



Curso de Javascript

Unidad Didáctica 01: Introducción



Ayuntamiento
de Vitoria-Gasteiz
Vitoria-Gasteizko
Udala

Índice de contenidos

- Introducción
- ¿Qué es JavaScript?
- Breve historia
- Especificaciones oficiales
- Cómo incluir JavaScript en documentos XHTML
- Etiqueta noscript
- Glosario básico
- Sintaxis
- Posibilidades y limitaciones
- Conclusiones

<http://cursosdedesarrollo.com/>



Introducción

JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico



Introducción

Se utiliza principalmente en el lado del cliente, implementado como parte de un navegador web permitiendo crear interacción con el usuario y páginas web dinámicas, aunque actualmente es posible ejecutar JavaScript en el propio servidor (NodeJS)



Introducción

Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF o aplicaciones de escritorio (mayoritariamente widgets) es también significativo



Introducción

JavaScript se diseñó con una sintaxis similar al lenguaje de programación C, aunque adopta nombres y convenciones del lenguaje de programación Java



Introducción

Sin embargo Java y JavaScript no están relacionados y tienen semánticas y propósitos diferentes



Breve Historia

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 complejos



Breve Historia

Con unas aplicaciones web cada vez más complejas y una velocidad de navegación tan lenta, surgió la necesidad de un lenguaje de programación que se ejecutara en el navegador del usuario



Breve Historia

De esta forma, 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



Breve Historia

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



Breve Historia

Posteriormente, Netscape firmó una alianza con Sun Microsystems para el desarrollo del nuevo lenguaje de programación. Además, justo antes del lanzamiento Netscape decidió cambiar el nombre por el de JavaScript



Breve Historia

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



Breve Historia

La primera versión de JavaScript fue un completo éxito y Netscape Navigator 3.0 ya incorporaba la siguiente versión del lenguaje, la versión 1.1

Al mismo tiempo, Microsoft lanzó JScript con su navegador Internet Explorer 3. JScript era una copia de JavaScript al que le cambiaron el nombre para evitar problemas legales



Breve Historia

Para evitar una guerra de tecnologías, Netscape decidió que lo mejor sería estandarizar el lenguaje JavaScript

De esta forma, en 1997 se envió la especificación JavaScript 1.1 al organismo ECMA (European Computer Manufacturers Association)



Breve Historia

ECMA creó el comité TC39 con el objetivo de
"estandarizar de un lenguaje de script
multiplataforma e independiente de cualquier
empresa"

El primer estándar que creó el comité TC39 se
denominó ECMA-262, en el que se definió por
primera vez el lenguaje ECMAScript



Breve Historia

Por este motivo, algunos programadores prefieren la denominación ECMAScript para referirse al lenguaje JavaScript

De hecho, JavaScript no es más que la implementación que realizó la empresa Netscape del estándar ECMAScript



Breve Historia

La organización internacional para la estandarización (ISO) adoptó el estándar ECMA-262 a través de su comisión IEC, dando lugar al estándar ISO/IEC-16262



Especificaciones Oficiales

ECMA ha publicado varios estándares relacionados con ECMAScript. En Junio de 1997 se publicó la primera edición del estándar ECMA-262

Un año después, en Junio de 1998 se realizaron pequeñas modificaciones para adaptarlo al estándar ISO/IEC-16262 y se creó la segunda edición



Especificaciones Oficiales

La quinta edición del estándar ECMA-262 (publicada en junio de 2011) es la versión que utilizan los navegadores actuales y se puede consultar gratuitamente en <http://www.ecma-international.org/publications/standards/Ecma-262.htm>



Cómo incluir Javascript en documentos XHTML

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



Cómo incluir Javascript en documentos XHTML

INCLUIR JAVASCRIPT EN EL MISMO DOCUMENTO XHTML

El código JavaScript se encierra entre etiquetas `<script>` y se incluye en cualquier parte del documento



Cómo incluir Javascript en documentos XHTML

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>`)



