Ejerccios JS

1: Bucle que genera la secuencia de colores seguros en Hexadecimal

2: Crear una página WEB con un DIV y un botón, en el botón deberá aparecer un color y cuando se pulsa el color que indica el botón deber usarse como fondo del DIV y el botón visualizará otro color. La secuencia de colores tiene que ser totalmente aleatoria.ágina WEB con un DIV y un botón, en el botón deberá aparecer un color y cuando se pulsa el color que indica el botón deber usarse como fondo del DIV y el botón visualizará otro color. La secuencia de colores tiene que ser totalmente aleatoria.

3: Hazme una pagina web que tu introduzcas una expresión regular y te la pruebe

4: Hazme una pagina web donde me permitas seleccionar todos los eventos y me puestres l oque hacen con un div .

5: Con Canvas dame una pelota que se mueva aleatoriamente por un cuadrado.

Ejercicos Repaso Temas 2-3:

Fundamentos de JavaScript

Tipos de Datos: Escribe un programa que determine si una entrada dada es un número, una cadena, un objeto, o un valor nulo.

Conversiones de Tipos: Implementa una función que convierta números a cadenas y cadenas a números. Valida entradas no convertibles.

Operadores Aritméticos: Crea una calculadora básica que permita realizar operaciones (+, -, \*, /, %) con dos números ingresados por el usuario.

Operadores Booleanos: Diseña un programa que simule una tabla de verdad para los operadores AND, OR y NOT.

Condiciones y Bucles: Escribe un programa que calcule todos los números primos entre 1 y 100 usando un bucle.

Estructuras de Control y Bucles

Validación de Edad: Crea una función que valide si una persona es mayor de edad según la edad ingresada. Usa if...else.

Serie de Fibonacci: Genera los primeros 20 números de la serie de Fibonacci utilizando un bucle while.

Sumatoria Condicional: Diseña un programa que calcule la suma de todos los números impares entre 1 y 100.

Juego de Adivinanza: Crea un programa que permita al usuario adivinar un número aleatorio entre 1 y 10. Usa un bucle para permitir intentos.

Contador de Vocales: Escribe una función que cuente cuántas vocales hay en una cadena de texto ingresada.

Objetos Predefinidos

Manipulación de Strings: Implementa una función que tome una frase y devuelva cada palabra en orden inverso.

Expresiones Regulares: Valida un número de teléfono que siga el formato (123) 456-7890 usando una expresión regular.

Objeto Math: Calcula el área y circunferencia de un círculo dado su radio. Usa Math.PI.

Objeto Date: Escribe un programa que muestre cuántos días faltan para el año nuevo.

Navegador del Usuario: Crea un script que detecte y muestre el navegador y sistema operativo del usuario usando el objeto navigator.

Manipulación del DOM

Interacción con el Usuario: Diseña una página que tenga un botón que, al hacer clic, cambie el color de fondo del documento a un color aleatorio.

Contador Dinámico: Implementa un contador que aumente y disminuya su valor con dos botones. Usa elementos <button> y onclick.

Formulario Dinámico: Valida en tiempo real un formulario con campos de "Nombre", "Correo" y "Contraseña". Muestra errores debajo de cada campo.

Manipulación de Tablas: Crea una tabla con filas dinámicas. Agrega un botón que permita añadir nuevas filas con contenido predefinido.

Uso de Canvas: Dibuja un semáforo en un elemento <canvas> con tres círculos que cambien de color al hacer clic en un botón.

Ejercicios mas dificles :

Conversión y Redondeo Avanzado: Diseña un programa que tome un número decimal y lo convierta a su representación en binario, octal y hexadecimal, mostrando cada resultado con al menos 8 dígitos de longitud, completando con ceros si es necesario.

Cálculos con Notación Científica: Escribe un programa que acepte dos números en notación científica (por ejemplo, 3.2e5) y realice las cuatro operaciones básicas entre ellos.

Jerarquía de Operaciones: Implementa un evaluador de expresiones matemáticas que soporte paréntesis, prioridades, y operaciones básicas. Por ejemplo, evalúa correctamente expresiones como (5 + 3) \* (2 - 4 / 2).

Estructuras Complejas y Algoritmos

Algoritmo de Ordenación Personalizada: Crea un programa que implemente el algoritmo quicksort para ordenar un array de objetos por una de sus propiedades. Permite al usuario seleccionar la propiedad a ordenar.

Generador de Números Primos: Diseña una función que reciba dos números como rango y devuelva todos los números primos dentro de ese rango utilizando la criba de Eratóstenes.

Cálculo Recursivo: Implementa una función recursiva que calcule el factorial de un número. Limita la recursión a 10 niveles y lanza un error si se supera.

