



EBook Gratuito

APPENDIMENTO

twitter-bootstrap

Free unaffiliated eBook created from
Stack Overflow contributors.

#twitter-
bootstrap

Sommario

Di.....	1
Capitolo 1: Iniziare con twitter-bootstrap.....	2
Osservazioni.....	2
Versioni.....	2
Examples.....	3
Installazione / Configurazione.....	3
Modello base.....	5
Quando usare Bootstrap.....	6
Pagina Web di base che utilizza componenti di bootstrap.....	6
Capitolo 2: Bootstrap Affix.....	9
Examples.....	9
Su Navbar.....	9
Affisso Esempio 2.....	9
Capitolo 3: Bootstrap Navbar.....	11
Examples.....	11
Bootstrap Navbar.....	11
Immagine di marca di Bootstrap.....	11
Capitolo 4: caroselli.....	13
Osservazioni.....	13
Examples.....	13
Utilizzo HTML di base.....	13
Utilizzo e inizializzazione di Javascript di base.....	14
Capitolo 5: Classi di utilità.....	15
Examples.....	15
Genera classi .hidden- * per tutti i punti di interruzione - SCSS.....	15
Capitolo 6: colonne.....	16
Examples.....	16
Colonne reattive stessa altezza (solo CSS o SASS).....	16
Capitolo 7: Componenti Bootstrap.....	20
Osservazioni.....	20

Examples.....	20
Esempi di componenti Bootstrap.....	20
Capitolo 8: Contenitori di bootstrap.....	21
introduzione.....	21
Examples.....	21
contenitori.....	21
Capitolo 9: Convalida Bootstrap.....	22
Osservazioni.....	22
Examples.....	22
Utilizzo di ASP.NET MVC e annotazioni di dati.....	22
Esempio di input che richiede la convalida.....	23
Opzionale.....	23
Capitolo 10: Dialoghi modali.....	25
Osservazioni.....	25
Examples.....	25
Utilizzo HTML di base.....	25
Utilizzo e inizializzazione di Javascript di base.....	25
Capitolo 11: Distintivi e etichette Bootstrap.....	27
Examples.....	27
badge.....	27
etichette.....	27
Capitolo 12: Dropdown di avvio.....	28
Parametri.....	28
Osservazioni.....	28
Examples.....	28
Come usare.....	28
Esempio di base.....	29
Capitolo 13: Elenco gruppo.....	30
Osservazioni.....	30
Examples.....	30
Esempio di base.....	30

badge.....	30
Articoli collegati.....	30
Elementi del pulsante.....	30
Articoli disabilitati.....	31
Classi contestuali.....	31
Contenuto personalizzato.....	31
Capitolo 14: Glyphicons.....	32
Osservazioni.....	32
Examples.....	32
Come usare i glyphicons.....	32
Capitolo 15: Grid Nesting.....	34
introduzione.....	34
Osservazioni.....	34
Examples.....	34
Colonne di nidificazione.....	34
Capitolo 16: Jumbotron.....	35
introduzione.....	35
Osservazioni.....	35
Examples.....	35
Jumbotron di base con due righe di testo e un pulsante.....	35
Capitolo 17: Le forme.....	36
Examples.....	36
Forma di base.....	36
Ingressi di sola lettura e disabilitati.....	36
Capitolo 18: menu a discesa.....	37
Osservazioni.....	37
Examples.....	37
Utilizzo HTML di base.....	37
Capitolo 19: Menu di navigazione.....	38
Examples.....	38
Menu pillola orizzontale.....	38
Menu pillola verticale.....	38

Pillola orizzontale reattiva a tutta larghezza.....	38
Capitolo 20: Mettere in guardia.....	39
Osservazioni.....	39
Examples.....	39
Tipi di avviso.....	39
Avviso di esempio di base.....	39
Avvisi animati.....	40
Avvisi non ammissibili.....	40
Collega il colore negli avvisi.....	40
Capitolo 21: Migrazione a Bootstrap 4.....	42
introduzione.....	42
Osservazioni.....	42
Examples.....	42
Cambiamenti nel layout di colonne del sistema di griglia in Bootstrap 4.....	42
Layout di griglia Bootstrap 4.....	42
Modifiche al supporto del browser.....	43
Applica la rimozione della classe.....	43
Bootstrap 4 Navbar.....	46
Bootstrap 3 to Bootstrap 4 Modifiche CSS.....	49
Bootstrap 4 Vertical Align.....	51
Bootstrap 4 Centering.....	53
Centro orizzontale.....	53
Centro verticale.....	54
Bootstrap 4 Column Order.....	54
Capitolo 22: Modals.....	56
Osservazioni.....	56
Examples.....	56
Modale HTML di base.....	56
Capitolo 23: Nav.....	57
Examples.....	57
Bootstrap Navs.....	57
Capitolo 24: Navbar.....	58

Examples.....	58
Barra di navigazione di base (fissa nella parte superiore della pagina).....	58
Sottomenu in navbar.....	58
Separatore navbar.....	59
Mantieni il link di navigazione corrente "attivo".....	59
Cambia punto di interruzione Navbar (mobile vs normale).....	59
Chiudi la barra di navigazione compressa quando fai clic all'esterno della barra di naviga.....	60
Capitolo 25: paginatura.....	61
introduzione.....	61
Examples.....	61
Un semplice esempio di impaginazione.....	61
Capitolo 26: pannelli.....	62
Osservazioni.....	62
Examples.....	62
Esempio di base.....	62
Pannello con intestazione.....	62
Pannello con il piè di pagina.....	62
Capitolo 27: Personalizzazione dello stile di bootstrap di Twitter.....	64
Osservazioni.....	64
Examples.....	64
Sostituzione del CSS predefinito.....	64
Capitolo 28: pulsanti.....	66
Sintassi.....	66
Examples.....	66
Classi di pulsanti.....	66
Capitolo 29: Sistema di griglia.....	69
introduzione.....	69
Osservazioni.....	69
Examples.....	69
Media Queries.....	69
Livelli griglia di bootstrap (punti di interruzione).....	69
Bootstrap Rows & Columns.....	71

contenitori.....	72
Colonne di compensazione.....	73
Manipolazione dell'ordine delle colonne tramite push e pull.....	73
Capitolo 30: Stampa in Bootstrap.....	75
Examples.....	75
Utilizzo HTML di base.....	75
Capitolo 31: tabelle.....	76
Examples.....	76
Tavolo semplice.....	76
Capitolo 32: tabelle.....	77
Osservazioni.....	77
Examples.....	77
Tavolo di base.....	77
Tavolo con stile avanzato.....	77
Righe a righe.....	77
Tavolo bordato.....	78
Passa il mouse sulle righe.....	78
Tavolo condensato.....	78
Classi contestuali.....	78
Tavoli reattivi.....	79
Reflow tabella: intestazioni verticali.....	79
Capitolo 33: Tabs.....	81
Examples.....	81
HTML di base.....	81
Schede animate.....	81
Capitolo 34: Temi Bootstrap.....	83
Examples.....	83
Temi di bootstrap rispetto alle sovrascritture delle regole.....	83
Capitolo 35: tooltip.....	84
Osservazioni.....	84
Examples.....	84

Tooltips di posizionamento.....	84
Esempio di base.....	84
Capitolo 36: Utilizzo di Clearfix in righe e colonne.....	86
introduzione.....	86
Osservazioni.....	86
Examples.....	86
The Naive First Attempt.....	86
Il problema dell'altezza.....	88
Clearfix a Rescue.....	89
Un cruscotto.....	92
2,4,6 Layout con Clearfix.....	97
Perché le colonne Bootstrap superano 12 in una riga?.....	99
Titoli di coda.....	102

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [twitter-bootstrap](#)

It is an unofficial and free twitter-bootstrap ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official twitter-bootstrap.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capitolo 1: Iniziare con twitter-bootstrap

Osservazioni

Bootstrap è un framework HTML, CSS e JavaScript utilizzato per creare siti Web creati utilizzando un paradigma mobile-first e responsive web design (RWD). Usa una combinazione di classi CSS premesse e JavaScript per creare una varietà di cose sul web. Include elementi come una griglia reattiva e personalizzata che consente di visualizzare i siti Web su qualsiasi schermo, una barra di navigazione a discesa che è in grado di essere reattiva e anche le cose più semplici che possono richiedere molto tempo come pulsanti premediti, moduli e fisarmoniche per nominare un pochi.

Bootstrap può essere utile per i seguenti motivi:

1. **Risparmio di tempo** : Bootstrap presenta molte cose che sono pre-costruite e che devono semplicemente essere richiamate durante la scrittura del codice. Ciò consente di risparmiare una notevole quantità di tempo e può ridurre notevolmente il tempo necessario per codificare un sito Web.
2. **Creato con responsive web design in mente** : Bootstrap consente agli sviluppatori web di non preoccuparsi di creare oggetti che si adattino alle dimensioni del loro schermo, dato che Bootstrap utilizza un design reattivo e mobile che consente loro di costruire cose che funzioneranno su qualsiasi schermo taglia.
3. **Semplifica il processo di progettazione** : Bootstrap viene pre-assemblato con elementi che hanno buone pratiche di progettazione. Questo può essere utile per coloro le cui abilità nel web design non sono così grandi, o per coloro che considerano il design un compito noioso, poiché molte delle classi Bootstrap sono esteticamente gradevoli e belle da guardare.

Questa sezione dovrebbe menzionare qualsiasi argomento di grandi dimensioni all'interno di twitter-bootstrap e collegarsi agli argomenti correlati. Poiché la documentazione per twitter-bootstrap è nuova, potrebbe essere necessario creare versioni iniziali di tali argomenti correlati.

Versioni

Versione	Data di rilascio
4.0	2999/01/01
3.3.7	2016/07/25
3.3.6	2015/11/24
3.3.5	2015/06/15

Versione	Data di rilascio
3.3.4	2015/03/16
3.3.1	2014/11/12
3.3.0	2014/10/29
3.2.0	2014/06/26
3.1.0	2014/01/30
3.0	2013/08/19
2.3	2013/02/07
2.2	2012/10/29
2.1	2012/08/20
2.0	2012-02-01
1.0	2011-08-18

Examples

Installazione / Configurazione

Download:

- Scarica Bootstrap [direttamente](#) o clona, ecc. Dal repository [GitHub](#)
- Scarica la versione personalizzata di Bootstrap da [documenti](#) ufficiali
- Installa con bower: `bower install bootstrap`
- Installa con npm: `npm install bootstrap`
- Installa con il compositore: `il composer require twbs/bootstrap`

La struttura del file

```

bootstrap/
|— css/
|   |— bootstrap.css
|   |— bootstrap.min.css
|   |— bootstrap-theme.css
|   |— bootstrap-theme.min.css
|— js/
|   |— bootstrap.js
|   |— bootstrap.min.js
|— fonts/
|   |— glyphicons-halflings-regular.eot
|   |— glyphicons-halflings-regular.svg
|   |— glyphicons-halflings-regular.ttf
|   |— glyphicons-halflings-regular.woff

```

Installazione:

All'interno della tua pagina HTML, includi CSS di Bootstrap, JS e la dipendenza di jQuery (pre versione 3, almeno fino all'ultima versione di Bootstrap). Tieni presente che se prevedi di utilizzare le funzionalità JavaScript di Bootstrap, il tuo riferimento jQuery deve *precedere* il tuo riferimento bootstrap.js all'interno del codice HTML.

È possibile utilizzare i file Bootstrap installati dalla sezione precedente o fare riferimento a un CDN fornito dai creatori di Bootstrap (collegamenti presi da [Guida introduttiva a Bootstrap](#)):

```

<!-- Latest compiled and minified CSS -->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
integrity="sha384-BVYiisSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u"
crossorigin="anonymous">

<!-- Optional theme -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap-
theme.min.css" integrity="sha384-
rHyoN1iRsVXV4nD0JutlnGaslCJuC7uwjduW9SVrLvRYooPp2bWYgmgJQIXw1/Sp" crossorigin="anonymous">

<!-- Latest compiled and minified JavaScript -->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"
integrity="sha384-Tc5IQib027qvyjSMfHjOMaLkfuWVxZxUPnPnCJA71l2mCWNIPg9mGCD8wGNICPD7Txa"
crossorigin="anonymous"></script>

```

Una pagina web Bootstrap molto semplice:

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- The above 3 meta tags *must* come first in the head; any other head content must come
*after* these tags -->

```

```

<title>Bootstrap 101 Template</title>

<!-- Bootstrap -->
<link href="css/bootstrap.min.css" rel="stylesheet">

<!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
<!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
<!--[if lt IE 9]>
  <script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
  <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
<![endif]-->
</head>
<body>
  <h1>Hello, world!</h1>

  <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
  <!-- Include all compiled plugins (below), or include individual files as needed -->
  <script src="js/bootstrap.min.js"></script>
</body>
</html>

```

Modello base

```

<!DOCTYPE html>
<html lang="en">

  <head>

    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- The above 3 meta tags *must* come first in the head; any other head content must
    come *after* these tags -->

    <title>Bootstrap 101 Template</title> <!-- The title of the Website -->

    <!-- Reference to Bootstrap's CSS file -->
    <!-- This is the line to reference the bootstrap's Stylesheet -->
    <link href="css/bootstrap.min.css" rel="stylesheet">

    <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
    <!-- [if lt IE 9] -->
    <script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
    <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
    <!--[endif]-->

  </head>

  <body>

    <h1>Hello, world!</h1>

    <!-- Referencing jQuery (necessary for Bootstrap JavaScript plugins(bootstrap.min.js)

```

```

to work) -->
    <script src =
"https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>

    <!-- Referencing Javascript Bootstrap Plugin to Facilitate Bootstrap Animations and
functionalities. -->
    <!-- (Necessary to run Bootstrap) -->
    <script src="js/bootstrap.min.js"></script>

</body>
</html>

```

Quando usare Bootstrap

Bootstrap è un framework supponente per HTML, CSS e Javascript. Contiene lo stile e le funzionalità di base per elementi che sono diventati elementi [Interfaccia utente] accettati, come elementi del modulo, pulsanti, finestre modali ed elementi di navigazione.

Bootstrap è un framework web reattivo, il che significa che è progettato per adattare layout e design a schermi di dimensioni grandi e piccole, come dispositivi mobili, tablet e computer desktop, il tutto in un unico codice base.

Uno dei concetti fondamentali di Bootstrap è il framework della griglia. Applicando le classi agli elementi HTML, è possibile creare layout complessi utilizzando una griglia di base di dodici colonne. Ad esempio, un layout a quattro colonne potrebbe adattarsi a due colonne su dispositivi tablet e una colonna su dispositivi mobili. La griglia utilizza `media queries`, un metodo CSS per il targeting di dimensioni dello schermo specifiche, per raggiungere questo obiettivo.

Bootstrap si comporta particolarmente bene se:

- Il design personalizzato non è una priorità assoluta
- Sei più a tuo agio nel modificare l'HTML e aggiungere classi piuttosto che creare CSS personalizzati
- Ti senti a tuo agio nell'utilizzare una struttura che presenterà molte somiglianze visive con molti altri siti web

Bootstrap può essere utilizzato da coloro che sono nuovi in HTML, CSS e Javascript, poiché la [documentazione](#) è eccellente. Tuttavia, esiste una curva di apprendimento per coloro che non sono del tutto a proprio agio con le tre tecnologie di base utilizzate da Bootstrap (HTML, CSS e Javascript).

È possibile acquistare o scaricare temi Bootstrap per modificare lo stile o la funzionalità di Bootstrap. È anche possibile utilizzare Bootstrap come punto di partenza, con la personalizzazione di CSS e Javascript.

Pagina Web di base che utilizza componenti di bootstrap

```

<!DOCTYPE html>
<html lang="en">

```

```

<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <!-- The above 3 meta tags *must* come first in the head; any other head content must come
*after* these tags -->
  <title>Bootstrap 101 Template</title>

  <!-- Bootstrap -->
  <link href="css/bootstrap.min.css" rel="stylesheet">

  <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
  <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
  <!--[if lt IE 9]>
    <script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
    <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
  <![endif]-->
</head>
<body>
  <!-- Fixed navbar -->
  <nav class="navbar navbar-default navbar-fixed-top">
    <div class="container">
      <div class="navbar-header">
        <button type="button" class="navbar-toggle collapsed" data-toggle="collapse"
data-target="#navbar" aria-expanded="false" aria-controls="navbar">
          <span class="sr-only">Toggle navigation</span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
        </button>
        <a class="navbar-brand" href="#">Project name</a>
      </div>
      <div id="navbar" class="navbar-collapse collapse">
        <ul class="nav navbar-nav">
          <li class="active"><a href="#">Home</a></li>
          <li><a href="#about">About</a></li>
          <li><a href="#contact">Contact</a></li>
          <li class="dropdown">
            <a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button"
aria-haspopup="true" aria-expanded="false">Dropdown <span class="caret"></span></a>
            <ul class="dropdown-menu">
              <li><a href="#">Action</a></li>
              <li><a href="#">Another action</a></li>
              <li><a href="#">Something else here</a></li>
              <li role="separator" class="divider"></li>
              <li class="dropdown-header">Nav header</li>
              <li><a href="#">Separated link</a></li>
              <li><a href="#">One more separated link</a></li>
            </ul>
          </li>
        </ul>
        <ul class="nav navbar-nav navbar-right">
          <li><a href="../navbar/">Default</a></li>
          <li><a href="../navbar-static-top/">Static top</a></li>
          <li class="active"><a href="#">Fixed top <span class="sr-
only">(current)</span></a></li>
        </ul>
      </div><!--/.nav-collapse -->
    </div>
  </nav>

```

```

<div class="container">

  <div class="jumbotron">
    <h1>Navbar example</h1>
    <p>This example is a quick exercise to illustrate how the default, static and
fixed to top navbar work. It includes the responsive CSS and HTML, so it also adapts to your
viewport and device.</p>
    <p>To see the difference between static and fixed top navbars, just scroll.</p>
    <p>
      <a class="btn btn-lg btn-primary" href="../../components/#navbar"
role="button">View navbar docs &raquo;</a>
    </p>
  </div>

</div> <!-- /container -->

<!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<!-- Include all compiled plugins (below), or include individual files as needed -->
<script src="js/bootstrap.min.js"></script>
</body>
</html>

```

Leggi Iniziare con twitter-bootstrap online: <https://riptutorial.com/it/twitter-bootstrap/topic/818/iniziare-con-twitter-bootstrap>

Capitolo 2: Bootstrap Affix

Examples

Su Navbar

html:

```
<nav class="navbar navbar-default" data-offset-top="120" data-spy="affix" >
  ...
</nav>
```

Css:

```
<style>
.navbar {
  background-color: red;
}
.navbar.affix {
  background-color: green;
}
</style>
```

Affisso Esempio 2

```
<div class="container" id="con">
  <div class="row">
    <div class="span12">
      <div class="well">
        <h1> Header </h1>
      </div>
    </div>
  </div>
</div>
<div class="container" data-spy="affix" data-offset-top="400" id="nav">
  <div class="navbar">
    <div class="navbar-inner">
      <div class="container">
        <div class="span12">
          <a class="brand" href="#">Home</a>
          <ul class="nav">
            <li class="active"><a href="#">Home</a></li>
            <li><a href="#">Link</a></li>
            <li><a href="#">Link</a></li>
          </ul>
        </div>
      </div>
    </div>
  </div>
</div>
<div class="container">
  <div class="span3">
```

```
<p style="height:1000px;padding:10px;">

    Long scrolling text here... Ovi lispmd idr. Blah goo bar foor foo. Ovi lispmd
idr. Blah goo bar foor foo. Ovi lispmd idr. Blah goo bar foor foo. Ovi lispmd idr. Blah goo
bar foor foo. Ovi lispmd idr. Blah goo bar foor foo. Ovi lispmd idr. Blah goo bar foor foo.
Ovi lispmd idr. Blah goo bar foor foo. Ovi lispmd idr. Blah goo bar foor foo. Ovi lispmd idr.
Blah goo bar foor foo. Ovi lispmd idr. Blah goo bar foor foo.

