Tarea para DWEC06.

Descripción de la tarea.

Caso práctico

Antonio está haciendo muchos avances en su proyecto. Por ejemplo, ya ha realizado casi toda la validación de los formularios de las páginas web empleando JavaScript, y les ha añadido mensajes de errores de forma dinámica. Ahora se encuentra con un pequeño problema: necesita poder acceder a algunas partes de su documento y modificar contenidos, pero no sabe cómo hacerlo. Le interesaría hacer cosas, como borrar celdas en tablas, añadir o modificar atributos a elementos, modificar contenido textual de cualquier parte del documento, etc. y todo ello usando JavaScript.

Juan le informa que todo eso que solicita se puede hacer con JavaScript, pero tendrá que hacer uso del DOM de una forma más intensiva. El DOM organiza todo el documento en una especie de árbol de elementos, de tal forma que, a través de JavaScript, podrá acceder a cada uno de esos nodos y modificar su contenido, añadir nuevos nodos, eliminarlos, recorrerlos, etc. Cualquier cambio que realice en el árbol de nodos, es reflejado de forma automática por el navegador web, con lo que las modificaciones son instantáneas de cara al cliente.

Por último, **Antonio** pregunta si es posible detectar qué botones del ratón han sido pulsados, o si se está utilizando una combinación de teclas en el teclado, ya que, por ejemplo, una de las cosas que quiere hacer en sus formularios es que, cuando se esté dentro de un campo de texto y se pulse *Enter* se pase automáticamente al siguiente campo, o que cuando se pulse una combinación de teclas determinada, se realice una tarea que tenga programada en JavaScript.

Juan le responde que para hacer eso tiene que profundizar un poco más en los eventos y, en especial, en el objeto Event, que le permite acceder a nuevas propiedades que le proporcionarán esa información específica que busca. **Juan** además puntualiza que, ahora que se mete de lleno en eventos más específicos, tendrá que tener en cuenta las incompatibilidades entre navegadores, para que sus aplicaciones sean multi-cliente. Para conseguirlo le da unas indicaciones de cómo tiene que programar los eventos, y qué diferencias va a encontrar entre los distintos navegadores.

¿Qué te pedimos que hagas?

Queremos hacer una aplicación en JavaScript que simule un pequeño tablero de dibujo. Para ello tendrás que dibujar una tablero de 30 celdas x 30 celdas con cada celda de ancho 10 px y alto 10 px. Para realizar el tablero de dibujo tendrás que emplear obligatoriamente los métodos de creación de nodos del DOM. Una vez generado el tablero lo meterás dentro del *div* con *id* "zonadibujo". Además tendrás una paleta con 5 colores de dibujo (que ya está creada y se facilita con el código *.html*)

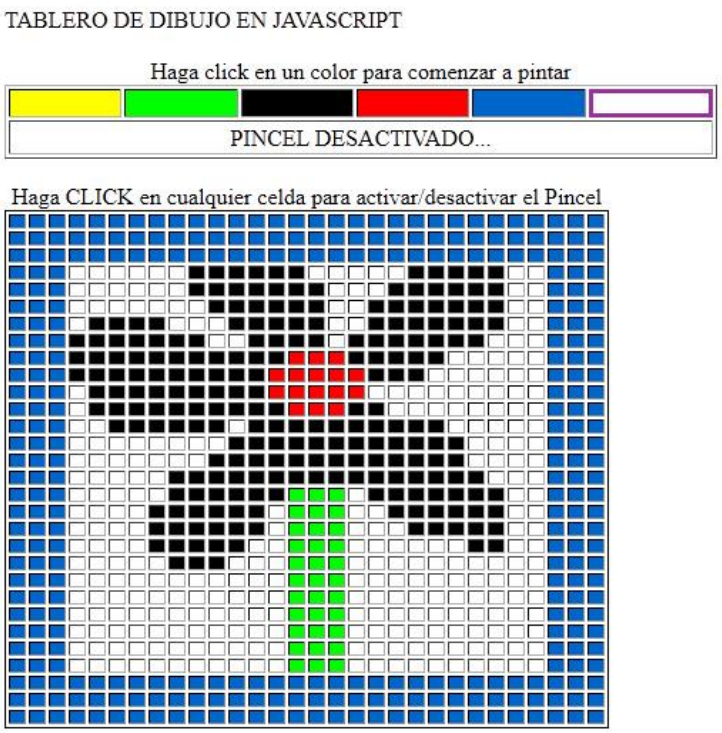
Se te facilitará un fichero *.html* y un fichero *.css* con los estilos que tendrás que utilizar. La programación de la aplicación JavaScript la tendrás que realizar en un fichero *.js* adicional.

Si se modifican los colores programados en los estilos CSS (color1 a color 6) la aplicación tendrá que seguir funcionando correctamente.

La forma de funcionamiento de la aplicación será la siguiente:

- Haremos click en alguno de los 5 colores de la paleta y se le asignará la clase "*seleccionado*".
- Una vez seleccionado un color de la paleta, haremos un click en una celda (que se pintará del color activo en la paleta) y desde ese momento al mover el ratón por el tablero pintará del color activo todas las celdas por las que vayamos pasando el ratón. En el momento que volvamos a hacer click en otra celda dejará de pintar.
- Podremos escoger un color diferente y repetir el proceso, incluso sobre celdas que ya han sido pintadas.
- Para borrar una celda pintaremos con el color blanco de la paleta.
- Cada vez que el pincel esté activado se mostrará un mensaje debajo de la paleta de colores indicando : PINCEL ACTIVADO o PINCEL DESACTIVADO.

Captura de cómo debería quedar la aplicación:



Información de interés.

Recursos necesarios

Editor web para teclear el código de la aplicación y diferentes navegadores web para ejecutar y probar su funcionamiento y compatibilidad cross-browser.

En el siguiente enlace podrás descargarte un fichero comprimido con los ficheros necesarios para la realización de la tarea (*index.html* y *estilos.css*).

Consejos y recomendaciones

Ya que vas a trabajar con clases CSS , te recomiendo que mires cómo acceder a las clases asignadas a un elemento usando los métodos del W3C.

Indicaciones de entrega

Una vez realizada la tarea elaborarás un único documento donde figuren las respuestas correspondientes. Una vez realizada la tarea, el envío se realizará a través de la plataforma. El archivo se nombrará siguiendo las siguientes pautas:

Apellido1_Apellido2_Nombre_DWEC06_Tarea

Evaluación de la tarea.

Criterios de evaluación implicados

a) Se ha reconocido el modelo de objetos del documento de una página Web.
b) Se han identificado los objetos del modelo, sus propiedades y métodos.
c) Se ha creado y verificado un código que acceda a la estructura del documento.
d) Se han creado nuevos elementos de la estructura y modificado elementos ya existentes.
e) Se han asociado acciones a los eventos del modelo.
f) Se han identificado las diferencias que presenta el modelo en diferentes navegadores.
g) Se han programado aplicaciones Web de forma que funcionen en navegadores con diferentes implementaciones del modelo.
h) Se han independizado las tres facetas (contenido, aspecto y comportamiento), en aplicaciones Web.

¿Cómo valoramos y puntuamos tu tarea?

Función de asignación de eventos.	1 punto.
Dibujar el tablero empleando métodos del DOM.	2.5 puntos.
Selección del color de dibujo en la paleta y ponerlo como color activo.	2 puntos.
Activar/desactivar el pincel.	0.5 puntos.
Pintar.	1 punto.
Que sea compatible con todos los navegadores.	2 puntos.
Comentarios e indentación del código.	1 punto.