

Source & Reference

<http://api.jquery.com/>

Selectors

<code>#id</code>	Select By Id
<code>element</code>	Select By Tag
<code>.class</code>	Select By CSS Class
<code>[attr='value']</code>	Select by attribute value
<code>:hidden</code>	must be hidden
<code>:visible</code>	must be visible
<code>:first-child</code>	First child in a sequence
<code>:last-child</code>	last Child in the sequence
<code>:nth-child(#)</code>	Selects all elements of the specified child
<code>:only-child</code>	Selects all elements that are the only child
<code>:contains('text')</code>	Selects all elements containing text
<code>:empty</code>	Selects all elements without children
<code>:has(selector)</code>	Selects all elements that contain at least one element that matches selector
<code>:not(selector)</code>	Selects all elements that do not match the given selector.
<code>:parent</code>	Selects all elements with children
<code>parent > child</code>	Selects all direct child elements specified by "child" of elements specified by "parent".
<code>'ancestor descendant'</code>	Selects all elements that are descendants of a given ancestor
<code>'previous + next'</code>	Selects all next elements matching "next" that are immediately preceded by a sibling "prev"
<code>:even</code>	Select all elements at an even index within the matched set. Zero based
<code>:odd</code>	Select all elements at an odd index within the matched set. Zero based
<code>:animated</code>	Select all elements that are in the progress of an animation at the time the selector is run.
<code>:first</code>	Select First matched element
<code>:last</code>	Select Last matched element

Selectors (cont)

<code>:eq(index)</code>	Select all elements at an index equal to <i>index</i> within the matched set. Zero based
<code>:gt(index)</code>	Select all elements at an index greater than <i>index</i> within the matched set. Zero based
<code>:lt(index)</code>	Select all elements at an index less than <i>index</i> within the matched set. Zero based
<code>:header</code>	Selects all elements that are headers, like h1 - h6

Attributes

<code>.attr(attributeName [,value])</code>	Get the first matched element attribute OR Set one or more attributes for the set of matched elements.
<code>.removeAttr(attributeName)</code>	Remove an attribute from each element in the set of matched elements.
<code>.prop(propertyName [,value])</code>	Get the first matched element OR Set one or more properties for the set of matched elements.
<code>.removeProp(propertyName)</code>	Remove a property for the set of matched elements.
<code>.val(value)</code>	Get the current value of the first matched element OR Set the value of each element in the set of matched elements.
<code>.html([htmlString])</code>	Get the HTML contents of the first element in the set of matched elements or set the HTML contents of every matched element.

Processing

<code>.each(function)</code>	Iterate over a jQuery object, executing a function for each matched element
------------------------------	---



By Sean J

cheatography.com/sean-j/

Published 18th May, 2017.

Last updated 18th May, 2017.

Page 1 of 8.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>

Processing (cont)

<code>.map()</code>	Pass each element in the current matched set through a function, producing a new jQuery object containing the return values.
<code>.noop()</code>	An empty function.
<code>\$.ready()</code>	Specify a function to execute when the DOM is fully loaded.
<code>\$.param()</code>	Create a serialized representation of an array, a plain object, or a jQuery object suitable for use in a URL query string or Ajax request
<code>.data([key] [, value])</code>	Retrieve all data OR retrieve first matched value or set all keyed values in a matched set

Miscellaneous

<code>.get([index])</code>	Retrieve the DOM elements matched by the jQuery object.
<code>.index([selector/element])</code>	Search for a given element from among the matched elements.
<code>.toArray()</code>	Retrieve all the elements contained in the jQuery set, as an array.

CSS - Effects

<code>.hide([duration], [easing], [complete])</code>	Display element
<code>.show([duration], [easing], [complete])</code>	Hide element
<code>.toggle([duration], [easing], [complete])</code>	Display or Hide element
<code>.fadeIn([duration], [easing], [complete])</code>	Display "fade in"
<code>.fadeOut([duration], [easing], [complete])</code>	Hide "fade out"
<code>.fadeTo(duration, opacity, [easing], [complete])</code>	Set Transparency
<code>.slideDown([duration], [easing], [complete])</code>	Display matched elements with a sliding motion
<code>.slideUp([duration], [easing], [complete])</code>	Hide matched elements with a sliding motion
<code>.slideToggle([duration], [easing], [complete])</code>	Display or hide matched elements with a sliding motion

CSS - Effects (cont)

<code>.animate(properties [, duration] [, easing] [, complete])</code>	Perform a custom animation of a set of CSS properties
<code>.finish([queue])</code>	Stop the currently-running animation, remove all queued animations, and complete all animations for the matched elements.
<code>.stop([clearQueue] [, jumpToEnd])</code>	Stop the currently-running animation on the matched elements.

