#### Bootstrap Versione 3.3.7

# BOOTSTRAP

# **II framework**

#### Autore

# Prof. Rio Chierego riochierego@libero.it

Ultimo aggiornamento 21 maggio 2020

#### Indice

| Siti utili:                          | slide 3                 |
|--------------------------------------|-------------------------|
| Introduzione:                        | slide 4                 |
| Una serie di domande su Bootstrap:   | da slide 5 alla slide 8 |
| Come usare il framework:             | da slide 9 a slide 10   |
| Novità versione 3 e suo contenuto:   | da slide 11 a slide 15  |
| Scheletro di una pagina base:        | da slide 16 a slide 20  |
| L'elemento GRID:                     | da slide 21 a slide 46  |
| La NAVBAR:                           | da slide 47 a slide 53  |
| Gestione delle immagini:             | da slide 54 a slide 56  |
| Formattazione del testo:             | da slide 57 a slide 59  |
| Le TABELLE:                          | da slide 59 a slide 66  |
| Layout di un FORM:                   | da slide 67 a slide 74  |
| L'elemento ACCORDION:                | da slide 75 a slide 76  |
| L'elemento CAROUSEL:                 | da slide 77 a slide 83  |
| Page header e JUMBOTRON:             | da slide 85 a slide 85  |
| Thumbnail all'interno della griglia: | da slide 86 a slide 88  |

#### Siti Utili

https://getbootstrap.com/docs/4.4/getting-started/introduction/

<u>https://getbootstrap.com/docs/4.4/getting-started/download/</u>

<u>http://www.html.it/guide/guida-bootstrap/</u>

https://www.mrwebmaster.it/web-design/guide/guida-bootstrap/

https://www.w3schools.com/bootstrap/default.asp

#### Introduzione

Premessa: queste slides si basano sulla versione 3.3.7 di Bootstrap del 2016 (l'ultima al momento è invece la 4.1.3).

Se si desidera utilizzare una versione precedente del **framework** si consiglia di effettuare un aggiornamento, se questo non è possibile considerate che diverse cose sono cambiate già con l'avvento della versione 3 quindi alcune parti potrebbero non essere applicabili al nostro progetto.

Con il termine **framework**, che può essere tradotto in italiano come intelaiatura o struttura, in informatica e specificatamente nello sviluppo software, si intende un'architettura logica di supporto (spesso un'implementazione logica di un particolare schema di progettazione concettuale) su cui un software può essere progettato e realizzato, spesso facilitandone lo sviluppo da parte del programmatore.

#### Prima domanda: Cosa è Bootstrap?

Bootstrap è un insieme di elementi grafici, stilistici, di impaginazione e Javascript pronti all'uso, nati in seno a Twitter ad opera degli sviluppatori Mark Otto e Jacob Thornton.

Oggi **Bootstrap** è un progetto indipendente ed è stato messo a disposizione degli sviluppatori di tutto il mondo che sono liberi di utilizzare questo **framework** come base per i propri progetti web. Ad oggi **Bootstrap** forte di una platea di utilizzatori in rapida espansione, rappresenta una delle soluzioni più utilizzate per la progettazione di **template** per il web, soprattutto in ottica **responsive**.

#### Seconda domanda: Cosa ci possiamo aspettare da Bootstrap?

La risposta è semplice: questo **framework** ci offre i mattoni con cui costruire pagine web **HTML5**, completamente **responsive**, **coerenti** e **funzionali**. L'utilità di Bootstrap è immediatamente evidente, soprattutto nella situazione attuale in cui le pagine web possono essere fruite su una miriade di dispositivi con caratteristiche diverse. Sarà Bootstrap ad occuparsi di mettervi a disposizione elementi di stile che permettono alla pagina di adattarsi al dispositivo utilizzando, al contempo, elementi di interfaccia comuni ai siti moderni, quelli cioè che l'utente si aspetta e di cui conosce comportamento e significato.

#### Terza domanda: Cosa è contiene realmente Bootstrap?

Possiamo suddividere i suoi elementi in quattro macro-aree:

#### 1) Scaffolding (o impalcatura)

Questa area contiene tutti quegli elementi CSS che permettono di definire la struttura della pagina, ossia il suo layout.

La parte costitutiva di questa pagina è il **Grid system** ossia una griglia, fissa o fluida, con una larghezza base di 960px nella quale possono essere definite righe e colonne in cui poi incasellare i contenuti.

Di base, la griglia è costituita da 12 colonne e ogni elemento del layout può espandersi su una o più colonne, semplicemente applicando una classe di stile che in qualche modo definisce la sua "estensione".

#### 2) CSS base

Questa area contiene degli stili predefiniti per diversi elementi della pagina, come i titoli (H1, H2,...), le tabelle, i pulsanti, gli elementi dei form, le immagini...

N.B. Con queste regole di stile realizzare pulsanti di varie dimensioni, con i bordi smussati, con un effetto over diventa davvero semplice e immediato. Lo stesso vale per la creazione di una tabella con righe a colori alterni.

#### Terza domanda: Cosa è contiene realmente Bootstrap? (continuazione)

#### 3) Componenti

Questa area contiene elementi più complessi di pulsanti o tabelle, ma ormai molto comuni nei siti web quali gruppi di pulsanti, barre di navigazione, menu a discesa, etc. etc.

Fra i componenti è compreso anche un set di icone, o meglio di *glifi* (dato che non si tratta di immagini ma di caratteri) di uso comune, messe a disposizione da <u>Glyphicons</u>.

Le icone/glifi **Glyphicons** di norma sono a pagamento, mentre il test distribuito con Bootstrap è gratuito. Le icone si utilizzano tramite apposite classi.

#### 4) Javascript

Quest'ultima area contiene **diversi plug-in jQuery** per realizzare effetti molto comuni come **transizioni, finestre modali, popup, carousel, accordion, tab**.

Anche questi plug-in sono semplici da usare e vi permettono di realizzare tantissime soluzioni interessanti.

**N.B. JQuery** è un libreria **javascript** che permette ai **Designer/Developer** di sviluppare pagine web con funzionalità interattive in maniera semplice.

La **libreria** è estremamente popolare, viene utilizzata da molti siti, piattaforme (es. WordPress) e framework HTML/CSS (es. Bootstrap).

#### Quarta domanda: Quando usare Bootstrap?

In realtà possiamo dire che conviene utilizzare **Bootstrap** quando si desidera creare un nuovo sito web senza dover scrivere tutto da zero.

Bootstrap, come ogni altro framework, consente di sviluppare un progetto in modo più rapido, tuttavia non può e non deve essere considerato la soluzione definitiva ad ogni problema di web-design.

Così come non è sempre corretto fare ricorso ad un framework Javascript come jQuery, lo stesso vale per Bootstrap: l'opportunità o meno di un suo utilizzo deve essere valutata caso per caso in base alle specifiche esigenze di sviluppo.

Può essere considerato **solo un tool** per gestire al meglio la **fase di avvio di un progetto**, un modo per poter contare su una serie di componenti riusabili e personalizzabili, comunque adattabili in termini stilistici ed estetici alle richieste del progetto e alla creatività di chi concepisce il sito.

### Bootstrap: Come usare il framework

#### N.B. In queste slides non troverete una trattazione sistematica dei singoli componenti

Per questo scopo si rimanda ad un'apposita <u>reference</u> (ispirata alla <u>documentazione ufficiale</u> e più sintetica), anche a beneficio di chi si scontra con l'ostacolo della lingua inglese.

Le slides vogliono invece essere un avvio all'uso del **framework**, un modo per definire il migliore approccio all'utilizzo. Ci concentreremo pertanto su:

- il reperimento del framework e la scelta delle modalità di download;
- la strutturazione di un template HTML di partenza;
- una panoramica generale dei componenti;
- l'impalcatura fondamentale di ogni progetto Bootstrap, ovvero la griglia e le funzionalità responsive del layout;
- le pratiche utili per la personalizzazione del framework.

# Quest'ultimo è un aspetto cruciale in quanto una delle critiche maggiori rivolte a Bootstrap e agli altri framework della stessa categoria, è che i siti con essi realizzati sembrano tutti uguali.

Il nostro scopo è dimostrare come con le opportune strategie e potendo affidarsi a un minimo di creatività, si possa variare l'estetica per realizzare creazioni di impatto e personali anche usando questo framework.

