

UNIDAD 6 Sube de nivel

setInterval(...)

Esta función nos permite hacer que una tarea se ejecute de manera repetitiva con un intervalo de tiempo concreto. Acepta dos argumentos: la función concreta a ejecutar, y el intervalo de tiempo en milisegundos a partir del cual se irá ejecutando esta función. Podemos cargar este **setInterval()** dentro de una variable para poder acceder a él más adelante y cancelarlo si es necesario.

Por ejemplo:

```
var cronometro = setInterval(actualizaContador, 1000);
```

clearInterval(...)

Esta función nos permite cancelar un intervalo iniciado por **setInterval()**. Para ello, debemos pasar el nombre de la variable que almacena el intervalo como argumento.

Ejemplo:

```
clearInterval(cronometro);
```

elemento.play() y elemento.pause()

Podemos controlar la reproducción en la etiqueta **<video>** o **<audio>** mediante los métodos **play()** para reproducir o **pause()** para pausar.

elemento.cloneNode()

Este método nos permite duplicar un nodo HTML. Este nodo existe inicialmente solo en la memoria de JavaScript, salvo que lo introduzcamos después en algún punto del DOM.

parseInt(...)

Esta función convierte un número dentro de un string en un valor numeral. Es una manera de trabajar con valores que recuperamos de atributos del DOM, ya que por defecto, estos valores son literales si van entrecomillados.

evento.stopPropagation()

Este método cancela la propagación de un evento, es decir, que un evento como **"click"** en un elemento no afecte a elementos superiores que contengan a este primer elemento y que tengan event listeners asociados.

evento.target

Esta propiedad nos indica el objetivo de un evento en cuestión. Por ejemplo, si el evento era **"click"**, nos indica sobre qué elemento concreto que ha hecho click (un enlace, un botón, etc).

evento.target.closest(...)

El método **closest()** de la propiedad **target** nos permite comprobar si el **target** del evento está contenido dentro de un **div** u otro nodo HTML que coincida con el argumento que pasamos.

Ejemplo:

```
evento.target.closest(".selecciona-nivel")
```

Este ejemplo devolvería **true** si el elemento en el que hemos hecho click está contenido en otro elemento con la clase "selecciona-nivel".

evento.key

En el caso de que el evento sea de apretar una tecla (**"keydown"**), la propiedad **key** nos indica el valor de la tecla apretada.