    </p>
</div>
</div>

<style>

#con .well {
    height:400px;
}

#nav.affix {
    position: fixed;
    top: 0;
    width: 100%
}
</style>

<script>
    $('#nav').affix();
</script>
```

Leggi Bootstrap Affix online: <https://riptutorial.com/it/twitter-bootstrap/topic/6639/bootstrap-affix>

Capitolo 3: Bootstrap Navbar

Examples

Bootstrap Navbar

Questo è un esempio di Navbar versione 3 di Bootstrap:

```
<nav class="navbar navbar-default" role="navigation">
<div class="container-fluid">
  <!-- Brand and toggle get grouped for better mobile display -->
  <div class="navbar-header">
    <button type="button" class="navbar-toggle" data-toggle="collapse" data-
target=".navbar-ex1-collapse">
      <span class="sr-only">Toggle navigation</span>
      <span class="icon-bar"></span>
      <span class="icon-bar"></span>
      <span class="icon-bar"></span>
    </button>
    <a class="navbar-brand" href="#">Title</a>
  </div>

  <!-- Collect the nav links, forms, and other content for toggling -->
  <div class="collapse navbar-collapse navbar-ex1-collapse">
    <ul class="nav navbar-nav">
      <li class="active"><a href="#">Link</a></li>
      <li><a href="#">Link</a></li>
    </ul>
    <form class="navbar-form navbar-left" role="search">
      <div class="form-group">
        <input type="text" class="form-control" placeholder="Search">
      </div>
      <button type="submit" class="btn btn-default">Submit</button>
    </form>
    <ul class="nav navbar-nav navbar-right">
      <li><a href="#">Link</a></li>
      <li class="dropdown">
        <a href="#" class="dropdown-toggle" data-toggle="dropdown">Dropdown <b
class="caret"></b></a>
        <ul class="dropdown-menu">
          <li><a href="#">Action</a></li>
          <li><a href="#">Another action</a></li>
          <li><a href="#">Something else here</a></li>
          <li><a href="#">Separated link</a></li>
        </ul>
      </li>
    </ul>
  </div><!-- /.navbar-collapse -->
</div>
```

Immagine di marca di Bootstrap

```
<nav class="navbar navbar-default">
  <div class="container-fluid">
```

```
<div class="navbar-header">
  <a class="navbar-brand" href="#">
    
  </a>
</div>
</div>
</nav>
```

Leggi Bootstrap Navbar online: <https://riptutorial.com/it/twitter-bootstrap/topic/6233/bootstrap-navbar>

Capitolo 4: caroselli

Osservazioni

Per ulteriori informazioni, visitare la documentazione ufficiale all'indirizzo <http://getbootstrap.com/javascript/#carousel> , da cui derivano gli esempi e le informazioni di utilizzo HTML e Javascript di base.

Va notato che i caroselli non funzionano correttamente in IE 9 e precedenti a causa dell'uso di transizioni / animazioni CSS3.

Examples

Utilizzo HTML di base

Un carosello Bootstrap è un componente Bootstrap che crea una presentazione che scorre attraverso gli elementi all'interno del carosello.

Ecco un esempio di utilizzo HTML di base:

```
<div id="carousel-example-generic" class="carousel slide" data-ride="carousel">
  <!-- Indicators -->
  <ol class="carousel-indicators">
    <li data-target="#carousel-example-generic" data-slide-to="0" class="active"></li>
    <li data-target="#carousel-example-generic" data-slide-to="1"></li>
    <li data-target="#carousel-example-generic" data-slide-to="2"></li>
  </ol>

  <!-- Wrapper for slides -->
  <div class="carousel-inner" role="listbox">
    <div class="item active">
      
      <div class="carousel-caption">
        ...
      </div>
    </div>
    <div class="item">
      
      <div class="carousel-caption">
        ...
      </div>
    </div>
    ...
  </div>

  <!-- Controls -->
  <a class="left carousel-control" href="#carousel-example-generic" role="button" data-
slide="prev">
    <span class="glyphicon glyphicon-chevron-left" aria-hidden="true"></span>
    <span class="sr-only">Previous</span>
  </a>
  <a class="right carousel-control" href="#carousel-example-generic" role="button" data-
```

```
slide="next">
  <span class="glyphicon glyphicon-chevron-right" aria-hidden="true"></span>
  <span class="sr-only">Next</span>
</a>
</div>
```

Utilizzo e inizializzazione di Javascript di base

I componenti del carosello possono essere istanziati tramite jQuery con la funzione `$('.carousel').carousel(options)`, dove `$('.carousel')` è un riferimento di livello superiore al carosello specifico e le `options` è un oggetto Javascript che specifica il carosello attributi predefiniti.

L'oggetto `options` consente di definire più proprietà che influiscono sul comportamento del carosello. Queste proprietà sono definite come tali:

- La proprietà `interval` accetta un tipo di `number` Javascript che consente a un utente di definire la quantità di tempo per cui il carosello visualizza una data diapositiva del carosello. Se viene specificato il valore booleano `false`, il carosello non ciclerà automaticamente.
- La proprietà `pause` accetta un tipo di `string` Javascript che attiva o disattiva il comportamento in cui il ciclo automatico del carousel viene sospeso quando il mouse dell'utente entra nel carosello. Il valore predefinito (e solo) accettato è "hover".
- La proprietà `wrap` accetta un tipo `boolean` Javascript che consente a un utente di definire se desidera che il carosello esegua un ciclo continuo senza fermarsi su una determinata diapositiva.
- La proprietà della `keyboard` accetta un tipo `boolean` Javascript che consente a un utente di definire se desidera che il carosello risponda agli eventi della tastiera.

Ecco un esempio dell'utilizzo Javascript di base:

```
$('#carCarousel').carousel({ interval: 2500, pause: "hover", wrap: false, keyboard: true });
```

Come con altri componenti Bootstrap, le opzioni del carousel possono anche essere specificate in HTML tramite attributi di dati.

Leggi caroselli online: <https://riptutorial.com/it/twitter-bootstrap/topic/1568/caroselli>

Capitolo 5: Classi di utilità

Examples

Genera classi .hidden- * per tutti i punti di interruzione - SCSS

```
// Mixin to generate hidden classes
@mixin generate-hidden-classes {
  @each $bp in map-keys($grid-breakpoints) {
    .hidden-#{$bp} {
      @include media-breakpoint-only($bp) {
        display: none !important;
      }
    }
  }
}

// Call to the mixin
@include generate-hidden-classes();
```

Leggi Classi di utilità online: <https://riptutorial.com/it/twitter-bootstrap/topic/6217/classi-di-utilita>

Capitolo 6: colonne

Examples

Colonne reattive stessa altezza (solo CSS o SASS)

Devi aggiungere un div con la classe `.row-height` all'interno della riga e anche aggiungere `.col-height` alle colonne. Se si desidera limitare l'effetto a una determinata query multimediale, è sufficiente utilizzare le `.row-height responsive` `.row-height` e `.col-height` : ad esempio `.row-sm-height` con `.col-sm-height` .

Versione CSS:

```
.row-height {
  display: table;
  table-layout: fixed;
  height: 100%;
  width: calc(100% + 30px);
}
.col-height {
  display: table-cell;
  float: none;
  height: 100%;
}
.col-top {
  vertical-align: top;
}
.col-middle {
  vertical-align: middle;
}
.col-bottom {
  vertical-align: bottom;
}

@media (min-width: 480px) {
  .row-xs-height {
    display: table;
    table-layout: fixed;
    height: 100%;
    width: 100%;
  }
  .col-xs-height {
    display: table-cell;
    float: none;
    height: 100%;
  }
  .col-xs-top {
    vertical-align: top;
  }
  .col-xs-middle {
    vertical-align: middle;
  }
  .col-xs-bottom {
    vertical-align: bottom;
  }
}
```

```

}

@media (min-width: 768px) {
  .row-sm-height {
    display: table;
    table-layout: fixed;
    height: 100%;
    width: 100%;
  }
  .col-sm-height {
    display: table-cell;
    float: none;
    height: 100%;
  }
  .col-sm-top {
    vertical-align: top;
  }
  .col-sm-middle {
    vertical-align: middle;
  }
  .col-sm-bottom {
    vertical-align: bottom;
  }
}

@media (min-width: 992px) {
  .row-md-height {
    display: table;
    table-layout: fixed;
    height: 100%;
    width: calc(100% + 30px);
  }
  .col-md-height {
    display: table-cell;
    float: none;
    height: 100%;
  }
  .col-md-top {
    vertical-align: top;
  }
  .col-md-middle {
    vertical-align: middle;
  }
  .col-md-bottom {
    vertical-align: bottom;
  }
  .row-md-height .col-md-3 {
    width: 25%;
    min-width: 25%;
    max-width: 25%;
  }
}

@media (min-width: 1200px) {
  .row-lg-height {
    display: table;
    table-layout: fixed;
    height: 100%;
    width: 100%;
  }
  .col-lg-height {

```

```

    display: table-cell;
    float: none;
    height: 100%;
  }
  .col-lg-top {
    vertical-align: top;
  }
  .col-lg-middle {
    vertical-align: middle;
  }
  .col-lg-bottom {
    vertical-align: bottom;
  }
}

```

Versione SASS (necessario bootstrap _variables.scss):

```

@import "../bootstrap/variables.scss";
$sizes: xs sm md lg;
$screens: $screen-xs-min $screen-sm-min $screen-md-min $screen-lg-min;

//general
.row-height {
  display: table;
  table-layout: fixed;
  height: 100%;
  width: calc(100% + $grid-gutter-width);
}
.col-height {
  display: table-cell;
  float: none;
  height: 100%;
}
.col-top {
  vertical-align: top;
}
.col-middle {
  vertical-align: middle;
}
.col-bottom {
  vertical-align: bottom;
}

//different sizes
@for $i from 1 through length($sizes) {
  $size: nth($sizes, $i);
  $screen: nth($screens, $i);

  @media (min-width: #{$screen}) {
    .row-#{$size}-height {
      display: table;
      table-layout: fixed;
      height: 100%;
      width: 100%;
    }
    .col-#{$size}-height {
      display: table-cell;
      float: none;
      height: 100%;
    }
  }
}

```

```
.col-#{$size}-top {  
  vertical-align: top;  
}  
.col-#{$size}-middle {  
  vertical-align: middle;  
}  
.col-#{$size}-bottom {  
  vertical-align: bottom;  
}  
}  
  
}
```

Leggi colonne online: <https://riptutorial.com/it/twitter-bootstrap/topic/6469/colonne>

Capitolo 7: Componenti Bootstrap

Osservazioni

Per ulteriori informazioni, visitare la documentazione ufficiale disponibile all'indirizzo <http://getbootstrap.com/javascript/> , da cui deriva l'elenco dei componenti.

Examples

Esempi di componenti Bootstrap

I componenti Bootstrap sono una raccolta di plugin jQuery opzionali che sono in bundle con Bootstrap.

Lo scopo dei componenti Bootstrap è quello di fornire funzionalità e funzionalità estese che sarebbero difficili (o impossibili) da realizzare senza l'uso di Javascript. Alcuni componenti forniscono funzionalità puramente funzionali, mentre alcuni componenti vengono utilizzati per definire la funzionalità di alcuni dei widget front-end speciali di Bootstrap.

Gli esempi includono *effetti di transizione* , *finestre di dialogo modali* , *elenchi a discesa* , *scrollspy* , *schede* , *descrizioni comandi* , *avvisi* , *popover* , *pulsanti* , *compressione* , *caroselli* e *affissi* .

Leggi Componenti Bootstrap online: <https://riptutorial.com/it/twitter-bootstrap/topic/6054/componenti-bootstrap>

Capitolo 8: Contenitori di bootstrap

introduzione

Utilizzare `.container` per un contenitore a larghezza fissa reattiva.

Utilizzare `.container-fluid` per un contenitore a larghezza intera, che copre l'intera larghezza del viewport.

Examples

contenitori

`.container` ha una larghezza fissa per ogni dimensione dello schermo in bootstrap (xs, sm, md, lg);

`.container-fluid` si espande per riempire la larghezza disponibile.

```
@media (min-width: 568px) {
  .container {
    width: 550px;
  }
}
@media (min-width: 992px) {
  .container {
    width: 970px;
  }
}
@media (min-width: 1200px) {
  .container {
    width: 1170px;
  }
}
```

A seconda della larghezza del viewport su cui viene visualizzata la pagina Web, la classe contenitore fornisce al div una larghezza fissa specifica.

Il tuo elemento `.container-fluid`, d'altra parte, verrà ridimensionato costantemente man mano che apporti anche le più piccole modifiche alla larghezza del tuo browser.

Leggi **Contenitori di bootstrap online**: <https://riptutorial.com/it/twitter-bootstrap/topic/10908/contenitori-di-bootstrap>

Capitolo 9: Convalida Bootstrap

Osservazioni

- Questa tecnica di convalida può essere utilizzata solo su input che si trovano all'interno di un modulo.
- Le proprietà devono avere almeno un requisito di convalida per mostrare l'evidenziazione su una convalida `onSubmit()` fallita. I tipi di dati (diversi dalla stringa) hanno un requisito di tipo di dati nascosto, quindi non richiedono un'annotazione esplicita dei dati. Le stringhe non hanno questo, quindi per forzare un controllo di validazione insieme agli altri campi, aggiungi l'annotazione dei dati `[MinLengthAttribute(0)]`.

Examples

Utilizzo di ASP.NET MVC e annotazioni di dati

Aggiungere quanto segue a `Web.config` (nella cartella `Views`), in `<appSettings>` :

```
<add key="ClientValidationEnabled" value="true"/>
<add key="UnobtrusiveJavaScriptEnabled" value="true"/>
```

Aggiungi il bundle `jqueryval` a `BundleConfig.cs`:

```
bundles.Add(new ScriptBundle("~/bundles/jqueryval").Include(
    "~/Scripts/jqueryval/jquery.validate*"));
```

Aggiungi quanto segue a tutte le pagine che necessitano di validazione (o `_Layout.cshml`):

```
<!-- Reference to the jqueryval bundle -->
@Scripts.Render("~/bundles/jqueryval")

<!-- jQuery to apply bootstrap validation classes and glyphs to inputs -->
<script type="text/javascript">
    $.validator.setDefaults({
        highlight: function (element) {
            $(element).closest('.form-group').removeClass('has-success has-
feedback').addClass('has-error has-feedback'); // red highlighting
            $(element).closest('.form-group').find('.form-control-
feedback').removeClass('glyphicon-ok').addClass('glyphicon-remove'); // red cross glyphicon
        },
        unhighlight: function (element) {
            $(element).closest('.form-group').removeClass('has-error has-
feedback').addClass('has-success has-feedback'); // green highlighting
            $(element).closest('.form-group').find('.form-control-
feedback').removeClass('glyphicon-remove').addClass('glyphicon-ok'); // green tick glyphicon
        }
    });
</script>
```

Aggiungi annotazioni di dati ai campi rilevanti nel modello:

```
using System.ComponentModel.DataAnnotations;

[Required(ErrorMessage = "This field is required.")
```

Nella vista, aggiungi quanto segue a ciascun input che deve essere convalidato:

```
<!-- Validation messages -->
<div class="text-danger">@Html.ValidationMessageFor(m => m.SomeField)</div>

<!-- Bootstrap feedback span: -->
<span class="glyphicon form-control-feedback"></span>
```

Aggiungere il seguente all'azione del controller pertinente per aggiungere la convalida sul lato server:

```
if (!ModelState.IsValid)
{
    return View(model);
}
else
{
    // continue with action
}
```

Esempio di input che richiede la convalida

Modello:

```
[Required(ErrorMessage = "This field is required.")
[StringLength(maximumLength: 10, ErrorMessage = "This field must be 10 characters or less.")]
public string SomeRequiredField { get; set; }
```

Vista:

```
<div class="form-group has-feedback">
  <div class="col-md-4">
    @Html.LabelFor(m => m.SomeRequiredField, new { @class = "control-label" })
  </div>
  <div class="col-md-8">
    @Html.TextBoxFor(m => m.SomeRequiredField, new { @class = "form-control" })
    <div class="text-danger">@Html.ValidationMessageFor(m => m.SomeRequiredField)</div>
    <span class="glyphicon form-control-feedback"></span>
  </div>
</div>
```

Opzionale

Aggiungi il seguente jQuery per convalidare gli input su sfocatura, nonché su submit:

```
$('#input').on('blur', function () {  
    $(this).valid();  
});
```

Leggi Convalida Bootstrap online: <https://riptutorial.com/it/twitter-bootstrap/topic/6388/convalida-bootstrap>

Capitolo 10: Dialoghi modali

Osservazioni

Per ulteriori informazioni, visitare la documentazione ufficiale all'indirizzo <http://getbootstrap.com/javascript/#modals> , da cui è stato tratto l'esempio "Utilizzo HTML di base".

Examples

Utilizzo HTML di base

Una finestra di dialogo modale Bootstrap è un componente Bootstrap che crea una finestra di dialogo modale che fluttua sul contenuto a livello di pagina.

Ecco un esempio dell'uso di base di una finestra di dialogo modale Bootstrap in HTML:

```
<div class="modal fade" tabindex="-1" role="dialog">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal" aria-label="Close"><span
aria-hidden="true">&times;</span></button>
        <h4 class="modal-title">Modal title</h4>
      </div>
      <div class="modal-body">
        <p>One fine body&hellip;</p>
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-default" data-dismiss="modal">Close</button>
        <button type="button" class="btn btn-primary">Save changes</button>
      </div>
    </div><!-- /.modal-content -->
  </div><!-- /.modal-dialog -->
</div><!-- /.modal -->
```

Utilizzo e inizializzazione di Javascript di base

Le componenti di dialogo modale possono essere istanziate tramite jQuery con la funzione `$('#myModal').modal(options)` , dove `$('#myModal')` è un riferimento di livello superiore alla finestra di dialogo modale specifica e le `options` sono un oggetto Javascript che specifica gli attributi di default della finestra di dialogo modale.

L'oggetto `options` consente di definire più proprietà che influiranno sul comportamento della finestra di dialogo modale. Queste proprietà sono definite come tali:

- La proprietà `backdrop` consente a un utente di definire se desidera che venga visualizzata una sovrapposizione di sfondo grigio dietro la finestra di dialogo modale. Entrambi i valori booleani e la stringa "statica" sono riconosciuti. Se viene specificato "statico", la finestra di

dialogo modale non verrà chiusa quando un utente fa clic sulla sovrapposizione dello sfondo.

- La proprietà della `keyboard` consente a un utente di definire se desidera che la finestra di dialogo modale venga chiusa quando si preme il tasto Esc sulla tastiera.
- La proprietà `show` consente a un utente di definire se desidera che la finestra di dialogo modale appaia quando viene inizializzata la modal.

Ecco un esempio dell'utilizzo Javascript di base:

```
$('#carModal').modal({ backdrop: false, keyboard: true, show: false });
```

Come con altri componenti Bootstrap, le opzioni del modale possono anche essere specificate in HTML tramite attributi di dati.

Leggi Dialoghi modali online: <https://riptutorial.com/it/twitter-bootstrap/topic/5927/dialoghi-modali>

Capitolo 11: Distintivi e etichette Bootstrap

Examples

badge

I badge sono indicatori numerici di quanti elementi sono associati a un collegamento:

Usa la classe `.badge` all'interno degli elementi `` per creare badge:

```
<a href="#">News <span class="badge">5</span></a><br>
<a href="#">Comments <span class="badge">10</span></a><br>
<a href="#">Updates <span class="badge">2</span></a>
```

Badge in Button

```
<button type="button" class="btn btn-primary">Primary <span class="badge">7</span></button>
```

etichette

Le etichette vengono utilizzate per fornire informazioni aggiuntive su qualcosa:

Utilizzare la classe `.label`, seguita da una delle sei classi contestuali `.label-default`, `.label-primary`, `.label-success`, `.label-info`, `.label-warning` o `.label-danger`, all'interno di un elemento `` per creare un'etichetta:

```
<h1>Example <span class="label label-default">New</span></h1>
<h2>Example <span class="label label-default">New</span></h2>
<h3>Example <span class="label label-default">New</span></h3>
<h4>Example <span class="label label-default">New</span></h4>
<h5>Example <span class="label label-default">New</span></h5>
<h6>Example <span class="label label-default">New</span></h6>
```

Leggi Distintivi e etichette Bootstrap online: <https://riptutorial.com/it/twitter-bootstrap/topic/7867/distintivi-e-etichette-bootstrap>

Capitolo 12: Dropdown di avvio

Parametri

metodi	Esempio
Chiama tramite Javascript	<code>\$('.dropdown-toggle').dropdown();</code>
Attiva / disattiva il menu a discesa	<code>\$('.dropdown-toggle').dropdown('toggle')</code>
Tipo di evento	Descrizione
show.bs.dropdown	Questo evento si attiva immediatamente quando viene chiamato il metodo show instance.
shown.bs.dropdown	Questo evento viene attivato quando il menu a discesa è stato reso visibile all'utente (attenderà le transizioni CSS, per completare).
hide.bs.dropdown	Questo evento viene generato immediatamente quando viene chiamato il metodo di istanza nascosta.
hidden.bs.dropdown	Questo evento viene attivato quando il menu a discesa ha finito di essere nascosto all'utente (attenderà le transizioni CSS, per completare).
Esempio di gestore di eventi	<code>\$(element).on('show.bs.dropdown', function () { // do something.. })</code>

Osservazioni

Quando si chiama Dropdown tramite Javascript `$('.dropdown-toggle').dropdown()`, il data-api cioè `data-toggle="dropdown"` ancora richiesto. [Leggi di più](#)

Examples

Come usare

Usa la classe `.dropdown` sull'elemento parent del menu a discesa.

Aggiungi la classe di menu `.dropdown` a un elemento per inizializzare il plugin del menu a discesa.

Chiama il plugin utilizzando l' `.dropdown-toggle` class `.dropdown-toggle` e `data-toggle="dropdown"` su

un pulsante o un collegamento ipertestuale.

Esempio di base

```
<div class="dropdown">
  <button class="btn btn-primary dropdown-toggle" type="button" data-
toggle="dropdown">Dropdown Example
  <span class="caret"></span></button>
  <ul class="dropdown-menu">
    <li><a href="#">Option One</a></li>
    <li><a href="#">Option two</a></li>
    <li><a href="#">More Options</a></li>
  </ul>
</div>
```

Leggi Dropdown di avvio online: <https://riptutorial.com/it/twitter-bootstrap/topic/6361/dropdown-di-avvio>

Capitolo 13: Elenco gruppo

Osservazioni

Dovresti sapere come usare i [pulsanti di bootstrap](#) e poche informazioni sulle classi [contestuali](#) .

Examples

Esempio di base

```
<ul class="list-group">
  <li class="list-group-item">Cras justo odio</li>
  <li class="list-group-item">Dapibus ac facilisis in</li>
  <li class="list-group-item">Morbi leo risus</li>
  <li class="list-group-item">Porta ac consectetur ac</li>
  <li class="list-group-item">Vestibulum at eros</li>
</ul>
```

badge

```
<ul class="list-group">
  <li class="list-group-item">
    <span class="badge">14</span>
    Cras justo odio
  </li>
</ul>
```

Articoli collegati

```
<div class="list-group">
  <a href="#" class="list-group-item active">
    Cras justo odio
  </a>
  <a href="#" class="list-group-item">Dapibus ac facilisis in</a>
  <a href="#" class="list-group-item">Morbi leo risus</a>
  <a href="#" class="list-group-item">Porta ac consectetur ac</a>
  <a href="#" class="list-group-item">Vestibulum at eros</a>
</div>
```

Elementi del pulsante

```
<div class="list-group">
  <button type="button" class="list-group-item">Cras justo odio</button>
  <button type="button" class="list-group-item">Dapibus ac facilisis in</button>
  <button type="button" class="list-group-item">Morbi leo risus</button>
  <button type="button" class="list-group-item">Porta ac consectetur ac</button>
  <button type="button" class="list-group-item">Vestibulum at eros</button>
</div>
```

Articoli disabilitati

```
<div class="list-group">
  <a href="#" class="list-group-item disabled">
    Cras justo odio
  </a>
  <a href="#" class="list-group-item">Dapibus ac facilisis in</a>
  <a href="#" class="list-group-item">Morbi leo risus</a>
  <a href="#" class="list-group-item">Porta ac consectetur ac</a>
  <a href="#" class="list-group-item">Vestibulum at eros</a>
</div>
```

Classi contestuali

```
<ul class="list-group">
  <li class="list-group-item list-group-item-success">Dapibus ac facilisis in</li>
  <li class="list-group-item list-group-item-info">Cras sit amet nibh libero</li>
  <li class="list-group-item list-group-item-warning">Porta ac consectetur ac</li>
  <li class="list-group-item list-group-item-danger">Vestibulum at eros</li>
</ul>
<div class="list-group">
  <a href="#" class="list-group-item list-group-item-success">Dapibus ac facilisis in</a>
  <a href="#" class="list-group-item list-group-item-info">Cras sit amet nibh libero</a>
  <a href="#" class="list-group-item list-group-item-warning">Porta ac consectetur ac</a>
  <a href="#" class="list-group-item list-group-item-danger">Vestibulum at eros</a>
</div>
```

Contenuto personalizzato

```
<div class="list-group">
  <a href="#" class="list-group-item active">
    <h4 class="list-group-item-heading">List group item heading</h4>
    <p class="list-group-item-text">...</p>
  </a>
</div>
```

Leggi Elenco gruppo online: <https://riptutorial.com/it/twitter-bootstrap/topic/6347/elenco-gruppo>

Capitolo 14: Glyphicons

Osservazioni

Questa sezione fornisce una panoramica sui glyphicon di Bootstrap e descrive come utilizzare i glyphicon.

Examples

Come usare i glyphicons

Twitter Bootstrap supporta icone chiamate glyphicons e possono essere utilizzate con tutti i tag HTML.

Tutte le icone richiedono una classe base e una singola classe di icone.

Tieni presente che le classi di icone non possono essere combinate direttamente con altri componenti, quindi utilizza sempre il `` .

Se il tuo codice HTML ha elementi figlio interni, non puoi utilizzare le classi di icone per quel particolare tag.

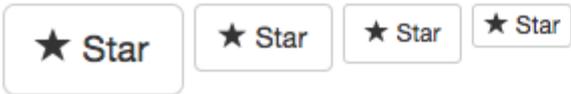
Esempi

Ad esempio, stai creando un pulsante di bootstrap, quindi la sintassi per questo pulsante dovrebbe essere come questa:

```
<button type="button" class="btn btn-default btn-lg">
  Star
</button>
```

Quindi nell'esempio precedente viene creato un semplice pulsante di avvio, ma ora si desidera aggiungere un glyphicon in questo pulsante, per cui è sufficiente aggiungere un elemento `` all'interno di un tag `<button>` . Come questo:

```
<button type="button" class="btn btn-default btn-lg">
  <span class="glyphicon glyphicon-star" aria-hidden="true"></span>Star
</button>
```



```
<button type="button" class="btn btn-default" aria-label="Left Align">
  <span class="glyphicon glyphicon-align-left" aria-hidden="true"></span>
</button>

<button type="button" class="btn btn-default btn-lg">
  <span class="glyphicon glyphicon-star" aria-hidden="true"></span> Star
</button>
```

Leggi Glyphicons online: <https://riptutorial.com/it/twitter-bootstrap/topic/6098/glyphicons>

Capitolo 15: Grid Nesting

introduzione

In Bootstrap è possibile utilizzare le colonne della griglia *all'interno di* altre colonne. Ciò è utile quando si creano layout responsive avanzati che utilizzano [più livelli di griglia](#) .

Osservazioni

Possiamo avere il maggior numero di colonne possibile nel modo sopra indicato.

Examples

Colonne di nidificazione

```
<div class="row">
  <div class="col-sm-9">
    Level 1: .col-sm-9
    <div class="row">
      <div class="col-xs-8 col-sm-6">
        Level 2: .col-xs-8 .col-sm-6
      </div>
      <div class="col-xs-4 col-sm-6">
        Level 2: .col-xs-4 .col-sm-6
      </div>
    </div>
  </div>
</div>
```

L'esempio è tratto da [<http://getbootstrap.com/css/#grid-nesting>] [1]

come suggerisce la documentazione del sito ufficiale

Per annidare il contenuto con la griglia predefinita, aggiungi un nuovo `.row` e un insieme di colonne `.col-sm-*` all'interno di una colonna `.col-sm-*` esistente. Le righe annidate devono includere un set di colonne che ne aggiungono fino a 12 o meno (non è necessario che tu utilizzi tutte le 12 colonne disponibili).

Level 1: .col-sm-9	
Level 2: .col-xs-8 .col-sm-6	Level 2: .col-xs-4 .col-sm-6

Leggi Grid Nesting online: <https://riptutorial.com/it/twitter-bootstrap/topic/9088/grid-nesting>

Capitolo 16: Jumbotron

introduzione

Jumbotron è un componente standard di Bootstrap per visualizzare alcuni contenuti importanti sul tuo sito web. Di solito è usato proprio sotto la barra di navigazione, prima del contenuto.

Osservazioni

All'interno del jumbotron funziona anche tutto il grid system, il container class e il row class.

Examples

Jumbotron di base con due righe di testo e un pulsante

Questo è un jumbotron con un titolo, un contenuto e un pulsante.

Codice

```
<div class="jumbotron">
  <h1>Title text</h1>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec tortor ipsum, convallis sit.</p>
  <p><a class="btn btn-default" href="#" role="button">A button</a></p>
</div>
```

Risultato

Title text

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Donec tortor ipsum, convallis sit.

A button

Leggi Jumbotron online: <https://riptutorial.com/it/twitter-bootstrap/topic/9188/jumbotron>

Capitolo 17: Le forme

Examples

Forma di base

I controlli del modulo hanno uno stile predefinito senza l'utilizzo di classi speciali.

Tuttavia etichette e controlli possono essere racchiusi in tag `.form-group` per una spaziatura ottimale.

```
<form>
  <div class="form-group">
    <label for="input-email">Email address</label>
    <input type="email" class="form-control" id="input-email" placeholder="Email">
  </div>
  <div class="form-group">
    <label for="input-password">Password</label>
    <input type="password" class="form-control" id="input-password" placeholder="Password">
  </div>
  <button type="submit" class="btn btn-default">Submit</button>
</form>
```

Ingressi di sola lettura e disabilitati

Aggiungi l'attributo `readonly` per impedire l'input dell'utente. Un campo di sola lettura non può essere modificato

```
<input class="form-control" type="text" placeholder="Readonly input here..." readonly>
```

Aggiungi l'attributo `disabled` per disabilitare un campo di input. Non è possibile modificare neanche un campo disattivo. Il cursore cambia per renderlo più evidente.

```
<input class="form-control" id="disabledInput" type="text" placeholder="Disabled input here..." disabled>
```

Leggi Le forme online: <https://riptutorial.com/it/twitter-bootstrap/topic/6251/le-forme>

Capitolo 18: menu a discesa

Osservazioni

Per ulteriori informazioni, visitare la documentazione ufficiale di Bootstrap disponibile all'indirizzo <http://getbootstrap.com/javascript/#dropdowns> , da cui deriva l'esempio di utilizzo HTML di base.

Examples

Utilizzo HTML di base

Un menu a discesa Bootstrap è un componente Bootstrap che consente a un elemento HTML di attivare la visualizzazione di un menu a discesa del sottomenu sull'elemento su cui si fa clic.

Ecco un esempio di utilizzo HTML di base:

```
<div class="dropdown">
  <button id="dLabel" type="button" data-toggle="dropdown" aria-haspopup="true" aria-
  expanded="false">
    Dropdown trigger
    <span class="caret"></span>
  </button>
  <ul class="dropdown-menu" aria-labelledby="dLabel">
    ...
  </ul>
</div>
```

Gli elementi del sotto menu a discesa possono essere specificati inserendo `li` elementati all'interno dell'elemento `ul` con la classe `.dropdown-menu` .

Leggi menu a discesa online: <https://riptutorial.com/it/twitter-bootstrap/topic/6033/menu-a-discesa>

Capitolo 19: Menu di navigazione

Examples

Menu pillola orizzontale

```
<ul class="nav nav-pills">
  <li role="presentation" class="active"><a href="#">Home</a></li>
  <li role="presentation"><a href="#">Profile</a></li>
  <li role="presentation"><a href="#">Messages</a></li>
</ul>
```

Menu pillola verticale

```
<ul class="nav nav-pills nav-stacked">
  <li role="presentation" class="active"><a href="#">Home</a></li>
  <li role="presentation"><a href="#">Profile</a></li>
  <li role="presentation"><a href="#">Messages</a></li>
</ul>
```

Pillola orizzontale reattiva a tutta larghezza

```
<ul class="nav nav-tabs nav-justified">
  <li role="presentation" class="active"><a href="#">Home</a></li>
  <li role="presentation"><a href="#">Profile</a></li>
  <li role="presentation"><a href="#">Messages</a></li>
</ul>
```

Leggi Menu di navigazione online: <https://riptutorial.com/it/twitter-bootstrap/topic/6404/menu-di-navigazione>

Capitolo 20: Mettere in guardia

Osservazioni

Vedi di più: <http://getbootstrap.com/components/#alerts>

Examples

Tipi di avviso

A differenza di altri componenti Bootstrap come i [pulsanti](#), gli [avvisi](#) **non** vengono forniti con uno stile `default` o `primary`, poiché sono pensati per avvisare l'utente in un modo specifico.

```
<div class="alert alert-success" role="alert">
  Some action was completed successfully
</div>
<div class="alert alert-info" role="alert">
  Here is some information. Just FYI.
</div>
<div class="alert alert-warning" role="alert">
  Careful! You're about to do something dangerous.
</div>
<div class="alert alert-danger" role="alert">
  An error (or something dangerous) happened!
</div>
```

Avviso di esempio di base

```
<div class="container">
  <h2>Alerts</h2>
  <div class="alert alert-success">
    <strong>Success!</strong>
  </div>
  <div class="alert alert-info">
    <strong>Info!</strong>
  </div>
  <div class="alert alert-warning">
    <strong>Warning!</strong> All foelds are required
  </div>
  <div class="alert alert-danger">
    The username is required and can't be empty
  </div>
</div>
```

Alerts

Success!

Info!

Warning! This alert box could indicate a warning that might need attention.

The username is required and can't be empty

Avvisi animati

Le classi `.fade` e `.in` aggiungono un effetto di dissolvenza quando si chiude il messaggio di avviso.

```
<div class="alert alert-success fade in">
  <a href="#" class="close" data-dismiss="alert" aria-label="close">&times;</a>
  <strong>Success!</strong> This is a good example!
</div>
```

Avvisi non ammissibili

Per dare una funzionalità di close alert, tutto ciò di cui abbiamo bisogno è aggiungere `data-dismiss="alert"` al nostro pulsante di chiusura.

```
<div class="alert alert-info alert-dismissible" role="alert">
  <button type="button" class="close" data-dismiss="alert" aria-label="Close">
    <span aria-hidden="true">&times;</span>
  </button>
  Sphinx of black quartz, judge my vow
</div>
```

Sphinx of black quartz, judge my vow



`.alert-dismissible` e `.close` classes sono facoltative, utili solo per lo stile.

Collega il colore negli avvisi

Per fornire rapidamente un colore corrispondente per i collegamenti all'interno di qualsiasi avviso,

possiamo utilizzare la classe di utilità `.alert-link`.

```
<div class="alert alert-success">
  You have won! Click <a href="#" class="alert-link">here</a> to claim your prize ...
</div>

<div class="alert alert-info">
  You might want to check <a href="#" class="alert-link">this</a> instead.
</div>

<div class="alert alert-warning">
  You are running out of coins. Buy more <a href="#" class="alert-link">here</a>.
</div>

<div class="alert alert-danger">
  Something went wrong. You can try <a href="#" class="alert-link">again</a> or ...
</div>
```

Congratulations! You have **WON!** Click **here** to claim your prize ...

You might want to check **this** instead.

You are running out of coins. Buy more **here**.

Something went wrong. You can try **again** or ...

Leggi Mettere in guardia online: <https://riptutorial.com/it/twitter-bootstrap/topic/6434/mettere-in-guardia>

Capitolo 21: Migrazione a Bootstrap 4

introduzione

Bootstrap 4 è una [riscrittura importante](#) e ci sono molte modifiche da tenere presente quando si esegue l'aggiornamento da Bootstrap 3. Ecco le modifiche al nome della classe, i suggerimenti e gli esempi di migrazione del codice Bootstrap 3.x in Bootstrap 4.x.

Osservazioni

Questo è solo un piccolo esempio più dettagliato da seguire.

Examples

Cambiamenti nel layout di colonne del sistema di griglia in Bootstrap 4

Il primo blocco di codice è scritto in Bootstrap 3. In Bootstrap 3 ci sono 4 tipi di specifiche di colonna, in particolare `col-md-*` `col-lg-*` `col-sm-*` `col-xs-*`. Un layout completamente reattivo sarà simile a questo in Bootstrap 3:

```
<div class="row">
  <div class="col-lg-4 col-md-8 col-sm-8 col-xs-8">
    contents
  </div>
  <div class="col-lg-4 col-md-4 col-sm-4 col-xs-4">
    contents
  </div>
</div>
```

In Bootstrap 4, hanno aggiunto un nuovo livello di griglia `sm` sotto `768px` per un controllo più granulare. Quindi Bootstrap 4 ha `col-*` (`xs`), `col-sm-*`, `col-md-*`, `col-lg-*`, and `col-xl-*`. Quindi quello che era `.col-md-6` in `v3` ora è `.col-lg-6` in `v4`. Si noti che il `-xs` è stato rimosso in modo che `.col-6` rappresenti 6 unità di colonna nel punto di interruzione extra piccolo (predefinito).

Quindi, se ora vogliamo scrivere lo stesso esempio sopra in Bootstrap 4, sarebbe simile a questo:

```
<div class="row">
  <div class="col-xl-8 col-lg-8 col-md-8 col-sm-8 col-8">
    contents
  </div>
  <div class="col-xl-8 col-lg-8 col-md-4 col-sm-4 col-4">
    contents
  </div>
</div>
```

Layout di griglia Bootstrap 4

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	Extra large ≥1200px
Grid behavior	Horizontal at all times	Collapsed to start, horizontal above breakpoints			
Max container width	None (auto)	540px	720px	960px	1140px
Class prefix	<code>.col-</code>	<code>.col-sm-</code>	<code>.col-md-</code>	<code>.col-lg-</code>	<code>.col-xl-</code>
# of columns	12				
Gutter width	30px (15px on each side of a column)				
Nestable	Yes				
Offsets	Yes				
Column ordering	Yes				

Modifiche al supporto del browser

In twitter-bootstrap 4 il supporto per IE8, IE9, and iOS 6 è stato eliminato. v4 ora è solo IE10+ and iOS 7+ . Per i siti che necessitano di uno di questi, utilizzare v3.

In twitter-bootstrap 4 è stato aggiunto il supporto ufficiale per Android v5.0 Lollipop's Browser and WebView . Le versioni precedenti di Android Browser e WebView rimangono unofficially supported solo in modo unofficially supported .

Applica la rimozione della classe

Affix viene rimosso da Bootstrap 4.

Si consiglia di utilizzare una `position: sticky polyfill` `position: sticky` invece.

Se si utilizza Affix per applicare stili aggiuntivi non di posizione, i polyfill potrebbero non supportare il caso d'uso. Un'opzione per tali usi è la libreria ScrollPos-Styler di terze parti.

[Secondo la documentazione di Bootstrap](#)

Dropped il plugin jQuery di Affix. Raccomandiamo di usare una posizione: polyfill appiccicoso. Vedi la voce HTML5 per i dettagli e i consigli specifici sul polyfill.

Se si utilizza Affix per applicare stili aggiuntivi non di posizione, i polyfill potrebbero non supportare il caso d'uso. Un'opzione per tali usi è la libreria ScrollPos-Styler di terze

parti.

Se qualcuno sta migrando da Bootstrap v3 a Bootstrap v4 l'approccio fallback è indicato di seguito:

HTML

