Видео недели:

1. [Фреймворк Bootstrap](https://youtu.be/auMuzykfoKo)
2. [Компоненты](https://www.youtube.com/watch?v=tLXeNa_acLA)
3. [Бонус! Верстаем форму логина на Bootstrap](https://youtu.be/CVC5-Qslof8)

**CSS Фреймворки**

**Bootstrap**

[Bootstrap](https://getbootstrap.com/) — это открытый и бесплатный HTML, CSS и JS фреймворк, который используется веб-разработчиками для быстрой вёрстки адаптивных дизайнов сайтов и веб-приложений.

Фреймворк Bootstrap используется по всему миру не только независимыми разработчиками, но иногда и целыми компаниями. На Bootstrap создано очень много различных сайтов, посмотреть на них можно на странице [Bootstrap Expo](https://expo.getbootstrap.com/).

Основная область его применения – это разработка **фронтенда сайтов и интерфейсов админок**. Среди аналогичных систем (Foundation, UIkit, Semantic UI, InK и др.) фреймворк Bootstrap является **самым популярным**.

**Почему Bootstrap так популярен?** Это связано с тем, что он **позволяет верстать сайты в несколько раз быстрее**, чем это можно выполнить на «чистом» CSS и JavaScript. А в нашем мире время – это самый дорогой ресурс. Также его популярность **связана с доступностью**. Она заключается в том, что на нём **даже начинающий разработчик может верстать достаточно качественные макеты**, которые трудно было бы выполнить без глубоких знаний веб-технологий и достаточной практики.

Версия Bootstap 4 сейчас является основной и поддерживает последние, стабильные версии всех основных браузеров и платформ. В операционной системе Windows эта версия поддерживает Internet Explorer 10-11 и Microsoft Edge.

Если же вам нужна поддержка более старых браузеров, например Internet Explorer 8 и 9, то в этом случае следует выбрать третью версию. В старых версиях Internet Explorer (8 и ниже) некоторые компоненты Bootstrap 3 могут отображаться без градиентов, теней и закруглённых углов. Это связано с тем, что эти версии браузеров не поддерживают необходимые для этого CSS3 свойства.

**Подключение Bootstrap к HTML странице**

Фреймворк Bootstrap представляет собой набор CSS и JavaScript файлов. Чтобы его использовать эти файлы необходимо просто подключить к странице. После подключения вам станут доступны **инструменты данного фреймворка: колоночная система (сетка Bootstrap), классы и компоненты**.

Процесс установки фреймворка Bootstrap состоит из подключения следующих файлов к HTML странице:

1. Bootstrap CSS (bootstrap.min.css);
2. <link rel="stylesheet" href="[<https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css>](<https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css>)" integrity="sha384-TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKI2xXr2" crossorigin="anonymous">
3. Последней версии библиотеки jQuery (необходима для работы JS плагинов Bootstrap);
4. Bootstrap JavaScript (bootstrap.min.js).
5. <script src="<https://code.jquery.com/jquery-3.5.1.slim.min.js>" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>
6. <script src="<https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js>" integrity="sha384-ho+j7jyWK8fNQe+A12Hb8AhRq26LrZ/JpcUGGOn+Y7RsweNrtN/tE3MoK7ZeZDyx" crossorigin="anonymous"></script>

**Примечание:** Файлы JavaScript лучше подключать перед закрывающим тегом body (</body>), так как это обеспечит более быструю загрузку и отображение основного контента веб-страницы.

1. Вставьте 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">

**Как это работает?**

Например, чтобы создать кнопку на странице с помощью Bootstrap, достаточно к ссылке или элементу button добавить всего несколько классов.

Создание кнопки в Bootstrap 4:

<!-- Чтобы сделать ссылку в виде кнопки добавим к ней 2 класса:

btn и btn-success -->

<a href="#" class="btn btn-success">Ссылка, оформленная в виде кнопки</a>

<a href="#">Обычная ссылка</a>

<button>Обычная кнопка</button>

**Ну как?** Сколько на это потребовалось времени? Да, всего несколько секунд.

А если, например, мне нужно **создать вкладки**, то для этого нужно будет лишь скопировать готовый HTML фрагмент, и отредактировать его так, чтобы в нём было нужное количество вкладок:

<ul class="nav nav-tabs" id="myTab" role="tablist">

<li class="nav-item">

<!-- атрибут href должен указывать на элемент, который необходимо показать -->

<a class="nav-link active" data-toggle="tab" href="#section-1" role="tab" aria-controls="section-1" aria-selected="true">Секция 1</a>

