



JAVASCRIPT

Capítulo 1

Lenguajes de Marcas y Sistemas de Gestión de Información

Curso 2019/2020



Capítulo 1: INTRODUCCIÓN

1. ¿Qué es JavaScript?
2. Especificaciones Oficiales
3. Cómo incluir JavaScript en documentos XHTML
4. Etiqueta NoScript
5. Glosario Básico
6. Sintaxis
7. Posibilidades y Limitaciones
8. JavaScript y Navegadores
9. JavaScript en Otros Entornos



1. ¿QUÉ ES JAVASCRIPT?

Es un lenguaje de programación que se utiliza principalmente para crear **páginas web dinámicas**.

- Incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario.

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 cualquier navegador sin necesidad de procesos intermedios.



2. ESPECIFICACIONES OFICIALES

ECMA (European Computer Manufacturers Association) ha publicado varios estándares relacionados con ECMAScript.

- **1ª edición** del estándar ECMA-262 (Junio 1997)
- **2ª edición:** Se realizaron pequeñas modificaciones para adaptarlo al estandar ISO/IEC-16262 (Junio 1998)
- **3ª edición** del estándar ECMA-262 (Diciembre 1999) es la versión que utilizan los navegadores actuales
- Actualmente se encuentra en desarrollo la **4ª versión** de ECMA-262, que podría incluir novedades como paquetes, *namespaces*, definición explícita de clases, etc.
- Estándar **ECMA-357**, que define una extensión conocida como E4X y que permite la integración de JavaScript y XML.



3. 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:

- Mismo documento XHTML
- Definir JavaScript en un Archivo Externo
- Incluir JavaScript en los elementos XHTML



3.1. INCLUSIÓN EN EL MISMO DOCUMENTO

El código JavaScript se encierra entre etiquetas `<script>` y se incluye en cualquier parte del documento. Se recomienda definir el código JavaScript dentro de la cabecera del documento:

```
<html >

  <head>

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

    <script type="text/javascript">

      alert("Un mensaje de prueba");

    </script>

  </head>

  <body>

    <p>Un párrafo de texto.</p>

  </body>

</html>
```



3.2. DEFINIR JAVASCRIPT EN UN ARCHIVO EXTERNO

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

Archivo *codigo.js*

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

Documento XHTML

```
<html>

  <head>

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

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

  </head>

  <body>

    <p>Un párrafo de texto.</p>

  </body>

</html>
```



3.3. INCLUIR JAVASCRIPT EN LOS ELEMENTOS XHTML

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

```
<html>

    <head>

        <title>Ejemplo      JavaScript      en      el      propio
documento</title>

    </head>

    <body>

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

    </body>

</html>
```




4. ETIQUETA NOSCRIPT

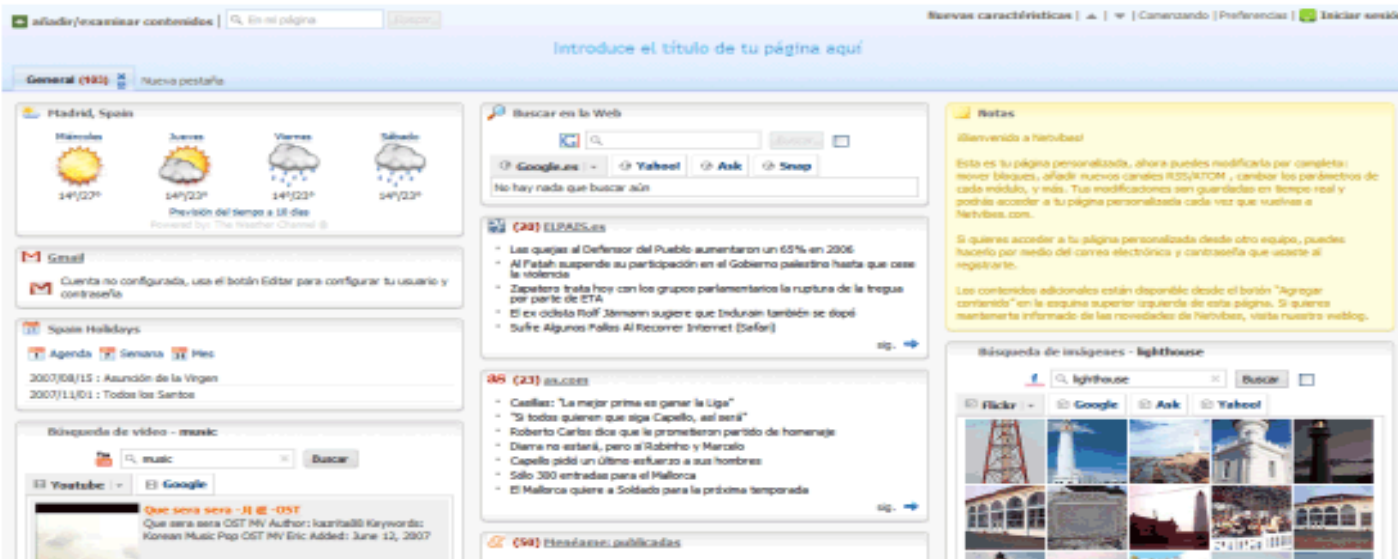
- Algunos navegadores no disponen de **soporte completo** de JavaScript.
- Otros navegadores permiten **bloquearlo parcialmente**.
- Algunos usuarios **bloquean completamente** el uso de JavaScript porque creen que así navegan de forma más segura.

