

Programación Web

Tema 3.1 Java Script - Introducción y Fundamentos

Miguel Ángel Manso
Emerson Castañeda
Ramón Alcarria
ETSI en Topografía, Geodesia y Cartografía - UPM

Universidad Politécnica de Madrid Emerson Castañeda/Miguel Ángel Manso

Contenido

- ¿Qué es JavaScript?
- Historia
- JavaScript en documentos HTML
- Etiqueta *noscript*
- Palabras reservadas
- Sintaxis
- Posibilidades y restricciones
- JavaScript en otros entornos

Universidad Politécnica de Madrid Emerson Castañeda/Miguel Ángel Manso

¿Qué es JavaScript? (I)

- **JavaScript** es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas e interactivas

El dinamismo en una página web incluye: efectos en el texto, animaciones, acciones que se activan al pulsar botones, ventanas con mensajes de aviso al usuario, validaciones, y el proporcionar mecanismos interactivos para la transmisión de información entre la página web y el internauta, y viceversa

- Es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos

*Los programas escritos con **JavaScript** se pueden probar directamente en el navegador sin necesidad de procesos intermedios*

Universidad Politécnica de Madrid Emerson Castañeda/Miguel Ángel Manso

¿Qué es JavaScript? (II)

- Implementación del estándar ECMAScript
 - **Scripting Language Standardized** por la **Ecma International Specification ECMA-262** y por **ISO/IEC 16262**
 - <http://www.ecma-international.org/publications/standards/Ecma-262.htm>
 - Edition 5.1 (June 2011)
 - Dialectos conocidos: JavaScript, JScript, y ActionScript
- Características:
 - Orientado a objetos
 - Basado en prototipos
 - Imperativo
 - Débilmente *tipado* (valores y expresiones en los tipos de datos)
 - Dinámico

Universidad Politécnica de Madrid Emerson Castañeda/Miguel Ángel Manso

¿Qué es JavaScript? (II)

- Más características
- Permite mayor interacción del usuario con el HTML: JavaScript permite mejorar la comunicación entre eventos producidos entre la página web y el usuario, por ejemplo: *"el usuario ha entrado dentro de una caja de texto, ¿realizamos alguna acción, por ejemplo, de cambio de color de la caja?"*
- Muy fácil de incrustar en una página HTML
- Licencia libre: no es necesario pagar por hacer uso del mismo

Universidad Politécnica de Madrid Emerson Castañeda/Miguel Ángel Manso

¿Cuándo usar JavaScript?

- Queremos insertar texto y formato de manera dinámica
 - *Si el usuario modifica un valor en una caja de texto queremos mostrarle un mensaje diciendo que este texto no cumple el formato*
- Queremos escuchar eventos de un usuario
 - *Cuando el usuario hace click en una tabla queremos modificar el color de la misma para que vea que la va a modificar*
- Queremos validar datos introducidos por el usuario
 - *Un usuario de una página web, después de rellenar un formulario, selecciona la opción de "Enviar". Antes de realizar la petición al servidor, JavaScript puede recoger este formulario, chequear los datos, y comprobar si estos son correctos*

Universidad Politécnica de Madrid Emerson Castañeda/Miguel Ángel Manso

¿Cuándo usar JavaScript?

- Queremos detectar el navegador del visitante
 - *Javascript puede detectar el navegador que ha llamado a nuestra página web, y poder adaptar el formato de la misma al formato que más le convenga al navegador*
- Queremos crear y manipular cookies
 - *Las cookies sirven para registrar las acciones de un usuario sobre una página Web, como la identidad, la fecha de la última visita etc.*
 - *Javascript es capaz de generar cookies por ejemplo con la información de usuario y contraseña de un usuario, para que no tenga que iniciar sesión cada vez que visite la página.*

Universidad Politécnica de Madrid Emerson Castañeda/Miguel Ángel Manso

Historia (I)

