j Q u e r y

www.jsrosettastone.com

G e t t i n g

jQuery 1.4.2	YUI 3.4.1	Notes

`$.foo.bar()`	`YUI().use('node', 'module2', 'module3', function (Y) {`
	`  Y.foo.bar()`
	`});`
		The `jQuery` and `$` objects are globals and the jQuery library itself is statically loaded, so they are available immediately.
		
		YUI is sandboxed and by default dynamically loaded. The `Y` object is local to the function you pass as the last argument to `YUI().use()`. Usually you will put all code that uses YUI inside one of these functions. This function executes **a f t** modules are loaded and accounted for.
		
		The return value of `YUI().use()` is also a Y object, which you can assign to a global variable [e.g. `var Y = YUI().use(…);`] and debug with it in a JavaScript console.

C o m m o n

jQuery 1.4.2	YUI 3.4.1	Notes
`$('div.foo:first')`	`Y.one('div.foo')`	jQuery and Y similar selecti syntax, but jC has added extensions, n convenience pseudo-class the Sizzle CSS3-compli selector engi YUI comes w three differen selector engi see the sectic

jQuery 1.4.2	YUI 3.4.1	Notes
		Selectors.

`var foo = $('div.foo:first');`
`foo.some_method();`
	`var foo = Y.one('div.foo');`
	
	`if (foo) {`
	`  foo.some_method();`
	`}`
		Return the fir element whic matches the selector. `:fi` a jQuery exte
		
		If no element match, `Y.one` returns `null` you should ch for it. jQuery selector meth always return object with 0 more elemen

`$('div.foo')`	`Y.all('div.foo')`	Select all div elements with class of foo.

`var foo = $('div.foo');`
 	
`if (foo.length) {`
`  // do something`
`}`
	`var foo = Y.all('div.foo');`
	
	`if (foo.size()) {`
	`  // do something`
	`}`
		If no element match the sel `Y.all()` retur empty `NodeLi` object. jQuery return an emp `[]` that is loac with special j(methods. Bot truthy even if contain no elements, so `NodeList.si` and `[].lengt` check for emptiness.

`.find('p.foo:first')` `.find('p.foo')`	`.one('p.foo')` `.all('p.foo')`	Finds `P` eleme with class `fo` are children c given node.
`$('<div/>')`	`Y.Node.create('<div/>')`	Create a new element. Doe add it to the document tre

`.html('foo')`
`.text('foo')`
`.val('foo')`
	`.setContent('foo')`
	`.set('text', 'foo')`
	`.set('value', 'foo')`
		`.set()` is a g method in YU modifying ele object proper Use `.setAtt`

jQuery 1.4.2	YUI 3.4.1	Notes
		modify eleme attributes. `.setContent` is a convenie wrapper arou `.set('innerH` *`html`*`)`

`.html()`
`.text()`
`.val()`
	`.get('innerHTML')`
	`.get('text')`
	`.get('value')`
		jQuery tends overload gett and setters in same methoc

`.attr('foo')` `.attr('foo', 'bar')`	`.getAttribute('foo')` `.setAttribute('foo', 'bar')`	Generic HTM attribute gette and setters.
`$.trim(' foo ');`	`Y.Lang.trim(' foo ');`	Strips leading trailing whites

`.click(fn)`
`.focus(fn)`
`.blur(fn)`
`.mouseout(fn)`
`.mouseover(fn)`
 	
`// jQuery 1.4.2 and later allows you to`
`// register events when creating the element`
`$('<p/>',{`
`  text      :'foo',`
`  className : 'bar',`
`  click     : fn,`
`  focus     : fn,`
`  blur      : fn`
`})`
	`.on('click', fn)`
	`.on('focus', fn)`
	`.on('blur', fn)`
	`.on('mouseout', fn)`
	`.on('mouseover', fn)`
	
	`// Alternatively, YUI allows you to attach multiple`
	`// subscribers with a single call.`
	`.on({`
	`  click    : fn,`
	`  focus    : fn,`
	`  blur     : fn,`
	`  mouseout : fn,`
	`  mouseover: fn`
	`})`
	
	`// Or attach a single subscriber to multiple events.`
	`.on(['click', 'focus', 'blur', 'mouseout', 'mouseover'], fn)`
		`.on()` is not chainable by default, but m subscribers c attached in o using the syn shown here.

