

 無料電子ブック

学習

twitter-bootstrap

Free unaffiliated eBook created from
Stack Overflow contributors.

#twitter-
bootstrap

.....	1
1: twitter-bootstrap	2
.....	2
.....	2
Examples.....	3
/.....	3
.....	4
.....	5
Web.....	6
2: Navbar	8
Examples.....	8
Navbar.....	8
.....	8
Navbar divider.....	9
.....	9
.....	9
.....	10
3: Navs	11
Examples.....	11
.....	11
4: Twitter	12
.....	12
Examples.....	12
CSS.....	12
5:	14
.....	14
Examples.....	14
.....	14
.....	14
.....	15
.....	15
.....	15

6:	17
.....	17
Examples.....	17
HTML.....	17
JavaScript.....	18
7:	19
.....	19
.....	19
Examples.....	19
.....	19
.....	19
.....	21
.....	22
.....	22
.....	23
8:	24
.....	24
.....	24
Examples.....	24
.....	24
9:	25
.....	25
Examples.....	25
.....	25
10:	26
.....	26
.....	26
Examples.....	26
2.....	26
11:	27

Examples.....	27
HTML.....	27
.....	27
12:	29
.....	29
Examples.....	29
.....	29
.....	29
13:	31
Examples.....	31
.....	31
14:	32
.....	32
Examples.....	32
.....	32
.....	32
.....	32
.....	32
.....	33
.....	33
.....	33
.....	34
.....	34
15:	36
.....	36
Examples.....	36
HTML.....	36
16:	37
Examples.....	37
.....	37
.....	37

17:	38
.....	38
Examples.....	38
.....	38
.....	38
.....	38
18: 4	40
.....	40
.....	40
Examples.....	40
4.....	40
4.....	40
.....	41
.....	41
4 Navbar.....	44
34CSS.....	47
4.....	49
4.....	51
.....	51
.....	52
4.....	52
19: Navbar	54
Examples.....	54
Navbar.....	54
Bootstrap.....	54
20:	56
.....	56
Examples.....	56
.....	56
21:	57

.....	57
Examples.....	57
.....	57
22:	58
Examples.....	58
.....	58
23:	59
Examples.....	59
HTML.....	59
24:	60
.....	60
.....	60
Examples.....	60
.....	60
.....	60
25:	62
Examples.....	62
.....	62
.....	62
26:	63
Examples.....	63
Navbar.....	63
2.....	63
27:	65
.....	65
Examples.....	65
ASP.NET MVC.....	65
.....	66
.....	66
28:	67
Examples.....	67
.....

.....	67
29:	68
.....	68
Examples.....	68
.....	68
30:	69
.....	69
Examples.....	69
.....	69
31:	71
.....	71
Examples.....	71
HTML.....	71
32:	72
.....	72
Examples.....	72
HTML.....	72
JavaScript.....	72
33:	74
Examples.....	74
.hidden- * - SCSS.....	74
34:	75
.....	75
Examples.....	75
.....	75
.....	75
.....	75
.....	75
.....	75
.....	75
.....	75
.....	75
.....	75
.....	75
.....	76
.....	76

35:	77
Examples.....	77
CSSSASS.....	77
36: Clearfix	81
.....	81
.....	81
Examples.....	81
.....	81
.....	82
.....	84
.....	86
Clearfix2,4,6.....	92
12.....	94
.....	97

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

1: twitter-bootstrapをいめる

ブートストラップは、モバイルファーストパラダイムとレスポンシブウェブデザインRWDをしてされたウェブサイトをするためにされるHTML、CSS、およびJavaScriptフレームワークです。これは、プレディアCSSクラスとJavaScriptのみわせをして、Webでさまざまなことをします。これには、ウェブサイトをどのでもることができるカスタムレスポンスグリッド、レスポンシブなドロップダウンナビゲーションバー、プレミードボタン、フォーム、アコーディオンなどのをするなものなどもまれていますいくつか。

ブートストラップは、のです。

1. のブートストラップには、あらかじめされたくのものがあり、コードをくときにびされるだけです。これにより、かなりのがされ、ウェブサイトのコーディングにするをにすることができます。
2. レスポンシブな**Web**デザインをにいてされています Bootstrapは、Webのがのきさにわせてするものをするををするはありません。サイズ。
3. プロセスのブートストラップには、れたをつがあらかじめみまれています。これは、Webデザインスキルがそれほどくないや、デザインをなとしてているにとっては、Bootstrapクラスのくがににいていてえがいためにちます。

このセクションでは、twitter-bootstrapのきなテーマについてし、トピックにリンクするがあります。 twitter-bootstrapのドキュメントはしくなっているので、それらのトピックのバージョンをするがあります。

バージョン

バージョン	
4.0	2999-01-01
3.3.7	2016-07-25
3.3.6	2015-11-24
3.3.5	2015-06-15
3.3.4	2015-03-16
3.3.1	2014-11-12
3.3.0	2014-10-29
3.2.0	2014-06-26

バージョン	
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

インストール/セットアップ

ダウンロード

- ブートストラップを [GitHub](#) リポジトリからダウンロードするか、クローンするなど
- カスタマイズされたバージョンのBootstrapを [ドキュメント](#) からダウンロードする
- `bower install bootstrap` `bower install bootstrap`
- `npm install bootstrap` `npm install bootstrap`
- `composer require twbs/bootstrap`とのインストール `composer require twbs/bootstrap`

ファイル

```
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
```

インストール

HTMLページには、BootstrapのCSS、JS、およびjQueryのなくともBootstrapバージョンのバージョン3がまれています。BootstrapのJavaScriptをするの、jQueryリファレンスは、HTMLのbootstrap.jsのなるがあります。

インストールされたBootstrapファイルをのセクションからするか、またはBootstrapのメーカーがするCDNをすることができますBootstrapからのリンク。

```
<!-- 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-rHyoN1iRsVXV4nD0JutlnGaslCJuC7uwjduW9SVrLvRYooPp2bWYgmgJQIXwl/Sp" crossorigin="anonymous">

<!-- Latest compiled and minified JavaScript -->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"
integrity="sha384-Tc5IQib027qvyjSMfHjOMaLkfuWVxZxUPnCJA712mCWNIPg9mGCD8wGNICPD7Txa"
crossorigin="anonymous"></script>
```

になブートストラップのウェブページ

```
<!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>
```

テンプレート

```

<!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>

```

ブートストラップをする

ブートストラップは、HTML、CSS、Javascriptのためのなフレームワークです。これには、フォーム、ボタン、モーダルウィンドウ、ナビゲーションなどの[ユーザーインターフェイス]がけられるようになったためのなスタイリングとがまれています。

ブートストラップはレスポンシブなWebフレームワークです。つまり、モバイルデバイス、タブレット、デスクトップコンピュータなどのさまざまなサイズのレイアウトとデザインをすべての

コードベースできるようにされています。

ブートストラップの1つがグリッドフレームワークです。HTMLにクラスをすることで、12のグリッドをしてレイアウトをすることができます。たとえば、4のレイアウトは、タブレットデバイスの2つとモバイルデバイスの1つのにしています。グリッドは、`media queries`、のサイズをターゲットにするCSSメソッドをしてこれをします。

ブートストラップは、ののにです。

- カスタムデザインではありません
- カスタムCSSをするよりも、HTMLのやクラスのがです。
- あなたはのくのウェブサイトとくのなをつフレームワークをしてです

ブートストラップは、HTML、CSS、Javascriptをめてするには、[ドキュメント](#)がれているためできます。しかし、BootstrapHTML、CSS、Javascriptでされている3つのなにはれていないにとっては、があります。

ブートストラップのスタイルやをするために、ブートストラップテーマをまたはダウンロードすることができます。CSSとJavascriptをカスタマイズして、ブートストラップをととしてすることもできます。

ブートストラップコンポーネントをするWebページ

```
<!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>

```

オンラインでtwitter-bootstrapをいめるをむ <https://riptutorial.com/ja/twitter-bootstrap/topic/818/twitter-bootstrapをいめる>

2: Navbar

Examples

なNavbar ページの

```
<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>
      <!-- ^^^ Hamburger icon that gets shown when window reaches a certain scale ^^^ -->
      <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>
```

ナビゲーションバーのサブメニュー

```
<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</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>

```

Navbar divider

```

<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>

```

のナビゲーションリンクを「アクティブ」につ

```

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

```

ナビゲーションバーのブレイクポイントをするモバイルとノーマル

max-widthはブレイクポイントです。

```

@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;
}
}
}

```

ナビゲーションバーのをクリックするとりたたまれたナビゲーションバーをじる

```

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');
        }
    }
});

```

オンラインでNavbarをむ <https://riptutorial.com/ja/twitter-bootstrap/topic/2267/navbar>

3: Navs

Examples

ブートストラップナビ

BootstrapでなNavは、ベースの.navクラスからマークアップをしています。スタイルをりえるためのクラスをねえます。

タブ

```
<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>
```

```
<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>
```

された

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

ドロップダウンで

```
<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>
```

オンラインでNavsをむ <https://riptutorial.com/ja/twitter-bootstrap/topic/6505/navs>

4: Twitterのブートストラップスタイルのカスタマイズ

すべきは、メインのbootstrap.cssのにcustom.cssのをやることです。そうしないと、custom.cssのはにはされません。

Examples

デフォルトの**CSS**のオーバーライド

もが**ツイッターのブートストラップ**がきですが、たちのにはデフォルトデザインがきではありません。そこで、ブートストラップのデザインをカスタマイズするをにします。クローンのTwitterのブートストラップは、デフォルトのCSSファイルのセットをしています。

たちはきするがあるメールCSSファイルがあるbootstrap.min.cssのbootstrap/dist/cssディレクトリ。

ブートストラップのデフォルトデザインをきするには、この2つ的なステップにいます。

1. custom.cssするcustom.css またはをcustom.cssけて、index.htmlリンクします

```
<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. カスタマイズをします。たとえば、デフォルトのボタンのをしたいとします。ブートストラップのデフォルトのボタンスタイルをするは、`<button class="btn">Sample</button>`タグにbtnクラスをするがあり`<button class="btn">Sample</button>`。custom.cssのコードをいってください。

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

のコードは、このようなものをします。

デフォルト

Sample

カスタム

Sample

これは、ブートストラップのコントリビュータによってすでにかれたボタンスタイルをきすをきま
す。これはのためのcssクラスをくことからたちをってくれました。はそれほどではありません
。

オンラインでTwitterのブートストラップスタイルのカスタマイズをむ

<https://riptutorial.com/ja/twitter-bootstrap/topic/6030/twitter>のブートストラップスタイルのカスタ
マイズ

5: アラート

<http://getbootstrap.com/components/#alerts>

Examples

アラートの

のようないくつかののブートストラップコンポーネントとはなりボタン、アラートがしていません default または primary それらがなのでユーザーにするためにされているので、スタイリング。

```
<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>
```

アラート

```
<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

アニメーションアラート

`.fade`および`.in`クラスは、メッセージをじるときにフェーディングをします。

```
<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>
```

なアラート

をじるには、じるボタンに`data-dismiss="alert"`を`data-dismiss="alert"`だけです。

```
<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`および`.close`クラスはオプションであり、スタイリングにのみちます。

アラートのリンクカラー

アラートのリンクにするをすばやくするには、`.alert-link`ユーティリティクラスをします。

```
<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 ...

