

Ejercicios 5: JavaScript

En esta esta tanda de ejercicios se propone la realización de cuatro ejercicios para practicar el manejo de eventos, tratado en el tema 6 del documento entregado. En todos los ejercicios se proporciona un fichero `.html` y un fichero `.js`, dentro de la carpeta `js`, que el alumno tiene que modificar para cumplir los requisitos solicitados en cada enunciado.

Una vez finalizados todos los ejercicios, el alumno debe comprimir todos los ficheros proporcionados con este enunciado en los que se han hecho las modificaciones necesarias para conseguir los requisitos pedidos en un único archivo `.zip` o `.rar` y subirlo a la plataforma virtual del curso. El nombre del fichero tiene que ser de la forma:

- **05**_apellido1_apellido2_nombre.zip
- **05**_apellido1_apellido2_nombre.rar

Ejercicio 5-1:

El `Ejercicio_5-1.html` muestra una imagen. Modificar este fichero para que reaccione a distintos eventos de ratón generados por el usuario:

- Si el puntero del ratón se coloca encima debe mostrar el contenido del fichero `Donald_2.png`.
- Si después de estar encima, el puntero del ratón se retira, debe mostrar el contenido del fichero `Donald_1.png`.
- Si se hace clic sobre la imagen debe mostrar el contenido del fichero `Goofi_1.png`.
- Si se hace doble clic sobre la imagen debe mostrar el contenido del fichero `Goofi_2.png`.

En este ejercicio vamos a registrar eventos de ratón empleando manejadores de eventos dentro del mismo objeto, por ejemplo:

```

```

Ayuda:

- http://www.w3schools.com/js/js_html_dom_events.asp
- <http://www.javascriptya.com.ar/index.php?inicio=60>

Ejercicio 5-2:

En este ejercicio vamos a repetir el ejercicio anterior pero registrando ahora los eventos con el método `addEventListener()` dentro del propio código JavaScript. Esta forma de registrar los eventos sólo la soportan las últimas versiones de los navegadores y permite agrupar toda la gestión de los eventos de una página en una misma zona del código.

Por ejemplo, si tenemos un elemento `<video>` en la página definido por:

```
<video id="player">
```

Podemos registrar el evento `timeupdate` añadiendo el siguiente código dentro de JavaScript si cuando se produce el evento se llama a la función `desplaza()`

```
document.getElementById("player").addEventListener("timeupdate"  
,desplaza)
```

El problema que nos vamos a encontrar en un caso real es que hay que garantizar que el evento se registra cuando el elemento ya se ha cargado. Para ello se podría también a usar el evento `load` del objeto `window` (se dispara cuando la página está completamente cargada) de la siguiente forma:

```
function registraEventos(){  
    document.getElementById("player").addEventListener("timeupdate", desplaza)  
    // Registrar aquí todos los eventos de la página  
}  
  
// Cuando página cargada, llamar función registraEventos()  
window.addEventListener("load", registraEventos);
```

Ayuda:

- http://www.w3schools.com/jsref/met_element_addeventlistener.asp
- <http://www.javascriptya.com.ar/index.php?inicio=60>

Ejercicio 5-3:

En este ejercicio vamos a darle una vuelta más al cálculo de la letra del DNI. En este caso el formulario permite ir tecleando los 8 dígitos numéricos del DNI y, cuando se complete el último, automáticamente se tiene que añadir la letra, sin necesidad de pulsar sobre ningún botón, y además modificar el texto que se muestra debajo del formulario y el color del fondo del recuadro, tal y como muestran las imágenes.



Para conseguirlo tendremos que practicar con los eventos relacionados con el teclado:

- **keypress:** este evento ocurre cuando el usuario presiona una tecla
- **keydown:** este evento ocurre cuando el usuario mantiene presionada una tecla
- **keyup:** este evento ocurre cuando el usuario libera una tecla presionada

En este ejercicio se proporcionan 3 ficheros:

- `Ejercicio_5-3.html`: descripción de la página web
- `Ejercicio_5-3.css`: hoja de estilos empleados

- Ejercicio_5-3.js: código Javascript

En este ejercicio es la primera vez que utilizamos JavaScript para interactuar con la hoja de estilo del documento descrita en el fichero `.css`. Para ello utilizaremos la propiedad `style` del elemento que queremos modificar:

Ayuda:

- http://www.w3schools.com/jsref/dom_obj_event.asp
- http://www.w3schools.com/jsref/dom_obj_style.asp

En este ejercicio el usuario va tecleando datos en el elemento:

```
<input type="text" id="textoDNI" maxlength="8" placeholder="8 números"/>
```

Cada vez que teclea un dato hay que comprobar si la longitud de la cadena introducida es de 8. Si es así, hay que calcular la letra del DNI, escribirla en el formulario y cambiar el color de fondo. Para realizar esto vamos a registrar el evento `"keyup"` en este elemento y que llame a la función `comprobar()`:

```
<input type="text" id="textoDNI" maxlength="8" placeholder="8 números"
onkeyup="comprobar()" />
```

Ahora pasemos a ver qué tiene que hacer esta función `comprobar()`. Como ya se ha comentado primero tiene que ver la longitud del dato introducido y, si es 8, desencadenar las acciones que menciona el enunciado:

```
function comprobar() {
var longitud = document.getElementById("textoDNI").value.length;
    if (longitud==8){
        escribirLetra();
        cambiarTexto();
        cambiarColorFondo();
    }
}
```