- Desarrollado por **Brendan Eich** de **Netscape** con el nombre de **Mocha**
- Renombrado posteriormente a **LiveScript**
- El cambio de nombre a **JavaScript** coincidió aproximadamente con el momento en que **Netscape** agregó soporte para la tecnología Java en su navegador web **Netscape Navigator** (Ver. 2.003 en Dic. de 1995)
- El cambio de nombre produjo confusión (hasta nuestros días), dando la impresión de que el lenguaje es una prolongación de **Java**
 - *Una estrategia de mercadotecnia de Netscape para obtener prestigio e innovar en lo que eran los nuevos lenguajes de programación web*

Universidad Politécnica de Madrid Emerson Castañeda/Miguel Ángel Manso

Historia (II)

- **JavaScript** es una marca registrada de **Oracle Corporation**. Es usada con licencia por los productos creados por Netscape Communications y la Fundación Mozilla
- Microsoft dio como nombre a su implementación de **JavaScript** «JScript», para evitar problemas relacionadas con la marca
- **JScript** fue adoptado en la versión 3.0 de **IE**, liberado en agosto de 1996, e incluyó como novedad la compatibilidad con el Efecto 2000
- Los dialectos parecen similares por lo tanto los términos «**JavaScript**» y «**JScript**» a menudo se utilizan indistintamente, pero la especificación de **JScript** es incompatible con la de **ECMA** en muchos aspectos

Universidad Politécnica de Madrid Emerson Castañeda/Miguel Ángel Manso

Historia (III)

- El **W3C** diseñó el estándar Document Object Model (**DOM**) con el propósito de aminorar las incompatibilidades
- **DOM** se incorpora en **Konqueror**, las versiones 6 de **IE** y **Netscape Navigator**, **Opera** la versión 7, **Mozilla Firefox**
- En 1997 fue adoptado como un estándar **ECMA**, con el nombre de **ECMAScript**
- Como estándar **ISO: ISO/IEC 16262 (1998, 2002, 2011)**
 - http://www.iso.org/iso/iso_catalogue/catalogue_ics/catalogue_detail_ics.htm?csnumber=29696
 - http://www.iso.org/iso/iso_catalogue/catalogue_ics/catalogue_detail_ics.htm?csnumber=33835
 - http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=55755

Universidad Politécnica de Madrid Emerson Castañeda/Miguel Ángel Manso

JavaScript en documentos HTML (I)

- La integración de **JavaScript** y **HTML** es flexible, existen tres formas generales para incluir código **JavaScript** en las páginas web
 - Incluir código **JavaScript** en el mismo documento **HTML**
 - Definir código **JavaScript** en un archivo externo
 - Incluir código **JavaScript** en los elementos **HTML**

Universidad Politécnica de Madrid Emerson Castañeda/Miguel Ángel Manso

JavaScript en documentos HTML (II)

- **JavaScript** en el mismo documento **HTML**
 - El código **JavaScript** se incluye entre etiquetas **<script>** **</script>**
 - Se puede incluir cualquier bloque de código en cualquier zona de la página
 - Se recomienda definir el código **JavaScript** dentro de la cabecera del documento **<head>** **</head>**

```

1 <html>
2 <head>
3   <title>Ejemplo de código JavaScript en el propio documento</title>
4   <script type="text/javascript">
5     alert("Un mensaje de prueba");
6   </script>
7 </head>
8 <body>
9   <p>Un párrafo de texto.</p>
10 </body>
11 </html>

```

Universidad Politécnica de Madrid Emerson Castañeda/Miguel Ángel Manso

JavaScript en documentos HTML (II)

- **JavaScript** en el mismo documento **HTML**
 - También incluyendo código Javascript en el body
 - **JavaScript** dentro de la cabecera del documento **<head>**
</head>

```

1 <html>
2 <head>
3   <title>Ejemplo de código JavaScript en el propio documento</title>
4   <script type="text/javascript">
5     alert("Un mensaje de prueba");
6   </script>
7 </head>
8 <body>
9   <p>Un párrafo de texto.</p>
10 </body>
11 </html>

```

Universidad Politécnica de Madrid Emerson Castañeda/Miguel Ángel Manso

JavaScript en documentos HTML (III)

