

Desarrollo Web en entorno cliente

Sintaxis de Javascript

Ejercicios propuestos

Ejercicios avanzados de sintaxis

Sudoku

Realiza una aplicación web que compruebe si una solución de un Sudoku es correcta o no.

Una sugerencia de diseño para la aplicación web es que debe tener 9x9 campos de texto y un botón “Comprobar”. Se recomienda tener una solución de un Sudoku válido para hacer pruebas.

La explicación de las reglas del Sudoku las tenéis aquí <https://es.wikipedia.org/wiki/Sudoku>

Internamente el programa realizará la comprobación de si el Sudoku es o no correcto en una función definida como **function esSudokuCorrecto(miArrayBi)** que devolverá true si es correcto, false en caso contrario.

SOLUCIÓN

```
//Ejemplo de sudokus
sudoku=new Array();
sudoku[0]=[1,2,3,4,5,6,7,8,9];
sudoku[1]=[7,8,9,1,2,3,4,5,6];
sudoku[2]=[4,5,6,7,8,9,1,2,3];
sudoku[3]=[3,1,2,6,4,5,9,7,8];
sudoku[4]=[9,7,8,3,1,2,6,4,5];
sudoku[5]=[6,4,5,9,7,8,3,1,2];
sudoku[6]=[2,3,1,5,6,4,8,9,7];
sudoku[7]=[8,9,7,2,3,1,5,6,4];
sudoku[8]=[5,6,4,8,9,7,2,3,1];

// Funcion que devuelve true si el horizontal de un sudoku es correcto, falso en caso contrario

function esCorrectoSudokuHorizontal(s,x){
    var i;
    // Array para marcar si una posicion esta o no, inicializado a 0
    posiciones=[0,0,0,0,0,0,0,0,0]
    for(i=0;i<9;i++){
        pos=s[x][i]-1;
        posiciones[pos]=1;
    }

    // Si estan las 9 posiciones marcadas
    for(i=0;i<9;i++){
        // Si alguna no se ha encontrado, devuelve falso
        if(posiciones[i]==0){
            return false;
        }
    }
    // Si estan todas, devuelve verdadero
    return true;
}

// Funcion que devuelve true si el vertical de un sudoku es correcto, falso en caso contrario

function esCorrectoSudokuVertical(s,x){
    var i;
    // Array para marcar si una posicion esta o no, inicializado a 0
    posiciones=[0,0,0,0,0,0,0,0,0]
    for(i=0;i<9;i++){
```

```

        pos=s[i][x]-1;
        posiciones[pos]=1;
    }

    // Si estan las 9 posiciones marcadas
    for(i=0;i<9;i++){
        // Si alguna no se ha encontrado, devuelve falso
        if(posiciones[i]==0){
            return false;
        }
    }
    // Si estan todas, devuelve verdadero
    return true;
}

// Funcion que devuelve true si la caja de un sudoku es correcto, falso en caso contrario

function esCorrectoSudokuCaja(s,x){

    var i,j;
    var iniX, finX;
    var iniY, finY;

    // Array para marcar si una posicion esta o no, inicializado a 0
    posiciones=[0,0,0,0,0,0,0,0,0]

    // Calculamos la iniX finX

    if(x==0 || x==1 || x==2){
        iniX=0;
        finX=3;
    }
    else if(x==3 || x==4 || x==5){
        iniX=3;
        finX=6;
    }
    else if(x==6 || x==7 || x==8){
        iniX=6;
        finX=9;
    }
    }

    /// Calculamos la iniY finY
    if(x==0 || x==3 || x==6){
        iniY=0;
        finY=3;
    }
    else if(x==1 || x==4 || x==7){
        iniY=3;
        finY=6;
    }
    else if(x==2 || x==5 || x==8){
        iniY=6;
        finY=9;
    }
    }

    // Comprobamos

    for(i=iniX;i<finX;i++){
        for(j=iniY;j<finY;j++){
            pos=s[i][j]-1;
            posiciones[pos]=1;
        }
    }
}

```

```

    }
}

// Si estan las 9 posiciones marcadas
for(i=0;i<9;i++){
    // Si alguna no se ha encontrado, devuelve falso
    if(posiciones[i]==0){
        return false;
    }
}
// Si estan todas, devuelve verdadero
return true;
}

// Funcion que devuelve true si un sudoku es correcto, falso en caso contrario

function esCorrectoSudoku(s){
    var i;
    // Presuponemos que es correcto e intentamos demostrar lo contrario
    esCorrecto=true;

    for(i=0;i<9;i++){
        // Si no es correcto...
        if(!esCorrectoSudokuHorizontal(s,i)){
            esCorrecto=false;
            break;
        }

        // Si no es correcto...
        if(!esCorrectoSudokuVertical(s,i)){
            esCorrecto=false;
            break;
        }

        // Si no es correcto...
        if(!esCorrectoSudokuCaja(s,i)){
            esCorrecto=false;
            break;
        }
    }
    return esCorrecto;
}

// Llamamos a la funcion para comprobar el sudoku
if(esCorrectoSudoku(sudoku)){
    alert("El sudoku es correcto");
}
else{
    alert("El sudoku no es correcto");
}

```

Patrones

Realiza una aplicación web que solicite una cadena de texto.

