Desarrollo de Aplicaciones Telemáticas, examen práctico, 30/06/2023
Grado en Ingeniería en Tecnologías de la Telecomunicación
Grado en Ingeniería en Sistemas de la Telecomunicación
Universidad Rey Juan Carlos

Instrucciones

1. Ejecuta en un terminal `~mortuno/prepara` y comprueba que esto ha creado los ficheros
   - `~/dat.junio.23/matrices.TULOGIN.js` donde resolverás el ejercicio 1.
   - `~/dat.junio.23/figuras.TULOGIN.html` y `~/dat.junio.23/figuras.TULOGIN.js` que contienen una copia de tus ficheros de la práctica 4.7. Deberás modificarlos para hacer el ejercicio 2.
2. Emplea lo visto en clase, y solo lo visto en clase (que puedes consultar en las transparencias). No será válido ni JavaScript más antiguo ni más avanzado.

Ejercicio 1. (3.5 puntos)

En este ejercicio escribirás un par de funciones para generar y procesar matrices. Una matriz no es más que una lista cuyos elementos son filas, y estas filas, son a su vez filas. Completa el fichero `~/dat.junio.23/matrices.TULOGIN.js` para que sea un programa en JavaScript de acuerdo con la siguiente especificación.

1. Los elementos de la matriz serán cadenas formadas por una letra minúscula, un único dígito para indicar la fila (i) y un único dígito para indicar la columna (j). P.e. la cadena `a23` sería el elemento de la segunda fila, tercera columna.

   Un ejemplo de una de estas matrices sería

   ```
   [
     [ 'a10', 'a11', 'a12' ],
     [ 'a20', 'a21', 'a22' ],
   ]
   ```

2. Escribe una función llamada *crea_fila* que devuelva una fila de esta estructura. Recibirá:
   - Un prefijo, que será una letra minúscula.
   - Un número (entero), formado por la concatenación de los índices i, j del primer elemento de la fila
   - Un número (entero), formado por la concatenación de los índices i, j del último elemento de la fila

   Así expresado parece un poco complicado, pero con un ejemplo verás que es muy fácil: dando como entrada por ejemplo la cadena a, el número 20 y el número 24, la función tendría que devolver la lista de cadenas `[ 'a20', 'a21', 'a22', 'a23', 'a24' ]`. Insertando varias de estas listas en otra lista, tendremos una matriz.

3. Escribe una función llamada *a_cadena* que reciba una matriz con este formato y devuelva una cadena de texto que represente esta matriz, sin corchetes, comas ni comillas. Los elementos de una fila estarán separados por un espacio. Entre una fila y otra habrá un salto de línea (`'\n'`). Ejemplo: si recibe como entrada esta matriz

   ```
   [
     [ 'a10', 'a11', 'a12', 'a13', 'a14' ],
     [ 'a20', 'a21', 'a22', 'a23', 'a24' ],
     [ 'a30', 'a31', 'a32', 'a33', 'a34' ]
   ]
   ```

la función devolverá la cadena

```
a10 a11 a12 a13 a14
a20 a21 a22 a23 a24
a30 a31 a32 a33 a34
```

4. Cada una de estas funciones puede llamar a otras funciones, si lo crees conveniente.

5. En cualquier programa es muy importante comprobar que los parámetros que reciben las funciones son los correctos: número, tipo, rango, etc. Por razones de tiempo, aquí no es necesario que lo hagas. Pero indica en cada función en un comentario qué habría que comprobar. No *cómo*, solamente *qué*.

Solución

```
'use strict'
function crea_fila(prefijo, i0, i_n){
// Habría que comprobar:
// Número de parámetros = 3
// Primer parámetro es una letra minúsculas
// Segundo parámetro: número entero entre 10 y 99
// Tercer parámetro: número entero entre 10 y 99
// Tercer parámetro >= segundo parámetro
// Segundo parámetro y tercer parámetro son de la misma fila
//     (misma decena, misma i)

    let rval = [];
    for (let i=i0; i <= i_n; ++i){
        let elemento = prefijo + String(i);
        rval.push(elemento)
    }
    return rval;
}

function a_cadena(m){
// Habría que comprobar:
// m es una lista
// Todos los elementos de m son filas
// Todas las filas tienen el mismo número de elementos
// Cada fila es una lista
// Cada elemento de la fila es una cadena
// Cada cadena tiene el formato aij, donde 'a' es
//    minúscula, ij son dígitos
    let cadena = ""
    for (let fila of m){
        for (let elemento of fila){
            cadena = cadena + elemento + " ";
        }
        cadena = cadena + "\n"
    }
    return cadena;
}


let f1 = crea_fila('a', 10, 14);
let f2 = crea_fila('a', 20, 24);
let f3 = crea_fila('a', 30, 34);
let m = [f1,f2,f3]
console.log(m)

// Resultado:
// [
//   [ 'a10', 'a11', 'a12', 'a13', 'a14' ],
//   [ 'a20', 'a21', 'a22', 'a23', 'a24' ],
//   [ 'a30', 'a31', 'a32', 'a33', 'a34' ]
// ]
```

```
console.log(a_cadena(m))

// Resultado:

//a10 a11 a12 a13 a14
//a20 a21 a22 a23 a24
//a30 a31 a32 a33 a34
```

Ejercicio 2. (6.5 puntos)

Modifica los ficheros `~/dat.junio.23/figuras.TULOGIN.html` y `~/dat.junio.23/figuras.TULOGIN.js` según esta especificación:

- En modo gráfico, el ejercicio tiene que seguir funcionando como hasta ahora. Para ello, copia el directorio donde tuvieras las imágenes de esta práctica al directorio del examen (`~/dat.junio.23`)
- Hasta ahora, el usuario podía marcar o desmarcar una figura haciendo *clic* sobre ella. Esto provocaba que o bien se mostrase la figura normalmente, o bien una imagen de reverso o similar. Cuando el usuario marcaba una carta o similar en modo texto, no sucedía nada.
- Modifica tu práctica para que las figuras en modo texto tengan un comportamiento similar al de las figuras en modo gráfico: al hacer clic sobre una, su texto será reemplazado por una serie de asteriscos, tantos como letras hubiera originalmente. Al volver a hacer clic, el texto volverá a su estado original. Las figuras en modo texto seleccionadas aparecerán en la misma estructura de datos que las figuras en modo gráfico. En esta estructura de datos no se distinguirán las figuras en modo texto o en modo gráfico.

Ejemplo: al hacer clic sobre *3 de corazones* el texto cambiará a ********** y el 3 de corazones aparecerá en la estructura que almacena las figuras seleccionadas. Al hacer clic sobre **********, el texto volverá a ser *3 de corazones* y esta figura desaparecerá de la estructura de figuras seleccionadas.