`$('#foo').trigger('click');`	`Y.one("#foo").simulate("click")`	Simulates a c event. In YUI need the *node-event-simulate* mod
`parent``.append('<div/>')`	*`parent`*`.append('<div/>')`	Creates a ne element and it a child of pa
`child``.appendTo(`*`parent`*`)`	*`child`*`.appendTo(`*`parent`*`)`	Appends `chi` `parent`, and r `child`. `.appendTo()` added to YUI 3.3.0.

jQuery 1.4.2	YUI 3.4.1	Notes

`parent = $('<div/>');`
`$('<p>foo<p>')`
`  .click(fn)`
`  .appendTo(parent);`
	`parent = Y.Node.create('<div/>');`
	`Y.Node.create('<p>foo</p>')`
	`  .appendTo(parent)`
	`  .on('click', fn);`
		Creates a ne element, ther appends a p element with `click` event subscription. that YUI's `on` method is not chainable, so returns an ev handle, not th node.

`.empty()`	`.empty(true)`	jQuery's `.emp` also deregiste any events associated wi elements bei destroyed. Pa `true` to `.empt` enables the s behavior in Y `.empty()` was added to YUI 3.3.0.

`.siblings()`
`.siblings(selector)`
	`.siblings()`
	`.siblings(selector)`
	`.siblings(function)`
		In addition to optional selec string, YUI als supports pass function to filt returned sibli

`.next()`
`.next(selector)`
	`.next()`
	`.next(selector)`
	`.next(fn)`
		Same consideration `.siblings()`.

`.prev()`
`.prev(selector)`
	`.previous()`
	`.previous(selector)`
	`.previous(fn)`
		Same consideration `.siblings()`.

`.parent()`	`.get('parentNode')`	Returns the p node of the g node.
`.children()`	`.get('children')`	Returns all th element child the given noc
`.closest(selector)`	`.ancestor(selector)` `.ancestor(fn)`	Returns the fi ancestor that matches the selector. In addition, YUI supports usin function inste

jQuery 1.4.2	YUI 3.4.1	Notes
		a selector to t ancestor.
`$.contains(node, descendant)`	`.contains(descendant)`	Check to see node contains certain desce
`.show()` `.hide()`	`.show()` `.hide()`	Make DOM n appear/disap
`.fadeIn()` `.fadeOut()`	`.show(true)` `.hide(true)`	In YUI, `.show` and `.hide()` be customize use transition supported by `transition` module. Thes methods wer added to YUI 3.3.0.
`$.parseJSON( '{"name":"Douglas"}' )`	`Y.JSON.parse( '{"name":"Douglas"}' )`	Converts a JS string into an object.
	`Y.JSON.stringify({name: "Douglas"});`	Converts an o to a JSON str No jQuery equivalent.
`$.proxy(fn, context)`	`Y.bind(fn, context)`	Creates a new function that call the suppl function in a particular con
`.data(key)` `.data(key, value)`	`.getData(key)` `.setData(key, value)`	Stores data associated to DOM elemen without modif the DOM.
`.removeData()` `.removeData(key)`	`.clearData()` `.clearData(key)`	Removes the associated da

S e l e c t o r

jQuery 1.4.2	YUI 3.4.1	Notes
`$('*')`	`Y.all('*')`	Select all nodes. Note that the default selector engine for YUI is CSS 2.1. For all examples in this section, use the `selector-css3` module for YUI.

