TEMA 8: CONTENIDOS WEB INTERACTIVOS

# 1.- Elementos interactivos.

El empleo de los elementos interactivos no es llenar la página web con recursos gráficos en movimiento, sino que, es una forma de acercarse al usuario y hacerlo participar, intentando lograr una web más dinámica mediante el **establecimiento de un canal de comunicación con el usuario**. Si los usuarios perciben la sensación de que hay alguien ahí detrás que está pendiente de ellos, que le contesta a sus dudas, que le permite opinar, sin duda volverá.

Al utilizar los elementos interactivos, no debemos olvidar el compromiso que tenemos con la **accesibilidad** de todos usuarios a la web y con la **usabilidad** del sitio. Recuerda siempre que no podemos sacrificar aspectos tan importantes como la accesibilidad y la usabilidad en aras de una mayor interactividad.

El diseño de un sistema interactivo debe estar **centrado en el usuario** y debe permitir a cualquier persona percibir el proceso interactivo como una experiencia agradable y con el que pueda obtener los resultados esperados.

*Dotar de interactividad a una página siempre debe pensarse como algo adicional a conseguir después de que ya se ha conseguido todo lo imprescindible: estandarización, accesibilidad y usabilidad.*

## 1.1.- Elementos básicos y avanzados.

El empleo de elementos interactivos en la web, nos van a servir para tratar de fidelizar a nuestros usuarios. Hay muchos elementos que permiten interactuar con el usuario: una zona de "Deja tu comentario", "Mándanos tu opinión", etcétera.

**Un elemento interactivo es aquel que cambia cuando el usuario interactúa con él.**

Los foros, blogs, encuestas, comentarios, etcétera son elementos interactivos que puede o no tener un sitio web porque permiten la comunicación con el usuario y también entre los usuarios, pero por la definición de elemento interactivo dada en el párrafo anterior podríamos considerar también como **elementos interactivos los siguientes**:

* Los **enlaces**. Todos los elementos ancla del HTML, son elementos interactivos, puesto que, aunque no se definan estilos para ellos, el color que muestran al principio, cuando se hace clic sobre ellos y después de ser visitados, es diferente. Además, una vez que se pulsa un enlace, debería haber algún cambio en la página.
* Todos los **objetos propios** de los **formularios** son elementos interactivos:
  + **Las cajas de texto son** los elementos donde el usuario puede escribir alguna información compuesta por un número de caracteres no muy extenso. Al principio pueden contener un texto por defecto o estar vacías y, mediante la interacción del usuario que escribe, borra, inserta o pega texto en ellas, cambian su contenido.
  + Los **botones de opción** son elementos donde el usuario puede elegir una opción entre varias. También se puede dejar uno de ellos preseleccionado por defecto o bien ninguno y, es el usuario el que decide cuál seleccionar. El aspecto cuando están seleccionados es diferente de cuando no lo están.
  + **Las casillas de verificación** son elementos que el usuario puede elegir seleccionar o no. No se suelen presentar preseleccionadas por defecto, pero cuando se seleccionan cambian su aspecto.
  + Las **áreas de texto** son parecidas a las cajas de texto, pero permiten escribir varias líneas de texto. Se suelen emplear para el envío de comentarios. Pueden, al igual que las cajas, mostrar un texto por defecto que guíe al usuario sobre lo que tiene que hacer. En la imagen que ilustra este apartado se ve un área de texto que contiene el texto "Escriba aquí sus temas de interés".
  + Las **listas de opciones** es un elemento interactivo en el que el usuario puede elegir uno o varios elementos de la lista, según cómo sea su configuración, con la misma funcionalidad que los botones de opción, si sólo permite elegir una opción, o con la misma funcionalidad que las casillas de verificación, si permite seleccionar varias opciones de la lista, aunque este uso no es el más común. La diferencia está en que ocupa menos espacio en el formulario. Se suele emplear para que el usuario elija su nacionalidad, el día, el mes o el año de su nacimiento.
  + **Los botones** son los elementos que permiten al usuario **confirmar** una determinada acción. Su aspecto suele cambiar cuando se pulsa de forma que el usuario identifica dicha pulsación. Todos los formularios suelen disponer de dos botones: uno que es el encargado de ejecutar la acción cuyo nombre dependerá de la acción a ejecutar y otro, que es el encargado de limpiar los datos y dejar el formulario vacío de nuevo.

