jQUERY 1.5 VISUAL CHEAT SHEET

Designed by Antonio Lupetti © 2011 • http://woorkup.com • http://twitter.com/woork | jQuery is © of John Resig and the jQuery Team.

★ = NEW IN jQUERY 1.5 / *f(x)* = FUNCTION / *a* = ARRAY / *jQ* = jQUERY / *El* = ELEMENT / 0–1 = BOOLEAN / *Obj* = OBJECT / *NUM* = NUMBER / *Str* = STRING

✻ SELECTORS / 1. BASIC

All Selector ("*")
Selects all elements. — <*El(s)*>

Class Selector (".class")
Matches all elements with the given name. — *a*<*El(s)*>

Element Selector ("element")
Selects all elements with the given tag name. — *a*<*El(s)*>

ID Selector ("#id")
Selects a single element with the given id attribute. — *a*<*El*>

Multiple Selector ("selector1, selector2, selectorN")
Selects the combined results of all the specified selectors. — *a*<*El(s)*>

✻ SELECTORS / 2. HIERARCHY

Child Selector ("parent > child")
Selects all direct child elements specified by "child" of elements specified by "parent". — *a*<*El(s)*>

Descendant Selector ("ancestor descendant")
Selects all elements that are descendants of a given ancestor. — *a*<*El(s)*>

Next Adjacent Selector ("prev + next")
Selects all next elements matching "next" that are immediately preceded by a sibling "prev". — *a*<*El(s)*>

Next Siblings Selector ("prev ~ siblings")
Selects all sibling elements that follow after the "prev" element, have the same parent, and match the filtering "siblings" selector. — *a*<*El(s)*>

✻ SELECTORS / 3. BASIC FILTER

:animated Selector
Select all elements that are in the progress of an animation at the time the selector is run. — *a*<*El(s)*>

:eq() Selector
Select the element at index n within the matched set. — *a*<*El*>

:even Selector
Selects even elements, zero-indexed — *a*<*El(s)*>

:first Selector
Selects the first matched element. — *a*<*El*>

:gt() Selector
Select all elements at an index greater than index within the matched set. — *a*<*El(s)*>

:header Selector
Selects all elements that are headers, like h1, h2, h3 and so on. — *a*<*El(s)*>

:last Selector
Selects the last matched element. — *a*<*El*>

:lt() Selector
Select all elements at an index less than index within the matched set. — *a*<*El(s)*>

:not() Selector
Selects all elements that do not match the given selector. — *a*<*El(s)*>

:odd Selector
Selects odd elements, zero-indexed. See also even. — *a*<*El(s)*>

✻ SELECTORS / 4. CONTENT FILTER

:contains() Selector
Select all elements that contain the specified text. — *a*<*El(s)*>

:empty Selector
Select all elements that have no children (including text nodes). — *a*<*El(s)*>

:has() Selector
Selects elements which contain at least one element that matches the specified selector. — *a*<*El(s)*>

:parent Selector
Select all elements that are the parent of another element, including text nodes. — *a*<*El(s)*>

✻ SELECTORS / 5. ATTRIBUTE

[name|=value]
Selects elements that have the specified attribute with a value either equal to a given string or starting with that string followed by a hyphen (-). — *a*<*El(s)*>

[name*=value]
Selects elements that have the specified attribute with a value containing the a given substring. — *a*<*El(s)*>

[name~=value]
Selects elements that have the specified attribute with a value containing a given word, delimited by spaces. — *a*<*El(s)*>

[name$=value]
Selects elements that have the specified attribute with a value ending exactly with a given string. — *a*<*El(s)*>

[name=value]
Selects all elements that are headers, like h1, h2, h3 and so on. — *a*<*El(s)*>

[name!=value]
Select elements that either don't have the specified attribute, or do have the specified attribute but not with a certain value. — *a*<*El(s)*>

[name^=value]
Selects elements that have the specified attribute with a value beginning exactly with a given string. — *a*<*El(s)*>

[name]
Selects elements that have the specified attribute, with any value. — *a*<*El(s)*>

[name=value][name2=value2]
Matches elements that match all of the specified attribute filters. — *a*<*El(s)*>

✻ SELECTORS / 6. CHILD FILTER

:first-child Selector
Selects all elements that are the first child of their parent. — *a*<*El(s)*>