El programa debe decir cuántas veces ocurre cada uno de estos patrones sin distinguir mayúsculas y minúsculas: “00” “101”, “ABC”, “HO”.

Un carácter puede formar parte de más de un patrón encontrado. Por ejemplo: En la cadena “000” el patrón “00” aparece dos veces (una empieza en la posición 0 y otra empieza en la posición 1).

Internamente el programa realizará la cuenta de patrones con una función definida como **function numeroPatrones(texto)** que devolverá un número entero con el número de patrones encontrados.

SOLUCIÓN

```
// Funcion que recibe un texto y un patron y dice cuantas veces esta el patron
// No distingue mayusculas y minusculas y una letra
// puede formar parte de mas de un patron encontrado

function buscarPatron(texto,patron)
{
    // Importante declarar la i por el ambito de variables (que no la coja del general)
    var i;
    // Contador de cuantos hemos encontrados
    encontrados=0;

    // PASamos todo a mayusculas para no distinguir mayusculas de minusculas
    textoMAY=texto.toUpperCase();
    patronMAY=patron.toUpperCase();

    for(i=0;i<textoMAY.length;i++){
        // Sacamos la cadena desde i a i + el tamanyo del patron
        cad=textoMAY.substring(i,i+patronMAY.length);

        if(cad==patronMAY){
            // Encontrada coincidencia
            encontrados++;
        }
    }

    // Devolvemos la solucion
    return encontrados;
}

var texto="000111101000ABCHO";
var patronesABuscar=["00","101","ABC","HO"];
var i;

// Total de coincidencias de todos los patrones

total=0;
// Recorremos el vector patronesABuscar

for(i=0;i<patronesABuscar.length;i++){

    // Buscamos el patron
    res=buscarPatron(texto,patronesABuscar[i]);
    alert("El partron "+patronesABuscar[i]+" esta "+res+" veces");
    // Acumulamos el total
    total+=res;
}
```

```

}

alert("El total de patrones encontrados es "+total);

```

Buscaminas

Realiza una aplicación web que reciba en código mediante un array bidimensional (de longitud variable) un escenario de Buscaminas, donde haya un 0 donde no hay minas y un -1 donde si hay.

Para cada casilla que no tenga una mina, diga cuantas minas adyacentes hay (en diagonal, horizontal y vertical).

Internamente el programa realizará las acciones con una función definida como **function contandoMinas(miCampo)** que devolverá un array bidimensional con el número de minas adyacentes en cada posición.

SOLUCIÓN

```

// Buscaminas

// Funcion que dado un tablero y una posicion, indica cuantas minas hay alrededor
function cuantasMinas(tab,x,y){
    minasEnc=0;
    //Comprobamos adyacentes
    if(x>0){
        if(tab[x-1][y]==-1){
            minasEnc++;
        }
    }

    if(x<tab.length-1){
        if(tab[x+1][y]==-1){
            minasEnc++;
        }
    }
    if(y>0){
        if(tab[x][y-1]==-1){
            minasEnc++;
        }
    }

    if(y<tab[x].length-1){
        if(tab[x][y+1]==-1){
            minasEnc++;
        }
    }

    if(y>0 && x>0){
        if(tab[x-1][y-1]==-1){
            minasEnc++;
        }
    }
    if(y>0 && x<tab.length-1){
        if(tab[x+1][y-1]==-1){
            minasEnc++;
        }
    }

    if(x>0 && y<tab[x].length-1){
        if(tab[x-1][y+1]==-1){
            minasEnc++;
        }
    }
}

```

```

    }
}

if(x<tab.length-1 && y<tab[x].length-1){
    if(tab[x+1][y+1]==-1){
        minasEnc++;
    }
}

//Devolvemos el resultado
return minasEnc;
}

// Declaramos tablero origen y destino vacios
var tablero;
var minas;

tablero=new Array();

tablero[0]=[0,0,-1,0];
tablero[1]=[0,-1,-1,0];

tablero=[[-1,0,0,0],[0,0,0,0],[0,-1,0,0],[0,0,0,0]]

/* Otros ejemplos

    tablero=[[-1,-1,-1],[-1,0,-1]]

    tablero=[[-1,0,0,0],[0,0,0,0],[0,-1,0,0],[0,0,0,0]]

*/

// Recorremos el vector

// Inicilaizamos el array de minas
minas=new Array();

for(i=0;i<tablero.length;i++){
    // Declaramos dinamicamente el sub array
    minas[i]=new Array();

    for(j=0;j<tablero[i].length;j++){
        if(tablero[i][j]==-1){
            minas[i][j]=-1;
        }
        else{
            minas[i][j]=cuantasMinas(tablero,i,j);
        }
    }
}

for(i=0;i<tablero.length;i++){
    document.write(minas[i]+"<br>");
}

```