

Umberto Emanuele

# Bootstrap

---

Novembre 2023



---

# Introduzione a Bootstrap

## Framework VS Librerie

La prima grande differenza tra un framework e una libreria è **chi ha il controllo del processo di sviluppo** .

Con una libreria, uno sviluppatore generalmente chiama la libreria ogni volta che lo ritiene opportuno. Un framework generalmente richiede che lo sviluppatore sia completamente immerso nel suo flusso di lavoro.

Come risultato spesso sembra che sia il framework a controllare il processo di sviluppo piuttosto che lo sviluppatore.

## Framework VS Librerie

Quando usi una libreria, sei responsabile del flusso dell'applicazione.

Sei tu a scegliere quando e dove richiamare la libreria.

Quando usi un framework, il framework è responsabile del flusso. Ti fornisce alcuni posti dove inserire il tuo codice, ma richiama il codice che hai inserito a seconda delle necessità.

## Framework VS Librerie

- Framework e librerie sono entrambi codici scritti da qualcun altro che ti aiuta ad eseguire alcuni compiti risparmiando tempo.
- Un framework **inverte il controllo del programma** . Dice allo sviluppatore cosa ha bisogno. Una libreria no. Il programmatore richiama la libreria dove e quando ne ha bisogno.

## Framework CSS VS Librerie CSS

Sentirete spesso due termini per i CSS: **librerie e framework** . Sono entrambi simili, ma ci sono alcune differenze chiave.

Una **libreria CSS** è come andare all'Ikea soltanto che il tutto è montato da un'altra persona. A volte non vuoi un tavolo da zero o di progettare le tue forchette.

I framework sono come una casa prefabbricata. Potresti essere in grado di prendere piccole decisioni estetiche qua e là ma il prodotto finale è interamente modellato dal framework.

Le librerie CSS sono più flessibili dei framework.

## Framework CSS VS Librerie CSS

### **Animate.css**

Animate.css è una libreria di animazioni CSS pronte all'uso che possono essere implementate su più browser.

### **Destyle.CSS**

Ti permette di ripristinare i margini e la spaziatura personalizzati e la dimensione dei caratteri al loro stato originale.

Destyle.css ti aiuta a implementare facilmente i tuoi progetti web su più browser.

### **Raisin.css**

Dispone di moduli pre-costruiti per FlexBox e CSS Grid. È anche completamente personalizzabile, leggera e facilmente implementabile.

## Framework CSS VS Librerie CSS

I **framework CSS** sono strumenti usati dagli sviluppatori per rendere il loro lavoro più facile. Mettono a disposizione strumenti per far funzionare rapidamente le **UI** che possono essere modificate durante lo sviluppo di un progetto invece di spendere tempo partendo da zero.

Un **framework CSS** è una collezione di CSS preparati e pronti all'uso .

Sono fatti su misura per l'uso in situazioni comuni, come l'impostazione delle navbar e sono spesso ampliati da altre tecnologie come **SASS e JavaScript** .



## Framework CSS VS Librerie CSS

Tailwind, Bulma, Materialize, Tachyons, Pure, Foundation, Skeleton, Semantic UI, UIKit, Primer...

## Bootstrap

**Bootstrap** è considerato uno dei migliori framework CSS grazie al suo design responsive.

È stato sviluppato da **Twitter** e rilasciato nel 2011 ed è gratuito.

È anche un framework di sviluppo front-end che permette a sviluppatori e designer di costruire rapidamente siti web completamente responsive.

Bootstrap ti risparmia di scrivere un sacco di codice CSS, dandoti più tempo da dedicare alla progettazione di pagine web.



## Bootstrap

Non c'è bisogno di un layout separato per la visualizzazione su mobile  
aggiungi le classi richieste ed il sito web si adatterà alla dimensione della  
finestra di visualizzazione a seconda del dispositivo.

HTML, SASS e JavaScript sono tutti configurati nel framework Bootstrap.



## Cosa c'è dentro Bootstrap?

Bootstrap fornisce una struttura di base con Grid System.

Impostazioni CSS globali, elementi HTML fondamentali stilizzati e migliorati con classi estensibili

Contiene oltre una dozzina di componenti riutilizzabili costruiti per fornire icone, dropdowns, navigazione, alert, pop-overs e molto altro.

Bootstrap contiene plugin jQuery personalizzati che puoi facilmente includerli tutti insieme o uno ad uno.

**Attenzione:** per Bootstrap 4 devi includere jQuery. Necessaria per i plugin.



## Vantaggi:

- È un framework responsive al 100%.
- Il codice di Bootstrap utilizza il preprocessore LESS e SASS.
- Grande comunità di sviluppatori.
- Ha il Grid System basato su flexbox e fornisce componenti riutilizzabili come tabelle, navbar, dropdown, ecc.
- Rari problemi di compatibilità browser.



## Svantaggi

- La dimensione della build dei progetti CSS è grande a causa dei tanti moduli caricati in Bootstrap. Non puoi selezionare i componenti specifici che vuoi. Puoi personalizzare il codice base a tuo rischio e pericolo se vuoi ridurre le dimensioni.
- I siti o i frontend costruiti con Bootstrap sembrano abbastanza simili e la variazione non è così grande in Bootstrap.



<https://getbootstrap.com>

- Documentazione ufficiale.
- Come installare ed utilizzare Bootstrap.



---

# Utilità generali: testi e box



## Testo / Colori

Bootstrap ha alcune classi che possono evidenziare il testo assegnando un particolare significato.

Questo testo è inattivo.

`.text-muted`

Questo testo è importante.

`.text-primary`

Questo testo indica il successo.

`.text-success`

Questo testo rappresenta un'informazione.

`.text-info`

Questo testo rappresenta un avvertimento.

`.text-warning`

Questo testo rappresenta un pericolo.

`.text-danger`

Testo secondario.

`.text-secondary`

Testo grigio scuro.

`.text-dark`

Colore predefinito del body (spesso nero).

`.text-body`

Puoi usare le classi `.link-*` per colorare i link.

```
<a href="#" class="link-primary"> link</a>
```

## Testo / Colori di sfondo

Oltre al colore del testo si può impostare il background color grazie a classi che seguono l'idea del significato visto per i colori del testo.

Questo testo è importante.

.bg-primary

Questo testo indica il successo.

.bg-success

Questo testo rappresenta alcune informazioni.

.bg-info

Questo testo rappresenta un avvertimento.

.bg-warning

Questo testo rappresenta un pericolo.

.bg-danger

Colore di sfondo secondario.

.bg-secondary

Colore di sfondo grigio scuro.

.bg-dark

Colore di sfondo grigio chiaro.

.bg-light

## Testo

```
<p class="bg-primary text-white">Questo testo è importante.</p>
```

```
<p class="bg-success text-white">Questo testo indica il successo.</p>
```

```
<p class="bg-info text-white">Questo testo rappresenta alcune informazioni.</p>
```

```
<p class="bg-warning text-white">Questo testo rappresenta un avvertimento.</p>
```

```
<p class="bg-danger text-white">Questo testo rappresenta un pericolo.</p>
```

```
<p class="bg-secondary text-white">Colore di sfondo secondario.</p>
```

```
<p class="bg-dark text-white">Colore di sfondo grigio scuro.</p>
```

```
<p class="bg-light text-dark">Colore di sfondo grigio chiaro.</p>
```

Questo testo è importante.