オンラインでアラートをむ <https://riptutorial.com/ja/twitter-bootstrap/topic/6434/アラート>

6: カルーセル

については、 <http://getbootstrap.com/javascript/#carousel>のドキュメントをしてください。ここで、なHTMLとJavaScriptのとは、しています。

CSS9のトランジション/アニメーションのにより、カルーセルはIE9でくしないことにしてください。

Examples

HTMLのない

ブートストラップカルーセルは、カルーセルのをするスライドショーをするブートストラップコンポーネントです。

なHTMLのをにします。

```
<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>
```

なJavaScriptのと

カルーセルコンポーネントは、`$('.carousel').carousel(options)` をしてjQueryによってインスタンスすることができます。`$('.carousel').carousel(options)` `$('.carousel')` はカルーセルのトップレベルで、`options` はカルーセルデフォルトの。

`options` オブジェクトをすると、カルーセルののにをえるのプロパティをでき `options` 。これらのプロパティはのようになされます。

- `interval` プロパティはJavaScriptの `number` をけります。これにより、ユーザーはカルーセルがのカルーセルスライドをするをできます。ブール `false` がされている、カルーセルはにしません。
- `pause` プロパティは、ユーザーのマウスがカルーセルにったときにカルーセルのサイクリングがするをトグルするJavaScript `string` をけれます。けられるデフォルトのそしてのは "ホバー" です。
- `wrap` プロパティはJavaScript `boolean` をけります。これにより、ユーザーは、カルノーセルがのスライドですることなくにするかどうかをすることができます。
- `keyboard` プロパティはJavaScriptの `boolean` をけります。これにより、カルーセルをキーボードイベントにさせるかどうかをユーザーができます。

に、なJavaScriptのをします。

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

のブートストラップコンポーネントとに、カルーセルのオプションはデータをしてHTMLですることできます。

オンラインでカルーセルをむ <https://riptutorial.com/ja/twitter-bootstrap/topic/1568/カルーセル>

7: グリッドシステム

き

ブートストラップのグリッドシステムは、ビューポートでコンテンツをからにレイアウトするためにされる**Columns** `.col-*-*` CSSクラスとばれる12でされています。はグループをするために `.row` CSSクラスにまれます。は、なわせのために、またはのコンテナそれぞれ `.container` または `.container-fluid` にされます。には、のコンテンツに「」とばれるをするめみがあります。

ブートストラップには、デバイスまたはビューポートのサイズがするにつれて12カラムまでにスケールアップする、のいモバイルの1グリッドシステムがまれています。コンテンツをするのをしてページレイアウトをすばやくするためのみのクラスがまれています。

Examples

メディアクエリ

ブートストラップのMedia Queriesをすると、ビューポートのサイズについてコンテンツを、にすることができます。ブートストラップグリッドシステムでキーブレイクポイントをするには、のメディアクエリがLESSファイルでされます。

```
/* 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) { ... }
```

によっては、これらをしてCSSをいデバイスセットにするmax-widthをめることができます。

```
@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) { ... }
```

ブートストラップグリッドブレイクポイント

ブートストラップには、のにて、としてられるなるブレイクポイントまたはグリッドサイズがあります。ブートストラップ3グリッドには、なるまたはビューポートにするために4つのがあります。ブートストラップ3は `xs`、 `sm`、 `md`、 `lg` です。ブートストラップのグリッドは、なる `col-{breakpoint}-{units}` CSSクラスによってされます。

グリッドには、デスクトップ、ラップトップ、タブレット、スマートフォンなどのなデバイスの

にながまれています。

ブートストラップは、CSSメディアクエリーをして、グリッドサイズのをするなブレイクポイントをします。これらのグリッドサイズをすると、さまざまなやデバイスにもよくするようにのレイアウトをすることができます。

- `col-xs-*` - スマートフォンのようなのは <768ピクセル
- `col-sm-*` - スマートフォンやタブレットのようなさなの > = 768 px
- `col-md-*` - タブレットやラップトップなどのの > = 992ピクセル
- `col-lg-*` - デスクトップなどのの > = 1200 px

	Extra small devices Phones (<768px)	Small devices Tablets (≥768px)	Medium devices Desktops (≥992px)	Large de (≥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			

リファレンス [グリッドシステム](#)

デバイスごとに

ビューポートの50すべてのデバイスのカラムをするには、ティアごとに `col-*-6` をすることができます。

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

しかし、`col-xs-6`は `xs` 6をするので、これはなマークアップではありません。したのティア `xs`、`sm`、または `md`も、のがきいのサイズをします。すべてのティアのじサイズについては、のビューポートのをします。

よりいコード

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

デバイスごとになるのいデザイン

`col-*-*`クラスをみわけて、なるグリッドサイズでのをすることができます。

たとえば、`sm`に50のをし、`md`に25のをします。

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

`sm`、`md`、および`lg`グリッドは、768ピクセルのビューポートですべてに「スタック」します。これは、`xs`グリッドがするです。`col-xs-*`クラスをするはにみねられず、のでしけます。

ブートストラップのと

ブートストラップのグリッドシステムには、ビューポートにコンテンツをレイアウトするためにできるとしてられている**12**があります。

12グリッド10,16などのわりには、12が6、4と33/3ににかけられているためです。これにより、さまざまなレイアウトににできます。ブートストラップのグリッドは、なる`col-{breakpoint}-{units}` CSSクラスによってされます。 [ビューポートのとブレイクポイントのAKA Tier](#)

たとえば、`col-md-3`は、`md`ののビューポートで12または25のうち3をめるをします。レイアウトでをするには、HTMLマークアップでな`col-{breakpoint}-{units}`クラスをします。

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

のはではないであるため、はコンテナのパーセンテージをします。

ブートストラップ3の

- `col-*-1` 1 of 128.33333333
- `col-*-2` 212 of 216.66666667
- `col-*-3` 25
- `col-*-4` 412 of 433.33333333
- `col-*-5` 512 of 541.66666667
- `col-*-6` 6 of 1250
- `col-*-7` 712 of 758.33333333
- `col-*-8` 812 of 866.66666667
- `col-*-9` 9 of 1275
- `col-*-10` 1010 of 1083.33333333
- `col-*-11` 1111 of 1191.66666667
- `col-*-12` of 12100

デモ - ブートストラップの12

ブートストラップ

ブートストラップの`.row`クラスは、をするためにされます。はににし、はにコンテナ `container`ま

は `container-fluid` のにするがあります。のとブラウザのののなをするために、はのマージン-15ピクセルをします。は、をにグループするためにされます。

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

はから `.row` をりつぶし、12ごとにしいにりします。したがって、`.row` をしてブレークをするか、1つの `.row` に12のをして、ビューポートをに `.row` またはスタックするをたせることができます。

のりし `.row` 12をするは、の**ないリセット**または `clearfix` をして、なのをにりすがあります。これは、ののさがるにはです。

ブートストラップグリッドのとの

ブートストラップ3グリッドレイアウトの

ブートストラップ3 - ネストされたは、12までのをできますか

ブートストラップのとの

ブートストラップグリッドのみ

コンテナ

ブートストラップでは、サイトのコンテンツをラップし、グリッドシステムをするために、がです。プロジェクトである2つのコンテナのいずれかをできます。

のあるのコンテナには `.container` クラスをします。

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

ビューポートののって、フルコンテナの `.container-fluid` クラスをします。

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

コンテナは、`padding` などのためにネストではありませんのコンテナにコンテナをくことはできません。

をオフセットする

これらのクラスは、のを*だけさせます。たとえば、`.col-md-offset-4` は4にわたって `.col-md-4` をし

ます。

```
<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>
```

プッシュプルをしたの

```
<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>
```

これにより、みみのグリッドのがされます。

.col-md-push- *および.col-md-pull- *。

もっと

Twitterのブートストラップ3でcol-lg-pushとcol-lg-pullをしたのの
ブートストラップ3さなサイズでのみをプッシュ/プルする
ブートストラップ3ののとスタック

オンラインでグリッドシステムをむ <https://riptutorial.com/ja/twitter-bootstrap/topic/3330/グリッドシステム>

8: グリッドネスト

き

ブートストラップでは、[このリンク](#)があるのでグリッドをすることができます。これは、[このリンク](#)のグリッドをするなレイアウトをするにちます。

なのでできるだけくのをつことができます。

Examples

のネスト

```
<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>
```

これは、[\[http://getbootstrap.com/css/#grid-nesting \]](http://getbootstrap.com/css/#grid-nesting) [1]

ウェブサイトののがするように

のグリッドでコンテンツをネストするには、[このリンク](#)の.col-sm- *にしい.rowと.col-sm- *のセットをします。ネストされたには、[このリンク](#)の12のがまれているがあります12のなをすべてするはありません。

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

オンラインでグリッドネストをむ <https://riptutorial.com/ja/twitter-bootstrap/topic/9088/グリッドネスト>

