Contenidos web interactivos.

Caso práctico

Como todas las semanas, se reúnen los miembros del equipo de trabajo de la empresa **BK programación** para comentar las incidencias de la semana anterior.

Esta semana tienen algo que celebrar ya que el proyecto de la panadería "Migas Amigas" está prácticamente listo y si lo entregan antes de la fecha límite, que es dentro de dos semanas, tendrán una bonificación ya que el gerente de la panadería quería que estuviera lista para la campaña navideña.

–¿Habéis verificado bien el código fuente con las herramientas de la W3C?–pregunta **Ada** a todos los presentes.

–Sí –contesta **Antonio** que continúa diciendo: –lo hemos validado para el XHTML 1.0 y las CSS3, y ya hemos añadido los iconos en el pie de página.

–Y ¿habéis verificado la accesibilidad?–vuelve a preguntar **Ada**.

–¡Pues claro! –le responde **María**, que continúa diciendo: –Hemos empleado las herramientas de verificación automática y –es **Juan** el que termina la frase– hemos revisado manualmente todos los criterios de éxito para que estuviera conforme con el nivel A. Estoy seguro de que conseguirá la subvención de su ayuntamiento.

Ada interviene preguntando: –¿Se la habéis dejado utilizar a alguna persona para ver si le resultaba fácil de usar?

–La verdad es que todos se la hemos enseñado a alguien para que nos diera su opinión. Yo en concreto se la enseñé a mi abuela que aunque todavía es joven no suele utilizar Internet muy a menudo y no tuvo ningún problema. Hasta me ha dicho que la avise cuando esté funcionando para poder escribirles con el formulario de contacto y encargar el roscón de Reyes –responde **Ana**.

–Habéis hecho un trabajo estupendo –les dice **Ada** y añade –y aún nos quedan dos semanas de plazo. ¿Creéis que se podría mejorar en estas dos semanas algo?

–Yo creo que sí –contesta **Juan**, que continúa diciendo: –Podríamos hacer la página más interactiva de modo que la experiencia del usuario sea más agradable. Aunque el formulario es accesible con el ratón y con el teclado y está preparado para las ayudas técnicas de las personas con discapacidad, podríamos controlar los eventos del usuario para que cuando éste pase por encima de algún enlace o algún elemento del formulario, o cuando un elemento se active, haya una cambio más evidente del que hay ahora. Y lo podemos hacer complementado la hoja de estilos y añadiendo algo de código con Javascript para controlar los eventos de los elementos. Sí, ya sé, no me mires de esa manera –**Juan** observa que **Ada** lo está mirando – ya se que hay que separar el contenido del aspecto y del comportamiento. No te preocupes, el script se puede vincular al documento igual que las hojas de estilo y después validaremos todo de nuevo.

–¿Qué pasa si un usuario no tiene activado el Javascript? –pregunta **Carlos**.

–Si se hace todo como dice Juan el único problema que tendrá ese usuario es que se perderá esa experiencia, porque el resto de la funcionalidad de la página permanecerá intacta ¿No es así **Juan**? –responde **Ada**, y como todas las semanas da por terminada la reunión diciendo: –Pues a trabajar todos.

1.- Elementos interactivos.

Caso práctico

Juan es un experto en desarrollo de aplicaciones informáticas y aplicaciones web. Conoce, al igual que **Ada**, muchos lenguajes de programación, tanto los que se utilizan del lado del cliente como los del lado del servidor.

Hasta ahora ha utilizado el lenguaje Javascript sin preocuparse demasiado de la accesibilidad y no tenía en cuenta si el usuario podía tener deshabilitado de funcionamiento de Javascript y podía perder parte de la funcionalidad de la página.

Ahora tiene mucha más experiencia y su labor en el equipo consiste en lograr que todos los compañeros y compañeras del equipo de **BK programación** realicen sus tareas teniendo en cuenta que deben hacer la vida más fácil y agradable al usuario.

Hemos visto en unidades anteriores algunos recursos gráficos: imágenes, sonido, vídeo y animaciones como complementos importantes del contenido de una página web y que toda persona que se dedica al diseño de interfaces web debe conocer.