### Bootstrap: Come usare il framework

#### Prima modalità: Inserire direttamente i file Bootstrap nel proprio progetto (versione compilata)

Procurarsi il framework Bootstrap è molto semplice. Basta visitare il sito <u>getbootstrap.com</u> e fare clic sul pulsante download che trovate sulla prima pagina del sito.

Dal download otterrete un pacchetto .zip **bootstrap-3.3.7.zip** (attualmente è in distribuzione la versione 4.1.3 beta)

Il pacchetto, una volta scompattato, conterrà la cartella **dist** che a sua volta contiene quello che ci occorre. Vediamo di seguito uno screenshot dell'albero della cartella una volta scompattata:

Seconda modalità: Utilizzare una Content Delivery Network (CDN) per includere Bootstrap

Se non si vuole caricare sul proprio server i file di Bootstrap, c'è anche la possibilità di includerli nelle proprie pagine richiamandoli da una CDN, ossia da un server che mette a disposizione questi file per il pubblico.

N.B. La scelta di utilizzare una CDN o caricare fisicamente i file sul proprio server deve essere fatta solo in base a criteri di performance: si ricordi, infatti, che utilizzare file esterni obbligherà il browser dei nostri utenti a fare ulteriori interrogazioni DNS e pertanto, presumibilmente, potrebbero aversi dei leggeri rallentamenti nel caricamento della pagina.

## Bootstrap: principali novità versione 3

#### Responsive design e griglia

La principale novità della versione 3 è senza dubbio rappresentata dal fatto che ora il framework è concepito nativamente come uno **strumento adatto alla realizzazione di siti responsivi.** 

Se prima le funzionalità responsive erano opzionali, ora si è abbracciato l'approccio mobile first.

Una pagina basata su Bootstrap è di default adattabile (responsive) ad una gamma ampia di device, dagli smartphone a crescere fino ai desktop ad altissima risoluzione.

In questo scenario, il componente che ha subito le modifiche più radicali è la griglia.

In Bootstrap versione 3 il layout è di default fluido, con larghezze espresse in percentuale, e pertanto maggiormente adattabile a differenti risoluzioni di schermo. Una serie di classi speciali abbinate a specifici breakpoint e di funzionalità avanzate rendono possibile la realizzazione di layout complessi e robusti in ogni scenario dettato dal dispositivo in uso.

Per chi volesse disabilitare le funzionalità **responsive** e operare con una **griglia fissa** sono disponibili sul sito ufficiale le istruzioni per implementare tale soluzione, anche se tale eventualità appare alquanto illogica.

### Bootstrap: principali novità versione 3

#### Supporto browser

Per quanto concerne il supporto dei vari browser va rilevato che con questa release Bootstrap abbandona al loro destino Internet Explorer 7 e Firefox 3.6. Ufficialmente il framework supporta le **ultime versioni di Chrome, Safari, Firefox, Opera e Internet Explorer.** 

Per **Internet Explorer 8** il supporto pieno è garantito solo adoperando lo script <u>Respond.js</u> in grado di aggiungere a quel browser la capacità di gestire le media query.

Altre funzionalità minori del framework non sono supportate su **Internet Explorer 9**.

## Bootstrap: Cosa contiene il framework

Una volta scompattato il pacchetto zip ci ritroviamo con questa struttura di cartelle e file (versione compilata):



## Bootstrap: Cosa contiene il framework

Una volta scompattato il pacchetto zip ci ritroviamo con questa struttura di cartelle e file (versione compilata):

| Cartella | Cosa contiene   |  |  |
|----------|---|--|--|
| CSS      | ci sono, in versione di sviluppo e compressa, i fogli di stile bootstrap.css e<br>bootstrap-theme.css. Il primo rappresenta il core del framework, il CSS<br>con tutti gli stili dei diversi componenti nella loro configurazione di base. Il<br>secondo foglio di stile comprende una serie di regole che ridefiniscono<br>alcuni di questi stili in modo da costituire una sorta di tema o skin rispetto<br>al CSS principale (gli <b>stili</b> riprendono quelli della vecchia versione di<br>Bootstrap) |  |  |
| js       | contiene la versione di sviluppo e quella minificata del file bootstrap.js. In<br>un unico file abbiamo dunque il codice di tutti i plugin e componenti<br>Javascript   |  |  |
| fonts    | l'icon font Glyphicons nei suoi diversi formati. Il font è già incorporato con tutti i riferimenti giusti all'interno del CSS principale  |  |  |

### Bootstrap: utilizzo del framework

Perché usare la versione compilata di Bootstrap?

Il download **della versione compilata** è da prediligere quando si voglia valutare il **framework** nel suo complesso, per la realizzazione di prototipi rapidi del sito, ma anche, in fase di sviluppo e produzione, in tutti quei casi in cui si pensa di sfruttare in un progetto l'intera gamma di componenti.

Chiaramente, è la via da scegliere anche quando si intenda lavorare direttamente con i **CSS** invece che con il **linguaggio LESS** <u>che è la base del codice sorgente di **Bootstrap**.</u>

16

Quando si tratta di mettere mano al codice, usare un **framework** come **Bootstrap** comporta essenzialmente la definizione di specifici componenti nel codice HTML attraverso la corretta strutturazione del markup e l'associazione agli elementi HTML di classi e stili CSS.

Prima, però, è opportuno crearsi una sorta di scheletro, <u>una</u> <u>struttura HTML di partenza per tutte la pagine del sito</u>.

Potrà chiaramente variare in base alle richieste di questo o quel progetto, ma una **base minima e condivisa** si può trovare.

Nella documentazione di Bootstrap viene suggerito di utilizzare questo snippet di codice HTML come template di base:

<!DOCTYPE html> <HTML> <HEAD>

Fa visualizzare correttamente i caratteri speciali quali lettere accentate, simboli e valute. <meta charset = "utf-8" > è per HTML5 nella versione "estesa" utilizzata è per HTML4

```
<TITLE><Scheletro pagina base</TITLE>
```

```
<!- META TAG viewport->
```

```
<meta name = "viewport" content="width=device-width, initial-scale=1.0" /> <meta http-equiv = "Content-Type" content = "text/html; charset = utf-8" /> <-- 
<!-- Bootstrap -->
```

<link href = "css/bootstrap.min.css" rel="stylesheet" media="screen" />
</HEAD>

<BODY>

```
<!-- jQuery (necessario per poter utilizzare i plugin javascript di Bootstrap) -->
<script src = "http://code.jquery.com/jquery-2.2.4.min.js"></script>
<!-- Include tutti I plugin compilati o solo quelli necessari-->
<script src = "js/bootstrap.min.js"></script>
</BODY>
```

</HTML>

Per prima cosa vorrei soffermare la vostra attenzione sulla presenza dell'istruzione META "viewport":

#### <meta name = "viewport" content="width=device-width, initial-scale=1.0" >

Questa linea di codice è necessaria per creare **layout responsive** (ovvero che si adattano alla dimensione del dispositivo in cui la pagina è visualizzata).

Come abbiamo detto, il poter creare layout perfettamente **responsivi**, infatti, è una delle peculiarità di questo framework.

Sempre nell HEAD del documento abbiamo incluso il file **CSS** di base del nostro framework.

#### k href = "css/bootstrap.min.css" rel="stylesheet" media="screen" >

N.B. Non abbiamo caricato, invece, il **tema** di bootstrap (**bootstrap-theme.min.css**), che può facilmente essere aggiunto mediante il richiamo all'apposito foglio di stile, in quanto abbiamo voluto mostrare una "installazione" minimale del framework che precludesse dal suo tema.

#### N.B. Al termine del documento abbiamo effettuato delle inclusioni di file Javascript

Innanzitutto includiamo la libreria **jQuery** che è una libreria **javascript** che permette ai **Designer/Developer** di sviluppare pagine web con funzionalità interattive in maniera semplice e che NON E' COMPRESA nel pacchetto scaricato, ma va scaricata direttamente dal suo sito oppure richiamata da un deposito on-line (CDN).

È molto importante che **jQuery sia caricata prima di Bootstrap**, in quanto quest'ultimo utilizza jQuery per le sue funzioni interne e quindi si aspetta che esista già un oggetto jQuery quando viene caricato.

N.B. Mentre il file di stile è caricato nell'intestazione del documento, **jQuery** e il file **bootstrap.min.js** vengono richiamati alla fine della pagina, **appena prima di chiudere il tag <body>**.