9: グリフィコン

このセクションでは、ブートストラップグリフコンのとグリフコンのいについてします。

Examples

グリフィコンのい

Twitter Bootstrapは、グリフィコンと呼ばれるアイコンをサポートしており、HTMLのすべてのタグでできます。

すべてのアイコンには、クラスとのアイコンクラスがです。

アイコンクラスはのコンポーネントとできないので、にのタグをしてください。

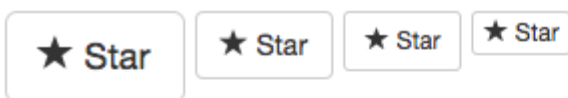
あなたのHTMLコードにのがある、そののタグのアイコンクラスをすることはできません。

たとえば、ブートストラップボタンをする、このボタンののはのようになります。

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

のではなブートストラップボタンがされましたが、このボタンにグリフアイコンをするがあります。これはを<button>タグにするだけです。このような

```
<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>
```

オンラインでグリフィコンをむ <https://riptutorial.com/ja/twitter-bootstrap/topic/6098/グリフィコン>

10: ジャンボトロン

き

Jumbotronは、あなたのウェブサイトになコンテンツをするためのBootstrapのコンポーネントです。、ナビゲーションバーのすぐ、コンテンツのにされます。

ジャンボトロンでは、すべてのグリッドシステム、コンテナクラス、クラスもします。

Examples

2のテキストとボタンがいたジャンボトロン

これは、タイトル、、ボタンきジャンボトロンです。

コード

```
<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>
```

Title text

Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Donec tortor ipsum, convallis sit.

A button

オンラインでジャンボトロンをむ <https://riptutorial.com/ja/twitter-bootstrap/topic/9188/ジャンボトロン>

11: タブ

Examples

HTML

```
<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>
```

これにより、3つのタブと3つのするコンテンツdivをつタブセットがされます。

アニメーションタブ

タブをフェードインさせるには、`.tab-pane .fade`をします。アクティブな`.in`はこのコンテンツをするための`.in`クラスもです。

```
<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>
```

オンラインでタブをむ <https://riptutorial.com/ja/twitter-bootstrap/topic/5980/タブ>

12: ツールチップ

ツールヒントは、さなポップアップボックスのようにえるユーザーインターフェイスです。これは、ユーザーがポインタをクリックせずにのにポインタをくとトリガーされます。

パフォーマンスのから、ツールチップをjQueryでするがあります。のコードは、DOMのすべてのツールチップをにします。

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

Examples

ツールチップの

デフォルトでは、のにツールチップがされます。 `data-placement` をして、の、、またはにツールチップのをできます。

```
<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**

`data-placement="auto"` をして、ツールチップをにすることもでき `data-placement="auto"` 。 ののツールチップは、であればツールチップをにし、そうでないはにします。

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

な

ツールチップをするには、 `data-toggle="tooltip"` と、ツールチップをつHTMLの `title` をするだけです。 `title` は、ツールチップにされるテキストをするためにされます。

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

Hello world!

Hover over me

オンラインでツールチップをむ <https://riptutorial.com/ja/twitter-bootstrap/topic/3731/ツールチップ>

13: テーブル

Examples

シンプルテーブル

スタイリングエフェクトはテーマによって異なることがありますが、`.table`クラスをして、アプリケーションのテーブルのデザインをします。

```
<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>
```

オンラインでテーブルをむ <https://riptutorial.com/ja/twitter-bootstrap/topic/6299/テーブル>

14: テーブル

コンテンツのとなテーブルテーブルのリフローズスタイルはコンテンツのをすることにしてください。でシンプルなデータテーブルにのみこのスタイルをしてくださいにレイアウトテーブルではしないでください。とになテーブルヘッダーセルをしてください。

さらに、このクラスは、のまたはにまたがるセルをつテーブルrowspanまたはcolspanをでしくしません。

Examples

テーブル

ブートストラップは、.tableクラスをしてテーブルのカスタムスタイリングをします。.tableクラスをの<table>にするだけで、ディバイダとパディングをすることができます

```
<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>
```

なスタイリングをえたテーブル

ブートストラップは、なテーブルスタイリングのための2つのクラスをします。

ストライプ

.table-stripedクラスをすると、がストライプされたがされます。

```
<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>
```

ご注意ください

ストライプされたのスタイルは、Internet Explorer 8ではできない:nth-child CSSセレクターによっています。

ボーダーテーブル

`.table-bordered` クラスをすると、とセルのすべてののにあるがされます。

```
<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>
```

にカーソルをわせる

`.table-hover` クラスをすると、ユーザーがをしたときにハイライトされたがされます。

```
<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>
```

テーブル

`.table-condensed` クラスをすると、デフォルトのセルのパディングがにカットされるため、よりコンパクトながされます。

```
<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>
```

コンテキストクラス

ブートストラップテーブルはコンテキストカラーをサポートします。テーブルのやセルのをするには、のなクラス `.active`、`.success`、`.info`、`.warning`、`.danger` いずれかをするだけ `.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>

```

レスポンステーブル

レスポンステーブルをするには、`.table-responsive` クラスをつHTMLコンテナのの`.table`をラップする`.table`があります。

```

<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>

```

レスポンステーブルは小さなデバイス<768pxでにスクロールします。サイズが768pxよりきいにはいはありません。

テーブルリフロー - ヘッダー

ヘッダーをつテーブルをする。

Twitterのブートストラップは、みなののヘッダーをサポートするようになりました。これをするには、`.table-reflow`クラスをするだけです

`.table-reflow` テーブルでは、twitterのブートストラップの`.table-reflow`クラスをして、ヘッダーをむテーブルをします。さらに、`.table-hover`、`.table-hover .table-striped`と`.table-hover .table-striped`をみわせて、カラムでのホバリングをうことができます。

```

<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>

```

```
</tbody>  
</table>
```

ここでv4のアルファドキュメントをチェックしてください [twitter-bootstrap .table-reflow](#)
オンラインでテーブルをむ <https://riptutorial.com/ja/twitter-bootstrap/topic/6360/テーブル>

15: ドロップダウン

については、 <http://getbootstrap.com/javascript/#dropdowns>にあるのBootstrapのドキュメントをしてください。ここでは、なHTMLのがあります。

Examples

HTMLのない

ブートストラップドロップダウンは、がクリックされたときにHTMLがサブメニュードロップダウンのをトリガできるようにするブートストラップコンポーネントです。

なHTMLのをにします。

```
<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>
```

プルダウンサブメニューをすることによってすることができる `li` element `ul` かつ `.dropdown-menu` クラス。

オンラインでドロップダウンをむ <https://riptutorial.com/ja/twitter-bootstrap/topic/6033/ドロップダウン>

16: ナビゲーションメニュー

Examples

ピルメニュー

```
<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>
```

ピルメニュー

```
<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>
```

ピル

```
<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>
```

オンラインでナビゲーションメニューをむ <https://riptutorial.com/ja/twitter-bootstrap/topic/6404/ナビゲーションメニュー>

17: パネル

ブートストラップのパネルコンポーネントは、のりにいくつかのパディングがあり、にじてしとフッターのコンテナがいたのボックスです。

Examples

な

デフォルトでは、すべての `.panel` は、いくつかのコンテンツをするためにいくつかのなとめをします。

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

しきパネル

`.panel-heading` を `.panel-heading` して、しコンテナをパネルににできます。 `pre-styled` しをするには、 `.panel-title` クラスをむ `<h1>`-`<h6>` をめることもできます。ただし、 `<h1>`-`<h6>` のフォントサイズは `.panel-heading` きされます。

なリンクのけをうには、 `.panel-title` のしにリンクをして `.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>
```

フッターきパネル

`.panel-footer` ボタンやテキストをラップします。フォアグラウンドではないため、コンテキストのバリエーションをする、パネルのフッターはやをしません。

```
<div class="panel panel-default">
```

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

オンラインでパネルをむ <https://riptutorial.com/ja/twitter-bootstrap/topic/2848/パネル>

18: ブートストラップ4への

き

ブートストラップ4はきなきえであり、ブートストラップ3からアップグレードするには、くのがあります。ブートストラップ3.xコードをブートストラップ4.xにするのクラスの、ヒント、をにします。

これはちょっとしたで、よりなをとっています。

Examples

ブートストラップ4におけるグリッドシステムのレイアウト

のコードブロックはブートストラップ3にきまれます。ブートストラップ3には4のカラム、つまりcol-md-* col-lg-* col-sm-* col-xs-*ます。にするレイアウトは、ブートストラップ3ではのようになります。

```
<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>
```

Bootstrap 4では、768pxよりしいsmグリッドがされ、かいがです。したがって、ブートストラップ4はcol-* (xs), col-sm-*, col-md-*, col-lg-*, and col-xl-*。だから、はv3の.col-md-6だったのがv4の.col-lg-6です。なさなデフォルトブレイクポイントでは、.col-6が6をすように、-xs infixがされていることにして-xs。

ブートストラップ4にじをいたければ、のようになります

```
<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>
```

グリッドレイアウトブートストラップ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				

ブラウザサポートの

twitter-bootstrap 4では、IE8, IE9, and iOS 6のサポートがされました。v4はIE10+ and iOS 7+のみIE10+ and iOS 7+。どちらかをとするサイトでは、v3をします。

twitter-bootstrap 4では、Android v5.0 Lollipop's Browser and WebViewサポートがされました。のバージョンのAndroidブラウザとWebViewはunofficially supportedしかunofficially supportedしていません。

アフィックスクラスの

はブートストラップ4からされます。

position: stickyをすることを勧めます。

Affixをしてのスタイルをしていた、ポリフィルはユースケースをサポートしていません。そのような1つのオプションは、サードパーティのScrollPos-Stylerライブラリです。

ブートストラップのによると

AffixのjQueryプラグインをしました。ポジションをすることを勧めます。およびのポリフィルについては、HTML5 Pleaseエントリをしてください。

Affixをしてのスタイルをしていた、ポリフィルはユースケースをサポートしていません

があります。そのような1つのオプションは、サードパーティのScrollPos-Stylerライブラリです。

かがBootstrap v3からBootstrap v4している、フォールバックアプローチがにされています。

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.</p>
    </div>
  </div>
</div>
```

```

        Sed ac orci quis tortor imperdiet venenatis. Duis elementum auctor accumsan.
Aliquam in felis sit amet augue.
        </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>
</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);
});

});

