

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Informatica Umanistica (1) (6 CFU)
A.A. 2023/2024

Sviluppo web (4)

HTML

Martina Dello Buono

Dottoranda in Digital Humanities

Dipartimento di Filologia Classica e Italianistica

Alma Mater Studiorum - Università di Bologna

<https://www.unibo.it/sitoweb/martina.dellobuono2/>

martina.dellobuono2@unibo.it

Orario delle lezioni → 5 lezioni

1. 4 dicembre → 13:00 – 15:00 → HTML.
2. 5 dicembre → 17:00 – 19:00 → HTML.
3. 7 dicembre → 15:00 – 17:00 → CSS.
4. 11 dicembre → 13:00 – 15:00 → Bootstrap Grid System.
5. 12 dicembre → 17:00 – 19:00 → Bootstrap components.

Argomenti affrontati:

HTML + CSS + Bootstrap

Materiale didattico:

<https://github.com/IU-lab-2023-2024>

La frequenza non è obbligatoria, ma fortemente consigliata.

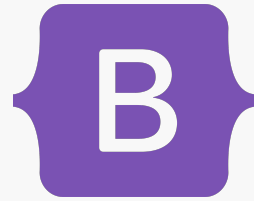




1. Getting started

Introduzione a Bootstrap

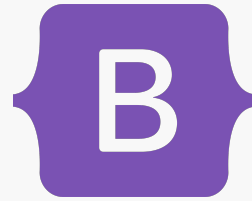
<https://getbootstrap.com/docs/5.0/getting-started/introduction/>



Che cos'è Bootstrap?

«A framework for building
responsive [...] sites»

<https://getbootstrap.com/docs/4.6/getting-started/introduction/>



1. Framework

«un'architettura logica di supporto [...] sulla quale un software può essere progettato e realizzato [...]. Alla base [...] c'è sempre una serie di librerie di codice [...]. Lo scopo [...] è infatti quello di risparmiare [...] la riscrittura di codice già scritto [...] per compiti simili»

<https://it.wikipedia.org/wiki/Framework>

2. Responsive



Monitor



Laptop



Tablet



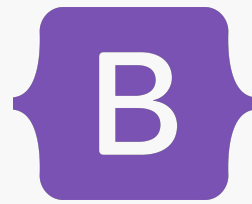
Smartphone

Gli utenti utilizzano schermi di diverse dimensioni per navigare nel web.

Le dimensioni della finestra del browser («**viewport**») e la quantità di informazioni visualizzate si adattano alle dimensioni dello schermo del device utilizzato per la visualizzazione di un sito.

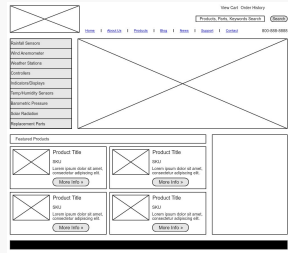
In fase di progettazione e sviluppo è necessario considerare un ventaglio di schermi di diverse dimensioni per rendere il sito «**responsive**» = in grado di cambiare dinamicamente il proprio aspetto a seconda delle dimensioni e dell'orientamento del dispositivo utilizzato per visualizzarlo.

<https://www.nngroup.com/articles/responsive-web-design-definition/>



A cosa serve dunque Bootstrap?

Riutilizzare codice già scritto per realizzare e gestire secondo i principi del responsive web design:



Esempio di layout

1. Il **layout** di un sito

2. Le **componenti di interfaccia** (es: bottoni, form, navbar, ecc.)

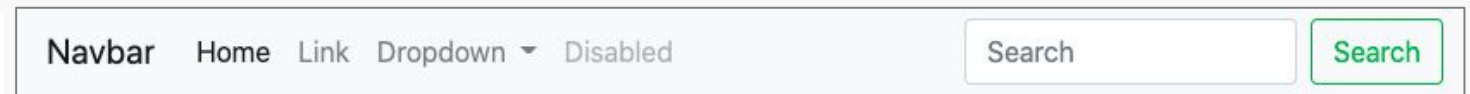
Email address

We'll never share your email with anyone else.

Password

☐ Check me out

Esempio di form



Esempio di navbar



Esempio di bottoni

```
bootstrap/
├── css/
│   ├── bootstrap-grid.css
│   ├── bootstrap-grid.css.map
│   ├── bootstrap-grid.min.css
│   ├── bootstrap-grid.min.css.map
│   ├── bootstrap-grid.rtl.css
│   ├── bootstrap-grid.rtl.css.map
│   ├── bootstrap-grid.rtl.min.css
│   ├── bootstrap-grid.rtl.min.css.map
│   ├── bootstrap-reboot.css
│   ├── bootstrap-reboot.css.map
│   ├── bootstrap-reboot.min.css
│   ├── bootstrap-reboot.min.css.map
│   ├── bootstrap-reboot.rtl.css
│   ├── bootstrap-reboot.rtl.css.map
│   ├── bootstrap-reboot.rtl.min.css
│   ├── bootstrap-reboot.rtl.min.css.map
│   ├── bootstrap-utilities.css
│   ├── bootstrap-utilities.css.map
│   ├── bootstrap-utilities.min.css
│   ├── bootstrap-utilities.min.css.map
│   ├── bootstrap-utilities.rtl.css
│   ├── bootstrap-utilities.rtl.css.map
│   ├── bootstrap-utilities.rtl.min.css
│   ├── bootstrap-utilities.rtl.min.css.map
│   ├── bootstrap.css
│   ├── bootstrap.css.map
│   ├── bootstrap.min.css
│   ├── bootstrap.min.css.map
│   ├── bootstrap.rtl.css
│   ├── bootstrap.rtl.css.map
│   ├── bootstrap.rtl.min.css
│   └── bootstrap.rtl.min.css.map
└── js/
    ├── bootstrap.bundle.js
    ├── bootstrap.bundle.js.map
    ├── bootstrap.bundle.min.js
    ├── bootstrap.bundle.min.js.map
    ├── bootstrap.esm.js
    ├── bootstrap.esm.js.map
    ├── bootstrap.esm.min.js
    ├── bootstrap.esm.min.js.map
    ├── bootstrap.js
    ├── bootstrap.js.map
    ├── bootstrap.min.js
    └── bootstrap.min.js.map
```

A cosa serve dunque Bootstrap?

CSS

Fogli di stile per gestire graficamente **layout** e **componenti di interfaccia** responsive.

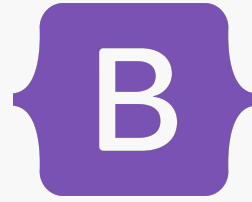
HTML

+

Per realizzare **layout** e **componenti di interfaccia** di un sito.

JavaScript (JS)

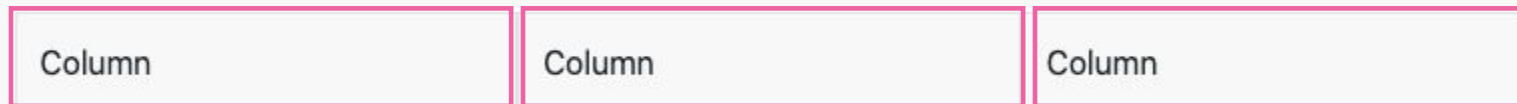
Funzioni per gestire la **dinamicità** delle componenti di interfaccia.



Un esempio pratico Layout

Task: Devo organizzare i contenuti della mia pagina web su 3 colonne. Come fare?

1. Potrei scrivere **partendo da 0** istruzioni HTML per costruire la struttura della pagina e inserire contenuti al suo interno, e istruzioni CSS per realizzare le 3 colonne e per renderle responsive/modificare il loro stile a seconda dei vari dispositivi.
2. **Utilizzare codice Bootstrap** che organizza automaticamente i miei contenuti su 3 colonne e regola il loro comportamento a seconda del dispositivo utilizzato per la visualizzazione.



Devo organizzare i contenuti della mia pagina web su 3 colonne. Come fare?