Los enlaces y objetos propios de formularios son los elementos interactivos básicos necesarios para crear los foros, blogs, etcétera.

## 1.2.- Comportamientos interactivos.

Para añadir un comportamiento interactivo a los elementos ya mencionados y que el usuario tenga la sensación de tener el control de la página podemos hacer uso, tal y como dijimos en la unidad anterior de:

* Las reglas de estilo.
* Los lenguajes de programación dinámicos.

Podemos utilizar las reglas de estilo para simular la interacción con los botones, enlaces, elementos de formulario, etc. Para ello nos servimos de las **pseudoclases** link, visited, hover, active, focus que ya vimos la unidad de hojas de estilo.

En la siguiente tabla podrás ver algunos de los muchos usos que se le pueden dar a las hojas de estilo para dotar a una interfaz de mayor interactividad.

|  |
| --- |
| *Mapas interactivos*  *Creando cualquier mapa de imagen que, al pasar el puntero del mouse por encima, despliegue una información adicional, haga una sustitución de una imagen, haga un cambio de color, etc.*  *http://green-beat.com/experiments/css\_map\_pop.php* |
| *Menús de navegación*  *El empleo de reglas de estilo para diseñar los menús de navegación es algo muy habitual para evitar la apariencia tan poco atractiva de los enlaces.*  *Estos menús se pueden crear:*   * *Utilizando las propiedades de los colores de la fuente y del fondo y variar propiedades usando las pseudoclases hover para definir los colores cuando el usuario pasa por encima o active para el momento en que el usuario hace clic sobre la opción del menú.* * *Utilizando imágenes de fondo que se pueden cambiar empleando las mismas pseudoclases* |
| *Alternancia de imágenes*  *Podemos usar las pseudoclases hover sobre un elemento de imagen para cambiar el archivo de imagen a mostrar cuando el usuario pasa el ratón por encima de la imagen* |
| *Galería de imágenes*  *CSS permite realizar algunos efectos muy interesantes sin necesidad de emplear ni una sola línea de código Javascript.*  *En el siguiente enlace puedes ver una página donde hay una galería de foros realizada sólo con CSS.*  *http://www.cssplay.co.uk/menu/gallery.html* |
| *Formularios más interactivos*  *Podemos alterar el color de fondo y de la fuente de cualquier elemento del formulario empleando las pseudoclases:*  *hover cuando el usuario pasa el ratón por encima del elemento del formulario.*  *focus cuando el usuario accede a un elemento ya sea con un clic de ratón o empleando la tecla de tabulación que se desplaza entre los diferentes elementos.* |

También podemos usar **el HTML Dinámico (DHTML**), del que la Wikipedia dice que "designa el conjunto de técnicas que permiten crear sitios web interactivos utilizando una combinación de lenguaje HTML estático, un lenguaje interpretado en el lado del cliente (como JavaScript), el lenguaje de hojas de estilo en cascada (**CSS**) y la jerarquía de objetos de un DOM".

Que el **HTML sea estático** significa que no está generado por ningún programa ejecutado en el lado del servidor escrito en un lenguaje de programación como podría ser PHP, ASP.NET o Perl, sino que se sirve el documento HTML tal cual está escrito. Todos los usuarios reciben el mismo documento.

**JavaScript** es un lenguaje de guiones que sirve para extender las capacidades del lenguaje HTML. Su sintaxis es muy simple. Es un lenguaje interpretado que se ejecuta en la máquina del usuario pudiendo éste ver su código cuando consulta el código fuente de la página. El código debe ser descargado antes de poder ser ejecutado por lo que suele ir escrito en la cabecera del documento, aunque también se pueden poner scripts en el cuerpo del documento.

El **modelo de objetos de documento** define la forma en la que se relacionan entre sí los objetos y elementos que forman parte de un documento, y su relación con el navegador. Los objetos del modelo tienen en el navegador una relación descendente entre sí. Esta relación es la que emplea Javascript para reconocer a cada objeto de una página de forma individual.