Questo in conformità alle più recenti indicazioni in fatto di programmazione web.

Il motivo di questa scelta è semplice: il browser, quando riceve dal server il primo tag <script> ha già ricevuto tutto il codice HTML e quindi può già iniziare il rendering della pagina (perchè il DOM della pagina è completo), senza essere rallentato dall'esecuzione di codice Javascript inclusi all'inizio.

Il risultato è quello di velocizzare la visualizzazione delle nostre pagine web.

#### Bootstrap: utilizzare una CDN per includerlo

Per finire questa sezione, come già detto in precedenza, se non volete caricare sul vostro server i file di Bootstrap, avete anche la possibilità di includerli nelle vostre pagine richiamandoli da una **CDN**, ossia da un server che mette a disposizione questi file per il pubblico.

Per richiamare Bootstrap da CND potete usare AD ESEMPIO le righe di codice che seguono:

k rel = "stylesheet" href = "https://stackpath.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" >

<script src = "https://stackpath.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js" ></script>

N.B. La scelta di utilizzare una CDN o caricare fisicamente i file sul proprio server deve essere fatta solo in base a criteri di performances: si ricordi, infatti, che utilizzare file esterni obbligherà il browser dei nostri utenti a fare ulteriori interrogazioni DNS e pertanto, presumibilmente, potrebbero aversi dei leggeri rallentamenti nel caricamento della pagina.

# Bootstrap: la griglia (GRID)

La griglia di Bootstrap (elemento **GRID**) è predisposta per la creazione di **layout fluidi.** Quando si vuole costruire un layout sfruttando la **griglia** di Bootstrap è bene tener presenti

tre semplici cose:

1) Essendo una *griglia* dobbiamo ragionare il nostro layout **distribuendolo per colonne e righe**;

2) Per impostazione predefinita, la griglia si compone di **12 colonne** strutturali, per cui bisogna fare in modo <u>che i vari elementi del nostro layout si distribuiscano sulle colonne</u> <u>della griglia in modo opportuno</u>.

Ogni elemento del nostro layout può occupare una o più colonne della griglia.

La somma totale delle colonne della griglia su cui si estendono in larghezza gli elementi del layout non può superare il numero di 12.

Se le colonne occupate dal layout sono meno di 12, viene lasciato dello spazio vuoto a destra.

Questo significa che, se desidero creare, ad esempio, un layout composto da tre colonne tutte uguali e occupare tutto lo spazio disponibile in larghezza, ciascuna delle colonne del mio layout si dovrà estendere su 4 colonne della griglia (4x3=12). Ma se si vuole una colonna centrale più larga, si potrebbero espandere la prima e la terza colonna del layout su due colonne della griglia e quella centrale su otto colonne (2+8+2=12). Ora importa chiarire il concetto, poi vedremo come implementarlo.

3) Ogni riga può avere un numero colonne diverso e una estensione sulle colonne della griglia diverso. by Prof. Rio Chierego

# Bootstrap: la griglia (GRID)

#### 22

Il **markup HTML** per la costruzione della griglia prevede l'inserimento delle colonne, ovvero dei **div con le classi corrispondenti al numero di colonne che l'elemento occupa**, all'interno di un **div con classe row** che rappresenta la riga.

La somma delle colonne inserita in una riga non può essere superiore a 12:

```
<div class = "container" >
<div class = "row" >
<div class = "col-sm-4" >
...
</div>
<div class = "col-sm-4" >
...
</div>
<div class = "col-sm-4" >
...
</div>
</div>
```

Per limitare la larghezza del layout e centrare al contempo la griglia, basta racchiuderla in un div con classe container (contenitore) con larghezza (width) prefissata in base alla dimensione del device

**N.B.** C'è anche la possibilità di avere una **griglia fluida**, ossia non delimitata in larghezza, sfruttando la proprietà CSS **max-width** ed un **div** di classe **container-fluid**, che permetterà al layout di occupare per intero la larghezza della **viewport** (design edge-to-edge)

### Bootstrap: le classi container e container-fluid

La classe **container** (<u>a larghezza fissa</u>) da usare per la griglia adatta il contenuto ed imposta i margini sinistro e destro cambiando il suo valore di width in base ai pixel del device che stiamo utilizzando.

|                 | Extra small devices<br>Phones (<768px)                          | Small devices Tablets<br>(≥768px)                | Medium devices Desktops<br>(≥992px) | Large devices Desktops<br>(≥1200px) |  |
|-----------------|---|--|-------------------------------------|-------------------------------------|--|
| Grid behavior   | Horizontal at all times   | Collapsed to start, horizontal above breakpoints |                                     |                                     |  |
| Container width | None (auto)   | 750px  | 970px                               | 1170px                              |  |
| Class prefix    | .col-xs-  | .col-sm-   | .col-md-                            | .col-lg-                            |  |
| # of columns    | 12  |  |                                     |                                     |  |
| Column width    | Auto  | ~62px  | ~81px                               | ~97px                               |  |
| Gutter width    | 30px (15px on each side of a column) N.B. Gutter = "spaziatura" |  |                                     |                                     |  |
| Nestable        | Yes   |  |                                     |                                     |  |
| Offsets         | Yes   |  |                                     |                                     |  |
| Column ordering | Yes   |  |                                     |                                     |  |

### Bootstrap: le classi container e container-fluid

24

Se si usa per la griglia la classe **container** (ossia a larghezza fissa) accadrà che, al variare dello schermo del device su cui viene visualizzato, esso assumerà una larghezza (width) prestabilita specificata:

| Device Width      | Container (fixed) Width |
|-------------------|-------------------------|
| 1200px o più      | 1170px                  |
| Da 992px a 1199px | 970px                   |
| Da 768px a 991px  | 750px                   |
| Meno di 768px     | automatico              |

Come già detto se invece si vuole massimizzare la larghezza del contenitore facendolo adattare sempre alla **max-width** del device in uso, allora si può anche scegliere di utilizzare un container fluido usando la classe **container-fluid**.

N.B. A questo punto la width corrisponderà sempre alla larghezza del device in uso

### Bootstrap: la griglia – breakpoint e classi relative

Nel foglio di stile di Bootstrap sono previsti di default quattro breakpoint fondamentali.

N.B. Per **breakpoint** si intende il punto nel quale le colonne da **flottate** (ossia poste una affianco all'altra) possono essere **impilate (o linearizzate** ossia poste una sopra l'altra) e viceversa

- Il primo è per schermi con larghezza inferiore a 768px
- Il secondo è per schermi dalla larghezza minima di 768px in su
- Il terzoè per schermi dalla larghezza minima di 992px in su
- Il quarto per schermi con larghezza minima da 1200px in su

(ossia  $0 < \text{larg} \le 767\text{px}$ ) (ossia **768px** < larg  $\le 991\text{px}$ ) (ossia **992px** < larg  $\le 1199\text{px}$ ) (ossia larg  $\ge 1200\text{px}$ )

A ciascun breakpoint è associato uno specifico prefisso nella definizione delle classi per la griglia:

Primo breakpoint: col-xs-\*(extra-small devices)Secondo breakpoint: col-sm-\*(small devices)Terzo breakpoint: col-md-\*(medium devices)Quarto brekpoint: col-lg-\*(large devices)

N.B. L'asterisco \* va sostituito con un numero da 1 a 12 che indica il **numero di colonne** occupato da un elemento.

### Bootstrap: la griglia – breakpoint e classi relative

26

**Passando da un breakpoint all'altro**, <u>ovvero al variare della larghezza dello schermo del</u> <u>device in pixel</u>, la griglia subisce due modifiche importanti:

- a) viene adeguata la larghezza massima del layout se esso è racchiuso in elemento contenitore con classe container (non così con la classe container-fluid);
- b) viene adeguata la larghezza massima delle colonne.

Tutto è riassunto in questa tabella:

27

Tutto sta nel definire in partenza una strategia, ponendosi due quesiti:

1) Voglio che il layout sia delimitato in larghezza con max-width e centrato rispetto alla finestra del browser?

2) In corrispondenza di quale breakpoint voglio che il layout sia flottato (ossia con le colonne che si dispongono una al fianco dell'altra) oppure sia impilato o linearizzato (ossia con le colonne che si dispongono una sull'altra)?