Questo testo indica il successo.

Questo testo rappresenta alcune informazioni.

Questo testo rappresenta un avvertimento.

Questo testo rappresenta un pericolo.

Colore di sfondo secondario.

Colore di sfondo grigio scuro.

Colore di sfondo grigio chiaro.

## Testi / Font

Bootstrap usa un font-size predefinito di **16px** e la font-family è "Helvetica Neue", Helvetica, Arial, sans-serif.

Tutti i tag HTML da `<h1>` a `<h6>` sono disponibili. Inoltre per creare una intestazione è possibile utilizzare le classi `.h1` fino a `.h6` per ottenere i medesimi risultati.

Le intestazioni **display** sono usate per risaltare di più delle intestazioni normali e sono presenti quattro classi: `.display -1`, `.display -2`, `.display -3` `.display -4`.

## Testi / Font

`<del>`, `<u>`, `<small>`, `<strong>`, `<em>`

Puoi usare il tag mark per evidenziare il testo.

~~Questa riga di testo deve essere trattata come testo cancellato.~~

Questa riga di testo sarà sottolineata

Questa riga di testo deve essere trattata come una stampa fine.

**Questa riga di testo verrà in grassetto.**

*Questa riga sarà in corsivo.*

## Testi / Allineamenti del testo

In Bootstrap è facile allineare i testi grazie alle classi di allineamento:

`.text-justify`, `.text-left`, `.text-center`, `.text-right` per Bootstrap 4.

`.text -start`, `text -center`, `text -end` per Bootstrap 5.

```
<p class="text-left">Testo allineato a sinistra.</p>
```

```
<p class="text-center">Allineato al centro.</p>
```

```
<p class="text-right">Allineato a destra.</p>
```

Per l'allineamento verticale puoi usare: `.align -baseline`, `.align -top`, `.align -middle`, `.align -bottom`, `.align -text -bottom`.

## Testi / Dimensioni del font

Bootstrap usa rem per unità di misura del font per la maggior parte dei suoi elementi.

Puoi comunque impostare la dimensione del font in px sull'elemento HTML nei tuoi CSS così riesci a cambiare il dimensionamento di default.

Se cambi il valore del font-size per il tag html puoi notare come il dimensionamento di tutti gli elementi testuali, come l'header, cambia rispetto all'elemento root.

## Box / Proprietà Display

La proprietà **display** in Bootstrap è usata per impostare la proprietà di visualizzazione di un elemento.

Le classi di proprietà display di Bootstrap aiutano a impostare direttamente la proprietà di visualizzazione CSS per un elemento.

**.d-block**, **.d-inline**, **.d-inline-block** sono equivalenti a `display: block;` `display: inline;` `display: inline-block;`



## Box / Margin e Padding

Bootstrap ha molte classi per modellare facilmente gli elementi HTML incluse varie classi per padding e margin per modificare l'aspetto dell'elemento.

Ci sono due modi per aggiungere la spaziatura agli elementi:

- **m**: questa proprietà definisce il margin.
- **p**: questa proprietà definisce il padding.

## Box / Margin e Padding

Le seguenti sintassi sono usate nelle varie classi per aggiungere delle spaziature:

- **(proprietà)(lati) -(dimensione)** per xs.
- **(proprietà)(lati) -(breakpoint) -(dimensione)** per sm, md, lg, e xl.

**Breakpoint** : sono punti in cui il contenuto del sito web può adattarsi in base al dispositivo e permettere di mostrare il miglior layout all'utente.

## Box / Margin e Padding

**Lati** : questo parametro permette di aggiungere la spaziatura nel contenuto ad un lato specifico.

**t** : margin-top/padding-top.

**b** : margin-bottom/padding-bottom.

**l** : margin-left/padding-left (**s** per Bootstrap 5).

**r** : margin-right/padding-right (**e** per Bootstrap 5).

**x** : per padding-left e padding-right/margin-left e margin-right.

**y** : per padding-top e padding-bottom/margin-top e margin-bottom.

**blank** : margin/padding su tutti i lati dell'elemento.

- (proprietà)(lati) -(dimensione) per xs.
- (proprietà)(lati) -(breakpoint) -(dimensione) per sm, md, lg, e xl.

## Box / Margin e Padding

**Dimensione** : questo parametro permette di aggiungere una quantità specifica di spaziatura.

**0** - 0px margin/padding.

**1** - 4px margin/padding.

**2** - 8px margin/padding.

**3** - 16px margin/padding.

**4** - 24px margin/padding.

**5** - 48px margin/padding.

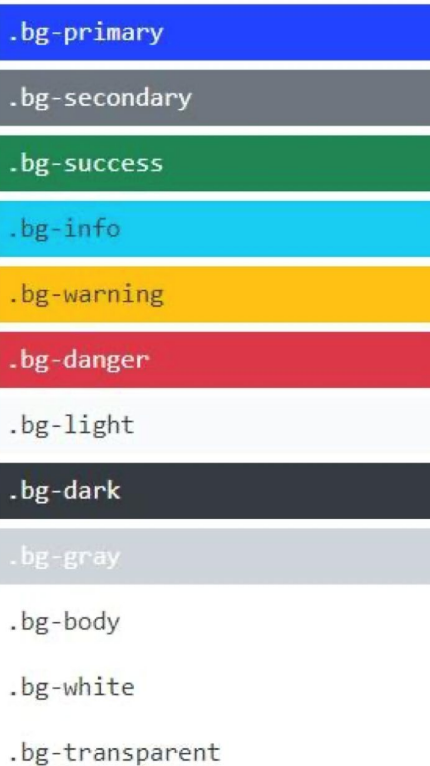
**auto** - margine automatico.

```
<div class="p-4 bg-success">Epicode School</div>
```

- (proprietà)(lati) -(dimensione) per xs.
- (proprietà)(lati) -(breakpoint) -(dimensione) per sm, md, lg, e xl.

## Box / Background Color

Come per il colore del testo puoi impostare il colore in background dando un significato in base al contesto che puoi chiaramente capire dalla nomenclatura.



## Box / Shadow e Overflow

**Shadow** è una proprietà che fornisce ombra a un elemento, l'intensità può variare da utente a utente. La proprietà shadow può essere molto utile quando l'utente ha bisogno di evidenziare qualcosa in particolare sulla pagina web.

Le classi in proporzione alla dimensione dell'ombra sono: **.shadow -none**, **.shadow -sm**, **.shadow**, **.shadow -lg**.

Puoi regolare la proprietà overflow con quattro classi predefinite: **.overflow -auto**, **.overflow -hidden**, **.overflow -visible**, **.overflow -scroll**.

