

# CURSO: CC3S2

## Laboratorio Dirigido 05

### 1. Introducción

Objetivo general: Entender y aprovechar la Interacción de JavaScript y el DOM.

### 2 . Recursos Informáticos

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

### 3. jshint

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.

Una vez que tenga Node.js instalado, cree un directorio lab05 y extraiga el contenido del archivo proporcionado en ese directorio.

Puede obtener la herramienta de calidad del código, JSHint (<http://jshint.com/about>), ejecutando el siguiente comando en el directorio lab05:  
***npm install***

Esto incluirá JSHint en el subdirectorio node\_modules. Podrá ejecutarlo en todos los archivos JavaScript que se encuentren en el directorio lab05 ejecutando el comando:

***npm run jshint***

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

### 4. Tareas

#### Tarea 1: Implementar un procesador de plantillas.

En su directorio lab05, cree un nuevo archivo llamado procesador-plantillas.js. El código para El procesador de plantillas irá en este archivo.

Cree una clase para el procesador de plantillas (TemplateProcessor) que se construya (tenga un constructor) y que reciba una cadena que será la plantilla como parámetro y además la clase tiene un método fillIn. Cuando se invoca con un argumento que es un objeto tipo diccionario del tipo {clave: “valor”,...} , fillIn devuelve una cadena con la plantilla llena con valores que toma del objeto de diccionario.

TemplateProcessor debe escribirse utilizando un constructor y prototipo estándar de JavaScript.

El método fillIn devuelve la cadena de plantilla con cualquier texto de la forma {{clave}} reemplazado con la clave correspondiente del objeto diccionario que se pasa a la función.

Si la plantilla especifica una clave que no está definida en el objeto diccionario, la propiedad debe ser reemplazada con una cadena vacía. Si la propiedad está entre dos palabras, se podrá notar que al reemplazar la propiedad con una cadena vacía dará como resultado dos espacios en blanco consecutivos. Ejemplo: "Este {{propiedadNoDefinida}} es genial " → "Este es genial ". Esto se considerará algo normal. El programa deberá manejar solamente claves correctamente formateadas. No se tomarán en cuenta claves anidadas o llaves que estén sin balancear p.ej tipo: {{foo {{bar}}}} ó {{bar}}}

Un ejemplo de uso de `TemplateProcessor` es:

```
var template = 'Mi mes favorito es {{mes}} , pero no me gustan los dias {{dia}} o el año {{año}}';

var dateTemplate = new TemplateProcessor(template);

var dictionary = {mes: 'Julio', dia: '13', año: '2020'};

var str = dateTemplate.fillIn(dictionary);

assert(str === 'Mi mes favorito es Julio , pero no me gustan los dias 13 o el año 2020 ');
```

## Tarea 2 : Implementar un Selector de Fechas.

Para este problema, implementará dos Selectores de Fechas interactivos utilizando una combinación de HTML, CSS y JavaScript.

En su directorio lab05 , cree un archivo **DatePicker.js** que implemente una clase de JavaScript denominada **DatePicker** que se puede usar como en el siguiente ejemplo:

```
var datePicker = new DatePicker ("div1", function (id, fixedDate) {

    console.log ("DatePicker con id", id,

        "fecha seleccionada:", FixedDate.month + "/" + FixedDate.day + "/" + FixedDate.year);

});

datePicker.render (new Date("1/1/2009"));
```

El constructor toma un argumento que consiste en el atributo **id** de un **div** existente y una función de callback para la selección de la fecha. Cuando se selecciona una fecha, la función **callback** debe ser llamada con el argumento **id** y un objeto que contiene las propiedades **mes**, **dia**, **año** con el número que codifica la fecha (por ejemplo, {mes: 1, día: 30, año: 2016} es el código del de 30 de enero , 2016).

