

<http://bootstrap-4.ru>

Bootstrap 4

Документация



Цикорина Д.С.
Волгоград 2018

Оглавление

Начало работы	3
Быстрый старт	3
CSS	3
JS	3
Начальный шаблон страницы	4
Важные глобальные атрибуты	4
HTML5 doctype	5
Загрузка	6
Компилированные CSS и JS	6
Исходные файлы	6
Bootstrap CDN	6
Системы управления пакетами (<i>Хождение по сайтам библиотек, скачивание и распаковка архивов, копирование файлов в проект — всё это заменяется парой команд в терминале</i>)	7
npm	7
RubyGems	7
Composer	7
NuGet	8
«Отзывчивый» мета-тэг	8
Размер ширины и высоты элемента	8
«Ребут» (новая библиотека CSS, не использующая классов, обеспечивающая «модульный подход»)	8
Контейнеры	10
Отзывчивые брейкпойнты	10
Z-индекс	12
Система сеток	14
Как это устроено	14
Параметры сеток	15
Автоматическое расположение с помощью колонок	15
Равная ширина	15
Установка ширины одной колонки	16
Содержимое адаптивной ширины	17
Мультиряд одинаковой ширины	18
Адаптивные классы	18
Все брейкпойнты	18

По горизонтали	18
Ищите и создавайте нужное.....	19
Уведомления	20
Примеры.....	20
Кнопки	22
Примеры.....	22
Тэг «кнопка».....	22
Контурные кнопки (без фона)	23
Размеры.....	23
Карусель	24
Как это работает	24
Пример	24
Только слайды	24
С индикаторами.....	25
Использование	26
Через атрибуты	26
Через JavaScript.....	26
Параметры	26
Методы.....	27
.carousel ('dispose')	27
События	28
Изображения	29
Отзывчивые изображения.....	29
Мини-версии изображения.....	29
Выравнивание изображений	29
Картинка	30

Начало работы

Начните работу с Bootstrap, самым популярным в мире фреймворком для создания быстродействующих мобильных сайтов с загрузочным CDN и шаблонами стартовой страницы.

Быстрый старт

Хотите использовать Bootstrap в своем проекте? Bootstrap – библиотека шаблонов CSS, распространяемая свободно командой сайта MaxCDN. Нужна система управления пакетами или исходники Bootstrap? [Посетите страницу загрузок](#).

CSS

Вставьте этот кусок кода в <head>, перед всеми прочими файлами CSS, для загрузки Bootstrap.

```
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">
```

JS

Многие из наших компонентов требуют использования JavaScript. Точнее - им требуются [jQuery](#), [Popper.js](#) и наши собственные плагины. Для активации плагинов разместите следующий кусок кода <script> в конце страниц, прямо перед закрывающим </body>. Сперва jQuery, потом Popper.js, потом наши.

Мы используем мини-сборку [jQuery](#), но и полную можно использовать.

```
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
integrity="sha384-
KJ3o2DKtIkVYIK3UENzmM7KCKRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
crossorigin="anonymous"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
integrity="sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
crossorigin="anonymous"></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpilMquVdAyjUar5+76PVCmY1"
crossorigin="anonymous"></script>
```

Любопытно, какие компоненты явно требуют jQuery, наших JS и Popper.js? Нажмите ссылку «Показать компоненты» ниже. Если вы не уверены в том, как правильно написать «коренную» структуру страницы, можете использовать наши шаблоны.

[Показать компоненты, требующие JavaScript](#)

Начальный шаблон страницы

Удостоверьтесь, что ваши страницы сверстаны по новейшим стандартам. Они должны включать и использовать HTML5 doctype и viewport meta tag для правильного «отзывчивого» поведения страниц. Вот как должен выглядеть «верхний», коренной код страницы:

```
<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">

    <title>Hello, world!</title>
  </head>
  <body>
    <h1>Hello, world!</h1>

    <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
integrity="sha384-
KJ3o2DKtIkVYIK3UENzmm7KCKkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
crossorigin="anonymous"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.j
s" integrity="sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
crossorigin="anonymous"></script>
    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpilMquVdAyjUar5+76PVCmY1"
crossorigin="anonymous"></script>
  </body>
</html>
```

Это все, что вам нужно из общих требований к странице. Посетите [Обзорный](#) раздел или [наши официальные примеры](#) для создания контента и компонентов вашего сайта.

Важные глобальные атрибуты

Bootstrap включает некоторые глобальные стили и настройки, о которых вам необходимо знать при работе с ним; все эти стили и настройки почти полностью имеют дело с и «заточены» под *нормализацию* кроссбраузерных стилей.

HTML5 doctype

Bootstrap требует использования doctype «HTML5». Без него у вас возникнут некоторые проблемы со стилями.

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

Загрузка

Загрузите Bootstrap, чтобы получить скомпилированный CSS и JavaScript исходный код или включите его с вашими любимыми менеджерами пакетов, такими как npm, Bower, RubyGems и другие.

Компилированные CSS и JS

Загружайте компилированный, готовый к использованию код **Bootstrap v4.0.0** (для легкого внедрения в ваш проект), включающие:

- Компилированные и «облегченные» бандлы (Смотрите [Сравнение файлов CSS](#))
- Компилированные и «облегченные» плагины JS.

Сюда не входит документация, исходники или сторонние JS-«зависимости» *Зависимости передаются (внедряются) сервису в момент его создания (jQuery или Popper.js).* Зависимости передаются (внедряются) сервису в момент его создания.

Исходные файлы

Компилируйте Bootstrap со своим «файлопроводом», загружая наши исходники Sass, JS и файлы документации. Это потребует некоторого дополнительного инструментария:

- Компилятор Sass (поддерживаются: Libsass или Ruby Sass) для компиляции ваших CSS.
- [Постпроцессоры](#) для создания своих нестандартных свойств CSS. (*-webkit-, -moz-, -o-, и т.д.*)

[Инструменты сборки](#) также могут понадобиться при работе с Bootstrap, но все-таки они не совсем подходят для ваших целей.