## Box / Position

Il framework bootstrap ci fornisce una serie di classi che ci permettono di cambiare la posizione di un elemento. Le classi sono cinque:

- **.position -static**
- **.position -relative**
- **.position -absolute**
- **.position -fixed**
- **.position -sticky**

```
<div class="position-relative">...</div>
```

Inoltre si sono altre tre classi in aggiunta:

- Inoltre si sono altre tre classi in aggiunta:
- **fixed -top** : fa sì che l'elemento si fissi nella parte superiore della schermata, da bordo a bordo.
- **fixed -bottom** : fa in modo che l'elemento si fissi sul fondo della schermata, da bordo a bordo.
- **sticky -top** : fa sì che l'elemento si fissi nella parte superiore della schermata, da bordo a bordo, dopo lo scorrimento superato l'elemento. Altrimenti, è posizionato come statico.

## Box / Position

Inoltre si possono posizionare gli elementi in modo più preciso grazie al formato `{proprietà} -{posizione}` .

### Proprietà:

- **top** : per la posizione verticale in alto.
- **bottom** : per la posizione verticale inferiore.
- **start** : per la posizione orizzontale a sinistra.
- **end** : per la posizione orizzontale destra.

### Posizione:

- **0** - per la posizione dal bordo 0%
- **50** - per la posizione del 50% dal bordo
- **100** - per la posizione del 100% dal bordo



---

# Grid System Parte I

## Grid System

Le griglie nel web design organizzano e strutturano il contenuto rendono i siti web facili da scansionare e riducono il carico cognitivo degli utenti.

Il grid system di Bootstrap usa una serie di contenitori, righe e colonne per impaginare e allineare il contenuto. È costruito con flexbox ed è completamente responsive.

Bootstrap è mobile first nel senso che il codice di Bootstrap ha come target schermi più piccoli come i dispositivi mobili, i tablet e poi "espande" i componenti e le griglie per schermi più grandi come i computer portatili e desktop.

## Grid System

Ecco come funziona il grid system di Bootstrap:

- Le righe devono essere posizionate all'interno di una classe `.container` per un corretto allineamento e padding.
- Le righe servono per creare gruppi orizzontali di colonne.
- Il contenuto dovrebbe essere posizionato all'interno delle colonne e solo le colonne possono essere i figli diretti delle righe.
- Le colonne creano degli spazi vuoti (gutter) tra il contenuto delle colonne tramite il padding. Quel padding è compensato nelle righe per la prima e l'ultima colonna tramite un margine negativo sulla classe `.row`.
- Le colonne della griglia sono create con dodici colonne disponibili che si desidera estendere. Per esempio, due colonne uguali dovresti usare `.col-xs-6`.

	Dispositivi extra small (telefoni)( $<768\text{px}$ )	Dispositivi piccoli (tablets) ( $\geq 768\text{px}$ )	Dispositivi medi (desktop) ( $\geq 992\text{px}$ )	Dispositivi larghi (desktop) ( $\geq 1200\text{px}$ )
<b>Comportamento della griglia</b>	Orizzontale in ogni momento	Collassato all'inizio, orizzontale sopra i breakpoint	Collassato all'inizio, orizzontale sopra i breakpoint	Collassato all'inizio, orizzontale sopra i breakpoint
<b>Larghezza massima del contenitore</b>	Nessuna (auto)	750px	970px	1170px
<b>Prefisso di classe</b>	.col-xs-	.col-sm-	.col-md-	.col-lg-
<b>Numero di colonne</b>	12	12	12	12

## Grid System / Breakpoint

I breakpoint sono punti in cui il contenuto del sito web può adattarsi in base al dispositivo e permettere di mostrare il miglior layout all'utente.

Bootstrap include sei breakpoint predefiniti. Questi breakpoint possono essere anche personalizzati nei file Sass originali.

Ogni breakpoint è stato creato per contenere comodamente dei contenitori le cui larghezze sono multipli di 12.

## Grid System / Breakpoint

Questi breakpoint sono personalizzabili tramite Sass, li trovi in `_variables.scss` .

Breakpoint	Class infix	Dimensione
X-Small	None	<576px
Small	sm	≥576px
Medium	md	≥768px
Large	lg	≥992px
Extra large	xl	≥1200px
Extra extra large	xxl	≥1400px

## Grid System / Container

I contenitori sono l'elemento di layout più basilare in Bootstrap e sono necessari quando si usa il nostro grid system. Quindi contiene elementi riga e gli elementi riga sono contenitori di colonne.

Bootstrap è dotato di tre diversi contenitori:

- `.container`: che imposta un max-width ad ogni breakpoint responsive.
- `.container-fluid`: con width al 100% in tutti i breakpoint.
- `.container-{breakpoint}`: con width al 100% fino al breakpoint specificato.

## Grid System / Container.

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	X-Large ≥1200px	XX-Large ≥1400px
<code>.container</code>	100%	540px	720px	960px	1140px	1320px
<code>.container -sm</code>	100%	540px	720px	960px	1140px	1320px
<code>.container -md</code>	100%	100%	720px	960px	1140px	1320px
<code>.container -lg</code>	100%	100%	100%	960px	1140px	1320px
<code>.container -xl</code>	100%	100%	100%	100%	1140px	1320px
<code>.container -xxl</code>	100%	100%	100%	100%	100%	1320px
<code>.container -fluid</code>	100%	100%	100%	100%	100%	100%



## Grid System / Row e Column

**Row:** le righe devono essere posizionate all'interno di un "container" o "container-fluid" per un corretto allineamento e padding. Si usano le righe per creare gruppi orizzontali di colonne.

```
<div class="row">  
  <div class="bg bg-success w-100">  
    Questa è una riga  
  </div>  
</div>
```

**Colonne:** le colonne sono create in numero di dodici disponibili che si può estendere. Per esempio, tre colonne uguali saranno tre "col-sm-4".

```
<div class="col-sm-4">  
  <div class="bg bg-danger">  
    Colonna  
  </div>  
</div>
```

## Grid System / Row e Column

Puoi usare le classi responsive **.row - cols - \*** per impostare rapidamente il numero di colonne che si adattano meglio al tuo layout.

Mentre le normali classi **.col - \*** si applicano alle singole colonne (ad esempio, `.col-md-4`), le classi `row-cols-*` sono impostate sul genitore `.row` come default per le colonne contenute.

Con **.row - cols - auto** puoi dare alle colonne la loro larghezza naturale.

## Grid System / Auto -layout

Puoi anche evitare di utilizzare grandezze per le colonne numerate come `.col-sm-4` ed utilizzare i breakpoint in modo più semplice. Aggiungi qualsiasi numero di classi senza numero specificato per ogni breakpoint necessario e ogni colonna avrà la stessa larghezza.

Prova ad implementare il seguente codice:

```
<div class="container">
  <div class="row">
    <div class="col">
      Prima colonna
    </div>
    <div class="col">
      Seconda colonna
    </div>
  </div>
</div>
```

### Grid System / Auto -layout

L'auto-layout per le colonne del grid system significa anche che puoi impostare la larghezza di una colonna e avere le colonne sorelle che si **ridimensionano automaticamente intorno ad essa** .

Le altre colonne si ridimensionano indipendentemente dalla larghezza della colonna.

Usa le classi **col-{breakpoint} -auto** per ridimensionare le colonne in base alla larghezza naturale del loro contenuto.

Per le griglie che sono uguali dal più piccolo dei dispositivi al più grande usa le classi **.col** e **.col -\***.

