



CSS. Bootstrap

Стилизация html-разметки

- Элементам Html можно задавать дополнительные настройки стиля с помощью атрибута style. Пример:

```
<p style="font-family:times;color:green;">
```

Этот текст написан зеленым цветом с помощью шрифта Times.

```
</p>
```



CSS

- CSS (Cascading Style Sheets) — код используемый для стилизации веб-страницы.
- Как и HTML, CSS не является языком программирования. Но это и не язык разметки - это *язык таблицы стилей*. Это означает, что он позволяет применять стили выборочно к элементам в документах HTML.
- Например, чтобы выбрать **все** элементы абзаца на HTML странице и изменить текст внутри них с черного на красный, вы должны написать этот CSS:

```
p {  
    color: red;  
}
```



Пример

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8">
    <title>Ter STYLE</title>
    <style type="text/css">
      H1 {
        color: red;
      }
    </style>
  </head>
  <body>
    <H1>Hello, world!</H1>
  </body>
</html>
```



Преимущества использования CSS-файлов

- Разграничение кода и оформления
- Ускорение загрузки сайта (кэширование)
- Централизованное хранение

```
<link href="styles/example.css" rel="stylesheet"  
      type="text/css" >
```

Структура CSS-правила

селектор	свойство	значение
<code>body</code>	<code>{ background:</code>	<code>#ffc910; }</code>

- Каждый набор правил должен быть обернут в фигурные скобки ({}).
- В каждом объявлении необходимо использовать двоеточие (:), чтобы отделить свойство от его значений.
- В каждом наборе правил вы должны использовать точку с запятой (;), чтобы отделить каждое объявление от следующего.

Выбор нескольких элементов

- Возможно выбрать несколько элементов разного типа и применить единый набор правил для всех из них. Например:

```
p, li, h1 {  
    color: red;  
    border: 1px solid black;  
}
```


Типы селекторов

Имя селектора	Что выбирает	Пример
Селектор элемента (иногда называемый селектором тега или типа)	Все HTML элемент(ы) указанного типа.	p
		Выбирает <p>
ID селектор	Элемент на странице с указанным ID на данной HTML. Лучше всего использовать один элемент для каждого ID (и конечно один ID для каждого элемента), даже если вам разрешено использовать один и тот же ID для нескольких элементов.	#my-id
		Выбирает <p id="my-id"> или
Селектор класса	Элемент(ы) на странице с указанным классом (множество экземпляров класса может объявляться на странице).	.my-class
		Выбирает <p class="my-class"> и
Селектор атрибута	Элемент(ы) на странице с указанным атрибутом.	img[src]
		Выбирает но не
Селектор псевдо-класса	Указанные элемент(ы), но только в случае определенного состояния, например, при наведении курсора.	a:hover
		Выбирает <a>, но только тогда, когда указатель мыши наведен на ссылку.



Типы носителя



Тип	Описание
all	Все типы. Это значение используется по умолчанию.
aural	Речевые синтезаторы, а также программы для воспроизведения текста вслух. Сюда, например, можно отнести речевые браузеры.
braille	Устройства, основанные на системе Брайля, которые предназначены для слепых людей.
handheld	Наладонные компьютеры и аналогичные им аппараты.
print	Печатающие устройства вроде принтера.
projection	Проектор.
screen	Экран монитора.
tv	Телевизор.

Примеры

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8">
    <title>Импорт стиля</title>
    <style>
      @import "/style/main.css" screen; /* Стил ь для вывода результата на монитор */
      @import "/style/smart.css" print, handheld; /* Стил ь для печати и смартфона */
    </style>
  </head>
  <body>
    <p>...</p>
  </body>
</html>
```

Медиа-запросы в CSS:

```
@media screen and (max-width: 575px) {
  #main {
    display: block;
  }
}
```

Единицы задания размеров

➤ Относительные

Единица	Описание
em	Размер шрифта текущего элемента
ex	Высота символа x
px	Пиксел
%	Процент

➤ Абсолютные

Единица	Описание
in	Дюйм (1 дюйм равен 2,54 см)
cm	Сантиметр
mm	Миллиметр
pt	Пункт (1 пункт равен 1/72 дюйма)
pc	Пика (1 пика равна 12 пунктам)



Задание цвета

```
BODY {  
    background-color: #3366CC; /* Цвет фона веб-страницы */  
}  
  
H1 {  
    background-color: RGB(249, 201, 16); /* Цвет фона под заголовком */  
}  
  
H2 {  
    background-color: RGBA(255, 99, 71, 1)  
}  
  
H3 {  
    background-color: HSLA(9, 90%, 90%, 1)  
}  
  
p{  
    background-color: red;  
}
```

