**FUNCIONES DE JQUERY**

**addClass()**

$("#contenido").addClass("nombreclase");

Con la función addClass() lo que hacemos añadir una clase ya establecida en nuestro CSS al contenedor con id=”contenido”. En e este caso le estamos añadiendo los estilos de la clase “nombreclase”.

**after()**

$("#contenido").after("<p>hola!</p>");

El método after() añade contenido html delante de lo que haya dentro del contenedor con id=”contenido”.

**ajax()**

var dataString = 'id='+service;

$.ajax({

type: "POST",

url: "archivo.php",

   data: dataString,

   success: function() {

   ...

   }

});

Con el método ajax() lo que conseguimos es realizar una llamada a un archivo php sin tener que recargar la página ni hacer una petición al servidor.

**append()**

$("#contenido").append("<p>hola!</p>");

El método append() añade contenido html al contenedor con id=”contenido”.

**appendTo()**

$("#contenido2").appendTo("#contenido1");

El método appendTo() copia o mueve el contenedor con id=”cotenedor2″ al contenedor con id=”contenido1″.

**attr()**

$("#contenido").attr("data");

$("#contenido").attr("title");

Con la función attr() retorna el valor del atributo que le pases como parámetro de un elemento o contenedor.

**before()**

$("#contenido").before("<p>hola!</p>");

El método before() añade contenido html detrás de lo que haya dentro del contenedor con id=”contenido”.

**blur()**

$("input").blur(function(){

    ...

});

Al salirse de en un campo o input de un formulario ejecutaríamos el código de dentro de la función.

**change()**

$("select#languages").change(function(){

    ...

});

Al cambiar de opción en un selector de opciones ejecutaríamos el código de dentro de la función.

**click()**

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

    ...

});

Al hacer click en el evento o contenedor con id=”button” ejecutaríamos el código de dentro de la función.

**css()**

$("#contenido").css("background-color","yellow");

Con la función css() podemos cambiar estilos al contenedor con id=”contenido”. En el ejemplo estamos cambiando el color de fondo.

**dblclick()**

$("#button").dblclick(function(){

    ...

});

Al hacer doble click en el evento o contenedor con id=”button” ejecutaríamos el código de dentro de la función.

**delay(time)**

$("#contenido").slideUp(300).delay(800).fadeIn(400);

Esta función permite retrasar la ejecución de las siguientes que estén en la cola. Es útil para añadir retrasos en una serie de animaciones. En el caso del código de ejemplo, primero hacemos slideup, hacemos una espera y seguidamente hacemos fadein.

**each()**

$("li").each(function(){

$(this).toggleClass("nombreclase");

});

Con la función each() lo que hacemos es iterar una serie de elementos. En el ejemplo estamos iterando una lista a la que a cada elemento le estamos cambiando el estilo o css utlizando la función toggleClass().

**empty()**

$("#contenido").empty();

Con empty() lo que conseguimos es vaciar el contenido de un contenedor antes de volverlo a rellenar.

**fadeIn()**

$("#contenido").fadeIn();

El método fadeIn() hace que el elemento que lo recibe aparezca en la página a través del cambio de su opacidad, haciendo una transición suavizada que acaba con el valor de opacity 1. Este método sólo podremos observarlo si el elemento sobre el que lo invocamos era total o parcialmente transparente, porque si era opaco al hacer un fadeIn() no se advertirá ningún cambio de opacidad.

**fadeOut()**

$("#contenido").fadeOut();

Este método hace que el elemento que lo recibe desaparezca de la página a través del cambio de su opacidad, haciendo una transición suavizada que acaba con el valor de opacity cero.

**fadeTo()**

$("#contenido").fadeTo("slow", 0.33);

El método fadeTo() es bastante más versátil, como ya se había adelantado. Para hacer un ejemplo interesante con este método tenemos que ver cómo se le pueden pasar distintos valores de opacidad y para ello hemos creado un campo select con distintos valores.

**focus()**

$("input").focus(function(){

    ...

});

Al situarte en un campo o input de un formulario ejecutaríamos el código de dentro de la función.

**focusout()**

$("input").focusout(function(){

    ...

});

Al salirse de en un campo o input de un formulario ejecutaríamos el código de dentro de la función.

**hasClass()**

$("#contenido").hasClass("nombreclase");

Con la función removeClass() comprobamos si el contenedor con id=”contenido” tiene la clase “nombreclase”. Devuelve True si la tiene.

**height()**

$("#contenido").height("200px");

Con la función height() modificamos el alto del contenedor con id=”contenido”.

**hide()**

$("#contenido").hide();

Con la función hide() lo que hacemos es ocultar el contenido del contenedor con id=”contenido”.

**hover()**

$('#element').hover(function(){

...

});

Con la función hover() lo que hacemos es ejecutar un código tras pasar el ratón por encima del contenedor con id=”element”.

**html()**

$("#contenido").html("<p>hola!</p>");

El método html() reemplaza el contenido html que haya dentro del contenedor con id=”contenido” por el que le pasas como parámetro.

**load()**

$("#contenido").load("archivo.php");

Función muy útil para cargar contenidos sin necesidad de recargar la página. En el caso del ejemplo, lo que haríamos es cargar el contenido html del “archivo.php” en el contenedor con id=”contenido”.

