

Ejercicios jquery

SEBASTIAN CAMILO CARDENAS RODRIGUEZ

**ING.
DIEGO ALONSO OJEDA MEDINA**

**ANALISIS Y DESARROLLO DE SISTEMAS DE INFORMACIÓN
CENTRO INDUSTRIAL DE MANTENIMIENTO Y MANUFACTURA (CIMM)**

SOGAMOSO-BOYACA

2016

Ejercicios jquery

(Document Reader, Inclusion, Sintaxis, Selectores): La sintaxis de jQuery es hecho a medida para la selección de los elementos HTML y realizar alguna acción sobre el elemento.

`$ (This) .hide ()` - oculta el elemento actual.

`$ ("P") ocultar ()` -. Disimular todo elementos `<p>`.

`$ (". Test") ocultar ()` -. Oculta todos los elementos con `class = "prueba"`.

`$ ("# Test") ocultar ()` -. Esconde el elemento con `id = "prueba"`.

Document Reader: Una página no puede ser manipulado de forma segura hasta que el documento está "listo". jQuery detecta este estado de preparación para usted. Código incluido dentro `$(document)`.

A `$(document).ready()` block.

`$(document).ready(function() {`

`console.log("ready!");`

`});`

Inclusion JQuery: Esta extensión requiere jQuery en su página web. Usted tiene que asegurarse de que usted siempre tiene una sola versión de jQuery incluido y que el jQuery incluye se encuentra antes de la inclusión de las librerías de JavaScript que provienen de la extensión.

`page.includeJSFooter {`

```

        bannerjQuery =
//ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js
        bannerjQuery.external = 1
        bannerjQuery.forceOnTop = 1
    }

```

Los selectores de jQuery: selectores de jQuery se utilizan para "encontrar" (o seleccionar) elementos HTML basándose en su nombre, id, clases, tipos, atributos, valores de atributos y mucho más. Está basado en los existentes.

El selector de elemento: El selector de elementos jQuery selecciona elementos basados en el nombre del elemento.

```

$(document).ready(function(){
    $("button").click(function(){
        $("p").hide();
    });
});

```

El Selector de #id: El selector de jQuery #id utiliza el atributo id de una etiqueta HTML para encontrar el elemento específico.

```

$(document).ready(function(){
    $("button").click(function(){
        $("#test").hide();
    });
});

```

El Selector de .class: El selector de clase jQuery encuentra elementos con una clase específica.

```

$(document).ready(function(){
    $("button").click(function(){

```

```
        $(".test").hide();  
    });  
});
```

EVENTO CLICK: El evento click se produce cuando se hace clic en un elemento. El método click () activa el evento clic, o atribuye una función a ejecutar cuando se produce un evento de clic.

```
$(".p").click(function(){  
    $(this).hide();  
});
```

EVENTO DBLCLICK: El evento dblclick se produce cuando un elemento se hace doble clic. El método dblclick () activa el evento dblclick, o atribuye una función a ejecutar cuando se produce un evento dblclick.

```
$(".p").dblclick(function(){  
    alert("The paragraph was double-clicked");  
});
```

EVENTO MOUSEENTER: El evento Mouseenter se produce cuando el puntero del ratón está sobre (entra) el elemento seleccionado.

```
$(".p").mouseenter(function(){  
    $(".p").css("background-color", "yellow");  
});
```

EVENTO MOUSELEAVE: El evento mouseleave se produce cuando el puntero del ratón sale del elemento seleccionado.

```
$(".p").mouseleave(function(){  
    $(".p").css("background-color", "gray");  
});
```

EVENTO KEYPRESS: El orden de los acontecimientos relacionados con el evento de pulsación de tecla. El método de pulsación de tecla () activa el evento de

pulsación de tecla, o atribuye una función a ejecutar cuando se produce un evento de pulsación de tecla.

```
$("#input").keypress(function(){  
    $("#span").text(i += 1);  
});
```

EVENTO KEYDOWN: El método `keydown ()` activa el evento `KeyDown` o concede una función a ejecutar cuando se produce un evento `KeyDown`.

```
$("#input").keydown(function(){  
    $("#input").css("background-color", "yellow");  
});
```

