

Selectors

Basics

```
#id
element
.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
```

Legend

Conventional signs

[obj] — array of objects
1.4+ — new in 1.4
1.4* — changed in 1.4

Data types

* — anything
\$ — jQuery object
arr — array
bool — boolean
el — DOM element
fn — function
int — integer
obj — object
str — string
XHR — XMLHttpRequest

<http://futurecolors.ru/jquery/>
ver. 22-10-2010

Core

jQuery function

```
$ .jQuery( selector [, context] ), .jQuery( element ),
  .jQuery( elementArray ),
  .jQuery( jQueryObject ), .jQuery( ) 1.4*
$ .jQuery( html [, ownerDocument] ),
  .jQuery( html, props ) 1.4*
$ .jQuery( fn )
```

jQuery Object Accessors

```
$ .each( fn(index, element) )
num .size( ), .length
str .selector
el .context
$ .eq( index )
  .jQuery.error( str ) 1.4+
[el],el .get( [index] )
num .index( ) 1.4*, .index( selector ) 1.4*,
  .index( element )
$ .jQuery.pushStack( elements, [name, arguments] )
arr .toArray( ) 1.4+
```

Interoperability

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

Attributes

Attributes

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

Class

```
$ .addClass( class ),
  .addClass( fn(index, class) ) 1.4*
bool .hasClass( class )
$ .removeClass( fn(index, class) ) 1.4*
$ .toggleClass( class [, switch] ),
  .toggleClass( fn(index, class) [, switch] ) 1.4*
```

HTML, text

```
str .html( )
$ .html( val ), .html( fn(index, html) ) 1.4*
str .text( )
$ .text( val ), .text( fn(index, html) ) 1.4*
```

Value

```
str, arr .val( )
$ .val( val ), .val( fn() ) 1.4*
```

CSS

CSS

```
str .css( name )
$ .css( name, val ), .css( map ),
  .css( name, fn(index, val) ) 1.4*
```

Positioning

```
obj .offset( )
$ .offset( coord ), .offset( fn( index, coord ) ) 1.4+
$ .offsetParent( )
obj .position( )
int .scrollTop( )
$ .scrollTop( val )
int .scrollLeft( )
$ .scrollLeft( val )
```

Height and Width

```
int .height( )
$ .height( val ), .height( fn(index, height) ) 1.4*
int .width( )
$ .width( val ), .width( fn(index, height) ) 1.4*
int .innerHeight( )
int .innerWidth( )
int .outerHeight( [margin] )
int .outerWidth( [margin] )
```

Traversing

Filtering

```
$ .eq( index )
$ .first( ) 1.4+
$ .last( ) 1.4+
$ .has( selector ),
  .has( element ) 1.4+
$ .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] ) 1.4*
arr .closest( selectors [, context] ) 1.4+
$ .find( selector )
$ .next( [selector] )
$ .nextAll( [selector] )
$ .nextUntil( [selector] ) 1.4+
$ .offsetParent( )
$ .parent( [selector] )
$ .parents( [selector] )
$ .parentsUntil( [selector] ) 1.4+
$ .prev( [selector] )
$ .prevAll( [selector] )
$ .prevUntil( [selector] ) 1.4+
$ .siblings( [selector] )
```

Miscellaneous

```
$ .add( selector [, context] ),
  .add( elements ), .add( html ) 1.4*
$ .andSelf( )
$ .contents( )
$ .end( )
```

Manipulation

Inserting Inside

```
$ .append( content ),
  .append( fn( index, html ) ) 1.4*
$ .appendTo( target )
$ .prepend( content ),
  .prepend( fn( index, html ) ) 1.4*
$ .prependTo( target )
```

Inserting Outside

```
$ .after( content ), .after( fn() ) 1.4*
$ .before( content ),
  .before( fn() ) 1.4*
$ .insertAfter( target )
$ .insertBefore( target )
```

Inserting Around

```
$ .unwrap( ) 1.4+
$ .wrap( wrappingElement ),
  .wrap( fn ) 1.4*
$ .wrapAll( wrappingElement ),
  .wrapAll( fn ) 1.4*
$ .wrapInner( wrappingElement ),
  .wrapInner( fn ) 1.4*
```

