

Actividades 3. Eventos JavaScript

1. Para esta parte vamos a trabajar con el curso publicado en el enlace <http://librosweb.es/libro/javascript/>
2. Leer el capítulo 6 y realizar los ejercicios 14, 15 y 16. Comprobad vuestra solución con la del sitio web.
3. En el ejercicio 14, hay que añadir los eventos necesarios para que cuando se pase el ratón por encima de los enlaces, aparezca el color de fondo en naranja (ver punto 6.3)
4. Modifica el ejercicio 12 para que cuando se pase el ratón sobre el mensaje “Seguir leyendo”, aparezca el contenido oculto sin necesidad de pinchar sobre él. Hacer lo mismo sobre el texto “Ocultar”
5. Crea una página que realice preguntas sobre la tabla de multiplicar:
El programa buscará dos números aleatorios y los mostrará un mensaje en la pantalla.
También mostrará tres posibles resultados (el bueno y dos buscados aleatoriamente)
El usuario tendrá que pinchar sobre la opción correcta.
Después de 10 rondas se mostrará un mensaje con la nota obtenida.
Ejemplo:

EL JUEGO DE LA TABLA DE MULTIPLICAR

¿Cuánto es 5 x 4?

18

20

25

6. Con el fichero Eventos.html del aula virtual vas a modificar las propiedades de la tabla con parámetros introducidos por el usuario.
En lugar de botones, pondremos una lista con los elementos a modificar: Borde, CellSpacing, CellPadding, Color del encabezado, Color del resto de las filas, tamaño de la letra del encabezado y tamaño de la letra del resto de las filas.
El usuario, al pinchar sobre cada uno de estos elementos, se capturará un evento que llame a la función correspondiente, la cual, preguntará al usuario sobre el valor que se quiere proporcionar para ese elemento.
Por ejemplo: si tenemos un elemento llamado “Tamaño de borde”, al pinchar sobre él, una ventana preguntará el valor para ese borde y lo asignará al borde de la tabla.
Es muy importante comprobar el valor pasado para que tenga un valor correcto y que esté dentro de un rango prudente. Si pasamos un tamaño de borde 100, se deberá volver a pedir otro valor; si por el contrario pasamos como tamaño de borde una palabra, no se deberá procesar y también se volverá a pedir otro valor.
7. A partir del ejercicio anterior, crea un evento para que cuando se pase sobre una línea, aparezca una ventana indicando la información de esa línea: número de orden en la tabla y los valores de sus celdas.
Añade una opción para añadir nuevas líneas a la tabla: una opción para añadir al principio y otra opción para añadir al final. A través de una ventana se pedirá al usuario los valores de la nueva línea.
Añade una opción para eliminar líneas de la tabla: Se pedirá al usuario el número de línea a eliminar y después de mostrar la información de la línea se pedirá la confirmación del borrado.

8. Vas a desarrollar el juego de los colores.

Utilizando el fichero que tienes en el aula virtual "propagacion.html", debes conseguir:

- Si pulsas sobre DIV3 se coloree DIV2 de azul
- Si pulsas sobre DIV2 se coloree DIV1 de rojo
- Si pulsas sobre DIV1 se coloree DIV3 de verde
- Finalmente, en el momento que pulses 3 veces sobre el mismo DIV se debe detener la propagación de ese DIV en cuestión.

9. Con el mismo fichero anterior, ahora en vez del juego de colores, debes conseguir, poniendo un alert, pulsando sobre DIV3, se ejecute propagando en este orden: DIV3 > DIV1 > DIV2

10. Comprime los ejercicios en un archivo comprimido que se llame "ApellidosNombre.zip" (o .rar, .7z, etc)

Poner especial interés en los siguientes puntos:

1. Los eventos hacen posible que los usuarios transmitan información a los programas, es decir, permiten una interacción completa entre el usuario y las páginas/aplicaciones web.
2. Existen tres modelos diferentes para manejar eventos: **Modelo básico, estándar o DOM nivel 2 y modelo de Internet explorer.**
3. El nombre de cada evento se construye mediante el prefijo on, seguido del nombre en inglés de la acción asociada al evento (onclick, onfocus, etc)
4. Se deben asociar funciones o código JavaScript a cada evento. De esta forma, cuando se produce un evento se ejecuta el código indicado.
5. Existen varias formas diferentes de indicar los manejadores:
 - Manejadores como atributos de los elementos XHTML (el menos recomendado de utilizar)
 - Manejadores como funciones JavaScript externas.
 - Manejadores "semánticos" (el que se debe utilizar)
6. El evento onload es uno de los más utilizados ya que, las funciones que permiten acceder y manipular los nodos del árbol DOM solamente están disponibles cuando la página se ha cargado completamente.
7. La segunda forma consiste en extraer todas las instrucciones de JavaScript y agruparlas en una función externa. Una vez definida la función, en el atributo del elemento XHTML se incluye el nombre de la función, para indicar que es la función que se ejecuta cuando se produce el evento (ej: onclick="funcionexterna()")
8. A esta función se le puede pasar como parámetro la variable this para poder utilizarla en el código (onclick="funcionexterna(this)")
9. Al asignar una función externa a un evento utilizando exclusivamente código JavaScript, **no se deben incluir los paréntesis**. Ejemplo:

```
document.getElementById("texto_pinchable").onclick = nombreFuncion;
```

10. La manera más profesional de utilizar los eventos, y que nos permite que un mismo evento disponga de varias acciones a la vez es usando el método addEventListener: Elemento.addEventListener("Nombre del evento", funcion);

11. Existe la propagación de eventos, bubbling o capturing cuando un elemento esta dentro de otro y todos tienen manejadores de eventos, entonces la existencia de un evento en el elemento interno hará que se lancen todos los eventos, de más significativo a menos (bubbling) o viceversa (capturing).
12. Tenemos el objeto Event que nos da información precisa de cada evento.
13. Observar el siguiente código:

```
var f1 = miFuncion;  
var f2 = miFuncion();
```

 - La diferencia entre las variables es que en f1 está la referencia a la función miFuncion y en f2 tiene el resultado que devuelve miFuncion (la llama)
14. Una función anónima es una función sin nombre y que se ejecuta inmediatamente al ser creada. No se puede volver a invocar desde otra parte del código ya que no tiene nombre.
15. La propiedad **innerHTML** permite incluir código html directamente en un nodo Element. Incluso es posible crear varias etiquetas en una sola sentencia en lugar de tener que crear varios nodos Text con sus correspondientes asignaciones al nodo Element.
16. **innerHTML** no está reconocido en el estándar, pero lo reconocen todos los navegadores y es ampliamente utilizado por los desarrolladores.
17. Es muy importante tener en cuenta que no todos los navegadores manejan de la misma manera los eventos. Hay que ver en cada caso cómo se debe implementar el código para que sea lo más compatible posible.
18. Para detectar si se está en Internet Explorer utilizar el siguiente código:

```
var ie = navigator.userAgent.toLowerCase().indexOf('msie')!=-1;  
  
if(ie) {  
    // Código para Internet explorer  
} else {  
    // Código para el resto de navegadores  
}
```

3. Materiales de apoyo:

- *Apuntes JavaScript.*
- *Introducción a JavaScript (libros web)*
- <http://librosweb.es/libro/javascript/>
- <http://www.w3schools.com/js/default.asp>
- <https://developer.mozilla.org/es/docs/Web/JavaScript>