

Bootstrap 4

Rev Digitale 2.3 del 30/03/2023

Introduzione a Bootstrap	2
Terminali, Containers e Grid System	3
Classi relative al testo / allineamento	6
Margin and Padding	8
Dimensions	8
Colors	9
Buttons	10
Modals	11
Forms e controlli.....	12
Icone	13
Immagini.....	13
Alert	14
Jumbotron	15
Page-header	15
Collapse	15
Card	15
Tabelle	16
Barre di navigazione e Menù	17
Dropdowns	18

Bootstrap 4

Bootstrap è un framework `css` e `Javascript` per la creazione di pagine web di tipo **responsive** (adattate) cioè in grado di adattarsi alle dimensioni dello schermo sul quale verranno visualizzate. Utilizza un approccio di tipo **mobile-first**, cioè orientato alla programmazione mobile. Sviluppato per creare **Twitter** e poi rilasciato come progetto open source.

Download

Dal sito ufficiale getbootstrap.com / **download** si può scaricare l'ultima versione. La versione **4.0** è stata rilasciata il 10 agosto 2017. Tra le altre cose, è stata eliminata la cartella **fonts** contenente le glyphicons (che erano distribuite come file di caratteri nei formati ttf, eot, svg e woff). Per l'utilizzo delle glyphicons si rimanda ora al sito **awesome**, aggregato al progetto bootstrap. Bootstrap utilizza al suo interno **jquery.js** e **popper.js** che è una libreria che facilita il posizionamento degli elementi html all'interno della pagina, soprattutto per quanto riguarda notification box e popup

Utilizzo delle librerie in locale

```
<link rel="stylesheet" href="bootstrap/css/bootstrap.css">
<script src="bootstrap/js/jquery.js"></script>
<script src="bootstrap/js/popper.js"></script>
<script src="bootstrap/js/bootstrap.js"></script>
```

Utilizzo delle librerie online

Si può utilizzare direttamente il **CDN** server di bootstrap (**bootstrapcdn.com**) :

Nella home page sono indicati i link ai server CDN per le varie versioni.
Per versioni differenti è sufficiente **modificare** il numero di versione.

```
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.12.1/css/all.min.css"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
```

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"></script>
```

E' anche disponibile una versione **bootstrap.bundle.min.js** che congloba al suo interno i riferimenti a Popper.js e Owl Carousel (necessario soltanto se si usano componenti di tipo Carousel per scorrere un set di immagini tramite le classiche frecce laterali), ma non a jQuery (abbandonato in bootstrap 5)

Il metatag viewport

Per poter utilizzare al meglio bootstrap ogni pagina html deve includere, prima di ogni altra cosa, i seguenti metatag:

```
<meta name="viewport"
      content="width=device-width, initial-scale=1">
```

Il meta-name **viewport** definisce le dimensioni della finestra in fase di apertura e il valore di zoom. I valori impostati fanno in modo che il browser adatti automaticamente la dimensione della pagina alle dimensioni dello schermo. `initial-scale=1` imposta il valore di zoom iniziale (100%)
`user-scalable=no` impedisce all'utente di effettuare zoom sulla pagina

I Terminali

Bootstrap suddivide i terminali in 5 categorie (4 in bootstrap v3) identificate tramite appositi **breakpoint** che indicano le dimensioni dello schermo:

- **xs** (**extra-small** for smaller phones) terminali con larghezza < **576** px
- **sm** (**small** for phones) terminali con larghezza < **768** px
- **md** (**medium** for tablets) terminali con larghezza < **992** px
- **lg** (**large** for desktops) terminali con larghezza < **1200** px
- **xl** (**extra-large** for larger desktops) terminali con larghezza > **1200** px

i Container

I **containers** sono delle classi che costituiscono la base del **Grid System** e che **consentono al loro interno di creare righe e colonne**.

- Il numero di righe può essere deciso dal programmatore
- Ogni riga si estende per tutta la larghezza del container
- Ogni riga viene automaticamente suddivisa in **12 celle** adattative di pari larghezza.

In realtà l'impostazione relativa al numero di colonne può essere modificata agendo all'interno del file di configurazione di bootstrap, ma si tratta di una esigenza abbastanza remota.

Il **container** viene sempre centrato orizzontalmente rispetto alla pagina o al genitore.

A seconda del terminale su cui la pagina viene visualizzata, la larghezza del **container** aumenta proporzionalmente rispetto alle dimensioni del terminale stesso.

La seguente tabella, scaricata dal sito ufficiale, riporta le dimensioni dei vari tipi di container:

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

La prima colonna riporta i **vari tipi di contenitori** disponibili in bootstrap, che sono essenzialmente 3:

container

Questa classe fa sì che il container assuma una larghezza fissa (che dipende dalla larghezza dello schermo) e venga visualizzato a centro video.

- In caso di terminale **extra-small** ogni cella occupa l'intera larghezza del terminale
Se il contenuto della cella eccede le dimensioni della cella viene nascosto
- In caso di terminale **small** ogni riga assume una larghezza pari a **540 px** e pertanto ogni cella assume una larghezza di $540/12 = 45$ px.
- In caso di terminale **medium** ogni riga assume una larghezza pari a **720 px** e pertanto ogni cella assume una larghezza di $720/12 = 60$ px
- In caso di terminale **large** ogni riga assume una larghezza pari a **960 px** e pertanto ogni cella assume una larghezza di $960/12 = 80$ px
- In caso di terminale **x-large** ogni riga assume una larghezza pari a **1140 px** e pertanto ogni cella assume una larghezza di $1140/12 = 95$ px. Anche nel caso di schermi molto grandi la dimensione del container non supererà mai i 1140 px

Man mano che si allarga la finestra del browser, nel momento in cui scattano le soglie indicate scatta anche l'allargamento di ogni singola riga e quindi anche degli elementi contenuti all'interno

container-fluid In questo caso il contenitore (e quindi ogni riga) occupa **sempre** l'intero spazio a sua disposizione (**100% del genitore**), indipendentemente dal tipo di terminale. La larghezza delle colonne si adatta in modo **fluid** (proporzionale) rispetto alle dimensioni del contenitore. Stringendo il contenitore si stringono fluidamente anche le colonne. Negli schermi grandi occupa però l'intera pagina con un effetto abbastanza scadente. **container-fluid** può andare bene per eventuali contenitori interni, ma **MAI** per il contenitore principale !

container-{breakpoint} Via di mezzo tra i due precedenti. Ogni colonna occupa il 100% della larghezza a sua disposizione fino a quando non si raggiunge il breakpoint indicato. Ad esempio nel caso del **container-lg** (che è il preferibile)

- la riga occuperà il 100% della larghezza a sua disposizione nel caso di terminali xs sm md (cioè come il **container-fluid**)
- la riga occuperà invece la larghezza prefissata nel caso di terminali lg e xl (cioè come il **container** normale)

Note sui container

- E' possibile utilizzare in sezioni diverse della stessa pagina contenitori diversi, ognuno con il proprio grid system. Ognuno può avere un suo tipo (fluid o non fluid).
- All'interno di un container è possibile annidare altri container di tipo differente (solo da bootstrap 4 in avanti) anche se si tratta di una situazione abbastanza improbabile

La larghezza delle celle

La classe **.row** aggiunge all'elemento una **width:100%** in modo che ogni riga occupi l'intera larghezza del terminale e dunque le righe vengano SEMPRE visualizzate una sotto l'altra.

Una row può contenere al suo interno come figli diretti soltanto celle.

I cosiddetti **modificatori di cella** (es **col-md-3**, **col-xl-6** etc) consentono di modificare la larghezza di una cella. **Ogni cella deve SEMPRE esporre almeno un modificatore di cella** altrimenti vscode segnala un errore di cella non valida. In assenza di modificatori le celle vengono comunque visualizzate e ad ogni cella verrà assegnato il valore **auto** che indica il numero minimo di colonne necessarie alla visualizzazione del contenuto presente nella cella.

I modificatori di cella consentono sostanzialmente di **accorp**are più celle in orizzontale (tipo **rowspan** dell'html) in modo da creare celle più larghe La somma deve comunque sempre fare 12. Esempio;

```
<div class="container">
  <h2>containter</h2>
  <div class='row'>
    <div class='col-md-3'>
      <button class="btn btn-secondary btn-block"> Secondary </button>
    </div>
    <div class='col-md-6'>
      <button class="btn btn-primary btn-block"> Primary </button>
    </div>
    <div class='col-md-3'>
      <button class="btn btn-secondary btn-block"> Secondary </button>
    </div>
  </div>
</div>
```

Nell'esempio, sui terminali **medium**

- la prima cella occupa 3 colonne
- la seconda 6 colonne
- la terza 3 colonne.

Nel momento in cui si imposta un valore su una singola classe :

- **il valore impostato verrà applicato su tutti i terminali di dimensione maggiore o uguale laddove non sia stata indicata una diversa classe di livello superiore.**
- **Nei terminali di dimensioni inferiore, in assenza di una apposita impostazione, le celle collassano, cioè vengono automaticamente visualizzate una sotto l'altra a piena larghezza**

Nell'esempio precedente le tre celle assumono la dimensione indicata nel caso di terminali **md**, **lg** e **xl**.

Nel caso invece di terminali inferiori (**xs** e **sm**) le celle vengono visualizzate una sotto l'altra.

Utilizzo parallelo di più modificatori

Le varie classi possono essere utilizzate **parallelamente** per ottimizzare il layout rispetto ai vari dispositivi, nel senso che ogni singolo oggetto potrà occupare un numero diverso di celle a seconda del contesto. Es:

```
<div class="row">
  <div class="col-sm-9 col-md-7" > </div>
  <div class="col-sm-3 col-md-5" > </div> </div>
```

Su una finestra di tipo **sm** il primo DIV occupa 9 colonne, il secondo 3 colonne.

Su una finestra di tipo **md** (o superiori) il primo DIV occupa 7 colonne, il secondo 5 colonne.

Siccome le classi relative a schermi maggiori 'coprono' le impostazioni delle classi relative a schermi minori, occorre scrivere le classi in ordine di terminale crescente.

Note sulle dimensioni delle celle

- Se si impostano ad esempio le classi **sm** ed **lg**, i terminali **sm** e **md** utilizzeranno la suddivisione "sm", mentre i terminali **lg** e **xl** utilizzeranno la suddivisione "lg".
Nei terminali **xs** le celle invece collassano e vengono visualizzate una sotto l'altra.
- Se si imposta la classe **md-12** l'oggetto sarà visualizzato a 12 colonne.
- Le classi **col-sm-0 col-md-0 col-lg-0 col-xl-0** non esistono.
- Le classi **col-xs-n** non esistono ed hanno come nome semplicemente **col-n**.
Se si impostano, le celle verranno comunque adattate tramite il solito schema a 12 colonne. I valori assegnati tramite questi modificatori valgono ovviamente anche per tutti i terminali superiori (a meno che non venga ridefinito un altro apposito modificatore). Nei terminali **xs** vengono rimosse tutte le spaziature fra una cella e l'altra in modo da ottimizzare l'occupazione dello spazio. I modificatori **xs** sono comunque abbastanza poco utilizzati perché sui terminali più piccoli di solito è bene avere le celle una sotto l'altra a piena larghezza.
- L'ultima cella, se vuota, può facoltativamente essere omessa. Bootstrap la aggiungerà in automatico estendendola per un numero di colonne tale per cui la somma complessiva arrivi a 12.

Utilizzo dei modificatori generici

Sono disponibili anche dei modificatori generici, cioè privi dell'indicazione del numero di colonne.

col-xs col-sm col-md col-lg col-xl

Se si applicano alle celle questi modificatori, lo spazio verrà diviso equamente fra le colonne impostate.

Ad esempio se si hanno 3 celle e si mette **lg-7** sulla prima cella e poi **lg lg** sulle altre due:

- la prima cella occuperà 7 colonne,
- le rimanenti 5 colonne verranno equamente suddivise fra le altre due celle. Cioè ognuna delle due celle occuperà 2.5 colonne, risultato NON ottenibile con i qualificatori tradizionali.

Offset fra le colonne

Per ogni colonna è possibile specificare un offset, inteso come numero di colonne vuote da aggiungere davanti alla colonna corrente che, di conseguenza, verrà spostata verso destra.

```
<div class="col-sm-6 col-sm-offset-3"> </div>
```

Questa classe è molto comoda nel caso in cui si voglia definire ad esempio una unica cella centrata di larghezza ad esempio 6. L'utilizzo dell'attributo `col-sm-offset-3` consente di evitare di scrivere la prima cella. Omettendo anche l'ultima cella la scrittura diventa estremamente compatta.

Annidamento delle colonne

E' possibile suddividere una colonna in sottocolonne annidate sempre con somma 12.

Le sottocolonne devono a loro volta essere sempre inserite all'interno di una row

```
<div class="row">
  <div class="col-sm-8">
    <div class="row">
      <div class="col-sm-6"> 4 colonne complessive (8/2) </div>
      <div class="col-sm-6"> 4 colonne complessive (8/2) </div>
    </div>
  </div>
  <div class="col-sm-4"> </div>
</div>
```

Classi relative al testo (Typography)

Allineamento Orizzontale del testo delle singole celle

Le classi

```
text-left
text-center
text-right
```

Consentono di allineare il testo interno ad una cella, e sono applicabili a qualunque tag di testo (div, paragrafi, span), così come ad esempio

```
text-lowercase
text-uppercase
font-weight-bold
```

Allineamento Verticale dell'intera riga

Le classi

```
align-items-start
align-items-center
align-items-end
```

applicabili soltanto ai tag `<div class="row">` consentono di allineare verticalmente il contenuto dell'intera row in alto, in basso oppure al centro

```
<div class="row align-items-start">
  <div class="col">
    One of two columns
  </div>
  <div class="col">
    One of two columns
  </div>
</div>
```

Allineamento Verticale delle singole celle

Le classi

align-self-start
align-self-center
align-self-end

sono invece **applicabili alle single celle**, e consentono di allineare verticalmente il contenuto della singola cella in alto, in basso oppure al centro

