

LOAD & READY

```
<script src="https://code.jquery.com/jquery-3.7.1.min.js" integrity="sha256-..." crossorigin="anonymous"></script>

$(function () { /* DOM ready */ });
$(document).ready(fn); // same
jQuery.noConflict(); // release $
```

slim build = no AJAX / effects shims.

SELECTING

<code>\$('p.intro')</code>	type / #id / .class / *
<code>\$('ul > li')</code>	child; a b descendant
<code>\$('h2 + p')</code>	adjacent; ~ general sibling
<code>\$('a[href^="https"]')</code>	<code>^= \$= *= ~=</code> attr match
<code>:first :last :eq(n)</code>	jq positional; <code>:gt</code> <code>:lt</code> <code>:even</code> <code>:odd</code>
<code>:not() :has()</code>	jq content filters
<code>:contains()</code>	
<code>:input :checkbox :selected</code>	jq form; <code>:checked</code> <code>:disabled</code>
<code>:visible :hidden</code>	jq layout state
<code>\$.sel.length</code>	0 = matched nothing

jq = jQuery-only, not valid CSS (slow path).

TRAVERSING

<code>.find(s)</code>	down
<code>.children(s)</code>	
<code>.parent()</code>	up; <code>.parentsUntil(s)</code>
<code>.parents(s)</code>	
<code>.closest(s)</code>	nearest ancestor incl. self
<code>.siblings(s)</code>	all siblings
<code>.next() .nextAll()</code>	forward; <code>.prev*</code> backward
<code>.nextUntil()</code>	
<code>.filter() .not()</code>	narrow a set
<code>.is()</code>	
<code>.eq(n) .first()</code>	by index
<code>.last() .slice()</code>	
<code>.each((i,el)=>{})</code>	<code>this</code> = raw el; args (i, el)
<code>.map()</code>	
<code>.end() .addBack()</code>	pop / merge the prior set

CONTENT & DOM

<code>.html() .text()</code>	get(first) / set(all); <code>.text</code> escapes
<code>.val()</code>	
<code>.append() .prepend()</code>	inside; <code>.appendTo()</code> swapped
<code>.before() .after()</code>	outside; <code>.insertBefore()</code> swapped
<code>.wrap() .wrapAll()</code>	wrapping
<code>.unwrap()</code>	
<code>.remove()</code>	drop + free data/handlers
<code>.detach()</code>	drop, keep data/handlers (re-insert)
<code>.empty()</code>	delete children only
<code>.clone(true)</code>	<code>true</code> = copy handlers too
<code>.replaceWith()</code>	
<code>\$('', {text, class, click})</code>	create with props object

CSS / CLASS / ATTR / SIZE

<code>.addClass()</code>	<code>.toggleClass(n, on?)</code>
<code>.removeClass()</code>	<code>.hasClass()</code>
<code>.css('color')</code>	get one / set object / fn setter
<code>.css({...})</code>	
<code>.attr()</code>	HTML attribute (default state)
<code>.prop()</code>	live DOM prop: checked selected disabled
<code>.width() .height()</code>	content box, unitless Number
<code>.innerWidth()</code>	+ padding
<code>.outerWidth(true?)</code>	+ border (+ margin if true)
<code>.offset()</code>	vs document / vs offset parent
<code>.position()</code>	
<code>.scrollTop(v?)</code>	get / set scroll
<code>.scrollLeft(v?)</code>	

EFFECTS

<code>.show() .hide()</code>	add (ms, done) to animate
<code>.toggle()</code>	
<code>.fadeIn/Out/Toggle(ms)</code>	<code>.fadeTo(ms, 0.4)</code>
<code>.slideDown/Up/Toggle(ms)</code>	animate height
<code>.animate({num}, ms, easing, done)</code>	<code>+= -=</code> 'toggle'; swing/linear
<code>.stop(clear?, jump?)</code>	control the fx queue
<code>.finish()</code>	
<code>.delay(ms)</code>	pause between queued effects
<code>jQuery.fx.off = true</code>	honour reduced-motion

Each element has its own FIFO effects queue.