```
<header>

</header>
<nav class="navbar navbar-light bg-faded" data-toggle="affix">
  <button class="navbar-toggler hidden-sm-up pull-xs-right" type="button" data-
toggle="collapse" data-target="#collapsingNavbar">
    ≡
  </button>
  <a class="navbar-brand" href="#">Brand</a>
  <div class="collapse navbar-toggleable-xs" id="collapsingNavbar">

    <ul class="nav navbar-nav pull-xs-right">
      <li class="nav-item dropdown">
        <a class="nav-link dropdown-toggle" data-toggle="dropdown" href="#"
role="button" aria-haspopup="true" aria-expanded="false">
          Menu
        </a>
        <div class="dropdown-menu" aria-labelledby="Preview">
          <a class="dropdown-item" href="#">Logout</a>

        </div>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Link</a>
      </li>
      <li class="nav-item ">
        <a class="nav-link" href="#">Link</a>
      </li>
      <li class="nav-item ">
        <a class="nav-link" href="#">Link</a>
      </li>
    </ul>
  </div>
</nav>

<div class="container" id="main">
  <h2>Hello Bootstrap 4.</h2>
  <div class="row">
    <div class="col-xs-12 col-sm-6 col-md-9">
      <p>3 wolf moon retro jean shorts chambray sustainable roof party. Shoreditch vegan
artisan Helvetica. Tattooed Codeply Echo Park Godard kogi, next level irony ennui twee squid
fap selvage. Meggings flannel Brooklyn literally small batch, mumblecore
PBR try-hard kale chips. Brooklyn vinyl lumbersexual bicycle rights, viral fap
cronut leggings squid chillwave pickled gentrify mustache. 3 wolf moon hashtag church-key Odd
Future. Austin messenger bag normcore, Helvetica Williamsburg
sartorial tote bag distillery Portland before they sold out gastropub
taxidermy Vice.</p>
    </div>
    <div class="col-xs-6 col-md-3">
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis pharetra codeply
varius quam sit amet vulputate. Quisque mauris augue, molestie tincidunt codeply condimentum
vitae, gravida a libero. Aenean sit amet felis dolor, in sagittis nisi.
Sed ac orci quis tortor imperdiet venenatis. Duis elementum auctor accumsan.
Aliquam in felis sit amet augue.</p>
  </div>
</div>
</div>
```

```

        </p>
        <hr>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis pharetra codeply
varius quam sit amet vulputate. Quisque mauris augue, molestie tincidunt codeply condimentum
vitae, gravida a libero. Aenean sit amet felis dolor, in sagittis nisi.
        Sed ac orci quis tortor imperdiet venenatis. Duis elementum auctor accumsan.
Aliquam in felis sit amet augue.
        </p>
    </div>
</div>
<div class="row">
    <div class="col-xs-6 col-sm-4">
        <div class="card card-outline-primary">
            <div class="card-block">
                <h3 class="card-title">Card</h3>
                <p class="card-text">With supporting text below as a natural lead-in to
additional content.</p>
                <a href="#" class="btn btn-outline-secondary">Outline</a>
            </div>
        </div>
    </div>
    <div class="col-xs-6 col-sm-4">
        <div class="card card-outline-primary">
            <div class="card-block">
                <h3 class="card-title">Card</h3>
                <p class="card-text">With supporting text below as a natural lead-in to
additional content.</p>
                <a href="#" class="btn btn-outline-secondary">Outline</a>
            </div>
        </div>
    </div>
    <div class="col-xs-6 col-sm-4">
        <div class="card card-outline-primary">
            <div class="card-block">
                <h3 class="card-title">Card</h3>
                <p class="card-text">With supporting text below as a natural lead-in to
additional content.</p>
                <a href="#" class="btn btn-outline-secondary">Outline</a>
            </div>
        </div>
    </div>
</div>
</div>

```

CSS

```

header {
height: 220px;
background: #ccc;
}

```

JAVASCRIPT

```

$(document).ready(function() {

    var toggleAffix = function(affixElement, scrollElement, wrapper) {

        var height = affixElement.outerHeight(),
            top = wrapper.offset().top;
    }
}

```

```

if (scrollElement.scrollTop() >= top){
    wrapper.height(height);
    affixElement.addClass("affix");
}
else {
    affixElement.removeClass("affix");
    wrapper.height('auto');
}

});

$(' [data-toggle="affix"] ').each(function() {
    var ele = $(this),
        wrapper = $('<div></div>');

    ele.before(wrapper);
    $(window).on('scroll resize', function() {
        toggleAffix(ele, $(this), wrapper);
    });

    // init
    toggleAffix(ele, $(window), wrapper);
});

});

```

Bootstrap 4 Navbar

Il nuovo componente di barra di avvio di Bootstrap 4 è stato migliorato rispetto al suo predecessore Bootstrap 3.x. In Bootstrap 4, Navbar è **reattivo per impostazione predefinita** e utilizza la tecnologia **Flexbox** per semplificare l'allineamento dei contenuti di Navbar. È anche semplice utilizzare le nuove `navbar-togglerable-*` per modificare il punto di interruzione della Navbar. Ora il Navbar ha 6 dimensioni di breakpoint o "stati" in modo da poter facilmente avere una delle seguenti opzioni di Navbar.

- Il Navbar *non* collassa *mai* nella vista mobile verticale ed è sempre orizzontale.
- La barra di navigazione è *sempre* collassata nella vista verticale e attivata tramite l'hamburger.
- Il Navbar collassa in visualizzazione verticale in *uno dei 4 punti di interruzione reattivi* .

Basic Bootstrap 4 Navbar

```

<nav class="navbar navbar-togglerable-md">
  <button class="navbar-toggler navbar-toggler-right" type="button" data-toggle="collapse"
  data-target="#navbar1">
    <span class="navbar-toggler-icon"></span>
  </button>
  <a class="navbar-brand" href="#">Navbar</a>
  <div class="collapse navbar-collapse" id="navbar1">
    <ul class="navbar-nav mr-auto">
      <li class="nav-item active">
        <a class="nav-link" href="#">Home</a>
      </li>
      <li class="nav-item">

```

```

    <a class="nav-link" href="#">Link</a>
  </li>
</ul>
</div>
</nav>

```

Come puoi vedere dal codice sopra la classe di `navbar-header` è stata rimossa da Bootstrap 4 e il `container-fluid` non è più necessario per un Navbar a larghezza piena.

Modifica del punto di interruzione della navbar

La classe `navbar-togglerable-md` fa collassare verticalmente la barra di navigazione sopra (e mostra l'icona di toggler) al punto di interruzione medio (md) di 992px. Per cambiare questo in un punto di interruzione diverso, dovremmo solo scambiare `navbar-togglerable-md` con uno di questi ..

- `navbar-togglerable` = compressione su larghezza `xs` <576 px
- `navbar-togglerable-sm` = compressione al `sm` larghezze <768px
- `navbar-togglerable-lg` = compressione su larghezza `lg` <1200px

Demo di Navbar Breakpoint 4 Breakpoint

Modifica dell'allineamento della barra di navigazione

Flexbox ci consente di modificare facilmente l'allineamento di Navbar e dei suoi contenuti (marchio, link, moduli o testo). Il contenuto predefinito di Navbar è allineato a sinistra. Naturalmente ci sono molti altri scenari di allineamento ...

- Marchio rimasto (predefinito), centro dei collegamenti e destra
- Centro marchi, collegamenti a sinistra ea destra
- Marchio a sinistra e collegamenti a destra
- Input di marca, link e larghezza di riempimento
- Nessuna marca, centro dei collegamenti e destra
- Marchio rimasto, collegamenti direttamente all'interno del contenitore
- Collegamenti giustificati (larghezza di riempimento) centrati

Bootstrap 4 barra di navigazione con marchio centrato e collegamenti sinistra / destra

```

<nav class="navbar navbar-togglerable-sm navbar-inverse bg-primary">
  <button class="navbar-toggler navbar-toggler-right" type="button" data-toggle="collapse"
  data-target=".dual-collapse">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="navbar-collapse collapse dual-collapse">
    <ul class="navbar-nav mr-auto">
      <li class="nav-item active">
        <a class="nav-link" href="#">Link</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#features">Link</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Link</a>
      </li>
    </ul>
  </div>
</nav>

```

```

        </li>
        <li class="nav-item">
            <a class="nav-link" href="#">Link</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href="#">Link</a>
        </li>
    </ul>
</div>
<a class="navbar-brand d-flex mx-auto" href="#">Navbar 2</a>
<div class="navbar-collapse collapse dual-collapse">
    <ul class="navbar-nav ml-auto">
        <li class="nav-item">
            <a class="nav-link" href="#">Link</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href="#">Link</a>
        </li>
    </ul>
</div>
</nav>

```

Bootstrap 4 Navbar con Marchio sinistro, Centro collegamenti e Destra

```

<nav class="navbar navbar-light navbar-togglerable-sm bg-faded justify-content-center">
    <button class="navbar-toggler navbar-toggler-right" type="button" data-toggle="collapse"
data-target="#collapsingNavbar3">
        <span class="navbar-toggler-icon"></span>
    </button>
    <a href="/" class="navbar-brand d-flex w-50 mr-auto">Brand</a>
    <div class="navbar-collapse collapse" id="collapsingNavbar3">
        <ul class="navbar-nav mx-auto w-100 justify-content-center">
            <li class="nav-item active">
                <a class="nav-link" href="#">Link</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="#">Link</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="#">Link</a>
            </li>
        </ul>
        <ul class="nav navbar-nav ml-auto w-100 justify-content-end">
            <li class="nav-item">
                <a class="nav-link" href="#">Right</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="#">Right</a>
            </li>
        </ul>
    </div>
</nav>

```

Demo di allineamento Navbar: <http://www.codeply.com/go/qhaBrcWp3v>

Altro su Bootstrap 4 Navbar

[Personalizzazione del colore, allineamento o altezza](#)

Bootstrap 3 to Bootstrap 4 Modifiche CSS

Poiché Bootstrap 4 è una riscrittura **importante**, molti dei nomi delle classi di Bootstrap 3.x sono stati modificati o rimossi. La ristrutturazione di componenti come il Navbar e l'introduzione di nuove classi CSS e supporto Flexbox significa che l'aggiornamento a 4.x *non* è un semplice processo di conversione da 3.x.

Tuttavia, esistono alcune classi CSS Bootstrap 3.x con una sostituzione Bootstrap 4 specifica.

Variazioni nome / classe selettore CSS da Bootstrap 3.3.7 a 4 (alpha 6)

{t} - rappresenta un **livello** o punto di interruzione (es .: sm, md, lg, ecc.).

xs tier è l'impostazione predefinita e non è necessario specificare: `col-3`, `col-6`, ecc.

{u} - rappresenta una dimensione **unità di misura** (es .: 1-12)

Bootstrap 3.x	Bootstrap 4
.col- {t} - {u}	.col- {t} - {u} (leave {t} blank for xs)
.col- {t} -offset- {u}	.offset- {t} - {u} (leave {t} blank for xs)
.col- {t} -push- {u}	.push- {t} - {u} (leave {t} blank for xs)
.col- {t} -pull- {u}	.pull- {t} - {u} (leave {t} blank for xs)
.pannello	.carta
.panel-rubrica	.card-header
.panel-titolo	.card-titolo
.panel-corpo	.card-block
.panel-footer	.card-footer
.panel-primaria	.card-primary.card-inversa
.panel-successo	.card-success.card-inversa
.panel-info	.card-info.card-inversa
.panel-warning	.card-warning.card-inversa
.panel-pericolo	.card-danger.card-inversa
.bene	.card.card-block
.thumbnail	.card.card-block
.list-inline> li	.list-inline-voce

Bootstrap 3.x	Bootstrap 4
.dropdown-menu> li	.dropdown-item
.nav navbar> li	.nav-item
.nav navbar> li> a	.nav-link
.navbar-destra	.ML-auto
.navbar-btn	.nav-item
.navbar-top fisso	.fixed-top
.nav-stacked	.flex colonne
.btn-default	.btn secondaria
.img-reattiva	.img-fluido
.img-cerchio	.rounded-cerchio
.sono in castigo	.rounded
.Form-orizzontale	(rimosso)
.Radio	.Form-check
.checkbox	.Form-check
.input-lg	.Form-control-lg
.input-sm	.Form-control-sm
Il controllo-label	.Form-control-label
.table-condensato	.table-sm
.paginazione> li	.pagina-item
.pagination> li> a	.pagina-link
.articolo	.carousel-item
.text-aiuto	.Form-control-feedback
.Estrarre-destra	.float-destra
.Estrarre-sinistra	.float-sinistra
.center-block	.MX-auto

Bootstrap 3.x	Bootstrap 4
.collapse.in	.collapse.show
.hidden-sm	.hidden-md-down
.hidden-md	.hidden-lg-down
.hidden-XS	.hidden-XS-down
.Visible-XS	.hidden-sm-up
.Visible-sm	.hidden-XS-down.hidden-md-up
.Visible-md	.hidden-sm-down.hidden-lg-up
.Visible-lg	.hidden-md-down.hidden-xl-up
.etichetta	.distintivo
.distintivo	.badge.badge-pill

Vedi anche:

[Bootstrap 3.x to 4 Strumento di migrazione](#)

[Novità di Bootstrap 4](#)

Bootstrap 4 Vertical Align

Ottenere elementi al **centro** o in **basso** allineati verticalmente è sempre stata una sfida con CSS e Bootstrap. L'allineamento verticale desiderato può essere all'interno di un contenitore genitore, o relativo ad elementi adiacenti.

Ora che Bootstrap 4 è **flexbox**, **di default** ci sono molti approcci diversi per l'allineamento verticale usando: [margini automatici](#) , [utilità Flexbox](#) o le [utilità di visualizzazione](#) insieme alle [utilità di allineamento verticale](#).

All'inizio, le [utilità di allineamento verticale](#) sembrerebbero una scelta ovvia, ma funzionano *solo* con elementi di visualizzazione inline e da tavolo. Ecco alcune opzioni e scenari di allineamento verticale di Bootstrap 4 ...

1 - Centro verticale che utilizza i margini automatici:

Un modo per centrare verticalmente è usare `my-auto` . Questo centererà l'elemento all'interno del suo contenitore. Ad esempio, `h-100` rende la riga a tutta altezza e `my-auto` centererà verticalmente la `col-sm-12` .

```
<div class="row h-100">
  <div class="col-sm-12 my-auto">
    <div class="card card-block w-25">Card</div>
  </div>
</div>
```

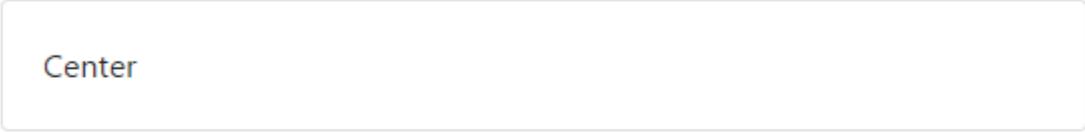
```
</div>
</div>
```

Centro verticale che utilizza la [demo dei margini automatici](#)

`my-auto` rappresenta i margini sull'asse y verticale ed è equivalente a:

```
margin-top: auto;
margin-bottom: auto;
```

2 - Centro verticale con Flexbox:



Center

Taller

With supporting text

Outline

Dal momento che Bootstrap 4 `.row` è ora `display:flex` puoi semplicemente usare l' `align-self-center` su qualsiasi colonna per `.row` verticalmente ...

```
<div class="row">
  <div class="col-6 align-self-center">
    <div class="card card-block">
      Center
    </div>
  </div>
  <div class="col-6">
    <div class="card card-inverse card-danger">
      Taller
    </div>
  </div>
</div>
```

oppure, usa `align-items-center` sull'intero `.row` per centrare verticalmente allineare tutti i `col-*` nella riga ...

```
<div class="row align-items-center">
  <div class="col-6">
    <div class="card card-block">
      Center
    </div>
  </div>
  <div class="col-6">
    <div class="card card-inverse card-danger">
      Taller
    </div>
  </div>
</div>
```

```
        </div>
    </div>
</div>
```

Demo di colonne verticali con altezza differente

3 - Centro verticale che utilizza i visualizzatori:

Bootstrap 4 ha [utilità di visualizzazione](#) che possono essere utilizzate per la `display:table` , `display:table-cell` , `display:inline` , ecc. Questi possono essere usati con l' [allineamento verticale](#) [util](#) per allineare elementi inline, inline-block o table cell.

```
<div class="row h-50">
  <div class="col-sm-12 h-100 d-table">
    <div class="card card-block d-table-cell align-middle">
      I am centered vertically
    </div>
  </div>
</div>
```

Centro verticale che utilizza la [demo di Display Utils](#)

Bootstrap 4 Centering

Come centrare un elemento, una colonna o un contenuto all'interno di una colonna funziona diversamente in Bootstrap 4.

Centro orizzontale

- `text-center` è ancora utilizzato per la `display:inline` elementi in `display:inline`
- `mx-auto` **sostituisce** `center-block` al centro `display:block` elementi
- `offset-* 0 mx-auto` può essere utilizzato per centrare le colonne della griglia

`mx-auto` (margini dell'asse x automatico) centererà il `display:block` o `display:flex` elementi `display:flex` che hanno una *larghezza definita* , (% , vw , px , ecc.). **Flexbox è usato di default** sulle colonne della griglia, quindi ci sono anche vari metodi di centraggio del flexbox.

Centro testo o elementi in linea: `text-center`

```
<div class="container">
  <h1 class="text-center">i'm centered</h1>
  <div class="row">
    <div class="col text-center">i'm centered!</div>
  </div>
</div>
```

`display:block` **centrale** `display:block` **o** `display:flex` : `mx-auto`

```
<div class="row">
```

```
<div class="col-12">
  
</div>
</div>
```

Centra le colonne usando gli offset: `offset-*`

```
<div class="row">
  <div class="col-4 offset-4">
    <h6>I'm .col-4 centered (offset 4)
  </div>
</div>
```

Le colonne possono *anche* essere centrate con: `mx-auto`

```
<div class="row">
  <div class="col-4 mx-auto">
    <h6>I'm .col-4 centered</h6>
  </div>
</div>
```

[Demo Bootstrap 4 Centering orizzontale](#)

Centro verticale

Per il centraggio verticale in Bootstrap 4 (asse y), consultare i documenti su: [Bootstrap 4 Vertical Align](#)

Bootstrap 4 Column Order

Cambiare l'ordine (o posizione) era possibile in Bootstrap 3 usando le classi `push pull`. In Bootstrap 4, le classi **push pull** funzionano ancora e si può utilizzare anche l' **ordine flexbox** .

In Bootstrap 4, le classi **push pull** ora sono `push-{viewport}-{units}` e `pull-{viewport}-{units}` e `xs-infix` è stato rimosso. Considera questo esempio che modifica l'ordine delle colonne nel layout 1-3-2 su `xs` e `sm` :

```
<div class="row">
  <div class="col-3 col-md-6">
    1
  </div>
  <div class="col-3 col-md-6 push-6 push-md-0">
    2
  </div>
  <div class="col-6 col-md-12 pull-3 pull-md-0">
    3
  </div>
</div>
```

[Bootstrap 4 Push Pull Demo](#)

Poiché la nuova versione 4 è flexbox, **un'altra opzione** è quella di utilizzare le classi **dell'utility flexbox** per modificare l'ordine delle colonne. Ora a piena larghezza, 12 colonne `col-*-12` possono essere invertite utilizzando l'ordine **flexbox** .

```
<div class="row">
  <div class="col-md-12">
    Col 1
  </div>
  <div class="col-md-12 flex-first flex-md-unordered">
    Col 2
  </div>
</div>
```

Flexbox Ordering Demo

Leggi Migrazione a Bootstrap 4 online: <https://riptutorial.com/it/twitter-bootstrap/topic/9090/migrazione-a-bootstrap-4>

Capitolo 22: Modals

Osservazioni

I modali richiedono bootstrap.min.js per funzionare correttamente.

Maggiori dettagli possono essere trovati qui: <http://getbootstrap.com/javascript/#modals>

Examples

Modale HTML di base

Una modale è una finestra di dialogo che può essere visualizzata sulla pagina corrente.