EVENTO KEYUP: El método `keyup ()` activa el evento `keyup`, o atribuye una función a ejecutar cuando se produce un evento `keyup`.

```
$("#input").keyup(function(){  
    $("#input").css("background-color", "pink");  
});
```

EVENTO SUBMIT: El método `submit ()` activa el evento `presentará`, o concede una función a ejecutar cuando un evento ocurre `enviar`.

```
$("#form").submit(function(){  
    alert("Submitted");  
});
```

EVENTO CHANGE: El método de cambio `()` se inicia el evento de cambio, o concede una función que se ejecute cuando se produce un evento de cambio.

```
$("#input").change(function(){  
    alert("The text has been changed.");  
});
```

EVENTO FOCUS: El evento de foco se produce cuando un elemento se centrará (cuando se selecciona mediante un clic del ratón o "ficha-navegación" a ella. El

método de enfoque () se inicia el evento de foco, o se fija una función para ejecutar cuando se produce un evento de foco.

```
$("#input").focus(function(){  
    $("#span").css("display", "inline").fadeOut(2000);  
});
```

EVENTO BLUR: El método de desenfoque () activa el evento de toma estable o atribuye una función a ejecutar cuando se produce un evento de desenfoque.

```
$("#input").blur(function(){  
    alert("This input field has lost its focus.");  
});
```

EVENTO LOAD: Este evento funciona con los elementos asociados a una dirección URL (imagen, escritura, marco, marco flotante), y el objeto ventana.

```
$("#img").load(function(){  
    alert("Image loaded.");  
});
```

EVENTO RESIZE: El evento de cambio de tamaño se produce cuando la ventana del navegador cambia de tamaño. El método de cambio de tamaño () activa el evento de cambio de tamaño, o se conecta una función a ejecutar cuando se produce un evento de cambio de tamaño.

```
$(window).resize(function(){  
    $('span').text(x += 1);  
});
```

EVENTO SCROLL: El evento de desplazamiento se produce cuando el usuario se desplaza en el elemento especificado. El evento de desplazamiento funciona para todos los elementos desplazables y el objeto de la ventana (ventana del navegador).

```
$("#div").scroll(function(){  
    $("#span").text(x += 1);
```

```
});
```

EVENTO UNLOAD: el método de descarga () fue desaprobado en jQuery versión 1.8 y eliminó en la versión 3.0. El evento de descarga se produce cuando el usuario se desplaza fuera de la página.

```
$(window).unload(function(){  
    alert("Goodbye!");  
});
```

Forms

Blur: su funcionalidad es la detección de la pérdida del foco en los diferentes elementos del DOM de una página Web. El evento se produce cuando falta la definición <input>

```
$(“input”).blur(function() {  
    alert(“this input field has lost its focus.”);  
});
```

Change: este método detecta el cambio de valor del elemento, es decir altera un texto cuando se cambia un <input> seleccionado en el DOM de nuestra Web.

```
$(“input”).change(function() {  
    alert(“the text has been changed.”);  
});
```

focus: su funcionalidad es la de detectar el foco en los diferentes elementos del DOM. El evento se produce cuando el foco <input> campo obtenido del foco.

```
$(“input”).focus(function() {  
    $(“span”).css(“display” , “inline”).fadeOut(2000);  
});
```

focusin: su funcionalidad es establecer el color de fondo de un elemento <div> cuando el elemento secundario de obtener el foco.

```
$(“div”).fucsin(function() {  
    $(this).css(“background-color” , “#FFFFCC”);  
});
```

focusOut: su funcionalidad es establecer de un elemento <div> cuando el elemento <div> o cualquier elemento secundario pierde el foco.

```
$(“div”).fucusout(function() {  
    $(this).css(“background-color” , “#FFFFFF”);  
});
```

param: su funcionalidad es crear una representación en serie de una matriz o un objeto, los valores serializados se pueden utilizar en la cadena de consulta URL al hacer una petición AJAX.

```
$(“button”).click(function() {  
    $(div).text($.param(personOBJ));  
});
```

select: su funcionalidad es enlazar un controlador de eventos “seleccionador” evento JavaScript o desencadenador ese evento es un elemento.

```
$(“#target”).select(function() {  
    alter(“Handler for .select() called.”);  
});
```

serialize: su funcionalidad es crear una cadena de texto URL codificada por la serialización de los valores del formulario, puede seleccionar uno o mas elementos del formulario o el propio formulario.

```
$(“button”).click(function() {  
    $(“div”).text($(“from”.serialize));  
});
```