The screenshot shows the Bootstrap 5.0 documentation page for the Grid system. The page has a purple header with navigation links: Home, Docs, Examples, Icons, Themes, Blog, and a Download button. Below the header is a search bar and a dropdown menu for Bootstrap v5.0. The main content area is titled 'Grid system' and includes a 'View on GitHub' button. The text describes the grid system as a powerful mobile-first flexbox grid with a twelve column system, six default responsive tiers, Sass variables, and mixins. An 'Example' section explains that the grid system uses containers, rows, and columns to layout and align content. A code snippet shows the basic HTML structure for a three-column grid. A 'Copy' button is next to the code. On the right side, there is a 'On this page' section with a list of links to various topics related to the grid system.

Home Docs Examples Icons Themes Blog

Search docs... Ctrl + /

Bootstrap v5.0

Getting started

Customize

Layout

Breakpoints

Containers

Grid

Columns

Gutters

Utilities

Z-index

Content

Forms

Components

Helpers

Utilities

Extend

About

Migration

Grid system

View on GitHub

Use our powerful mobile-first flexbox grid to build layouts of all shapes and sizes thanks to a twelve column system, six default responsive tiers, Sass variables and mixins, and dozens of predefined classes.

Example

Bootstrap's grid system uses a series of containers, rows, and columns to layout and align content. It's built with [flexbox](#) and is fully responsive. Below is an example and an in-depth explanation for how the grid system comes together.

New to or unfamiliar with flexbox? [Read this CSS Tricks flexbox guide](#) for background, terminology, guidelines, and code snippets.

Column	Column	Column
--------	--------	--------

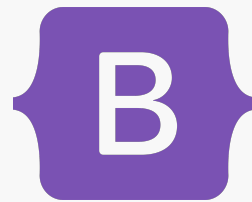
```
<div class="container">
  <div class="row">
    <div class="col">
      Column
```

Copy

On this page

- Example
- How it works
- Grid options
- Auto-layout columns
 - Equal-width
 - Setting one column width
 - Variable width content
- Responsive classes
 - All breakpoints
 - Stacked to horizontal
 - Mix and match
 - Row columns
- Nesting
- Sass
 - Variables
 - Mixins
 - Example usage
- Customizing the grid
 - Columns and gutters
 - Grid tiers

<https://getbootstrap.com/docs/5.0/layout/grid/>



Un esempio pratico Componenti di interfaccia

Task: Devo inserire nella mia pagina web un bottone cliccando sul quale si apre un menu «a tendina» (*dropdown menu*) contenenti le voci «x», «y» e «z». Come fare?

1. Potrei scrivere **partendo da 0** istruzioni HTML per realizzare la struttura e inserire i contenuti, istruzioni CSS per assegnare alle componenti uno stile e per renderle responsive/modificare il loro stile a seconda dei vari dispositivi, e una funzione JS per gestire la funzionalità di apertura e chiusura del menu «a tendina».
2. **Utilizzare codice Bootstrap** che costruisce e controlla struttura (HTML), stile (CSS) e dinamicità (JS) di tali componenti.

Devo inserire nella mia pagina web un bottone cliccando sul quale si apre un menu «a tendina» contenenti le voci «x», «y» e «z». Come fare?

The screenshot shows the Bootstrap 5.0 documentation page for the 'Dropdowns' component. The page has a purple header with navigation links: Home, Docs, Examples, Icons, Themes, and Blog. A search bar is located below the header. The main content area is titled 'Dropdowns' and includes a 'View on GitHub' link. The text describes how to use the Bootstrap dropdown plugin, mentioning the 'Popper' library and the 'bootstrap.bundle.min.js' file. A sidebar on the left lists various components, with 'Dropdowns' highlighted under the 'Components' section. A sidebar on the right lists sub-topics for the dropdowns, such as 'Overview', 'Accessibility', 'Examples', 'Sizing', 'Dark dropdowns', 'Directions', 'Menu items', 'Menu alignment', 'Menu content', 'Dropdown options', and 'Sass'.

Home Docs Examples Icons Themes Blog

Search docs... Ctrl + /

Bootstrap v5.0

View on GitHub

Dropdowns

Toggle contextual overlays for displaying lists of links and more with the Bootstrap dropdown plugin.

Overview

Dropdowns are toggleable, contextual overlays for displaying lists of links and more. They're made interactive with the included Bootstrap dropdown JavaScript plugin. They're toggled by clicking, not by hovering; this is [an intentional design decision](#).

Dropdowns are built on a third party library, [Popper](#), which provides dynamic positioning and viewport detection. Be sure to include [popper.min.js](#) before Bootstrap's JavaScript or use [bootstrap.bundle.min.js](#) / [bootstrap.bundle.js](#) which contains Popper. Popper isn't used to position dropdowns in navbars though as dynamic positioning isn't required.

Accessibility

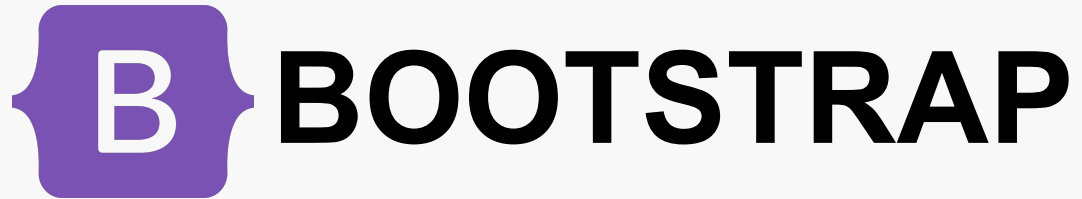
The [WAI-ARIA](#) standard defines an actual [role="menu" widget](#), but this is specific to application-like menus which trigger actions or functions. ARIA menus can only contain menu items, checkbox menu items, radio button menu items, radio button groups, and sub-menus.

Bootstrap's dropdowns, on the other hand, are designed to be generic and applicable to a variety of situations and markup structures. For instance, it is possible to create dropdowns that contain additional inputs and form controls, such as search fields or login forms. For this reason, Bootstrap does not expect (nor automatically add) any of the [role](#) and [aria-](#) attributes required for true ARIA menus. Authors will have to include these

On this page

- Overview
- Accessibility
- Examples
 - Single button
 - Split button
- Sizing
- Dark dropdowns
- Directions
 - Dropup
 - Dropright
 - Dropleft
- Menu items
 - Active
 - Disabled
- Menu alignment
 - Responsive alignment
 - Alignment options
- Menu content
 - Headers
 - Dividers
 - Text
 - Forms
- Dropdown options
 - Auto close behavior
- Sass

<https://getbootstrap.com/docs/5.0/components/dropdowns/>



2. Setup

Download, CDN

<https://getbootstrap.com/docs/5.0/getting-started/introduction/>

<https://getbootstrap.com/docs/5.0/getting-started/download/>

Esistono più versioni di Bootstrap. Verrà analizzata la v5.0.2.



Prima di poter utilizzare gli strumenti messi a disposizione, è necessario **integrare Bootstrap** nel proprio progetto.

Non è sufficiente riutilizzare il codice HTML presente sul sito di Bootstrap perché tutte le componenti funzionino a dovere (JS) e appaiano correttamente dal punto di vista stilistico (CSS).

È fondamentale creare innanzitutto **un collegamento fra il proprio progetto e i file contenenti le istruzioni CSS/JS fornite da Bootstrap.**

Setup Due metodi

1. CDN



2. Download



CDN

Setup

Metodo 1: usare una CDN



È possibile includere nei propri file HTML i **link ai file CSS e JS Bootstrap**, seguendo il procedimento riportato di seguito:

Tutorial

Pro: il processo di setup è molto rapido.

Contro: la performance del sito peggiora in termini di velocità. Le istruzioni CSS/JS Bootstrap prima di essere lette e via via applicate devono essere reperite tramite connessione ad altro server, diverso da quello che ospita il nostro sito!

Setup

Metodo 2: scaricare Bootstrap

Download



Il nostro progetto

File che abbiamo in locale (sulla nostra macchina/computer)



Bootstrap

File CSS e JS che contengono istruzioni per impostare un certo stile e funzionalità

<https://getbootstrap.com/docs/5.0/getting-started/download/>

È possibile scaricare il pacchetto di file CSS e JS Bootstrap da includere nella cartella del proprio progetto. Una volta incluso il pacchetto, è necessario eseguire gli step esemplificati di seguito:

Tutorial



Pro: la performance del sito migliora in termini di velocità perché le istruzioni CSS/JS non devono essere reperite su altri server.

