

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
```

Child Filters

```
:nth-child(expr)
:first-child
:last-child
:only-child
```

Forms

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

Core

jQuery function

```
$ .jQuery( selector [, context] ), .jQuery( element ),
  .jQuery( elementArray ), .jQuery( jQueryObject ),
  .jQuery( )
$ .jQuery( html [, ownerDocument] ),
  .jQuery( html, props )
$ .jQuery( fn )
```

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 )index( element )' )
$ .jQuery.pushStack( elements, [name, arguments] )
arr .toArray( )
```

Interoperability

```
$ .jQuery.noConflict( [extreme] )
```

Hierarchy

```
ancestor descendant
parent > child
prev + next
prev ~ siblings
```

Content Filters

```
:contains(text)
:empty
:has(selector)
:parent
```

Visibility Filters

```
:hidden
:visible
```

Attribute Filters

```
[attribute]
[attribute=value]
[attribute!=value]
[attribute^=value]
[attribute$=value]
[attribute*=value]
[attribute|=value]
[attribute~value]
[attribute]
[attribute2]
```

Form Filters

```
:enabled
:disabled
:checked
:selected
```

Attributes

Attributes

```
str .attr( name )
$ .attr( name, val ), .attr( map ),
  .attr( name, fn(index, attr) )
$ .removeAttr( name )
```

Class

```
$ .addClass( class ),
  .addClass( fn(index, class) )
bool .hasClass( class )
$ .removeClass( [class] ),
  .removeClass( fn(index, class) )
$ .toggleClass( class [, switch] ),
  .toggleClass( fn(index, class) [, switch] )
```

HTML, text

```
str .html( )
$ .html( val ), .html( fn(index, html) )
str .text( )
$ .text( val ), .text( fn(index, html) )
```

Value

```
str,arr .val( )
$ .val( val ), .val( fn( ) )
```

CSS

CSS

```
str .css( name )
$ .css( name, val ), .css( map ),
  .css( name, fn(index, val) )
```

Positioning

```
obj .offset( )
$ .offset( coord ),
  .offset( fn( index, coord ) )
$ .offsetParent( )
obj .position( )
int .scrollTop( )
$ .scrollTop( val )
int .scrollLeft( )
$ .scrollLeft( val )
```

Height and Width

```
int .height( )
$ .height( val ),
  .height( fn(index, height) )
int .width( )
$ .width( val ), .width( fn(index, height) )
int .innerHeight( )
int .innerWidth( )
int .outerHeight( [margin] )
int .outerWidth( [margin] )
```

Deferred

```
deferred object = {
  def done(doneCallbacks)
  def fail(failCallbacks)
  bool isRejected()
  bool isResolved()
  def reject([args])
  def rejectWith(context, [args])
  def resolve([args])
  def resolveWith(context, [args])
  def then(doneCallbacks, failCallbacks)
} 1.5+
def .when(deferreds) 1.5+
```

Traversing

Filtering

```
$ .eq( index )
$ .first( )
$ .last( )
$ .has( selector ), .has( element )
$ .filter( selector ), .filter( fn(index) )
bool .is( selector )
$ .map( fn(index, element) )
$ .not( selector ), .not( elements ), .not( fn( index ) )
$ .slice( start [, end] )
```

Tree traversal

```
$ .children( [selector] )
$ .closest( selector [, context] )
arr .closest( selectors [, context] )
$ .find( selector )
$ .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] ), .add( elements ), .add( html )
$ .andSelf( )
$ .contents( )
$ .end( )
```

Manipulation

Inserting Inside

```
$ .append( content ), .append( fn( index, html ) )
$ .appendTo( target )
$ .prepend( content ), .prepend( fn( index, html ) )
$ .prependTo( target )
```

Inserting Outside

```
$ .after( content ), .after( fn( ) )
$ .before( content ), .before( fn( ) )
$ .insertAfter( target )
$ .insertBefore( target )
```

Inserting Around

```
$ .unwrap( )
$ .wrap( wrappingElement ), .wrap( fn )
$ .wrapAll( wrappingElement ), .wrapAll( fn )
$ .wrapInner( wrappingElement ), .wrapInner( fn )
```

Replacing

```
$ .replaceWith( content ), .replaceWith( fn )
$ .replaceAll( selector )
```

Removing

```
$ .detach( [selector] )
$ .empty( )
$ .remove( [selector] )
```

Copying

```
$ .clone( [withDataAndEvents], [deepWithDataAndEvents] ) 1.5+
```

Events

Page Load

```
$ .ready( fn( ) )
```

Event Handling

```
$ .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]] )
```

Live Events

```
$ .live( eventType [, data], fn( ) )
$ .die( ), .die( eventType [, fn( )] )
```

Interaction Helpers

```
$ .hover( fnIn(eventObj), fnOut(eventObj) )
$ .toggle( fn(eventObj), fn2(eventObj) [, ...] )
```

Event Helpers

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

Event object

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

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( [clearQueue] [, jumpToEnd] )
$ .delay( duration [, queueName] )
```

Settings

```
bool jQuery.fx.off
num jQuery.fx.interval
```

AJAX

Low-Level Interface

```
jqXHR jQuery.ajax( options, [settings] ) 1.5+
{
  map accepts bool async = true
  fn beforeSend( jqXHR, config )
  bool cache = true
  fn complete( jqXHR, status )
  map contents str contentType
  obj context map converters
  bool crossDomain obj, str data
  fn dataFilter( data, type )
  bool global = true map headers
  bool ifModified = false str jsonp
  fn jsonpCallback str password
  bool processData = true
  str scriptCharset map statusCode
  num timeout bool traditional
  str type = 'GET' str url = curr. page
  str username fn xhr
  str dataType ∈ {xml, json, script, html}
  fn error( jqXHR, status, errorThrown )
  fn success( data, status, jqXHR )
}
jQuery.ajaxSetup( options )
```

Shorthand Methods

```
$ .load( url [, data] [, fn(.responseText, status, XHR) ] )
jqXHR jQuery.get( url [, data] [, fn( data, status, XHR) ] [, type] )
jqXHR jQuerygetJSON( url [, data] [, fn( data, status ) ] )
jqXHR jQuery.getScript( url [, fn( 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, errorThrown ) )
$ .ajaxSend( fn( event, XHR, options ) )
$ .ajaxStart( fn( ) )
$ .ajaxStop( fn( ) )
$ .ajaxSuccess( fn( event, XHR, options ) )
```

Miscellaneous

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

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( index, valueOfElement ) )
obj jQuery.extend( [deep,] target, obj1 [, objN] )
arr jQuery.grep( array, fn( element, index ) [, invert] )
arr jQuery.makeArray( obj )
arr jQuery.map( array, fn( element, index ) )
num jQuery.inArray( val, array )
```


```
arr jQuery.merge( first, second )
fn jQuery.noop
fn jQuery.proxy( fn, scope ), jQuery.proxy( scope, name )
fn jQuery.sub( ) 1.5+
arr jQuery.unique( array )
str jQuery.trim( str )
obj jQuery.parseJSON( str )
```

Data functions

```
$ .clearQueue( [name] )
$ .dequeue( [name] ), jQuery.dequeue( [name] )
obj jQuery.data( element, key ), jQuery.data( )
obj .data( ), .data( key )
$ .data( key, val ), .data( obj )
$ .removeData( [name] )
[fn] .queue( [name] ) jQuery.queue( [name] )
$ .queue( [name,] fn( next ) ),
  jQuery.queue( [name,] 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 )
```

http://
ver. 04-02-2011

jQuery 1.5 API Cheat Sheet