Specifica una classe numerata quando hai bisogno di una colonna di dimensioni particolari, altrimenti, puoi usare **.col** .

---

# Grid System Parte II

## Grid System / Allineamento delle colonne

Le colonne si possono allineare **verticalmente ed orizzontalmente** grazie alle utility di allineamento **flexbox**.

Usa le classi **.align-items-start**, **.align-items-center**, **.align-items-end** associate alla classe **.row** per **allineare verticalmente** le colonne di una riga intera, vedi l'esempio sottostante:

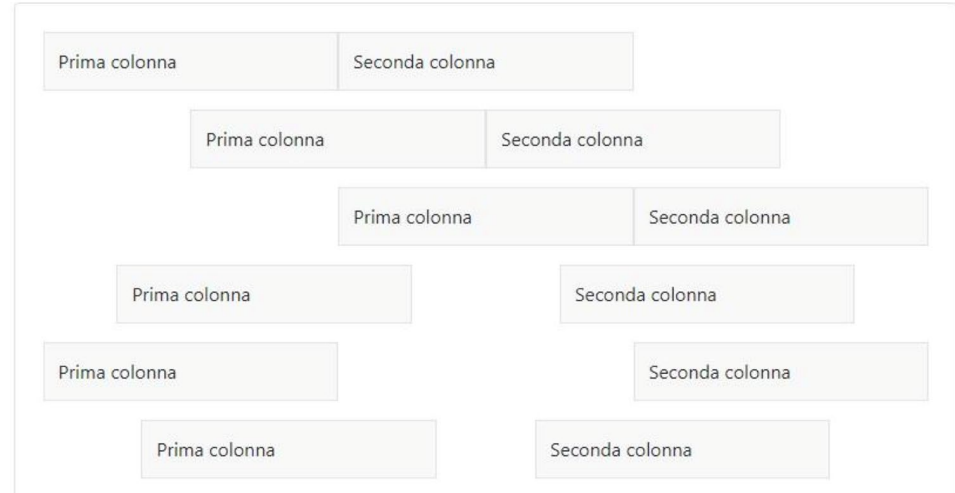
```
<div class="row align-items-start">
```

Usa le classi **.align-self-start**, **.align-self-center**, **.align-self-end** associate alla classe **.col** per **allineare verticalmente** una colonna singolarmente in una riga, vedi l'esempio sottostante:

```
<div class="col align-self-start">
```

### Grid System / Allineamento delle colonne

Usa le classi `.justify -content -start` , `.justify -content -center` , `.justify -content -end` , `.justify -content -around` , `.justify -content -between` , `.justify -content -start` a livello di `.row` per allineare le colonne orizzontalmente.



## Grid System / Allineamento delle colonne

Posizionare le colonne su una nuova riga in flexbox richiede un piccolo trucco.

Aggiungi un elemento con **width: 100%** (**classe .w - 100**) ovunque tu voglia posizionare le tue colonne su una nuova riga.

Usa le classi **.order** - per controllare l'ordine del tuo contenuto. Queste classi sono responsive, quindi puoi impostare l'ordine per breakpoint (ad esempio, `.order-1.order-md-2`). Include il supporto da 1 a 5 per tutti e sei i livelli della griglia. `<div class="col order-5">`

Ci sono anche le classi responsive **.order -first** e **.order -last** che cambiano l'ordine di un elemento applicando `order: -1` e `order: 6`, rispettivamente.



## Grid System / Offset delle colonne

Hai la possibilità di spostare le colonne verso destra usando le classi `.offset -md -*`.

Queste classi aumentano il margine sinistro, di una colonna, di \* colonne.

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

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

.col-md-4

.col-md-4 .offset-md-4

## Grid System / Gutter

Il **gutter** è la spaziatura, padding, tra le colonne, usato per spaziare ed allineare il contenuto in modo responsive. Puoi impostare padding-right e padding-left su ogni colonna, usa margin negativo per compensare, all'inizio ed alla fine di ogni riga per allineare il contenuto.

Il gutter parte da **1.5rem (24px)** di larghezza.

Le classi **.gx-\*** possono essere usate per controllare le larghezze dei gutter orizzontali.

Le classi **.gy-\*** possono essere usate per controllare le larghezze dei gutter verticali.

	BOOTSTRAP 4	BOOTSTRAP 5
<b>Grid System</b>	Ha 5 livelli (xs, sm, md, lg, xl).	Ha 6 livelli (xs, sm, md, lg, xl, xxl).
<b>Colori</b>	Ha colori limitati.	Colori extra aggiunti. Una tavolozza di colori migliorata.
<b>jQuery</b>	Ha jQuery e tutti i plugin correlati.	jQuery è stato rimosso ed è passato a vanilla JS con alcuni plugin funzionanti.
<b>Internet Explorer</b>	Bootstrap 4 supporta sia IE 10 che 11.	Bootstrap 5 non supporta IE 10 e 11.
<b>Elementi Form</b>	I pulsanti radio e le caselle di controllo hanno un aspetto diverso in diversi sistemi operativi e browser.	L'aspetto degli elementi dei form non cambierà, su diversi sistemi operativi o browser.

<b>Utilità</b>	Non possiamo modificare le utilità in bootstrap 4	Bootstrap 5 ha dato la libertà di modificare e anche di creare la nostra utilità
<b>Gutter</b>	Si usa <code>.gutter</code> con fontsize in px	Si usa <code>.g*</code> con fontsize in rem
<b>Classi verticali</b>	Le colonne possono essere posizionate relativamente	Le colonne non possono essere posizionate relativamente
<b>Icone</b>	Bootstrap 4 non ha le proprie icone SVG	Bootstrap 5 ha le sue icone SVG
<b>Jumbotron</b>	Supportato	Non supportato

---

# Elementi di layout

## Navbar

Una **barra di navigazione** (o navbar) è la parte superiore della pagina che viene utilizzata per identificare la struttura di un sito web e di conseguenza la possibilità di navigare in essa. Una barra di navigazione può estendersi o collassare, a seconda delle dimensioni dello schermo.

Una barra di navigazione standard è creata con la classe `.navbar` seguita da una classe responsive: `.navbar-expand{-sm|-md|-lg|-xl|-xxl}` (impila la barra di navigazione verticalmente su schermi extra large, large, medium o small).

Le navbar e i loro contenuti sono fluidi e responsive per impostazione predefinita. Ma puoi facilmente modificare questo comportamento che dipende dal plugin **JavaScript Collapse** .

Assicurare l'accessibilità utilizzando il tag `<nav>` o, se si utilizza un elemento più generico come un `<div>`, aggiungere un `role="navigation"`.

## Navbar

Per aggiungere link all'interno della navbar, usate un elemento `<ul>` con `class="navbar-nav"`.