```
<!-- Clicking the button will open the modal window -->
<button type="button" class="btn btn-success btn-lg" data-toggle="modal" data-
target="#theModal">Open The Modal</button>

<!-- The Modal -->
<div id="theModal" class="modal fade" role="dialog">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal">&times;</button>
        <h4 class="modal-title">Text For The Modal Header</h4>
      </div>
      <div class="modal-body">
        <p>Text for The Modal Body.</p>
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-default" data-
dismiss="modal">Close</button>
      </div>
    </div>
  </div>
</div>
</div>
```

Leggi Modals online: <https://riptutorial.com/it/twitter-bootstrap/topic/6320/modals>

Capitolo 23: Nav

Examples

Bootstrap Navs

I Nav disponibili in Bootstrap hanno markup condiviso, a partire dalla classe `.nav` base, oltre agli stati condivisi. Scambia le classi di modifica per passare da uno stile all'altro.

Tabs

```
<ul class="nav nav-tabs">
  <li role="presentation" class="active"><a href="#">Home</a></li>
  <li role="presentation"><a href="#">Profile</a></li>
  <li role="presentation"><a href="#">Messages</a></li>
</ul>
```

pillole

```
<ul class="nav nav-pills">
  <li role="presentation" class="active"><a href="#">Home</a></li>
  <li role="presentation"><a href="#">Profile</a></li>
  <li role="presentation"><a href="#">Messages</a></li>
</ul>
```

giustificato

```
<ul class="nav nav-tabs nav-justified">
  ...
</ul>
<ul class="nav nav-pills nav-justified">
  ...
</ul>
```

Con Dropdowns

```
<ul class="nav nav-tabs">
  <li role="presentation" class="dropdown">
    <a class="dropdown-toggle" data-toggle="dropdown" href="#" role="button" aria-
haspopup="true" aria-expanded="false">
      Dropdown <span class="caret"></span>
    </a>
    <ul class="dropdown-menu">
      ...
    </ul>
  </li>
</ul>
```

Leggi Nav online: <https://riptutorial.com/it/twitter-bootstrap/topic/6505/nav>

Capitolo 24: Navbar

Examples

Barra di navigazione di base (fissa nella parte superiore della pagina)

```
<div class="navbar navbar-inverse navbar-fixed-top">
  <div class="container">
    <div class="navbar-header">
      <!-- vvv Hamburger icon that gets shown when window reaches a certain scale vvv -->
      <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">
        <span class="sr-only">Toggle navigation</span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button>
      <!-- ^^v Hamburger icon that gets shown when window reaches a certain scale ^^v -->
      <a class="navbar-brand" href="#">WebSite Title</a>
    </div>
    <div class="navbar-collapse collapse">
      <ul class="nav navbar-nav">
        <li><a href="#">Home</a></li>
        <li><a href="#">About</a></li>
        <li><a asp-controller="Home" asp-action="Contact">Contact</a></li>
      </ul>
    </div>
  </div>
</div>
```

Sottomenu in navbar

```
<div class="navbar navbar-inverse navbar-fixed-top">
  <div class="container">
    <div class="navbar-header">
      <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">
        <span class="sr-only">Toggle navigation</span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button>
      <a class="navbar-brand" href="#">WebSite Title</a>
    </div>
    <div class="navbar-collapse collapse">
      <ul class="nav navbar-nav">
        <li><a href="#">Home</a></li>
        <li><a href="#">About>About</a></li>
        <li><a asp-controller="Home" asp-action="Contact">Contact</a></li>
        <!-- vvv Create a submenu in the navbar vvv --->
        <li class="dropdown">
          <a href="#" class="dropdown-toggle" data-toggle="dropdown">Testing Stuff
          <b class="caret"></b></a>
```

```

        <ul class="dropdown-menu">
            <li><a href="#">SubItem</a></li>
            <li><a href="#">Something Sub-y</a></li>
        </ul>
    </li>
    <!-- ^^ Create a submenu in the navbar ^^ -->
</ul>
</div>
</div>
</div>

```

Separatore navbar

```

<div class="navbar navbar-inverse navbar-fixed-top">
  <div class="container">
    <div class="navbar-header">
      <button type="button" class="navbar-toggle" data-toggle="collapse" data-
target=".navbar-collapse">
        <span class="sr-only">Toggle navigation</span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button>
      <a class="navbar-brand" href="#">WebSite Title</a>
    </div>
    <div class="navbar-collapse collapse">
      <ul class="nav navbar-nav">
        <li><a href="#">Home</a></li>
        <!-- vvv Create a divider in the nav vvv -->
        <li class="divider"></li>
        <!-- ^^ Create a divider in the nav ^^ -->
        <li><a href="#">About</a></li>
        <li><a asp-controller="Home" asp-action="Contact">Contact</a></li>
      </ul>
    </div>
  </div>
</div>

```

Mantieni il link di navigazione corrente "attivo"

```

// Add active class to active navigation link
$(document).ready(function () {
  $('ul.nav.navbar-nav').find('a[href="' + location.pathname + '"]')
    .closest('li').addClass('active');
});

```

Cambia punto di interruzione Navbar (mobile vs normale)

max-width è il punto di interruzione

```

@media (max-width: 1200px) {
  .navbar-header {
    float: none;
  }
  .navbar-left, .navbar-right {

```

```

        float: none !important;
    }
    .navbar-toggle {
        display: block;
    }
    .navbar-collapse {
        border-top: 1px solid transparent;
        box-shadow: inset 0 1px 0 rgba(255,255,255,0.1);
    }
    .navbar-fixed-top {
        top: 0;
        border-width: 0 0 1px;
    }
    .navbar-collapse.collapse {
        display: none!important;
    }
    .navbar-nav {
        float: none!important;
        margin-top: 7.5px;
    }
    .navbar-nav>li {
        float: none;
    }
    .navbar-nav>li>a {
        padding-top: 10px;
        padding-bottom: 10px;
    }
    .collapse.in{
        display:block !important;
    }
    .navbar-nav .open .dropdown-menu {
        position: static;
        float: none;
        width: auto;
        margin-top: 0;
        background-color: transparent;
        border: 0;
        -webkit-box-shadow: none;
        box-shadow: none;
    }
}
}

```

Chiudi la barra di navigazione compressa quando fai clic all'esterno della barra di navigazione

```

jQuery('body').bind('click', function(e) {
    if(jQuery(e.target).closest('#navbar').length == 0) {
        // click happened outside of .navbar, so hide
        var opened = jQuery('.navbar-collapse').hasClass('collapse in');
        if ( opened === true ) {
            jQuery('#navbar2 .navbar-collapse').collapse('hide');
        }
    }
});

```

Leggi Navbar online: <https://riptutorial.com/it/twitter-bootstrap/topic/2267/navbar>

Capitolo 25: paginatura

introduzione

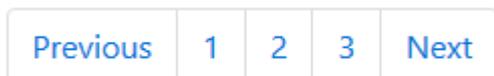
I link di impaginazione indicano che esiste una serie di contenuti correlati su più pagine. In genere, questi vengono utilizzati quando un approccio multipagina a lunghi elenchi di contenuti migliora le prestazioni generali, ad esempio nei risultati di ricerca o nelle caselle di posta.

Examples

Un semplice esempio di impaginazione

```
<nav aria-label="Page navigation example">
  <ul class="pagination">
    <li class="page-item"><a class="page-link" href="#">Previous</a></li>
    <li class="page-item"><a class="page-link" href="#">1</a></li>
    <li class="page-item"><a class="page-link" href="#">2</a></li>
    <li class="page-item"><a class="page-link" href="#">3</a></li>
    <li class="page-item"><a class="page-link" href="#">Next</a></li>
  </ul>
</nav>
```

che ti dà:



Leggi paginatura online: <https://riptutorial.com/it/twitter-bootstrap/topic/10605/paginatura>

Capitolo 26: pannelli

Osservazioni

Il componente del pannello in bootstrap è una casella (con bordi) con alcune imbottiture attorno al suo contenuto e facoltativamente i contenitori di intestazione e piè di pagina.

Examples

Esempio di base

Per impostazione predefinita, tutto il `.panel` fa applicare alcuni bordi e padding di base per contenere del contenuto.

```
<div class="panel panel-default">
  <div class="panel-body">
    Basic panel example
  </div>
</div>
```

Pannello con intestazione

Aggiungi facilmente un contenitore di `.panel-heading` al tuo pannello con `.panel-heading`. Puoi anche includere qualsiasi `<h1>-<h6>` con una classe `.panel-title` per aggiungere un'intestazione pre-styled. Tuttavia, le dimensioni dei caratteri di `<h1>-<h6>` sono sovrascritte da `.panel-heading`.

Per una corretta colorazione del link, assicurati di inserire collegamenti nei titoli all'interno di `.panel-title`.

```
<div class="panel panel-default">
  <div class="panel-heading">Panel heading without title</div>
  <div class="panel-body">
    Panel content
  </div>
</div>

<div class="panel panel-default">
  <div class="panel-heading">
    <h3 class="panel-title">Panel title</h3>
  </div>
  <div class="panel-body">
    Panel content
  </div>
</div>
```

Pannello con il piè di pagina

Avvolgi i pulsanti o il testo secondario in `.panel-footer`. Si noti che i piè di pagina del pannello **non**

ereditano colori e bordi quando si usano variazioni contestuali in quanto non sono pensati per essere in primo piano.

```
<div class="panel panel-default">
  <div class="panel-body">
    Panel content
  </div>
  <div class="panel-footer">Panel footer</div>
</div>
```

Leggi pannelli online: <https://riptutorial.com/it/twitter-bootstrap/topic/2848/pannelli>

Capitolo 27: Personalizzazione dello stile di bootstrap di Twitter

Osservazioni

Una cosa da notare è che si deve citare il nome `custom.css` dopo il `bootstrap.css` principale, altrimenti i valori di `custom.css` non verranno effettivamente implementati.

Examples

Sostituzione del CSS predefinito

Tutti adorano [twitter bootstrap](#) , ma a qualcuno di noi non piace il design predefinito. Ecco una semplice guida su come iniziare a personalizzare il design di bootstrap. Twitter bootstrap quando clonato fornisce un set di file css predefiniti che possiamo sovrascrivere.

Il file css della posta che dobbiamo sovrascrivere è `bootstrap.min.css` nella `bootstrap.min.css` `bootstrap/dist/css` .

Per sovrascrivere il design predefinito di bootstrap segui questi 2 semplici passaggi.

1. `custom.css` un file `custom.css` (o puoi `custom.css` come vuoi) e collegarlo al tuo `index.html`

```
<html>
<head>
  <title>Customize Bootstrap</title>

  <link rel="stylesheet" type="text/css" href="path/to/bootstrap.min.css">
  <!-- This must be declared after the bootstrap.min.css -->
  <link rel="stylesheet" type="text/css" href="path/to/your/custom.css">
</head>
<body>
  <!-- Do something -->
</body>
</html>
```

2. Inizia a personalizzare. Ad esempio vogliamo cambiare il colore del pulsante predefinito. Se si desidera utilizzare lo stile di pulsante predefinito di bootstrap, è necessario aggiungere la classe `btn` al `<button class="btn">Sample</button>` . Basta scrivere il seguente codice sul tuo `custom.css` .

```
.btn{
  background-color:red;
}
```

Il codice sopra produrrà qualcosa di simile.

Predefinito :



Personalizzato:



Questa tecnica ci salverà dalla riscrittura degli stili dell'intero pulsante che sono stati già scritti dai contributori di bootstrap. Questo ci ha anche impedito di scrivere la nostra classe CSS che per me è meno noiosa.

Leggi Personalizzazione dello stile di bootstrap di Twitter online: <https://riptutorial.com/it/twitter-bootstrap/topic/6030/personalizzazione-dello-stile-di-bootstrap-di-twitter>

Capitolo 28: pulsanti

Sintassi

- Classi: `.btn-default` | `.btn-primary` | `.btn-success` | `.btn-info` | `.btn-warning` | `.btn-danger` | `.btn-link`;
- Dimensioni: `.btn-lg` | `.btn-md` | `.btn-sm` | `.btn-xs`;
- Stato: attivo | disabled.

Examples

Classi di pulsanti

Bootstrap offre più classi per i pulsanti di stile e facendoli risaltare.

I pulsanti Bootstrap possono essere creati aggiungendo la classe `.btn` a un elemento.

Bootstrap Class	Ruolo (colore)
<code>.btn-default</code>	Pulsante standard (bianco)
<code>.btn-primary</code>	Fornisce un peso visivo extra e identifica l'azione primaria (blu)
<code>.btn-success</code>	Utilizzato per indicare un'azione riuscita (verde)
<code>.btn-info</code>	Pulsante contestuale per fornire informazioni (blu chiaro)
<code>.btn-warning</code>	Indica che l'utente deve essere cauto (giallo)
<code>.btn-danger</code>	Indica un'azione pericolosa o negativa (rosso)
<code>.btn-link</code>	Fai apparire il tuo pulsante come un tag di ancoraggio.

Dimensioni dei pulsanti

Puoi anche creare diverse dimensioni di pulsanti con le classi `.btn-size`

Bootstrap Class	Risultato
<code>.btn-lg</code>	Crea un pulsante di dimensioni maggiori
<code>.btn-sm</code>	Crea un pulsante di dimensioni più piccole
<code>.btn-xs</code>	Crea un pulsante extra-piccolo
<code>.btn-block</code>	I pulsanti diventano elementi a livello di blocco e coprono l'intera larghezza

Bootstrap Class	Risultato
	del genitore

Rendi attivo il pulsante

La classe `active` farà apparire premuto un pulsante.

```
<button type="button" class="btn btn-primary active">Active Primary</button>
```

Disabilita un pulsante

L'aggiunta della classe `disabled` a un pulsante renderà il pulsante non selezionabile e mostrerà un cursore proibito quando ci si passa sopra.

```
<button type="button" class="btn btn-primary disabled">Disabled Primary</button>
```

Renderizza i pulsanti orizzontalmente insieme

Più pulsanti possono essere resi orizzontalmente con la classe `.btn-group`. Basta avvolgere i pulsanti all'interno di un elemento contenitore e assegnare a quell'elemento la classe `btn-group`.

```
<div class="btn-group">
  <button type="button" class="btn btn-primary">Apples</button>
  <button type="button" class="btn btn-primary">Oranges</button>
  <button type="button" class="btn btn-primary">Pineapples</button>
</div>
```

Renderizza i pulsanti verticalmente

Applica la `.btn-group-vertical` all'elemento contenitore

```
<div class="btn-group-vertical">
  <button type="button" class="btn btn-primary">Apples</button>
  <button type="button" class="btn btn-primary">Oranges</button>
  <button type="button" class="btn btn-primary">Pineapples</button>
</div>
```

Fai in modo che il gruppo di pulsanti occupi tutta la larghezza

I pulsanti racchiusi all'interno di un elemento `.btn-group` occupano solo la larghezza necessaria. Per fare in modo che il gruppo si `.btn-group-justified` l'intera larghezza dello schermo, usa invece `.btn-group-justified`.

```
<div class="btn-group btn-group-justified">
  <a href="#" class="btn btn-primary">Apples</a>
  <a href="#" class="btn btn-primary">Oranges</a>
  <a href="#" class="btn btn-primary">Pineapples</a>
</div>
```

Leggi pulsanti online: <https://riptutorial.com/it/twitter-bootstrap/topic/4757/pulsanti>

Capitolo 29: Sistema di griglia

introduzione

Il sistema di griglia di Bootstrap consiste di 12 unità note come **Colonne** (`.col-*` class CSS) che vengono utilizzate per impaginare il contenuto da *sinistra a destra* attraverso il viewport. Le colonne sono contenute all'interno delle **righe** (classe `.row` CSS) per creare gruppi orizzontali di colonne. Righe sono collocati all'interno di un **contenitore** fisso o full-width (`.container` o `.container-fluid` rispettivamente) per il corretto allineamento. Le colonne hanno una spaziatura che crea spaziatura (nota come "grondaia") tra il contenuto delle colonne.

Osservazioni

Bootstrap include un sistema di grid fluido fluido reattivo e mobile che consente di scalare opportunamente fino a 12 colonne man mano che aumenta la dimensione del dispositivo o della vista. Include classi predefinite per la rapida creazione di layout di pagina attraverso una serie di righe e colonne che ospitano i tuoi contenuti.

Examples

Media Queries

Le query multimediali in Bootstrap ti consentono di spostare, mostrare e nascondere i contenuti in base alla dimensione della finestra. Le seguenti media query vengono utilizzate nei file LESS per creare i punti di interruzione delle chiavi nel sistema di griglia Bootstrap:

```
/* Small devices (tablets, 768px and up) */
@media (min-width: @screen-sm-min) { ... }

/* Medium devices (desktops, 992px and up) */
@media (min-width: @screen-md-min) { ... }

/* Large devices (large desktops, 1200px and up) */
@media (min-width: @screen-lg-min) { ... }
```

Occasionalmente questi vengono espansi per includere una larghezza massima per limitare i CSS a un insieme più ristretto di dispositivi:

```
@media (max-width: @screen-xs-max) { ... }
@media (min-width: @screen-sm-min) and (max-width: @screen-sm-max) { ... }
@media (min-width: @screen-md-min) and (max-width: @screen-md-max) { ... }
@media (min-width: @screen-lg-min) { ... }
```

Livelli griglia di bootstrap (punti di interruzione)

Oltre al concetto di **unità di colonna**, Bootstrap ha diversi **punti di interruzione** o dimensioni di

griglia noti come livelli. La griglia Bootstrap 3 ha quattro (4) livelli per adattarsi alle diverse larghezze dello schermo (o della finestra). I livelli Bootstrap 3 sono `xs`, `sm`, `md` e `lg`. Le colonne della griglia di Bootstrap sono identificate da diverse `col-{breakpoint}-{units}` classi CSS.

Ogni livello della rete **comprende una gamma** progettata per adattarsi al meglio alle larghezze tipiche dello schermo del dispositivo, come quella dei desktop, dei laptop, dei tablet e degli smartphone.

Bootstrap utilizza query multimediali CSS per creare punti di interruzione reattivi che stabiliscono un limite per ciascuna dimensione della griglia. Queste dimensioni della griglia consentono di modificare il layout delle colonne per adattarsi al meglio a diverse larghezze e dispositivi dello schermo, l'essenza del design reattivo.

- `col-xs-*` - per le larghezze dello schermo *più piccole* come gli smartphone <768 px
- `col-sm-*` - per schermi di *piccole* dimensioni come smartphone e tablet >= 768 px
- `col-md-*` - per larghezze *medie* dello schermo come tablet e laptop >= 992 px
- `col-lg-*` - per larghezze dello schermo di *grandi* dimensioni come i desktop >= 1200 px

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

Riferimento: [Grid System](#)

Stessa larghezza della colonna per ogni dispositivo

Per creare una colonna che sia sempre il 50% della larghezza della vista (su tutti i dispositivi), puoi impostare `col-*-6` per ogni livello.

```
<div class="col-xs-6 col-sm-6 col-md-6 col-lg-6">..</div>
```

Tuttavia, questo è markup extra non necessario, poiché `col-xs-6` significa 6 unità su `xs` e su. Il livello più piccolo impostato (`xs`, `sm` o `md`) definisce anche la dimensione per le larghezze dello schermo più grandi. Per la colonna della stessa dimensione su tutti i livelli, basta impostare la larghezza per la vista più piccola.

Codice più breve:

```
<div class="col-xs-6">..</div>
```

Larghezza della colonna diversa per ciascun dispositivo (progettazione reattiva)

Le classi `col-*-*` possono essere **combinare** per controllare le larghezze delle colonne su diverse dimensioni della griglia.

Ad esempio, crea una colonna di larghezza del 50% a livello `sm` e una colonna di larghezza del 25% a livello `md` ...

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

Le griglie `sm`, `md` e `lg` verranno tutte "impilate" verticalmente con una larghezza di visualizzazione inferiore a 768 pixel. È qui che si inserisce la griglia `xs`. Le colonne che utilizzano le classi `col-xs-*` non si impilano verticalmente e continuano a ridimensionarsi sugli schermi più piccoli.

Bootstrap Rows & Columns

Il sistema di griglia di Bootstrap ha **12 unità** note come **colonne** che possono essere utilizzate per impaginare il contenuto orizzontalmente attraverso il viewport.

Il motivo per una griglia di 12 unità (anziché 10, 16, ecc.) È che 12 si divide in modo uniforme in 6 (metà), 4 (quarti) e 3 (terzi). Ciò rende molto più semplice l'adattamento a una varietà di layout. Le colonne della griglia di Bootstrap sono identificate da diverse `col-{breakpoint}-{units}` classi CSS. [Ulteriori informazioni sulla larghezza e sui punti di interruzione della vista \(livelli AKA\)](#)

Ad esempio, `col-md-3` rappresenta una colonna che occupa 3 delle 12 unità (o 25%) in una finestra di visualizzazione media (`md`). Per usare una larghezza di colonna nel tuo layout, usa semplicemente la colonna `col-{breakpoint}-{units}` appropriata nel tuo markup HTML.