Objetos Predefinidos Avanzados

Validador de URLs: Diseña un programa que valide si una cadena es una URL válida. Extrae el protocolo, dominio y ruta, y muestra un mensaje en caso de error si la cadena no es válida.

Análisis de Texto Complejo: Escribe una función que tome un párrafo y determine las tres palabras más usadas. Incluye el número de veces que aparecen y muestra los resultados en orden descendente.

Calendario Personalizado: Implementa un calendario que permita al usuario ingresar un mes y un año, mostrando los días de ese mes y destacando los fines de semana.

Manipulación del DOM Avanzada

Edición de Texto en Tiempo Real: Diseña un editor de texto que permita al usuario seleccionar texto y aplicar estilos como negrita, cursiva y subrayado. Guarda el contenido editado en un formato HTML.

Constructor de Gráficos: Crea un programa que permita al usuario ingresar valores numéricos para generar un gráfico de barras dinámico dentro de un <canvas>. Personaliza los colores de las barras.

Sistema de Tabs Dinámicas: Diseña un sistema de pestañas dinámico que permita al usuario agregar nuevas pestañas, eliminar pestañas existentes, y editar el contenido de cada pestaña.

Eventos y Asincronía

Juego del Cronómetro: Diseña un juego en el que el usuario deba detener un cronómetro lo más cerca posible de los 10 segundos. Calcula la desviación en milisegundos y muestra un mensaje en función del resultado.

Arrastrar y Soltar Complejo: Implementa una funcionalidad que permita mover elementos entre dos listas mediante arrastrar y soltar. Las listas deben actualizarse dinámicamente para reflejar los cambios.

Temporizador de Eventos: Escribe un programa que permita programar varios temporizadores simultáneos. Cada temporizador debe ejecutar una acción diferente al finalizar, como cambiar el color del fondo o mostrar un mensaje.

Interacción con el Usuario

Buscador Inteligente: Diseña un buscador en el que, al escribir una palabra, se muestren automáticamente las coincidencias disponibles en una lista precargada de opciones.

Control de Animaciones: Usa requestAnimationFrame para animar un círculo en un <canvas> que siga el movimiento del mouse, pero con un retraso configurable.

Validación Dinámica: Crea un formulario que valide dinámicamente cada campo a medida que el usuario escribe. Si el usuario intenta enviar el formulario con errores, muestra un mensaje debajo del campo incorrecto.

Lógica Compleja y Programación Funcional

Simulador de Conexiones: Diseña un programa que simule una red social representada como un grafo, donde los nodos son usuarios y las conexiones son amistades. Implementa una función que encuentre los amigos en común entre dos usuarios.

Intérprete de Código Reducido: Crea un intérprete básico que acepte instrucciones como ADD 5, SUB 3, MULT 2 y PRINT, y ejecute los comandos en el orden indicado, manteniendo un acumulador.

Soluciones:

1:

for(R=0;R<=255;R=R+51){   
 if(R==0){  
    RR = "0"+R.toString(16);  
  }  
 else{  
    RR = R.toString(16);  
  }  
 for(G=0;G<=255;G=G+51){  
  if(G==0){  
    GG = "0"+G.toString(16);  
   }  
  else{  
    GG = G.toString(16);  
   }  
   for(B=0;B<=255;B=B+51){  
    if(B==0){  
     BB = "0"+B.toString(16);  
    }  
    else{  
    BB = B.toString(16);  
   }  
   console.log(RR+GG+BB);  
  }  
 }  
}

2:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<style>

div{

background-color: lightblue;

width: 100px;

height: 100px;

}

</style>

</head>

<body><script>

function generarAleatorio(){

//genero los numeros

R = Math.round( (Math.random() \* 15 ) );

G = Math.round( (Math.random() \* 15 ) );

B = Math.round( (Math.random() \* 15 ) );

//los paso a String

R = R.toString(16);

G = G.toString(16);

B = B.toString(16);

//los junto en una variable

RGB = "#" + R + "" + G + "" + B;

return RGB;

}

function cambioColor(){

RGB = generarAleatorio();

//si el aux no esta vacio le doy el valor

if(document.getElementById("botoncilloA").value != ""){

RGB = document.getElementById("botoncilloA").value ;

}

//le doy al fondo el valor del auxiliar

document.getElementById("miCelda").style.backgroundColor = document.getElementById("botoncilloA").value;

//vuelvo a generar el aleatorio

RGB = generarAleatorio();

//aigno los valores el del boton y el del fondo que va a tener

document.getElementById("botoncillo").value = RGB;

document.getElementById("botoncilloA").value = RGB;

}

//asigno el valor al cambio

document.getElementById("botoncillo").onclick = cambioColor;

window.onload = function() {

colorInicial = generarAleatorio();

document.getElementById("botoncillo").value = colorInicial;

document.getElementById("botoncilloA").value = colorInicial;

document.getElementById("miCelda").style.backgroundColor = document.getElementById("botoncilloA").value;