- Código **JavaScript** en un archivo externo
 - Las instrucciones **JavaScript** se pueden incluir en un archivo externo de tipo **js**, ejemplo: codigo.js
 - Los documentos **HTML** enlazan mediante la etiqueta **<script>**
`1 alert("Un mensaje de prueba");`
 - Se pueden crear tantos archivos **JavaScript** como sean necesarios y cada documento **HTML** puede enlazar tantos como requiera

```

1 <html>
2 <head>
3   <title>Ejemplo de código JavaScript en el propio documento</title>
4   <script type="text/javascript" src="/js/codigo.js"></script>
5 </head>
6 <body>
7   <p>Un párrafo de texto.</p>
8 </body>
9 </html>

```

Universidad Politécnica de Madrid Emerson Castañeda/Miguel Ángel Manso

JavaScript en documentos HTML (IV)

- Código **JavaScript** en los elementos **HTML**
 - Consiste en incluir trozos de **JavaScript** dentro de las etiquetas de código **HTML** de la página
 - Este último método, es el menos usado

```

1 <html>
2 <head>
3   <title>Ejemplo de código JavaScript en el propio documento</title>
4   <script type="text/javascript" src="/js/codigo.js"></script>
5 </head>
6 <body>
7   <p onclick="alert('Un mensaje de prueba')">Un párrafo de texto.</p>
8 </body>
9 </html>

```

Universidad Politécnica de Madrid Emerson Castañeda/Miguel Ángel Manso

Etiqueta noscript (I)

- Se usa cuando el navegador no dispone de soporte completo de **JavaScript**, o los usuarios bloquean parcial o completamente el uso del mismo
- Se incluye un mensaje de aviso al usuario indicándole que debería activar **JavaScript** mediante el uso de la etiqueta **<noscript>**

```

1 <head> ... </head>
2 <body>
3 <noscript>
4   <p>Bienvenido a Mi Sitio</p>
5   <p>La página que estás viendo requiere para su funcionamiento
6     el uso de JavaScript.
7     Si lo has deshabilitado intencionadamente, por favor vuelve
8     a activarlo.
9   </p>
10 </noscript>
11 </body>

```

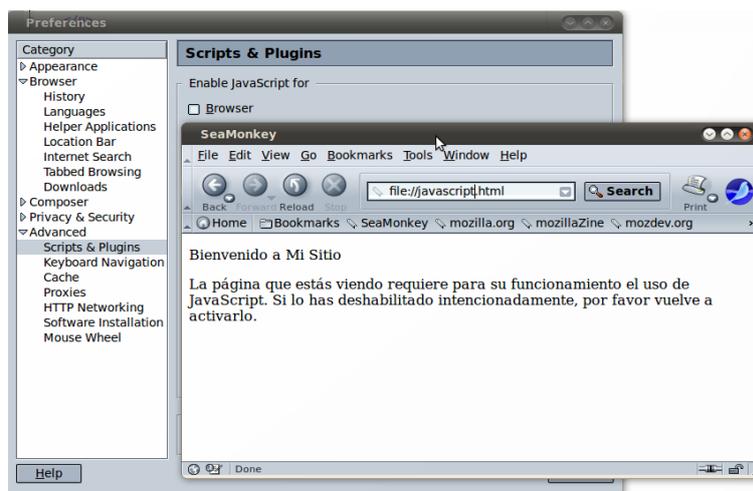
Universidad Politécnica de Madrid Emerson Castañeda/Miguel Ángel Manso

Etiqueta noscript (II)

- 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**

Universidad Politécnica de Madrid Emerson Castañeda/Miguel Ángel Manso

Etiqueta noscript (III)



Universidad Politécnica de Madrid Emerson Castañeda/Miguel Ángel Manso

Palabras Reservadas

- Son palabras, en inglés, que se utilizan para construir las sentencias de **JavaScript** y por tanto no pueden utilizarse libremente
- Son palabras reservadas en **JavaScript**:

**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.**

Universidad Politécnica de Madrid Emerson Castañeda/Miguel Ángel Manso

Sintaxis (I)