```
<div class="row">
  <div class="col align-self-start">
    One of three columns
  </div>
  <div class="col align-self-center">
    One of three columns
  </div>
  <div class="col align-self-end">
    One of three columns
  </div>
</div>
```

Allineamento Orizzontale delle singole celle

Quando non si specificano tutte le celle di una riga (cioè quando la somma non fa 12) è possibile definire la posizione delle celle inesistenti utilizzando, sulla riga, le seguenti classi:

```
<div class="row justify-content-start">
  <div class="col-4">
    One of two columns
  </div>
  <div class="col-4">
    One of two columns
  </div>
</div>
```

justify-content-start	lascia 4 colonne vuote a destra (default)
justify-content-end	lascia 4 colonne vuote a sinistra
justify-content-center	lascia 2 colonne vuote a sinistra e due a destra
justify-content-between	lascia 4 colonne vuote al centro
justify-content-around	lascia 1 colonna vuota a sx, 1 a dx, 2 in centro

La classe no-gutters

The gutters between columns in the predefined grid classes can be removed with `.no-gutters`. This removes the negative margins from `.row` and the horizontal padding from all immediate children columns.

```
<div class="row no-gutters">
  <div class="col-12 col-sm-6 col-md-8">.col-12 .col-sm-6 .col-md-8</div>
  <div class="col-6 col-md-4">.col-6 .col-md-4</div>
</div>
```

Margin and Padding

Per margini e padding si utilizzano apposite classi scritte nel seguente formato:

```
{property}{sides}-{breakpoint}-{size}
```

dove :

property può assumere i valori

m = margin
p = padding

sides può assumere i valori

t - for classes that set margin-top or padding-top
b - for classes that set margin-bottom or padding-bottom
s - (start) for classes that set margin-left or padding-left in LTR, margin-right or padding-right in RTL
e - (end) for classes that set margin-right or padding-right in LTR, margin-left or padding-left in RTL
x - for classes that set both *-left and *-right
y - for classes that set both *-top and *-bottom
blank - for classes that set a margin or padding on all 4 sides of the element

breakpoint può assumere i soliti valori sm md lg xl (e maggiori)

Se non è indicato significa xs (e maggiori)

size può assumere i valori

0 - for classes that eliminate the margin or padding by setting it to 0
1 - for classes that set the margin or padding to \$spacer * .25
2 - for classes that set the margin or padding to \$spacer * .5
3 - for classes that set the margin or padding to \$spacer
4 - for classes that set the margin or padding to \$spacer * 1.5
5 - for classes that set the margin or padding to \$spacer * 3
auto - for classes that set the margin to auto

dove \$spacer rappresenta il valore di default di margin / padding utilizzato dal browser in uso

Esempio: `class="py-sm-1.5"`

Dimensions

La seguenti classi impostano una larghezza in % rispetto al genitore. Idem per le altezze

w-25
w-50
w-75
w-100
h-25
h-50
h-75
h-100

Colors

Colori relativi al testo

Sono disponibili le seguenti classi per definire il colore di un testo:

<code>.text-muted</code>	// grigio chiaro	This text is muted.
<code>.text-primary</code>	// blu	This text is important.
<code>.text-success</code>	// verde	This text indicates success.
<code>.text-info</code>	// celeste	This text represents some information.
<code>.text-warning</code>	// giallo	This text represents a warning.
<code>.text-danger</code>	// rosso	This text represents danger.
<code>.text-secondary</code>	// grigio chiaro	Secondary text.
<code>.text-dark</code>	// grigio scuro	Dark grey text.
<code>.text-light</code>	// grigio chiarissimo	Body text.
<code>.text-white</code>	// bianco	.
<code>.text-body</code>	// nero	

Colori relativi allo sfondo

Sono disponibili le seguenti classi per definire il colore di uno sfondo:

<code>.bg-primary</code>	// blu (con testo bianco)
<code>.bg-success</code>	// verde
<code>.bg-info</code>	// celeste
<code>.bg-warning</code>	// giallo
<code>.bg-danger</code>	// rosso
<code>.bg-secondary</code>	// grigio chiaro
<code>.bg-dark</code>	// grigio scuro
<code>.bg-light</code>	// quasi bianco

This text is important.

This text indicates success.

This text represents some information.

This text represents a warning.

This text represents danger.

Secondary background color.

Dark grey background color.

Light grey background color.

Buttons

Oltre alla classe base **btn** che aggiunge semplicemente un padding sui 4 lati, in bootstrap 4.x.x sono disponibili le seguenti classi :

Basic Primary Secondary Success Info Warning Danger Dark Light Link