Setup

Approccio 2: scaricare Bootstrap

Download



Il nostro progetto

File che abbiamo in locale (sulla nostra macchina/computer)



Bootstrap

File CSS e JS che contengono istruzioni per impostare un certo stile e funzionalità

<https://getbootstrap.com/docs/5.0/getting-started/download/>

Contro:

1. Il processo di setup e il tempo di apprendimento degli step necessari a completarlo sono più lunghi.
2. Non tutti i file scaricati e presenti nel pacchetto Bootstrap sono effettivamente necessari. Vi consiglio di mantenere soltanto `bootstrap.min.css` e `bootstrap.bundle.min.js`.



3. Layout

Container, Grid system

<https://getbootstrap.com/docs/5.0/layout/>

Layout



Monitor



Laptop



Tablet



Smartphone

In fase di progettazione e sviluppo è necessario considerare un ventaglio di schermi di diverse dimensioni per rendere il sito **«responsive»** = in grado di cambiare dinamicamente il proprio aspetto a seconda delle dimensioni e dell'orientamento del dispositivo utilizzato per visualizzarlo.



Come gestisce Bootstrap il processo di design responsive?

1. Breakpoint

Per gestire in maniera più agevole il processo di design responsive è stato introdotto il concetto di «**breakpoint**». Bootstrap v.5.0.x ne fornisce una definizione:

I breakpoint sono larghezze personalizzabili che determinano il comportamento del layout responsive tra i dispositivi o le dimensioni della finestra [...].

Bootstrap definisce dunque delle soglie entro le quali classificare i dispositivi a seconda delle loro dimensioni.

1. Breakpoint

I dispositivi sono suddivisi in **6 classi** in base alla loro **width** (larghezza) → 6 breakpoint:

	Class infix	Breakpoint	Dimensioni	Esempio di tipo di dispositivo
X-Small	<i>None</i>		<576px	
Small	sm	576px	≥576px	Smartphone orientato verticalmente (<i>portrait phone</i>)
Medium	md	768px	≥768px	Smartphone orientato orizzontalmente (<i>landscape phone</i>)
Large	lg	992px	≥992px	Tablet
Extra large	xl	1200px	≥1200px	Desktop
Extra extra large	xxl	1400px	≥1400px	Large desktop

<https://getbootstrap.com/docs/5.0/layout/breakpoints/>

2. Media query

Le media query sono un meccanismo che si avvale dei breakpoint per permettere di adattare i fogli di stile in CSS alle diverse dimensioni dei viewport/device. Nello specifico, la **CSS at-rule** (dichiarazione CSS) **@media** permette di definire delle condizioni alle quali, se vere, applicare certe regole.

https://developer.mozilla.org/it/docs/Web/CSS/Media_Queries

Es: le dimensioni dei titoli `h1` sono state impostate a 4em. Nei dispositivi medium, i titoli `h1` devono essere ridotti a 2em. Come specificare questa regola in CSS?

```
h1 {font-size: 4em;}
```

Aggiungiamo una media query:

```
@media screen and (min-width: 768px) and (max-width: 991px) {  
    h1 {font-size: 2em;}  
}
```

2. Media query

Regola applicata di default

```
h1 {font-size: 2em;}
```

Media query

Bootstrap solitamente utilizza **min-width** come condizione. Può essere usata anche max-width oppure, come in questo caso, entrambe.

Se è vero che la larghezza dello schermo è minimo 576px e massimo 767px

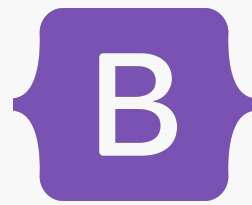
```
@media screen and (min-width: 768px) and (max-width: 991px) {
```

Allora gli h1 devono avere le dimensioni di 2rem

```
h1 {font-size: 2em;}
```

```
}
```

Sottinteso: se la larghezza, invece, è minore di 576px o maggiore di 767px, applica la regola di default.



Come gestisce e implementa il layout Bootstrap?

3. Container

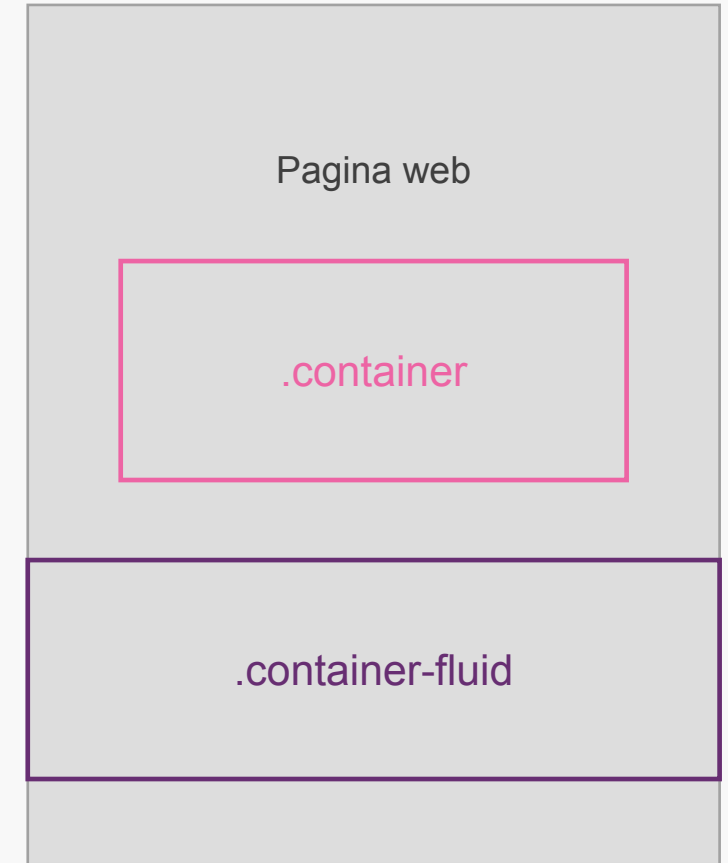
I container sono gli elementi fondativi del layout in Bootstrap. È necessario includerli nella struttura HTML della pagina se si intende utilizzare il «grid system» (sistema a griglia) messo a punto da Bootstrap per costruire un layout basato su una struttura a 12 colonne.

Essi sono dei contenitori in grado contenere qualsiasi elemento al loro interno: titoli, paragrafi, immagini, ecc. Essi possono essere innestati.

Esistono 3 tipi di container, ognuno dei quali è individuato da una classe da assegnare a un elemento HTML che funge da contenitore (**section**, **div**, ecc.):