Bootstrap CDN

Пользуйтесь кэшированной версией Bootstrap для «облегчения» страницы и ускорения загрузки.

```
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmY1"
crossorigin="anonymous"></script>
```

Если вы пользуетесь нашим компилированным JS, не забудьте подключить наши CDN версии jQuery и Popper.js перед вашими JS-файлами.

```
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KcKRp/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
crossorigin="anonymous"></script>
```

```
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-
ApNbggh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
crossorigin="anonymous"></script>
```

Системы управления пакетами (*Хождение по сайтам библиотек, скачивание и распаковка архивов, копирование файлов в проект — всё это заменяется парой команд в терминале*)

Используйте **исходники** Bootstrap в любом проекте с помощью популярных систем управления пакетами. С любой системой управления пакетами Bootstrap **потребует компилятор Sass (препроцессор) и постпроцессор** для правильной установки наших компилированных версий.

npm

Устанавливайте Bootstrap в вашем приложении на движке Node с помощью пакета [npm](#):

```
npm install bootstrap
```

`require('bootstrap')` загрузит все плагины JS Bootstrap в объект jQuery. Модуль bootstrap сам не экспортирует ничего. Вы можете вручную загрузить плагины JS Bootstrap, загружая файлы `/js/*.js` в корневую папку пакета.

пакет.json Bootstrap содержит некоторые дополнительные метаданные со следующими ключами:

- `sass` - путь к главным исходникам [Sass](#) Bootstrap
- `style` - путь к полной версии CSS Bootstrap, которая предварительно откомпилирована с помощью установок по умолчанию (без настройки)

RubyGems

Устанавливайте Bootstrap в ваши приложения на Ruby с помощью [Bundler](#) (**рекомендовано**) и системы управления пакетами [«RubyGems»](#), просто добавив следующую строку в ваш [Gemfile](#):

```
gem 'bootstrap', '~> 4.0.0'
```

Еще один способ – если вы не используете Bundler, вы можете установить gem-файл такой командой:

```
gem install bootstrap -v 4.0.0
```

[Смотри gem's README.](#)

Composer

Вы также можете устанавливать и управлять Sass и JS в Bootstrap с помощью [пакетного менеджера](#) уровня приложений для языка программирования PHP:

```
composer require twbs/bootstrap:4.0.0
```

NuGet

Если вы занимаетесь разработкой в .NET, вы также можете устанавливать и управлять базами [CSS](#), [Sass](#) и JS в вашем Bootstrap с помощью [NuGet](#):

```
Install-Package bootstrap
```

```
</html>
```

«Отзывчивый» мета-тэг

Bootstrap разрабатывался как *mobile first*, т.е. его настройки прежде всего оптимизированы под мобильные устройства, а уж потом с помощью медиа-запросов мы подгоняем масштаб компонентов как нам необходимо на прочих устройствах. **Вставьте этот кусок кода в <head>**:

```
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
```

Вы можете посмотреть пример этого в действии на странице [starter template](#).

Размер ширины и высоты элемента

Для большей простоты масштабирования в CSS мы изменяем глобальное значение `box-sizing` с `content-box` на `border-box`. Это гарантирует, что `padding` не будет влиять на конечную вычисленную ширину элемента; однако может создать проблемы для сторонних приложений, таких как Google Maps или поисковой движок Google.

В том редком случае, когда вам надо преодолеть эти возможные баги, делайте так:

```
.selector-for-some-widget {  
  box-sizing: content-box;  
}
```

С использованием фрагмента кода, приведенного выше, вложенные элементы – включая сгенерированный через `:before` и `:after` контент – будут все наследовать заданный `box-sizing` для класса `.selector-for-some-widget`.

Узнайте больше о блоковой модели и параметризации размеров на [CSS Tricks](#).

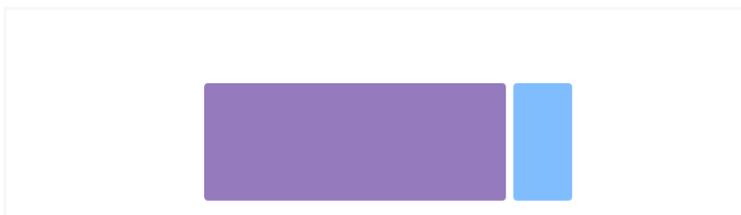
«Ребут» (новая библиотека CSS, не использующая классов, обеспечивающая «модульный подход»)

Для улучшенной кроссбраузерной визуализации мы применили «[Ребут](#)», обеспечивающий более продвинутое «сбросы стилей» для элементов HTML для коррекции мелких багов в браузерах и девайсах.

Контейнеры

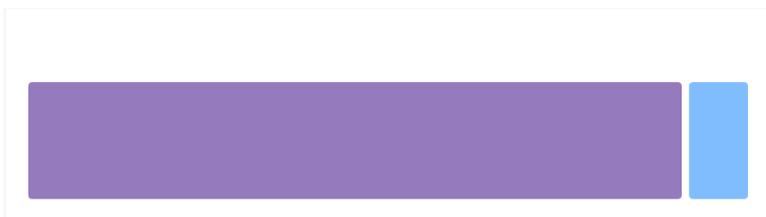
Это базовый элемент в Bootstrap и они необходимы при использовании нашей **стандартной сеточной системы**. Выбирайте отзывчивый, с фиксированной шириной (что значит, что его `max-width` изменяется на каждом брейкпойнте) или контейнер с плавающей шириной (`width ==100%` всегда).

Контейнеры *могут* иметь вложенные элементы, но в большинстве случаев можно обойтись без них.



```
<div class="container">
  <!-- Content here -->
</div>
```

Используйте `.container-fluid` для создания контейнера полной ширины, занимающий 100% зоны просмотра.



