenes una rudimentaria, aunque útil interactividad y respuesta al visitante. Cuando un visitante de una página Web rellena un formulario, Javascript podrá validar instantáneamente los contenidos, en lugar de tener que pasar al servidor. Esto es una forma excelente de ayudar a acelerar las aplicaciones y ya no es necesario que el servidor se ocupe de la mayor carga de procesamiento, trasladando esta mayor carga al navegador web del cliente.

Importante: es imperativo probar las páginas web en diferentes navegadores.

¿QUÉ PUEDE HACER JAVASCRIPT?

Javascript es un lenguaje complementario, lo que significa que no es habitual encontrar una aplicación que esté sólo escrita en javascript sin la ayuda de otros lenguajes, como (X)html, y sin presentación en un navegador.

Javascript puede hacer muchas cosas del lado del cliente en una aplicación. Puede añadir interactividad en una página web con acciones como crear menús desplegables, transformar texto en una página, añadir elementos dinámicos a una página, y ayudar para introducir datos en un formulario.

¿QUÉ NO PUEDE HACER JAVASCRIPT?

NO PUEDE FORZARSE A UN CLIENTE

En términos prácticos, javascript depende de otra interfaz o programa anfitrión para su funcionamiento. Este programa anfitrión (host) habitualmente es el navegador web del cliente. Como javascript es un lenguaje del lado del cliente, sólo puede hacer lo que éste le permite.

Debemos trabajar un poco más para asegurarnos de que la página web que estamos diseñando estará disponible para quienes no posean javascript.

Si escribimos una aplicación que funciona bien desde nuestro equipo, no tiene por qué funcionar en otro. Una vez enviada la aplicación al cliente, ya no es posible controlar cómo funcionará.

NO PUEDE GARANTIZAR LA SEGURIDAD DE LOS DATOS

Dado que javascript se ejecuta enteramente en el cliente, el programador ha de saber renunciar a su control. Una vez que está en el equipo del cliente éste puede manipular de cualquier forma el código antes de enviarlo de nuevo al servidor. Nunca debemos confiar en un dato que venga de vuelta del cliente. Incluso si se han utilizado las funciones javascript para validar el contenido de formularios, tendremos que validar los datos de nuevo una vez que llegan al servidor. Un cliente con javascript deshabilitado podría enviar datos corrompidos a través de un formulario (javascript se puede inhabilitar en el equipo de nuestro visitante).

NO PUEDE CRUZAR DOMINIOS

Los programas ejecutados desde un dominio no tienen acceso a las propiedades de otro dominio de internet, ni tampoco pueden afectar a los programas y datos de otro dominio. Por ejemplo, javascript se puede usar para abrir una nueva ventana del navegador, pero el contenido de esa ventana está restringido al programa de origen. Si una página de mi sitio web contiene javascript, no puede acceder a ningún javascript ejecutado desde otro dominio.

Lo esencial es ser conscientes de que javascript se limita a hacer cosas desde su propia ventana del navegador.

NO HACE SERVIDORES

Cuando se programa un código del lado del servidor, como VB .net o PHP, casi podemos asegurar que el servidor desarrollará determinadas funciones, como comunicarse con una base de datos o dar acceso a módulos necesarios para la aplicación web. Javascript no tiene acceso a las variables del lado del servidor. Por ejemplo, Javascript no puede acceder a bases de datos situadas en el servidor. El código javascript se limita a lo que se puede hacer dentro de la plataforma en la que se está ejecutando el programa, que suele ser el navegador.

¿QUÉ EDITOR USAR?

La mejor forma de aprender JavaScript es tecleando el código HTML y JavaScript en un simple documento de texto. La elección del **editor** depende de nosotros, pero aquí tenemos algunas pistas para realizar una buena elección.

Para aprender JavaScript no se recomiendan editores del estilo **WYSIWYG** (What You See is What You Get) como Dreamweaver o FrontPage, ya que estas herramientas están más orientadas a la modificación de contenido y presentación, y nosotros nos vamos a centrar más en el código fuente de la página.

Uno de los **factores** importantes que tenemos que tener en cuenta a la hora de elegir un editor, es ver la facilidad con la que se pueden grabar los ficheros con extensión .html. Independientemente del sistema operativo que estemos utilizando cualquier programa que nos permita grabar ficheros directamente con la

extensión .htm o .html nos evitaría un gran número de problemas. También hay que tener en cuenta la **codificación** que emplea ese programa para grabar los ficheros. Por ejemplo en Microsoft Word cuando intentamos guardar archivos éste intenta almacenarlos en un formato binario de Word - algo que los navegadores web no pueden cargar. Para grabar ese archivo con extensión .txt o .html requiere moverse un poco más por el menú de diálogo "Guardar como", lo cual es realmente una clara desventaja.

