



# ОСНОВЫ WEB-ТЕХНОЛОГИЙ

Лекция 6

# СТРУКТУРА ЗАНЯТИЯ

- Bootstrap – это?
- Компоненты

# МОДУЛЬНЫЙ ПОДХОД К РАЗРАБОТКЕ

Table

Float  
(Bootstrap)

Display:  
inline-block

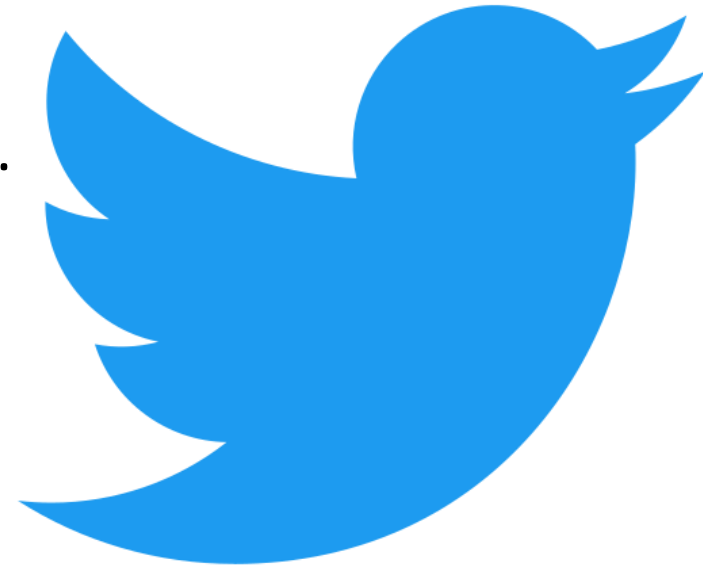
Flex

Grid

# ЧТО ТАКОЕ BOOTSTRAP?

Bootstrap — это CSS/HTML фреймворк для создания веб-сайтов. Другими словами, это набор инструментов для создания веб-макета.

Эта библиотека начала разрабатываться как внутренняя библиотека компании Twitter\* под названием **Twitter Blueprint**.



\*Организация,  
заблокированная  
на территории РФ

# ИЗ ИСТОРИИ

**19 августа 2011** Bootstrap 1

**31 января 2012** Bootstrap 2(1 2-колонок, поддержка адаптивности)

**19 августа 2013** Bootstrap 3(mobile first, плоский дизайн)

**18 января 2018** года выпущена первая стабильная версия Bootstrap 4

**5 мая 2021** Bootstrap 5

# ПРЕИМУЩЕСТВА BOOTSTRAP

1. **Скорость работы** – создание макетов с Bootstrap занимает меньше времени благодаря большому набору готовых к использованию элементов.
2. **Гибкость** – добавление новых элементов не нарушает общую структуру благодаря динамически изменяющейся сетке.
3. **Легкая изменяемость** – правка стилей достигается за счет добавления новых CSS правил, которые переопределяют существующие. При этом, вам не нужно использовать атрибуты типа !important.

# ПРЕИМУЩЕСТВА BOOTSTRAP

1. **Большое количество шаблонов** – этот момент будет отдельно рассмотрен далее.
2. **Огромное сообщество сторонников/разработчиков.**
3. **Широкий спектр применения** – Bootstrap используется для создания тем почти для любой CMS (Magento, Joomla, WordPress или любой другой), включая одностраничные лендинги.
4. **Замечательная официальная документация.**

# СОСТАВЛЯЮЩИЕ BOOTSTRAP

1. CSS/HTML
2. JS компоненты
3. Иконочный шрифт



# ГДЕ ВЗЯТЬ?

<https://github.com/twbs/bootstrap>

<https://getbootstrap.com/>

<https://bootstrap-4.ru/>

# СЕТКА

Используется система  
«12 колонок»

5 адаптивных ярусов

# КАК ЭТО УСТРОЕНО?

Система сеток Bootstrap 4 использует контейнеры, ряды и колонки, чтобы удобно располагать содержимое. Бутстрап реализован с помощью флексбокса и полностью адаптивен.

# ЧТО ТАКОЕ FLEXBOX?

**Флексбокс** — это CSS-механизм, который позволяет контролировать размер, порядок и выравнивание элементов по нескольким осям, распределение свободного места между элементами и многое другое.