serializeArray: su funcionalidad es codificar un conjunto de elementos de formulario como una serie de un nombre y valores.

```
$(“from”).submit(function( event ) {  
    console.log($(this).serializeArray() );  
    event.preventDefault();  
});
```

submit: su funcionalidad es mostrar una alerta cuando se esta enviando un formulario.

```
$(“form”).submit(function() {  
    alert(“submitted”);  
});
```

val: su función es obtener el valor actual del primer elemnto en el conjunto de los elementos coincidentes.

```
$(“button”).click(function() {  
    $(“input:text”).val(“Glen Quagmire”);  
});
```

Efectos.

Hide: su función es ocultar los elementos coincidentes.

```
$(“.target”).hide();
```

show: su función es mostrar los elementos ocultos seleccionados, no funciona en los elementos ocultos con los métodos de jQuery y diplay.

```
$(“button”).click(function() {  
    $(“p”).show();  
});
```

slide: su función es deslizar el elemento fuera de la ventana gráfica.

```
$("#clickme").click(function() {  
    $("#book").slide("slow", function() {  
    });  
});
```

animate: su función es realizar una animación personalizada de un conjunto de propiedades de CSS.

```
$( "#clickme" ).click(function() {  
    $( "#book" ).animate({  
        opacity: 0.25,  
        left: "+=50",  
        height: "toggle"  
    }, 5000, function()  
    });  
});
```

stop: su función es detener la animación en ejecución en los elementos coincidentes.

```
$( "#hoverme-stop-2" ).hover(function() {  
    $( this ).find( "img" ).stop( true, true ).fadeOut();  
    }, function() {  
    $( this ).find( "img" ).stop( true, true ).fadeIn();  
    });
```

callbacks: su función es una sentencia de JavaScript que ejecuta línea por línea. sin embargo, con efectos, la línea de código se puede ejecutar a pesar de que el efecto no se ha terminado. esto puede crear errores.

```
$( "jQuery" ).callbacks( flags );  
var acciones = $.callabacks( once );
```

chaining: su función es devolver un objeto jQuery, que luego se puede utilizar para llamar a otro método. esto permite hacer el encadenamiento de comandos donde se pueden realizar varios métodos en el mismo conjunto de elementos.

```
$("#p1").css("color", "red")  
    .slideUp(2000)  
    .slideDown(2000);
```

manejo de html y css.

text: se utiliza para establecer la alineación horizontal de un texto.

```
h1 {  
    text-align: center;  
}  
h2 {  
    text-align: left;  
}
```

attr: su función es obtener el valor de un atributo para el primer elemento en el conjunto de los elementos coincidentes. cuando se utiliza este método para devolver el valor de atributos, devuelve el valor del elemento emparejado primero.

```
$("button").click(function(){  
    $("img").attr("width", "500");  
});
```

append: su función es la inserción de contenido, especificado por el parámetro, al final de cada elemento en el conjunto de los elementos coincidentes.

```
$("p").append("Some appended text.");
```

prepend: su función es insertar el contenido, especificado por el parámetro, al principio de cada elemento en el conjunto de los elementos coincidentes.

```
$("#button").click(function(){  
    $("#p").prepend("<b>Prepended text</b>");  
});
```

after: su función es insertar el contenido, especificado por el parámetro, después de cada elemento en el conjunto de los elementos coincidentes.

```
$("#button").click(function(){  
    $("#p").after("<p>Hello world!</p>");  
});
```

before: su función es insertar el contenido, especificado por el parámetro, antes de cada elemento en el conjunto de elementos coincidentes.

```
$("#button").click(function(){  
    $("#p").before("<p>Hello world!</p>");  
});
```

remove: su función es quitar los elementos HTML existentes, el método remove elimina elementos seleccionados y sus elementos secundarios.

```
$("#div1").remove();
```

empty: su función es eliminar todos los nodos hijos secundarios y el contenido de los elementos seleccionados. Este método no elimina el elemento en sí, o sus atributos.

```
$("#button").click(function(){  
    $("#div").empty();  
});
```