1. `.container`
2. `.container-fluid`
3. `.container-{breakpoint}`

<https://getbootstrap.com/docs/5.0/layout/containers/>

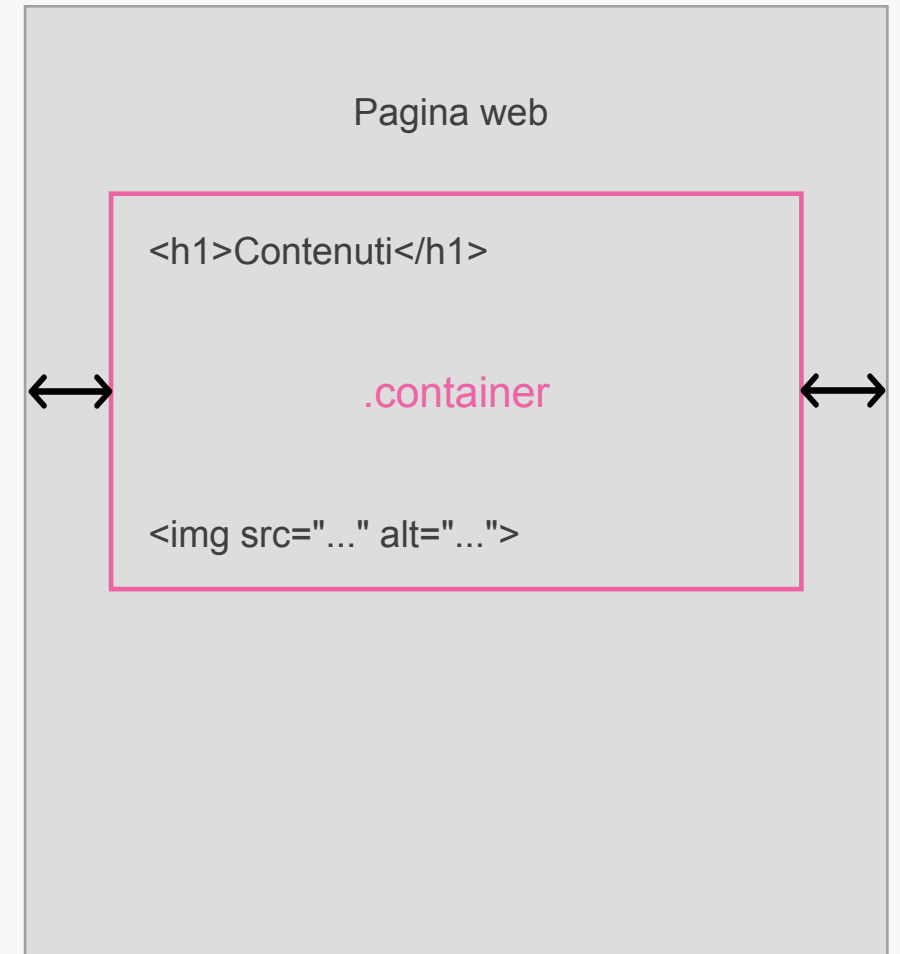


.container

```
<div class=".container">  
  <!-- Content here -->  
</div>
```

La classe **.container** assegnata a un elemento contenitore (nel caso riportato, **<div>**) imposta una larghezza massima (**max-width**) ad ogni breakpoint.

La caratteristica principale dei **.container** è la presenza di **netti margini laterali a destra e a sinistra** del contenitore in tutti i viewport/device a parte che negli extra small.

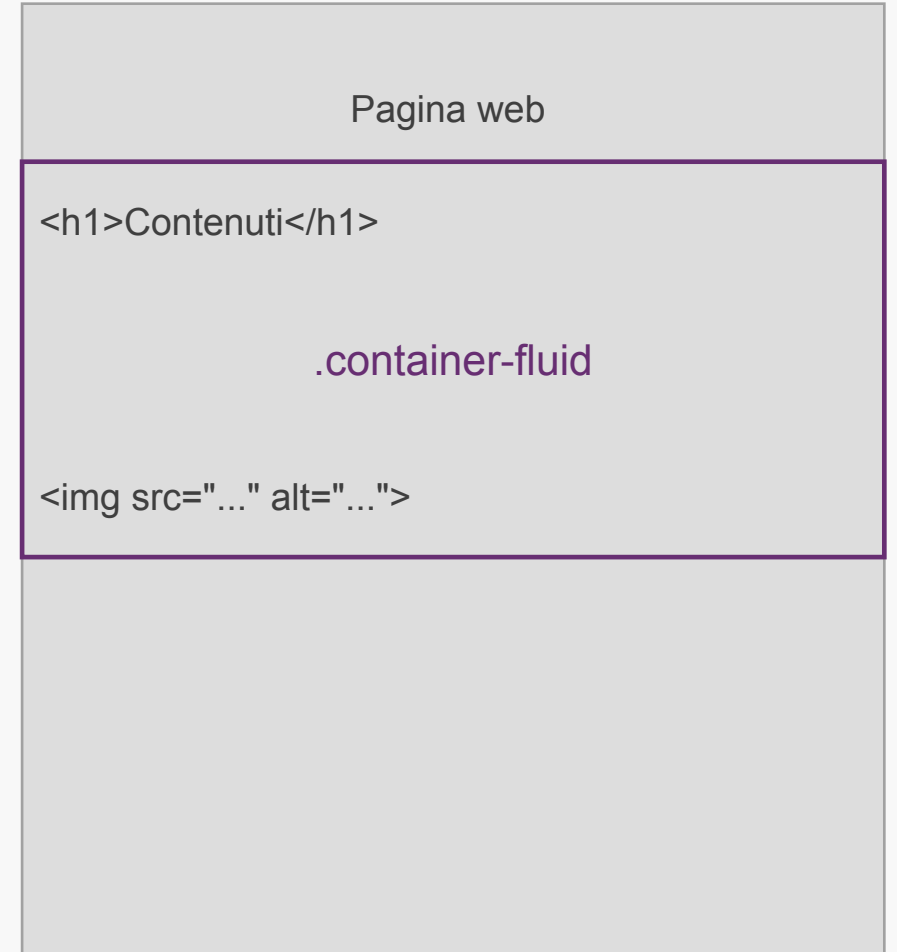


.container-fluid

```
<div class="container-fluid">  
  <!-- Content here -->  
</div>
```

La classe **.container-fluid** assegnata a un elemento contenitore (nel caso riportato, **<div>**) imposta una **larghezza fissa width: 100%** qualsiasi sia la dimensione del viewport/device.

Il container-fluid occupa in larghezza l'**intera larghezza della pagina**.



Come gestisce e implementa il layout Bootstrap?

.container e **.container-fluid**

Containers

Additional classes added in Bootstrap v4.4 allow containers that are 100% wide until a particular breakpoint.

.container

.container-sm

.container-md

.container-lg

.container-xl

.container-fluid

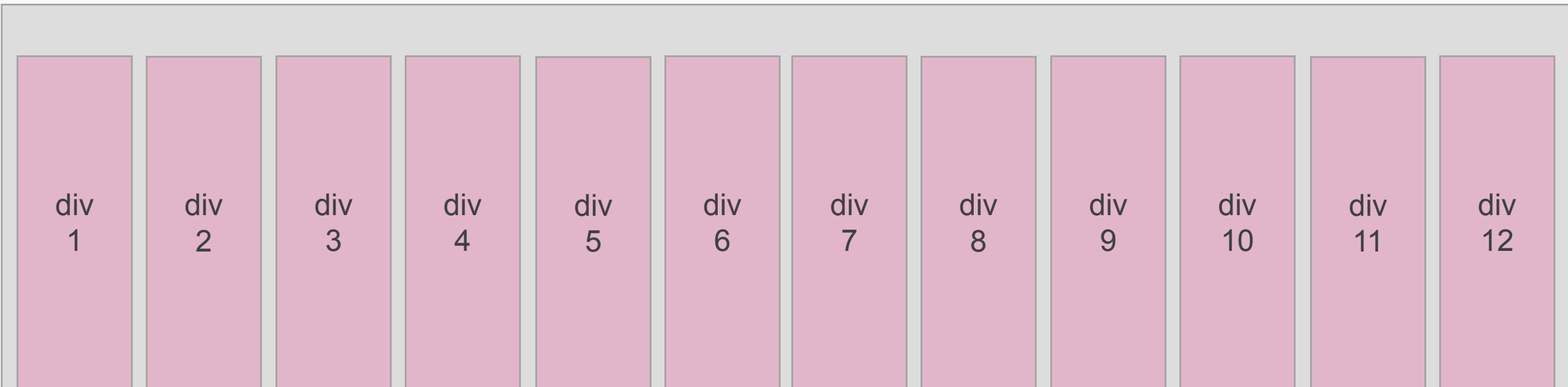
<https://getbootstrap.com/docs/4.6/examples/grid/#containers>



Come gestisce e implementa il layout Bootstrap?

4. Grid system - Colonne

Il «grid system» (sistema a griglia) Bootstrap è un sistema che permette di **strutturare layout responsive** su un **sistema a 12 colonne**. Ogni colonna viene identificata da un `<div>`.

