**Теория 1. Введение и подключение сетки Bootstrap**

**Что такое Bootstrap?**

**Bootstrap** — свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML и CSS-шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript расширения.

**Зачем использовать Bootstrap?**

* **Мобильный подход** – Bootstrap 3, структура состоит из мобильных стилей на протяжении всей библиотеки вместо них не в отдельных файлах.
* **Поддержка браузеров** – Она поддерживается всеми популярными браузерами.
* **Легко для того чтобы начать** – С помощью всего лишь знание HTML и CSS каждый может начать с Bootstrap. Кроме того, официальный сайт Bootstrap имеет хорошую документацию.
* **Адаптивный дизайн** – отзывчивый CSS Bootstrap приспосабливается к десктопу, планшету и мобильным телефонам. Подробнее об адаптивном дизайне в главе адаптивный дизайн в Bootstrap.
* Обеспечивает чистое и единое решение для построения интерфейса для разработчиков.
* Он содержит красивые и функциональные встроенные компоненты, которые легко настроить.
* И лучше всего это, то что он с открытым исходным кодом.

**Из чего состоит Bootstrap?**

* инструментов для создания макета (обёрточных контейнеров, мощной системы сеток, гибких медиа-объектов, адаптивных утилитных классов);
* классов для стилизации базового контента: текста, изображений, кода, таблиц и figure;
* готовых компонентов: кнопок, форм, горизонтальных и вертикальных навигационных панелей, слайдеров, выпадающих списков, аккордеонов, модальных окон, всплывающих подсказок и др.;
* утилитных классов для решения традиционных задач наиболее часто возникающими перед веб-разработчиками: выравнивание текста, отображение и скрытие элементов, задания цвета, фона, margin и padding отступов, и т.д.

**Преимущества и недостатки фреймворка Bootstrap**

* **высокая скорость** создания качественной адаптивной вёрстки даже начинающими веб-разработчиками (достигается это благодаря использованию готовых классов и 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 миксинов (можно изменить количество колонок, цвета, радиус скруглений, отступы между колонками и т.д.);
* **низкий порог вхождения**; для работы с фреймворком не обязательно иметь «глубокие» знания по HTML, CSS, JavaScript и jQuery (достаточно знать только основы этих технологий);
* **однородность дизайна и его согласованность между различными компонентами** (в Bootstrap все компоненты выполнены в едином стиле);
* **наличие огромных сообществ и обучающих материалов**; при желании это поможет не только хорошо разобраться в фреймворке, но и найти ответы практически на любые вопросы.

**Какую версию Bootstrap выбрать**

На текущий момент есть 3 версии, которые можно использовать в своих проектах:

* v3 - это 3.4.1
* v4 - это 4.6.0
* v5 - это 5.3.0

Bootstrap 5 рекомендуется для проектов, которые будут использоваться только в современных браузерах (поддержка IE и других браузеров не нужна). В других случаях - Bootstrap 4.

**Назначение Bootstrap сетки**

**Сетка Bootstrap5 – это основная часть фреймворка. Она предназначена для создания адаптивных шаблонов**.

Сетка Bootstrap 5основывается на CSS Flexbox и медиа-запросах (CSS mediaqueries). Чтобы понять, как она работает более точно, рекомендуется сначала изучить эти CSS технологии.

Следующий шаг – это разобраться с тем, что такое адаптивный макет сайта.

**Адаптивный макет** – это такой макет, вид которого может изменяться в зависимости от того, какую ширину viewport имеет браузер. Это означает, что при одних значениях ширины viewport адаптивный макет может выглядеть одним образом, а при иных – совершенно по-другому.

**В Bootstrap изменения 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 рисунке 1 приведены основные контрольные точки, которые Bootstrap5 имеет по умолчанию:

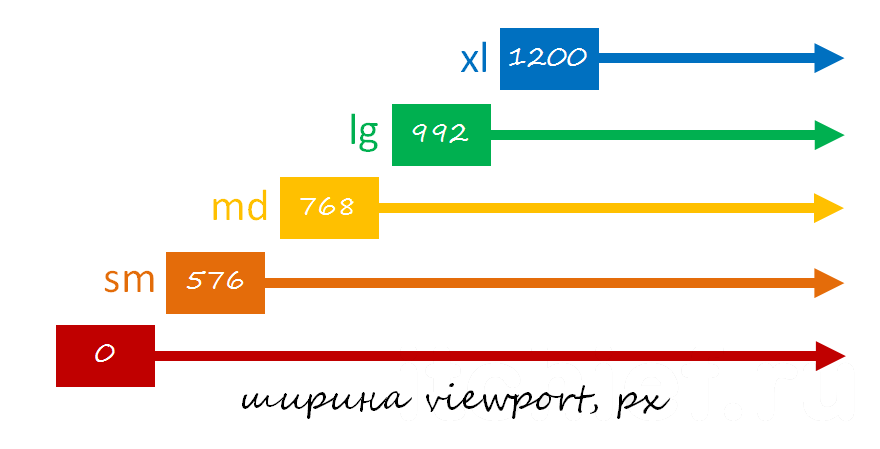


Рисунок 1 — Контрольные точки Bootstrap сетки

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 имеют обозначение**. Первый breakpoint **не имеет обозначения** (xs), второй называется – sm, третий – md, четвёртый – lg и пятый – xl (рисунок 2). Эти обозначения необходимо запомнить, т.к. они используются в классах, которые мы будем добавлять к элементам. **Эти обозначения в имени класса будут указывать на то с какой ширины viewport стили, определённые в нём, будут применяться к элементу**.

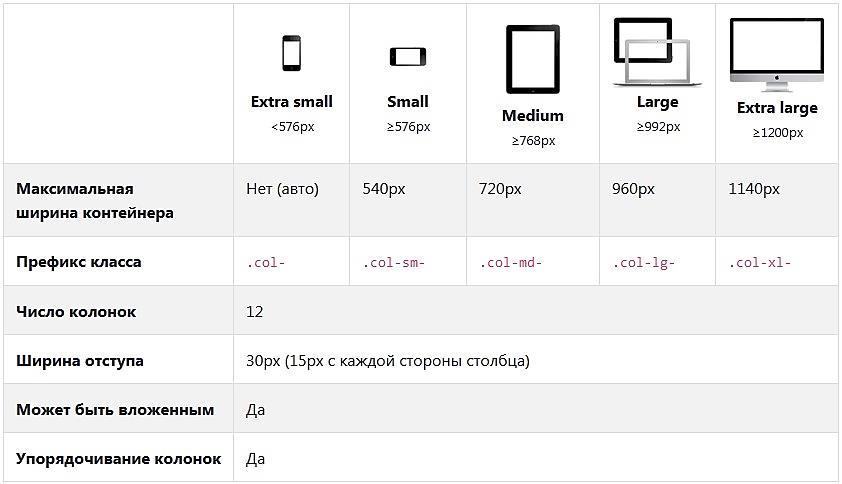


Рисунок 2 — Система сеток

При этом,**контрольные точки задают только минимальную ширину**. Т.е., например, если вы определили макет, используя в нём классы без обозначения контрольной точки (xs) и с использованием md, то он на sm будет иметь такую же структуру как на xs, а на lg и xl - как на md.

**Из чего состоит сетка?**

Сетка Bootstrap 5состоит из:

* **обёрточных контейнеров** (элементов с классом container и container-fluid);
* **рядов** (элементов с классом row);
* **адаптивных блоков** (элементов, имеющих один или несколько классов, равных col или начинающихся с col).

Все эти части сетки – это обычные HTML элементы, к которым просто добавлены определённые классы.

Обёрточный контейнер (.container и .container-fluid) — это элемент сетки Bootstrap5, с которого обычно **начинается создание макета страницы или её части**. Другими словами, это базовый элемент, в котором необходимо размещать все другие элементы сетки (ряды и адаптивные блоки). **Его основная цель — это установить шаблону ширину и выровнять его по центру страницы** (рисунок 3).



Рисунок 3 — Обёрточный контейнер

В Bootstrap5 имеются два вида обёрточных контейнеров: **адаптивно-фиксированный и резиновый**.

**Адаптивно-фиксированный контейнер** предназначен для создания контейнера с постоянной шириной, которая будет оставаться постоянной только в пределах действия определённой контрольной точки. Т.е., например, на контрольной точке sm до действия контрольной точки md, он будет иметь одну фиксированную ширину, а на md до действия lg – другую фиксированную ширину. Единственная контрольная точка, на которой данный контейнер не будет иметь фиксированную ширину это breakpoint **без обозначения**. Здесь контейнер будет занимать **100% ширину** (рисунок 4).