jQuery 1.4.2	YUI 3.4.1	Notes
`$(':animated')`		Psuedoclass to select all elements currently being animated. No YUI equivalent.
`$(':button')`	`Y.all('input[type=button], button')`	Extension. In both jQuery and YUI you can run multiple selectors separated by commas.
`$(':checkbox')`	`Y.all('input[type=checkbox]')`	Extension.
`$(':checked')`	`Y.all(':checked')`	CSS3
`$('parent > child')`	`Y.all('parent > child')`	Immediate child selector (child must be one level below parent)
`$('parent child')`	`Y.all('parent child')`	Descendent selector (child can be at any level below parent)
`$('div.class')`	`Y.all('div.class')`	Class selector
`$(":contains('foo')")`	`Y.all(':contains(foo)')`	Extension to select all elements whose text matches 'foo'. jQuery can take quotes or not. YUI requires no quotes. The text matching is plain string comparison, not glob or regexp. Be careful with this one as it will return all matching ancestors, eg `[html, body, div]`.
`$(':disabled')` `$(':enabled')`	`Y.all(':disabled')` `Y.all(':enabled')`	CSS3. `'input[disabled]'` and `'input:not([disabled])'` also work in both libraries.
`$(':empty')`	`Y.all(':empty')`	CSS3. Selects all elements that have no child nodes (excluding text nodes).
`$(':parent')`	`Y.all(':not(:empty)')`	Inverse of `:empty`. Will find all elements that are a parent of at least one element. jQuery's version is an extension. YUI's is CSS3.
`$('div:eq(n)')`	`Y.all('div').item(n)`	Extension. Selects *nth* element. YUI's `item()` will return `null` if there is no `nth` element. jQuery's selector will return an empty list `[]` on a match failure.
`$('div:even')` `$('div:odd')`	`Y.all('div').even()` `Y.all('div').odd()`	Extension. Selects all even or odd elements. Note that elements are 0-indexed and the 0th element is considered even. See also YUI's `NodeList.modulus(n, offset)`.
`$(':file')`	`Y.all('input[type=file]')`	Extension. Find input elements whose type=file.
`$('div:first-child')`	`Y.all('div:first-child')`	CSS3. Selects the first child element of divs.
`$('div:first)`	`Y.one('div')`	The `.one()` method returns `null` if there is no match, and a single Node object if there

jQuery 1.4.2	YUI 3.4.1	Notes
		is.

`$('div:gt(n)');`
`$('div:lt(n)');`
`// or`
`$('div').slice(n + 1);`
`$('div').slice(0,n);`
	`Y.all('div').slice(n + 1);`
	`Y.all('div').slice(0, n);`
		Extension. `:gt` (greater than) selects all elements from index `n+1` onwards. `:lt` (less than) selects all nodes from 0 up to `n-1`.

`$('div:has(p)')`	`var nodes = [];`
	
	`Y.all('div').each(function (node) {`
	`  if (node.one('p')) {`
	`  nodes.push(node);`
	`  }`
	`});`
	
	`nodes = Y.all(nodes);`
		Extension. Selects elements which contain at least one element that matches the specified selector. In this example, all `div` tags which have a `p` tag descendent will be selected.

`$(':header')`	`Y.all('h1,h2,h3,h4,h5,h6')`	Extension. Selects all heading elements. Rarely used.

`$('div:hidden')`	`var hidden = [];`
	`Y.all('div').each(function(node) {`
	`  if ((node.get('offsetWidth') === 0 &&`
	`  node.get('offsetHeight') === 0) ||`
	`  node.get('display') === 'none') {`
	`  hidden.push(node);`
	`  }`
	`});`
	`hidden = Y.all(hidden);`
		Extension. In jQuery > 1.3.2 `:hidden` selects all elements (or descendents of elements) which take up no visual space. Elements with `display:none` or whose `offsetWidth/offsetHeight` equal 0 are considered hidden. Elements with `visibility:hidden` are not considered hidden.
		
		The YUI equivalent would essentially be a port of the jQuery code that implements `:hidden`. This might be a good candidate for a patch to YUI.

`$('#id')`	`Y.all('#id')`	CSS3. Identity selector.
`$('input:image')`	`Y.all('input[type=image]')`	Extension. Selects all inputs of type image.
`$(':input')`	`Y.all('input,textarea,select,button')`	Extension. Selects all user-editable form elements.
`$(':last-child')`	`Y.all(':last-child')`	CSS3.

`$('div:last')`	`var list = Y.all('div'),`
	`  last;`
	
	`if (list.size()) {`
	`  last = list.item(list.size() - 1);`
	`}`
		Extension. Selects the last element matched by the selector.

`$('input[type=checkbox][checked]')`	`Y.all('input[type=checkbox][checked]')`	CSS3, multiple attribute selector
`$(':not(div)')`	`Y.all(':not(div)')`	CSS3. Negation selector.
`$(':password')`	`Y.all('input[type=password]')`	Extension.
`$(':radio')`	`Y.all('input[type=radio]')`	Extension.
`$(':reset')`	`Y.all('input[type=reset]')`	Extension.
`$(':selected')`	`Y.all('option[selected]')`	Extension.

jQuery 1.4.2	YUI 3.4.1	Notes
`$(':submit')`	`Y.all('input[type=submit]')`	Extension.
`$(':text')`	`Y.all('input[type=text]')`	Extension. Does not select `textarea` elements.