Se la risposta alla prima domanda è SI, inizierò col definire un div con classe container e al suo interno inserirò righe e colonne:

```
<div class="container">
<!-- Qui va la griglia di Bootstrap -->
</div>
```

La risposta alla seconda domanda determinerà l'uso delle classi (breakpoint) da adottare per impostare le colonne e dipenderà da come sì è progettato il design del layout.

Poniamo di volere un layout che mostri le colonne **impilate sotto i 768px** e **floattate dai 768px** in su. <a href="https://www.selfacture.com/dimensionle-solution-complexity-

Useremo quindi solo le classi col-sm-\*

Ecco una parte del codice:

```
<div class="row">
     <div class="col-sm-4">
     </div>
     <div class="col-sm-4">
     ...
     </div>
     <div class="col-sm-4">
     </div>
  </div>
   <div class="row">
     <div class="col-sm-6">
      . . .
     </div>
     <div class="col-sm-2">
     ...
     </div>
     <div class="col-sm-4">
     </div>
  </div>
</div>
```

#### 29

#### Il layout apparirà impilato (o linearizzato) su uno smartphone (display < 768px):



#### Ma anche su uno schermo di un tablet largo 600px:

| 500x800 💋 | Small Tablet<br>Size: 600x800 (Portrait), Ratio: 3:4 |       |
|-----------|--|-------|
| col-sm-4  |  |       |
| col-sm-4  |  | 11111 |
|           |  |       |

31

Fin quando le colonne non si dispongono una accanto all'altra a partire da 768px:



N.B. Dal momento che il breakpoint è basato su min-width, ovvero sulla larghezza minima, non è necessario specificare le classi per gli altri breakpoint, tutto si adatta semplicemente con il meccanismo dell'ereditarietà.

Se voglio che:

- il layout sia impilato o linearizzato sotto i 992px, userò solo le classi: col-md-\*

- il layout flottante solo sopra i 1200px (scenario poco probabile)userò solo le classi col-lg-\*

Provate a restringere e allargare la finestra del browser visualizzando gli esempi per verificare il tutto.

#### Fin qui le soluzioni minimali.

IMPORTANTE: Nulla ci impedisce infatti di mescolare le classi per modificare a piacere la configurazione del layout in corrispondenza dei breakpoint.

Vediamo l'esempio seguente:

Voglio in questo caso che le colonne della prima riga siano **flottate anche sotto i 768px** ma in maniera differente

La classe corrispondente al breakpoint <768px è **col-xs-**\*, che andremo ad aggiungere in questo modo:



#### 34

È evidente che questa opzione consente un controllo estremo sulla configurazione della griglia. Per ulteriori esempi di classi combinate si può fare riferimento anche agli <u>esempi ufficiali</u> della documentazione di Bootstrap.



### Bootstrap: la griglia – offset

La **griglia di Bootstrap** offre ulteriori funzionalità in grado di accrescere considerevolmente la flessibilità di questo componente.

Abbiamo detto che **non è obbligatorio usare tutte le colonne** della griglia per il layout e che se la somma delle colonne utilizzate è inferiore a 12, le colonne rimanenti restano vuote.

Naturalmente, per impostazione predefinita rimangono "vuote" le colonne a destra.

Parlando di griglie e layout, l'offset si riferisce alla possibilità di spostare verso destra una colonna per uno spazio, definibile a piacere, (quindi lasciando libero lo spazio alla sua sinistra)

Lo spazio liberato con questo spostamento rimane non occupato mantenendo comunque inalterata la struttura del layout.

L'offset, pertanto, si rivela utile quando si voglia creare un margine più ampio tra le colonne oppure quando si voglia creare una colonna centrata e centrale (si veda l'esempio contenuto nella slide seguente)

### Bootstrap: la griglia – offset

L'offset si definisce con classi che richiamano quelle già viste per i breakpoint.

Il prefisso assume la forma di **col-sm-offset-\***, **col-md-offset-\*** e **col-lg-offset-\***, **dove l'asterisco va sostituito con un numero** da **1** a **11** che corrisponde allo spazio in colonne dello spostamento.

#### IMPORTANTE: L'offset non è attivo per il breakpoint extra-small.

#### N.B. La somma delle colonne, compreso l'offset, non deve essere mai superiore a 12

Esempio: Se vogliamo centrare un layout composto da un unico elemento che si estende su 6 colonne strutturali, dovremo impostare 3 di offset. Il codice sarà il seguente

```
<div class="container">
<div class="row">
<div class="col-md-6 col-md-offset-3" >
<h1>Colonna 1</h1> "Lorem ipsum dolor sit amet, ..."
</div>
</div>
```



Ovviamente, giocando con l'offset è possibile creare effetti diversi ed ottenere layout asimmetrici ed "originali" spezzando la monotonia visiva della griglia (vedere slide seguenti)
| col-md-4 |                          | col-md-4 cc | ol-md-offset-4           |
|----------|--------------------------|-------------|--------------------------|
|          | col-md-3 col-md-offset-3 |             | col-md-3 col-md-offset-3 |
|          | col-md-6 col-md-offset-3 |             |                          |



Esempio di applicazione: Supponiamo di voler creare il form seguente

| odulo di  | contatto                       |
|-----------|--------------------------------|
| Nome      | Nome e cognome                 |
| Email     | Indirizzo e-mail               |
| Messaggio | Inserisci qui il tuo messaggio |
|           | Invia                          |

**Abbiamo quattro righe**. Nelle prime tre, la label occupa 3 colonne, la parte con il campo di testo o la textarea occupa 9 colonne:

```
<div class="container">
  <div class="row">
                                         <!-- prima riga: campo Nome -->
      <div class="col-sm-3">
          <label>Nome</label>
      </div>
      <div class="col-sm-9">
          <input type="text" placeholder="Nome e cognome">
      </div>
  </div>
                <!-- seconda riga: campo Email -->
                <!-- terza riga: campo Messaggio -->
                <!-- quarta riga: button -->
    . . . . . . . . . .
</div>
```

Abbiamo quattro righe. Nelle prime tre, la label occupa 3 colonne, la parte con il campo di testo o la textarea occupa 9 colonne.

```
<div class="row"> <!-- seconda riga: campo Email -->
<div class="col-sm-3">
<label>E-mail</label>
</div>
<div class="col-sm-9">
<linput type="text" name="Email" placeholder="Email">
</div>
</div>
</div>
```

Abbiamo quattro righe. Nelle prime tre, la label occupa 3 colonne, la parte con il campo di testo o la textarea occupa 9 colonne.

```
<div class="row"> <!-- terza riga: campo Messaggio -->
<div class="col-sm-3">
<label>Messaggio</label>
</div>
<div class="col-sm-9">
<div class="col-sm-9">
<div class="col-sm-9">
<div class="col-sm-9">
<div class="col-sm-9">
<div class="col-sm-9">
</div>
</div>
</div>
```

Fin qui nulla di particolare. Il modulo si chiude con il pulsante.

Ad esso non è associata però una label! Non possiamo replicare quindi la struttura di prima, ma vogliamo che il bottone sia allineato rispetto alla **textarea** e agli altri campi di testo.

Come fare? Ci viene in soccorso l'offset.

Inseriremo il pulsante in un div pari a 9 colonne e lo sposteremo verso destra per uno spazio pari a 3 colonne (la larghezza delle label), così:

```
<div class="row">
    <div class="col-sm-9 col-sm-offset-3"> <!-- quarta riga: button -->
        <button type="submit" class="btn btn-success btn-block">lnvia</button>
        </div>
</div>
```

#### Bootstrap: la griglia – annidare le colonne

Bootstrap permette di **annidare le colonne**, cioè di inserire una nuova griglia all'interno di un'altra griglia.

Per farlo è sufficiente definire una riga (un div con **classe "row"**) all'interno di una colonna e quindi definire le colonne annidate all'interno di ciascuna riga.

Ogni riga annidata contiene a sua volta una griglia di 12 colonne che si possono riempire esattamente come quella della griglia di primo livello.

| colonna 1        |                   |                   |                    | colonna 2 |  |
|------------------|-------------------|-------------------|--------------------|-----------|--|
| colonne arendata | LUNCOMO ANIVADATA | colonna annistata | colonina anniciata |           |  |
| 1                | 2                 | 3                 | 4                  |           |  |

#### Bootstrap: la griglia – annidare le colonne



# Bootstrap: la griglia – classi di visibilità

Come altri framework della stessa categoria, Bootstrap offre una serie di classi attraverso cui è possibile **nascondere e/o mostrare selettivamente un elemento** in base al breakpoint e alla larghezza corrente dello schermo.

