

Desarrollo de Aplicaciones Telemáticas, examen práctico, 11/05/2023

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.mayo.23/listas.TULOGIN.js` donde resolverás el ejercicio 1.
 - `~/dat.mayo.23/cuentaImagen.TULOGIN.html` y `~/dat.mayo.23/cuentaImagen.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. 4 puntos

Escribe un programa en JavaScript en el fichero `~/dat.mayo.23/listas.TULOGIN.js` de acuerdo con la siguiente especificación.

1. El programa tendrá una función llamada *extraer_elemento* que:
 - a) Recibirá dos parámetros: una lista y un elemento.
 - b) Si el elemento pertenece a la lista, lo extraerá. De forma destructiva, esto es, la lista quedará modificada, dejará de contener el elemento.
 - c) Si el elemento está incluido más de una vez, extraerá la primera aparición y solo la primera aparición.
 - d) Si el elemento no pertenece a la lista, la función no hará nada.
 - e) No es necesario que compruebes si el programador invoca correctamente a esta función.
 - f) La función devolverá *true* si ha modificado la lista, *false* en otro caso.
 - g) Ejemplo de traza:

```
[ 'SOTA', 'sota', 'sota', 'caballo' ]
extraemos  sota
[ 'SOTA', 'sota', 'caballo' ]
extraemos  Comodín
[ 'SOTA', 'sota', 'caballo' ]
extraemos  sota
[ 'SOTA', 'caballo' ]
```

2. El programa tendrá una función llamada *extraer_elementos* que:
 - a) Recibirá dos parámetros: una lista y un elemento
 - b) Llamando a la función anterior, eliminará de forma destructiva todas las apariciones del elemento en la lista. Si no hay ninguna, no hará nada.
 - c) La lista quedará modificada, si procede, pero la función no devolverá ningún valor.
 - d) Ejemplo de traza:

```
[ 'SOTA', 'sota', 'sota', 'caballo' ]
extraemos todas las apariciones de sota
[ 'SOTA', 'caballo' ]
```

3. El programa tendrá una función llamada *contar_elemento* que:

- a) Recibirá dos parámetros: una lista y un elemento
- b) Devolverá el número de veces que el elemento aparece en la lista.
- c) Ejemplo de traza:

```
[ 'sota', 'caballo', 'sota', 'Sota' ]
apariciones de rey : 0
apariciones de sota : 2
```

4. El programa contendrá llamadas a estas funciones para probar su funcionamiento. Podrá tener funciones adicionales si lo crees conveniente.

Solución

```
'use strict'
function extraer_elemento(lista, elemento){
  let rval = false;
  let indice = lista.indexOf(elemento);
  if (indice!==-1){
    lista.splice(indice,1);
    rval = true;
  }
  return rval;
}

function extraer_elementos(lista, elemento){
  while( extraer_elemento(lista, elemento) ){
  }
}

function contar_elemento(lista, elemento){
  let rval = 0;
  for (let l of lista){
    if (l === elemento) {
      rval = rval+1;
    }
  }
  return rval;
}

let listado = ['SOTA', 'caballo', 'sota', 'sota', 'sota', 'sota']
let elemento;

console.log('listado:', listado)
elemento = "rey";
console.log("apariciones de",elemento,":",contar_elemento(listado,elemento))

elemento = "sota";
console.log("apariciones de",elemento,":",contar_elemento(listado,elemento))
elemento="sota"

extraer_elemento(listado,elemento)
console.log("extraemos",elemento)
console.log(listado)

elemento="caballo"
extraer_elementos(listado,elemento)
console.log("extraemos todos los ",elemento)
console.log(listado)

elemento="sota";
console.log("extraemos todos los ",elemento)
extraer_elementos(listado,elemento)
console.log(listado)

elemento="Comodín";
```

```
console.log("extraemos ",elemento)
extraer_elemento(listado,elemento)
console.log(listado)
```

Ejercicio 1. 6 puntos

Modifica los ficheros `~/dat.mayo.23/cuentaImagen.TULOGIN.html` y `~/dat.mayo.23/cuentaImagen.TULOGIN.js` de forma que sigan haciendo todo lo que hacían hasta ahora, y además, lo siguiente.

1. Cada vez que crees una de tus figuras en modo *sin repetición*, junto a ella aparecerá un botón al que llamaremos *sumar*, cuyo texto será el signo *más*. El *id* de este botón será el prefijo `sum_`, concatenado con el *id* de la imagen. Ejemplo, si la imagen es `reyPicas`, el botón será `sum_reyPicas`.
2. Cada vez que crees una de tus figuras en modo *sin repetición*, junto a ella aparecerá, además, un botón al que llamaremos *restar* cuyo texto será el signo *menos*. El *id* de este botón será el prefijo `res_`, concatenado con el *id* de la imagen. Ejemplo, si la imagen es `reyPicas`, el botón será `res_ReyPicas`.
3. Cuando tu programa trabaje en modo texto, no habrá novedades, no aparecerán estos botones. Tampoco cuando tu programa trabaje en modo imagen con repetición.
4. No te preocupes por el aspecto y disposición de estos dos botones. A menos que hayas resuelto el examen completo y te sobre tiempo.
5. Cuando el usuario haga *click* en alguna imagen o en los botones *sumar* o en los botones *restar*, en algún lugar de la pantalla se verá el *id* de la imagen o el botón.
6. Tu programa contendrá un array llamado *listado*, que inicialmente estará vacío.
7. Cada vez que el usuario haga *click* sobre un botón *sumar*, el programa añadirá a *listado* el *id* de la imagen correspondiente al botón a *listado*. Si ese *id* ya estaba en *listado*, aparecerá una vez más. Ejemplo de traza:

```
Listado: [ 'reinaCorazones', 'reyPicas']
El usuario hace click en sum_ReyPicas
Listado: [ 'reinaCorazones', 'reyPicas', 'reyPicas']
```

8. Cada vez que el usuario haga *click* sobre un botón *restar*, el programa eliminará de *listado* una aparición del *id* de la imagen correspondiente. Si el *id* no estaba en la lista, no hará nada. Emplea para esto la función *extraer_elemento* del ejercicio anterior (si te funciona)

Ejemplo de traza:

```
Listado: [ 'reinaCorazones', 'reyPicas', 'reyPicas']
El usuario hace click en res_reyPicas
Listado: [ 'reinaCorazones', 'reyPicas']
```

9. Observa que una vez que conozcas el *id* de un botón, podrás obtener el *id* de la imagen asociada quedándote con una subcadena. Ejemplo: la imagen asociada al botón `sum_reyPicas` es `reyPicas`.
10. En alguna parte de la pantalla se verá el *listado* y se actualizará cada vez que se modifique.
11. Recuerda que es muy conveniente hacer trazas detalladas. Si quieres puedes dejar todas las trazas, pero es preferible dejar sólo las que consideres más importantes.

Solución

Una solución completa real tendría que estar basada en una práctica 4.7. A continuación se muestra un programa similar, que contiene código como el que habría que añadir a una práctica.

<https://gsyc.urjc.es/~mortuno/cuentaImagen.html>