Чтобы включить флексбокс, нужно задать элементу свойство `display: flex;`. После этого:

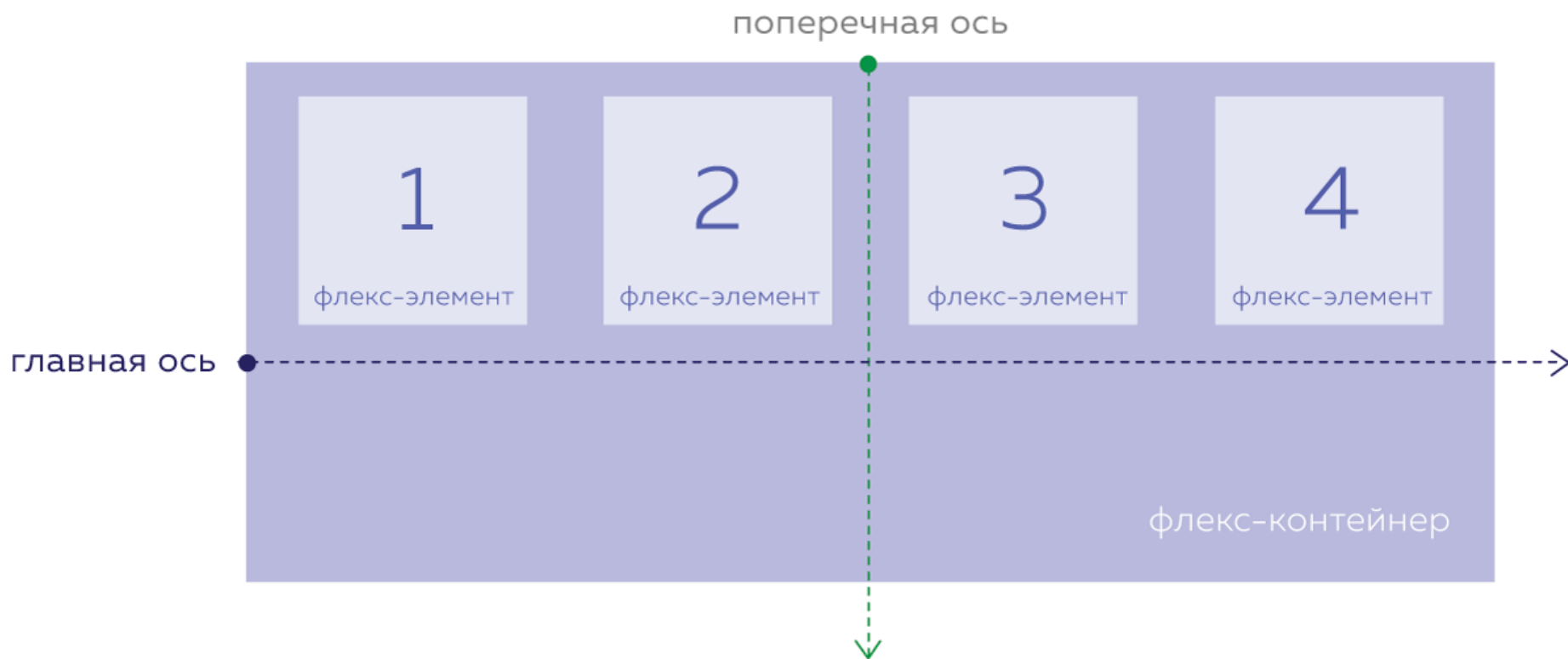
1. Элемент с `display: flex;` превращается во «флекс-контейнер».
2. Непосредственные потомки этого элемента превращаются во «флекс-элементы» и начинают распределяться по новым правилам.

# ГЛАВНАЯ И ПОПЕРЕЧНАЯ ОСИ

**Главной осью** flex-контейнера является направление, в соответствии с которым располагаются все его дочерние элементы. Поток флекс-элементов «течёт» вдоль главной оси от её начала к её концу.

**Поперечной осью** называется направление, перпендикулярное главной оси. Вдоль этой оси работают «вертикальные» выравнивания.