**keyup()**

$("input").keyup(function () {

      var value = $(this).val();

      $("p").text(value);

    }).keyup();

Con keyup() capturamos los eventos del teclado. En el ejemplo lo que se escriba en el input se va escribiendo también en el párrafo.

**mouseleave()**

$("#contenido").mouseleave(function(){

    ...

});

Al sacar el ratón de encima del contenedor o elemento con id=”contenido” ejecutaríamos el código de dentro de la función.

**mouseout()**

$("#contenido").mouseout(function(){

    ...

});

Al sacar el ratón de encima del contenedor o elemento con id=”contenido” ejecutaríamos el código de dentro de la función.

**mouseover()**

$("#contenido").mouseover(function(){

    ...

});

Al pasar el ratón por encima del contenedor o elemento con id=”contenido” ejecutaríamos el código de dentro de la función.

**nextAll()**

$("#contenido").nextAll().remove();

El método nextAll() selecciona todos los contenedores siguientes al que tiene id=”contenido”. En el caso del ejemplo, se eliminarán todos los contenedores siguientes a “#contenido”.

**next()**

$("#contenido").next().remove();

El método next() selecciona el contenedor siguiente al que tiene id=”contenido”. En el caso del ejemplo, se eliminará el contenedor siguiente a “#contenido”.

**parent()**

$("#contenido").parent();

El método parent() retorna un objeto que contiene información del contenedor padre del contenedor con id=”contenido”.

**position()**

$("#contenido").position().top;

$("#contenido").position().left;

La función position() permite obtener la posición de un elemento en relación a su “padre”.

**ready()**

$(document).ready(function(){

//Aqui tu codigo

});

Ready es un método propio de jQuery, que revisa si el DOM está listo para usarse. Es más util que el window.onload, pues este debe esperar a que todos los elementos de la pagina esten cargados (como scripts e imagenes) paa ejecutar. El “ready”, en cambio, espera solo a la estructura.

**removeClass()**

$("#contenido").removeClass("nombreclase");

Con la función removeClass() lo que hacemos quitar una clase o estilo CSS al contenedor con id=”contenido”. En e este caso le estamos quitando los estilos de la clase “nombreclase”.

**slideDown()**

$("#contenido").slideDown();

Con la función slideDown() lo que hacemos es mostrar hacia abajo el contenido del contenedor con id=”contenido” que inicialmente o no podría estar oculto.

**slideToggle()**

$("#contenido").slideToggle();

Con la función slideToggle() lo que hacemos es mostrar y ocultar el contenido del contenedor con id=”contenido” que inicialmente o no podría estar oculto.

**slideUp()**

$("#contenido").slideUp();

Con la función slideUp() lo que hacemos es mostrar hacia arriba el contenido del contenedor con id=”contenido” que inicialmente o no podría estar oculto.

**show()**

$("#contenido").show();

Con la función show() lo que hacemos es mostrar el contenido del contenedor con id=”contenido” que inicialmente o no podría estar oculto.

**size()**

$("li").size();

La función size() en este caso retorna el número de elementos que hay en la lista.

**submit()**

$("#formulario").submit(function() {

      if ($("#code").val() == "correct") {

        $.ajax({

            type: "POST",

            url: "archivo.php",

            data: dataString,

            success: function() {

                $("#message").empty();

                $("#message").append('<div>Hemos recibido su codigo correctamente</div>');

                $(this).remove();

            }

        });

      }

      $("#message").empty();

      $("#message").append('<div>Codigo no valido!</div>');

      return false;

});

La función submit() se suele utilizar para obtener la información de formularios. En el código anterior vemos un ejemplo de lo que se podría hacer. El formulario debe tener id=”formulario” y método POST.

**remove()**

$("#contenido").remove();

Con la función remove() lo que hacemos es quitar de la vista el contenido del contenedor con id=”contenido”.

**removeAttr()**

$("#image").removeAttr("width");

Con la función removeAttr() lo que hacemos es quitar un atributo del elemento con id=”image”.

**text()**

$("#contenido").text();

Con la función text() obtiene el contenido textual del elemento o contenedor con id=”contenido”.

**toggle()**

$("#contenido").toggle();

Con la función toggle() mostramos y ocultamos con efecto el contenedor con id=”contenido”.

**toggleClass()**

$("#contenido").toggleClass("nombreclase");

Con la función toggleClass() podemos añadir y quitar la clase “nombreclase” al contenedor con id=”contenido”.

**val()**

$("input").val();

$('select.foo option:selected').val(); // get the value from a dropdown select

$('select.foo').val(); // get the value from a dropdown select even easier

$('input:checkbox:checked').val(); // get the value from a checked checkbox

$('input:radio[name=bar]:checked').val(); // get the value from a set of radio buttons

Con la función val() obtenemos el valor de los inputs, selects, textareas de nuestros formularios.

**width()**

$("#contenido").width("200px");

Con la función width() modificamos el ancho del contenedor con id=”contenido”.

FUENTES DE INFORMACIÓN

<http://www.jose-aguilar.com/blog/funciones-utiles-para-utilizar-en-jquery/>