CSS - General

<code>.addClass(classNames)</code>	Adds the specified class(es) to each matched element
<code>.css(propertyName [, value])</code>	Get the value of a computed style property for the first element in the set of matched elements or set one or more CSS properties for every matched element
<code>.hasClass()</code>	Determine whether any of the matched elements are assigned the given class.
<code>\$.escapeSelector()</code>	Escapes any character that has a special meaning in a CSS selector.
<code>.removeClass()</code>	Remove a single class, multiple classes, or all classes from each element in the set of matched elements.
<code>.toggleClass()</code>	Add or remove one or more classes from each element in the set of matched elements, depending on either the class's presence or the value of the state argument.

Events - Mouse

<code>.click([handler])</code>	Bind an event handler to the "click" JavaScript event, or trigger that event on an element.
--------------------------------	---

Events - Mouse (cont)

`.contextmenu([handler])` Bind an event handler to the "contextmenu" JavaScript event, or trigger that event on an element.

`.dblclick([handler])` Bind an event handler to the "dblclick" JavaScript event, or trigger that event on an element

`.hover([handler])` Bind one or two handlers to the matched elements, to be executed when the mouse pointer enters and leaves the elements.

`.mousedown([handler])` Bind an event handler to the "mousedown" JavaScript event, or trigger that event on an element.

`.mouseenter([handler])` Bind an event handler to be fired when the mouse enters an element, or trigger that handler on an element.

`.mouseleave([handler])` Bind an event handler to be fired when the mouse leaves an element, or trigger that handler on an element

`.mousemove([handler])` Bind an event handler to the "mousemove" JavaScript event, or trigger that event on an element

`.mouseout([handler])` Bind an event handler to the "mouseout" JavaScript event, or trigger that event on an element

`.mouseover([handler])` Bind an event handler to the "mouseover" JavaScript event, or trigger that event on an element

`.mouseup([handler])` Bind an event handler to the "mouseup" JavaScript event, or trigger that event on an element

Events - Keyboard

`.keydown(handler)` Bind an event handler to the "keydown" JavaScript event, or trigger that event on an element.

`.keypress(handler)` Bind an event handler to the "keypress" JavaScript event, or trigger that event on an element.

`.keyup(handler)` Bind an event handler to the "keyup" JavaScript event, or trigger that event on an element.

Traversing - Tree & Misc

`.children([selector])` Get the children of each element in the set of matched elements, optionally filtered by a selector.

`.closest([selector])` For each element in the set, get the first element that matches the selector by testing the element itself and traversing up through its ancestors in the DOM tree.

`.find([selector])` Get the descendants of each element in the current set of matched elements, filtered by a selector, jQuery object, or element.

`.next([selector])` Get the immediately following sibling of each element in the set of matched elements. If a selector is provided, it retrieves the next sibling only if it matches that selector.

`.nextAll([selector])` Get all following siblings of each element in the set of matched elements, optionally filtered by a selector.

`.nextUntil([selector], [filter])` Get all following siblings of each element up to but not including the element matched by the selector, DOM node, or jQuery object passed.

`.offsetParent()` Get the closest ancestor element that is positioned.

`.parent([selector])` Get the parent of each element in the current set of matched elements, optionally filtered by a selector.

`.parents([selector])` Get the ancestors of each element in the current set of matched elements, optionally filtered by a selector.