Por tanto, ahora tendríamos que desarrollar las tres funciones que son llamadas cuando la longitud del dato es 8:

```
function escribirLetra() {
    var dni = document.getElementById("textoDNI").value;
    var letras="TRWAGMYFPDXBNJZSQVHLCKET"
    var posicion=dni%23;
    letra = letras.charAt(posicion);
    document.getElementById("textoDNI").value = dni+"-"+letra;
}
```

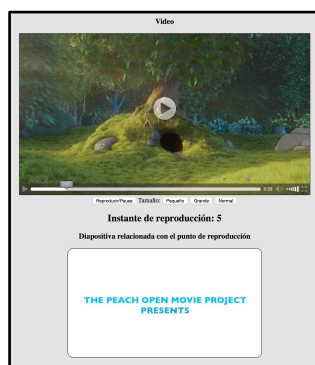
```
function cambiarTexto(){
    document.getElementById("capaTextoId").innerHTML = "DNI completo";
}

function cambiarColorFondo() {
    document.getElementById("form").style.backgroundColor = "#81F7D8";
}
```

Evidentemente todo se podría haber agrupado en una única función, pero se ha hecho así, en tres funciones, para que quede un código más claro.

Ejercicio 5-4: Sincronización de video

Al abrir con el navegador el `Ejercicio_5-4.html` se muestra en el navegador el siguiente contenido:



En la parte superior aparece un video creado con la etiqueta `<VIDEO>` de HTML5. Siguiendo las recomendaciones habituales, para intentar conseguir la mayor compatibilidad con todos los navegadores, el video se ha codificado en tres formatos diferentes: **MP4**, **Ogg** y **WebM**.

En la etiqueta `<video>` se ha incluido el atributo `controls` para que se muestren los controles de reproducción en el propio video. También se ha añadido el atributo `loop` para que, al finalizar el video, continúe por el principio.

Debajo del video se han añadido cuatro botones con los que también se va a interactuar con el video empleando JavaScript:

- Botón “Reproducir/Pausa”. Pulsando sobre este botón se inicia, o se interrumpe, la reproducción del video de forma alternativa.
- Botón “Pequeño”: disminuye el tamaño del video a un ancho de 480 pixels.
- Botón “Grande”: aumenta el tamaño del video a un ancho de 840 pixels.
- Botón “Normal”: ajusta el tamaño del video a su ancho original de 720 pixels.

En el fichero entregado estos cuatro botones no hacen nada. En este ejercicio hay que hacer las modificaciones necesarias en el fichero HTML y generar el código JavaScript oportuno, para que se comporten como se ha mencionado anteriormente.

Más abajo, en la página web, también se muestra un mensaje con el instante de reproducción del video expresado en segundos. En este ejercicio hay que hacer las modificaciones necesarias en el fichero HTML y generar el código JavaScript oportuno, para que se muestre este dato en pantalla.

Finalmente, en la parte inferior de la página Web, se muestra una imagen. En el fichero entregado se muestra siempre el fichero `img/Imagen01_000.jpg`. Pero además de este fichero, dentro de la carpeta `img` se han añadido 8 ficheros más con imágenes relacionadas con el video.: `Imagen01_000.jpg` a `Imagen08_032.jpg`.

En este ejercicio hay que hacer las modificaciones necesarias en el fichero HTML y generar el código JavaScript oportuno para que, según se vaya reproduciendo el video, estas imágenes vayan cambiando y se muestre siempre la imagen que corresponda con el instante de reproducción del video.

Para facilitar el trabajo, los últimos tres números del nombre de la imagen hacen referencia instante de tiempo de reproducción del video, expresado en segundos, en el que la imagen debería comenzar a mostrarse. Por ejemplo, el fichero `Imagen04_017.jpg` se tiene que mostrar en el segundo 17 de reproducción y permanecer hasta el segundo 23, en el que debería mostrarse el fichero `Imagen05_023.jpg`. Es decir, `Imagen04_017.jpg` es visible en la parte inferior si el instante de reproducción es igual o mayor que 17, y menor que 23.

En resumen, este ejercicio consta de tres partes:

1. Configurar los botones de navegación.
2. Mostrar el mensaje con el segundo de reproducción.
3. Sincronizar las imágenes con el video.

Ayuda:

- Etiqueta video de HTML5:
http://www.w3schools.com/html/html5_video.asp
- Lista de métodos, propiedades y eventos de los elementos Audio y Video en HTML5. Útil para saber qué evento hay que registrar y ver qué propiedades del video hay que consultar, para sincronizar el video con el resto de la página.
http://www.w3schools.com/tags/ref_av_dom.asp
- Para gestionar las imágenes que tienen que aparecer en la parte inferior de la página, quizás convendría crear dos *arrays*, uno con el nombre de los ficheros de las imágenes y otro *array* con el segundo en el que hay que hacer el cambio. Si es necesario, esta idea se puede debatir en el foro del curso y el Equipo Docente sugerir cómo sacarla adelante.

Comentario final:

Cualquier dificultad, cuestión, sugerencia, etc. sobre el desarrollo de estos ejercicios, o sobre el contenido del curso, puede y debe plantearla en el foro del curso.