



Бесплатная электронная книга

УЧУСЬ

twitter-bootstrap

Free unaffiliated eBook created from
Stack Overflow contributors.

#twitter-
bootstrap

.....	1
1: twitter-bootstrap	2
.....	2
.....	2
Examples.....	3
/.....	3
.....	5
Bootstrap.....	6
-.....	7
2: Bootstrap Navbar	9
Examples.....	9
Bootstrap Navbar.....	9
Bootstrap.....	9
3: Dropdowns	11
.....	11
Examples.....	11
HTML.....	11
4: Glyphicons	12
.....	12
Examples.....	12
.....	12
5: JumboTron	14
.....	14
.....	14
Examples.....	14
jumbotron.....	14
6: Navbar	16
Examples.....	16
Basic Navbar ().....	16
navbar.....	16
Navbar.....	17

"active".....	17
Navbar ().....	17
.....	18
7: Navs	19
Examples.....	19
Navs.....	19
8:	21
.....	21
Examples.....	21
.....	21
.....	21
.....	22
.....	22
.....	22
9: -	24
Examples.....	24
Navbar.....	24
2.....	24
10:	26
Examples.....	26
HTML.....	26
.....	26
11: Bootstrap	28
.....	28
.....	28
Examples.....	28
.....	28
.....	29
12:	30
Examples.....	30
.....	30
.....	30

13: Clearfix Cols	31
.....	31
.....	31
Examples	31
.....	31
.....	33
Clearfix	34
.....	37
2,4,6 Clearfixes	42
Bootstrap 12 ?	44
14:	47
.....	47
Examples	47
HTML	47
Javascript	48
15:	49
.....	49
Examples	49
.....	49
16:	52
Examples	52
(CSS SASS)	52
17: Bootstrap	56
.....	56
Examples	56
Bootstrap	56
18: Bootstrap	57
.....	57
Examples	57
.....	57
19:	58

Examples.....	58
.....	58
.....	58
.....	58
20: Bootstrap 4.....	59
.....	59
.....	59
Examples.....	59
Bootstrap 4.....	59
- Bootstrap 4.....	60
.....	60
Affix.....	60
Bootstrap 4 Navbar.....	63
Bootstrap 3 Bootstrap 4 CSS.....	66
4.....	68
4.....	70
.....	70
.....	71
4.....	72
21:	73
.....	73
Examples.....	73
HTML-.....	73
22:	74
.....	74
Examples.....	74
HTML.....	74
Javascript.....	74
23: Bootstrap.....	76
.....	76
Examples.....	76
CSS.....	76

24:	78
	78
Examples	78
	78
25:	79
	79
Examples	79
	79
	79
	79
26: Bootstrap	81
Examples	81
HTML	81
27:	82
	82
Examples	82
	82
	82
28:	84
Examples	84
.hidden- * - SCSS	84
29:	85
	85
Examples	85
ASP.NET MVC	85
,	86
	87
30:	88
	88
	88
Examples	88
	

()	89
Bootstrap	90
	92
	92
push pull	92
31:	94
	94
	94
Examples	94
	94
32:	96
	96
Examples	96
	96
	96
	96
	96
	96
	97
	97
	97
33:	98
Examples	98
	98
34:	99
	99
Examples	99
	99
	99
	99
	100
	100

.....	100
.....	100
.....	101
Reflow -	101
35:	103
Examples.....	103
Bootstrap	103
36:	104
Examples.....	104
.....	104
.....	104
.....	105

Около

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

глава 1: Начало работы с twitter-bootstrap

замечания

Bootstrap - это HTML, CSS и JavaScript-фреймворк, используемый для создания веб-сайтов, созданных с использованием парадигмы мобильных устройств, а также адаптивного веб-дизайна (RWD). Он использует комбинацию готовых CSS-классов и JavaScript для создания множества вещей в Интернете. Он включает в себя такие вещи, как настраиваемая сетчатая сетка, позволяющая просматривать веб-сайты на любом экране, раскрывающийся навигационный бар, способный реагировать и даже простые вещи, которые могут быть интенсивными во времени, такие как кнопки, формы и аккордеоны, чтобы назвать мало.

Bootstrap может быть полезен по следующим причинам:

1. **Экономия времени** . В Bootstrap есть много вещей, которые предварительно построены, и их просто нужно называть при написании кода. Это экономит значительное количество времени и может значительно сократить время, необходимое для кодирования веб-сайта.
2. **Построенный с учетом** адаптивного **веб-дизайна** : Bootstrap позволяет веб-разработчикам не беспокоиться о создании вещей, которые будут масштабироваться с размером их экрана, так как Bootstrap использует мобильный, первый отзывчивый дизайн, который позволяет им создавать вещи, которые будут работать на любом экране размер.
3. **Упрощает процесс проектирования** : Bootstrap поставляется заранее с элементами, которые имеют хорошие методы проектирования. Это может быть полезно для тех, чьи навыки веб-дизайна не так велики или для тех, кто рассматривает дизайн как утомительную задачу, так как многие из классов Bootstrap эстетически приятны и замечательны.

В этом разделе должны быть указаны любые большие темы в twitter-bootstrap и ссылки на связанные темы. Поскольку Документация для twitter-bootstrap является новой, вам может потребоваться создать начальные версии этих связанных тем.

Версии

Версия	Дата выхода
4,0	2999-01-01

Версия	Дата выхода
3.3.7	2016-07-25
3.3.6	2015-11-24
3.3.5	2015-06-15
3.3.4	2015-03-16
3.3.1	2014-11-12
3.3.0	2014-10-29
3.2.0	2014-06-26
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

Установка / настройка

Загрузка:

- Загрузите Bootstrap [прямо](#) или клонировать и т. Д. Из репозитория [GitHub](#)
- Загрузите настроенную версию Bootstrap из официальных [документов](#)
- Устанавливать с помощью колокольчика: `bower install bootstrap`
- Установить с npm: `npm install bootstrap`
- Установить с композитором: `composer require twbs/bootstrap`

Структура файла

```

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

```

Установка:

В HTML-странице можно включить CSS, JS Bootstrap и зависимость jQuery (предварительная версия 3, по крайней мере, с последней версией Bootstrap). Обратите внимание, что если вы планируете использовать функции JavaScript Bootstrap, ваша ссылка jQuery должна появиться *перед* ссылкой bootstrap.js в вашем HTML.

Вы можете использовать установленные файлы Bootstrap из приведенного выше раздела или сослаться на CDN, предоставленные создателями Bootstrap (ссылки, взятые из «[Начало работы с Bootstrap](#)»):

```

<!-- Latest compiled and minified CSS -->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
integrity="sha384-BVYiisIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u"
crossorigin="anonymous">

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

<!-- Latest compiled and minified JavaScript -->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"
integrity="sha384-Tc5IQib027qvyjSMfHjOMaLkfuWVxZxUPnCJA712mCWNIpG9mGCD8wGNICPD7Txa"
crossorigin="anonymous"></script>

```

Очень простая веб-страница Bootstrap:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">

```

```

    <!-- The above 3 meta tags *must* come first in the head; any other head content must come
*after* these tags -->
    <title>Bootstrap 101 Template</title>

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

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

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

```

Основной шаблон

```

<!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>
```

Когда использовать Bootstrap

Bootstrap - это самоуверенная структура для HTML, CSS и Javascript. Он содержит базовый стиль и функциональность для того, что стало принятым элементами [User Interface], например элементами формы, кнопками, модальными окнами и элементами навигации.

Bootstrap - это адаптивная веб-инфраструктура, то есть она предназначена для адаптации макета и дизайна для размеров экрана больших и малых размеров, таких как мобильные устройства, планшеты и настольные компьютеры, все в одной базе кода.

Одной из основных концепций Bootstrap является структура сетки. Применяя классы к элементам HTML, можно создавать сложные макеты, используя базовую сетку из двенадцати столбцов. Например, расположение четырех столбцов может адаптироваться к двум столбцам на планшетных устройствах и одному столбцу на мобильных устройствах. Для достижения этого сетка использует `media queries`, метод CSS для таргетинга на конкретные размеры экрана.

Bootstrap работает особенно хорошо, если:

- Пользовательский дизайн не является главным приоритетом
- Вам удобнее редактировать HTML и добавлять классы, чем создавать пользовательский CSS
- Вам удобнее использовать фреймворк, который будет иметь много визуальных сходств со многими другими веб-сайтами

Bootstrap может использоваться теми, кто не знаком с HTML, CSS и Javascript, поскольку [документация](#) отличная. Тем не менее, есть кривая обучения для тех, кто не совсем устраивает три основные технологии, используемые Bootstrap (HTML, CSS и Javascript).

Можно приобрести или загрузить темы Bootstrap, чтобы изменить стиль или функциональность Bootstrap. Также возможно использовать Bootstrap в качестве отправной точки, с настройкой CSS и Javascript.

Основная веб-страница с использованием компонентов начальной загрузки

```
<!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>
          </ul>
        </div>
      </div>
    </nav>
```

```

        <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>

```

Прочитайте Начало работы с twitter-bootstrap онлайн: <https://riptutorial.com/ru/twitter-bootstrap/topic/818/начало-работы-с-twitter-bootstrap>

глава 2: Bootstrap Navbar

Examples

Bootstrap Navbar

Это пример Bootstrap версии 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>
```

Изображение бренда Bootstrap

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

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

Прочитайте Bootstrap Navbar онлайн: <https://riptutorial.com/ru/twitter-bootstrap/topic/6233/bootstrap-navbar>

глава 3: Dropdowns

замечания

Для получения дополнительной информации посетите официальную документацию Bootstrap, расположенную по адресу <http://getbootstrap.com/javascript/#dropdowns> , откуда получен базовый пример использования HTML.

Examples

Основное использование HTML

Выпадающее окно Bootstrap - это компонент Bootstrap, который позволяет элементу HTML запускать отображение выпадающего списка подменю при щелчке элемента.

Вот пример использования 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-menu` **МОЖНО** указать, вставив элемент `li` в элемент `ul` с `.dropdown-menu` .

Прочитайте Dropdowns онлайн: <https://riptutorial.com/ru/twitter-bootstrap/topic/6033/dropdowns>

глава 4: Glyphicons

замечания

В этом разделе приведен обзор глификонов Bootstrap и описано, как использовать глификоны.

Examples

Как использовать глификон

Twitter Bootstrap поддерживает значки, называемые глификонами, и их можно использовать со всеми тегами HTML.

Все значки требуют базового класса и индивидуального класса значков.

Имейте в виду, что классы значков нельзя напрямую комбинировать с другими компонентами, поэтому всегда используйте `` .

Если ваш HTML-код содержит внутренние дочерние элементы, вы не можете использовать классы значков для этого конкретного тега.

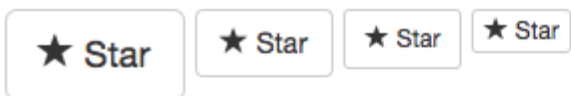
Примеры

Например, вы создаете кнопку начальной загрузки, тогда синтаксис этой кнопки должен выглядеть следующим образом:

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

Таким образом, в приведенном выше примере создается простая кнопка начальной загрузки, но теперь вы хотите добавить глификон в эту кнопку, так как это просто добавляет элемент `` внутри `<button>` . Как это:

```
<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>
```

Прочитайте Glyphicons онлайн: <https://riptutorial.com/ru/twitter-bootstrap/topic/6098/glyphicons>

глава 5: JumboTron

Вступление

Jumbotron является стандартным компонентом Bootstrap для отображения некоторого важного содержимого на вашем веб-сайте. Он обычно используется прямо под навигационной панелью, перед контентом.

замечания

Внутри jumbotron также работает вся сетка, класс контейнера и класс строк.

Examples

Основной jumbotron с двумя строками текста и кнопкой

Это jumbotron с заголовком, содержанием и кнопкой.

Код

```
<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>
```

Результат

Title text

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

A button

Прочитайте JumboTron онлайн: <https://riptutorial.com/ru/twitter-bootstrap/topic/9188/jumbotron>

глава 6: Navbar

Examples

Basic Navbar (исправлено в верхней части страницы)

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

Подменю в navbar

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



```

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

```

Разделитель Navbar

```

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

```

Сохранять текущую ссылку навигации "active"

```

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

```

Изменить точку останова Navbar (мобильный или обычный)

max-width - точка останова

```

@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;
}
}
}

```

Закреть рухнувший навигатор при нажатии за пределами панели навигации

```

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');
        }
    }
});

```

Прочитайте Navbar онлайн: <https://riptutorial.com/ru/twitter-bootstrap/topic/2267/navbar>

глава 7: Navs

Examples

Загрузочные Navs

Navs, доступные в Bootstrap, имеют общую разметку, начиная с базового класса .nav, а также совместно используемых состояний. Модификаторы Swap переключаются между каждым стилем.

Вкладки

```
<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>
```

Таблетки

```
<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>
```

Оправдано

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

С раскрывающимися окнами

```
<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>
```

Прочитайте Navs онлайн: <https://riptutorial.com/ru/twitter-bootstrap/topic/6505/navs>

глава 8: бдительный

замечания

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

Examples

Типы предупреждений

В отличие от некоторых других компонентов Bootstrap, таких как [кнопки](#) , [оповещения](#) не имеют `default` или `primary` стиля, потому что они предназначены для оповещения пользователя определенным образом.

```
<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>
```

Основной пример оповещения

```
<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

Анимированные оповещения

`.fade` и `.in` добавляют эффект замирания при закрытии сообщения предупреждения.

```
<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>
```

Недопустимые оповещения

Чтобы дать оповещение о тесной функциональности, нам нужно только добавить `data-dismiss="alert"` к нашей кнопке закрытия.

```
<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` и `.close` классы являются необязательными, полезными только для стилизации.

Цвет ссылки в оповещениях

Чтобы быстро предоставить соответствующий цвет для ссылок внутри любого предупреждения, мы можем использовать класс утилиты `.alert-link`.

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

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

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

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

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

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

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

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

Прочитайте бдительный онлайн: <https://riptutorial.com/ru/twitter-bootstrap/topic/6434/>
бдительный

глава 9: Бутстреп-аффикс

Examples

В 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>
```

Пример Аффикса 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>
```

Прочитайте Бутстреп-аффикс онлайн: <https://riptutorial.com/ru/twitter-bootstrap/topic/6639/>
бутстреп-аффикс

глава 10: Вкладки

Examples

Основной 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>
```

Это создаст вкладку, состоящую из 3 вкладок и 3 связанных разделов содержимого.

Анимированные вкладки

Чтобы сделать вкладки затухающими, добавьте `.fade` в каждую `.tab-pane`. В активной панели вкладки также должен быть класс `.in` чтобы сделать видимым начальное содержимое.

```
<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>
```

Прочитайте Вкладки онлайн: <https://riptutorial.com/ru/twitter-bootstrap/topic/5980/вкладки>

глава 11: Выпадающие окна Bootstrap

параметры

методы	пример
Вызов через Javascript	<code>\$('.dropdown-toggle').dropdown();</code>
Переключает выпадающий список	<code>\$('.dropdown-toggle').dropdown('toggle')</code>
Тип события	Описание
show.bs.dropdown	Это событие срабатывает сразу же после вызова метода show instance.
shown.bs.dropdown	Это событие запускается, когда выпадающее окно становится видимым для пользователя (будет ждать перехода CSS для завершения).
hide.bs.dropdown	Это событие запускается сразу же после вызова метода экземпляра hide.
hidden.bs.dropdown	Это событие запускается, когда выпадающий список скрывается от пользователя (будет ждать перехода CSS для завершения).
Пример обработчика событий	<pre>\$(element).on('show.bs.dropdown', function () { // do something... })</pre>

замечания

При вызове Dropdown через Javascript `$('.dropdown-toggle').dropdown()` Dropdown `$('.dropdown-toggle').dropdown()` Dropdown `$('.dropdown-toggle').dropdown()` все равно требуется data-api, т.е. `data-toggle="dropdown"` . [Прочитайте больше](#)

Examples

Как пользоваться

Используйте класс `.dropdown` в родительском элементе выпадающего меню.

Добавьте класс `.dropdown-menu` в элемент для инициализации плагина выпадающего меню.

Вызовите плагин, используя класс `.dropdown-toggle` и атрибут `data-toggle="dropdown"` на кнопке или гиперссылке.

Основной пример

```
<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>
```

Прочитайте Выпадающие окна Bootstrap онлайн: <https://riptutorial.com/ru/twitter-bootstrap/topic/6361/выпадающие-окна-bootstrap>

глава 12: Значки загрузки и ярлыки загрузки

Examples

Значки

Значки - это числовые показатели того, сколько элементов связано со ссылкой:

Используйте класс `.badge` в `` для создания значков:

```
<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>
```

Значок в кнопке

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

Этикетки

Этикетки используются для предоставления дополнительной информации о чем-то:

Используйте класс `.label`, за которым следует один из шести контекстных классов `.label-default`, `.label-primary`, `.label-success`, `.label-info`, `.label-warning` или `.label-danger`, внутри элемента `` для создания метки:

```
<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>
```

Прочитайте Значки загрузки и ярлыки загрузки онлайн: <https://riptutorial.com/ru/twitter-bootstrap/topic/7867/значки-загрузки-и-ярлыки-загрузки>

глава 13: Использование Clearfix в рядах и Cols

Вступление

При создании расширенных макетов могут возникать сценарии, когда вам понадобится использовать **более 12 столбцов** в одном элементе `.row`. Концепция **обтекания столбцов** и **отзывчивых сбросов (АКА clearfixes)** *необходимы* для понимания **адаптивного** дизайна с помощью Bootstrap.

[Основы сетки Bootstrap](#)

замечания

Решетки Bootstraps замечательно мощные и элегантные. Однако вы должны помнить, что имя фреймворка - «Bootstrap», а не «WeDidItForYou». Bootstrap **обеспечивает** гибкий дизайн, он не **гарантирует** его.

Вам все равно, чтобы ваш дизайн был действительно отзывчивым и предоставлял вашим пользователям наилучшие возможности для конечных пользователей.

Examples

Наивная первая попытка

Прежде чем мы начнем, давайте определим CSS для примеров. Это `head` часть нашего образца. Я всегда использую `border-radius` и `background-color` при тестировании, потому что он упрощает просмотр делений ячейки без добавления размера границы, который может повлиять на размер ячеек.

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

```

    }
    /* 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>

```

С этой точки зрения давайте определим сетку и посмотрим на идеальные результаты при всех размерах видового экрана!

Использование `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`



Предыдущие два изображения показывают рендеринг на средних и малых размерах экрана. Помните, что мы получим четыре столбца на среде + из-за `col-md-3` и двух ячеек на маленьком - из-за `col-xs-6`.

Выглядит неплохо, правда? Я думаю, что мы закончили здесь! Сказал много наивных сайтов Bootstrap, которые просто ждут выхода ...

Проблема с высотой

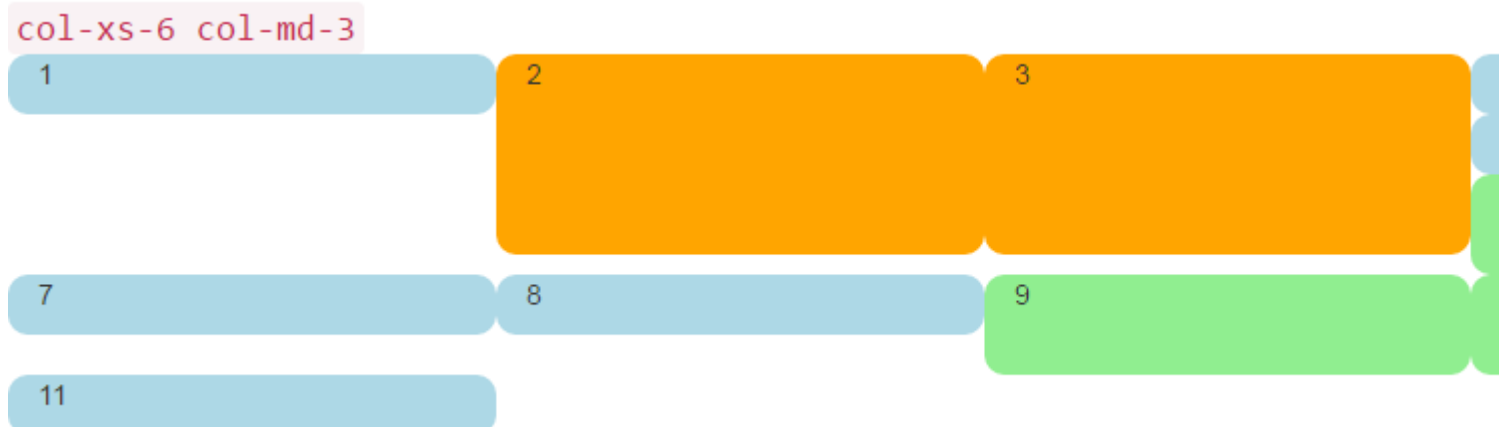
В нашем « [наивном примере](#) » все наши клетки были одинаковой высоты. Браузер охотно сломал линии именно там, где мы хотели, и все казалось правильным с миром. Пока высота не появится на картинке.

Давайте рассмотрим предыдущий пример и придадим некоторую высоту некоторым ячейкам, возможно, вы увидите на странице типа dashboard.

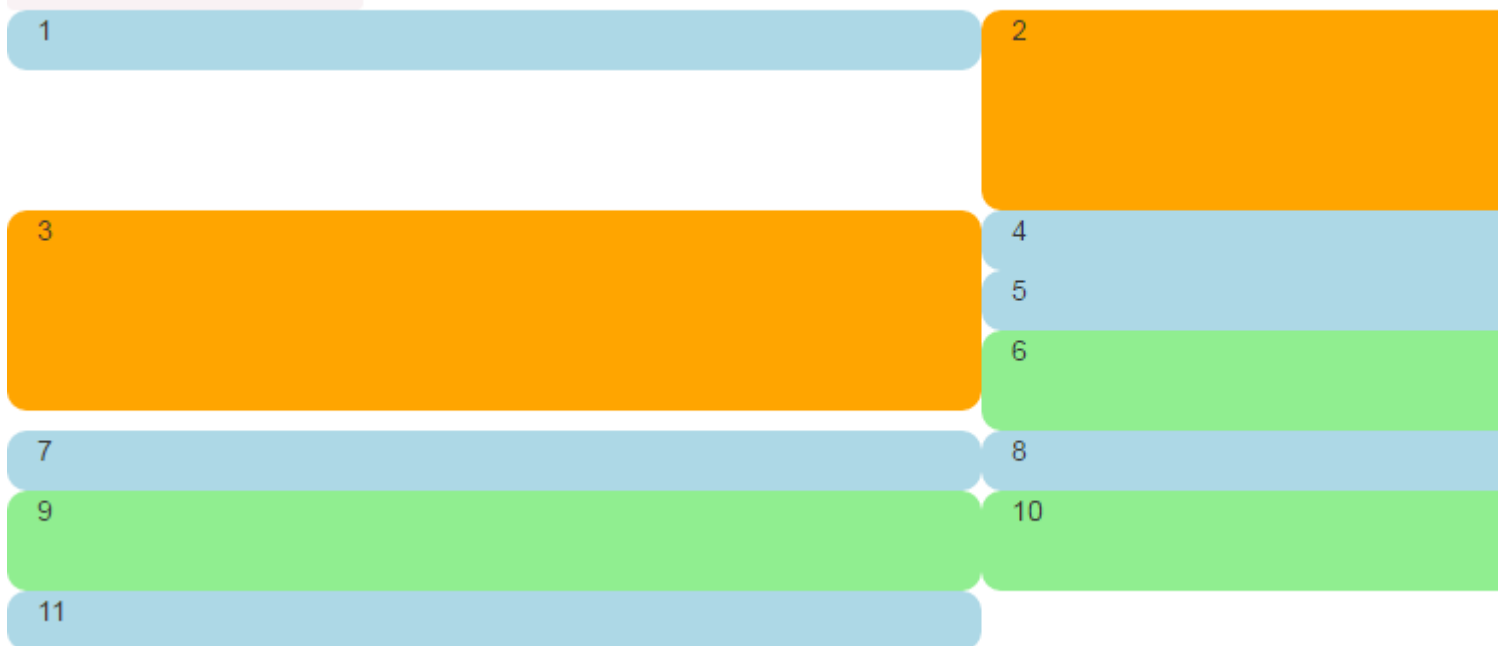
```
<div class="container-fluid">
  <div class="row">
    <div class="col-xs-6 col-md-3">1</div>
    <div class="col-xs-6 col-md-3 cell-tall">2</div>
    <div class="col-xs-6 col-md-3 cell-tall">3</div>
    <div class="col-xs-6 col-md-3">4</div>
    <div class="col-xs-6 col-md-3">5</div>
    <div class="col-xs-6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

Здесь мы добавили некоторый CSS с `cell-tall` `cell-med` и `cell-med` которые мы определили выше. Это приведет к изменению высоты некоторых ячеек. Интересно, как это будет выглядеть ...

Здесь они снова на средних и малых размерах экрана:



col-xs-6 col-md-3



О, мой, какой беспорядок. Я не думаю, что это то, что мы хотели. На средних и больших размерах 5 и 6 являются неуместными, и как-то 7 закончили тем, что начали новый ряд. При небольшом размере у нас есть две ячейки в первом ряду, а **четыре** во втором ряду, с 4, 5 и 6, все скреплены справа на обоих размерах экрана!

Итак, как мы это решаем?

Clearfix для спасения

Конечно, одним из способов помочь ситуации было бы использовать несколько `row` :

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

Это, как правило, первое, что пробуют новые Bootstrappers. Это, кажется, имеет смысл: «Я хочу четыре ячейки в каждой строке, так что я просто создать новую `row` для каждой 4 `col` дивы».

Но есть проблема с этой линией рассуждений: весь смысл Bootstrap 3 и предстоящей версии 4 должен быть **отзывчивым**. Поместив «четыре `col row`», вы на самом деле не «мысленно реагируете».

Хорошее понимание `clearfix` класса CSS поможет вам начать видеть , что несколько `row` дивы действительно было помутнение вашего понимания так , что адаптивный дизайн **предназначались** для работы. Короче говоря, вы просто **не можете** знать, сколько `col` вывести в `row` любом случае - браузер еще не выполнил вашу работу!

Помните в First Things First, мы сказали, что вам нужно думать в «обратном 12»? Без дальнейших церемоний, давайте исправим нашу проблему здесь, используя комментарии прямо в коде, чтобы, надеюсь, устранить любую путаницу. Да, это выглядит намного больше кода, но **большая** часть дополнительных комментариев.

```
<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">1</div>
</div>
</div>
```

`clearfix` - это класс CSS, который отображает крошечный (практически невидимый) `div`, и его цель - «очистить» `left` поплавки, которые были использованы `col` `div`.

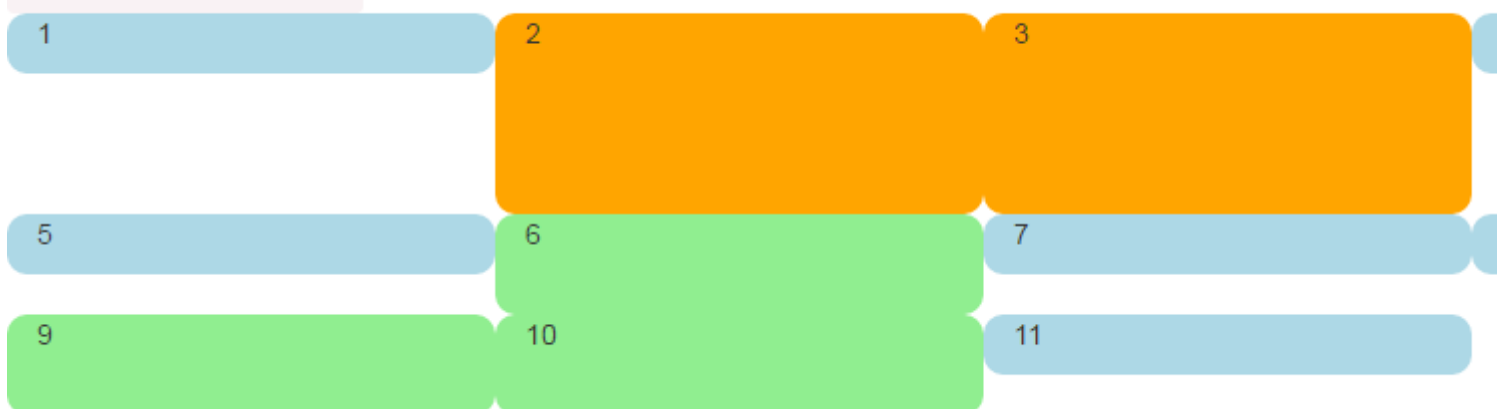
Гений действительно находится в классах `hidden-sm`, `hidden-md` и т. Д. Эти классы расположены **на `clearfix DIV`**, а не на `col` дивы! Это приводит к тому, что `clearfix div` волшебным образом появляется или исчезает из потока рендеринга при определенных ширинах видowego экрана! Genius!

У Bootstrap есть озадачивающий массив `hidden-*` и `visible-*` классов в версии 3, и, к сожалению, они не являются «обратными» друг к другу. Поэтому я считаю, что яснее всего и безопаснее просто использовать `hidden-*` классы в `clearfix`.

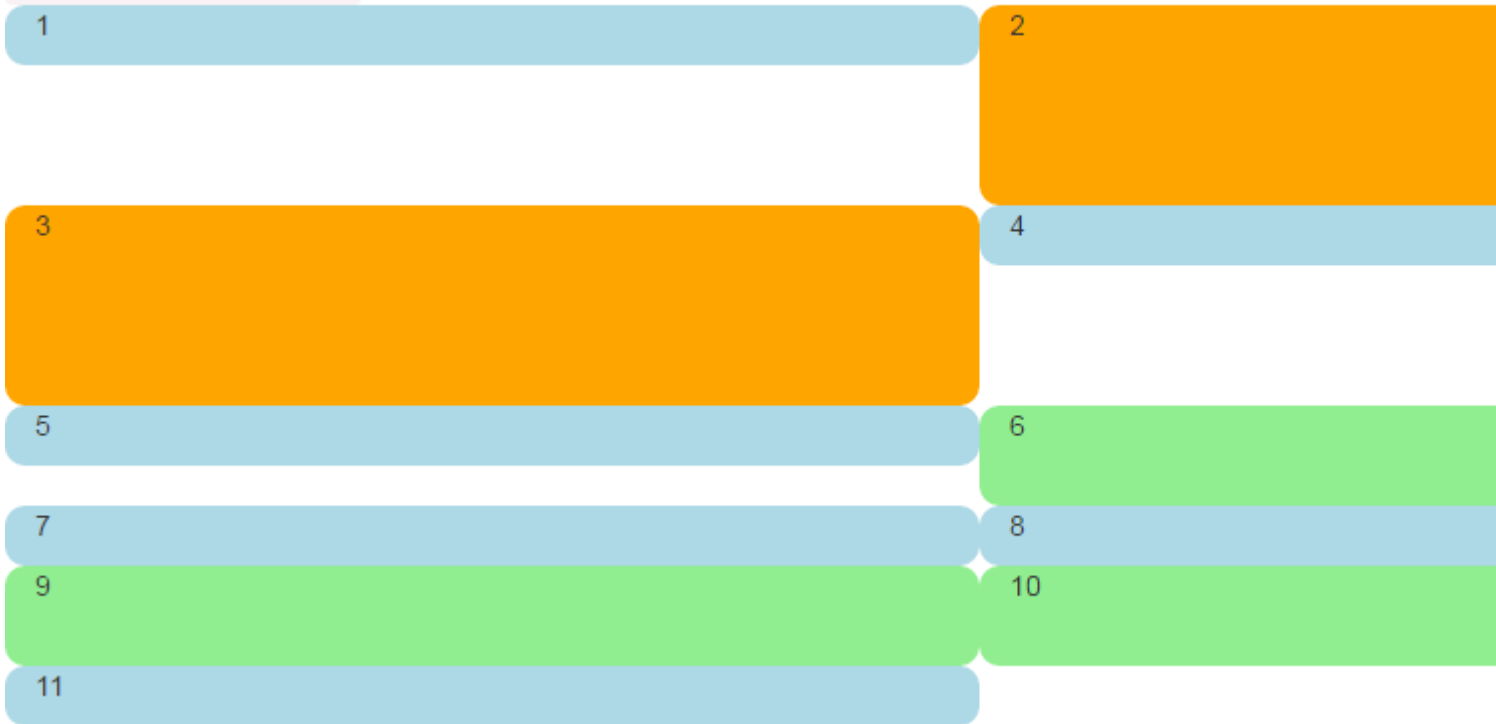
Похоже, что это может измениться к лучшему в Bootstrap 4, с такими классами, как `hidden-*-up` и `hidden-*-down` (они полностью избавляются от классов `visible-*`).

Достаточно достаточно слов, как это выглядит сейчас?

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



col-xs-6 col-md-3



Это то, чего мы хотим! На большом экране у нас всегда есть ЧЕТВЕРТАЯ, на меньшем экране, всегда ДВА. Больше нет штабелирования в странных местах, и пробелы - это то место, где мы ожидаем их.

Панель инструментов

Достаточно хорошо из этих цветных закругленных вещей, давайте сделаем что-то более интересное, чем цифры в этих div. Возьмем тот же набор столбцов и создадим реальную панель. Используйте следующий 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>
```

И вот код «приборной панели»:

```
<div class="container-fluid">
  <div class="row">
    <div class="col-xs-6 col-md-3">
      <div class="panel panel-default panel-med">
        <div class="panel-heading">
          Heading 1
        </div>
        <div class="panel-body">
          Body 1
        </div>
        <div class="panel-footer">
          Footer 1
        </div>
      </div>
    </div>
    <div class="col-xs-6 col-md-3 cell-tall">
      <div class="panel panel-danger panel-tall">
        <div class="panel-heading">
          Heading 2
        </div>
        <div class="panel-body">
          Body 2. Look out, this needs some attention!
        </div>
        <div class="panel-footer">
          Footer 2
        </div>
      </div>
    </div>
  </div>
  <!--
  On small and extra small devices, the viewport will render TWO cells
  (12 / 6 = 2), so we need a clearfix every TWO cells. We also need to
  say "don't show this clearfix when the viewport will render FOUR cells",
  which it will do at medium size and up (12 / 3 = 4). We do that by adding
  hidden-md and hidden-lg to our clearfix div, in effect instructing the
  browser to not show it at all on a wider screen.
  -->
  <div class="clearfix hidden-md hidden-lg"></div>
  <!-->
  <div class="col-xs-6 col-md-3 cell-tall">
    <div class="panel panel-success panel-short">
      <div class="panel-heading">
        Heading 3
      </div>
      <div class="panel-body">
        Body 3. The file has successfully uploaded.
      </div>
      <div class="panel-footer">
        Footer 3
      </div>
    </div>
  </div>
  </div>
  <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>
  </div>
```

```

        <div class="panel-footer">
            Footer 4
        </div>
    </div>
</div>
<!--
    We are never going to have more than FOUR cells. So every FOURTH cell,
    we place a clearfix that will ALWAYS show. We do this by just leaving off
    any of the hidden-* classes.
-->
<div class="clearfix"></div>
<!------>
<div class="col-xs-6 col-md-3">
    <div class="panel panel-warning panel-short">
        <div class="panel-heading">
            Heading 5
        </div>
        <div class="panel-body">
            Body 5.
        </div>
        <div class="panel-footer">
            Footer 5
        </div>
    </div>
</div>
<div class="col-xs-6 col-md-3 cell-med">
    <div class="panel panel-warning panel-tall">
        <div class="panel-heading">
            Heading 6
        </div>
        <div class="panel-body">
            Body 6.
        </div>
    </div>
</div>
</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>

```

Этот код будет выглядеть так:

Heading 1
Body 1
Footer 1

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

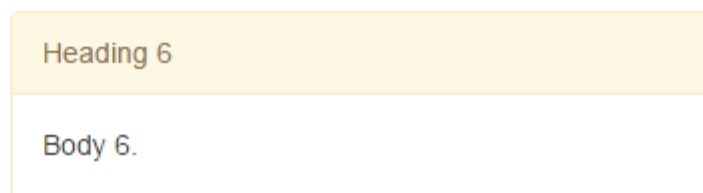
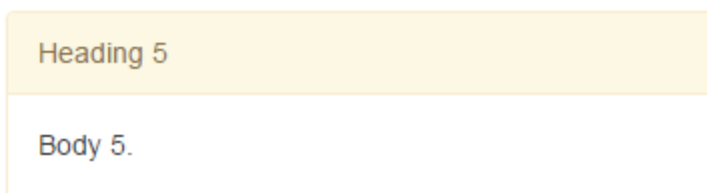
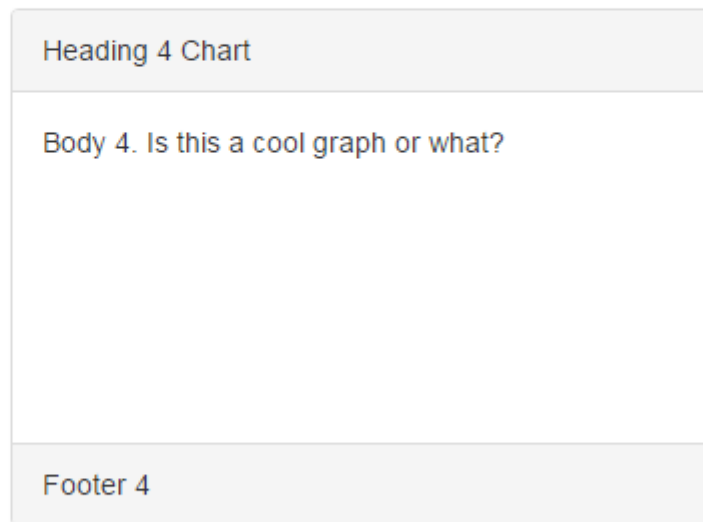
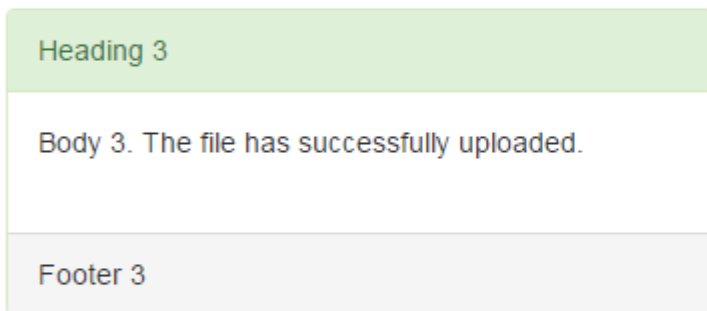
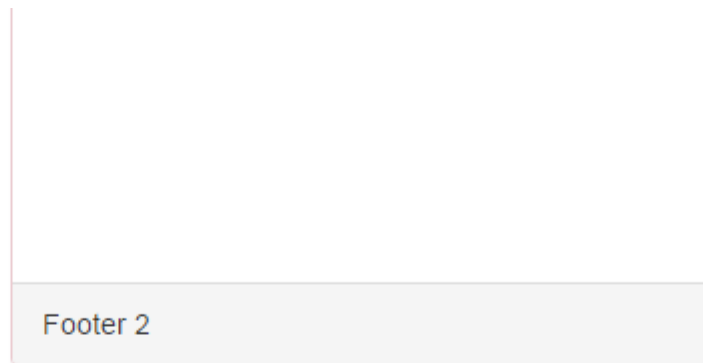
Heading 3
Body 3. The file has successfully uploaded.
Footer 3

Heading 5
Body 5.
Footer 5

Heading 6
Body 6.

Heading 7
Body 7.
Footer 7

И, как это в небольших видовых экранах:



Кстати, я использую класс `panel` Bootstrap 3, который уйдет в Bootstrap 4 и будет заменен гораздо более описательной и конкретной `card`. Посмотрев на эти изображения, вы можете понять, почему `card` будет намного лучше, чем двусмысленная `panel`.

2,4,6 Макет с Clearfixes

Вот макет, который отображает две, четыре или шесть ячеек в зависимости от размера экрана.

```
<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.
  -->
```

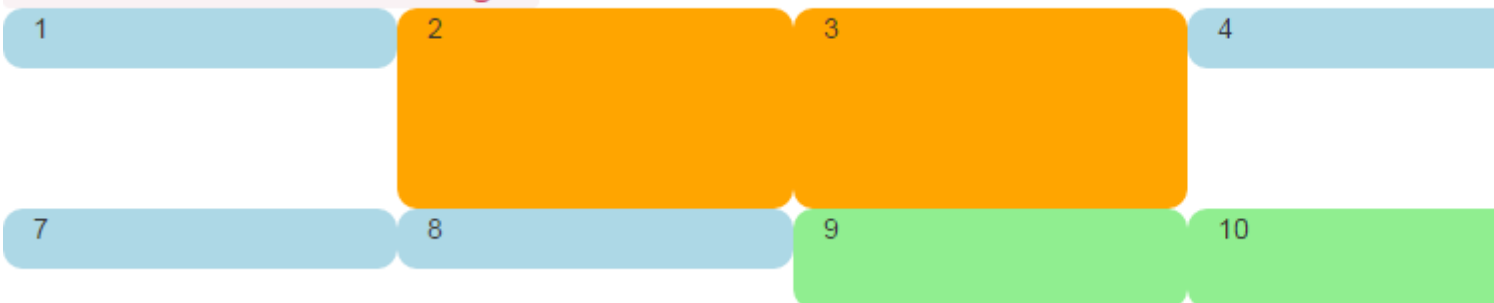
```

<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>

```

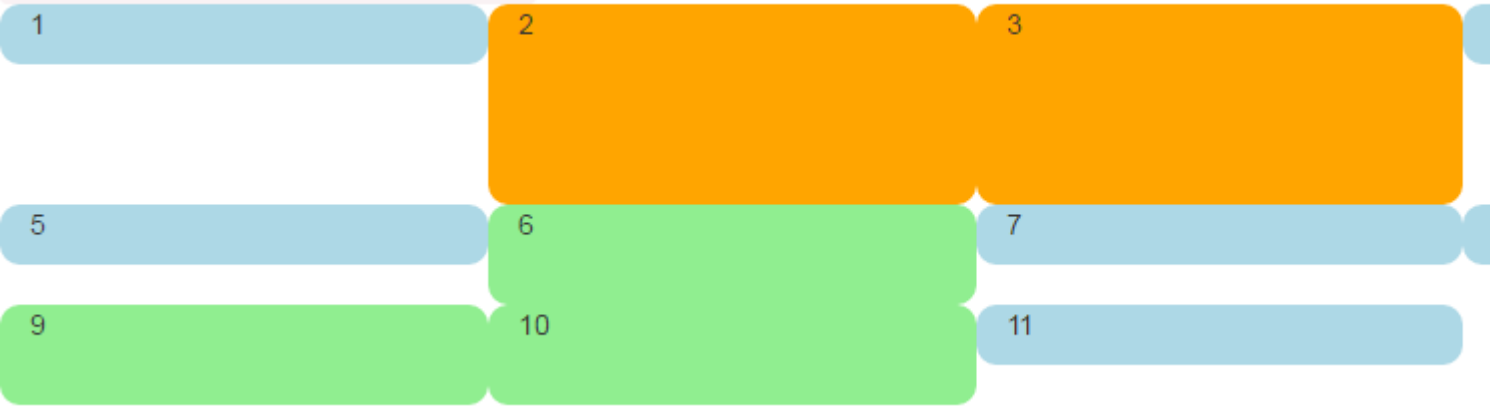
Большой экран:

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



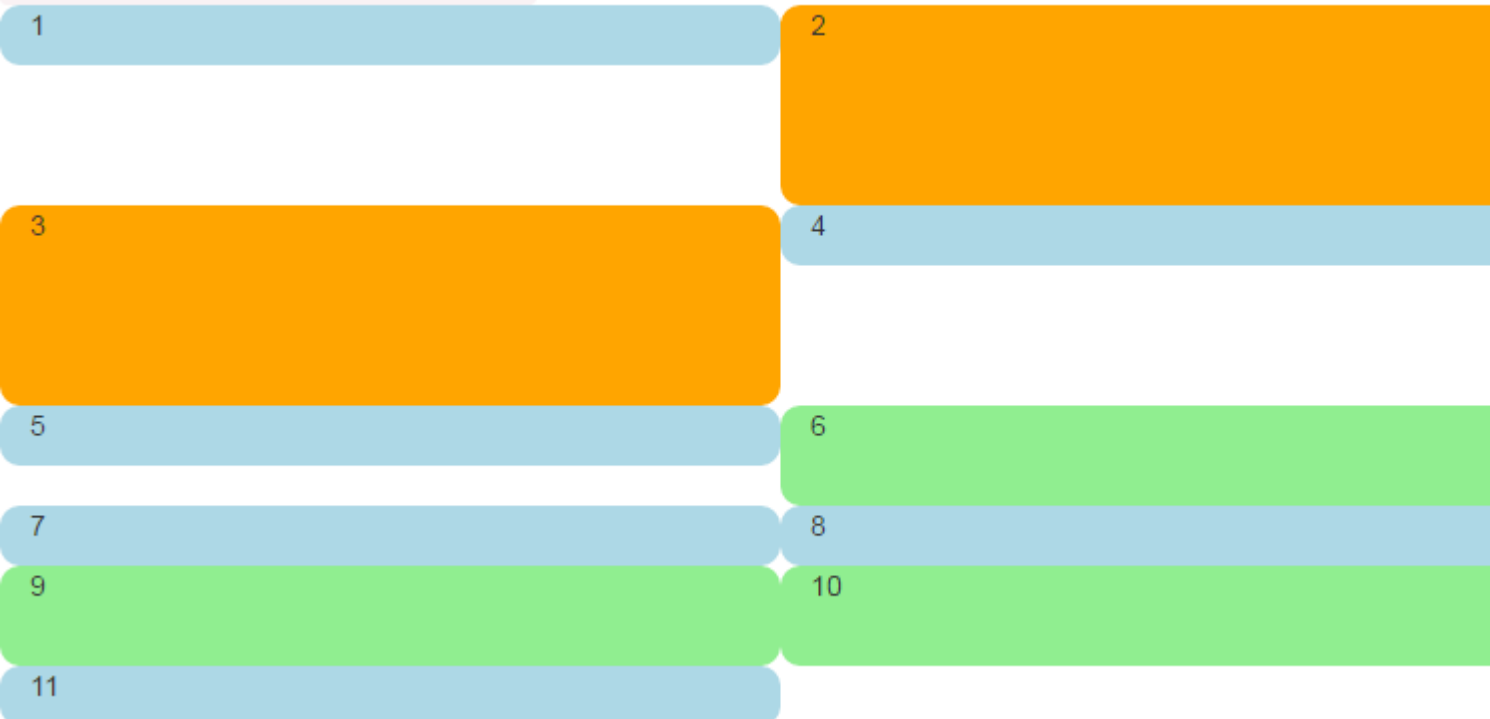
Средний экран:

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



Маленький экран:

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



Почему столбцы Bootstrap превысят 12 строк?

Существует множество сценариев реагирования, в которых *необходимо*, чтобы единицы столбца превышали 12 в одном элементе `.row`. Это называется **обертыванием столбцов**.

Если в одной строке помещено более 12 столбцов, каждая группа дополнительных столбцов будет, как одна единица, обернуться на новую строку.

Например, рассмотрим макет, где мы хотим ...

- 3 столбца на больших и средних устройствах и
- 2 столбца на маленьких и маленьких устройствах

Large



Small



Чтобы получить этот макет в Bootstrap, мы будем использовать (правильно).

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

Правильная демонстрация метода

Как вы видите в примере, *общее* количество столбцов в элементе `.row` **превышает 12**. Этот метод известен как **обертка колонок**, и это одна из самых мощных функций быстрого тестирования Bootstrap. Желаемый макет был *бы невозможен* (кроме дублирования разметки), если мы попытались придерживаться **общего заблуждения, что единицы**

столбца должны содержать до 12 в одной строке .

Компоновка *не* представляется возможным , когда мы не превышаем 12 (неправильно) ..

```
<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>
```

Неверная демонстрация метода (не работает 3 столбца на большом)

Помните, что `.row` *не* совпадает с одной строкой в окне просмотра. `.row` - это группировка столбцов. Столбцы, превышающие 12 единиц в одном элементе `.row` **на новую строку** (вниз по области просмотра). Вот почему важно понимать, что 12 столбцов представляют горизонтальные **единицы** в окне просмотра.

Кроме того, **отзывчивые сбрасывания** (clearfix) должны использоваться для равномерной упаковки, когда **столбцы отличаются по высоте** .

Прочитайте **Использование Clearfix в рядах и Cols** онлайн: <https://riptutorial.com/ru/twitter-bootstrap/topic/6124/использование-clearfix-в-рядах-и-cols>

глава 14: Карусели

замечания

Для получения дополнительной информации посетите официальную документацию по адресу <http://getbootstrap.com/javascript/#carousel> , где основаны основные примеры использования и информации об использовании JavaScript и Javascript.

Следует отметить, что карусели не функционируют корректно в IE 9 и ранее из-за использования переходов / анимаций CSS3.

Examples

Основное использование HTML

Карусель Bootstrap - это компонент Bootstrap, который создает слайд-шоу, которое проходит через элементы внутри карусели.

Вот пример использования 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>
```

Использование и инициализация основного Javascript

Компоненты карусели могут быть созданы с помощью jQuery с функцией

`$('.carousel').carousel(options)`, где `$('.carousel')` - ссылка верхнего уровня на конкретную карусель, а `options` - объект Javascript с указанием карусели атрибуты по умолчанию.

Объект `options` позволяет определить несколько свойств, которые влияют на поведение карусели. Эти свойства определяются как таковые:

- Свойство `interval` принимает тип `number` Javascript, который позволяет пользователю определить время, в течение которого карусель отображает данный карусель. Если указано логическое значение `false`, карусель не будет циклически переключаться.
- Свойство `pause` принимает тип `string` Javascript, который переключает поведение, когда автоматическая цикличность карусели приостанавливается, когда мышь пользователя входит в карусель. Принятое по умолчанию (и только) значение является «зависанием».
- Свойство `wrap` принимает `boolean` тип Javascript, который позволяет пользователю определять, хотите ли они, чтобы карусель непрерывно выполнял цикл без остановки на данном слайде.
- Свойство `keyboard` принимает `boolean` тип Javascript, который позволяет пользователю определить, хотите ли они, чтобы карусель отвечала на события клавиатуры.

Ниже приведен пример использования базового Javascript:

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

Как и в других компонентах Bootstrap, параметры карусели также можно указать в HTML через атрибуты данных.

Прочитайте Карусели онлайн: <https://riptutorial.com/ru/twitter-bootstrap/topic/1568/карусели>

глава 15: Кнопки

Синтаксис

- Классы: `.btn-default` | `.btn-primary` | `.btn-success` | `.btn-info` | `.btn-warning` | `.btn-danger` | `.btn-link`;
- Размеры: `.btn-lg` | `.btn-md` | `.btn-sm` | `.btn-XS`;
- Состояние: активное | `disabled`.

Examples

Клавишные классы

Bootstrap предоставляет несколько классов для кнопок стилизации и делает их выделяющимися.

Кнопки Bootstrap могут быть созданы путем добавления класса `.btn` к элементу.

Класс начальной загрузки	Роль (цвет)
<code>.btn-default</code>	Стандартная кнопка (белый)
<code>.btn-primary</code>	Обеспечивает дополнительный визуальный вес и определяет основное действие (синее)
<code>.btn-success</code>	Используется для указания успешного действия (зеленый)
<code>.btn-info</code>	Контекстная кнопка для предоставления информации (светло-голубой)
<code>.btn-warning</code>	Указывает, что пользователь должен будет соблюдать осторожность (желтый)
<code>.btn-danger</code>	Указывает на опасное или отрицательное действие (красный)
<code>.btn-link</code>	Сделать кнопку похожим на якорный тег.

Размеры кнопок

Вы также можете создавать разные размеры кнопок с `.btn-size`

Класс начальной загрузки	Результат
<code>.btn-lg</code>	Создает кнопку большего размера
<code>.btn-sm</code>	Создает кнопку меньшего размера
<code>.btn-xs</code>	Создает очень маленькую кнопку
<code>.btn-block</code>	Кнопки становятся элементами уровня блока и охватывают всю ширину их родительских

Активна кнопка

`active` класс заставит кнопку нажать.

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

Отключить кнопку

Добавление `disabled` класса к кнопке приведет к тому, что кнопка будет `disabled` и покажет запретный курсор при зависании над ней.

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

Кнопки рендеринга горизонтально

Несколько кнопок могут отображаться горизонтально с `.btn-group`. Просто оберните ваши кнопки внутри элемента контейнера и дайте этому элементу класс `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>
```

Кнопки рендеринга по вертикали

Применить `.btn-group-vertical` к элементу контейнера

```
<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>
```

Группировать группы кнопок в полную ширину

Кнопки, заключенные внутри элемента `.btn-group` занимают столько же ширины, сколько

необходимо. Чтобы группа `.btn-group-justified` всю ширину экрана, используйте `.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>
```

Прочитайте Кнопки онлайн: <https://riptutorial.com/ru/twitter-bootstrap/topic/4757/кнопки>

глава 16: Колонны

Examples

Ответственные столбцы одинаковой высоты (только CSS или SASS)

Вы должны добавить div с классом `.row-height` внутри строки, а также добавить `.col-height` в столбцы. Если вы хотите ограничить эффект определенным запросом на мультимедиа, просто используйте чувствительные `.row-height` и `.col-height` : например `.row-sm-height` с `.col-sm-height` .

Версия CSS:

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

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

```

}
}

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

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

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

```

```

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

```

Версия SASS (требуется 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;  
    }  
}  
  
}
```

Прочитайте Колонны онлайн: <https://riptutorial.com/ru/twitter-bootstrap/topic/6469/колонны>

глава 17: Компоненты Bootstrap

замечания

Для получения дополнительной информации посетите официальную документацию, расположенную по адресу <http://getbootstrap.com/javascript/> , откуда получен список компонентов.

Examples

Примеры компонентов Bootstrap

Компоненты Bootstrap представляют собой набор дополнительных плагинов jQuery, которые поставляются вместе с Bootstrap.

Цель компонентов Bootstrap - предоставить расширенные функции и возможности, которые было бы трудно (или невозможно) выполнить без использования Javascript. Некоторые компоненты обеспечивают чисто функциональный характер, тогда как некоторые компоненты используются для определения функциональности некоторых специальных виджетов Bootstrap.

Примеры включают [эффекты перехода](#) , [модальные диалоги](#) , [выпадающие списки](#) , [scrollspy](#) , [вкладки](#) , [всплывающие подсказки](#) , [предупреждения](#) , [popovers](#) , [кнопки](#) , [коллапс](#) , [карусели](#) и [аффиксы](#) .

Прочитайте Компоненты Bootstrap онлайн: <https://riptutorial.com/ru/twitter-bootstrap/topic/6054/компоненты-bootstrap>

глава 18: Контейнеры Bootstrap

Вступление

Используйте `.container` для адаптивного контейнера с фиксированной шириной.

Используйте `.container-fluid` для контейнера полной ширины, охватывающего всю ширину окна просмотра.

Examples

Контейнеры

`.container` имеет одну фиксированную ширину для каждого размера экрана в бутстрапе (`xs`, `sm`, `md`, `lg`);

`.container-fluid` расширяется, чтобы заполнить доступную ширину.

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

В зависимости от ширины окна просмотра, на котором просматривается веб-страница, класс контейнера дает свой `div` конкретную фиксированную ширину.

С другой стороны, ваш элемент `.container-fluid` будет постоянно изменяться, поскольку вы вносите даже самые незначительные изменения в ширину вашего браузера.

Прочитайте Контейнеры Bootstrap онлайн: <https://riptutorial.com/ru/twitter-bootstrap/topic/10908/контейнеры-bootstrap>

глава 19: Меню навигации

Examples

Меню горизонтальных таблеток

```
<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>
```

Вертикальное меню таблеток

```
<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>
```

Полноразмерная ответная горизонтальная таблетка

```
<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>
```

Прочитайте Меню навигации онлайн: <https://riptutorial.com/ru/twitter-bootstrap/topic/6404/меню-навигации>

глава 20: Миграция в Bootstrap 4

Вступление

Bootstrap 4 является **основным переписыванием**, и есть много изменений, которые следует учитывать при обновлении с Bootstrap 3. Ниже приведены изменения имени, советы и примеры переноса вашего кода Bootstrap 3.x в Bootstrap 4.x.

замечания

Это лишь небольшой пример более подробных примеров.

Examples

Изменения компоновки столбцов сетки в Bootstrap 4

Первый блок кода написан в Bootstrap 3. В Bootstrap 3 существует 4 типа спецификаций столбцов, а именно `col-md-*` `col-lg-*` `col-sm-*` `col-xs-*`. Полностью отзывчивый макет будет выглядеть так в Bootstrap 3:

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

В Bootstrap 4 они добавили новый уровень сетки `sm` ниже `768px` для более детального контроля. Таким образом, Bootstrap 4 имеет `col-*` (`xs`), `col-sm-*`, `col-md-*`, `col-lg-*`, and `col-xl-*`. Итак, то, что раньше было `.col-md-6` в `v3`, теперь `.col-lg-6` в `v4`. Обратите внимание, что infix `-xs` был удален, поэтому `.col-6` представляет 6 столбцов на дополнительной небольшой (по умолчанию) точке останова.

Итак, если теперь мы хотим написать тот же пример выше в Bootstrap 4, он будет выглядеть так:

```
<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>
```

Сетка-макет Bootstrap 4

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

Изменения в поддержке браузера

В twitter-bootstrap 4 поддержка IE8, IE9, and iOS 6 была удалена. v4 теперь только IE10+ and iOS 7+ . Для сайтов, нуждающихся в любом из них, используйте v3.

В twitter-bootstrap 4 была добавлена официальная поддержка Android v5.0 Lollipop's Browser and WebView . Более ранние версии Android-браузера и WebView остаются только unofficially supported .

Удаление класса Affix

Аффикс удален из Bootstrap 4.

Вместо этого рекомендуется использовать `position: sticky` полипол.

Если вы использовали Affix для применения дополнительных стилей без позиции, полисы могут не поддерживать ваш прецедент. Одним из вариантов такого использования является сторонняя библиотека ScrollPos-Styler.

[Согласно документации Bootstrap](#)

Выбросил плагин Affix jQuery. Вместо этого рекомендуем использовать позицию: липкий полипол. См. HTML5. Пожалуйста, введите подробную информацию и конкретные рекомендации по заполнению полиса.

Если вы использовали Affix для применения дополнительных стилей без позиции, полисы могут не поддерживать ваш прецедент. Одним из вариантов такого использования является сторонняя библиотека ScrollPos-Styler.

Если кто-то мигрирует с Bootstrap v3 на Bootstrap v4 то приведенный ниже подход приведен ниже:

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

```

CSS

```

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

```

JAVASCRIPT

```
$(document).ready(function() {

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

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

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

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

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

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

Bootstrap 4 Navbar

Новый компонент Bootstrap 4 Navbar улучшен по сравнению с предшественником Bootstrap 3.x. В Bootstrap 4 Navbar **реагирует по умолчанию** и использует **flexbox** для упрощения выравнивания контента Navbar. Также просто использовать новые `navbar-toggleable-*` для изменения точки останова Navbar. Теперь Navbar имеет 6 точек останова или «состояний», так что вы можете легко иметь один из следующих вариантов Navbar.

- Navbar *никогда не* сворачивается в вертикальное мобильное представление и всегда горизонтально.
- Navbar *всегда* сворачивается в вертикальный вид и переключается через гамбургер.
- Navbar сворачивается в вертикальный вид на *одной из 4 чувствительных точек останова*.

Базовый Bootstrap 4 Navbar

```
<nav class="navbar navbar-toggleable-md">
  <button class="navbar-toggler navbar-toggler-right" type="button" data-toggle="collapse"
```

```

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

```

Как видно из приведенного выше `navbar-header` класс `navbar-header` был удален из Bootstrap 4, и `container-fluid` больше не требуется для Navbar полной ширины.

Изменение точки останова Navbar

Класс `navbar-toggleable-md` делает вышеупомянутый краф Navbar вертикальным (и показывает значок toggler) в точке останова среды (md) 992px. Чтобы изменить это на другую `navbar-toggleable-md` останова, нам просто нужно поменять swar `navbar-toggleable-md` с одним из них.

- `navbar-toggleable` = свернуть по ширине `xs` <576px
- `Navbar-переключаемые-sm` = коллапс на `sm` ширине <768px
- `navbar-toggleable-lg` = свернуть по ширине `lg` <1200px

[Bootstrap 4 Breakpoint Navbar Demo](#)

Изменение выравнивания Navbar

Flexbox позволяет нам легко изменять выравнивание Navbar и его содержимого (бренд, ссылки, формы или текст). Содержимое Navbar по умолчанию выравнивается влево. Конечно, есть много других сценариев выравнивания ...

- Бренд слева (по умолчанию), центр ссылок и право
- Центр брендов, ссылки слева и справа
- Бренд слева и ссылки справа
- Входной сигнал формы, ссылок и заполнения
- Нет бренда, центр ссылок и права
- Бренд слева, ссылки прямо внутри контейнера
- Обозначенные ссылки (ширина заливки) с центрированием

Bootstrap 4 Navbar с центрированным маркером и левыми / правыми ссылками

```
<nav class="navbar navbar-toggleable-sm navbar-inverse bg-primary">
```



```

    <button class="navbar-toggler navbar-toggler-right" type="button" data-toggle="collapse"
data-target=".dual-collapse">
      <span class="navbar-toggler-icon"></span>
</button>
<div class="navbar-collapse collapse dual-collapse">
  <ul class="navbar-nav mr-auto">
    <li class="nav-item active">
      <a class="nav-link" href="#">Link</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#features">Link</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">Link</a>
    </li>
    <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 с маркой Left, Links Center и 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>
    </ul>
  </div>
</nav>

```

```

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

```

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

Подробнее о Bootstrap 4 Navbar

[Настройка цвета, выравнивания или высоты](#)

Bootstrap 3 в Bootstrap 4 Изменения CSS

Поскольку Bootstrap 4 является **основным** переписыванием, многие из названий классов Bootstrap 3.x были изменены или удалены. Перестройка таких компонентов, как Navbar и внедрение новых классов CSS и поддержка Flexbox, означает, что обновление до 4.x *не* является простым процессом преобразования из 3.x.

Тем не менее, есть некоторые классы CSS Bootstrap 3.x, которые имеют определенную замену Bootstrap 4.

Имя / селектор класса CSS изменяется от Bootstrap 3.3.7 до 4 (alpha 6)

{t} - представляет собой **уровень** или точку останова (т.е.: sm, md, lg и т. д.).

xs является стандартным, и его не нужно указывать: col-3 , col-6 и т. д.

{u} - представляет собой размер **единицы измерения** (то есть: 1-12)

Bootstrap 3.x	Бутстрап 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	.карта
.panel-заголовок	.card-заголовок
.panel-заголовок	.card-заголовок
.panel тела	.card-блок

Bootstrap 3.x	Бутстрап 4
.panel-сноска	.card-сноска
.panel первичной	.card-primary.card-обратная
.panel-успех	.card-success.card-обратная
.panel-инфо	.card-info.card-обратная
.panel предупреждения	.card-warning.card-обратная
.panel-опасность	.card-danger.card-обратная
.Что ж	.card.card-блок
.thumbnail	.card.card-блок
.list-inline> li	.list-рядный элемент
.dropdown-menu> li	.dropdown-пункт
.nav navbar> li	.nav-пункт
.nav navbar> li> a	.nav-ссылка
.navbar-право	.ml-авто
.navbar-BTN	.nav-пункт
.navbar фиксированной верхней	.fixed-топ
.nav стеками	.flex колонки
.btn по умолчанию	.btn-вторичный
.img реагирующих	.img-жидкость
.img-круг	.rounded-круг
.img-закругленные	.rounded
.form-горизонтальный	(Удалено)
.радио	.form проверка
.checkbox	.form проверка
.Входной-Л.Г.	.form-контроль-Л.Г.

Bootstrap 3.x	Бутстрап 4
.Входной-см	.form-контроль, см
.control метки	.form-контроль метки
.table-конденсируют	.table-см
.pagination> li	.стр-пункт
.pagination> li> a	.стр-ссылка
.вещь	.carousel-пункт
.text-помощь	.form-контроль с обратной связью
.Снять-право	.float-право
.Снять налево	.плыть налево
.center-блок	.mx-авто
.collapse.in	.collapse.show
.hidden-см	.hidden-мкр-вниз
.hidden-мкр	.hidden-Л.Г. вниз
.hidden-хз	.hidden-хз-вниз
.Visible-хз	.hidden-см-вверх
.Visible-см	.hidden-хз-down.hidden-мкр-вверх
.Visible-мкр	.hidden-см-down.hidden-LG-вверх
.Visible-Л.Г.	.hidden-мкр-down.hidden-х-вверх
.этикетка	.badge
.badge	.badge.badge-таблетки

Также см:

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

[Что нового в Bootstrap 4](#)

Бутстрап 4 Вертикальное выравнивание

Приведение элементов в **центр** или **нижнее** выравнивание по вертикали всегда было проблемой с помощью CSS и Bootstrap. Желаемое выравнивание по вертикали может находиться внутри родительского контейнера или относительно смежных элементов.

Теперь, когда Bootstrap 4 является **Flexbox по умолчанию** существует множество различных подходов к вертикальной линии с использованием: [Авто-поля](#) , [Flexbox Utilities](#) , или [Display Utilities](#) вместе с [вертикальным Совместите Utilities](#) .

Сначала [Утилиты вертикального выравнивания](#) кажутся очевидным выбором, но они работают *только* с встроенными и табличными элементами отображения. Вот некоторые варианты и сценарии вертикального выравнивания Bootstrap 4 ...

1 - Вертикальный центр с использованием автоматических полей:

Один из способов вертикального центра - использовать `my-auto` . Это будет центрировать элемент внутри контейнера. Например, `h-100` делает высоту строки, а `my-auto` будет вертикально `col-sm-12` столбец `col-sm-12` .

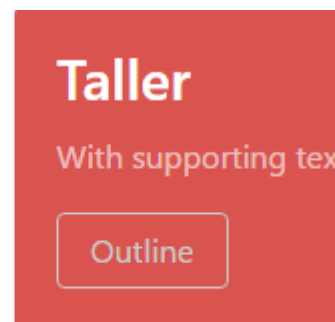
```
<div class="row h-100">
  <div class="col-sm-12 my-auto">
    <div class="card card-block w-25">Card</div>
  </div>
</div>
```

Вертикальный центр Использование Авто Margins Demo

`my-auto` представляет поля на вертикальной оси `y` и эквивалентен:

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

2 - Вертикальный центр с Flexbox:



Поскольку Bootstrap 4 `.row` теперь `display: flex` вы можете просто использовать `align-self-center` для любого столбца, чтобы вертикально `align-self-center` его ...

```

<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>

```

или, используйте `align-items-center` на всем `.row` чтобы вертикально центрировать
выровнять все `col-*` в строке ...

```

<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>

```

Вертикальный центр с разной высотой

3 - Вертикальный центр с использованием дисплеев:

В Bootstrap 4 есть [утилиты отображения](#), которые могут использоваться для `display:table`, `display:table-cell`, `display:inline` и т. Д. Они могут использоваться с [утилитами вертикального выравнивания](#) для выравнивания встроенных, встроенных блоков или элементов ячейки таблицы.

```

<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>

```

Вертикальный центр с использованием демонстрационных приложений

Бутстрап 4 Центрирование

Как центрировать элемент, столбец или содержимое внутри столбца работает по-разному в Bootstrap 4.

Горизонтальный центр

- `text-center` по-прежнему используется для `display:inline` элементы
- `mx-auto` заменяет `center-block` на центральный `display:block` элементы `display:block`
- `offset-*` *или* `mx-auto` можно использовать для центрирования столбцов сетки

`mx-auto` (auto x-axis margin) будет `display:block` центре `display:block` или `display:flex` элементы с определенной шириной (%, vw, px и т. д.). **Flexbox** используется по умолчанию для столбцов сетки, поэтому существуют также различные методы центрирования флексонов.

Текст центра или встроенные элементы: `text-center`

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

Центральный `display:block` или `display:flex` : `mx-auto`

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

Центральные столбцы с использованием смещений: `offset-*`

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

Столбцы также могут быть центрированы: `mx-auto`

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

[Демо- бутстрап 4 Горизонтальное центрирование](#)

Вертикальный центр

Для вертикального центрирования в Bootstrap 4 (ось y) см. Документы по: [Bootstrap 4 Vertical Align](#)

Бутстрап 4 Колонный заказ

Изменение порядка (или позиции) возможно в Bootstrap 3 с использованием классов push pull. В Bootstrap 4 классы **push pull** все еще работают, и, кроме **того**, можно использовать **порядок flexbox**.

В Bootstrap 4 классы **push pull** теперь `push-{viewport}-{units}` и `pull-{viewport}-{units}` а `xs-infix` удалены. Рассмотрим этот пример, который изменяет порядок столбцов на 1-3-2 макета на `xs` и `sm`:

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

[Bootstrap 4 Push Pull Demo](#)

Поскольку новая версия 4 является гибкой, **другой вариант** заключается в использовании **классов утилиты flexbox** для изменения порядка столбцов. Теперь полная ширина, 12 единиц столбца `col-*-12` столбцов могут быть отменены с **помощью** упорядочивания **гибких дисков**.

```
<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>
```

[Демо-версия Flexbox](#)

Прочитайте Миграция в Bootstrap 4 онлайн: <https://riptutorial.com/ru/twitter-bootstrap/topic/9090/миграция-в-bootstrap-4>

глава 21: модальности

замечания

Для модалов требуется, чтобы функция bootstrap.min.js функционировала должным образом.

Более подробную информацию можно найти здесь:

<http://getbootstrap.com/javascript/#modals>

Examples

Основные HTML-модальные

Модальное диалоговое окно, которое можно отобразить на текущей странице.

```
<!-- 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>
```

Прочитайте модальности онлайн: <https://riptutorial.com/ru/twitter-bootstrap/topic/6320/>
модальности

глава 22: Модальные диалоги

замечания

Для получения дополнительной информации посетите официальную документацию по адресу <http://getbootstrap.com/javascript/#modals> , откуда был получен пример «Основное использование HTML».

Examples

Основное использование HTML

Модальный диалог Bootstrap - это компонент Bootstrap, который создает модальное диалоговое окно, которое перемещается по содержимому на уровне страницы.

Ниже приведен пример базового использования модального диалога Bootstrap в 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 -->
```

Использование и инициализация основного Javascript

Модальные диалоговые компоненты могут быть созданы через jQuery с помощью функции `$('#myModal').modal(options)` , где `$('#myModal')` является ссылкой верхнего уровня на конкретный модальный диалог, а `options` - это объект Javascript, определяющий атрибуты модального диалога по умолчанию.

Объект `options` позволяет определить несколько свойств, которые влияют на поведение модального диалогового окна. Эти свойства определяются как таковые:

- Свойство `backdrop` позволяет пользователю определить, хотите ли они, чтобы серый

фоновой надписью отображался за модальным диалогом. Будут распознаны как логические значения, так и строка «статические». Если указано «статический», модальный диалог не будет закрыт, когда пользователь нажмет на фоновое оверлей.

- Свойство `keyboard` позволяет пользователю определить, нужно ли закрывать модальное диалоговое окно, когда клавиша эвакуации нажата на клавиатуре.
- Свойство `show` позволяет пользователю определить, хотите ли они, чтобы модальный диалог появлялся, когда модаль инициализирован.

Ниже приведен пример использования базового Javascript:

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

Как и в случае с другими компонентами Bootstrap, параметры модальности также могут быть указаны в HTML через атрибуты данных.

Прочитайте Модальные диалоги онлайн: <https://riptutorial.com/ru/twitter-bootstrap/topic/5927/модальные-диалоги>

глава 23: Настройка стиля загрузки Bootstrap

замечания

Следует отметить, что стоит упомянуть имя `custom.css` после основного `bootstrap.css`, иначе значения `custom.css` не будут реализованы.