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

Отзывчивые брейкпойнты

Поскольку Bootstrap разработан как `mobile-first`, тут используются [медиа-запросы](#) `@media` для создания брейкпойнтов контента и интерфейсов. Эти брейкпойнты в основном сделаны на минимальных широтах зоны просмотра и позволяют масштабировать элементы по мере изменения размера зоны просмотра.

В исходниках Sass Bootstrap главным образом записаны следующие ранги `@media` (брейкпойнты) для разметки, сеточной системы и компонентов.

```
// Экстремальные девайсы («телефоны», < 576px)
// Нет @media, т.к. это по умолчанию в Bootstrap

// Малые девайсы («ландшафтные телефоны», >= 576px)
@media (min-width: 576px) { ... }

// Средние девайсы («планшеты», >= 768px)
@media (min-width: 768px) { ... }

// Большие девайсы (десктопы, >= 992px)
```

```
@media (min-width: 992px) { ... }

// Экстрабольшие девайсы (большие десктопы, >= 1200px)
@media (min-width: 1200px) { ... }
```

Так как все эти параметры записаны в исходниках как Sass, все `@media` доступны через миксины:

```
@include media-breakpoint-up(xs) { ... }
@include media-breakpoint-up(sm) { ... }
@include media-breakpoint-up(md) { ... }
@include media-breakpoint-up(lg) { ... }
@include media-breakpoint-up(xl) { ... }

// Example usage:
@include media-breakpoint-up(sm) {
  .some-class {
    display: block;
  }
}
```

По случаю можно использовать и другие размеры (`max-width: 575px` == «данный размер или меньше»):

```
// Экстремальные девайсы («телефоны», < 576px)
@media (max-width: 575.98px) { ... }

// Малые девайсы («ландшафтные», < 768px)
@media (max-width: 767.98px) { ... }

// Средние девайсы («таблетки», < 992px)
@media (max-width: 991.98px) { ... }

// Большие девайсы (большие десктопы, < 1200px)
@media (max-width: 1199.98px) { ... }

// Экстрабольшие девайсы (большие десктопы)
// Тут нет @media, т.к. такие брейкпойнты не имеют верхней границы по ширине
width
```

Обратите внимание, что поскольку браузеры в настоящее время не поддерживают [запросы диапазона](#), мы используем ограничения [min- и max- prefixes](#) префиксов и видовых экранов с дробной шириной (что может произойти при определенных условиях на устройствах с высоким разрешением на дюйм, например), используя значения с более высокой точностью для этих сравнений.

Опять же, эти `@media` доступны через Sass миксины:

```
@include media-breakpoint-down(xs) { ... }
@include media-breakpoint-down(sm) { ... }
@include media-breakpoint-down(md) { ... }
@include media-breakpoint-down(lg) { ... }
```

Также в наличии `@media` и миксины размеров экрана, использующие минимальные и максимальные ширины брейкпойнтов.

```
// Экстремальные девайсы («портретные телефоны», < 576px)
@media (max-width: 575.98px) { ... }
```

```
// Малые девайсы («ландшафтные», >= 575px)
@media (min-width: 576px) and (max-width: 767.98px) { ... }

// Средние девайсы («таблетки», >= 768px)
@media (min-width: 768px) and (max-width: 991.98px) { ... }

// Большие девайсы (большие десктопы, >= 992px)
@media (min-width: 992px) and (max-width: 1199.98px) { ... }

// Большие девайсы (большие десктопы, >= 1200px)
@media (min-width: 1200px) { ... }
```

Эти `@media` также доступны из миксинов Sass:

```
@include media-breakpoint-only(xs) { ... }
@include media-breakpoint-only(sm) { ... }
@include media-breakpoint-only(md) { ... }
@include media-breakpoint-only(lg) { ... }
@include media-breakpoint-only(xl) { ... }
```

Также, `@media` могут занимать несколько значений ширины брейкпойнта:

```
// Пример
// Здесь стили применяются от средних девайсов до XL-девайсов
@media (min-width: 768px) and (max-width: 1199.98px) { ... }
```

Миксины Sass для захвата таких же параметров (выше) выглядели бы так:

```
@include media-breakpoint-between(md, xl) { ... }
```

Z-индекс

Несколько компонентов Bootstrap используют данный индекс `z-index`, т.е. свойство CSS, которое помогает контролировать расположение контента с помощью третьей оси. В Bootstrap используется «стандартный» Z-индекс, спроектированный для правильного расположения навигации, всплывающих подсказок, всплывающих окон и т.н. “modals” (pop-up окно наверху страницы).

Эти высшие значения могут начинаться с любого числа, в идеале достаточно большого и специфичного, чтобы избежать конфликтов. Нам нужен стандартный набор этих чисел для использования с нашими компонентами: всплывающими подсказками, окнами, навбарами, выпадающими элементами, модальными элементами - так, чтобы мы могли бы разумно стандартизировать их поведение. Не существует ограничений - используйте хоть 100, хоть 500 и более.

Мы не советуем настраивать эти значения самому, т.к. если потребуется изменить одно, придется менять все.

```
$zindex-dropdown: 1000 !default;
$zindex-sticky: 1020 !default;
$zindex-fixed: 1030 !default;
$zindex-modal-backdrop: 1040 !default;
$zindex-modal: 1050 !default;
$zindex-popover: 1060 !default;
$zindex-tooltip: 1070 !default;
```

Для того, чтобы эффективно обращаться с перекрывающимися границами внутри компонентов (например, кнопки и формы ввода в группах форм ввода), мы используем по

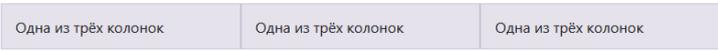
умолчанию значения `z-index` 1, 2, 3, а также `hover` и "активные состояния". В `hover/focus/active` мы подводим частный элемент к "переднему краю" использованием более высокого `z`-индекса, для показа его границы над вложенными элементами.