El objeto de nivel superior en el **DOM** es el Navegador en sí (Browser). El objeto del siguiente nivel en el DOM es la ventana del navegador (**Window**) y el siguiente son los propios documentos visualizados en el navegador (**Document**).

## 1.3.- Propiedades de los elementos.

El modelo **DOM** que vimos en el apartado anterior establece una **relación jerárquica** entre los elementos (objetos) que forman parte de un documento HTML y la relación de éstos con el navegador.

Cada uno de estos elementos tiene una serie de **propiedades** que se podrán modificar en función de las acciones del usuario pero para hacerlo es necesario que cada elemento esté identificado de forma unívoca.

*¿Recuerdas cómo se podía identificar un elemento del HTML de forma unívoca?*

Había que nombrar el elemento mediante el atributo id.

Cuando un elemento está identificado mediante un nombre único en todo el documento, el lenguaje utilizado para implementar la interactividad puede localizar el elemento fácilmente y acceder a sus propiedades, o atributos, y cambiarlas. Esto se debe a que, en el modelo DOM, los atributos de una etiqueta HTML son traducidos por el navegador en propiedades de un objeto.

Pero en un documento HTML puede haber un número muy elevado de elementos div o de elementos input, o cualquier otro elemento repetido cada uno con su identificador exclusivo. En ese caso *¿cómo identifica este modelo a cada uno de esos elementos?* La respuesta es muy simple, utiliza un vector.

*Acceso a un elemento del DOM*

|  |
| --- |
| *Supongamos el siguiente código*  *<!-- Este es un comentario de HTML para indicar que aquí iría la*  *cabecera y parte del cuerpo -->*  *<div id=”cabecera”>*  *<!-- Este comentario indica que aquí iría todo el texto HTML*  *correspondiente a la cabecera del cuerpo de la pagina -->*  *</div>*  *<div id=”contenido”>*  *<!-- Este comentario indica que aquí iría todo el texto HTML*  *correspondiente al contenido del cuerpo de la pagina -->*  *</div>*  *<!-- Este es otro comentario de HTML para indicar que aquí iría*  *el resto del cuerpo -->*  *Podemos acceder al elemento div con identificador “contenido”*  ***Forma 1***  *document.getElementById(“contenido”)*  *Esta forma utiliza el método getElementById al que se le pasa como argumento el valor “contenido” del atributo id del elemento al que queremos hacer referencia.*  ***Forma 2***  *document.getElementByTagName(“div”)[1]*  *Esta forma utiliza el método getElementsByTagName al que se le pasa como argumento el nombre de la etiqueta div a la que queremos hacer referencia.*  *Este método crea siempre un vector de elementos div. Como los vectores comienzan a numerar sus elementos en 0 el número 1 que está entre corchetes hacer referencia al segundo div del documento.*  *El índice [1] hace referencia a la posición relativa que ocupa el elemento div en el documento.*  ***Forma 3***  *document.getElementsByTagName(“div”)[“contenido”]*  *Esta forma utiliza el método getElementsByTagName para crear un vector asociativo de elementos div donde, en lugar de emplear la posición relativa del elemento div en el documento para hacer referencia a un elemento del vector, emplea el nombre del identificador “contenido” por el cual es conocido.* |

Los siguientes son enlaces a las páginas de sus autores que explican, con ejemplos, el modelo de objetos de documento.

Introducción al Modelo de Objetos de Documento (DOM). <http://kusor.net/traducciones/brainjar.es/introdom1.es.html>

¿Qué es el DOM?

<http://codexexempla.org/curso/curso_4_3.php>

Ahora que ya has visto cuáles son los elementos de un documento, sus propiedades y los métodos de que dispone cada uno, cómo hay que acceder a cada uno de ellos y conoces la sintaxis básica del Javascript, puedes modificar el comportamiento de estos elementos cambiando sus propiedades, empleando como herramienta este lenguaje.

