APRENDERAPROGRAMAR.COM

EJEMPLO RELOJ JAVASCRIPT. SETTIMEOUT, CLEARTIMEOUT, SETINTERVAL, REQUEST ANIMATIONFRAME. (CU01164E)

Sección: Cursos
Categoría: Tutorial básico del programador web: JavaScript desde cero

Fecha revisión: 2029

Resumen: Entrega nº64 del Tutorial básico "JavaScript desde cero".

Autor: César Krall

TIMERS JAVASCRIPT

Además de Date en JavaScript disponemos de otros objetos y métodos que nos permiten realizar tareas relacionadas con el tiempo. En concreto el objeto global window dispone de varios métodos pensados para ejecutar una función con un cierto retardo, o periódicamente con cierto intervalo de tiempo entre cada ejecución.

TIMERS: SETTIMEOUT, SETINTERVAL, REQUESTANIMATIONFRAME

Normalmente el código JavaScript se ejecuta secuencialmente, pero existen funciones especiales denominadas timers que permiten establecer la ejecución de funciones en determinados momentos del tiempo.

Los timers son funciones predefinidas del objeto window (por tanto se pueden invocar usando window.nombreDeLaFuncion(...) o simplemente usando nombreDeLaFuncion(...).

SINTAXIS TIMER	UTILIDAD	EJEMPLOS aprenderaprogramar.com
var referenciaTimer = setTimeout(nombreFuncion, tiempoMiliseg); *** Para función sin parámetros**	La función nombreFuncion se ejecutará con un retraso de tiempoMiliseg respecto a lo que sería su ejecución natural.	var ejecT = setTimeout(mostrarAlerta, 5000); //mostrarAlerta se ejecuta con 5 segundos de retraso
var referenciaTimer = setTimeout(function() {nombreFuncion (par1, par2, ..., parN)},tiempoMiliseg); *** Para función con parámetros**	La función nombreFuncion se ejecutará con un retraso de tiempoMiliseg respecto a lo que sería su ejecución natural.	var control = setTimeout(function() {ejemploAccion(nodosTituloCurso, nodosCambiados, contador, tocaCambiar)}, 1500); //La función se ejecuta con 1.5 s de retraso
clearTimeOut(referenciaTimer)	Detiene la ejecución programada por referenciaTimer mediante setTimeOut (si se ejecuta antes del tiempo programado)	clearTimeOut(ejecT)
var referenciaTimer = setInterval(nombreFuncion(par1, par2, ... parN), tiempoMiliseg); *** Para función sin parámetros**	Ejecuta periódicamente la función nombreFuncion con un intervalo entre ejecuciones de tiempoMiliseg.	setInterval(mostrarAlerta, 5000); //mostrarAlerta se ejecuta periódicamente cada 5 segundos

SINTAXIS TIMER	UTILIDAD	EJEMPLOS aprenderaprogramar.com
var referenciaTimer = setInterval(function() {nombreFuncion(par1, par2, ... parN)}, tiempoMiliseg); *** Para función con parámetros**	Ejecuta periódicamente la función nombreFuncion con un intervalo entre ejecuciones de tiempoMiliseg.	var t = setInterval(function(){reloj('Chile') },2000); //reloj se ejecuta periódicamente cada 2 s
clearInterval(referenciaTimer)	Detiene la ejecución programada por referenciaTimer mediante setInterval	clearInterval(ejecT)
var referenciaTimer = requestAnimationFrame(nombreFuncion) *** Para función sin parámetros**	Crea un bucle de repintado delegando el control del mismo en el navegador.	globalID = requestAnimationFrame(animacionRepetimos);
var referenciaTimer = requestAnimationFrame(function(){nombreFuncion(par1, par2, ..., parN)}); *** Para función con parámetros**	Crea un bucle de repintado delegando el control del mismo en el navegador.	globalID = requestAnimationFrame(function(){animacionRepetimos('estiloDivertido')});
cancelAnimationFrame(referenciaTimer)	Detiene la ejecución programada por referenciaTimer mediante RequestAnimationFrame.	Ver ejemplo más abajo