Система сеток

Используйте нашу мощную систему гибких сеток, предоставляющую большие преимущества для пользователей в первую очередь мобильных устройств, для удобной верстки и расположения элементов любых видов и размеров, которая работает благодаря системе «12 колонок», 5-ти адаптивных ярусов, препроцессоров Sass и десятков предустановленных классов.

Как это устроено

Система сеток Bootstrap 4 использует контейнеры, ряды и колонки, чтобы удобно располагать содержимое. Бутстрап реализован с помощью [флексбокса](#) и полностью адаптивен. Ниже приведен пример и глубокий взгляд на объединение сетки.



```
<div class="container">
  <div class="row">
    <div class="col-sm">
      Одна из трёх колонок
    </div>
    <div class="col-sm">
      Одна из трёх колонок
    </div>
    <div class="col-sm">
      Одна из трёх колонок
    </div>
  </div>
</div>
```

В данном примере с помощью наших предустановленных классов сеток были созданы 3 колонки равной ширины для маленьких, средних и больших девайсов. Эти колонки выровнены с помощью родительского класса `.container`.

Вот небольшое объяснение работы Bootstrap 4:

- Инструмент для центровки контента сайта. Используйте класс `.container` для фиксированной ширины или `.container-fluid` для 100%-ной ширины.
- Ряды – это обертка для колонок. Каждая колонка имеет горизонтальный `padding` (называемый `gutter`) для контроля пространства между колонками. Этот `padding` (падинг) влияет на ряды с отрицательным марджином. В этом случае все содержимое ваших колонок будет визуально центрировано внизу с левой стороны.
- Содержимое должно быть расположено в колонках, и только колонки могут быть расположены в рядах.
- Благодаря флексбксу колонки сетки без установленного атрибута «ширина» автоматически получают равную ширину. Например, четыре экземпляра класса `.col-sm` автоматически (на малых точках брейкпойнта) получают ширину одной колонки = 25%. Подробнее смотрите раздел [Автоматическое расположение с помощью колонок](#).
- Цифры в наименовании классов колонок показывают, сколько колонок из 12-ти возможных в ряду вы бы хотели использовать. Так, если вы хотите использовать три колонки одной ширины, используйте `.col-sm-4`.
- Ширина колонок `width` задана в процентах, что позволяет колонокам быть гибкими и изменять размер относительно их родительского элемента.

- Колонки имеют горизонтальный паддинг `padding` для создания отступов между отдельными колонками, но вы можете удалить марджины `margin` из рядов и паддинги `padding` из колонок, добавив класс `.no-gutters` в `.row`.
- Есть 5 «ярусов» сеток, по одному для каждого [«брейкпойнта»](#): все точки (экстра маленький), маленький, средний, большой и экстрабольшой.
- Ярусы сетки основаны на минимальной широте, т.е. **они подходят для каждого вышестоящего яруса** (т.е., `.col-sm-4` подходит для маленьких, средних, больших и XL девайсов).
- Вы можете использовать predefined классы сетки (например `.col-4`) или препроцессоры Sass для создания своей разметки.

Знайте об [ограничениях и багах](#), существующих во флексбоксах, таких как [невозможность использования некоторых элементов HTML как гибкие контейнеры](#).

Параметры сеток

Бутстрап использует `em` и `rem` для задания большинства размеров, а пиксели `px` — для «брейкпойнтов» сетки и ширин контейнеров. Это происходит потому, что ширина зоны видимости на каждом устройстве измеряется в пикселях и не изменяется с [размером шрифта](#).

Посмотрим, как действуют некоторые аспекты системы сеток Bootstrap на разных «ручных» устройствах.

	 Extra small <576px	 Small ≥576px	 Medium ≥768px	 Large ≥992px	 Extra large ≥1200px
Максимальная ширина контейнера	None (auto)	540px	720px	960px	1140px
Префикс класса	<code>.col-</code>	<code>.col-sm-</code>	<code>.col-md-</code>	<code>.col-lg-</code>	<code>.col-xl-</code>
Число колонок	12				
Ширина отступа	30px (15px с каждой стороны столбца)				
Может быть вложенным	Да				
Упорядочивание колонок	Да				

Автоматическое расположение с помощью колонок

Используйте классы колонок со специальными точками брейкпойнта (например, `.col-sm-6`) для легкого расположения колонок без использования явно обозначенных номеров классов.

Равная ширина

Например, здесь мы видим две сетки, которые подойдут к любому устройству и зоне видимости, от `xs` до `xl`. Добавляйте любое количество простых классов для каждого брейкпойнта, и каждая колонка будет одинаковой ширины.

1 из 2	2 из 2	
1 из 3	2 из 3	3 из 3

```

<div class="container">
  <div class="row">
    <div class="col">
      1 из 2
    </div>
    <div class="col">
      2 из 2
    </div>
  </div>
  <div class="row">
    <div class="col">
      1 из 3
    </div>
    <div class="col">
      2 из 3
    </div>
    <div class="col">
      3 из 3
    </div>
  </div>
</div>

```

Колонки одинаковой ширины могут быть представлены в виде строк, но тут может помешать [баг](#), который нуждается в добавлении атрибута `flex-basis` или `border`. Наш пример работает благодаря установленному значению `border`; вы можете сделать то же самое: `.col { border: 1px solid transparent; }`. Есть иной способ: вы можете добавить `flex-basis` к ширине колонки, т.е. `.col { flex: 1 0 50%; }`.

Лечение этого бага документировано в [reduced test case outside Bootstrap](#).

Колонка	Колонка
Колонка	Колонка

```

<div class="container">
  <div class="row">
    <div class="col">Колонка</div>
    <div class="col">Колонка</div>
    <div class="w-100"></div>
    <div class="col">Колонка</div>
    <div class="col">Колонка</div>
  </div>
</div>

```

Установка ширины одной колонки

Авто-расположение колонок в сетке флексбокса также дает возможность установить ширину одной колонки, при этом остальные «родственные» колонки автоматически изменят свой размер вокруг нее. Вы можете использовать предустановленные классы сеток (как показано внизу), миксины или инлайн-ширину. Заметим, что другие колонки будут изменять размер независимо от ширины центральной колонки.