```

ブートストラップ4 Navbar

新しいBootstrap 4 Navbar Componentは、Bootstrap 3.xのよりもされました。ブートストラップ4では、ナビゲーションバーはデフォルトでし、ナビゲーションバーのコンテンツのアライメントがはるかににするためにフレキシボックスをしています。新しい `navbar-toggleable-*` クラスをしてNavbarブレイクポイントをすることもです。Navbarには6つのブレイクポイントサイズまたは「」があり、のNavbarオプションのいずれかをにつことができます。

- Navbarはのモバイルビューにりまれることはなく、にです。
- Navbarはにビューにりたたまれ、ハンバーガーでりえられます。
- Navbar は、4つのブレイクポイントの1つでビューにりまれます。

なブートストラップ4 Navbar

```

<nav class="navbar navbar-toggleable-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>
```

のコードからわかるように、`navbar-header`クラスはBootstrap 4からされており、フルのNavbarでは`container-fluid`はです。

Navbar ブレークポイントの

`navbar-toggableable-md`クラスは、992pxのmdブレークポイントで、のNavbarをにりたたんでそしてトグルアイコンをします。これをのブレークポイントにするには、`navbar-toggableable-md`をこれらのいずれかとするだけです。

- `navbar-toggableable` = `xs`のりたたみ<576px
- `navbar-toggableable-sm` = `sm`でのりたたみ<768px
- `navbar-toggableable-lg` = `lg`でのりたたみ<1200px

ブートストラップ4ブレークポイントNavbarデモ

Navbar Alignmentの

Flexboxにより、Navbarとそのコンテンツブランド、リンク、フォームまたはテキストのをにすることができます。デフォルトのNavbarコンテンツはえになります。もちろん、のくのアライメントシナリオがあります...

- ブランドデフォルト、リンクセンター
- ブランドセンター、リンクの
- のブランドとのリンク
- ブランド、リンク、のフォーム
- ブランドなし、リンクセンター
- のブランド、コンテナのリンク
- みリンクをめるをに

ブートストラップ4 **Navbar with Centered Brand**、/リンク

```
<nav class="navbar navbar-toggableable-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>
    <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>

```

ブーツストラップ4 **Navbar** ブランド、リンクセンター、

```

<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>

```

Navbar Alignment  <http://www.codeply.com/go/qhaBrcWp3v>

Bootstrap 4 Navbarの

カラー、アライメント、またはさのカスタマイズ

ブートストラップ3からブートストラップ4へのCSS

ブートストラップ4はきなきえであるため、くのブートストラップ3.xクラスがされたりされたりしています。Navbarなどのコンポーネントの、しいCSSクラスとFlexboxサポートのにより、4.xへのアップグレードは3.xからのなプロセスではありません。

しかし、のブートストラップ4のをつBootstrap 3.x CSSクラスがいくつかあります。

CSSクラス/セレクタがBootstrap 3.3.7から4alpha 6にされました

{t} - またはブレイクポイントをしますsm、md、lgなど。

xsティアがデフォルトであり、col-3、col-6などするはありません。

{u} - colのサイズをします112

ブートストラップ3.x	ブートストラップ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
。パネル	。カード
.panel-heading	.card-header
.panel-title	.card-title
.panel-body	カードブロック
.panel-footer	.card-footer
.panel-primary	.card-primary.card-inverse
.panel-success	.card-success.card-inverse
.panel-info	.card-info.card-inverse
.panel-warning	.card-warning.card-inverse
.panel-danger	.card-danger.card-inverse
.well	.card.card-block

ブートストラップ3.x	ブートストラップ4
.thumbnail	.card.card-block
.list-inline> li	.list-inline-item
.dropdown-menu> li	.dropdown-item
.nav navbar> li	.nav-item
.nav navbar> li> a	.nav-link
.navbar-right	.ml-auto
.navbar-btn	.nav-item
.navbar-fixed-top	.fixed-top
.nav-stacked	.flex-column
.btn-default	.btn-セカンダリ
.img-responsive	イム -
.img-circle	みをびた
.img-rounded	.rounded
.form-horizontal	された
。	.form-check
.checkbox	.form-check
.input-lg	.form-control-lg
.input-sm	.form-control-sm
.control-label	.form-control-label
.table-condensed	.table-sm
.pagination> li	.page-item
.pagination> li> a	.page-link
。	。 カルーセルアイテム
.text-help	.form-control-feedback
。	。 フロート

ブートストラップ3.x	ブートストラップ4
.pull-left	。フロート
.center-block	.mx-auto
.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
。ラベル	。バッジ
。バッジ	.badge.badge-pill

またしてください

[ブートストラップ3.xから4へのツール](#)

[ブートストラップ4の](#)

ブートストラップ4

またはするようになっがににCSSやブートストラップでととなっています。ののは、コンテナにあってもよいし、するにしてであっててもよい。

、ブートストラップ4は、デフォルトではフレキシボックスであることをしたにはくのならアプローチがあるマージン、フレキシボックスユーティリティ、またはディスプレイユーティリティとにエユーティリティ。

は、[Vertical Alignment Utilities](#)がなにえますが、これらはインラインとテーブルののみします。ブートストラップ4のオプションとシナリオをします...

1 - マージンをするセンター

にする1つのは、`my-auto`をする`my-auto`です。これは、をそのコンテナのにします。たとえば、`h-100`はをさにし、`my-auto`は`col-sm-12`をににします。

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

マージンデモをしたセンター

`my-auto`はyのマージンをし、とです。

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

2 - Flexboxきセンター



Taller

With supporting text

Outline

ブートストラップ4の`.row`が`display:flex`、`display:flex`をする`align-self-center`、どのにも`align-self-center`をするだけでに`.row`することができます。

```
<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>
```

または、のすべての`.row - col-*`をにえにするために、`.row`に`align-items-center`をし`align-items-center`。

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

```
<div class="col-6">
  <div class="card card-inverse card-danger">
    Taller
  </div>
</div>
</div>
```

センターなるさののデモ

3- ディスプレイ・ユースをするセンター

ブートストラップ4には、`display:table`、`display:table-cell`、`display:inline`など、`display:inline`にできるユーティリティがあります。これらは、インライン、インラインブロック、またはのセルをするためにのユーティリティでできます。

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

Display Utils デモをしたセンター

ブートストラップ4センタリング

に、またはコンテンツをするは、ブートストラップ4ではなっています。

センター

- `text-center`はきき`display:inline`され`display:inline`
- `mx-auto`は、`center-block`をセンター`display:block`きえ`display:block`
- `offset-*`または`mx-auto`をして、グリッドを`offset-*`することができます

`mx-auto` x マージンは`display:block`または`display:flex`、`%`、`vw`、`px`などがされた`display:flex`。Flexboxはデフォルトでグリッドカラムでされるため、さまざまなフレックスボックスのセンタリングもあります。

センターテキストまたはインライン `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>
```

センター—`display:block`または`display:flex` `mx-auto`

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

オフセットをした `offset-*`

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

は、`mx-auto`にくともできます。

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

デモブートストラップ4センタリング

センター

ブートストラップ4Yのセンタリングについては、のドキュメントをしてください。 [ブートストラップ4](#)

ブートストラップ4の

ブートストラップ3では、プッシュプルクラスをしてまたはをすることができました。ブートストラップ4では、プッシュプルクラスはききし、さらにフレックスボックスオーダーをすることができます。

ブートストラップ4において、プッシュプルクラスになりました`push-{viewport}-{units}`と`pull-{viewport}-{units}`と`xs`-インフィックスがされました。 `xs`と`sm`のを1-3-2のレイアウトにするのをえてみましょう。

```
<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>
```

ブートストラップ4プッシュブルのデモ

しいバージョン4は、フレキシボックスがあるので、のオプションは、のをするフレキシボックスユーティリティクラスをすることです。は、12のcol-*-12はフレックスボックスのをしてすることができます。

```
<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デモの

オンラインでブートストラップ4へのをむ <https://riptutorial.com/ja/twitter-bootstrap/topic/9090/ブートストラップ4への>

19: ブートストラップNavbar

Examples

ブートストラップNavbar

これは、Bootstrapバージョン3 Navbarのです。

```
<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>
```

Bootstrap ブランド イメージ

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

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

オンラインでブートストラップNavbarをむ <https://riptutorial.com/ja/twitter-bootstrap/topic/6233/ブートストラップnavbar>

20: ブートストラップコンテナ

き

のあるのコンテナには.containerをします。

ビューポートのによって、フルコンテナに.container-fluidをします。

Examples

コンテナ

.containerは、ブートストラップxs、sm、md、lgのサイズにして1つのをちます。

.container-fluidは、なをめるようにされます。

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

Webページがされているビューポートののじて、コンテナクラスはdivにのをえます。

、.container-fluidは、ブラウザののをにしても、にサイズがされます。

オンラインでブートストラップコンテナをむ <https://riptutorial.com/ja/twitter-bootstrap/topic/10908/ブートストラップコンテナ>

21: ブートストラップコンポーネント

については、 <http://getbootstrap.com/javascript/>のドキュメントをしてください。コンポーネントのは、 <http://getbootstrap.com/javascript/>からできます。

Examples

ブートストラップコンポーネントの

ブートストラップコンポーネントは、ブートストラップにバンドルされているオプションのjQueryプラグインのです。

ブートストラップコンポーネントのは、Javascriptをせずにするのがしいまたはなをすることです。のコンポーネントはにですが、のコンポーネントはBootstrapのなフロントエンドウィジェットののをするためにされます。

としては、 [トランジション](#)、 [モーダルダイアログボックス](#)、 [ドロップダウン](#)、 [scrollspy](#)、 [タブ](#)、 [ツールチップ](#)、 [アラート](#)、 [popovers](#)、 [ボタン](#)、 [カルーセル](#)、 [および](#)。

