


JavaScript: Introducción

A principios de los años 90, la mayoría de usuarios que se conectaban a Internet lo hacían con módems a una velocidad máxima de 28.8 kbps. En esa época, empezaban a desarrollarse las primeras aplicaciones web y por tanto, las páginas web comenzaban a incluir formularios cada vez más complejos.

De ahí surgió la necesidad de un lenguaje de programación que se ejecutara en el navegador del usuario. Así, si el usuario no rellenaba correctamente un formulario, no se le hacía esperar mucho tiempo hasta que el servidor volviera a mostrar el formulario indicando los errores existentes.



JavaScript: Introducción

Brendan Eich, un programador que trabajaba en Netscape, pensó que podría solucionar este problema adaptando otras tecnologías existentes (como ScriptEase) al navegador Netscape Navigator 2.0, que iba a lanzarse en 1995. Inicialmente, Eich denominó a su lenguaje LiveScript, pero justo antes del lanzamiento Netscape decidió cambiar el nombre por el de **JavaScript**. La razón del cambio de nombre fue exclusivamente por marketing, ya que *Java* era la palabra de moda en el mundo informático y de Internet de la época.



JavaScript: Incluir JavaScript en HTML

La integración de JavaScript y XHTML es muy flexible, ya que existen al menos tres formas para incluir código JavaScript en las páginas web.

- **Incluir JavaScript en el mismo documento HTML**

El código JavaScript se encierra entre etiquetas `<script>` y se incluye en cualquier parte del documento. Aunque es correcto incluir cualquier bloque de código en cualquier zona de la página, se recomienda definir el código JavaScript dentro de la cabecera del documento (dentro de la etiqueta `<head>`)



JavaScript: Incluir JavaScript en HTML

Nuestro primer código JavaScript, Método 1:

```
<head>  
  <meta charset="utf-8">  
  <meta name="description" content="">  
  <meta name="keywords" content="">  
  <title>Ejemplo de código JavaScript en el propio  
  documento</title>  
  <script type="text/javascript">  
    alert("Un mensaje de prueba");  
  </script>  
</head>
```



JavaScript: Incluir JavaScript en HTML

La integración de JavaScript y XHTML es muy flexible, ya que existen al menos tres formas para incluir código JavaScript en las páginas web.

- **Definir JavaScript en un archivo externo**

Las instrucciones JavaScript se pueden incluir en un archivo externo de tipo JavaScript que los documentos HTML enlazan mediante la etiqueta `<script>`. Se pueden crear todos los archivos JavaScript que sean necesarios y cada documento HTML puede enlazar tantos archivos JavaScript como necesite.



JavaScript: Incluir JavaScript en HTML

Método 2, archivo externo:

```
<head>  
  <meta charset="utf-8">  
  <meta name="description" content="">  
  <meta name="keywords" content="">  
  <title>Ejemplo de código JavaScript en el propio  
  documento</title>  
  <script type="text/javascript" src="codigo.js">  
  </script>  
</head>
```



JavaScript: Incluir JavaScript en HTML

Método 2, archivo externo:

Nombre Archivo: `codigo.js`

Contenido:

```
alert("Un mensaje de prueba");
```



JavaScript: Incluir JavaScript en HTML

La integración de JavaScript y XHTML es muy flexible, ya que existen al menos tres formas para incluir código JavaScript en las páginas web.

- **Incluir JavaScript en los elementos HTML**

Este último método es el menos utilizado, ya que consiste en incluir trozos de JavaScript dentro del código HTML de la página. El mayor inconveniente de este método es que ensucia innecesariamente el código HTML de la página y complica el mantenimiento del código JavaScript.



JavaScript: Incluir JavaScript en HTML

Método 3, incluir JavaScript en los elementos HTML:

...

```
</head>
```

```
<body>
```

```
<p onclick="alert('Un mensaje de prueba')">Un  
párrafo de texto.</p>
```

```
</body>
```

```
</html>
```



JavaScript: Incluir JavaScript en HTML

Glosario Básico:

Script: cada uno de los programas, aplicaciones o trozos de código creados con el lenguaje de programación *JavaScript*. Unas pocas líneas de código forman un script y un archivo de miles de líneas de *JavaScript* también se considera un script. A veces se traduce al español directamente como "guión", aunque script es una palabra más adecuada y comúnmente aceptada.



JavaScript

Glosario Básico:

Sentencia: cada una de las instrucciones que forman un script.

Palabras reservadas: son las palabras (en inglés) que se utilizan para construir las sentencias de *JavaScript* y que por tanto no pueden ser utilizadas libremente. Las palabras actualmente reservadas por *JavaScript* son: **break, case, catch, continue, default, delete, do, else, finally, for, function, if, in, instanceof, new, return, switch, this, throw, try, typeof, var, void, while, with**



JavaScript: Sintaxis

Sintaxis:

Las normas básicas que definen la sintaxis de JavaScript son las siguientes:

- **No se tienen en cuenta los espacios en blanco y las nuevas líneas**
- **Se distinguen las mayúsculas y minúsculas**
- **No se define el tipo de las variables**
- **No es necesario terminar cada sentencia con el carácter de punto y coma (;)**
- **Se pueden incluir comentarios**



JavaScript: Sintaxis

Sintaxis:

No se tienen en cuenta los espacios en blanco y las nuevas líneas:

como sucede con HTML, el intérprete de *JavaScript* ignora cualquier espacio en blanco sobrante, por lo que el código se puede ordenar de forma adecuada para entenderlo mejor (tabulando las líneas, añadiendo espacios, creando nuevas líneas, etc.)