Replacing

```
$ .replaceWith( content ),
  .replaceWith( fn ) 1.4*
$ .replaceAll( selector )
```

Removing

```
$ .detach( [selector] ) 1.4+
$ .empty( )
$ .remove( [selector] )
```

Copying

```
$ .clone( [withDataAndEvents] )
```

Events

Page Load

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

Event Handling

```
$ .bind( type [, data ], fn(eventObj) )
$ .bind( type [, data ], false ) 1.4*
$ .bind( array ) 1.4*
$ .unbind( [type] [, fn] )
$ .one( type [, data ], fn(eventObj) )
$ .trigger( event [, data] )
obj .triggerHandler( event [, data] )
$ .delegate( selector, type, [data], handler ) 1.4+
$ .undelegate( [selector, type, [handler]] ) 1.4+
```

Live Events

```
$ .live( eventType [, data ], fn() )
$ .die( ) 1.4+, .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, 1.4+ .mouseup,
  .focusout, 1.4+ .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, 1.4+
  num pageX,
  num pageY,
  preventDefault(),
  el relatedTarget,
  obj result,
  stopImmediatePropagation(),
  stopPropagation(),
  el target,
  num timeStamp,
  str type,
  str which
}
```

Effects

Basics 1.4*

```
$ .show( [ duration ], [ easing ] [, fn] )
$ .hide( [ duration ], [ easing ] [, fn] )
$ .toggle( [showOrHide] )
$ .toggle( duration [, easing] [, fn] )
```

Sliding 1.4*

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

Fading 1.4*

```
$ .fadeIn( duration [, easing] [, fn] )
$ .fadeOut( duration [, easing] [, fn] )
$ .fadeTo( [duration], opacity [, easing] [, fn] )
```

Custom

```
$ .animate( params [, duration] [, easing] [, fn] )
$ .animate( params, options )
$ .stop( [clearQueue] [, jumpToEnd] )
$ .delay( duration [, queueName] ) 1.4+
```

Settings

```
bool jQuery.fx.off
num jQuery.fx.interval 1.4+
```

AJAX

Low-Level Interface

```
XHR jQuery.ajax( options )
bool async = true fn beforeSend( XHR )
bool cache = true fn complete( XHR, status )
str contentType obj context
obj, str data fn dataFilter( data, type )
bool global = true bool ifModified = false
str jsonp fn jsonpCallback
str password bool processData = true
num timeout str type = 'GET'
str url = curr. page str username
fn xhr str scriptCharset
str dataType ∈ {xml, json, script, html}
fn error( XHR, status, errorThrown )
fn success( data, status, XHR )
jQuery.ajaxSetup( options )
```

Shorthand Methods

```
$ .load( url [, data ], [fn( responseText, status, XHR )] )
XHR jQuery.get( url [, data ], [fn( data, status, XHR )] [, type] )
XHR jQuery.getJSON( url [, data ], [fn( data, status )] )
XHR jQuery.getScript( url [, fn( data, status )] )
XHR 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 ) )
```

Miscellaneous

```
str .serialize( )
[obj] .serializeArray( )
str jQuery.param( obj, [traditional] ) 1.4*
```

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 1.4+
fn jQuery.proxy( fn, scope ), jQuery.proxy( scope, name ) 1.4+
arr jQuery.unique( array )
str jQuery.trim( str )
obj jQuery.parseJSON( str ) 1.4+
```

Data functions

```
$ .clearQueue( [name] ) 1.4+
$ .dequeue( [name] ), jQuery.dequeue( [name] )
obj jQuery.data( element, key ), jQuery.data( ) 1.4+
obj .data( ), .data( key )
$ .data( key, val ), .data( obj ) 1.4*
$ .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 ) 1.4+
bool jQuery.isFunction( obj )
bool jQuery.isPlainObject( obj ) 1.4+
bool jQuery.isWindow( obj ) 1.4+
```

jQuery 1.4.3 API Cheat Sheet