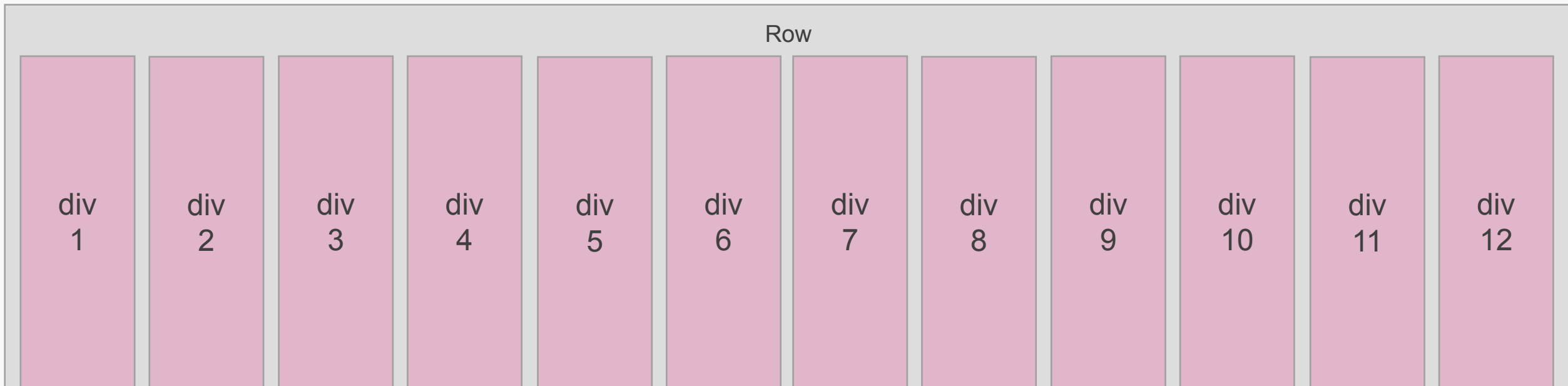


<https://getbootstrap.com/docs/5.0/layout/grid/>

Guida al CSS layout **flexbox** su cui si basa il grid system Bootstrap: <https://css-tricks.com/snippets/css/a-guide-to-flexbox/#flexbox-background>

4. Grid system - Row

Una somma massima di 12 colonne è racchiusa in una riga, detta *row*, corrispondente a `<div class="row">`.

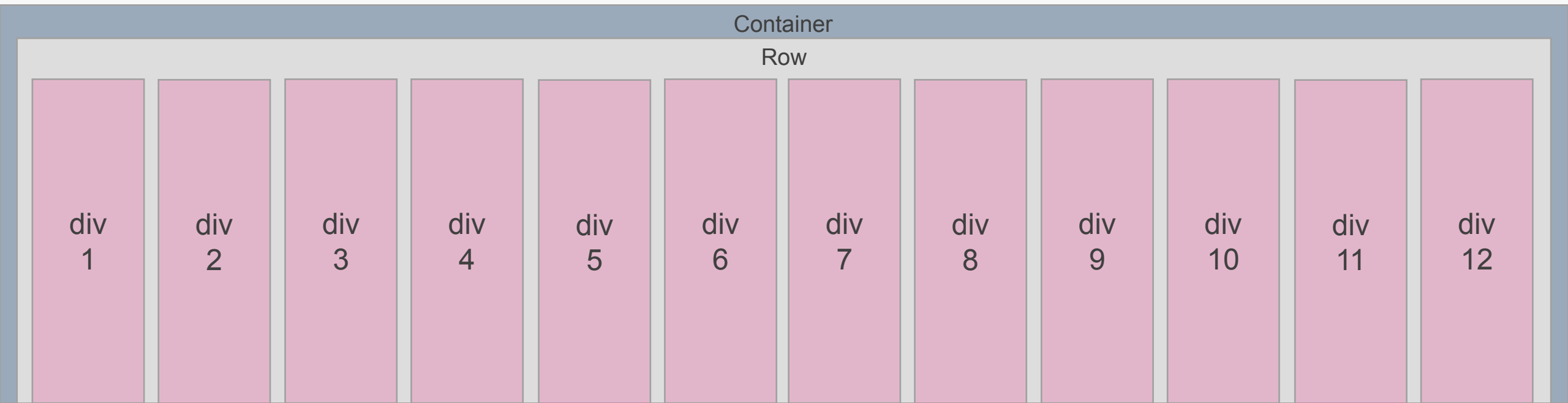


<https://getbootstrap.com/docs/5.0/layout/grid/>

Guida al CSS layout **flexbox** su cui si basa il grid system Bootstrap: <https://css-tricks.com/snippets/css/a-guide-to-flexbox/#flexbox-background>

4. Grid system - Container

Le *row* sono contenute da un container, corrispondente a `<div class="container">` oppure `<div class="container-fluid">`.



<https://getbootstrap.com/docs/5.0/layout/grid/>

Guida al CSS layout **flexbox** su cui si basa il grid system Bootstrap: <https://css-tricks.com/snippets/css/a-guide-to-flexbox/#flexbox-background>

Il Messaggiero

In Roma cent. 8

N. 1 — LUNEDÌ 16 DICEMBRE 1878

FUORI GI' INFERNO CANT. I

AI LETTORI.

Non occorre inventare programmi politici, o addirittura di ciò che ragionare: il lettore lo vedrà subito da sé. Ci basta indicare le strade sotterranee a quali esigenze e con che mezzi questo nuovo pestifero si propone di soddisfare.

Non a tutti è dato abbonarsi a più giornali; ed una sola « due, per quanto il formato sia ampio e la redazione diligente, sia potersi dar notizia di tutto, avendo ciascuno limitate fonti d'informazione. Anzi meno possibile, per i rapporti che hanno nelle parti politiche, riferire ciascuno le notizie e diverse opinioni sugli atti del governo ed i movimenti dei partiti.

Noi ci proponiamo di fare appieno ciò che gli altri per la loro natura non possono riferire, cioè, sopra ogni questione tutte le opinioni di ogni avvenimento tutte le notizie. A tale scopo riprodurremo tutti i giorni, integralmente o in parte secondo l'importanza e lo spazio, gli articoli, le corrispondenze, le informazioni di tutta la stampa italiana sopra le questioni e gli avvenimenti del giorno. Il nostro giornale li riferirà, compendiali, tutti sono più diretti, e così il Giornale dei giornali.

[illegible][illegible]

col

Tutti i giornali, più o meno basati sulle gli uomini
il cinque del giornale, hanno avuto ragione. Gli altri
Bale i qualcuno per l'errore del governo. Tra
sempre, se dovessero riprendere solo al suo solito
in questi ore di tempeste, a quella persona da consigli
il potere è l'organizzazione!

The masses of turning points, under various q regions: —

strano, di un galeotto storico come i suoi cari i greci — da Demostene, da Eschilo, da Socrate, da Solone. La religione non è la loro — né che ne ispiri mai né Plutarco, né Crisostomo, né Cicerone. I suoi cristiani, con il commercio con gli egiziani, siatando, non lo sopprimono ormai pure con l'abolizione della Morte! Questo non è tutto, il non è mai l'ha colpito al cuore, colpendo il galeotto Carlo, altri, che gli concedono, possono, rimanere, non, neanche. La lezione non è cambiata. Carlo era il protago della giustizia personale, dell'idea, della proba politica. Chi non, sempre, che lo ispiri non pure, questo protago, non ha mai, come a fare — e, nel frattempo, altri ne sostituiscono con definiti e voluti, spaziarli così non per i suoi sogni della giustizia collettiva, sottile, di terribile verità.

... il potere deve fare appello alla no-
...? ma che programma? Qualco-
... le persone cadute in questi or-
... le immedie le programmi di lavoro: i re-
... dal punto di vista: non sono migliori? non
... ma che religione della repubblica, come i Fran-
...? E le chiedo: quando — le ripete — repubbli-
... e, addosso da clericali. Noi della Democrazia Legione
... le vogliono, se vengono con le loro donne —
... i loro programmi sono al livello morale di Giu-
... e Lazzarini — che, come il governo di Bologna
... una volta. L'ultimo l'avevano il centro — le ripete
... di lui e che con le si si, sempre, con loro
... loro, dalla nostra stessa.

la casa di chi? con chi programma — la rubrica — il nuovo giornale — leggere e analizzare

...che tutti gli uomini sono liberi, per la natura
e la legge vengono.

Una cosa è di cui il nostro modo di vivere è
costante del tutto invariante.

Es, naturalmente alle altre promesse del nostro
no, alla realizzazione degli aumenti per cassa, —
senza che le conseguenti perdite in natura pesi
di fatto all'aumento di un dividendo che, per quanto
inferiore, da tutti per legge dovrebbe prevalere.

