

Bootstrap

Практически любая веб-страница содержит множество похожих компонентов, которые встречаются и на других сайтах. Это меню, навигация, элементы форм, заголовки и др. Не говоря уже про многоколоночную верстку, без которой вообще сложно превратить картинку макета в готовую веб-страницу. Чтобы меньше писать кода и по максимуму задействовать уже готовые решения применяются специализированные библиотеки, их часто называют фреймворками. Одним из таких фреймворков для верстки является Bootstrap.

Bootstrap разработали Марк Отто и Якоб Торнтон, сотрудники Twitter, именно поэтому в названии фигурирует имя компании. Их цель понятна любому разработчику — создать единый стандартный набор инструментов для сотрудников компании, ускоряющий их работу.

На сегодняшний день Bootstrap давно перерос рамки одной компании, это открытый продукт применяемый веб-разработчиками для верстки сайтов во всём мире.

Какие плюсы несёт в себе использование библиотеки.

Высокая скорость разработки

Фактически Bootstrap представляет собой конструктор, фрагменты которого вы включаете в свой проект при необходимости. Это уменьшает время разработки, потому что не требуется придумывать и писать их самостоятельно.

Адаптивный дизайн

Bootstrap направлен на создание макета под разные устройства — ноутбуки, планшеты, смартфоны. При этом код пишется один, а масштабирование в зависимости от ширины устройства берёт на себя фреймворк.

Открытая программа

Вы можете свободно изучать исходные коды библиотеки, изменять и расширять её под собственные нужды.

Совместимость с браузерами

Компоненты библиотеки написаны и протестированы с учётом работы разных браузеров. Это гарантирует, что макет будет выглядеть одинаково независимо от выбранного браузера.

Низкий порог вхождения

Чтобы использовать библиотеку в своей работе, требуется обладать минимальными знаниями по HTML, CSS и JavaScript. Это позволяет создавать эффектные сайты даже начинающим разработчикам.

Единая работа компонент

Bootstrap не просто вставляет какие-то элементы на страницу, но сразу же устанавливает их оформление и взаимодействие с пользователем через JavaScript. Вы получаете полностью работающий набор компонент, который достаточно добавить и настроить под себя.

Недостатки, конечно же, тоже имеются и они вытекают в основном из универсальности системы.

Во-первых, файлы библиотеки, даже сжатые, занимают довольно много места и увеличивают нагрузку на сервер. С повышением опыта нужно загружать и устанавливать только требуемые компоненты Bootstrap, сокращая тем самым объём файлов.

Во-вторых, предлагаемые в библиотеке стили могут не подходить под дизайн разрабатываемого сайта и придётся много переделывать. В таких случаях, как известно, проще написать всё самому с нуля.

В общем, Bootstrap годится для типовых сайтов, дизайн которых ориентирован на библиотеку. А сайтов, которые предлагают темы и шаблоны, в том числе бесплатные, в последнее время родилось довольно много. Достаточно поискать по ключевым словам «Bootstrap Theme».

Автор

Влад Мержевич

Веб-разработчик, автор нескольких книг, посвящённых созданию сайтов, HTML и CSS. Кандидат технических наук.

WebReference.ru

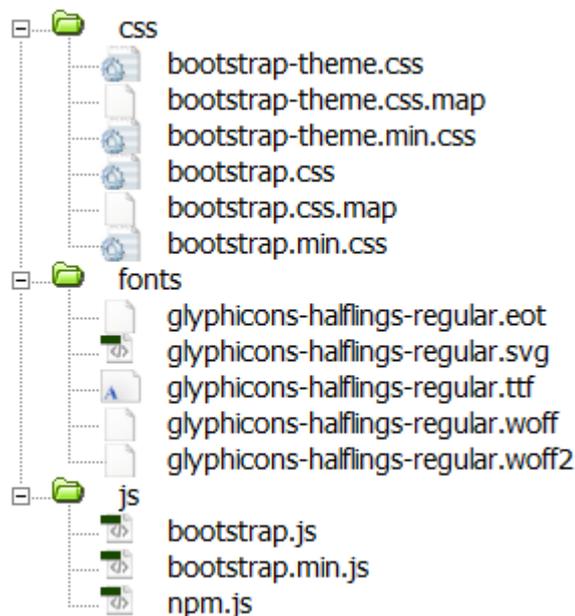
Руководство взято с сайта webref.ru.

Установка Bootstrap

Для начала необходимо скачать Bootstrap с сайта разработчика по следующему адресу:

<http://getbootstrap.com/getting-started/#download>

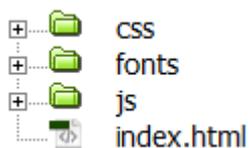
Это полная версия, содержащая все необходимые скрипты и стили. Внутри находится три папки: `css`, `fonts` и `js`.



Структура файлов Bootstrap

В папках со стилями и скриптами приложено две версии файлов — исходная и компактная (в имени содержится `min`). Компактная отличается лишь размером файла и снижением читаемости кода. Лучше всего на рабочий сайт добавлять именно эту версию, так мы чуть ускорим загрузку веб-страниц.

Копируем все папки в наш проект и в корне создаём `index.html`. В итоге структура нашего проекта будет выглядеть следующим образом.



Файлы и папки нашего проекта

Содержимое `index.html` включает в себя ссылку на стилевой файл `bootstrap.css` и внизу страницы мы вызываем `bootstrap.js`. Больше пока ничего не нужно.

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Bootstrap</title>
    <link href="css/bootstrap.css" rel="stylesheet">
  </head>
  <body>
    <h1>Привет, мир!</h1>
    <script src="js/bootstrap.js"></script>
  </body>
</html>
```

Если в процессе работы потребуется переопределить стили каких-то элементов, то вы можете подключить ещё один собственный стилевой файл и в нём задать необходимые свойства. Bootstrap также предлагает сразу настроить желаемые параметры прямо у них на сайте, отметить требуемые компоненты и скачать персонифицированную версию рабочих файлов. Это делается по следующей ссылке:

<http://getbootstrap.com/customize/>

Также в дальнейшем нам понадобится библиотека [jQuery](#), её надо скачать и добавить вызов перед bootstrap.js.

Модульные сетки

Bootstrap характерен своей продуманной и гибкой системой модульных сеток, в основе которой лежит 12-колоночный макет.

1	1	1	1	1	1	1	1	1	1	1	1
2		2		2		2		2		2	
3			3			3			3		
4				4				4			
6						6					
12											

Колонки одинаковой ширины

Сами колонки не обязательно должны иметь одинаковую ширину, можно комбинировать любое число колонок, главное, чтобы они в сумме давали 12.

2		8								2	
3			9								
4				1	3			4			

Колонки разной ширины

Создание колонок

Так, чтобы создать макет из трёх колонок используем `<div>` с классом `container`, внутри которого будут располагаться наши колонки. Сами колонки находятся внутри `<div>` с классом `row` и содержат имена вида `col-xs-N`, где `N` — число колонок от 1 до 12 (пример 1).

Пример 1. Макет с тремя колонками

```

<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Трёхколоночный макет</title>
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <style>
      [class*="col-"] {
        background-color: #eee;
        border-right: 2px solid #fff;
        text-align: center;
        padding-top: 15px;
        padding-bottom: 15px;
        font-size: 2rem;
      }
    </style>
  </head>
  <body>

```

```

<div class="container">
  <div class="row">
    <div class="col-xs-3">3 колонки</div>
    <div class="col-xs-7">7 колонок</div>
    <div class="col-xs-2">2 колонки</div>
  </div>
</div>
<script src="js/jquery.min.js"></script>
<script src="js/bootstrap.min.js"></script>
</body>
</html>

```

Стиль добавлен лишь для наглядного выделения колонок.

Класс container создаёт макет фиксированной ширины, значение которой зависит от размера устройства. Для мониторов максимальная ширина составляет 1170 пикселей, для смартфонов макет будет занимать всю доступную ширину. Если вам не требуется ограничивать ширину макета, то вместо класса container следует использовать container-fluid (пример 2).

Пример 2. Резиновый макет

```

<div class="container-fluid">
  <div class="row">
    <div class="col-xs-4">4 колонки</div>
    <div class="col-xs-6">6 колонок</div>
    <div class="col-xs-2">2 колонки</div>
  </div>
</div>

```

Отступы между колонок

Колонки изначально плотно прилегают друг к другу, что не всегда полезно для макета. Для добавления пустого пространства между колонками предназначен класс col-xs-offset-N, где N изменяется от 0 до 12. Отступ добавляется слева от текущей колонки (пример 3).

Пример 3. Добавление отступов

```

<div class="container">
  <div class="row">
    <div class="col-xs-3">3 колонки</div>
    <div class="col-xs-5 col-xs-offset-1">5 колонок</div>
    <div class="col-xs-2 col-xs-offset-1">2 колонки</div>
  </div>
</div>

```

Результат данного примера в браузере показан на рис. 3.

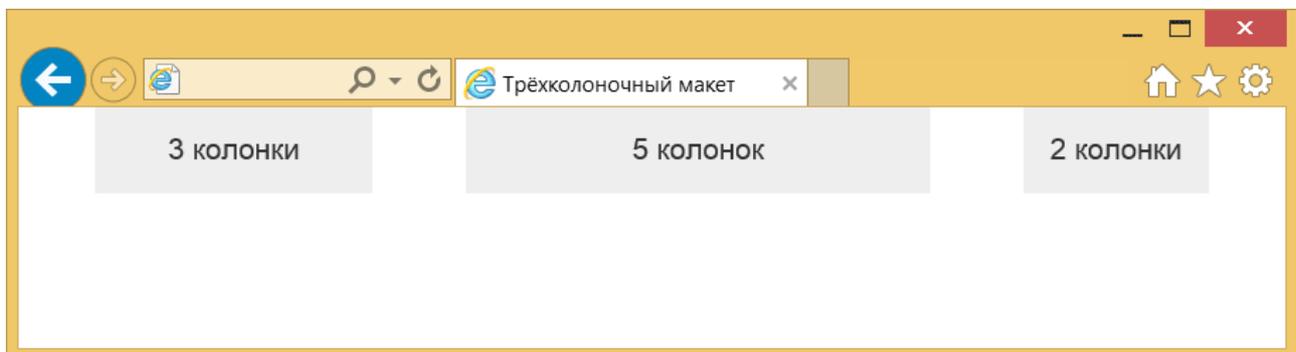


Рис. 3. Колонки с отступами между ними

Помните, что отступы добавляются к общему числу колонок, сумма которых не должна превышать 12,

в противном случае колонки начнут перемещаться на другую строку.

Сдвиг колонок

Каждую колонку можно сдвигать влево или вправо на указанное число колонок. Опять же это делается с помощью классов. `col-xs-pull-N` — сдвигает колонку влево на заданное число, а `col-xs-push-N` сдвигает вправо (пример 4). Здесь N может меняться от 0 до 12.

Пример 4. Сдвиг колонок вправо