:last-child Selector
Selects all elements that are the last child of their parent. — *a*<*El(s)*>

:nth-child Selector
Selects all elements that are the nth-child of their parent. — *a*<*El(s)*>

:only-child Selector
Selects all elements that are the only child of their parent. — *a*<*El(s)*>

✻ SELECTORS / 7. VISIBILITY FILTER

:hidden Selector
Selects all elements that are hidden. — *a*<*El(s)*>

:visible Selector
Selects all elements that are visible. — *a*<*El(s)*>

✻ SELECTORS / 8. FORM

:button Selector
Selects all button elements and elements of type button. — *a*<*El(s)*>

:checkbox Selector
Selects all elements of type checkbox. — *a*<*El(s)*>

:checked Selector
Matches all elements that are checked. — *a*<*El(s)*>

:disabled Selector
Selects all elements that are disabled. — *a*<*El(s)*>

:enabled Selector
Selects all elements that are enabled. — *a*<*El(s)*>

:file Selector
Selects all elements of type file. — *a*<*El(s)*>

:image Selector
Selects all elements of type image. — *a*<*El(s)*>

:input Selector
Selects all input, textarea, select and button elements. — *a*<*El(s)*>

:password Selector
Selects all elements of type password. — *a*<*El(s)*>

:radio Selector
Selects all elements of type radio. — *a*<*El(s)*>

:reset Selector
Selects all elements of type reset. — *a*<*El(s)*>

:selected Selector
Selects all elements that are selected. — *a*<*El(s)*>

:submit Selector
Selects all elements of type submit. — *a*<*El(s)*>

:text Selector
Selects all elements of type text. — *a*<*El(s)*>

✻ CORE / 1. THE jQUERY FUNCTION

jQuery()
Accepts a string containing a CSS selector which is then used to match a set of elements. — *jQ*

jQuery.sub() ★
Creates a new copy of jQ whose properties and methods can be modified without affecting the original jQuery object. — *jQ*

jQuery.when() ★
Provides a way to execute callback functions based on one or more objects, usually Deferred objects that represent asynchronous events. — *Deferred*

jQuery.noConflict( )
Relinquish jQuery's control of the $ variable. — *Obj*

jQuery.extend(object)
Extends the jQuery object itself. — *jQ*

✻ CORE / 2. OBJECT ACCESSORS

.context
The DOM node context originally passed to jQuery(). — *El*

.each(function(index, Element))
Iterate over a jQ object, executing a function for each matched element. — *jQ*

.get([index])
Retrieve the DOM elements matched by the jQuery object. — *El* | *a*

.index()
Search for a given element from among the matched elements. — *Num*

.length
The number of elements in the jQuery object. — *Num*

.selector
A selector representing selector originally passed to jQuery(). — *Str*

.size()
Return the number of DOM elements matched by the jQuery object. — *Num*

.toArray()
Retrieve all the DOM elements contained in the jQuery set, as an array. — *a*

✻ CORE / 3. DATA

.queue([queueName], newQueue)
Show the queue of functions to be executed on the matched elements. — *jQ*

.data(obj)
Store arbitrary data associated with the matched elements. — *jQ*

.removeData([name])
Remove a previously-stored piece of data. — *jQ*

.dequeue([queueName])
Execute the next function on the queue for the matched elements. — *jQ*

✻ CORE / 4. INTEROPERABILITY

jQuery.fn.extend(object)
Extends the jQuery element set to provide new methods (used to make a typical jQuery plugin). — *jQ*

jQuery.extend(object)
Extends the jQuery object itself. — *jQ*

✻ ATTRIBUTES / 1. ATTR

.attr(attributeName)
Get the value of an attribute for the first element in the set of matched elements. — *Obj*

.attr(attributeName, value)
Set one or more attributes for the set of matched elements. — *Obj*

.removeAttr()
Remove an attribute from each element in the set of matched elements. — *jQ*

jQUERY 1.5
VISUAL CHEAT SHEET

Designed by Antonio Lupetti © 2011 • http://woorkup.com • http://twitter.com/woork | jQuery is © of John Resig and the jQuery Team.

★ = NEW IN jQUERY 1.5 / *f(x)* = FUNCTION / *a* = ARRAY / *jQ* = jQUERY / *El* = ELEMENT / 0–1 = BOOLEAN / *Obj* = OBJECT / *NUM* = NUMBER / *Str* = STRING

✲ ATTRIBUTES / 2. CLASS

.addClass(class) — jQ
Adds the specified class(es) to each of the set of matched elements.