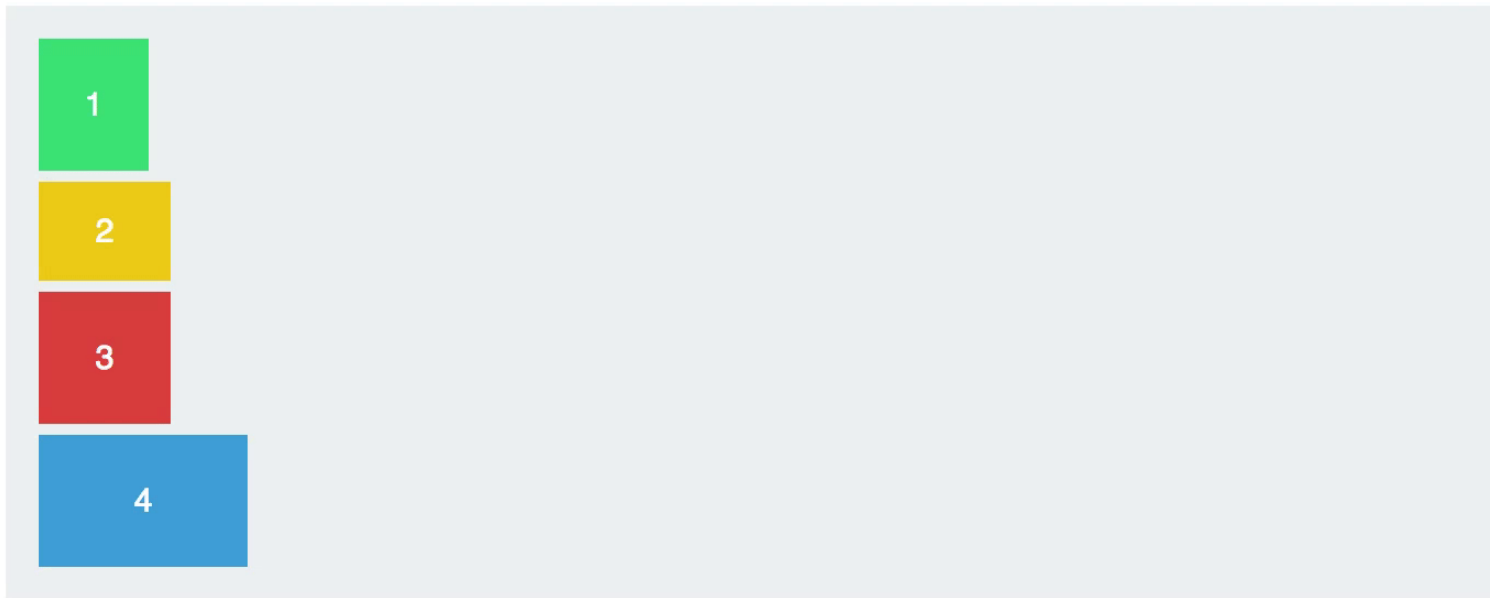
# ГЛАВНАЯ И ПОПЕРЕЧНАЯ ОСИ



# КАК ЭТО УСТРОЕНО?

Пример

**display: block;**



# ГЛАВНАЯ ОСЬ

По умолчанию главная ось направлена слева направо, но её можно разворачивать во всех направлениях с помощью свойства `flex-direction`, которое задаётся для флекс-контейнера. Значения свойства:

- Значение по умолчанию **row** — главная ось направлена слева направо.
- **column** — главная ось направлена сверху вниз.
- **row-reverse** — главная ось направлена справа налево.
- **column-reverse** — главная ось направлена снизу вверх.