La vigilia del mio perfezionarsi, nel pieno della vita pregressa, si ha promissione sui relativi compiti da svolgere. I compagni vengono assegnati a vari uffici, come il controllo delle voci importanti nel probabile campo della popolazione, controllo il ministero della casa ecc. ecc. E' una scoperta anche quella per mettere ordine e senso alla confusione.

Ma l'organo del governo, che profile volute accanto alla sede al centro di una massiccia costruzione, si spande agli estremi abilitati all'occasione ogni cosa. « Se potremo, il piano di via Cavour e sui suoi confini, ancora si regge, resta, con una riga di acciottoli Cavour, sull'immagine, profano per la rappresentazione necessaria. — L'Italia, obliqua abituata a libertà per arrivare l'agente il simbolo della Cavour e la derivata della Cavour, potremo con noi, la politica del governo che per accendere al governo Cavour, sarà guidata dal piano con cui molti provinciali, con un'immagine di acciottoli, su alla compa degli acciottoli. »

Ed il paese, — quale è nel nostro Schizmatismo
l'un governo che può di tutto esserle a fianco, —
essendo in tale stato, ed avendo questa

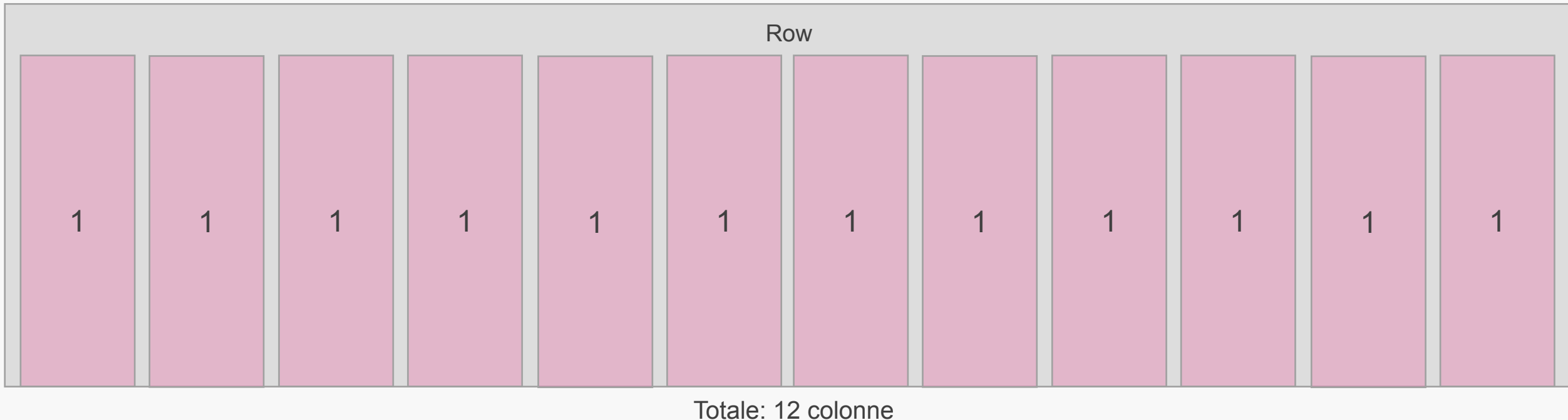
colonne

Esempio di griglia. Immagine da: https://it.wikipedia.org/wiki/File:Il_Messaggero_1878.jpg



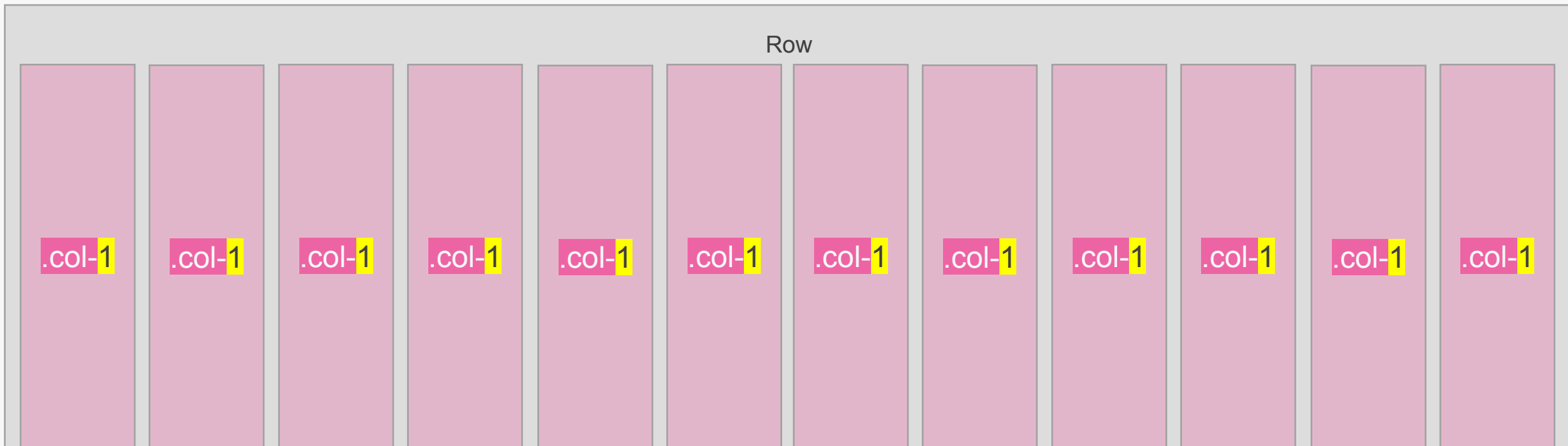
Grid system

Ciascuna **colonna** è **1/12 dell'intero** sistema, dunque ciascuna colonna ha **valore 1** perché **occupa 1 parte dell'intero**. Sommando i valori di tutte le colonne otteniamo come risultato **12**, ossia 12 parti, e cioè il **numero massimo di colonne in cui può essere divisa la pagina**.



Ogni colonna corrisponde a un `<div>` con classe `.col-*`

In Bootstrap una colonna è identificata sempre da un `<div>` con classe `.col-` a cui è necessario aggiungere un **valore numerico corrispondente al numero di parti dell'intero che la colonna deve occupare**. Di default l'intero sistema è costituito da 12 colonne=12 parti, ognuna delle quali vale 1=1 parte. Allora ogni colonna in un sistema da 12 colonne sarà identificata dalla classe `.col-` (che identifica sempre una colonna) e **1** (perché ogni colonna occupa 1 parte su 12). Il risultato è la classe `.col-1` da assegnare a ciascuna colonna.



Totale: 12 colonne in cui ogni colonna vale 1 parte dell'intero

Somma dei valori numerici assegnati alle classi

Sommando tutti i valori assegnati alle classi di ciascuna colonna otteniamo **12** e cioè il numero di parti di cui è costituito l'intero
= 12 parti = 12 colonne.

Qualsiasi sia il valore numerico assegnato alle classi delle colonne, dobbiamo ottenere sempre 12. Se il risultato è maggiore o minore, c'è un errore e il layout non risulterà strutturato correttamente! Le colonne in più andranno a capo!