```
<div class="container">
  <div class="row">
    <div class="col-xs-3">3 колонки</div>
    <div class="col-xs-5 col-xs-push-1">5 колонок</div>
    <div class="col-xs-2 col-xs-push-2">2 колонки</div>
  </div>
</div>
```

Заметьте, что сдвиг это не дополнительная колонка как при использовании `offset`, так что суммировать все значения не нужно. Ещё надо следить за тем, чтобы колонки не накладывались друг на друга.

Вложенные колонки

При вёрстке сложных макетов двенадцати колонок может не хватить, к тому же в одной колонке могут встречаться ещё дополнительные. Так что нам потребуются вложения одних колонок в другие. Делается это похожим образом (пример 5).

Пример 5. Вложенные колонки

```
<div class="container">
  <div class="row">
    <div class="col-xs-12">Заголовок</div>
    <div class="col-xs-5">
      <div class="row">
        <div class="col-xs-12">Колонка 1</div>
        <div class="col-xs-6">Колонка 1-1</div>
        <div class="col-xs-6">Колонка 1-2</div>
      </div>
    </div>
    <div class="col-xs-5 col-xs-offset-2">Колонка 2</div>
  </div>
</div>
```

Чтобы создать вложенные колонки опять добавляем `<div>` с классом `row`, который содержит желаемую структуру вложенных колонок. Таким образом можно сверстать какие угодно сложные макеты.

Заметьте, что заголовок заполняет всю ширину макета, потому что он занимает все 12 колонок. Дополнительный контейнер `row` для подобных вещей вводить не обязательно, перенос остальных колонок на другую строку произойдет автоматически (рис. 4).

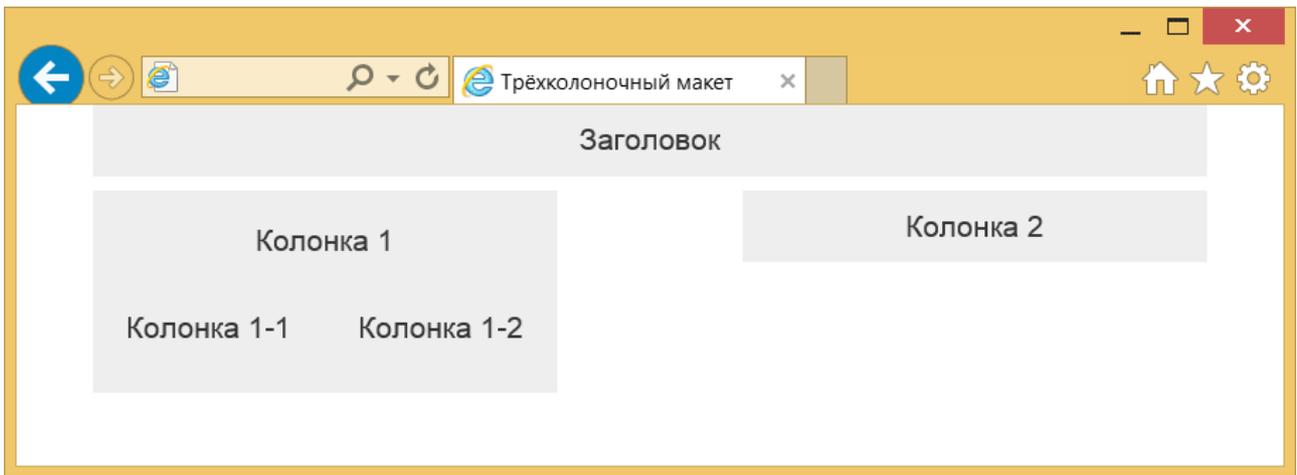


Рис. 4. Вложенные колонки

Адаптивный дизайн

Адаптивным дизайном называется способ вёрстки, когда ширина макета веб-страницы подстраивается под ширину устройства. Самый простой вариант — это пропорциональное изменение масштаба, но на деле лучше менять и расположение элементов и их размеры. Так мы получим наиболее оптимальный вид для смартфонов, планшетов и мониторов.

В Bootstrap уже заложены возможности адаптивного дизайна. Если вы сделаете простой многоколоночный макет, а затем начнёте уменьшать ширину окна браузера, то заметите, что изменяется и сам макет. Но этим всё не ограничивается, гораздо интереснее немного менять сам макет чтобы он наилучшим образом соответствовал устройству. Для этого введены ключевые слова, которые встречаются в именах классов, они представлены в табл. 1.

Табл. 1. Ключевые слова

Устройство	Имя	Ширина макета
Смартфон	xs	Менее 768px
Планшет	sm	768px и больше
Монитор	md	992px и больше
Монитор с высоким разрешением	lg	1200px и больше

Чтобы задать ширину колонок для смартфонов достаточно в код включить класс col-xs-N, для мониторов он уже будет именоваться col-md-N. Любые классы можно комбинировать между собой, если класс для выбранного устройства не указан, то он наследуется снизу вверх. Это значит, что макет для смартфона будет выглядеть так же, как и макет для монитора. Но не наоборот. Именно поэтому вёрстка всегда начинается с макета для смартфона, затем уже идёт планшет и монитор.

Поскольку мы теперь ориентируемся на мобильные устройства, то внутри <head> надо добавить следующую строку.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Теперь наш дизайн будет масштабироваться по ширине устройства и выглядеть оптимально, особенно на устройствах с маленьким экраном.

С учётом этой строки и новых классов сделаем простой макет с двумя колонками (пример 1).

Пример 1. Адаптивный макет

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Адаптивный макет</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <style>
      [class*="col-"] {
        background-color: #eee;
        text-align: center;
        padding-top: 10px;
        padding-bottom: 10px;
        margin-bottom: 10px;
        font-size: 2rem;
      }
    </style>
  </head>
  <body>
    <div class="container">
      <div class="row">
        <div class="col-xs-12">Заголовок</div>
```

```
<div class="col-xs-12 col-sm-8 col-md-8">Колонка 1</div>
<div class="col-xs-12 col-sm-4 col-md-3 col-md-offset-1">Колонка 2</div>
</div>
</div>
<script src="js/jquery.min.js"></script>
<script src="js/bootstrap.min.js"></script>
</body>
</html>
```

Для тестирования макета не обязательно смотреть его на каком-то заданном устройстве, достаточно уменьшить ширину окна браузера и мы сразу же увидим изменения. На рис. 1-3 показан результат данного примера при разной ширине окна.