JavaScript: Sintaxis

Sintaxis:

Se distinguen las mayúsculas y minúsculas:

al igual que sucede con la sintaxis de las etiquetas y elementos HTML. Sin embargo, si en una página HTML se utilizan indistintamente mayúsculas y minúsculas, la página se visualiza correctamente, siendo el único problema la no validación de la página. En cambio, si en *JavaScript* se intercambian mayúsculas y minúsculas el script no funciona.



JavaScript: Sintaxis

Sintaxis:

No se define el tipo de las variables:

al crear una variable, no es necesario indicar el tipo de dato que almacenará. De esta forma, una misma variable puede almacenar diferentes tipos de datos durante la ejecución del script.



JavaScript: Sintaxis

Sintaxis:

No es necesario terminar cada sentencia con el carácter de punto y coma (;):

en la mayoría de lenguajes de programación, es obligatorio terminar cada sentencia con el carácter (;). Aunque *JavaScript* no obliga a hacerlo, es conveniente seguir la tradición de terminar cada sentencia con el carácter del punto y coma (;).



JavaScript: Sintaxis

Sintaxis:

Se pueden incluir comentarios:

los comentarios se utilizan para añadir información en el código fuente del programa. Aunque el contenido de los comentarios no se visualiza por pantalla, si que se envía al navegador del usuario junto con el resto del script, por lo que es necesario extremar las precauciones sobre la información incluida en los comentarios.



JavaScript: Sintaxis

Sintaxis:

Se pueden incluir comentarios:

JavaScript define dos tipos de comentarios: los de una sola línea y los que ocupan varias líneas.

Los comentarios de una sola línea se definen añadiendo dos barras oblicuas (//) al principio de la línea.

Los comentarios multilínea se definen encerrando el texto del comentario entre los símbolos /* y */.



JavaScript: Limitaciones

Desde su aparición, JavaScript siempre fue utilizado de forma masiva por la mayoría de sitios de Internet. La aparición de Flash disminuyó su popularidad, ya que Flash permitía realizar algunas acciones imposibles de llevar a cabo mediante JavaScript.

Sin embargo, la aparición de las aplicaciones AJAX programadas con JavaScript le ha devuelto una popularidad sin igual dentro de los lenguajes de programación web.



JavaScript: Limitaciones

En cuanto a las limitaciones, JavaScript fue diseñado de forma que se ejecutara en un entorno muy limitado que permitiera a los usuarios confiar en la ejecución de los scripts.

Así, los scripts de JavaScript no pueden comunicarse con recursos que no pertenezcan al mismo dominio desde el que se descargó el script. Los scripts tampoco pueden cerrar ventanas que no hayan abierto esos mismos scripts. Las ventanas que se crean no pueden ser demasiado pequeñas ni demasiado grandes ni colocarse fuera de la vista del usuario.



JavaScript: Limitaciones

Además, los scripts no pueden acceder a los archivos del ordenador del usuario (ni en modo lectura ni en modo escritura) y tampoco pueden leer o modificar las preferencias del navegador.

Por último, si la ejecución de un script dura demasiado tiempo (por ejemplo por un error de programación) el navegador informa al usuario de que un script está consumiendo demasiados recursos y le da la posibilidad de detener su ejecución.



JavaScript: Limitaciones

Además, los scripts no pueden acceder a los archivos del ordenador del usuario (ni en modo lectura ni en modo escritura) y tampoco pueden leer o modificar las preferencias del navegador.

Por último, si la ejecución de un script dura demasiado tiempo (por ejemplo por un error de programación) el navegador informa al usuario de que un script está consumiendo demasiados recursos y le da la posibilidad de detener su ejecución.



JavaScript: Etiqueta noscript

Algunos navegadores no disponen de soporte completo de JavaScript, otros navegadores permiten bloquearlo parcialmente e incluso algunos usuarios bloquean completamente el uso de JavaScript porque creen que así navegan de forma más segura.

El lenguaje HTML define la etiqueta **<noscript>** para mostrar un mensaje al usuario cuando su navegador no puede ejecutar JavaScript. El siguiente código muestra un ejemplo del uso de la etiqueta **<noscript>**:



JavaScript: Etiqueta noscript

Algunos navegadores no disponen de soporte completo de JavaScript, otros navegadores permiten bloquearlo parcialmente e incluso algunos usuarios bloquean completamente el uso de JavaScript porque creen que así navegan de forma más segura.

El lenguaje HTML define la etiqueta **<noscript>** para mostrar un mensaje al usuario cuando su navegador no puede ejecutar JavaScript. La etiqueta **<noscript>** se debe incluir en el interior de la etiqueta **<body>**. El mensaje que muestra **<noscript>** puede incluir cualquier elemento o etiqueta HTML.



JavaScript: Etiqueta noscript

```
<head> ... </head>
```

```
<body>
```

```
  <noscript>
```

```
    <p>Bienvenido a Mi Sitio</p>
```

```
    <p>La página que estás viendo requiere para su  
funcionamiento el uso de JavaScript.
```

```
    Si lo has deshabilitado intencionadamente, por  
    favor vuelve a activarlo.</p>
```

```
  </noscript>
```

```
</body>
```



JavaScript: Etiqueta noscript

Ejercicio 1

Modificar el primer script para que:

- 1) Todo el código JavaScript se encuentre en un archivo externo llamado `codigo.js` y el script siga funcionando de la misma manera.
- 2) Después del primer mensaje, se debe mostrar otro mensaje que diga "Soy el primer script"
- 3) Añadir algunos comentarios que expliquen el funcionamiento del código
- 4) Añadir en la página HTML un mensaje de aviso para los navegadores que no tengan activado el soporte de JavaScript