Beatriz Buyo Pérez (CC BY-NC-SA)

En esta unidad trataremos el empleo de los elementos interactivos en una web. 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.

Es importante tener en cuenta que, 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.

Reflexiona

Visitas una web que habla de un tema de tu interés. Es una web agradable, con contenidos interesantes, fácil de usar y, además, está actualizándose constantemente con nuevos artículos. Te gusta, pero hay veces que te surge alguna duda sobre alguno de sus comentarios y te gustaría preguntar. ¿Está la página preparada para interactuar contigo?

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:

Beatriz Buyo Pérez (CC BY-NC-SA)

- 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:
  - La 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. Con los botones de opción, también se puede dejar uno de ellos preseleccionado por defecto o bien ninguno y, es el usuario el que decide cuál seleccionar. El aspecto de los botones de opción cuando están seleccionados es diferente de cuando no lo están.

- Las casillas de verificación son elementos que el usuario puede elegir seleccionar o no. Las casillas de verificación 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 éste 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. Aunque en la imagen no se ve ningún botón, 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, etcétera. Para ello nos servimos de las pseudoclases link, visited, hover, active, focus que ya vimos la unidad de hojas de estilo.

En la siguiente presentación 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.

Interactividad usando CSS

Usos

Interactividad usando CSS

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, etcétera.

En el siguiente enlace puedes ver un ejemplo de despliegue de información adicional cuando el usuario hace clic en una zona de la imagen:

Para integración:

Título: Ejemplo de mapa interactivo.

URL: http://green-beast.com/experiments/css_map_pop.php.html

Texto enlace: Enlace a la página de Mike Cherim donde hace un pop-up con un mapa de imagen.

Mike Cherim y pop-up hay que marcarlas en idioma inglés.

Fin integración.

Interactividad usando CSS

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 estas 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.

Interactividad usando CSS

Alternancia de imágenes

Podemos usar la pseudoclase **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.

Interactividad usando CSS

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 fotos realizada sólo con CSS. Al tratarse de una página contenedora de un ejemplo, es posible que te líes un poco con el código fuente. Puedes aislar el ejemplo creando un nuevo archivo .html con el código mínimamente imprescindible:

- Las reglas de estilo que se encuentran incrustadas en la última etiqueta **<style …>…</style>** justo antes de la etiqueta **</head>**.
- La parte del <body> que se encuentra encerrada entre ** … **.

Para integración:

Título: Ejemplo de galería de imágenes.

URL: http://www.cssplay.co.uk/menu/gallery.html

Texto enlace: Enlace al ejemplo de una galería de imágenes realizada con CSS del autor Stu Nicholls.

Stu Nicholls hay que marcarlo en idioma inglés.

Fin integración.

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".

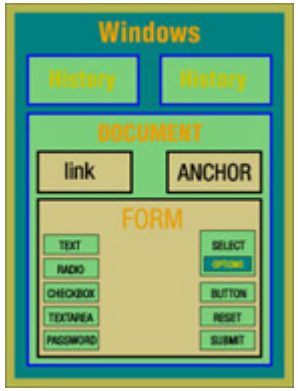
Que el **HTML** sea **estático** significa que no está generado por ningún programa ejecutado en el lado del servidor escrito en un lenguaje de programación como podría ser PHP, ASP.NET o Perl, sino que se sirve el documento HTML tal cual está escrito. Todos los usuarios reciben el mismo documento.

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). La imagen que ilustra este apartado muestra mediante cajas los objetos del modelo y la relación jerárquica que existe entre ellos. Las cajas más externas representan un nivel superior y las más internas un nivel inferior.



Autoevaluación

El título incluido dentro de una etiqueta del HTML <h1> ¿es un elemento interactivo?

- ○ Verdadero.
- ○ Falso.

No es correcto. Un título no es un elemento interactivo. Sería correcta ese título hiciera a la vez de enlace.

Muy bien. La etiqueta <h1> se suele utilizar para destacar un título mediante una letra de gran tamaño.