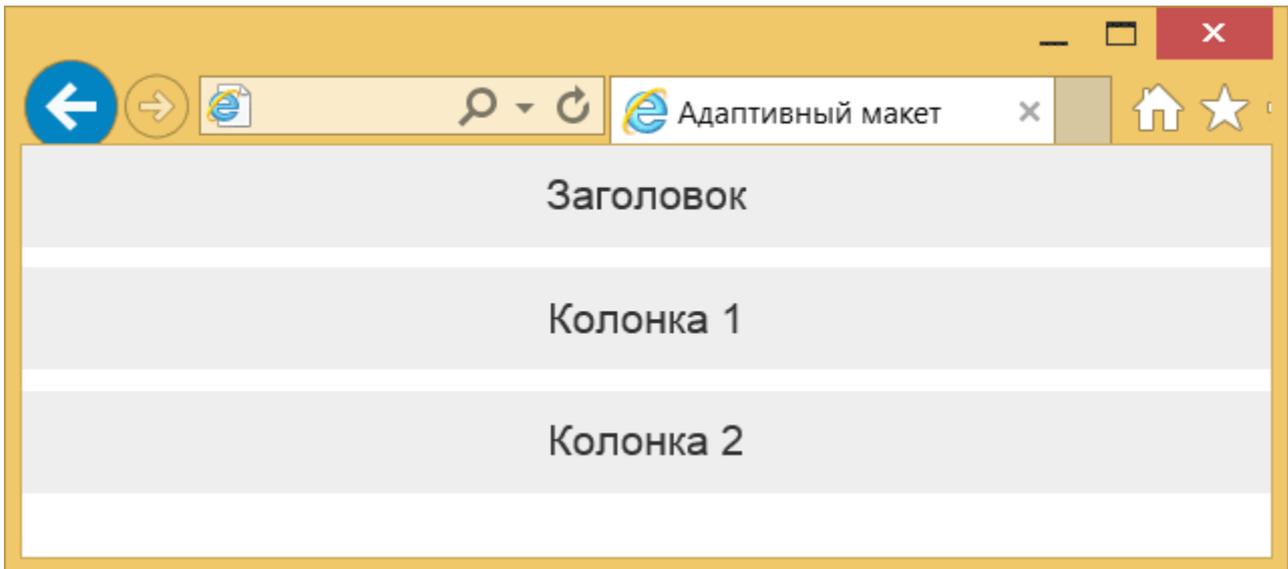


Рис. 1. Вид на смартфоне

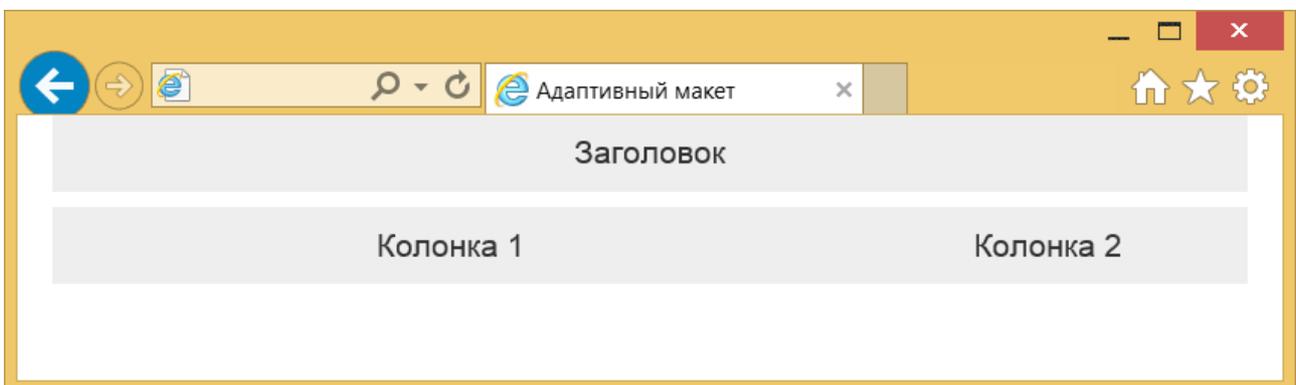


Рис. 2. Вид на планшете

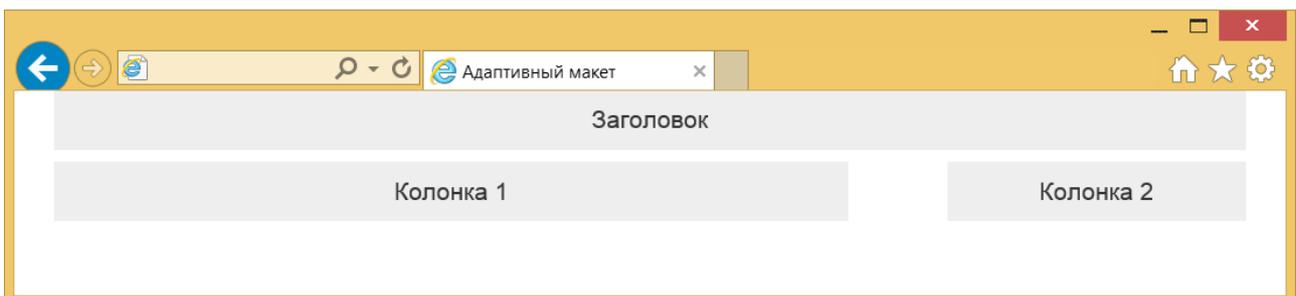


Рис. 3. Вид на мониторе

Мы рассмотрели самый простой случай, когда расположение элементов практически не меняется

относительно друг друга. Как быть, если потребуется переставить элементы местами или кое-что убрать для маленьких экранов? Напрямую это сделать не получится, поэтому пойдём на небольшую хитрость — добавим два одинаковых элемента в разных местах и будем один прятать, а другой отображать. Bootstrap предлагает два набора классов для сокрытия и показа элементов в зависимости от ширины окна — `hidden-xs` и `visible-xs-block`. Принцип тот же, что и при работе с колонками, вместо `xs` подставляем нужное ключевое слово. В примере 2 показано как «перемещать» таким образом заголовок.

Пример 2. Изменение положения заголовка

```
<div class="container">
  <div class="row">
    <div class="col-xs-12 hidden-xs">Заголовок</div>
    <div class="col-xs-12 col-sm-8 col-md-8">Колонка 1</div>
    <div class="col-xs-12 col-sm-4 col-md-3 col-md-offset-1">Колонка 2</div>
    <div class="col-xs-12 visible-xs-block">Заголовок</div>
  </div>
</div>
```

Результат данного примера показан на рис. 4.

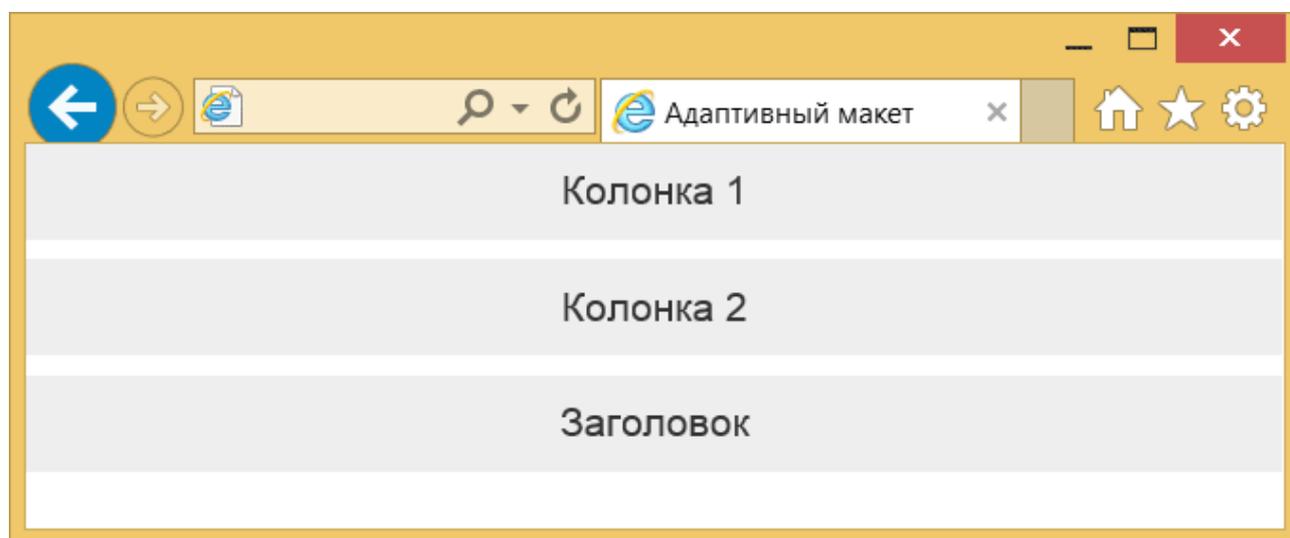


Рис. 4. Заголовок внизу колонок при узкой ширине окна

В данном примере класс `hidden-xs` означает, что колонку для смартфонов следует прятать, а `visible-xs-block` — наоборот, показывать. При просмотре на широком экране заголовок будет всегда вверху, а на узком внизу.

Стили

Bootstrap содержит множество готовых стилей для оформления элементов. Применять их достаточно просто, всего то надо добавить к элементу определённый в стилях класс. Это сокращает время на разработку проекта, поскольку не приходится задумываться о дизайне отдельных элементов. Стиль некоторых элементов, вроде заголовков `<h1>` также отличается от исходного, это надо учитывать.

Цвет и фон

Для веб-страницы устанавливается белый цвет фона, набор шрифтов Helvetica Neue, Helvetica, Arial и цвет текста #333. Вы также можете использовать заданный набор цветов для привлечения внимания к тексту и создания информационных сообщений (пример 1).

Пример 1. Цвета текста

```
<p class="text-muted">Серый цвет</p>
<p class="text-primary">Основной текст</p>
<p class="text-success">Цвет для сообщения об успехе</p>
<p class="text-info">Цвет для информации</p>
<p class="text-warning">Цвет для предупреждений</p>
<p class="text-danger">Цвет для сообщений об ошибках</p>
```

Результат применения этих классов показан на рис. 1.

Серый цвет
Основной текст
Цвет для сообщения об успехе
Цвет для информации
Цвет для предупреждений
Цвет для сообщений об ошибках

Рис. 1. Цвета для разных сообщений

Подобным же образом можно менять цвет фона у текста или другого элемента, например, кнопок. Немного поменяются имена классов, но принцип сохранится (пример 2).

Пример 2. Фон элементов

```
<p class="bg-primary">Основной текст</p>
<p class="bg-success">Цвет для сообщения об успехе</p>
<p class="bg-info">Цвет для информации</p>
<p class="bg-warning">Цвет для предупреждений</p>
<p class="bg-danger">Цвет для сообщений об ошибках</p>
```

Результат применения этих классов показан на рис. 2.

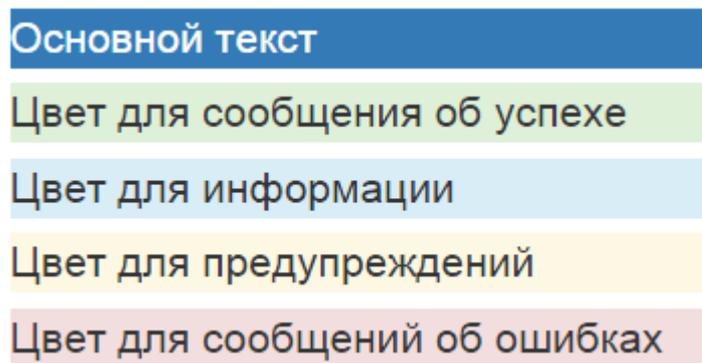


Рис. 2. Фон для разных сообщений

Врезки

Для создания врезок предназначены классы `pull-left` и `pull-right`, которые выравнивают элемент, соответственно, по левому и правому краям. Врезкой называется блок с рисунками и текстом, который встраивается в основной текст. Врезка обычно располагается по левому или правому краю текстового блока, а основной текст обтекает её с других сторон (пример 3).

Пример 3. Создание врезки

```
<h1>Борщ</h1>
<div class="pull-right bg-info" style="width: 20em; padding: 10px;">
  <h2><small>Хозяйке на заметку</small></h2>
  <p>Борщ получится вкуснее, если добавить в него немного соли.</p>
</div>
<p>Мясо отварить до готовности. Промыть свеклу, очистить,
нарезать соломкой и тушить с помидорами до полуготовности.</p>
<p>Бульон процедить, мясо нарезать кусочками. В бульон добавить
нарезанный дольками картофель, довести до кипения, опустить нарезанную
соломкой свежую капусту и варить 10-15 минут, добавить пассерованные
овощи, болгарский перец, нашинкованный тонкой соломкой,
специи и довести до готовности.</p>
<p>Готовому борщу дать настояться в течение 20-25 минут.
При подаче к столу добавить сметану, мясо, зелень.</p>
```

Результат данного примера показан на рис. 3.