Cómo incluir Javascript en documentos XHTML

```
<head>
```

```
  <meta http-equiv="Content-Type" content="text/html;  
  charset=UTF-8" />
```

```
  <title>Ejemplo de código JavaScript en el propio documento</  
  title>
```

```
  <script type="text/javascript">
```

```
    console.log("Un mensaje de prueba");
```

```
  </script>
```

```
</head>
```

<http://cursosdedesarrollo.com/>



Cómo incluir Javascript en documentos XHTML

Para que la página XHTML resultante sea válida, es necesario añadir el atributo type a la etiqueta `<script>`

Los valores que se incluyen en el atributo type están estandarizados y para el caso de JavaScript, el valor correcto es `text/javascript`



Cómo incluir Javascript en documentos XHTML

Este método se emplea cuando se define un bloque pequeño de código o cuando se quieren incluir instrucciones específicas en un determinado documento HTML que completen las instrucciones y funciones que se incluyen por defecto en todos los documentos del sitio web



Cómo incluir Javascript en documentos XHTML

El principal inconveniente es que si se quiere hacer una modificación en el bloque de código, es necesario modificar todas las páginas que incluyen ese mismo bloque de código JavaScript



Cómo incluir Javascript en documentos XHTML

DEFINIR JAVASCRIPT EN UN ARCHIVO EXTERNO

Las instrucciones JavaScript se pueden incluir en un archivo externo de tipo JavaScript que los documentos XHTML enlazan mediante la etiqueta `<script>`



Cómo incluir Javascript en documentos XHTML

Se pueden crear todos los archivos JavaScript que sean necesarios y cada documento XHTML puede enlazar tantos archivos JavaScript como necesite

Ejemplo:

Archivo codigo.js

```
console.log("Un mensaje de prueba");
```



Cómo incluir Javascript en documentos XHTML

```
<head>
```

```
  <meta http-equiv="Content-Type" content="text/html;  
  charset=UTF-8" />
```

```
  <title>Ejemplo de código JavaScript en el propio  
  documento</title>
```

```
  <script type="text/javascript" src="/js/codigo.js"></  
  script>
```

```
</head>
```



Cómo incluir Javascript en documentos XHTML

Además del atributo type, este método requiere definir el atributo src, que es el que indica la URL correspondiente al archivo JavaScript que se quiere enlazar

Cada etiqueta <script> solamente puede enlazar un único archivo, pero en una misma página se pueden incluir tantas etiquetas <script> como sean necesarias



Cómo incluir Javascript en documentos XHTML

Los archivos de tipo JavaScript son documentos normales de texto con la extensión .js, que se pueden crear con cualquier editor de texto como Notepad, Wordpad, EmEditor, UltraEdit, Vi, etc...



Cómo incluir Javascript en documentos XHTML

La principal ventaja de enlazar un archivo JavaScript externo es que se simplifica el código XHTML de la página, que se puede reutilizar el mismo código JavaScript en todas las páginas del sitio web y que cualquier modificación realizada en el archivo JavaScript se ve reflejada inmediatamente en todas las páginas XHTML que lo enlazan



Cómo incluir Javascript en documentos XHTML

INCLUIR JAVASCRIPT EN LOS ELEMENTOS XHTML

Este último método es el menos utilizado, ya que consiste en incluir instrucciones JavaScript dentro del código XHTML de la página



Cómo incluir Javascript en documentos XHTML

```
<body>
```

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

```
</body>
```



Cómo incluir Javascript en documentos XHTML

El mayor inconveniente de este método es que ensucia innecesariamente el código XHTML de la página y complica el mantenimiento del código JavaScript

En general, este método sólo se utiliza para definir algunos eventos y en algunos otros casos especiales, como se verá más adelante



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



Etiqueta noscript

En estos casos, es habitual que si la página web requiere JavaScript para su correcto funcionamiento, se incluya un mensaje de aviso al usuario indicándole que debería activar JavaScript para disfrutar completamente de la página