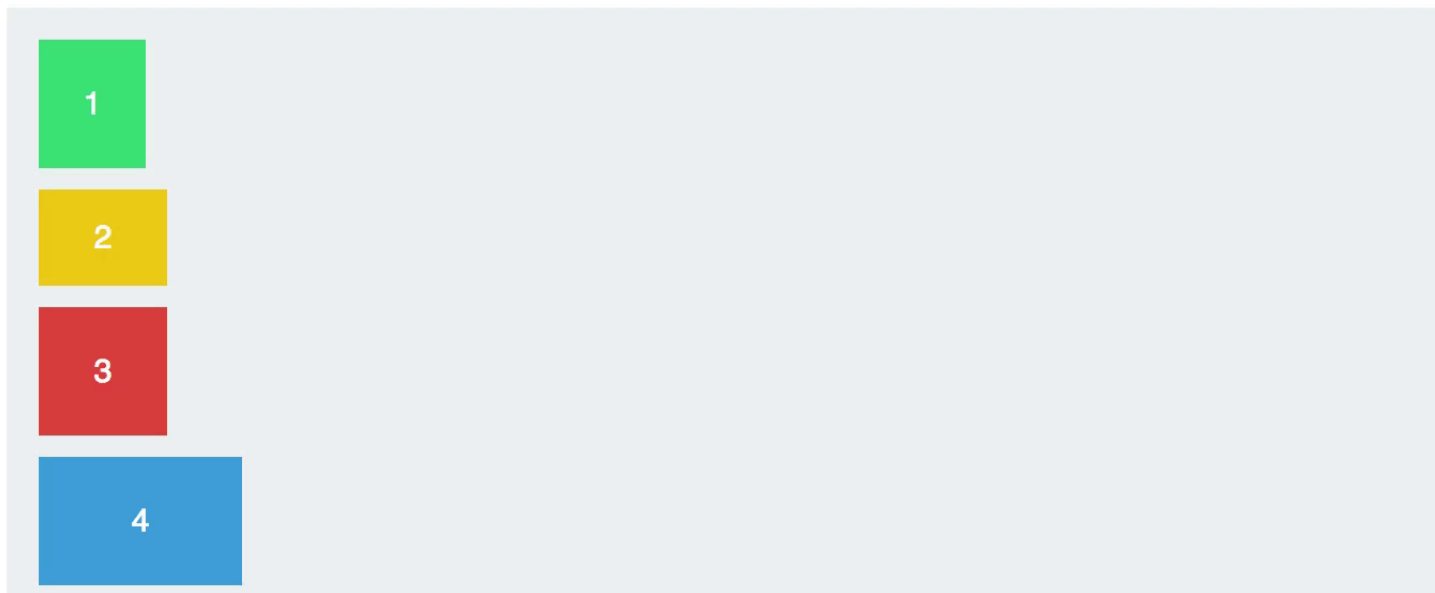
# КАК ЭТО УСТРОЕНО?

**flex-direction: row;**



# КАК ЭТО УСТРОЕНО?

`flex-direction: column;`



# ПОПЕРЕЧНАЯ ОСЬ

Поперечная ось всегда перпендикулярна главной оси и поворачивается вместе с ней:

- Если главная ось направлена горизонтально, то поперечная ось смотрит вниз.
- Если главная ось направлена вертикально, то поперечная ось смотрит направо.

Таким образом, поперечная ось никогда не смотрит вверх или влево, и свойства для поворота поперечной оси нет.

# РАСПРЕДЕЛЕНИЕ ФЛЕКС-ЭЛЕМЕНТОВ

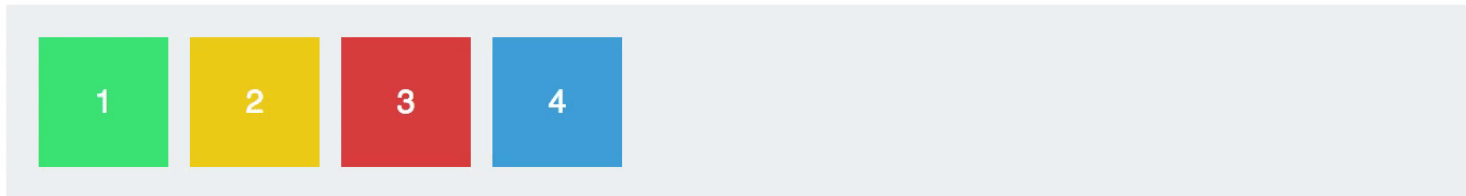
## *Выравнивание по главной оси*

CSS-свойство **justify-content** определяет то, как будут выровнены элементы вдоль главной оси. Доступные значения justify-content:

- Значение по умолчанию **flex-start** — элементы располагаются у начала главной оси.
- **flex-end** — элементы располагаются в конце главной оси.
- **center** — элементы располагаются по центру главной оси.
- **space-between** — элементы располагаются так, что расстояния между соседними одинаковые, а между элементами и краями флекс-контейнера отступов нет.
- **space-around** — элементы располагаются так, что расстояния между соседними одинаковые, а между элементами и краями флекс-контейнера есть отступ, равный половине расстояния между соседними элементами.
- **space-evenly** — расстояния между соседними элементами и краями флекс-контейнера одинаковые.

# КАК ЭТО РАБОТАЕТ?

**`justify-content: flex-start;`**



# РАСПРЕДЕЛЕНИЕ ФЛЕКС-ЭЛЕМЕНТОВ

## *Выравнивание по поперечной оси*

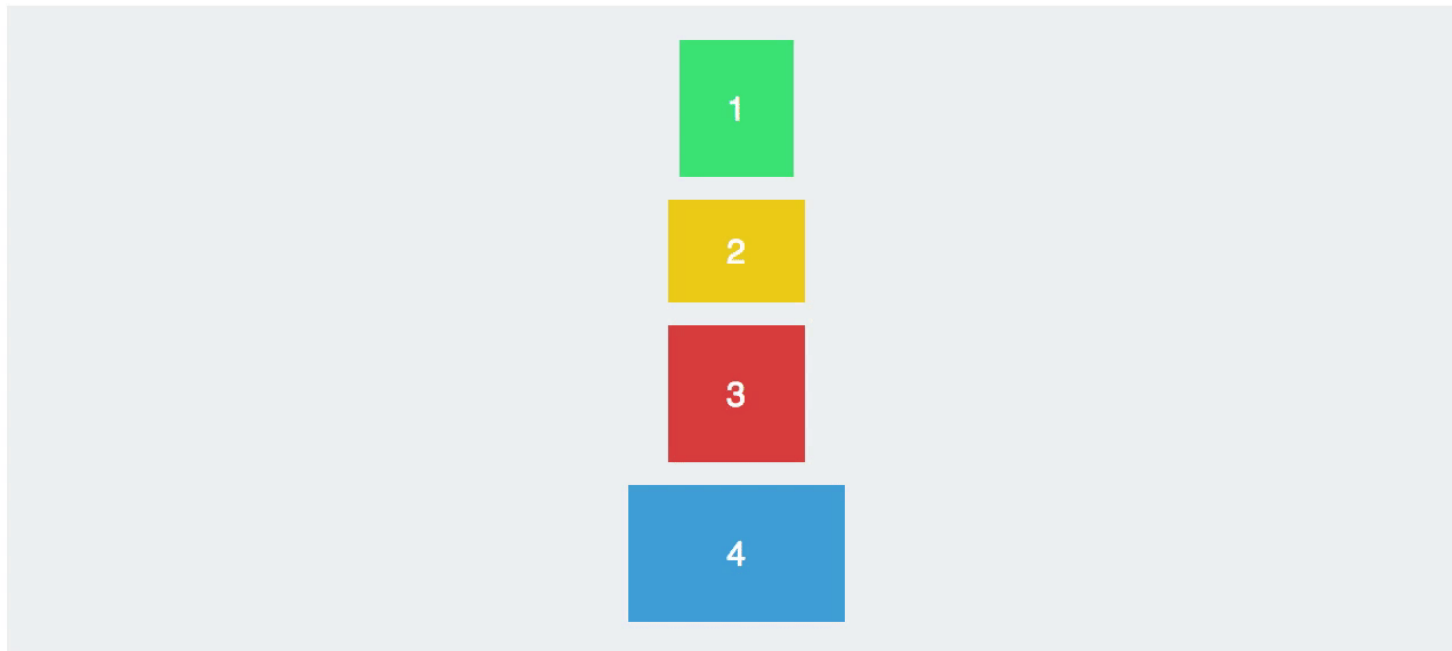
CSS-свойство **align-items** определяет то, как будут выровнены элементы вдоль поперечной оси. Доступные значения align-items:

- Значение по умолчанию **stretch** — элементы растягиваются на всю «высоту» флекс-контейнера.
- **flex-start** — элементы располагаются у начала поперечной оси.
- **flex-end** — элементы располагаются в конце поперечной оси.
- **center** — элементы располагаются по центру поперечной оси.
- **baseline** — элементы выравниваются по базовой линии текста внутри них.

Распределение элементов по главной оси задаётся для всего флекс-контейнера и на все флекс-элементы действует одинаково, задать какому-то элементу отличное от других распределение по главной оси нельзя.

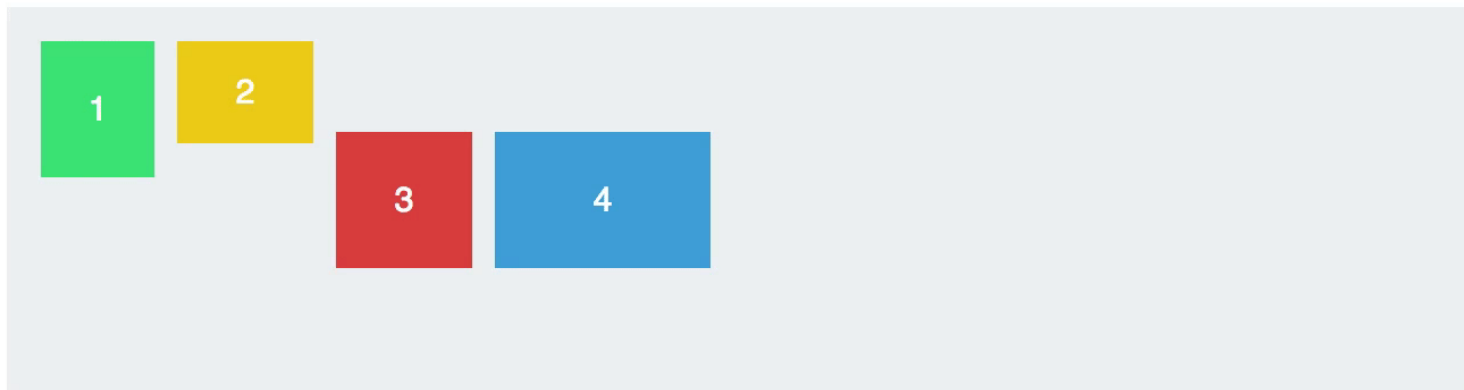
# КАК ЭТО УСТРОЕНО?

**flex-direction: column;**  
**justify-content: center; align-items: center;**



# КАК ЭТО УСТРОЕНО?

**align-self: flex-start;**





# ПОРЯДКОВЫЙ НОМЕР ФЛЕКС-ЭЛЕМЕНТА

Порядок следования флекс-элементов в потоке можно изменять с помощью свойства `order`, порядкового номера флекс-элемента, не меняя при этом HTML-код.

По умолчанию порядковый номер флекс-элементов равен 0, а сортировка элементов производится по возрастанию номера. Порядковый номер задаётся целым числом, положительным или отрицательным.

# КОНТЕЙНЕР

Контейнеры являются фундаментальным строительным блоком Bootstrap, которые содержат ваш контент, дополняют и выравнивают его на устройствах или в областях просмотра.

# КОНТЕЙНЕРЫ

В Bootstrap используется три разных типа контейнеров:






- `.container`, который устанавливает максимальную ширину `max-width` в каждой отзывчивой контрольной точке;
- `.container-fluid`, ширина которого `width: 100%` на всех контрольных точках;
- `.container-{breakpoint}`, то есть ширина `width: 100%` до указанной контрольной точки

# БРЕЙКПОЙНТ – ЭТО?

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.

# ПАРАМЕТРЫ СЕТОК

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

# ПАРАМЕТРЫ СЕТОК

Бутстрап использует **em** и **rem** для задания большинства размеров, а пиксели **px** — для «брейкпойнтов» сетки и ширин контейнеров.

# АВТОМАТИЧЕСКОЕ РАСПОЛОЖЕНИЕ С ПОМОЩЬЮ КОЛОНОК

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

# РАСПОЛОЖЕНИЕ КОЛОНОК

.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1
.col-md-8								.col-md-4			
.col-md-4			.col-md-4				.col-md-4				
.col-md-6						.col-md-6					

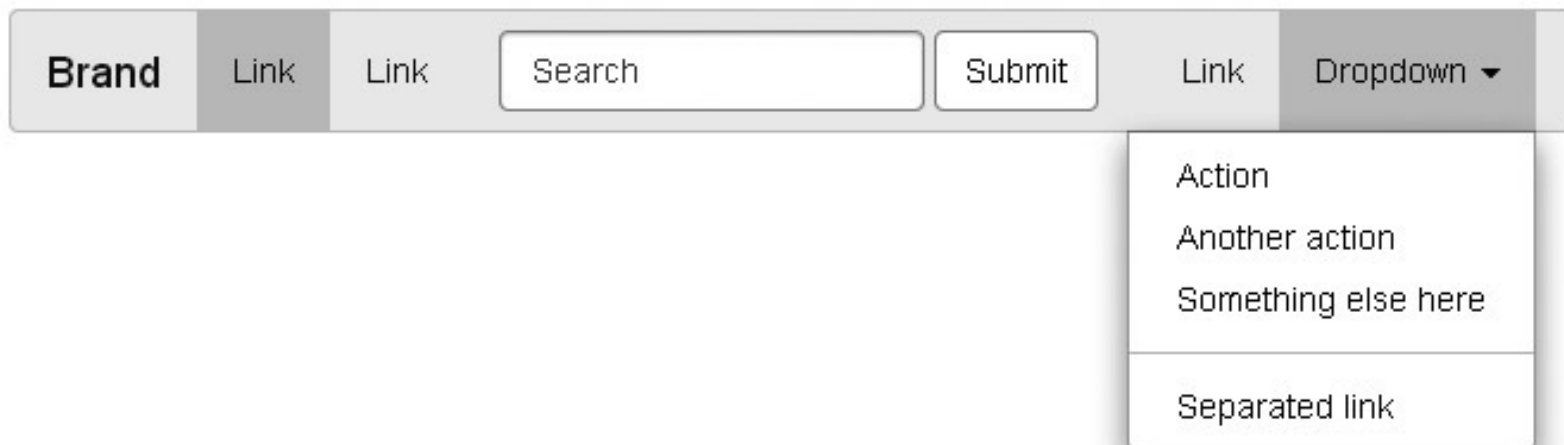


# НАВИГАЦИЯ

Навигации в Bootstrap уделено особое внимание:

- фреймворк содержит дизайны для вкладок (табов),
- постраничной навигации (пагинации),
- боковых меню,
- «хлебных крошек»,
- основного меню,
- панели инструментов (тулбара) и т.д.

# ПРИМЕР ШАБЛОНА НАВИГАЦИИ



A horizontal navigation bar template with the following elements from left to right: a 'Brand' label, a 'Link' label, another 'Link' label, a 'Search' input field, a 'Submit' button, a 'Link' label, and a 'Dropdown' menu with a downward arrow. The dropdown menu is open, showing four items: 'Action', 'Another action', 'Something else here', and 'Separated link'.

Brand	Link	Link	Search	Submit	Link	Dropdown ▼
						Action
						Another action
						Something else here
						Separated link

# ОПОВЕЩЕНИЯ (АЛЕРТЫ)

Любое оповещение может быть представлено в 4 стандартных форматах:

- положительный,
- информационный,
- предупреждающий,
- отрицательный.

# ПАГИНАЦИЯ

Пагинация выглядит вот так:



# КНОПКИ

Default

Primary

Success

Info

Warning

Danger

Link

# ТАБЛИЦЫ

#	First Name	Last Name	Username
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry	the Bird	@twitter

# ФОРМЫ

## Email address

## Password

## File input

 No file selected.

Example block-level help text here.

☐ Check me out

# ТИПОГРАФИКА

h1. Bootstrap heading

---

h2. Bootstrap heading

---

h3. Bootstrap heading

---

h4. Bootstrap heading

---

h5. Bootstrap heading

---

h6. Bootstrap heading



# ИКОНОЧНЫЙ ШРИФТ



# ПОРЯДОК РАБОТЫ

Бутстрап подключаем  
вначале, как базу, а потом  
подключаем свои стили и  
переопределяем стили