</li>

<li class="nav-item">

<!-- атрибут href должен указывать на элемент, который необходимо показать -->

<a class="nav-link" data-toggle="tab" href="#section-2" role="tab" aria-controls="section-2"

aria-selected="false">Секция 2</a>

</li>

<li class="nav-item">

<!-- атрибут href должен указывать на элемент, который необходимо показать -->

<a class="nav-link" data-toggle="tab" href="#section-3" role="tab" aria-controls="section-3"

aria-selected="false">Секция 3</a>

</li>

</ul>

<div class="tab-content">

<!-- атрибут id определяет вкладку, которая будет его показывать -->

<div class="tab-pane fade show active" id="section-1" role="tabpanel" aria-labelledby="home-tab">...</div>

<!-- атрибут id определяет вкладку, которая будет его показывать -->

<div class="tab-pane fade" id="section-2" role="tabpanel" aria-labelledby="profile-tab">...</div>

<!-- атрибут id определяет вкладку, которая будет его показывать -->

<div class="tab-pane fade" id="section-3" role="tabpanel" aria-labelledby="contact-tab">...</div>

</div>

**Всё, вкладки готовы.** На это тоже ушли секунды. Правда, потрясающая скорость. В этом сила Bootstrap.

**Из чего состоит Bootstrap?**

Bootstrap состоит из:

* сетки;
* классов для стилизации текста, изображений, таблиц и другого контента;
* компонентов предназначенных для создания кнопок, различных форм на странице, горизонтальных и вертикальных навигационных меню, слайдеров, выпадающих списков, аккордеонов, модальных окон, всплывающих подсказок и других элементов интерфейса;
* классов для решения вспомогательных задач наиболее часто возникающими перед веб-разработчиками (выравнивание текста, скрытие или отображение элемента, задания цвета и фона элементу, задание margin и padding отступов, и др.)

Здесь можно найти решения практически на все случаи жизни!

**С чего начать изучение Bootstrap?**

Первое – это **нужно научиться работать с сеткой**. В Bootstrap она используется как для создания основного каркаса страницы, так и для разметки всего остального. Сетка в Bootstrap 4 построена на CSS Flexbox.

Следующее – это **изучить компоненты и классы Bootstrap**.

После изучение этих базовых вещей уже можно будет переходить к вёрстке сайта или веб-приложения.

**Сетка Bootstrap**

Сетку Bootstrap 4 можно использовать отдельно (без других компонентов). Этот вариант применения фреймворка тоже используется веб-разработчиками, т.к. не всегда готовые Bootstrap элементы, такие как кнопки, навигационные панели, таблицы и прочие компоненты можно использовать в рамках того или иного проекта.

Зачастую это касается проектов с уникальным дизайном. В этом случае нужные элементы проще написать с нуля самостоятельно, вместо того чтобы переписывать готовые компоненты Bootstrap.

**Сетка Bootstrap 4 – это основная часть фреймворка. Она предназначена для создания адаптивных шаблонов.**

Вот самый простой пример применения сетки:

<div class="container">

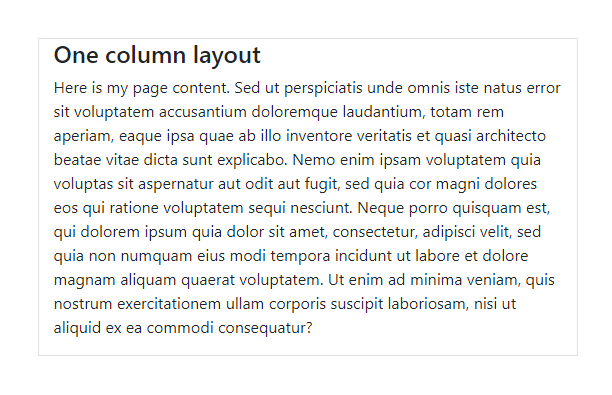
<div class="row">

<div class="col">I'm your content inside the grid!</div>

</div>

</div>

Этот код выдаст нам одну большую колонку на всю ширину экрана.



А вот уже две колонки:

<div class="container">

<div class="row">

<div class="col">Left column</div>

<div class="col">Right column</div>

</div>

</div>



И вот на три колонки:

<div class="container">

<div class="row">

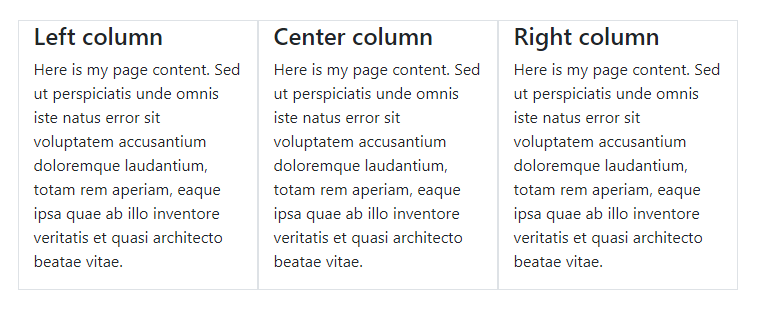
<div class="col">Left column</div>

<div class="col">Center column</div>

<div class="col">Right column</div>

</div>

</div>



Вы наверное заметили, что здесь используется блок .container, чтобы обернуть .row. Контейнер - это вообще ключевой элемент сетки в Bootstrap.