.hasClass(class) — 0–1
Determine whether any of the matched elements are assigned the given class.

.removeClass(class) — jQ
Remove a single class, multiple classes, or all classes from each element in the set of matched elements.

.toggleClass(class, switch) — jQ
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 switch argument.

✲ ATTRIBUTES / 3. HTML

.html() — Str
Get the HTML contents of the first element in the set of matched elements.

.html(htmlString) — jQ
Set the HTML contents of each element in the set of matched elements.

✲ ATTRIBUTES / 4. TEXT

.text() — Str
Get the combined text contents of each element in the set of matched elements, including their descendants.

.text(textString) — jQ
Set the content of each element in the set of matched elements to the specified text.

✲ ATTRIBUTES / 5. VALUE

.val() — Str | a
Get the current value of the first element in the set of matched elements.

.val(value) — jQ
Set the value of each element in the set of matched elements.

✲ CSS / 1. CSS

.css(propertyName) — Str
Get the value of a style property for the first element in the set of matched elements.

.css(propertyName, value) — jQ
Set one or more CSS properties for the set of matched elements.

✲ CSS / 2. POSITIONING

.scrollLeft() — Int
Get the current horizontal position of the scroll bar for the first element in the set of matched elements.

.scrollLeft(value) — jQ
Set the current horizontal position of the scroll bar for each of the set of matched elements.

.offset() — Obj {top, left}
Get the current coordinates of the first element in the set of matched elements, relative to the document.

.offset(coordinates) — jQ
Set the current coordinates of every element in the set of matched elements, relative to the document.

.position() — Obj {top, left}
Get the current coordinates of the first element in the set of matched elements, relative to the offset parent.

.scrollTop() — Int
Get the current vertical position of the scroll bar for the first element in the set of matched elements.

.scrollTop(value)
Set the current vertical position of the scroll bar for each of the set of matched elements.

✲ CSS / 3. HEIGHT & WIDTH

.height(value) — jQ
Set the CSS height of every matched element.

.height() — Int
Get the current computed height for the first element in the set of matched elements.

.innerHeight() — Int
Get the current computed height for the first element in the set of matched elements, including padding but not border.

.innerWidth() — Int
Get the current computed width for the first element in the set of matched elements, including padding but not border.

.outerHeight() — Int
Get the current computed height for the first element in the set of matched elements, including padding, border, and optionally margin.

.outerWidth() — Int
Get the current computed width for the first element in the set of matched elements, including padding and border.

.width(value) — jQ
Set the CSS width of each element in the set of matched elements.

.width() — Int
Get the current computed width for the first element in the set of matched elements.

✲ TRAVERSING / 1. FILTERING

.eq(- index) — jQ
Reduce the set of matched elements to the one at the specified index.

.eq(index) — jQ
Reduce the set of matched elements to the one at the specified index.

.filter(selector) — jQ
Reduce the set of matched elements to those that match the selector or pass the function's test.

.is(selector) — 0–1
Check the current matched set of elements against a selector and return true if at least one of these elements matches the selector.

.map(callback(index, domEl)) — jQ
Pass each element in the current matched set through a function, producing a new jQuery object containing the return values.

.not() — jQ
Remove elements from the set of matched elements.

.slice(start, [end]) — jQ
Reduce the set of matched elements to a subset specified by a range of indices.

✲ TRAVERSING / 2. TREE TRAVERSAL

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

.closest(selector) — jQ
Get the first ancestor element that matches the selector, beginning at the current element and progressing up through the DOM tree.

.closest(selectors, [context]) — jQ
Get the first ancestor element that matches the selector, beginning at the current element and progressing up through the DOM tree.

.find(selector) — jQ
Get the descendants of each element in the current set of matched elements, filtered by a selector.

.next([selector]) — jQ
Get the immediately following sibling of each element in the set of matched elements, optionally filtered by a selector.

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

.nextUntil([selector]) — jQ
Get all following siblings of each element up to but not including the element matched by the selector.

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

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

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

.parentsUntil([selector]) — jQ
Get the ancestors of each element in the current set of matched elements, up to but not including the element matched by the selector.

.prev([selector]) — jQ
Get the immediately preceding sibling of each element in the set of matched elements, optionally filtered by a selector.

