

# Tema 1. Introducción

---

- JavaScript
- Características del lenguaje
- Acciones y limitaciones
- Un poco de historia
- La etiqueta *<script>*
- La etiqueta *<noscript>*
- Hola Mundo
- JavaScript en un archivo externo

# JavaScript

---

- JavaScript es un lenguaje de *scripts* desarrollado por Netscape para incrementar las funcionalidades del lenguaje HTML. La funcionalidad que incorpora el lenguaje dota de dinamismo a las páginas Web.
- Una página web dinámica es aquella que incorpora efectos como:
  - Texto que aparece y desaparece
  - Animaciones
  - Acciones que se activan al pulsar botones
  - Ventanas con mensajes de aviso al usuario
  - Validación de formularios, etc
- JavaScript no es Java. La razón de su nombre fue exclusivamente por marketing, ya que Java era la palabra de moda en el mundo informático y de Internet de la época.

# Características del lenguaje

- Es un lenguaje interpretado, es decir, no requiere compilación.
  - El navegador del usuario se encarga de interpretar las sentencias JavaScript contenidas en una página HTML y ejecutarlas adecuadamente.
  - Es un lenguaje del lado cliente.
- Es un lenguaje orientado a objetos.
  - El modelo de objetos de JavaScript está reducido y simplificado.
  - No dispone de forma nativa ni de herencia ni de ámbito.
  - Incluye objetos predefinidos para que los *scripts* puedan acceder a la información de una página y puedan actuar sobre la interfaz del navegador.
- Es un lenguaje orientado a eventos.
  - Cuando un usuario pincha sobre un enlace, pulsa una tecla o mueve el puntero sobre un elemento de la página Web, se produce un evento.
  - El lenguaje es capaz de capturar el evento y de realizar la acción asociada al mismo.

# Acciones y limitaciones

- Lo que puede hacer JavaScript
  - Resaltar, hacer parpadear o cambiar la imagen de un elemento de la página al pasar el cursor por encima
  - Validar un formulario y/o realizar los cálculos necesarios antes de enviarlo
  - Crear páginas personalizadas. Según el comportamiento del usuario, la página se modifica cambiando sus elementos e incluso creando páginas nuevas
  - Detectar plug-ins instalados y requerir su instalación, modo de visualización de la pantalla, navegador utilizado, etc.
- Lo que no puede hacer JavaScript
  - No permite escribir ni leer archivos en la máquina cliente. Sólo *cookies* y con limitaciones.
  - No permite escribir ni leer archivos en el servidor
  - No puede cerrar ventanas que no haya abierto

# Un poco de historia

- A principios de los años 90, con unas aplicaciones web cada vez más complejas y con poco ancho de banda, surgió la necesidad de un lenguaje de programación que se ejecutara en el navegador del usuario.
- 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.
- 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*.
- Posteriormente, Netscape firmó una alianza con Sun Microsystems para el desarrollo del nuevo lenguaje de programación.
- 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.
- 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*).
- 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.

# La etiqueta `<script>`

---

- `<script>...</script>`
- Sirve para definir bloques de código de *script* que se deban ejecutar en nuestra página. Este código puede estar escrito en diferentes lenguajes de programación.
- El código puede incluirse en cualquier parte del documento. No obstante, se recomienda:
  - Incluir el código JavaScript con las definiciones (variables, funciones) dentro de la cabecera del documento (`<head>`)
  - Incluir el código con la ejecución dentro del bloque `<body>`
- Se pueden incluir tantas etiquetas `<script>` como se desee.

# Atributos de <script>

- *language=...*
  - Especifica el lenguaje de scripts de los contenidos de este elemento. Su valor es un identificador del lenguaje, pero debido a que estos identificadores no son estándar, este atributo ha sido desaprobado en favor de *type*.
- *type=...*
  - Indica el lenguaje de scripts de los contenidos del elemento y prevalece sobre el lenguaje de scripts por defecto. El lenguaje de scripts se especifica como un tipo de contenido.
  - Valores:
    - text/javascript (JavaScript)
    - text/jscript (versión Microsoft de JavaScript)
    - text/vbscript (VBScript)
    - text/php (PHP).
- *src=...*
  - Especifica el nombre del archivo que contiene el código de script.

# La etiqueta *<noscript>*

- *<noscript>*
  - Proporciona código HTML alternativo para los navegadores que no soportan el lenguaje de *script* utilizado o lo tienen bloqueado.
  - Si el navegador interpreta JavaScript, ignorará esta etiqueta. En caso contrario, ignorará la etiqueta *<script>* e interpretará el código HTML de la etiqueta *<noscript>*.
- En estos casos, es conveniente ocultar el código JavaScript. Para ellos, incluiremos el código entre comentarios.

```
<script type="text/javascript">  
<!--  
...  
//-->  
</script>
```



# Hola Mundo

```
<html>
<head>
<title>Exemple 1</title>
</head>
<body>
<script type="text/javascript">
<!--
window.alert("Hola Mundo");
//-->
</script>
<noscript>
<p>Esta página requiere JavaScript</p>
</noscript>
</body>
</html>
```

# JavaScript en un archivo externo

- Las instrucciones JavaScript se pueden incluir en un archivo externo de extensión *js* 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.

```
<html>
<head>
<title>Exemple 1</title>
<script type="text/javascript" src="miscript.js">
</script>
</head>
<body>
<h3>La página del "Hola Mundo"</h3>
</body>
</html>
```

```
<!--
window.alert("Hola Mundo");
//→
```