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.

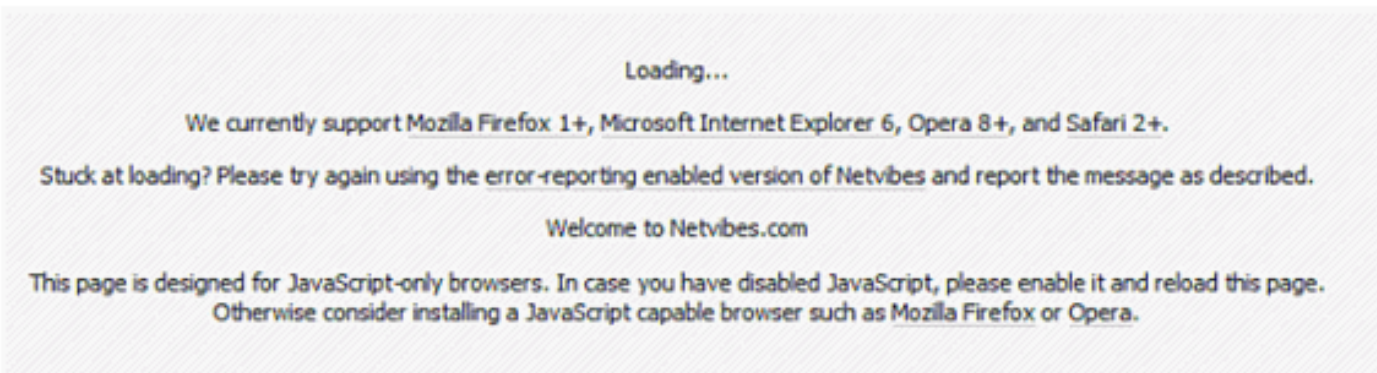


4. ETIQUETA NOSCRIPT

Web con JavaScript activado



Web con JavaScript desactivado





4. ETIQUETA NOSCRIPT

El lenguaje HTML define la etiqueta `<noscript>` para mostrar un mensaje al usuario cuando su navegador no puede ejecutar JavaScript:

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

```
    <body>
```

```
        <noscript>
```

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

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

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

```
        </noscript>
```

```
    </body>
```

La etiqueta `<noscript>` se debe incluir en el interior de la etiqueta `<body>`. El mensaje que muestra `<noscript>` puede incluir cualquier elemento o etiqueta XHTML.

5. GLOSARIO BÁSICO

- **Script:** cada uno de los programas, aplicaciones o trozos de código creados con el lenguaje de programación JavaScript.
- **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.`

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.

La sintaxis de JavaScript es muy **similar** a la de otros lenguajes de programación como Java y C.

6. SINTAXIS: NORMAS

- **No se tienen en cuenta los espacios en blanco y las nuevas líneas:** el intérprete de JavaScript ignora cualquier espacio en blanco sobrante, por lo que el código se puede ordenar de forma adecuada para entenderlo mejor.
- **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**.

6. SINTAXIS: NORMAS

- **No se define el tipo de las variables:** al crear una variable, no es necesario indicar el tipo de dato que almacenará → Una misma variable puede almacenar diferentes tipos de datos durante la ejecución del script.
- **No es necesario terminar cada sentencia con punto y coma (;):** en los lenguajes de programación es obligatorio terminar cada sentencia con dicho carácter. JavaScript no obliga a hacerlo, pero es conveniente seguir la tradición de terminar cada sentencia con el carácter punto y coma.
- **Se pueden incluir 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.

6. SINTAXIS: COMENTARIOS

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

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

Ejemplo de comentario de varias líneas

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

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



7. POSIBILIDADES Y LIMITACIONES

Desde su aparición, JavaScript siempre fue utilizado de forma masiva por la mayoría de sitios de Internet hasta la aparición de Flash. Sin embargo, la aparición de las aplicaciones AJAX programadas con JavaScript le ha devuelto una popularidad dentro de los lenguajes de programación web.

LIMITACIONES

- Fue diseñado de forma que se ejecutara en un **entorno muy limitado** que permitiera a los usuarios confiar en la ejecución de los scripts.
- Los scripts **no pueden comunicarse** con recursos que no pertenezcan al **mismo dominio** desde el que se descargó el script.
- Los scripts **no pueden acceder** a los archivos del ordenador del usuario y tampoco 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 permite detener su ejecución.

8. JAVASCRIPT Y NAVEGADORES

Los navegadores más modernos disponibles actualmente incluyen soporte de JavaScript hasta la versión correspondiente a la **3ª edición** del estándar **ECMA-262**.

La mayor diferencia reside en el **dialecto** utilizado.

JScript



JavaScript



+ 9. JAVASCRIPT EN OTROS ENTORNOS

La inigualable popularidad de JavaScript como lenguaje de programación de aplicaciones web se ha extendido a **otras aplicaciones y otros entornos** no relacionados con la web.

- **Adobe Acrobat** permiten incluir código JavaScript en archivos PDF.
- Otras herramientas de Adobe como **Flash** utilizan *ActionScript*, un dialecto del mismo estándar de JavaScript.
- **Photoshop** permite realizar pequeños scripts mediante JavaScript.
- La versión 6 de **Java** incluye un nuevo paquete (denominado *javax.script*) que permite integrar ambos lenguajes.
- Aplicaciones como **Yahoo Widgets** y el **Dashboard de Apple** utilizan JavaScript para programar sus *widgets*.