



**I.I.S. Marconi Dalmine**

Dipartimento di Informatica

# Esercitazione 07

## ChartJs

# Introduzione

**Chart.js** è una libreria JavaScript open source che permette di creare grafici animati, reattivi e facilmente personalizzabili. Con pochi comandi possiamo rappresentare dati sotto forma di:

- grafici a barre (`bar`);
- grafici a linee (`line`);
- torte o ciambelle (`pie`, `doughnut`);
- altri tipi come `radar`, `scatter`, `polarArea`.

È perfetta per introdurre la visualizzazione dei dati in JavaScript perché:

- è leggera (un solo file `.js`);
- è semplice da usare;
- funziona in qualsiasi browser moderno.

## Passaggio 1 — Importare la libreria

Il modo più rapido per usare Chart.js è includerlo tramite un **CDN**, cioè un link che carica la libreria da Internet. Inserisci questo tag nel tuo file HTML, *prima* del tuo script personale:

```
1 <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
```

Ecco la struttura minima di una pagina HTML pronta per Chart.js:

```
1 <!DOCTYPE html>
2 <html lang="it">
3 <head>
4   <meta charset="utf-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1">
6   <title>Primo grafico con Chart.js</title>
7 </head>
8 <body>
9   <canvas id="grafico"></canvas>
10
11  <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
12  <script>
13    // qui andrà il codice JavaScript
14  </script>
15 </body>
16 </html>
```

Il tag `<canvas>` è il contenitore grafico dove Chart.js disegnerà il grafico. L'attributo `id` serve per identificarlo nel codice.

## Passaggio 2 — Selezionare il canvas

Nel JavaScript, il primo passo è selezionare l'elemento `<canvas>`:

```
1 const ctx = document.getElementById('grafico');
```

La variabile `ctx` (abbreviazione di *context*) rappresenta il contesto grafico in cui il grafico verrà creato.

## Passaggio 3 — Definire i dati

Un grafico di Chart.js è sempre composto da:

- le **etichette** (`labels`) che identificano le categorie sull'asse X;
- uno o più **dataset** che contengono i valori numerici.

```
1 const data = {  
2   labels: ['Lun', 'Mar', 'Mer', 'Gio', 'Ven'],  
3   datasets: [{  
4     label: 'Visitatori',  
5     data: [12, 19, 3, 5, 2]  
6   }]  
7 };
```

Puoi aggiungere più oggetti dentro l'array `datasets` per confrontare serie di dati differenti.

## Passaggio 4 — Creare il grafico

Ora creiamo l'oggetto `Chart`, specificando il tipo di grafico, i dati e le opzioni:

```
1 new Chart(ctx, {  
2   type: 'bar', // bar, line, pie, doughnut, ecc.  
3   data: data,  
4   options: {}  
5 });
```

Il parametro `type` definisce la forma del grafico. L'oggetto `options` consente di modificare colori, titoli, assi e legenda.

## Passaggio 5 — Esempio completo

Salva il file come `chart_base.html` e aprilo nel browser:

```

1 <!DOCTYPE html>
2 <html lang="it">
3 <head>
4   <meta charset="utf-8" />
5   <meta name="viewport" content="width=device-width, initial-scale=1" />
6   <title>Chart.js base</title>
7   <style>
8     body { font-family: system-ui, sans-serif; margin: 2rem; max-width: 700px; }
9     canvas { width: 100%; max-width: 600px; }
10  </style>
11 </head>
12 <body>
13   <h1>Il mio primo grafico con Chart.js</h1>
14   <canvas id="grafico"></canvas>
15
16   <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
17   <script>
18     const ctx = document.getElementById('grafico');
19
20     const data = {
21       labels: ['Lun', 'Mar', 'Mer', 'Gio', 'Ven'],
22       datasets: [{
23         label: 'Visitatori',
24         data: [12, 19, 3, 5, 2]
25       }]
26     };
27
28     new Chart(ctx, {
29       type: 'bar',
30       data: data,
31       options: {
32         plugins: {
33           title: { display: true, text: 'Visitatori settimanali' },
34           legend: { position: 'bottom' }
35         },
36         scales: {
37           y: { beginAtZero: true }
38         }
39       }
40     });
41   </script>
42 </body>
43 </html>

```

Aprendo la pagina, vedrai un grafico a barre animato con cinque colonne.

## Documentazione ufficiale

Per approfondire tutte le funzionalità di Chart.js consulta la documentazione ufficiale:

<https://www.chartjs.org/docs/latest/>

Lì troverai esempi interattivi, guide dettagliate e le specifiche per ogni tipo di grafico.