<div class="container">

<div class="row">

<div class="col">I'm content inside the grid!</div>

</div>

</div>

Контейнер может использоваться для хранения любых элементов и самого контента. Он используется не только для строк и колонок сетки. Для примера, вот идеальная и правильная разметка Bootstrap:

<div class="container">

<h2>My Heading</h2>

<div class="row">

<div class="col">I'm content inside the grid!</div>

</div>

</div>

С первого взгляда, контейнер может показаться пустяковым и не сильно уж нужным, но он очень важен для контроля ширины шаблона. Контейнер также используется для равномерного выравнивания граней шаблона внутри ширины окна браузера.

У Bootstrap 4 есть 2 типа контейнера. В примерах выше использован .container, но также есть и полноэкранный .container-fluid. Вы можете использовать любой из них:

1. Контейнер с фиксированной шириной, для центровки контейнера по середине шаблона.

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

1. Контейнер с шириной во весь экран.

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

.container масштабируется адаптивно по ширине экрана, так что в конце концов он может стать шириной на весь экран, как и .container-fluid, но на маленьких устройствах.

Помните, что контейнер может использоваться для любого контента, а не только строк и колонок сетки. Но! Если вы используете последние два элемента, то строки сетки должны быть размещены внутри контейнера.

**Важные правила сетки!**

Колонки col должны быть прямыми потомками row

row используются только для того, чтобы включать в себя колонки и ни для чего больше

row должны быть помещены внутри контейнера

Ждите косяка, если вы не будете соблюдать эти три простые правила.

**Строки (rows) и Колонки (columns)**

Когда вы думаете о строке, то вероятно вы думаете о горизонтальной линии, что вполне нормально, НО лучше думать о строке, как о **родителе для колонок** или о **группе для колонок**.

Это потому, что колонки внутри .row не всегда располагаются горизонтально вдоль вьюпорта. Иногда нам надо, чтобы колонки в шаблоне были горизонтальны, а иногда нам надо, чтобы они располагались вертикально. Концепция горизонтального vs. вертикального шаблона является сущностью адаптивного дизайна. Единственным предназначением “строки”, является содержание одной или более “колонки”.

Не вставляйте контент прямо в “строку”!

**Так делать нельзя:**

<div class="row">

О нет, это ошибка!

</div><div class="row">

<p>Это ужасно, не надо так!!</p>

</div><div class="row">

<h2>Даже заголовки нельзя! Нет, нет и нет!</h2>

</div>

**“колонки” и только колонки, размещаются внутри “строк”.**

А контент размещается уже внутри “колонок”.

<div class="row">

<div class="col">Да! Надо так!</div>

</div>

Так же очень важно упомянуть, что .row имеет display: flex. А как потомок в Flexbox, “колонка” в каждой строке одной и той же высоты.

Благодаря Flexbox, горизонтальное и вертикальное выравнивание легко делается с использованием рабочих [классов Bootstrap 4 — flex и auto-margin](http://getbootstrap.com/docs/4.1/utilities/flex/).

А теперь настало время углубиться в строки и колонки и то, как они на самом деле взаимодействуют. Есть разные типы колонок и разные способы использования их в шаблоне. Они как магия.



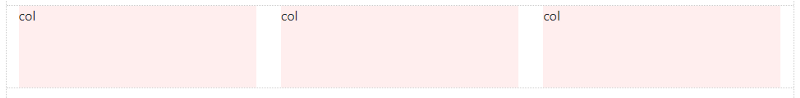
**Счастье в колонках!**

* Создают горизонтальные разделители по вьюпорту (ширине экрана).
* Могут иметь разную ширину.
* Раскладку *горизонтально* слева направо, *вертикально* сверху вниз.
* Могут изменять позицию (порядок) относительно родственных элементов в той же строке.
* Имеют ту же высоту, что и другие родственные элементы в той же строке.
* Могут “расти” или “урезаться” по ширине.
* Могут автоматически врапиться или вертикально “слепляться” при необходимости или при нужной ширине экрана.
* Могут содержать больше строк и колонок при вложении.

**Всё что нужно знать о колонках Bootstrap**

Снова не забываем, что колонки должны быть прямыми потомками строк.

Итак, колонки создают горизонтальные деления по вьюпорту. Пространство между колонками называется “gutter”.