|  |  |
| --- | --- |
|  | <divclass="container">...</div> |
|  |  |

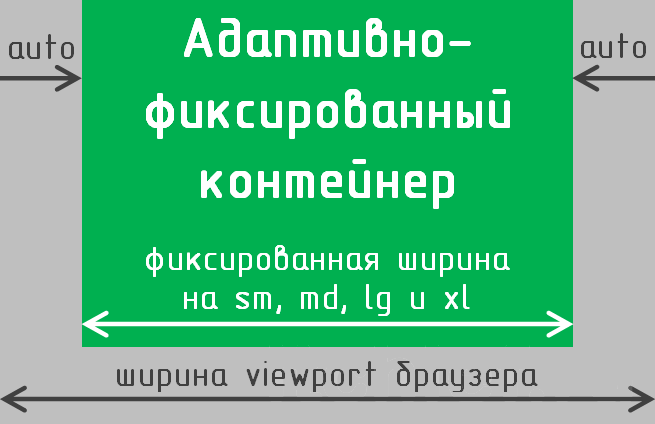


Рисунок 4 — Ширина .container

Адаптивно-фиксированный контейнер будет иметь:

* 100% ширину при ширине viewport до 576px;
* 540px при ширине viewport от 576 до 768px;
* 720px при ширине viewport от 768 до 992px и т.д

В горизонтальном направлении контейнер располагается по центру. Осуществляется это посредством установки ему CSS свойств margin-left: auto и margin-right: auto в файле «bootstrap.css».

Второй вид контейнера – это **резиновый**. Он применяется тогда, когда вам необходимо создать полностью гибкий макет целой страницы или её части. **Данный контейнер имеет на любых контрольных точках 100% ширину**. (рисунок 5)

|  |  |
| --- | --- |
|  | <divclass="container-fluid">...</div> |

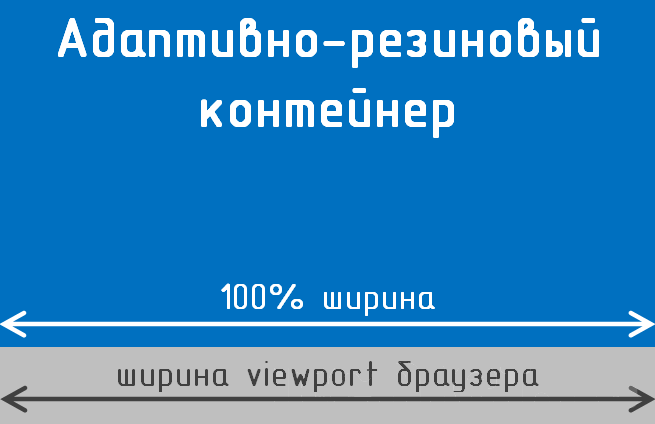


Рисунок5 — Ширина .container-fluid

Кроме этого, контейнеры (.container и .container-fluid) имеют ещё внутренние отступы слева и справа по 15px. Установка внутренних отступов контейнеров задаётсяв «bootstrap.css» посредством CSS-свойств **padding-left: 15px** и **padding-right: 15px**.

При верстке макета не следует одни обёрточные контейнеры помещать внутрь других.

**Ряды (.row)**

**Ряд** – это элемент бутстрап сетки, который выступает в роли непосредственного контейнера для адаптивных блоков (рисунок 6).

Это означает что ряд – это **просто родительский элемент для адаптивных блоков**. Т.е. адаптивные блоки всегда должны быть размещены непосредственно в нём.

С точки зрения CSS Flexbox, ряд – это flex-контейнер, а адаптивные блоки – это flex-элементы.