Traversing - Tree & Misc (cont)

<code>.parentsUntil([selector] [,filter])</code>	Get the ancestors of each element in the current set of matched elements, up to but not including the element matched by the selector, DOM node, or jQuery object.
<code>.prev([selector])</code>	Get the immediately preceding sibling of each element in the set of matched elements. If a selector is provided, it retrieves the previous sibling only if it matches that selector.
<code>.prevAll([selector])</code>	Get all preceding siblings of each element in the set of matched elements, optionally filtered by a selector.
<code>.prevUntil([selector] [,filter])</code>	Get all preceding siblings of each element up to but not including the element matched by the selector, DOM node, or jQuery object.
<code>.siblings([selector])</code>	Get the siblings of each element in the set of matched elements, optionally filtered by a selector.
<code>.add([selector elements html])</code>	Create a new jQuery object with elements added to the set of matched elements.
<code>.addBack([selector])</code>	Add the previous set of elements on the stack to the current set, optionally filtered by a selector.
<code>.contents()</code>	Get the children of each element in the set of matched elements, including text and comment nodes.
<code>.end()</code>	End the most recent filtering operation in the current chain and return the set of matched elements to its previous state.

Utilities

<code>\$.contains(container, contained)</code>	Check to see if a DOM element is a descendant of another DOM element.
<code>\$.each(array, callback)</code>	A generic iterator function, which can be used to seamlessly iterate over both objects and arrays.
<code>\$.extend(target [,object1] [,objectN])</code>	Merge the contents of two or more objects together into the first object.
<code>\$.fn.extend(obj)</code>	Merge the contents of an object onto the jQuery prototype to provide new jQuery instance methods.
<code>\$.globalEval(code)</code>	Execute some JavaScript code globally.
<code>\$.grep(array, function [,invert])</code>	Finds the elements of an array which satisfy a filter function. The original array is not affected.
<code>\$.inArray(value, array [,fromIndex])</code>	Search for a specified value within an array and return its index (or -1 if not found).
<code>\$.isArray(obj)</code>	Determine whether the argument is an array.
<code>\$.isEmptyObject(obj)</code>	Check to see if an object is empty (contains no enumerable properties).
<code>\$.isFunction(obj)</code>	Determine if the argument passed is a JavaScript function object.
<code>\$.isNumeric(value)</code>	Determines whether its argument represents a JavaScript number.
<code>\$.isPlainObject(obj)</code>	Check to see if an object is a plain object (created using <code>{}</code> or <code>new Object()</code>).
<code>\$.isWindow(obj)</code>	Determine whether the argument is a window.



Utilities (cont)

<code>\$.isXMLDoc(<i>node</i>)</code>	Check to see if a DOM node is within an XML document (or is an XML document).
<code>\$.makeArray(<i>obj</i>)</code>	Convert an array-like object into a true JavaScript array.
<code>\$.map(<i>array</i>, <i>callback</i>)</code>	Translate all items in an array or object to new array of items.
<code>\$.merge(<i>first</i>, <i>second</i>)</code>	Merge the contents of two arrays together into the first array.
<code>\$.noop()</code>	An empty function.
<code>\$.now()</code>	Return a number representing the current time.
<code>\$.parseHTML(<i>data</i> [<i>context</i>] [<i>keepScripts</i>])</code>	Parses a string into an array of DOM nodes.
<code>\$.parseJSON(<i>json</i>)</code>	Takes a well-formed JSON string and returns the resulting JavaScript value
<code>\$.parseXML(<i>data</i>)</code>	Parses a string into an XML document.
<code>\$.trim(<i>string</i>)</code>	Remove the whitespace from the beginning and end of a string.
<code>\$.type(<i>obj</i>)</code>	Determine the internal JavaScript [[Class]] of an object.
<code>\$.unique(<i>array</i>)</code>	Sorts an array of DOM elements, in place, with the duplicates removed. Note that this only works on arrays of DOM elements, not strings or numbers.
<code>\$.uniqueSort(<i>array</i>)</code>	Sorts an array of DOM elements, in place, with the duplicates removed.

Core

<code>jQuery()</code> <code>\$()</code>	Return a collection of matched elements either found in the DOM based on passed argument(s) or created by passing an HTML string.
<code>\$.holdReady(<i>hold</i>)</code>	Holds or releases the execution of jQuery's ready event.
<code>\$.noConflict(<i>removeAll</i>)</code>	Relinquish jQuery's control of the \$ variable.
<code>\$.ready</code>	A Promise-like object (or "thenable") that resolves when the document is ready.
<code>.ready(<i>handler</i>)</code>	Specify a function to execute when the DOM is fully loaded.
<code>\$.readyException(<i>error</i>)</code>	Handles errors thrown synchronously in functions wrapped in jQuery().
<code>\$.when(<i>deferreds</i>)</code>	Provides a way to execute callback functions based on zero or more Thenable objects, usually Deferred objects that represent asynchronous events.

