



## Referencia Básica JQuery 1.2

### Selectores

#### Básicos

<b>.clase</b>	Elementos con atributo class="clase"
<b>#id</b>	Elementos con id="id"
<b>xxx</b>	Elementos con nombre xxx (Ej: \$("P") = todos los párrafos)
<b>*</b>	Todos los elementos

#### Jerárquicos

<b>el1 el2</b>	Elementos el2 descendientes de elementos el1
<b>p &gt; h</b>	Elementos h que sean hijos de elementos p
<b>ant+sig</b>	Elementos sig que estén después de elementos ant
<b>ant~sig</b>	Elementos sig que estén después de elementos ant y sean hermanos del mismo

### Filtros

(Se aplican opcionalmente después del selector)

**Atributos** Los filtros seleccionan elementos tales que el atributo *atrib...*

<b>[atrib]</b>	... está presente	<b>[atrib=val]</b>	... es igual a val
<b>[atrib!=val]</b>	... es != de val.	<b>[atrib^=val]</b>	... comienza por val
<b>[atrib*=val]</b>	... contiene a val	<b>[atrib\$=val]</b>	... termina en val

**Posicionales** Seleccionan elementos cuya posición sea...

<b>:first</b>	... la primera (0)	<b>:last</b>	... la última
<b>:even</b>	... par (0,2,4,...)	<b>:odd</b>	... impar
<b>:eq(n)</b>	... igual a n		
<b>:lt(n)</b>	... menor que n	<b>:gt(n)</b>	... mayor que n

#### Contenido

<b>:contains(txt)</b>	Elementos que contienen el texto txt
<b>:empty</b>	Elementos vacíos

**Estado Actual** Seleccionan elementos que están...

<b>:animated</b>	... siendo animados
<b>:visible</b>	... visibles
<b>:hidden</b>	... ocultos

**Jerárquicos** Seleccionan elementos que son... (respecto a su padre)

<b>:parent</b>	... padres	<b>:only-child</b>	... hijos únicos
<b>:first-child</b>	... el 1er hijo	<b>:last-child</b>	... el último hijo
<b>:nth-child(n)</b>	Elementos que son el <i>n</i> -ésimo hijo de su padre. En lugar de <i>n</i> , se puede indicar la cadena "odd" o "even", que dejaría los hijos en posiciones impares o pares, respectivamente		

#### Campos de Formulario - Estado

<b>:enabled</b>	Habilitados	<b>:disabled</b>	Deshabilitados
<b>:checked</b>	Marcados	<b>:selected</b>	Seleccionados

#### Campos de Formulario - Tipo

<b>:input</b>	<b>:text</b>	<b>:checkbox</b>	<b>:radio</b>
<b>:hidden</b>	<b>:submit</b>	<b>:reset</b>	<b>:image</b>
<b>:file</b>	<b>:password</b>		

### Eventos

evt es cualquier evento JavaScript sin el on delante (Ej: click, mouseover, etc.)

<b>evt(fn)</b>	Asigna la función fn al evento evt. La función recibe como parámetro el objeto de evento. En ella, this=origen del evento
<b>bind(evt,datos,fn)</b>	Vincula la función fn al evento xxx. Los datos son opcionales y se pasan como event.data
<b>unbind(evt,fn)</b>	Desvincula la función fn del evento xxx
<b>hover(fn1,fn2)</b>	Llama a fn1 al entrar y fn2 al salir el ratón
<b>trigger(evt,datos)</b>	Simula la ocurrencia del evento evt

### Efectos

En todos los efectos, la velocidad puede ser "slow", "fast", "normal" o la duración del efecto en ms. Los elementos deben ocultarse con display:none y no con visibility:hidden. En todas las rutinas fn representa la función a llamar cuando el efecto ha concluido

<b>show()</b>	<b>show(velocidad,fn)</b>	Muestra el elemento
<b>hide()</b>	<b>hide(velocidad,fn)</b>	Oculto el elemento
<b>toggle()</b>	<b>toggle(velocidad,fn)</b>	Invierte la visibilidad
<b>fadeIn(velocidad,fn)</b>		Aparición mediante fundido
<b>fadeOut(velocidad,fn)</b>		Ocultación mediante fundido
<b>slideDown(velocidad,fn)</b>		Aparición mediante efecto persiana
<b>slideUp(velocidad,fn)</b>		Desaparición mediante efecto persiana
<b>slideToggle(velocidad,fn)</b>		Invierte visibilidad mediante efecto persiana
<b>animate(param,duracion,algoritmo,fn)</b>		Animación desde el valor actual de los estilos hasta el valor final indicado en param. Sólo para estilos numéricos. Algoritmo puede ser "linear" o "swing". Ej: <code>animate({height:"20px",width:"30px"}, 2000,"swing", ... )</code>
<b>stop()</b>		Detiene las animaciones en curso

### Peticiones AJAX

<b>\$.ajax(param)</b>	Lanza una petición AJAX con los parámetros indicados. Devuelve el objeto XMLHttpRequest creado. Los parámetros pueden ser:
<b>\$.ajax({</b>	
<b>async:</b>	¿Petición asíncrona? (por defecto : true ):
<b>beforeSend:</b>	Función a invocar antes de la llamada. Recibe como parámetro el objeto XMLHttpRequest
<b>complete:</b>	Función llamada al completar la petición
<b>error:</b>	Función llamada en caso de error. Parámetros: XMLHttpRequest, mensaje de Error, Excepción
<b>success:</b>	Función llamada en caso de éxito. Parámetros: datos, mensaje de estado
<b>timeout:</b>	Tiempo de espera máximo en ms para la petición
<b>username:</b>	Nombre de usuario (para autenticación)
<b>password:</b>	Clave de usuario (para autenticación)
<b>url:</b>	URL a solicitar
<b>dataType:</b>	Tipo de datos: "xml", "html", "script", "json", "text", "jsonp"
<b>data:</b>	Datos a enviar al servidor {nombre:valor, nombre:valor}
<b>cache:</b>	¿Permitir el uso de la cache? (por defecto: true, salvo para dataType="script")
<b>contentType:</b>	Tipo de datos enviados. Por defecto "application/x-www-form-urlencoded"
<b>})</b>	

### Modificación HTML

<b>css(nom)</b>	Devuelve el valor de la propiedad nom del primer elemento
<b>css(nom,val)</b>	Establece la propiedad CSS nom=val
<b>css(param)</b>	Modifica varias propiedades. Ej: <code>css({height:"...",width:"..."})</code>
<b>html()</b>	Devuelve el HTML del primer elemento
<b>html(val)</b>	Establece el HTML del elemento a val.
<b>attr(nom)</b>	Devuelve el valor del atributo nom del primer elemento
<b>attr(nom,val)</b>	Establece los atributos nom a val
<b>removeAttr(nom)</b>	Elimina los atributos nom.

### Manipulación de la selección

<b>get(n)</b>	Devuelve el <i>n</i> -ésimo elemento DOM seleccionado
<b>add(selector)</b>	Añade nuevos elementos a la lista
<b>not(selector)</b>	Elimina de la lista los elementos que cumplen el selector