El objeto debe tener un método **render** que toma un argumento consistente en un objeto **Date** que seleccione un mes en particular (el objeto puede referirse a cualquier día dentro del mes). Cuando **render** es invocado, reemplaza el contenido del **div** del selector de fecha, con código

HTML que muestra un pequeño calendario de un mes como los que puede ver en un sitio web de reserva de viajes:

- El calendario debe mostrar los días del mes seleccionado en una cuadrícula con una línea para cada semana y una columna para cada día de la semana.
- Las semanas van desde el domingo a la izquierda hasta el sábado a la derecha. El calendario debe contener una fila de encabezado que muestre las abreviaturas de los días de la semana, como "Dom", "Lu", etc.
- Cada día del mes se muestra como un número.
- Algunas semanas, cuando se muestran en el selector de fechas, pueden contener días que no están en el mes seleccionado. Estos días deben mostrarse como el número de su mes respectivo, pero de forma atenuada, para indicar que no forman parte del mes actual.
- Todas las semanas mostradas deben contener al menos un día perteneciente al mes actual. La mayoría de los meses mostrarán 5 semanas, pero algunos meses pueden mostrar 4 o 6 según los días. El número de filas en su calendario no debe ser fijo.
- El calendario debe mostrar el nombre del mes y el año en la parte superior del calendario. Además, debe mostrar controles como "<" y ">" en los que se puede hacer clic para cambiar la visualización del calendario al mes anterior o al mes siguiente.
- Al hacer clic en un día válido del mes actual, se debe invocar el callback especificado en el constructor con los argumentos descritos anteriormente. Al hacer clic en días que pertenecen a meses que no sean el mes actual, no se debe invocar el callback.

Una vez que haya creado la clase JavaScript, deberá probarlo en el archivo proporcionado **datepicker.html** (no necesita modificar este archivo html) Este archivo contiene dos elementos **div** vacíos, más algo de código JavaScript que invoca la clase DatePicker para mostrar un selector de fecha en cada uno de los divs. Uno de los selectores de fecha muestra inicialmente el mes actual y el otro muestra el mes de enero de 2009. Debería ser posible cambiar el mes de cada selector de fecha de forma independiente utilizando los controles de ese selector de fecha.

El archivo html proporcionado no tiene estilo, así que cree una hoja de estilo con el nombre del archivo **datepicker.css** para aplicar estilo a los calendarios y hacer que se vean bien. La etiqueta de enlace correspondiente que requiere el archivo css ya se ha agregado en el archivo html.

#### Notas:

No puede usar jQuery ni ningún otro paquete de biblioteca JavaScript en esta tarea. El objetivo de esta tarea es conocer las características fundamentales de JavaScript / DOM de bajo nivel.

Sin embargo, puede utilizar cualquier función incorporada de JavaScript. Por ejemplo, puede encontrar útiles las siguientes funciones integradas

- Métodos de la clase Date.
- parseInt, parseFloat, isNaN.

Algunas de las siguientes propiedades de los elementos DOM y otras funciones, pueden resultarle útiles en esta tarea:

- `cells`
- `firstChild`
- `nextSibling`
- `rows`
- `tagName`
- `textContent`
- `innerHTML`
- `document.createElement(tagName, [options])`
- `RegExp`

Puede usar la función `console.log` para depurar.

También puede encontrar útil la declaración del debugger en la depuración. Si el texto `"debugger";` se ejecuta en un programa JavaScript, hace que el programa se ejecute en el depurador JavaScript. Esto es análogo a un punto de interrupción, excepto que su código puede controlar cuándo se activa.

Para saber si existen errores abra la consola de JavaScript.

La consola de JavaScript y Chrome DevTools serán herramientas muy útiles para el desarrollo de sus tareas.

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

Comprima los archivos resultantes de los ejercicios de la tarea, incluyendo un archivo **CC3S2\_Lab05\_<Nombre\_apellido>.doc** con screenshots de las 2 tareas y el resultado de correr jshint.

**Suba el archivo comprimido (CC3S2\_Lab05\_<Nombre\_apellido>.zip) al repositorio del curso.**