Poi aggiungere elementi `<li>` con una classe **.nav-item** seguita da un elemento `<a>` con una classe **.nav-link**.

```
<nav class="navbar navbar-expand-sm|md bg-light">
<ul class="navbar-nav">
  <li class="nav-item">
    <a class="nav-link" href="#">Link 1</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Link 2</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Link 3</a>
  </li>
</ul>
</nav>
```

## Navbar

Le barre di navigazione sono dotate di sottocomponenti e sono:

- **.navbar -brand** : spazio dedicato al logo.
- **.navbar -nav** : una barra completa a tutta altezza (compreso il supporto per i dropdown).
- **.navbar -toggler** : per l'uso alternato del plugin collapse ed altri comportamenti di navigazione.
- Utilità **flex e spacing** per qualsiasi controllo e azione del modulo.
- **.navbar -text** : per aggiungere stringhe di testo centrate verticalmente.
- **.collapse.navbar -collapse** : per raggruppare e nascondere i contenuti della navbar in base a un punto di interruzione.
- Aggiungi **.navbar -nav -scroll** a un **.navbar -nav** (o ad un altro sottocomponente della navbar) per abilitare lo scorrimento verticale all'interno dei contenuti di una barra di navigazione collassata. Per impostazione predefinita, lo scorrimento inizia a **75vh** (o 75% dell'altezza del viewport), ma è possibile sovrascriverlo con la proprietà personalizzata CSS **--bs-navbar -height** o con stili personalizzati.
- **Scrollspy** è una possibilità che ti permette di aggiornare il menu di navigazione in base alla posizione della pagina evidenziando il link corrente.



## Navbar

La personalizzazione della navbar non è mai stata così facile grazie alla combinazione di classi per i colori di sfondo.

Puoi utilizzare **.navbar -light** per un background chiaro o **.navbar -dark** per colori di sfondo scuri.

Poi anche personalizzarla con le classi **.bg -\***.

Usa le utility di posizione per mettere la navbar in posizioni non statiche. Scegli tra **fixed -top**, **fixed -bottom** o **sticky -top**. Le navbar fisse usano **position: fixed**, il che significa che sono estratte dal normale flusso del DOM e possono richiedere CSS personalizzati.

## Navbar

Combinando alcuni attributi è possibile creare facilmente un'interfaccia a schede (**tab**) .

Per creare un menu di navigazione a schede inserisci un elenco non ordinato con la classe **.nav** e poi aggiungere la classe **.nav-tabs** .

```
<ul class="nav nav-tabs">  
  <li class="nav-item">  
    <a class="nav-link active" aria-current="page"  
href="#">Attivo</a>  
  </li>  
  <li class="nav-item">  
    <a class="nav-link" href="#">Link 1</a>  
  </li>  
  <li class="nav-item">  
    <a class="nav-link" href="#">Link 2</a>  
  </li>  
</ul>
```

## Navbar

Le **breadcrumb** , in italiano briciole di pane, servono a far vedere al visitatore la pagina corrente e la gerarchia di navigazione, come è arrivato a quella determinata pagina.

Per crearle si utilizza sempre il tag `<nav>` e `<ol>` o `<ul>` con elementi di lista collegati per creare un breadcrumb minimale.

```
<nav aria-label="breadcrumb">
  <ol class="breadcrumb">
    <li class="breadcrumb-item"><a href="#">Home</a></li>
    <li class="breadcrumb-item active" aria-current="page">Prima pagina</li>
  </ol>
</nav>
```

I divisori sono aggiunti automaticamente nei CSS attraverso `::before` . Possono essere cambiati modificando una proprietà CSS -- **bs-breadcrumb-divider** .

```
<li class="breadcrumb-item active">Home</li>
```

## Paginazione

Se hai un sito web con molte pagine, potresti voler aggiungere una sorta di paginazione ad ogni pagina. Per indicare che una serie di contenuti correlati esistono su più pagine.

La paginazione è costruita con elementi HTML `<li>` in modo che i screen reader possano annunciare il numero di link disponibili. Il tutto inserito dentro un tag `<nav>`.

Si usano le classi `.pagination` per `<ul>`, `.page-item` per i `<li>` e per l'anchor tag `.page-link`.

```
<nav aria-label="Esempio pagina di navigazione">
  <ul class="pagination">
    <li class="page-item"><a class="page-link" href="#">Prima</a></li>
    <li class="page-item"><a class="page-link" href="#">1</a></li>
    <li class="page-item"><a class="page-link" href="#">2</a></li>
    <li class="page-item"><a class="page-link" href="#">Dopo</a></li>
  </ul>
</nav>
```

## Paginazione

I link di paginazione sono personalizzabili. Usa **.disabled** per i link non cliccabili e **.active** per indicare la pagina corrente.

```
<nav aria-label="Esempio pagina di navigazione">
  <ul class="pagination">
    <li class="page-item" disabled><a class="page-link" href="#">Prima</a></li>
    <li class="page-item" active><a class="page-link" href="#">1</a></li>
    <li class="page-item"><a class="page-link" href="#">2</a></li>
    <li class="page-item"><a class="page-link" href="#">Dopo</a></li>
  </ul>
</nav>
```

## Elementi modali

Gli **elementi modali** o finestre modali sono ad esempio le notifiche, i lightboxes.

I modali sono costruiti con HTML, CSS e JavaScript. Sono posizionati sopra tutto il resto del documento e rimuovono lo scroll dal `<body>` in modo che il contenuto modale scorra al suo posto.

Bootstrap supporta solo una finestra modale alla volta. Le finestre modali annidate non sono supportate.

I modali usano **position: fixed**. Quando possibile, metti l'elemento modale in una posizione primaria per evitare potenziali interferenze da altri elementi.

Per creare una finestra modale si utilizzano le seguenti classi con i `<div>` annidati: **.modal**, **.modal -dialog**, **.modal -content**, **.modal -header**, **.modal -title**, **.modal -body**, **.modal -footer**.

---

# Il contenuto

## Immagini

Bootstrap offre diverse classi per le immagini per migliorare il loro aspetto e per renderle responsive.

Ti ricordo che rendere un'immagine responsive significa che dovrebbe scalare secondo il suo elemento genitore. Cioè la dimensione dell'immagine non dovrebbe superare il suo elemento genitore e crescerà e si ridurrà in base al cambiamento della dimensione del suo genitore senza perdere il suo aspect ratio.

La classe **.img-fluid** è usata all'interno del tag `<img>` per creare l'immagine responsive. Questa classe applica `max-width: 100%` e `height: auto`.

```

```



## Immagini

La classe **.rounded** è usata per creare un'immagine con gli angoli arrotondati. Questa classe è usata con il tag `<img>`.

La classe **.img-thumbnail** è usata per creare un'immagine in miniatura (con bordo).

Per allineare un'immagine a destra e sinistra puoi utilizzare le classi **.float-right** e **.float-left** sempre nel tag `<img>`.

Le classi **.mx-auto** (`margin:auto`) e **.d-block** (`display:block`) sono usate per centrare l'immagine.

## Tabelle

Bootstrap fornisce una serie di classi che possono essere utilizzate per applicare vari stili alle tabelle, come cambiare l'aspetto dell'intestazione, delle righe, aggiungere o rimuovere i bordi.

Inoltre è possibile rendere le tabelle responsive.

Per creare una semplice tabella aggiungi la classe **.table** all'interno del tag `<table>` come indicato di seguito:

```
<table class="table">
```

Puoi usare le classi semantiche per colorare le tabelle, le righe della tabella o le singole celle.

```
<table class="table-primary">...</table>
```

```
<tr class="table-primary">...</tr>
```

```
<td class="table-primary">...</td>
```

Aggiungere **.table -hover** per abilitare uno stato hover sulle righe della tabella all'interno di un `<tbody>`.