Examples

Переопределение CSS по умолчанию

Все любят [twitter bootstrap](#), но некоторым из нас не нравится дизайн по умолчанию. Итак, вот простой справочник о том, как начать настройку дизайна bootstrap. Twitter bootstrap при клонировании предоставляет набор файлов css по умолчанию, которые мы можем переопределить.

Файл электронной почты css, который нужно переопределить, - это `bootstrap.min.css` под `bootstrap/dist/css`.

Чтобы переопределить проект по умолчанию, выполните следующие два простых шага.

1. Создайте `custom.css` (или вы можете назвать его как хотите) и свяжите его с вашим `index.html`

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

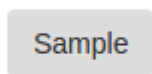
  <link rel="stylesheet" type="text/css" href="path/to/bootstrap.min.css">
  <!-- This must be declared after the bootstrap.min.css -->
  <link rel="stylesheet" type="text/css" href="path/to/your/custom.css">
</head>
<body>
  <!-- Do something -->
</body>
</html>
```

2. Начните настройку. Например, мы хотим изменить цвет кнопки по умолчанию. Если вы хотите использовать стиль кнопки загрузки по умолчанию, вам нужно добавить класс `btn` **B** `<button class="btn">Sample</button>`. Просто напишите следующий код на вашем `custom.css`.

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

Вышеприведенный код даст что-то подобное.

По умолчанию :



Пользовательский:



Этот метод избавит нас от перезаписи всех стилей кнопок, которые уже были написаны вкладчиками-разработчиками. Это также спасло нас от написания нашего собственного класса css, который для меня менее утомительный.

Прочитайте [Настройка стиля загрузки Bootstrap онлайн: https://riptutorial.com/ru/twitter-bootstrap/topic/6030/настройка-стиля-загрузки-bootstrap](https://riptutorial.com/ru/twitter-bootstrap/topic/6030/настройка-стиля-загрузки-bootstrap)

глава 24: пагинация

Вступление

Разделительные ссылки указывают, что существует ряд связанных материалов на нескольких страницах. Обычно они используются, когда многостраничный подход к длинным спискам контента улучшает общую производительность, например, в результатах поиска или в почтовых ящиках.

Examples

Простой пример разбивки на страницы

```
<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>
```

который дает вам:



Прочитайте пагинация онлайн: <https://riptutorial.com/ru/twitter-bootstrap/topic/10605/пагинация>

глава 25: Панели

замечания

Компонент панели в бутстрапе - это (ограниченное) поле с некоторым дополнением вокруг его содержимого и, возможно, контейнеры заголовка и нижнего колонтитула.

Examples

Основной пример

По умолчанию, все `.panel` делает это, применяя некоторые основные границы и отступы, чтобы содержать некоторый контент.

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

Панель с заголовком

Легко добавьте контейнер заголовка в панель с `.panel-heading`. Вы можете также включить любой `<h1>`-`<h6>` с `.panel-title` чтобы добавить заголовок с предварительным стилем.

Однако размер шрифта `<h1>`-`<h6>` переопределяется `.panel-heading`.

Для правильной окраски ссылок обязательно размещайте ссылки в заголовках внутри `.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>
```

Панель с нижним колонтитулом

Кнопки `.panel-footer` или дополнительный текст в `.panel-footer`. Обратите внимание, что нижние колонтитулы панели **не** наследуют цвета и границы при использовании контекстных вариаций, поскольку они не должны быть на переднем плане.

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

Прочитайте Панели онлайн: <https://riptutorial.com/ru/twitter-bootstrap/topic/2848/панели>

глава 26: Печать в Bootstrap.

Examples

Основное использование HTML

Элементы печати Bootstrap позволяют указать, какие элементы должны быть видны при печати и которые должны быть скрыты.

Чтобы сделать что-то видимое, используйте одно из следующих значений в зависимости от элемента и как оно должно появляться при печати:

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

Чтобы скрыть что-то от печати, используйте следующее:

```
.hidden-print
```

Прочитайте Печать в Bootstrap. онлайн: <https://riptutorial.com/ru/twitter-bootstrap/topic/6707/печать-в-bootstrap->

глава 27: подсказка

замечания

Всплывающая подсказка представляет собой элемент пользовательского интерфейса, который выглядит как маленький всплывающий ящик. Обычно это срабатывает, когда пользователь наводит указатель на другой элемент, не нажимая его.

По соображениям производительности всплывающие подсказки должны быть инициализированы с помощью jQuery. Следующий код будет включать все всплывающие подсказки в DOM:

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

Examples

Подсказки по позиционированию

По умолчанию всплывающая подсказка появится поверх элемента. Мы можем использовать атрибут `data-placement` чтобы установить положение всплывающей подсказки сверху, снизу, слева или справа от элемента.

```
<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

Мы также можем использовать `data-placement="auto"`, чтобы динамически переориентировать всплывающую подсказку. Всплывающая подсказка в следующем примере подсказка будет отображаться слева, когда это возможно, иначе она будет отображаться справа.

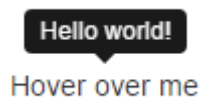
```
<a href="#" data-toggle="tooltip" data-placement="auto left" title="To the left?">Hover</a>
```

Основной пример

Чтобы создать всплывающую подсказку, нам нужно добавить атрибут `data-toggle="tooltip"`

и `title` элемента HTML, который будет иметь всплывающую подсказку. Атрибут `Title` используется для указания текста, отображаемого внутри всплывающей подсказки.

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



Прочитайте подсказка онлайн: <https://riptutorial.com/ru/twitter-bootstrap/topic/3731/подсказка>

глава 28: Полезные классы

Examples

Сгенерировать классы `.hidden-*` для всех точек останова - 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();
```

Прочитайте Полезные классы онлайн: <https://riptutorial.com/ru/twitter-bootstrap/topic/6217/полезные-классы>

глава 29: Проверка бутстрапа

замечания

- Этот метод валидации можно использовать только на входах, которые находятся внутри формы.
- Свойства должны иметь хотя бы одно требование проверки, чтобы показывать выделение при неудачной `onSubmit()`. Типы данных (кроме строки) имеют требование скрытого типа данных, поэтому не требуют явной аннотации данных. Строки не имеют этого, поэтому для принудительной проверки проверки вместе с другими полями добавьте аннотацию данных `[MinLengthAttribute(0)]`.

Examples

Использование ASP.NET MVC и аннотации данных

Добавьте в `Web.config` (в папке `Views`) `<appSettings>` : `<appSettings>` :

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

Добавьте пакет `jqueryval` в `BundleConfig.cs`:

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

Добавьте все страницы, требующие проверки (или `_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>
```

Добавьте аннотации данных в соответствующие поля в модели:

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

В представлении добавьте следующее для каждого ввода, который требует проверки:

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

Добавьте дополнение к соответствующему действию контроллера для добавления проверки на стороне сервера:

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

Пример ввода, требующий проверки

Модель:

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

Посмотреть:

```
<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>
```

Необязательный

Добавьте следующий jQuery для проверки входов при размытии, а также для отправки:

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

Прочитайте Проверка бутстрапа онлайн: <https://riptutorial.com/ru/twitter-bootstrap/topic/6388/проверка-бутстрапа>

глава 30: Сетевая система

Вступление

Сетчатая система Bootstrap состоит из 12 единиц, известных как **Columns** (`.col-*-*` CSS classes), которые используются для компоновки содержимого *слева направо* через видовое окно. Столбцы содержатся в классе **строк** (`.row` CSS) для создания горизонтальных групп столбцов. Строки помещаются в **контейнер** с фиксированной или полной шириной (`.container` или `.container-fluid`, соответственно) для правильного выравнивания. В столбцах есть отступы, которые создают промежуток (известный как «водосточный желоб») между содержимым в столбцах.

замечания

Bootstrap включает в себя гибкую мобильную первую систему флюидов, которая соответственно масштабирует до 12 столбцов при увеличении размера устройства или видового экрана. Он включает в себя predefined классы для быстрого создания макетов страниц через ряд строк и столбцов, в которых размещается ваш контент.

Examples

Запросы мультимедиа

Медиа-запросы в Bootstrap позволяют перемещать, отображать и скрывать контент на основе размера видового экрана. Следующие медиа-запросы используются в файлах LESS для создания ключевых контрольных точек в сетке Bootstrap:

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

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

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

Иногда они расширяются до максимальной ширины для ограничения CSS для более узкого набора устройств:

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


Уровни сетки бутстрапа (точки останова)

В дополнение к концепции **блоков столбцов**, Bootstrap имеет разные **точки останова** или размеры сетки, известные как уровни. В сетке Bootstrap 3 есть четыре (4) уровня для размещения различных экранов (или видовых экранов). Уровнями Bootstrap 3 являются `xs`, `sm`, `md` и `lg`. Столбцы сетки Bootstrap идентифицируются с помощью различных классов CSS-столбцов `col-{breakpoint}-{units}`.

Каждый уровень сетки **охватывает диапазон**, который предназначен для наилучшего соответствия типичной ширине экрана устройства, такой как рабочие столы, ноутбуки, планшеты и смартфоны.

Bootstrap использует мультимедийные запросы CSS для создания ответных контрольных точек, которые устанавливают границу для каждого размера сетки. Эти размеры сетки позволяют изменять расположение колонок, чтобы наилучшим образом соответствовать разным ширинам экрана и устройствам. Суть гибкого дизайна.

- `col-xs-*` - для самых *маленьких* экранов, таких как смартфоны <768 px
- `col-sm-*` - для *небольших* экранов, таких как смартфоны и планшеты > = 768 px
- `col-md-*` - для *средних* экранов, таких как планшеты и ноутбуки > = 992 px
- `col-lg-*` - для *больших* экранов, таких как рабочие столы > = 1200 пикселей

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

Ссылка: [Grid System](#)

Та же ширина столбца для каждого устройства

Чтобы создать столбец, который всегда равен 50% ширины окна просмотра (на всех устройствах), вы можете установить `col-*-6` для каждого уровня.

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

Однако это необязательная дополнительная разметка, так как `col-xs-6` означает 6 единиц на `xs` и выше. Самый маленький уровень, который вы установили (`xs`, `sm` или `md`), также определяет размер для большей ширины экрана. Для столбца того же размера на всех уровнях просто установите ширину для самого маленького окна просмотра.

Более короткий код:

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

Разная ширина столбца для каждого устройства (отзывчивый дизайн)

Классы `col-*-*` можно **комбинировать** для управления шириной столбцов по разным размерам сетки.

Например, создайте столбец шириной 50% на уровне `sm` и столбце ширины 25% на уровне `md`
...

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

`sm`, `md` и `lg` будут «стекаться» вертикально при ширине видowego экрана менее 768 пикселей. Здесь помещается сетка `xs`. Столбцы, которые используют классы `col-xs-*`, не будут складываться вертикально и будут уменьшаться на самых маленьких экранах.

Строки и столбцы Bootstrap

Система сетки Bootstrap имеет **12 единиц**, известных как **столбцы**, которые можно использовать для размещения содержимого по горизонтали через область просмотра.

Причиной для 12-элементной сетки (вместо 10, 16 и т. Д.) Является то, что 12 равномерно делит на 6 (половинки), 4 (четверти) и 3 (трети). Это облегчает адаптацию к различным макетам. Столбцы сетки Bootstrap идентифицируются с помощью различных классов CSS-столбцов `col-{breakpoint}-{units}`. [Узнайте больше о ширине и точках обзора \(AKA Tiers\)](#)

Так, например, `col-md-3` представляет столбец, который занимает 3 из 12 единиц (или 25%) в окне просмотра ширины среды (`md`). Чтобы использовать ширину столбца в макете, просто используйте соответствующий класс `col-{breakpoint}-{units}` в своей разметке HTML.

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

Ширина столбца - текучая (не фиксированная ширина), поэтому столбцы потребляют *процент* от их контейнера.

Единицы столбцов в Bootstrap 3

- `col-*-1` : 1 из 12 (8,33333333% ширины)
- `col-*-2` : 2 из 12 (16.66666667% ширина)
- `col-*-3` : 3 из 12 (ширина 25%)
- `col-*-4` : 4 из 12 (ширина 33.3333333%)
- `col-*-5` : 5 из 12 (41.66666667% ширина)
- `col-*-6` : 6 из 12 (ширина 50%)
- `col-*-7` : 7 из 12 (58.33333333% ширина)
- `col-*-8` : 8 из 12 (66.66666667% ширина)
- `col-*-9` : 9 из 12 (ширина 75%)
- `col-*-10` : 10 из 12 (83.33333333% ширина)
- `col-*-11` : 11 of 12 (91.66666667% ширина)
- `col-*-12` : 12 из 12 (100% ширина)

Демо- версия - 12 колонных блоков Bootstrap

Строка Bootstrap

Класс Bootstrap `.row` используется для размещения столбцов. Столбцы *всегда* должны помещаться в строки, а строки всегда должны помещаться внутри контейнера (`container` или `container-fluid`). Строка использует отрицательные поля (`-15px`) для обеспечения правильного расстояния между содержимым столбца и краем браузера. Строки используются для группировки столбцов по горизонтали.

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

Колонны заполняют `.row` по горизонтали влево-вправо, и [завернуть](#) на новую строку каждые 12 единиц столбцов. Поэтому вы можете использовать `.row s` для создания **горизонтальных разрывов** или добавить **более 12** столбцов в один элемент `.row` чтобы иметь **столбцы, которые обертывают** (или складывают) вертикально вниз по области просмотра.

При использовании переноса столбцов (более 12 единиц в `.row`) вам необходимо [использовать отзывчивые сбрасывания \(или clearfixes\)](#), чтобы обеспечить равномерную упаковку неравномерного содержимого столбцов. Это важно, когда содержимое столбцов изменяется по высоте.

Подробнее о столбцах и строках сетки Bootstrap

Проблемы с компоновкой бутстрапа 3?

[Bootstrap 3 - вложенная строка может иметь столбцы, содержащие более 12?](#)

[Объяснение строки и столбца Bootstrap](#)

[Как работает сетка Bootstrap \(средняя\)](#)

Контейнеры

Bootstrap требует наличия содержащего элемента для обертывания содержимого сайта и размещения нашей сетки. Вы можете выбрать один из двух контейнеров для использования в ваших проектах.

Используйте класс `.container` для `.container` контейнера с фиксированной шириной.

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

Используйте класс `.container-fluid` для контейнера полной ширины, охватывающий всю ширину вашего окна просмотра.

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

Примечание. Контейнеры не являются гнездовыми (вы не можете поместить контейнер в другой контейнер) из-за `padding` и т. Д.

Смещение столбцов

Эти классы увеличивают левое поле столбца на * столбцы. Например, `.col-md-offset-4` перемещается `.col-md-4` на четыре столбца.

