

Guida completa a jQuery

Stefanoo94

Contents

Guida rapida a jQuery	1
Scopo	1
Prerequisiti	1
1. Aggiungere jQuery al progetto	1
2. Selettori e accesso al DOM	2
3. Manipolazione del DOM	2
4. Eventi	2
5. Animazioni ed effetti	3
6. AJAX	3
7. Plugin e estensioni	4
8. NoConflict	4
9. Performance e best practice	4
10. Testing	4
11. Migrazione da jQuery a Vanilla / Framework moderno	4
12. Integrazione in progetti moderni (Webpack / Vite)	4
13. Sicurezza	5
14. Risorse utili	5
Esempio minimo (script)	5

Guida rapida a jQuery

Versione: 1.0

Autore: Stefanoo94

Scopo

Questa guida fornisce i comandi e gli esempi pratici per iniziare e lavorare con jQuery: selettori, manipolazione del DOM, eventi, AJAX, effetti, plugin, best practice e indicazioni per integrazione e build.

Prerequisiti

- Conoscenze base di HTML, CSS e JavaScript
- Browser moderni o ambiente che supporti ES5
- jQuery (incluso via CDN o installato con npm/yarn)

1. Aggiungere jQuery al progetto

Via CDN (semplice):

```
<script src="https://code.jquery.com/jquery-3.6.0.min.js" integrity="sha256-/xUj+30J..." crossorigin="a
```

Con npm/yarn (per bundler):

```
npm install jquery  
# oppure  
yarn add jquery
```

E nell'entry JavaScript (es. con Webpack/Vite):

```
import $ from 'jquery';  
window.$ = window.jQuery = $; // se necessario per plugin legacy
```

2. Selettori e accesso al DOM

jQuery usa selettori simili a CSS:

```
const $items = $('.item');           // class  
const $button = $('#myButton');      // id  
const $lis = $('ul > li');           // combinatori
```

Document ready:

```
$(function() {  
    // codice eseguito dopo che il DOM è pronto  
});
```

3. Manipolazione del DOM

Lettura/scrittura testo/HTML:

```
$('#title').text('Nuovo titolo');  
$('#content').html('<strong>HTML</strong>');
```

Gestione attributi e proprietà:

```
$('#img').attr('src', 'img.jpg');  
$('#checkbox').prop('checked', true);
```

Aggiungere/rimuovere elementi:

```
$('#<li>Nuovo</li>').appendTo('#list');  
$('#item-to-remove').remove();
```

Classi:

```
$('#box').addClass('active');  
$('#box').removeClass('active');  
$('#box').toggleClass('hidden');
```

4. Eventi

Sintassi base:

```
$('#btn').on('click', function(e) {  
  e.preventDefault();  
  // ...  
});
```

Delegazione (per elementi dinamici):

```
$(document).on('click', '.dynamic-btn', function() {  
  // gestore per elementi creati dopo il binding  
});
```

Rimozione handler:

```
$('#btn').off('click');
```

5. Animazioni ed effetti

Effetti built-in:

```
$('#box').hide();  
$('#box').show();  
$('#box').fadeIn(300);  
$('#box').slideToggle();
```

Animazioni personalizzate:

```
$('#box').animate({ left: '100px', opacity: 0.5 }, 400);
```

Usa CSS transitions per performance migliori su animazioni complesse.

6. AJAX

Con jQuery AJAX:

```
$.ajax({  
  url: '/api/items',  
  method: 'GET',  
  dataType: 'json'  
}).done(data => console.log(data))  
  .fail(err => console.error(err));
```

Shortcut:

```
$.getJSON('/api/items').then(data => { /* ... */ });  
$.post('/login', { user, pass }).done(res => { /* ... */ });
```

Impostare header globali / CSRF:

```
$.ajaxSetup({  
  headers: { 'X-CSRF-Token': token }  
});
```

Promesse e async/await: - jQuery Deferred non è una vera Promise standard, ma puoi usare `.then()`; per integrazione moderna preferisci `fetch/axios` o converti deferred con `Promise.resolve()`.

7. Plugin e estensioni

Usa plugin per componenti (slider, datepicker, etc.) o scrivine di tuoi: Esempio semplice plugin:

```
(function($) {  
  $.fn.highlight = function(color = 'yellow') {  
    return this.css('background-color', color);  
  };  
})(jQuery);  
  
// uso  
$('.item').highlight();
```

Ricorda di mantenere i plugin isolati e compatibili con moduli se usi bundler.

8. NoConflict

Se usi altre librerie che usano \$, evita conflitti:

```
const jq = jQuery.noConflict();  
// poi usa jq(...) al posto di $(...)
```

9. Performance e best practice

- Minimizza le selezioni ripetute: salva in variabili jQuery (\$el = \$('#id')).
 - Usa delegation per elementi dinamici.
 - Evita manipolazioni DOM multiple: costruisci un fragment e inseriscilo una volta.
 - Semplifica selettori: selettori complessi sono più lenti.
 - Valuta di usare Vanilla JS (querySelector, fetch) per operazioni semplici nelle app moderne.
-

10. Testing

- Per unit test: usa Jest + jsdom o Karma + Jasmine (legacy).
 - Per test end-to-end: Cypress o Playwright.
 - Isola comportamenti jQuery in funzioni puramente JS per testabilità.
-

11. Migrazione da jQuery a Vanilla / Framework moderno

- Valuta sostituire funzionalità con API native (fetch, classList, addEventListener).
 - Migra gradualmente con wrapper e feature flags.
 - Per grandi codebase, crea un piano di migration per rimuovere dipendenze da plugin jQuery.
-

12. Integrazione in progetti moderni (Webpack / Vite)

- Installa via npm e importalo:

```
import $ from 'jquery';  
window.jQuery = $; // se plugin legacy lo richiedono
```

- Usa alias se necessario e configura externals per CDN in produzione.
-

13. Sicurezza

- Non inserire direttamente HTML non sanitizzato tramite .html().
 - Sanifica input lato server e client.
 - Proteggi request AJAX con CSRF token.
-

14. Risorse utili

- jQuery docs: <https://api.jquery.com>
 - Migration guide: <https://jquery.com/upgrade-guide/>
 - Plugins popolari: Slick, Owl Carousel, jQuery UI (se serve UI legacy)
-

Esempio minimo (script)

```
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script>
$(function() {
  $('#load').on('click', function() {
    $.getJSON('/api/items', function(data) {
      $('#list').empty();
      data.forEach(i => $('#list').append($('- ').text(i.name)));
    });
  });
});
</script>

```

Vuoi che aggiunga anche esempi di plugin specifici o una sezione su integrazione con TypeScript? Dimmi e la estendo.