Attributes / 2. Class

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

- .hasClass(class)
  Determines whether any of the matched elements are assigned the given class.

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

- .toggleClass(class, switch)
  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()
  Get the HTML contents of each element in the set of matched elements.

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

Attributes / 4. Text

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

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

Attributes / 5. Value

- .val()
  Get the current value of the first element in the set of matched elements.

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

CSS / 2. Positioning

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

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

- .offset()
  Get the coordinates of the first element in the set of matched elements, relative to the document.

- .offset(coordinates)
  Set the current coordinates of every element in the set of matched elements, relative to the offset parent.

- .position()
  Get the coordinates of the first element in the set of matched elements, relative to the offset parent.

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

- .scrollTop()
  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

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

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

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

- .height()
  Get the current computed height for each element in the set of matched elements.

CSS / 1. CSS

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

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

CSS / 2. Filtering

- .eq(index)
  Reduce the set of matched elements to those at the specified index.

- .eq(index)
  Reduce the set of matched elements to those at the specified index.

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

- .is(selector)
  Checks the current matched set of elements against a selector and return true if at least one of these elements matches the selector.

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

- .not()
  Remove elements from the set of matched elements.

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

- .slice()
  Reduce the set of matched elements to a subset specified by a range of indices.

CSS / 3. Miscellaneous

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

- .closest(selector)
  Get the first ancestor element that matches the selector.

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

- .children selectors, [ context ]
  Get the first ancestor element that matches the selector, beginning at the current element and progressing up through the DOM tree.

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

- .add(selector)
  Add elements to the set of matched elements.

- .add selectors, [ context ]
  Add elements to the set of matched elements.
**EVENTS / 1. DOCUMENT LOADING**

- `.load(handler(eventObject))`:
  Bind an event handler to the "load" JavaScript event.

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

- `.unload(handler(eventObject))`:
  Bind an event handler to the "unload" JavaScript event.

**EVENTS / 2. HANDLER ATTACHMENT**

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

- `.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.

- `.die()`:
  Remove all event handlers previously attached using `.jcall()` from the elements.

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

- `.on( eventType, [ eventData ], handler (eventObject))`:
  Attach an event handler for the matched elements for the given event type.

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

- `.triggerHandler( eventType, extraParameters)`:
  Execute all handlers attached to an event for an event.

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

- `.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.

**EVENTS / 3. MOUSE EVENTS**

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

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

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

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

- `.mouseover( handlerIn(eventObject), handlerOut(eventObject))`:
  Bind two handlers to the matched elements, to be executed when the mouse pointer enters and leaves the elements.

- `.mouseout( handler(eventObject))`:
  Bind a single handler to the matched elements, to be executed when the mouse pointer enters and leaves the elements.

**EVENTS / 4. FORM EVENTS**

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

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

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

- `.focusout( handler(eventObject))`:
  Bind an event handler to the "focusout" JavaScript event.

- `.clickIn(eventObject)`:
  Execute all handlers attached to an element.

**EVENTS / 5. KEYBOARD EVENTS**

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

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

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

**EVENTS / 6. EVENT OBJECT**

- `.event.currentTarget`:
  The current DOM element within the event bubbling phase.

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

**EVENTS / 7. TRAVERSING**

- `.parent()`:
  Returns the parent of the matched elements.

- `.parents( selector)`:
  Returns all ancestors of the matched elements.

**EVENTS / 8. MANIPULATION**

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

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

- `.html( content)`:
  Replace the content of each element in the set of matched elements.

- `.replaceWith( content)`:
  Remove the set of matched elements from the DOM, and replace them with the provided new content.

- `.after( [ newContent ] )`:
  Insert new content immediately after each element in the set of matched elements.

- `.before( [ content ] )`:
  Insert content immediately before each element in the set of matched elements.

- `.append( [ content ] )`:
  Insert content immediately after each element in the set of matched elements.

- `.prepend( [ content ] )`:
  Insert content immediately before each element in the set of matched elements.

- `.remove()`:
  Remove each element in the set of matched elements.

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

- `.wrapInner( wrappingElement)`:
  Wrap an HTML structure around each element in the set of matched elements.

- `.hasClass( className )`:
  Return true if the element has the specified class name.

- `.addClass( className )`:
  Add one or more class names to an element.

- `.removeClass( className )`:
  Remove one or more class names from an element.

- `.toggleClass( className, [ state ] )`:
  Toggle the specified class name on an element.

- `.attr( attributeName, value )`:
  Set or get the value of an attribute.

- `.css( propertyName, value )`:
  Set or get the value of a CSS property.

- `.last()`:
  Return the last element in the set of matched elements.

- `.first()`:
  Return the first element in the set of matched elements.

- `.siblings()`:
  Return all sibling elements of the current element.

- `.next()`:
  Return the next sibling element of the current element.

- `.prev()`:
  Return the previous sibling element of the current element.

**EVENTS / 9. SELECTORS**

- `.filter( selector )`:
  Select elements that match the specified selector.

- `.unwrap()`:
  Remove the parents of the set of matched elements from the DOM.