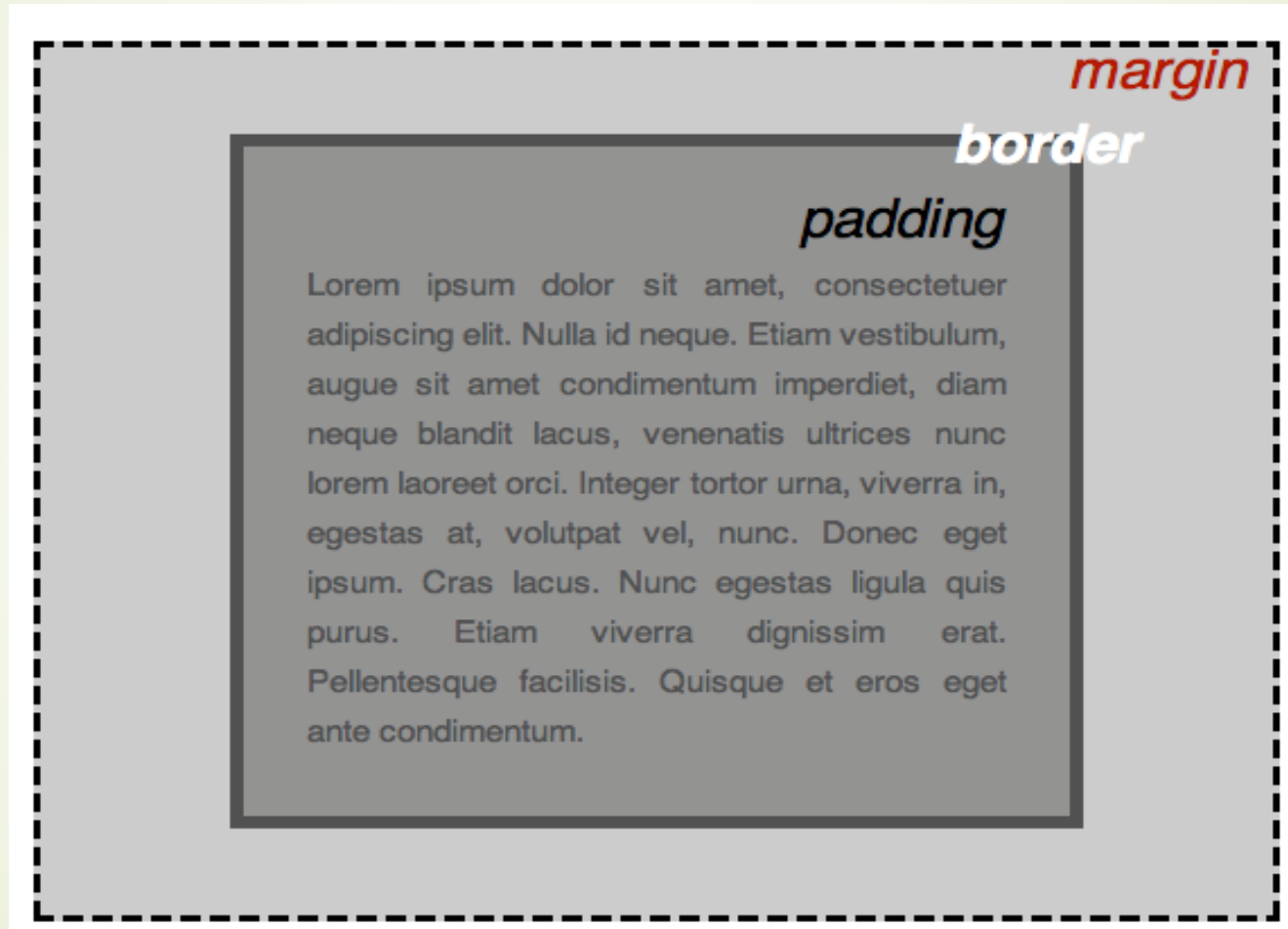


Использование адресов

```
body {  
    background: url('http://webimg.ru/images/156_1.png') no-repeat;  
}
```

```
div {  
    background: url(images/warning.png) no-repeat;  
    padding-left: 20px;  
    margin-left: 200px;  
}
```

БЛОКИ





Ключевые слова

- В качестве значений активно применяются ключевые слова, которые определяют желаемый результат действия стилевых свойств. Ключевые слова пишутся без кавычек.

Правильно: `P { text-align: right; }`

Неверно: `P { text-align: "right"; }`


Переменные

- CSS переменные - это сущности, определяемые автором CSS, хранящие конкретные значения, которые можно повторно использовать в документе. Они устанавливаются с использованием custom property нотации (например, `--main-color: black;`) и доступны через функцию `var()` (например, `color: var(--main-color);`)

```
:root {  
    --main-bg-color: brown;  
}  
  
.one {  
    color: white;  
    background-color: var(--main-bg-color);  
}  
  
.two {  
    color: var(--main-bg-color, red); /* red если --my-var не определена */  
}
```

