
jQuery
write less, do more.

Características

- Facilita la manipulación de los elementos del DOM (Document Object Model)
- Permite moverse hacia arriba, hacia abajo y hacia los lados en el árbol de elementos del DOM
- Facilita la manipulación de eventos
- Permite crear animaciones básicas
- Facilita el manejo de peticiones ajax

Selectores Básicos

- selector universal : $("*")
- selector por id : $("#about_us") $("#wrapper")
- selector por clase : $(".white_border")
- selector por etiqueta : $("article") $("h2")
- selector múltiple $("#about_us, h3")

Selectores Por Atributo

- El valor del atributo contenga la cadena : **[name*="value"]**

  $("a[href*='http']").css("color", "red")

- El valor del atributo sea igual a la cadena : **[name="value"]**
- El valor del atributo no sea igual a la cadena : **[name!="value"]**

- El valor del atributo empiece con : **[name^="value"]**
- El valor del atributo termine con : **[name$="value"]**

- Los elementos que contengan el atributo: **[name]**

- Selector con múltiples atributos : **[name="value"][name2="value2"]** en este caso se seleccionan los elementos que cumplan con todos los parámetros dados

Filtros Básicos

:eq() : selecciona dentro de los elementos que coinciden con la búsqueda sólo el elemento ubicado en el índice especificado .

$("article:eq(1)").css("background", "pink")

:lt() : selecciona dentro de los elementos que coinciden con la búsqueda todos los elementos que están en un índice menor que el especificado .

:gt() : selecciona dentro de los elementos que coinciden con la búsqueda todos los elementos que están en un índice mayor que el especificado .

:even : selecciona los elementos que están en índices pares, empezando con el cero

$("a:even").css("background", "pink")

:odd : selecciona los elementos que están el índices impares.

Filtros Básicos

:first : selecciona el primer elemento dentro conjunto de elementos seleccionados
:last selecciona el último elemento dentro conjunto de elementos seleccionados

:focus : selecciona el elemento que tiene el foco

:header : slecciona todos los elementos que son encabezados como el h1, h2, h3, ...
$(":header").css("background-color", "pink")

:not() : selecciona los elementos que no coinciden con el selector proporcionado

:contains() : selecciona todos los elementos que contienen el texto especificado.
$("a:contains('2014-11-07')")

Selectores De Elementos De Formularios

:input : selecciona todos los elementos tipo input, tipo textarea, tipo select y tipo button.

:radio : selecciona todos los elementos tipo 'radio'.

:button : selecciona todos los botones y elementos tipo 'button'.

:checkbox : selecciona todos los elementos tipo 'checkbox'.

:password : selecciona todos los elementos tipo 'password'.

:file :selecciona todos los elementos tipo 'file'.

:image : selecciona todos los elementos tipo 'image'.

Selectores De Elementos De Formularios

:reset selecciona todos los elementos tipo 'reset'.

:submit : selecciona todos los elementos tipo submit

:text : selecciona todos los input tipo 'text'

:checked : selecciona todos los elementos que están chequeados o seleccionados. Funciona con checkboxes, radio buttons, y elementos tipo 'select'

:selected : selecciona todos los elementos que están seleccionados, pero sóloa plica para los elementos tipo 'select'

:disabled : selecciona todos los elementos que están deshabilitados

:enabled : selecciona todos los elementos que están habilitados

Selectores de Jerarquía

- Selector de descendientes : $("#wrapper div")
- Selector de hijos : $("#wrapper > div")
- Selector de hermanos : $("aside li:eq(7) ~ li")
- Selector de elemento adyacente : $("aside li:eq(7) + li")

Selectores De Visibilidad

:hidden : selecciona todos los elementos ocultos

:visible : selecciona todos los elementos visibles

Moverse a través del DOM: Padres

.parent() : selecciona el padre directo $("audio").parent()

.parents() : selecciona todos los padres de un elemento. Puede recibir un selector para filtrar el resultado. $("audio").parents() $("audio").parents("div")

.parentsUntil() : selecciona todos los padres de un elemento hasta el el selector indicado, pero sin incluirlo.

$("audio").parentsUntil("#content").css('border', '3px solid pink')

.closest() selecciona el padre más cercano que coincida con el selector proporcionado.