## Form

In generale per creare un form si utilizzano `<input>`, `<select>`, `<textarea>` per l'aspetto generale per il rispettivo campo di input, elemento di selezione e textarea.

La classe usata per il controllo form è **`.form-control`**.

La dimensione del controllo del form può essere regolata utilizzando `.form-control-lg` e `.form-control-sm`.

La classe utilizzata per le etichette è **`.form-label`**. Per quanto riguarda invece un testo informativo da posizionare ad esempio sotto il campo di input puoi usare la classe **`.form-text`**. Per l'input select puoi usare **`.form-select`**.

Il tipo di input file può essere definito con la classe **`.form-control-file`**. Il tipo di input range può essere definito con la classe **`.form-control-range`**.

## Form

La classe usata per i radio e le checkbox è una sola **.form-check**. Per default è impostata su checkbox, per impostare radio bisogna specificarlo con l'attributo `type` nell'input.

Aggiungi l'attributo **disabled** su un input per dargli un aspetto grigio e rimuovere gli eventi del puntatore.

Per impostare la selezione di un colore usa la classe **.form-control-color** sempre nell'input dopo la classe **form-control** specificando nell'attributo `value` il colore.

Form più complessi possono essere costruiti usando le classi griglia. Usale per i layout dei form che richiedono colonne multiple, larghezze diverse e opzioni di allineamento aggiuntive.

Infine puoi usare le classi **.row-cols-\*** per creare layout orizzontali responsive.

## Form

I gruppi di input (group input) in Bootstrap sono utilizzati per estendere i controlli predefiniti dei form aggiungendo testo, pulsanti, su entrambi i lati di input testuali o input personalizzati.

Le seguenti classi sono le classi base che vengono utilizzate per aggiungere i gruppi ai lati delle caselle di input.

- La classe **.input-group-prepend** è usata per aggiungere i gruppi davanti l'input.
- La classe **.input-group-append** è usata per aggiungerli alla fine dell'input.
- La classe **.input-group-text** è usata per definire il testo che viene visualizzato all'interno del gruppo.

Puoi inserire checkbox o radio all'interno di un gruppo di input al posto del testo. Aggiungi **.mt-0** al **.form-check-input** quando non c'è testo visibile accanto all'input.

## Form

Riprendendo direttamente la comunicazione attuale sulla documentazione ufficiale di Bootstrap:

“Siamo consapevoli che attualmente gli stili di convalida personalizzati lato client e i tooltip non sono accessibili. Mentre lavoriamo su una soluzione, raccomandiamo di usare l'opzione lato server o il metodo di validazione predefinito del browser.”

La validazione dei moduli HTML è applicata tramite due pseudo-classi CSS, **:invalid** e **:valid**. Si applica agli elementi `<input>`, `<select>` e `<textarea>`.

Quando provi ad effettuare l'invio vedrai i CSS **:invalid** e **:valid** applicati al form.

## Icone

Puoi includerle nel tuo codice HTML in vari modi in base al tipo di progetto.

- **Copia incolla l'SVG come HTML incorporato** : incorpora le tue icone all'interno dell'HTML della tua pagina.
- **Sprite SVG con elemento `<use>`** : usa il nome del file dell'icona come identificatore dell'icona (ad esempio, il cuore è `#heart`). Gli sprite SVG ti permettono di fare riferimento a un file esterno in modo simile ad un elemento `<img>`, ma con la potenza dell'attributo `fill` e di `currentColor` per una facile personalizzazione.
- **Come immagine esterna** : copia le icone di Bootstrap in qualsiasi directory tu voglia ed utilizza il tag `<img>` come le normali immagini.

Le icone includono una larghezza e un'altezza di **1em** di default per permettere un facile ridimensionamento tramite il `font-size`.

---

## **Altri elementi utili**



## Buttons

Qualsiasi cosa a cui viene data una classe **.btn** erediterà l'aspetto predefinito di un pulsante grigio con angoli arrotondati. Tuttavia Bootstrap fornisce alcune opzioni per lo stile dei pulsanti. Ognuno delle quali serve al proprio scopo semantico.

- **.btn** : pulsante default.
- **.btn -primary** : identifica l'azione primaria in un insieme di pulsanti.
- **.btn -success** : indica un'azione riuscita o positiva.
- **.btn -info** : pulsante per messaggi informativi di avviso.
- **.btn -warning** : indica che bisogna fare attenzione con questa azione.
- **.btn -danger** : indica un'azione pericolosa o potenzialmente negativa.
- **.btn -link** : sottolinea un pulsante facendolo apparire come un link, pur mantenendo il comportamento del pulsante.

Inoltre è possibile avere un pulsante solo con i bordi colorati senza riempimento sostituendo le classi sopra indicate con **.btn-outline-\***. Sostituendo l'asterisco con **primary**, **success**, etc.

## Buttons



## Buttons

Per impostare le dimensioni dei pulsanti puoi usare le seguenti classi:

- **.btn -lg**: rende la dimensione del pulsante grande.
- **.btn -sm**: rende la dimensione del pulsante piccola.
- **.btn -xs**: rende la dimensione del pulsante extra piccola.
- **.btn -block** : questo crea pulsanti a livello di blocco, coprono l'intera larghezza dell'elemento genitore.

```
<button type = "button" class = "btn btn-primary btn-lg">
```

```
    Pulsante grande primary
```

```
</button>
```

## Buttons

Bootstrap fornisce classi che permettono di cambiare lo stato dei pulsanti come attivo e disabilitato.

**Stato attivo** : i pulsanti appariranno premuti (con uno sfondo più scuro) quando sono attivi. Sia per il `<button>` che per `<a>` la classe è la medesima, **.active**. Ricorda di inserire la classe `aria-pressed="true "` per le tecnologie assistive.

**Stato disabilitato** : quando si disabilita un pulsante, il suo colore sbiadisce del 50%. Sia per il `<button>` che per `<a>` la classe è la medesima, **.disabled** . Ricorda di inserire la classe `aria-disabled="true "` per le tecnologie assistive.

**Attenzione** : il `disabled` su `<a>` disabilita solo l'aspetto non la sua funzionalità.

## Alert

Gli **alert** forniscono un modo per stilizzare dei messaggi per l'utente. Forniscono feedback contestuali alle azioni tipiche dell'utente.

Puoi aggiungere un avviso creando un `<div>` contenitore e aggiungendo la classe **.alert** e una delle quattro classi semantiche: **.alert -success** , **.alert -info** , **.alert -warning** , **.alert -danger** .

Per creare un alert che si può chiudere devi soddisfare le seguenti istruzioni:

- Aggiungi un alert creando un `<div>` contenitore e aggiungendo una classe di **.alert** e una delle quattro classi semantiche.
- Aggiungi anche **.alert -dismissable** al `<div>` di cui sopra.
- Aggiungi un pulsante di chiusura con class **.close** e l'attributo **data -dismiss="alert"** e **aria -hidden="true"** .