Псевдоклассы

- :link – выберет любую ссылку, которая еще не была посещена, даже те, которые уже стилизованы
- :visited – позволяет вам выбирать ссылки, которые были посещены.
- :active – соответствует элементу в момент, когда он активизируется пользователем (TAB)
- :hover – срабатывает, когда пользователь наводит на элемент мышью, но не обязательно активизирует его
- :focus – применяется, когда элемент (такой как input формы) получает фокус. Обычно он активизируется при клике мышью пользователем или при выборе элемента с использованием клавиши "tab" на клавиатуре.
- :first-child – находит любой элемент, являющийся первым в своем родителе.
- :last-child – любой элемент, являющийся последним в его родителе.
- :nth-child() – находит один или более элементов, основываясь на их позиции среди группы соседних элементов ($2n$, $2n+1$ и т.д.).
- :nth-last-child() – находит элемент, имеющий $a+n-1$ потомков после данной позиции в дереве документа, значение для n может быть положительным или нулевым, а также имеющий родительский элемент
- :nth-of-type() – находит один или более элементов с заданным тегом, основываясь на их позиции среди группы соседних элементов.
- :first-of-type – находит первого потомка своего типа среди детей родителя.

- 
- 
- :last-of-type – находит последнего потомка с заданным тегом в списке детей родительского элемента.
 - :empty – находит любой элемент, у которого нет потомков.
 - :target – представляет уникальный элемент (целевой элемент) с подходящим id URL-фрагментом (модальное окно, вкладка и т.п.)
 - :checked – находит отмеченные radio-buttons и check-buttons
 - :enabled – находит любой включенный элемент. Элемент включен, если его можно активировать (например, выбрать, нажать на него или ввести текст) или поставить фокус. У элемента также есть отключенное состояние, когда его нельзя активировать или сфокусировать.
 - :disabled – находит любой не включенный элемент
 - :after – применяется для вставки назначенного контента после содержимого элемента. Этот псевдоэлемент работает совместно со стилевым свойством content, которое определяет содержимое для вставки.
 - :before – по своему действию :before аналогичен псевдоэлементу :after, но вставляет контент до содержимого элемента.
 - :first-letter – определяет стиль первого символа в тексте элемента, к которому добавляется. Это позволяет создавать в тексте буквицу и выступающий инициал.
 - :first-line – определяет стиль первой строки блочного текста.



Комбинированный селектор

- Возможно комбинировать селектор, создавая более конкретный селектор. Например, селектор `.key` выбирает все элементы, с ключом имени класса `key`. Селектор `p.key` отбирает только `<p>` элементы, которые имеют имя класса `key`.
- Если таблица стилей имеет противоречивые правила, и они одинаково специфичны, тогда CSS дает приоритет правилу, которое позже находится в таблице стилей.

Селекторы на основе отношений

Селектор	Выбрано
A E	Любой E элемент, что является потомком одного из A элемента (то есть: дочерний, или один из дочернего, т.д.)
A > E	Любой E элемент, что является дочерним (т.е. прямой потомок) A элемента.
E:first-child	Любой E элемент, что является первым дочерним элементом родительского элемента.
B + E	Любой E элемент, что является следующим "братом" B элемента (то есть: следующий ребенок того же родителя)