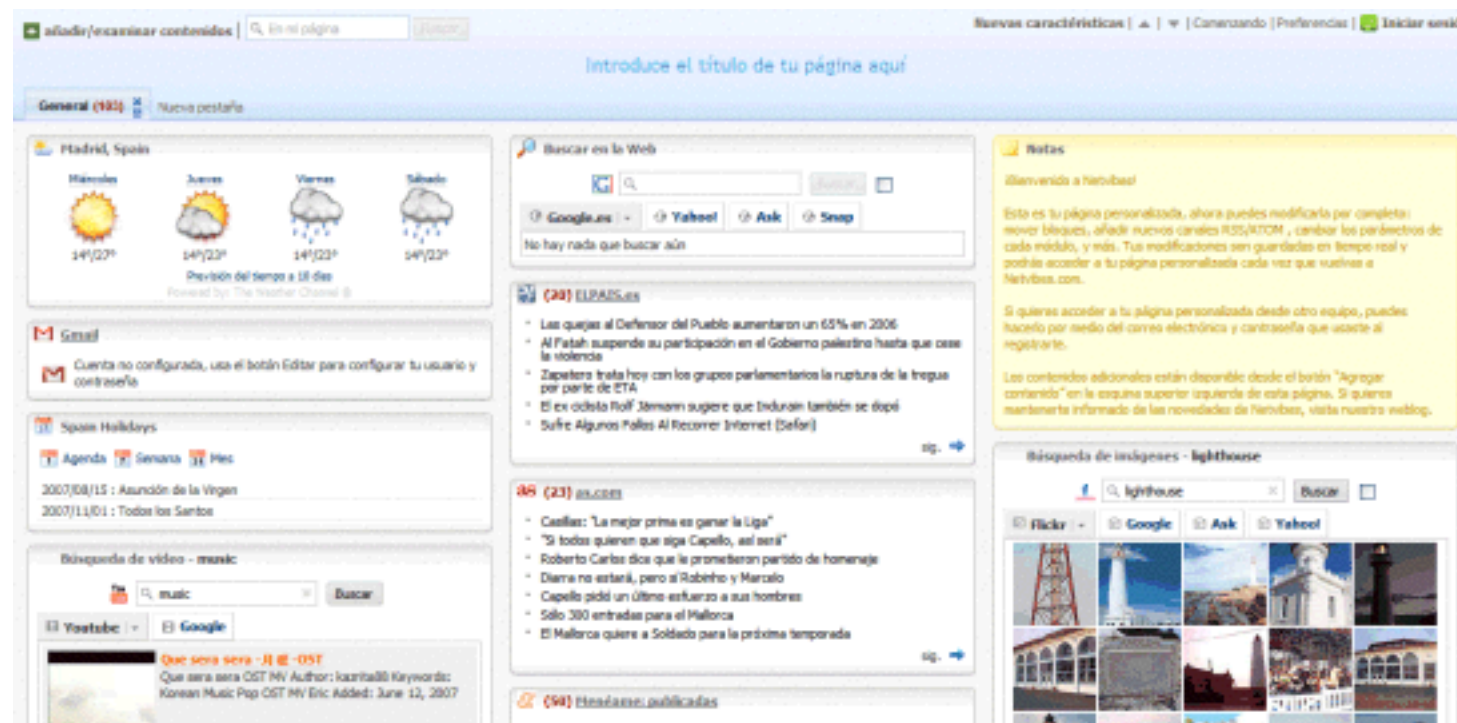


Etiqueta noscript

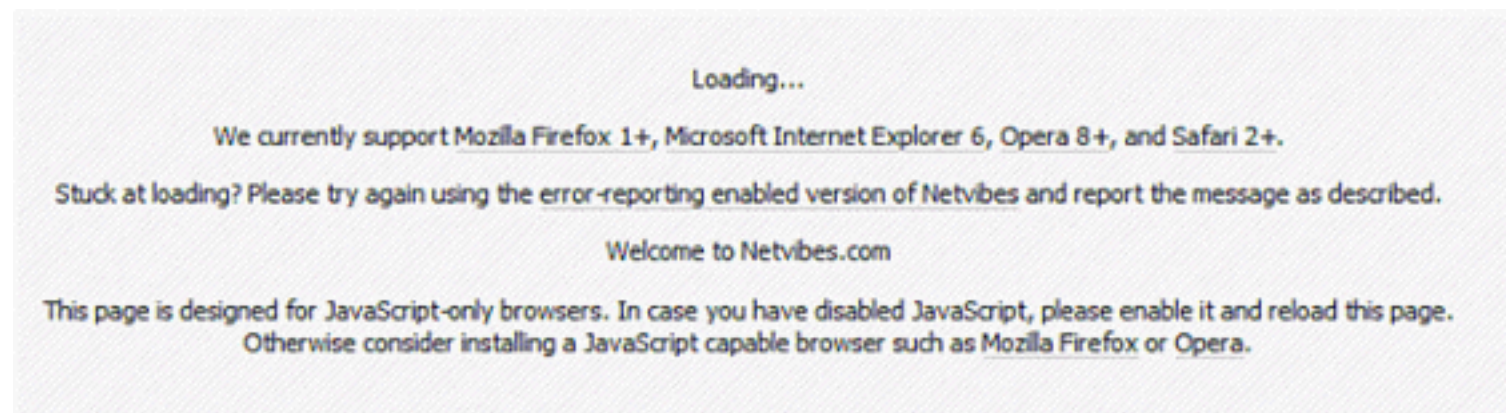
El siguiente ejemplo muestra una página web basada en JavaScript cuando se accede con JavaScript activado y cuando se accede con JavaScript completamente desactivado



Etiqueta noscript



Etiqueta noscript



Etiqueta noscript

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>`



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>
```

<http://cursosdedesarrollo.com/>



Etiqueta noscript

La etiqueta `<noscript>` se debe incluir en el interior de la etiqueta `<body>` (normalmente se incluye al principio de `<body>`)

El mensaje que muestra `<noscript>` puede incluir cualquier elemento o etiqueta XHTML



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



Glosario Básico

Sentencia: cada una de las instrucciones que forman un script



Glosario Básico

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



Glosario Básico

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



Sintaxis

La sintaxis de un lenguaje de programación se define como el conjunto de reglas que deben seguirse al escribir el código fuente de los programas para considerarse como correctos para ese lenguaje de programación



Sintaxis

La sintaxis de JavaScript es muy similar a la de otros lenguajes de programación como Java y C. Las normas básicas que definen la sintaxis de JavaScript son las siguientes



Sintaxis

No se tienen en cuenta los espacios en blanco y las nuevas líneas: como sucede con XHTML, 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.)



Sintaxis

Se distinguen las mayúsculas y minúsculas: al igual que sucede con la sintaxis de las etiquetas y elementos XHTML. Sin embargo, si en una página XHTML 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



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



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 (;)



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



Sintaxis

