



**Kostenloses eBook**

**LERNEN**

**twitter-bootstrap**

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

**#twitter-**

**bootstrap**

# Inhaltsverzeichnis

Über.....	1
<b>Kapitel 1: Erste Schritte mit twitter-bootstrap.....</b>	<b>2</b>
Bemerkungen.....	2
Versionen.....	2
Examples.....	3
Installation / Setup.....	3
Grundlegende Vorlage.....	5
Wann sollte Bootstrap verwendet werden?.....	6
Grundlegende Webseite mit Bootstrap-Komponenten.....	6
<b>Kapitel 2: Anpassung des Twitter-Bootstrap-Stils.....</b>	<b>9</b>
Bemerkungen.....	9
Examples.....	9
Standard-CSS überschreiben.....	9
<b>Kapitel 3: Bootstrap Badges und Labels.....</b>	<b>11</b>
Examples.....	11
Abzeichen.....	11
Etiketten.....	11
<b>Kapitel 4: Bootstrap Dropdowns.....</b>	<b>12</b>
Parameter.....	12
Bemerkungen.....	12
Examples.....	12
Wie benutzt man.....	12
Basisbeispiel.....	13
<b>Kapitel 5: Bootstrap Navbar.....</b>	<b>14</b>
Examples.....	14
Bootstrap Navbar.....	14
Bootstrap-Markenbild.....	14
<b>Kapitel 6: Bootstrap-Befestigung.....</b>	<b>16</b>
Examples.....	16
In der Navbar.....	16

Fixierungsbeispiel 2.....	16
<b>Kapitel 7: Bootstrap-Container.....</b>	<b>18</b>
Einführung.....	18
Examples.....	18
Behälter.....	18
<b>Kapitel 8: Bootstrap-Komponenten.....</b>	<b>19</b>
Bemerkungen.....	19
Examples.....	19
Beispiele für Bootstrap-Komponenten.....	19
<b>Kapitel 9: Bootstrap-Themen.....</b>	<b>20</b>
Examples.....	20
Bootstrap-Designs im Vergleich zu Regelüberschreibungen.....	20
<b>Kapitel 10: Bootstrap-Überprüfung.....</b>	<b>21</b>
Bemerkungen.....	21
Examples.....	21
Verwenden von ASP.NET MVC und Datenanmerkungen.....	21
Beispieleingabe, für die eine Überprüfung erforderlich ist.....	22
Wahlweise.....	23
<b>Kapitel 11: Dropdowns.....</b>	<b>24</b>
Bemerkungen.....	24
Examples.....	24
Grundlegende HTML-Verwendung.....	24
<b>Kapitel 12: Drucken in Bootstrap.....</b>	<b>25</b>
Examples.....	25
Grundlegende HTML-Verwendung.....	25
<b>Kapitel 13: Formen.....</b>	<b>26</b>
Examples.....	26
Grundform.....	26
Schreibgeschützte und deaktivierte Eingänge.....	26
<b>Kapitel 14: Gebrauchsklassen.....</b>	<b>27</b>
Examples.....	27

Generiere .hidden- * Klassen für alle Haltepunkte - SCSS.....	27
<b>Kapitel 15: Gitternetz.....</b>	<b>28</b>
Einführung.....	28
Bemerkungen.....	28
Examples.....	28
Spalten verschachteln.....	28
<b>Kapitel 16: Glyphicons.....</b>	<b>29</b>
Bemerkungen.....	29
Examples.....	29
Wie benutze ich Glyphicons?.....	29
<b>Kapitel 17: Gruppe auflisten.....</b>	<b>31</b>
Bemerkungen.....	31
Examples.....	31
Grundlegendes Beispiel.....	31
Abzeichen.....	31
Verknüpfte Elemente.....	31
Schaltflächenelemente.....	31
Deaktivierte Elemente.....	32
Kontextuelle Klassen.....	32
Benutzerdefinierter Inhalt.....	32
<b>Kapitel 18: Jumbotron.....</b>	<b>33</b>
Einführung.....	33
Bemerkungen.....	33
Examples.....	33
Grundlegender Jumbotron mit zwei Textzeilen und einem Button.....	33
<b>Kapitel 19: Karussells.....</b>	<b>35</b>
Bemerkungen.....	35
Examples.....	35
Grundlegende HTML-Verwendung.....	35
Grundlegende Verwendung und Initialisierung von Javascript.....	36
<b>Kapitel 20: Migration zu Bootstrap 4.....</b>	<b>37</b>

Einführung .....	37
Bemerkungen .....	37
Examples .....	37
Spaltenlayoutänderungen des Rastersystems in Bootstrap 4 .....	37
Gitterlayout Bootstrap 4 .....	37
Änderungen der Browserunterstützung .....	38
Entfernen Sie die Klassen .....	38
Bootstrap 4 Navbar .....	41
Bootstrap 3 bis Bootstrap 4 CSS-Änderungen .....	44
Bootstrap 4 Vertikal ausrichten .....	46
Bootstrap 4 Zentrieren .....	48
<b>Horizontales Zentrum .....</b>	<b>48</b>
<b>Vertikales Zentrum .....</b>	<b>49</b>
Bootstrap 4 Column Order .....	49
<b>Kapitel 21: Modale Dialoge .....</b>	<b>51</b>
Bemerkungen .....	51
Examples .....	51
Grundlegende HTML-Verwendung .....	51
Grundlegende Verwendung und Initialisierung von Javascript .....	51
<b>Kapitel 22: Modals .....</b>	<b>53</b>
Bemerkungen .....	53
Examples .....	53
Basic HTML Modal .....	53
<b>Kapitel 23: Navbar .....</b>	<b>54</b>
Examples .....	54
Grundlegende Navigationsleiste (am oberen Rand der Seite befestigt) .....	54
Untermenü in der Navigationsleiste .....	54
Navbar-Teiler .....	55
Aktuellen Navigationslink "aktiv" halten .....	55
Navbar-Haltepunkt ändern (Handy vs. Normal) .....	55
Schließen Sie die minimierte Navigationsleiste, wenn Sie außerhalb der Navigationsleiste k. ....	56
<b>Kapitel 24: Navigationsmenüs .....</b>	<b>57</b>

Examples.....	57
Horizontales Pillenmenü.....	57
Vertikale Pillenmenü.....	57
Responsive horizontale Pille mit voller Breite.....	57
<b>Kapitel 25: Navs.....</b>	<b>58</b>
Examples.....	58
Bootstrap-Navs.....	58
<b>Kapitel 26: Panels.....</b>	<b>59</b>
Bemerkungen.....	59
Examples.....	59
Grundlegendes Beispiel.....	59
Panel mit Überschrift.....	59
Panel mit Fußzeile.....	59
<b>Kapitel 27: QuickInfo.....</b>	<b>61</b>
Bemerkungen.....	61
Examples.....	61
Tooltips positionieren.....	61
Basisbeispiel.....	61
<b>Kapitel 28: Rastersystem.....</b>	<b>63</b>
Einführung.....	63
Bemerkungen.....	63
Examples.....	63
Medien-Anfragen.....	63
Bootstrap-Raster (Haltepunkte).....	63
Bootstrap-Zeilen und -Säulen.....	65
Behälter.....	66
Versetzung von Spalten.....	67
Säulenordnungsmanipulation mit Push und Pull.....	67
<b>Kapitel 29: Säulen.....</b>	<b>69</b>
Examples.....	69
Responsive Spalten gleicher Höhe (nur CSS oder SASS).....	69
<b>Kapitel 30: Seitennummerierung.....</b>	<b>73</b>

Einführung .....	73
Examples .....	73
Ein einfaches Paginierungsbeispiel .....	73
<b>Kapitel 31: Tabellen .....</b>	<b>74</b>
Examples .....	74
Einfache Tabelle .....	74
<b>Kapitel 32: Tabellen .....</b>	<b>75</b>
Bemerkungen .....	75
Examples .....	75
Grundtabelle .....	75
Tisch mit modernem Styling .....	75
<b>Gestreifte Reihen .....</b>	<b>75</b>
<b>Tisch mit Rahmen .....</b>	<b>76</b>
<b>Schweben Sie in den Reihen .....</b>	<b>76</b>
<b>Kurzer Tisch .....</b>	<b>76</b>
<b>Kontextuelle Klassen .....</b>	<b>76</b>
Responsive Tabellen .....	77
Table Reflow - Vertikale Header .....	77
<b>Kapitel 33: Tabs .....</b>	<b>79</b>
Examples .....	79
Grundlegendes HTML .....	79
Animierte Registerkarten .....	79
<b>Kapitel 34: Tasten .....</b>	<b>81</b>
Syntax .....	81
Examples .....	81
Button-Klassen .....	81
<b>Kapitel 35: Verwenden von Clearfix in Zeilen und Spalten .....</b>	<b>84</b>
Einführung .....	84
Bemerkungen .....	84
Examples .....	84
Der naive erste Versuch .....	84

Das Höhenproblem.....	86
<b>Clearfix zur Rettung.....</b>	<b>87</b>
<b>Ein Dashboard.....</b>	<b>90</b>
2,4,6 Layout mit Clearfixes.....	95
Warum sollten Bootstrap-Spalten mehr als 12 hintereinander überschreiten?.....	97
<b>Kapitel 36: Warnen.....</b>	<b>100</b>
Bemerkungen.....	100
Examples.....	100
Warnungsarten.....	100
Warnungsgrundbeispiel.....	100
Animierte Benachrichtigungen.....	101
Ablehnbare Alarmer.....	101
Linkfarbe in Alerts.....	101
<b>Credits.....</b>	<b>103</b>



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

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

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

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

---

# Kapitel 1: Erste Schritte mit twitter-bootstrap

## Bemerkungen

Bootstrap ist ein HTML-, CSS- und JavaScript-Framework, das zum Erstellen von Websites verwendet wird, die mit einem Mobile-First-Paradigma sowie Responsive Web Design (RWD) erstellt werden. Es verwendet eine Kombination aus vorgefertigten CSS-Klassen und JavaScript, um verschiedene Dinge im Web zu erstellen. Dazu gehören ein benutzerdefiniertes, responsives Grid, mit dem Websites auf jedem Bildschirm angezeigt werden können, eine Dropdown-Navigationsleiste, die ansprechbar ist, und sogar einfache Dinge, die zeitaufwändig sein können, wie beispielsweise vorgefertigte Schaltflächen, Formulare und Akkordeons wenige.

Bootstrap kann aus folgenden Gründen nützlich sein:

1. **Zeitersparnis** : Bootstrap bietet viele Funktionen, die beim Erstellen von Code vorgefertigt sind und lediglich aufgerufen werden müssen. Dies spart viel Zeit und kann die Codierung einer Website erheblich verkürzen.
2. Entwickelt für **responsives Webdesign** : Bootstrap ermöglicht Webentwicklern, sich keine Gedanken über das Erstellen von Dingen zu machen, die mit der Größe des Bildschirms skaliert werden können, da Bootstrap ein Mobile-First-Responsive-Design verwendet, mit dem Dinge erstellt werden können, die auf jedem Bildschirm funktionieren Größe.
3. **Vereinfacht den Design-Prozess** : Bootstrap wird mit vorgefertigten Elementen geliefert, die über gute Design-Praktiken verfügen. Dies kann für diejenigen nützlich sein, deren Webdesign-Kenntnisse nicht besonders hoch sind, oder für diejenigen, die Design als langwierige Aufgabe betrachten, da viele Bootstrap-Klassen ästhetisch ansprechend und schön anzusehen sind.

---

Dieser Abschnitt sollte alle großen Themen in Twitter-Bootstrap erwähnen und auf die verwandten Themen verweisen. Da die Dokumentation für twitter-bootstrap neu ist, müssen Sie möglicherweise erste Versionen dieser verwandten Themen erstellen.

## Versionen

Ausführung	Veröffentlichungsdatum
4,0	2999-01-01
3.3.7	2016-07-25
3.3.6	2015-11-24
3.3.5	2015-06-15

Ausführung	Veröffentlichungsdatum
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

### Installation / Setup

Wird heruntergeladen:

- Laden Sie Bootstrap [direkt](#) herunter oder klonen Sie usw. aus dem [GitHub](#)- Repository
- Laden Sie Ihre angepasste Version von Bootstrap von offiziellen [Dokumenten](#) herunter
- `bower install bootstrap` mit **Bower**: `bower install bootstrap`
- `npm install bootstrap` mit **npm**: `npm install bootstrap`
- Mit **Composer** installieren: `composer require twbs/bootstrap`

### Die Dateistruktur

```

bootstrap/
├── css/
│   ├── bootstrap.css
│   ├── bootstrap.min.css
│   ├── bootstrap-theme.css
│   └── bootstrap-theme.min.css
├── js/
│   ├── bootstrap.js
│   └── bootstrap.min.js
├── fonts/
│   ├── glyphicons-halflings-regular.eot
│   ├── glyphicons-halflings-regular.svg
│   ├── glyphicons-halflings-regular.ttf
│   └── glyphicons-halflings-regular.woff

```

### Installation:

Fügen Sie in Ihre HTML-Seite Bootstraps CSS, JS und die Abhängigkeit von jQuery (Vorversion 3, mindestens ab der neuesten Bootstrap-Version) ein. Wenn Sie beabsichtigen, die JavaScript-Funktionen von Bootstrap zu verwenden, muss Ihre jQuery-Referenz *vor* Ihrer bootstrap.js-Referenz in Ihrem HTML-Code stehen.

Sie können Ihre installierten Bootstrap-Dateien aus dem obigen Abschnitt verwenden oder auf ein CDN verweisen, das von den Herstellern von Bootstrap bereitgestellt wird (Links, die aus [Erste Schritte mit Bootstrap stammen](#)):

```

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

<!-- Optional theme -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap-
theme.min.css" integrity="sha384-
rHyoNliRsVXV4nD0JutlnGaslCJuC7uwjduW9SVrLvRYooPp2bWYgmgJQIXw1/Sp" crossorigin="anonymous">

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

```

### Eine sehr einfache Bootstrap-Webseite:

```

<!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>

```

## Grundlegende Vorlage

```

<!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>
```

## Wann sollte Bootstrap verwendet werden?

Bootstrap ist ein Opinionated Framework für HTML, CSS und Javascript. Es enthält grundlegende Gestaltungsmerkmale und Funktionen für Elemente, die als Benutzeroberfläche akzeptiert wurden, z. B. Formularelemente, Schaltflächen, modale Fenster und Navigationselemente.

Bootstrap ist ein responsives Web-Framework, das heißt, Layout und Design für große und kleine Bildschirmgrößen wie mobile Geräte, Tablets und Desktop-Computer in einer einzigen Codebasis anpassen.

Eines der grundlegenden Konzepte von Bootstrap ist das Grid-Framework. Durch das Anwenden von Klassen auf HTML-Elemente können komplexe Layouts mit einem Grundraster mit zwölf Spalten erstellt werden. Beispielsweise kann sich ein Layout mit vier Spalten an zwei Spalten auf Tablet-Geräten und eine Spalte an mobilen Geräten anpassen. Das Raster verwendet `media queries`, eine CSS-Methode zur Ausrichtung auf bestimmte Bildschirmgrößen, um dies zu erreichen.

Bootstrap funktioniert besonders gut, wenn:

- Custom Design hat keine oberste Priorität
- Sie können HTML-Code leichter bearbeiten und Klassen hinzufügen als benutzerdefinierte CSS
- Sie können ein Framework verwenden, das viele visuelle Ähnlichkeiten mit vielen anderen Websites aufweist

Bootstrap kann von Neulingen in HTML, CSS und Javascript verwendet werden, da die [Dokumentation](#) hervorragend ist. Es gibt jedoch eine Lernkurve für diejenigen, die mit den drei von Bootstrap verwendeten Basistechnologien (HTML, CSS und Javascript) nicht ganz vertraut sind.

Sie können Bootstrap-Designs kaufen oder herunterladen, um den Stil oder die Funktionalität von Bootstrap zu ändern. Es ist auch möglich, Bootstrap als Ausgangspunkt für die Anpassung von CSS und Javascript zu verwenden.

## Grundlegende Webseite mit Bootstrap-Komponenten

```
<!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>

```

Erste Schritte mit twitter-bootstrap online lesen: <https://riptutorial.com/de/twitter-bootstrap/topic/818/erste-schritte-mit-twitter-bootstrap>

# Kapitel 2: Anpassung des Twitter-Bootstrap-Stils

## Bemerkungen

Zu beachten ist, dass der Name von `custom.css` nach der Hauptdatei `bootstrap.css` angegeben werden muss, andernfalls werden die Werte von `custom.css` nicht wirklich implementiert.

## Examples

### Standard-CSS überschreiben

Jeder liebt [Twitter-Bootstrap](#), aber einige von uns mögen das Standarddesign nicht. Hier ist eine einfache Anleitung, wie Sie das Design von bootstrap anpassen können. Wenn Twitter geklont wird, stellt Twitter bootstrap eine Reihe von Standard-CSS-Dateien bereit, die wir überschreiben können.

Die Mail-CSS-Datei, die wir überschreiben müssen, ist die `bootstrap.min.css` im `bootstrap/dist/css`.

Um das Standarddesign von bootstrap zu überschreiben, folgen Sie diesen zwei einfachen Schritten.

1. Erstellen Sie eine `custom.css` (oder Sie können sie beliebig benennen) und verknüpfen Sie sie mit Ihrer `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. Starten Sie das Customizing. Zum Beispiel möchten wir die Farbe der Standardschaltfläche ändern. Wenn Sie den standardmäßigen Schaltflächenstil von Bootstrap verwenden möchten, müssen Sie die `btn` Klasse in `btn <button class="btn">Sample</button>` -Tag `<button class="btn">Sample</button>`. Schreiben Sie einfach den folgenden Code in Ihre `custom.css`.

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

Der obige Code wird so etwas produzieren.

**Standard :**



**Brauch :**



Diese Technik erspart uns das Umschreiben der gesamten Button-Styles, die bereits von bootstrap-Mitwirkenden geschrieben wurden. Dies hat uns auch davon abgehalten, unseren eigenen CSS-Kurs zu schreiben, der für mich weniger langweilig ist.

Anpassung des Twitter-Bootstrap-Stils online lesen: <https://riptutorial.com/de/twitter-bootstrap/topic/6030/anpassung-des-twitter-bootstrap-stils>

---

# Kapitel 3: Bootstrap Badges und Labels

## Examples

### Abzeichen

Badges sind numerische Indikatoren dafür, wie viele Elemente mit einem Link verknüpft sind:

Verwenden Sie die `.badge` Klasse innerhalb von `<span>` -Elementen, um Badges zu erstellen:

```
<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>
```

### Abzeichen im Button

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

### Etiketten

Labels werden verwendet, um zusätzliche Informationen zu etwas bereitzustellen:

Verwenden Sie die `.label` Klasse, gefolgt von einer der sechs `.label-default`, `.label-primary`, `.label-success`, `.label-info`, `.label-warning` oder `.label-danger` innerhalb eines `<span>` -Elements ein Label erstellen:

```
<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>
```

Bootstrap Badges und Labels online lesen: <https://riptutorial.com/de/twitter-bootstrap/topic/7867/bootstrap-badges-und-labels>

# Kapitel 4: Bootstrap Dropdowns

## Parameter

Methoden	Beispiel
Über Javascript anrufen	<code>\$('.dropdown-toggle').dropdown();</code>
Schaltet das Dropdown um	<code>\$('.dropdown-toggle').dropdown('toggle')</code>
Ereignistyp	Beschreibung
show.bs.dropdown	Dieses Ereignis wird sofort ausgelöst, wenn die Show-Instanzmethode aufgerufen wird.
shown.bs.dropdown	Dieses Ereignis wird ausgelöst, wenn die Dropdown-Liste für den Benutzer sichtbar ist (wartet auf CSS-Übergänge, bis der Vorgang abgeschlossen ist).
hide.bs.dropdown	Dieses Ereignis wird sofort ausgelöst, wenn die Instanzmethode hide eingeblendet wurde.
hidden.bs.dropdown	Dieses Ereignis wird ausgelöst, wenn die Dropdown-Liste für den Benutzer ausgeblendet ist (wartet, bis die CSS-Übergänge abgeschlossen sind).
<b>Event-Handler-Beispiel</b>	<code>\$(element).on('show.bs.dropdown', function () { // do something.. })</code>

## Bemerkungen

Beim Aufruf von Dropdown über Javascript `$('.dropdown-toggle').dropdown()` ist das data-api dh `data-toggle="dropdown"` noch erforderlich. [Weiterlesen](#)

## Examples

### Wie benutzt man

Verwenden Sie die `.dropdown` Klasse für das übergeordnete Element des Dropdown-Menüs.

Fügen Sie einem Element die Dropdown-Menüklasse hinzu, um das Plugin für das Dropdown-Menü zu initialisieren.

Rufen Sie das Plugin mit der Klasse `.dropdown-toggle` und dem Attribut `data-toggle="dropdown"` für eine Schaltfläche oder einen Hyperlink auf.

## Basisbeispiel

```
<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>
```

Bootstrap Dropdowns online lesen: <https://riptutorial.com/de/twitter-bootstrap/topic/6361/bootstrap-dropdowns>

# Kapitel 5: Bootstrap Navbar

## Examples

### Bootstrap Navbar

Dies ist ein Beispiel für Bootstrap Version 3 Navbar:

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

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

### Bootstrap-Markenbild

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

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

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

# Kapitel 6: Bootstrap-Befestigung

## Examples

### In der 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>
```

### Fixierungsbeispiel 2

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

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

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

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

<style>

#con .well {
    height:400px;
}

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

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

**Bootstrap-Befestigung online lesen:** <https://riptutorial.com/de/twitter-bootstrap/topic/6639/bootstrap-befestigung>

---

# Kapitel 7: Bootstrap-Container

## Einführung

Verwenden Sie `.container` für einen responsiven Container mit fester Breite.

Verwenden Sie `.container-fluid` für einen Container mit voller Breite, der sich über die gesamte Breite des Viewports erstreckt.

## Examples

### Behälter

`.container` hat eine feste Breite für jede Bildschirmgröße in Bootstrap (xs, sm, md, lg);

`.container-fluid` dehnt sich aus, um die verfügbare Breite auszufüllen.

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

Abhängig von der Breite des Ansichtsfensters, auf dem die Webseite angezeigt wird, gibt die Containerklasse ihrem Div eine bestimmte feste Breite.

Andererseits wird Ihr `.container-fluid` Element ständig verkleinert, wenn Sie selbst kleinste Änderungen an der Breite Ihres Browsers vornehmen.

**Bootstrap-Container online lesen:** <https://riptutorial.com/de/twitter-bootstrap/topic/10908/bootstrap-container>

---

# Kapitel 8: Bootstrap-Komponenten

## Bemerkungen

Weitere Informationen finden Sie in der offiziellen Dokumentation unter <http://getbootstrap.com/javascript/> , von der die Komponentenliste abgeleitet ist.

## Examples

### Beispiele für Bootstrap-Komponenten

Bootstrap-Komponenten sind eine Sammlung optionaler jQuery-Plugins, die im Lieferumfang von Bootstrap enthalten sind.

Der Zweck der Bootstrap-Komponenten besteht darin, erweiterte Funktionen und Funktionen bereitzustellen, die ohne Javascript nur schwer oder gar nicht zu realisieren wären. Einige Komponenten sind rein funktional, während einige Komponenten dazu dienen, die Funktionalität für einige der speziellen Front-End-Widgets von Bootstrap zu definieren.

Beispiele hierfür sind [Übergangseffekte](#) , [modale Dialoge](#) , [Dropdown - Menüs](#) , [scrollspy](#) , [Tabs](#) , [tooltips](#) , [Warnungen](#) , [popovers](#) , [Knöpfe](#) , [Kollaps](#) , [Karussells](#) , und [Affixe](#) .

**Bootstrap-Komponenten online lesen:** <https://riptutorial.com/de/twitter-bootstrap/topic/6054/bootstrap-komponenten>

---

# Kapitel 9: Bootstrap-Themen

## Examples

### Bootstrap-Designs im Vergleich zu Regelüberschreibungen

#### Was sind Themen?

Es gibt verschiedene visuelle Erscheinungsbilder für Bootstrap, die aus Quellen wie [Bootswatch zu finden sind](#) , die die Datei `bootstrap.min.css` [ändern](#) . Sie können auf diese Weise auch ein eigenes Thema erstellen.

#### Wann müssen Designs geändert und neue Regeln zu einer `site.css`-Datei hinzugefügt werden?

Wann ändern Sie die

`bootstrap.min.css`

Datei und wann Sie Ihre eigene `.css`-Datei hinzufügen möchten, z

`site.css`

?

Manchmal gibt es Stilanforderungen, die unabhängig von dem von Ihnen verwendeten Thema erfüllt werden müssen. Diese Regeln sollten in Ihrer eigenen `.css`-Datei enthalten sein, z. B. `site.css` . Damit das *Hauptdesign* geändert werden kann, gelten die Regeln aus `site.css` *trotzdem* . Dazu müssen Sie nur das Bootstrap-Design und Ihre eigenen Regeln verknüpfen, um die vorhandenen zu überschreiben:

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

Auf diese Weise können Bootstrap-Designs jederzeit geändert werden, ohne die zwingenden Regeln zu verlieren, die von `site.css` angewendet *werden* .

Bootstrap-Themen online lesen: <https://riptutorial.com/de/twitter-bootstrap/topic/6381/bootstrap-themen>

# Kapitel 10: Bootstrap-Überprüfung

## Bemerkungen

- Diese Validierungstechnik kann nur für Eingaben verwendet werden, die sich innerhalb eines Formulars befinden.
- Eigenschaften müssen mindestens eine Validierungsanforderung aufweisen, um die Hervorhebung bei einer fehlgeschlagenen `onSubmit()` Validierung `onSubmit()`. Datentypen (mit Ausnahme von `string`) erfordern eine versteckte Datentypenforderung. Daher ist keine explizite Datenanmerkung erforderlich. Zeichenfolgen haben dies nicht.  
`[MinLengthAttribute(0)]` eine Validierungsprüfung zusammen mit den anderen Feldern zu erzwingen, fügen Sie die `[MinLengthAttribute(0)]`.

## Examples

### Verwenden von ASP.NET MVC und Datenanmerkungen

Fügen Sie in `<appSettings>` Folgendes zu `Web.config` (im Ordner `Views`) `<appSettings>`:

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

Fügen Sie das `BundleConfig.cs`-Paket `jqueryval` hinzu:

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

Fügen Sie allen Seiten, für die eine Validierung erforderlich ist, Folgendes hinzu (oder `_Layout.cshml`):

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

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

```
});  
</script>
```

Fügen Sie den relevanten Feldern des Modells Datenanmerkungen hinzu:

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

Fügen Sie in der Ansicht zu jeder Eingabe, die überprüft werden muss, Folgendes hinzu:

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

Fügen Sie der relevanten Controller-Aktion Folgendes hinzu, um die serverseitige Validierung hinzuzufügen:

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

---

## Beispieleingabe, für die eine Überprüfung erforderlich ist

Modell:

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

Aussicht:

```
<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>
```

## Wahlweise

Fügen Sie die folgende jQuery hinzu, um die Eingaben für Unschärfe sowie beim Senden zu überprüfen:

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

Bootstrap-Überprüfung online lesen: <https://riptutorial.com/de/twitter-bootstrap/topic/6388/bootstrap-uberprufung>

---

# Kapitel 11: Dropdowns

## Bemerkungen

Weitere Informationen finden Sie in der offiziellen Bootstrap-Dokumentation unter <http://getbootstrap.com/javascript/#dropdowns> , von der das grundlegende HTML-Verwendungsbeispiel abgeleitet ist.

## Examples

### Grundlegende HTML-Verwendung

Ein Bootstrap-Dropdown ist eine Bootstrap-Komponente, mit der ein HTML-Element die Anzeige eines Untermenü-Dropdowns veranlasst, wenn das Element angeklickt wird.

Hier ist ein einfaches Beispiel für die Verwendung von HTML:

```
<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>
```

Dropdown-Untermenüelemente können durch Einfügen von `li` Elementen innerhalb des `ul` Elements mit der `.dropdown-menu` .

Dropdowns online lesen: <https://riptutorial.com/de/twitter-bootstrap/topic/6033/dropdowns>

---

# Kapitel 12: Drucken in Bootstrap.

## Examples

### Grundlegende HTML-Verwendung

Mit den Druckelementen von Bootstrap können Sie festlegen, welche Elemente beim Drucken sichtbar sind und welche ausgeblendet werden sollen.

Um etwas sichtbar zu machen, verwenden Sie je nach Element und wie es beim Ausdruck aussehen soll:

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

Um etwas vor dem Ausdruck zu verbergen, verwenden Sie Folgendes:

```
.hidden-print
```

Drucken in Bootstrap. online lesen: <https://riptutorial.com/de/twitter-bootstrap/topic/6707/drucken-in-bootstrap->

# Kapitel 13: Formen

## Examples

### Grundform

Formularsteuerelemente weisen einige Standardstile auf, ohne spezielle Klassen zu verwenden.

Beschriftungen und Steuerelemente können jedoch für einen optimalen Abstand in `.form-group` Tags eingeschlossen werden.

```
<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>
```

### Schreibgeschützte und deaktivierte Eingänge

Fügen Sie das `readonly` Attribut hinzu, um Benutzereingaben zu verhindern. Ein Readonly-Feld kann nicht bearbeitet werden

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

Fügen Sie das `disabled` Attribut hinzu, um ein Eingabefeld zu deaktivieren. Ein deaktiviertes Feld kann ebenfalls nicht bearbeitet werden. Der Cursor ändert sich, um ihn auffälliger zu machen.

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

Formen online lesen: <https://riptutorial.com/de/twitter-bootstrap/topic/6251/formen>

---

# Kapitel 14: Gebrauchsklassen

## Examples

### Generiere `.hidden-*` Klassen für alle Haltepunkte - 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();
```

Gebrauchsklassen online lesen: <https://riptutorial.com/de/twitter-bootstrap/topic/6217/gebrauchsklassen>

# Kapitel 15: Gitternetz

## Einführung

In Bootstrap ist es möglich, Rasterzellen *in* anderen Spalten zu verwenden. Dies ist hilfreich, wenn Sie erweiterte responsive Layouts mit **mehreren Rasterebenen** erstellen.

## Bemerkungen

Auf die oben genannte Weise können wir so viele Spalten wie möglich haben.

## Examples

### Spalten verschachteln

```
<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>
```

Das Beispiel stammt aus [ <http://getbootstrap.com/css/#grid-nesting> ]

wie die Dokumentation der offiziellen Website vorschlägt

Um Ihren Inhalt mit dem Standardraster zu verschachteln, fügen Sie eine neue `.row` und einen Satz von `.col-sm-*`-Spalten innerhalb einer vorhandenen `.col-sm-*`-Spalte hinzu. Verschachtelte Zeilen sollten einen Satz von Spalten enthalten, der maximal 12 ergibt (es ist nicht erforderlich, dass Sie alle 12 verfügbaren Spalten verwenden).

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

Gitternetz online lesen: <https://riptutorial.com/de/twitter-bootstrap/topic/9088/gitternetz>

---

# Kapitel 16: Glyphicons

## Bemerkungen

Dieser Abschnitt enthält eine Übersicht über Bootstrap-Glyphicons und beschreibt die Verwendung von Glyphicons.

## Examples

### Wie benutze ich Glyphicons?

Twitter Bootstrap unterstützt Symbole, die als Glyphicons bezeichnet werden, und sie können mit allen HTML-Tags verwendet werden.

Alle Icons erfordern eine Basisklasse und eine individuelle Icon-Klasse.

Beachten Sie, dass Icon-Klassen nicht direkt mit anderen Komponenten kombiniert werden können. Verwenden Sie daher immer das innere `<span></span>` -Tag.

Wenn Ihr HTML-Code innere untergeordnete Elemente enthält, können Sie für dieses bestimmte Tag keine Symbolklassen verwenden.

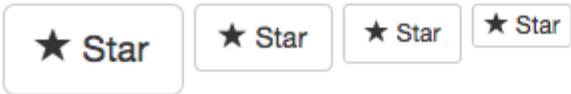
### Beispiele

Wenn Sie beispielsweise eine Bootstrap-Schaltfläche erstellen, sollte die Syntax für diese Schaltfläche folgendermaßen aussehen:

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

Im obigen Beispiel wird also eine einfache Bootstrap-Schaltfläche erstellt. Jetzt möchten Sie ein Glyphicon in diese Schaltfläche einfügen. Dazu fügen Sie einfach ein `<span>` -Element in ein `<button>` -Tag ein. So was:

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



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

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

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

---

# Kapitel 17: Gruppe auflisten

## Bemerkungen

Sie sollten wissen, wie man Bootstrap- [Buttons verwendet](#), und wenig Informationen zu [kontextuellen](#) Klassen.

## Examples

### Grundlegendes Beispiel

```
<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>
```

### Abzeichen

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

### Verknüpfte Elemente

```
<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>
```

### Schaltflächenelemente

```
<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>
```

## Deaktivierte Elemente

```
<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>
```

## Kontextuelle Klassen

```
<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>
```

## Benutzerdefinierter Inhalt

```
<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>
```

Gruppe auflisten online lesen: <https://riptutorial.com/de/twitter-bootstrap/topic/6347/gruppe-auflisten>

---

# Kapitel 18: Jumbotron

## Einführung

Jumbotron ist eine Standardkomponente von Bootstrap, um einige wichtige Inhalte auf Ihrer Website anzuzeigen. Es wird normalerweise direkt unter der Navigationsleiste vor dem Inhalt verwendet.

## Bemerkungen

Innerhalb des Jumbotron funktionieren auch alle Rastersysteme, Containerklassen und Zeilenklassen.

## Examples

### Grundlegender Jumbotron mit zwei Textzeilen und einem Button

Dies ist ein Jumbotron mit einem Titel, einem Inhalt und einer Schaltfläche.

#### Code

```
<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>
```

#### Ergebnis



# Title text

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

A button

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

# Kapitel 19: Karussells

## Bemerkungen

Weitere Informationen finden Sie in der offiziellen Dokumentation unter <http://getbootstrap.com/javascript/#carousel> , von der die grundlegenden HTML- und Javascript-Verwendungsbeispiele und Informationen abgeleitet werden.

Es ist zu beachten, dass Karussells aufgrund der Verwendung von CSS3-Übergängen / Animationen im IE 9 und früheren Versionen nicht ordnungsgemäß funktionieren.

## Examples

### Grundlegende HTML-Verwendung

Ein Bootstrap-Karussell ist eine Bootstrap-Komponente, die eine Diashow erstellt, die Elemente innerhalb des Karussells durchläuft.

Hier ist ein einfaches Beispiel für die Verwendung von HTML:

```
<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>
```

## Grundlegende Verwendung und Initialisierung von Javascript

Karussellkomponenten können über jQuery mit der Funktion `$('.carousel').carousel(options)` instanziiert werden, wobei `$('.carousel')` eine Referenz auf ein bestimmtes Karussell ist und `options` ein Javascript-Objekt, das das Karussell angibt Standardattribute.

Mit dem `options` können mehrere Eigenschaften definiert werden, die das Verhalten des Karussells beeinflussen. Diese Eigenschaften sind als solche definiert:

- Die `interval` akzeptiert einen Javascript- `number` , mit dem ein Benutzer die Zeitdauer definieren kann, für die das Karussell eine bestimmte Karussell-Folie anzeigt. Wenn der boolesche Wert `false` angegeben wird, wird das Karussell nicht automatisch durchlaufen.
- Die `pause` Eigenschaft akzeptiert einen Javascript- `string` das Verhalten umschaltet, wenn der automatische Zyklus des Karussells angehalten wird, wenn die Maus des Benutzers in das Karussell eintritt. Der standardmäßige (und einzige) akzeptierte Wert ist "hover".
- Die `wrap` Eigenschaft akzeptiert einen `boolean` Javascript-Typ, mit dem ein Benutzer definieren kann, ob das Karussell fortlaufend durchlaufen werden soll, ohne auf einer bestimmten Folie anhalten zu müssen.
- Die `keyboard` akzeptiert einen `boolean` Javascript-Typ, mit dem ein Benutzer definieren kann, ob das Karussell auf Tastaturreignisse reagieren soll oder nicht.

Hier ist ein Beispiel für die grundlegende Verwendung von Javascript:

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

Wie bei anderen Bootstrap-Komponenten können die Optionen des Karussells auch über Datenattribute in HTML festgelegt werden.

**Karussells online lesen:** <https://riptutorial.com/de/twitter-bootstrap/topic/1568/karussells>

---

# Kapitel 20: Migration zu Bootstrap 4

## Einführung

Bootstrap 4 ist eine **wichtige** Neuentwicklung. Es gibt viele Änderungen, die beim Upgrade von Bootstrap 3 zu beachten sind. Hier finden Sie Änderungen, Tipps und Beispiele für den Klassennamen, sowie die Migration Ihres Bootstrap 3.x-Codes zu Bootstrap 4.x.

## Bemerkungen

Dies ist nur ein kleines Beispiel, um detailliertere Beispiele zu befolgen.

## Examples

### Spaltenlayoutänderungen des Rastersystems in Bootstrap 4

Der erste Codeblock ist in Bootstrap 3 geschrieben. In Bootstrap 3 gibt es vier Arten von Spaltenspezifikationen, nämlich `col-md-*` `col-lg-*` `col-sm-*` `col-xs-*`. Ein vollständig ansprechendes Layout sieht in Bootstrap 3 folgendermaßen aus:

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

In Bootstrap 4 wurde eine neue Sm-Grid-Stufe unterhalb von 768px für eine detailliertere Steuerung hinzugefügt. Bootstrap 4 hat also `col-*` (`xs`), `col-sm-*`, `col-md-*`, `col-lg-*`, and `col-xl-*`. Was früher `.col-md-6` in v3 war, ist jetzt `.col-lg-6` in v4. Beachten Sie, dass das `-xs` Infix entfernt wurde, sodass `.col-6` 6 Spalteneinheiten am besonders kleinen (Standard) Haltepunkt darstellt.

Wenn wir nun dasselbe obige Beispiel in Bootstrap 4 schreiben möchten, würde es so aussehen:

```
<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>
```

## Gitterlayout Bootstrap 4

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

## Änderungen der Browserunterstützung

In twitter-bootstrap 4 wurde die Unterstützung für `IE8`, `IE9`, and `iOS 6` . `v4` ist jetzt nur noch `IE10+` and `iOS 7+` . Verwenden Sie für Sites, die diese benötigen, Version 3.

In twitter-bootstrap 4 wurde die offizielle Unterstützung für `Android v5.0 Lollipop's Browser` and `WebView` für `Android v5.0 Lollipop's Browser` and `WebView` hinzugefügt. Frühere Versionen von `Android Browser` und `WebView` werden nur `unofficially supported` .

## Entfernen Sie die Klassen

`Affix` wird aus Bootstrap 4 entfernt.

Es wird empfohlen, eine `position: sticky` stattdessen `position: sticky` Polyfill.

Wenn Sie `Affix` zum Anwenden zusätzlicher, nicht positionierter Stile verwendet haben, unterstützen die Polyfills möglicherweise Ihren Anwendungsfall nicht. Eine Option für solche Anwendungen ist die `ScrollPos-Styler`-Bibliothek eines Drittanbieters.

## Gemäß der Bootstrap-Dokumentation

Das `Affix`-Plugin für `jQuery` wurde gelöscht. Wir empfehlen die Verwendung einer `Position: stattdessen` klebrige `Polyfill`. Weitere Informationen und spezifische Empfehlungen für `Polyfill` finden Sie im `HTML5`-Eintrag "`Please`".

Wenn Sie Affix zum Anwenden zusätzlicher, nicht positionierter Stile verwendet haben, unterstützen die Polyfills möglicherweise Ihren Anwendungsfall nicht. Eine Option für solche Anwendungen ist die ScrollPos-Styler-Bibliothek eines Drittanbieters.

Wenn jemand von `Bootstrap v3` zu `Bootstrap v4` migriert, wird der Fallback-Ansatz unten angegeben:

## HTML

```
<header>

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

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

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

<div class="container" id="main">
  <h2>Hello Bootstrap 4.</h2>
  <div class="row">
    <div class="col-xs-12 col-sm-6 col-md-9">
      <p>3 wolf moon retro jean shorts chambray sustainable roof party. Shoreditch vegan
artisan Helvetica. Tattooed Codeply Echo Park Godard kogi, next level irony ennui twee squid
fap selvage. Meggings flannel Brooklyn literally small batch, mumblecore
      PBR try-hard kale chips. Brooklyn vinyl lumbersexual bicycle rights, viral fap
cronut leggings squid chillwave pickled gentrify mustache. 3 wolf moon hashtag church-key Odd
Future. Austin messenger bag normcore, Helvetica Williamsburg
      sartorial tote bag distillery Portland before they sold out gastropub
taxidermy Vice.</p>
    </div>
    <div class="col-xs-6 col-md-3">
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis pharetra codeply
```

varius quam sit amet vulputate. Quisque mauris augue, molestie tincidunt codeply condimentum vitae, gravida a libero. Aenean sit amet felis dolor, in sagittis nisi.

Sed ac orci quis tortor imperdiet venenatis. Duis elementum auctor accumsan. Aliquam in felis sit amet augue.

```
</p>
```

```
<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>
```

## 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

Die neue Bootstrap 4 Navbar-Komponente wurde gegenüber ihrem Vorgänger Bootstrap 3.x verbessert. In Bootstrap 4 reagiert die Navbar **standardmäßig** und verwendet **Flexbox**, um die Ausrichtung von Navbar-Inhalten wesentlich zu vereinfachen. Es ist auch eine einfache Sache, die neuen `navbar-toggleable-*` Klassen zu verwenden, um den Navbar-Haltpunkt zu ändern. Die Navbar verfügt jetzt über 6 Haltpunktgrößen oder "Zustände", so dass Sie problemlos eine der folgenden Navbar-Optionen verwenden können.

- Die Navbar bricht *niemals* in die vertikale mobile Ansicht ein und ist immer horizontal.
- Die Navbar ist *immer* in die vertikale Ansicht eingeblendet und wird über den Hamburger umgeschaltet.
- Die Navbar wird an *einem der 4 reagierenden Haltpunkte* in die vertikale Ansicht eingeblendet.

## Grundlegende Bootstrap 4-Navigationsleiste

```

<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>

```

Wie Sie dem obigen Code `navbar-header` die `navbar-header` Klasse aus Bootstrap 4 entfernt, und die `container-fluid` wird für eine Navbar mit voller Breite nicht mehr benötigt.

## Ändern des Navbar-Haltepunkts

Die `navbar-togglerable-md` Klasse lässt die obige Navbar vertikal (und zeigt das Toggle-Symbol) am mittleren Haltepunkt (md) von 992px an. Um dies auf einen anderen Haltepunkt zu ändern, müssen Sie einfach `navbar-togglerable-md` mit einem dieser `navbar-togglerable-md`.

- `navbar-togglerable` = Zusammenbruch bei `xs` Breite <576px
- `navbar-togglerable-sm` = Zusammenbruch bei `sm` Breiten <768px
- `navbar-togglerable-lg` = Zusammenbruch bei `lg` Breiten <1200px

## Bootstrap 4 Breakpoint Navbar Demo

## Ändern der Navbar-Ausrichtung

Mit Flexbox können wir die Ausrichtung der Navbar und deren Inhalt (Marke, Links, Formulare oder Text) einfach ändern. Der Standardinhalt der Navbar ist linksbündig. Natürlich gibt es viele andere Ausrichtungsszenarien ...

- Marke links (Standard), Links Mitte & Rechts
- Markcenter, Links links und rechts
- Marke links und Links rechts
- Eingabe von Marke, Links und Füllbreite
- Keine Marke, Linkscenter & rechts
- Marke links, Links rechts im Container
- Justierte Links (Füllbreite) zentriert

## Bootstrap 4-Navbar mit zentrierter Marke und Links / Rechts-Links

```

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

```

```

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

```

## Bootstrap 4 Navbar mit Brand Left, Links Center und Right

```

<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>

```

## Mehr zur Bootstrap 4 Navbar

[Anpassen von Farbe, Ausrichtung oder Höhe](#)

## Bootstrap 3 bis Bootstrap 4 CSS-Änderungen

Da es sich bei Bootstrap 4 um ein **grundlegendes** Umschreiben handelt, wurden viele der Bootstrap 3.x-Klassennamen geändert oder entfernt. Die Umstrukturierung von Komponenten wie der Navbar sowie die Einführung neuer CSS-Klassen und Flexbox-Unterstützung bedeuten, dass ein Upgrade auf 4.x *kein* einfacher Konvertierungsprozess von 3.x ist.

Es gibt jedoch einige Bootstrap 3.x-CSS-Klassen, für die ein bestimmter Bootstrap 4-Ersatz gilt.

### CSS-Klassenname / -selektor ändert sich von Bootstrap 3.3.7 zu 4 (Alpha 6)

{t} - steht für eine **Stufe** oder einen Haltepunkt (z. B. sm, md, lg usw.).

xs tier ist die Standardeinstellung und muss nicht angegeben werden: col-3 , col-6 usw.

{u} - steht für eine **Spaltengröße** (dh: 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	.Karte
.panel-Überschrift	.card-Header
.panel-title	.card-title
.panel-körper	.card-block
.panel-footer	.card-footer
.panel primär	.card-primary.card-inverse
.panel-erfolg	.card-success.card-inverse
.panel-info	.card-info.card-inverse
.Panel-Warnung	.card-warning.card-inverse

Bootstrap 3.x	Bootstrap 4
.panel-gefahr	.card-danger.card-inverse
.Gut	.card.card-block
.Miniaturansicht	.card.card-block
.list-inline> li	.list-Inline-Element
Dropdown-Menü> li	. Dropdown-Element
.nav navbar> li	.nav-item
.nav navbar> li> a	.nav-Link
.navbar-right	.ml-auto
.navbar-btn	.nav-item
.navbar-Fixed-Top	.fixed-top
.nav gestapelt	.flex-Spalte
.btn-default	.btn-sekundär
.img-responsive	.img-Flüssigkeit
.img-Kreis	runder Kreis
.Ich bin geerdet	.gerundet
.form-horizontal	(entfernt)
.Radio	.form-check
.checkbox	.form-check
.input-lg	.form-control-lg
.input-sm	.form-control-sm
.control-label	.form-control-label
tisch-kondensiert	.tabelle-sm
.pagination> li	.page-item
.pagination> li> a	.page-link
.Artikel	Karussell-Artikel

Bootstrap 3.x	Bootstrap 4
.text-Hilfe	.form-control-feedback
.Pull-Right	.schweben rechts
.pull-left	.float-left
.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
.Etikette	.Abzeichen
.Abzeichen	badge.badge-pill

### Siehe auch:

[Bootstrap 3.x zu 4 Migrationstool](#)

[Was ist neu in Bootstrap 4?](#)

## Bootstrap 4 Vertikal ausrichten

Erste Elemente zur **Mitte** oder **unten** ausrichten vertikal ist immer eine Herausforderung , mit CSS und Bootstrap gewesen. Die gewünschte vertikale Ausrichtung kann sich in einem übergeordneten Container oder relativ zu benachbarten Elementen befinden.

Da Bootstrap 4 **nun standardmäßig Flexbox ist** , gibt es viele verschiedene Ansätze für die vertikale Ausrichtung: [Auto-Ränder](#) , [Flexbox-Dienstprogramme](#) oder die [Anzeige-Dienstprogramme](#) zusammen mit den [Dienstprogrammen](#) für die [vertikale Ausrichtung](#) .

Zunächst scheinen die [Dienstprogramme](#) für die [vertikale Ausrichtung](#) naheliegend zu sein, diese funktionieren jedoch *nur* mit Inline- und Tabellenanzeigeelementen. Hier sind einige vertikale Ausrichtungsoptionen und -szenarien von Bootstrap 4 ...

### 1 - Vertikales Zentrum mit automatischen Rändern:

Eine Möglichkeit, vertikal zu zentrieren, ist `my-auto`. Dadurch wird das Element in seinem Container zentriert. Mit `h-100` wird beispielsweise die ganze Höhe der Zeile erreicht, und `my-auto` zentriert die `col-sm-12` vertikal.

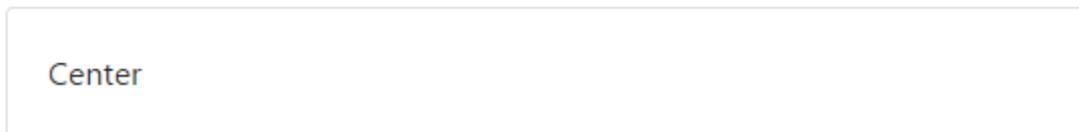
```
<div class="row h-100">
  <div class="col-sm-12 my-auto">
    <div class="card card-block w-25">Card</div>
  </div>
</div>
```

## Vertikales Zentrum mit Auto Margins Demo

`my-auto` repräsentiert Ränder auf der vertikalen Y-Achse und entspricht:

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

## 2 - Vertikales Zentrum mit Flexbox:



**Taller**  
With supporting text  
[Outline](#)

Da Bootstrap 4 `.row` jetzt `display:flex`, können Sie einfach die Funktion zum `align-self-center` in jeder Spalte verwenden, um sie vertikal zu `.row` ...

```
<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>
```

oder verwenden Sie `align-items-center` für die gesamte `.row`, um alle `col-*` in der Zeile vertikal auszurichten ...

```
<div class="row align-items-center">
```

```

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

```

## Demo für vertikale Höhe mit unterschiedlichen Höhen

### 3 - Vertikales Zentrum mit Display Utils:

Bootstrap 4 verfügt über [Anzeigeinstrumente](#), die für die `display:table`, `display:table-cell`, `display:inline` usw. Diese können mit den [vertikalen Ausrichtungshilfsmitteln](#) zum Ausrichten von Inline-, Inline-Block- oder Tabellenzellenelementen verwendet werden.

```

<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>

```

## Vertikales Zentrum Verwenden von Display Utils Demo

### Bootstrap 4 Zentrieren

Wie ein Element, eine Spalte oder ein Inhalt innerhalb einer Spalte zentriert wird, funktioniert in Bootstrap 4 anders.

## Horizontales Zentrum

- `text-center` wird weiterhin für die `display:inline` Elemente
- `mx-auto` ersetzt den `center-block` in die `Center-` `display:block`
- `offset-*` *oder* `mx-auto` können Sie `offset-*`

`mx-auto` (automatische x-Achsen-Ränder) von `display:block` oder `display:flex` Elemente mit einer *definierten Breite* (`%`, `vw`, `px` usw.). **Flexbox wird standardmäßig** für Rastersäulen verwendet, daher gibt es auch verschiedene Flexbox-Zentrierungsmethoden.

**Text oder Inline-Elemente zentrieren:** `text-center`

```

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

```

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

**Zentrierte** `display:block` oder `display:flex : mx-auto`

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

**Spalten mit** `offset-* : offset-*`

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

Spalten können *auch* mit: `mx-auto` zentriert werden

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

[Demo Bootstrap 4 Horizontales Zentrieren](#)

---

## Vertikales Zentrum

Informationen zur vertikalen Zentrierung in Bootstrap 4 (y-Achse) finden Sie in den folgenden Dokumenten: [Bootstrap 4 Vertical Align](#)

### Bootstrap 4 Column Order

Das Ändern der Reihenfolge (oder Position) war in Bootstrap 3 mithilfe der Push-Pull-Klassen möglich. In Bootstrap 4 funktionieren die **Push-Pull**- Klassen weiterhin, und zusätzlich kann die **Flexbox-Reihenfolge** verwendet werden.

In Bootstrap 4 sind die **Push- Pull**- Klassen jetzt `push-{viewport}-{units}` und `pull-{viewport}-{units}` und der `xs-` Infix wurde entfernt. Betrachten Sie dieses Beispiel, in dem die Spaltenreihenfolge für `xs` und `sm` in das Layout 1-3-2 geändert wird:

```
<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

---

Da die neue Version 4 Flexbox ist, ist **eine weitere Option**, die **Flexbox Utility** - Klassen zu verwenden, um die Reihenfolge der Spalten zu ändern. Mit der **Flexbox**- Reihenfolge können nun **col-\*-12 Spalten mit Farbe col-\*-12** umgedreht werden.

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

## Flexbox Bestell-Demo

Migration zu Bootstrap 4 online lesen: <https://riptutorial.com/de/twitter-bootstrap/topic/9090/migration-zu-bootstrap-4>

# Kapitel 21: Modale Dialoge

## Bemerkungen

Weitere Informationen finden Sie in der offiziellen Dokumentation unter <http://getbootstrap.com/javascript/#modals> , von der das Beispiel 'Basic HTML Usage' abgeleitet wurde.

## Examples

### Grundlegende HTML-Verwendung

Ein modifizierter Bootstrap-Dialog ist eine Bootstrap-Komponente, die ein modales Dialogfenster erstellt, das über den Inhalt auf Seitenebene schwebt.

Hier ein Beispiel für die grundlegende Verwendung eines Bootstrap-Modaldialogfelds in HTML:

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

### Grundlegende Verwendung und Initialisierung von Javascript

Modale Dialogkomponenten können über die jQuery-Funktion mit der Funktion `$('#myModal').modal(options)` instanziiert werden, wobei `$('#myModal')` eine Referenz auf den spezifischen modalen Dialog ist und `options` ein Javascript-Objekt ist die Standardattribute des modalen Dialogs.

Mit dem `options` können mehrere Eigenschaften definiert werden, die das Verhalten des modalen Dialogfelds beeinflussen. Diese Eigenschaften sind als solche definiert:

- Die `backdrop` Eigenschaft ermöglicht es einen Benutzer zu definieren , ob sie ein grauer Hintergrund Overlay hinter dem modalen Dialogfeld angezeigt werden sollen. Beide booleschen Werte und der String "static" werden erkannt. Wenn "statisch" angegeben ist,

wird das modale Dialogfeld nicht geschlossen, wenn ein Benutzer auf die Hintergrundüberlagerung klickt.

- Mit der `keyboard` kann ein Benutzer festlegen, ob der modale Dialog geschlossen werden soll, wenn die Escape-Taste auf der Tastatur gedrückt wird.
- Mit der `show` Eigenschaft kann ein Benutzer festlegen, ob der modale Dialog angezeigt werden soll, wenn der modale Modus initialisiert wird.

Hier ist ein Beispiel für die grundlegende Verwendung von Javascript:

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

Wie bei anderen Bootstrap-Komponenten können die Optionen des Modals auch über Datenattribute in HTML angegeben werden.

Modale Dialoge online lesen: <https://riptutorial.com/de/twitter-bootstrap/topic/5927/modale-dialoge>

---

# Kapitel 22: Modals

## Bemerkungen

Modals benötigen bootstrap.min.js, um ordnungsgemäß zu funktionieren.

Weitere Informationen finden Sie hier: <http://getbootstrap.com/javascript/#modals>

## Examples

### Basic HTML Modal

Ein Modal ist ein Dialogfenster, das über der aktuellen Seite angezeigt werden kann.

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

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

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

# Kapitel 23: Navbar

## Examples

### Grundlegende Navigationsleiste (am oberen Rand der Seite befestigt)

```
<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>
```

### Untermenü in der Navigationsleiste

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

```

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

```

## Navbar-Teiler

```

<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>

```

## Aktuellen Navigationslink "aktiv" halten

```

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

```

## Navbar-Haltepunkt ändern (Handy vs. Normal)

### max-width ist der Haltepunkt

```

@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;
    }
}

```

## Schließen Sie die minimierte Navigationsleiste, wenn Sie außerhalb der Navigationsleiste klicken

```

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');
        }
    }
});

```

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

---

# Kapitel 24: Navigationsmenüs

## Examples

### Horizontales Pillenmenü

```
<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>
```

### Vertikale Pillenmenü

```
<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>
```

### Responsive horizontale Pille mit voller Breite

```
<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>
```

Navigationsmenüs online lesen: <https://riptutorial.com/de/twitter-bootstrap/topic/6404/navigationsmenus>

# Kapitel 25: Navs

## Examples

### Bootstrap-Navs

In Bootstrap verfügbare Navs verfügen über ein gemeinsames Markup, beginnend mit der Basis-`.nav`-Klasse sowie gemeinsam genutzten Status. Tauschen Sie Modifikator-Klassen aus, um zwischen den einzelnen Stilen zu wechseln.

#### Tabs

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

#### Pillen

```
<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>
```

#### Gerechtfertigt

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

#### Mit 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>
```

Navs online lesen: <https://riptutorial.com/de/twitter-bootstrap/topic/6505/navs>

# Kapitel 26: Panels

## Bemerkungen

Die Panel-Komponente in Bootstrap ist eine (umrahmte) Box mit einigen Füllungen um den Inhalt und optional Kopf- und Fußzeile-Containern.

## Examples

### Grundlegendes Beispiel

Standardmäßig `.panel` das `.panel` nur einige grundlegende Rahmen und Füllungen an, um Inhalte zu enthalten.

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

### Panel mit Überschrift

`.panel-heading` Ihrem Panel `.panel-heading` einfach einen Überschriftencontainer mit `.panel-heading`. Sie können auch beliebige `<h1>-<h6>` mit einer `.panel-title` Klasse hinzufügen, um eine vorab gestylte Überschrift hinzuzufügen. Die Schriftgrößen von `<h1>-<h6>` werden jedoch durch die `.panel-heading` überschrieben.

`.panel-title` für eine korrekte Link-Farbe sicher, dass Sie Links in Überschriften innerhalb des `.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 mit Fußzeile

`.panel-footer` Sie Schaltflächen oder sekundären Text in die `.panel-footer` . Beachten Sie, dass Bedienfeldfußzeilen bei kontextabhängigen Variationen **keine** Farben und Rahmen erben, da sie nicht im Vordergrund sein sollen.

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

Panels online lesen: <https://riptutorial.com/de/twitter-bootstrap/topic/2848/panels>

# Kapitel 27: QuickInfo

## Bemerkungen

Der Tooltip ist ein Benutzeroberflächenelement, das wie ein kleines Popup-Fenster aussieht. Es wird normalerweise ausgelöst, wenn ein Benutzer seinen Zeiger über ein anderes Element bewegt, ohne darauf zu klicken.

Aus Leistungsgründen müssen Tooltips mit jQuery initialisiert werden. Mit dem folgenden Code werden alle Tooltips im DOM aktiviert:

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

## Examples

### Tooltips positionieren

Standardmäßig wird die QuickInfo über dem Element angezeigt. Wir können ein `data-placement`, um die Position des Tooltips oben, unten, links oder rechts vom Element festzulegen.

```
<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

Wir können auch `data-placement="auto"`, um den Tooltip dynamisch neu auszurichten. Der Tooltip wird im nächsten Beispiel nach Möglichkeit links angezeigt, ansonsten rechts.

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

### Basisbeispiel

Um einen Tooltip zu erstellen, müssen wir nur das Attribut `data-toggle="tooltip"` und einen `title` zum HTML-Element hinzufügen, das den Tooltip enthält. Das Titelattribut wird verwendet, um den Text anzugeben, der in der QuickInfo angezeigt wird.

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

Hello world!

Hover over me

QuickInfo online lesen: <https://riptutorial.com/de/twitter-bootstrap/topic/3731/quickinfo>

---

# Kapitel 28: Rastersystem

## Einführung

Das Rastersystem von Bootstrap besteht aus 12 Einheiten, die als **Columns** ( `.col-*-*` CSS-Klassen) bezeichnet werden und dazu dienen, den Inhalt des Viewports von *links nach rechts zu gestalten*. Spalten sind in **Zeilen** ( `.row` CSS-Klasse) enthalten, um horizontale `.row` zu erstellen. Zeilen sind in einem festen oder voller Breite **Container** (platziert `.container` oder `.container-fluid` bezeichnet) für die richtige Ausrichtung. Spalten haben eine Auffüllung, die einen Abstand zwischen den Inhalten in den Spalten erzeugt.

## Bemerkungen

Bootstrap umfasst ein reaktionsfähiges, mobiles erstes Fluid-Grid-System, das mit zunehmender Geräte- oder Viewport-Größe bis zu 12 Spalten skaliert. Es enthält vordefinierte Klassen zum schnellen Erstellen von Seitenlayouts durch eine Reihe von Zeilen und Spalten, in denen sich Ihre Inhalte befinden.

## Examples

### Medien-Anfragen

Medienabfragen in Bootstrap ermöglichen das Verschieben, Anzeigen und Ausblenden von Inhalten basierend auf der Größe des Darstellungsbereichs. Die folgenden Medienabfragen werden in LESS-Dateien verwendet, um die Schlüsselhaltepunkte im Bootstrap-Grid-System zu erstellen:

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

Gelegentlich werden diese um eine maximale Breite erweitert, um CSS auf eine engere Gruppe von Geräten zu beschränken:

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

## Bootstrap-Raster (Haltepunkte)

Neben dem Konzept der [Säuleneinheiten](#) verfügt Bootstrap über verschiedene **Haltepunkte** oder Rastergrößen, so genannte Tiers. Das Bootstrap 3-Raster verfügt über vier (4) Ebenen, um unterschiedliche Bildschirmbreiten (oder Ansichtsfenster) aufzunehmen. Die Bootstrap 3-Ebenen sind `xs`, `sm`, `md` und `lg`. Die Rasterzellen von Bootstrap werden durch verschiedene `col-{breakpoint}-{units}` CSS-Klassen identifiziert.

Jede Grid-Tier-Ebene **umfasst einen Bereich**, der die Bildschirmbreiten typischer Geräte wie Desktops, Laptops, Tablets und Smartphones optimal anpasst.

Bootstrap verwendet CSS-Medienabfragen, um responsive Haltepunkte zu erstellen, die eine Begrenzung für jede Rastergröße festlegen. Mit diesen Rastergrößen können Sie das Layout von Spalten ändern, um die unterschiedlichen Bildschirmbreiten und Geräte optimal an das Responsive Design anzupassen.

- `col-xs-*` - für *kleinste* Bildschirmbreiten wie Smartphones <768 px
- `col-sm-*` - für *kleine* Bildschirmbreiten wie Smartphones und Tablets >= 768 px
- `col-md-*` - für *mittlere* Bildschirmbreiten wie Tablets und Laptops >= 992 px
- `col-lg-*` - für *große* Bildschirmbreiten wie Desktops >= 1200 px

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

Referenz: [Rastersystem](#)

### Gleiche Spaltenbreite für jedes Gerät

Um eine Spalte zu erstellen, die immer 50% der Breite des Ansichtsfensters (auf allen Geräten) beträgt, können Sie für jede Schicht `col-*-6` festlegen.

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

Dies ist jedoch ein unnötiges zusätzliches Markup, da `col-xs-6` 6 Einheiten auf `xs` und mehr bedeutet. Die kleinste von Ihnen festgelegte Ebene (`xs`, `sm` oder `md`) definiert auch die Größe für größere Bildschirmbreiten. Legen Sie für *dieselbe* Spalte in allen Ebenen die Breite für das kleinste Ansichtsfenster fest.

*Kürzere Code:*

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

## Unterschiedliche Säulenbreite für jedes Gerät (Responsive Design)

Die `col-*-*` Klassen können **kombiniert werden**, um die Spaltenbreiten bei verschiedenen Rastergrößen zu steuern.

Erstellen Sie beispielsweise eine Spalte mit *50%* Breite in der `sm` Tier und eine *25%* Breite in der `md` Tier ...

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

Die Gitter `sm`, `md` und `lg` "stapeln" alle vertikal bei Darstellungsbreite von weniger als 768 Pixel. An diesem Punkt passt das `xs` Raster. Spalten, die die `col-xs-*` -Klassen verwenden, werden nicht vertikal gestapelt und werden auf den kleinsten Bildschirmen weiter verkleinert.

## Bootstrap-Zeilen und -Säulen

Das Rastersystem von Bootstrap besteht aus **12 Einheiten, den sogenannten Columns (Spalten)**, mit denen der Inhalt horizontal im Ansichtsfenster angeordnet werden kann.

Der Grund für ein 12-Einheiten-Raster (anstelle von 10, 16 usw.) ist, dass 12 gleichmäßig in 6 (Hälften), 4 (Quartale) und 3 (Drittel) unterteilt ist. Dies erleichtert die Anpassung an verschiedene Layouts. Die Rasterspalten von Bootstrap werden durch verschiedene `col-{breakpoint}-{units}` CSS-Klassen identifiziert. [Weitere Informationen zu Breite und Haltepunkten des Ansichtsfensters \(AKA-Ebenen\)](#)

Zum Beispiel stellt `col-md-3` eine Spalte dar, die 3 der 12 Einheiten (oder 25%) über ein Viewport mit mittlerer Breite (`md`) aufnimmt. Um eine Spaltenbreite in Ihrem Layout zu verwenden, verwenden Sie einfach die entsprechende `col-{breakpoint}-{units}` -Klasse in Ihrem HTML-Markup.

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

Die Säulenbreite ist fließend (keine feste Breite), daher verbrauchen die Säulen einen *Prozentsatz* ihres Behälters.

### Säuleneinheiten in Bootstrap 3

- `col-*-1` : 1 von 12 (8.33333333% Breite)
- `col-*-2` : 2 von 12 (16.66666667% Breite)
- `col-*-3` : 3 von 12 (25% Breite)
- `col-*-4` : 4 of 12 (33.33333333% Breite)

- `col-*-5` : 5 von 12 (41.66666667% Breite)
- `col-*-6` : 6 von 12 (50% Breite)
- `col-*-7` : 7 von 12 (58.33333333% Breite)
- `col-*-8` : 8 von 12 (66.66666667% Breite)
- `col-*-9` : 9 von 12 (75% Breite)
- `col-*-10` : 10 von 12 (83.33333333% Breite)
- `col-*-11` : 11 von 12 (91.66666667% Breite)
- `col-*-12` : 12 von 12 (100% Breite)

## Demo - Bootstrap 12 Säuleneinheiten

### Die Bootstrap Row

Die Bootstrap- `.row` Klasse enthält die Spalten. Säulen sollten *immer* in Reihen platziert werden und Reihen sollten immer in einem Behälter ( `container` oder `container-fluid` ) platziert werden. Die Zeile verwendet negative Ränder (-15px), um den richtigen Abstand zwischen dem Inhalt der Spalte und dem Rand des Browsers sicherzustellen. Zeilen werden verwendet, um Spalten horizontal zu gruppieren.

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

Spalten füllen die `.row` horizontal von links nach rechts und werden alle 12 Spalteneinheiten in eine neue Zeile **umbrochen** . Daher können Sie `.row` s verwenden, um **horizontale Unterbrechungen** zu erstellen, oder Sie können **mehr als 12** Spalteneinheiten in einem einzelnen `.row` Element `.row` , um **Spalten** zu haben , die den Viewport vertikal `.row` (oder stapeln).

Bei der Verwendung von Spaltenumbrüchen (mehr als 12 Einheiten in einer `.row` ) müssen Sie **responsive Reset (oder Clearfixes) verwenden**, um sicherzustellen, dass ungerade Spalteninhalte gleichmäßig **umbrochen** werden. Dies ist wichtig, wenn der Inhalt der Spalten in der Höhe variiert.

### Mehr zu Bootstrap Grid Columns & Rows

[Probleme mit dem Bootstrap 3-Fluidgitter?](#)

[Bootstrap 3 - verschachtelte Zeile Kann ich mehr als 12 Spalten hinzufügen?](#)

[Bootstrap Zeile und Spalte Erklärung](#)

[Wie das Bootstrap-Grid funktioniert \(Medium\)](#)

### Behälter

Bootstrap erfordert ein umfassendes Element, um den Inhalt der Site zu verpacken und unser Rastersystem unterzubringen. Sie können einen von zwei Containern für Ihre Projekte auswählen.

Verwenden Sie die `.container` Klasse für einen responsiven Container mit fester Breite.

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

Verwenden Sie die `.container-fluid` Klasse für einen Container mit voller Breite, der sich über die gesamte Breite Ihres Viewports erstreckt.

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

Hinweis: Container können nicht verschachtelt werden (Sie können einen Container nicht in einen anderen Container stellen), da sich das `padding` und vieles mehr.

## Versetzung von Spalten

Diese Klassen vergrößern den linken Rand einer Spalte um \* Spalten. Zum Beispiel bewegt `.col-md-offset-4` `.col-md-4` über vier Spalten.

```
<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>
```

## Säulenordnungsmanipulation mit Push und 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>
```

Dies ändert die Reihenfolge der integrierten Rasterspalten.

Syntax: `.col-md-push- *` und `.col-md-pull- *`.

Mehr:

[Säulenreihenfolge-Bearbeitung mit col-lg-push und col-lg-pull in Twitter Bootstrap 3](#)

Bootstrap 3: Push / Pull-Spalten nur bei kleineren Bildschirmgrößen  
Spaltenanordnung und -stapelung in Bootstrap 3

Rastersystem online lesen: <https://riptutorial.com/de/twitter-bootstrap/topic/3330/rastersystem>

# Kapitel 29: Säulen

## Examples

### Responsive Spalten gleicher Höhe (nur CSS oder SASS)

Sie müssen ein div mit der Klasse `.row-height` in der Zeile hinzufügen und den Spalten `.col-height` hinzufügen. Wenn Sie den Effekt auf eine bestimmte Medienabfrage beschränken möchten, verwenden Sie einfach die `.row-height` Klassen `.row-height` und `.col-height` : beispielsweise `.row-sm-height` mit `.col-sm-height` .

CSS-Version:

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

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

```
}

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

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

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

```

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

```

## SASS-Version (erforderliche 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;
}
}
}
```

Säulen online lesen: <https://riptutorial.com/de/twitter-bootstrap/topic/6469/saulen>

---

# Kapitel 30: Seitennummerierung

## Einführung

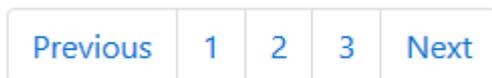
Paginierungs-Links zeigen an, dass eine Reihe verwandter Inhalte auf mehreren Seiten vorhanden ist. Normalerweise werden diese verwendet, wenn ein mehrseitiger Ansatz für lange Inhaltslisten die allgemeine Leistung verbessert, beispielsweise in Suchergebnissen oder Posteingängen.

## Examples

### Ein einfaches Paginierungsbeispiel

```
<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>
```

was gibt dir:



Seitennummerierung online lesen: <https://riptutorial.com/de/twitter-bootstrap/topic/10605/seitennummerierung>

---

# Kapitel 31: Tabellen

## Examples

### Einfache Tabelle

Die Stileffekte können je nach Thema variieren. Mit der `.table`-Klasse wird ein einheitliches und konsistentes Erscheinungsbild für Tabellen in einer Anwendung erstellt:

```
<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>
```

Tabellen online lesen: <https://riptutorial.com/de/twitter-bootstrap/topic/6299/tabellen>

---

# Kapitel 32: Tabellen

## Bemerkungen

**Inhaltsreihenfolge und komplexe Tabellen** Beachten Sie, dass der Stil der Tabellenumwandlung die visuelle Reihenfolge des Inhalts ändert. Stellen Sie sicher, dass Sie diesen Stil nur auf **wohlgeformte** und einfache Datentabellen anwenden (und insbesondere nicht für Layouttabellen verwenden), wobei für jede Zeile und Spalte die entsprechenden Tabellenkopfzeilen vorhanden sind.

Darüber hinaus funktioniert diese Klasse nicht ordnungsgemäß für Tabellen mit Zellen, die sich über mehrere Zeilen oder Spalten erstrecken (mit Rowspan- oder Colspan-Attributen).

## Examples

### Grundtabelle

Bootstrap definiert ein benutzerdefiniertes Styling für eine Tabelle mit der Klasse `.table`. Einfach die `.table` Klasse zu einer beliebigen `<table>`, um horizontale Trennlinien und Füllungen.

```
<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>
```

### Tisch mit modernem Styling

Bootstrap bietet eine Reihe von Klassen für die erweiterte Tabellengestaltung.

---

## Gestreifte Reihen

Sie haben eine Tabelle mit gestreiften Zeilen, wenn Sie die Klasse `.table-striped` hinzufügen:

```
<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>
```

Beachten Sie, dass:

Gestreifte Tabellen werden über den `:nth-child` CSS-Selektor gestaltet, der in Internet Explorer 8 nicht verfügbar ist.

---

## Tisch mit Rahmen

Sie haben eine Tabelle mit Rändern an allen Seiten der Tabelle und Zellen, wenn Sie die Klasse mit dem Zusatz `.table-bordered` hinzufügen:

```
<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>
```

---

## Schweben Sie in den Reihen

Wenn Sie die `.table-hover` Klasse hinzufügen, haben Sie eine Tabelle mit hervorgehobenen Zeilen, wenn der Benutzer über eine Zeile `.table-hover` :

```
<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>
```

---

## Kurzer Tisch

Wenn Sie die `.table-condensed` Klasse hinzufügen, wird die Standardauffüllung der Zelle halbiert, sodass Sie eine kompaktere Tabelle haben:

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

---

## Kontextuelle Klassen

Bootstrap-Tabellen unterstützen kontextabhängige Farben. Um die Hintergrundfarbe einer Tabellenzeile oder Zelle zu ändern, müssen Sie lediglich eine der folgenden kontextuellen Klassen

.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>
```

## Responsive Tabellen

Sie müssen jede `.table` in den HTML-Container mit der `.table-responsive` Klasse `.table-responsive`, um responsive Tabellen zu erstellen:

```
<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>
```

Responsive Tabellen werden auf kleinen Geräten horizontal gescrollt (<768px). Bei Bildschirmen mit einer Breite von mehr als 768 Pixeln gibt es keine Unterschiede.

## Table Reflow - Vertikale Header

**Eine Tabelle mit vertikalen Kopfzeilen erhalten.**

Twitter Bootstrap unterstützt jetzt vertikale Header auf einer gut formatierten normalen Tabelle. Um dies zu erreichen, verwenden `.table-reflow` einfach die `.table-reflow` Klasse

Verwenden Sie die `.table-reflow` Klasse twitter bootstrap für eine gut geformte Tabelle, um eine Tabelle mit vertikalen Headern zu erhalten. Außerdem können Sie `.table-striped` die Verwendung von `.table-striped` und `.table-hover` für das Schweben auf Spalten verwenden.

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

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

Sie sollten die v4-Alpha-Dokumente hier überprüfen: [twitter-bootstrap .table-reflow](#)

Tabellen online lesen: <https://riptutorial.com/de/twitter-bootstrap/topic/6360/tabellen>

# Kapitel 33: Tabs

## Examples

### Grundlegendes HTML

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

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

Dadurch wird ein Registerkartensatz mit 3 Registerkarten und 3 zugehörigen Inhaltsdivs erstellt.

### Animierte Registerkarten

Um Tabs verblassen in, fügen `.fade` zu jedem `.tab-pane`. Der aktive Registerkartenbereich muss auch über eine `.in` Klasse verfügen, damit der ursprüngliche Inhalt sichtbar wird.

```
<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>
```

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

# Kapitel 34: Tasten

## Syntax

- Klassen: `.btn-default` | `.btn-primary` | `.btn-Erfolg` | `.btn-info` | `.btn-Warnung` | `.btn-danger` | `.btn-link`;
- Größen: `.btn-lg` | `.btn-md` | `.btn-sm` | `.btn-xs`;
- Zustand: aktiv | disabled.

## Examples

### Button-Klassen

Bootstrap bietet mehrere Klassen zum Gestalten von Schaltflächen und zum Hervorheben.

Bootstrap-Schaltflächen können erstellt werden, indem einem Element die `.btn` Klasse hinzugefügt wird.

Bootstrap-Klasse	Rolle (Farbe)
<code>.btn-default</code>	Standardknopf (weiß)
<code>.btn-primary</code>	Bietet zusätzliches visuelles Gewicht und identifiziert die primäre Aktion (blau)
<code>.btn-success</code>	Wird verwendet, um eine erfolgreiche Aktion anzuzeigen (grün)
<code>.btn-info</code>	Kontexttaste zur Bereitstellung von Informationen (hellblau)
<code>.btn-warning</code>	Zeigt an, dass der Benutzer mit Vorsicht vorgehen sollte (gelb)
<code>.btn-danger</code>	Weist auf eine gefährliche oder negative Aktion hin (rot)
<code>.btn-link</code>	Lassen Sie den Button wie ein Anker-Tag aussehen.

### Tastengrößen

Sie können auch verschiedene Größen von Schaltflächen mit den `.btn-size` erstellen

Bootstrap-Klasse	Ergebnis
<code>.btn-lg</code>	Erzeugt eine größere Schaltfläche
<code>.btn-sm</code>	Erzeugt eine kleinere Schaltfläche

Bootstrap-Klasse	Ergebnis
<code>.btn-xs</code>	Erzeugt eine extra kleine Schaltfläche
<code>.btn-block</code>	Schaltflächen werden zu Elementen auf Blockebene und erstrecken sich über die gesamte Breite des übergeordneten Elements

## Schaltfläche aktivieren

Die `active` Klasse lässt eine Schaltfläche gedrückt erscheinen.

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

## Deaktivieren Sie eine Schaltfläche

Wenn Sie die `disabled` Klasse zu einer Schaltfläche hinzufügen, ist die Schaltfläche nicht anklickbar und zeigt einen verbotenen Cursor, wenn Sie mit der Maus darüber fahren.

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

## Schaltflächen horizontal zusammen rendern

Mit der `.btn-group` Klasse können mehrere Schaltflächen horizontal gerendert werden. Wickeln Sie einfach Ihre Schaltflächen in ein Containerelement und geben Sie diesem die Klasse `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>
```

## Schaltflächen vertikal rendern

Wenden Sie die Klasse `.btn-group-vertical` auf das Containerelement an

```
<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>
```

## Schaltflächengruppe in voller Breite aufnehmen

In einem `.btn-group` Element eingeschlossene Schaltflächen `.btn-group` nur so viel Breite wie benötigt. Um die Gruppe über die gesamte Breite des Bildschirms zu bringen, verwenden `.btn-group-justified` stattdessen `.btn-group-justified`.

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

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

Tasten online lesen: <https://riptutorial.com/de/twitter-bootstrap/topic/4757/tasten>

# Kapitel 35: Verwenden von Clearfix in Zeilen und Spalten

## Einführung

Beim Erstellen von erweiterten Layouts kann es Szenarien geben, in denen Sie **mehr als 12 Spalteneinheiten** in einem einzelnen `.row` Element verwenden müssen. Das Konzept des **Spaltenumbruchs** und der **responsiven Rücksetzungen (AKA-Clearfixes)** ist für das Verständnis des responsiven Designs mit Bootstrap *wesentlich*.

[Grundlagen des Bootstrap-Gitters](#)

## Bemerkungen

Bootstraps-Gitter sind bemerkenswert kraftvoll und elegant. Beachten Sie jedoch, dass der Name des Frameworks "Bootstrap" und nicht "WeDidItForYou" lautet. Bootstrap **ermöglicht** responsives Design, **garantiert** es jedoch nicht.

Es liegt immer noch an Ihnen, Ihr Design wirklich ansprechend zu gestalten und Ihren Benutzern die bestmögliche Endbenutzererfahrung zu bieten.

## Examples

### Der naive erste Versuch

Bevor wir beginnen, definieren wir ein paar CSS für die Beispiele. Dies ist der `head` unserer Probe. Ich verwende beim Testen immer den `border-radius` und die `background-color`, da es einfach ist, Zellteilungen zu sehen, ohne eine Randgröße hinzuzufügen, die die Größe der Zellen beeinflussen könnte.

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

```

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

```

Lassen Sie uns ein Raster definieren und die perfekten Ergebnisse für alle Ansichtsfenstergrößen betrachten!

Verwenden von `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`



Die beiden vorherigen Bilder zeigen das Rendern bei mittleren und kleinen Bildschirmgrößen. Denken Sie daran, wir erhalten VIER Spalten auf medium + wegen `col-md-3` und ZWEI Zellen bei

small- wegen `col-xs-6` .

Sieht ziemlich gut aus, oder? Ich denke wir sind hier fertig! Sagte eine Menge naiver Bootstrap-Sites, die nur darauf warten, zu brechen ...

## Das Höhenproblem

In unserem " [naiven Beispiel](#) " hatten alle unsere Zellen die gleiche Höhe. Der Browser brach willkürlich die Zeilen genau dort, wo wir wollten, und alles schien mit der Welt recht zu sein. Bis die Höhe ins Bild kommt.

Nehmen wir das vorige Beispiel und geben einigen Zellen eine gewisse Höhe, vielleicht so, wie Sie es auf einer Dashboard-Seite sehen würden.

```
<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>
```

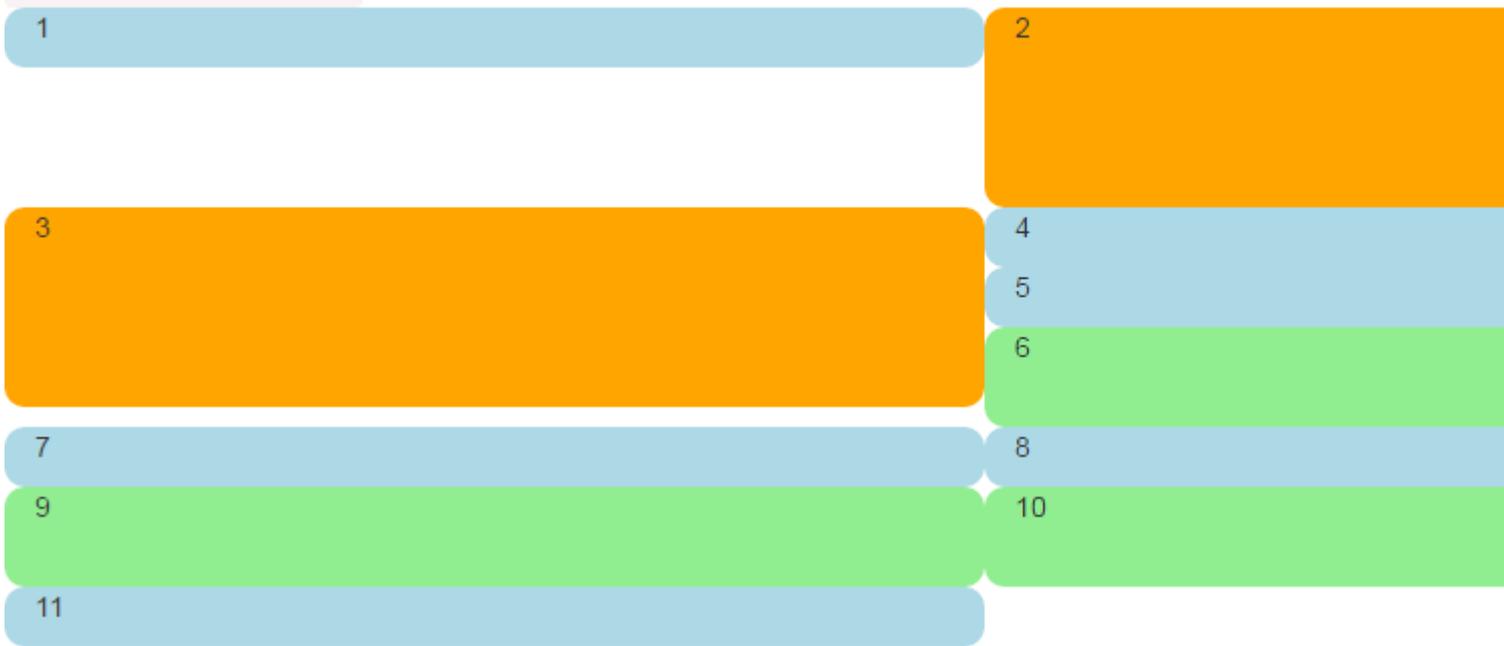
Hier haben wir einige `cell-tall` und `cell-med` CSS hinzugefügt, die wir oben definiert haben. Dies hat den Effekt, dass die Höhe einiger Zellen geändert wird. Ich frage mich, wie es aussehen wird ...

Hier sind sie wieder bei mittleren und kleinen Bildschirmgrößen:

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



col-xs-6 col-md-3



Oh mein Gott, was für ein Schlamassel. Ich glaube nicht, dass wir das wollten. Bei einer mittelgroßen Größe sind 5 und 6 weit weg, und irgendwie endeten 7 mit einer neuen Reihe. Bei einer kleinen Größe haben wir zwei Zellen in der ersten Reihe und **vier** in der zweiten Reihe, wobei 4, 5 und 6 bei beiden Bildschirmgrößen rechts gestapelt sind!

Wie lösen wir das?

## Clearfix zur Rettung

Eine Möglichkeit, die Situation sicher zu verbessern, besteht darin, mehr als eine `row` :

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

Dies ist normalerweise das erste, was neue Bootstrapper versuchen. Es scheint sinnvoll zu sein: "Ich möchte vier Zellen in jeder Zeile, also erstelle ich für jede 4- `col` Divs eine neue `row` ".

Es gibt jedoch ein Problem mit dieser Argumentation: Der Kernpunkt von Bootstrap 3 und der kommenden Version 4 besteht darin, darauf zu **reagieren** . Wenn Sie "vier `col row` " platzieren, denken Sie nicht wirklich "reagieren".

Ein gutes Verständnis der `clearfix` CSS - Klasse werden Sie feststellen , dass mehrere sehen helfen beginnen `row` divs haben wirklich Ihr Verständnis für die Art und Weise wurde Eintrübung dass Responsive Design arbeiten **sollte**. Kurz gesagt, Sie können einfach **nicht** wissen, wie viele

col sowieso row einfügen müssen - der Browser hat Ihre Arbeit noch nicht gerendert!

Erinnern Sie sich an die ersten Dinge Zuerst sagten wir, Sie müssten in "Umkehrung von 12" denken? Lassen Sie uns dieses Problem ohne weiteres beheben, indem Sie die richtigen Kommentare im Code verwenden, um die Verwirrung hoffentlich zu beseitigen. Ja, es sieht nach viel mehr Code aus, aber das **meiste** davon sind Kommentare.

```
<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>
```

Der `clearfix` ist eine CSS-Klasse, die ein winziges (praktisch unsichtbares) `div` darstellt. Der Zweck besteht darin, die von den `col` `divs` verwendeten `left` Floats "zu löschen".

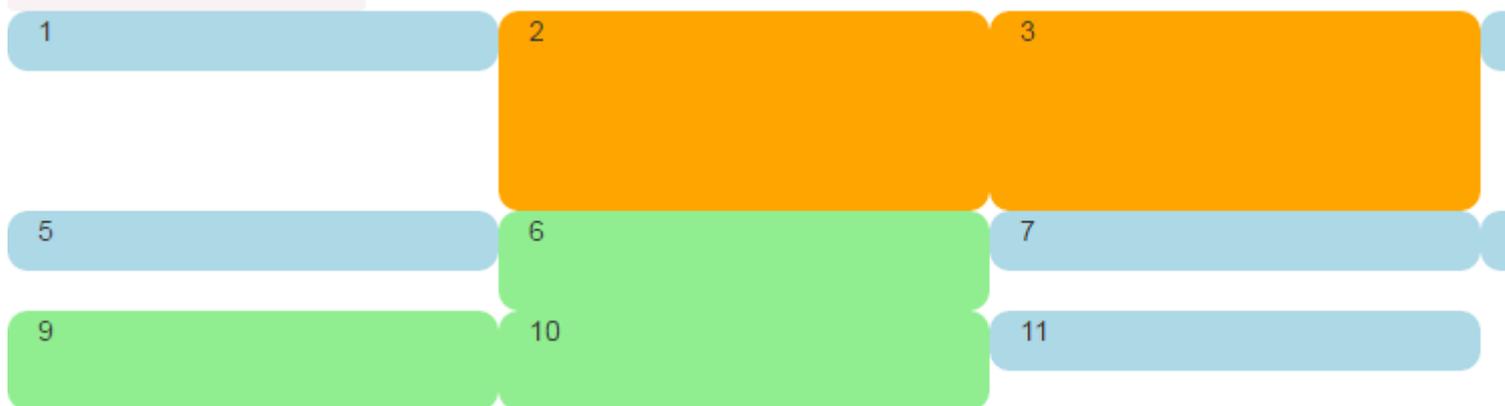
Das Genie ist wirklich in den Klassen `hidden-sm`, `hidden-md` usw. Diese Klassen werden **auf das `clearfix` `div` gestellt**, NICHT auf die `col` `divs`! Dies bewirkt, dass das `clearfix` `div` bei bestimmten Viewport-Breiten magisch erscheint oder aus dem Render-Stream verschwindet! Genius!

Bootstrap verfügt in Version 3 über ein verwirrendes Array von `hidden-*` und `hidden-* visible-*` Klassen. Leider sind sie nicht wirklich "invers". Daher finde ich es am klarsten und sichersten, nur die `hidden-*` -Klassen für `clearfixes` zu verwenden.

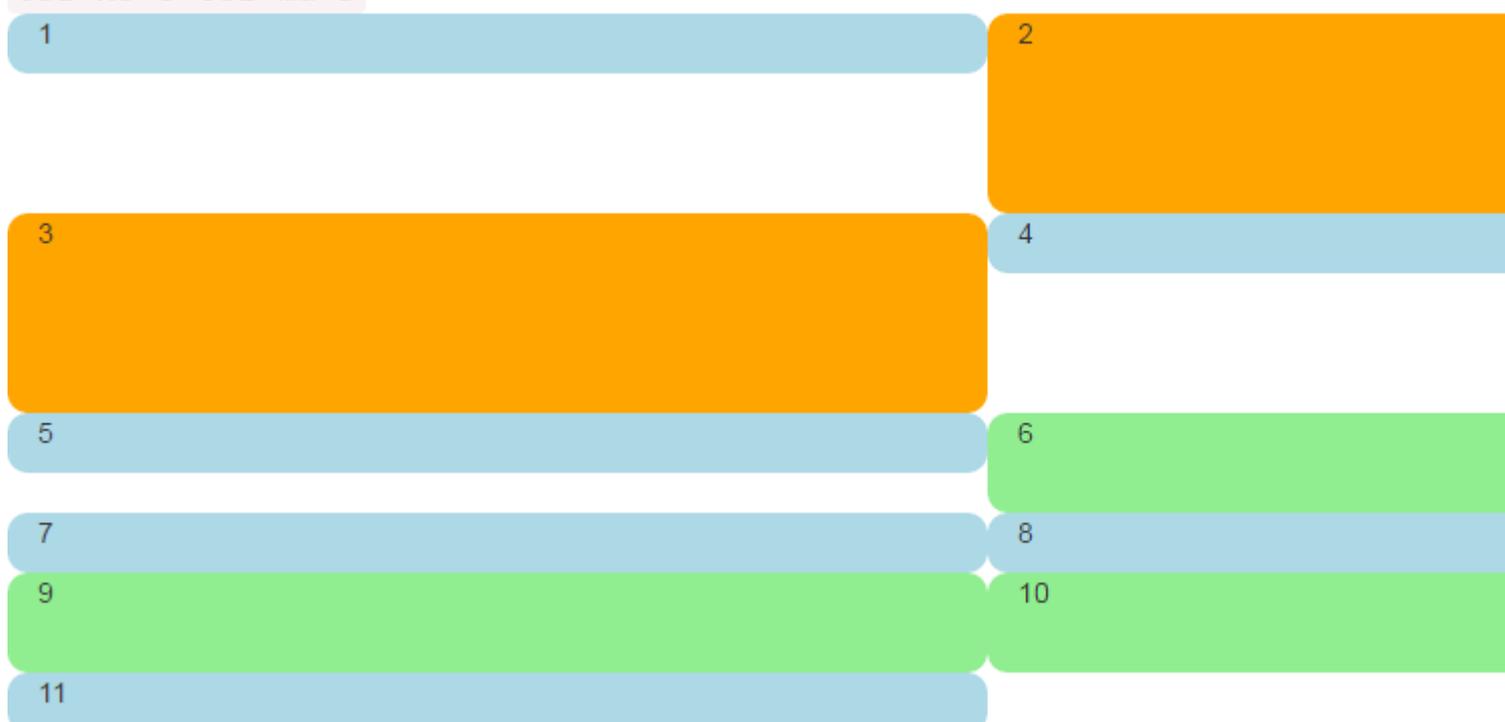
Es sieht so aus, als könnte sich das in Bootstrap 4 zum Besseren ändern, mit Klassen wie `hidden-*-up` und `hidden-*-down` (sie werden vollständig von `visible-*` -Klassen entfernt).

Nun genug wörtlich, wie sieht es jetzt aus?

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



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



Das wollen wir! Auf dem großen Bildschirm haben wir immer VIER, auf dem kleineren Bildschirm immer ZWEI. Kein Stapeln mehr an seltsamen Orten und Lücken sind, wo wir sie erwarten würden.

## Ein Dashboard

Nun gut genug von diesen farbigen abgerundeten Dingen, lasst uns etwas interessanteres als Zahlen in diese Divs einfügen. Nehmen wir den gleichen Satz von Spalten und erstellen Sie ein echtes Dashboard. Verwenden Sie das folgende 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>
```

Und hier ist der "Dashboard" -Code:

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

```

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

```

```

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

```

```

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

```

Dieser Code wird folgendermaßen aussehen:

Heading 1
Body 1
Footer 1

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

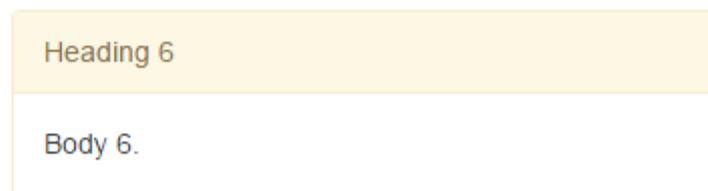
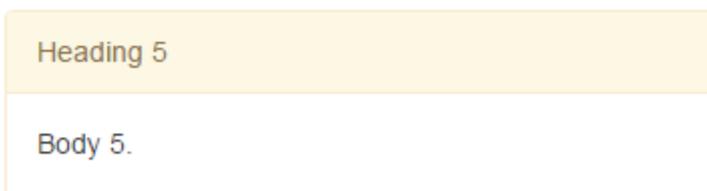
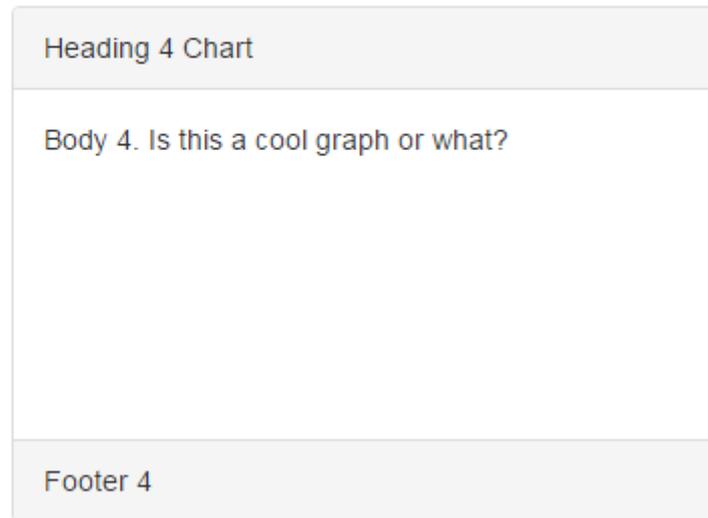
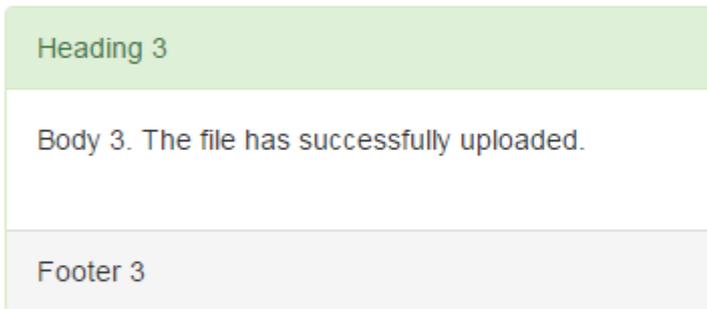
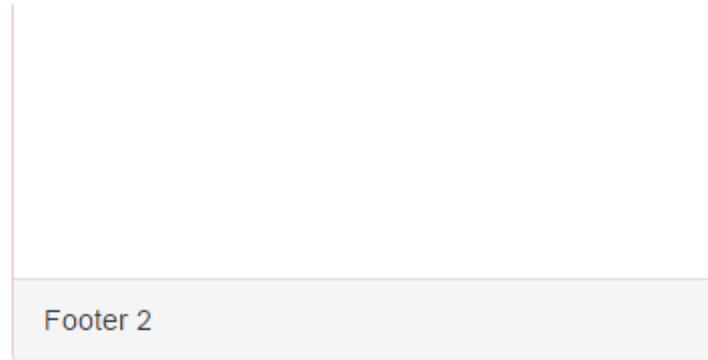
Heading 3
Body 3. The file has successfully uploaded.
Footer 3

Heading 5
Body 5.
Footer 5

Heading 6
Body 6.

Heading 7
Body 7.
Footer 7

Und so in kleineren Viewports:



Übrigens verwende ich die Bootstrap 3- `panel` Klasse, die in Bootstrap 4 wegfällt und durch die viel aussagekräftigere und spezifischere `card`. Wenn Sie sich diese Bilder ansehen, können Sie sehen, warum die `card` ein viel besserer Name ist als das mehrdeutige `panel`.

## 2,4,6 Layout mit Clearfixes

Hier ist ein Layout, das je nach Bildschirmgröße zwei, vier oder sechs Zellen darstellt.

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

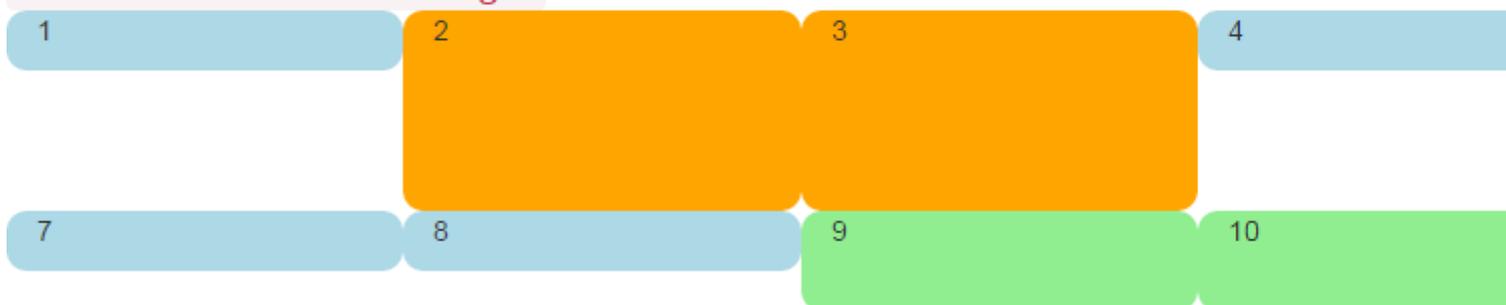
```

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

```

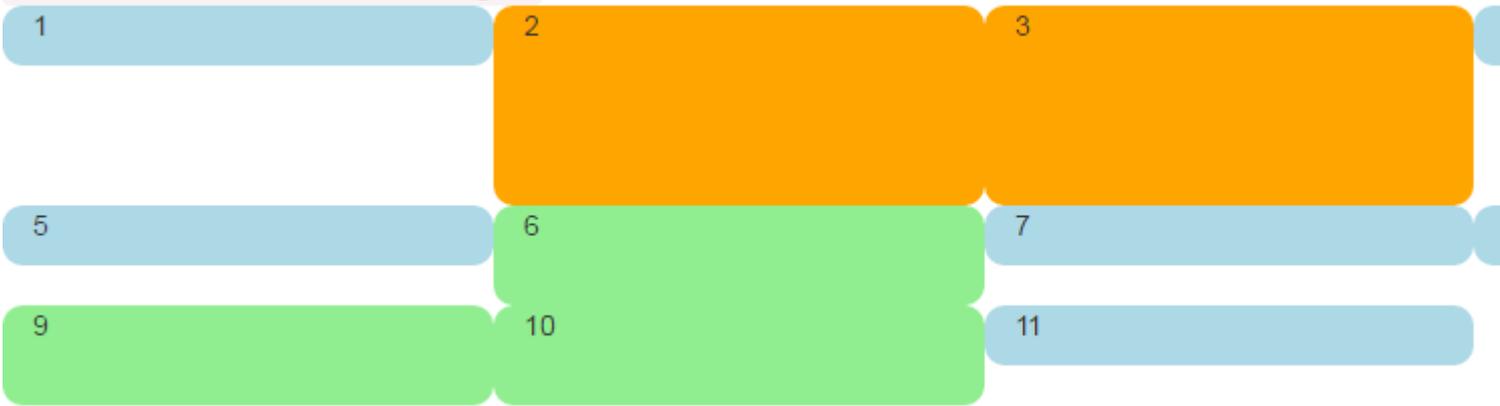
Großer Bildschirm:

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



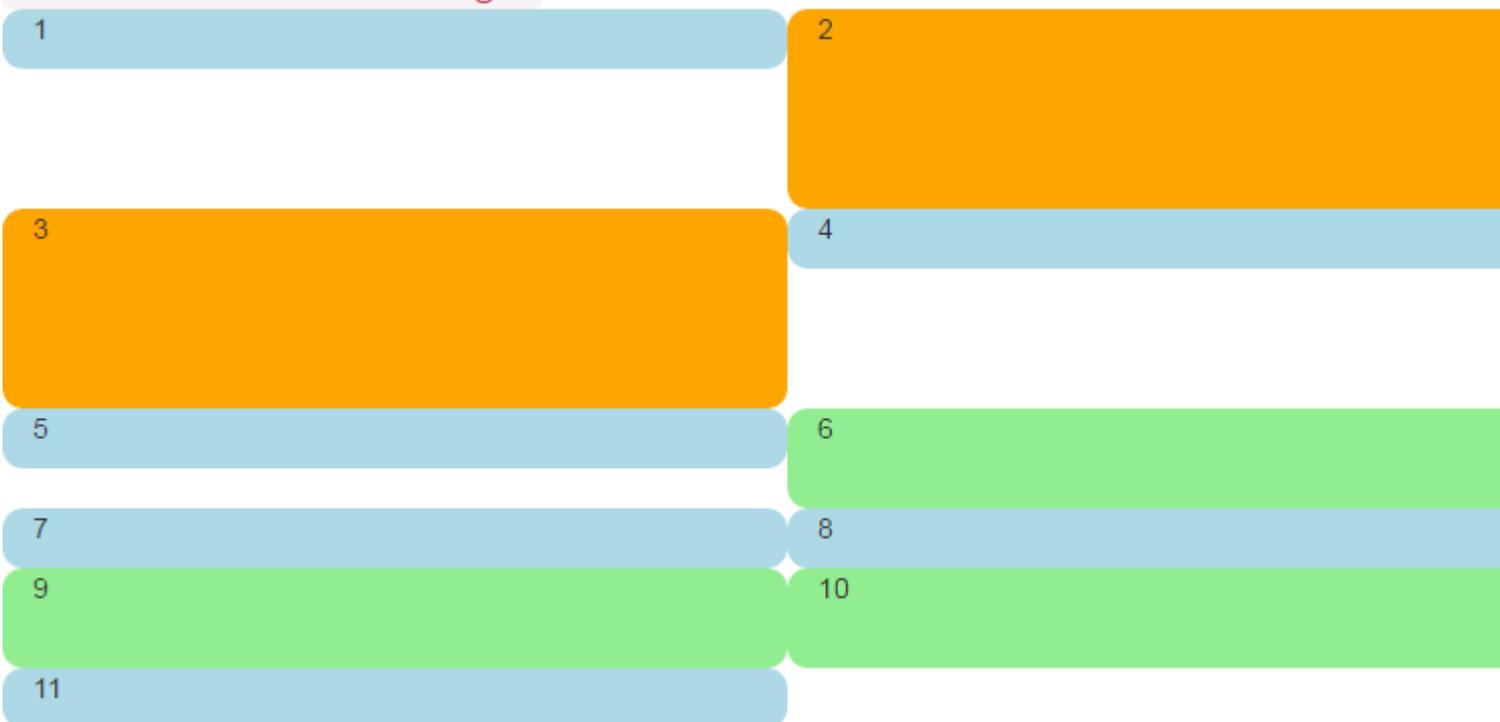
Mittlerer Bildschirm:

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



Kleiner Bildschirm:

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



## Warum sollten Bootstrap-Spalten mehr als 12 hintereinander überschreiten?

Es gibt viele responsive Szenarien, in denen es *erforderlich ist*, Spalteneinheiten zu haben, die 12 in einem einzelnen `.row` Element `.row`. Dies wird als **Spaltenumbruch bezeichnet**.

Wenn sich mehr als 12 Spalten in einer einzelnen Zeile befinden, wird jede Gruppe zusätzlicher Spalten als eine Einheit in eine neue Zeile umgebrochen.

Stellen Sie sich beispielsweise ein Layout vor, in dem wir wollen ...

- 3 Spalten über große und mittlere Geräte und
- 2 Spalten über kleine und kleinste Geräte

## Large



## Small



Um dieses Layout in Bootstrap zu erhalten, verwenden wir (richtig).

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

### Richtige Methodendemo

Wie Sie im Beispiel sehen, **überschreitet** die *Summe* der Spalteneinheiten im `.row` Element **12**. Diese Technik wird als **Spaltenumbruch bezeichnet** und ist eine der leistungsfähigsten responsiven Designfunktionen von Bootstrap. Das gewünschte Layout wäre *nicht möglich* (mit Ausnahme des Duplizierens von Markup), wenn wir versuchen, bei dem weit **verbreiteten Missverständnis festzuhalten, dass Spalteneinheiten bis zu 12 in einer einzigen Zeile**

## addieren müssen .

Das Layout ist *nicht* möglich, wenn wir nicht 12 (falsch) überschreiten.

```
<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>
```

### Falsche Methodendemo (schlägt auf großen drei Spalten fehl)

Denken Sie daran, dass eine `.row` *nicht* mit einer einzelnen Zeile im Ansichtsfenster `.row` ist. Eine `.row` ist eine Gruppierung von Spalten. Die Spalten mit mehr als 12 Einheiten in einem einzelnen `.row` Element werden **in eine neue Zeile** (unten im Ansichtsfenster) `.row` . Daher ist es wichtig zu verstehen, dass die 12 Spalten horizontale **Einheiten** im Ansichtsfenster darstellen.

Außerdem müssen **responsive Reset** (clearfix) für gleichmäßiges Wrapping verwendet werden, wenn die Höhe der **Spalten variiert** .

Verwenden von Clearfix in Zeilen und Spalten online lesen: <https://riptutorial.com/de/twitter-bootstrap/topic/6124/verwenden-von-clearfix-in-zeilen-und-spalten>

---

# Kapitel 36: Warnen

## Bemerkungen

Weitere [Informationen](http://getbootstrap.com/components/#alerts) finden Sie unter <http://getbootstrap.com/components/#alerts>

## Examples

### Warnungsarten

Im Gegensatz zu einigen anderen Bootstrap-Komponenten wie z. B. [Buttons](#) verfügen die [Alerts](#) **nicht** über ein `default` oder `primary` Styling, da sie den Benutzer auf bestimmte Weise warnen sollen.

```
<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>
```

### Warnungsgrundbeispiel

```
<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

## Animierte Benachrichtigungen

Die Klassen `.fade` und `.in` fügen beim Schließen der Warnmeldung einen Fading-Effekt hinzu.

```
<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>
```

## Ablehnbare Alarme

Um eine Alarm-Close-Funktionalität zu erhalten, müssen Sie dem Schließen-Button nur `data-dismiss="alert"` hinzufügen.

```
<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` und `.close` Klassen sind optional und nur für das Styling nützlich.

## Linkfarbe in Alerts

Um schnell eine passende Farbe für Links in einer Warnung bereitzustellen, können Sie die

.alert-link Dienstprogrammklasse verwenden.

```
<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 ...

Warnen online lesen: <https://riptutorial.com/de/twitter-bootstrap/topic/6434/warnen>

# Credits

S. No	Kapitel	Contributors
1	Erste Schritte mit twitter-bootstrap	<a href="#">andreaem</a> , <a href="#">Atul Mishra</a> , <a href="#">bpoiss</a> , <a href="#">Community</a> , <a href="#">Evan</a> , <a href="#">Gleb Kemarsky</a> , <a href="#">H. Pauwelyn</a> , <a href="#">kernal lora</a> , <a href="#">Kritner</a> , <a href="#">MattD</a> , <a href="#">Mingle Li</a> , <a href="#">Nhan</a> , <a href="#">Prashanth Benny</a> , <a href="#">the12</a> , <a href="#">tmg</a> , <a href="#">Toby</a> , <a href="#">VincenzoC</a> , <a href="#">Vishnu Y S</a>
2	Anpassung des Twitter-Bootstrap-Stils	<a href="#">CENT1PEDE</a> , <a href="#">Vikas Yadav</a>
3	Bootstrap Badges und Labels	<a href="#">mmativ</a>
4	Bootstrap Dropdowns	<a href="#">Ismail Farooq</a> , <a href="#">MattD</a>
5	Bootstrap Navbar	<a href="#">Ilyas karim</a>
6	Bootstrap-Befestigung	<a href="#">Ilyas karim</a>
7	Bootstrap-Container	<a href="#">Neha Chopra</a>
8	Bootstrap-Komponenten	<a href="#">alex</a>
9	Bootstrap-Themen	<a href="#">KAI</a>
10	Bootstrap-Überprüfung	<a href="#">Amy Barrett</a> , <a href="#">mnoronha</a>
11	Dropdowns	<a href="#">alex</a>
12	Drucken in Bootstrap.	<a href="#">MattD</a>
13	Formen	<a href="#">Community</a> , <a href="#">Jens</a> , <a href="#">Owen Pauling</a>
14	Gebrauchsklassen	<a href="#">ajju</a>
15	Gitternetz	<a href="#">neophyte</a> , <a href="#">ZimSystem</a>
16	Glyphicons	<a href="#">Madalina Taina</a> , <a href="#">tmg</a> , <a href="#">Umer Farooq</a>

17	Gruppe auflisten	<a href="#">Ilyas karim</a>
18	Jumbotron	<a href="#">Gabriel Chi Hong Lee</a>
19	Karussells	<a href="#">alex</a> , <a href="#">Boysenb3rry</a>
20	Migration zu Bootstrap 4	<a href="#">Chris Farmer</a> , <a href="#">neophyte</a> , <a href="#">ZimSystem</a>
21	Modale Dialoge	<a href="#">alex</a> , <a href="#">mnoronha</a>
22	Modals	<a href="#">John Blanchard</a>
23	Navbar	<a href="#">Kritner</a> , <a href="#">Krunal Mevada</a> , <a href="#">kybernaut.cz</a>
24	Navigationsmenüs	<a href="#">Ignacio Correia</a>
25	Navs	<a href="#">leowebguy</a>
26	Panels	<a href="#">JackPoint</a> , <a href="#">tmg</a>
27	QuickInfo	<a href="#">Madalina Taina</a> , <a href="#">tmg</a>
28	Rastersystem	<a href="#">Ani Menon</a> , <a href="#">Boysenb3rry</a> , <a href="#">bpoiss</a> , <a href="#">Harshal Patil</a> , <a href="#">leowebguy</a> , <a href="#">Madalina Taina</a> , <a href="#">Mingle Li</a> , <a href="#">mmativ</a> , <a href="#">Stephen Leppik</a> , <a href="#">the12</a> , <a href="#">tmg</a> , <a href="#">ZimSystem</a>
29	Säulen	<a href="#">kybernaut.cz</a>
30	Seitennummerierung	<a href="#">TheDarkKnight</a>
31	Tabellen	<a href="#">atjoedonahue</a>
32	Tabs	<a href="#">DavidG</a> , <a href="#">tmg</a>
33	Tasten	<a href="#">Madalina Taina</a> , <a href="#">Muhammad Abdullah</a> , <a href="#">Richard Hamilton</a> , <a href="#">the12</a>
34	Verwenden von Clearfix in Zeilen und Spalten	<a href="#">Bruce Pierson</a> , <a href="#">ZimSystem</a>
35	Warnen	<a href="#">JHS</a> , <a href="#">Madalina Taina</a> , <a href="#">tmg</a>