

Bootstrap

Rev Digitale 1.2 del 20/01/2018

Introduzione a Bootstrap	2
I Containers	2
Bootstrap grid system	2
Classi relative al testo	4
Forms, controlli, glyphicon	5
Buttons	7
Immagini.....	8
Tabelle	8
Jumbotron	9
Page-header	9
Barre di navigazione e Menù	10
Dropdowns	11
Altre classi	12

Bootstrap

Bootstrap è un framework `css` e `Javascript` per la creazione di pagine web di tipo **responsive** (adattative) 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.

E' costituito da un insieme di classi spiegate molto bene su "**w3school bootstrap**", in cui le varie classi sono suddivise per gruppi.

Download

Dal sito [getbootstrap](#) scarica la versione **minified** che è più che sufficiente. Lo zip scaricato contiene 3 cartelle (versione 3.3.7 a novembre 2016) che dovranno essere posizionate all'interno della cartella di lavoro.

- **js** contenente il file `bootstrap.js`
- **css** contenente i vari `css` di bootstrap
- **fonts** contenente il font `glyphicons-halflings`

La libreria bootstrap utilizza al suo interno anche **jQuery**, che quindi dovrà essere linkato al progetto prima dei riferimenti a bootstrap.

Il metatag viewport

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

```
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
```

Il meta-name **viewport** fa in modo che il browser adatti automaticamente la dimensione della pagina ad un valore pari alle dimensioni dello schermo, in modo da attivare la media query corrispondente.

`initial-scale=1` imposta il valore di zoom iniziale (100%)

`user-scalable=no` impedisce all'utente di effettuare zoom sulla pagina

I Containers

Per poter utilizzare le classi bootstrap, **l'intera pagina** deve essere avvolta all'interno di un tag che implementi una classe **container**.

Le classi container soltanto due, **entrambe centrate** all'interno della pagina:

container	La larghezza delle colonne è fissa e cambia soltanto quando la larghezza della finestra oltrepassa una delle 4 soglie del Grid System (lo spazio eccedente le dimensioni del container sarà libero ai due lati del container stesso).
container-fluid	La larghezza delle colonne varia in modo fluido (proporzionale) rispetto alla larghezza della finestra. In questo modo il container occupa sempre l'intera larghezza della finestra.

I contenitori non sono annidabili, cioè non posso utilizzare un contenitore all'interno di un altro.

Bootstrap Grid System

Ciò che rende bootstrap facilmente responsive è il suo Grid System. L'area interna al Container viene automaticamente suddivisa in **12 colonne** adattative di pari larghezza. Se le dimensioni della finestra si riducono, la larghezza delle colonne si riduce proporzionalmente, portando automaticamente alcune celle a capo in caso di ulteriore riduzione. Più colonne possono essere accorpate per creare colonne più larghe per spaziare correttamente gli elementi sulla pagina. La somma deve comunque sempre fare 12.

Il grid system utilizza 4 classi:

- **xs** (for phones) [width < 768px]
- **sm** (for tablets) [width > 768px]
- **md** (for desktops) [width > 992px]
- **lg** (for larger desktops) [width > 1200px]

La pagina deve sempre essere strutturata per righe. Le colonne **devono** sempre essere inserite all'interno di una row. **Una row può contenere al suo interno come figli diretti soltanto colonne.** Es:

```
<div class="row">
  <div class="col-md-3"> 3 colonne di spaziatura</div>
  <div class="col-md-6"> 6 colonne di testo</div>
  <div class="col-md-3"> 3 colonne di spaziatura </div>
</div>
```

La dimensione delle colonne per le quattro classi è la seguente:

- **xs** auto
- **sm** 62 px
- **md** 81 px
- **lg** 97 px

Per cui man mano che si allarga la finestra del browser, nel momento in cui scattano le soglie indicate scatta anche l'allargamento di ogni singola colonna e quindi anche degli elementi contenuti all'interno. Se all'interno di una riga si inserisce un unico elemento DIV senza specificare il numero di colonne, esso occuperà automaticamente tutte e 12 le colonne.