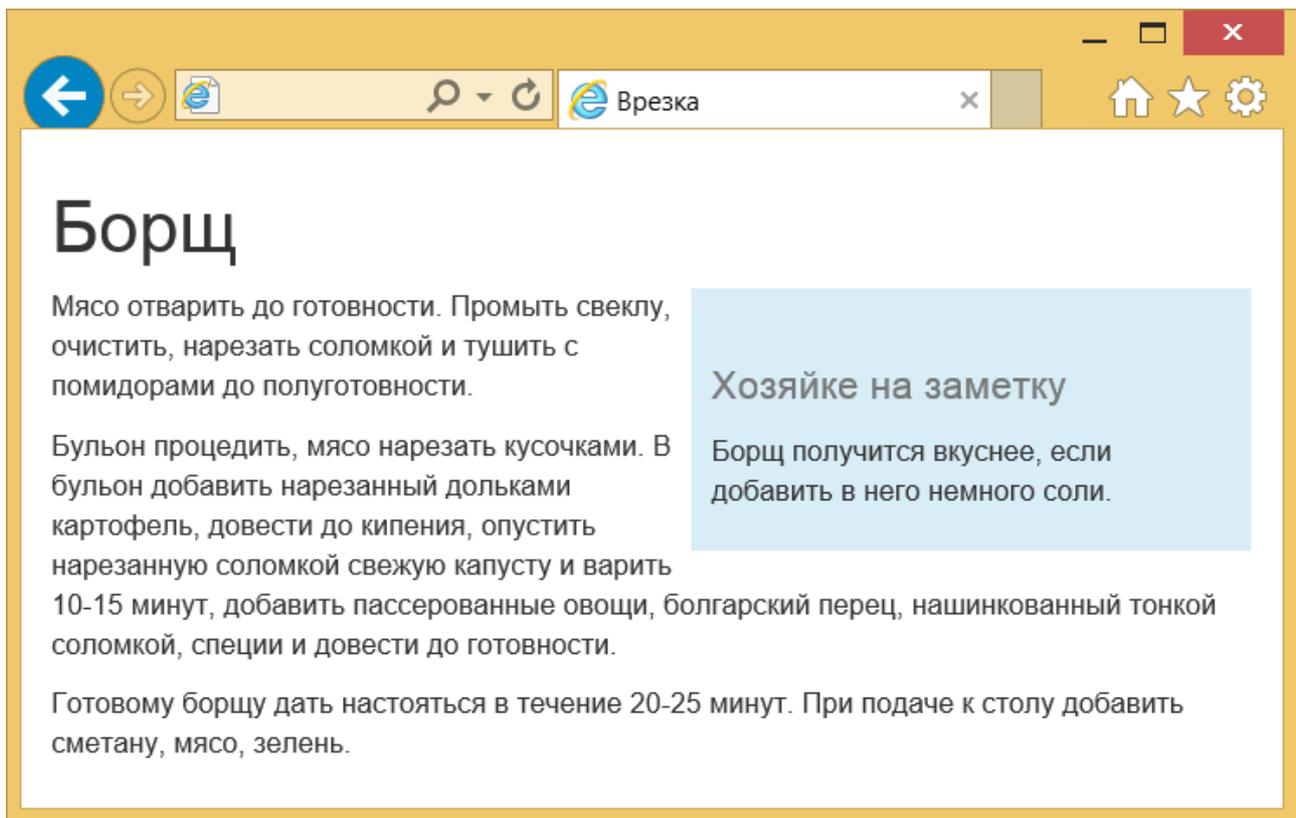


Рис. 3. Вид врезки в тексте

Чтобы наша врезка была более заметна, к `<div>` добавлен класс `bg-info`, а также ограничена его ширина. Само обтекание в стилях делается с помощью свойства `float`, которое продолжает воздействовать и на нижние элементы. Для отмены действия `float` применяется класс `clearfix`, он добавляется к любому нижележащему элементу.

См. также

Документация по использованию классов в Bootstrap

<http://getbootstrap.com/css/>

Текст

По умолчанию размер шрифта основного текста установлен как 14px, а межстрочное расстояние (значение свойства line-height) — 1.428. Между абзацами текста (элемент <p>) введено дополнительное расстояние в 10 пикселей.

Заголовки

Для заголовков применяются знакомые из HTML элементы с <h1> по <h6>. Их стиль немного отличается от исходного (рис. 1).

Заголовок первого уровня

Заголовок второго уровня

Заголовок третьего уровня

Заголовок четвёртого уровня

Заголовок пятого уровня

Заголовок шестого уровня

Рис. 1. Вид заголовков текста

Заголовки допустимо немного уменьшить самостоятельно, если внутрь вставить элемент <small>, получится такая комбинация.

```
<h1><small>Заголовок первого уровня</small></h1>
```

Такого типа заголовки по виду несколько иные и предназначены для других разделов сайта, например, боковой панели (рис. 2).

Заголовок первого уровня

Заголовок второго уровня

Заголовок третьего уровня

Заголовок четвёртого уровня

Заголовок пятого уровня

Заголовок шестого уровня

Рис. 2. Уменьшенные заголовки

Лид

Лидом в журналистике называют первый абзац статьи привлекающий внимание и дающий краткое представление о содержании статьи. Обычно он по своему виду немного отличается от основного текста. Чтобы указать лид достаточно к абзацу или колонке добавить класс lead, как показано в примере 1.

Пример 1. Создание лида

```
<p class="lead">Дорогой друг! Я расскажу тебе удивительную историю про маленького червячка, который жил в яблоке, о его путешествии в волшебную страну Нортландия, о том, какие опасности подстерегали его на пути, кого он повстречал по дороге, какие приключения и испытания прошёл, и что он рассказал по возвращению домой.</p>  
<p>Итак, сказка про маленького червячка, который жил в яблоке, о его путешествии в волшебную страну Нортландия, о том, какие опасности подстерегали его на пути, кого он повстречал по дороге, какие приключения и испытания прошёл, и что он рассказал по возвращению домой.</p>
```

Результат данного примера показан на рис. 3.



Рис. 3. Вид лида в браузере

Аббревиатура

Аббревиатура это сложносокращённое слово (колхоз) или слово, полученное из первых букв слов предложения (КГБ). Для обозначения аббревиатур в тексте применяется элемент <abbr>, а для расшифровки внутри него добавляется атрибут title (пример 2).

Пример 2. Аббревиатура

```
<p><abbr title="Cascading Style Sheets, Каскадные таблицы стилей">CSS</abbr> – набор параметров форматирования, который применяется к элементам документа, чтобы изменить их внешний вид. </p>
```

Аббревиатура обозначается точечным подчёркиванием, при наведении на слово курсора мыши он меняет свой вид на подсказывающий и появляется всплывающая подсказка, если она есть.

При добавлении класса `initialism` к элементу `<abbr>` для него устанавливается свойство `font-size` со значением `90%`, тем самым аббревиатура выводится уменьшенного размера.

```
<abbr title="Cascading Style Sheets, каскадные таблицы стилей" class="initialism">CSS</abbr>
```

Цитаты

Для оформления больших цитат в HTML применяется элемент `<blockquote>`, но Bootstrap немного меняет его оформление (рис. 4). Слева от цитаты отображается серая вертикальная линия, а имя автора или источник цитаты выводится уменьшенным размером и серого цвета.

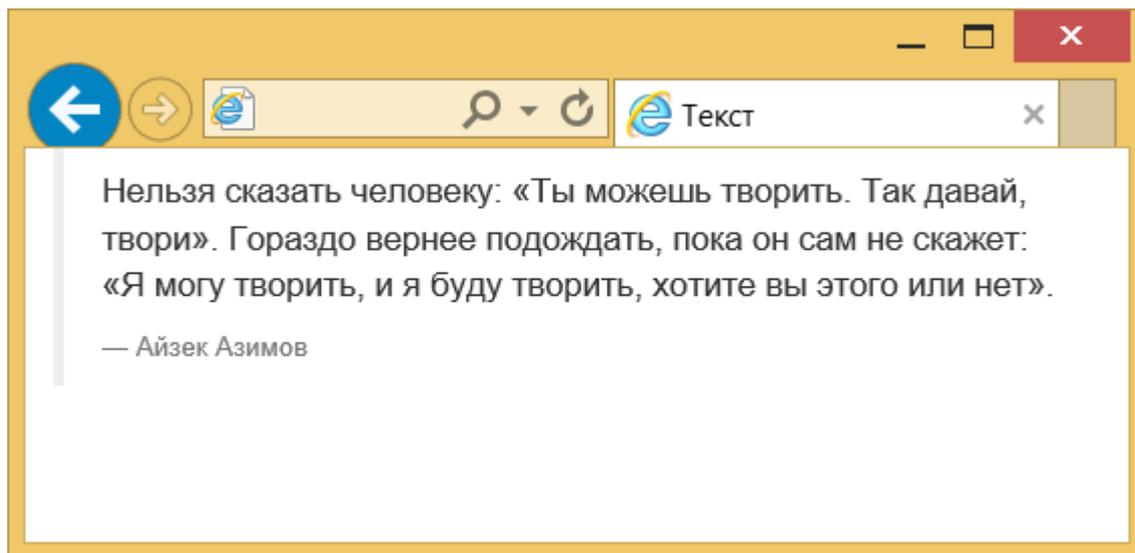


Рис. 4. Вид цитаты в браузере

Если сама цитата делается весьма традиционно, то подпись к ней добавляется с помощью элемента `<small>` или `<footer>`, как показано в примере 3.

Пример 3. Добавление цитаты

```
<blockquote>
  <p>Нельзя сказать человеку: «Ты можешь творить.
  Так давай, твори». Гораздо вернее подождать, пока он
  сам не скажет: «Я могу творить, и я буду творить, хотите
  вы этого или нет».</p>
  <p><small>Айзек Азимов</small></p>
</blockquote>
```

Цитату можно инвертировать, тогда линия будет справа, а текст выравниваться также по правому краю (рис. 5).

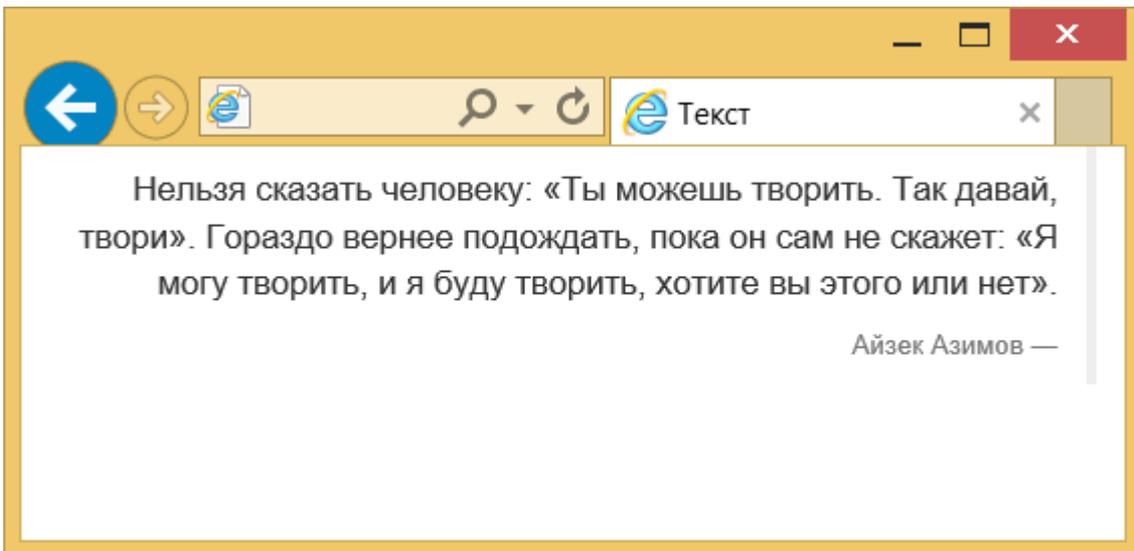


Рис. 5. Цитата, выравненная по правому краю

Просто добавьте класс `pull-right` к `<blockquote>`.

```
<blockquote class="pull-right">...</blockquote>
```

Листинг

Нам часто приходится выкладывать листинги программ и указывать переменные. В HTML для этого есть несколько готовых элементов — `<var>` обозначает отдельные переменные, `<code>` — строчный фрагмент кода, а `<pre>` используется для многостраничного кода (пример 4).