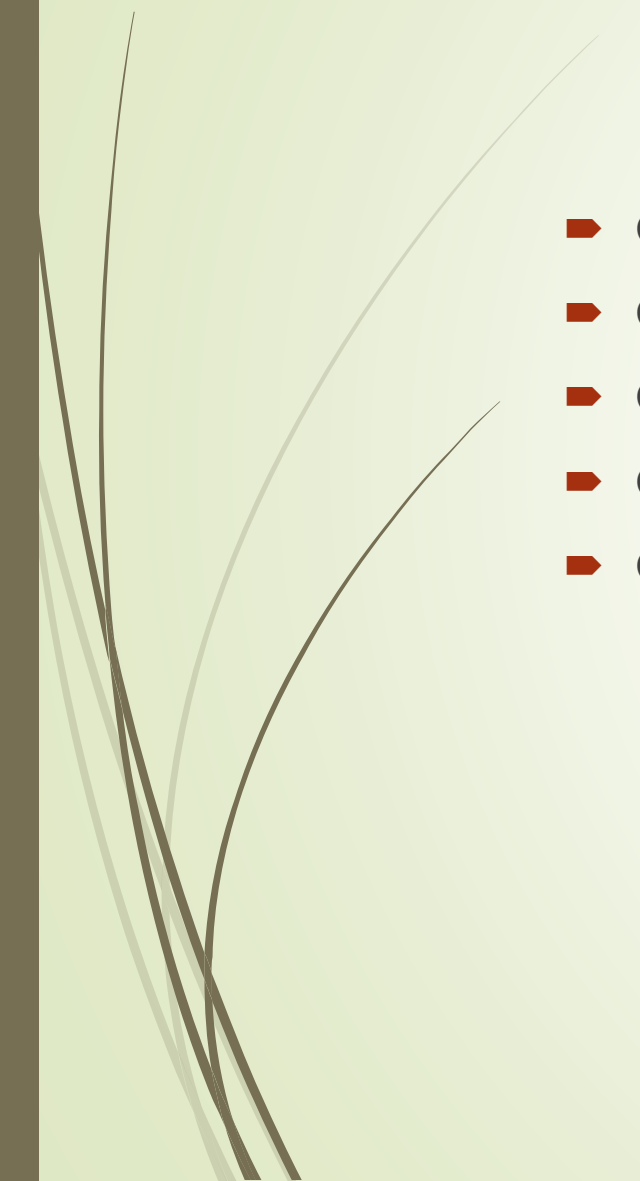


Наследование правил

- Наследованием называется перенос правил форматирования для элементов, находящихся внутри других. Такие элементы являются дочерними, и они наследуют **некоторые** стилевые свойства своих родителей, внутри которых располагаются. Пример: таблицы



Приоритеты стилевых правил

- Стиль браузера.
 - Стиль автора.
 - Стиль пользователя.
 - Стиль автора с добавлением !important.
 - Стиль пользователя с добавлением !important.
- 



Специфичность

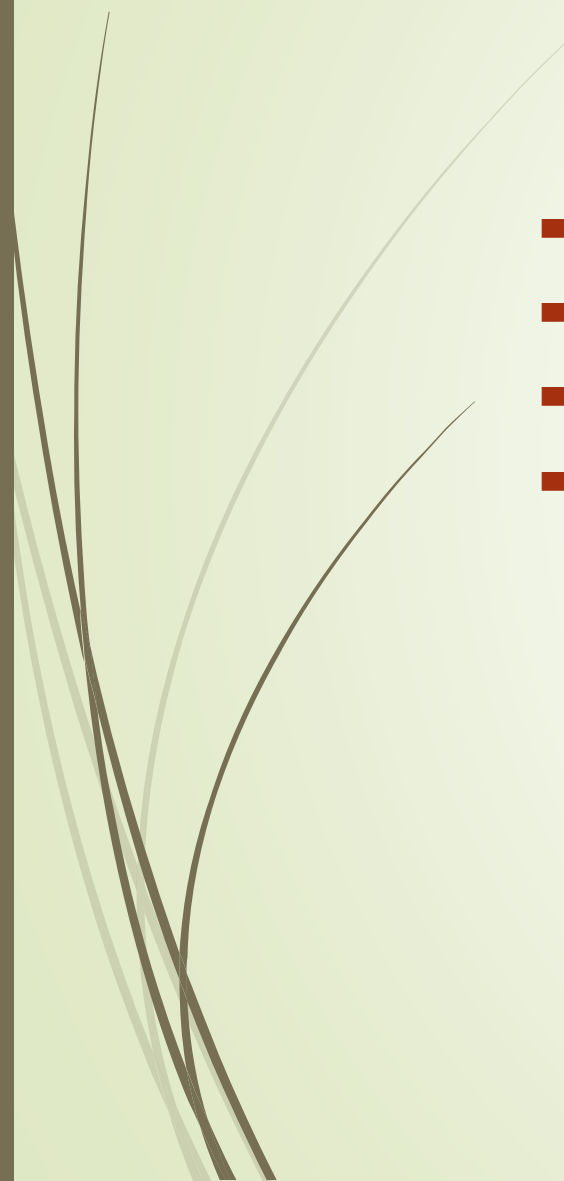
- Если к одному элементу одновременно применяются противоречивые стилевые правила, то более высокий приоритет имеет правило, у которого значение специфичности селектора больше. Специфичность это некоторая условная величина, вычисляемая следующим образом. За каждый идентификатор (в дальнейшем будем обозначать их количество через *a*) начисляется 100, за каждый класс и псевдокласс (*b*) начисляется 10, за каждый селектор тега и псевдоэлемент (*c*) начисляется 1. Складывая указанные значения в определенном порядке, получим значение специфичности для данного селектора.
- Встроенный стиль, добавляемый к тегу через атрибут `style`, имеет специфичность 1000, поэтому всегда перекрывает связанные и глобальные стили. Однако добавление `!important` перекрывает в том числе и встроенные стили.
- Если два селектора имеют одинаковую специфичность, то применяться будет тот стиль, что указан в коде ниже.

Особенности HTML5

- Поддержка аудио и видео, элементы
 <audio>
 <video>
- Рисование, элемент:
 <canvas>
- Новые атрибуты форм, такие как placeholder, require, novalidate
- Использование локального хранилища данных
- Media-запросы



