



SELECTORS
 Basics
#id
element
.class
.class.class

selector1
selector2

Basic Filters
:first
:last
:not(selector)
:even
:odd
:eq(index)
:gt(index)
:lt(index)
:header
:animated
:focus

Child Filters
:nth-child(expr)
:first-child
:last-child
:only-child

Forms
:input
:text
:password
:radio
:checkbox
:checkbox
:submit
:image
:reset
:button
:file

CORE
 jQuery function
jQuery
jQuery(selector [, context] | element | elementArray)
jQuery(html [, owner] | html, props)
jQuery(fn)
 def **.when(deferreds)**
 fn **jQuery.sub()**
\$.holdReady(hold)

jQuery Object Accessors
\$.each(fn(index, element))
 num **.size()**, **.length**
 str **.selector**
 el **.context**
\$.eq(index)
jQuery.error(str)
 [el,el] **.get(index)**
 num **.index()**, **.index(selector | element)**
jQuery.pushStack(elements, [name, args])
 arr **.toArray()**

Interoperability
jQuery.noConflict([extreme])

Attributes
 str **.attr(name | name, value)**
.attr(name, val | map | name, fn(index, attr))
\$.removeAttr(name)
\$.prop(name)
\$.removeProp(name)

Class
\$.addClass(class | fn(index, class))
 bool **.hasClass(class)**
\$.removeClass([class] | fn(index, class))
\$.toggleClass(class [, switch] | fn(index, class) | [switch])

HTML, text
 str **.html()**
\$.html(val | fn(index, html))
 str **.text()**
\$.text(val | fn(index, html))

Value
 str, arr **.val()**
\$.val(val | fn())

AJAX
 Low-Level Interface
 jqXHR **jQuery.ajax(options, [settings])**
 jQueryObject accepts
 bool async = true
 fn beforeSend(jqXHR, config) bool cache = true
 fn complete(jqXHR, status) obj context
 str contentType
 map converters
 obj data
 bool dataFilter(data, type)
 bool global = true
 bool ifModified = false
 fn jsonpCallback
 bool processData = true
 map statusCodes
 bool traditional
 str url = curr. page
 fn xhr
 str dataType = (xml, json, script, html)
 fn error(jqXHR, status, errorThrown)
 fn success(data, status, jqXHR)

Miscellaneous
 str **.serialize()**
 [obj] **.serializeArray()**
 str **jQuery.param(obj, [traditional])**

CSS
 str **.css(name)**
.css(name, val | map | name, fn(index, val))

Positioning
 obj **.offset()**
\$.offset(coord | fn(index, coord))
 obj **.offsetParent()**
 int **.position()**
 int **.scrollTop(val)**
 int **.scrollLeft(val)**
 int **.scrollLeft(val)**

Height and Width
 int **.height()**
\$.height(val | fn(index, height))
 int **.width()**
\$.width(val | fn(index, height))
 int **.innerHeight()**
 int **.innerWidth()**
 int **.outerHeight([includeMargin])**
\$.outerHeight(val | fn(index, outerHeight)) 1.8+
 int **.outerWidth([includeMargin])**
\$.outerWidth(val | fn(index, outerWidth)) 1.8+

Shorthand Methods
\$.load(url [data] [, fn(responseText, textStatus, XMLHttpRequest)])
 jqXHR **jQuery.get(url [data] [, fn(data, status, XMLHttpRequest)])**
 jqXHR **jQuery.getJSON(url [data] [, fn(data, status, XMLHttpRequest)])**
 jqXHR **jQuery.getScript(url [data, status])**
 jqXHR **jQuery.post(url [data] [, fn(data, status)] [, type])**

Global Ajax Event Handlers
\$.ajaxComplete(fn(event, XHR, options))
\$.ajaxError(fn(event, XHR, options, thrownError))
\$.ajaxSend(fn(event, XHR, options))
\$.ajaxStart(fn())
\$.ajaxStop(fn())
\$.ajaxSuccess(fn(event, XHR, options))

Events
 Page Load
\$.ready(fn())

Event Handling
\$.on(events [selector] [data], handler) 1.7+
\$.on(events-map [selector] [data]) 1.7+
\$.off(events [selector] [handler]) 1.7+
\$.off(events-map [selector]) 1.7+
\$.bind(type [data], fn(eventObj))
\$.bind(type [data], false)
\$.bind(array)
\$.unbind([type] [, fn])
\$.one(type [data], fn(eventObj))
\$.trigger(event [data])
 obj **.triggerHandler(event [data])**
\$.delegate(selector, type, [data], handler)
\$.undelegate([selector, type, [handler]] | selector, events | namespace)

Live Events
\$.live(eventType [data], fn())
\$.die(), **\$.die(eventType [data], fn())**

Interaction Helpers
\$.hover(fnIn(eventObj), fnOut(eventObj))
\$.toggle(fn(eventObj), fn2(eventObj) [...])