.prevAll([selector]) — jQ
Get all preceding siblings of each element in the set of matched elements, optionally filtered by a selector.

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

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

✲ TRAVERSING / 3. MISCELLANEOUS

.add() — jQ
Add elements to the set of matched elements.

.add(selectors, [context]) — jQ
Add elements to the set of matched elements.

.andSelf() — jQ
Add the previous set of elements on the stack to the current set.

.contents() — jQ
Get the children of each element in the set of matched elements, including text nodes.

.end() — jQ
End the most recent filtering operation in the current chain and return the set of matched elements to its previous state.

✲ MANIPULATION / 1. INSIDE

.append(content) — jQ
Insert content, specified by the parameter, to the end of each element in the set of matched elements.

.append(function(index, html)) — jQ
Insert content, specified by the parameter, to the end of each element in the set of matched elements.

.appendTo(target) — jQ
Insert every element in the set of matched elements to the end of the target.

.prepend(content) — jQ
Insert content, specified by the parameter, to the beginning of each element in the set of matched elements.

.prependTo(target) — jQ
Insert content, specified by the parameter, to the end of each element in the set of matched elements.

✲ MANIPULATION / 2. OUTSIDE

.after(content) — jQ
Insert content, specified by the parameter, after each element in the set of matched elements.

.after(function(index)) — jQ
Insert content, specified by the parameter, to the end of each element in the set of matched elements.

.before(content) — jQ
Insert content, specified by the parameter, before each element in the set of matched elements.

.before(function) — jQ
Insert content, specified by the parameter, before each element in the set of matched elements.

.insertAfter(target) — jQ
Insert every element in the set of matched elements after the target.

.insertBefore(target) — jQ
Insert every element in the set of matched elements before the target.

jQUERY 1.5
VISUAL CHEAT SHEET

Designed by Antonio Lupetti © 2011 • http://woorkup.com • http://twitter.com/woork |

★ = NEW IN jQUERY 1.5 / *f(x)* = FUNCTION / *a* = ARRAY / *jQ* = jQUERY / *El* = ELEMENT / 0–1 = BOOLEAN / *Obj* = OBJECT / *NUM* = NUMBER / *Str* = STRING

✲ MANIPULATION / 3. AROUND

.unwrap()
Remove the parents of the set of matched elements from the DOM, leaving the matched elements in their place. jQ

.wrap(wrappingElement)
Wrap an HTML structure around each element in the set of matched elements. jQ

.wrap(wrappingFunction)
Wrap an HTML structure around each element in the set of matched elements. jQ

.wrapAll(wrappingElement)
Wrap an HTML structure around all elements in the set of matched elements. jQ

.wrapInner(wrappingElement)
Wrap an HTML structure around the content of each element in the set of matched elements. jQ

.wrapInner(wrappingFunction)
Wrap an HTML structure around the content of each element in the set of matched elements. jQ

✲ MANIPULATION / 4. REPLACING

.replaceWith(newContent)
Replace each element in the set of matched elements with the provided new content. jQ

.replaceWith(function)
Replace each element in the set of matched elements with the provided new content. jQ

.replaceAll()
A selector expression indicating which element(s) to replace. jQ

✲ MANIPULATION / 5. REMOVING

.detach([selector])
Remove the set of matched elements from the DOM. jQ

.empty()
Remove all child nodes of the set of matched elements from the DOM. jQ

.remove([selector])
Remove the set of matched elements from the DOM. jQ

✲ MANIPULATION / 6. COPYING

.clone([withDataAndEvents])
Create a deep copy of the set of matched elements. jQ

✲ EVENTS / 1. DOCUMENT LOADING

.load(handler(eventObject))
Bind an event handler to the "load" JavaScript event. jQ

.ready(handler)
Specify a function to execute when the DOM is fully loaded. jQ

.unload(handler(eventObject))
Bind an event handler to the "unload" JavaScript event. jQ

✲ EVENTS / 2. HANDLER ATTACHMENT

.bind(eventType, [eventData], handler (eventObject))
Attach a handler to an event for the elements. jQ

.delegate(selector, eventType, handler)
Attach a handler to one or more events for all elements that match the selector, now or in the future, based on a specific set of root elements. jQ

.die()
Remove all event handlers previously attached using .live() from the elements. jQ