addclass: su función es añadir uno o más nombres de clase a los elementos seleccionados, este método no elimina los atributos de clase existentes, solo añade uno o más nombres de clase para el atributo de clase.

```
$("#button").click(function(){  
    $("#p:first").addClass("intro");  
});
```

removeClass: su función es eliminar uno o varias clases de los elementos seleccionados, si no se especifica ningún parámetro, este método eliminara todos los nombres de las clases de los elementos seleccionados.

```
$("#button").click(function(){  
    $("#p").removeClass("intro");  
});
```

toggleClass: su función es añadir o quitar una o mas clases de cada elemento en el conjunto de elementos coincidentes, ya sea en función de la presencia de la clase o el valor del argumento del estado.

```
$("#button").click(function(){  
    $("#p").toggleClass("main");  
});
```

css: su función es obtener las propiedades de estos estilos calculados para el primer elemento en el conjunto de los elementos coincidentes.

```
$("#div.exmaple").css("width" , function ( index ) {  
    retun index * 50;  
});
```

width: su función es obtener la anchura calculada actual para el primer elemento en el conjunto de los elementos seleccionados.

```
$("#button").click(function(){  
    alert($("#div").width());
```

```
});
```

height: su función es obtener la altura calculada actual para el primer elemento en el conjunto de los elementos seleccionados.

```
$("#button").click(function(){  
    alert($("#div").height());  
});
```

innerWidth: su función es obtener la anchura interior computarizado actual para el primer elemento en el conjunto de elementos emparejados, incluyendo el relleno, pero no frontera.

```
$(selector).innerWidth()
```

outerWidth: su función es obtener la altura calculada actual para el primer elemento en el conjunto de elementos emparejados, incluyendo el relleno.

```
$(selector).innerHeight()
```

outerHeight: su función es obtener la altura exterior computarizado actual (incluid el relleno, borde y margen) para el primer elemento en el conjunto de los elementos seleccionados.

```
$(selector).outerHeight(includeMargin)
```

Modificar

Parent: su función es obtener el padre de cada elemento en el conjunto actual de los elementos seleccionados opcionalmente filtradas por un selector.

```
$(document).ready(function(){  
    $("#span").parent().css({"color": "red", "border": "2px solid red"});  
});
```

parentsUntil: su función es obtener los antepasados de cada elemento en el conjunto actual de los elementos seleccionados, y hasta, pero sin incluir el elemento emparejado por el selector, nodo DOMS, o un objeto jQuery.

```
$(document).ready(function(){
    $("span").parentsUntil("div").css({"color": "red", "border": "2px solid red"});
});
```

Children: su función es obtener los hijos d cada elemento en el conjunto de los elementos seleccionados, opcionalmente filtradas por un selector.

```
$(document).ready(function(){
    $("ul").children().css({"color": "red", "border": "2px solid red"});
});
```

Find: su función es obtener los descendientes de cada elemento en el conjunto actual de los elementos coincidentes, filtrada por un selector, objetos jQuery o un elemento.

```
$(document).ready(function(){
    $("ul").find("span").css({"color": "red", "border": "2px solid red"});
});
```

Siblings: su función es obtener los hermanos de cada elemento en el conjunto de elementos seleccionados opcionalmente filtradas por un selector.

```
$(document).ready(function(){
    $("li.start").siblings().css({"color": "red", "border": "2px solid red"});
});
```

Next: su función es obtener el hermano inmediatamente después de cada elemento en el conjunto de elementos seleccionados. Si se proporciona un selector.

```
$(document).ready(function(){
```

```
    $("li.start").next().css({"color": "red", "border": "2px solid red"});  
});
```

Nextall: su función es obtener todos los siguientes hermanos de cada elemento en el conjunto de elementos coincidentes, opcionalmente filtrados por un selector.

```
$(document).ready(function(){  
    $("li.start").nextAll().css({"color": "red", "border": "2px solid red"});  
});
```

nextUntil: su función es obtener todos los siguientes hermanos de cada elemento hasta, pero sin incluir el elemento emparejado por el selector, nodo DOM o un objeto jQuery pasado.

```
$(document).ready(function(){  
    $("li.start").nextUntil("li.stop").css({"color": "red", "border": "2px solid  
red"});  
});
```

prev: su función es obtener el hermano inmediatamente anterior de cada elemento en el conjunto de elementos seleccionados.

```
$(document).ready(function(){  
    $("li.start").prev().css({"color": "red", "border": "2px solid red"});  
});
```