Il caso forse più tipico è quello di un layout a più colonne in cui si voglia mostrare una colonna aggiuntiva solo quando la larghezza supera una certa soglia.

Il meccanismo che regola il funzionamento di queste classi è intuitivo, è sufficiente fare riferimento alla tabella allegata alla <u>documentazione</u> per comprendere come e su quali breakpoint agisce una specifica classe tra quelle disponibili.

Il prefisso hidden- equivale semanticamente a 'nascosto su', mentre visible- a 'visibile su'

Il prefisso è accompagnato dall'indicazione abbreviata dei soliti quattro breakpoint predefiniti di Bootstrap:

xs per extra-small;sm per smallmd per medium;Ig per large.

## Bootstrap: la griglia – classi di visibilità

|            | Extra small devices<br>Phones (<768px) | Small devices<br>Tablets (≥768px) |
|------------|--|-----------------------------------|
| visible-xs | Visible                                | Hidden                            |
| visible-sm | Hidden                                 | Visible                           |
| visible-md | Hidden                                 | Hidden                            |
| visible-lg | Hidden                                 | Hidden                            |
| .hidden-xs | Hidden                                 | Visible                           |
| .hidden-sm | Visible                                | Hidden                            |
| .hidden-md | Visible                                | Visible                           |
| .hidden-lg | Visible                                | Visible                           |

# Bootstrap: la griglia – classi di visibilità (esempio)

#### <div class="container">

```
<div class="row">
 <div class="col-md-4 hidden-sm">
  col-md-4<br>hidden-sm
 </div>
 <div class="col-md-4">
  col-md-4
 </div>
 <div class="col-md-4">
  col-md-4
 </div>
</div>
<div class="row">
 <div class="col-md-4 hidden-sm hidden-xs">
  col-md-4<br>hidden-sm<br>hidden-xs
 \langle div \rangle
 <div class="col-md-4">
  col-md-4
                                  md.
 </div>
 <div class="col-md-4">
  col-md-4
 </div>
</div>
```

In questa riga appariranno inizialmente 3 colonne flottate su schermi md.

Raggiunto il breakpoint **sm**, la prima colonna verrà nascosta mentre le altre 2 colonne verranno impilate.

Raggiunto il breakpoint xs, ritorneranno ad essere visibili ed *impilate tutte e 3 le colonne* 

Anche in questa riga appariranno inizialmente 3 colonne flottate su schermi

Anche qui raggiunto il breakpoint **sm**, la prima colonna verrà nascosta mentre le altre 2 colonne verranno impilate TUTTAVIA tale design rimarrà lo stesso anche una volta raggiunto il breakpoint xs

#### </div>

#### Bootstrap: la barra di navigazione (NAVBAR)

Uno degli elementi comuni a tutti i siti web è la barra di navigazione (o navbar).

La *navbar* è l'elemento strutturale che contiene i link che consentono di muoversi tra le varie sezioni del sito web: in poche parole è l'area della pagina che ospita il **menu** ed altre utilità di navigazione come, ad esempio, un motore di ricerca interna.

**Bootstrap** propone diverse *navbar* e offre la possibilità di inserirvi molti elementi oltre ai collegamenti verso altre pagine: è possibile inserire un campo di ricerca, un elenco a discesa e molti altri elementi.

Le barre di navigazione di **Bootstrap** sono *responsive* per impostazione predefinita: quando la finestra del browser diventa piccola, gli elementi della barra vengono nascosti e, al loro posto, viene visualizzato un pulsante che permette di ri-visualizzarli in pila, quindi non più in orizzontale.

La barra di navigazione è predisposta per contenere **un logo** che rimane sempre visibile anche quando la barra è minimizzata (opzionale).

N.B. Una *navbar* deve ospitare al suo interno un elemento container (fisso o fluido) che non va assolutamente confuso con l'altro elemento container (fisso o fluido) che contiene la struttura responsive della pagina (griglia o GRID)

#### 50

In questa slide proponiamo una barra di navigazione semplificata in modo da focalizzarci sul meccanismo di funzionamento della barra stessa.



La barra di navigazione è realizzata mediante **il tag HTML 5 nav** che appunto è deputato a contenere gli elementi di navigazione di una pagina.

All'interno di questo tag abbiamo due tag **div** che costituiscono **i due elementi** fondamentali dell'elemento navbar:

- 1) navbar-header
- 2) navbar-collapse

La prima parte della barra di navigazione realizzata con la classe navbar-header contiene gli elementi che devono essere visibili anche quando la barra è minimizzata per i display di piccole dimensioni:

- il logo, realizzato con la classe navbar-brand;

- il **pulsante**, realizzato con la classe **navbar-toggle**, da mostrare quando la barra di navigazione è minimizzata e serve, quando cliccato, per visualizzare (n.b. **impilato**) il contenuto della barra stessa.

A tal fine è molto importante l'attributo **data-target**" come vedremo in seguito.

Il pulsante contiene in genere tre elementi span di classe icon-bar.

Ciascuno di questi disegna una lineetta sul pulsante.

La seconda parte della barra è costituita dagli elementi di navigazione veri e propri: essi sono racchiusi in un div di classe collapse navbar-collapse e in una lista non ordinata di classe nav navbar-nav.

E' bene precisare che l'identificativo **myNavbar** non ha effetti stilistici ma è **FONDAMENTALE** per potere associare l'azione del bottone all'apertura del menù.

# N.B. Tale identificativo deve corrispondere al valore di data-target del bottone con classe navbar-toggle facendo attenzione a farlo precedere dal carattere '#'

Si noti, ancora, che tra i vari item della lista uno è contrassegnato con la classe **active**: ciò serve a formattare quella specifica voce di menu in modo distinto dalle altre e segnala che il collegamento di quella voce porta alla pagina corrente (cioè quella che stiamo già visualizzando).

**Suggerimento**: se desiderate distanziare la barra di navigazione del bordo della finestra del browser sarà sufficiente aggiungere un po' di padding all'elemento body della pagina. Ad esempio utilizzando la possibilità di inserire elementi di CSS in linea:

<body style="padding-top: 10px" >

|                                       | A Direction of the second seco |  |
|---------------------------------------|--|--|
| WebSiteName Home Page 1 Page 2 Page 3 | WebSiteName  |  |
| $\bigwedge$                           | Home   |  |
| Barra di navigazione completa         | Page 1   |  |
| 7                                     | Page 2   |  |
|                                       | Page 3   |  |
|                                       |  |  |
| Barra di navigazione collassata       |  |  |

### Bootstrap: la barra di navigazione (invertita)

<nav class="navbar navbar-inverse navbar-fixed-top"> <div class="container-fluid"> <div class="navbar-header"> <button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#myNavbar"</pre> <span class="icon-bar"></span> <span class="icon-bar"></span> <span class="icon-bar"></span> Prima novità </button> <a class="navbar-brand" href="#">WebSiteName</a> Voce di menù con sottomenu </div> <div class="collapse navbar-collapse" id="myNavbar</pre> class="active"><a href="#">Home</a> <a class="dropdown-toggle" data-toggle="dropdown" href="#">Page 1 <span class="caret"></span></a> <a href="#">Page 1-1</a> <a href="#">Page 1-2</a> <a href="#">Page 1-3</a> </1i> La classe caret inserisce un al fianco <a href="#">Page 2</a> <a href="#">Page 3</a> della voce di menù con sottomenu <a href="#"><span class="glyphicon glyphicon-user"></span> Sign Up</a> <a href="#"><span class="glyphicon glyphicon-log-in"></span> Login</a> </div> Seconda novità </div> by Prof. Rio Chierego Uso dell'elemento nav navbar-nav navbar-right e delle glyphicon </nav>





#### Bootstrap: immagini

Le forme (**ATTENZIONE: NON RESPONSIVE**) che è possibile assegnare alle immagini con Bootstrap, utilizzando il tag HTML IMG, sono realizzabili attraverso l'utilizzo delle seguenti **classi**:

Rounded Corners:



A) La classe img-rounded aggiunge gli spigoli arrotondati ad una immagine
 N.B. IE8 non supporta questa caratteristica

<IMG src="foto.jpg" class="img-rounded" alt="Foto" width="304" height="236" >

Circle:



Thumbnail:

B) La classe img-circle adatta l'immagine ad un cerchio
 N.B. IE8 non supporta questa caratteristica

<IMG src="foto.jpg" class="img-circle" alt="Foto" width="304" height="236" >



C) La classe img-thumbnail adatta l'immagine ad una anteprima (bordata)
<IMG src="foto.jpg" class="img-thumbnail" alt="Foto" width="304" height="236" >

#### Bootstrap: immagini responsive

57

Nelle scorse slide abbiamo già visto **che le "celle" delle colonne del layout** (creato con l'ausilio della "griglia" di Bootstrap) **sono completamente** *responsive*, ossia si adattano alle dimensioni del display del dispositivo in cui vengono visualizzate.