Особенности CSS3

- Градиенты цветов
 - Скругленные элементы
 - Тени
 - Анимация
- 



Flexbox

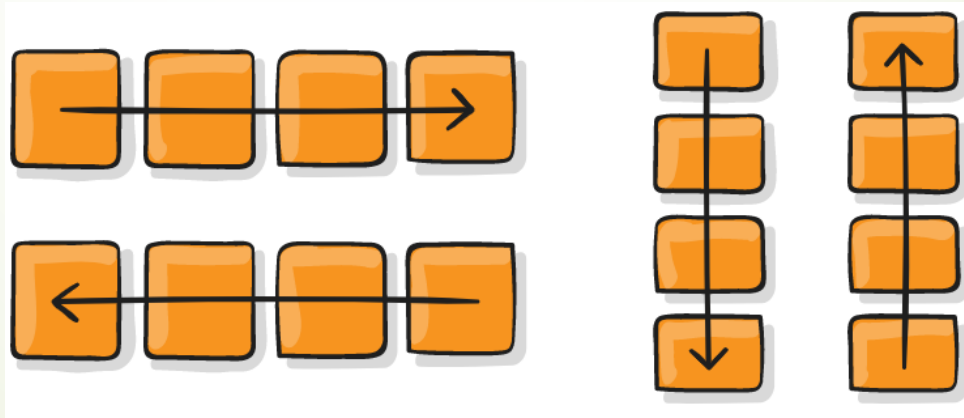
- CSS Flexbox — это технология для создания сложных гибких макетов за счёт правильного размещения элементов на странице.
- Для создания макета необходимо поместить контент внутрь flexbox-контейнера:

```
.container {  
    display: flex; /* or inline-flex */  
}
```

Flex-direction

- Далее необходимо установить направление основной оси

```
.container {  
  flex-direction: row | row-reverse | column | column-reverse;  
}
```



justify-content

flex-start



flex-end



center



space-between



space-around

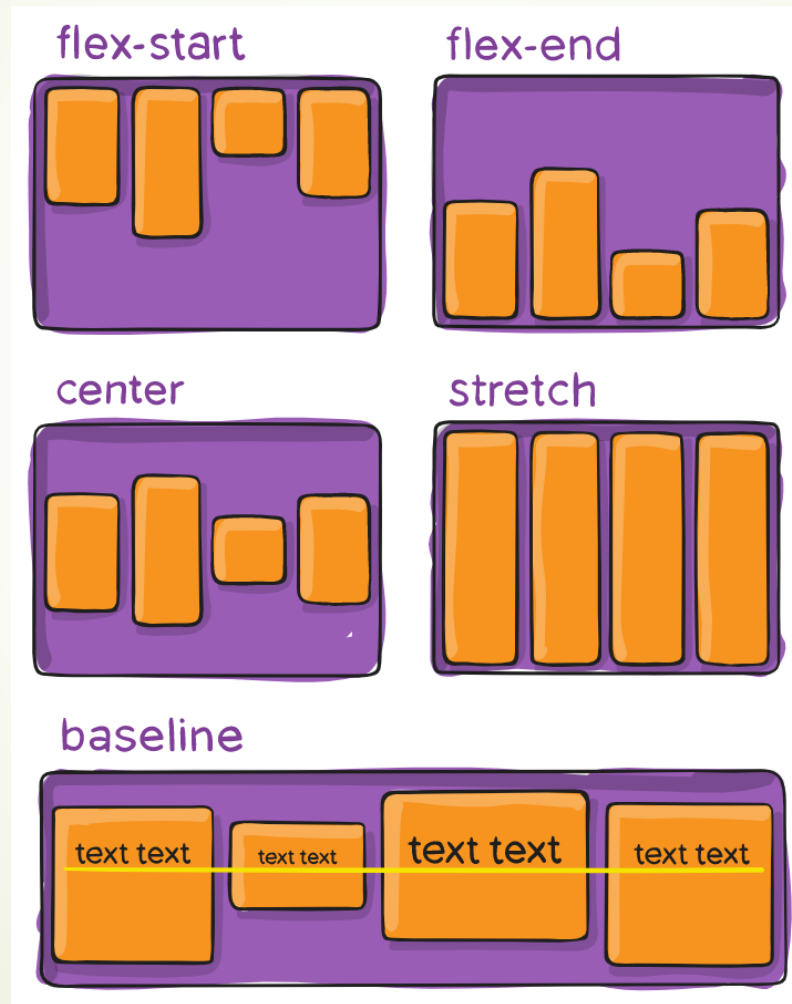


space-evenly



align-items

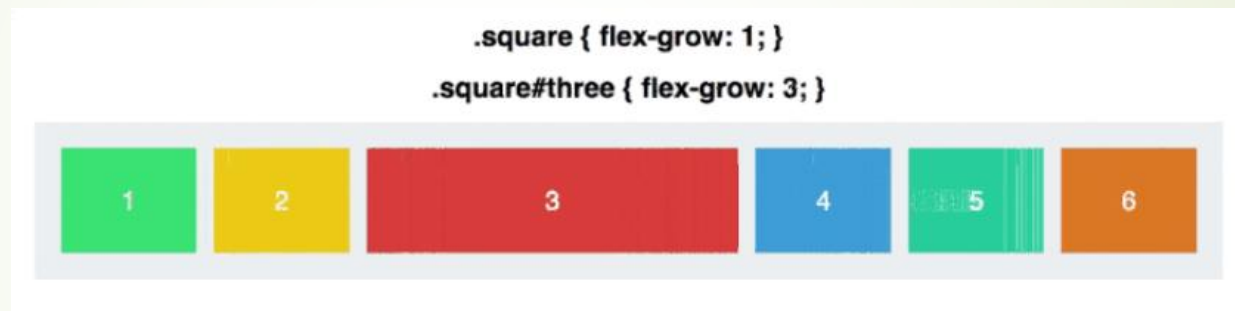
(аналогично align-self для конкретного элемента)