Queue

<code>.clearQueue(<i>queueName</i>)</code>	Remove from the queue all items that have not yet been run.
<code>.dequeue(<i>queueName</i>)</code>	Execute the next function on the queue for the matched elements.
<code>.queue(<i>queueName</i>)</code> <code>[<i>newQueue</i> <i>callback</i>]</code>	Show or manipulate the queue of functions to be executed on the matched elements.

Ajax

<code>\$.ajax(<i>url</i> [<i>settings</i>])</code>	Perform an asynchronous HTTP (Ajax) request.
<code>.ajaxComplete(<i>handler</i>)</code>	Register a handler to be called when Ajax requests complete.
<code>.ajaxError(<i>handler</i>)</code>	Register a handler to be called when Ajax requests complete with an error.

Ajax (cont)

<code>.ajaxSend(handler)</code>	Attach a function to be executed before an Ajax request is sent
<code>.ajaxStart(handler)</code>	Register a handler to be called when the first Ajax request begins.
<code>.ajaxStop(handler)</code>	Register a handler to be called when all Ajax requests have completed
<code>.ajaxSuccess(handler)</code>	Attach a function to be executed whenever an Ajax request completes successfully.
<code>\$.ajaxPrefilter([dataTypes], handler)</code>	Handle custom Ajax options or modify existing options before each request is sent and before they are processed by <code>\$.ajax()</code> .
<code>\$.ajaxSetup(options)</code>	Set default values for future Ajax requests. Its use is not recommended
<code>\$.ajaxTransport(dataType, handler)</code>	Creates an object that handles the actual transmission of Ajax data.
<code>\$.get(url [,data] [,success] [,dataType])</code>	Load data from the server using a HTTP GET request
<code>\$.getJSON(url [,data] [,success])</code>	Load JSON-encoded data from the server using a GET HTTP request.
<code>\$.getScript(url [,success])</code>	Load a JavaScript file from the server using a GET HTTP request, then execute it
<code>\$.param(obj)</code>	Create a serialized representation of an array, a plain object, or a jQuery object suitable for use in a URL query string or Ajax request.
<code>\$.post(url [,data] [,success] [,dataType])</code>	Load data from the server using a HTTP POST request.
<code>.load(url [,data] [,complete])</code>	Load data from the server and place the returned HTML into the matched element.

Ajax (cont)

<code>.serialize()</code>	Encode a set of form elements as a string for submission.
<code>.serializeArray()</code>	Encode a set of form elements as an array of names and values.

CSS - Dimensions

<code>.offset()</code>	Get the current coordinates of the first element, or set the coordinates of every element, in the set of matched elements, relative to the document.
<code>.position()</code>	Get the current coordinates of the first element in the set of matched elements, relative to the offset parent.
<code>.scrollLeft()</code>	Get the current horizontal position of the scroll bar for the first element in the set of matched elements or set the horizontal position of the scroll bar for every matched element.
<code>.scrollTop()</code>	Get the current vertical position of the scroll bar for the first element in the set of matched elements or set the vertical position of the scroll bar for every matched element.
<code>.height()</code>	Get the current computed height for the first element in the set of matched elements or set the height of every matched element.
<code>.innerHeight()</code>	Get the current computed inner height (including padding but not border) for the first element in the set of matched elements or set the inner height of every matched element.
<code>.outerHeight()</code>	Get the current computed outer height (including padding, border, and optionally margin) for the first element in the set of matched elements or set the outer height of every matched element.
<code>.width()</code>	Get the current computed width for the first element in the set of matched elements or set the width of every matched element.



CSS - Dimensions (cont)

<code>.inner-Width()</code>	Get the current computed inner width (including padding but not border) for the first element in the set of matched elements or set the inner width of every matched element.
<code>.outer-Width()</code>	Get the current computed outer width (including padding, border, and optionally margin) for the first element in the set of matched elements or set the outer width of every matched element.

Events - Browser

<code>.resize([handler])</code>	Bind an event handler to the "resize" JavaScript event, or trigger that event on an element.
<code>.scroll([handler])</code>	Bind an event handler to the "scroll" JavaScript event, or trigger that event on an element.