Naturalmente se le celle contengono testo questo non è un problema: il testo si adatta facilmente alle diverse dimensioni del contenitore che lo ospita e lo abbiamo già visto negli esempi proposti nella slide precedenti.

Altrettanto, purtroppo, non si può dire delle immagini se trattate con il tag IMG senza utilizzo di opportune classi Bootstrap

Infatti per rendere l'immagine *responsiva* è sufficiente assegnare all'immagine stessa la classe **img-responsive** 

Questa classe assegna all'immagine le proprietà CSS

#### max-width: 100%; e height: auto

in modo che l'immagine si ridimensioni automaticamente in base all'elemento genitore senza mai varcarne i confini.

Esempio:

```
<img src="foto.jpg" class="img-responsive" >
```

# Bootstrap: formattazione del testo nei paragrafi

Il testo del corpo della pagina, per impostazione predefinita, in Boostrap, è reso con dimensione (font-size) 14px e una interlinea (line-height) di 20px.

Inoltre ai paragrafi viene applicato un margine inferiore (margin-bottom) di 10 px.

Ovviamente esistono delle **classi** che permettono di modificare velocemente queste impostazioni base.

Ad esempio, assegnando ad un paragrafo la classe **lead**, lo si evidenzia rispetto agli altri

.....

Esistono classi che applicano dei colori per veicolare un significato:

- text-muted testo grigio poco enfatizzato;
- text-primary testo blu evidenziato;
- text-success testo verde elemento positivo;
- text-info testo azzurro informazione;
- text-warning testo arancio bisogna prestare attenzione;
- text-danger testo rosso pericolo.

# Bootstrap: formattazione del testo nei paragrafi

Bootstrap dispone anche di classi specifiche per l'allineamento dei paragrafi che non richiedono un particolare commento in quanto il loro nome è sufficientemente auto-esplicativo:

- text-left testo allineato a sinistra
- text-center testo allineato al centro
- text-right testo allineato a destra
- text-justify testo giustificato

In Bootstrap esistono inoltre tre classi per la trasformazione del testo:

- text-lowercase trasforma il testo in minuscolo
- text-uppercase trasforma il testo in maiuscolo
- text-capitalize scrive il testo con la prima lettera maiuscola

Esempio di utilizzo combinato:

N.B. Il paragrafo sarà evidenziato rispetto agli altri, il testo scritto in blu, centrato e tutto in maiuscolo

#### Bootstrap: le tabelle (tabella base)

La classe **.table** aggiunge alla tabella solo elementi di stile di base (una leggera spaziatura tra le celle and divisori orizzontali)

| Nome     | Cognome  | Email                |
|----------|----------|----------------------|
| Giovanni | Bianchi  | bianchi@example.com  |
| Maria    | Salemme  | salemme@example.com  |
| Giacomo  | Barletta | barletta@example.com |

#### Bootstrap: le tabelle (tabella base)

#### <div class="container">

```
<h2>Tabella base</h2>
La classe <b>.table</b> aggiunge alla tabella solo elementi di stile di base
<thead>
  \langle t,r \rangle
    Nome
    Cognome
    Email
  </thead>
 \langle tr \rangle
    Giovanni
    Bianchi
    bianchi@example.com
  \langle tr \rangle
    Maria
    Salemme
    salemme@example.com
  \langle t,r \rangle
    Giacomo
    Barletta
    barletta@example.com
```

#### Bootstrap: le tabelle (a strisce e bordata)

#### La classe .table-striped aggiunge delle strisce alternate alla tabella (zebratura)

| Nome     | Cognome  | Email                |
|----------|----------|----------------------|
| Giovanni | Bianchi  | bianchi@example.com  |
| Maria    | Salemme  | salemme@example.com  |
| Giacomo  | Barletta | barletta@example.com |

# La classe **.table-bordered** aggiunge i lati a tutte le celle della tabella compresa l'intestazione

| Nome     | Cognome  | Email                |
|----------|----------|----------------------|
| Giovanni | Bianchi  | bianchi@example.com  |
| Maria    | Salemme  | salemme@example.com  |
| Giacomo  | Barletta | barletta@example.com |

#### Bootstrap: le tabelle (Hover rows e condensata)

La classe **.table-hover** aggiunge un effetto hover su sfondo grigio alle righe della tabella mentre si scorrono con il puntatore

| Nome     | Cognome  | Email                |
|----------|----------|----------------------|
| Giovanni | Bianchi  | bianchi@example.com  |
| Maria    | Salemme  | salemme@example.com  |
| Giacomo  | Barletta | barletta@example.com |

#### La classe .table-condensed compatta la spaziatura tra tutte le celle

| Nome     | Cognome  | Email                |
|----------|----------|----------------------|
| Giovanni | Bianchi  | bianchi@example.com  |
| Maria    | Salemme  | salemme@example.com  |
| Giacomo  | Barletta | barletta@example.com |

#### Bootstrap: le tabelle (contestuale)

La classe **.table** può essere usata per colorare le righe (rows) o le celle (cells) di una tabella in aggiunta alle classi contestuali:

| .active  | applica il colore hover alla cella oppure alla riga selezionata |
|----------|---|
| .success | indica un'azione che è avvenuta con successo                    |
| .info    | indica un'azione neutra o informativa                           |
| .warning | indica un avviso da tenere in considerazione                    |
| .danger  | indica un'azione potenzialmente pericolosa o negativa           |
|          |   |

Vediamo come.....

#### Bootstrap: le tabelle (contestuale)

La classe **.table** può essere usata per colorare le righe (rows) o le celle (cells) di una tabella in aggiunta alle classi contestuali:

.activeapplica il colore hover alla cella oppure alla riga selezionata.successindica un'azione che è avvenuta con successo.infoindica un'azione neutra o informativa.warningindica un avviso da tenere in considerazione.dangerindica un'azione potenzialmente pericolosa o negativa

applicate ai tag TR oppure TD

| Nome    | Cognome    | Email                |
|---------|------------|----------------------|
| Default | Defaultson | default@example.com  |
| Active  | Bianchi    | bianchi@example.com  |
| Success | Salemme    | salemme@example.com  |
| Info    | Barletta   | barletta@example.com |
| Warning | Rossi      | rossi@example.com    |
| Danger  | Verdi      | verdi@example.com    |
| Warning | Rossi      | rossi@example.com    |

# Bootstrap: le tabelle (responsive)

Tutte le tabelle create in precedenza adatteranno il loro contenuto man mano che le misure dei display aumenteranno o diminuiranno

Nel caso in cui si volesse mantenere un certo aspect-ratio dei dati tabellati per certi dispositivi è possibile utilizzare la classe **.table-responsive** 

La classe .table-responsive crea una tabella responsive le cui righe scrolleranno (mantenendo inalterato quindi il contenuto) su dispositivi con schermi al di sotto dei 768 pixel e rimarranno inalterate su quelli più grandi.

Per fare questo occorrerà inglobare il tag TABLE con un tag di tipo DIV

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

# Bootstrap: le tabelle (responsive)

67

| # | Nome    | Cognome  | Età | Città    | Paese  |
|---|---------|----------|-----|----------|--------|
| 1 | Anna    | Pitt     | 35  | New York | USA    |
| 2 | Maria   | Stuarda  | 45  | Pasadena | USA    |
| 3 | Giacomo | Barletta | 45  | Napoli   | Italia |

La tabella viene mostrata per intero su device con display maggiori di 768px

| # | Nome    | Cognome  | Età | Città    |
|---|---------|----------|-----|----------|
| 1 | Anna    | Pitt     | 35  | New York |
| 2 | Maria   | Stuarda  | 45  | Pasadena |
| 3 | Giacomo | Barletta | 45  | Napoli   |
| • |         |          |     | •        |

La tabella viene mostrata con una barra di scrolling orizzontale su device con display maggiori di 768px

# Bootstrap: i layout dei form (responsive)

I controlli presenti nei form ricevono alcune caratteristiche di default con Bootstrap.

