

NOMBRE ALUMNO/A Iván Hernández Fuentes

Curso 2018/2019	ACTIVIDADES	CALIFICACIÓN
Fecha:	Grupo: 2º DEW	

MÓDULO	UNIDAD DE TRABAJO	
DEW	UT3	A8

La tarea consiste en la implementación de varios scripts en lenguaje Javascript. Es obligatorio el uso de las funciones que se detallan

La tarea consiste en la realización de varios scripts en Javascript:

1. Realiza un script de comprobación de palíndromos, que determine si la cadena de texto introducida por el usuario se lee de la misma forma desde la derecha y desde la izquierda (exceptuando los espacios)

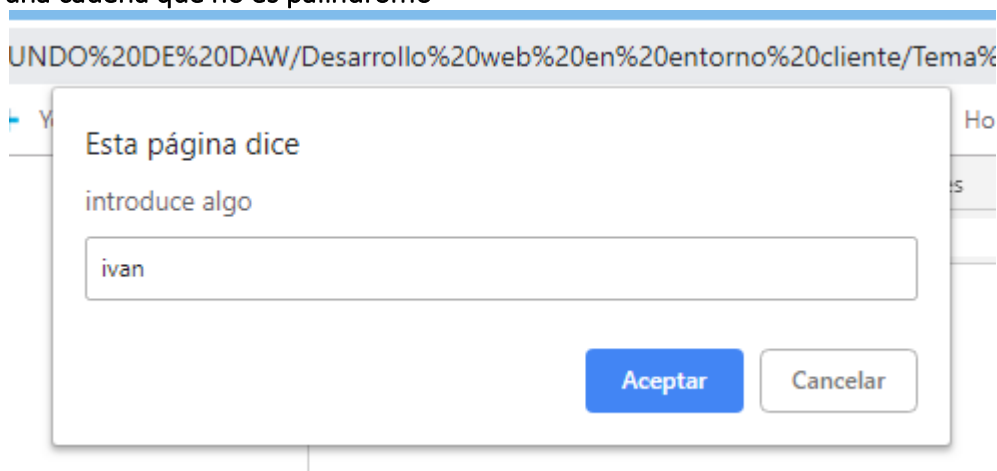
El script pedirá frases sucesivamente hasta que se introduzca la cadena "FIN" (en mayúsculas o minúsculas).

Es obligatorio el uso de al menos las siguientes funciones de **String**:

- **charAt()** para buscar el carácter en una posición dada.
- **toLowerCase()** / **toUpperCase()** para no discriminar entre mayúsculas y minúsculas.

Función	Recibe	Devuelve	Descripción
esPalindromo	Cadena	boolean	Determina si la cadena introducida es palíndromo (independientemente de mayúsculas/minúsculas)

Introducimos una cadena que no es palindromo



A continuación introducimos otra cadena que si es Palindromo

Esta página dice

introduce algo

AceptarCancelar

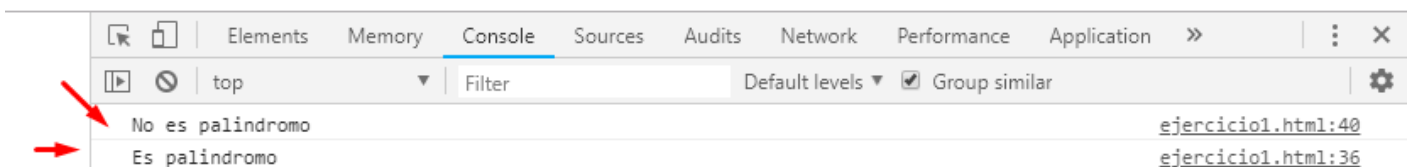
Introducimos fin para terminar la ejecución

Esta página dice

introduce algo

AceptarCancelar

Nos devuelve que la primera no es palíndromo y la segunda si



Código

```
<script>

function esPalindromo(cadena){

    var l = cadena.length;
    var i = 0;
    while(i != l - i - 1){

        var a = cadena.charAt(i);
        var b = cadena.charAt(l - i - 1);

        if(a != b){
            return false;
        }
        i++;
    }

    return true;
}
```

```

do{
  var cadena = prompt("introduce algo").toLowerCase().replace(/\s/g,"");
  if(esPalindromo(cadena)){
    console.log("Es palindromo");
  }
  if(!esPalindromo(cadena)) {
    console.log("No es palindromo");
  }
}while(cadena != "fin");
</script>