オンラインでブートストラップコンポーネントをむ <https://riptutorial.com/ja/twitter-bootstrap/topic/6054/ブートストラップコンポーネント>

22: ブートストラップテーマ

Examples

ブートストラップテーマとルールのオーバーライド

テーマはですか

Bootstrapには、`bootstrap.min.css`ファイルをしている [Bootswatch](#) などのソースからいくつかのがあります。このようにテーマをすることもできます。

いつテーマをし、いつ `site.css` ファイルにしいルールをするのですか

するタイミング

`bootstrap.min.css`

ファイル、およびいつの `.css` ファイルをするか

`site.css`

には、あなたがっているテーマになく、スタイルがです。これらのルールは、`site.css`などの `.css` ファイルにするがあります。メインのテーマをすることができます。その、`site.css`のルールがされます。これをうには、ブートストラップのテーマとのルールをリンクして、のルールをきするだけです。

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

こうすることで、ブートストラップのテーマは、`site.css`からされたルールをうことなくいつでもできます。

オンラインでブートストラップテーマをむ <https://riptutorial.com/ja/twitter-bootstrap/topic/6381/ブートストラップテーマ>

23: ブートストラップです。

Examples

HTMLのない

ブートストラップのプリントをすると、にされるアイテムとにするアイテムをできます。

かをえるようにするには、にじてのいずれかをし、にどのようにされるかをします。

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

かがされないようにするには、をしてください

```
.hidden-print
```

オンラインでブートストラップです。をむ <https://riptutorial.com/ja/twitter-bootstrap/topic/6707/>
ブートストラップです。

24: ブートストラップドロップダウン

パラメーター

メソッド	
Javascriptでびす	<code>\$('.dropdown-toggle').dropdown();</code>
ドロップダウンをりえ ます。	<code>\$('.dropdown-toggle').dropdown('toggle')</code>
イベントタイプ	
show.bs.dropdown	このイベントは、show instanceメソッドがびされるとすぐにしま す。
された.bs.dropdown	このイベントは、ユーザーにドロップダウンがされたときにしま すCSSトランジションがするまでします。
hide.bs.dropdown	このイベントは、hideインスタンスメソッドがびされたときにち にされます。
hidden.bs.dropdown	このイベントは、ドロップダウンがユーザーからされたときにし ますCSSトランジションがするのをちます。
イベントハンドラの	<code>\$(element).on('show.bs.dropdown', function () { // do something... })</code>

ドロップダウンをJavascript `$('.dropdown-toggle').dropdown()` びすと、data-apiすなわち `data-toggle="dropdown"` としてです。 [きをむ](#)

Examples

い

ドロップダウンメニューので `.dropdown` クラスをします。

`.dropdown-menu` クラスをにして、ドロップダウンメニュープラグインをします。

ボタンまたはハイパーリンクのクラス `.dropdown-toggle` と `data-toggle="dropdown"` `.dropdown-toggle` `data-toggle="dropdown"` をして、プラグインをびします。

な

```
<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>
```

オンラインでブートストラップドロップダウンをむ <https://riptutorial.com/ja/twitter-bootstrap/topic/6361/ブートストラップドロップダウン>

25: ブートストラップのバッジとラベル

Examples

バッジ

バッジとは、リンクにけられているアイテムのです。

バッジをするには、``の`.badge`クラスをします。

```
<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>
```

ボタンのバッジ

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

ラベル

ラベルは、かにするをするためにされます

`.label`クラスのに、``の6つのコンテキストクラス`.label-default`、`.label-primary`、`.label-success`、`.label-info`、`.label-warning`または`.label-danger`いずれかを`.label-danger`ますラベルをする

```
<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>
```

オンラインでブートストラップのバッジとラベルをむ <https://riptutorial.com/ja/twitter-bootstrap/topic/7867/ブートストラップのバッジとラベル>

26: ブートストラップ

Examples

Navbar で

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>
```

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>
```

オンラインでブートストラップをむ <https://riptutorial.com/ja/twitter-bootstrap/topic/6639/ブートストラップ>

27: ブートストラップ

- このは、フォームのにしてのみできます。
- した `onSubmit()` でをするには、プロパティになくとも1つのがです。データにはのデータがあるため、なデータはありません。 `[MinLengthAttribute(0)]` はこれがありません。したがって、のフィールドとにチェックをするには、データ `[MinLengthAttribute(0)]` します。

Examples

ASP.NET MVCとデータアノテーションの

<appSettings> Web.configViewsフォルダにをします。

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

BundleConfig.csにjqueryvalバンドルをします。

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

がなすべてのページまたは _Layout.cshmlにをします。

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

<!-- jQuery to apply bootstrap validation classes and glyphs to inputs -->
<script type="text/javascript">
    $.validator.defaults({
        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>
```

モデルのするフィールドにデータをします。

```
using System.ComponentModel.DataAnnotations;

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

ビューで、がなにをします。

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

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

サーバーのをするには、するコントローラーのアクションにをします。

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

がな

モデル

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

```
<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>
```

オプション

のjQueryをして、ほかしとのをします。

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

オンラインでブートストラップをむ <https://riptutorial.com/ja/twitter-bootstrap/topic/6388/ブートストラップ>

28: フォーム

Examples

フォームコントロールにはなクラスをせずにデフォルトスタイルがいくつかあります。

ただし、ラベルとコントロールを `.form-group` タグで `.form-group`、`na` をすることができます。

```
<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>
```

みりおよびされた

`readonly` をして、ユーザーをします。みりフィールドはできません

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

フィールドをにするには、`disabled` をし `disabled`。なフィールドはすることもできません。カーソルがされ、つようになります。

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

オンラインでフォームをむ <https://riptutorial.com/ja/twitter-bootstrap/topic/6251/フォーム>

29: ページネーション

き

ページネーションリンクは、のページにわたってのコンテンツがすることをします。に、これらは、コンテンツのいリストにするページのアプローチにより、やボックスなどのなパフォーマンスがするにされます。

Examples

なページネーションの

```
<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>
```

それはあなたにえる



オンラインでページネーションをむ <https://riptutorial.com/ja/twitter-bootstrap/topic/10605/ページネーション>

30: ボタン

- クラス `.btn-default` | `.btn-primary` | `.btn-success` | `.btn-info` | `.btn-warning` | `.btn-danger` | `.btn-link`;
- サイズ `.btn-lg` | `.btn-md` | `.btn-sm` | `.btn-xs`;
- アクティブ|した。

Examples

ボタンクラス

ブートストラップは、スタイリングボタンのためののクラスをし、それらをたせます。

ブートストラップボタンは、 `.btn` クラスをにすることでできます。

ブートストラップクラス	
<code>.btn-default</code>	ボタン
<code>.btn-primary</code>	ななをえ、なをします
<code>.btn-success</code>	したアクションをすためにされます。
<code>.btn-info</code>	をすためのコンテキストボタンい
<code>.btn-warning</code>	ユーザーがをうがあることをします
<code>.btn-danger</code>	なまたはな
<code>.btn-link</code>	ボタンをアンカータグのようにせます。

ボタンのサイズ

また、 `.btn-size` クラスでさまざまなサイズのボタンをすることもできます

ブートストラップクラス	
<code>.btn-lg</code>	よりきなサイズのボタンをします。
<code>.btn-sm</code>	さなサイズのボタンをします。
<code>.btn-xs</code>	のボタンをす
<code>.btn-block</code>	ボタンはブロックレベルのになり、のにります

ボタンをアクティブにする

`active` クラスはボタンがされたようにせます。

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

ボタンをにする

`disabled` クラスをボタンにすると、そのボタンをクリックできなくなり、そのにカーソルをくとされたカーソルがされます。

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

ボタンをにレンダリングする

のボタンは、`.btn-group` クラスでにレンダリングできます。コンテナにボタンをラップし、そのに `btn-group` クラスをします。

```
<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>
```

ボタンをにレンダリングする

コンテナに `.btn-group-vertical` クラスをする

```
<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>
```

ボタングループをにする

`.btn-group` にラップされたボタンは、なだけのをめます。グループをのに `.btn-group-justified` は、わりに `.btn-group-justified` します。

```
<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>
```

オンラインでボタンをむ <https://riptutorial.com/ja/twitter-bootstrap/topic/4757/ボタン>

31: モーダル

モーダルをするには、bootstrap.min.jsがしくするがあります。

は<http://getbootstrap.com/javascript/#modals>をしてください。

Examples

HTML モーダル

モーダルはこのページにできるダイアログウィンドウです。

```
<!-- 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>
```

オンラインでモーダルをむ <https://riptutorial.com/ja/twitter-bootstrap/topic/6320/モーダル>

32: モーダルダイアログ

については、 <http://getbootstrap.com/javascript/#modals>のドキュメントをしてください。ここでは、「なHTMLの」のがあります。

Examples

HTMLのない

ブートストラップモーダルダイアログは、ページレベルのコンテンツにするモーダルダイアログウィンドウをするブートストラップコンポーネントです。

に、HTMLのBootstrapモーダルダイアログのなをします。

```
<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 -->
```

なJavaScriptのと

モーダルダイアログコンポーネントは、 `$('#myModal').modal(options)` でjQueryによってインスタスすることができます `$('#myModal').modal(options)` ここで `$('#myModal')` はのモーダルダイアログのトップレベルので、 `options` はJavascriptオブジェクトですモーダルダイアログのデフォルト

`options` オブジェクトをすると、モーダルダイアログのにするのプロパティをでき `options` 。これらのプロパティはのようになれます。

- `backdrop` プロパティをすると、のオーバーレイをモーダルダイアログのにするかどうかをユーザーができます。プールと `"static"` がなれます。「」をすると、ユーザーがオーバーレイをクリックしたときにモーダルダイアログがじられることはありません。
- `keyboard` プロパティでは、エスケープキーがキーボードでされたときにモーダルダイアログをじるかどうかをすることができます。
- `show` プロパティをすると、モーダルがされたときにモーダルダイアログをするかどうかをユ

—ザーができます。

に、なJavaScriptのをします。

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

のブートストラップコンポーネントとに、モーダルのオプションはデータをしてHTMLですること
ともできます。

オンラインでモーダルダイアログをむ [https://riptutorial.com/ja/twitter-bootstrap/topic/5927/モーダ
ルダイアログ](https://riptutorial.com/ja/twitter-bootstrap/topic/5927/モーダルダイアログ)