1 из 3	2 из 3 (широкая)	3 из 3
1 из 3	2 из 3 (широкая)	3 из 3

```

<div class="container">
  <div class="row">
    <div class="col">
      1 из 3
    </div>
    <div class="col-6">
      2 из 3 (широкая)
    </div>
    <div class="col">
      3 из 3
    </div>
  </div>
  <div class="row">
    <div class="col">
      1 из 3
    </div>
    <div class="col-5">
      2 из 3 (широкая)
    </div>
    <div class="col">
      3 из 3
    </div>
  </div>
</div>

```

Содержимое адаптивной ширины

Используйте классы `col-{breakpoint}-auto` для создания колонок, изменяющих свою ширину по ширине содержимого.

1 из 3	Содержимое адаптивной ширины	3 из 3
1 из 3	Содержимое адаптивной ширины	3 из 3

```

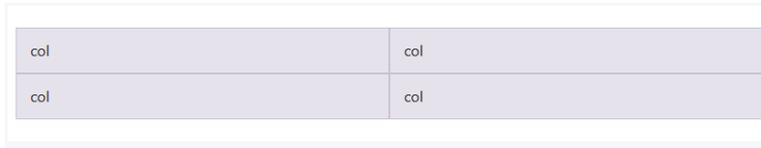
<div class="container">
  <div class="row justify-content-md-center">
    <div class="col col-lg-2">
      1 из 3
    </div>
    <div class="col-md-auto">
      Содержимое адаптивной ширины
    </div>
    <div class="col col-lg-2">
      3 из 3
    </div>
  </div>
  <div class="row">
    <div class="col">
      1 из 3
    </div>
    <div class="col-md-auto">
      Содержимое адаптивной ширины
    </div>
    <div class="col col-lg-2">
      3 из 3
    </div>
  </div>

```

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

Мультиряд одинаковой ширины

Создавайте колонки одной ширины, которые охватывают множественные ряды, добавлением `.w-100` туда, где необходимо сместить вниз на новую строку. Делайте их адаптивными, применяя `.w-100` вместе с некоторыми [адаптивными утилитами отображения](#).



col	col
col	col

```
<div class="row">
  <div class="col">col</div>
  <div class="col">col</div>
  <div class="w-100"></div>
  <div class="col">col</div>
  <div class="col">col</div>
</div>
```

Адаптивные классы

Сетка Bootstrap включает 5 «ярусов» predetermined классов, используемых для построения сложного адаптивного контента. Адаптируйте размеры своих колонок для правильного их отображения на всех видах и размерах устройств.

Все брейкпойнты

Для сеток, которые выглядят и располагаются одинаково на всех устройствах любого размера, используйте классы `.col` и `.col-*`. Определите именованный класс с цифрой, когда вам нужна колонка определенного размера, во всех остальных случаях свободно пользуйтесь `.col`.



col	col	col	col
col-8	col-4		

```
<div class="row">
  <div class="col">col</div>
  <div class="col">col</div>
  <div class="col">col</div>
  <div class="col">col</div>
</div>
<div class="row">
  <div class="col-8">col-8</div>
  <div class="col-4">col-4</div>
</div>
```

По горизонтали

Используя один набор из классов `.col-sm-*`, вы можете создать базовую сетку, которая вначале сложена по вертикали, а потом ее колонки становятся горизонтальными (на экстремальных девайсах).

col-sm-8	col-sm-4	
col-sm	col-sm	col-sm

<div class="row">

```

<div class="col-sm-8">col-sm-8</div>
<div class="col-sm-4">col-sm-4</div>
</div>
<div class="row">
  <div class="col-sm">col-sm</div>
  <div class="col-sm">col-sm</div>
  <div class="col-sm">col-sm</div>
</div>

```

Ищите и создавайте нужное

Не хотите, чтобы ваши колонки шли в одном порядке? Используйте комбинацию разных классов для каждого яруса. Смотрите пример внизу.

.col-12 .col-md-8	.col-6 .col-md-4	
.col-6 .col-md-4	.col-6 .col-md-4	.col-6 .col-md-4
.col-6	.col-6	

<!-- Размещайте колонки для

мобильных устройств: делайте одну – полной ширины, и другую - половинной -->

```

<div class="row">
  <div class="col-12 col-md-8">.col-12 .col-md-8</div>
  <div class="col-6 col-md-4">.col-6 .col-md-4</div>
</div>

```

<!-- Колонки начинаются с 50% ширины на мобильных и меняют ее на 33,3% на мониторе -->

```

<div class="row">
  <div class="col-6 col-md-4">.col-6 .col-md-4</div>
  <div class="col-6 col-md-4">.col-6 .col-md-4</div>
  <div class="col-6 col-md-4">.col-6 .col-md-4</div>
</div>

```

<!-- Колонки всегда 50% ширины, везде -->

```

<div class="row">
  <div class="col-6">.col-6</div>
  <div class="col-6">.col-6</div>
</div>

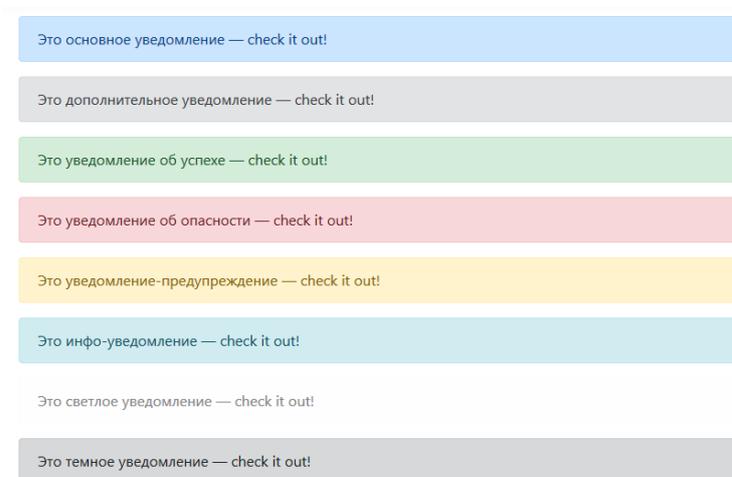
```

Уведомления

Предоставляйте контекстные сообщения обратной связи для типичных действий пользователя с помощью нескольких доступных и гибких предупреждающих сообщений.