Пример 4. Вывод программы

```
<p>Функция checkParent() вызывается  
с параметрами src и dest.</p>  
<pre>function checkParent (src, dest) {  
  while (src != null) {  
    if (src.tagName == dest)  
      src = src.parentElement  
  }  
  return null  
</pre>
```

Результат данного примера показан на рис. 6.

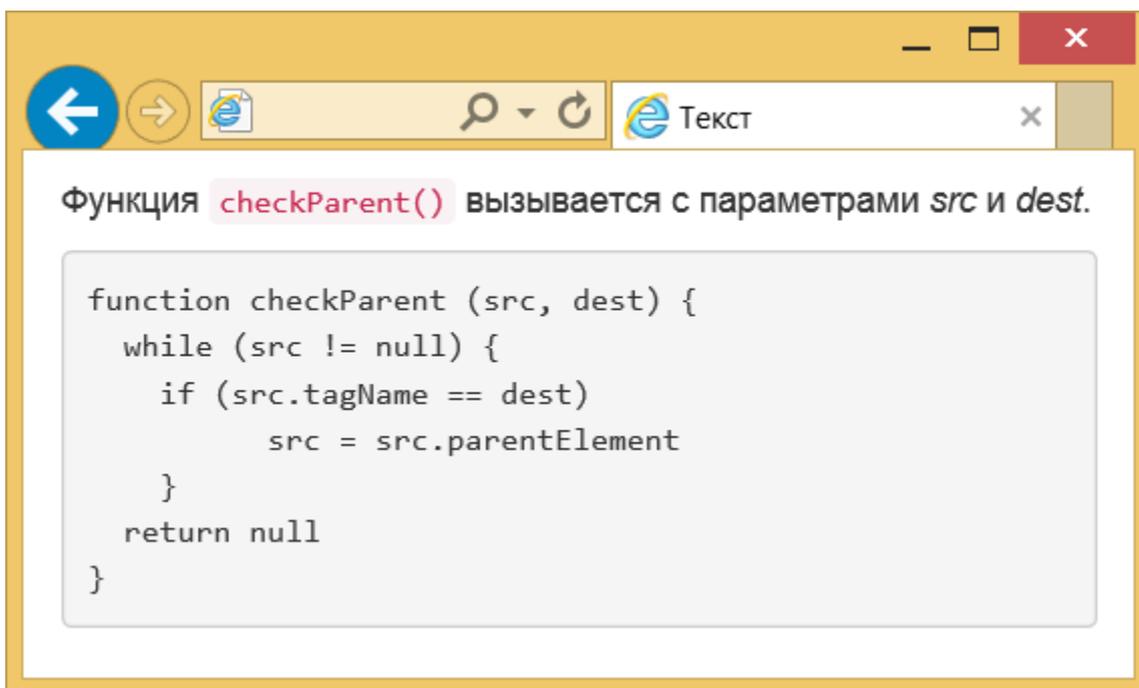


Рис. 6. Вид кода программы в браузере

Элемент `<pre>` занимает высоту равную его содержимому, что для больших листингов может быть неудобно. Чтобы ограничить высоту достаточно добавить класс `pre-scrollable`, тогда высота будет равна 350 пикселей и появится вертикальная полоса прокрутки.

Изображения

Для оформления изображений есть несколько классов — `img-rounded`, `img-circle` и `img-thumbnail`. Чтобы изменить стиль отображения картинок просто добавьте один из этих классов к элементу ``. На рис. 1 показаны результаты применения каждого класса.



а. Оригинальное изображение



б. `img-rounded`



в. `img-circle`



г. `img-thumbnail`

Рис. 1. Изменение стиля изображения

Класс `img-rounded` скругляет уголки у картинки, `img-circle` делает её круглой, а `img-thumbnail` добавляет рамку вокруг изображения.

В примере 1 показано, как вывести фотографию и задать её стиль.

Пример 1. Круглая картинка

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Изображения</title>
    <link href="css/bootstrap.min.css" rel="stylesheet">
  </head>
  <body>
    <div class="container">
      <div class="row">
```

```
<div class="col-xs-4"></div>
<div class="col-xs-8">Настоящий боевой котик. Почему боевой, спросите вы?
  Так ведь только опасных животных держат на привязи,
  чтобы они никому случаем не навредили.</div>
</div>
</div>
<script src="js/jquery.min.js"></script>
<script src="js/bootstrap.min.js"></script>
</body>
</html>
```

Любые выводимые изображения имеют исходные размеры, но для адаптивного дизайна лучше, когда ширина картинка подстраивается под ширину колонки. Для этого к элементу `` достаточно добавить класс `img-responsive`, как показано в примере выше. Заметьте, что классы для адаптивного размера и стиля изображения вполне сочетаются между собой. В результате при уменьшении размера окна мы увидим и уменьшение самой картинки (рис. 2).

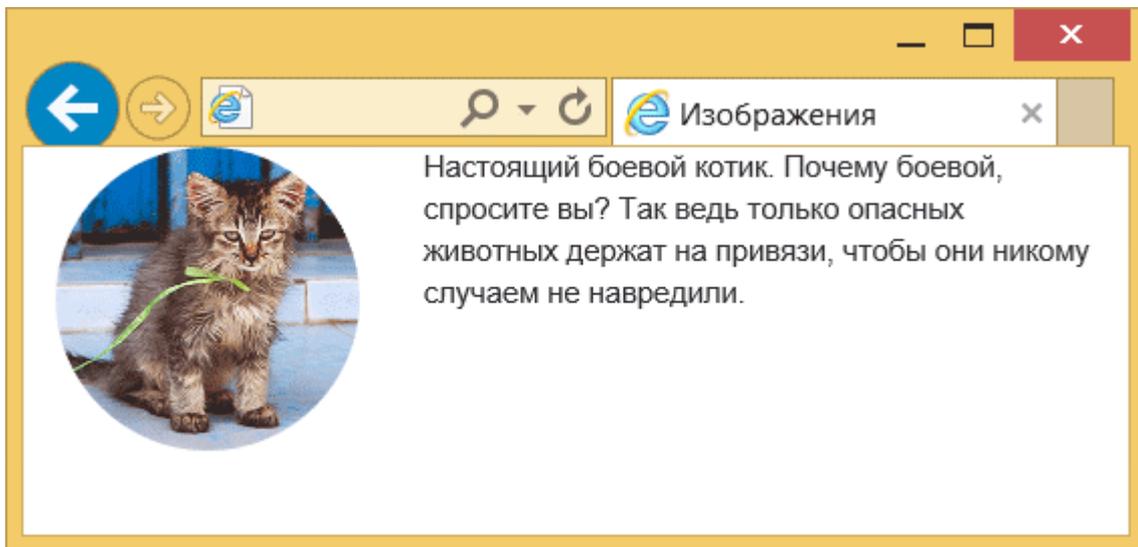


Рис. 2. Вид фотографии при уменьшении размеров окна

Списки

На сайтах списки применяются как по прямому назначению — для вывода именно списков, так и для создания различной навигации. Это связано с тем, что сами элементы списков имеют строгую иерархию и позволяют делать вложенную структуру, вдобавок видом списков легко управлять через стили.

Горизонтальный список

Такой список обычно используется для создания меню, к элементу `` для этого нужно только добавить класс `list-inline` (пример 1).

Пример 1. Горизонтальное меню

```
<ul class="list-inline">
  <li><a href=#>Сепульки</a></li>
  <li><a href=#>Сепулькирии</a></li>
  <li><a href=#>Сепуление</a></li>
</ul>
```

Вот как выглядит подобное меню (рис. 1). Стиль средней ссылки меняется при наведении на неё курсора.

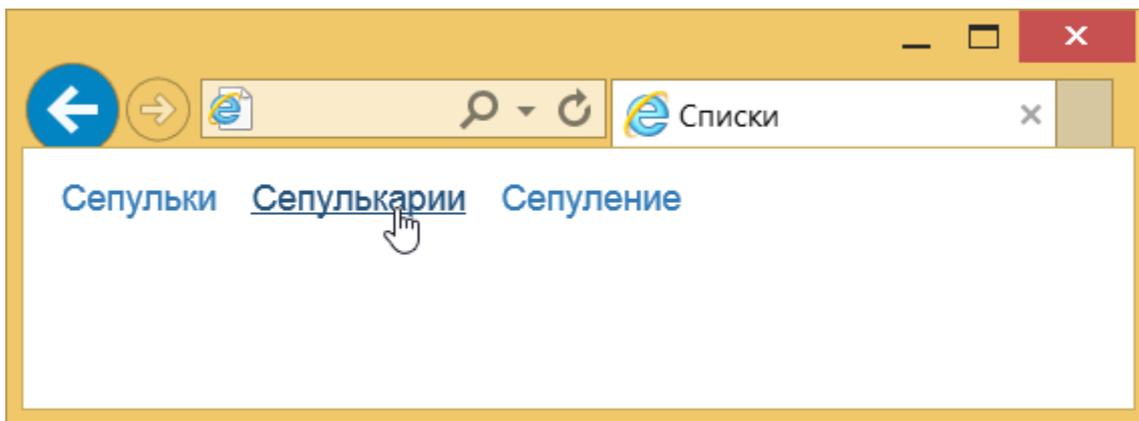


Рис. 1. Горизонтальный список

Список определений

Список определений состоит из двух элементов — термина и его определения. Сам список создаётся с помощью контейнера `<dl>`, термин — элементом `<dt>`, а его определение — с помощью `<dd>` (пример 2).

Пример 2. Список определений

```
<dl>
  <dt>Сепульки</dt>
  <dd>Важный элемент цивилизации ардритов с планеты Энтеропия.</dd>
  <dt>Сепулькирии</dt>
  <dd>Устройства для сепуления.</dd>
  <dt>Сепуление</dt>
  <dd>Занятие ардритов с планеты Энтеропия.</dd>
</dl>
```

Bootstrap предлагает сделать такой список нагляднее за счёт горизонтального расположения термина и определения. К элементу `<dl>` достаточно добавить класс `dl-horizontal` (пример 3).

Пример 3. Горизонтальный список определений

```
<dl class="dl-horizontal">
  <dt>Сепульки</dt>
  <dd>Важный элемент цивилизации ардритов с планеты Энтеропия.</dd>
  <dt>Сепулькарии</dt>
  <dd>Устройства для сепуления.</dd>
  <dt>Сепуление</dt>
  <dd>Занятие ардритов с планеты Энтеропия.</dd>
</dl>
```

Результат данного примера показан на рис. 2.

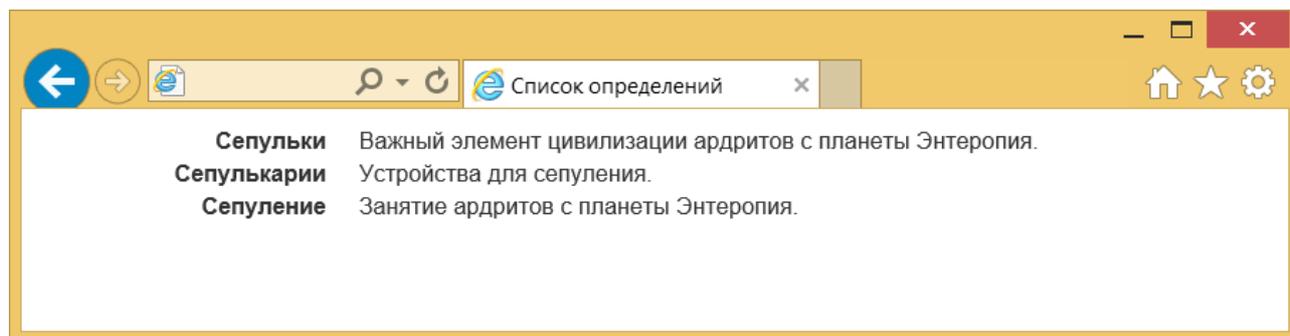


Рис. 2. Горизонтальный список определений

Если термин слишком длинный, то он будет обрезан и к нему добавлено многоточие. Также при уменьшении размеров окна вид списка меняется на традиционный (рис. 3).

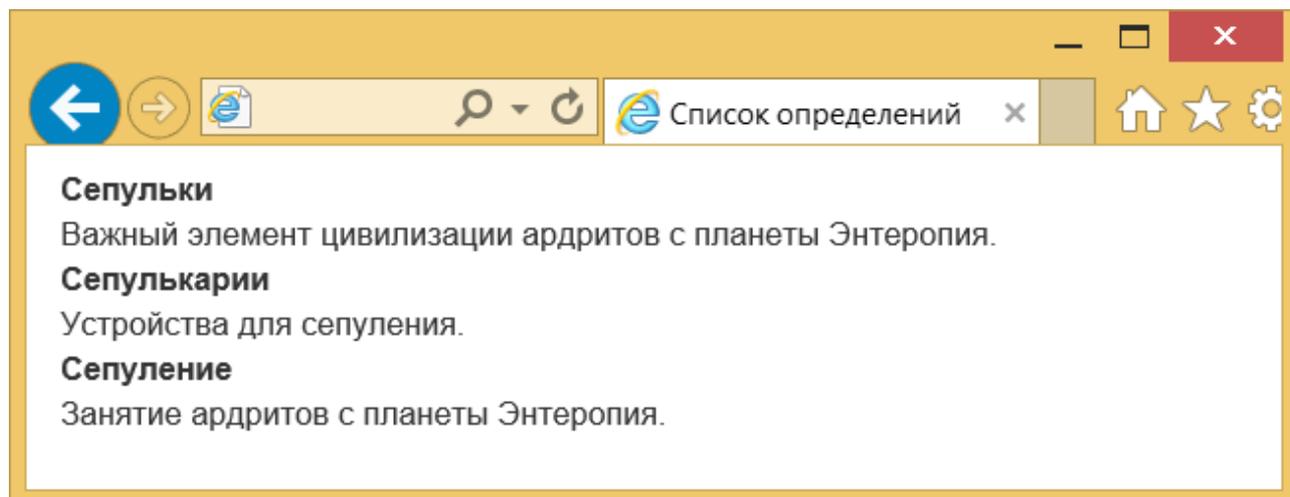
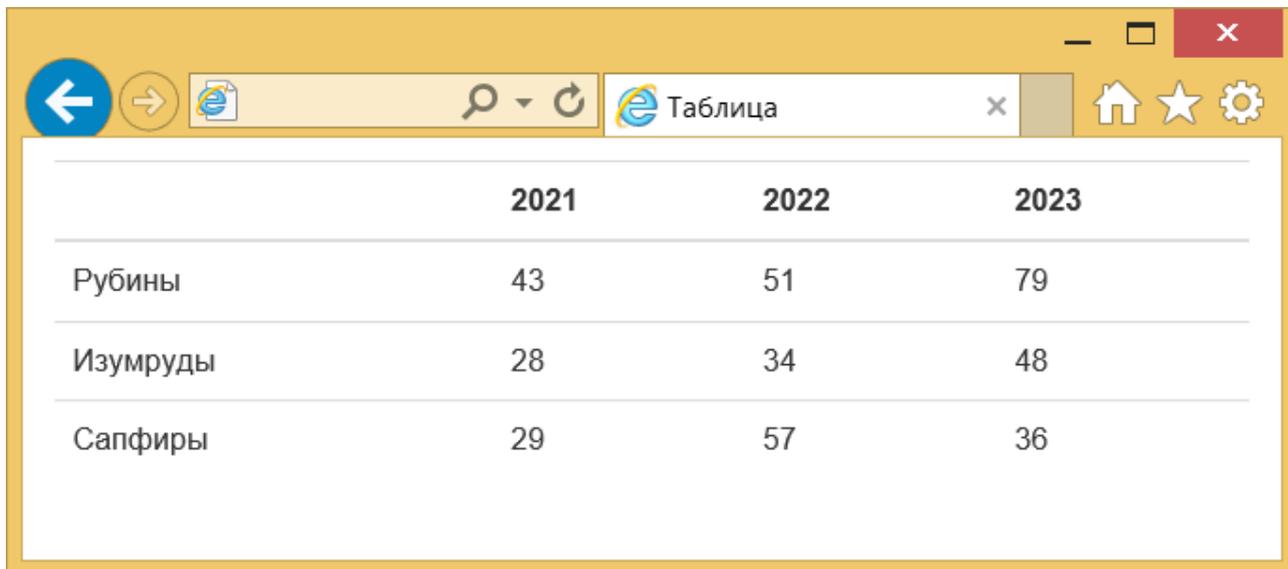


Рис. 3. Список определений

Таблицы

Для базового оформления таблиц в элемент `<table>` достаточно включить класс `table`. Вид оформленной через стили таблицы показан на рис. 1.



The screenshot shows a browser window with a yellow title bar and a navigation bar. The address bar contains the text "Таблица". The main content area displays a table with the following data:

	2021	2022	2023
Рубины	43	51	79
Изумруды	28	34	48
Сапфиры	29	57	36

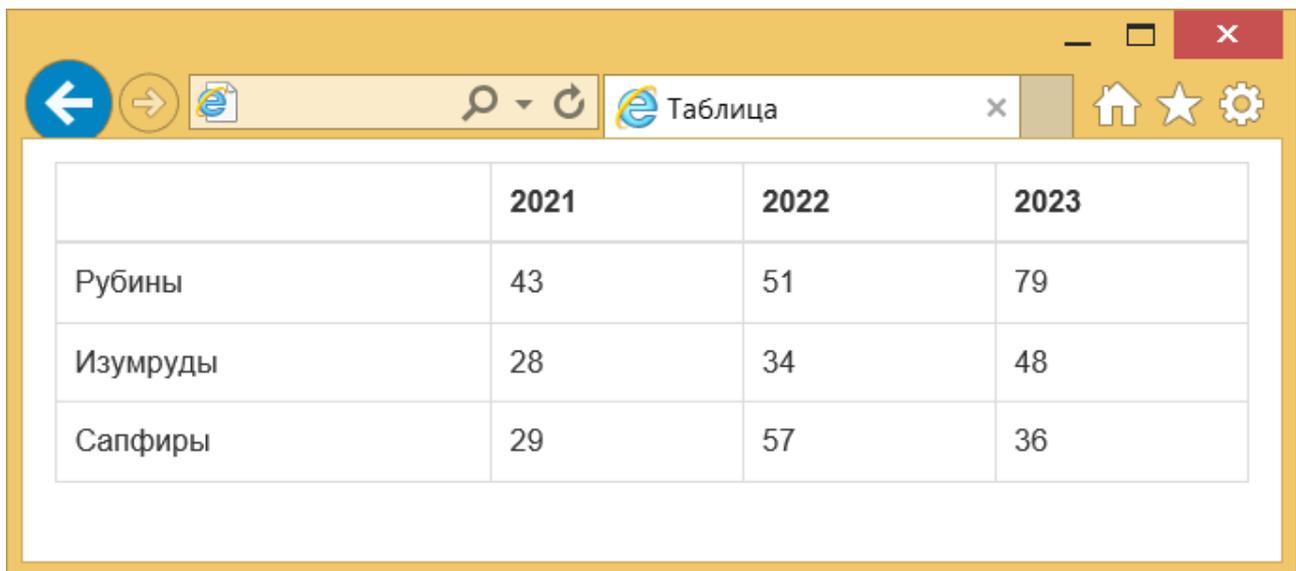
Рис. 1. Таблица

Рамки вокруг таблицы нет, строки таблицы отделяются друг от друга линиями. Чтобы таблица представляла собой сетку добавьте класс `table-bordered` (пример 1).

Пример 1. Сетка

```
<table class="table table-bordered">
  <thead>
    <tr>
      <th></th><th>2021</th><th>2022</th><th>2023</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Рубины</td><td>43</td><td>51</td><td>79</td>
    </tr>
    <tr>
      <td>Изумруды</td><td>28</td><td>34</td><td>48</td>
    </tr>
    <tr>
      <td>Сапфиры</td><td>29</td><td>57</td><td>36</td>
    </tr>
  </tbody>
</table>
```

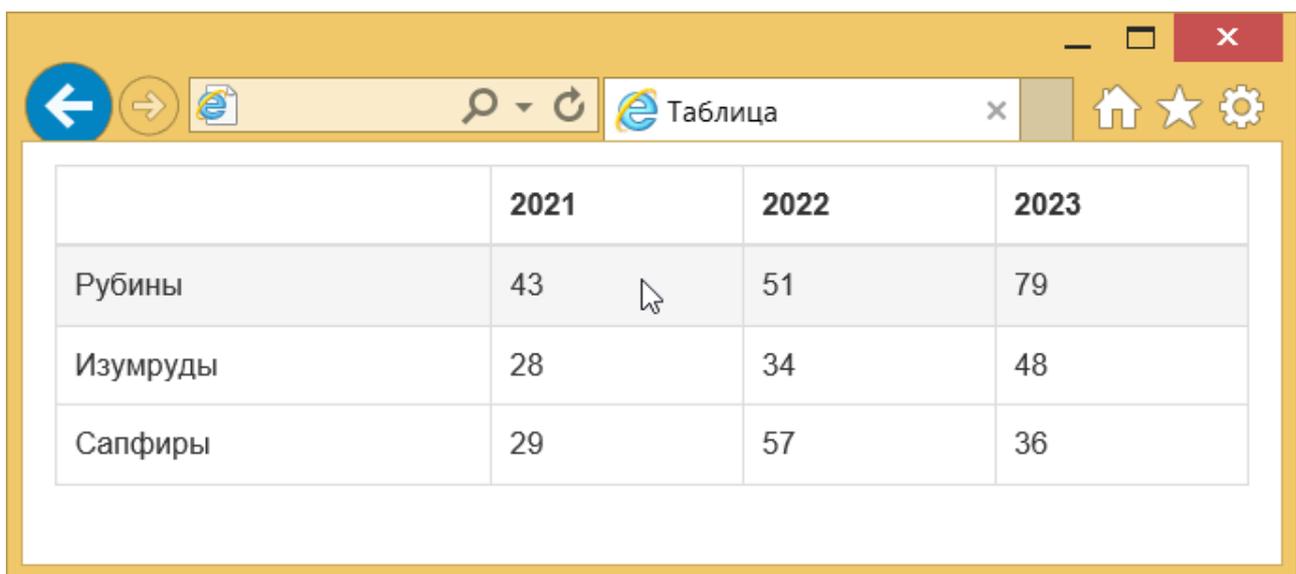
Заметьте, что класс `table` мы сохраняем и добавляем к нему новые классы. Результат данного примера показан на рис. 2.



	2021	2022	2023
Рубины	43	51	79
Изумруды	28	34	48
Сапфиры	29	57	36

Рис. 2. Сетка

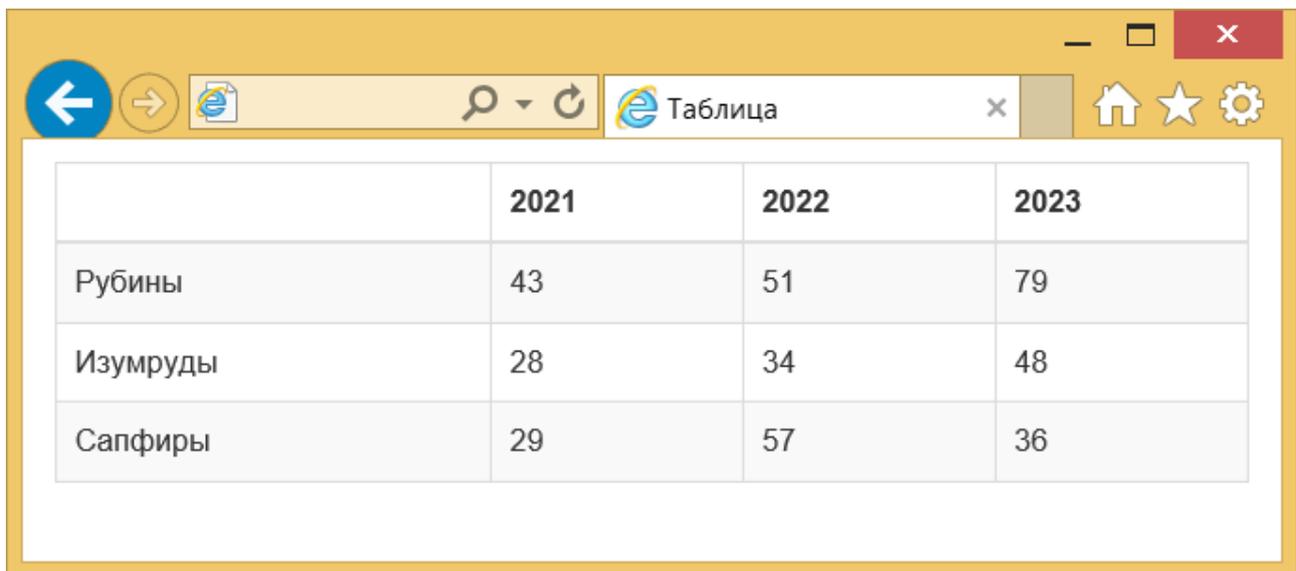
Можно добавить к нашему набору ещё один класс `table-hover` и тогда при наведении на строку таблицы она будет подсвечиваться серым цветом (рис. 3). Строки внутри `<thead>` не выделяются таким образом.



	2021	2022	2023
Рубины	43	51	79
Изумруды	28	34	48
Сапфиры	29	57	36

Рис. 3. Выделение строки таблицы

Для удобного представления табличных данных имеет смысл делать строки таблицы чередующимися, чтобы цвет фона чётных и нечётных строк различался (рис. 4). Такой вид таблицы часто называют «зеброй».



	2021	2022	2023
Рубины	43	51	79
Изумруды	28	34	48
Сапфиры	29	57	36

Рис. 4. Чередующиеся строки

Такой формат опять же делается путём добавления класса `table-striped` к элементу `<table>`. Всё вышеперечисленное можно объединить в одной таблице.

```
<table class="table table-bordered table-hover table-striped">
```

Кнопки

Кнопки применяются не только как элементы форм, в кнопку вполне можно превратить и ссылку, придав ей прямоугольный вид. Она по-прежнему останется ссылкой со всем доступным функционалом, но по своему оформлению будет напоминать ту же кнопку. Аналогично и наоборот — кнопке допустимо придать вид ссылки.

Для создания кнопок добавьте к элементу `<a>`, `<input>` или `<button>` класс `btn`. Второй класс нужен для изменения стиля кнопки. В примере 1 показаны возможные варианты кнопок и для каких целей и предназначены.

Пример 1. Кнопки

```
<button type="button" class="btn btn-default">Кнопка по умолчанию</button>  
<button type="button" class="btn btn-primary">Основная кнопка</button>  
<button type="button" class="btn btn-info">Информация</button>  
<button type="button" class="btn btn-warning">Предупреждение</button>  
<button type="button" class="btn btn-danger">Ошибка</button>  
<button type="button" class="btn btn-link">Кнопка как ссылка</button>
```

Результат данного примера показан на рис. 1.

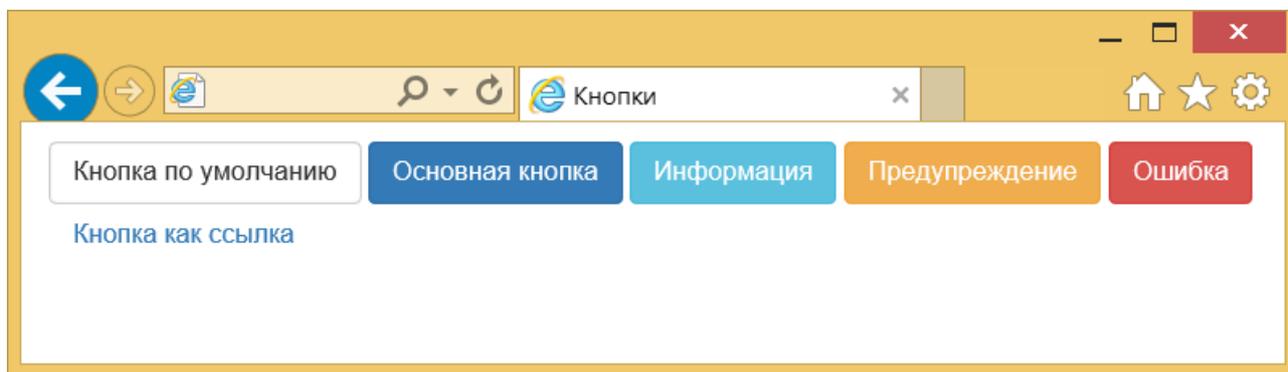


Рис. 1. Разные кнопки

При наведении курсора на кнопку она становится чуть темнее.

Размеры кнопок

Если вам требуется сделать кнопку больше или меньше, то для этого применяется три дополнительных класса. Класс `btn-lg` устанавливает кнопку большого размера, `btn-sm` — маленького, а `btn-xs` — очень маленького (пример 2).

Пример 2. Кнопки разного размера