Effects

jQuery 1.4.2	YUI 3.4.1	Notes

`$('#foo').animate(`
`  {`
`  width: 100,`
`  height: 100,`
`  opacity: 0.5`
`  },`
`  {`
`  duration: 600,`
`  easing: 'swing'`
`  }`
`);`
	`var a = new Y.Anim(`
	`  {`
	`  node: '#foo',`
	`  to: {`
	`  width: 100,`
	`  height: 100,`
	`  opacity: 0.5`
	`  },`
	`  duration: 0.6,`
	`  easing: Y.Easing.bounceOut`
	`  }`
	`);`
	`a.run();`
		The basic syntax and capabilities of both animation libraries are very similar. jQuery has convenience methods for effects like `.fadeIn()`, `.slideUp()`, etc. jQuery core has two easing functions: 'linear' and 'swing', but jQuery UI comes with many more effects as plugins.
		
		YUI has several easing algorithms built-in, and offers additional tools such as animations over Bezier curves. Make sure to load the `'anim'` module in your call to `YUI().use()`.

`$('#.foo').fadeOut();`
 	
`// or`
 	
`$('#.foo').hide(600);`
	`Y.one('#foo').hide(true)`	jQuery's `.fadeOut()` fades the opacity to 0, then sets `display:none` on the element. `.fadeIn()` is naturally the inverse. The YUI equivalents are `.hide(true)` and `.show(true)` (note that the `transition` module must be loaded in order to get the fade effect).
		
		jQuery effects tend to default to 200 or 600ms while YUI's show/hide transitions default to 500ms. YUI durations are in fractions of seconds; jQuery durations are set in milliseconds.

Array vs

jQuery 1.4.2	YUI 3.4.1	Notes
`$('.foo').array_method(args)`	`Y.all('.foo').array_method(args)`	Any Array operation that you can perform on a jQuery list can be translated to YUI in this form. YUI NodeList objects are not native Arrays, but do provide wrapper functions for the most common array methods as of 3.3.0.
`$('div').slice(x, y)`	`Y.all('div').slice(x, y)`	Return the *xth* to the *yth* div elements.
`$('div').add('p')`	`Y.all('div').concat(Y.all('p'));`	Add nodes that match the specified selector.

`$('.foo').each(`
`  function() {`
`  this.some_method();`
`  }`
	`Y.all('.foo').each(`
	`  function() {`
	`  this.some_method();`
	`  }`
		`.each()` is like the `for` loop. YUI's each() returns the original NodeList to help with chaining.

jQuery 1.4.2	YUI 3.4.1	Notes
`);`	`);`	
`$('.foo').filter('.bar')`	`Y.all('.foo').filter('.bar')`	The `.filter()` method in both libraries both take CSS selectors as filter criteria. jQuery's `.filter()` can also take a function.

`var fn = function(idx) {`
`  return this.property === 'value';`
`};`
`$('.foo').filter(fn);`
	`var filtered = [];`
	`Y.all('.foo').each(`
	`  function(node) {`
	`  if (node.get('property') === 'value') {`
	`  filtered.push(node);`
	`  }`
	`  }`
	`);`
	`filtered = Y.all(filtered);`
		Classic functional programming filter function. Given a list of elements, run the function on each and return a list of those which evaluated true. `NodeList.filter(fn)` is coming to a future point release of YUI.

`$('.foo').map(`
`  function(idx, el) {`
`  return some_function(el);`
`  }`
`)`
	`var mapped = [];`
	`Y.all('.foo').each(`
	`  function(node) {`
	`  mapped.push(`
	`  return some_function(node)`
	`  );`
	`  }`
	`);`
	`mapped = Y.all(mapped);`
		jQuery's `.map()` returns a jQuery-wrapped array of the return values of calls to the given function. `NodeList.map(fn)` is coming to a future point release of YUI.

A j a x

jQuery 1.4.2	YUI 3.4.1	Notes

`$.ajax({`
`  url:  url,`
`  data:  data,`
`  success:  successFn`
`});`
	`Y.io(url, {`
	`  data: data,`
	`  on:  {success: successFn}`
	`});`
		YUI.io has extra options for failure mode callbacks, headers, cross-frame i/o, etc. jQuery.ajax() has some interesting options for async, context, and filtering. Make sure to load the YUI `'io'` module.

	`Y.io(url, {`
	`  data: data,`
	`  on:  {success: successFn},`
	`  xdr:  {use: 'flash'}`
	`});`
		Cross-domain requests via a Flash helper. No jQuery equivalent.