.live(eventType, eventData, handler)
Attach a handler to the event for all elements which match the current selector, now or in the future. jQ

.one(eventType, [eventData], handler (eventObject))
Attach a handler to an event for the elements. The handler is executed at most once per element. jQ

.trigger(eventType, extraParameters)
Execute all handlers and behaviors attached to the matched elements for the given event type. jQ

.triggerHandler(eventType, extraParameters)
Execute all handlers attached to an element for an event. jQ

.unbind(eventType, handler (eventObject))
Remove a previously-attached event handler from the elements. jQ

.undelegate()
Remove a handler from the event for all elements which match the current selector, now or in the future, based upon a specific set of root elements. jQ

✲ EVENTS / 3. MOUSE EVENTS

.click(handler(eventObject))
Bind an event handler to the "click" JavaScript event, or trigger that event on an element. jQ

.dblclick(handler(eventObject))
Bind an event handler to the "dblclick" JavaScript event, or trigger that event on an element. jQ

.focusin(handler(eventObject))
Bind an event handler to the "focusin" JavaScript event. jQ

.focusout(handler(eventObject))
Bind an event handler to the "focusout" JavaScript event. jQ

.hover(handlerIn(eventObject), handlerOut(eventObject))
Bind two handlers to the matched elements, to be executed when the mouse pointer enters and leaves the elements. jQ

.hover(handler(eventObject))
Bind a single handler to the matched elements, to be executed when the mouse pointer enters or leaves the elements. jQ

.mousedown(handler(eventObject))
Bind an event handler to the "mousedown" JavaScript event, or trigger that event on an element. jQ

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

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

.mousemove(handler(eventObject))
Bind an event handler to the "mousemove" JavaScript event, or trigger that event on an element. jQ

.mouseout(handler(eventObject))
Bind an event handler to the "mouseout" JavaScript event, or trigger that event on an element. jQ

.mouseover(handler(eventObject))
Bind an event handler to the "mouseover" JavaScript event, or trigger that event on an element. jQ

.mouseup(handler(eventObject))
Bind an event handler to the "mouseup" JavaScript event, or trigger that event on an element. jQ

✲ EVENTS / 4. FORM EVENTS

.blur(handler(eventObject))
Bind an event handler to the "blur" JavaScript event, or trigger that event on an element. jQ

.change(handler(eventObject))
Bind an event handler to the "change" JavaScript event, or trigger that event on an element. jQ

.focus(handler(eventObject))
Bind an event handler to the "focus" JavaScript event, or trigger that event on an element. jQ

.select(handler(eventObject))
Bind an event handler to the "select" JavaScript event, or trigger that event on an element. jQ

.submit(handler(eventObject))
Bind an event handler to the "submit" JavaScript event, or trigger that event on an element. jQ

✲ EVENTS / 5 KEYBOARD EVENTS

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

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

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

✲ EVENTS / 6. EVENT OBJECT

event.currentTarget
The current DOM element within the event bubbling phase. El

event.data
Contains the optional data passed to jQuery.fn.bind when the current executing handler was bound.

event.isDefaultPrevented()
Returns whether event.preventDefault() was ever called on this event object. 0–1

event.isImmediatePropagationStopped()
Returns whether event.stopImmediatePropagation() was ever called on this event object. 0–1

event.isPropagationStopped()
Returns whether event.stopPropagation() was ever called on this event object. 0–1

event.pageX
The mouse position relative to the left edge of the document. Num

event.pageY
The mouse position relative to the top edge of the document. Num

event.preventDefault()
If this method is called, the default action of the event will not be triggered.

event.relatedTarget
The other DOM element involved in the event, if any. El

event.result
This attribute contains the last value returned by an event handler that was triggered by this event, unless the value was undefined. Obj Anything

event.stopImmediatePropagation()
Prevents other event handlers from being called.

event.stopPropagation()
Prevents the event from bubbling up the DOM tree, preventing any parent handlers from being notified of the event.

event.target
The DOM element that initiated the event. El

event.timeStamp
This attribute returns the number of milliseconds since January 1, 1970, when the event is triggered. Num

event.type
Describes the nature of the event. Str

event.which
For key or button events, this attribute indicates the specific button or key that was pressed. Str

✲ EVENTS / 6. BROWSER EVENTS

.error(handler(eventObject))
Bind an event handler to the "error" JavaScript event. jQ

.resize(handler(eventObject))
Bind an event handler to the "resize" JavaScript event, or trigger that event on an element. jQ

.scroll(handler(eventObject))
Bind an event handler to the "scroll" JavaScript event, or trigger that event on an element. jQ

jQUERY 1.5 VISUAL CHEAT SHEET

Designed by Antonio Lupetti © 2011 • http://woorkup.com • http://twitter.com/woork | jQuery is © of John Resig and the jQuery Team.