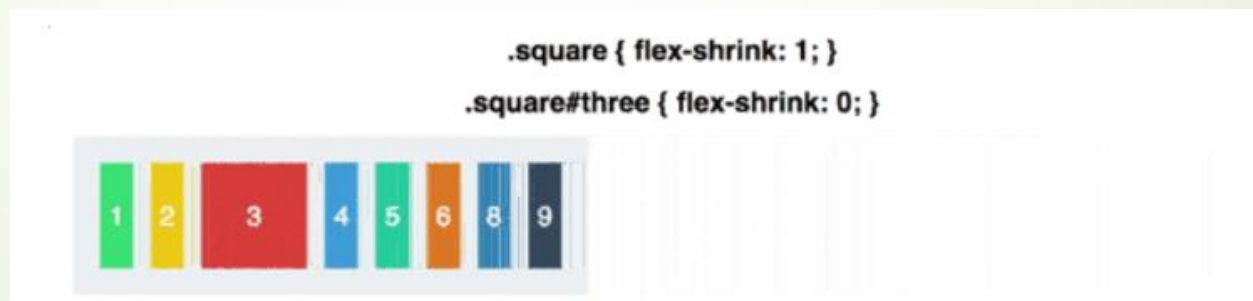
- **flex-basis** отвечает за изначальный размер элементов до того, как они будут изменены другими свойствами CSS Flexbox:



- **flex-grow** определяет на сколько блок может увеличиться в размерах



- **flex-shrink** определяет, насколько блоку можно уменьшиться в размере





Лень – двигатель прогресса!

- **flex** заменяет flex-grow, flex-shrink и flex-basis. Значения по умолчанию: 0 (grow) 1 (shrink) auto (basis):

```
.square#one {  
    flex: 2 1 300px;  
}
```

Bootstrap

- Bootstrap — это открытый и бесплатный CSS фреймворк, который используется веб-разработчиками для быстрой верстки адаптивных дизайнов сайтов и веб-приложений.

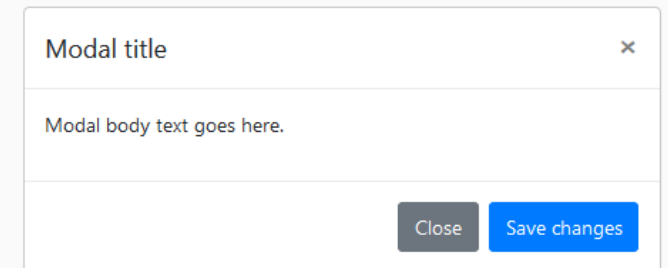


Пример элементов



```
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-secondary">Secondary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-light">Light</button>
<button type="button" class="btn btn-dark">Dark</button>

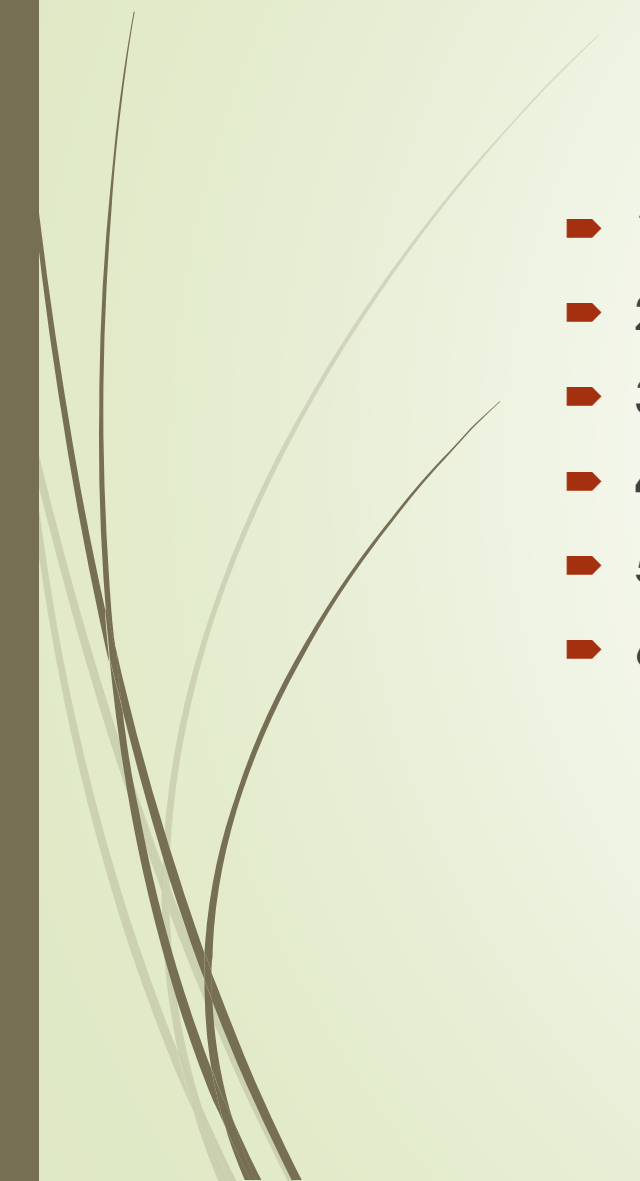
<button type="button" class="btn btn-link">Link</button>
```



```
<div class="modal" tabindex="-1" role="dialog">
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title">Modal title</h5>
        <button type="button" class="close" data-dismiss="modal" aria-label="Close">
          <span aria-hidden="true">&times;</span>
        </button>
      </div>
      <div class="modal-body">
        <p>Modal body text goes here.</p>
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-primary">Save changes</button>
        <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
      </div>
    </div>
  </div>
</div>
```



Преимущества

- 1. Уменьшение количества времени, затрачиваемого на разработку
 - 2. Адаптивность
 - 3. Кросс-браузерность
 - 4. Легкость в использовании и быстрота в освоении
 - 5. Понятный код
 - 6. Единство стилей
- 



Недостатки

- 1. Шаблонность
- 2. Дополнительная нагрузка на сеть
- 3. Использование библиотеки JQuery (в версиях до 5.0), что тормозит работу страниц

Система сеток. Контейнер

- Система сеток Bootstrap 4 использует контейнеры, ряды и колонки, чтобы удобно располагать содержимое.
- `.container` является корневым блоком сетки в Bootstrap, то есть располагается на внешнем уровне.

```
<div class="container">  
  <div class="row">  
    <div class="col">Этот текст находится внутри сетки Bootstrap</div>  
  </div>  
</div>
```

- Контейнер может показаться избыточным элементом, но это не так. Он определяет ширину макета и выравнивает его по горизонтали в области просмотра. Кроме того, контейнер нужен, чтобы контролировать отрицательные внешние отступы рядов.



Ряды

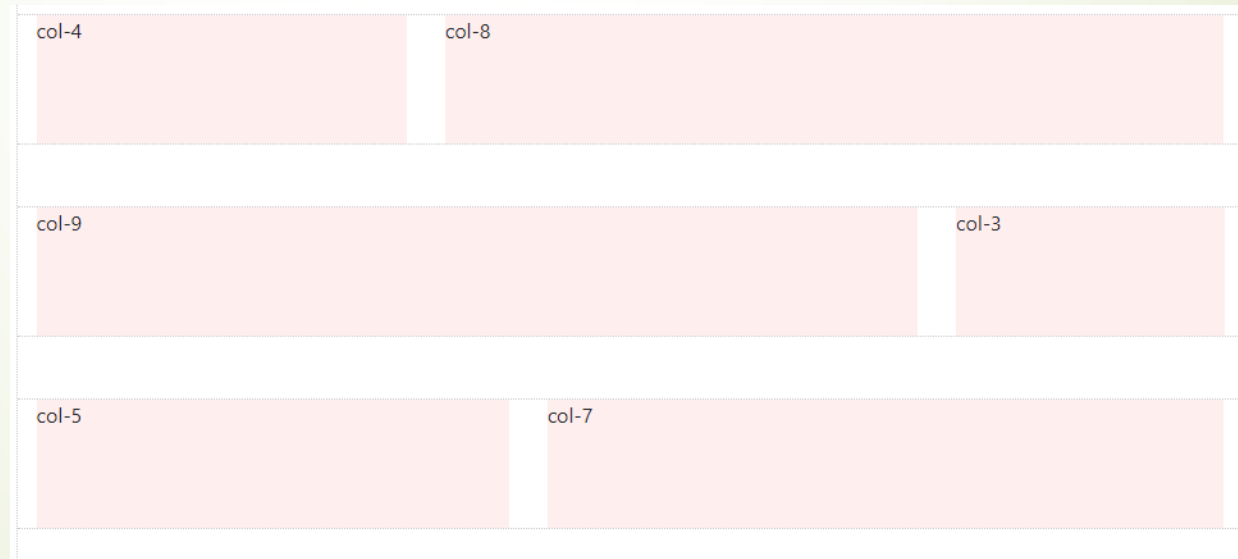


Название "ряд" (row) часто вводит в заблуждение и скрывает настоящее назначение рядов в сетке Bootstrap. Оно вызывает ошибочное представление о горизонтальной линии, однако колонки внутри ряда не всегда располагаются в одну строку. Их положение может меняться. Например, на маленьких экранах они могут выстраиваться друг под другом, а на больших – рядом. Концепция изменения макета для разных областей просмотра – это основа адаптивного дизайна.