```

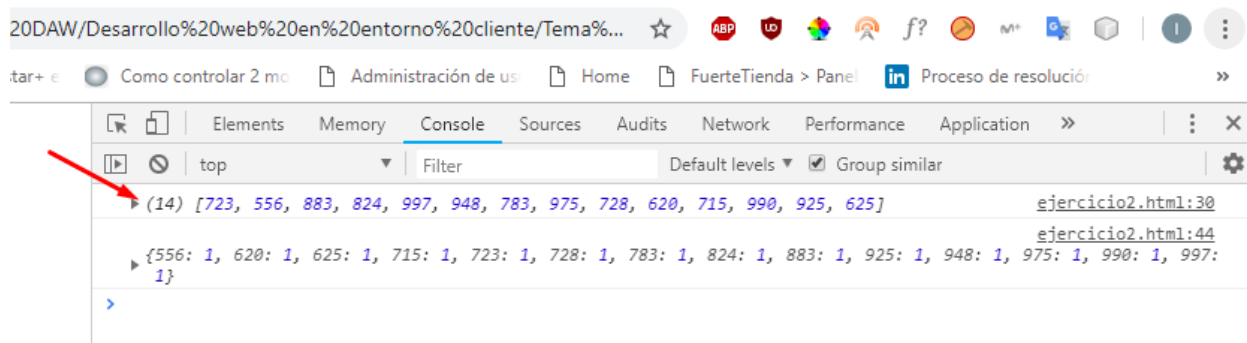
- Realiza un script que genere de manera aleatoria una cantidad de números cuyo valor varía entre 1 y 50 (la cantidad a generar también será un número aleatorio entre 500 y 1000). Dichos scripts deberán ser guardados en un array.

Una vez generados los números, se mostrará un mensaje indicativo de cuál ha sido el número más frecuente. Para ello se contabilizarán las repeticiones de todos los números usando un array. El array tendrá 50 posiciones, y en la posición *i* se indicará el número de repeticiones del valor *i*.

Función	Recibe	Devuelve	Descripción
generaAleatorio(min,max)	min:entero max:entero	entero	Genera un aleatorio entre máximo y mínimo

			(incluidos).
generaNAleatorios(n,min,max)	n:entero min:entero max:entero	entero[]	Genera un array con n aleatorios entre máximo y mínimo.
cuentaFrecuencias(numeros)	numeros: entero[]	entero[]	Genera un array que indica, las frecuencias de repetición de cada valor contenido en numeros. La posición i-ésima contendrá el número de repeticiones del valor i.
maximaFrecuencia(repeticiones)	repeticiones:entero[]	-	Imprime las frecuencias de repetición de todos los elementos del array, indicando además cual es el elemento que más se repite.

Generamos el array según el numero que haya tocado, en este caso el 14 eso quiere decir que hay 12 elementos en el array.

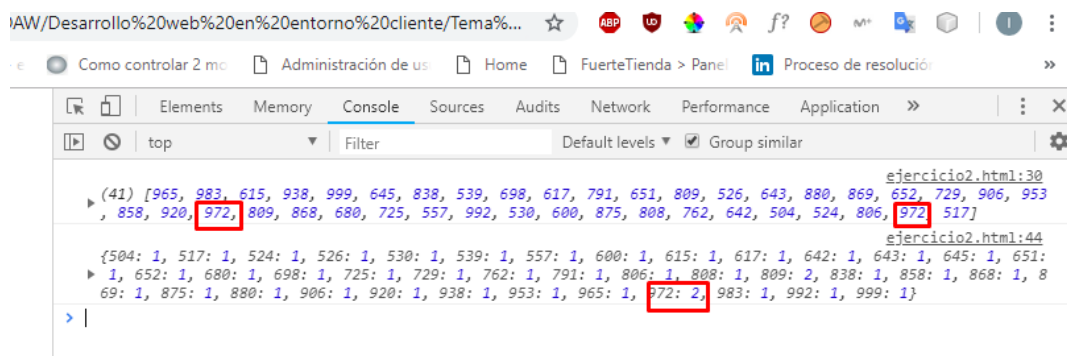


```

(14) [723, 556, 883, 824, 997, 948, 783, 975, 728, 620, 715, 990, 925, 625]
{556: 1, 620: 1, 625: 1, 715: 1, 723: 1, 728: 1, 783: 1, 824: 1, 883: 1, 925: 1, 948: 1, 975: 1, 990: 1, 997: 1}
  
```

Abajo nos muestra las veces que sale el número en este caso solo una vez.

Generamos otra vez y en este caso si detecta que ha salido más de una vez un número.



```

(41) [965, 983, 615, 938, 999, 645, 838, 539, 698, 617, 791, 651, 809, 526, 643, 880, 869, 652, 729, 906, 953, 858, 920, 972, 809, 868, 680, 725, 557, 992, 530, 600, 875, 808, 762, 642, 504, 524, 806, 972, 517]
{504: 1, 517: 1, 524: 1, 526: 1, 530: 1, 539: 1, 557: 1, 600: 1, 615: 1, 617: 1, 642: 1, 643: 1, 645: 1, 651: 1, 652: 1, 680: 1, 698: 1, 725: 1, 729: 1, 762: 1, 791: 1, 806: 1, 808: 1, 809: 2, 838: 1, 858: 1, 868: 1, 869: 1, 875: 1, 880: 1, 906: 1, 920: 1, 938: 1, 953: 1, 965: 1, 972: 2, 983: 1, 992: 1, 999: 1}
  