Event Helpers
 function ([data,] [fn])
\$.blur()
\$.change()
\$.click()
\$.dblclick()
\$.error()
\$.focus()
\$.focusin()
\$.focusout()
\$.keydown()
\$.keypress()
\$.keyup()
\$.load([data,] fn)
\$.mousedown()
\$.mouseenter()
\$.mouseleave()
\$.mousemove()
\$.mouseout()
\$.mouseover()
\$.mouseup()
\$.resize()
\$.scroll()
\$.select()
\$.submit()
\$.unload([data,] fn)

Effects
 Basics
\$.show([duration [easing] [, fn]])
\$.hide([duration [easing] [, fn]])
\$.toggle([showOrHide])
\$.toggle(duration [easing] [, fn])

Sliding
\$.slideDown(duration [easing] [, fn])
\$.slideUp(duration [easing] [, fn])
\$.slideToggle(duration [easing] [, fn])

Fading
\$.fadeIn(duration [easing] [, fn])
\$.fadeOut(duration [easing] [, fn])
\$.fadeTo([duration,] opacity [easing] [, fn])
\$.fadeToggle([duration,] [, easing] [, fn])

Custom
\$.animate(params [duration] [easing] [, fn])
\$.animate(params, options)
\$.stop([queue] [, clearQueue] [, jumpToEnd]) 1.7+
\$.delay(duration [queueName])

Settings
 bool **jQuery.fx.off**
 num **jQuery.fx.interval**

Traversing
 Filtering
\$.eq(index)
\$.first()
\$.last()
\$.has(selector), **\$.has(element)**
\$.filter(selector), **\$.filter(fn(index))**
 bool
\$.is(selector | function(index) | jQuery object | element) 1.7
\$.map(fn(index, element))
\$.not(selector), **\$.not(elements)**
\$.not(fn(index))
\$.slice(start [, end])

Tree traversal
\$.children([selector])
\$.closest(selector [context] | jQuery object | element)
 arr **\$.closest(selectors [context] removed)**
\$.find(selector | jQuery object | element)
\$.next([selector])
\$.nextAll([selector])
\$.nextUntil([selector])
\$.offsetParent()
\$.parent([selector])
\$.parents([selector])
\$.parentsUntil([selector])
\$.prev([selector])
\$.prevAll([selector])
\$.prevUntil([selector])
\$.siblings([selector])

Miscellaneous
\$.add(selector [context] | elements | html)
\$.andSelf()
\$.contents()
\$.end()

Manipulation
 Inserting Inside
\$.append(content | fn(index, html))
\$.appendTo(target)
\$.prepend(content | fn(index, html))
\$.prependTo(target)

Inserting Outside
\$.after(content | fn())
\$.before(content | fn())
\$.insertAfter(target)
\$.insertBefore(target)

Replacing
\$.replaceWith(content | fn)
\$.replaceAll(selector)

Removing
\$.detach([selector])
\$.empty()
\$.remove([selector])

Copying
\$.clone([withDataAndEvents] [, deepWithDataAndEvents])

Callbacks
 callbacks object = { 1.7+
 and **.add(callbacks)**
 and **.disable()**
 and **.empty()**
 and **.fire(arguments)**
 bool **.fired()**
 and **.fireWith([context] [, args])**
 bool **.has(callback)**
 and **.lock()**
 bool **.locked()**
 and **.remove(callbacks)**
\$.Callbacks(flags)
 def **\$.progress(progressCallbacks) 1.7+**
 def **\$.reject(args)**
 def **\$.rejectWith(context, [args])**
 def **\$.resolve(args)**
 def **\$.resolveWith(context, [args])**
 str **state() 1.8+**
 def **.notify(args) 1.7+**
 def **.notifyWith(context, [args]) 1.7+**
 def **.pipe([doneFilter] [, failFilter] [, progressFilter]) 1.7+**
 def **\$.Callbacks(flags)**
 def **\$.reject(args)**
 def **\$.rejectWith(context, [args])**
 def **\$.resolve(args)**
 def **\$.resolveWith(context, [args])**
 str **state()**
 def **\$.then(doneCallbacks, failCallbacks [, progressCallbacks]) 1.7+**
 def **\$.promise(target)**