EVENTS

<code>.on('click', fn)</code>	the one binder; <code>.off()</code> <code>.one()</code>
<code>.on('click', 'li', fn)</code>	delegation : future rows too; <code>this=li</code>
<code>.on('click', {id}, fn)</code>	data → <code>e.data.id</code>
<code>.trigger('x', [args])</code>	<code>.triggerHandler()</code> = no bubble/default
<code>'click.ns'</code>	namespaced; <code>.off('ns')</code>
event obj	<code>.type</code> <code>.target</code> <code>.currentTarget</code> <code>.which</code> <code>.key</code> <code>.pageX</code> <code>.data</code>
stop	<code>.preventDefault()</code> <code>.stopPropagation()</code> ; return false = both

Shorthands `.click()` etc.; no-arg trigger form deprecated (3.3+).

DATA & UTILITIES

<code>.data('k', v)</code>	cache; reads <code>data-*</code> once, no write-back
<code>.data('k') .data()</code>	
<code>.index() .index(sel)</code>	position among siblings / in set
<code>.get(i) .toArray()</code>	raw DOM nodes; <code>.get(-1)</code> = last
<code>\$.set[i]</code>	
<code>\$.each \$.map \$.grep</code>	<code>\$.extend(true, t, s)</code> = deep
<code>\$.extend</code>	
<code>\$.merge \$.inArray</code>	<code>\$.fn.jquery</code> = version
<code>\$.now</code>	
deprecated	<code>\$.trim</code> <code>\$.type</code> <code>\$.isArray</code> <code>\$.parseJSON</code> → native

AJAX

```
jQuery.ajax('/api', {
  method: 'POST', data: {x:1}, dataType: 'json',
  headers: {}, timeout: 8000, beforeSend: fn })
  .done((data)=>{}).fail((jqXHR)=>{}).always(fn);
```

<code>\$.get</code> <code>\$.post</code>	shorthands
<code>\$.getJSON</code>	
<code>\$.getScript</code>	
<code>\$('#p').load('url #sel')</code>	inject HTML fragment
<code>.then()</code>	Promises/A+ since JQ 3; <code>await</code> ok
<code>\$.when(a, b).done()</code>	combine; <code>\$.Deferred()</code>
<code>.serialize()</code>	<code>\$.param(obj)</code>
<code>.serializeArray()</code>	
<code>ajaxStart</code> <code>ajaxStop</code>	global, bound on document
<code>ajaxError</code>	

PLUGIN SKELETON

```
(function ($) {
  $.fn.highlight = function (options) {
    var o = $.extend({}, $.fn.highlight.defaults, options);
    return this.each(function () {
      $(this).css('background', o.color);
    });
  };
  $.fn.highlight.defaults = { color: '#ff0' };
})(jQuery);
```

Iterate with `.each`, return `this`, namespace handlers.

JQUERY → NATIVE

<code>\$(sel)</code>	<code>document.querySelector(sel)</code>
<code>.find(sel)</code>	<code>el.querySelector(sel)</code>
<code>.addClass('x')</code>	<code>el.classList.add('x')</code>
<code>.on('click', fn)</code>	<code>el.addEventListener('click', fn)</code>
<code>.on('click', 'li', fn)</code>	<code>e.target.closest('li')</code> inside listener
<code>.append(n) .remove()</code>	same-named native methods
<code>.css('c','red')</code>	<code>el.style.color = 'red'</code>
<code>.each(fn)</code>	for (const el of list)
<code>\$.ajax / \$.getJSON</code>	<code>fetch()</code> + <code>AbortController</code>
<code>\$(fn)</code>	<code><script defer></code> / <code>DOMContentLoaded</code>

Migrate with the official `jquery-migrate` plugin.