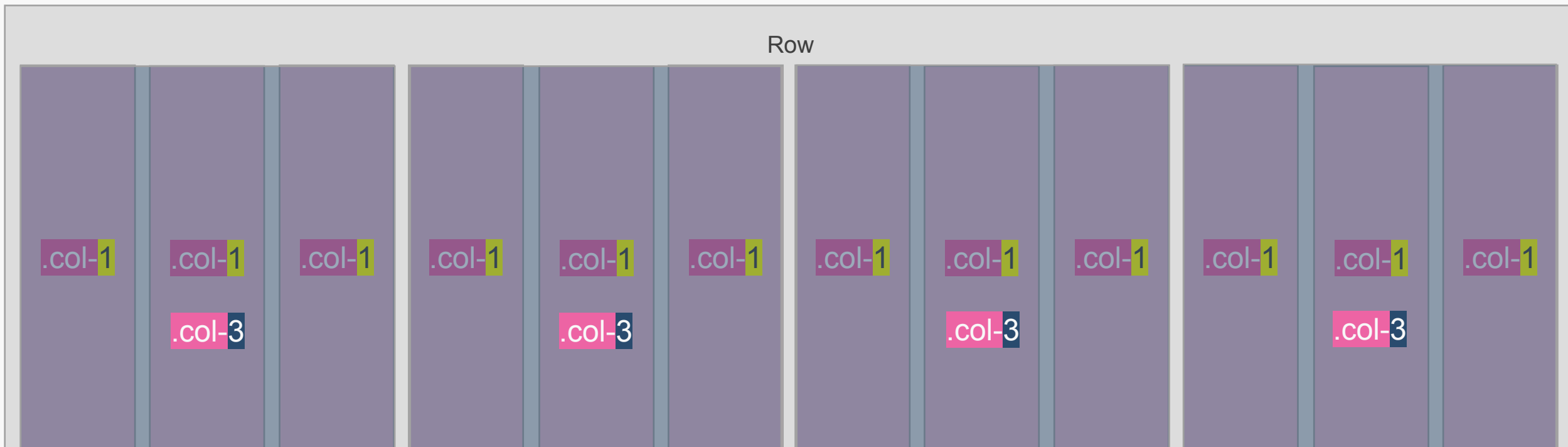
Row											
.col-1	.col-1	.col-1	.col-1	.col-1	.col-1	.col-1	.col-1	.col-1	.col-1	.col-1	.col-1

Totale: 12 colonne in cui ogni colonna vale 1 parte dell'intero

Colonne di dimensioni uguali che occupano più di 1 parte dell'intero

Non è detto che si abbia la necessità di costruire un layout disposto su 12 colonne e quindi di distribuire i propri contributi su 12 colonne. **È infatti possibile dividere lo spazio della pagina da un massimo di 12 colonne a un minimo di 1 colonna.**

Ragioniamo per esempi. Mettiamo che abbiamo un lungo testo da disporre su 4 colonne. Ciò significa che dobbiamo dividere lo spazio della pagina in 4 colonne = 4 parti. **Lo spazio in totale vale sempre 12** perché 12 sono le colonne in totale. Come dividerlo allora in 4 parti?



Totale: 12 colonne in cui ogni colonna vale 3 parti dell'intero

Colonne di dimensioni uguali che occupano più di 1 parte dell'intero

12 (spazio totale) : 4 (parti che voglio ottenere) = 3 (numero che devo associare alla classe .col-)

Divido lo spazio totale per il numero di parti che voglio ottenere. Il risultato è il numero che devo associare a `.col-`.

SPAZIO TOTALE : PARTI CHE VOGLIO OTTENERE = NUMERO DA ASSOCIARE A `.col-`



Qualsiasi sia il valore numerico assegnato alle classi delle colonne, sommando tutti i valori assegnati alle classi di ciascuna colonna dobbiamo ottenere sempre 12. Se il risultato è maggiore o minore, c'è un errore e il layout non risulterà strutturato correttamente!



Regola generale

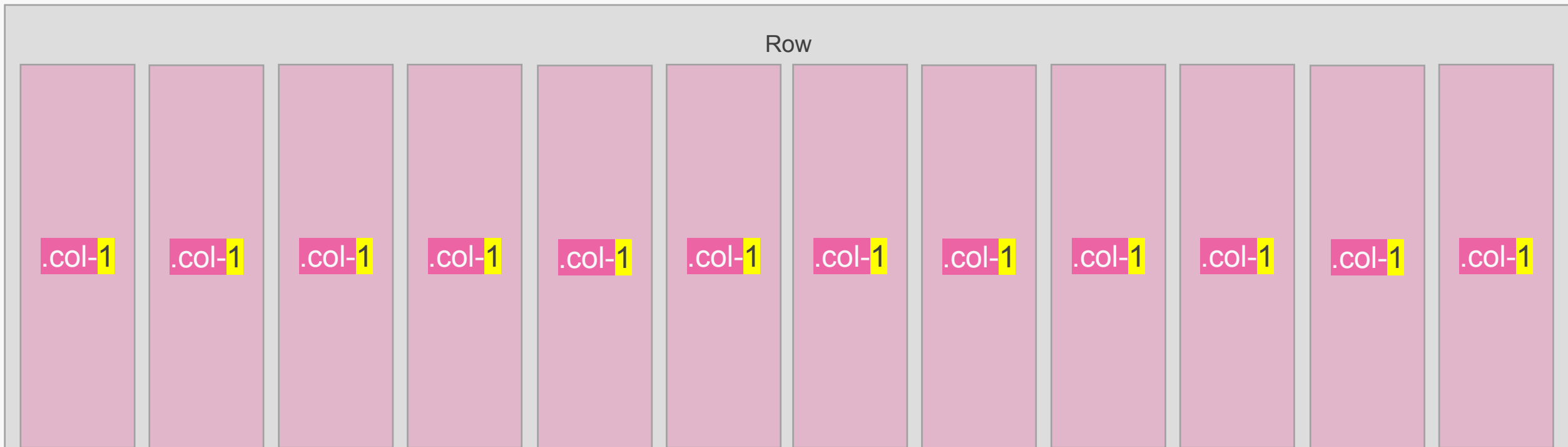
1. A `.col-1` associamo sempre un numero che corrisponde al numero di parti che una colonna deve occupare.
2. Le parti occupate in totale devono essere sempre 12!



Qualsiasi sia il valore numerico assegnato alle classi delle colonne, sommando tutti i valori assegnati alle classi di ciascuna colonna dobbiamo ottenere sempre 12. Se il risultato è maggiore o minore, c'è un errore e il layout non risulterà strutturato correttamente!

Ora tocca a voi

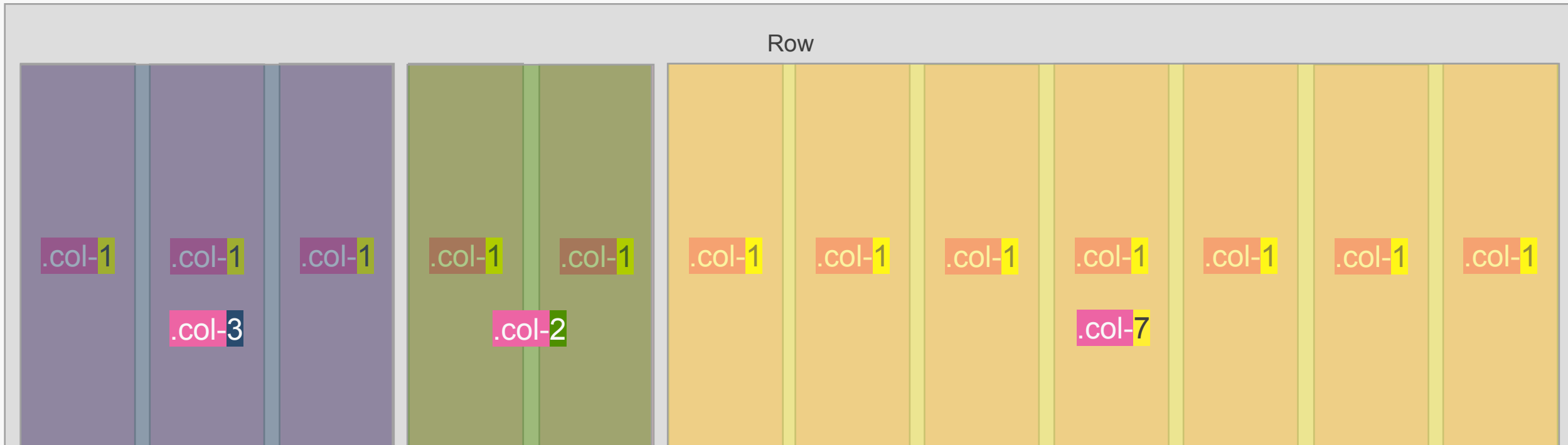
Devo dividere lo spazio in 6 parti. Quanti `<div>` devo creare e quali valori devo assegnare alle classi `.col-?`



Colonne di dimensioni differenti (larghezza)

Si possono realizzare anche colonne di dimensioni differenti e quindi assegnare un certo valore numerico a ciascuna classe `.col-1` – corrispondente al numero di parti che ciascuna colonna deve occupare. L'importante ancora una volta è che la somma di tutti i valori numerici sia 12.