Utilities
 Browser and Feature Detection
 obj **jQuery.support**
 obj **jQuery.browser** deprecated
 str **jQuery.browser.version** deprecated
 bool **jQuery.boxModel** deprecated

Basic operations
 obj **jQuery.each(obj, fn(i, valueOfElement))**
 obj **jQuery.extend([deep,] target, obj1 [, objN])**
 arr **jQuery.grep(arr, fn(el, i) [, invert])**
 arr **jQuery.makeArray(obj)**
 arr **jQuery.map(arrayOrObject, fn(el, i))**
 num **jQuery.inArray(val, arr)**
 arr **jQuery.merge(first, second)**
 fn **jQuery.noop**
 fn **jQuery.proxy(fn, scope | scope, name)**
 arr **jQuery.unique(arr)**
 str **jQuery.trim(str)**
 obj **jQuery.parseJSON(str)**

Data functions
\$.clearQueue(name)
\$.dequeue(name), **\$.jQuery.dequeue(name)**
 obj **jQuery.data(el, key)**
jQuery.data()
 obj **\$.data()**, **\$.data(key)**
\$.data(key, val | obj)
\$.removeData(name) [list] 1.7+
 [fn] **\$.queue(name)**
\$.jQuery.queue(name)
\$.queue(name, fn(next))
\$.queue(name, queue, fn())
\$.queue(name, queue)
\$.jQuery.queue(name, queue)

Test operations
 str **jQuery.type(obj)**
 bool **jQuery.isArray(obj)**
 bool **jQuery.isEmptyObject(obj)**
 bool **jQuery.isFunction(obj)**
 bool **jQuery.isPlainObject(obj)**
 bool **jQuery.isWindow(obj)**
 bool **jQuery.isNumeric(val) 1.7+**

Event object
 event = {
 el **currentTarget**,
data,
 bool **isDefaultPrevented()**,
 bool **isImmediatePropagationStopped()**,
 bool **isPropagationStopped()**,
 str **namespace**,
 num **pageX**,
 num **pageY**,
 el **preventDefault()**,
 el **relatedTarget**,
 obj **result**,
 def **stopImmediatePropagation()**,
stopPropagation(),
 el **target**,
 num **timeStamp**,
 str **type**,
 str **which**
 }

Deferred
 deferred object = {
 def **\$.always(alwaysCallbacks [, alwaysCallbacks])**
 def **\$.done(doneCallbacks)**
 def **\$.fail(failCallbacks)**
 bool **\$.isRejected()** deprecated
 bool **\$.isResolved()** deprecated
 str **state() 1.8+**
 def **.notify(args) 1.7+**
 def **.notifyWith(context, [args]) 1.7+**
 def **.pipe([doneFilter] [, failFilter] [, progressFilter]) 1.7+**
 def **\$.progress(progressCallbacks) 1.7+**
 def **\$.reject(args)**
 def **\$.rejectWith(context, [args])**
 def **\$.resolve(args)**
 def **\$.resolveWith(context, [args])**
 str **state()**
 def **\$.then(doneCallbacks, failCallbacks [, progressCallbacks]) 1.7+**
 def **\$.promise(target)**

Event object
 event = {
 el **currentTarget**,
data,
 bool **isDefaultPrevented()**,
 bool **isImmediatePropagationStopped()**,
 bool **isPropagationStopped()**,
 str **namespace**,
 num **pageX**,
 num **pageY**,
 el **preventDefault()**,
 el **relatedTarget**,
 obj **result**,
 def **stopImmediatePropagation()**,
stopPropagation(),
 el **target**,
 num **timeStamp**,
 str **type**,
 str **which**
 }

Event object
 event = {
 el **currentTarget**,
data,
 bool **isDefaultPrevented()**,
 bool **isImmediatePropagationStopped()**,
 bool **isPropagationStopped()**,
 str **namespace**,
 num **pageX**,
 num **pageY**,
 el **preventDefault()**,
 el **relatedTarget**,
 obj **result**,
 def **stopImmediatePropagation()**,
stopPropagation(),
 el **target**,
 num **timeStamp**,
 str **type**,
 str **which**
 }