};

</script>

<center><div id="miCelda" >

</div></center>

<center><input type="button" value="" id="botoncillo" onclick="cambioColor()"></center>

<input type="hidden" value="" id="botoncilloA">

</body>

</html>

3: <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<script>

function testea () {

var expresion = document.miform.expresion1.value;

var cadena = document.miform.cadena1.value;

var OpExp = new RegExp(expresion,"");

document.miform.resultado1.value = OpExp.test(cadena);

}

</script>

</head>

<body>

<form name="miform">

<table>

<tr><td>Expresión Regular</td><td> <input type="text" name="expresion1" size="40"></td><td rowspan="3">Resultado <input type="text" name="resultado1"> </td></tr>

<tr><td>Cadena</td><td> <input type="text" name="cadena1" value="hola"></td></tr>

<tr><td colspan="3"><input type="button" value="Text Expresión regular" onclick="testea()"></td></tr>

</table>

</form>

</body>

</html>

4: <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<script>

function init() {

var midiv = document.getElementsByTagName('div')[0].style.backgroundColor = "#EE0000";

var miboton = document.getElementsByTagName('input')[0];

var mibotonS = document.getElementsByTagName('input')[1];

var mibotonR = document.getElementsByTagName('input')[2];

miboton.onclick = function () {console.log('\*onclick\*');};

miboton.onfocus = function () {console.log('\*onfocus\*');};

miboton.onblur = function () {console.log('\*onblur\*');};

miboton.ondblclick = function () {console.log('\*ondblclick\*');};

miboton.onblur = function () {console.log('\*onblur\*');};

mibotonS.onsubmit = function () {console.log('\*onblur\*');};

mibotonR.onreset = function () {console.log('\*onblur\*');};

}

window.onload = init;

</script>

</head>

<body>

<div

onclick="console.log('Has pinchado con el ratón \*onclick\*');"

ondblclick="console.log('Has hecho dobleClick con el ratón \*ondblclick\*');"

onmouseover="console.log('Acabas de pasar el ratón por encima \*onmouseover\*');"

onmouseout="console.log('Acabas de perder el foco del ratón \*onmouseout\*');"

onmousedown="console.log('Acabas de pulsar el ratón por -sin soltar- \*onmousedown\*');"

onmouseup="console.log('Acabas de soltar la pulsación del ratón \*onmouseup\*');">

Puedes pinchar sobre este elemento o simplemente pasar el ratón por encima<br><br>

</div>

<form name="miform">

<input type="button" value="eventos"><br>

<input type="submit" id="S" value="Envio"><br>

<input type="reset" id="R" value="Limpiar">

</form></body></html>

5:

<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

<title>CANVAS</title>

<style>

#micanvas{

border: #92C428 solid 3px;

}

</style>

<script>

var miCanvas;

var lienzo;

var x = 20;

var y = 45;

var radio = 20;

var anchuraCanvas;

var alturaCanvas;

var fin = anchuraCanvas - radio;

var inicio = radio;

var intervalo;

var sentidoX = 1;

var sentidoY = 1;

function carga(){

miCanvas = document.getElementById("micanvas");

lienzo = miCanvas.getContext('2d');

anchuraCanvas = window.parseInt(document.getElementById("micanvas").width);

alturaCanvas = window.parseInt(document.getElementById("micanvas").height);

lienzo.clearRect(0,0,anchuraCanvas,alturaCanvas);

}

function dibuja(){

carga();

lienzo.beginPath();

lienzo.fillStyle = "rgb(200,0,0)";

lienzo.arc(x,y,radio,0,2\*Math.PI,true);

lienzo.fill();

x = x + sentidoX \* 10;

y = y + sentidoY \* 10;

if (x+radio>=anchuraCanvas && sentidoX == 1){

sentidoX = -1;

}

if (x-radio<=0 && sentidoX == -1){

sentidoX = 1;

}

if (y+radio>=alturaCanvas && sentidoY == 1){

sentidoY = -1;

}

if (y-radio<=0 && sentidoY == -1){

sentidoY = 1;

}

}

function mover(){

intervalo = window.setInterval(dibuja,20);

document.getElementById("miBoton").value = "Parar";

document.getElementById("miBoton").onclick = parar;

}

function parar(){

window.clearInterval(intervalo);

document.getElementById("miBoton").value = "Iniciar";

document.getElementById("miBoton").onclick = mover;

}

function inicia(){

document.getElementById("miBoton").onclick = mover;

}

window.onload = inicia;

</script>

</head>

<body>

<form>

<input type="button" id="miBoton" value="Iniciar">

</form>

<br>

<canvas id="micanvas" width="500" height="500">

</canvas>

</body>

</html>