

CAPÍTULO 12

PROGRAMACIÓN DE DOCUMENTOS WEB
UTILIZANDO LENGUAJES DE SCRIPT DEL
CLIENTE

OBJETIVOS

- Diferenciar ejecución web del lado del cliente y del servidor.
- Introducir el modelo de objetos de documento DOM.
- Conocer el lenguaje de programación web del lado del cliente: JavaScript.
- Realizar ejemplos y prácticas sobre dicho lenguaje.

ÍNDICE

- 12.1 DIFERENCIAS ENTRE LA EJECUCIÓN EN LADO DEL CLIENTE Y DEL SERVIDOR
- 12.2 MODELO DE OBJETOS DEL DOCUMENTO DOM
- 12.3 JAVASCRIPT
- 12.4 VALIDACIÓN DE FORMULARIOS CON JAVASCRIPT
- 12.5 COMPORTAMIENTOS DINÁMICOS. CAPTURA DE EVENTOS CON JAVASCRIPT
- 12.6 LIMITACIONES Y RIESGOS DE ATAQUES

12.1 DIFERENCIAS ENTRE LA EJECUCIÓN EN LADO DEL CLIENTE Y DEL SERVIDOR

- El navegador es una aplicación capaz de interpretar las órdenes recibidas en forma de código HTML fundamentalmente y convertirlas en las páginas que son el resultado de dicha orden.
- Cuando nosotros pulsamos sobre un enlace hipertexto, en realidad lo que pasa es que establecemos una petición de un archivo HTML residente en el servidor (un ordenador que se encuentra continuamente conectado a la red) el cual es enviado e interpretado por nuestro navegador (el cliente).

- Así pues, podemos hablar de lenguajes de lado servidor que son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él. Por otro lado, los lenguajes de lado cliente (entre los cuales no solo se encuentra el HTML sino también el Java y el JavaScript, los cuales son simplemente incluidos en el código HTML) son aquellos que pueden ser directamente "digeridos" por el navegador y no necesitan un pretratamiento.

Lenguajes del lado del Cliente	Lenguajes del lado del Servidor	Lenguaje del lado del Cliente y del Servidor
HTML JavaScript Applets de Java Visual Basic Script Flash CSS	CGI PHP ASP PERL JSP	DHTML XML

12.2 MODELO DE OBJETOS DEL DOCUMENTO DOM

- El Modelo de Objetos de Documento, o DOM, es la interfaz que te permite acceder y manipular, mediante la programación, los contenidos de una página web (o documento). Proporciona una representación estructurada, orientada a objetos, de los elementos individuales y el contenido de una página, con métodos para recuperar y fijar las propiedades de dichos objetos. Además, proporciona métodos para agregar y eliminar dichos objetos, permitiéndote crear contenido dinámico.

12.3 JAVASCRIPT

- JavaScript es un lenguaje de programación creado por la empresa Netscape. Es el lenguaje de programación más utilizado en Internet para añadir interactividad a las páginas web. No confundir el JavaScript con el Java. El Java es un lenguaje de programación de propósito general como lo son el C++ o el Visual Basic.
- Un programa en JavaScript se integra en una página web (entre el código HTML) y es el navegador el que lo interpreta (ejecuta). Es decir el JavaScript es un lenguaje interpretado, no compilado (no se genera ningún tipo de fichero objeto o exe).