```

Código

```
<script>

var min = 1;
var max = 50;

function generaAleatorio(min,max){ //generamos un aleatorio entre los parámetros que le entran
    var numero = Math.floor((Math.random() * max)+min);
    return numero;
}

function generaNaletarios(n,min,max){ //generamos un array de N numeros que haya salido en la anterior funcion y haciendo un random segun
var numeros = [];

    for(var i = 0; i < n; i++){
        var nuevoNumero = Math.floor((Math.random() * (max - min))+min); //genera el nuevo numero
        numeros.push(nuevoNumero); //añadimos el numero al array
    }
    console.log(numeros);
    return numeros;
}

function cuentaFrecuencias(arr) {
    array = arr.reduce((prev, curr) => (prev[curr] = ++prev[curr] || 1, prev), {}); //obtenemos las veces que se repite ese numero en el arra
    return array;
}

function maximaFrecuencia(number){
    console.log(number);
}

maximaFrecuencia(cuentaFrecuencias(generaNaletarios(generaAleatorio(min,max),500,1000)));

</script>
```

- Realiza un script de que cambie automáticamente la URL de la página una vez transcurridos 10 segundos. La URL se tomará aleatoriamente de entre un array con 10 URLs, definido estáticamente al comienzo del script. Utiliza `setTimeout()` para establecer el tiempo de espera.

Función	Recibe	Devuelve	Descripción
cambiaURL(url)	url:cadena	-	Cambia la URL del navegador

Código

```
<script>

var url = [
  "https://www.google.es",
  "https://moodle.cifpmajadamarcial.com",
  "http://51.254.116.159:8080/",
  "http://51.254.116.159/cesar/video/video.html",
  "http://51.254.116.159/dsw",
  "http://marca.com",
  "https://github.com/",
  "https://elpais.com/",
  "http://web.mit.edu/6.813/www/sp17/programming-labs/lab1-html-css/",
  "https://yahoo.com"
];

function cambiaURL(){
  random = Math.floor(Math.random()*url.length);
  window.location.href=url[random];
}
setTimeout(cambiaURL, 10000);
```

- Realiza un script que muestre la fecha actual. Utiliza una hoja de estilos que vincularás desde la página HTML para mejorar la apariencia del calendario:



Función	Recibe	Devuelve	Descripción
nombreMes(indice)	indice:entero	cadena	Devuelve el nombre del mes correspondiente al número de mes: 1 es enero, 2 es febrero... 12 es diciembre
muestraCalendario()	--	-	Muestra el calendario correspondiente al día Actual

← → ↻ ⓘ file:///C:/Users/ivan/

Aplicaciones Aulas Virtuales CIFP

2018
23
Octubre

También hay un switch pero no se muestra debido a que es muy largo

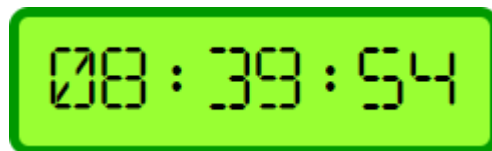
```
function muestraCalendario(){
    var dia = fecha.getUTCDate();
    var anio = fecha.getFullYear();

    document.write(`<div id ="calendario">
        <span id="anio"> ${anio}</span><br/>
        <span id="dia"> ${dia}</span>
        <span id="mes"> ${nombreMes(mes)}</span>
    </div>`);
}

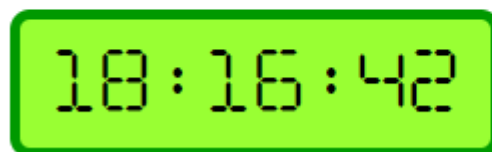
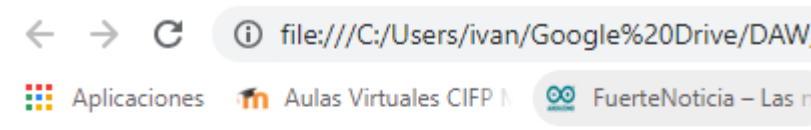
muestraCalendario();
```

5. Realiza un script que muestre la hora actual, que se actualizará de manera dinámica usando la función `setInterval()`. El mensaje se mostrará con el formato HH:MM:SS

Función	Recibe	Devuelve	Descripción
formateaHora (fecha)	fecha:Date	cadena	Devuelve una cadena que representa la hora actual en formato HH:MM:SS
actualizaReloj()	--	-	Muestra un mensaje en consola con la hora actual



Resultado del reloj



FuerteNoticia - L
apw1617iher.esy

```

<script>
  function formateaHora(){
    tiempo = new Date();
    h = tiempo.getHours();
    m = tiempo.getMinutes();
    s = tiempo.getSeconds();

    if(h < 10){
      h = '0' + h;
    }else if(m < 10 ){
      m = '0' + m;
    }else if(s < 10) {
      s = '0' + s;
    }

    var reloj;

    reloj = document.getElementById("reloj").innerHTML = `${h}:${m}:${s}`;

    return reloj;
  }

  function actualizaReloj(){
    setInterval(formateaHora,1000);
  }

  actualizaReloj();
</script>
<div id="reloj">

```