CURSO: CC3S2

Laboratorio Dirigido 03

1. Introducción

Objetivo general:
Afianzar el conocimiento y aplicación de Javascript

2 . Recursos Informáticos

- Recursos del sitio web del curso CC3S2
- http://jshint.com/about

3. DESARROLLO

Aunque en este laboratorio tiene que ejecutar código en su navegador, necesita tener Node.js instalado en su sistema para ejecutar el verificador de calidad del código. Si aún no ha instalado Node.js y el paquete npm efectúe su instalación ahora.
sudo apt install nodejs
node –version
sudo apt install npm
npm –version

Una vez que tenga Node.js instalado, cree un directorio lab03 y extraiga el contenido del archivo proporcionado en ese directorio. El archivo zip contiene el archivo cc3s2-test-lab03.html
y cc3s2-test-lab03.js que actúan como marco de prueba para el código que escribirá en este laboratorio.

Puede obtener la herramienta de calidad del código, JSHint (http://jshint.com/about), ejecutando el siguiente comando en el directorio lab03:
npm install -G JSHINT

Esto incluirá JSHint en el subdirectorio node_modules. Podrá ejecutarlo en todos los archivos JavaScript que se encuentren en el directorio lab03 ejecutando el comando:

npm run jshint

El código javascript que envie como resultado de su tarea debe empezar con ***“use strict”;*** Cuando se corra JSHint no deberían emitirse ningún tipo de warnings.

Para ejecutar la tarea, abra el archivo cc3s2-test-lab03.html en su navegador. La página web cargará el código JavaScript y ejecutará algunas pruebas. Dado que Node.js contiene una máquina virtual JavaScript, también puede correr los tests sin un navegador, usando el comando

npm test

Nota: estas pruebas no cubren todos los casos extremos. Están ahí para guiarlo y hacerle saber cuando cuente con la funcionalidad básica. Es su responsabilidad cumplir con todo lo que se pide en las siguientes especificaciones y con lo que no se prueba explícitamente en el archivo de prueba que le proporcionamos.

En este laboratorio deberá escribir o modificar algunas funciones de JavaScript.

Dada la disponibilidad de bibliotecas de JavaScript para resolver o ayudar a resolver prácticamente cualquier tarea de JavaScript que se le pueda asignar, y dado que el objetivo del laboratorio es aprender JavaScript, se prohíbe utilizar bibliotecas de JavaScript en sus soluciones. Las funciones integradas en las **matrices** JavaScript y los objetos **Date**, si se pueden utilizar.

TAREA 1

En su directorio lab03 haga un archivo nuevo denominado **cc3s2-multifiltro.js**

Declare una función global denominada **cc3s2HacerMultiFiltro** que toma un arreglo (**arregloOriginal**) como parámetro, y retorna una función que puede ser usada para filtrar los elementos de este arreglo.

La función retornada (**filtradorDeArreglo**) internamente da seguimiento a un arreglo denominado (**arregloActual**). Inicialmente **arregloActual** es igual al **arregloOriginal**.

La función **filtradorDeArreglo** toma como parámetros a 2 funciones: **criterioDeFiltrado** y **callback**

1. **criterioDeFiltrado** - Una función que toma como parámetro un elemento de un arreglo y retorna un booleano. Esta función se llama con cada elemento de **arregloActual** y **arregloActual** se actualiza para reflejar los resultados de la funcion **criterioDeFiltrado**. La funcion **criterioDeFiltrado** retorna falso para un elemento de que debe ser removido de **arregloActual**. De otro modo el elemento debe quedar en **arregloActual**. Si **criterioDeFiltrado** no es una función, la función retornada (**filtradorDeArreglo**) deberá retornar el valor de **arregloActual** sin haber hecho ninguna acción de filtrado.

2. **callback** – Una función que será llamada cuando se haya terminado el filtrado. **callback** toma el valor de **arregloActual** como un argumento. Accediendo a ***this*** dentro de la funcion **callback** debe referenciar el valor de **arregloOriginal**. Si **callback** no es una función, debe ser ignorada. **callback** no tiene valor de retorno.

La función **filtradorDeArreglo** debe retornarse así misma, a menos que el parámetro **criterioDeFiltrado** no se especifique. En éste caso deberá retornar el **arregloOriginal**. Deberá ser posible tener varias funciones **filtradorDeArreglo** operando al mismo tiempo.

El siguiente código muestra un ejemplo de cómo se podría usar las funciones definidas en esta tarea.

```
// Invocando a cc3s2HacerMultifiltro() con arregloOriginal = [1,2,3] retorna
// una función, guardada en la variable filtradorDeArreglo1,
// que puede ser usada repetidamente para filtrar el arreglo de entrada.
var filtradorDeArreglo1 = cc3s2HacerMultifiltro([1,2,3]);

// Invocando filtradorDeArreglo1 (con la función callback)
// para eliminar todos los números que no son igual a 2
filtradorDeArreglo1(function (elem) {
return elem !== 2; // verificar si el elemento no es igual a 2
}, function (arregloActual) {
// imprimir 'this' desde la función callback debería imprimir
// arregloOriginal que es [1,2,3]
console.log(this);
console.log(arregloActual); // imprime [1, 3]
});
```

```
// Invocando filtradorDeArreglo1 (sin la función callback )
// para filtrar (eliminar)todos los elementos diferentes a 3
filtradorDeArreglo1(function (elem) {
return elem !== 3; // verificar si el elemento no es igual a 3
});

// Invocando filtradorDeArreglo1 sin criterios de filtrado
// debería retornar el arregloActual
var arregloActual = filtradorDeArreglo1();
console.log('arregloActual', arregloActual); // imprime [1] ya que se filtraron 2 y 3

// ya que filtradorDeArreglo se retorna a si mismo, las llamadas pueden concatenarse
function filterTwos(elem) { return elem !== 2; }
function filterThrees(elem) { return elem !== 3; }
var filtradorDeArreglo2 = cc3s2HacerMultifiltro([1,2,3]);
var arregloActual2 = filtradorDeArreglo2(filterTwos)(filterThrees)();
console.log('arregloActual2', arregloActual2); // imprime [1] ya que se filtraron 2 y 3

// varios filtros activos al mismo tiempo
var filtradorDeArreglo3 = cc3s2HacerMultifiltro([1,2,3]);
var filtradorDeArreglo4 = cc3s2HacerMultifiltro([4,5,6]);
console.log(filtradorDeArreglo3(filterTwos)()); // imprime [1,3]
console.log(filtradorDeArreglo4(filterThrees)()); // imprime [4,5,6]
```

==========

Asegúrese que sus archivos js corren ***npm run jshint*** y ***npm test*** sin errores.

Comprima los archivos resultantes de los ejercicios de la tarea y suba el archivo comprimido **(CC3S2_Lab03_<Nombre_apellido>.zip)** al repositorio del curso.