

**Jquery 1.1.3 Cheat Sheet 2007-08-20 <http://www.JavascriptToolbox.com/jquery/>**

Core		Selectors	
<b>\$(html) / \$(element) / \$(selector [,context])</b>	<b>\$(func) / \$(document).ready(func)</b>	E:nth-child(n) / E:first-child / E:last-child / E:only-child	
<b>each (func)</b>	<code>\$( "img" ).each(function(i){</code>	E:empty	has no children (including text nodes)
<b>\$(i) o/\$(i).get(i)</b>	<code>  this.src = "test" + i + ".jpg";</code>	E:enabled	E:disabled
<b>\$(i).length / \$(i).size()</b>	<code>  return false; // stop looping over each</code>	E:checked	E:selected
	<code>});</code>	E:#id	E.className
<b>Attributes</b>		E:not(s)	does not match simple selector s
<b>attr(name) / attr(name,val) / attr({name:val})</b>	<b>attr(name, func)</b>	E F	F element descendant of an E element
<b>removeAttr(name)</b>	<code>\$( "img" ).attr("title", function() { return this.src });</code>	E > F	F element child of an E element
<b>addClass(c) / removeClass(c) / toggleClass(c)</b>		E + F	F element immediately preceded by an E element
<b>html() / html(content)</b>	<b>text() / text(content)</b>	E ~ F	F element preceded by an E element
<b>Traversing</b>		E[@foo]	contains a "foo" attribute
<b>add(expr) / add(html) / add(Element)</b>	Add more elements to the set of matched elements	E[@foo=bar]	"foo" attribute value is exactly equal to "bar"
<b>contains(text)</b>	Leave only elements that contain 'text'	E[@foo^=bar]	"foo" attribute value begins exactly with "bar"
<b>filter(expr) / filter(func)</b>	Leave elements matching expr or func returning true	E[@foo\$=bar]	"foo" attribute value ends exactly with "bar"
<b>find(expr)</b>	<code>\$( "p" ).find("span") === \$( "p span" )</code>	E[@foo*=bar]	"foo" attribute value contains the substring "bar"
<b>is(expr [,expr])</b>	returns boolean	E[@foo!=bar]	"foo" attribute is not equal to "bar"
<b>next([expr])</b>	<b>prev([expr])</b>	E[@foo~=bar]	space-delimited "foo" attribute contains "bar"
<b>not(expr) / not(Element)</b>	Removes matched elements from list	E[@foo=bar][@baz=bop]	Match multiple attributes
<b>parent([expr]) / parents([expr])</b>	<b>children([expr])</b>	:gt(N) / :lt(N)	:eq(0) and :nth(0)
<b>siblings([expr])</b>	<b>end()</b>	:first / :last	:even / :odd
<b>DOM Manipulation</b>		:parent	elements which have child elements (including text).
<b>before(content) / after(content)</b>	Creates a new sibling before/after element	:contains('test')	elements which contain the specified text.
<b>prepend(content) / append(content)</b>	Creates a new child node at the beginning/end	:visible / :hidden	
<b>clone([deep])</b>	<b>remove()</b>	:input	All form elements, not just type=input
<b>empty()</b>	Removes all child nodes and content from E	Input Type Selectors:	
<b>wrap(html)</b>	<code>\$( "p" ).wrap("&lt;div class='wrap'&gt;&lt;/div&gt;");</code>	:password :radio :checkbox :submit :image :reset :button :file	
<b>CSS</b>		<b>Effects</b>	
<b>css(name)</b>	get val from first element in list only	Speed Values:	slow / normal / fast / #ms
<b>css(key,val) / css( {key:val} )</b>	<b>height() / height(val) / width() / width(val)</b>	<b>animate(params, speed, easing, callback)</b>	
<b>Events - return false from handlers to cancel default action</b>		<b>fadeIn/fadeOut(speed, callback)</b>	<b>fadeTo(speed, opacity, callback)</b>
<b>bind(type,data,func) / unbind(type,func)</b>	function handler(event) {	<b>hide/show(speed, callback)</b>	<b>slideDown/slideUp(speed, callback)</b>
<b>one(type, data, func) - execute only once</b>	<code>  alert(event.data.foo);</code>	<b>toggle()</b>	<b>slideToggle(speed,callback)</b>
<b>hover(overfunc, outfunc)</b>	}	<b>AJAX</b>	
<b>Event Types:</b>	<code>\$( "p" ).bind("click", {foo: "bar"}, handler)</code>	<b>\$.ajax( properties )</b>	async: boolean
blur, change, click, dblclick, error, focus, keydown, keypress, keyup, load, mousedown, mousemove, mouseout, mouseover, mouseup, ready, resize, scroll, select, submit, unload		<b>\$.ajaxSetup( properties )</b>	beforeSend: func
<b>toggle(eventfunc, oddfunc)</b>		<b>\$.get( url, properties, fn(data) )</b>	complete: func
<b>trigger(type, data)</b>	Executes browser's default action also	<b>\$.getIfModified(url,props,fn(data) )</b>	contentType: string
		<b>\$.getJSON(url,props,fn(json) )</b>	data: {obj}/String
<b>Misc</b>		<b>\$.getScript( url, callback )</b>	dataType: [xml,html,script,json]
<b>\$.browser.[safari, opera, msie, mozilla]</b>	<b>\$.browser.version</b>	<b>\$.post( url, props, fn(data) )</b>	error: func
<b>\$.each( obj, func )</b>	<b>\$.trim(str) - Trim a string</b>	<b>ajaxComplete( fn(xhr,props) )</b>	global: boolean
<code>\$.each( [0,1,2], function(i, n){</code>	<b>\$.extend( target, prop1, propN )</b>	<b>ajaxError( fn(xhr,props) )</b>	ifModified: boolean
<code>  alert( "Item #" + i + ": " + n );</code>	<code>var settings = { validate: false, limit: 5, name: "foo" };</code>	<b>ajaxSend( fn(xhr,props) )</b>	processData: boolean
<code>});</code>	<code>var options = { validate: true, name: "bar" };</code>	<b>ajaxStart( fn(xhr,props) )</b>	success: func
<code>\$.each( { name:"John",lang:"JS" },function(i,n){</code>	<code>jQuery.extend(settings, options);</code>	<b>ajaxStop( fn(xhr,props) )</b>	timeout: number
<code>  alert( "Name: " + i + ", Value: " + n );</code>	<code>settings == { validate: true, limit: 5, name: "bar" }</code>	<b>ajaxSuccess( fn(xhr,props) )</b>	type: [POST,GET]
<code>});</code>	<b>\$.map( array, func )</b>	<b>serialize()</b>	url: string
<b>\$.grep( array, func, invert)</b>	<code>\$.map( [0,1,2], function(n){</code>	<b>load/loadIfModified( url, props, fn(responseText, status, response) )</b>	
<code>  \$.grep( [0,1,2], function(n,i){</code>	<code>  return n + 4;</code>	<code>\$( "#feeds" ).load("feeds.php",</code>	
<code>  return n &gt; 0;</code>	<code>  });</code>	<code>  {limit: 25,</code>	
<code>  });</code>	<b>\$.merge( array, array ) - removes dupes</b>	<code>  function() { alert("The last 25 entries in the feed have been loaded"); }</code>	
Result = [ 1, 2 ]	<code>\$.merge( [0,1,2], [2,3,4] ) === [0,1,2,3,4]</code>	<code>  );</code>	