```
<button type="button" class="btn">Basic</button>
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-secondary">Secondary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-dark">Dark</button>
<button type="button" class="btn btn-light">Light</button>
<button type="button" class="btn btn-link">Link</button>
```

Queste classi possono essere applicate ai tag `<a>` `<input type=button>` `<button>`

In parallelo può essere applicata una delle seguenti classi che assegna una specifica dimensione al pulsante (la differenza comunque è di pochi px)

- **.btn-xs**
- **.btn-sm**
- **.btn-md** (default)
- **.btn-lg**
- **.btn-xl**

che consentono di impostare in modo più fine larghezza e altezza del pulsante (sempre che il pulsante non implementi le classi `form-control` o `btn-block` nel qual caso la larghezza coprirà tutta l'area disponibile)

La classe **disabled** visualizza il pulsante come disabilitato

La classe **active** visualizza il pulsante come premuto (con sfondo un po' più scuro).

La classe btn-block

Il pulsante per default utilizza una larghezza pari alla larghezza del suo contenuto

Se il pulsante è inserito all'interno di un grid system, **btn-block** fa sì che il pulsante si estenda per l'intera lunghezza delle celle all'interno delle quali è inserito

```
<button type="button" class="btn btn-primary btn-block"> Invia </button>
```

Attenzione però che se sul pulsante si applica la classe **input-group**, il pulsante, pur in presenza di **btn-block**, utilizzerà soltanto lo spazio necessario al proprio contenuto.

Button Outline

Bootstrap 4.x.x fornisce i seguenti buttons outline, dotati di un semplice bordino colorato :

Primary Secondary Success Info Warning Danger Dark Light

```
<button type="button" class="btn btn-outline-primary">Primary</button>
<button type="button" class="btn btn-outline-secondary">Secondary</button>
<button type="button" class="btn btn-outline-success">Success</button>
<button type="button" class="btn btn-outline-info">Info</button>
```

Modals

Esempio di finestra di login:

```
<div class="modal fade" id="loginModal" tabindex="-1">
  <div class="modal-dialog ">
    <div class="modal-content">

      <div class="modal-body">
        <form>
          <div class="form-group input-group pt-sm-4">
            <span class="input-group-text">
              <i class="fas fa-user"> </i>
            </span>
            <input type="text" class="form-control" id="user" placeholder="">
          </div>
          <div class="form-group input-group">
            <span class="input-group-text">
              <i class="fas fa-lock"> </i>
            </span>
            <input type="password" class="form-control" id="pass" placeholder="">
          </div>
        </form>
        <div class="text-right pt-sm-3">
          <button type="button" class="btn btn-secondary"
            data-dismiss="modal">Cancel</button>
          <button type="button" class="btn btn-primary" id="btnOk">Ok</button>
        </div>
      </div>
    </div>
  </div>
</div>
```

Per aprire la finestra:

```
<button type="button" class="btn btn-primary" data-toggle="modal"
  data-target="#loginModal">Login</button>
```

Per Gestire il click sul pulsante OK:

jQuery:

```
$("#loginModal").modal("hide")
```

pure javascript

```
modalClose(loginModal)
function modalClose(modal){
  modal.classList.remove('show');
  modal.setAttribute('style', 'display: none');
  const modalBackdrops = document.getElementsByClassName('modal-backdrop');
  document.body.removeChild(modalBackdrops[0]);
}
```

Per assegnare alla finestra un offset rispetto al default:

```
.modal {
  position: absolute;
  top: 350px;
  left: 10px;
  z-index: 10040;
  overflow: auto;
  overflow-y: auto; }
```

Per modificare la dimensione:

```
.modal-content{
  max-width:300px;
}
```

Forms e Controlli

Bootstrap fornisce tre possibili layout per le form:

- **vertical-form** (default)
- **inline-form** (class="form-inline")
- **horizontal-form** (class="form-horizontal")

Nel caso di **vertical form** tutti i controlli interni alla form vengono visualizzati uno sotto l'altro, indipendentemente da form-group (ogni controllo occupa l'intera riga)

Nel caso di **inline form** tutti i controlli interni alla form vengono visualizzati su una stessa riga, indipendentemente da form-group

Nel caso di **horizontal-form** tutti i controlli interni alla classe **form-group** vengono visualizzati sulla stessa riga, mentre form-group diversi vengono visualizzati su righe successive.

form-group

Tutti i gruppi di controlli interni ad una form dovrebbero implementare una classe **form-group** che funge da "contenitore" per i controlli. In realtà si limita ad aggiungere un margine inferiore di 15px in modo da distanziare leggermente i vari gruppi.

La classe **form-group** diventa significativa se abbinata alla classe **horizontal-form**.

*Questa classe non può essere applicata ad un tag che implementa la classe **row** la quale **deve** essere applicata ad un elemento dedicato.*

form-control

La classe **form-control** consente al controllo di estendersi per l'intera larghezza a sua disposizione. E' equivalente a **btn-block** che però vale solo per i pulsanti

Input-group

La classe **input-group** consente di estendere un Text Box aggiungendo una icona o un testo o un pulsante prima o dopo del TextBox medesimo.

L'elemento da aggiungere davanti / dietro implementa solitamente la classe **input-group-text**