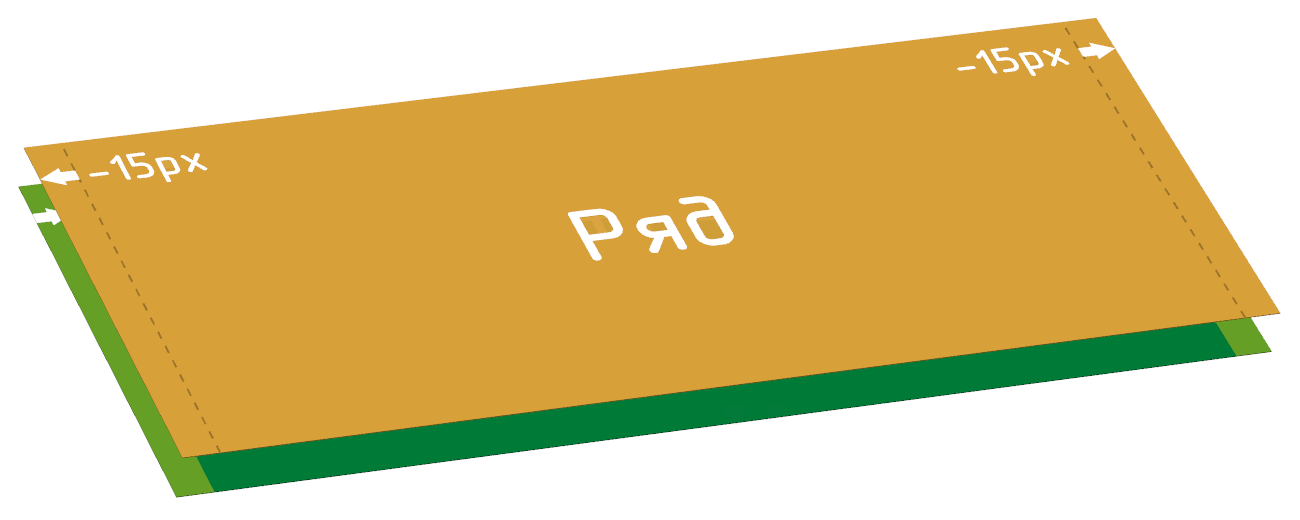


Рисунок 6 — .row

HTML-разметка ряда:

|  |  |
| --- | --- |
|  | <div class="row">...</div> |

В отличие от Bootstrap 3, в которой ряд выступал только для компенсации левых и правых внутренних padding полей обёрточных контейнеров или адаптивных блоков, в Bootstrap 5**он играет очень важную роль**. Это связано с тем, что данная сетка построена на CSS Flexbox. В этой сетки он выступает в роли **flex-контейнера** для **flex-элементов** (адаптивных блоков). Т.е. если вы будете использовать адаптивные блоки вне ряда, они работать не будут. **В Bootstrap 5адаптивные блоки должны обязательно** находиться в блоке с классом row.

Компенсация внутренних padding полей осуществляется так же как и в Bootstrap 3, за счёт отрицательных левых и правых внешних отступов, равных 15px (margin-left:-15px и margin-right:-15px).

**Адаптивные блоки (.col)**

**Адаптивные блоки – это основные строительные элементы сетки**. Именно от них будет зависеть, как макет веб-страницы будет отображаться на разных контрольных точках (без обозначения, sm, md, lg и xl) (рисунок 7).

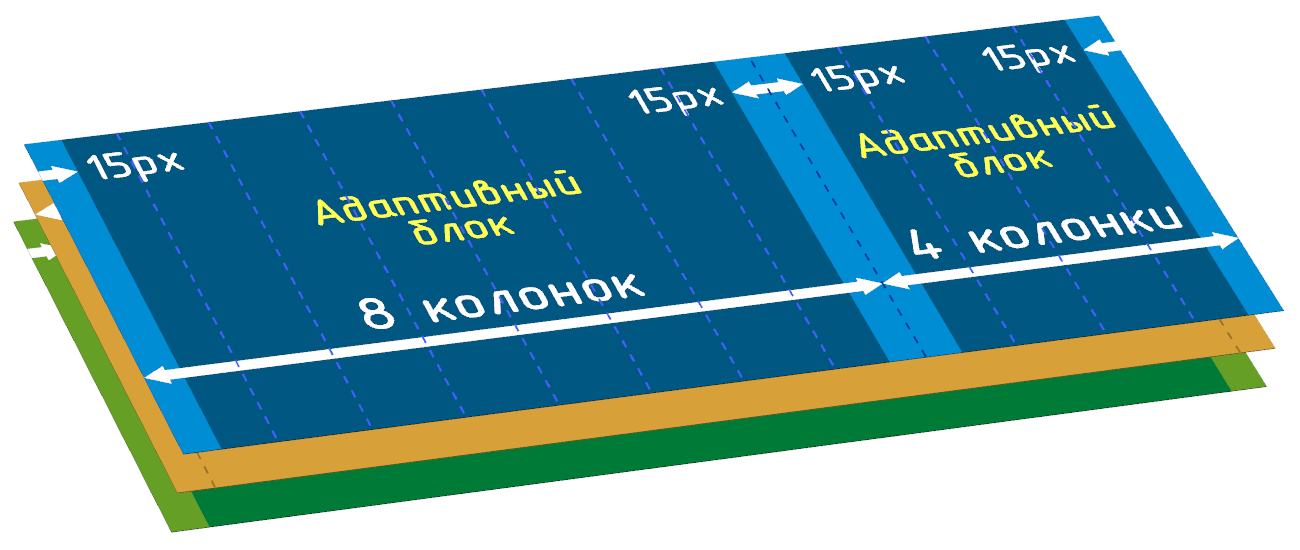


Рисунок 7 — .col

Если простыми словами, то **адаптивные блоки – это блоки, ширина которых (в процентном отношении от родителя) на разных breakpoint может быть различной**. Т.е., например, адаптивный блок на sm может иметь ширину, равную 50% от родительской, а на md – 25%.