# Tipi, colori e opzioni essenziali

In questa parte vediamo come:

- cambiare **tipo** di grafico (bar, line, doughnut, pie, ...);
- impostare **colori e stili** dei dataset;
- utilizzare le **opzioni** più comuni (titolo, legenda, assi).

## Cambiare il tipo di grafico

Il tipo si cambia modificando la stringa `type` in fase di creazione:

```
1 new Chart(ctx, {
2   type: 'line',    // 'bar', 'line', 'doughnut', 'pie', 'radar', 'polarArea', ...
3   data,
4   options: {}
5 });
```

## Colori e stile dei dataset

Puoi controllare colori e stile direttamente nel dataset:

```
1 const data = {
2   labels: ['A', 'B', 'C', 'D'],
3   datasets: [{
4     label: 'Esempio',
5     data: [5, 7, 3, 9],
6     backgroundColor: ['#A78BFA', '#818CF8', '#6366F1', '#4F46E5'], // riempimento barre
7     // ↪ o area
8     borderColor: '#4F46E5',    // bordo (linee/barre)
9     borderWidth: 1,
10    tension: 0.3,               // solo line: curvatura
11    pointRadius: 3              // solo line/scatter: dimensione punti
12  }]
13 };
```

## Opzioni comuni: titolo, legenda, assi

```
1  const options = {
2    responsive: true,
3    plugins: {
4      title: { display: true, text: 'Titolo del grafico' },
5      legend: { position: 'bottom' }
6    },
7    scales: {
8      y: {
9        beginAtZero: true,
10       ticks: { callback: v => v + ' unità' }
11     },
12     x: {
13       title: { display: true, text: 'Categoria' }
14     }
15   }
16 };
17 new Chart(ctx, { type: 'bar', data, options });
```

### Suggerimenti rapidi:

- `maintainAspectRatio:false` + contenitore con altezza fissa  $\Rightarrow$  grafici davvero responsive.
- Ogni modifica a `chart.data` richiede `chart.update()` per essere visualizzata.
- Per ricreare un grafico nello stesso canvas: `chart.destroy()` prima di istanziarne uno nuovo.

# Esempi da provare

Gli esempi seguenti sono pagine HTML complete. Copiali in file separati (ad es. 01\_barre\_base.html, 02\_linee\_scale.html, ...) e aprili nel browser.

## 01) Barre base — primo grafico in 60 secondi

```
1 <!DOCTYPE html>
2 <html lang="it">
3 <head>
4   <meta charset="utf-8" />
5   <meta name="viewport" content="width=device-width, initial-scale=1" />
6   <title>Chart.js - Barre base</title>
7   <style>body{font-family:system-ui;margin:2rem;max-width:900px}</style>
8 </head>
9 <body>
10  <h1>Barre (base)</h1>
11  <canvas id="chart"></canvas>
12
13  <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
14  <script>
15    const ctx = document.getElementById('chart');
16    const data = {
17      labels: ['Lun', 'Mar', 'Mer', 'Gio', 'Ven'],
18      datasets: [{ label: 'Visitatori', data: [12, 19, 3, 5, 2] }]
19    };
20    new Chart(ctx, {
21      type: 'bar',
22      data,
23      options: {
24        responsive: true,
25        plugins: { legend: { position: 'top' }, title: { display: true, text:
26          ↪ 'Settimana' } }
27      }
28    });
29  </script>
30 </body>
31 </html>
```

## 02) Linee + scale e titolo assi

```
1 <!DOCTYPE html>
2 <html lang="it">
3 <head><meta charset="utf-8"><meta name="viewport"
  ↳ content="width=device-width, initial-scale=1">
4 <title>Chart.js - Linee</title>
5 <style>body{font-family:system-ui;margin:2rem;max-width:900px}</style></head>
6 <body>
7   <h1>Linee con scale e titoli assi</h1>
8   <canvas id="chart"></canvas>
9   <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
10  <script>
11    const ctx = document.getElementById('chart');
12    const data = {
13      labels: ['Gen', 'Feb', 'Mar', 'Apr', 'Mag', 'Giu', 'Lug'],
14      datasets: [{
15        label: 'Temperatura media',
16        data: [6, 7, 12, 16, 20, 24, 27],
17        tension: 0.3,
18        pointRadius: 3
19      }]
20    };
21    new Chart(ctx, {
22      type: 'line',
23      data,
24      options: {
25        responsive: true,
26        scales: {
27          y: { beginAtZero: false, ticks: { callback: v => v + '°C' }, title: {
28            ↳ display: true, text: '°C' } },
29          x: { title: { display: true, text: 'Mese' } }
30        }
31      });
32    </script>
33  </body>
34 </html>
```



# Esercizi guidati

Gli esercizi seguenti sono **guidati**. Lavora in coppia e parti dai due esempi tenuti sopra.

