

## SECCION 6 ( Estilos – Eventos )

### Practica 32. (32\_color\_caja.html)

Crear una web con una caja en su interior que al posicionarnos sobre ella cambie el color de fondo y texto interior, y viceversa. Se debe realizar utilizando los métodos de Javascript ['mouseover()'] y ['mouseout()']. NO se debe utilizar la pseudoclase [ :hover ] desde CSS que facilitaría parte del ejercicio.

\*\*\*\*\*Resultado\*\*\*\*\*



### Practica 33. (33\_eventos\_cajas.html)

Creamos una web con una caja y una imagen de fondo. Al hacer doble clic [ 'dblclick' ] sobre la imagen aumentando de forma proporcional tanto el ancho como el alto un 10%. Al pulsar el botón 'Volver a Inicio' la imagen recuperará su tamaño inicial. Las propiedades 'clientWidth' y 'clientHeight' del objeto Event son necesarias para realizar el ejercicio.

Los eventos que invocan funciones a través de Listeners dan la posibilidad de obtener la información del evento como parámetro de la función receptora. De esta forma podemos obtener datos sobre el objeto que se ha pulsado y actuar en consecuencia.

\*\*\*\*\*Resultado\*\*\*\*\*



### Practica 34. (34\_alternar\_cajas.html)

Disponemos de una página web con dos cajas y un botón de tipo `<INPUT>` con el texto "*Cambiar valores*". El ejercicio consiste en cambiar los valores y las propiedades de las cajas cada vez que se pulse el botón. En la medida de lo posible se utilizará el método de escucha [`'addEventListener()'`] para dejar el ejercicio con un diseño más profesional.

- El texto de las cajas estará en mayúsculas y el color de fondo es el que se muestra en la imagen.
- Cuando se haya realizado el cambio 7 veces el juego finalizará desactivando el botón a través del correspondiente atributo [`'disabled'`].

\*\*\*\*\*Resultado\*\*\*\*\*

**AMARILLO**

**Cambiar  
valores**

**VERDE**

### Practica 35. (35\_alineacion\_cajas.html)

Disponemos de una página web con 3 cajas distribuidas horizontalmente en la página.

Cada una de las cajas tiene un color de fondo ('background') diferente. Los colores iniciales de cada una de las cajas son **Amarillo – Rojo – Verde** como se muestra en la imagen inferior.

Cada vez que realicemos doble clic ('dblclick') sobre alguna de las cajas esta cambiará de color. La secuencia de los colores siempre será ( Rojo – Amarillo – Verde – Rojo – Amarillo – Verde – etc..).

En el momento que se produzca una alineación con alguno de los 3 colores se mostrará un mensaje por pantalla informando de la alineación de los colores.

- Los textos que se pueden mostrar en la caja de resultado podrán ser:
  - " << NO hay alineación de colores>>" (las tres cajas tienen colores diferentes).
  - " << El color alineado es X >>" (donde la X corresponde el nombre del color).
- El color de fondo inicial de cada una de las cajas se deberá dibujar desde los métodos proporcionados por Javascript ('element.style.backgroundColor').

\*\*\*\*\*Resultado \*\*\*\*\*

<< No hay alineación de colores >>



**Practica 36.** (36\_movimiento\_cajas.html) -> Métodos [ 'los que sean necesarios' ]

El siguiente ejercicio es una página web que permitirá mover una imagen por toda la web que se encuentra inicialmente centrada.

La web se realiza construyendo una botonera que nos permitirá mover el elemento en la posición especificada. Los botones de las flechas podrán ser imágenes (formato .svg o .png), o iconos enlazados desde algún CDN como Font Awesome o Google Icons.

Para la realización de la práctica las cajas se posicionan de forma absoluta.

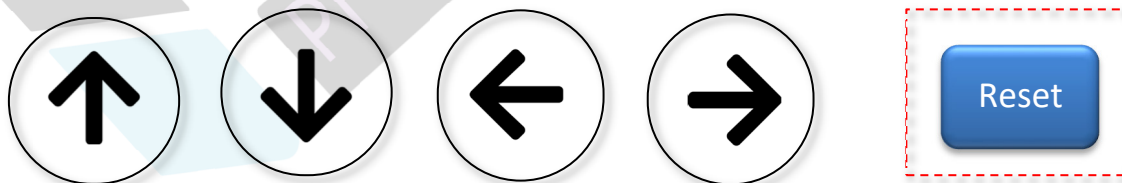
Cada vez que se pulse un botón se realizará un desplazamiento de la forma en la misma dirección de la flecha. Para ello será necesario conocer la posición del objeto con ['offsetTop'] y ['offsetLeft'].

- Las cajas que contienen las imágenes serán circulares.
- El botón 'Reset' devolverá la imagen a su posición inicial que será el centro de la página.

**OPCIONAL:** Si el alumno dispone de tiempo suficiente puede realizar las siguientes mejoras:

- La imagen no podrá sobresalir nunca de la ventana del navegador. Básicamente es controlar que los atributos ['offsetTop'] y ['offsetLeft'] no adquieran un valor en negativo
- Los botones se pintaran de color de fondo (background) verde cuando sean pulsados ['onmousedown'] y nuevamente recuperarán su fondo inicial cuando sean despusados ['onmouseup'].