Создание адаптивного блока осуществляется очень просто посредством добавления одного или нескольких классов col-?-? к необходимому HTML элементу, расположенному непосредственно в ряду.

В классе col-?-? **вместо первого знака вопроса указывается название контрольной точки** (без обозначения, sm, md, lg или xl). **Вместо второго знака вопроса указывается ширина адаптивного блока**, которую он должен иметь на указанной контрольной точке. Ширина адаптивного блока задаётся в относительной форме **по умолчанию с помощью числа от 1 до 12** (колонок Bootstrap).

Данное число определяет, какую часть ширины будет занимать адаптивный блок начиная с указанной контрольной точки от ширины родительского блока, т.е. ряда. При этом ширина ряда в числовом выражении (колонках Bootstrap) по умолчанию равна 12.

Например, блок с классом col-md-4 начиная с контрольной точки md будет занимать 4/12 ширины ряда, т.е. 33,3% (т.е. 4/12\*100% = 33.3%).

Адаптивные блоки как обёрточные контейнеры имеют внутренние отступы слева и справа по 15px. Данные отступы адаптивным блокам в фреймворкеBootstrap 5устанавливаются с помощью CSS свойств padding-left: 15px и padding-right: 15px.

Размещать адаптивные блоки необходимо непосредственно в ряд. Т.е. у любого адаптивного блока в качестве родителя должен быть обязательно элемент .row.

Например, рассмотрим, какую ширину будет иметь следующий адаптивный блок:

|  |  |
| --- | --- |
|  | <div class="row">      <!--Адаптивный блок Bootstrap-->      <div class="col-12 col-sm-9 col-md-7 col-lg-5 col-xl-3">        ...      </div>  </div> |

Список, объясняющий какую ширину адаптивный блок, будет иметь на разных контрольных точках:

* класс col-12 будет задавать ширину, которую блок будет иметь по умолчанию; она будет равна 12 колонкам Bootstrap (т.е. 12/12\*100%=100% от ширины ряда); эту ширину этот блок будет иметь начиная с xs;
* класс col-sm-9 переопределит ширину блока начиная с действия контрольной точки sm; начиная с sm этот блок будет иметь ширину, равную 9 колонкам Bootstrap (т.е. 9/12\*100%=75% от ширины ряда);
* класс col-md-7 установит ширину блока, равную 7 колонкам Bootstrap (т.е. 7/12\*100%=58,3% от ширины ряда), начиная с md;
* класс col-lg-5 установит начиная с действия контрольной точки lg ширину блоку, равную 5 колонкам Bootstrap (т.е. 5/12\*100%=41,6% от ширины ряда);
* класс col-xl-3 задаст блоку, на экранах с шириной viewport больше xl, ширину равную 3 колонкам Bootstrap (т.е. 3/12\*100%=25% от ширины ряда).

При указании ширины адаптивному блоку мы указываем класс, содержащий контрольную точку, начиная с которой данная ширина будет действовать. Эту ширину данный блок будет иметь до тех пор, пока она будет переопределена с помощью другого класса, действие которого начинается с более большой ширины viewport.

**Адаптивные блоки без указания количества колонок**

В сетке Bootstrap 5имеются специальные классы col, col-sm, col-md, col-lg, col-xl, col-auto, col-sm-auto, col-md-auto, col-lg-auto и col-xl-auto.

Первая группа классов (col, col-sm, col-md, col-lg, col-xl) предназначена для создания адаптивных блоков, ширина которых будет зависеть от свободного пространства в линии. Распределение незанятой ширины (свободного пространства) линии между всеми такими блоками осуществляется равномерно. Кроме этого, данные адаптивные блоки перед распределением свободного пространства линии (по умолчанию) имеют нулевую ширину.

Создание в ряду 5 адаптивных блоков с одинаковой шириной.

|  |  |
| --- | --- |
|  | <divclass="row">      <div class="col"> 1/5 </div>      <div class="col"> 2/5 </div>      <div class="col"> 3/5 </div>      <div class="col"> 4/5 </div>      <div class="col"> 5/5 </div>  </div> |

Расчёт ширины:

* свободное пространство линии - 100% (т.к. их ширина равна 0);
* ширина каждого адаптивного блока - 20% (100%/5);

**Расположение адаптивных блоков в ряду**

Адаптивные блоки в ряду по умолчанию располагаются горизонтальными линиями. В пределах горизонтальной линии адаптивные блоки выстраиваются последовательно слева направо. В одну горизонтальную линию могут поместиться адаптивные блоки с суммарным числом колонок не более 12. Адаптивные блоки, которые не помещаются в текущую линию переходят на следующую. На рисунке 10 изображен пример расположения адаптивных блоков в ряду.



Рисунок 10 — Расположение адаптивных блоков в ряду

|  |  |
| --- | --- |
|  | <div class="container">      <div class="row">        <div class="col-6"> (1) </div>        <div class="col-6"> (2) </div>        <div class="col-12"> (3) </div>        <div class="col-8"> (4) </div>      </div>  </div> |

**Основной принцип верстки макета**

Основной принцип верстки макета веб-страницы на сетке Bootstrap 5заключается во вкладывании одних адаптивных блоков в другие.

При этом ширина адаптивных блоков — это всегда относительная величина, которая задаётся в колонках Bootstrap и зависит только от ширины родителя, т.е. ряда.

Размещать контент веб-страницы следует только в адаптивных блоках.

Например, в имеющийся макет, а именно в адаптивный блок col-8 вложим ещё 2 блока:

|  |  |
| --- | --- |
|  | <div class="container">      <div class="row">        <div class=col-8>...</div>        <div class=col-4>...</div>      </div>  </div> |

Для этого предварительно необходимо в блок col-8 положить ряд (рисунок 11):

|  |  |
| --- | --- |
|  | <div class="container">      <div class="row">        <div class=col-8>          <div class="row">...</div>        <div class=col-4>...</div>      </div>  </div> |

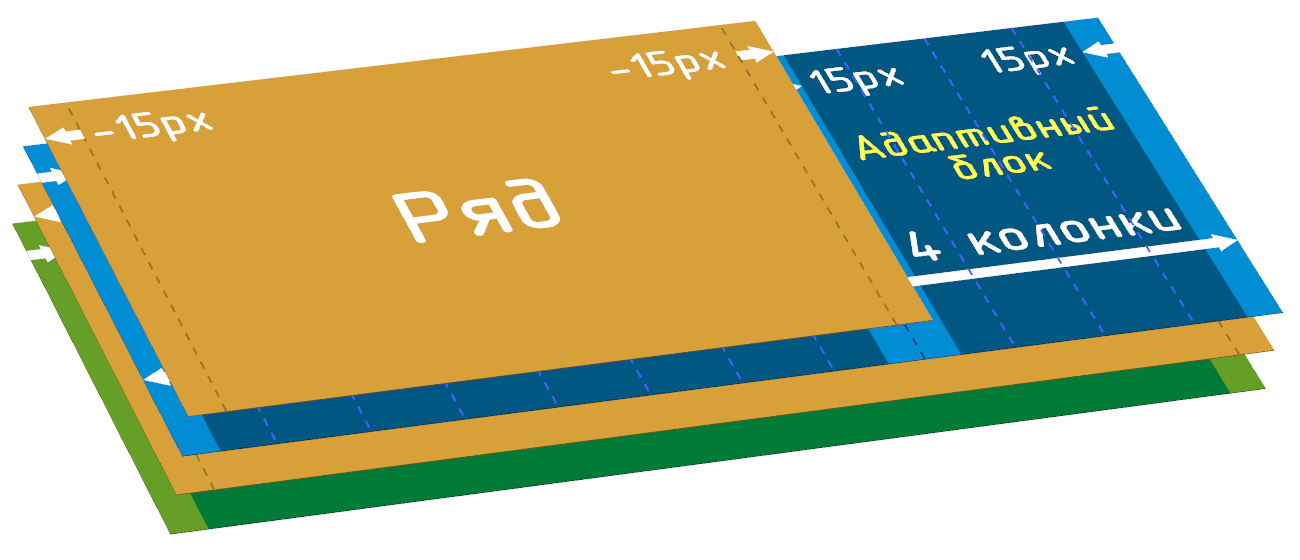


Рисунок 11 — Вставка ряда в содержимое адаптивного блока col-8

После этого добавим 2 адаптивных блока в ряд (рисунок 12):

|  |  |
| --- | --- |
|  | <div class="container">      <div class="row">          <div class=col-8>              <div class="row">                  <div class=col-6>...</div>                  <div class=col-6>...</div>              </div>          <div class=col-4>...</div>      </div>  </div> |