Events - Handler Attachment

<code>jQuery.proxy(function, context [,additionalArguments])</code>	Takes a function and returns a new one that will always have a particular context.
<code>.off(events [,selector] [,handler])</code>	Remove an event handler.
<code>.on(events [,selector] [,data] [,handler])</code>	Attach an event handler function for one or more events to the selected elements.
<code>.one(events [,data], handler)</code>	Attach a handler to an event for the elements. The handler is executed at most once per element per event type.
<code>.trigger(eventType [,extraParameters])</code>	Execute all handlers and behaviors attached to the matched elements for the given event type.
<code>.triggerHandler(eventType [,extraParameters])</code>	Execute all handlers attached to an element for an event

*Most event methods have additional overloads

Events - Forms

<code>.blur(handler)</code>	Bind an event handler to the "blur" JavaScript event, or trigger that event on an element.
<code>.change(handler)</code>	Bind an event handler to the "change" JavaScript event, or trigger that event on an element.
<code>.focus(handler)</code>	Bind an event handler to the "focus" JavaScript event, or trigger that event on an element.
<code>.focusin(handler)</code>	Bind an event handler to the "focusin" event.
<code>.focusout(handler)</code>	Bind an event handler to the "focusout" JavaScript event.
<code>.select(handler)</code>	Bind an event handler to the "select" JavaScript event, or trigger that event on an element.
<code>.submit(handler)</code>	Bind an event handler to the "submit" JavaScript event, or trigger that event on an element.

Traversing - Filtering

<code>.eq(index)</code>	Reduce the set of matched elements to the one at the specified index.
<code>.filter(selector)</code>	Reduce the set of matched elements to those that match the selector or pass the function's test.
<code>.first()</code>	Reduce the set of matched elements to the first in the set.
<code>.has(selector)</code>	Reduce the set of matched elements to those that have a descendant that matches the selector or DOM element.
<code>.is(selector/function/selectorion/elements)</code>	Check the current matched set of elements against a selector, element, or jQuery object and return true if at least one of these elements matches the given arguments.



Traversing - Filtering (cont)

<code>.last()</code>	Reduce the set of matched elements to the final one in the set.
<code>.map(callback)</code>	Pass each element in the current matched set through a function, producing a new jQuery object containing the return values.
<code>.not(selector)</code>	Remove elements from the set of matched elements.
<code>.slice(start [,end])</code>	Reduce the set of matched elements to a subset specified by a range of indices.

Manipulation

<code>.after(content/function)</code>	Insert content, specified by the parameter, after each element in the set of matched elements.
<code>.append(content/function)</code>	Insert content, specified by the parameter, to the end of each element in the set of matched elements.
<code>.appendTo(target)</code>	Insert every element in the set of matched elements to the end of the target.
<code>.before(content/function)</code>	Insert content, specified by the parameter, before each element in the set of matched elements.
<code>.clone([withDataAndEvents])</code>	Create a deep copy of the set of matched elements.
<code>.detach(selector)</code>	Remove the set of matched elements from the DOM.
<code>.empty()</code>	Remove all child nodes of the set of matched elements from the DOM.
<code>.insertAfter(target)</code>	Insert every element in the set of matched elements after the target
<code>.insertBefore(target)</code>	Insert every element in the set of matched elements before the target.

Manipulation (cont)

<code>.prepend(content/function)</code>	Insert content, specified by the parameter, to the beginning of each element in the set of matched elements.
<code>.prependTo(target)</code>	Insert every element in the set of matched elements to the beginning of the target.
<code>.remove(selector)</code>	Remove the set of matched elements from the DOM.
<code>.replaceAll(target)</code>	Replace each target element with the set of matched elements.
<code>.replaceWith(newContent/function)</code>	Replace each element in the set of matched elements with the provided new content and return the set of elements that was removed.
<code>.text()</code>	Get the combined text contents of each element in the set of matched elements, including their descendants, or set the text contents of the matched elements.
<code>.unwrap(selector)</code>	Remove the parents of the set of matched elements from the DOM, leaving the matched elements in their place.
<code>.wrap([wrappingElement/function])</code>	Wrap an HTML structure around each element in the set of matched elements.
<code>.wrapAll([wrappingElement/function])</code>	Wrap an HTML structure around all elements in the set of matched elements.
<code>.wrapInner([wrappingElement/function])</code>	Wrap an HTML structure around the content of each element in the set of matched elements.