prevAll: su función es obtener todos los hermanos anteriores de cada elemento en el conjunto de elementos seleccionados, opcionalmente filtrados por un selector.

```
$(document).ready(function(){  
    $("li.start").prevAll().css({"color": "red", "border": "2px solid red"});  
});
```


prevUntil: su función es obtener todos los hermanos anteriores de cada elemento hasta, pero sin incluir el elemento emparejado por el selector, nodo DOM o un elemento jQuery.

```
$(document).ready(function(){  
    $("li.start").prevUntil("li.stop").css({"color": "red", "border": "2px solid  
red"});  
});
```

First: su función es reducir el conjunto de elementos emparejados a la primera en el conjunto.

```
$("div p").first()
```

Last: su función es reducir el conjunto de elementos que coinciden con el final en el conjunto.

```
$("p").last()
```

Eq: su función es reducir el conjunto de elementos que coincidan con el que está en el índice especificado.

```
$("p").eq(1).css("background-color", "yellow");
```

Filter: su función es reducir el conjunto de elementos emparejados a los que coinciden con el selector o pasan la prueba de la función.

```
$("p").filter(".intro")
```

Not: su función es eliminar los elementos del conjunto de elementos seleccionados.

```
$("p").not(".intro")
```

Ajax: su función es realizar una solicitud HTTP asíncrona (AJAX).

```
$("#button").click(function(){  
    $.ajax({url: "demo_test.txt", success: function(result){  
        $("#div1").html(result);  
    });  
});
```

Load: su función es cargar los datos desde el servidor y colocar el HTML devuelto en el elemento seleccionado.

```
$("#button").click(function(){  
    $("#div1").load("demo_test.txt");  
});
```

Get: su función es recuperar uno de los elementos que coinciden con el objeto jQuery.

```
$("#button").click(function(){  
    $.get("demo_test.asp", function(data, status){  
        alert("Data: " + data + "\nStatus: " + status);  
    });  
});
```

Post: su función es cargar los datos desde el servidor mediante una petición HTTP POST.

```
$("#button").click(function(){  
    $.post("demo_test.asp", function(data, status){  
        alert("Data: " + data + "\nStatus: " + status);  
    });  
});
```

```
});  
});
```

Json sintaxis: los datos JSON se escribe como pares de nombre / valor. Un par nombre / valor consiste en un nombre de campo (entre comillas), seguido de dos puntos y seguido por un valor.

```
"firstName":"John"
```

Json HTTP: un uso común de JSON es para leer datos desde un servidor web, y mostrar los datos en una página web.

```
function myFunction(arr) {  
    var out = "";  
    var i;  
    for(i = 0; i < arr.length; i++) {  
        out += '<a href="' + arr[i].url + '">' + arr[i].display + '</a><br>';  
    }  
    document.getElementById("id01").innerHTML = out;  
}
```

Json files: un uso común es para leer los datos desde un servidor web y mostrar los datos en una página web.

```
function myFunction(arr) {  
    var out = "";  
    var i;  
    for(i = 0; i < arr.length; i++) {  
        out += '<a href="' + arr[i].url + '">' + arr[i].display + '</a><br>';  
    }  
    document.getElementById("id01").innerHTML = out;  
}
```

Json SQL: su uso común es para leer datos JSON desde un servidor web con PHP y MySQL.

jQuery UI Interacciones P1

Draggable: su función es permitir que los elementos puedan moverse con el ratón. Habilitar la función que puede arrastrarse en cualquier elemento DOM, mover el objeto arrastrable haciendo clic sobre ella con el ratón y arrastrándolo en cualquier lugar dentro de la ventana.

```
$(function() {  
    $("#draggable").draggable({  
        revert : function(event, ui)  
    });  
});
```

Droppable: su función es crear objetos para los elementos que puedan arrastrarse, permite a cualquier elemento DOM ser lanzables, un objetivo para los elementos que pueden arrastrarse.

```
$("#dock").droppable({  
    drop: function(event, ui) {  
    }  
});
```

Resizable: su función es cambiar el tamaño de un elemento con el ratón, permitir a cualquier elemento DOM ser de tamaño variable, con el cursor agarrar el borde derecho inferior y arrastre a la anchura o la altura que desea.

```
$(selector, context).resizable({option1: value1, option2: value2..... });
```

