

# jQuery Riferimento Rapido

Selettori, DOM, eventi, AJAX, effetti, navigazione

## Selettori

### Selettori di Base

```
$("#p") // all <p> elements
$(".btn") // class selector
$("#main") // ID selector
$("ul > li") // direct children
$("input[type='text']") // attribute selector
```

### Selettori Comuni

<code>\$('.*')</code>	Tutti gli elementi
<code>\$('.a, span')</code>	Selettori multipli
<code>\$('.:first')</code>	Primo elemento corrispondente
<code>\$('.:last')</code>	Ultimo elemento corrispondente
<code>\$('.:even') / \$('.:odd')</code>	Elementi con indice pari/dispari
<code>\$('.:visible') / \$('.:hidden')</code>	Filtri di visibilità
<code>\$('.:contains(text)')</code>	Elementi che contengono testo
<code>\$('.:has(selector)')</code>	Elementi che contengono un corrispondente

## Manipolazione DOM

### Contenuto e Struttura

```
$("#el").html("<b>Bold</b>"); // set inner HTML
$("#el").text("Plain text"); // set text content
$("ul").append("<li>New</li>"); // add at end
$("ul").prepend("<li>First</li>"); // add at start
```

### Inserimento e Rimozione

```
$("#<p>Hello</p>").insertAfter("#el"); // insert after
$("#<p>Hi</p>").insertBefore("#el"); // insert before
$("#el").remove(); // remove element from DOM
$("#el").empty(); // remove all children
```

## Metodi

<code>.html()</code>	Ottieni/imposta HTML interno
<code>.text()</code>	Ottieni/imposta contenuto testuale
<code>.append()</code> / <code>.prepend()</code>	Aggiungi figlio alla fine/inizio
<code>.after()</code> / <code>.before()</code>	Inserisci fratello dopo/prima
<code>.wrap()</code>	Avvolgi l'elemento con un nuovo genitore
<code>.clone()</code>	Copia profonda dell'elemento
<code>.replaceWith()</code>	Sostituisce l'elemento

## Eventi

### Associazione di Eventi

```
$("#.btn").on("click", function () {
  $(this).toggleClass("active");
});
$(document).on("click", ".dynamic", handler); // delegation
$("#.btn").off("click"); // remove handler
```

### Metodi Abbreviati

<code>.click(fn)</code>	Gestore click
<code>.hover(fnIn, fnOut)</code>	Entrata/uscita del mouse
<code>.focus()</code> / <code>.blur()</code>	Focus guadagnato/perso
<code>.change(fn)</code>	Valore cambiato (input/select)
<code>.submit(fn)</code>	Invio del form
<code>.keydown()</code> / <code>.keyup()</code>	Eventi tastiera

### Document Ready

```
$(document).ready(function () { /* DOM loaded */ });
$(function () { /* shorthand */ });
```

## Effetti e Animazioni

### Mostra / Nascondi / Alterna

```
$("#el").hide(400); // hide with animation
$("#el").show(400); // show with animation
$("#el").toggle(); // toggle visibility
$("#el").fadeIn(600); // fade in
$("#el").slideDown(400); // slide down
```

### Animazione Personalizzata

```
$("#el").animate({
  opacity: 0.5,
  left: "+=50px",
  height: "toggle"
}, 1000, function () { /* complete */ });
```

### Metodi degli Effetti

<code>.fadeIn()</code> / <code>.fadeOut()</code>	Anima l'opacità
<code>.fadeToggle()</code>	Alterna con dissolvenza
<code>.slideDown()</code> / <code>.slideUp()</code>	Anima l'altezza
<code>.animate(props, ms)</code>	Animazione CSS personalizzata
<code>.stop()</code>	Ferma l'animazione corrente
<code>.delay(ms)</code>	Ritarda il prossimo effetto in coda

## AJAX

### Metodi Abbreviati

```
$.get("/api/data", function (data) { console.log(data); });
$.post("/api/data", { name: "Alice" }, function (res) {});
$.getJSON("/api/items", function (json) {});
$("#el").load("/fragment.html"); // load HTML into element
```