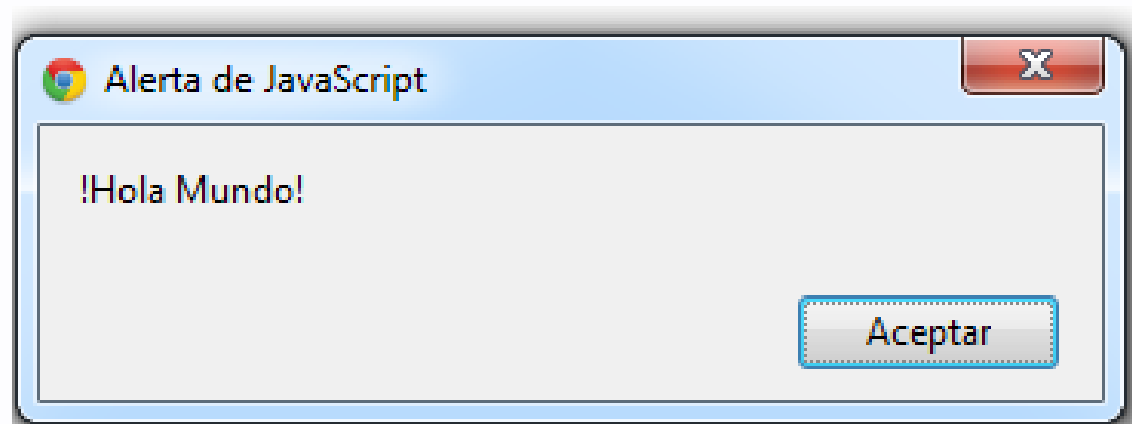
```
<SCRIPT LANGUAGE="JavaScript"> ... </SCRIPT>
```

```
<SCRIPT>...</SCRIPT>
```

12.3.2

PRIMER EJEMPLO

```
<HTML>
<HEAD>
<TITLE>EJE1.HTML</TITLE>
</HEAD>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
alert("¡Hola Mundo!");
</SCRIPT>
</BODY>
</HTML>
```



12.4 VALIDACIÓN DE FORMULARIOS CON JAVASCRIPT

12.4.1 EVENTO ONSUBMIT

```
<form name="miFormulario"  
action="mailto:mi@mail.com">  
<input type="submit" value="Enviar" name="enviar">  
</form>
```


12.4.1 EVENTO ONSUBMIT

```
<form name="miFormulario"  
action="mailto:mi@mail.com">  
onSubmit="alert('Has pulsado enviar. '); return  
false; ">  
<input type="submit" value="Enviar" name="enviar">  
</form>
```

12.4.1 EVENTO ONSUBMIT

```
<form name="miFormulario"  
action="mailto:mi@mail.com"  
onSubmit="alert('Has pulsado enviar. '); return  
true; ">  
<input type="submit" value="Enviar" name="enviar">  
</form>
```

12.4.2 EL OBJETO THIS

```
<form name="miFormulario"
action="mailto:mi@mail.com" onSubmit="return
Valida(this);">
Campo1 <input type="text" name="campo1">
<br>
Campo2 <input type="text" name="campo2">
<input type="submit" value="Enviar"
name="enviar">
</form>
```

12.4.3 VALIDACIÓN DE CAMPOS VACÍOS

```
<html>
<head>
<script>
function Valida(formulario) {
    if ((formulario.campo1.value.length == 0) ||
(formulario.campo2.value.length ==0)) {
        alert('falta informaci&oacute;n');
        return false
    }
    return true
}
</script>
```

12.4.3 VALIDACIÓN DE CAMPOS VACÍOS

```
<body>
<form name="miFormulario"
action="mailto:mi@mail.com"
onSubmit="return Valida(this);">
Campo1 <input type="text" name="campo1">
<br>
Campo2 <input type="text" name="campo2">
<br>
<input type="submit" value="Enviar" name="enviar">
</form>
</body>
</html>
```

12.4.3 VALIDACIÓN DE CAMPOS NUMÉRICOS

```
function Valida(formulario) {  
    if ((formulario.campo1.value.length == 0) ||  
        (formulario.campo2.value.length == 0)) {  
        alert('falta informaci&oacute;n');  
        return false;  
    }  
    if (isNaN(parseInt(formulario.campo2.value))) {  
        alert('el campo2 debe ser un n&uacute;mero');  
        return false;  
    }  
  
    return true  
}
```

12.4.3 VALIDACIÓN DE CAMPOS NUMÉRICOS DE LONGITUD FIJA

```
function valida2(formulario) {  
    var cp=/^( [0-9]{5,5})|^)$/;  
    var dni=/^( [0-9]{8,8}\-[A-Z])|^)$/;  
    if (!(cp.test(formulario.campo1.value))) {  
        alert('Contenido del campo1 no es un  
C&Oacute;DIGO POSTAL');  
        return false; }  
  
    if (!(dni.test(formulario.campo2.value))) {  
        alert('Contenido del campo2 no es un DNI v&a  
acute;lido.');        return false; }  
    return true;  
}
```