La idea es decantarse por **editores** que posean **características** que te faciliten la programación web, como por ejemplo:

- Sintaxis con codificación de colores. Que resalte automáticamente en diferente color o tipos de letra los elementos del lenguaje tales como objetos, comentarios, funciones, variables, etc.
- Verificación de sintaxis. Que te marque los errores en la sintaxis del código que estás escribiendo.
- Que te permita diferenciar los comentarios del resto del código.
- Que genere automáticamente partes del código tales como bloques, estructuras, etc.
- Que disponga de utilidades adicionales, tales como cliente FTP para enviar tus ficheros automáticamente al servidor, etc.

Dependiendo del **sistema operativo** que utilicemos en nuestro ordenador disponemos de múltiples **opciones de editores**. Cada una es perfectamente válida para poder programar en JavaScript. Algunos ejemplos de editores web gratuitos son:

- Para Windows tienes: Visual Code, Notepad++, Aptana Studio, Bluefish, Eclipse, NetBeans, etc.
- Para Mac OS tienes: Aptana Studio, Bluefish, Eclipse, KompoZer, Nvu, etc.
- Para Linux tienes: KompoZer, Amaya, Quanta Plus, Bluefish, codetch, etc.

¿QUÉ NAVEGADOR WEB USAR?

Otro de los componentes obligatorio para aprender JavaScript es el **navegador web**. No es necesario que nos conectemos a Internet para comprobar nuestros scripts realizados con JavaScript. Podemos realizar dicha tarea sin conexión. Ésto quiere decir que podemos aprender JavaScript y crear aplicaciones con un ordenador portátil y desde cualquier lugar sin necesitar Internet para ello.

El tipo de **navegador web** que utilicemos es elección nuestra. Eso sí, se recomienda que se usen las últimas versiones disponibles para evitar problemas de seguridad e incompatibilidades. Algunos ejemplos de navegadores gratuitos son:

- Para Windows: Mozilla Firefox, Google Chrome, Safari, Opera, Microsoft Edge, etc.
- Para Mac OS: Mozilla Firefox, Safari, Google Chrome, etc.

- Para Linux: Mozilla Firefox, Konqueror, Opera, etc.

Una recomendación muy interesante es el disponer de 2 ó 3 tipos de navegadores diferentes, ya que así podremos comprobar la compatibilidad de nuestra página web y ver si nuestro código fuente de JavaScript se ejecuta correctamente en todos ellos.

Para ajustar un poco más nuestro entorno de trabajo, lo último que necesitamos es el poder ejecutar nuestro editor web y nuestro navegador de forma simultánea, ya que el **flujo típico de trabajo en JavaScript** va a ser:

1. Introducir HTML, JavaScript y CSS en el documento original en el editor web.
2. Guardarlo en disco.
3. Cambiarnos al navegador web.
4. Realizar una de las siguientes tareas:
 - Si es un nuevo documento, abrirlo a través de la opción Abrir del menú Archivo > Abrir Archivo.
 - Si el documento ya está cargado en el navegador pues simplemente recargar la página.

Los pasos 2 al 4 son acciones que se van a ejecutar muy frecuentemente. Esa secuencia grabar-cambiar-recargar la realizaremos tantas veces cuando estemos escribiendo y depurando nuestro script, que llegará a ser prácticamente un acto reflejo. Algunos editores ya disponen de teclas rápidas para realizar esta tarea. Todo ello dependerá del tipo de editor, navegador y sistema operativo que utilicemos.

VALIDACIÓN

Otro aspecto muy importante es la **validación**. Podemos ahorrarnos muchas horas de comprobaciones simplemente asegurándonos de que nuestro código HTML es válido. *Si nuestro código HTML contiene imperfecciones, tenemos muchas posibilidades de que nuestro JavaScript o CSS no funcionen de la manera esperada, ya que ambos dependen de los elementos HTML y sus atributos. Cuanto más nos ajustemos a las especificaciones del estándar, mejor resultado obtendremos entre los diferentes tipos de navegadores.*

El consorcio W3C, el cuál diseñó el lenguaje HTML, desarrolló un validador que nos permitirá chequear si nuestra página web cumple las **especificaciones** indicadas por el elemento DOCTYPE que se incluye al principio de cada página web que realicemos. El Validador será nuestro amigo y nos permitirá realizar unas páginas más consistentes.

La dirección del **Validador W3C** es: [Validador W3C](#)