Puoi usare la classe **.alert -link** per fornire rapidamente collegamenti colorati corrispondenti al contenuto di qualsiasi alert.

## Tab

Le navigazioni a schede (**tab**) sono uno dei modi più efficaci per visualizzare una grande quantità di contenuti su una singola pagina web.

L'utente può accedere rapidamente al contenuto passando da un pannello all'altro senza lasciare la pagina.

Per creare un menu di navigazione a schede, crea un elenco non ordinato di base con le voci dell'elenco come link. Poi aggiungi le classi **.nav** e **.nav-tabs** alla lista non numerata e la classe **.nav-items** agli elementi della lista.

## Accordion

Gli accordion sono utili quando si vuole passare da nascondere a mostrare una certa quantità di contenuto.

L'accordion usa internamente il collapse per renderlo richiudibile. Per espanderlo aggiungi la classe **.open** su **.accordion** .

Per crearlo devi utilizzare le seguenti classi: **.accordion** , **.accordion -item** , **.accordion -header** , **.accordion -body** e **.accordion -button** nei rispettivi `<div>`.

## Tooltip

Un **tooltip** è usato per fornire all'utente suggerimenti testuali interattivi su un determinato elemento HTML e si visualizzano quando il puntatore del mouse vi passa sopra.

L'attributo **data-bs-toggle="tooltip"** è usato per creare un tooltip. L'attributo **title** è usato per specificare il testo che dovrebbe essere visualizzato all'interno del tooltip.

Un modo per inizializzare tutti i tooltip di una pagina sarebbe quello di selezionarli in base al loro attributo **data-bs-toggle**. Questo si può fare con JavaScript puro oppure con jQuery (che consiglio per la semplicità).

```
<div class="container">
  <h1 data-bs-toggle="tooltip" title="Epico!">Epicode School</h1>
</div>
<script>
  $(document).ready(function() {
    $('[data-bs-toggle="tooltip"]').tooltip();
  });
</script>
```



## Tooltip

L'attributo **data-bs-placement** è usato per impostare la posizione dell'attributo tooltip. L'elemento tooltip può essere visualizzato sul lato superiore, sinistro, destro o inferiore dell'elemento HTML.

Es.: `data-bs-placement="top"`.

Ai tooltip si possono aggiungere anche delle opzioni. Le opzioni possono essere passate tramite attributi di dati o JavaScript. Per gli attributi di dati, aggiungi il nome dell'opzione a **data-bs-**, come in **data-bs-animation=""** .

## Tooltip

Nome	Type	Default	Descrizione
<b>animation</b>	boolean	true	Applica una transizione fade (dissolvenza) al tooltip.
<b>container</b>	string   element   false	false	Aggiunge il tooltip ad un elemento specifico. Esempio: container: 'body'. Questa opzione è particolarmente utile in quanto permette di posizionare il tooltip nel flusso del documento vicino all'elemento di attivazione. Questo eviterà che il tooltip si allontani dall'elemento durante un ridimensionamento della finestra.
<b>delay</b>	number   object	0	Applica un ritardo nel mostrare e nascondere il tooltip (in ms). Se viene fornito un solo numero, il ritardo viene applicato sia nel nascondere che nel mostrare. La struttura dell'oggetto è: delay: { "show": 500, "hide": 100 }

## Cards

La **card (scheda)** è un componente che fornisce un contenitore di contenuto flessibile ed estensibile con molteplici varianti e opzioni.

Le card supportano un'ampia varietà di contenuti, tra cui immagini, testo, gruppi di liste, link e altro.

Il componente di base di una card è la classe **.card -body** unita alla classe **.card** come genitore possiamo creare una card base.

La classe **.card -header** fornisce l'intestazione alle carte e **.card -footer** fornisce il piè di pagina.

**.card -title** è usata per dare un titolo alla scheda, **.card -text** per contenuto testuale e **.card -link** è usata per fornire un link se richiesto.

Le immagini nella card sono inserite con **.card -img -top** e con **.card -img -bottom** dentro il tag `<img>`.

Infine puoi creare liste di contenuti in una scheda con **.list -group** con elementi sovrapposti.

## Carousel

Il **carousel** è uno slideshow per scorrere una serie di contenuti. Al momento non sono supportati correttamente in Internet Explorer, questo perché usano transizioni e animazioni CSS3.

Carousel è un modo flessibile e responsive per aggiungere uno slider al tuo sito.

Le classi per creare il carosello con Bootstrap in ordine nei rispettivi `<div>` sono: **.carousel** associata con **.slide**, **.carousel -inner**, **.carousel -item** (il primo elemento deve avere anche **.active** altrimenti il carosello non sarà visibile). **.d-block** e **.w-100** sulle immagini del carosello per evitare l'allineamento predefinito del browser.

Se vuoi aggiungere i controlli nelle slide usa `<button>` ma puoi usare `<a>` con `role="button"`. Le classi da utilizzare sono **.carousel -control -next(o prev)**, **.carousel -control -next -icon**.

Gli elementi controllo e gli eventuali indicatori (identificati dalla classe **.carousel -indicators**) devono avere un attributo **data -bs -target** (o `href` per i link) che corrisponde all'id dell'elemento **.carousel**.

## Carousel

Lo scopo delle barre di avanzamento (**progress bar** ) è quello di mostrare che le risorse si stanno caricando, in corso, o che c'è un'azione in corso per quanto riguarda gli elementi della pagina.

Per creare una barra di avanzamento serve:

- Aggiungere un `<div>` con la classe **.progress** .
- All'interno del `<div>` aggiungi un `<div>` con la classe **.progress -bar** .
- Aggiungi un attributo `style` con `width` in percentuale. Per esempio, **style = "40%"** ; indica che la barra di progresso è al 40%.

Per creare una barra di progresso con diversi colori aggiungi la classe **progress -bar -\*** dove **\*** può avere i classici `primary`, `success`, `info`, `warning`, `danger`, etc.

---

# Customizzazione base

## Flex

**Flex** (che richiama direttamente il flexbox, Flexible Box Layout) è usato per progettare la struttura per un layout flessibile e responsive.

La classe **.d-flex** è usata per creare un semplice contenitore flexbox. La classe **.d-inline-flex** è usata per creare un contenitore flexbox in linea.

**.d-flex** e **.d-inline-flex** possono essere usati per tutti i breakpoint (sm, md, lg, xl) ad esempio **.d-sm-flex**, **.d-sm-inline-flex**, ecc.

La direzione degli elementi può essere posizionata direttamente in un contenitore flex.

Per impostare una direzione orizzontale da sinistra verso destra usa **.flex-row**. Invece per creare l'inverso usa **.flex-row-reverse**.

Per impostare una direzione verticale usa **.flex-column**. Per l'inverso usa **.flex-column-reverse**.

Le varie direzioni possono essere usate per tutti i breakpoints (sm, md, lg, xl) come ad esempio **.flex-lg-row**, **flex-lg-row-reverse**, ecc.

## Flex

Per cambiare l'allineamento degli elementi flex si può usare la classe **.justify -content -\***.

L'asterisco può essere uno qualsiasi tra **start** , **end** , **center** , **between** o **around** .