\*\*\*\*\*Resultado\*\*\*\*\*



### Practica 37. (37\_estilos\_form.html)

Realizamos un formulario con 2 elementos de formulario de tipo `<input>` que al posicionarnos dentro de alguno de ellos [ `'onfocus()'` ] el elemento aumentará automáticamente el tamaño, el tipo de letra, dibuje un sombreado exterior y rellene el color de su interior en color verde. Al salir del elemento [ `'onblur()'` ] volverá a su estado inicial, es decir, todos los estilos que se han aplicado al entrar volverán a su estado de inicio. En la medida de lo posible se utilizará el método de escucha [ `'addEventListener()'` ] para dejar el ejercicio con un diseño más profesional.

\*\*\*\*\*Resultado\*\*\*\*\*

Nombre:

Mail:

a@a.com

### Practica 38. (38\_estilos\_menu.html)

La práctica consiste en crear un elemento de formulario del tipo menú desplegable con 5 nombres de ciudades y una caja a su derecha. El ejercicio consiste en seleccionar un elemento de la lista desplegable y automáticamente se modificará el fondo de la caja como se muestra en la imagen inferior.

- El cambio de imagen se producirá cada vez que se seleccione un elemento del menú desplegable lanzará el evento `'onchange()'`.
- Los bordes de la caja estarán redondeados y al posicionarnos sobre la caja [ `'mouseover'` ] cambiará su borde a otro color.

\*\*\*\*\*Resultado\*\*\*\*\*

Selecciona Ciudad:

--- Selecciona opción --- ▼



### Practica 39. (39\_tranform\_image.html)

Disponemos de una web con un rectángulo (caja) que en su interior contiene una imagen. Cada vez que realicemos un clic sobre la imagen esta rotará 90 grados [ `'transform()'` ] en el sentido de las agujas del reloj. Cada vez que el objeto de una vuelta el contador de vueltas que hay a su lado irá aumentando en función del número de vueltas que haya realizado. Cuando la imagen haya realizado 5 vueltas automáticamente no podrá continuar girando aunque se realice clic.

- La vuelta finaliza cuando la imagen vuelve a su posición inicial.
- El botón `'Reset'` volverá a dejar todo como al principio para volver a empezar.

- La caja que muestra las vueltas se pintará de color rojo cuando llegue al valor de 5.

\*\*\*\*\*Resultado\*\*\*\*\*

Contador de vueltas:

4

Reset



#### Practica 40. (40\_tranform\_titulo.html)

Esta página contiene un título introducido en una caja y cinco botones en su zona inferior.

- Los botones pueden ser cajas que al ser pulsadas ['onclick()'] invoquen un evento que ejecute el método necesario en cada situación.
- El primer botón permitirá aumentar el tamaño de la letra.
- El segundo botón permitirá disminuir el tamaño de la letra.
- El tercer botón aplicará una sombra al título a partir de las preferencias del alumno/a.
- El cuarto botón irá alternando el color entre rojo y azul.
- El quinto botón reseteará los estilos volviendo a dejar el texto como estaba inicialmente.

\*\*\*\*\*Resultado\*\*\*\*\*

Título en Javascript

Subir letra

Bajar letra

Aplicar sombra

Cambio color

Reset



#### Practica 41. (41\_transition\_image.html)

Tenemos una página con 4 imágenes distribuidas en una única línea. El efecto que se producirá al posicionarnos sobre ellas [ 'mouseover' ] aumentará el tamaño de la imagen, aplicará un borde de color rojo no excesivamente grueso y dispondrá de un sombreado alrededor de la imagen. Los bordes de las imágenes son en ángulo recto y al posicionarnos se producirá un efecto de redondeos de las mismas [ 'borderRadius' ]. Al salir del elemento [ 'mouseout()' ] el elemento volverá a su posición original.

\*\*\*\*\*Resultado\*\*\*\*\*



#### Practica 42. (42\_animation\_menu.html)

Diseñamos una página web que contenga una caja de tipo "hamburguesa" que al ser pulsada despliegue todas las opciones que disponga nuestro menú. El ejercicio se realizará mostrando las diferentes opciones de forma progresiva [ 'animation' ].

- Al pulsar nuevamente la hamburguesa el menú se cerrará.
- Las cajas que muestran las diferentes opciones del menú deberán estar inicialmente ocultas de alguna forma ya que al ser de mayor tamaño que la imagen de la hamburguesa se podrían visualizar antes de realizar un clic.
- Las propiedades de animación igual que en CSS3 se deben de configurar para que sean efectivas desde JS. Algunos de los métodos en JS que se pueden utilizar para personalizar las animaciones de las cajas serian [ 'animationDuration()', 'animationDelay()', 'animation()' ] – como metodo abreviado].

\*\*\*\*\*Resultado\*\*\*\*\*



**Practica 43.** (43\_video\_cajas.html)

Diseñamos una pequeña aplicación web que contenga en su interior un elemento de formulario del tipo menú desplegable con 4 opciones. Las opciones serán 4 videos de corta duración que se descargará el alumno/a (10 – 20 seg máximo). La aplicación permitirá seleccionar un video desde el menú desplegable que cargará de forma automática el video en la caja de al lado. Los controles del video (pause, play stop) se realizarán mediante cajas personalizadas que diseñará el alumno/a conteniendo una imagen significativa de la acción que realiza.

- Al finalizar el video mostrará un mensaje de advertencia informando que el video ha finalizado.
- Al pulsar el botón de 'Stop' se cargará en la caja de visualización el texto '*Selecciona un video*' con el fondo negro.
- Los eventos a utilizar para la realización del ejercicio son básicamente 'play', 'stop'.
- Las imágenes de los botones se deberán seleccionar de las librerías de iconos estandarizadas como Fonts Awesome, Google Icons o Bootstrap Icons.

\*\*\*\*\*Resultado\*\*\*\*\*

Selecciona película:

