



EBook Gratis

APRENDIZAJE

twitter-bootstrap

Free unaffiliated eBook created from
Stack Overflow contributors.

#twitter-
bootstrap

Tabla de contenido

Acerca de.....	1
Capítulo 1: Empezando con el twitter-bootstrap.....	2
Observaciones.....	2
Versiones.....	2
Examples.....	3
Instalación / Configuración.....	3
Plantilla Básica.....	5
Cuando usar Bootstrap.....	6
Página web básica utilizando componentes bootstrap.....	6
Capítulo 2: Afijo Bootstrap.....	9
Examples.....	9
En navbar.....	9
Ejemplo de afijo 2.....	9
Capítulo 3: Alerta.....	11
Observaciones.....	11
Examples.....	11
Tipos de alerta.....	11
Ejemplo básico de alerta.....	11
Alertas animadas.....	12
Alertas Desechables.....	12
Color de enlace en Alertas.....	12
Capítulo 4: Bootstrap Dropdowns.....	14
Parámetros.....	14
Observaciones.....	14
Examples.....	14
Cómo utilizar.....	14
Ejemplo básico.....	15
Capítulo 5: Bootstrap Navbar.....	16
Examples.....	16
Bootstrap Navbar.....	16

Imagen de la marca Boostrap.....	16
Capítulo 6: Botones.....	18
Sintaxis.....	18
Examples.....	18
Clases de botones.....	18
Capítulo 7: Carruseles.....	21
Observaciones.....	21
Examples.....	21
Uso básico de HTML.....	21
Uso básico de Javascript e inicialización.....	22
Capítulo 8: Clases de utilidad.....	23
Examples.....	23
Generar clases .hidden- * para todos los puntos de interrupción - SCSS.....	23
Capítulo 9: Columnas.....	24
Examples.....	24
Columnas sensibles de la misma altura (solo CSS o SASS).....	24
Capítulo 10: Componentes Bootstrap.....	28
Observaciones.....	28
Examples.....	28
Ejemplos de componentes Bootstrap.....	28
Capítulo 11: Contenedores Bootstrap.....	29
Introducción.....	29
Examples.....	29
Contenedores.....	29
Capítulo 12: Diálogos modales.....	30
Observaciones.....	30
Examples.....	30
Uso básico de HTML.....	30
Uso básico de Javascript e inicialización.....	30
Capítulo 13: Formas.....	32
Examples.....	32

Forma básica.....	32
Entradas de solo lectura y deshabilitadas.....	32
Capítulo 14: Glifos.....	33
Observaciones.....	33
Examples.....	33
Cómo utilizar Glyphicons.....	33
Capítulo 15: Grupo de lista.....	35
Observaciones.....	35
Examples.....	35
Ejemplo basico.....	35
Insignias.....	35
Artículos vinculados.....	35
Elementos de botones.....	35
Artículos deshabilitados.....	36
Clases contextuales.....	36
Contenido personalizado.....	36
Capítulo 16: Imprimiendo en bootstrap.....	37
Examples.....	37
Uso básico de HTML.....	37
Capítulo 17: Información sobre herramientas.....	38
Observaciones.....	38
Examples.....	38
Herramientas de posicionamiento.....	38
Ejemplo básico.....	38
Capítulo 18: Insignias y etiquetas de Bootstrap.....	40
Examples.....	40
Insignias.....	40
Etiquetas.....	40
Capítulo 19: Jumbotron.....	41
Introducción.....	41
Observaciones.....	41
Examples.....	41

Jumbotron básico con dos líneas de texto y un botón.....	41
Capítulo 20: Listas desplegables.....	43
Observaciones.....	43
Examples.....	43
Uso básico de HTML.....	43
Capítulo 21: Los paneles.....	44
Observaciones.....	44
Examples.....	44
Ejemplo basico.....	44
Panel con encabezado.....	44
Panel con pie de página.....	44
Capítulo 22: Menús de navegación.....	46
Examples.....	46
Menú horizontal de la píldora.....	46
Menú vertical de la píldora.....	46
Píldora horizontal receptiva de ancho completo.....	46
Capítulo 23: Mesas.....	47
Examples.....	47
Mesa simple.....	47
Capítulo 24: Mesas.....	48
Observaciones.....	48
Examples.....	48
Tabla basica.....	48
Mesa con estilo avanzado.....	48
Filas de rayas.....	48
Mesa bordeada.....	49
Flotar en las filas.....	49
Mesa condensada.....	49
Clases contextuales.....	49
Tablas responsivas.....	50
Mesa de reflujo - encabezados verticales.....	50

Capítulo 25: Migración a Bootstrap 4	52
Introducción	52
Observaciones	52
Examples	52
Cambios en el diseño de la columna del sistema de cuadrícula en Bootstrap 4	52
Diseño de cuadrícula Bootstrap 4	52
Cambios en el soporte del navegador	53
Remover clase de eliminación	53
Bootstrap 4 Navbar	56
Bootstrap 3 to Bootstrap 4 Cambios CSS	59
Bootstrap 4 Vertical Align	61
Bootstrap 4 Centrado	63
Centro horizontal	63
Centro vertical	64
Bootstrap 4 Column Order	64
Capítulo 26: Modales	66
Observaciones	66
Examples	66
HTML básico modal	66
Capítulo 27: Navbar	67
Examples	67
Barra de navegación básica (fijada en la parte superior de la página)	67
Submenú en la barra de navegación	67
Divisor navbar	68
Mantener el enlace de navegación actual "activo"	68
Cambiar punto de interrupción de la barra de navegación (móvil vs normal)	68
Cerrar la barra de navegación contraída al hacer clic fuera de la barra de navegación	69
Capítulo 28: Naves	70
Examples	70
Naves Bootstrap	70
Capítulo 29: Nido de rejilla	71
Introducción	71

Observaciones.....	71
Examples.....	71
Columnas de anidación.....	71
Capítulo 30: Paginación.....	72
Introducción.....	72
Examples.....	72
Un simple ejemplo de paginación.....	72
Capítulo 31: Personalización de estilo de Bootstrap de Twitter.....	73
Observaciones.....	73
Examples.....	73
Anulación de CSS predeterminado.....	73
Capítulo 32: Pestañas.....	75
Examples.....	75
HTML básico.....	75
Pestañas animadas.....	75
Capítulo 33: Sistema de red.....	77
Introducción.....	77
Observaciones.....	77
Examples.....	77
Preguntas de los medios.....	77
Niveles de rejilla de arranque (puntos de interrupción).....	77
Bootstrap filas y columnas.....	79
Contenedores.....	80
Columnas de compensación.....	81
Manipulación del orden de las columnas mediante push and pull.....	81
Capítulo 34: Temas Bootstrap.....	83
Examples.....	83
Temas bootstrap versus anulaciones de reglas.....	83
Capítulo 35: Usando Clearfix en Filas y Cols.....	84
Introducción.....	84
Observaciones.....	84

Examples.....	84
El primer intento ingenuo.....	84
El problema de la altura.....	86
Clearfix al rescate.....	87
Un tablero de instrumentos.....	90
Disposición 2.4.6 con Clearfixes.....	95
¿Por qué las columnas de Bootstrap superan 12 en una fila?.....	97
Capítulo 36: Validación Bootstrap.....	100
Observaciones.....	100
Examples.....	100
Uso de ASP.NET MVC y anotaciones de datos.....	100
Ejemplo de entrada que requiere validación.....	101
Opcional.....	101
Creditos.....	103

Acerca de

You can share this PDF with anyone you feel could benefit from it, download 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

Capítulo 1: Empezando con el twitter-bootstrap

Observaciones

Bootstrap es un marco HTML, CSS y JavaScript que se utiliza para crear sitios web que se crean utilizando un primer paradigma móvil, así como un diseño web sensible (RWD). Utiliza una combinación de clases de CSS prefabricadas y JavaScript para hacer una variedad de cosas en la web. Incluye elementos tales como una cuadrícula sensible y personalizada que permite ver los sitios web en cualquier pantalla, una barra de navegación desplegable que es capaz de responder e incluso cosas simples que pueden requerir mucho tiempo, como botones, formularios y acordeones prefabricados para nombrar un pocos.

Bootstrap puede ser útil por las siguientes razones:

- 1. Ahorro de tiempo :** Bootstrap presenta muchas cosas que están preconfiguradas, y que simplemente necesitan ser llamadas cuando se escribe un código. Esto ahorra una cantidad considerable y puede reducir considerablemente el tiempo necesario para codificar un sitio web.
- 2. Diseñado pensando en el diseño web sensible :** Bootstrap permite que los desarrolladores web no se preocupen por crear cosas que se amplíen con el tamaño de su pantalla, ya que Bootstrap usa un diseño móvil y sensible que les permite crear cosas que funcionarán en cualquier pantalla tamaño.
- 3. Simplifica el proceso de diseño :** Bootstrap viene premontado con elementos que tienen buenas prácticas de diseño. Esto puede ser útil para aquellos cuyas habilidades de diseño web no son tan buenas, o para aquellos que ven el diseño como una tarea tediosa, ya que muchas de las clases de Bootstrap son estéticamente agradables y excelentes para mirar.

Esta sección debe mencionar cualquier tema grande dentro de twitter-bootstrap y vincular a los temas relacionados. Dado que la Documentación para twitter-bootstrap es nueva, es posible que deba crear versiones iniciales de esos temas relacionados.

Versiones

Versión	Fecha de lanzamiento
4.0	2999-01-01
3.3.7	2016-07-25
3.3.6	2015-11-24

Versión	Fecha de lanzamiento
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
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

Instalación / Configuración

Descargando:

- Descargue Bootstrap [directamente](#) o clone, etc. desde el repositorio de [GitHub](#)
- Descarga tu versión personalizada de Bootstrap desde [documentos](#) oficiales
- Instalar con bower: `bower install bootstrap`
- Instalar con npm: `npm install bootstrap`
- Instalar con el compositor: el `composer require twbs/bootstrap`

La estructura de archivos

```

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

```

Instalación:

Dentro de su página HTML, incluya CSS, JS de Bootstrap y la dependencia de jQuery (pre versión 3, al menos a partir de la última versión de Bootstrap). Tenga en cuenta que si planea utilizar las funciones de JavaScript de Bootstrap, su referencia jQuery debe aparecer *antes de* su referencia bootstrap.js dentro de su HTML.

Puede utilizar sus archivos instalados de Bootstrap de la sección anterior, o hacer referencia a un CDN provisto por los creadores de Bootstrap (enlaces tomados de [Getting Started with Bootstrap](#)):

```

<!-- Latest compiled and minified CSS -->
<link rel="stylesheet"
      href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
      integrity="sha384-BVYii3IF8eGqWzjZD2E4PcbywLrZjx5E4I655izICOYZhierYoJwIlkHvPQKXn"
      crossorigin="anonymous">

<!-- Optional theme -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap-theme.min.css" integrity="sha384-BVHyNlEJj+OggPnq4jwI6o+O7a+QfZqF6i5Kj07d+u6ZD+gkZqD6QbdEjh4u" crossorigin="anonymous">

<!-- Latest compiled and minified JavaScript -->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"
      integrity="sha384-Tc5IQib027qvysJSMFHjOMaLkfUWVxZXUPnPJA712mCWNIPG9mGCD8wGNICPD7Txa"
      crossorigin="anonymous"></script>

```

Una página web muy básica de Bootstrap:

```

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

```

```

*after* these tags -->
<title>Bootstrap 101 Template</title>

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

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

```

Plantilla Básica

```

<!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>

```

Cuando usar Bootstrap

Bootstrap es un marco de opinión para HTML, CSS y Javascript. Contiene un estilo y una funcionalidad básicos para lo que se han aceptado los elementos [Interfaz de usuario], como elementos de formulario, botones, ventanas modales y elementos de navegación.

Bootstrap es un marco web sensible, lo que significa que está diseñado para adaptar el diseño y el diseño para tamaños de pantalla grandes y pequeños, como dispositivos móviles, tabletas y computadoras de escritorio, todo en un solo código.

Uno de los conceptos fundamentales de Bootstrap es el marco de la cuadrícula. Al aplicar clases a elementos HTML, es posible crear diseños complejos utilizando una cuadrícula básica de doce columnas. Por ejemplo, un diseño de cuatro columnas podría adaptarse a dos columnas en dispositivos de tableta y una columna en dispositivos móviles. La cuadrícula utiliza `media queries`, un método CSS para apuntar a tamaños de pantalla específicos, para lograr esto.

Bootstrap funciona particularmente bien si:

- El diseño personalizado no es una prioridad
- Se siente más cómodo editando HTML y agregando clases que creando CSS personalizado
- Se siente cómodo utilizando un marco que tendrá muchas similitudes visuales con muchos otros sitios web

Bootstrap puede ser utilizado por aquellos que son nuevos en HTML, CSS y Javascript, ya que la [documentación](#) es excelente. Sin embargo, hay una curva de aprendizaje para aquellos que no están del todo cómodos con las tres tecnologías básicas utilizadas por Bootstrap (HTML, CSS y Javascript).

Es posible comprar o descargar temas de Bootstrap para modificar el estilo o la funcionalidad de Bootstrap. También es posible utilizar Bootstrap como punto de partida, con personalización de CSS y Javascript.

Página web básica utilizando componentes bootstrap.

```

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

```

```

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

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

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

```

```
<div class="container">

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

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

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

Lea Empezando con el twitter-bootstrap en línea: <https://riptutorial.com/es/twitter-bootstrap/topic/818/empezando-con-el-twitter-bootstrap>

Capítulo 2: Afijo Bootstrap

Examples

En navbar

HTML:

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

Css:

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

Ejemplo de afijo 2

```
<div class="container" id="con">  
    <div class="row">  
        <div class="span12">  
            <div class="well">  
                <h1> Header </h1>  
            </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 class="container">  
    <div class="span3">
```

```

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

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

```

Lea Afijo Bootstrap en línea: <https://riptutorial.com/es/twitter-bootstrap/topic/6639/afijo-bootstrap>

Capítulo 3: Alerta

Observaciones

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

Examples

Tipos de alerta

A diferencia de otros componentes de Bootstrap, como los [Botones](#), las [Alertas](#) **no** vienen con un estilo `default` o `primary`, ya que tienen la intención de alertar al usuario de una manera específica.

```
<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>
```

Ejemplo básico de alerta.

```
<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

Alertas animadas

Las clases `.fade` y `.in` agregan un efecto de desvanecimiento al cerrar el mensaje de alerta.

```
<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>
```

Alertas Desechables

Para dar una funcionalidad de cierre de alerta, todo lo que necesitamos es agregar `data-dismiss="alert"` a nuestro botón de cierre.

```
<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` y `.close` son opcionales, solo útiles para el estilo.

Color de enlace en Alertas

Para proporcionar rápidamente un color coincidente para los enlaces dentro de cualquier alerta,

podemos usar la clase de utilidad .alert-link .

```
<div class="alert alert-success">  
    You have won! Click <a href="#" class="alert-link">here</a> to claim your prize ...  
</div>  
  
<div class="alert alert-info">  
    You might want to check <a href="#" class="alert-link">this</a> instead.  
</div>  
  
<div class="alert alert-warning">  
    You are running out of coins. Buy more <a href="#" class="alert-link">here</a>.  
</div>  
  
<div class="alert alert-danger">  
    Something went wrong. You can try <a href="#" class="alert-link">again</a> or ...  
</div>
```

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

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

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

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

Lea Alerta en línea: <https://riptutorial.com/es/twitter-bootstrap/topic/6434/alerta>

Capítulo 4: Bootstrap Dropdowns

Parámetros

Métodos	Ejemplo
Llamar a través de Javascript	<code>\$('.dropdown-toggle').dropdown();</code>
Altera el menú desplegable	<code>\$('.dropdown-toggle').dropdown('toggle')</code>
Tipo de evento	Descripción
show.bs.dropdown	Este evento se dispara inmediatamente cuando se llama al método de instancia de show.
shown.bs.dropdown	Este evento se activa cuando el menú desplegable se ha hecho visible para el usuario (esperará a que se completen las transiciones de CSS).
hide.bs.dropdown	Este evento se activa inmediatamente cuando se llama al método de instancia de ocultación.
hidden.bs.dropdown	Este evento se activa cuando el menú desplegable ha terminado de ocultarse al usuario (esperará a que se completen las transiciones de CSS).
Ejemplo de controlador de eventos	<code>\$(element).on('show.bs.dropdown', function () { // do something... })</code>

Observaciones

Cuando se llama a Dropdown Via Javascript `$('.dropdown-toggle').dropdown()`, el data-api es decir, `data-toggle="dropdown"` aún es necesario. [Lee mas](#)

Examples

Cómo utilizar

Utilice la clase `.dropdown` en el elemento principal del menú desplegable.

Agregue la clase `.dropdown-menu` a un elemento para inicializar el complemento del menú desplegable.

Llame al complemento utilizando la clase `.dropdown-toggle` y el atributo `data-toggle="dropdown"` en un botón o un hipervínculo.

Ejemplo básico

```
<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>
```

Lea Bootstrap Dropdowns en línea: <https://riptutorial.com/es/twitter-bootstrap/topic/6361/bootstrap-dropdowns>

Capítulo 5: Bootstrap Navbar

Examples

Bootstrap Navbar

Este es un ejemplo de la barra de navegación de Bootstrap versión 3:

```
<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>
```

Imagen de la marca Boostrap

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

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

Lea Bootstrap Navbar en línea: <https://riptutorial.com/es/twitter-bootstrap/topic/6233/bootstrap-navbar>

Capítulo 6: Botones

Sintaxis

- Clases: .btn-default | .btn-primario | .btn-éxito | .btn-info | .btn-advertencia | .btn-danger | .btn-link;
- Tamaños: .btn-lg | .btn-md | .btn-sm | .btn-xs;
- Estado: activo | incapacitado

Examples

Clases de botones

Bootstrap proporciona varias clases para diseñar botones y hacer que se destaquen.

Los botones de arranque pueden crearse agregando la clase `.btn` a un elemento.

Clase bootstrap	Rol (color)
.btn-default	Botón estándar (blanco)
.btn-primary	Proporciona peso visual adicional e identifica la acción primaria (azul)
.btn-success	Se utiliza para indicar una acción exitosa (verde)
.btn-info	Botón contextual para proporcionar información (azul claro)
.btn-warning	Indica precaución que debe ser aplicada por el usuario (amarillo)
.btn-danger	Indica una acción peligrosa o negativa (rojo)
.btn-link	Haz que tu botón parezca una etiqueta de ancla.

Tamaños de botones

También puede crear diferentes tamaños de botones con las clases de `.btn-size`

Clase bootstrap	Resultado
.btn-lg	Crea un botón de mayor tamaño.
.btn-sm	Crea un botón de tamaño más pequeño.
.btn-xs	Crea un botón extra-pequeño.
.btn-block	Los botones se convierten en elementos a nivel de bloque y abarcan todo

Clase bootstrap	Resultado
	el ancho de su padre

Hacer el botón activo

La clase `active` hará que un botón aparezca presionado.

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

Deshabilitar un botón

Agregar la clase `disabled` a un botón hará que el botón no se pueda hacer clic y mostrará un cursor prohibido al pasar sobre él.

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

Renderizar botones horizontalmente juntos

Se pueden representar varios botones horizontalmente con la clase `.btn-group`. Simplemente envuelva sus botones dentro de un elemento contenedor y asigne a ese elemento la clase `btn-group`.

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

Renderizar botones verticalmente

Aplicar la `.btn-group-vertical` al elemento contenedor

```
<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>
```

Hacer que el grupo de botones ocupe todo el ancho

Los botones envueltos dentro de un elemento `.btn-group` solo ocupan el ancho necesario. Para hacer que el grupo abarque todo el ancho de la pantalla, use `.btn-group-justified` lugar.

```
<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>
```

Lea Botones en línea: <https://riptutorial.com/es/twitter-bootstrap/topic/4757/botones>

Capítulo 7: Carruseles

Observaciones

Para obtener más información, visite la documentación oficial en <http://getbootstrap.com/javascript/#carousel>, donde se derivan ejemplos e información de uso de HTML básico y Javascript.

Cabe señalar que los carruseles no funcionan correctamente en IE 9 y anteriores debido al uso de transiciones / animaciones CSS3.

Examples

Uso básico de HTML

Un carrusel Bootstrap es un componente de Bootstrap que crea una presentación de diapositivas que recorre los elementos dentro del carrusel.

Aquí hay un ejemplo de uso de HTML básico:

```
<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>
```

Uso básico de Javascript e inicialización.

Los componentes del carrusel se pueden instanciar a través de jQuery con la función `$('.carousel').carousel(options)`, donde `$('.carousel')` es una referencia de nivel superior al carrusel específico y `options` es un objeto de Javascript que especifica el carrusel atributos por defecto.

El objeto de `options` permite que se definan múltiples propiedades, lo que afectará el comportamiento del carrusel. Estas propiedades se definen como tales:

- La propiedad de `interval` acepta un tipo de `number` Javascript que permite a un usuario definir la cantidad de tiempo durante el cual el carrusel muestra una diapositiva de carrusel determinada. Si se especifica el valor booleano `false`, el carrusel no realizará un ciclo automáticamente.
- La propiedad de `pause` acepta un tipo de `string` Javascript que alterna el comportamiento donde el ciclo automático del carrusel se detiene cuando el mouse del usuario ingresa al carrusel. El valor predeterminado (y único) aceptado es "flotar".
- La propiedad de `wrap` acepta un tipo `boolean` Javascript que permite a un usuario definir si desea que el carrusel realice un ciclo continuo sin detenerse en una diapositiva determinada.
- La propiedad del `keyboard` acepta un tipo `boolean` Javascript que permite a un usuario definir si desea que el carrusel responda o no a los eventos del teclado.

Aquí hay un ejemplo del uso básico de Javascript:

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

Al igual que con otros componentes de Bootstrap, las opciones del carrusel también se pueden especificar en HTML a través de atributos de datos.

Lea Carruseles en línea: <https://riptutorial.com/es/twitter-bootstrap/topic/1568/carruseles>

Capítulo 8: Clases de utilidad

Examples

Generar clases .hidden-* para todos los puntos de interrupción - 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();
```

Lea Clases de utilidad en línea: <https://riptutorial.com/es/twitter-bootstrap/topic/6217/clases-de-utilidad>

Capítulo 9: Columnas

Examples

Columnas sensibles de la misma altura (solo CSS o SASS)

.row-height agregar un div con la clase .row-height dentro de la fila y también agregar .col-height a las columnas. Si desea restringir el efecto a una determinada consulta de medios, solo use las .row-height responsivas de .row-height y .col-height : por ejemplo .row-sm-height con .col-sm-height .

Versión CSS:

```
.row-height {  
  display: table;  
  table-layout: fixed;  
  height: 100%;  
  width: calc(100% + 30px);  
}  
.col-height {  
  display: table-cell;  
  float: none;  
  height: 100%;  
}  
.col-top {  
  vertical-align: top;  
}  
.col-middle {  
  vertical-align: middle;  
}  
.col-bottom {  
  vertical-align: bottom;  
}  
  
@media (min-width: 480px) {  
  .row-xs-height {  
    display: table;  
    table-layout: fixed;  
    height: 100%;  
    width: 100%;  
  }  
  .col-xs-height {  
    display: table-cell;  
    float: none;  
    height: 100%;  
  }  
  .col-xs-top {  
    vertical-align: top;  
  }  
  .col-xs-middle {  
    vertical-align: middle;  
  }  
  .col-xs-bottom {  
    vertical-align: bottom;  
  }  
}
```

```

}

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

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

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

```

```

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

```

Versión de SASS (se necesita bootstrap _variables.scss):

```

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

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

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

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

```

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

Lea Columnas en línea: <https://riptutorial.com/es/twitter-bootstrap/topic/6469/columnas>

Capítulo 10: Componentes Bootstrap

Observaciones

Para obtener más información, visite la documentación oficial que se encuentra en <http://getbootstrap.com/javascript/>, donde se deriva la lista de componentes.

Examples

Ejemplos de componentes Bootstrap

Los componentes de Bootstrap son una colección de complementos de jQuery opcionales que se incluyen con Bootstrap.

El propósito de los componentes de Bootstrap es proporcionar características y capacidades extendidas que serían difíciles (o imposibles) de lograr sin el uso de Javascript. Algunos componentes son puramente funcionales, mientras que algunos componentes se usan para definir la funcionalidad de algunos de los widgets frontales especiales de Bootstrap.

Los ejemplos incluyen [efectos de transición](#) , [cuadros de diálogo modales](#) , [menús desplegables](#) , [scrollspy](#) , [pestañas](#) , [información sobre herramientas](#) , [alertas](#) , [popovers](#) , [botones](#) , [colapso](#) , [carruseles](#) , y [afijos](#) .

Lea Componentes Bootstrap en línea: <https://riptutorial.com/es/twitter-bootstrap/topic/6054/componentes-bootstrap>

Capítulo 11: Contenedores Bootstrap

Introducción

Utilice .container para un contenedor de ancho fijo sensible.

Use .container-fluid para un contenedor de ancho completo, que abarque todo el ancho de la ventana gráfica.

Examples

Contenedores

.container tiene un ancho fijo para cada tamaño de pantalla en bootstrap (xs, sm, md, lg);

El líquido del contenedor se expande para llenar el ancho disponible.

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

Dependiendo del ancho de la ventana gráfica en la que se está viendo la página web, la clase contenedora le da a su div un ancho fijo específico.

Por otro lado, su elemento .contener-fluid, cambiará de tamaño constantemente a medida que realice los cambios más pequeños en el ancho de su navegador.

Lea Contenedores Bootstrap en línea: <https://riptutorial.com/es/twitter-bootstrap/topic/10908/contenedores-bootstrap>

Capítulo 12: Diálogos modales

Observaciones

Para obtener más información, visite la documentación oficial en <http://getbootstrap.com/javascript/#modals>, donde se deriva el ejemplo de 'Uso básico de HTML'.

Examples

Uso básico de HTML

Un diálogo modal de Bootstrap es un componente Bootstrap que crea una ventana de diálogo modal que flota sobre el contenido a nivel de la página.

Aquí hay un ejemplo del uso básico de un diálogo modal de Bootstrap en HTML:

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

Uso básico de Javascript e inicialización.

Los componentes del diálogo modal se pueden crear una instancia a través de jQuery con la función `$('#myModal').modal(options)`, donde `$('#myModal')` es una referencia de nivel superior al diálogo modal específico y las `options` son un objeto de Javascript que especifica los atributos por defecto del diálogo modal.

El objeto de `options` permite que se definan múltiples propiedades, lo que afectará el comportamiento del diálogo modal. Estas propiedades se definen como tales:

- La propiedad de `backdrop` permite al usuario definir si desea que aparezca una superposición de fondo gris detrás del cuadro de diálogo modal. Se reconocen tanto los valores booleanos como la cadena "estática". Si se especifica "estático", el diálogo modal no se cerrará cuando un usuario haga clic en la superposición de fondo.

- La propiedad del `keyboard` permite al usuario definir si desea que el diálogo modal se cierre cuando se presiona la tecla Escape en el teclado.
- La propiedad `show` permite al usuario definir si desea que aparezca el diálogo modal cuando se inicializa el modal.

Aquí hay un ejemplo del uso básico de Javascript:

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

Al igual que con otros componentes de Bootstrap, las opciones modales también pueden especificarse en HTML a través de atributos de datos.

Lea Diálogos modales en línea: <https://riptutorial.com/es/twitter-bootstrap/topic/5927/dialogos-modales>

Capítulo 13: Formas

Examples

Forma básica

Los controles de formulario tienen un estilo predeterminado sin usar ninguna clase especial.

Sin embargo, las etiquetas y los controles se pueden envolver en etiquetas `.form-group` para un espaciado óptimo.

```
<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>
```

Entradas de solo lectura y deshabilitadas

Agregue el atributo de `readonly` para evitar la entrada del usuario. Un campo de solo lectura no puede ser editado

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

Añadir el `disabled` atributo para desactivar un campo de entrada. Un campo inhabilitado tampoco puede ser editado. El cursor cambia para hacerlo más notable.

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

Lea Formas en línea: <https://riptutorial.com/es/twitter-bootstrap/topic/6251/formas>

Capítulo 14: Glifos

Observaciones

Esta sección proporciona una descripción general de los glifos Bootstrap y describe cómo usar los glifos.

Examples

Cómo utilizar Glyphicons

Twitter Bootstrap admite íconos llamados glyphicons y se pueden usar con todas las etiquetas de HTML.

Todos los iconos requieren una clase base y una clase de ícono individual.

Tenga en cuenta que las clases de íconos no se pueden combinar directamente con otros componentes, así que siempre use la etiqueta `` interna.

Si su código HTML tiene elementos secundarios internos, no podrá utilizar las clases de íconos para esa etiqueta en particular.

Ejemplos

Por ejemplo, si está creando un botón de arranque, entonces la sintaxis de este botón debería ser así:

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

Entonces, en el ejemplo anterior, se crea un simple botón de arranque, pero ahora desea agregar un glyphicon en este botón, para esto simplemente agregue un elemento `` dentro de una etiqueta `<button>`. Me gusta esto:

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

 Star

 Star

 Star

 Star

```
<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>
```

Lea Glifos en línea: <https://riptutorial.com/es/twitter-bootstrap/topic/6098/glifos>

Capítulo 15: Grupo de lista

Observaciones

Debe saber cómo usar los [botones de arranque](#) y poca información sobre las clases [contextuales](#)

Examples

Ejemplo basico

```
<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>
```

Insignias

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

Artículos vinculados

```
<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>
```

Elementos de botones

```
<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>
```

Artículos deshabilitados

```
<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>
```

Clases contextuales

```
<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>
```

Contenido personalizado

```
<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>
```

Lea Grupo de lista en línea: <https://riptutorial.com/es/twitter-bootstrap/topic/6347/grupo-de-lista>

Capítulo 16: Imprimiendo en bootstrap.

Examples

Uso básico de HTML

Los elementos de impresión de Bootstrap le permiten designar qué elementos deben estar visibles cuando se imprimen y cuáles deben estar ocultos.

Para hacer que algo esté visible, utilice una de las siguientes opciones, según el elemento y cómo debe aparecer cuando se imprima:

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

Para ocultar algo de la impresión, use lo siguiente:

```
.hidden-print
```

Lea [Imprimiendo en bootstrap](https://riptutorial.com/es/twitter-bootstrap/topic/6707/imprimiendo-en-bootstrap). en línea: <https://riptutorial.com/es/twitter-bootstrap/topic/6707/imprimiendo-en-bootstrap>

Capítulo 17: Información sobre herramientas

Observaciones

La información sobre herramientas es un elemento de la interfaz de usuario que parece un pequeño cuadro emergente. Por lo general, se activa cuando un usuario coloca su puntero sobre otro elemento, sin hacer clic en él.

Por razones de rendimiento, la información sobre herramientas debe inicializarse con jQuery. El siguiente código habilitará todas las informaciones sobre herramientas en el DOM:

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

Examples

Herramientas de posicionamiento

Por defecto, la información sobre herramientas aparecerá en la parte superior del elemento. Podemos usar el atributo de `data-placement` para establecer la posición de la información sobre herramientas en la parte superior, inferior, izquierda o derecha del elemento.

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

Hover Hover Hover Hover **Right tooltip**

También podemos usar `data-placement="auto"`, para reorientar dinámicamente la información sobre herramientas. La información sobre herramientas en el siguiente ejemplo se mostrará a la izquierda cuando sea posible, de lo contrario se mostrará a la derecha.

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

Ejemplo básico

Para crear una información sobre herramientas, solo necesitamos agregar el atributo `data-toggle="tooltip"` y un `title` al elemento HTML que tendrá la información sobre herramientas. El atributo de título se usa para especificar el texto que se muestra dentro de la información sobre herramientas.

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

Hello world!

Hover over me

Lea Información sobre herramientas en línea: <https://riptutorial.com/es/twitter-bootstrap/topic/3731/informacion-sobre-herramientas>

Capítulo 18: Insignias y etiquetas de Bootstrap

Examples

Insignias

Las insignias son indicadores numéricos de cuántos elementos están asociados con un enlace:

Use la clase `.badge` dentro de los elementos `` para crear distintivos:

```
<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>
```

Insignia en el botón

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

Etiquetas

Las etiquetas se utilizan para proporcionar información adicional sobre algo:

Utilice la clase `.label` , seguida de una de las seis clases contextuales `.label-default` , `.label-primary` , `.label-success` , `.label-info` , `.label-warning` o `.label-danger` , dentro de un elemento `` para crear una etiqueta:

```
<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>
```

Lea Insignias y etiquetas de Bootstrap en línea: <https://riptutorial.com/es/twitter-bootstrap/topic/7867/insignias-y-etiquetas-de-bootstrap>

Capítulo 19: Jumbotron

Introducción

Jumbotron es un componente estándar de Bootstrap para mostrar algunos contenidos importantes en su sitio web. Normalmente se usa justo debajo de la barra de navegación, antes del contenido.

Observaciones

Dentro del jumbotron, todo el sistema de cuadrícula, clase de contenedor y clase de fila también funciona.

Examples

Jumbotron básico con dos líneas de texto y un botón.

Este es un jumbotron con un título, un contenido y un botón.

Código

```
<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>
```

Resultado

Title text

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Donec tortor ipsum, convallis sit.

A button

Lea Jumbotron en línea: <https://riptutorial.com/es/twitter-bootstrap/topic/9188/jumbotron>

Capítulo 20: Listas desplegables

Observaciones

Para obtener más información, visite la documentación oficial de Bootstrap que se encuentra en <http://getbootstrap.com/javascript/#dropdowns>, donde se deriva el ejemplo de uso de HTML básico.

Examples

Uso básico de HTML

Un menú desplegable Bootstrap es un componente Bootstrap que permite que un elemento HTML active la visualización de un menú desplegable de submenú al hacer clic en el elemento.

Aquí hay un ejemplo de uso de HTML básico:

```
<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>
```

Los elementos del submenú desplegable pueden especificarse insertando `li` elementos dentro del elemento `ul` con la clase `.dropdown-menu`.

Lea Listas desplegables en línea: <https://riptutorial.com/es/twitter-bootstrap/topic/6033/listas-desplegables>

Capítulo 21: Los paneles

Observaciones

El componente del panel en bootstrap es un cuadro (bordeado) con algo de relleno alrededor de su contenido y, opcionalmente, contenedores de encabezado y pie de página.

Examples

Ejemplo basico

De forma predeterminada, todo lo que hace el `.panel` es aplicar algunos bordes y rellenos básicos para contener algo de contenido.

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

Panel con encabezado

Agregue fácilmente un contenedor de encabezado a su panel con `.panel-heading`. También puede incluir cualquier `<h1>-<h6>` con una clase `.panel-title` para agregar un encabezado de estilo anterior. Sin embargo, los tamaños de fuente de `<h1>-<h6>` se `.panel-heading`.

Para colorear correctamente los enlaces, asegúrese de colocar enlaces en los encabezados dentro de `.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>
```

Panel con pie de página

Ajustar botones o texto secundario en `.panel-footer`. Tenga en cuenta que los pies de página del

panel **no** heredan los colores y los bordes cuando se usan variaciones contextuales, ya que no están destinados a estar en primer plano.

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

Lea Los paneles en línea: <https://riptutorial.com/es/twitter-bootstrap/topic/2848/los-paneles>

Capítulo 22: Menús de navegación

Examples

Menú horizontal de la píldora

```
<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>
```

Menú vertical de la píldora

```
<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>
```

Píldora horizontal receptiva de ancho completo

```
<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>
```

Lea Menús de navegación en línea: <https://riptutorial.com/es/twitter-bootstrap/topic/6404/menus-de-navegacion>

Capítulo 23: Mesas

Examples

Mesa simple

Si bien los efectos de estilo pueden variar según el tema, la clase .table se utiliza para crear una apariencia uniforme y consistente para las tablas en una aplicación:

```
<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>
```

Lea Mesas en línea: <https://riptutorial.com/es/twitter-bootstrap/topic/6299/mesas>

Capítulo 24: Mesas

Observaciones

Orden de contenido y tablas complejas Tenga en cuenta que el estilo de reflujo de tabla cambia el orden visual del contenido. Asegúrese de que solo aplique este estilo a tablas de datos simples y **bien formadas** (y en particular, no use esto para tablas de diseño) con celdas de encabezado de tabla adecuadas para cada fila y columna.

Además, esta clase no funcionará correctamente para tablas con celdas que abarcan varias filas o columnas (utilizando los atributos rowspan o colspan).

Examples

Tabla basica

Bootstrap define un estilo personalizado para la tabla utilizando la clase `.table`. Simplemente agregue la clase `.table` a cualquier `<table>` para ver los separadores horizontales y el relleno:

```
<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>
```

Mesa con estilo avanzado.

Bootstrap proporciona un par de clases para el estilo avanzado de tablas.

Filas de rayas

Tendrá una tabla con filas de rayas, si agrega la clase `.table-striped`:

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

Tenga en cuenta que:

Las tablas con rayas se diseñan a través del selector de CSS `:nth-child`, que no está

disponible en Internet Explorer 8.

Mesa bordeada

Tendrá una tabla con bordes en todos los lados de la tabla y las celdas, si agrega la clase `.table-bordered`:

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

Flotar en las filas

Si agrega la clase `.table-hover`, tendrá una tabla con filas resaltadas cuando el usuario se desplace sobre una fila:

```
<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>
```

Mesa condensada

Si agrega `.table-condensed` clase `.table-condensed`, el relleno de celda predeterminado se reducirá a la mitad, por lo que tendrá una tabla más compacta:

```
<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>
```

Clases contextuales

Las tablas bootstrap soportan colores contextuales. Para cambiar el color de fondo de una fila o celda de una tabla, solo tiene que agregar una de las siguientes clases `.active` : `.active` , `.success` , `.info` , `.warning` , `.danger`

```

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

```

Tablas responsivas

Tiene que envolver cualquier `.table` en el contenedor `html` con la clase `.table-responsive` para crear tablas responsivas:

```

<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>

```

Las tablas receptivas se desplazarán horizontalmente en dispositivos pequeños (<768px). No habrá diferencias para pantallas de más de 768px de ancho.

Mesa de reflujo - encabezados verticales

Conseguir una mesa con encabezados verticales.

El bootstrap de Twitter ahora admite el encabezado vertical en una tabla normal bien formateada. Para lograr esto solo usa la clase `.table-reflow`

Use la clase bootstrap `.table-reflow` twitter en una tabla bien formada para lograr una tabla con encabezados verticales. Además, puede combinar con el uso de `.table-striped` y `.table-hover` para `.table-hover` en las columnas esta vez.

```

<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>
    </tr>
  </tbody>
</table>

```

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

Debes revisar los documentos alfa de v4 aquí: [twitter-bootstrap .table-reflow](#)

Lea Mesas en línea: <https://riptutorial.com/es/twitter-bootstrap/topic/6360/mesas>

Capítulo 25: Migración a Bootstrap 4

Introducción

Bootstrap 4 es una [reescritura importante](#) y hay muchos cambios que debe tener en cuenta al actualizar desde Bootstrap 3. Estos son los cambios de nombre de clase, consejos y ejemplos de cómo migrar su código Bootstrap 3.x a Bootstrap 4.x.

Observaciones

Este es solo un pequeño ejemplo de ejemplos más detallados a seguir.

Examples

Cambios en el diseño de la columna del sistema de cuadrícula en Bootstrap 4

El primer bloque de código se escribe en Bootstrap 3. En Bootstrap 3 hay 4 tipos de especificaciones de columna, a saber, `col-md-* col-lg-* col-sm-* col-xs-*`. Un diseño totalmente sensible se verá así en Bootstrap 3:

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

En Bootstrap 4, agregaron un nuevo nivel de cuadrícula sm por debajo de 768 px para un control más granular. Así que Bootstrap 4 tiene `col-*` (xs), `col-sm-*`, `col-md-*`, `col-lg-*`, and `col-xl-*`. Entonces, lo que solía ser `.col-md-6` en v3 ahora es `.col-lg-6` en v4. Observe que el prefijo `-xs` se ha eliminado, por lo que `.col-6` representa 6 unidades de columna en el punto de interrupción extra pequeño (predeterminado).

Entonces, si ahora queremos escribir el mismo ejemplo anterior en Bootstrap 4, se vería así:

```
<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>
```

Diseño de cuadrícula Bootstrap 4

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

Cambios en el soporte del navegador

En twitter-bootstrap 4 se ha eliminado el soporte para IE8, IE9, and iOS 6. v4 ahora es solo IE10+ and iOS 7+. Para sitios que necesiten cualquiera de esos, use v3.

En twitter-bootstrap 4 se agregó el soporte oficial para Android v5.0 Lollipop's Browser and WebView. Las versiones anteriores del Navegador de Android y WebView solo son unofficially supported.

Remover clase de eliminación

El afijo se elimina de Bootstrap 4.

Se recomienda usar una `position: sticky` lugar de un relleno de polietileno `position: sticky`.

Si estaba usando el Afijo para aplicar estilos adicionales, no de posición, los polyfills podrían no ser compatibles con su caso de uso. Una opción para tales usos es la biblioteca ScrollPos-Styler de terceros.

Según la documentación de Bootstrap

Se eliminó el complemento Affix jQuery. Recomendamos el uso de una posición: polietileno adhesivo en su lugar. Consulte la entrada de Por favor en HTML5 para obtener detalles y recomendaciones específicas de polyfill.

Si estaba usando el Afijo para aplicar estilos adicionales, no de posición, los polyfills podrían no ser compatibles con su caso de uso. Una opción para tales usos es la biblioteca ScrollPos-Styler de terceros.

Si alguien está migrando de Bootstrap v3 a Bootstrap v4 el enfoque alternativo:

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 Echo Park Godard kogi, next level irony ennui twee squid fap selvage. Meggings flannel Brooklyn literally small batch, mumblecore PBR try-hard kale chips. Brooklyn vinyl lumbersexual bicycle rights, viral fap cronut leggings squid chillwave pickled gentrify mustache. 3 wolf moon hashtag church-key Odd Future. Austin messenger bag normcore, Helvetica Williamsburg sartorial tote bag distillery Portland before they sold out gastropub taxidermy Vice.</p>
        </div>
        <div class="col-xs-6 col-md-3">
            <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis pharetra codeply varius quam sit amet vulputate. Quisque mauris augue, molestie tincidunt codeply condimentum
```

```

vitae, gravida a libero. Aenean sit amet felis dolor, in sagittis nisi.
    Sed ac orci quis tortor imperdiet venenatis. Duis elementum auctor accumsan.
Aliquam in felis sit amet augue.
</p>
<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>

```

CSS

```

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

```

JAVASCRIPT

```

$(document).ready(function() {

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

```

```

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

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

};

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

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

Bootstrap 4 Navbar

El nuevo componente Bootstrap 4 Navbar Component se ha mejorado con respecto a su predecesor Bootstrap 3.x. En Bootstrap 4, la barra de navegación **responde de manera predeterminada** y utiliza **flexbox** para facilitar la alineación del contenido de la barra de navegación. También es una cuestión simple de usar las nuevas clases `navbar-toggleable-*` para cambiar el punto de interrupción de Navbar. Ahora la barra de navegación tiene 6 tamaños de punto de interrupción o "estados" para que pueda tener fácilmente una de las siguientes opciones de la barra de navegación.

- La barra de navegación *nunca* se colapsa en la vista móvil vertical, y siempre es horizontal.
- La barra de navegación *siempre* se contrae en la vista vertical y se alterna a través de la hamburguesa.
- La barra de navegación se colapsa en vista vertical en *uno de los 4 puntos de interrupción de respuesta*.

Basic Bootstrap 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>

```

Como puede ver en el código de arriba, la `navbar-header` se ha eliminado de Bootstrap 4 y ya no se requiere el `container-fluid` del `container-fluid` para una barra de navegación de ancho completo.

Cambiar el punto de interrupción de la barra de navegación

La `navbar-toggleable-md` hace que la barra de navegación de arriba se contraiga verticalmente (y muestre el icono de alternar) en el punto de interrupción medio (md) de 992px. Para cambiar esto a un punto de interrupción diferente, solo tendríamos que intercambiar `navbar-toggleable-md` con uno de estos ...

- `navbar-toggleable` = contraer en anchos `xs` <576px
- `navbar-toggleable-sm` = colapso en anchos `sm` <768px
- `navbar-toggleable-lg` = colapso en anchos de `lg` <1200px

[Bootstrap 4 Breakpoint Navbar Demo](#)

Cambiar la alineación de la barra de navegación

Flexbox nos permite cambiar fácilmente la alineación de la barra de navegación y su contenido (marca, enlaces, formularios o texto). El contenido predeterminado de la barra de navegación se alinea a la izquierda. Por supuesto que hay muchos otros escenarios de alineación ...

- Marca izquierda (por defecto), enlaces centro y derecha
- Centro de marca, enlaces izquierda y derecha.
- Marca izquierda y enlaces a la derecha.
- Marca, enlaces y ancho de relleno.
- Sin marca, centro de enlaces y derecha.
- Marca izquierda, enlaces a la derecha dentro del contenedor.
- Enlaces justificados (ancho de relleno) centrados

Bootstrap 4 Navbar con marca centrada, y enlaces de izquierda / derecha

```

<nav class="navbar navbar-toggleable-sm navbar-inverse bg-primary">
  <button class="navbar-toggler navbar-toggler-right" type="button" data-toggle="collapse"
  data-target=".dual-collapse">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="navbar-collapse collapse dual-collapse">

```

```

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

```

Bootstrap 4 Navbar con marca izquierda, centro de enlaces y derecha

```

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

Demostraciones de alineación de la barra de navegación:

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

Más sobre el Bootstrap 4 Navbar

[Personalización del color, alineación o altura](#)

Bootstrap 3 to Bootstrap 4 Cambios CSS

Dado que Bootstrap 4 es una reescritura **importante**, muchos de los nombres de clase de Bootstrap 3.x han cambiado o se han eliminado. La reestructuración de componentes, como la barra de navegación, y la introducción de nuevas clases de CSS y el soporte de Flexbox significa que la actualización a 4.x *no* es un proceso de conversión simple desde 3.x.

Sin embargo, hay algunas clases CSS de Bootstrap 3.x que tienen un reemplazo específico de Bootstrap 4.

Cambios de nombre / selector de clase CSS de Bootstrap 3.3.7 a 4 (alfa 6)

{t}: representa un **nivel** o punto de interrupción (es decir, sm, md, lg, etc.).

xs nivel xs es el predeterminado, y no necesita ser especificado: col-3 , col-6 , etc.

{u} - representa un tamaño de **unidad de col** (es decir: 1-12)

Bootstrap 3.x	Bootstrap 4
.col- {t} - {u}	.col- {t} - {u} (leave {t} blank for xs)
.col- {t} -offset- {u}	.offset- {t} - {u} (leave {t} blank for xs)
.col- {t} -push- {u}	.push- {t} - {u} (leave {t} blank for xs)
.col- {t} -pull- {u}	.pull- {t} - {u} (leave {t} blank for xs)
.panel	.tarjeta
.panel-título	.card-header
título .panel	.card-titulo
.panel-cuerpo	.card-block
.panel-pie de página	.card-footer
.panel-primario	.card-primary.card-inverso
.panel-exito	.card-success.card-inverso

Bootstrap 3.x	Bootstrap 4
.panel-info	.card-info.card-inverse
.panel-aviso	.card-warning.card-inverso
.panel-peligro	.card-danger.card-inverso
.bien	.card.card-block
.miniatura	.card.card-block
.list-inline> li	.list-inline-item
menú desplegable> li	ítem desplegable
.nav navbar> li	.nav-item
.nav navbar> li> a	.nav-link
.navbar-derecha	.ml-auto
.navbar-btn	.nav-item
.navbar-fixed-top	.fixed-top
.nav-stacked	.flex-column
.btn-predeterminado	.btn-secundaria
.img-responsive	.img-fluid
.img-circulo	círculo redondo
.estoy castigado	.redondeado
.form-horizontal	(remoto)
.radio	.form-check
.caja	.form-check
.input-lg	.form-control-lg
.input-sm	.form-control-sm
.control-label	.form-control-label
mesa-condensada	.table-sm
.paginación> li	.page-item

Bootstrap 3.x	Bootstrap 4
.paginación> li> a	.page-link
.ít	.carousel-item
.text-help	.form-control-feedback
.pull-right	.flotar derecho
.pulgar a la izquierda	.flotador izquierdo
.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
.etiqueta	.placa
.placa	.badge.badge-pill

Ver también:

[Bootstrap 3.x to 4 Migration Tool](#)

[What's New in Bootstrap 4](#)

Bootstrap 4 Vertical Align

Elementos que consiguen a **centro** o Alinear **abajo** verticalmente siempre ha sido un reto con CSS y Bootstrap. La alineación vertical deseada puede estar dentro de un contenedor principal, o en relación con elementos adyacentes.

Ahora que Bootstrap 4 es **Flexbox por defecto**, hay muchos enfoques diferentes para la alineación vertical utilizando: [márgenes automáticos](#) , [Utilidades de Flexbox](#) o [Utilidades de pantalla](#) junto con [Utilidades de alineación vertical](#) .

Al principio, las [Utilidades de alineación vertical](#) parecerían una opción obvia, pero estas *solo* funcionan con elementos de visualización en línea y de tabla. Aquí hay algunas opciones y

1 - Centro vertical utilizando márgenes automáticos:

Una forma de centrar verticalmente es usar `my-auto`. Esto centrará el elemento dentro de su contenedor. Por ejemplo, `h-100` hace que la fila tenga su altura completa, y `my-auto` centrará verticalmente la `col-sm-12`.

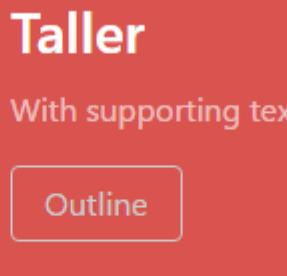
```
<div class="row h-100">
  <div class="col-sm-12 my-auto">
    <div class="card card-block w-25">Card</div>
  </div>
</div>
```

Centro vertical usando la [demo de Auto Margins](#)

`my-auto` representa márgenes en el eje vertical y es equivalente a:

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

2 - Centro vertical con Flexbox:



Ya que Bootstrap 4 `.row` está ahora en `display:flex`, simplemente puede usar `align-self-center` en cualquier columna para centrarlo verticalmente ...

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

o, use `align-items-center` en toda la `.row` para alinear verticalmente todos los `col-*` en la fila ...

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

Centro vertical Demostración de diferentes columnas de altura

3 - Centro vertical usando Utilidades de pantalla:

Bootstrap 4 tiene **utilidades de visualización** que se pueden usar para `display:table`, `display:table-cell`, `display:inline`, etc. Se pueden usar con las **utilidades de alineación vertical** para alinear elementos de celda en línea, bloque en línea o tabla.

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

Centro vertical usando la demostración de utilidades de pantalla

Bootstrap 4 Centrado

Cómo centrar un elemento, columna o contenido dentro de una columna funciona de manera diferente en Bootstrap 4.

Centro horizontal

- `text-center` todavía se utiliza para la `display:inline` elementos en `display:inline`
- `mx-auto` reemplaza `center-block` central al centro de la `display:block` elementos de `display:block`
- `offset-* 0 mx-auto` se puede usar para centrar las columnas de la cuadrícula

`mx-auto` (los márgenes del eje x automáticos) centrarán la `display:block` o `display:flex` elementos `display:flex` que tienen un *ancho definido*, (`%` , `vw` , `px` , etc.). **Flexbox se utiliza de forma predeterminada** en las columnas de la cuadrícula, por lo que también hay varios métodos de centrado de flexbox.

Centro de texto o elementos en línea: text-center

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

display: block **central** display: block O display: flex : mx-auto

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

offset-* **columnas usando compensaciones:** offset-*

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

Las columnas *también* se pueden centrar con: mx-auto

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

[Demo Bootstrap 4 Centrado horizontal](#)

Centro vertical

Para el centrado vertical en Bootstrap 4 (eje y), consulte la documentación en: [Alineamiento vertical de Bootstrap 4](#)

Bootstrap 4 Column Order

Cambiar el orden (o la posición) fue posible en Bootstrap 3 usando las clases push-pull. En Bootstrap 4, las clases **push-pull** siguen funcionando y, además, se puede utilizar el **orden flexbox**.

En Bootstrap 4, las clases **push- pull** ahora son `push-{viewport}-{units}` y `pull-{viewport}-{units}` y se ha eliminado el infix `xs-`. Considere este ejemplo que cambia el orden de las columnas al diseño 1-3-2 en `xs` y `sm`:

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

Bootstrap 4 Push Pull Demo

Como la nueva versión 4 es flexbox, **otra opción** es usar las clases de la **utilidad flexbox** para cambiar el orden de las columnas. Ahora todo el ancho, 12 unidades `col-*-12` columnas se pueden revertir usando el ordenamiento de **flexbox** .

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

Demostración de pedidos de Flexbox

Lea Migración a Bootstrap 4 en línea: <https://riptutorial.com/es/twitter-bootstrap/topic/9090/migracion-a-bootstrap-4>

Capítulo 26: Modales

Observaciones

Los modales requieren que bootstrap.min.js funcione correctamente.

Más detalles se pueden encontrar aquí: <http://getbootstrap.com/javascript/#modals>

Examples

HTML básico modal

Un modal es una ventana de diálogo que se puede mostrar en la página actual.

```
<!-- 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>
```

Lea Modales en línea: <https://riptutorial.com/es/twitter-bootstrap/topic/6320/modales>

Capítulo 27: Navbar

Examples

Barra de navegación básica (fijada en la parte superior de la página)

```
<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>
```

Submenú en la barra de navegación

```
<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>
                </li>
            </ul>
        </div>
    </div>
</div>
```

```

        <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>

```

Divisor navbar

```

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

```

Mantener el enlace de navegación actual "activo"

```

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

```

Cambiar punto de interrupción de la barra de navegación (móvil vs normal)

max-width es el punto de ruptura

```

@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;
    }
}
}

```

Cerrar la barra de navegación contraída al hacer clic fuera de la barra de navegación

```

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');
        }
    }
});

```

Lea Navbar en línea: <https://riptutorial.com/es/twitter-bootstrap/topic/2267/navbar>

Capítulo 28: Naves

Examples

Naves Bootstrap

Los navegadores disponibles en Bootstrap tienen un marcado compartido, comenzando con la clase base .nav, así como también los estados compartidos. Intercambiar clases modificadoras para cambiar entre cada estilo.

Pestañas

```
<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>
```

Pastillas

```
<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>
```

Justificado

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

Con dropdowns

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

Lea Naves en línea: <https://riptutorial.com/es/twitter-bootstrap/topic/6505/naves>

Capítulo 29: Nido de rejilla

Introducción

En Bootstrap es posible usar columnas de cuadrícula *dentro* de otras columnas. Esto es útil cuando se crean diseños de respuesta avanzados que utilizan [múltiples niveles de cuadrícula](#).

Observaciones

Podemos tener la mayor cantidad de columnas posible de la manera mencionada anteriormente.

Examples

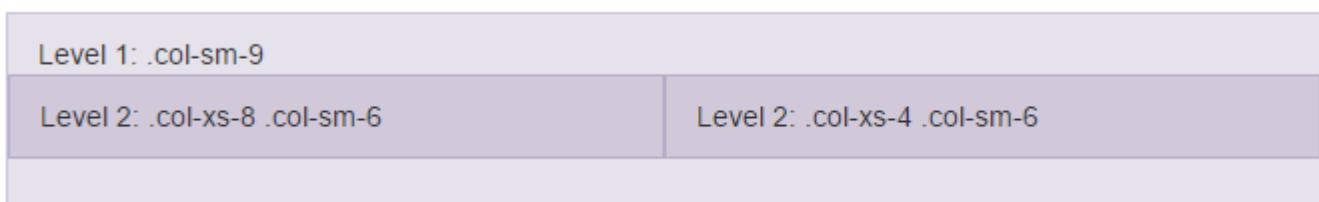
Columnas de anidación

```
<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>
```

El ejemplo se toma de [<http://getbootstrap.com/css/#grid-nesting>]

Como lo sugiere la documentación del sitio web oficial.

Para anidar su contenido con la cuadrícula predeterminada, agregue una nueva columna `.row` y un conjunto de columnas `.col-sm-*` dentro de una columna existente `.col-sm-*`. Las filas anidadas deben incluir un conjunto de columnas que suman hasta 12 o menos (no es necesario que use las 12 columnas disponibles).



Lea Nido de rejilla en línea: <https://riptutorial.com/es/twitter-bootstrap/topic/9088/nido-de-rejilla>

Capítulo 30: Paginación

Introducción

Los enlaces de paginación indican que existe una serie de contenido relacionado en varias páginas. Normalmente, estos se utilizan cuando un enfoque de varias páginas para largas listas de contenido mejora el rendimiento general, como en los resultados de búsqueda o en las bandejas de entrada.

Examples

Un simple ejemplo de paginación.

```
<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>
```

lo que te da:

Previous	1	2	3	Next
----------	---	---	---	------

Lea Paginación en línea: <https://riptutorial.com/es/twitter-bootstrap/topic/10605/paginacion>

Capítulo 31: Personalización de estilo de Bootstrap de Twitter

Observaciones

Una cosa a tener en cuenta es que uno tiene que mencionar el nombre custom.css después del bootstrap.css principal, de lo contrario, los valores de custom.css no se implementarán realmente.

Examples

Anulación de CSS predeterminado

Todo el mundo ama el [bootstrap de Twitter](#), pero a algunos de nosotros no nos gusta su diseño predeterminado. Así que aquí hay una guía simple sobre cómo comenzar a personalizar el diseño de bootstrap. El refuerzo de Twitter cuando se clona proporciona un conjunto de archivos css predeterminados que podemos anular.

El archivo css de correo que debemos anular es `bootstrap.min.css` en el directorio `bootstrap/dist/css`.

Para anular el diseño predeterminado de bootstrap, siga estos 2 sencillos pasos.

1. `custom.css` un `custom.css` (o puede `custom.css` nombre que desee) y vincularlo a su `index.html`

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

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

2. Empezar a personalizar. Por ejemplo, queremos cambiar el color del botón predeterminado. Si desea utilizar el estilo de botón predeterminado de bootstrap, debe agregar la clase `btn` en su etiqueta `<button class="btn">Sample</button>`. Simplemente escriba el siguiente código en su `custom.css`.

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

El código de arriba producirá algo como esto.

Defecto :

Sample

Personalizado :

Sample

Esta técnica nos evitará reescribir todos los estilos de botones que ya fueron escritos por los colaboradores de bootstrap. Esto también nos salvó de escribir nuestra propia clase de css, que para mí es menos tediosa.

Lea Personalización de estilo de Bootstrap de Twitter en línea: <https://riptutorial.com/es/twitter-bootstrap/topic/6030/personalizacion-de-estilo-de-bootstrap-de-twitter>

Capítulo 32: Pestañas

Examples

HTML básico

```
<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>
```

Esto creará un conjunto de pestañas con 3 pestañas y 3 divs de contenido asociados.

Pestañas animadas

Para hacer que las pestañas .fade , agregue .fade a cada .tab-pane . El panel de pestañas activo también debe tener la clase .in para que el contenido inicial esté visible.

```
<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>
```

Lea Pestañas en Línea: <https://riptutorial.com/es/twitter-bootstrap/topic/5980/pestanas>

Capítulo 33: Sistema de red

Introducción

El sistema de cuadrícula de Bootstrap consta de 12 unidades conocidas como **Columnas** (`.col-*` clases CSS) que se utilizan para distribuir el contenido de *izquierda a derecha* en la ventana gráfica. Las columnas están contenidas en **Filas** (clase CSS de `.row`) para crear grupos horizontales de columnas. Las filas se colocan dentro de un **contenedor** fijo o de ancho completo (`.container` o `.container-fluid`, respectivamente) para una alineación adecuada. Las columnas tienen un relleno que crea espacio (conocido como "canalón") entre el contenido de las columnas.

Observaciones

Bootstrap incluye un primer sistema de cuadrícula fluida móvil y sensible que escala de manera apropiada hasta 12 columnas a medida que aumenta el tamaño del dispositivo o la ventana gráfica. Incluye clases predefinidas para crear rápidamente diseños de página a través de una serie de filas y columnas que albergan su contenido.

Examples

Preguntas de los medios

Las consultas de medios en Bootstrap le permiten mover, mostrar y ocultar contenido según el tamaño de la ventana gráfica. Las siguientes consultas de medios se utilizan en archivos MENOS para crear los puntos de interrupción clave en el sistema de rejilla de Bootstrap:

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

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

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

Ocasionalmente, estos se expanden para incluir un ancho máximo para limitar CSS a un conjunto más estrecho de dispositivos:

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

Niveles de rejilla de arranque (puntos de interrupción)

Además del concepto de [unidades de columna](#), Bootstrap tiene diferentes **puntos de corte** o

tamaños de cuadrícula conocidos como niveles. La cuadrícula de Bootstrap 3 tiene cuatro (4) niveles para acomodar diferentes anchos de pantalla (o ventana gráfica). Los niveles de Bootstrap 3 son `xs` , `sm` , `md` y `lg` . Las columnas de la cuadrícula de Bootstrap se identifican mediante diferentes `col-{breakpoint}-{units}` clases CSS.

Cada nivel de cuadrícula **aborda un rango** que está diseñado para ajustarse a los anchos de pantalla típicos de dispositivos, como el de computadoras de escritorio, computadoras portátiles, tabletas y teléfonos inteligentes.

Bootstrap utiliza consultas de medios CSS para crear puntos de interrupción sensibles que establecen un límite para cada tamaño de cuadrícula. Estos tamaños de cuadrícula le permiten cambiar el diseño de las columnas para adaptarse mejor a los diferentes anchos de pantalla y dispositivos: la esencia del diseño sensible.

- `col-xs-*` - para los anchos de pantalla *más pequeños* , como los teléfonos inteligentes <768 px
- `col-sm-*` - para anchos de pantalla *pequeños* como teléfonos inteligentes y tabletas> = 768 px
- `col-md-*` - para anchos de pantalla *medios* como tabletas y portátiles> = 992 px
- `col-lg-*` - para anchos de pantalla *grandes* como escritorios> = 1200 px

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

Referencia: [Sistema Grid](#)

El mismo ancho de columna para cada dispositivo

Para crear una columna que siempre sea el 50% del ancho de la ventana gráfica (en todos los dispositivos), puede configurar `col-*-6` para cada nivel.

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

Sin embargo, este es un margen adicional innecesario, ya que `col-xs-6` significa 6 unidades en `xs` y superiores. El nivel más pequeño que establezca (`xs`, `sm` o `md`) también define el tamaño para anchos de pantalla más grandes. Para la columna del *mismo* tamaño en todos los niveles, simplemente configure el ancho para la ventana más pequeña.

Código más corto:

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

Ancho de columna diferente para cada dispositivo (diseño sensible)

Las clases `col-*-*` se pueden **combinar** para controlar los anchos de columna en diferentes tamaños de cuadrícula.

Por ejemplo, cree una columna de 50% de ancho en el nivel `sm`, y una columna de 25% de ancho en el nivel `md` ...

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

Las cuadrículas `sm`, `md` y `lg` se "apilarán" verticalmente en anchos de ventana de visualización de menos de 768 píxeles. Aquí es donde encaja la cuadrícula `xs`. Las columnas que usan las clases `col-xs-*` no se apilarán verticalmente y continuarán reduciéndose en las pantallas más pequeñas.

Bootstrap filas y columnas

El sistema de cuadrícula de Bootstrap tiene **12 unidades** conocidas como **Columnas** que se pueden usar para distribuir el contenido horizontalmente en la ventana gráfica.

El motivo de una cuadrícula de 12 unidades (en lugar de 10, 16, etc.) es que 12 se divide en 6 (mitades), 4 (cuartos) y 3 (tercios). Esto hace que la adaptación a una variedad de diseños sea mucho más fácil. Las columnas de la cuadrícula de Bootstrap se identifican mediante diferentes `col-{breakpoint}-{units}` clases CSS. [Obtenga más información sobre el ancho de la ventana gráfica y los puntos de interrupción \(niveles AKA\)](#)

Entonces, por ejemplo, `col-md-3` representa una columna que ocupa 3 de las 12 unidades (o 25%) en una ventana de ancho medio (`md`). Para usar un ancho de columna en su diseño, simplemente use la clase de `col-{breakpoint}-{units}` apropiada en su marca HTML.

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

El ancho de la columna es fluido (no el ancho fijo), por lo que las columnas consumen un *porcentaje* de su contenedor.

Unidades de columna en Bootstrap 3

- `col-*-1` : 1 de 12 (8.33333333% de ancho)
- `col-*-2` : 2 de 12 (16.66666667% de ancho)
- `col-*-3` : 3 de 12 (25% de ancho)

- col-*-4 : 4 de 12 (33.3333333% de ancho)
- col-*-5 : 5 de 12 (41.66666667% de ancho)
- col-*-6 : 6 de 12 (50% de ancho)
- col-*-7 : 7 de 12 (58.3333333% de ancho)
- col-*-8 : 8 de 12 (66.66666667% de ancho)
- col-*-9 : 9 de 12 (75% de ancho)
- col-*-10 : 10 de 12 (83.3333333% de ancho)
- col-*-11 : 11 de 12 (91.66666667% de ancho)
- col-*-12 : 12 de 12 (100% de ancho)

Demo - 12 unidades de la columna de Bootstrap

La fila de Bootstrap

La clase `.row` Bootstrap se utiliza para contener las columnas. Las columnas *siempre* se deben colocar en filas, y las filas siempre se deben colocar dentro de un contenedor (`container` o `container-fluid`). La fila utiliza márgenes negativos (-15px) para garantizar un espacio adecuado entre el contenido de la columna y el borde del navegador. Las filas se utilizan para agrupar columnas horizontalmente.

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

Columnas llenarán el `.row` horizontal de izquierda a derecha, y se [envuelva](#) a una nueva línea cada 12 unidades de columna. Por lo tanto, puede usar `.row`s para crear **saltos horizontales**, o puede agregar **más de 12** unidades de columna en un solo elemento `.row` para que las **columnas** se `.row` (o apilen) verticalmente en la ventana gráfica.

Cuando use el `.row` columnas (más de 12 unidades en una `.row`), deberá [usar restablecimientos de respuesta \(o correcciones claras\)](#) para garantizar un [ajuste](#) uniforme del contenido de la columna desigual. Esto es esencial cuando el contenido de las columnas varía en altura.

Más sobre las columnas y filas de la rejilla de arranque

[¿Bootstrap 3 problemas de diseño de cuadrícula fluida?](#)

[Bootstrap 3 - fila anidada ¿puedo tener columnas que sumen más de 12?](#)

[Bootstrap fila y explicación col](#)

[Cómo funciona la rejilla Bootstrap \(Medio\)](#)

Contenedores

Bootstrap requiere un elemento contenedor para envolver los contenidos del sitio y alojar nuestro sistema de cuadrícula. Puede elegir uno de los dos contenedores para usar en sus proyectos.

Utilice la clase `.container` para un contenedor de ancho fijo sensible.

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

Use `.container-fluid` clase `.container-fluid` para un contenedor de ancho completo, que abarca todo el ancho de su ventana gráfica.

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

Nota: los contenedores no son encajables (no puede colocar un contenedor dentro de otro contenedor), debido al `padding` y más.

Columnas de compensación

Estas clases aumentan el margen izquierdo de una columna en * columnas. Por ejemplo, `.col-md-offset-4` mueve `.col-md-4` en cuatro columnas.

```
<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>
```

Manipulación del orden de las columnas mediante push and pull

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

Esto cambia el orden de las columnas de cuadrícula incorporadas.

Sintaxis: `.col-md-push- *` y `.col-md-pull- *`.

Más:

[Manipulación del orden de las columnas mediante col-lg-push y col-lg-pull en Twitter Bootstrap 3](#)

Bootstrap 3: empujar / tirar columnas solo en tamaños de pantalla más pequeños
Columna ordenando y apilando en Bootstrap 3

Lea Sistema de red en línea: <https://riptutorial.com/es/twitter-bootstrap/topic/3330/sistema-de-red>

Capítulo 34: Temas Bootstrap

Examples

Temas bootstrap versus anulaciones de reglas

¿Qué son los temas?

Existen varios aspectos visuales para Bootstrap, que se pueden encontrar en fuentes, como [Bootswatch](#), que están modificando el archivo `bootstrap.min.css`. También puedes crear tu propio tema de esta manera.

¿Cuándo modificar los temas y cuándo agregar nuevas reglas a un archivo site.css?

Cuándo modificar el

`bootstrap.min.css`

archivo, y cuándo agregar su propio archivo `.css`, como

`site.css`

?

A veces hay requisitos de estilo, que deben hacerse, independientemente del tema que esté utilizando. Estas reglas deben ir a su propio archivo `.css`, como `site.css`, para que se pueda cambiar el tema principal, las reglas de `site.css` se aplicarán de todos modos. Para hacer eso, solo tienes que vincular el tema `bootstrap`, y tus propias reglas, para anular las existentes:

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

De esta manera, los temas de Bootstrap se pueden cambiar en cualquier momento sin perder las reglas obligatorias, aplicadas desde `site.css`.

Lea Temas Bootstrap en línea: <https://riptutorial.com/es/twitter-bootstrap/topic/6381/temas-bootstrap>

Capítulo 35: Usando Clearfix en Filas y Cols

Introducción

Al crear diseños avanzados, puede haber escenarios en los que necesite usar **más de 12 unidades de columna** en un solo elemento `.row`. El concepto de **ajuste de columnas** y **restablecimientos de respuesta** (también conocidos como **clearfixes**) son **esenciales** para entender el diseño de respuesta con Bootstrap.

[Fundamentos de la rejilla de Bootstrap](#)

Observaciones

Las rejillas Bootstraps son notablemente poderosas y elegantes. Sin embargo, debe recordar que el nombre del marco es "Bootstrap", no "WeDidItForYou". Bootstrap **permite** un diseño sensible, no lo **garantiza**.

Aún depende de usted hacer que su diseño sea verdaderamente receptivo y brindar a sus usuarios la mejor experiencia de usuario final posible.

Examples

El primer intento ingenuo

Antes de comenzar, vamos a definir algunos CSS para los ejemplos. Esta es la `head` la sección de la muestra. Siempre uso `border-radius` y `background-color` cuando estoy probando, porque hace que ver las divisiones de celda sea simple sin agregar ningún tamaño de borde que pueda afectar el tamaño de las celdas.

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

```

        }
        /* padding top-bottom for some row examples */
        .row.padded {
            padding: 1rem 0 1rem 0;
        }
    </style>
</head>

```

¡Con eso fuera del camino, definamos una cuadrícula y veamos los resultados perfectos en todos los tamaños de vistas!

Utilizando `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



Las dos imágenes anteriores muestran la representación en tamaños de pantalla medianos y pequeños. Recuerde, obtendremos CUATRO columnas en medio + debido a `col-md-3`, y DOS celdas en pequeño, debido a `col-xs-6`.

Se ve bastante bien, ¿verdad? Creo que hemos terminado aquí! Dijo un montón de sitios

ingenuos de Bootstrap ahí fuera esperando romper ...

El problema de la altura

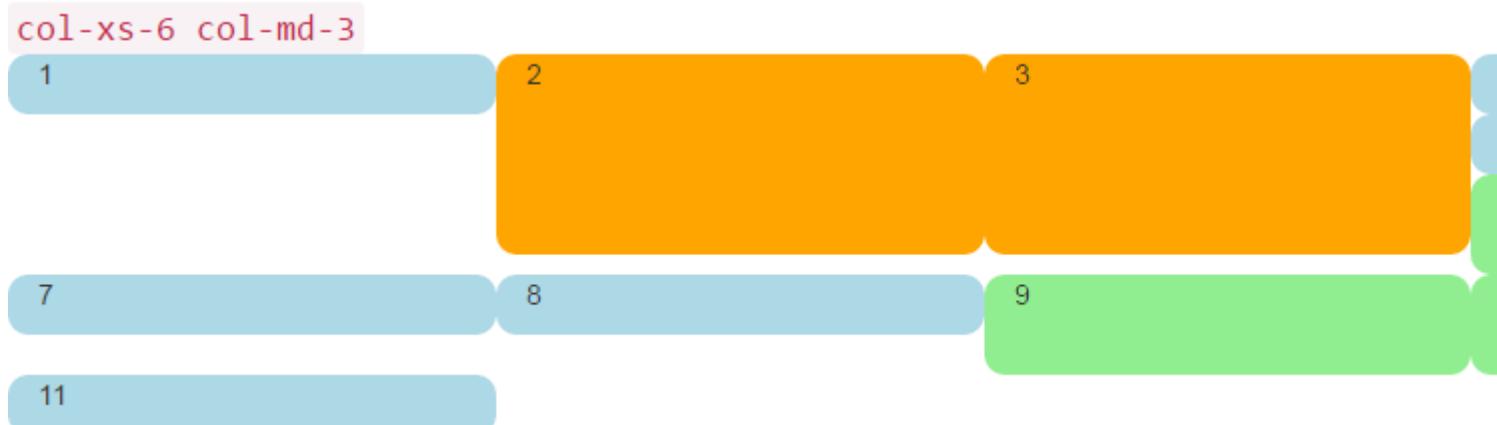
En nuestro " [ejemplo ingenuo](#) ", todas nuestras células tenían la misma altura. El navegador rompió voluntariamente las líneas exactamente donde queríamos, y todo parecía ir bien con el mundo. Hasta que la altura entra en escena.

Tomemos el ejemplo anterior y asignemos cierta altura a algunas de las celdas, tal vez como se vería en una página de tipo tablero de control.

```
<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>
```

Aquí hemos añadido algunos CSS de `cell-tall` y `cell-med` que definimos anteriormente. Esto tendrá el efecto de cambiar la altura de algunas de las celdas. Me pregunto cómo se verá ...

Aquí están de nuevo en tamaños de pantalla medianos y pequeños:



col-xs-6 col-md-3



Oh mi, qué desastre. No creo que eso es lo que queríamos. En tamaño medio-grande, 5 y 6 están fuera de lugar, y de alguna manera 7 terminaron comenzando una nueva fila. ¡En tamaño pequeño tenemos dos celdas en la primera fila, y **cuatro** en la segunda fila, con 4, 5 y 6 todas apiladas a la derecha en ambos tamaños de pantalla!

Entonces, ¿cómo resolvemos esto?

Clearfix al rescate

Una forma de ayudar a la situación sin duda, sería usar más de una `row`:

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

Esto suele ser lo primero que intentan los nuevos Bootstrappers. Parece tener sentido: "Quiero cuatro celdas en cada fila, así que solo crearé una nueva `row` para cada 4 `col` divs".

Pero hay un problema con esta línea de razonamiento: todo el punto de Bootstrap 3 y la próxima versión 4 es ser **receptivo**. Al colocar "cuatro `col` en una `row`", no estás realmente "pensando responsivamente".

Una buena comprensión de la `clearfix` clase CSS le ayudará a empezar a ver que varias `row` divs realmente han sido nublando su comprensión de la forma en que el diseño de respuesta estaba **destinado** a trabajar. En resumen, de todos modos, simplemente **no puede** saber cuántos `col`

ponen en `row`. ¡El navegador aún no ha hecho su trabajo!

¿Recuerdas en First Things First, dijimos que debes pensar en "inverso a 12"? Sin más preámbulos, solucionemos nuestro problema aquí, utilizando comentarios en el código para aclarar cualquier confusión. Sí, parece mucho más código, pero la **mayoría** de los extras son comentarios.

```
<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>
```

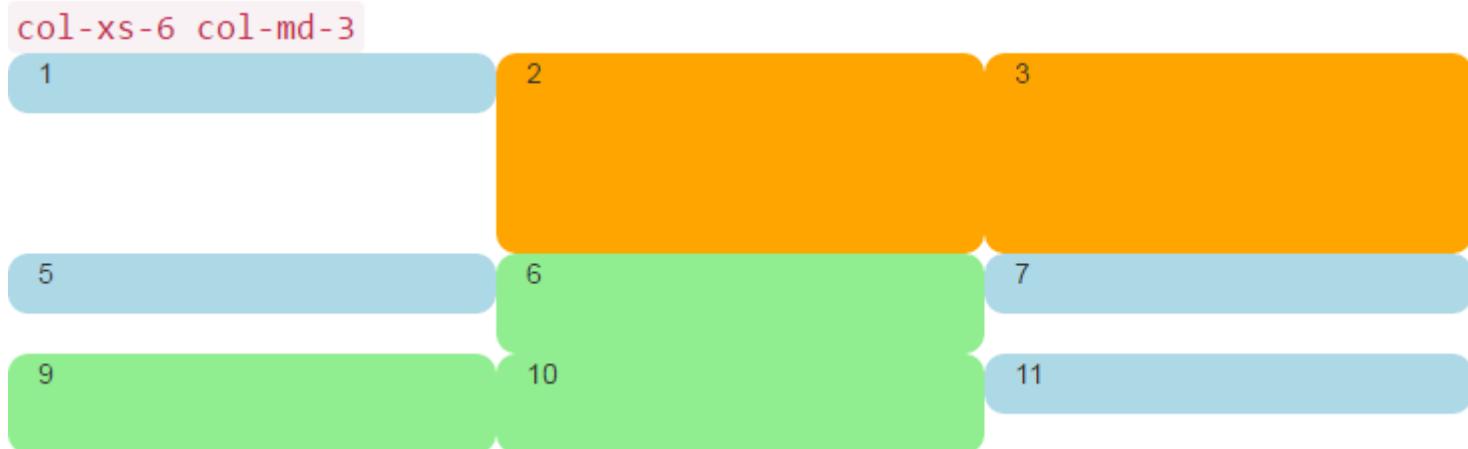
El `clearfix` es una clase de CSS que genera una pequeña `clearfix` (virtualmente invisible), y su propósito es "borrar" los flotadores de la `left` que han sido utilizados por los divs de `col`.

El genio está realmente en las clases `hidden-sm`, `hidden-md`, etc. Estas clases se colocan **en el div clearfix**, no en el `col` divs! ¡Esto hace que el div `clearfix` aparezca mágicamente o desaparezca de la secuencia de representación en ciertos anchos de ventana gráfica! ¡Genio!

Bootstrap tiene una variedad desconcertante de `hidden-*` y `visible-*` clases en la versión 3, y por desgracia no son realmente la "inversa" de unos a otros. Por lo tanto, me parece más claro y más seguro utilizar siempre las clases `hidden-*` en los arreglos transparentes.

Esto parece que puede cambiar para mejor en Bootstrap 4, con clases como `hidden-*-up` y `hidden-*-down` (se están deshaciendo de las clases `visible-*`) por completo.

Bastante verborrea, ¿qué aspecto tiene ahora?



Eso es lo que queremos! En la pantalla grande, siempre tenemos CUATRO en la pantalla más pequeña, siempre DOS en la pantalla grande. No más apilamiento en lugares extraños, y las brechas son donde esperaríamos que estuvieran.

Un tablero de instrumentos

Bastante bien de esas cosas redondeadas de colores, pongamos algo más interesante que los números en esos divs. Tomemos ese mismo conjunto de columnas y hagamos un tablero real. Usa el siguiente CSS:

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

Y aquí está el código del "tablero":

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

```

<div class="panel-footer">
    Footer 2
</div>
</div>
<!--
    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>
<!--
    Now we have rendered EIGHT cells, which is a multiple of TWO AND FOUR,
    so we put in a clearfix that's always visible.
-->
<div class="clearfix"></div>
<!--
<div class="col-xs-6 col-md-3 cell-med">
    <div class="panel panel-info panel-short">
        <div class="panel-heading">
            Heading 9
        </div>
        <div class="panel-body">
            Body 9.
        </div>
        <div class="panel-footer">
            Footer 9
        </div>
    </div>
</div>

```

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

Ese código se verá así:

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

Y así en ventanas más pequeñas:

Footer 1

Footer 2

Heading 3

Body 3. The file has successfully uploaded.

Footer 3

Heading 4 Chart

Body 4. Is this a cool graph or what?

Footer 4

Heading 5

Body 5.

Heading 6

Body 6.

Por cierto, estoy usando la clase de `panel` Bootstrap 3, que desaparecerá en Bootstrap 4 y será reemplazada por la `card` mucho más descriptiva y específica. Al observar estas imágenes, puede ver por qué la `card` será un nombre mucho mejor que el `panel` ambiguo.

Disposición 2.4.6 con Clearfixes

Aquí hay un diseño que representa dos, cuatro o seis celdas según el tamaño de la pantalla.

```
<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 aclearfix 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 ourclearfix 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>

```

Pantalla grande:



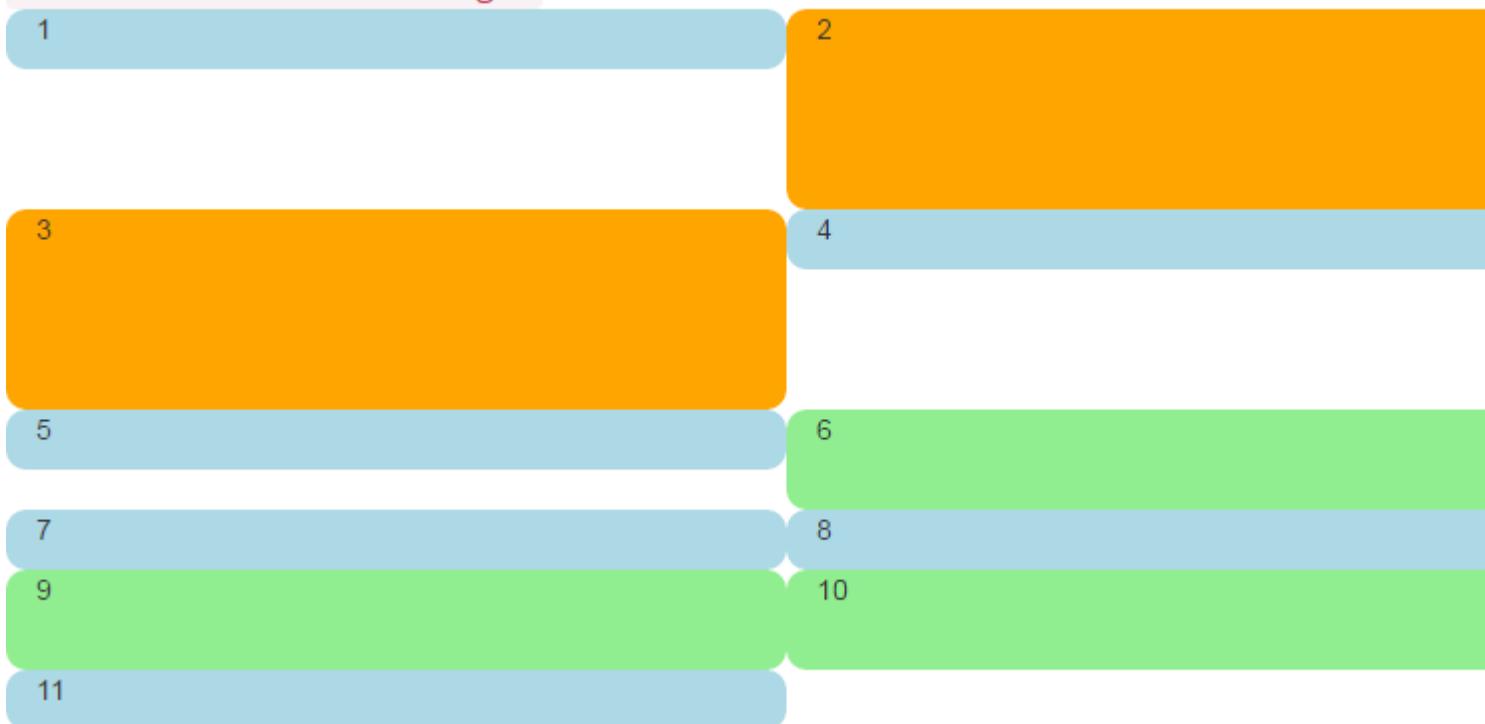
Pantalla mediana:

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



Pantalla pequeña:

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



¿Por qué las columnas de Bootstrap superan 12 en una fila?

Hay muchos escenarios de respuesta en los que es *necesario* tener unidades de columna que excedan 12 en un solo elemento `.row`. Esto se conoce como [ajuste de columnas](#).

Si se colocan más de 12 columnas dentro de una sola fila, cada grupo de columnas adicionales se agrupará, como una unidad, en una nueva línea.

Por ejemplo, considere un diseño donde queremos ...

- 3 columnas en dispositivos grandes y medianos, y
- 2 columnas en dispositivos pequeños y pequeños

Large



Small



Para obtener este diseño en Bootstrap, usaríamos (correcto) ..

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

Método correcto de demostración

Como puede ver en el ejemplo, el *total* de unidades de columna en el elemento `.row` **superá 12**. Esta técnica se conoce como **ajuste de columnas** y es una de las características de diseño de respuesta más poderosas de Bootstrap. El diseño deseado *no sería posible* (aparte de duplicar el marcado) si intentamos mantenernos en la **idea errónea de que las unidades de columna deben sumar 12 en una sola fila**.

El diseño *no* es posible cuando no superamos 12 (incorrecto) ..

```
<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>
```

Demostración del método incorrecto (falla 3 columnas en grande)

Recuerde, un `.row` *no* es *lo* mismo que una sola línea en la ventana gráfica. Una `.row` es una agrupación de columnas. Las columnas que superen las 12 unidades en un solo elemento `.row` se ajustarán **a una nueva línea** (en la ventana). Por eso es esencial entender que las 12 columnas representan **unidades** horizontales en la ventana gráfica.

Además, los **restablecimientos de respuesta** (clearfix) deben usarse para un ajuste uniforme cuando las **columnas varían en altura** .

Lea Usando Clearfix en Filas y Cols en línea: <https://riptutorial.com/es/twitter-bootstrap/topic/6124/usando-clearfix-en-filas-y-cols>

Capítulo 36: Validación Bootstrap

Observaciones

- Esta técnica de validación solo se puede utilizar en entradas que están dentro de un formulario.
- Las propiedades deben tener al menos un requisito de validación para mostrar el resaltado en una validación `onSubmit()` fallida. Los tipos de datos (aparte de la cadena) tienen un requisito de tipo de datos oculto, por lo que no requieren una anotación de datos explícita. Las cadenas no tienen esto, por lo que para forzar una verificación de validación junto con los otros campos, agregue la anotación de datos `[MinLengthAttribute(0)]`.

Examples

Uso de ASP.NET MVC y anotaciones de datos

Agregue lo siguiente a Web.config (en la carpeta Vistas), dentro de `<appSettings>`:

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

Agregue el paquete jqueryval a BundleConfig.cs:

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

Agregue lo siguiente a todas las páginas que necesitan validación (o _Layout.cshml):

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

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

Agregue anotaciones de datos a los campos relevantes en el modelo:

```
using System.ComponentModel.DataAnnotations;  
  
[Required(ErrorMessage = "This field is required.")]
```

En la vista, agregue lo siguiente a cada entrada que necesite validación:

```
<!-- Validation messages -->  
<div class="text-danger">@Html.ValidationMessageFor(m => m.SomeField)</div>  
  
<!-- Bootstrap feedback span: -->  
<span class="glyphicon form-control-feedback"></span>
```

Agregue lo siguiente a la acción del controlador relevante para agregar la validación del lado del servidor:

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

Ejemplo de entrada que requiere validación

Modelo:

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

Ver:

```
<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>
```

Opcional

Agregue el siguiente jQuery para validar las entradas en el desenfoque, así como en el envío:

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

Lea Validación Bootstrap en Línea: <https://riptutorial.com/es/twitter-bootstrap/topic/6388/validacion-bootstrap>

Creditos

S. No	Capítulos	Contributors
1	Empezando con el twitter-bootstrap	andreaem, Atul Mishra, bpoiss, Community, Evan, Gleb Kemarsky, H. Pauwelyn, kernal lora, Kritner, MattD, Mingle Li, Nhan, Prashanth Benny, the12, tmg, Toby, VincenzoC, Vishnu Y S
2	Afijo Bootstrap	Ilyas karim
3	Alerta	JHS, Madalina Taina, tmg
4	Bootstrap Dropdowns	Ismail Farooq, MattD
5	Bootstrap Navbar	Ilyas karim
6	Botones	Madalina Taina, Muhammad Abdullah, Richard Hamilton, the12
7	Carruseles	alex, Boysenb3rry
8	Clases de utilidad	ajju
9	Columnas	kybernaut.cz
10	Componentes Bootstrap	alex
11	Contenedores Bootstrap	Neha Chopra
12	Diálogos modales	alex, mnoronha
13	Formas	Community, Jens, Owen Pauling
14	Glifos	Madalina Taina, tmg, Umer Farooq
15	Grupo de lista	Ilyas karim
16	Imprimiendo en bootstrap.	MattD
17	Información sobre herramientas	Madalina Taina, tmg
18	Insignias y etiquetas	mmativ

	de Bootstrap	
19	Jumbotron	Gabriel Chi Hong Lee
20	Listas deplegables	alex
21	Los paneles	JackPoint, tmg
22	Menús de navegación	Ignacio Correia
23	Mesas	atjoedonahue
24	Migración a Bootstrap 4	Chris Farmer, neophyte, ZimSystem
25	Modales	John Blanchard
26	Navbar	Kritner, Krunal Mevada, kybernaut.cz
27	Naves	leowebguy
28	Nido de rejilla	neophyte, ZimSystem
29	Paginación	TheDarkKnight
30	Personalización de estilo de Bootstrap de Twitter	CENT1PEDE, Vikas Yadav
31	Pestañas	DavidG, tmg
32	Sistema de red	Ani Menon, Boysenb3rry, bpoiss, Harshal Patil, leowebguy, Madalina Taina, Mingle Li, mmativ, Stephen Leppik, the12, tmg, ZimSystem
33	Temas Bootstrap	KAI
34	Usando Clearfix en Filas y Cols	Bruce Pierson, ZimSystem
35	Validación Bootstrap	Amy Barrett, mnoronha