Al igual que ocurría con las hojas de estilo, que podíamos utilizarlas en **línea** (mezcladas con el contenido), **incrustadas** (escribiendo los estilos en la cabecera del documento) y **enlazadas** (escritas en uno o varios ficheros externos que se vinculaban en la cabecera mediante un código parecido a:

**<link href="miHohaDeEstilo.css" type="text/css" rel="stylesheet" media="screen" />**

Con los scripts vuelve a pasar los mismo, ya que podríamos usar código script para capturar un evento en un elemento de un formulario escribiendo el script en el propio elemento. Puedes ver cómo, en el ejemplo:

**<input type="text" name="fecha" onchange="validateDate(this);" />**

Pero se **recomienda no hacer esto** ya que aquellos usuarios que no tengan activado el Javascript nopodrían validar la fecha. Así que, en aras de mejorar la accesibilidad, e independizar el comportamiento de la apariencia y del contenido, se recomienda tener el código separado en un archivo y enlazarlo en la cabecera del documento mediante un código parecido a:

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

La **validación** de datos de **formularios** es uno de los usos más habituales de Javascript. Se suele realizar la validación de los campos que son obligatorios para garantizar que estén cubiertos y de aquellos que tienen que ajustarse a un patrón, como puede ser una fecha o un correo electrónico para garantizar que son válidos antes de enviarlos al servidor. Así, se mejora el tiempo de respuesta del usuario que recibe antes la notificación de un dato incorrecto. Pero ten en cuenta siempre que, por el hecho de realizar la validación antes del envío, no estás libre de tener que realizarla también del lado del servidor ya que aquellos usuarios que no tengan activado Javascript te pueden enviar un formulario lleno de datos incorrectos o incompletos.

Desde el siguiente enlace accederás al artículo "Técnicas de accesibilidad: el JavaScript no obstrusivo. La teoría de la mejora progresiva" escrito por Saúl González Fernández. En él se desarrolla el tema relacionando el Javascript con las pautas de la accesibilidad. Artículo "Técnicas de accesibilidad: el JavaScript no obstrusivo. La teoría de la mejora progresiva". <http://codexexempla.org/curso/curso_4_5.php>

## 1.5.- Reproducción de sonido, vídeo y animación.

Anteriormente, en otra unidad, vimos la forma de incluir en el código XHTML el código necesario para incrustar el reproductor en la página de modo que se pudieran reproducir los archivos de sonido y/o vídeo.

En el enlace que debiste visitar en aquel momento, y que ponemos a tu disposición de nuevo, también se mostraba el código fuente necesario para incluir una animación realizada con Flash.

Uso de los elementos object y embed.

<https://developer.mozilla.org/es/Usando_el_marcado_correcto_para_llamar_a_los_plugins#El_elemento_.3cobject.3e.3a_Est.c3.a1ndares_W3C_y_los_navegadores>

*Sea cual sea el modo que empleemos para incluir sonido, vídeo y/o animaciones en una página web, los archivos deben estar en un formato apropiado para la web, teniendo el menor tamaño posible sin que afecte a la calidad necesaria para que el usuario pueda oír y/o ver su contenido correctamente.*

*Ten en cuenta, además que, si quieres hacer una página accesible y el enlace es a un archivo de vídeo o una animación sin sonido, se debe proporcionar una alternativa textual donde se realice una descripción de la acción que se está desarrollando.*

**El usuario es el que tiene la última palabra.**

Muchas veces, las personas que desarrollan interfaces están tan ensimismadas en su labor creativa que se olvidan de que el usuario es el que decide sobre qué es lo que quiere ver u oír y qué es lo que no.

Al fin y al cabo, es el usuario el que sabe lo que le interesa de un determinado sitio web.

El hecho de desactivar estas opciones mejora la experiencia visual de las personas con problemas de visión o de aquellas a las que las imágenes demasiado llamativas les producen molestias. Y la velocidad de descarga de la página mejora sustancialmente.

Aun así, si decides incorporar este tipo de recursos gráficos recuerda siempre que:

* Es mejor utilizarlos como complemento del contenido y no como elemento decorativo.
* Debes comprobar que funcionan correctamente con distintos navegadores.
* El marcado del código debe estar conforme a los estándares.

fin