



Selección y alcance

Los selectores en jQuery y cómo alcanzar atravesando desde unos elementos a otros

Resumen del API de jQuery

http://api.jquery.com/

Este es un resumen de referencia rápida del API (juego de funciones) existente en jQuery. Hay algunas funciones que pertenecen a varias clasificaciones, aunque en este esquema aparecen una única vez.



Clase 3: Selectores

http://api.jquery.com/category/selectors/

Seleccionar elementos de la página, usando toda la potencia de las CSS.

Dispones de selectores que están en los estándares CSS1, CSS2 y CSS3 para usar según tus necesidades.

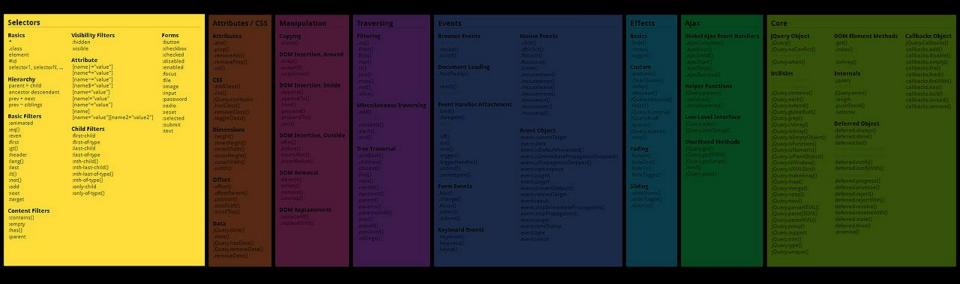
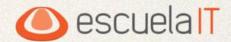


Imagen resumen de jQuery extraida de: http://oscarotero.com/jquery/



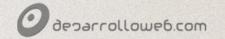




Exploremos la documentación

http://api.jquery.com/category/selectors/





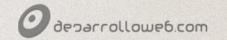
selectores



Básicamente tienes los selectores disponibles en CSS1, CSS2 y CSS3 para para seleccionar aquellos elementos que necesites.

Para aprenderlos y que no se te olviden rápidamente debes de practicar a menudo con ellos.

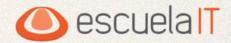


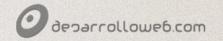


selectores básicos



- identificador "#micapa"
- Clase ".miclase"
- Etiqueta "body" "p"

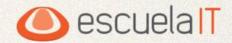


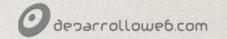


selectores básicos



- Varios selectores a la vez "sel1, sel2, sel3"
- Selector de contexto:
 ("selector", "contexto") ej: \$(".mc", "#x")





Selectores jerarquía



- Jerarquía a cualquier nivel ("p span")
- Hijos directos ("p > span")
- Hermanos ("div ~ p")
- Siguientes ("#p2 + p")

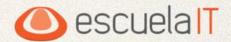




Filtros básicos



- Primer elemento ("p:first")
- Último elemento ("tr:last")
- Pares / Impares ("tr:even") / ("tr:odd")
- Selección por índice igual ("tr:eq(2)")
- Indice mayor que ("p:gt(1)")
- Indice menor que ("td:lt(2)")
- Encabezamientos ("article:header")
- No es selector ("p:not(#p2)")

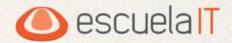


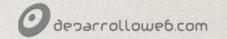


Filtros hijos



- Primer hijo ("p:first-child")
 - El selector :first-child puede seleccionar varios elementos, uno por cada padre. No hay que confundir esto con :first, que sólo selecciona un único elemento (ver la sección Los filtros jQuery básicos El primer elemento, en este capítulo).
- Último hijo ("tr:last-child")
- Enésimo hijo ("tr td:nth-child(2)")
 Al contrario de lo que sucede con muchos índices en jQuery, el índice empieza por 1, y no por 0.
- Hijos pares e impares ("table tr:nth-child(odd)")
- Único hijo ("p a:only-child")

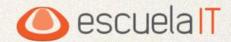


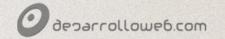


Filtros de contenido



- Tiene un texto dado ("p:contains('probando')")
- Contenido vacío ("td:empty")
- Tiene hijos (incluyendo nodos de texto) \$("td:parent")
- tiene otros elementos dados ("div:has(p)")

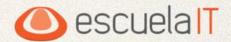




Filtros de visibilidad



- Elementos visibles ("p:visible")
- Elementos no visibles ("p:hidden")

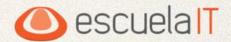


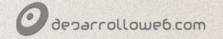


Filtros de atributo



- Tiene atributo ("p[id]")
- Tiene atributo con valor dado ("[align=center]")
- Atributo no tiene valor determinado ("a[title!='enlace interno']")
- Atributo empieza por un valor ("a[title^='enlace']")
- Atributo que termina por valor ("a[title\$='externo']")
- Atributo que contiene valor ("p[id*='p']")
- Aplicar múltiples filtros de atributo a la vez ("p[id][align=center]")





Selectores de formul.



:input Selecciona todos los elementos

:text Selecciona todos los input type text

:password Selecciona contraseña.

:radio Selecciona botones de radio.

:checkbox Selecciona casillas de selección.

:submit Selecciona submit.

:image Selecciona imagen.

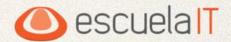
:reset Selecciona tipo reset.