### \$.ajax Completo

```
$.ajax({
  url: "/api/data",
  method: "POST",
  data: JSON.stringify({ key: "val" }),
  contentType: "application/json",
  success: (res) => console.log(res),
  error: (xhr) => console.error(xhr.status)
});
```

### Opzioni AJAX

<code>url</code>	URL della richiesta
<code>method</code>	Metodo HTTP (GET, POST, ...)
<code>data</code>	Dati da inviare
<code>dataType</code>	Tipo di risposta atteso
<code>contentType</code>	Tipo di contenuto della richiesta
<code>success</code> / <code>error</code>	Funzioni di callback
<code>headers</code>	Header di richiesta personalizzati

## Navigazione

### Navigazione nell'Albero

```
$("#li").parent(); // direct parent
$("#li").parents("ul"); // all matching ancestors
$("#ul").children("li"); // direct children
$("#ul").find(".active"); // all matching descendants
$("#li").siblings(); // sibling elements
```

## Metodi di Navigazione

<code>.parent()</code> / <code>.parents()</code>	Genitore diretto / tutti gli antenati
<code>.children()</code> / <code>.find()</code>	Figli diretti / tutti i discendenti
<code>.siblings()</code>	Tutti gli elementi fratelli
<code>.next()</code> / <code>.prev()</code>	Fratello successivo/precedente
<code>.closest(sel)</code>	Antenato più vicino che corrisponde al selettore
<code>.first()</code> / <code>.last()</code>	Primo/ultimo nel set corrispondente
<code>.eq(index)</code>	Elemento all'indice
<code>.filter(sel)</code>	Riduce agli elementi corrispondenti

## Attributi e CSS

### Attributi

```
$("#img").attr("src"); // get attribute
$("#img").attr("src", "new.jpg"); // set attribute
$("#img").removeAttr("title"); // remove attribute
$("input").prop("checked", true); // set property
```

### CSS e Classi

```
$(".box").css("color", "red"); // set single style
$(".box").css({ color: "red", fontSize: "14px" }); // multi
$("#p").addClass("highlight");
$("#p").removeClass("highlight");
$("#p").toggleClass("active");
$("#p").hasClass("active"); // returns boolean
```

## Form

### Valori e Serializzazione

```
$("#input").val(); // get value
$("#input").val("new value"); // set value
$("#form").serialize(); // "name=Alice&age=30"
$("#form").serializeArray(); // [{name, value}, ...]
```

### Selettori per Form

<code>\$("#:input")</code>	Tutti gli elementi del form
<code>\$("#:checked")</code>	Checkbox/radio selezionati
<code>\$("#:selected")</code>	Elementi option selezionati
<code>\$("#:disabled")</code>	Elementi disabilitati
<code>\$("#:text")</code>	Input di testo

## Utilità

### Funzioni per Array e Oggetti

```
$.each([1, 2, 3], function (i, val) { /* iterate */ });
$.each(obj, function (key, val) { /* iterate object */ });
$.extend({}, defaults, options); // merge objects
$.inArray("b", ["a", "b", "c"]); // returns 1 (index)
```

### Metodi Utili

<code>\$.each(arr, fn)</code>	Itera array o oggetto
<code>\$.extend(target, ...)</code>	Unisce oggetti (shallow)
<code>\$.inArray(val, arr)</code>	Indice del valore (-1 se mancante)
<code>\$.isArray(obj)</code>	Verifica se è un array
<code>\$.trim(str)</code>	Rimuove spazi iniziali/finali
<code>\$.parseJSON(str)</code>	Analizza una stringa JSON
<code>\$.map(arr, fn)</code>	Trasforma gli elementi dell'array

# jQuery Riferimento Rapido

---

## Pattern Comuni

---

### Chaining

---

```
$( "ul" )  
  .find( "li" )  
  .addClass( "item" )  
  .first()  
  .css( "font-weight", "bold" );
```

### Pattern Plugin

---

```
$.fn.highlight = function (color) {  
  return this.each(function () {  
    $(this).css("background", color || "yellow");  
  });  
};  
$("p").highlight("pink");
```

### Risoluzione Conflitti

---

```
// Avoid $ conflicts with other libraries  
jQuery.noConflict();  
jQuery(document).ready(function ($) {  
  $(".btn").click(handler); // $ safe inside  
});
```