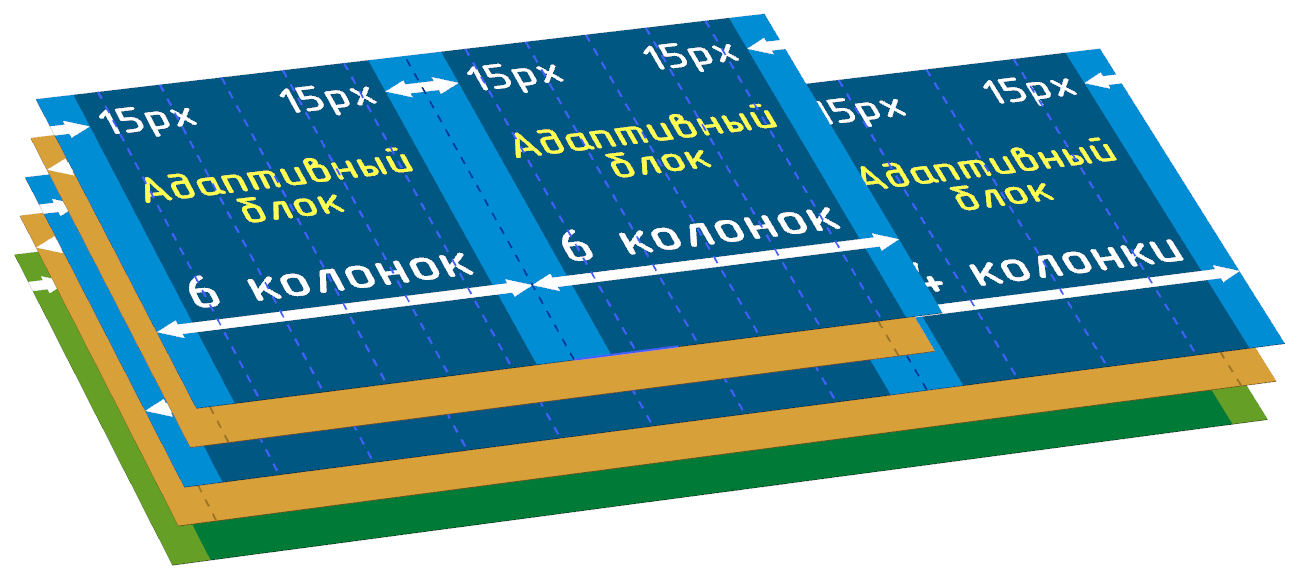


Рисунок 12 — Вставка адаптивных блоков в ряд

**Выравнивание адаптивных блоков**

Выравнивания адаптивных блоков в горизонтальном и вертикальном направлении осуществляется в Bootstrap 5с помощью служебных flex-классов.

**Вертикальное выравнивание адаптивных блоков**

Выравнивание адаптивных блоков в пределах линии ряда по вертикали осуществляется посредством одного из следующих классов, который необходимо дополнительно добавить к row:

* align-items-start (относительно начала линии);
* align-items-center (поцентру);
* align-items-end (относительно конца).

Например, выровняем все адаптивные блоки по центру линии ряда:

|  |  |
| --- | --- |
|  | <div class="row align-items-center">      <div class="col"> 1/2 </div>      <div class="col"> 2/2 </div>  </div> |

По умолчанию адаптивные элементы занимают всю высоту линии ряда, в которой они расположены.

Выравнивание какого-то определённого адаптивного блока по вертикали в пределах линии может осуществляться одним из следующих классов:

* align-self-start (относительно начала линии);
* align-self-center (поцентру);
* align-self-end (относительно конца).

Данные классы необходимо добавлять к адаптивным блокам, а не к ряду.

Например, выравниваем адаптивный блок 2 по нижнему краю линии:

|  |  |
| --- | --- |
|  | <div class="row align-items-center">      <div class="col"> (1) </div>      <div class="col align-self-end"> (2) </div>  </div> |

**Горизонтальное выравнивание адаптивных блоков**

Для выравнивания адаптивных блоков в горизонтальном направлении предназначены следующие классы:

* justify-content-start (относительно начала линии ряда - по умолчанию);
* justify-content-center (поцентру);
* justify-content-end (относительно конца линии);
* justify-content-around (равномерно, с учётом пространства перед первым и последним адаптивным блоком);
* justify-content-between (равномерно, с одинаковым пространством между адаптивными блоками).

Например, распределим адаптивные блоки в горизонтальном направлении равномерно:?

|  |  |
| --- | --- |
|  | <div class="row justify-content-around">      <div class="col-4"> (1) </div>      <div class="col-4"> (2) </div>  </div> |

**Смещение адаптивных блоков**

В Bootstrap 5смещение адаптивных блоков можно выполнить с помощью:

* классов **offset** (на определённое количество колонок);
* служебных (утилитных) **margin классов**.