```
<form>
  <div class="col-sm-4">
    <div class="form-group input-group">
      <label for="email" class="input-group-text">Username:</label>
      <input type="email" class="form-control" id="email">
    </div>
  </div>
</form>
```

Le icone

w3school / icons. Da bootstrap 4 le icone non fanno più parte del pacchetto base ma devono essere scaricate ed installate separatamente. In alternativa è possibile usare le icone di fontawesome.com/icons

A tal fine occorre scaricare la libreria awesome oppure impostare un link ad un CDN.

All'inizio del sito c'è un comodissimo campo di **search** che consente di ricercare le icone desiderate

Le icone users sono contenute nel gruppo **Users e People**

Le icone password sono contenute nel gruppo **Security**

Per inserire una icona si utilizza il tag **<i>** :

```
<i class="fas fa-user"></i>
```

La prima classe indica il tipo di icona:

fas = icona solida

far = icona regolare stilizzata

fal = icona light con i bordi più sottili

fad = icona duotone in due colori

Per centrare e spaziare adeguatamente l'icona occorre "avvolgerla in un tag **** con associata la classe **input-group-text** che ha sostituito la classe **input-group-addon** di bootstrap 3.x.x.

A differenza di **input-group-addon** questa nuova classe non consente però di ricolorare l'icona in caso di errore.

```
<div class="form-group input-group">
  <span class="input-group-text">
    <i class="fas fa-user"> </i>
  </span>
  <input type="text" class="form-control" id="usr" placeholder="Username">
</div>
```

	Email
	Password

La classe **"is-invalid"** consente di ricolorare di rosso il bordo di un Text Box interno ad una form in cui sia stato inserito un valore non valido (**Form Validation** al fondo della sezione Forms)

Immagini

Le immagini non hanno una classe predefinita necessaria per poter applicare le classi aggiuntive, che possono invece essere applicate direttamente al tag **IMG**.

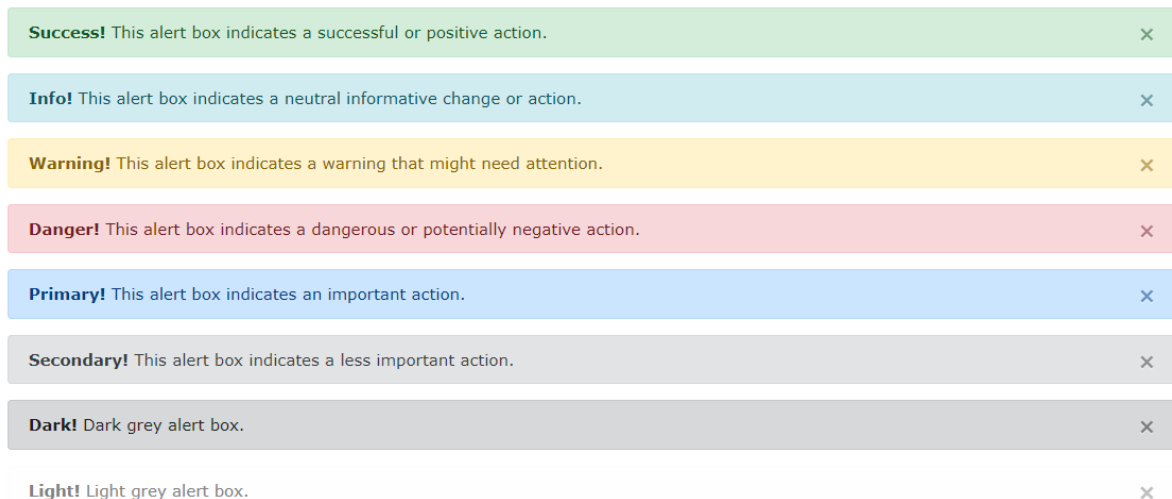
Le classi aggiuntive applicabili al tag **IMG** sono le seguenti :

.rounded	con spigoli arrotondati.
.rounded-circle	rende l'immagine ovale. Le dimensioni rimangono quelle indicate da width
.img-thumbnail	aggiunge un bordo con trattino nero e padding bianco e arrotonda un pochino gli spigoli. Le dimensioni sono quelle indicate da width
.float-right	rende l'immagine fluttuante a destra
.float-left	rende l'immagine fluttuante a sinistra
.img-fluid	visualizza l'immagine in tutta la sua grandezza, indipendentemente da width, adattandosi automaticamente alle dimensioni dello schermo o alle dimensioni del genitore. L'immagine si riduce automaticamente con la riduzione del genitore.

Per poter inserire le immagini nel contesto a righe e colonne di bootstrap occorre inserire l'immagine all'interno di un tag DIV in quanto non è consentito applicare al tag IMG le classi relative alle colonne (ad esempio la classe `class="col-sm-6"`)

La classe Alert

Applicata ad un contenitore, visualizza l'intero contenuto (testo e sfondo) con il colore impostato:



```
<div class="alert alert-success">
  <strong>Success!</strong> Indicates a successful or positive action.
</div>
```

La classe alert applica il color di sfondo anche al padding circostante al testo. Se il testo è costituito da un controllo diventa un po' pesante.

Pulsante di chiusura

E' possibile aggiungere alla alert un pulsane di chiusura:

```
<div class="alert alert-danger alert-dismissible" id="msg">
  <button type="button" class="close" data-dismiss="alert">&times;</button>
  <strong>Attenzione!</strong> username non valido.
</div>
```

La classe alert-dismissible aggiunge un extra-padding intorno al pulsante di chiusura.

La classe close sposta il pulsante sulla destra.

L'attributo data-dismiss è quello che forza la chiusura della finestra in corrispondenza del click sulla x.

Per avere maggiore controllo, invece di utilizzare data-dismiss si può gestire la chiusura manualmente da javascript:

```
_msg.children("button").on("click", function(){
  _msg.hide();
})
```

In tal caso `_msg.show()` consente di riaprire la finestra.

Jumbotron

Utilizzato per creare titoli con ampio risalto, sfondo grigio, spigoli arrotondati e font-size aumentato. Utilizzato per richiamare maggiormente l'attenzione del visitatore.

```
<div class="jumbotron">
  <h1>Bootstrap Tutorial</h1>
  <p>Bootstrap is the most popular framework for developing
      responsive, mobile-first projects on the web.</p>
</div>
```

Page-header

Consente di creare una intestazione con una leggera riga orizzontale sottostante più alcune spaziature aggiuntive intorno all'elemento. Simile al jumbotron ma meno evidente

```
<div class="page-header">
  <h1>Example Page Header</h1>
</div>
```

Collapse

Assegnando questa classe ad un qualunque oggetto della pagina, questo diventa 'collapsible' cioè **inizialmente nascosto** e visualizzabile mediante un click su un pulsante o altro elemento della pagina.

Elemento nascosto:

```
<div class="collapse" id="myID">
```

Elemento 'controllante':

```
<div data-toggle="collapse"
      data-target="#idElementoDaControllare"> </div>
```

Il primo attributo indica che l'elemento corrente può controllare un elemento 'collapsible' della pagina. Il secondo attributo indica l'elemento da controllare. Nella stessa pagina possono esistere più elementi 'collapsible' controllati ognuno mediante un apposito elemento controllante.

La classe collapse non può essere applicata ad un elemento che implementa la classe **form-group** ma deve essere applicata ad un apposito elemento interno.

Card

E' semplicemente un box bordato con un certo padding. Se non si specificano altre classi viene implementato il "basic card" a sfondo bianco. Altrimenti si può aggiungere una delle solite classi per la definizione dello sfondo, ad esempio **bg-primary**

Tables

La classe `.table` applicata ad un tag html `table` applica uno stile “base” della tabella in cui le righe hanno sfondo bianco e bordo orizzontale di separazione. Le colonne non hanno bordo. In assenza della classe `.table` le classi interne non funzionano correttamente.

Se si utilizza la classe `.table`, occorre utilizzare anche i tag **THEAD** e **TBODY**, altrimenti le classi bootstrap non funzionano correttamente.

Insieme a `.table`, può essere applicata alla tabella una delle seguenti classi:

<code>.table-striped</code>	le righe vengono visualizzate con sfondo alternato, una a sfondo bianco ed una a sfondo grigio chiaro.
<code>.table-hover</code>	viene aggiunto un effetto di roll over sulle righe. Al passaggio del mouse su una riga, lo sfondo assumerà una tonalità leggermente più scura
<code>.table-dark</code>	applica un colore di sfondo dark. Può essere utilizzata in abbinamento a <code>table-striped</code> per ottenere uno sfondo alternato grigio e dark. Oppure può essere utilizzata in abbinamento con <code>table-hover</code>
<code>.table-bordered</code>	viene aggiunto un bordo anche alle colonne.
<code>.table-borderless</code>	Vengono eliminati tutti i bordi anche sulle righe
<code>.table-sm</code>	rende la tabella più compatta dimezzando tutti i padding

Contextual classes

Senza impostare nessuna delle classi precedenti (con la sola `.table`) è possibile assegnare un colore specifico ad ogni singola cella. I colori utilizzabili sono i soliti colori relativi agli sfondi

```
<table class="table">
  <thead>
    <tr class="table-primary">
      <th>Firstname</th>
      <th>Lastname</th>
      <th>Email</th>
    </tr>
  </thead>
  <tbody>
    <tr class="table-secondary">
```

Le classi `thead-color` consentono di applicare un colore di sfondo all'intera `thead`