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

Ejemplo de comentario de una sola línea:

```
// a continuación se muestra un mensaje  
  
console.log("mensaje de prueba");
```

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



Sintaxis

Ejemplo de comentario de varias líneas:

```
/* Los comentarios de varias líneas son muy útiles  
cuando se necesita incluir bastante información  
en los comentarios */  
console.log("mensaje de prueba");
```

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



Posibilidades y 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



Posibilidades y Limitaciones

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



Posibilidades y 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

De esta forma, los scripts de JavaScript no pueden comunicarse con recursos que no pertenezcan al mismo dominio desde el que se descargó el script



Posibilidades y Limitaciones

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 (aunque los detalles concretos dependen de cada navegador)



Posibilidades y 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



Posibilidades y Limitaciones

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



Posibilidades y Limitaciones

A pesar de todo, existen alternativas para poder saltarse algunas de las limitaciones anteriores. La alternativa más utilizada y conocida consiste en firmar digitalmente el script y solicitar al usuario el permiso para realizar esas acciones



Conclusiones

Hemos visto cómo funciona
de manera básica
Javascript y cómo incluirlo
en ficheros XHTML

<http://cursosdedesarrollo.com/>



Datos de Contacto

<http://www.cursosdedesarrollo.com>
info@cursosdedesarrollo.com

<http://cursosdedesarrollo.com/>



Licencia



David Vaquero Santiago

Esta obra está bajo una
Licencia Creative Commons Atribución-
NoComercial-CompartirIgual 4.0
Internacional

Derivada de:

<http://www.arkaitzgarro.com/javascript/>

y

<http://javiereguiluz.com/>

<http://cursosdedesarrollo.com/>