33: ユーティリティクラス

Examples

すべてのブレイクポイントにして.hidden-*クラスをする - 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();
```

オンラインでユーティリティクラスをむ <https://riptutorial.com/ja/twitter-bootstrap/topic/6217/ユーティリティクラス>

34: リストグループ

あなたは、ブートストラップボタンをうとコンテキストクラスにするをっているべきです。

Examples

な

```
<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>
```

バッジ

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

リンクされたアイテム

```
<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>
```

ボタン

```
<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>
```

```
<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>
```

コンテキストクラス

```
<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>
```

カスタムコンテンツ

```
<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>
```

オンラインでリストグループをむ <https://riptutorial.com/ja/twitter-bootstrap/topic/6347/リストグループ>

35:

Examples

のいCSSまたはSASSのみ

.row-height クラスをつdivをし、に.col-height をする.col-height があります。をのメディアクエリにしたいは、.row-height と.col-height といった.col-height クラスをしてください。たとえば、.row-sm-height .col-sm-height です。

CSSのバージョン

```
.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;
}
}
}

```

SASSバージョンなブートストラップ_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;
    }
}
}
```

オンラインでをむ <https://riptutorial.com/ja/twitter-bootstrap/topic/6469/>

36: とでのClearfixの

き

なレイアウトをする、1つの`.row`に**12**のをやるがあるがあります。ブートストラップをしたレスポンシブデザインには、[カラムラッピング](#)とレスポンシブリセットAKAクリアフィックスのがです。

ブートストラップグリッドの

ブートストラップグリッドはにでエレガントです。ただし、フレームワークのは "`WeDidItForYou`"ではなく "`Bootstrap`"であることをえておくがあります。ブートストラップはレスポンシブデザインをにしますが、するものではありません。

デザインをににし、ユーザーにのエンドユーザーエクスペリエンスをすることは、としてあなたです。

Examples

ナイーブなのみ

めるに、のためにいくつかのCSSをしましょう。これはたちのサンプルの`head`です。テストにはに`border-radius`と`background-color`します。セルのきさにするのサイズをすることなく、セルをにすることができるからです。

```
<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>
```

```
    }  
  </style>  
</head>
```

それで、グリッドをし、すべてのビューポートサイズでなをてみましょう

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



の2つの、およびサイズでのレンダリングをしています。えておきますが、 col-md-3ために medium +に4、 col-xs-6ため col-md-3セルがられます。

かなりいいよねたちはここでわったとうブレイクアウトをとっているナイーブブートストラップサイトがたくさんあると...

さの

たちの「[な](#)」では、すべてののはじさでした。ブラウザはにったところでのにったラインをしてしま

いました。さがになるまで。

のをて、いくつかのセルにいくつかのさをえてみましょう。、ダッシュボードのページのようにえます。

```
<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>
```

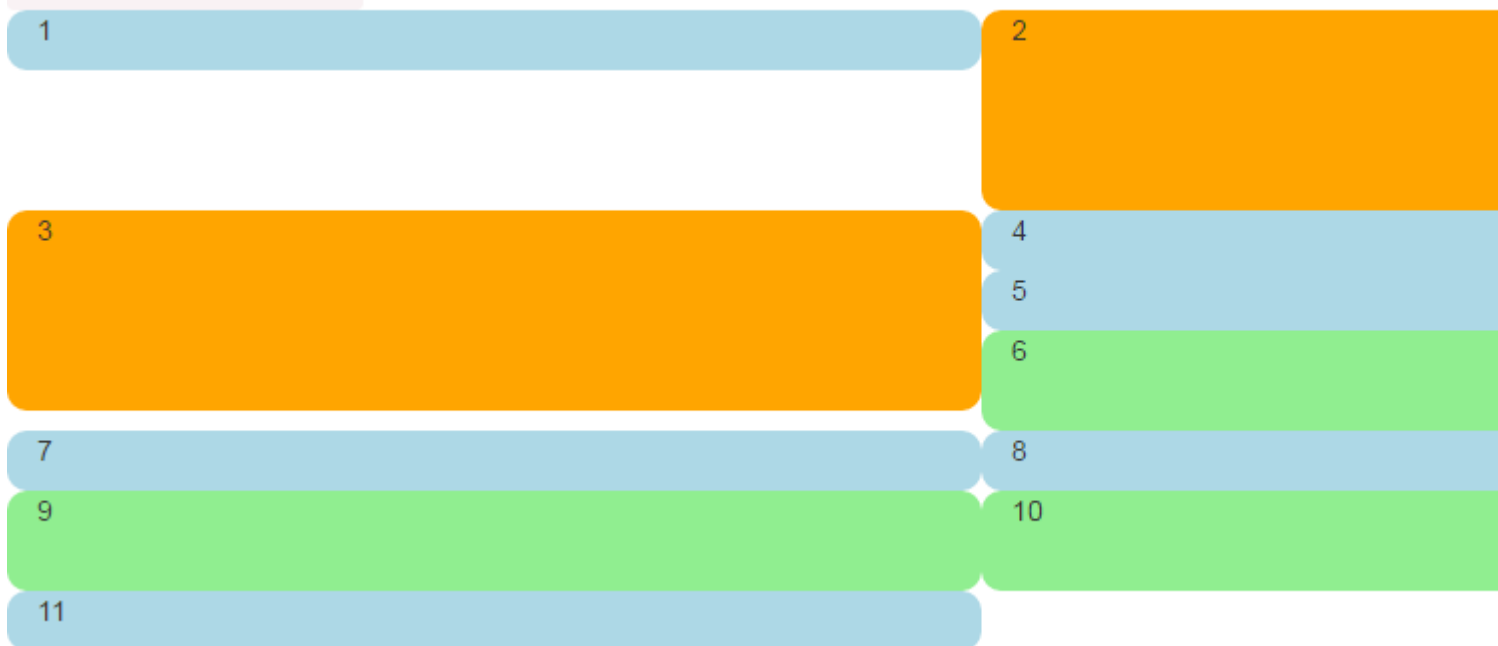
ここでは、でした `cell-tall` と `cell-med` CSS をしました。これにより、いくつかのセルのさをするがあります。どのようにえるのだろう...

ここでもらはサイズでび

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



col-xs-6 col-md-3



オハイオは、どのような。はそれがたちがかんでいたとはわない。のサイズでは、5と6がれてしま
い、とか7がしいをしました。さなサイズでは、のに2つのセルがあり、2のに4つのセルがあり、
のサイズでに4,5,6がすべてみなっています。

では、どうやってこれをするのですか

レスキューへのクリアフィックス

をにける1つのは、の`row`をすることです。

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

これは、しいBootstrappersがみるのもので。 「に4つのセルがなので、4つの`col` divごとにしい
`row`をするだけです」というになります。

しかし、こののにはがあります。ブートストラップ3とのバージョン4ののポイントはです。 「
4つの`col`を`row`する」ことによって、あなたはに「にえる」というわけではありません。

`clearfix` CSSクラスをよくすることで、の`row` divがにレスポンシブデザインがするをできないこ
とがわかります。するに、あなたは、にどのようにくをることができない `col` にれて `row` とにかく
ブラウザは、まだあなたのをレンダリングしていません

のことをいしてくださいまず、「の12」でえるがあるといましたかさらにすることなく、コードのコメントをってをするためにをしましょう。はい、もっとくのコードのようにえますが、なのほとんどはコメントです。

```
<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>
```

clearfixは、さなにえないdivをレンダリングするCSSクラスであり、そのはcol divによってされたleftを"クリア"することです。

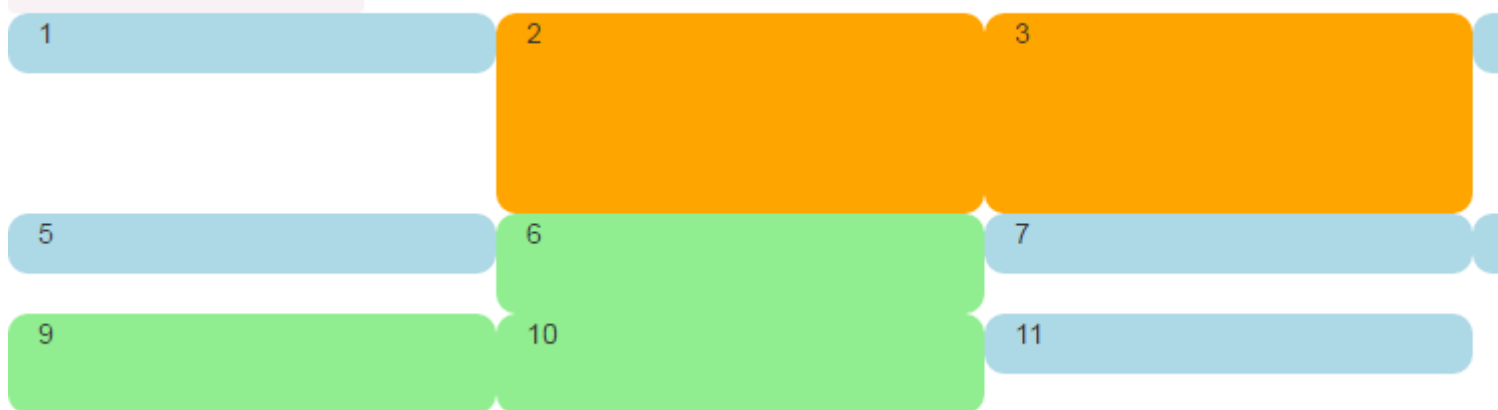
はに `hidden-sm`、`hidden-md` などのクラスです。これらのクラスは `col div` ではなく、`clearfix div` にされます。これにより、`clearfix div` がのビューポートでレンダリングストリームにののようにされたりえたりします。

ブートストラップはのながある `hidden-*` と `visible-*` バージョン3のクラスを、ながらはにの「」ではありません。だから、はにすることがもかつもなつける `hidden-*` `clearfixes` のクラスを。