- 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.** El código se puede ordenar de forma adecuada para entenderlo mejor
 - **Se distinguen las mayúsculas y minúsculas (Case-Sense).** Si en JavaScript se intercambian mayúsculas y minúsculas el script no funciona

Universidad Politécnica de Madrid Emerson Castañeda/Miguel Ángel Manso

Sintaxis (II)

- **No es necesario definir el tipo de dato de las variables.** De esta forma, una misma variable puede almacenar diferentes tipos de datos durante la ejecución del script
- **No es necesario terminar cada sentencia con el carácter de punto y coma (;).** En algunos lenguajes de programación, es obligatorio terminar cada sentencia con (;). Aunque JavaScript no obliga a hacerlo, es aconsejable terminar cada sentencia con el carácter del punto y coma

Universidad Politécnica de Madrid Emerson Castañeda/Miguel Ángel Manso

Sintaxis (III)

- **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** define comentarios de una sola línea y los que ocupan varias líneas (`//` y `/* */`)

```

1 // Comentario de una sola línea: a continuación se muestra un mensaje
2 alert("mensaje de prueba");
3
4 /* Los comentarios de varias líneas son muy útiles
5 cuando se necesita incluir bastante información
6 en los comentarios */
7 alert("mensaje de prueba");

```

Universidad Politécnica de Madrid Emerson Castañeda/Miguel Ángel Manso

Posibilidades y Restricciones (I)

- **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
 - No pueden comunicarse con recursos que no pertenezcan al mismo dominio desde el que se descargó el script (*cross-domain*)
 - No pueden cerrar ventanas que no hayan abierto ellos mismos
 - Las ventanas que se crean no pueden ser demasiado pequeñas ni demasiado grandes ni colocarse fuera de la vista del usuario

Universidad Politécnica de Madrid Emerson Castañeda/Miguel Ángel Manso

Posibilidades y Restricciones (II)

- No pueden acceder a los archivos del ordenador del usuario (ni en modo lectura ni en modo escritura)
- No pueden leer o modificar las preferencias del navegador
- Si la ejecución de un script dura demasiado tiempo, el navegador informa al usuario de que un script está consumiendo demasiados recursos y le da la posibilidad de detener su ejecución
- La alternativa más utilizada y conocida para saltarse algunas de las anteriores limitaciones consiste en firmar digitalmente el script y solicitar al usuario el permiso para realizar esas acciones

Universidad Politécnica de Madrid Emerson Castañeda/Miguel Ángel Manso

JavaScript en otros entornos (I)

- La popularidad de **JavaScript** como lenguaje se ha extendido a otras aplicaciones y otros entornos que no se relacionan directamente con las paginas web
 - Aplicaciones como **Yahoo Widgets** ó el **Dashboard de Apple** utilizan **JavaScript** para programar sus Widgets
 - <http://widgets.yahoo.com/>
 - <http://www.apple.com/downloads/dashboard/>
 - Herramientas como **Adobe Acrobat** permiten incluir código **JavaScript** en archivos **PDF**
 - **Adobe Flash** y **Flex** utilizan **ActionScript**, un dialecto del mismo estándar

Universidad Politécnica de Madrid Emerson Castañeda/Miguel Ángel Manso

JavaScript en otros entornos (II)

- **Photoshop** permite realizar pequeños scripts mediante **JavaScript**
- En la versión 6 de Java se incluye un nuevo paquete denominado **javax.script** que permite integrar ambos lenguajes
- Las nuevas máquinas virtuales de **JavaScript** y los frameworks basados en ellas, también han aumentado la popularidad de **JavaScript** para aplicaciones web del lado del servidor

- Node.js <http://nodejs.org/> - CommonJS
www.commonjs.org/



Universidad Politécnica de Madrid Emerson Castañeda/Miguel Ángel Manso

Herramientas

- Desarrollo para escritorio: Eclipse, NetBeans, Aptana Studio, PHP Storm
- Desarrollo en Web: <http://jsbin.com/>
- Depuración en Web: Firebug para Firefox y Developer Tools para Chrome (F12)