:button Selecciona <button> y input button.

:file Selecciona elementos enviar archivo.

:hidden Selecciona los ocultos con hidden.

:focus Selecciona el elemento del foco.

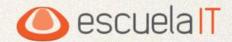




Filtros de formulario



- Todos los checked ("input:checked")
- Selecciona elemento seleccionado \$("#navegadores option:selected")





Caracteres especiales



Ponemos dos barras invertidas \$("#p\\.x")

Clase 3: Traversing

http://api.jquery.com/category/traversing/

Acceder desde unos elmentos a otros, atravesando el DOM a través de la navegación por el árbol de nodos mediante diversas operaciones.

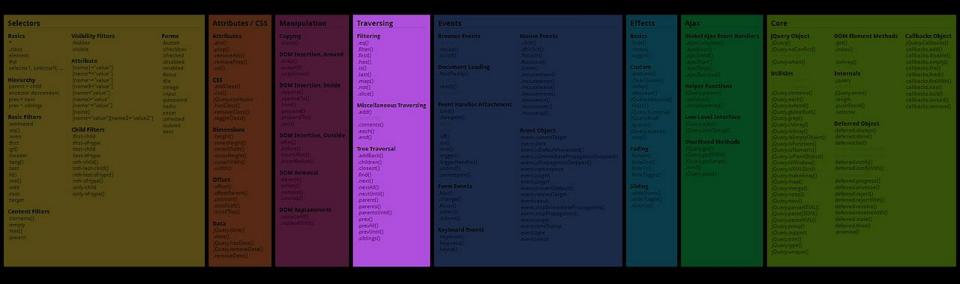
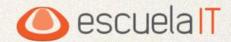
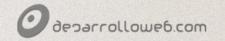


Imagen resumen de jQuery extraida de: http://oscarotero.com/jquery/

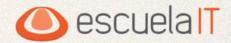






Exploremos la documentación

http://api.jquery.com/category/traversing/



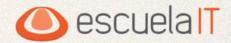


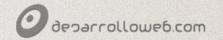
eq()



Reducir una colección para traerse un elemento único definido a través del índice (si es negativo, se cuenta el índice por la cola)

http://api.jquery.com/eq/



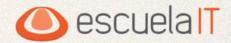


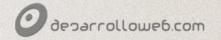
children()



Obtener los hijos que hay en los elementos seleccionados. Opcionalmente puedes filtrar por un selector esos hijos.

http://api.jquery.com/children/





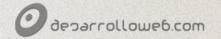
filter()



De una colección de elementos nos quedamos solo con unos pocos, aquellos que casen con el selector usado para filtrado.

http://api.jquery.com/filter/



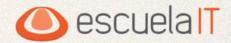


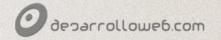
find()



Obtienes los descendientes de la colección que además casen con un selector.

http://api.jquery.com/find/





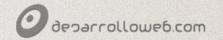
next()



Obtienes el siguiente elemento, hermano del seleccionado y opcionalmente puedes obligar a que concuerde con un selector.

http://api.jquery.com/next/



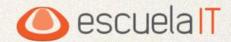


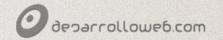
parent()



Obtienes el padre del elemento/s de la colección, puedes filtrar por un selector.

http://api.jquery.com/parent/



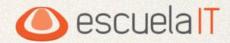


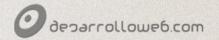
siblings()



Obtienes los hermanos, opcionalmente puedes filtrar por selector.

http://api.jquery.com/siblings/



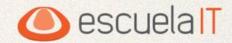


not()



Elimina elementos de una colección, le pasas el selector

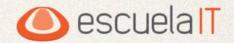
http://api.jquery.com/not/





Buenas prácticas

Algunas notas sobre selectores y buenas prácticas



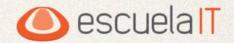


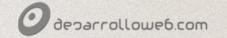
Cacheado de objetos



```
$("p.miparrafo").css(...);
$("p.miparrafo").show(...);
$("p.miparrafo").animate(...);

Mejor:
var p = $("p.miparrafo");
p.css(...);
p.show(...);
p.animate(...);
```





Encadenar métodos

```
Signature of the second of the
```

```
$("p.miparrafo").css(...);
$("p.miparrafo").show(...);
$("p.miparrafo").animate(...);
```

Mejor:

```
$("p.miparrafo").css().show().animate();
```

Alternativa:

```
$("p.miparrafo")
.css()
.show()
.animate();
```





Rendimiento de selectores

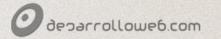
No todos los selectores son igual de rápidos.

Hay selectores inmediatos (id) y otros que obligas a jQuery a recorrrer todos los nodos (elementos de la página) para encontrar aquellos que concuerdan con un selector.

```
$("#miidentificador");
$(".miclase");
$("p");
```







Partir de un contexto

Puedes ayudar al motor de jQuery indicando el contexto desde donde buscar tus elementos. (También te ayudas a ti mismo)





También puedes filtrar

Las funciones traversing también te ayudan a optimizar.





Rendimiento

Test de rendimiento entre:

```
$("#x .mc") //css de toda la vida
$(#x).find(".mc") //buscando con traversing
$(".mc", "#x") selector de contexto
```

http://jsperf.com/finding-on-context-vs-combination-selector