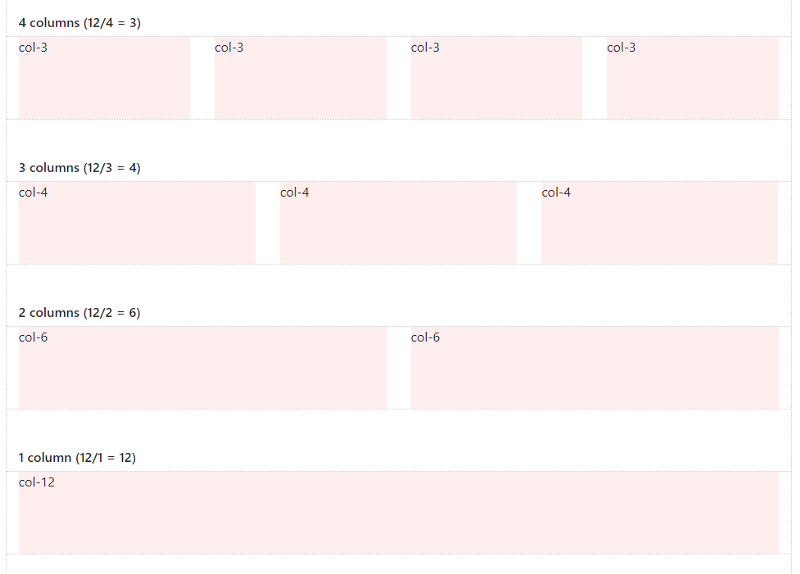
Классическая сетка Bootstrap имеет 12 колонок:



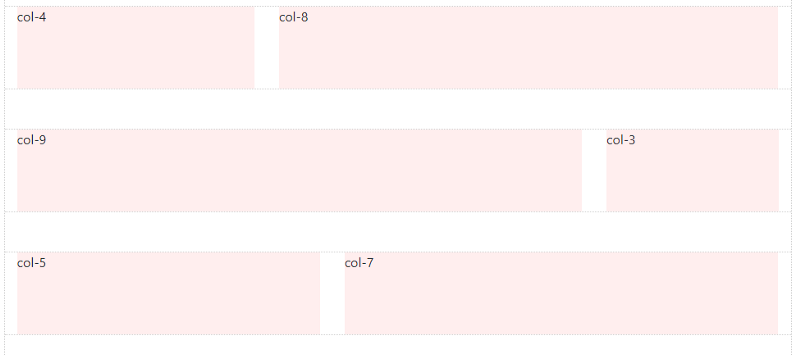
Таким образом, колонки могут быть равномерно разделены на 12 частей. Вот пример, 6 колонок (12/6=2) :



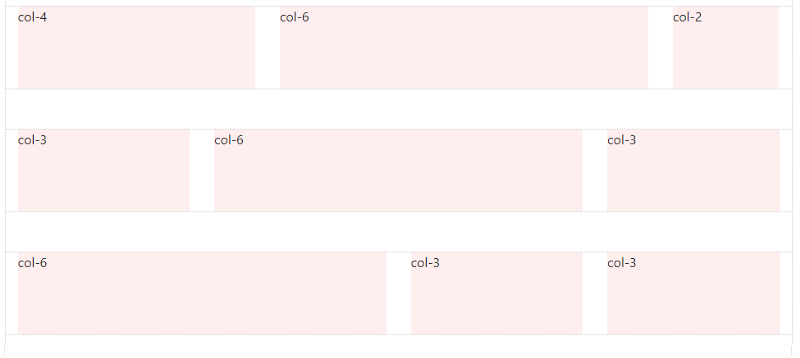
И дальше уже дело простой математики:



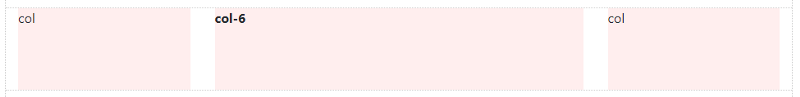
Колонки могут быть разделены с использованием любой части из 12 элементов. И это нормально — использовать меньше 12. И также, это нормально — использовать больше 12-ти, но это я покажу вам дальше.



Учитывая такую гибкость, возможности шаблона кажутся бесконечными.



Как вы видели выше, колонки могут быть разной ширины:



Но естественно, можно и нужно делать так, чтобы ширина колонки могла изменяться в зависимости от ширины экрана.



Bootstrap превосходно работает с адаптивной версткой.

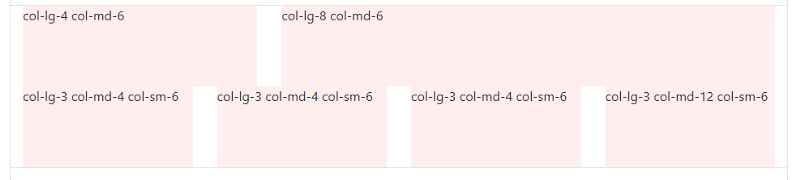
Но все по порядку, давайте сначала разберемся, как можно использовать больше 12-ти колонок в строке? Ведь сетка не всегда может быть из 12 элементов. Спасибо flexbox, у Bootstrap 4 есть новые auto-layout колонки. Такие безразмерные колонки дают вам больше гибкости при разработке шаблонов.