## Esercizio 1 — Linea con dati reali

**Obiettivo:** creare un grafico a linee che mostri l'andamento di una variabile nel tempo (es. temperatura media settimanale).

**Consegne**

- Duplica l'esempio 02) *Linee* in un nuovo file `ex1_linea.html`.
- Sostituisci le etichette con i giorni della settimana e inserisci 7 valori numerici.
- Attiva il titolo dell'asse Y (`scales.y.title.display=true`) e scegli un titolo significativo.
- Aggiungi una seconda serie opzionale (es. minima vs. massima) e verifica la legenda.

**Checkpoint:** nel tooltip compaiono i valori corretti; la linea è liscia (`tension` tra 0.2 e 0.4).

## Esercizio 2 — Barre comparativo

**Obiettivo:** confrontare due categorie su 6–8 elementi (es. vendite 2024 vs 2025).

**Consegne**

- Parti dall'esempio 01) *Barre* e crea `ex2_barre_comparative.html`.
- Aggiungi un secondo dataset con label diversa.
- Imposta `beginAtZero=true` su Y e un titolo per l'asse X.
- Personalizza lo spessore delle barre (`barThickness` o `categoryPercentage`).

**Checkpoint:** le due serie sono distinguibili e la legenda non copre il grafico.

## Esercizio 3 — Torta/Ciambella con percentuali

**Obiettivo:** visualizzare la ripartizione percentuale di 4–6 voci (es. quote di mercato).

**Consegne**

- Crea `ex3_torta.html` e cambia `type` in `'doughnut'` o `'pie'`.
- Inserisci etichette e valori; abilita la legenda in alto (`plugins.legend.position='top'`).
- Mostra la percentuale nel tooltip (usa `tooltip.callbacks.label` con `total` e `value`).

**Checkpoint:** la somma delle percentuali è 100% (arrotondata).

## Esercizio 4 — Radar o Scatter con opzioni

**Obiettivo:** sperimentare un tipo meno comune e curare le opzioni.

**Consegne**

- Scegli `'radar'` (competenze) oppure `'scatter'` (coppie x,y).
- Aggiungi titolo del grafico (`plugins.title.display=true`).
- Per `scatter`: configura i limiti degli assi (`min/max`). Per `radar`: regola `suggestedMin/suggestedMax`.

- Rendi il grafico responsive dentro un contenitore al 60% della larghezza.

**Checkpoint:** nessuna sovrapposizione di etichette; il grafico resta leggibile ridimensionando la finestra.

## Esercizi

### Linea: serie doppia e media mobile

**Compito:** dati i valori mensili di due prodotti, crea un grafico a linee con entrambe le serie e aggiungi una terza serie calcolata: la media mobile a 3 mesi del prodotto A.

**Controllo:** titoli assi presenti; legenda chiara; tooltip leggibile; media corretta.

### Barre: classifica con ordinamento

**Compito:** mostra una classifica di 8 categorie con barre orizzontali; implementa un pulsante che ordina i dati crescente/decrescente (puoi usare `chart.data.datasets[0].data.sort()` e `chart.update()`).

**Controllo:** pulsante funzionante; etichette non troncate; asse X da zero.

### Torta/Ciambella: confronto % due anni

**Compito:** realizza due grafici `doughnut` affiancati (2024 vs 2025) con le stesse categorie. Inserisci al centro di ciascuno la somma totale (puoi usare un elemento HTML sovrapposto).

**Controllo:** percentuali coerenti; legende chiare; layout responsive (due colonne sopra 800px, una colonna sotto).

# Appendice — Mini Cheatsheet

## API veloci

```
1 // creare
2 const chart = new Chart(canvasEl, { type: 'bar', data, options });
3 // aggiornare
4 chart.data.labels.push('Nuova');
5 chart.data.datasets[0].data.push(42);
6 chart.update();
7 // rimuovere
8 chart.data.labels.pop();
9 chart.data.datasets[0].data.pop();
10 chart.update();
11 // distruggere (se serve ricreare nello stesso canvas)
12 chart.destroy();
```

## Problemi comuni

- **Grafico non appare:** verifica che il `<canvas>` esista e che lo script di Chart.js sia caricato.
- **Dati non cambiano:** hai chiamato `chart.update()` dopo le modifiche?
- **Errori di registrazione:** usando il CDN <https://cdn.jsdelivr.net/npm/chart.js> ottieni il build UMD con tutto già registrato.