```
<thead class="thead-dark">
```

.table-responsive

Da applicare su un tag **DIV esterno** alla tabella, rende la tabella **responsive**, nel senso che compare automaticamente una scroll bar quando lo schermo assume una larghezza inferiore a 992px

```
<div class="table-responsive">
  <table class="table">
    ...
  </table>
</div>
```


Barre di Navigazione e Menù (bootstrap 3.x.x.)

Lo scopo è quello di eseguire link (tag a href) oppure procedure javascript.
A questo scopo esistono tre gruppi di controllo:

- Il **pager** che è costituito semplicemente da una successione orizzontale di pulsanti
- Le barre di navigazione classiche (**navbar**) strutturate come un menù orizzontale
- Le **dropdowns** che sono menù a discesa che consentono di associare ad ogni singola voce un'azione javascript oppure un link. Si tratta comunque sempre di menù e NON di ListBox

pager

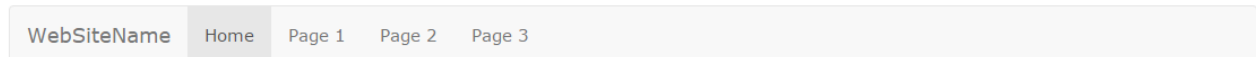
Crea una serie di pulsanti per la navigazione del sito
E' sufficiente applicare la classe "pager" ad un tag ul

```
<ul class="pager">  
  <li><a href="#">Previous</a></li>  
  <li><a href="#">Next</a></li>  
</ul>
```

Navbar

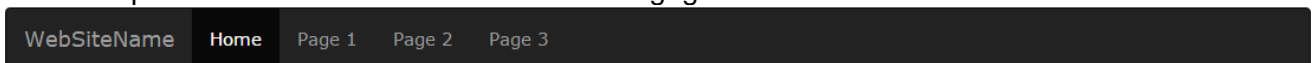
Scopo primo è quello di consentire la navigazione fra le pagine.
Esistono diversi tipi di navbar.

Standard NavBar



Inversione dei colori

Come la precedente ma con sfondo nero a testo grigio chiaro



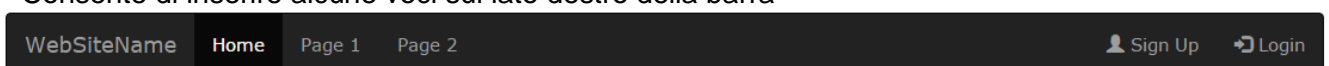
NavBar with DropDown

Le varie voci possono contenere un ulteriore drop down menù



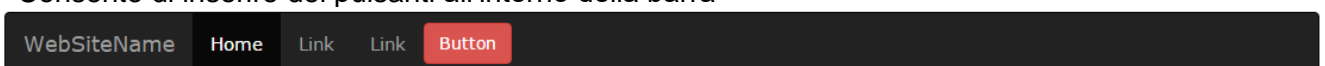
Right-Aligned NavBar

Consente di inserire alcune voci sul lato destro della barra



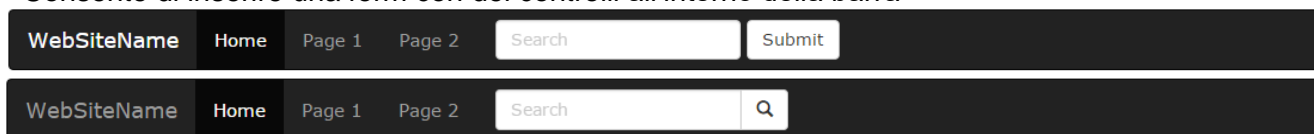
Buttons NavBar

Consente di inserire dei pulsanti all'interno della barra



Forms NavBar

Consente di inserire una form con dei controlli all'interno della barra



Forms NavBar

Oltre ai Link, può contenere anche semplice testo



Fixing the NavBar

Le classi **navbar-fixed-top** e **navbar-fixed-bottom** applicabili a qualsiasi delle navbar precedenti, fissano la barra in cima o in fondo alla pagina rendendola insensibile allo scroll

Collapsing the NavBar

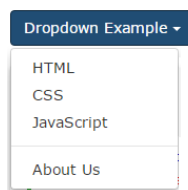
Da utilizzare quando si ha poco spazio a disposizione. In caso di bassi valori di width i vari link scompaiono ed al loro posto viene visualizzato un pulsante di apertura. In caso di click la barra viene aperta in orizzontale.



Dropdowns

Anche nel caso dei dropdowns esistono diversi tipi:

Basic Dropdown



Header

Consente di aggiungere una intestazione prima delle voci

Disable and Active Items

Consente di attivare / disattivare le singole voci

DropDownPosition

Consente di applicare un offset orizzontale e verticale al punto di apertura del sottomenù

DropUp

Consente di aprire il menù in alto sovrapponendolo al pulsante stesso