Escribe este código y comprueba sus resultados:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html><head><title>Ejemplo aprenderaprogramar.com</title><meta charset="utf-8">
<script type="text/javascript">
function alertaTras5seg() { setTimeout(mostrarAlerta, 5000); }
function mostrarAlerta() {alert('Han pasado 5 segundos desde la carga de la página');}
</script>
</script></head>
<body onload="alertaTras5seg()" >
<div id="cabecera"><h2>Cursos aprenderaprogramar.com</h2><h3>Ejemplos JavaScript</h3></div>
</body></html>
```

El resultado esperado es que cuando transcurran 5 segundos desde la carga de la página se muestre el mensaje “Han pasado 5 segundos desde la carga de la página”.

Si queremos introducir la repetición periódica de la ejecución de la función podemos escribir una llamada con setTimeOut dentro de la propia función. Por ejemplo:

```
function mostrarAlerta() {
alert('Han pasado 5 segundos');
setTimeout(mostrarAlerta, 5000);
}
```

Hará que se repita periódicamente la alerta indicando que han pasado 5 segundos.

Aunque si se quiere una repetición periódica será más cómodo usar setInterval:

```
function alertaTras5seg() {setInterval(mostrarAlerta, 5000);}
```

REQUESTANIMATIONFRAME

Tradicionalmente los efectos de progresión o animación incremental con JavaScript se lograban con un código de este tipo:

```
setInterval(function() {  // Código a ejecutar}, 1000/60);
```

Con este código se lograba ejecutar 60 repintados por segundo de una función que por ejemplo iba dibujando una línea. Esta cadencia daba lugar a que pareciera que la línea se dibujaba progresivamente.

A partir de cierto momento, se introdujo una forma de crear estos efectos (puede que no funcione en algunas versiones de navegadores) que trataba de trasladar el control de la cadencia de la animación al propio navegador con varios objetivos:

a) Permitir que el navegador detuviera el proceso en una pestaña si esta pasaba a estar inactiva (dejando así de consumir recursos).

b) Permitir que el navegador optimizara el redibujado, optimizando los recursos y evitando bloqueos.

c) Buscar un menor consumo de energía (CPU).

Algunos programadores adoran requestAnimationFrame y otros lo ignoran o simplemente no lo usan.

El esquema básico para trabajar con requestAnimationFrame es el siguiente:

```
var globalID;
function animacionRepetimos() {
  //Código a ejecutar
  globalID = requestAnimationFrame(animacionRepetimos);
}
function detener() {  cancelAnimationFrame(globalID);}
```

La animación comienza cuando se invoca a una función a la que hemos denominado (a modo de ejemplo) animacionRepetimos. Dentro de esta función existe una llamada recursiva que da lugar a que

el navegador ejecute el repitando periódico ejecutando previamente la función. La función puede ser detenida usando cancelAnimationFrame(nombreDelIdentificadorEmpleado);

Ejecuta este código y comprueba sus resultados (ten en cuenta que en algunos navegadores puede no funcionar, especialmente si son más antiguos):

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html><head><title>Ejemplo aprenderaprogramar.com</title><meta charset="utf-8">
<style type="text/css">
button {margin:10px;}
#cabecera{position:absolute; color:white; margin:5px; font-family:verdana, arial;}
.relleno { width: 5px; height: 20px; background: black; float: left;}
</style>
<script type="text/javascript">
window.requestAnimationFrame = window.requestAnimationFrame || window.mozRequestAnimationFrame ||
                    window.webkitRequestAnimationFrame || window.oRequestAnimationFrame;
var globalID;
function animacionRepetimos() {
  var nuevoDiv = document.createElement("div");
  document.getElementById('oculto').appendChild(nuevoDiv);
  nuevoDiv.innerHTML='<div class="relleno"  \> </div>';
  globalID = requestAnimationFrame(animacionRepetimos);
}
function empezar() {globalID = requestAnimationFrame(animacionRepetimos);}
function detener() {  cancelAnimationFrame(globalID);}
</script></head>
<body >
<button id="start" onclick="empezar()">Empezar</button><button id="stop" onclick = "detener()">Detener</button>
<div id="oculto"></div>
<div id="cabecera"><h2>Cursos aprenderaprogramar.com</h2><h3>Ejemplos JavaScript</h3></div>
</body></html>
```

El resultado esperado es que una el color negro se vaya extendiendo de izquierda a derecha rellenando líneas (al acumularse div de fondo negro) sobre la pantalla haciendo visible el texto blanco que no se desplaza por tener la propiedad CSS position: absolute;

CREAR UN RELOJ CON JAVASCRIPT

Con las herramientas de que disponemos estamos en disposición de crear un reloj con JavaScript.