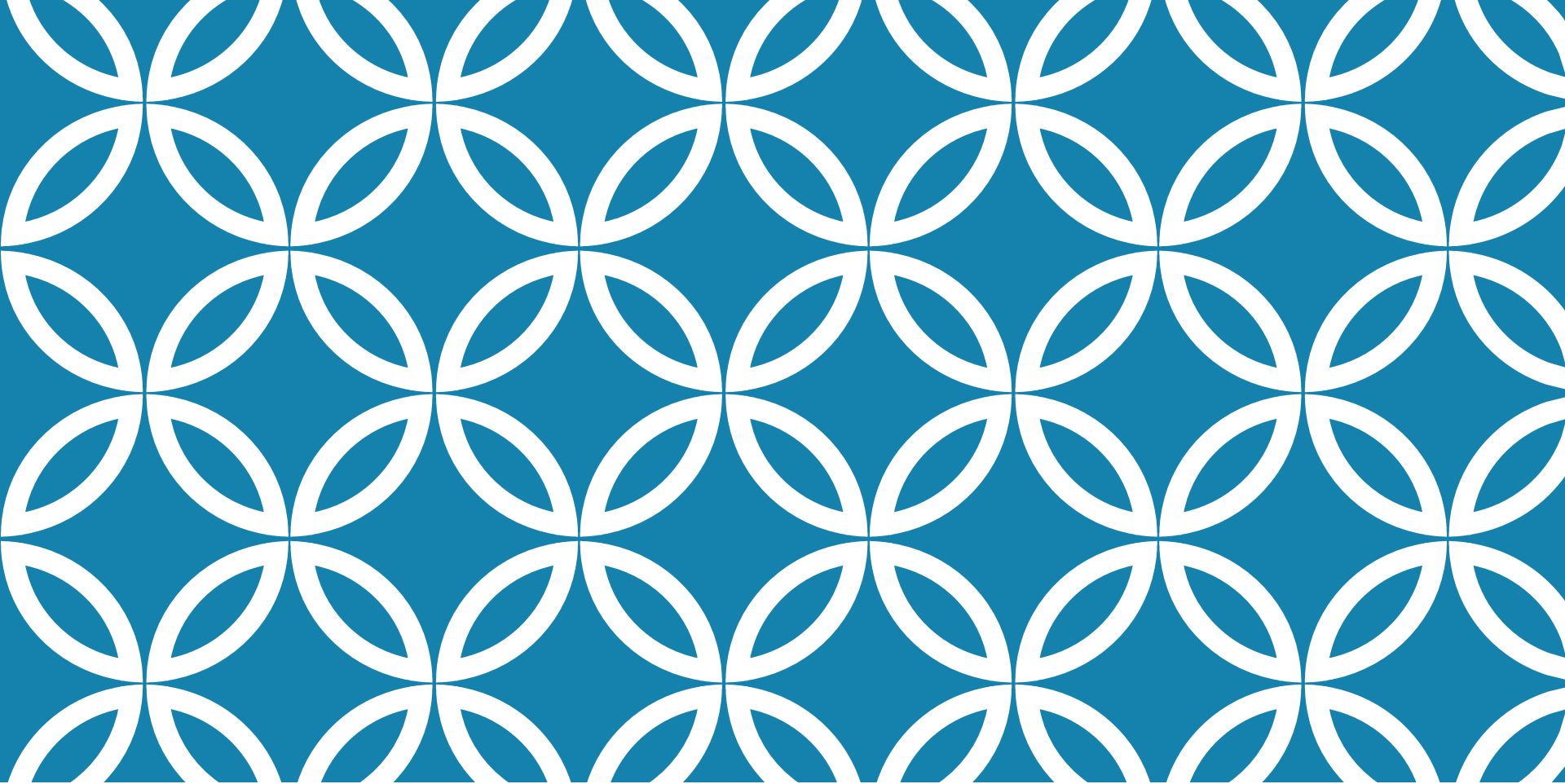
# ШАБЛОНЫ BOOTSTRAP

Шаблоны в Bootstrap позволяют вам изменять уже модифицированные элементы под ваши нужды.

Подключаются шаблоны Bootstrap очень просто: после подключения самого Bootstrap вы просто добавляете вызов CSS шаблона.

# КОМПОНЕНТЫ JAVASCRIPT

Чтобы управлять этими компонентами, потребуется библиотека jQuery (для BS-4); не забудьте подключить ее к файлу bootstrap.js



**СПАСИБО ЗА  
ВНИМАНИЕ!**

конец лекции 4