★ = NEW IN jQUERY 1.5 / *f(x)* = FUNCTION / *a* = ARRAY / *jQ* = jQUERY / *El* = ELEMENT / 0-1 = BOOLEAN / *Obj* = OBJECT / *NUM* = NUMBER / *Str* = STRING

✻ EFFECTS / 1. BASIC

.hide(duration, [callback]) — *jQ*
Hide the matched elements.

.show(duration, [callback]) — *jQ*
Display the matched elements.

✻ EFFECTS / 2. SLIDING

.slideDown([duration], [callback]) — *jQ*
Display the matched elements with a sliding motion.

.slideToggle([duration], [callback]) — *jQ*
Display or hide the matched elements with a sliding motion.

.slideUp([duration], [callback]) — *jQ*
Hide the matched elements with a sliding motion.

✻ EFFECTS / 3. FADING

.fadeIn([duration], [callback]) — *jQ*
Display the matched elements by fading them to opaque.

.fadeOut([duration], [callback]) — *jQ*
Hide the matched elements by fading them to transparent.

.fadeTo(duration, opacity, [callback]) — *jQ*
Adjust the opacity of the matched elements.

✻ EFFECTS / 4. CUSTOM

.animate(properties, options) — *jQ*
Perform a custom animation of a set of CSS properties.

.delay(duration, [queueName]) — *jQ*
Set a timer to delay execution of subsequent items in the queue.

.stop([clearQueue], [jumpToEnd]) — *jQ*
Stop the currently-running animation on the matched elements.

jQuery.fx.off — 0-1
Globally disable all animations.

✻ EFFECTS / 4. CUSTOM

.animate(properties, options) — *jQ*
Perform a custom animation of a set of CSS properties.

.delay(duration, [queueName]) — *jQ*
Set a timer to delay execution of subsequent items in the queue.

.stop([clearQueue], [jumpToEnd]) — *jQ*
Stop the currently-running animation on the matched elements.

jQuery.fx.off — 0-1
Globally disable all animations.

✻ AJAX / 1. LOW-LEVEL INTERFACE

jQuery.ajax(url, [settings]) ★ — *jqXHR*
Perform an asynchronous HTTP (Ajax) request.

jQuery.ajax(settings) — *jqXHR*
Perform an asynchronous HTTP (Ajax) request.

jQuery.ajaxSetup(option) — 0-1
Set default values for future Ajax requests.

✻ AJAX / 2. SHORTHAND METHODS

jQuery.get(url, [data], [callback(data, textStatus, XMLHttpRequest)], [dataType]) — *jqXHR*
Load data from the server using a HTTP GET request.

jQuery.getJSON(url, [data], [callback(data, textStatus)]) — *jqXHR*
Load JSON-encoded data from the server using a GET HTTP request.

jQuery.getScript(url, [success(data, textStatus)]) — *jqXHR*
Load a JavaScript file from the server using a GET HTTP request, then execute it.

.load(url, [data], [complete (responseText, textStatus, XMLHttpRequest)]) — *jQ*
Load data from the server and place the returned HTML into the matched element.

jQuery.post(url, [data], [success (data, textStatus, XMLHttpRequest)], [dataType]) — *jqXHR*
Load data from the server using a HTTP POST request.

✻ AJAX / 3. AJAX EVENT HANDLERS

.ajaxComplete(handler(event, XMLHttpRequest, ajaxOptions)) — *jQ*
Register a handler to be called when Ajax requests complete.

.ajaxStart(handler()) — *jQ*
Register a handler to be called when the first Ajax request begins.

.ajaxStop(handler()) — *jQ*
Hide a loading message after all the Ajax requests have stopped.

.ajaxError(handler(event, XMLHttpRequest, ajaxOptions, thrownError)) — *jQ*
Register a handler to be called when Ajax requests complete with an error.

.ajaxSend(handler(event, XMLHttpRequest, ajaxOptions)) — *jQ*
Show a message before an Ajax request is sent.

