

Fundamentos de Programación

JavaScript
Interruptores
IES La Nía



Interrupciones

- *JavaScript* permite ejecutar fragmentos de código tras un periodo de tiempo, o repitiéndolo de manera indeterminada.
- Estas funciones no bloquean la ejecución del resto del código

setTimeout

- Ejecuta una acción tras una duración determinada
- **setTimeout**(función, tiempoMS)

```
var miFuncion = function() {
    console.log("Batman vuelve");
};
setTimeout(miFuncion, 2000);
```

Ejercicios



- (ej709timeout) A partir de un número (una caja de un formulario), ejecutar una función que muestre una alerta con dicho número tras número segundos.
- (ej710timeout) A partir de una pregunta y dos botones con respuesta SI/NO, ejecuta una función tras dos segundos que elimine el botón de NO.

setInterval

- Realiza una llamada a la función de manera repetida, con un intervalo determinado
- **setInterval**(función, intervaloMs)

```
miFuncion = function() {
    console.log("Batman vuelve");
};
setInterval(miFuncion, 2000);
```

Ejercicios

- (`ej711interval`) A partir de un número (una caja de un formulario), ejecutar una función que muestre por pantalla de manera decreciente un temporizador que cambie cada segundo hasta llegar a 0.
- (`ej712intervalform`) A partir del formulario de persona, ejecuta una función para que cada segundo guarde una copia en el almacén local de manera automática.

clearInterval

- Los timers de pueden detener/cancelar
- `clearInterval(timer)`
- Al crear un *timer*, necesitamos asignarlo a una variable

```
var i = 1;
miFuncion = function() {
    console.log("Batman vuelve " + i);
    i++;
    if (i === 20) {
        clearInterval(miTimer);
    }
};

var miTimer = setInterval(miFuncion, 2000);
```