Allo stesso modo anche **justify -content** può essere usato per tutti i breakpoint (sm, md, lg, xl) da inserire prima della definizione dell'allineamento.

Gli elementi Flex possono essere forzati ad avere larghezze uguali usando la classe **.flex -fill**.

**.flex-fill** può essere usato per tutti i breakpoint come per le altre classi.



## CSS custom

**Le proprietà custom (a volte chiamate variabili CSS)** sono entità definite dagli autori CSS che contengono valori specifici da riutilizzare in un documento.

Sono impostate usando la notazione di proprietà personalizzata (ad esempio, `--backgr-color: blue;`).

Come hai potuto notare un CSS custom è fatto usando un nome per la proprietà personalizzata che inizia con un doppio trattino (--), una proprietà che può essere qualsiasi CSS valido.

Bootstrap include proprietà personalizzate (CSS custom) per una migliore personalizzazione. Tutte le proprietà CSS custom hanno il prefisso **bs-** per evitare conflitti con CSS di terze parti.

## CSS custom

```
--bs-blue: #0d6efd;  
--bs-indigo: #6610f2;  
--bs-purple: #6f42c1;  
--bs-pink: #d63384;  
--bs-red: #dc3545;  
--bs-orange: #fd7e14;  
--bs-yellow: #ffc107;  
--bs-green: #198754;  
--bs-teal: #20c997;  
--bs-cyan: #0dcaf0;  
--bs-white: #fff;  
--bs-gray: #6c757d;  
--bs-gray-dark: #343a40;  
--bs-primary: #0d6efd;  
--bs-secondary: #6c757d;  
--bs-success: #198754;  
--bs-info: #0dcaf0;  
--bs-warning: #ffc107;  
--bs-danger: #dc3545;  
--bs-light: #f8f9fa;  
--bs-dark: #212529;
```

```
--bs-font-sans-serif: system-ui, -apple-system, "Segoe UI", Roboto, "Helvetica Neue", Arial, "Noto Sans",  
"Liberation Sans", sans-serif, "Apple Color Emoji", "Segoe UI Emoji", "Segoe UI Symbol", "Noto Color Emoji";  
--bs-font-monospace: SFMono-Regular, Menlo, Monaco, Consolas, "Liberation Mono", "Courier New", monospace;
```

## CSS Override

Puoi sovrascrivere gli stili predefiniti degli elementi di Bootstrap usando due metodi.

Il primo usando **CSS override** si applica ai siti che usano BootstrapCDN o le versioni precompilate di Bootstrap. Il secondo usando le variabili Sass e si applica ai siti che usano la versione con codice sorgente.

Ogni metodo è utile e dipende dai casi d'uso. L'uso del CSS override è **più semplice ed anche più dispendioso in termini di tempo**. Usare le **variabili Sass**, d'altra parte, è ideale **per fare cambiamenti più estesi** in tutto il tuo sito ma è progettato per utenti più avanzati.

Se usi Bootstrap CDN, non c'è bisogno di scaricare alcun file. Se vuoi usare la versione compilata di Bootstrap, allora ti ricordo che devi scaricare i file CSS compilati e salvarli nella stessa cartella del tuo file index.html. Una volta completato questo passo è il momento di creare il tuo foglio CSS.

Per farlo, basta creare un altro file .css nella stessa cartella del tuo file index.html.

## CSS Override

Il riferimento al nuovo file CSS deve venire dopo quello di Bootstrap funzioni.

A causa della regola della specificità dei CSS. Se gli elementi HTML sono targettizzati da più selettori CSS della stessa specificità, l'ultimo sovrascriverà gli altri e solo le sue proprietà di stile saranno applicate. Collegando il tuo file CSS dopo il .css predefinito, gli dai un **livello più alto di specificità** .

Ad esempio a dimensione predefinita del carattere per la classe .btn è 16px.  
Puoi cambiarla a 22px usando la proprietà CSS font-size nel tuo file .css.

```
.btn { font-size: 22px }
```

Seguendo lo stesso processo, puoi fare tutte le modifiche necessarie per trasformare il tuo sito Bootstrap a tuo piacimento.

## CSS Override

Attualmente la versione 5 di Bootstrap è scritto in Sass, un linguaggio di estensione CSS che deve essere compilato perché i browser lo capiscano e lo utilizzino correttamente.

Avrai bisogno di un compilatore Sass per fare questo ma una volta fatto questo avrai il controllo completo sul tuo codice e su quello di Bootstrap.

Se scarichi il codice sorgente di Bootstrap usando npm e poi osservi la cartella noterai la seguente struttura:

```
tuo-progetto/  
├── scss  
│   └── custom.scss  
├── node_modules/  
│   └── bootstrap  
│       ├── js  
│       └── scss
```

## CSS Override

È importante notare che se non usi un gestore di pacchetti, dovrai impostare manualmente una struttura di file simile che tenga separati i file sorgente di Bootstrap dai tuoi.

Nel tuo file `custom.scss` puoi importare tutte le sorgenti Sass di Bootstrap includendo la seguente linea di codice:

```
@import "../node_modules/bootstrap/scss/bootstrap";
```

Una volta fatto questo puoi iniziare a fare le modifiche. Non c'è bisogno di aggiungere un riferimento al tuo file CSS nell'`index.html`.

Puoi ad esempio modificare il `background-color` ed il `color` nel seguente modo:

```
$body-bg: #000;
```

```
$body-color: #111;
```

## Navbar custom

Vista la possibilità di personalizzare i CSS, ora hai gli strumenti per personalizzare a tuo piacimento anche la tua barra di navigazione.

Per prima cosa ad esempio voglio cambiare la `.navbar-brand`, che identifica una società, un prodotto o il nome di un servizio. Per default la classe `.navbar-brand` ha un colore nero e una dimensione del font di 20px. Utilizzerò il metodo **CSS override**.

Cambio il colore con una tonalità di blu e ridimensiono a 25px usando rispettivamente le proprietà CSS `color` e `font-size` nel file CSS. Per farlo devo usare due selettori di classe: il nome della classe `.navbar` e `.navbar-brand`.

```
.navbar .navbar-brand {  
  color: #47A0FF;  
  font-size: 25px;  
}
```

## Navbar custom

L'uso di CSS override apporta modifiche limitate a singoli elementi o gruppi di elementi e non cambia l'aspetto dei pseudo-stati come ":hover", ":active" e "disabled".

Se vuoi fare modifiche più approfondite allora devi usare Sass.

Se voglio cambiare il colore del carattere degli elementi della navbar e usare un colore diverso per quando sono attivi e quando un utente ci passa sopra devo modificare le seguenti variabili Sass predefinite:

```
$navbar-light-color: rgba($black, .5) !default;  
$navbar-light-hover-color: rgba($black, .7) !default;  
$navbar-light-active-color: rgba($black, .9) !default;
```

Ora copio ed incollo le variabili nel mio file custom.scss, cambio i loro valori con del codice esadecimale e cancello il flag !default. Il flag !default significa che puoi sovrascrivere il valore predefinito della variabile nel tuo Sass senza modificare il codice sorgente di Bootstrap.





*shaping the skills of tomorrow*

challengenetwork.it