Колонки в одной строке располагаются вдоль горизонтально, а затем встают вертикально вниз. Это *вертикальное сталкивание* или “обертывание”, происходит когда элементы колонок в одном ряду превышают количество двенадцати. Этот [процесс известен как “враппинг колонок”](http://getbootstrap.com/docs/4.1/layout/grid/#column-wrapping)

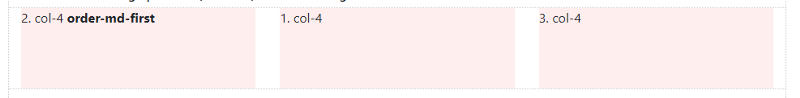
Колонки в одной строке переходят на следующую строку, через каждые 12 элементов (4+4+4 =12 на примере):



Ширина колонки и “обертывание” может контролироваться, используя разные ряды адаптивной сетки (чуть дальше мы поговорим о контрольных точках в адаптивной верстке).



Колонки могут менять позицию относительно потомков в том же ряду:



Колонки могут содержать других потомков row & col. Это называется вложением:



Колонки могут “расти” и “урезаться” по ширине. Это auto-layout колонки:



**Как делать адаптивный дизайн с Bootstrap**

**В Bootstrap изменение вида макета реализовано через медиа-запросы.** Каждый медиа-запрос в Bootstrap строится на основании **минимальной ширины viewport браузера**. В Bootstrap ключевое значение ширины viewport в медиа-запросе называется **breakpoint** (контрольной точкой, или более просто - классом устройства).

На схеме приведены основные контрольные точки, которые Bootstrap 4 имеет по умолчанию:

Это означает что до 576px макет сайта может отображаться одним образом, от 576px до 768px – другим образом и т.д. Таким образом, можно создать макет, который на каждом из этих участков может выглядеть по-разному.

Контрольные точки (далее - брейкпойнты) имеют обозначение. Некоторые из них вы, возможно, заметили в предыдущих примерах ( col-lg-4, col-md).

Адаптивные брейкпоинты, основаны на ширине экрана:

(xs) — ширина экрана < 576px. Это стандарт.

sm — ширина экрана ≥ 576px

md — ширина экрана ≥ 768px

lg — ширина экрана ≥ 992px

xl — ширина экрана ≥ 1200px

(xs) стоит в скобках, так как он используется по умолчанию. xs как запись появился в Bootstrap 3, но больше не используется в Bootstrap 4. Так что вместо col-xs-6 пишем просто  col-6.

Для примера: вот 2 колонки, каждая шириной 50%:

<div class="container">

<div class="row">

<div class="col-sm-6">Column 1</div>

<div class="col-sm-6">Column 2</div>

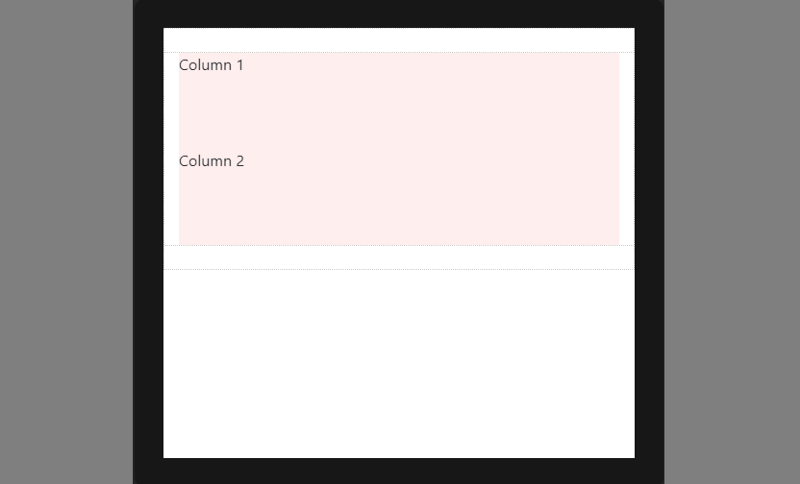
</div>

</div>

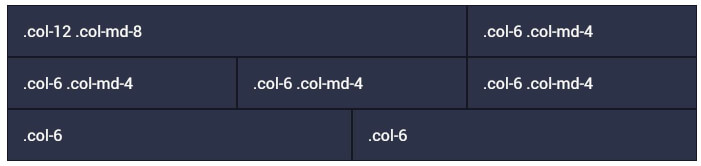
сol-sm-6 означает использование 6 колонок из 12, то есть 50% ширины на типичных маленьких размерах экранов. Которые больше или равны 576px:



На экранах меньше, чем 576px, 2 колонки станут шириной 100% и встанут вертикально:



С помощью Bootstrap вы можете создавать любые комбинации колонок при создании сетки. Для каждой колонки можно задать любое поведение на различных разрешениях с помощью адаптивных классов.