```
<div class="col-{breakpoint}-{units}">
```

La larghezza della colonna è fluida (larghezza non fissa), quindi le colonne consumano una *percentuale* del loro contenitore.

Unità di colonna in Bootstrap 3

- `col-*-1` : 1 di 12 (larghezza 8.33333333%)
- `col-*-2` : 2 di 12 (16,66666667% di larghezza)
- `col-*-3` : 3 di 12 (larghezza 25%)
- `col-*-4` : 4 di 12 (33,33333333% di larghezza)
- `col-*-5` : 5 di 12 (41,66666667% di larghezza)
- `col-*-6` : 6 di 12 (larghezza 50%)

- `col-*-7` : 7 di 12 (larghezza 58.33333333%)
- `col-*-8` : 8 di 12 (66,66666667% di larghezza)
- `col-*-9` : 9 di 12 (larghezza 75%)
- `col-*-10` : 10 di 12 (83,33333333% di larghezza)
- `col-*-11` : 11 di 12 (91,66666667% di larghezza)
- `col-*-12` : 12 di 12 (larghezza 100%)

Demo - Le 12 unità di colonna di Bootstrap

La riga Bootstrap

La classe Bootstrap `.row` viene utilizzata per contenere le colonne. Le colonne devono essere *sempre* posizionate in righe e le righe devono essere sempre collocate all'interno di un contenitore (`container` o `container-fluid`). La riga utilizza margini negativi (-15px) per garantire una spaziatura corretta tra il contenuto della colonna e il bordo del browser. Le righe vengono utilizzate per raggruppare le colonne orizzontalmente.

```
<div class="container">
  <div class="row">
    <!-- one more columns -->
    <div class="col-{breakpoint}-{units}">..</div>
  </div>
</div>
```

Le colonne riempiranno il file `.row` orizzontalmente da sinistra a destra e si **avvolgeranno** in una nuova riga ogni 12 colonne. Pertanto, è possibile utilizzare `.row` s per creare **interruzioni orizzontali** oppure è possibile aggiungere **più di 12** unità di colonna in un singolo elemento `.row` per disporre di **colonne che si avvolgono** (o impilano) verticalmente lungo il viewport.

Quando si utilizza il wrapping di colonne (più di 12 unità in un file `.row`), è necessario **utilizzare le risposte reattive (o le correzioni rapide)** per assicurare anche il wrapping del contenuto di colonne irregolari. Questo è essenziale quando il contenuto delle colonne varia in altezza.

Altro su Bootstrap Grid Columns & Rows

[Problemi di layout della griglia fluida Bootstrap 3?](#)

[Bootstrap 3 - riga nidificata posso aggiungere colonne a più di 12?](#)

[Riga di bootstrap e spiegazione del colle](#)

[Come funziona la griglia Bootstrap \(media\)](#)

contenitori

Bootstrap richiede un elemento contenente per avvolgere il contenuto del sito e ospitare il nostro sistema di rete. Puoi scegliere uno dei due contenitori da utilizzare nei tuoi progetti.

Utilizzare la classe `.container` per un contenitore a larghezza fissa reattiva.

```
<div class="container">
  ...
</div>
```

Utilizzare `.container-fluid` classe del `.container-fluid` per un contenitore a larghezza intera, che copre l'intera larghezza della vista.

```
<div class="container-fluid">
  ...
</div>
```

Nota: i contenitori non sono nidificabili (non è possibile inserire un contenitore in un altro contenitore), a causa del `padding` e altro.

Colonne di compensazione

Queste classi aumentano il margine sinistro di una colonna per * colonne. Ad esempio, `.col-md-offset-4` sposta `.col-md-4` su quattro colonne.

```
<div class="row">
  <div class="col-lg-4"></div>
  <div class="col-lg-4 col-lg-offset-4"></div>
</div>
<div class="row">
  <div class="col-lg-5 col-lg-offset-1"></div>
  <div class="col-lg-5 col-lg-offset-1"></div>
</div>
```

Manipolazione dell'ordine delle colonne tramite push e pull

```
<div class="container content">
  <div class="row">
    <!--Main Content-->
    <div class="col-lg-9 col-lg-push-3">
      Main Content
    </div>

    <!--Sidebar-->
    <div class="col-lg-3 col-lg-pull-9">
      Sidebar
    </div>
  </div>
</div>
```

Questo cambia l'ordine delle colonne della griglia incorporate.

Sintassi: `.col-md-push-*` e `.col-md-pull-*`.

Di Più:

[Manipolazione dell'ordine delle colonne usando col-lg-push e col-lg-pull in Twitter Bootstrap 3](#)
[Bootstrap 3: spingere / tirare le colonne solo su schermi di dimensioni più piccole](#)
[Ordinamento e impilamento colonne in Bootstrap 3](#)

Leggi Sistema di griglia online: <https://riptutorial.com/it/twitter-bootstrap/topic/3330/sistema-di-griglia>

Capitolo 30: Stampa in Bootstrap.

Examples

Utilizzo HTML di base

Gli elementi di stampa di Bootstrap consentono di indicare quali oggetti devono essere visibili quando stampati e quali devono essere nascosti.

Per rendere visibile qualcosa utilizzare uno dei seguenti a seconda dell'elemento e come dovrebbe apparire quando stampato:

```
.visible-print-block  
.visible-print-inline  
.visible-print-inline-block
```

Per nascondere qualcosa dalla stampa, usa quanto segue:

```
.hidden-print
```

Leggi **Stampa in Bootstrap**. online: <https://riptutorial.com/it/twitter-bootstrap/topic/6707/stampa-in-bootstrap->

Capitolo 31: tabelle

Examples

Tavolo semplice

Mentre gli effetti di stile possono variare a seconda del tema, la classe `.table` viene utilizzata per creare un aspetto uniforme e coerente per le tabelle di un'applicazione:

```
<table class="table">
  <tr>
    <th>Season</th>
    <th>Doctor</th>
    <th>Companion</th>
  </tr>
  <tr>
    <td>1</td>
    <td>Christopher Eccleston</td>
    <td>Rose Tyler</td>
  </tr>
</table>
```

Leggi tabelle online: <https://riptutorial.com/it/twitter-bootstrap/topic/6299/tabelle>

Capitolo 32: tabelle

Osservazioni

Ordine dei contenuti e tabelle complesse Fate attenzione che lo stile di ridimensionamento della tabella cambia l'ordine visivo del contenuto. Assicurati di applicare questo stile solo a tabelle di dati **ben formate** e semplici (e in particolare, non utilizzarle per tabelle di layout) con le celle di intestazione di tabella appropriate per ogni riga e colonna.

Inoltre, questa classe non funzionerà correttamente per le tabelle con celle che si estendono su più righe o colonne (utilizzando gli attributi `rowspan` o `colspan`).

Examples

Tavolo di base

Bootstrap definisce uno stile personalizzato per la tabella utilizzando la classe `.table`. Basta aggiungere la classe `.table` a qualsiasi `<table>` per vedere i divisori orizzontali e il riempimento:

```
<table class="table">
  <thead><tr><th>First Name</th><th>Last name</th></tr></thead>
  <tbody>
    <tr><td>John</td><td>Doe</td></tr>
    <tr><td>Fred</td><td>Bloggs</td></tr>
  </tbody>
</table>
```

Tavolo con stile avanzato

Bootstrap fornisce un paio di classi per lo stile di tabella avanzato.

Righe a righe

Avrai una tabella con righe a strisce, se aggiungi la classe `.table-striped`:

```
<table class="table table-striped">
  <thead><tr><th>First Name</th><th>Last name</th></tr></thead>
  <tbody>
    <tr><td>John</td><td>Doe</td></tr>
    <tr><td>Fred</td><td>Bloggs</td></tr>
  </tbody>
</table>
```

Nota che:

Le tabelle a strisce sono progettate tramite il selettore CSS `:nth-child`, che non è

disponibile in Internet Explorer 8.

Tavolo bordato

Avrai una tabella con bordi su tutti i lati del tavolo e delle celle, se aggiungi la classe `.table-bordered` :

```
<table class="table table-bordered">
  <thead><tr><th>First Name</th><th>Last name</th></tr></thead>
  <tbody>
    <tr><td>John</td><td>Doe</td></tr>
    <tr><td>Fred</td><td>Bloggs</td></tr>
  </tbody>
</table>
```

Passa il mouse sulle righe

Se aggiungi la classe `.table-hover` , avrai una tabella con righe evidenziate quando l'utente si posiziona su una riga:

```
<table class="table table-hover">
  <thead><tr><th>First Name</th><th>Last name</th></tr></thead>
  <tbody>
    <tr><td>John</td><td>Doe</td></tr>
    <tr><td>Fred</td><td>Bloggs</td></tr>
  </tbody>
</table>
```

Tavolo condensato

Se aggiungi la classe `.table-condensed` , il riempimento della cella predefinito verrà dimezzato, quindi avrai una tabella più compatta:

```
<table class="table table-condensed">
  <thead><tr><th>First Name</th><th>Last name</th></tr></thead>
  <tbody>
    <tr><td>John</td><td>Doe</td></tr>
    <tr><td>Fred</td><td>Bloggs</td></tr>
  </tbody>
</table>
```

Classi contestuali

Le tabelle Bootstrap supportano i colori contestuali. Per cambiare il colore di sfondo di una riga o di una cella è sufficiente aggiungere una delle seguenti classi context: `.active` , `.success` , `.info` , `.warning` , `.danger`

```

<table class="table">
  <thead><tr><th>First Name</th><th>Last name</th></tr></thead>
  <tbody>
    <tr class="success"><td>John</td><td>Doe</td></tr>
    <tr><td>Fred</td><td class="info">Bloggs</td></tr>
  </tbody>
</table>

```

Tavoli reattivi

Devi racchiudere qualsiasi `.table` nel contenitore html con la classe `.table-responsive` per creare tabelle reattive:

```

<div class="table-responsive">
  <table class="table">
    <thead><tr><th>First Name</th><th>Last name</th></tr></thead>
    <tbody>
      <tr><td>John</td><td>Doe</td></tr>
      <tr><td>Fred</td><td>Bloggs</td></tr>
    </tbody>
  </table>
</div>

```

Le tabelle reattive scorreranno orizzontalmente su dispositivi di piccole dimensioni (<768px). Non ci saranno differenze per gli schermi più grandi di 768 px.

Reflow tabella: intestazioni verticali

Ottenere un tavolo con intestazioni verticali.

Il bootstrap di Twitter ora supporta l'intestazione verticale su una tabella normale ben formattata. Per ottenere ciò basta usare la classe `.table-reflow`

Usa la classe `.table-reflow` bootstrap di twitter su una tabella ben formata per ottenere una tabella con intestazioni verticali. Inoltre è possibile combinare con l'uso di `.table-striped` e `.table-hover` per il `.table-hover` del mouse sulle colonne questa volta.

```

<table class="table table-striped table-hover table-reflow">
  <thead>
    <tr>
      <th ><strong> First Name: </strong></th>
      <th ><strong> Last Name: </strong></th>
      <th ><strong> Email: </strong></th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td > John </td>
      <td > Doe </td>
      <td > john.doe@email.com </td>
    </tr>
    <tr>
      <td > Joane </td>
      <td > Donald </td>

```

```
        <td> jane@email.com </td>
    </tr>
</tbody>
</table>
```

Dovresti controllare i documenti alpha v4 qui: [twitter-bootstrap .table-reflow](#)

Leggi tabelle online: <https://riptutorial.com/it/twitter-bootstrap/topic/6360/tabelle>

Capitolo 33: Tabs

Examples

HTML di base

```
<ul class="nav nav-tabs" role="tablist">
  <li role="presentation">
    <a href="#id-of-content-1" role="tab" data-toggle="tab">Tab 1</a>
  </li>
  <li role="presentation">
    <a href="#id-of-content-2" role="tab" data-toggle="tab">Tab 2</a>
  </li>
  <li role="presentation">
    <a href="#id-of-content-3" role="tab" data-toggle="tab">Tab 3</a>
  </li>
</ul>

<div class="tab-content">
  <div role="tabpanel" id="id-of-content-1" class="tab-pane">Tab content 1</div>
  <div role="tabpanel" id="id-of-content-2" class="tab-pane">Tab content 2</div>
  <div role="tabpanel" id="id-of-content-3" class="tab-pane">Tab content 3</div>
</div>
```

Questo creerà un set di schede con 3 schede e 3 div contenuti associati.

Schede animate

Per rendere le schede sbiadite, aggiungi `.fade` in ogni `.tab-pane`. Il pannello delle schede attivo deve avere anche `.in` class per rendere visibile il contenuto iniziale.

```
<ul class="nav nav-tabs" role="tablist">
  <li role="presentation">
    <a href="#id-of-content-1" role="tab" data-toggle="tab">
      Tab 1
    </a>
  </li>
  <li role="presentation" class="active">
    <a href="#id-of-content-2" role="tab" data-toggle="tab">
      Tab 2
    </a>
  </li>
  <li role="presentation">
    <a href="#id-of-content-3" role="tab" data-toggle="tab">
      Tab 3
    </a>
  </li>
</ul>

<div class="tab-content">
  <div role="tabpanel" id="id-of-content-1" class="tab-pane fade">
    Tab content 1
  </div>
  <div role="tabpanel" id="id-of-content-2" class="tab-pane fade active in">
```

```
    Tab content 2
</div>
<div role="tabpanel" id="id-of-content-3" class="tab-pane fade">
    Tab content 3
</div>
</div>
```

Leggi Tabs online: <https://riptutorial.com/it/twitter-bootstrap/topic/5980/tabs>

Capitolo 34: Temi Bootstrap

Examples

Temi di bootstrap rispetto alle sovrascritture delle regole

Quali sono i temi?

Ci sono diversi aspetti visivi là fuori per Bootstrap, che possono essere trovati da fonti, come [Bootswatch](#), che stanno modificando il file `bootstrap.min.css`. Puoi anche creare il tuo tema in questo modo.

Quando modificare temi e quando aggiungere nuove regole a un file `site.css`?

Quando modificare il

`bootstrap.min.css`

file e quando aggiungere il proprio file `.css`, ad esempio

`site.css`

?

A volte ci sono requisiti di stile, che devono essere fatti, indipendentemente dal tema che stai usando. Queste regole dovrebbero essere inserite nel tuo file `.css`, come `site.css`, in modo che il tema principale possa essere modificato, le regole da `site.css` si applicheranno comunque. Per fare ciò, devi solo collegare il tema bootstrap e le tue regole, per sovrascrivere quelli esistenti:

```
<link href="../../../Content/bootstrap.min.css" rel="stylesheet">
<link href="../../../Content/site.css" rel="stylesheet">
```

In questo modo, i temi Bootstrap possono essere modificati in qualsiasi momento senza perdere le regole obbligatorie, applicate da `site.css`.

Leggi Temi Bootstrap online: <https://riptutorial.com/it/twitter-bootstrap/topic/6381/temi-bootstrap>

Capitolo 35: tooltip

Osservazioni

Il suggerimento è un elemento dell'interfaccia utente che assomiglia a una piccola finestra pop-up. Solitamente viene attivato quando un utente passa il puntatore su un altro elemento, senza cliccarlo.

Per motivi di prestazioni, i tooltip devono essere inizializzati con jQuery. Il seguente codice abiliterà tutti i tooltip nel DOM:

```
<script>
  $(document).ready(function() {
    $('[data-toggle="tooltip"]').tooltip();
  });
</script>
```

Examples

Tooltips di posizionamento

Per impostazione predefinita, il suggerimento apparirà in cima all'elemento. Possiamo usare `data-placement` attributo `data-placement` per impostare la posizione del tooltip in alto, in basso, a sinistra o a destra dell'elemento.

```
<a href="#" data-toggle="tooltip" data-placement="top" title="Top tooltip">Hover</a>
<a href="#" data-toggle="tooltip" data-placement="bottom" title="Bottom tooltip">Hover</a>
<a href="#" data-toggle="tooltip" data-placement="left" title="Left tooltip">Hover</a>
<a href="#" data-toggle="tooltip" data-placement="right" title="Right tooltip">Hover</a>
```

Hover Hover Hover Hover **Right tooltip**

Possiamo anche usare `data-placement="auto"`, per riorientare dinamicamente il suggerimento. Il tooltip nel prossimo esempio il tooltip verrà visualizzato a sinistra quando possibile, altrimenti verrà visualizzato a destra.

```
<a href="#" data-toggle="tooltip" data-placement="auto left" title="To the left?">Hover</a>
```

Esempio di base

Per creare un suggerimento, è sufficiente aggiungere `data-toggle="tooltip"` attributo `data-toggle="tooltip"` e un `title` all'elemento HTML che avrà il suggerimento. L'attributo `Title` è usato per specificare il testo che viene visualizzato all'interno del tooltip.

```
<span data-toggle="tooltip" title="Hello world! ">Hover over me</span>
```



Leggi tooltip online: <https://riptutorial.com/it/twitter-bootstrap/topic/3731/tooltip>

Capitolo 36: Utilizzo di Clearfix in righe e colonne

introduzione

Quando si creano layout avanzati, potrebbero verificarsi degli scenari in cui sarà necessario utilizzare **più di 12 unità di colonna** in un singolo elemento `.row`. Il concetto di **wrapping di colonne** e **resettaggi reattivi (clearfix di AKA)** sono *essenziali* per comprendere la progettazione reattiva con Bootstrap.

[Nozioni di base sulla griglia Bootstrap](#)

Osservazioni

Le griglie di bootstrap sono straordinariamente potenti ed eleganti. Tuttavia, è necessario ricordare che il nome del framework è "Bootstrap", non "WeDidItForYou". Bootstrap **consente** un design reattivo, non lo **garantisce**.

Spetta a te rendere il tuo design veramente reattivo e offrire ai tuoi utenti la migliore esperienza possibile per l'utente finale.

Examples

The Naive First Attempt

Prima di iniziare, definiamo alcuni CSS per gli esempi. Questo è il `head` sezione del nostro campione. Io uso sempre `border-radius` e `background-color` quando sto testando, perché rende semplice vedere le divisioni cellulari senza aggiungere alcuna dimensione del bordo che potrebbe influenzare la dimensione delle celle.