.ajaxSuccess(handler(event, XMLHttpRequest, ajaxOptions)) — *jQ*
Show a message when an Ajax request completes successfully.

✻ AJAX / 4. DATA

.serialize() — *Str*
Encode a set of form elements as a string for submission.

.serializeArray() — *a*
Encode a set of form elements as an array of names and values.

✻ UTILITIES / 1. UTILITIES

jQuery.browser — *Map*
Contains flags for the useragent, read from navigator.userAgent. While jQuery.browser will not be removed from future versions of jQuery, every effort to use jQuery.support and proper feature detection should be made.

jQuery.browser.version — *Str*
Returns the version number of the rendering engine for the user's browser.

jQuery.contains(container, contained) — 0-1
Check to see if a DOM node is within another DOM node.

jQuery.each(collection, callback (indexInArray, valueOfElement)) — *Obj*
Iterates through the array displaying each number as both a word and numeral

jQuery.extend(target, [object1], [objectN]) — *Obj*
Merge the contents of two or more objects together into the first object.

jQuery.globalEval(code)
Execute some JavaScript code globally.

jQuery.hasData(element) ★ — 0-1
Determine whether an element has any jQuery data associated with it.

jQuery.grep(array, function (elementOfArray, indexInArray), [invert]) — *a*
Finds the elements of an array which satisfy a filter function. The original array is not affected.

jQuery.inArray(value, array) — *Num*
Search for a specified value within an array and return its index (or -1 if not found).

jQuery.isArray(obj) — 0-1
Determine whether the argument is an array.

jQuery.isEmptyObject(obj) — 0-1
Check to see if an object is empty (contains no properties).

jQuery.isFunction(obj) — 0-1
Determine if the argument passed is a Javascript function object.

jQuery.isPlainObject(obj) — 0-1
Check to see if an object is a plain object (created using "{}" or "new Object").

jQuery.isXMLDoc(node) — 0-1
Check to see if a DOM node is within an XML document (or is an XML document).

jQuery.makeArray(obj) — *a*
Convert an array-like object into a true JavaScript array.

jQuery.map(array, callback (elementOfArray, indexInArray)) — *a*
Translate all items in an array or array-like object to another array of items.

jQuery.merge(first, second) — *a*
Merge the contents of two arrays together into the first array.

jQuery.noop()
An empty function.

jQuery.parseJSON(json) — *Obj*
Takes a well-formed JSON string and returns the resulting JavaScript object.

jQuery.proxy(function, context) — *f(x)*
Takes a function and returns a new one that will always have a particular context.

jQuery.queue(element, [queueName]) — *a*
Show the queue of functions to be executed on the matched element.

jQuery.queue(element, queueName, newQueue) — *a*
Show the queue of functions to be executed on the matched element.

jQuery.removeData(element, [name]) — *jQ*
Remove a previously-stored piece of data.

jQuery.support — *Obj*
A collection of properties that represent the presence of different browser features or bugs.

jQuery.trim(str) — *Obj*
Remove the whitespace from the beginning and end of a string.

jQuery.parseXML(data) ★ — *XMLdoc*
Parses a string into an XML document.

jQuery.unique() — *jQ*
Sorts an array of DOM elements, in place, with the duplicates removed.

✻ DEFERRED OBJECT ★

deferred.done(doneCallbacks) — *Deferred*
Add handlers to be called when the Deferred object is resolved.

deferred.fail(failCallbacks) — *Deferred*
Add handlers to be called when the Deferred object is rejected.

deferred.isRejected() — 0-1
Determine whether a Deferred object has been rejected.

deferred.isResolved() — 0-1
Determine whether a Deferred object has been resolved.

deferred.promise() — *Promise*
Return a Deferred's Promise object.

deferred.reject(args) — *Deferred*
Reject a Deferred object and call any failCallbacks with the given args.

deferred.rejectWith(context,[args]) — *Deferred*
Reject a Deferred object and call any failCallbacks with the given context and args.

deferred.resolve(args) — *Deferred*
Resolve a Deferred object and call any doneCallbacks with the given args.

deferred.resolveWith(args) — *Deferred*
Resolve a Deferred object and call any doneCallbacks with the given context and args.

deferred.then(doneCallbacks, failCallbacks) — *Deferred*
Add handlers to be called when the Deferred object is resolved or rejected.