L'altezza delle righe dipende invece dal contenuto dei vari blocchi che compongono la riga

Utilizzo parallelo di più classi

Le varie classi possono essere utilizzate **parallelamente** per ottimizzare il layout rispetto ai vari dispositivi, nel senso che la singola cella 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** la prima cella occupa 9 colonne, la seconda 3 colonne.
Su una finestra di tipo **md** la prima cella occupa 7 colonne, la seconda 5 colonne.

Utilizzo di una sola classe

Se si intende utilizzare lo stesso numero di colonne per tutti i vari device, è sufficiente definire una sola classe. Nel caso in cui la finestra abbia dimensioni diverse rispetto alla classe utilizzata, automaticamente la larghezza delle colonne scala al valore opportuno. In questo caso è raccomandato l'utilizzo della classe **sm** in quanto la scalatura verso l'alto è più facile e precisa.

Se non si esplicita uno specifico valore per la classe **xs**, nel momento in cui la larghezza dello schermo scende sotto 768px i vari blocchi costruiti all'interno della riga vengono visualizzati **uno sotto l'altro**, ciascuno per una altezza pari al proprio contenuto. Se invece si specifica la classe **xs**, i vari blocchi che costituiscono la riga continuano ad essere ridotti proporzionalmente, ma l'effetto è peggiore rispetto al caso precedente in quanto ad un certo punto la visibilità diventa estremamente ridotta.

Spaziatura fra le colonne

Tra una colonna e l'altra viene creata automaticamente una spaziatura (ottenuta attraverso il padding delle colonne) detta **gutter** (grondaia, rilegatura). Per tutte e quattro le classi il valore di default di questo **gutter** è 30 px, suddiviso in 15 pixel su entrambi i lati della colonna.

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-8 col-sm-offset-4"> </div>
```

Colonne annidate

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 </div>
      <div class="col-sm-6"> 4 colonne </div>
    </div>
  </div>
  <div class="col-sm-4"> </div>
</div>
```

Typography (classi relative al testo)

Colori per il testo

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

```
.text-primary    // bianco
.text-success    // verde
.text-danger     // rosso
.text-warning    // giallo
.text-info       // celeste
.text-muted      // grigio
```

Colori per il testo

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

```
.bg-primary    // bianco
.bg-success    // verde
.bg-danger     // rosso
.bg-warning    // giallo
.bg-info       // celeste
```

Esiste infine un nutrito elenco di classi applicabili a sezioni di testo (div, paragrafi, span):

```
text-left
text-center
text-right
text-lowercase
text-uppercase
small (dimensione ridotta all'85% della dimensione del contenitore)
```

Forms e Controlli

Bootstrap fornisce tre possibili layout per le form:

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

Tutti i gruppi di controlli interni alla form dovrebbero implementare una classe **form-group** che si limita ad aggiungere un margine inferiore di 15px in modo da distanziare leggermente i vari gruppi. Questa classe non può essere applicata ad un tag che implementa la classe row, che **deve** essere invece essere applicata ad un elemento posizionato su una riga dedicata.

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

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 **horizontal-form** tutti i controlli interni ad un form-group vengono visualizzati sulla stessa riga, mentre form-group diversi vengono visualizzati su righe successive

Tutti i controlli interni alla form devono invece necessariamente implementare la classe **form-control** che consente al controllo di estendersi per l'intera larghezza a sua disposizione.

La classe **input-group** consente di raggruppare più elementi sulla stessa riga, indipendentemente dal tipo di layout impostato

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

```
<div class="form-group col-sm-4">
  <div class="input-group">
    <label for="pwd">Password:</label>
    <input type="password" class="form-control" id="pwd">
  </div>
</div>
```

Le pagine **input** e **input2** contengono tutte le classi applicabili ai singoli controlli di una form.

Le glyphicon e le classi input-group e input-group-addon

Applicando al contenitore la classe **input-group**, diventa possibile raggruppare una icona ed un textBox sulla stessa riga.