Un ottimo metodo per progettare il layout è fare uno schema di 12 colonne su un foglio e provare a dividerle in diverse parti secondo le necessità.



ADVANCED: Colonne di dimensioni differenti a seconda delle dimensioni del viewport

È possibile stabilire anche quante parti debbano occupare le colonne su viewport/device di diverse dimensioni. I dispositivi sono stati suddivisi da Bootstrap nelle seguenti classi, ognuna delle quali è identificata da un infisso:

	Class infix
X-Small	None
Small	sm
Medium	md
Large	lg
Extra large	xl
Extra extra large	xxl

Inserendo uno specifico infisso fra `.col-` e il valore numerico, seguito da `-`, si specificherà che le dimensioni impostate delle colonne sono valide solo e soltanto per quella specifica classe di dispositivi.

ESEMPIO
`.col-md-` indica che questa suddivisione è da applicare solo per viewport/device medi





Regola generale = utilizzate l'infisso md

Senza addentrarci troppo nei dettagli, vi consiglio di assegnare sempre l'infisso **md** alle classi che regolano le dimensioni delle colonne. È un ottimo compromesso:

- su viewport/device più grandi le dimensioni delle colonne impostate rimarranno invariate.
- su viewport/device piccoli (generalmente su smartphone), ciascuna colonna occuperà l'intera larghezza del viewport = le colonne appariranno una sotto l'altra.

ADVANCED: Colonne di dimensioni differenti a seconda delle dimensioni del viewport/device > più classi `.col-*-*` a un solo `<div>`

	Class infix
X-Small	<i>None</i>
Small	<code>sm</code>
Medium	<code>md</code>
Large	<code>lg</code>
Extra large	<code>xl</code>
Extra extra large	<code>xxl</code>

Si possono assegnare più classi a un solo elemento `<div>` con infissi diversi. Utilizzando per esempio `.col-md-*` e `.col-sm-*`, si specificherà che le dimensioni delle colonne sono valide solo e soltanto per quella specifica classe di viewport/dispositivi.

ESEMPIO

`.col-md-` indica che questa suddivisione è da applicare solo per viewport/device medi.
`.col-sm-` per i viewport/device small.



Più righe di colonne

`<div class="container-fluid">` (o `container` a seconda delle vostre esigenze)

`<div class="row">` # identifica la 1°RIGA DELLA GRIGLIA

`<div class="col-md-4">...</div>`

`<div class="col-md-8">...</div>`

`</div>` # chiusura 1°RIGA DELLA GRIGLIA

`<div class="row">` # identifica la 2°RIGA DELLA GRIGLIA

`<div class="col-md-5">...</div>`

`<div class="col-md-7">...</div>`

`</div>` # chiusura 2°RIGA DELLA GRIGLIA

`</div>` # chiusura container

Codice HTML finale

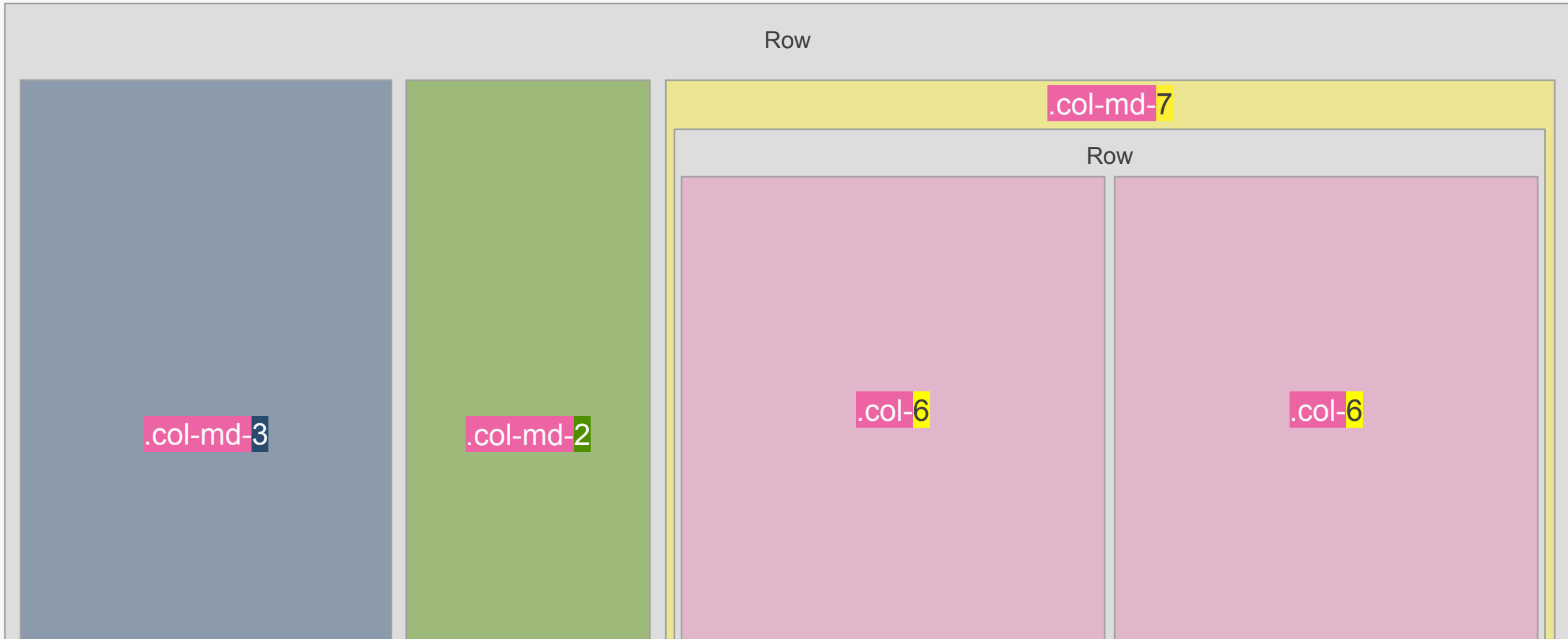
```
<!-- container -->
<div class="container-fluid">
  <!-- riga 1 -->
  <div class="row">
    <div class="col-md-4">Colonna da 4</div>
    <div class="col-md-8">Colonna da 8</div>
  </div>
  <!-- riga 2 -->
  <div class="row">
    <div class="col-md-5">Colonna da 5</div>
    <div class="col-md-7">Colonna da 7</div>
  </div>
</div>
```

Controlla il risultato online: <https://sviluppo-web-dhch.github.io/grid-system/>
Scarica il codice sorgente: <https://github.com/sviluppo-web-dhch/grid-system>



ADVANCED: Colonne innestate

È possibile innestare, e cioè annidare, più colonne all'interno di singole colonne → all'interno di una colonna `<div class="col-*">` riaprire una nuova row `<div class="row">` contenente altrettante colonne secondo la necessità disposte secondo le regole.



Codice HTML finale

```
<!-- container -->
<div class="container-fluid">
  <!-- riga 1 -->
  <div class="row">
    <div class="col-md-4">Colonna da 4</div>
    <div class="col-md-8">Colonna da 8</div>
  </div>
  <!-- riga 2 -->
  <div class="row">
    <div class="col-md-5">Colonna da 5</div>
    <div class="col-md-7">
      Colonna da 7
      <!-- colonne innestate -->
      <div class="row">
        <div class="col-md-6">Colonna innestata da 6</div>
        <div class="col-md-6">Colonna innestata da 6</div>
      </div>
    </div>
  </div>
</div>
```

Esercizio:

1. All'interno della cartella del progetto «I periodi artistici di Pablo Picasso», creare una pagina per il catalogo, es. `catalogo.html`.
2. Distinguere nel mockup della pagina *catalogo* (prossima slide) il container, le row e le colonne (innestate oppure semplici).
3. Riprodurre il layout identificato.



Catalogo

Riordina la lista

Ordina per data

Ordina per titolo (A-Z)

Ordina per titolo (Z-A)

Tutte le opere

Giovinezza

Periodo blu

Periodo rosa

Cubismo

Giovinezza 1888-1900