<!-- На мобильных устройствах: Первая колонка - на полную ширину, вторая на половину -->

<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>

<!-- На десктопе: 3 колонки по 4 в каждой. На мобильных: все колонки вполовину контейнера -->

<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 занимают 6 из 12 колонок на любом разрешении -->

<div class="row">

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

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

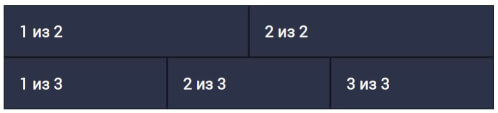
</div>

**Давайте посмотрим на auto-layout сетку**

Auto-layout колонки в Bootstrap 4 также работают адаптивно. Из-за их простоты часто даже предпочтительнее использовать их, а не классические элементы 12 колонок. Auto-layout колонки идеальны для любых сценариев шаблонов, где необходима одинаковая ширина колонок. Но не забудьте, что 12-ти колоночные варианты могут быть смешаны при необходимости.

С помощью нового универсального класса col можно указать до 12 колонок в ряду (родитель row), ширина которых будет автоматически вычислена в зависимости от количества элементов и будет равна.

Например:



col остаются горизонтальными на всей широте и не встают вертикально, так как xs прерывание дефолтно:

<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>

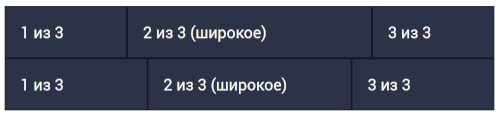
<div class="col">2 из 3</div>

<div class="col">3 из 3</div>

</div>

</div>

Вы также можете явно установить ширину одной колонки, а остальные оставить автоматическими.



<div class="row">

<div class="col">1 из 3</div>

<div class="col-6">2 из 3 (широкое)</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 (широкое)</div>

<div class="col">3 из 3</div>

</div>

В данном примере второй элемент в третьем ряду имеет класс col-6 и второй элемент во втором ряду имеет класс col-5, которые занимают соответствующее количество колонок на всех разрешениях экрана. Ширина остальных колонок является отзывчивой и вычисляется автоматически, занимая всё оставшееся пространство.

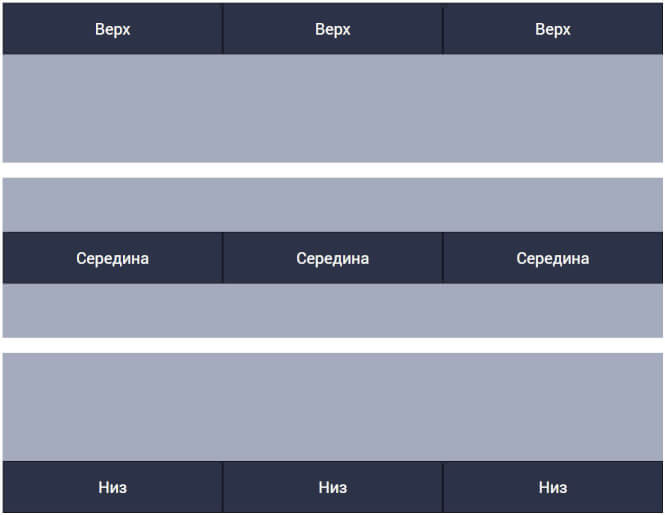
**Ключевые моменты адаптивного дизайна используемые в сетке Bootstrap 4:**

* Колонки встанут вертикально и станут шириной во весь экран на устройствах с маленьким разрешением, если вы не используете col-\* класс в HTML разметке. Используйте col-\* для предотвращения такого вертикального выстраивания.
* Классы сеток поменьше, также применяются на больших экранах, пока не перепишутся конкретно под ширину большего экрана. Следовательно, <div class="col-md-6"></div> в сущности тоже самое, что и <div class="col-md-6 col-lg-6"></div>. Следовательно, вам надо использовать класс для самых маленьких разрешений, которые вам нужно поддерживать.
* Строки (row) — имеют display: flex и следовательно колонки имеют равную высоту в одном и том же ряду. Используйте auto-margin или Flexbox align-item и justify-content для горизонтального или вертикального выравнивания.

**Выравнивания в Bootstrap**

Bootstrap 4 основан на «флексах» и дарит нам все его возможности, которые доступны в простых готовых классах, в том числе вертикальное и горизонтальное выравнивание.