**Классы offset**

Классы **offset** предназначены для смещения адаптивных блоков вправо на определённое количество колонок.

**Данные классы имеют следующий синтаксис:**

оffset-{1} или offset-{breakpoint}-{1}

{breakpoint} – контрольная точка, начиная с которой к данному блоку будет применено смещение (если она не указана, то смещение будет применено, начиная с самых крохотных устройств).

{2} – величина смещения, указываемая с помощью количества колонок Bootstrap.

В качестве примера установим смещение адаптивным блокам так, как это показано на рисунке 13.

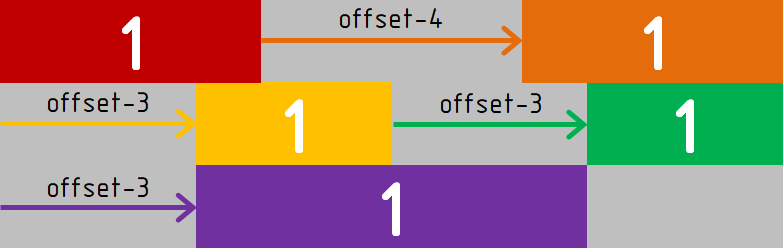


Рисунок 13 — Смещение адаптивных блоков с помощью offset классов

|  |  |
| --- | --- |
|  | <div class="row">      <div class="col-4"> (1) </div>      <div class="col-4 offset-4"> (2) </div>  </div>  <div class="row">      <div class="col-3 offset-3"> (1) </div>      <div class="col-3 offset-3"> (2) </div>  </div>  <div class="row">      <div class="col-6 offset-3"> (1) </div>  </div> |

**Смещение с помощью margin классов**

В пятой версии Bootstrap устанавливать смещение адаптивным блокам также можно с помощью **margin отступов** (margin-left: auto и (или) margin-right: auto). Эта вариант смещения появился благодаря тому, что сетка в новой версии Bootstrap (4) основывается на **CSS Flexbox**.

Данный вариант смещения (с помощью margin классов) используют, когда **блоки необходимо сместить относительно друг от друга не которую переменную величину**. На рисунке 14 показывется пример смещение адаптивных блоков с помощью margin классов.

В Bootstrap 5для более удобного и адаптивного задания блокам margin отступов (margin-left: auto и (или) margin-right: auto) можно использовать классы ml-auto, mr-auto, ml-{breakpoint}-auto и mr-{breakpoint}-auto.

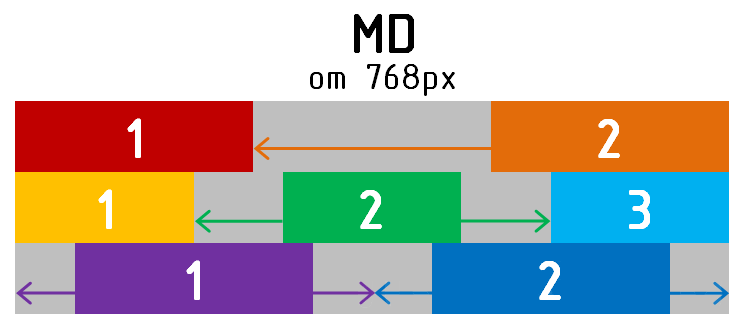


Рисунок 14 — Смещение адаптивных блоков с помощью margin классов

|  |  |
| --- | --- |
|  | <div class="row">      <div class="col-4"> (1) </div>      <div class="col-4 ml-auto"> (2) </div>  </div>  <div class="row">      <div class="col-3"> (1) </div>      <div class="col-3 ml-auto mr-auto"> (2) </div>      <div class="col-3"> (3)</div>  </div>  <div class="row">      <div class="col-4 ml-auto mr-auto"> (1) </div>      <div class="col-4 ml-auto mr-auto"> (2) </div>  </div> |

**Контрольные вопросы**

1. Что такое Twitter Bootstrap?
2. Какие основные преимущества Bootstrap?
3. Что включает в себя пакет Bootstrap?
4. Какие основные инструменты Bootstrap?
5. Из чего состоит Bootstrap?
6. Какие преимущества есть фреймворка Bootstrap
7. Какие недостатки есть у фреймворка Bootstrap
8. Какие версии Botstrap есть на текущий момент?
9. Основные отличия Bootstrap 5от Bootstrap 3?
10. Какие виды контейнеров в Bootstrap вы знаете?