Solución

1. Incorrecto
2. Opción correcta

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 tienen 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.

Reflexiona

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

Mostrar retroalimentación

Había que nombrar el elemento mediante el atributo id. Por ejemplo: `<p id="primer_parrafo">`.

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.

Debes saber

En el siguiente enlace puedes ver todos los objetos (elementos) que forman parte de un documento con sus propiedades y con los métodos o acciones que se pueden realizar sobre el mismo.

Modelo de objetos de documento (HTML), Nivel 1.

Para saber más

Desde el siguiente enlace puedes acceder a las especificaciones del W3C de los tres niveles del modelo DOM.

Modelo de objetos de documento (HTML), Nivel 1, 2 y 3.

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. En en la siguiente presentación puedes ver un ejemplo comentado de las formas que tenemos de acceder a un elemento del modelo de objetos de documento.

Acceso a un elemento del DOM

Acceso a un elemento del DOM

Supongamos el siguiente código

HTML

```
<!-- 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 cu
</div>
<div id="contenido">
<!-- Este comentario indica que aquí iría todo el texto HTML correspondiente a  el contenido del cuer
```

```
</div>
<!-- Este es otro comentario de HTML para indicar que aquí iría el resto del cuerpo -->
```

Acceso a un elemento del DOM

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.

Acceso a un elemento del DOM

Podemos acceder al elemento div con identificador “contenido”

Forma 2

document.getElementsByTagName(“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.

Acceso a un elemento del DOM

Podemos acceder al elemento div con identificador "contenido"

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.

Recomendación

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)

¿Qué es el DOM?

1.4.- Ejecución de secuencias de comandos.

Debes saber

Desde el siguiente enlaces accederás a la página de Saúl González Fernández donde se dan las nociones de la sintaxis básica de Javascript.

Sintaxis básica de Javascript.

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.

Pero es muy importante que tengas en cuenta que, 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 sacado de la wikipedia:

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

Pero se recomienda no hacer esto ya que aquellos usuarios que no tengan activado el Javascript no podrían 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 incompleto.

Debes saber

Desde el siguiente enlace accederás al artículo "Los Principios del Javascript No Obstrusivo" escrito por Peter-Paul Koch de la página DEV.OPERA. En este artículo se argumentan las razones por las cuáles se debe separar el contenido del comportamiento. Son especialmente interesantes los ejemplos que aporta.

Artículo "Los Principios del Javascript No Obstrusivo".

Recomendación

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".

Autoevaluación

El uso de Javascript es la única manera de proporcionar interactividad a una página ¿Verdadero o falso?

- ○ Verdadero.
- ○ Falso.

No es correcto. También están las reglas de estilo que permiten cambiar de color los enlaces o de los elementos de un formulario cuando el ratón pasa por encima, etcétera.

Muy bien. Has prestado mucha atención a los contenidos.

Solución

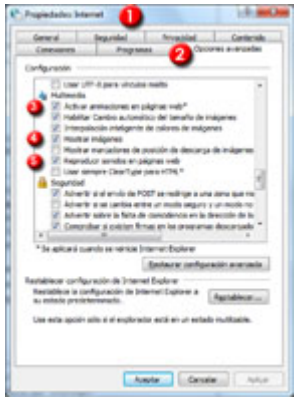
1. Incorrecto
2. Opción correcta

1.5.- Reproducción de sonido, vídeo y animación.

Anteriormente, en otra unidad, vimos la forma de incluir en nuestra página 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.**



Beatriz Buyo Pérez (CC BY-NC-SA)

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.

Citas para pensar

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.

En la imagen que ilustra este apartado se muestra la forma en la que el sistema Windows permite al usuario tomar la decisión de ver o no las imágenes, o reproducir o no los sonidos y/o las animaciones. Basta con que el usuario vaya al Panel de control, a la opción Redes e Internet y entrar en Opciones de Internet en la pestaña de Opciones avanzadas para que con un simple clic de ratón se pierda todo aquello a lo que hemos dedicado horas y horas de esfuerzo.

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.

Aún 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.