Ejecuta este código y comprueba sus resultados (ten en cuenta que en algunos navegadores puede no funcionar, especialmente si son más antiguos):

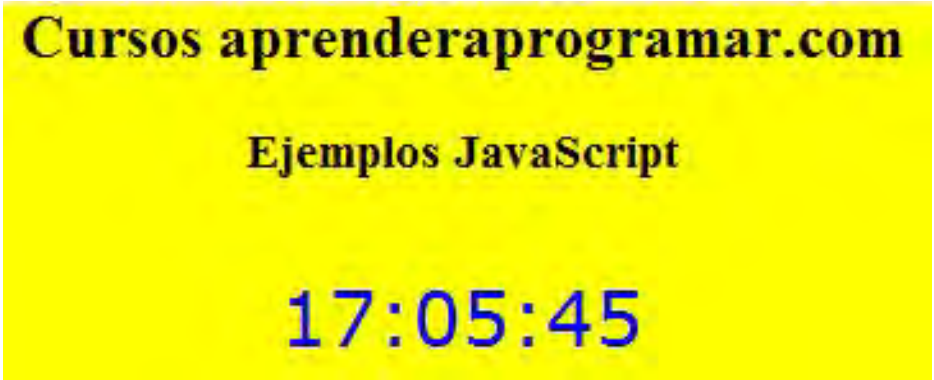
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Ejemplo aprenderaprogramar.com</title>
<meta charset="utf-8">
<script type="text/javascript">

function reloj() {
   var hoy=new Date(); var h=hoy.getHours(); var m=hoy.getMinutes(); var s=hoy.getSeconds();
   m = actualizarHora(m);    s = actualizarHora(s);
   document.getElementById('displayReloj').innerHTML = h+":"+m+":"+s;
   var t = setTimeout(function(){reloj()},500);
}

function actualizarHora(i) {
   if (i<10) {i = "0" + i};  // Añadir el cero en números menores de 10
   return i;
}
</script>

</script>
</head>
<body onload="reloj()" >
<div style="text-align:center;">
<div id="cabecera"><h2>Cursos aprenderaprogramar.com</h2><h3>Ejemplos JavaScript</h3></div>
<div style="color:blue; font-family: verdana, arial; font-size:30px; padding:15px;" id ="displayReloj" >   </div>
</div>
</body>
</html>
```

El resultado esperado es que se muestre un reloj marcando los segundos.



EJERCICIO 1

Crea un reloj JavaScript que marque los segundos usando el método setInterval.

Para comprobar si tus respuestas y código son correctos puedes consultar en los foros aprenderaprogramar.com.

EJERCICIO 2

Crea un reloj JavaScript que marque los segundos usando el método requestAnimationFrame.

Para comprobar si tus respuestas y código son correctos puedes consultar en los foros aprenderaprogramar.com.

EJERCICIO 3

Crea un reloj JavaScript que marque inicialmente 01:00 (es decir, 1 minuto) y que marche hacia atrás (00:59, 00:58, 00:57 ... etc.) hasta llegar a 00:00. Cuando llegue a 00:00 debe deternerse y mostrar el mensaje: "Tu tiempo ha terminado".

Para comprobar si tus respuestas y código son correctos puedes consultar en los foros aprenderaprogramar.com.

Próxima entrega: CU01165E

Acceso al curso completo en aprenderaprogramar.com -- > Cursos, o en la dirección siguiente: http://aprenderaprogramar.com/index.php?option=com_content&view=category&id=78&Itemid=206