Moverse a través del DOM: Hijos

.children() : selecciona los hijos directos del elemento seleccionado

```
$("#about_us").children("h6").css('font-size','1.2em')
```

.find() : selecciona todos los descendientes del elemento aplicando el filtro dado

```
$("#about_us").find('a').css('color','green')
```

Moverse a través del DOM: Hermanos

.next() : selecciona el elemento que está después del elemento seleccionado

.prev() : selecciona el elemento que está antes del elemento seleccionado

.nextAll() : selecciona todos los hermanos (osea los que tienen el mismo padre) que están después del elemento seleccionado

.prevAll() : selecciona todos los hermanos (osea los que tienen el mismo padre) que están antes del elemento seleccionado

.siblings() : selecciona todos los hermanos del elemento seleccionado en todas las direcciones

Encadenamiento de Métodos (Chaining)

La mayoría de los métodos de jQuery retornan un objeto jQuery el cual podemos usar para invocar otro método.

```
$("aside").find("a").eq(2).html( "Esta es una nuevo artículo" ).css("font-size", "1.5em");
```

Mover, Insertar, Copiar, y Eliminar Elementos

1. Ubicar el elemento seleccionado teniendo como referencia otro elemento que se pasa como parámetro

2. Ubicar un elemento que se pasa como parámetro teniendo como referencia el elemento seleccionado.

Primer enfoque : mover el elemento seleccionado

.insertAfter() : inserta el elemento seleccionado después (por fuera) del elemento que se pasa como parámetro

.insertBefore() : inserta el elemento seleccionado antes (por fuera) del elemento que se pase como parámetro

.appendTo() : agrega el elemento seleccionado al final del elemento que se pasa como parámetro (adentro)

.prependTo() : agrega el elemento seleccionado al principio del elemento que se pasa como parámetro (adentro)

Segundo enfoque : se manipula el parámetro

.after() : inserta el elemento que se pasa como parámetro después del elemento seleccionado (por fuera de él).

.before() inserta el elemento que se pasa como parámetro antes del elemento seleccionado (por fuera de él).

.append(): agrega al final (pero dentro) del elemento seleccionado el elemento que se pasa como parámetro

.prepend() agrega al final (pero dentro) del elemento seleccionado el elemento que se pasa como parámetro

Eliminar Elementos

- **.detach()** y **.remove()** eliminan elementos de la página. La diferencia entre las dos es que **.detach()** conserva los datos y eventos asociados al elemento y **.remove()** lo elimina permanentemente.
- **.empty()** deja el elemento seleccionado en la página pero elimina su contenido.

Copiar elementos

.clone() crea una copia del elemento seleccionado.

```
$("#older_posts li")
.first().clone().appendTo("#older_posts");
```

Eventos de elementos de formularios

.focus() : ocurre cuando un elemento obtiene el foco.

.blur() : ocurre cuando un elemento pierde el foco.

.change() : ocurre cuando el valor de un elemento cambia. Aplica sólo para elementos tipo input, textarea y select.

.select() : ocurre cuando el usuario hace una selección de texto. Sólo aplica para inputs tipo text y para textareas.

.submit() : este evento ocurre cuando el usuario envía un formulario, es decir, cuando hace click a un input tipo submit o un botón tipo submit.

Eventos de Teclado

.keydown() : ocurre cuando el usuario presiona una tecla.

.keyup() : ocurre cuando el usuario suelta una tecla.

.keypress() : es similar a keydown, excepto que las teclas que no se imprimen como shift, esc, ctrl, suprimir, flechas, etc, activan el evento keydown pero no el keypressed.

El evento keypressed responde ante la entrada de texto en cambio keydown responde ante cualquier tecla.