12.4.3 VALIDACIÓN DE CAMPOS DE CORREO ELECTRÓNICO

```
function valida3(formulario) {  
    var correo /^[^@\s]+@[^\s]+(\.[^\s]+)+$/  
    if (!(correo.test(formulario.campo1.value))) {  
        alert('Contenido del campo1 no es CORREO  
ELECTRÓNICO válido.');        return false; }  
  
    return true;  
}
```


12.5 COMPORTAMIENTOS DINÁMICOS. CAPTURA DE EVENTOS CON JAVASCRIPT

Sintaxis	Descripción	Evento de...
onAbort	Ocurre cuando el usuario aborta la carga de una imagen (pulsando sobre un enlace o sobre el botón Stop).	Image.
onBlur	Ocurre cuando el usuario quita el foco de un elemento de un formulario, de una ventana o de un frame.	Button, Checkbox, Password, Radio, Reset, Select, Submit, Text, Textarea y Window.
onChange	Ocurre cuando un select, text o textArea pierde el foco y su valor ha sido modificado.	FileUpload, Select, Text y Textarea.
onClick	Ocurre cuando el usuario hace un clic sobre un enlace o un elemento de un formulario.	Button, Checkbox, Link, Radio, Reset y Submit.
onDbClick	Ocurre cuando el usuario hace un doble clic sobre un enlace u un elemento de un formulario.	Document, Area y Link.
onFocus	Ocurre cuando se produce un error en la carga de un documento o una imagen.	Button, Checkbox, FileUpload, Layer, Password, Radio, Reset, Select, Submit, Text, Textarea y Window.
onKeyDown	Ocurre cuando el usuario pulsa una tecla.	Document, Image, Link and Textarea.
onKeyPress	Ocurre cuando el usuario pulsa o mantiene pulsada una tecla.	Document, Image, Link y Textarea.
onKeyUp	Ocurre cuando el usuario libera una tecla.	Document, Image, Link y Textarea.
onLoad	Ocurre cuando el navegador ha terminado de cargar la página.	Image, Layer y Window.
onMouseDown	Ocurre cuando el usuario pulsa un botón del ratón.	Button, document y Link.
onMouseMove	Ocurre cuando el usuario mueve el ratón.	Se debe asociar explícitamente a algún objeto.
onMouseOut	Ocurre cuando el usuario mueve el ratón fuera de un objeto.	Area, Layer y Link.
onMouseOver	Ocurre cuando el usuario mueve el ratón dentro de un objeto.	Area, Layer y Link.
onMove	Ocurre cuando se mueve una ventana.	Window.
onReset	Ocurre cuando el usuario pulsa el botón reset de un formulario.	Form.
onResize	Ocurre cuando el usuario cambia el tamaño de la ventana.	Window.
onSelect	Ocurre cuando el usuario selecciona algún texto en un objeto text o textarea.	Text y Textarea.
onSubmit	Ocurre cuando el usuario envía un formulario.	Form.

```
<body onLoad="alert('est&aacute;s en mi  
p&aacute;gina web')" onResize="alert('has cambiado  
el tama&ntilde;o de la p&aacute;gina')">
```

Ejemplo de onclick


```
<a href="http://www.google.es"  
onclick="alert('adi&oacute;s')">Me voy de la  
p&aacute;gina</a>
```

```
<br><br>
```

```

```

12.6 LIMITACIONES Y RIESGOS DE ATAQUES

- JavaScript tiene algunas limitaciones, las cuales derivan de su objetivo principal, que es adicionar interactividad a las páginas web. JavaScript no puede interactuar con el servidor.
- Una de sus principales desventajas es su fortaleza: JavaScript trabaja enteramente dentro del navegador del cliente y no se comunica con el servidor ni con otras máquinas. JavaScript no puede crear gráficos: JavaScript no pueden crear sus propios gráficos, solamente puede manipular gráficos hechos en otros programas.

- JavaScript trabaja de forma diferente en diferentes navegadores.
- Este es el principal problema, su comportamiento depende del navegador.
- Esta problemática se debe a que las empresas que han introducido JavaScript lo han hecho de forma independiente, sin considerar los problemas que esto trae al programador.