

Selector Type	Usage Example	Description Of The Selector
*	<code>\$('*');</code>	Returns all the elements in the document (including head, body and all other elements)
#id	<code>\$('#id')</code>	Returns an element with the given ID.
.class	<code>\$('.class')</code> or <code>\$(a.class')</code>	Returns all elements in the document with the given class name (or where prefixed with a selector, it's matches using that class name)
element	<code>\$('div')</code>	Returns all elements in a document with the given tag name i.e. div, table, ul, li, a etc.
a, b, c. ... n	<code>\$(th, td, .class, #id')</code>	Returns all elements in the document that match the multiple selection patterns specified
parent child	<code>\$('li a')</code>	Returns all child elements that are a descendant of parent, ie. "a" elements that are a descendant of "li"
a > b	<code>\$(table > tr');</code>	Returns all 'b' elements which are a child element of a or in our example all tr elements in a table or tables.
a + b	<code>\$(li + a');</code>	Returns the "a" elements that are an immediate descendant of b, ie. "li" in our example.
a ~ b	<code>\$(p ~ ul');</code>	Returns the 'b' elements that are a sibling of 'a', ie. "ul" elements that are a sibling of "p"
:first	<code>\$(ul li:first');</code>	Returns the first element in a result set
:first-child	<code>\$(ul li:first-child');</code>	Returns the first child element of the parent element.
:last	<code>\$(ul li:last');</code>	Returns the last element in a result set
:last-child	<code>\$(ul li:last-child');</code>	Returns the last child element of the parent element.
:only-child	<code>\$(div p:only-child');</code>	Returns the only child of the parent element.
:not(a)	<code>\$(input:not(:checked));</code>	Returns all elements that are not "a" on in our example all input elements that are not checked
:has(a)	<code>\$(div:has(p));</code>	Returns all elements with a descendant that matches a in out example a "div" that contains a "p".
:odd	<code>\$(ul li:odd');</code>	Returns all odd elements in a result set (zero based)
:even	<code>\$(ul li:even');</code>	Returns all even elements in a result set (zero based)
:eq(n)	<code>\$(ul li:eq(n));</code>	Returns a numbered element identified by n (zero based)
:gt(n)	<code>\$(ul li:gt(n));</code>	Returns all elements in a result set greater than n (zero based)
:lt(n)	<code>\$(ul li:lt(n));</code>	Returns all elements in a result set less than n (zero based)
:nth-child(n)	<code>\$(ul li:nth-child(n));</code>	Returns the nth child in a result set (one based)
:nth-child(odd)	<code>\$(ul li:nth-child(odd));</code>	Returns all odd numbered elements in a result set (one based)
:nth-child(even)	<code>\$(ul li:nth-child(even));</code>	Returns all even numbered elements in a result set (one based)
:nth-child(formula)	<code>\$(ul li:nth-child(3n));</code>	Returns every nth child in a result set. In our example every third child (one based)
:header	<code>\$(header');</code>	Returns all heading elements e.g. h1, h2, etc.
:animated	<code>\$(ul:animated');</code>	Returns elements with an animation currently in progress
:contains(text)	<code>\$(div:contains(resig));</code>	Returns all elements containing the passed string (eg. 'resig')
:empty	<code>\$(div:empty');</code>	Returns all elements that contain no child nodes
:parent	<code>\$(li:parent');</code>	Returns all elements that a parent nodes to any other DOM element including text nodes.
:hidden	<code>\$(ul:hidden');</code>	Returns all hidden elements that are hidden with CSS or input fields of the type "hidden"
:visible	<code>\$(ul:visible');</code>	Returns all visible elements
[attribute]	<code>\$(div[href]);</code>	Returns all elements that contain the specified attribute - in our example any element with a "href" attribute
[attribute=value]	<code>\$(div[rel=ext]);</code>	Returns all elements where the specified attribute value is equal to the passed value. Eg. element with a "rel" attribute equal to "ext"
[attribute!=value]	<code>\$(div[rel!=ext]);</code>	Returns all elements where the specified attribute value is not equal to the passed value. Eg. element with a "rel" attrib not equal to "ext"
[attribute!=value]	<code>\$(div[class=gloss]);</code>	Returns all elements where the specified attribute value begins with the passed value. Eg. element whose "class" attribute value begins with "gloss"
[attribute\$=value]	<code>\$(div[id\$=-container]);</code>	Returns all elements where the passed attribute value ends with the passed value. Eg. any element whose "id" ends with "-container"
[attribute*=value]	<code>\$(div[class=bar]);</code>	Returns all elements where the passed attribute value contains the passed value. Eg. any element whose "class" contains the string "bar"
[attribute =value]	<code>\$(div[a alt=hello]);</code>	Returns matched elements that have the specified attribute with a value either equal to/starting with that string followed by a hyphen
[name=a][name2=b]	<code>\$(input[id][name\$='b']);</code>	Returns matched elements that match all of the attributes which have been specified eg. inputs with an id and a 'name' ending in b
:input	<code>\$(input');</code>	Returns only those input elements of the tag name input, select, text-area and button
:text	<code>\$(input:text');</code>	Returns only those input elements of the type "text"
:password	<code>\$(input:password');</code>	Returns only those input elements of the type "password"
:radio	<code>\$(input:radio');</code>	Returns only those input elements of the type "radio"
:checkbox	<code>\$(input:checkbox');</code>	Returns only those input elements of the type "checkbox"
:submit	<code>\$(input:submit');</code>	Returns only those input elements of the type "submit"
:image	<code>\$(input:image');</code>	Returns only those input elements of the type "image"
:reset	<code>\$(input:reset');</code>	Returns only those input elements of the type "reset"
:file	<code>\$(input:file');</code>	Returns only those input elements of the type "file"
:button	<code>\$(input:button');</code>	Returns only those input elements of the type "button"
:enabled	<code>\$(input:enabled');</code>	Returns all enabled input elements
:selected	<code>\$(input:selected');</code>	Returns the selected element in a select list.
:disabled	<code>\$(input:disabled');</code>	Returns disabled input elements
:checked	<code>\$(input:checked');</code>	Returns checked input elements of the type radio or checkbox.