**Вертикальное выравнивание**



<div class="row align-items-start">

<div class="col">Верх</div>

<div class="col">Верх</div>

<div class="col">Верх</div>

</div>

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

<div class="col">Середина</div>

<div class="col">Середина</div>

<div class="col">Середина</div>

</div>

<div class="row align-items-end">

<div class="col">Низ</div>

<div class="col">Низ</div>

<div class="col">Низ</div>

</div>

Помимо управления выравниванием через родительский row, можно выравнивать колонки, задавая им соответствующие классы:



<div class="row">

<div class="col align-self-start">Верх</div>

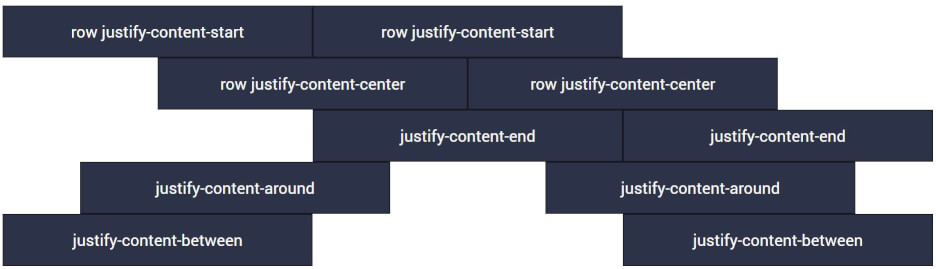
<div class="col align-self-center">Середина</div>

<div class="col align-self-end">Низ</div>

</div>

**Горизонтальное выравнивание**

Кроме того, Bootstrap 4 имеет в своём арсенале инструменты для горизонтального выравнивания колонок при помощи префикса justify-content- у row



<div class="row justify-content-start">

<div class="col-4">row justify-content-start</div>

<div class="col-4">row justify-content-start</div>

</div>

<div class="row justify-content-center">

<div class="col-4">row justify-content-center</div>

<div class="col-4">row justify-content-center</div>

</div>

<div class="row justify-content-end">

<div class="col-4">justify-content-end</div>

<div class="col-4">justify-content-end</div>

</div>

<div class="row justify-content-around">

<div class="col-4">justify-content-around</div>

<div class="col-4">justify-content-around</div>

</div>

<div class="row justify-content-between">

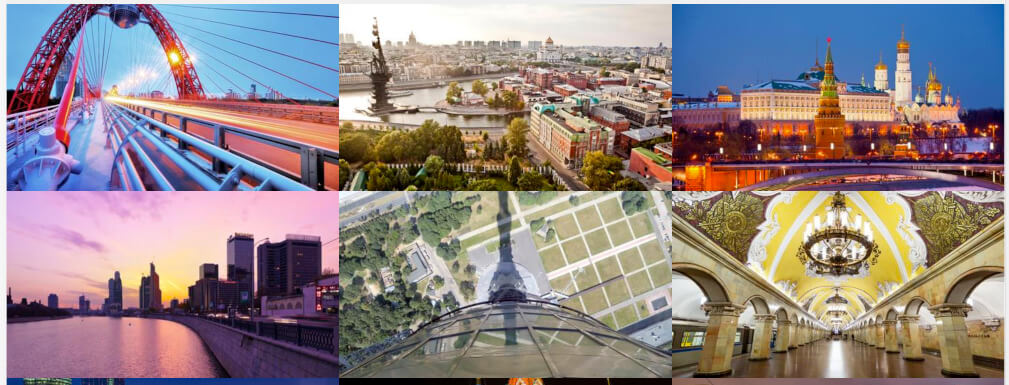
<div class="col-4">justify-content-between</div>

<div class="col-4">justify-content-between</div>

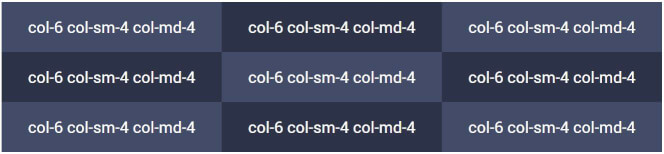
</div>

**Удаление полей между колонками**

Очень часто встречаются ситуации, когда необходимо убрать поля между колонками. Например, если вы создаете галерею и элементы должны быть расположены вплотную, например так:



Для этого достаточно у элемента row задать дополнительный класс .no-gutters.



<div class="row no-gutters">

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

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

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

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

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

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

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

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

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

</div>

**Компоненты Bootstrap 4**

Десятки полезных компонентов уже встроены в Bootstrap - такие как навигация, кнопки, сообщения, инфоблоки и многое другое.