Примеры

Уведомления доступны для любой длины как текста, так и опциональной кнопки отмены. Для правильной стилизации используйте один из 8 **требуемых** контекстуальных классов (например, `.alert-success`). Для строчного отклонения используйте [плагин уведомлений jQuery](#).



```
<div class="alert alert-primary" role="alert">
  Это основное уведомление – check it out!
</div>
<div class="alert alert-secondary" role="alert">
  Это дополнительное уведомление – check it out!
</div>
<div class="alert alert-success" role="alert">
  Это уведомление об успехе – check it out!
</div>
<div class="alert alert-danger" role="alert">
  Это уведомление об опасности – check it out!
</div>
<div class="alert alert-warning" role="alert">
  Это уведомление-предупреждение – check it out!
</div>
<div class="alert alert-info" role="alert">
  Это инфо-уведомление – check it out!
</div>
<div class="alert alert-light" role="alert">
  Это инфо-уведомление – check it out!
</div>
<div class="alert alert-dark" role="alert">
  Это темное уведомление – check it out!
</div>
```

Использование вспомогательных технологий

Использование цвета как дополнительного инструмента информативности доступно только в визуальной сфере, что ограничивает пользователей вспомогательных технологий,

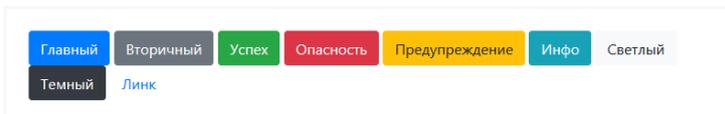
например, программ для чтения текста с экрана. Удостоверьтесь, что информация, обозначенная цветом, также доступна из самого контента (т.е. в тексте) или содержится в альтернативных средствах – таких как дополнительный скрытый в классе `.sr-only` текст.

Кнопки

Используйте стили кнопок Bootstrap для взаимодействия с формами, диалогами, т.п. Имеется поддержка некоторых контекстуальных вариаций, размеров, состояний и т.п.

Примеры

Bootstrap имеет некоторые predefined стили кнопок, каждый из которых имеет свою семантическую цель, и имеет дополнительные параметры для большего контроля и гибкости.



```
<button type="button" class="btn btn-primary">Главный</button>
<button type="button" class="btn btn-secondary">Вторичный</button>
<button type="button" class="btn btn-success">Успех</button>
<button type="button" class="btn btn-danger">Опасность</button>
<button type="button" class="btn btn-warning">Предупреждение</button>
<button type="button" class="btn btn-info">Инфо</button>
<button type="button" class="btn btn-light">Светлый</button>
<button type="button" class="btn btn-dark">Темный</button>
<button type="button" class="btn btn-link">Линк</button>
```

Использование вспомогательных технологий

Использование цвета как дополнительного инструмента информативности доступно только в визуальной сфере, что ограничивает пользователей вспомогательных технологий, например, программ для чтения текста с экрана. Удостоверьтесь, что информация, обозначенная цветом, также доступна из самого контента (т.е. в тексте) или содержится в альтернативных средствах – таких как дополнительный скрытый в классе `.sr-only` текст.

Тэг «кнопка»

Классы `.btn` созданы для использования с элементом `<button>`. Однако их также можно использовать внутри `<a>` или `<input>` (хотя из-за этого некоторые браузеры могут рендерить это немного иначе).

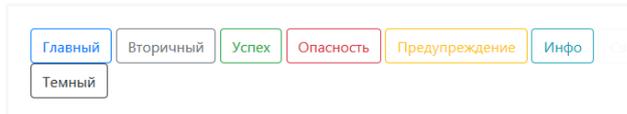
При использовании классов `.btn` в элементах `<a>`, которые в данном случае используются не как линк, ведущий на новые страницы или как якорь, а как триггер внутривстраничной функциональности (такой как сворачивание содержимого), необходимо задать этим `<a>` атрибут `role="button"` для правильного поведения в случае применения вспомогательных технологий таких как «экранный читалка».



```
<a class="btn btn-primary" href="#" role="button">Link</a>
<button class="btn btn-primary" type="submit">Button</button>
<input class="btn btn-primary" type="button" value="Input">
<input class="btn btn-primary" type="submit" value="Submit">
<input class="btn btn-primary" type="reset" value="Reset">
```

Контурные кнопки (без фона)

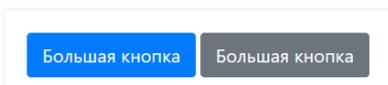
Нужна кнопка, но без «тяжелых» фоновых цветов? Замените дефолтные классы-модификаторы на классы `.btn-outline-*` для удаления всех фоновых изображений и цветов в любой кнопке.



```
<button type="button" class="btn btn-outline-primary">Главный</button>
<button type="button" class="btn btn-outline-secondary">Вторичный</button>
<button type="button" class="btn btn-outline-success">Успех</button>
<button type="button" class="btn btn-outline-danger">Опасность</button>
<button type="button" class="btn btn-outline-warning">Предупреждение</button>
<button type="button" class="btn btn-outline-info">Инфо</button>
<button type="button" class="btn btn-outline-light">Светлый</button>
<button type="button" class="btn btn-outline-dark">Темный</button>
```

Размеры

Нужно изменить размеры кнопки? Добавьте в нее классы `.btn-lg` или `.btn-sm` для получения дополнительных размеров.



```
<button type="button" class="btn btn-primary btn-lg">Большая кнопка</button>
<button type="button" class="btn btn-secondary btn-lg">Большая
кнопка</button>
```



```
<button type="button" class="btn btn-primary btn-sm">Маленькая
кнопка</button>
<button type="button" class="btn btn-secondary btn-sm">Маленькая
кнопка</button>
```

Карусель

Компонент слайд-шоу для циклического повторения элементов - карусель изображения или текстовые слайды.

Как это работает

«Карусель» - это слайд-шоу для отображения серии содержимого, созданная на основе CSS 3D-трансформаций и немного на JS. Она работает с текстом, изображениями или обычной разметкой. Она также поддерживает кнопки “next/prev”.

В браузерах с поддержкой [API видимости страниц](#) элементы карусели не будут сменяться, когда страница невидима пользователю (например, когда вкладка браузера неактивна, или окно браузера «минимизировано» - кнопка «квадратик» слева от крестика «закрыть браузер», и т.д.).

Пожалуйста, знайте, что вложенные карусели не поддерживаются, а карусели в целом не очень желательны «стандартам доступности».

В итоге, если вы используете карусели BS4, это требует [подключения util.js](#).

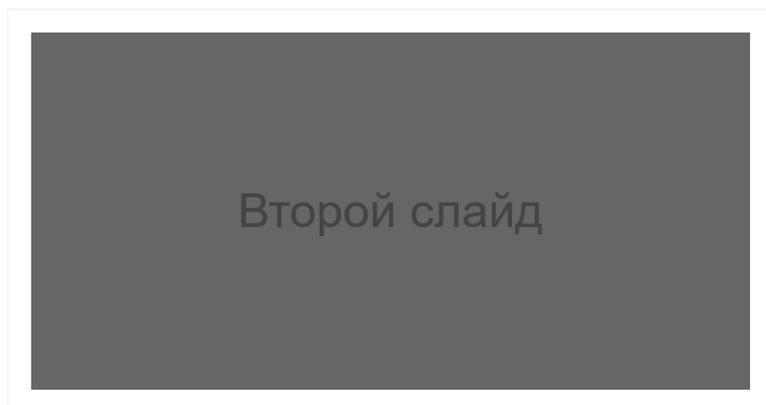
Пример

Карусель не нормализует автоматически размер содержимого слайдов (наверно нечто вроде подстройки размеров под размер слайда или типа того). Поэтому вам могут понадобиться дополнительные утилиты для придания их содержимому нужного размера. Пока карусели поддерживают кнопки «prev/next», их не надо добавлять явно. Добавьте и настраивайте их самостоятельно.

Задавайте уникальный id классу `.carousel` для возможности гибкого управления, особенно если вы используете много каруселей на странице.

Только слайды

Вот пример карусели только со слайдами. Отметим наличие классов `.d-block` и `.img-fluid` в изображениях карусели – их цель в том, чтобы предотвратить дефолтное выравнивание изображений браузером.



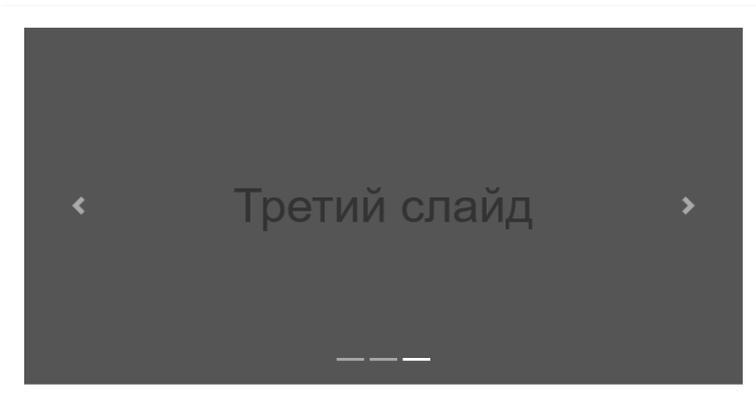
```

<div id="carouselExampleSlidesOnly" class="carousel slide" data-
ride="carousel">
  <div class="carousel-inner">
    <div class="carousel-item active">
      
    </div>
    <div class="carousel-item">
      
    </div>
    <div class="carousel-item">
      
    </div>
  </div>
</div>

```

С индикаторами

Вы также можете добавить индикаторы к карусели, наряду с органами управления.



```

<div id="carouselExampleIndicators" class="carousel slide" data-
ride="carousel">
  <ol class="carousel-indicators">
    <li data-target="#carouselExampleIndicators" data-slide-to="0"
class="active"></li>
    <li data-target="#carouselExampleIndicators" data-slide-to="1"></li>
    <li data-target="#carouselExampleIndicators" data-slide-to="2"></li>
  </ol>
  <div class="carousel-inner">
    <div class="carousel-item active">
      
    </div>
    <div class="carousel-item">
      
    </div>
    <div class="carousel-item">
      
    </div>
  </div>
  <a class="carousel-control-prev" href="#carouselExampleIndicators"
role="button" data-slide="prev">
    <span class="carousel-control-prev-icon" aria-hidden="true"></span>
    <span class="sr-only">Previous</span>
  </a>
  <a class="carousel-control-next" href="#carouselExampleIndicators"
role="button" data-slide="next">
    <span class="carousel-control-next-icon" aria-hidden="true"></span>

```

```
<span class="sr-only">Next</span>
</a>
</div>
```

Требуется начальный активный элемент

Необходимо добавлять класс `.active` в один из слайдов, иначе карусель не будет видна.

Использование

Через атрибуты

Атрибут `data-slide` принимает значения `prev` или `next`, которые изменяют позицию слайда относительно его текущей позиции. Или используйте `data-slide-to` для перехода на слайд с индексом, например, 2: `data-slide-to="2"`, индексы слайдов начинаются с 0.