Selectable: su función es usar el ratón para seleccionar los elementos de forma individual o en grupo, habilitar un elemento DOM para ser seleccionable.

```
$(function() {
```

```
$( "#selectable-2" ).selectable({  
    delay : 1000  
});
```

Sortable: su función es reordenar los elementos de una lista o de cuadrícula utilizando el ratón, permitiendo a un grupo de elementos DOM para ser clasificable.

```
$(function () {  
    $("ol.example").sortable();  
});
```

jQuery Widgets: su función es convertir un par de colectores y paneles de contenido en un acordeón, el margen de beneficio de su contenedor necesita pares de cabecera y los paneles de contenido.

```
$( ".selector" ).accordion({  
    active: 2  
});
```

Autocomplete: su función es permitir al usuario seleccionar de una lista pre-poblada de los valores a medida que escribe, el aprovechamiento de búsqueda y filtrado de forma rápida.

```
$( ".selector" ).autocomplete({  
    appendTo: "#someElem"  
});
```

Button: su función es realizar los elementos por defecto de forma como los botones, las entradas y los anclajes a los botones temable con vuelo estacionario apropiado y estilos activos.

```
$( ".selector" ).button({
```

```
classes: {  
    "ui-button": "highlight"  
}  
});
```

Checkboxradio: su función es convertir entrada de tipo radio a la casilla de verificación a los botones temable.

```
$( ".selector" ).checkboxradio({  
    classes: {  
        "ui-checkboxradio": "highlight"  
    }  
});
```

Controlgroup: su función es proporcionar una agrupación visual de botón y otros widgets de entrada.

```
$( ".selector" ).controlgroup({  
    classes: {  
        "ui-controlgroup": "highlight"  
    }  
});
```

Datepicker: su función es seleccionar una fecha de un calendario emergente o en línea.

```
$( "#datepicker" ).datepicker({  
    beforeShowDay: $.datepicker.noWeekends  
});
```

Dialog: su función es obtener el contenido abierto en una superposición interactiva.

```
$( ".selector" ).dialog({  
    appendTo: "#someElem"});
```

jQuery UI Interacciones

Menú: su función es configurar con el ratón y el teclado interacciones para la navegación.

```
$( ".selector" ).menu({  
    classes: {  
        "ui-menu": "highlight"  
    }  
});
```

Progressbar: su función es mostrar el estado de un determinado proceso o indeterminado.

```
$( ".selector" ).progressbar({  
    classes: {  
        "ui-progressbar": "highlight"  
    }  
});
```

Selectmenu: su función es duplicados y se extiende la funcionalidad de un elemento select de HTML nativa para superar las limitaciones del control nativo.

```
$( ".selector" ).selectmenu({  
    appendTo: "#someElem"  
});
```

Slider: su función es arrastrar un controlador para seleccionar un valor numérico.

```
$( ".selector" ).slider({  
    animate: "fast"  
});
```

Spinner: su función es mejora de una entrada de texto para introducir valores numéricos, con los botones ARRIBA / ABAJO y flecha entrega de llaves.

```
$( ".selector" ).spinner({  
  classes: {  
    "ui-spinner": "highlight"  
  }  
});
```

Tabs: su función es obtener el área de contenido individual con múltiples paneles, cada uno asociado con un encabezamiento en una lista.

```
$( ".selector" ).tabs({  
  classes: {  
    "ui-tabs": "highlight"  
  }  
});
```

Tooltip: su función es mostrar información sobre herramientas temable personalizables, que sustituye a la información de herramientas nativas.

```
$( ".selector" ).tooltip({  
  classes: {  
    "ui-tooltip": "highlight"  
  }  
});
```

jQuery UI Temas y Utilidades

Position: su función es obtener las coordenadas actuales del primer elemento en el conjunto de elementos coincidentes, en relación con el padre de desplazamiento.

```
$("button").click(function(){
```



```
var x = $("p").position();  
alert("Top: " + x.top + " Left: " + x.left);  
});
```

Widget Factory: su función es crear plugins jQuery con estado utilizando la misma abstracción como todos los widgets jQuery UI.

```
$( ".selector" ).progressbar({  
  classes: {  
    "ui-progressbar": "highlight"  
  }  
});
```