```
<head>
  <title></title>
  <link rel="stylesheet"
href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
  <style>
    /* colorize all col- */
    [class^="col-"] {
      min-height: 30px;
      border-radius: 10px;
      background-color: lightblue;
    }
    /* a tall cell */
    .cell-tall {
      height: 100px;
      background-color: orange;
    }
    /* a medium-height cell */
```

```

.cell-med {
  height: 50px;
  background-color: lightgreen;
}
/* padding top-bottom for some row examples */
.row.padded {
  padding: 1rem 0 1rem 0;
}
</style>
</head>

```

Con quello fuori mano, definiamo una griglia e guardiamo i risultati perfetti in tutte le dimensioni della vista!

Utilizzando `col-xs-6 col-md-3`

```

<div class="container-fluid">
  <div class="row">
    <div class="col-xs-6 col-md-3">1</div>
    <div class="col-xs-6 col-md-3">2</div>
    <div class="col-xs-6 col-md-3">3</div>
    <div class="col-xs-6 col-md-3">4</div>
    <div class="col-xs-6 col-md-3">5</div>
    <div class="col-xs-6 col-md-3">6</div>
    <div class="col-xs-6 col-md-3">7</div>
    <div class="col-xs-6 col-md-3">8</div>
    <div class="col-xs-6 col-md-3">9</div>
    <div class="col-xs-6 col-md-3">10</div>
    <div class="col-xs-6 col-md-3">11</div>
  </div>
</div>

```

`col-xs-6 col-md-3`



`col-xs-6 col-md-3`



Le due immagini precedenti mostrano il rendering su schermi di medie e piccole dimensioni. Ricorda, otterremo QUATTRO colonne in media + a causa di `col-md-3` e DUE celle in piccolo, a

causa di `col-xs-6` .

Sembra abbastanza buono, giusto? Penso che abbiamo finito qui! Ha detto un sacco di siti bootstrap ingenui là fuori che aspettano solo di rompere ...

Il problema dell'altezza

Nel nostro " [esempio ingenuo](#) ", tutte le nostre celle avevano la stessa altezza. Il browser volutamente rompe le linee esattamente dove volevamo, e tutto sembrava giusto con il mondo. Fino a quando l'altezza entra in scena.

Prendiamo l'esempio precedente e diamo un po 'di altezza ad alcune celle, forse come vedreste in una pagina di tipo dashboard.

```
<div class="container-fluid">
  <div class="row">
    <div class="col-xs-6 col-md-3">1</div>
    <div class="col-xs-6 col-md-3 cell-tall">2</div>
    <div class="col-xs-6 col-md-3 cell-tall">3</div>
    <div class="col-xs-6 col-md-3">4</div>
    <div class="col-xs-6 col-md-3">5</div>
    <div class="col-xs-6 col-md-3 cell-med">6</div>
    <div class="col-xs-6 col-md-3">7</div>
    <div class="col-xs-6 col-md-3">8</div>
    <div class="col-xs-6 col-md-3 cell-med">9</div>
    <div class="col-xs-6 col-md-3 cell-med">10</div>
    <div class="col-xs-6 col-md-3">11</div>
  </div>
</div>
```

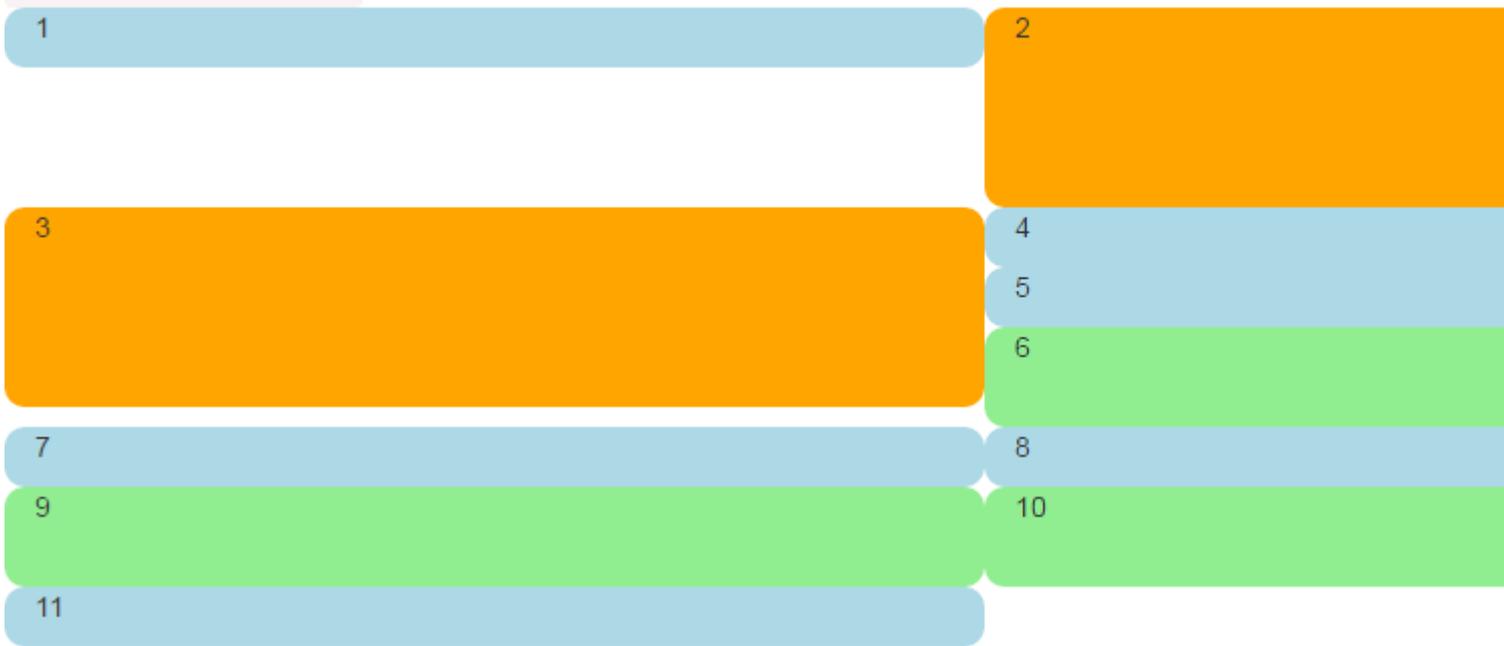
Qui abbiamo aggiunto qualche `cell-tall` e `cell-med` CSS che abbiamo definito in precedenza. Ciò avrà l'effetto di cambiare l'altezza di alcune celle. Mi chiedo come apparirà ...

Eccoli di nuovo nelle dimensioni di schermi medi e piccoli:

`col-xs-6 col-md-3`



col-xs-6 col-md-3



Oh mio Dio, che casino. Non penso che sia quello che volevamo. A dimensioni medio-grandi, 5 e 6 sono fuori luogo, e in qualche modo 7 hanno finito per iniziare una nuova riga. A piccole dimensioni abbiamo due celle nella prima fila e **quattro** nella seconda fila, con 4, 5 e 6 tutte impilate sulla destra in entrambe le dimensioni dello schermo!

Quindi, come lo risolviamo?

Clearfix a Rescue

Un modo per aiutare la situazione sicuramente, sarebbe usare più di una `row` :

```
<div class="container-fluid">
  <div class="row">
    <!-- cols -->
  </div>
  <div class="row">
    <!-- cols -->
  </div>
</div>
```

Questa è solitamente la prima cosa che provano i nuovi Bootstrapper. E sembra avere senso: "Voglio quattro celle in ogni riga, quindi mi limiterò a creare una nuova `row` per ogni 4 `col` div".

Ma c'è un problema con questa linea di ragionamento: l'intero punto di Bootstrap 3 e la prossima versione 4 devono essere **reattivi** . Inserendo "quattro `col` in una `row` ", non si sta veramente "pensando in modo reattivo".

Una buona comprensione della classe CSS `clearfix` ti aiuterà a capire che le `div` multi `row` hanno offuscato la tua comprensione del modo in cui il responsive design **doveva** funzionare. In breve, semplicemente **non puoi** sapere quanti `col` mettere in `row` - il browser non ha ancora reso il tuo

lavoro!

Ricorda in First Things First, abbiamo detto che devi pensare in "inverse of 12"? Senza ulteriori indugi, risolviamo il problema qui, utilizzando i commenti direttamente nel codice per chiarire ogni confusione. Sì, sembra molto più codice, ma la **maggior parte** dei commenti extra.

```
<div class="container-fluid">
  <div class="row">
    <div class="col-xs-6 col-md-3">1</div>
    <div class="col-xs-6 col-md-3 cell-tall">2</div>
    <!--
      We have rendered TWO cells.
      On small and extra small devices, the viewport will render TWO cells
      (12 / 6 = 2), so we need a clearfix every TWO cells. We also need to
      say "don't show this clearfix when the viewport will render FOUR cells",
      which it will do at medium size and up (12 / 3 = 4). We do that by adding
      hidden-md and hidden-lg to our clearfix div, in effect instructing the
      browser to not show it at all on a wider screen.
    -->
    <div class="clearfix hidden-md hidden-lg"></div>
    <!-->
    <div class="col-xs-6 col-md-3 cell-tall">3</div>
    <div class="col-xs-6 col-md-3">4</div>
    <!--
      We have now rendered FOUR cells.
      We are never going to have more than FOUR cells side by side. So every
      FOURTH cell, we place a clearfix that will ALWAYS show. We do this by
      just leaving off any of the hidden-* classes.
    -->
    <div class="clearfix"></div>
    <!-->
    <div class="col-xs-6 col-md-3">5</div>
    <div class="col-xs-6 col-md-3 cell-med">6</div>
    <!--
      We have now rendered SIX cells.
      After the sixth cell, we are at a multiple of TWO, but not FOUR so we
      repeat the clearfix that we used after cell TWO.
    -->
    <div class="clearfix hidden-md hidden-lg"></div>
    <!-->
    <div class="col-xs-6 col-md-3">7</div>
    <div class="col-xs-6 col-md-3">8</div>
    <!--
      Now we have rendered EIGHT cells, which is a multiple of TWO AND FOUR,
      so we put in a clearfix that's always visible.
    -->
    <div class="clearfix"></div>
    <!-->
    <div class="col-xs-6 col-md-3 cell-med">9</div>
    <div class="col-xs-6 col-md-3 cell-med">10</div>
    <!--
      After the 10th cell, once again a multiple of TWO but not FOUR...
    -->
    <div class="clearfix hidden-md hidden-lg"></div>
    <!-->
    <div class="col-xs-6 col-md-3">11</div>
  </div>
</div>
```

Il `clearfix` è una classe CSS che rende un div piccolo (praticamente invisibile), e il suo scopo è quello di "cancellare" i float di `left` che sono stati usati dai div `col`.

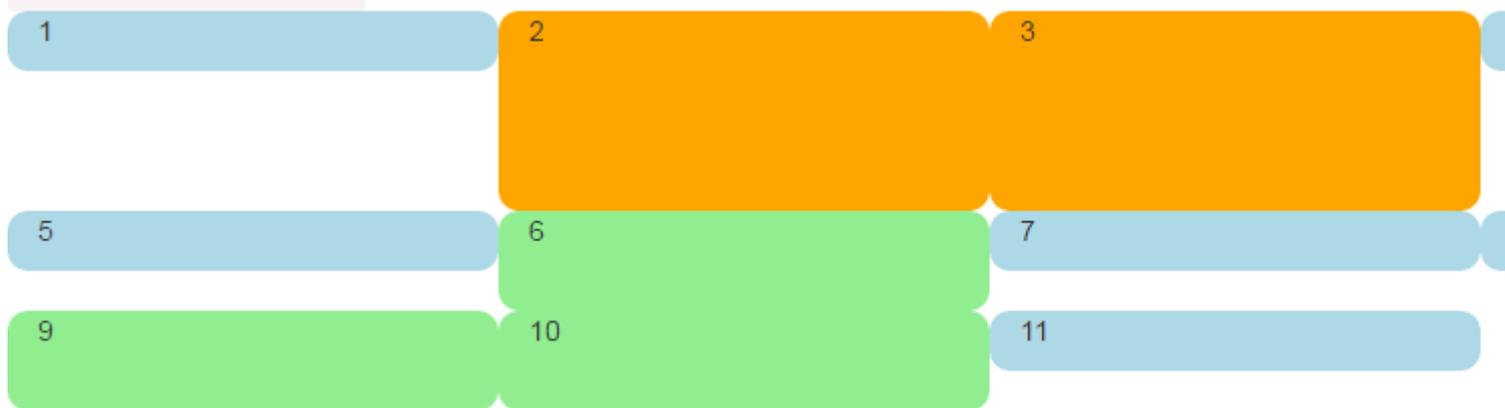
Il genio è davvero nelle classi `hidden-sm`, `hidden-md`, ecc. Queste classi sono posizionate **sul div `clearfix`**, NON sulle `col` div! Questo fa sì che il div `clearfix` magicamente o scompaia dal flusso di rendering a certe larghezze di vista! Genio!

Bootstrap ha una serie sconcertante di `hidden-*` e `visible-*` classi nella versione 3, e purtroppo non sono davvero il "inversa" l'uno dall'altro. Quindi trovo più chiaro e sicuro usare sempre le classi `hidden-*` sugli aggiornamenti chiari.

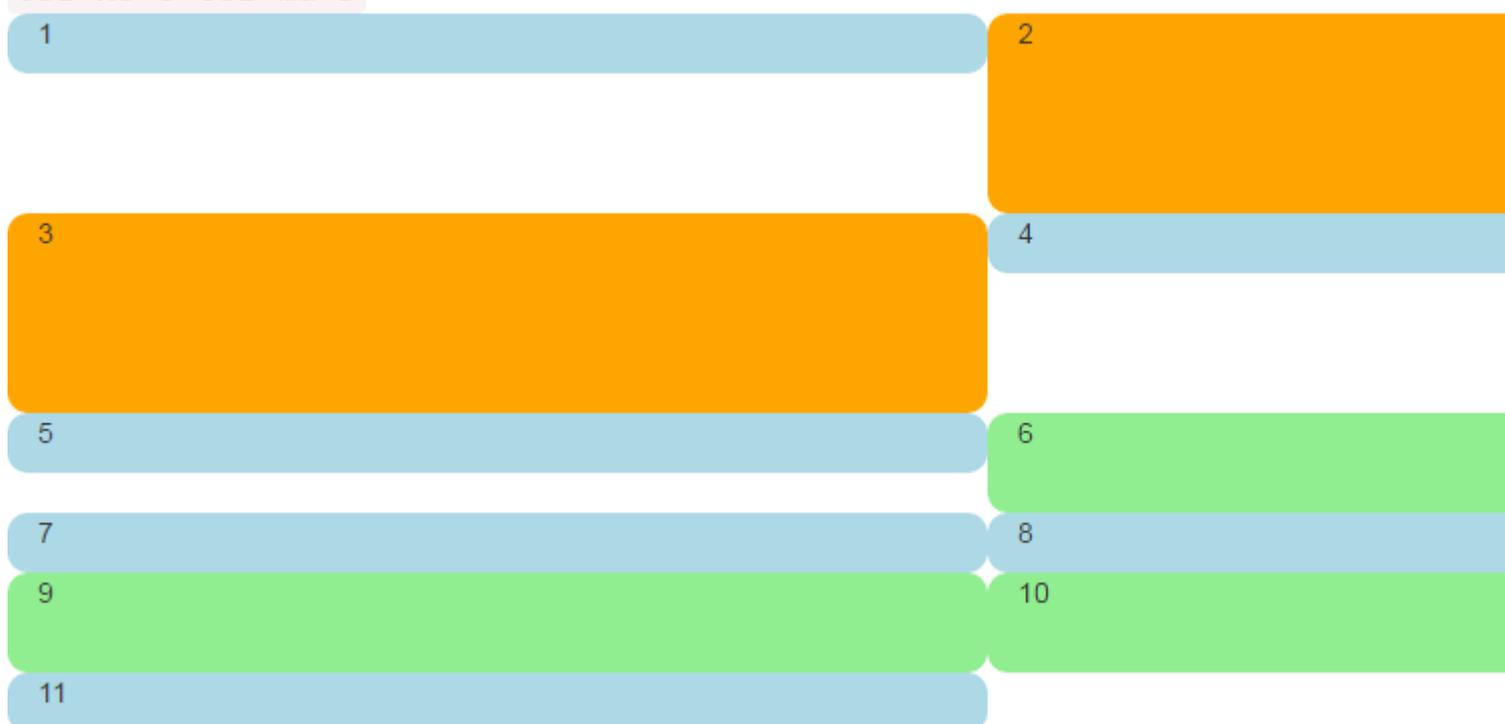
Sembra che potrebbe cambiare in meglio in Bootstrap 4, con classi come `hidden-*-up` e `hidden-*-down` (si stanno sbarazzando completamente delle classi `visible-*`).

Abbastanza verbosa, com'è adesso?

`col-xs-6 col-md-3`



`col-xs-6 col-md-3`



Questo è quello che vogliamo! Nel grande schermo, abbiamo sempre QUATTRO attraverso, nello schermo più piccolo, sempre DUE attraverso. Niente più impilature in posti strani, e le lacune sono dove ci aspetteremmo che fossero.

Un cruscotto

Abbastanza bene di quelle cose colorate arrotondate, mettiamo qualcosa di più interessante dei numeri in quelle div. Prendiamo lo stesso set di colonne e realizziamo una vera dashboard. Usa il seguente CSS:

```
<head>
  <title></title>
  <link rel="stylesheet"
href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
  <style>
    body {
      padding-top: 15px;
    }
    .panel-tall .panel-body {
      height: 175px;
    }
    .panel-med .panel-body {
      height: 100px;
    }
    .panel-short .panel-body {
      height: 70px;
    }
  </style>
</head>
```

Ed ecco il codice "dashboard":

```
<div class="container-fluid">
  <div class="row">
    <div class="col-xs-6 col-md-3">
      <div class="panel panel-default panel-med">
        <div class="panel-heading">
          Heading 1
        </div>
        <div class="panel-body">
          Body 1
        </div>
        <div class="panel-footer">
          Footer 1
        </div>
      </div>
    </div>
    <div class="col-xs-6 col-md-3 cell-tall">
      <div class="panel panel-danger panel-tall">
        <div class="panel-heading">
          Heading 2
        </div>
        <div class="panel-body">
          Body 2. Look out, this needs some attention!
        </div>
      </div>
    </div>
  </div>
</div>
```

```

        <div class="panel-footer">
            Footer 2
        </div>
    </div>
</div>
<!--
    On small and extra small devices, the viewport will render TWO cells
    (12 / 6 = 2), so we need a clearfix every TWO cells. We also need to
    say "don't show this clearfix when the viewport will render FOUR cells",
    which it will do at medium size and up (12 / 3 = 4). We do that by adding
    hidden-md and hidden-lg to our clearfix div, in effect instructing the
    browser to not show it at all on a wider screen.
-->
<div class="clearfix hidden-md hidden-lg"></div>
<!-->
<div class="col-xs-6 col-md-3 cell-tall">
    <div class="panel panel-success panel-short">
        <div class="panel-heading">
            Heading 3
        </div>
        <div class="panel-body">
            Body 3. The file has successfully uploaded.
        </div>
        <div class="panel-footer">
            Footer 3
        </div>
    </div>
</div>
<div class="col-xs-6 col-md-3">
    <div class="panel panel-default panel-tall">
        <div class="panel-heading">
            Heading 4 Chart
        </div>
        <div class="panel-body">
            Body 4. Is this a cool graph or what?
        </div>
        <div class="panel-footer">
            Footer 4
        </div>
    </div>
</div>
</div>
<!--
    We are never going to have more than FOUR cells. So every FOURTH cell,
    we place a clearfix that will ALWAYS show. We do this by just leaving off
    any of the hidden-* classes.
-->
<div class="clearfix"></div>
<!-->
<div class="col-xs-6 col-md-3">
    <div class="panel panel-warning panel-short">
        <div class="panel-heading">
            Heading 5
        </div>
        <div class="panel-body">
            Body 5.
        </div>
        <div class="panel-footer">
            Footer 5
        </div>
    </div>
</div>
</div>

```

```

<div class="col-xs-6 col-md-3 cell-med">
  <div class="panel panel-warning panel-tall">
    <div class="panel-heading">
      Heading 6
    </div>
    <div class="panel-body">
      Body 6.
    </div>
  </div>
</div>
<!--
  After the sixth cell, we are at a multiple of TWO, but not FOUR so we
  repeat the clearfix that we used after cell TWO.
-->
<div class="clearfix hidden-md hidden-lg"></div>
<!------>
<div class="col-xs-6 col-md-3">
  <div class="panel panel-info panel-tall">
    <div class="panel-heading">
      Heading 7
    </div>
    <div class="panel-body">
      Body 7.
    </div>
    <div class="panel-footer">
      Footer 7
    </div>
  </div>
</div>
<div class="col-xs-6 col-md-3">
  <div class="panel panel-info panel-med">
    <div class="panel-heading">
      Heading 8
    </div>
    <div class="panel-body">
      Body 8.
    </div>
    <div class="panel-footer">
      Footer 8
    </div>
  </div>
</div>
<!--
  Now we have rendered EIGHT cells, which is a multiple of TWO AND FOUR,
  so we put in a clearfix that's always visible.
-->
<div class="clearfix"></div>
<!------>
<div class="col-xs-6 col-md-3 cell-med">
  <div class="panel panel-info panel-short">
    <div class="panel-heading">
      Heading 9
    </div>
    <div class="panel-body">
      Body 9.
    </div>
    <div class="panel-footer">
      Footer 9
    </div>
  </div>
</div>

```