La classe **input-group-addon** consente di applicare ad un controllo interno a **input-group** una *icona* oppure *un testo* oppure *un pulsante*.

	Email
	Password
Text	Additional Info

```
<form>
  <div class="input-group">
    <span class="input-group-addon">
      <i class="glyphicon glyphicon-user"> </i>
    </span>
    <input type="text" class="form-control" placeholder="Email">
  </div>

  <div class="input-group">
    <span class="input-group-addon">
      <i class="glyphicon glyphicon-lock"> </i>
    </span>
    <input type="password" class="form-control" placeholder="Password">
  </div>

  <div class="input-group">
    <span class="input-group-addon">Text</span>
    <input type="text" class="form-control" placeholder="Additional Info">
  </div>
</form>
```

Il tag **<i>** viene utilizzato per inserire le icone. La classe che segue rappresenta sostanzialmente il nome della glyphicon da utilizzare.

Per vedere l'elenco completo delle glyphicon si rimanda direttamente al sito di bootstrap (digitare su google "bootstrap glyphicon").

Awesome icons

Consente di scaricare icone di ogni genere che devono essere utilizzate sempre all'interno di un tag `<i>`

Il menù **Examples** contiene moltissimi esempi.

Il menù **Icons** contiene l'elenco delle icone disponibili.

Di seguito è riportato il codice necessario per l'associazione di una icona ad un campo password.

Le icone sono perfettamente compatibili con la struttura di bootstrap.

```
<div class="input-group">
  <span class="input-group-addon"> <i class="fa fa-key fa-fw"> </i> </span>
  <input class="form-control" type="password" placeholder="Password">
</div>
```

Buttons

Sono disponibili le seguenti sette classi:



- `btn-default`
- `btn-primary`
- `btn-success`
- `btn-info`
- `btn-warning`
- `btn-danger`
- `btn-link`

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

In parallelo può essere applicata una delle seguenti classi che consentono di aumentare larghezza e altezza del pulsante, sempre che il pulsante non implementi le classi `form-control` o `btn-block`, nel qual caso la sua larghezza coprirà tutta l'area disponibile. In questo caso le classi indicate di seguito aumentano soltanto l'altezza del pulsante.

- `.btn-lg`
- `.btn-md` (default)
- `.btn-sm`
- `.btn-xs`

Esempio:

```
<div class="form-group col-sm-4">
  <button type="button" class="btn btn-default btn-sm "> Invia </button>
</div>
```

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

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

La classe btn-block

Esponde il pulsante sull'intero spazio a sua disposizione (effetto ottenibile anche mediante la classe **form-control**). 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.

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 :

.img-rounded	si limita ad arrotondare gli angoli. Le dimensioni sono quelle indicate da width
.img-circle	rende l'immagine ovale. Le dimensioni sono 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
.img-responsive	visualizza l'immagine in tutta la sua grandezza, indipendentemente da width, adattandosi automaticamente alle dimensioni dello schermo o alle dimensioni dell'elemento che la contiene (parent).

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

Tabelle

La classe **.table** applicata ad un tag **table** definisce lo stile base della tabella.

Deve sempre essere richiamata, altrimenti le classi interne non funzionano correttamente.

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

.table-striped	le righe vengono visualizzate con sfondo alternato, una a sfondo chiaro ed una a sfondo scuro.
.table-bordered	viene aggiunto un bordo ad ogni cella ed alla tabella esterna.
.table-hover	viene aggiunto un effetto di roll over sulle righe. Al passaggio del mouse su una riga, lo sfondo assumerà una tonalità leggermente più scura
.table-condensed	rende la tabella più compatta dimezzando tutti i padding

.table-responsive

Da applicare su un tag div esterno alla tabella. Rende la tabella maggiormente responsiva, nel senso che compare automaticamente una scroll bar quando lo schermo assume una larghezza inferiore a 768

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


Stili contestuali applicabili alle righe e/o alle celle

Cambiano il colore di sfondo della singola riga o cella

```
.primary      // bianco
.success      // verde
.danger       // rosso
.warning      // giallo
.info         // blu
```

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.