Tutti gli elementi contenenti testo come **<input>**, **<textarea>**, **and <select>** se utilizzati con la class **form-control** saranno visualizzati su di uno schermo con un'ampiezza pari al 100%

Inoltre Bootstrap mette a disposizione degli utenti tre tipi di layout per i form:

- Verticale (default)
- Orizzontale
- In linea

Ci sono delle regole standard obbligatorie comuni da rispettare qualunque sarà il layout che si intenderà utilizzare tra i tre previsti :

Occorre usare sia le etichette sia l controlli dei form all'interno di un div utilizzando la classe form-group (necessario per ottenere un'ottima spaziatura con il ridimensionamento degli schermi dei device)

Occorre aggiungere la classe form-control a tutti gli elementi del form di tipo <input>, <textarea>, e <select>

### Bootstrap: il Form Verticale (default)

```
<div class="container">
    <form>
       <div class="form-group">
           <label for="email">Indirizzo Email:</label>
          <input type="email" class="form-control" id="email">
       </div>
       <div class="form-group">
          <label for="pwd">Password:</label>
          <input type="password" class="form-control" id="pwd">
       </div>
       <div class="checkbox">
          <label><input type="checkbox"> Ricordami</label>
       </div>
       <button type="submit" class="btn btn-default">Invia</button>
    </form>
</div>
```

#### Bootstrap: il Form Verticale (default)

| ndirizzo Email: |  |  |
|-----------------|--|--|
|                 |  |  |
| Password:       |  |  |
| Ricordami       |  |  |

#### Bootstrap: il Form in linea

In un form in linea tutti I suoi elementi appariranno su di un unica linea, allineati a sinistra con le etichette poste a lato

N.B. Questo layout si applica solo a form che si trovano all'interno di una viewport (ossia dimensioni dell'area di visualizzazione di un browser) larga almeno 768px. In caso contrario verrà mostrato per il form il layout verticale di default anche se dichiarato in linea

Regola aggiuntiva da applicare nel caso si desideri un form con layout in linea:

Occorre aggiungere la classe form-inline all'elemento <form>

**Suggerimento:** Se non si include una label per ciascun campo di input, **gli screen readers** (ossia quelle applicazioni software che identificano ed interpretano il testo mostrato sullo schermo di un computer, presentandolo tramite sintesi vocale o attraverso un display braille) potrebbero avere problemi nella visualizzazione del form.

E' possibile, utilizzando la **classe sr-only**, nascondere le label per tutti i tipi di device ad eccezione proprio degli screen readers

#### Bootstrap: il Form in linea

```
Password:
Indirizzo Email:
                                                                                                                                                                                                                                                                                           Ricordami
                                                                                                                                                                                                                                                                                                                                       Invia
<div class="container">
                    <form class="form-inline">
                                     <div class="form-group">
                                                       <label class="sr-only" for="email">Indirizzo Email:</label>
                                                    <input type="email" class="form-control" id="email">
                                     </div>
                                     <div class="form-group">
                                                    <label class="sr-only" for="pwd">Password:</label>
                                                    <input type="password" class="form-control" id="pwd">
                                     </div>
                                      <div class="checkbox">
                                                    <label><input type="checkbox"> Ricordami</label>
                                     </div>
                                   <br/>

                    </form>
</div>
```
### Bootstrap: il Form orizzontale

Per form con layout orizziontale si intende un form in cui le etichette risultano allineate ai campi di input per gli schermi **medi** e **grandi** (**dimensione maggiore o uguale a 768px**)

Sugli schermi **normali** (con **dimensione minore di 768px**), il form assumerà il layout verticale di default con le etichette posizionate sopra ciascun campo di input

Regole aggiuntive da applicare nel caso si desideri un form con layout orizzontale:
Occorre aggiungere la classe form-horizontal all'elemento <form>
Occorre aggiungere la classe control-label a tutte le etichette presenti

Suggerimento importante: Utilizzare le classi della griglia predefinite in Bootstrap per allineare orizzontalmente le etichette ed i controlli del form

# Bootstrap: il Form orizzontale

#### 74

```
<form class="form-horizontal">
  <div class="form-group">
    <a>label class="control-label col-sm-2" for="email">Email:</a>
    <div class="col-sm-10">
       <input type="email" class="form-control" id="email" placeholder="Digita l'email">
    </div>
 </div>
 <div class="form-group">
    <label class="control-label col-sm-2" for="pwd">Password:</label>
     <div class="col-sm-10">
       <input type="password" class="form-control" id="pwd" placeholder="Digita la password">
     </div>
  </div>
 <div class="form-group">
     <div class="col-sm-offset-2 col-sm-10">
       <div class="checkbox">
           <label><input type="checkbox"> Ricordami</label>
       </div>
     </div>
 </div>
 <div class="form-group">
   <div class="col-sm-offset-2 col-sm-10">
      <button type="submit" class="btn btn-default">Invia</button>
   </div>
 </div>
</form>
```

## Bootstrap: il Form orizzontale

#### 75

Layout orizzontale per schermi medi e grandi (dimensione maggiore o uguale a 768px)

| Email:    | Digita l'email           |  |  |  |  |
|-----------|--------------------------|--|--|--|--|
| Password: | vord: Digita la password |  |  |  |  |
|           | Ricordami                |  |  |  |  |
|           | Invia                    |  |  |  |  |

Layout orizzontale per schermi normali e piccoli (dimensione minore di 768px)

| Email:             |  |  |
|--------------------|--|--|
| Digita l'email     |  |  |
| Password:          |  |  |
| Digita la password |  |  |
| Ricordami          |  |  |
| Invia              |  |  |

#### **Bootstrap:** Accordion

L'accordion è uno strumento utile per presentare una grande quantità di contenuto nelle pagine occupando poco spazio. Il contenuto, in pratica, viene suddiviso in blocchi che poi vengono mostrati singolarmente attraverso un meccanismo a "fisarmonica" (quando si apre una "voce" si chiude quella che era aperta in precedenza).

Per costruire l'accordion, si usa il plugin **collapsible** integrato in Bootstrap.

| Pannello 1                        |  |  |  |  |  |  |
|-----------------------------------|--|--|--|--|--|--|
| Lorem ipsum dolor sit amet        | Se clicco su "Pannello 1" e si apre il suo contenuto chiudendosi quello di tutti gli altri |  |  |  |  |  |
| Pannello 2                        |  |  |  |  |  |  |
|                                   |  |  |  |  |  |  |
| Pannello 1                        |  |  |  |  |  |  |
| Pannello 2                        | Se invece clicco su "Pannello 2" e si anre il suo  |  |  |  |  |  |
| Curabitur pretium tincidunt lacus | contenuto chiudendosi quello aperto prima cliccando<br>su "Pannello 1"                     |  |  |  |  |  |

#### **Bootstrap: Accordion**

```
<div class="panel-group" id="accordion">
 <div class="panel panel-default">
   <div class="panel-heading">
     <h4 class="panel-title">
       <a class="accordion-toggle" data-toggle="collapse"
          data-parent="#accordion" href="#uno">
         Pannello 1
       \langle a \rangle
     </h4>
   </div>
   <div id="uno" class="panel-collapse collapse in">
     <div class="panel-body">
       Lorem ipsum dolor sit amet... 
     </div>
   </div>
 </div>
 <div class="panel panel-default">
   <div class="panel-heading">
     <h4 class="panel-title">
       <a class="accordion-toggle" data-toggle="collapse"
          data-parent="#accordion" href="#due">
         Pannello 2
       </a>
     </h4>
   </div>
   <div id="due" class="panel-collapse collapse">
     <div class="panel-body">
       Curabitur pretium tincidunt lacus... 
     </div>
   </div>
 </div>
</div>
```

Questo plugin Bootstrap permette di creare una sequenza di immagini o altri elementi a scorrimento.

La realizzazione di un carousel richiede diversi elementi. Innanzi tutto, è necessario un <div> che funga da contenitore di tutti gli elementi del carousel e che deve avere le classi **carousel** e **slide**.