```

<div class="col-xs-6 col-md-3 cell-med">
  <div class="panel panel-info panel-tall">
    <div class="panel-heading">
      Heading 10
    </div>
    <div class="panel-body">
      Body 10.
    </div>
    <div class="panel-footer">
      Footer 10
    </div>
  </div>
</div>
<!--
  After the 10th cell, once again a multiple of TWO but not FOUR...
-->
<div class="clearfix hidden-md hidden-lg"></div>
<!------>
<div class="col-xs-6 col-md-3">
  <div class="panel panel-info panel-tall">
    <div class="panel-heading">
      Heading 11
    </div>
    <div class="panel-body">
      Body 11.
    </div>
    <div class="panel-footer">
      Footer 11
    </div>
  </div>
</div>
</div>
</div>
</div>

```

Quel codice sarà simile a questo:

Heading 1
Body 1
Footer 1

Heading 2
Body 2. Look out, this needs some attention!
Footer 2

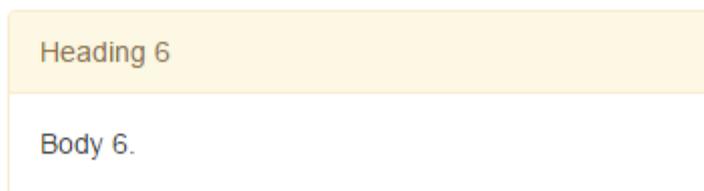
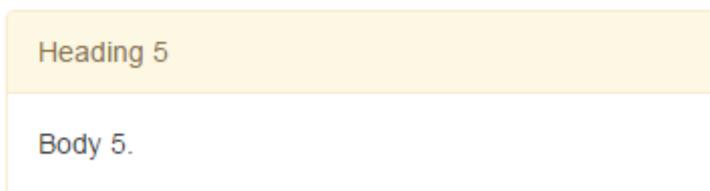
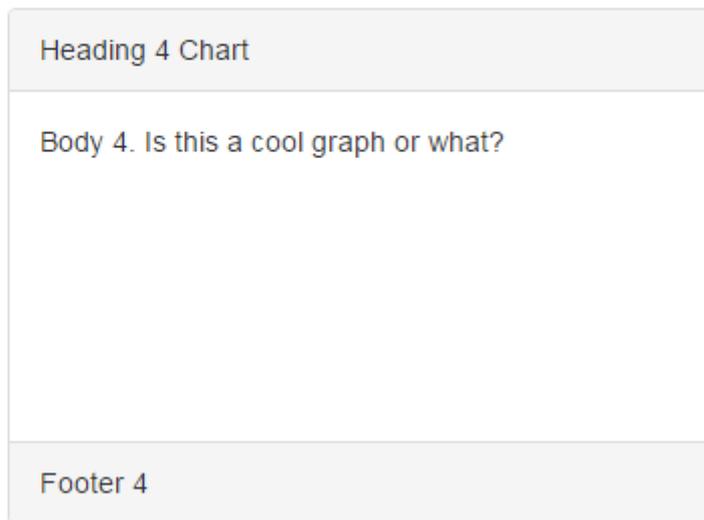
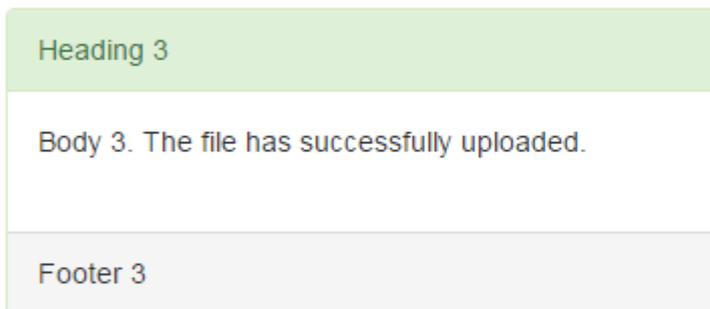
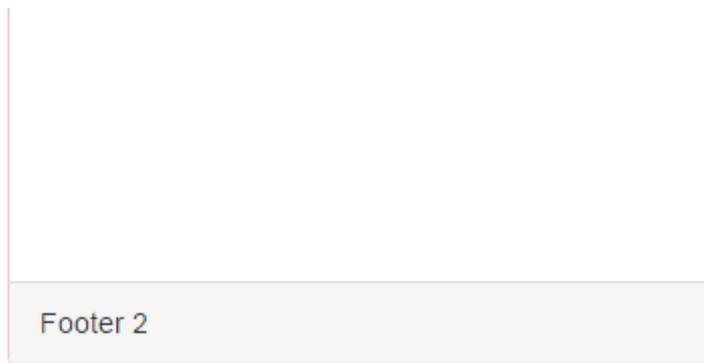
Heading 3
Body 3. The file has successfully uploaded.
Footer 3

Heading 5
Body 5.
Footer 5

Heading 6
Body 6.

Heading 7
Body 7.
Footer 7

E così in finestre più piccole:



Dal modo in cui sto usando la classe del `panel` Bootstrap 3, che andrà via in Bootstrap 4 e sarà sostituita dalla `card` molto più descrittiva e specifica. Guardando queste immagini, puoi vedere perché la `card` sarà un nome molto migliore rispetto al `panel` ambiguo.

2,4,6 Layout con Clearfix

Ecco un layout che esegue il rendering di due, quattro o sei celle a seconda delle dimensioni dello schermo.

```
<div class="container-fluid">
  <div class="row">
    <div class="col-xs-6 col-md-3 col-lg-2">1</div>
    <div class="col-xs-6 col-md-3 col-lg-2 cell-tall">2</div>
  <!--
    On small and extra small devices, the viewport will render TWO cells
    (12 / 6 = 2), so we need a clearfix every TWO cells. We also need to
    say "don't show this clearfix when the viewport will render FOUR cells",
    which it will do at medium size (12 / 3 = 4). We do that by adding
    hidden-md and hidden-lg to our clearfix div, in effect instructing the
    browser to not show it at all on a wider screen.
  -->
  <div class="clearfix hidden-md hidden-lg"></div>
```

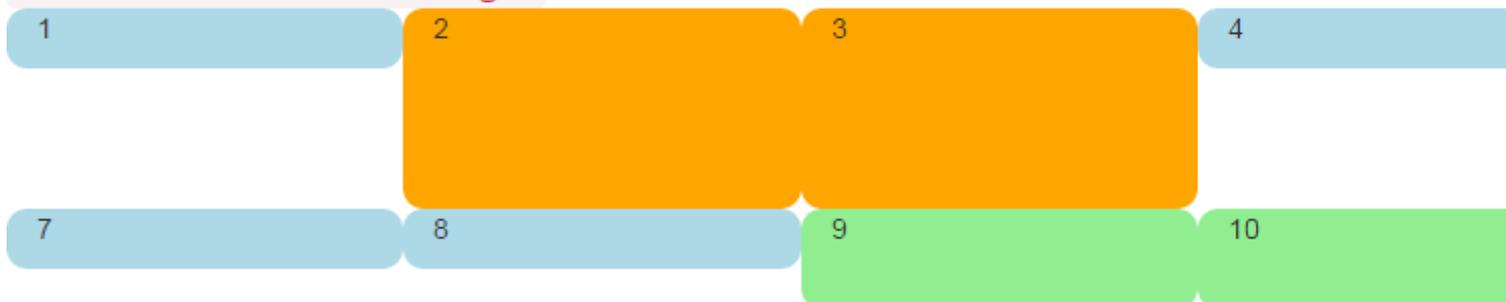
```

<!-->
<div class="col-xs-6 col-md-3 col-lg-2 cell-tall">3</div>
<div class="col-xs-6 col-md-3 col-lg-2">4</div>
<!--
    After the FOURTH cell, we need a clearfix, but it still needs to be
    hidden on LARGE viewports, because remember we will have a maximum of
    SIX cells now.
-->
<div class="clearfix hidden-lg"></div>
<!-->
<div class="col-xs-6 col-md-3 col-lg-2">5</div>
<div class="col-xs-6 col-md-3 col-lg-2 cell-med">6</div>
<!--
    After the SIXTH cell, we need to show on SMALL and LARGE, but not on
    MEDIUM. Remember, our MEDIUM viewport only wants a clearfix when we
    are at a multiple of FOUR.
-->
<div class="clearfix hidden-md"></div>
<!-->
<div class="col-xs-6 col-md-3 col-lg-2">7</div>
<div class="col-xs-6 col-md-3 col-lg-2">8</div>
<!--
    Now we have rendered EIGHT cells, which is a multiple of TWO AND FOUR,
    so we put in a clearfix that's not visible on LARGE, because we are NOT
    at a multiple of SIX.
-->
<div class="clearfix hidden-lg"></div>
<!-->
<div class="col-xs-6 col-md-3 col-lg-2 cell-med">9</div>
<div class="col-xs-6 col-md-3 col-lg-2 cell-med">10</div>
<!--
    After the 10th cell, small only.
-->
<div class="clearfix hidden-md hidden-lg"></div>
<!-->
<div class="col-xs-6 col-md-3 col-lg-2">11</div>
</div>
</div>

```

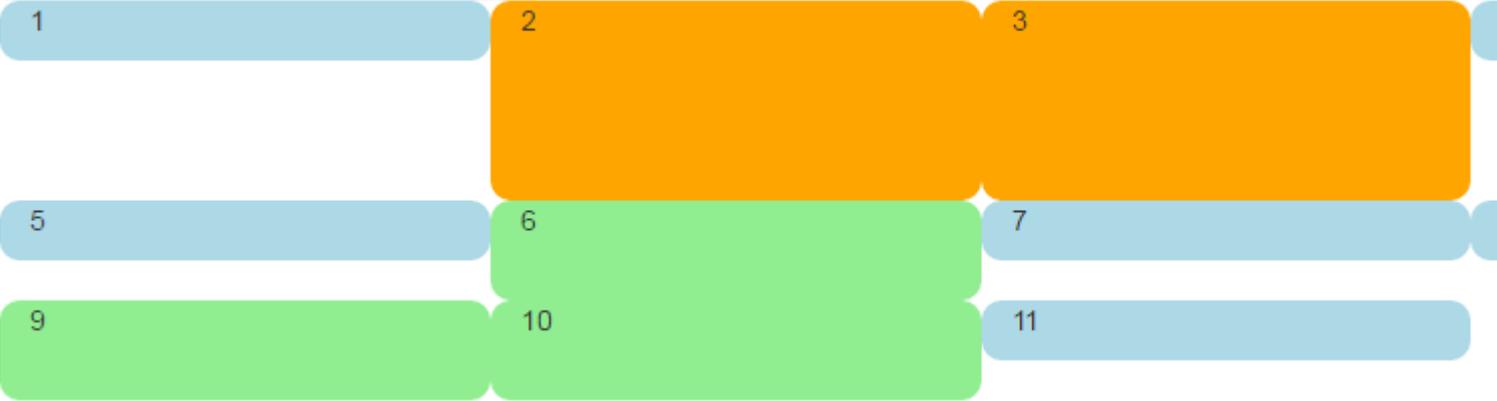
Schermo grande:

`col-xs-6 col-md-3 col-lg-2`



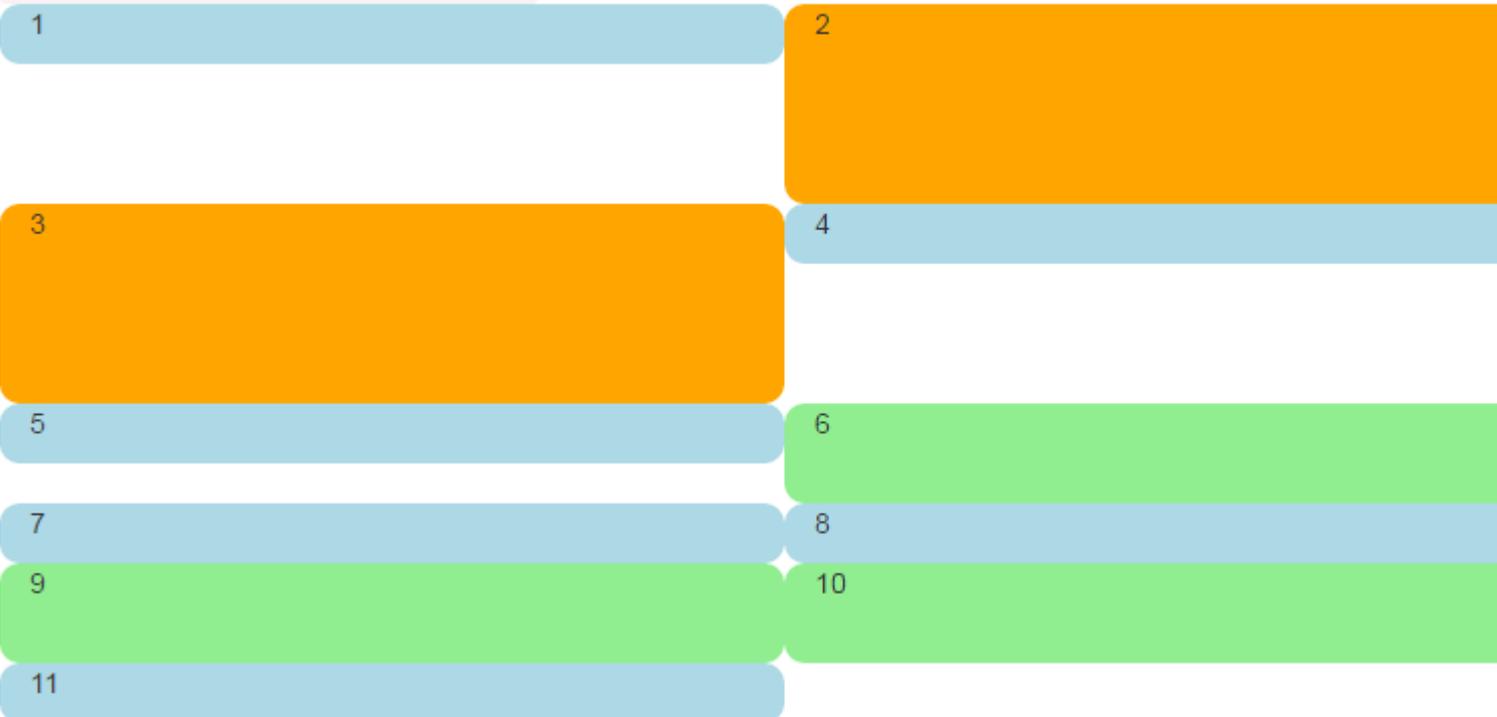
Schermo medio:

```
col-xs-6 col-md-3 col-lg-2
```



Schermo piccolo:

```
col-xs-6 col-md-3 col-lg-2
```



Perché le colonne Bootstrap superano 12 in una riga?

Esistono molti scenari reattivi in cui è *necessario* disporre di unità di colonna superiori a 12 in un singolo elemento `.row`. Questo è noto come [avvolgimento di colonne](#).

Se più di 12 colonne sono posizionate all'interno di una singola riga, ciascun gruppo di colonne aggiuntive, come una unità, verrà inserito in una nuova riga.

Ad esempio, considera un layout dove vogliamo ...

- 3 colonne su dispositivi di grandi e medie dimensioni e
- 2 colonne su dispositivi piccoli e piccoli

Large



Small



Per ottenere questo layout in Bootstrap, useremmo (correggere) ..

```
<div class="row">
  <div class="col-xs-6 col-md-4"> x </div>
  <div class="col-xs-6 col-md-4"> x </div>
</div>
```

[Demo del metodo corretto](#)

Come vedi nell'esempio, il *totale* delle unità di colonna nell'elemento `.row` **supera 12** . Questa tecnica è nota come [avvolgimento di colonne](#) ed è una delle funzionalità di progettazione responsive più potenti di Bootstrap. Il layout desiderato *non sarebbe possibile* (a parte la duplicazione del markup) se avessimo cercato di rimanere con l' **equivoco comune che le unità di colonna devono aggiungere fino a 12 in una singola riga** .

Il layout *non* è possibile quando non superiamo 12 (sbagliato) ..

```
<div class="row">
  <div class="col-xs-6 col-md-4"> x </div>
  <div class="col-xs-6 col-md-4"> x </div>
</div>
<div class="row">
  <div class="col-xs-6 col-md-4"> x </div>
  <div class="col-xs-6 col-md-4"> x </div>
</div>
<div class="row">
  <div class="col-xs-6 col-md-4"> x </div>
  <div class="col-xs-6 col-md-4"> x </div>
</div>
```

Demo del metodo errato (non riesce 3 colonne trasversali)

Ricorda, un `.row` *non* è la stessa di una singola linea attraverso il viewport. A `.row` è un raggruppamento di colonne. Le colonne che superano 12 unità in un singolo elemento `.row` si **avvolgeranno in una nuova riga** (in basso nella vista). Ecco perché è essenziale capire che le 12 colonne rappresentano **unità** orizzontali attraverso il viewport.

Inoltre, è necessario utilizzare resettaggi **reattivi** (clearfix) per il wrapping uniforme quando le **colonne variano in altezza** .

Leggi **Utilizzo di Clearfix in righe e colonne online**: <https://riptutorial.com/it/twitter-bootstrap/topic/6124/utilizzo-di-clearfix-in-righe-e-colonne>

Titoli di coda

S. No	Capitoli	Contributors
1	Iniziare con twitter-bootstrap	andreaem , Atul Mishra , bpoiss , Community , Evan , Gleb Kemarsky , H. Pauwelyn , kernal lora , Kritner , MattD , Mingle Li , Nhan , Prashanth Benny , the12 , tmg , Toby , VincenzoC , Vishnu Y S
2	Bootstrap Affix	Ilyas karim
3	Bootstrap Navbar	Ilyas karim
4	caroselli	alex , Boysenb3rry
5	Classi di utilità	ajju
6	colonne	kybernaut.cz
7	Componenti Bootstrap	alex
8	Contenitori di bootstrap	Neha Chopra
9	Convalida Bootstrap	Amy Barrett , mnoronha
10	Dialoghi modali	alex , mnoronha
11	Distintivi e etichette Bootstrap	mmativ
12	Dropdown di avvio	Ismail Farooq , MattD
13	Elenco gruppo	Ilyas karim
14	Glyphicons	Madalina Taina , tmg , Umer Farooq
15	Grid Nesting	neophyte , ZimSystem
16	Jumbotron	Gabriel Chi Hong Lee
17	Le forme	Community , Jens , Owen Pauling
18	menu a discesa	alex
19	Menu di navigazione	Ignacio Correia

20	Mettere in guardia	JHS , Madalina Taina , tmg
21	Migrazione a Bootstrap 4	Chris Farmer , neophyte , ZimSystem
22	Modals	John Blanchard
23	Nav	leowebguy
24	Navbar	Kritner , Krunal Mevada , kybernaut.cz
25	paginatura	TheDarkKnight
26	pannelli	JackPoint , tmg
27	Personalizzazione dello stile di bootstrap di Twitter	CENT1PEDE , Vikas Yadav
28	pulsanti	Madalina Taina , Muhammad Abdullah , Richard Hamilton , the12
29	Sistema di griglia	Ani Menon , Boysenb3rry , bpoiss , Harshal Patil , leowebguy , Madalina Taina , Mingle Li , mmativ , Stephen Leppik , the12 , tmg , ZimSystem
30	Stampa in Bootstrap.	MattD
31	tabelle	atjoedonahue
32	Tabs	DavidG , tmg
33	Temi Bootstrap	KAI
34	tooltip	Madalina Taina , tmg
35	Utilizzo di Clearfix in righe e colonne	Bruce Pierson , ZimSystem