Código inicial

```
$( document ).ready( function() { … } );

$( function() { … } );

$( document ).ready( inicializar );
function inicializar() { … }

$( window ).load( function() { ... } );
```

Eventos de Mouse

.click()

.dblclick()

.mouseenter() : ocurre cuando el puntero del mouse entra en un elemento

.mouseleave() : ocurre cuando el puntero del mouse sale de un elemento

.hover() : maneja los eventos .mouseenter() y .mouseleave()

.mousedown() : ocurre cuando el puntero del mouse está sobre un elemento y se presiona el botón

.mouseup() : ocurre cuando el puntero del mouse está sobre un elemento y se suelta el botón

.mouseover() : tiene el mismo funcionamiento de evento .mouseenter() excepto en el manejo de la propagación del evento. (bubbling)

.mouseout() : tiene el mismo funcionamiento de evento mouseleave excepto en el manejo de la propagación del event. (bubbling)

.mousemove() : este evento se activa cuando el mouse se mueve dentro del elemento.

Bubbling

En javascript los eventos se propagan hacia arriba en el árbol de elementos del DOM, es decir, se van disparando sobre todos los ancestros del elemento original hasta llegar al body, al html y al document root.

e.stopPropagation() : detiene la propagación del evento

Propiedades del objeto del evento

- pageX, pageY : contienen la posición del mouse al momento de ocurrir el evento.
- type : el tipo de evento, por ejemplo "click"
- which : el botón del mouse o la tecla del teclado que fue presionada
- data : datos adicionales que se hayan pasado al evento cuando fué asociado al elemento
- target : el elemento del DOM element que inició el evento

Métodos .on() y .off()

$("h1").click(function(){})

es equivalente a la expresión

$("h1").on("click", function(){})

.off() : se utiliza para dejar de escuchar un evento en un elemento del DOM.

Por ejemplo, si se quiere eliminar el manejador del evento mouseover() sobre el footer de la página, se utliza la instrucción $("footer").off("mouseover");

Efectos

.show() : muestra los elementos seleccionados.

.hide() : oculta los elementos seleccionados.

.toggle() : muestra los elementos si están ocultos o los oculta si están visibles.

.slideDown() : muestra los elementos seleccionados por medio de una animación de su altura.

.slideUp() : oculta los elementos seleccionados por medio de una animación de su altura.

.slideToggle() : muestra los elementos si están ocultos o los oculta si están visibles, animando su altura.

Efectos

.fadeIn() : muestra los elementos seleccionados por medio de la animación de su transparencia.

.fadeOut() : oculta los elementos seleccionados por medio de la animación de su transparencia.

.fadeToggle() : muestra los elementos si están ocultos o los oculta si están visibles por medio de la animación de su transparencia.

.fadeTo() : ajusta la opacidad los elementos seleccionados.

Parámetros:

- duración de la animación en ms

- valor entre 0 y 1 que indica la opacidad a la que se quiere llegar

- función que se ejecuta al terminar la animación (opional)

Iterando sobre colecciones de JQuery

Algunos métodos que no realizan una iteración implícita sobre los elementos seleccionados son:

.css() (getter)

.height() (getter)

.html() (getter)

.val() (getter)

.width() (getter)

Iterando sobre colecciones de JQuery

.each() recibe como parámetro una función que se ejecuta sobre todos los elementos que hacen parte de la selección.

```
$(".article_title").each(function(){
    $(this).css("background-color", "blue");
});
```

AJAX

AJAX nos permite hacer peticiones al servidor y procesar la respuesta para actualizar el contenido de la página sin tener que recargarla

```
$.ajax(url, {
    data: { id: 123 }, // los datos que se van a enviar
    type: "GET", // tipo de petición: POST, GET, PUT, DELETE
    dataType : "json", // text, xml, json, jsonp, script, o html
    success: function( response ) { ... },
    error: function( xhr, status, errorThrown ) { ... },
    complete: function( xhr, status ) { ...  }
});
```

Funciones AJAX de alto nivel

$.get(url [, data] [, success] [, dataType]) : obtiene datos del servidor utilizando una petición GET. Debe usarse para hacer peticiones que no hacen cambios en el servidor

$.getScript(url [, success]) : carga un archivo JavaScript desde el servidor usando una petición GET y luego lo ejecuta.

$.getJSON(url [, data] [, success]) : carga información codificada en formato JSON desde el servidor.

$.post(url [, data] [, success] [, dataType]) : obtiene datos del servidor utilizando una petición POST. Las peticiones POST deben usarse para operaciones que hacen cambios en el servidor

Gracias