```
<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>
```

Управление порядком столбцов с помощью push и 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>
```

Это изменит порядок встроенных столбцов сетки.

Синтаксис: `.col-md-push- *` и `.col-md-pull- *`.

Больше:

[Управление порядком столбцов с использованием col-lg-push и col-lg-pull в Twitter.](#)

[Загрузочный файл 3](#)

[Bootstrap 3: Push / pull колонки только на меньших размерах экрана](#)

[Колонка заказа и штабелирования в Bootstrap 3](#)

Прочитайте [Сетевая система онлайн](https://riptutorial.com/ru/twitter-bootstrap/topic/3330/): <https://riptutorial.com/ru/twitter-bootstrap/topic/3330/>
[сетевая-система](#)

глава 31: Сетка вложения

Вступление

В Bootstrap можно использовать столбцы сетки *внутри* других столбцов. Это полезно при создании расширенных адаптивных макетов, которые используют **несколько уровней сетки**.

замечания

У нас может быть как можно больше столбцов.

Examples

Вложенные столбцы

```
<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>
```

Пример взят из [<http://getbootstrap.com/css/#grid-nesting>][1]

как предполагает документация официального веб-сайта

Чтобы вложить содержимое в сетку по умолчанию, добавьте новый `.row` и набор столбцов `.col-sm-*` в существующий столбец `.col-sm-*`. Вложенные строки должны включать в себя набор столбцов, которые составляют до 12 или менее (не требуется, чтобы вы использовали все 12 доступных столбцов).

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

Прочитайте Сетка вложения онлайн: <https://riptutorial.com/ru/twitter-bootstrap/topic/9088/>

глава 32: Список групп

замечания

Вы должны знать, как использовать [кнопки](#) бутстрапа и небольшую информацию о классах [контекста](#) .

Examples

Основной пример

```
<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>
```

Значки

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

Связанные элементы

```
<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>
```

Элементы кнопки

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



```
</div>
```

Отключенные предметы

```
<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>
```

Контекстные классы

```
<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>
```

Пользовательский контент

```
<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>
```

Прочитайте Список групп онлайн: <https://riptutorial.com/ru/twitter-bootstrap/topic/6347/список-групп>

глава 33: таблицы

Examples

Простая таблица

Хотя эффекты стилизации могут варьироваться в зависимости от темы, класс `.table` используется для создания единообразного и последовательного внешнего вида таблиц в приложении:

```
<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>
```

Прочитайте таблицы онлайн: <https://riptutorial.com/ru/twitter-bootstrap/topic/6299/таблицы>

глава 34: таблицы

замечания

Контент содержания и сложные таблицы. Остерегайтесь того, что стиль `table-reflow` изменяет визуальный порядок содержимого. Убедитесь, что вы применяете этот стиль только к **хорошо сформированным** и простым таблицам данных (и, в частности, не используйте это для таблиц макета) с соответствующими ячейками заголовка таблицы для каждой строки и столбца.

Кроме того, этот класс не будет корректно работать для таблиц с ячейками, которые охватывают несколько строк или столбцов (используя атрибуты `rowspan` или `colspan`).

Examples

Основная таблица

Bootstrap определяет пользовательский стиль для таблицы, используя класс `.table`. Просто добавьте класс `.table` в любую `<table>` чтобы увидеть горизонтальные разделители и отступы:

```
<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>
```

Стол с расширенным дизайном

Bootstrap предоставляет несколько классов для расширенного стиля таблицы.

Полосатые ряды

У вас будет таблица с полосатыми рядами, если вы добавите `.table-striped`:

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

Обратите внимание, что:

Полосатые таблицы стилизованы с помощью селектора CSS `:nth-child`, который недоступен в Internet Explorer 8.

Граничный стол

У вас будет таблица с границами со всех сторон таблицы и ячеек, если вы добавите `.table-bordered`:

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

Наведение по строкам

Если вы добавите `.table-hover`, у вас будет таблица с выделенными строками, когда пользователь будет `.table-hover` над строкой:

```
<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>
```

Сжатый стол

Если вы добавите `.table-condensed`, то по умолчанию для ячейки будет разрезано пополам, поэтому у вас будет более компактная таблица:

```
<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>
```

Контекстные классы

Столбы Bootstrap поддерживают контекстные цвета. Для того, чтобы изменить цвет фона строки таблицы или ячейки нужно просто добавить один из следующих классов: `contextual`, `.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>
```

Реагирующие таблицы

Вы должны обернуть любой `.table` в контейнере `html` `.table-responsive` для создания `.table-responsive` таблиц:

```
<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>
```

Ревизионные таблицы будут прокручиваться горизонтально на небольших устройствах (<768 пикселей). Не будет различий для экранов размером более 768 пикселей.

Таблица Reflow - Вертикальные заголовки

Получение таблицы с вертикальными заголовками.

Twitter bootstrap теперь поддерживает вертикальный заголовок в хорошо отформатированной нормальной таблице. Для этого просто используйте `.table-reflow`

Используйте класс twitter bootstrap `.table-reflow` на хорошо сформированном столе, чтобы получить таблицу с вертикальными заголовками. Кроме того, вы можете комбинировать с использованием `.table-striped` и `.table-hover` для зависания столбцов на этот раз.

```
<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>
```

Вы должны проверить v4 alpha docs здесь: [twitter-bootstrap .table-reflow](#)

Прочитайте таблицы онлайн: <https://riptutorial.com/ru/twitter-bootstrap/topic/6360/таблицы>

глава 35: Темы загрузки

Examples

Темы Bootstrap и правила перекрываются

Что такое темы?

Существует несколько визуальных представлений для Bootstrap, которые можно найти из источников, таких как [Bootswatch](#), которые изменяют файл *bootstrap.min.css*. Вы также можете создать свою собственную тему таким образом.

Когда изменять темы и когда добавлять новые правила в файл *site.css*?

Когда изменять

`bootstrap.min.css`

файла, а когда добавить свой собственный `.css`-файл, например

`site.css`

?

Иногда существуют требования к стилю, которые необходимо выполнить, независимо от того, какую тему вы используете. Эти правила должны входить в ваш собственный `.css`-файл, такой как *site.css*, поэтому основная тема может быть изменена, правила от *site.css* будут применяться в любом случае. Для этого вам просто нужно связать тему начальной загрузки и свои собственные правила, чтобы переопределить существующие:

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

Таким образом, темы Bootstrap могут быть изменены в любое время без потери обязательных правил, применяемых с сайта *site.css*.

Прочитайте Темы загрузки онлайн: <https://riptutorial.com/ru/twitter-bootstrap/topic/6381/темы-загрузки>

глава 36: формы

Examples

Основная форма

Элементы управления формой имеют некоторый стиль по умолчанию без использования каких-либо специальных классов.

Однако метки и элементы управления могут быть обернуты `.form-group` для оптимального расстояния.

```
<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>
```

Только для чтения и отключенных входов

Добавьте атрибут `readonly` чтобы предотвратить ввод пользователя. Поле `readonly` не может быть отредактировано

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

Добавьте `disabled` атрибут, чтобы отключить поле ввода. Невозможно отредактировать раздробленное поле. Курсор изменится, чтобы сделать его более заметным.

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

Прочитайте формы онлайн: <https://riptutorial.com/ru/twitter-bootstrap/topic/6251/формы>

кредиты

S. No	Главы	Contributors
1	Начало работы с twitter-bootstrap	andreaem , Atul Mishra , bpoiss , Community , Evan , Gleb Kemarsky , H. Pauwelyn , kernal lora , Kritner , MattD , Mingle Li , Nhan , Prashanth Benny , the12 , tmg , Toby , VincenzoC , Vishnu Y S
2	Bootstrap Navbar	Ilyas karim
3	Dropdowns	alex
4	Glyphicons	Madalina Taina , tmg , Umer Farooq
5	JumboTron	Gabriel Chi Hong Lee
6	Navbar	Kritner , Krunal Mevada , kybernaut.cz
7	Navs	leowebguy
8	бдительный	JHS , Madalina Taina , tmg
9	Бутстреп-аффикс	Ilyas karim
10	Вкладки	DavidG , tmg
11	Выпадающие окна Bootstrap	Ismail Farooq , MattD
12	Значки загрузки и ярлыки загрузки	mmativ
13	Использование Clearfix в рядах и Cols	Bruce Pierson , ZimSystem
14	Карусели	alex , Boysenb3rry
15	Кнопки	Madalina Taina , Muhammad Abdullah , Richard Hamilton , the12
16	Колонны	kybernaut.cz
17	Компоненты Bootstrap	alex

18	Контейнеры Bootstrap	Neha Chopra
19	Меню навигации	Ignacio Correia
20	Миграция в Bootstrap 4	Chris Farmer , neophyte , ZimSystem
21	модальности	John Blanchard
22	Модальные диалоги	alex , mnoronha
23	Настройка стиля загрузки Bootstrap	CENT1PEDE , Vikas Yadav
24	пагинация	TheDarkKnight
25	Панели	JackPoint , tmg
26	Печать в Bootstrap.	MattD
27	подсказка	Madalina Taina , tmg
28	Полезные классы	ajju
29	Проверка бутстрапа	Amy Barrett , mnoronha
30	Сетевая система	Ani Menon , Boysenb3rry , bpoiss , Harshal Patil , leowebguy , Madalina Taina , Mingle Li , mmativ , Stephen Leppik , the12 , tmg , ZimSystem
31	Сетка вложения	neophyte , ZimSystem
32	Список групп	Ilyas karim
33	таблицы	atjoedonahue
34	Темы загрузки	KAI
35	формы	Community , Jens , Owen Pauling