`$('#message').load('/ajax/test.html');`	`Y.one('#message').load('/ajax/test.html');` `Y.one('#message').load('/ajax/test.html', '#foo');`	Load the content of a given URL and replace the contents of `#message` with it. In YUI, the `node-load` module provides this functionality. YUI also optionally supports extracting only a portion of the loaded content if a selector string is passed as the second argument (assuming the content is HTML).

C S S

jQuery 1.4.2	YUI 3.4.1	Notes

`.addClass('foo')`
`.removeClass('foo')`
`.toggleClass('foo')`
`.hasClass('foo')`
	`.addClass('foo')`
	`.removeClass('foo')`
	`.toggleClass('foo')`
	`.hasClass('foo')`
		CSS class name manipulation.

`.removeClass('foo').addClass('bar')`	`.replaceClass('foo', 'bar')`	Replace node's CSS class 'foo' with 'bar'.
`.css('display', 'block')`	`.setStyle('display', 'block')`	Set a single CSS property

`.css({`
`  height:  100,`
`  width:  100,`
`  display: 'block'`
`})`
	`.setStyles({`
	`  height:  100,`
	`  width:  100,`
	`  display: 'block'`
	`})`
		Set multiple CSS properties with a dictionary.

`.css('display')`	`.getStyle('display')`	Get the current value for a CSS property.
`.height()` `.width()`	`.getComputedStyle('height')` `.getComputedStyle('width')`	Computed height / width. Excludes padding and borders.
`.innerHeight()` `.innerWidth()`	`.get('clientHeight')` `.get('clientWidth')`	Includes padding but not border
`.outerHeight()` `.outerWidth()`	`.get('offsetHeight')` `.get('offsetWidth')`	Includes padding and border
`.offset()` `// {left: 123, top: 456, width: 789, height: 1011}`	`.getXY()` `// [123, 456]`	Get the computed x,y coordinates relative to the document. jQuery also returns the size of the node
`.offset({ left: 123, top: 456 })`	`.setXY(123, 456)`	Set the x,y coordinates relative to the document.

L a n g u a g

jQuery 1.4.3	YUI 3.4.1	Notes
`$.each([1, 2, 3], fn(index, value))` `$.each({ key: value }, fn(key, value))`	`Y.Array.each([1, 2, 3], fn(value, index))` `Y.Object.each({ key: value }, fn(key, value))`	Iterate through an array or object. YUI is compatible with the Array forEach method so the first parameter the callback receives when iterating through an array is the value. In jQuery, the first parameter is the index of the value in the array.
`$.inArray(value, array)`	`Y.Array.indexOf(value, array)`	Returns the index of the searched value in the specified array.
`$.type(obj)`	`Y.Lang.type(obj)`	Returns a string representing the type of the specified object. YUI and jQuery results are compatible

jQuery 1.4.3	YUI 3.4.1	Notes
		(see jQuery's).
`$.isPlainObject(obj)`	`Y.Lang.isObject(obj)` `Y.Lang.isObject(obj, true)`	In YUI, `Y.Lang.isObject` returns true for arrays and functions. Passing true as a second parameter makes it return false if the input is a function.

`$.isArray(obj)`
`$.isFunction(obj)`
	`Y.Lang.isArray(obj)`
	`Y.Lang.isFunction(obj)`
	`Y.Lang.isString(obj)`
	`Y.Lang.isBoolean(obj)`
	`Y.Lang.isDate(obj)`
	`Y.Lang.isNumber(obj)`
	`Y.Lang.isNull(obj)`
	`Y.Lang.isUndefined(obj)`
	`Y.Lang.isValue(obj)`
		YUI also has some extra type checking functions. In particular, `Y.Lang.isValue()` returns false if the object is null, undefined or not a number, and true in any other case.

`$.isEmptyObject(obj)`	`Y.Object.isEmpty(obj)`	Check if the given object doesn't have any properties.
`$.makeArray(obj)`	`Y.Array(obj)`	Make an array-like object, for instance the return value of `document.getElementsByTagName`, into a true array.
`$.now()`	`Y.Lang.now()`	Return the current time in milliseconds.

C r e d i t s

The jQuery - YUI 3 Rosetta Stone was originally created by Carlos Bueno. It's now maintained by Ryan Grove and Paul Irish.

www.jsrosettastone.com