> <div id="mioCarousel" class="carousel slide"> Contenuto del carousel </div>

All'interno del div vanno poi creati tre sezioni. Ne vedremo uno alla volta.

- 1) Sezione degli **indicatori:** lista di elementi vuoti che segnaleranno all'utente quanti sono gli elementi del carousel e quale si sta visualizzando
- 2) Sezione degli **schermate/slides:** contiene gli oggetti che andranno mostrati uno alla volta nel carousel all'utente
- 3) Sezione delle **frecce avanti/indietro:** vanno create le frecce laterali che permetteranno lo spostamento in avanti ed al'indietro tra gli oggetti del carousel

 Sezione degli indicatori: è costituita da una lista di elementi vuoti che costituiranno gli indicatori che segnalano all'utente quanti sono gli elementi del carosuel e qual'è quello attualmente visualizzato.

```
    data-target="#mioCarousel" data-slide-to="0" class="active">
    data-target="#mioCarousel" data-slide-to="1">
    data-target="#mioCarousel" data-slide-to="1">
    data-target="#mioCarousel" data-slide-to="2">
```

Ovviamente, gli elementi devono essere in numero uguale alle "schermate" da mostrare nel carousel e devono contenere l'attributo **data-target** il cui valore deve corrispondere all'ID del <div> contenitore del carousel

2) Sezione degli **schermate/slides:** contiene gli oggetti che andranno mostrati uno alla volta nel carousel all'utente

Occorre creare un div di classe **carousel-inner** che conterrà una serie di div di classe **item** ognuno dei quali contiene ciò che deve essere mostrato in ogni schermata/slide del nostro carousel.

La prima schermata mostrata avrà anche la classe **active** che poi sarà assegnata dal plugin alle successive schermate, man mano che vengono mostrate.

```
<div class="carousel-inner">
<div class="item active">
<img src="img/foto1.JPG" alt="...">
</div>
<div class="item">
<img src="img/foto2.JPG" alt="...">
</div>
<div class="item">
</div>
<div class="item">
</div>
</div>
```

2) Sezione degli **schermate/slides:** contiene gli oggetti che andranno mostrati uno alla volta nel carousel all'utente

In ogni scheda può essere creato un titolo che viene mostrato appena sopra agli indicatori. Il titolo viene realizzato con un div di classe **carousel-caption**. In questo div sono inseriti i tag necessari a creare il titolo (ad esempio un H1 ed un paragrafo).

```
<div class="item active">
<img src="img/foto 1.JPG" alt="...">
<div class="carousel-caption">
<h3>Primo passaggio</h3>
i materiali necessari
</div>
```

3) Sezione delle **frecce avanti/indietro:** vanno create le frecce laterali che permetteranno lo spostamento in avanti ed al'indietro tra gli oggetti del carousel

Per concludere la creazione del nostro carousel, occorre creare le frecce laterali (una a destra ed una a sinistra) che permettono all'utente di spostarsi da una schermata all'altra senza attendere lo scorrimento automatico del carousel: si tratta dei classici comandi *Prev* e *Next*.

Questi elementi di navigazione vengono realizzati mediante dei tag **<a>** cui viene assegnata la classe **carousel-control** in aggiunta alla classe **left** o **right**.

E' inoltre necessario assegnare all'attributo **href** di queste due ancore l'ID del div contenitore del carousel.

```
<a class="left carousel-control" href="#mioCarousel" data-slide="prev">
<span class="icon-prev"></span>
</a>
```

<a class="right carousel-control" href="#mioCarousel" data-slide="next">

```
<span class="icon-next"></span>
```

</a>

#### Gestire la larghezza del Carousel

Per impostazione predefinita un carousel è largo quanto il suo elemento genitore e se le schede che lo costituiscono sono più piccole rimane diverso spazio intorno ad esse. Tra l'altro, questo spazio si dispone sul lato destro a meno che non si centri il div che costituisce la scheda.

Per ridurre le dimensioni del carousel, basta creare una regola CSS (che andremo ad inserire nel nostro foglio di stile) per l'elemento div che racchiude tutti gli elementi del carousel.

Ad esempio:

#### **Opzioni del Carousel**

Il comportamento del carousel può essere influenzato dall'uso di opzioni passate attraverso una serie di attributi di tipo **data-**\* oppure mediante codice JavaScript.

La tabella che segue riepiloga le opzioni disponibili per la serie di attributi di tipo **data\_\***, i loro valori predefiniti e il loro significato.

| Nome     | Tipo                    | Valore<br>predefinito | Descrizione  |
|----------|-------------------------|-----------------------|--|
| interval | numero<br>o<br>booleano | 5000                  | Indica il tempo in millisecondi trascorso il quale viene visualizzata<br>una nuova schermata. Se assume il valore false il carousel non<br>avanza automaticamente, ma solo tramite le apposite frecce. |
| pause    | stringa                 | "hover"               | Se si configura questa opzione, l'avanzamento del carousel si ferma<br>quando il mouse passa su di esso, riprende quando esce.   |
| wrap     | bo <mark>o</mark> leano | true                  | Indica se il carousel riprende dall'inizio l'avanzamento dopo aver<br>visualizzato tutte le schede. Se assume il valore false, il carousel si<br>interrompe quando raggiunge l'ultima scheda.          |

# Bootstrap: i titoli delle pagine (page-header)

#### **PAGE-HEADER**

Per creare l'header delle pagine (intesa come area di intestazione visibile all'utente), Bootstrap dispone di una apposita classe, **page-header**, che appunto crea una area in cui inserire il titolo H1 ed eventuale testo supplementare.

Vediamo il codice HTML:

#### <div class="page-header">

<h1>Intestazione pagina <small>testo aggiuntivo</small></h1></div>

Naturalmente, nell'intestazione della pagina è possibile aggiungere altri elementi come, ad esempio, un pulsante di navigazione.

# Bootstrap: i titoli delle pagine (jumbotron)

86

Se però si desidera una intestazione di pagina più "evidente" allora potrebbe essere utile ricorrere al componente **Jumbotron**, che crea una area rettangolare coi bordi arrotondati estesa per tutta l'area orizzontale disponibile.

L'area ha uno sfondo scuro che la fa risaltare nettamente rispetto al resto della pagina.

Il componente Jumbotron si crea con un elemento div a cui viene applicata la classe **jumbotron**, in questo modo:

```
<div class="jumbotron">
```

<h1>Intestazione pagina <small>testo aggiuntivo</small></h1> Lorem ipsum dolor sit amet, ...

```
<a class="btn btn-primary btn-lg">Learn more</a>
```

</div>



by Prof. Rio Chierego

#### Bootstrap: gestire le thumbnail all'interno della griglia

Il *grid system* di Bootstrap di cui abbiamo parlato nelle lezioni iniziali del corso può essere espanso con l'uso del componente **thumbnail** che permette di creare facilmente griglie ordinate di immagini, video, e molto altro.

Creando un elemento di classe **thumbnail** in ogni colonna della griglia si potrà mostrare una immagine o altro contenuto riquadrato.

Ecco una griglia che mostra 6 immagini distribuite in due righe da tre foto ciascuna.



by Prof. Rio Chierego

#### Bootstrap: gestire le thumbnail all'interno della griglia

```
<div class="row">
                                                       PRIMA RIGA
   <div class="col-md-4">
    <div class="thumbnail">
        <img src="lunaPiccola.JPG" class="img-responsive" alt="...">
    </div>
   </div>
   <div class="col-md-4">
     <div class="thumbnail">
        <img src="lunaPiccola.JPG" class="img-responsive" alt="...">
     </div>
  </div>
  <div class="col-md-4">
     <div class="thumbnail">
        <img src="lunaPiccola.JPG" class="img-responsive"alt="...">
     </div>
  </div>
</div>
```

#### Bootstrap: gestire le thumbnail all'interno della griglia

```
<div class="row">
                                                       SECONDA RIGA
   <div class="col-md-4">
    <div class="thumbnail">
        <img src="lunaPiccola.JPG" class="img-responsive" alt="...">
    </div>
   </div>
   <div class="col-md-4">
     <div class="thumbnail">
        <img src="lunaPiccola.JPG" class="img-responsive" alt="...">
     </div>
  </div>
  <div class="col-md-4">
     <div class="thumbnail">
        <img src="lunaPiccola.JPG" class="img-responsive"alt="...">
    </div>
  </div>
</div>
```