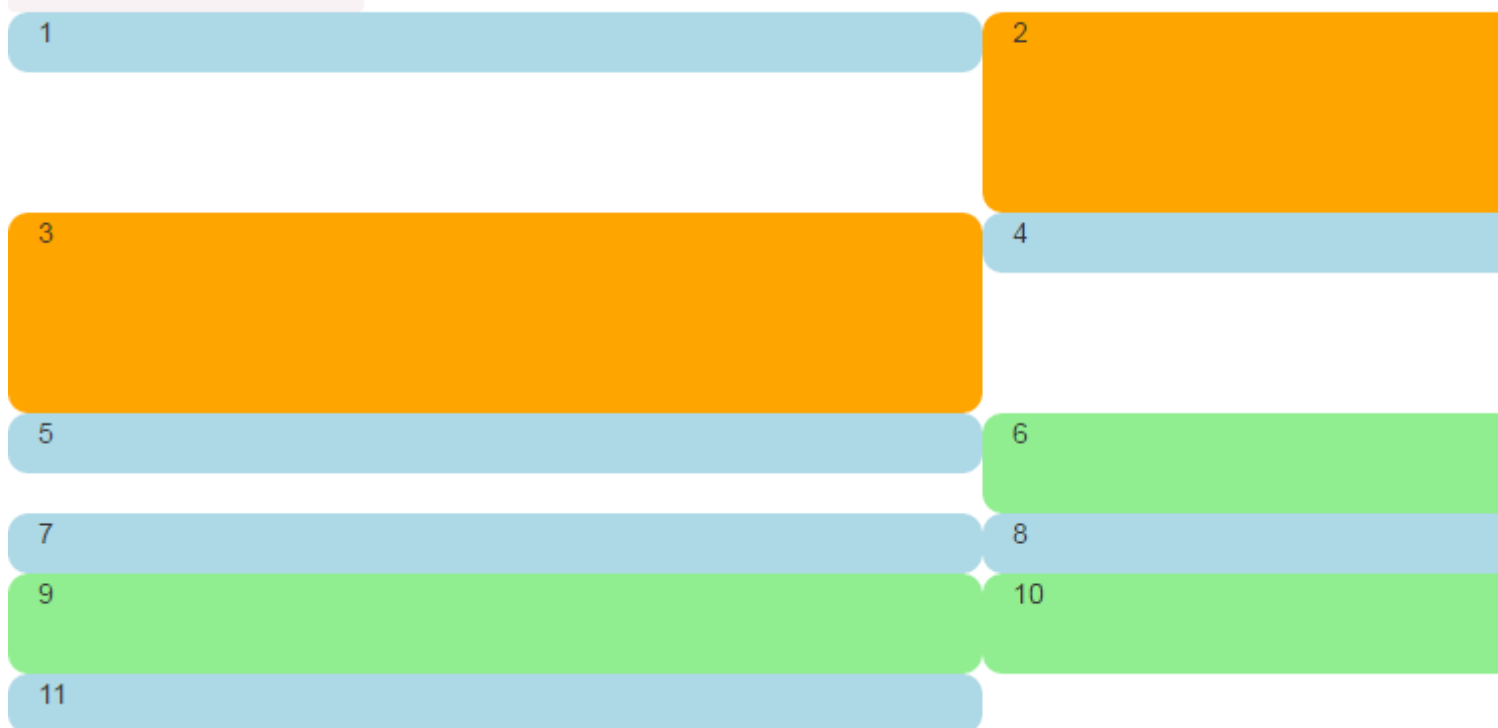
これは、`hidden-*-down` や `hidden-*-down hidden-*-up` ようなクラスで、Bootstrap 4のほうがなくなるかもしれないようですらは `visible-*` クラスをにりいています。

に、それはのよようにえる

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



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



それがたちがしいものですでは、にさいではに4つあり、に2つあります。なにみねるはなく、ギャップはたちがするところです。

ダッシュボード

まあまあのもものうち、それらのdivのよりもいいものをれてみましょう。じのセットをして、のダッシュボードをてみましょう。のCSSをしてください

```
<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>
```

そしてここに「ダッシュボード」コードがあります

```
<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 class="panel-footer">
          Footer 2
        </div>
      </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>
<!--
  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 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>
</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>
<!--
    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>
```

そのコードはのようになります。

Heading 1
Body 1
Footer 1

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

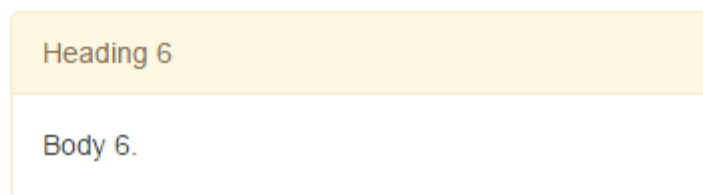
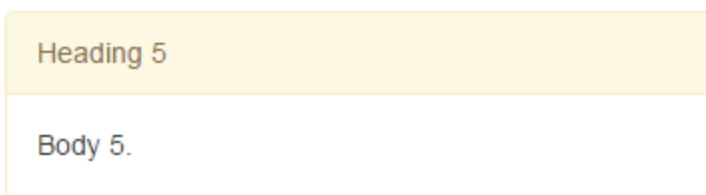
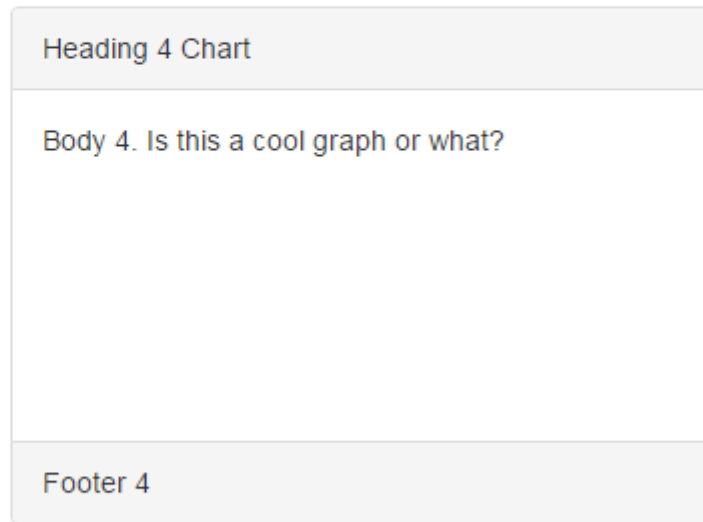
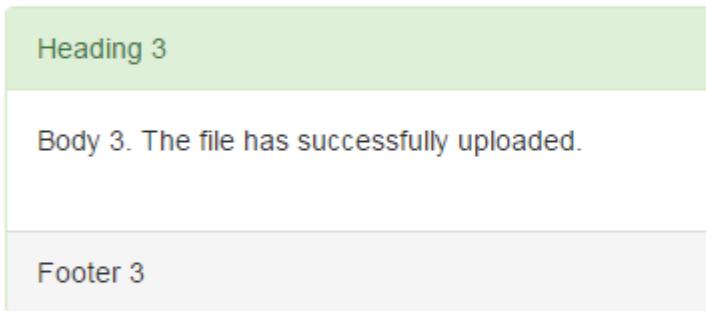
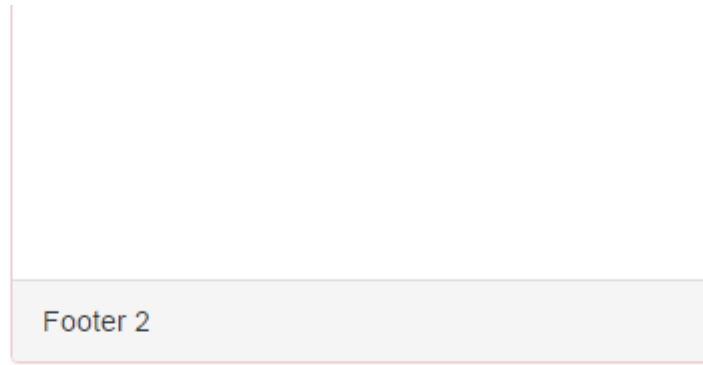
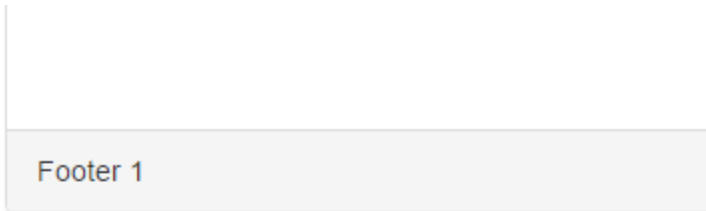
Heading 3
Body 3. The file has successfully uploaded.
Footer 3

Heading 5
Body 5.
Footer 5

Heading 6
Body 6.