* **Уведомления**
* **Кнопки**
* **Карточки**
* **Выпадающие списки**
* **Навигация**

С полным списком компонент и их настроек можно ознакомиться по ссылке: <https://bootstrap-4.ru/docs/4.0/components/alerts/>

**Преимущества и недостатки фреймворка Bootstrap**

Преимущества, которые даёт фреймворк Bootstrap при разработке на его основе frontend части сайтов и интерфейсов админок:

* **высокая скорость** создания качественной адаптивной вёрстки даже начинающими веб-разработчиками (достигается это благодаря использованию готовых компонентов, созданных профессионалами);
* **кроссбраузерность и кроссплатформеннось** (корректное отображение и работа сайта во всех поддерживаемых этим фреймворком браузерах и операционных системах);
* **наличие большого количество готовых хорошо продуманных компонентов**, протестированных огромным сообществом веб-разработчиков на различных устройствах;
* **возможность настройки под свой проект**, достигается это посредством изменения SCSS переменных и использования Bootstrap миксинов (можно изменить количество колонок, цвета, радиуса скруглений углов элементов, отступы между колонками и многое другое);
* **низкий порог вхождения**; для работы с фреймворком не обязательно иметь глубокие знания по HTML, CSS, JavaScript и jQuery (достаточно знать только основы вышеперечисленных технологий);
* **наличие хорошо продуманного дизайна компонентов и согласованности** (в Bootstrap все компоненты выполнены в едином стиле);
* **наличие огромного сообщества, большого количества статей, рецептов и видеоматериалов**; всё это при желании поможет не только хорошо разобраться в фреймворке, но и найти ответы практически на любые вопросы.

Фреймворк Bootstrap – это **проект с открытым исходным кодом**, доступным на Github. Он имеет лицензию MIT. Это означает, что его можно использовать бесплатно как в личных, так и в коммерческих проектах.

На Bootstrap, конечно, **верстают далеко не всё**. Его не используют, например, для:

* создания фронтенда проектов с уникальным дизайном;
* разработки проектов, в которых заказчик готов платить за проект на «чистом» CSS и JavaScript (в большинстве случаях такая разработка осуществляется в команде, в которой каждый её участник выполняет какой-то свой определенный набор функций);
* вёрстки личных проектов, если у вас есть достаточно количество времени и ваш уровень знания технологий HTML, CSS и JavaScript достаточный, чтобы это осуществить.

Bootstrap, так и большинство подобных фреймворков, **имеет недостатки**. Среди них:

* более большой размер конечных CSS и JavaScript файлов, чем если их написать конкретно под этот проект на чистом CSS и JavaScript (это связано с тем стили и код JavaScript содержат много всего и на большинстве сайтах нужна только та или иная часть из этого);
* использование Bootstrap для вёрстки сайтов, в которых потребуется значительное переписывание его CSS и JavaScript кода (простой настройкой Bootstrap переменных тут не обойтись).

Первый недостаток не такой существенный, тем более что его можно немного «сгладить» выполнив самостоятельную сборку проекта из исходных кодов и включить в неё только те компоненты, которые нужны. В этом случае код этого фреймворка будет значительно меньше.

**Домашнее задание на эту неделю:**

**Вопросы 💎**

1. Какие основные преимущества Bootstrap?
2. Как сделать желтую кнопку с помощью Bootstrap?
3. Каким кодом можно сделать такую сетку?
4. Каким компонентом Bootstrap можно задать такой элемент?
5. Как убрать поля между колонками?
6. Как сделать навигацию на Bootstrap?
7. Что такое колонка **auto-layout ?**
8. Как сделать центрирование содержимого по горизонтали? А по вертикали?
9. Как задать адаптивность для колонки шириной в 50%, чтобы при ширине экрана меньше 768px она становилась шириной на весь экран?
10. Что произойдет, если колонок станет больше 12?
11. Как задать иконку инстаграм через font awasome?
12. Чем отличается container от container-fluid?

**Практическое задание 💎**

1. Сделать макет с применением Bootstrap и прислать ссылку на Github
2. Сверстать страничку <https://www.figma.com/file/Furcf4cCqnxLTrNDjykGtg/VR_Colors> по методологии БЭМ и прислать ссылку на Github

**Ссылка на свою страничку из википедии 💎**

ОТВЕТЫ

1. **Уменьшение количества времени, затрачиваемого на разработку, адаптивность, кроссбраузерность, легкость в использовании и быстрота в освоении, понятный код, единство стилей**
2. <button type="button" class="btn btn-warning">Предупреждение</button>
3. Ответ по ссылке на гитхаб
4. progress-bar
5. <div class="row no-gutters">
6. class="nav"
7. auto-layout - это новые колонки(безразмерные), дающие больше гибкости при разработке шаблонов
8. по горизонтали justify-content-center

по вертикали align-items-center

1. col-md-12
2. *вертикальное сталкивание* или “обертывание”, происходит когда элементы колонок в одном ряду превышают количество двенадцати. Этот [процесс известен как “враппинг колонок”](http://getbootstrap.com/docs/4.1/layout/grid/#column-wrapping)
3. найти на сайте нужную иконку , скопировать код и вставить в свой документ

пример: <i class="fab fa-instagram"></i>

1. container-fluid занимает всю доступную ширину страницы, а container бывает с фиксированной шириной, для центровки контейнера по середине шаблона.