Лучше всего думать о ряде, как о родительском элементе для группы колонок.






Колонки

- Колонки нужны для разделения области просмотра по горизонтали, при этом в одном ряду могут быть столбцы разной ширины. Их размер может изменяться в зависимости от некоторых факторов. Пространство между колонками называется "желоб" (gutter).



Адаптивность сетки

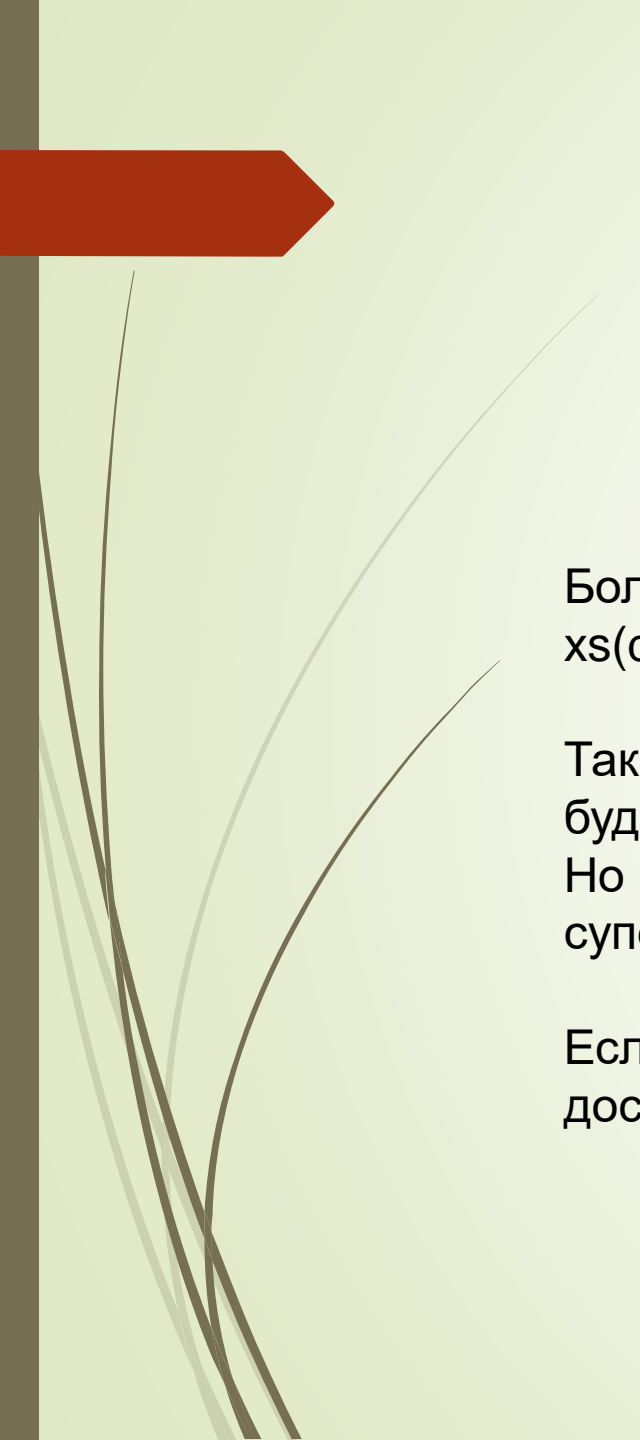
- Ширина одной и той же колонки, а также общая структура сетки может меняться для разных областей просмотра. Изменениями можно управлять с помощью специальных классов
- Фреймворк определяет 5 уровней адаптивности (брейкпоинтов), которые основаны на ширине области просмотра:

	 Extra small <576px	 Small ≥576px	 Medium ≥768px	 Large ≥992px	 Extra large ≥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				



Column 1	Column 2

Column 1
Column 2



Большие брейкпоинты переопределяют меньшие
`xs(default) > sm > md > lg > xl`

Таким образом, класс `.col-sm-6` на самом деле означает, что ширина колонки будет составлять 50% на всех экранах размера `sm` и больше.
Но класс `.col-lg-4` может переопределить это правило для больших и супербольших областей видимости.

Если требуется установить одинаковую ширину столбцов на всех уровнях, достаточно явно указать ее для наименьшего.


Автоматический расчет столбцов

► Пример 1.

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

► Пример 2.

```
<div class="container">  
  <div class="row">  
    <div class="col-sm-2">left</div>  
    <div class="col">main content</div>  
    <div class="col-auto">right</div>  
  </div>  
</div>
```



Житейские советы для недизайнеров

- Кругом только враги и идиоты
 - Использование рамок и акцентов
 - При использовании меню пользователь должен постоянно знать в каком пункте он находится
 - Правило «шести»
 - Использование двух основных контрастирующих цветов (возможен 1 дополнительный)
- 