Heading 7
Body 7.
Footer 7

さなビューポートではこうです



ところで、はBootstrap 3 `panel` クラスをしています。これはBootstrap 4でえてしまい、よりでな `card` きえられ `card`。これらのをと、 `card` があいまいな `panel` よりもはるかにれたになるがわかります。

Clearfixをした2,4,6のレイアウト

ここでは、サイズにじて2つ、4つ、または6つのセルをレンダリングするレイアウトをします。

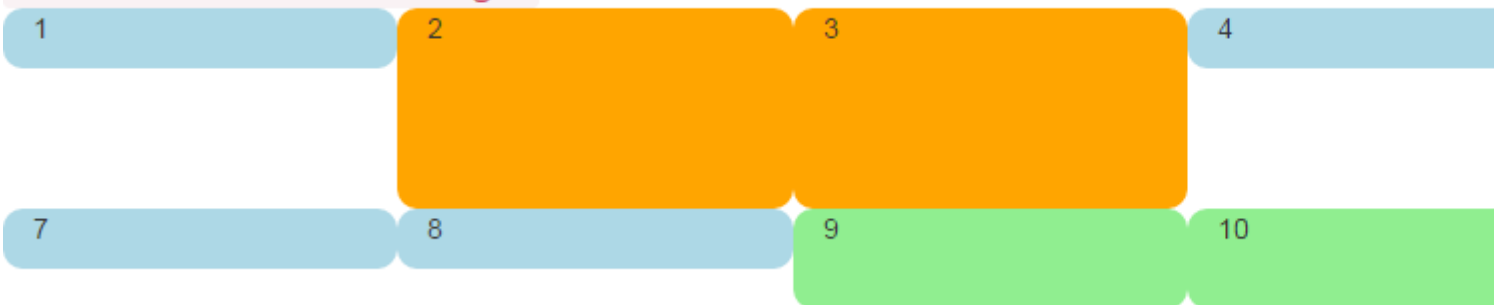
```
<div class="container-fluid">
  <div class="row">
    <div class="col-xs-6 col-md-3 col-lg-2">1</div>
    <div class="col-xs-6 col-md-3 col-lg-2 cell-tall">2</div>
  <!--
    On small and extra small devices, the viewport will render TWO cells
    (12 / 6 = 2), so we need a clearfix every TWO cells. 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>
```

```

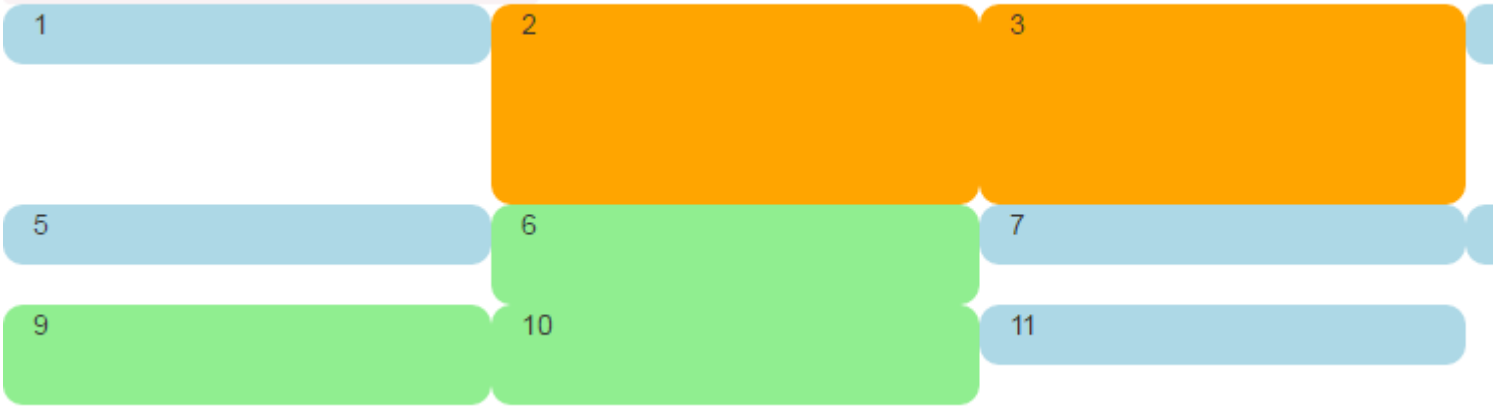
<!-->
<div class="col-xs-6 col-md-3 col-lg-2 cell-tall">3</div>
<div class="col-xs-6 col-md-3 col-lg-2">4</div>
<!--
    After the FOURTH cell, we need a clearfix, but it still needs to be
    hidden on LARGE viewports, because remember we will have a maximum of
    SIX cells now.
-->
<div class="clearfix 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>

```

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

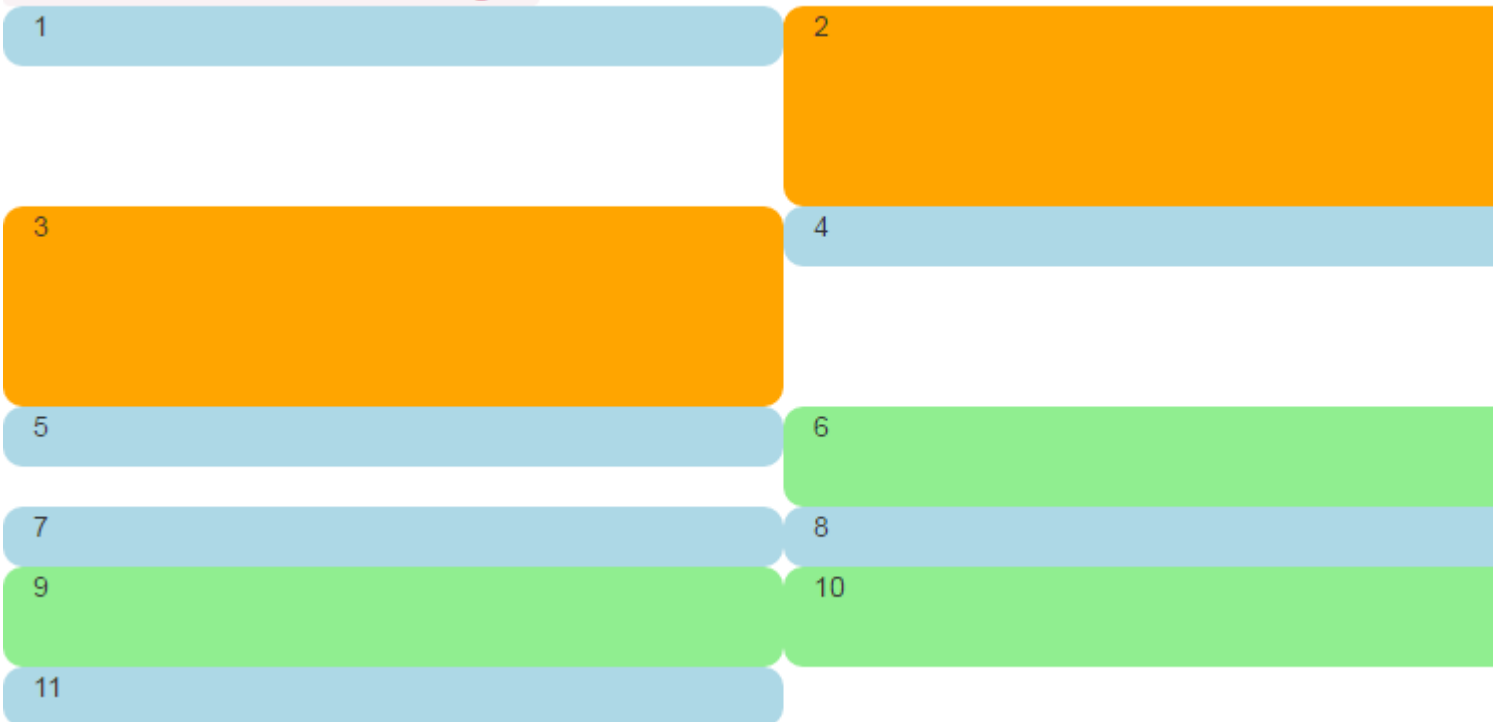


col-xs-6 col-md-3 col-lg-2



スマートフォン

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



なぜブートストラップが12をえるのですか

1つの.rowに12をえるをつがある、くのシナリオがあります。これはのりしとばれます。

1つのに12のがされている、なのグループは1つのとしてしいにりされます。

たとえば、たちがむレイアウトをえてみましょう...

- デバイスとデバイスの3つの
- デバイスとデバイスの2つの

Large



Small



ブートストラップでこのレイアウトをするには、`col-xs-6`をします。

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

メソッドのデモを

ここでは、`.row`のものが**12**をえています。これは**カラムラッピング**とばれ、ブートストラップの有名なレスポンスデザインの1つです。々は、`col`に**12**までしなければならないなにしてしようとしたのにのレイアウトは、マークアップをする ではないであろう。

12をえないとレイアウトができませんっています。

```
<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>
```

ったメソッドのデモな3つのにする

.rowはビューポートの1とじではないことに.rowください。 .rowはのグループです。 1つの.rowの12をえるは、しいビューポートのにりされます。 そのため、12はビューポートのをしていることをすることがです。

さらに、のさがするでも、リセット clearfixをしてラッピングをうがあります。

オンラインでとでのClearfixのをむ <https://riptutorial.com/ja/twitter-bootstrap/topic/6124/>とでのclearfixの

クレジット

S. No		Contributors
1	twitter-bootstrapをいめる	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	Navbar	Kritner , Krunal Mevada , kybernaut.cz
3	Navs	leowebguy
4	Twitterのブートストラップスタイルのカスタマイズ	CENT1PEDE , Vikas Yadav
5	アラート	JHS , Madalina Taina , tmg
6	カルーセル	alex , Boysenb3rry
7	グリッドシステム	Ani Menon , Boysenb3rry , bpoiss , Harshal Patil , leowebguy , Madalina Taina , Mingle Li , mmativ , Stephen Leppik , the12 , tmg , ZimSystem
8	グリッドネスト	neophyte , ZimSystem
9	グリフィコン	Madalina Taina , tmg , Umer Farooq
10	ジャンボトロン	Gabriel Chi Hong Lee
11	タブ	DavidG , tmg
12	ツールチップ	Madalina Taina , tmg
13	テーブル	atjoedonahue
14	ドロップダウン	alex
15	ナビゲーションメニュー	Ignacio Correia
16	パネル	JackPoint , tmg
17	ブートストラップ4への	Chris Farmer , neophyte , ZimSystem

18	ブートストラップ Navbar	Ilyas karim
19	ブートストラップコ ンテナ	Neha Chopra
20	ブートストラップコ ンポーネント	alex
21	ブートストラップテ ーマ	KAI
22	ブートストラップで する。	MattD
23	ブートストラップド ロップダウン	Ismail Farooq , MattD
24	ブートストラップの バッジとラベル	mmativ
25	ブートストラップ	Ilyas karim
26	ブートストラップ	Amy Barrett , mnoronha
27	フォーム	Community , Jens , Owen Pauling
28	ページネーション	TheDarkKnight
29	ボタン	Madalina Taina , Muhammad Abdullah , Richard Hamilton , the12
30	モーダル	John Blanchard
31	モーダルダイアログ	alex , mnoronha
32	ユーティリティクラ ス	ajju
33	リストグループ	Ilyas karim
34		kybernaut.cz
35	とでのClearfixの	Bruce Pierson , ZimSystem