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

Barre di Navigazione e Menù

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

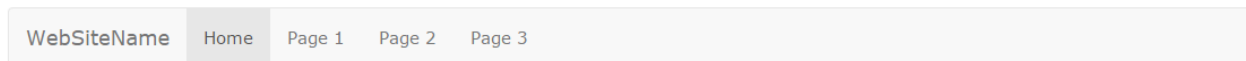
Crea una serie di pulsanti per la navigazione del sito
E' sufficiente applicare la classe "pager" ad un tag ul

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

navbar

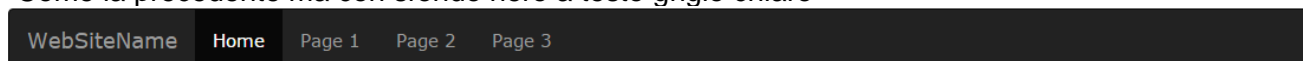
Scopo primo è quello di consentire la navigazione fra le pagine.
Esistono diversi tipi di navbar.

Standard NavBar



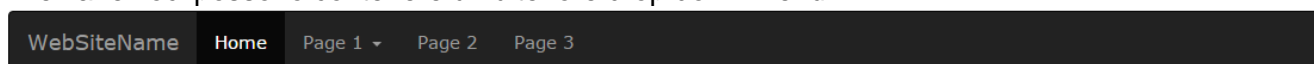
Inversione dei colori

Come la precedente ma con sfondo nero a testo grigio chiaro



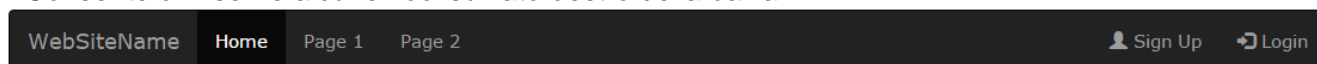
NavBar with DropDown

Le varie voci possono contenere un ulteriore drop down menù



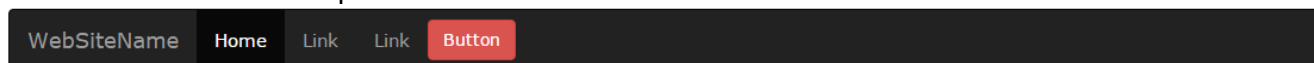
Right-Aligned NavBar

Consente di inserire alcune voci sul lato destro della barra



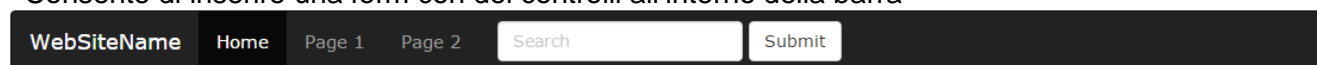
Buttons NavBar

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

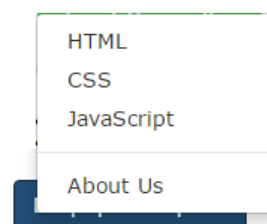
Consente di attivare / disattivare le singole voci

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



Cenni su altre classi

La classe 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.

Per visualizzare / nascondere un elemento 'collapsible' occorre aggiungere sull'elemento 'controllante' i seguenti attributi:

```
data-toggle="collapse"  
data-target="#idElementoDaControllare"
```

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 eventualmente deve essere applicata ad un apposito elemento interno.

La classe Alert

Da un punto di vista grafico visualizza un messaggio a sfondo colorato in modo simile a quello delle classi page-header e jumbotron. Rispetto allo jumbotron

- Non consente l'effetto collapsible su altri oggetti
- Presenta la possibilità di utilizzare diversi colori di sfondo (jumbotron solo grigio) applicando una delle seguenti classi:

```
.alert-success // verde  
.alert-info    // celeste  
.alert-warning // giallo  
.alert-danger  // rosso
```
- La classe page-header non applica colori allo sfondo