Атрибут `data-ride="carousel"` используется для создания анимации карусели. **Его нельзя сочетать с явной инициализацией карусели через JavaScript.**

Через JavaScript

Вызывайте карусель вручную:

```
$('.carousel').carousel()
```

Параметры

Параметры можно передавать через атрибуты или JavaScript. Для использования атрибутов добавьте название параметра в `data-`, например: `data-interval=""`.

Имя	Тип	По умолчанию	Описание
<code>interval</code>	число	5000	Время задержки между автоматической сменой слайда. Если <code>false</code> – карусель не будет автоматически сменять слайды.
<code>keyboard</code>	boolean	<code>true</code>	Будет ли карусель реагировать на события клавиатуры.
<code>pause</code>	строка boolean	<code>"hover"</code>	Если стоит <code>"hover"</code> – смена слайдов тормозится по <code>mouseenter</code> , и начинает смену по <code>mouseleave</code> . Если <code>false</code> – наведение на карусель не остановит смену слайдов.
			Устройства, активируемые касанием: <code>"hover"</code> – пауза при <code>touchend</code> (когда пользователь закончил взаимодействие с каруселью) на два интервала, потом опять смена слайдов. Заметьте, что это поведение – дополнение к описанному выше поведению мыши.
<code>ride</code>	строка	<code>false</code>	Автосмена слайдов карусели после первой ручной смены слайда юзером. Если <code>carousel</code> – автосмена включается после загрузки.

Имя	Тип	По умолч.	Описание
wrap	boolean	true	Должна ли карусель сменяться плавно или дискретно.

Методы

Асинхронные методы и переходы

Все методы API **асинхронны** и запускают **переход**. Они возвращаются функции, вызвавшей их, с началом перехода, но **до его конца**. Плюс, вызов метода к компоненту, выполняющему переход, **будет проигнорирован**.

[Смотрите нашу документацию по JavaScript.](#)

```
.carousel(options)
```

Инициализирует объект `object` карусели с установленными параметрами и начинает смену слайдов.

```
$('.carousel').carousel({
  interval: 2000
})
.carousel('cycle')
```

Сменяет слайды карусели слева направо.

```
.carousel('pause')
```

Останавливает смену слайдов.

```
.carousel(number)
```

Прокручивает слайды до определенного момента (основано на 0, схоже с рядами).

Возвращает значение функции-вызову до того, как элемент-«цель» показан (т.е. до того, как возникает событие `slid.bs.carousel`).

```
.carousel('prev')
```

Прокручивает к предыдущему слайду. **Возвращает значение функции-вызову до того, как элемент-«цель» показан** (т.е. до того, как возникает событие `slid.bs.carousel`).

```
.carousel('next')
```

К следующему слайду. **Возвращает значение функции-вызову до того, как элемент-«цель» показан** (т.е. до того, как возникает событие `slid.bs.carousel`).

```
.carousel('dispose')
```

Уничтожает карусель элемента.

События

Карусель в Bootstrap располагает 2-мя событиями для применения функциональности. У обоих событий есть следующие дополнительные свойства:

- `direction`: Направление, в котором движутся слайды ("left" или "right").
- `relatedTarget`: Элемент DOM, который движется на место «пролистанного» слайда.
- `from`: Индекс текущего слайда
- `to`: Индекс следующего слайда

Все события карусели запускаются непосредственно в каруселу (т.е. в `<div class="carousel">`).

Тип события	Описание
<code>slide.bs.carousel</code>	Это событие запускается немедленно, когда вызван метод <code>slide</code> .
<code>slid.bs.carousel</code>	Это событие запускается, когда карусель завершила переходы своих слайдов.

```
$('#myCarousel').on('slide.bs.carousel', function () {  
  // do something..  
})
```

Изображения

Документация и примеры приведения изображений к отзывчивому поведению (так, чтобы их размер не выходил за рамки родительского элемента) и добавления к ним «легковесных» стилей – всё посредством классов.

Отзывчивые изображения

Изображения в Bootstrap делаются «отзывчивыми» с помощью `.img-fluid`. `max-width: 100%`; и `height: auto`; применяемых к изображению, которое т.о. масштабируется по родительскому элементу.



```

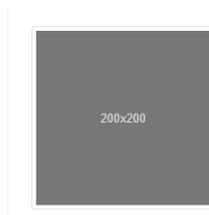
```

[Изображения SVG и IE10](#)

В IE10 SVG-изображения с классом `.img-fluid` выглядят диспропорционально. Для решения этой проблемы добавьте `width: 100% \9;` где необходимо. Это решение затрагивает и размеры изображений других форматов, поэтому Bootstrap не применяет его автоматически.

Мини-версии изображения

В дополнение к нашим [утилитам бордер-радиуса](#) вы можете использовать класс `.img-thumbnail` для придания изображению границы 1px.

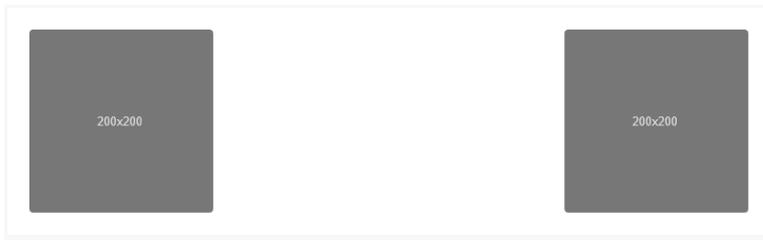


```

```

Выравнивание изображений

Выравнивайте изображения [вспомогательными классами](#) выравнивания по стороне или [классами выравнивания текста](#). Блочные изображения `block` можно центрировать использованием класса утилиты «марджина» [the `.mx-auto`](#).



```
  

```



```

```

[Картинка](#)

Если вы используете элемент `<picture>` для привязки множественных элементов `<source>` к определенному ``, добавьте классы `.img-*` в ``, а не в `<picture>`.

```
<picture>  
  <source srcset="..." type="image/svg+xml">  
    
</picture>
```