**EVENTS / 10. EFFECTS**

- `.fadeTo( duration, opacity )`:
  Fade the opacity of the matched elements.

- `.animate(...)`:
  Animate the properties of the matched elements.

- `.animateFunction(...)`:
  Animate a custom function applied to the matched elements.

- `.animateSet(...)`:
  Animate a set of properties on the matched elements.

**EVENTS / 11. AJAX**

- `.ajaxStart(...)`:
  Fire the event when the request begins.

- `.ajaxComplete(...)`:
  Fire the event when the request completes.

- `.ajaxError(...)`:
  Fire the event if an error occurs during the request.

- `.ajaxSuccess(...)`:
  Fire the event when the request is successful.

- `.ajaxStop(...)`:
  Fire the event when the request ends.

**EVENTS / 12. UTILITIES**

- `.is(...)`:
  Test if the elements match a selector.

- `.map(...)`:
  Map the elements using a function and return an array.

- `.val(...)`:
  Get or set the value of an input element.

- `.empty(...)`:
  Test if the elements are empty.

- `.length(...)`:
  Return the number of matched elements.

- `.offset(...)`:
  Return the offset of the matched elements.

- `.parent(...)`:
  Return the parent of the matched elements.

- `.parents(...)`:
  Return all ancestors of the matched elements.

- `.prev(...)`:
  Return the previous sibling of the matched elements.

- `.prevAll(...)`:
  Return all previous siblings of the matched elements.

- `.siblings(...)`:
  Return all siblings of the matched elements.

- `.swap(...)`:
  Swap the position of the matched elements with another set.

- `.wrapAll(...)`:
  Wrap an HTML structure around all elements in the set of matched elements.
jQuery 1.5
VISUAL CHEAT SHEET

### EFFECTS / 1. BASIC
- `.show(duration, [callback])`
  - Hide the matched elements.
- `.hide(duration, [callback])`
  - Display the matched elements.

### EFFECTS / 2. SLIDING
- `.slideUp([duration], [callback])`
  - Display or hide the matched elements with a sliding motion.
- `.slideDown([duration], [callback])`
  - Hide the matched elements with a sliding motion.

### EFFECTS / 3. FADEING
- `.fadeIn([duration], [callback])`
  - Display the matched elements by fading them to opaque.
- `.fadeOut([duration], [callback])`
  - Adjust the opacity of the matched elements.
- `.fadeTo(duration, opacity, [callback])`
  - Globally disable all animations.

### EFFECTS / 4. CUSTOM
- `.animate(properties, options)`
  - Perform a custom animation of a set of CSS properties.
- `.delay(duration, [queueName])`
  - Set a timer to delay execution of subsequent items in the queue.
- `.stop([clearQueue], [jumpToEnd])`
  - Stop the currently-running animation on the matched elements.

### SHORTHAND METHODS
- `.ajaxSetup(options)`
  - Set default values for future Ajax requests.
- `.ajaxStart(handler)`
  - Register a handler to be called when Ajax requests complete.
- `.ajaxComplete(handler, event, XMLHttpRequest, AjaxOptions)`
  - Add handlers to be called when the first array item has completed.
- `.ajaxSend(handler, event, XMLHttpRequest, AjaxOptions)`
  - Add handlers to be called when an Ajax request is sent.
- `.ajaxStart(handler, event, XMLHttpRequest, AjaxOptions)`
  - Perform an asynchronous HTTP (Ajax) request.
- `.ajaxSuccess(handler, event, XMLHttpRequest, AjaxOptions)`
  - Show a message when an Ajax request completes successfully.
- `.ajaxError(handler, event, XMLHttpRequest, AjaxOptions, thrownError)`
  - Register a handler to be called when an XMLHttpRequest request completes with an error.

### ARRAYS
- `.clearQueue()`
  - Remove a previously-stored piece of data.
- `. jentePost(array, function(element, indexInArray, [invert]))`
  - Finds the elements of an array which satisfy a filter function. The original array is not affected.
- `.inArray(value, array)`
  - Check to see if an object is empty (contains no properties).
- `.isEmptyObject(obj)`
  - Check to see if an object is a plain object (created using {} or “new Object”).
- `.isPlainObject(obj)`
  - A collection of properties that represent the presence of different browser features or bugs.
- `.isXMLDoc(node)`
  - Check to see if a DOM node is within an XML document (or is an XML document).
- `.map(array, callback)`
  - Convert an array-like object into a true JavaScript array.
- `.querySelector(element, [queueName])`
  - Sorts an array of DOM elements, in place, with the duplicates removed.
- `.resolveWith(context, [args])`
  - Reuse a Deferred object and call any doneCallbacks with the given context.
- `.resolveWith([args])`
  - Reuse a Deferred object and call any doneCallbacks with the given context.
- `.rejectWith([context, [args])`
  - Reject a Deferred object and call any failCallbacks with the given context.
- `.rejectWith([args])`
  - Add handlers to be called when the Deferred object is resolved or rejected.

### UTILITIES
- `.add(element)`
  - Add handlers to be called when the Deferred object is resolved or rejected.