```
<button type="button" class="btn btn-primary btn-lg">Большая кнопка</button>  
<button type="button" class="btn btn-primary">Размер по умолчанию</button>  
<button type="button" class="btn btn-primary btn-sm">Маленькая кнопка</button>  
<button type="button" class="btn btn-primary btn-xs">Очень маленькая</button>
```

Результат данного примера показан на рис. 2.

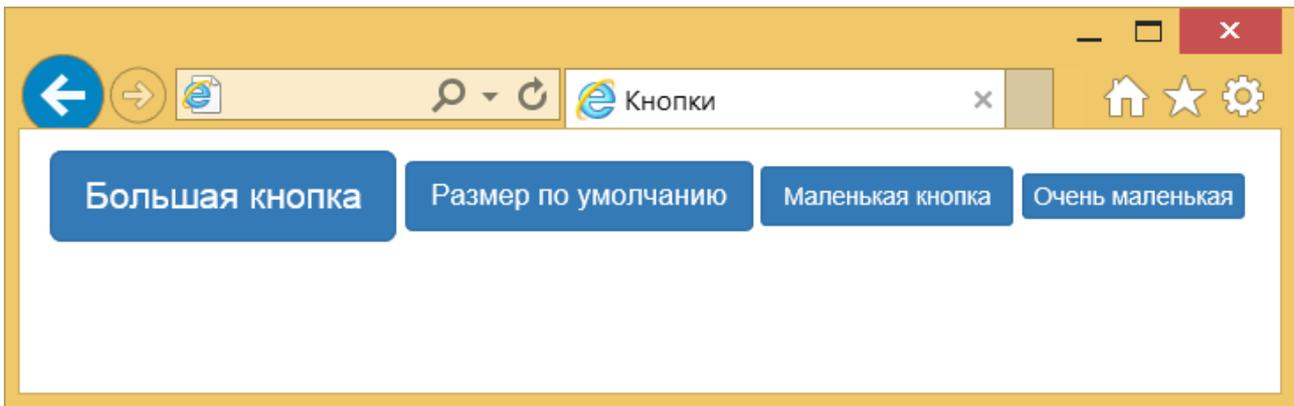


Рис. 2. Кнопки разного размера

Кнопку можно сделать и блочной, занимающей всю доступную ей ширину. Добавьте к ней только класс `btn-block`, в результате получится следующая кнопка (рис. 3).

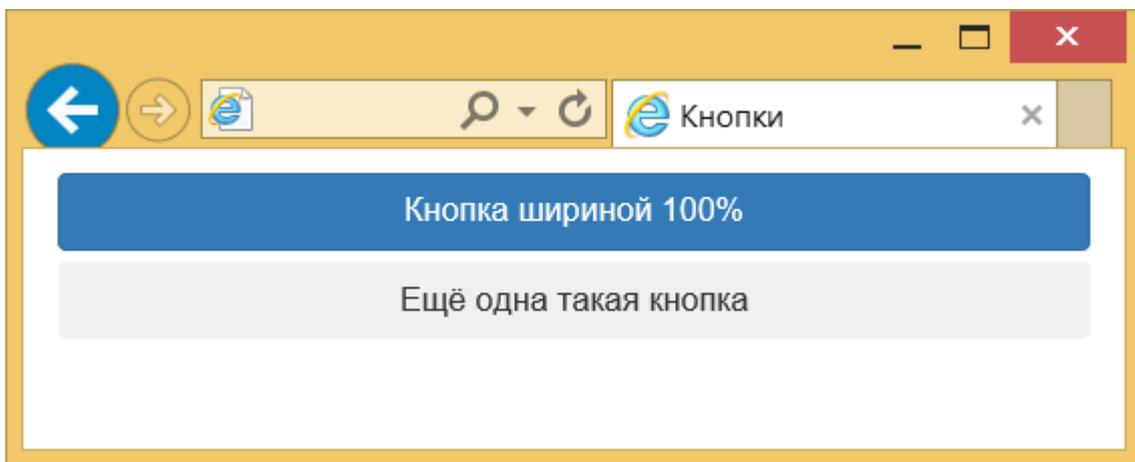


Рис. 3. Блочная кнопка

Код для создания таких кнопок показан в примере 3.

Пример 3. Блочные кнопки

```
<button type="button" class="btn btn-primary btn-block">Кнопка шириной 100%</button>
<button type="button" class="btn btn-block">Ещё одна такая кнопка</button>
```

Блокирование кнопок

Чтобы кнопка перестала работать, для элементов `<input>` и `<button>` достаточно добавить атрибут `disabled`. Bootstrap сам изменит стиль таких кнопок, показав наглядно, что нажимать на них бесполезно. Для ссылок, оформленных как кнопки, нужно включить класс `disabled`, как показано в примере 4.

Пример 4. Блокированные кнопки

```
<button type="button" class="btn btn-default">Обычная кнопка</button>
<button type="button" class="btn btn-primary" disabled>Ты не нажмёшь!</button>
<input type="submit" class="btn btn-default" disabled value="Блокированная
кнопка">
<a href="link1.html" class="btn btn-default disabled">Блокированная ссылка</a>
```

Результат данного примера показан на рис. 4.

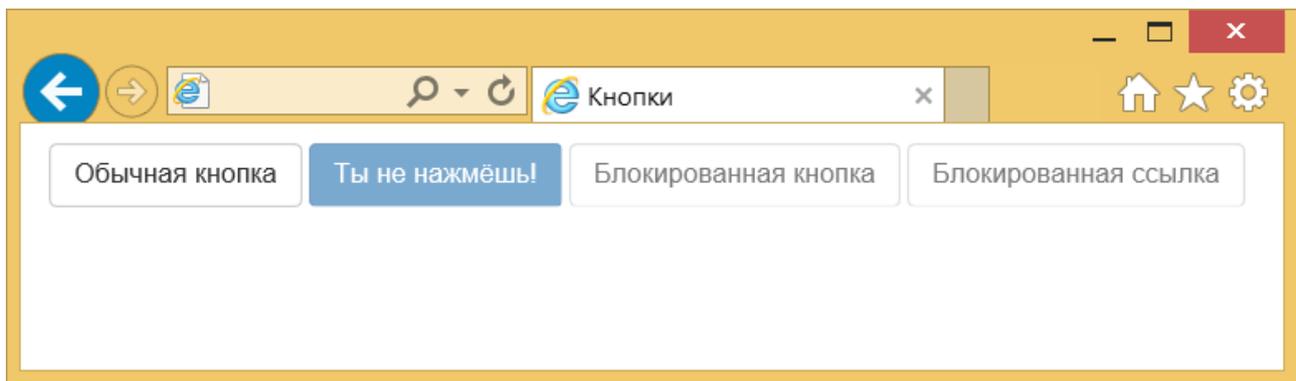


Рис. 4. Вид обычных и неактивных кнопок

Иконки

Bootstrap включает несколько десятков иконок, которые можно применять в тексте, для кнопок, в формах, в навигации и в других местах. Иконки сделаны в виде символов шрифта, поэтому могут масштабироваться в больших пределах без ухудшения качества и занимают при этом небольшой объём. Для добавления желаемой иконки обычно используется пустой элемент `<i>`, внутри которого пишем класс иконки. Вот, например, как на кнопку добавить символ лупы, она обычно символизирует поиск.

```
<button class="btn btn-primary"><i class="glyphicon glyphicon-search"></i></button>
```

В примере 1 показано создание панели инструментов с кнопками и иконками на них.

Пример 1. Панель инструментов

```
<div class="btn-toolbar">
  <div class="btn-group">
    <a class="btn btn-default" href="#"><i class="glyphicon glyphicon-align-left"></i></a>
    <a class="btn btn-default" href="#"><i class="glyphicon glyphicon-align-center"></i></a>
    <a class="btn btn-default" href="#"><i class="glyphicon glyphicon-align-right"></i></a>
    <a class="btn btn-default" href="#"><i class="glyphicon glyphicon-align-justify"></i></a>
  </div>
  <div class="btn-group">
    <a class="btn btn-default" href="#"><i class="glyphicon glyphicon-bold"></i></a>
    <a class="btn btn-default" href="#"><i class="glyphicon glyphicon-italic"></i></a>
  </div>
</div>
```

Результат данного примера показан на рис. 1.

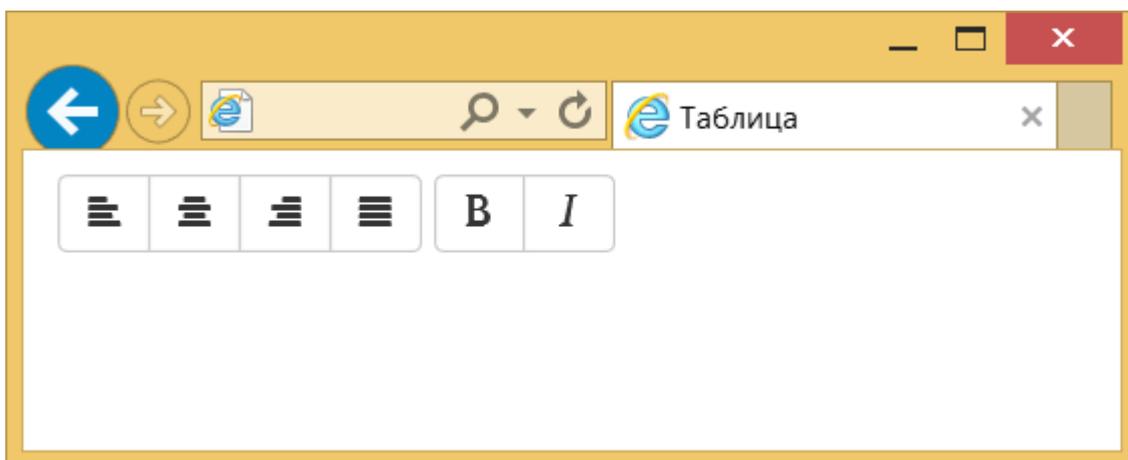


Рис. 1. Вид кнопок с иконками

См. также

Список всех иконок и классов для них

<http://getbootstrap.com/components/#glyphicons>

Формы

Вёрстка форм — это одна из сильных сторон Bootstrap. Фреймворк не только помогает расположить элементы форм в нужном порядке, но и обеспечивает изменение вида элементов и их взаимодействие с пользователем.

Давайте рассмотрим типовую форму для входа на сайт и на её примере разберём, как работает Boostrap для форм. В примере 1 показано создание базовой формы пока без добавления дополнительных классов.

Пример 1. Форма для входа на сайт

```
<form>
  <fieldset>
    <legend>Вход на сайт</legend>
    <p><label for="inputLogin">Логин</label></p>
    <p><input id="inputLogin" placeholder="Введите логин"></p>
    <p> <label for="inputPassword">Пароль</label></p>
    <p><input type="password" id="inputPassword" placeholder="Введите пароль"></p>
    <p><label><input type="checkbox"> Запомнить меня</label></p>
    <p><button type="submit">Войти</button></p>
  </fieldset>
</form>
```

Форма выглядит скучно (рис. 1), поскольку какое-либо оформление ещё не включено.

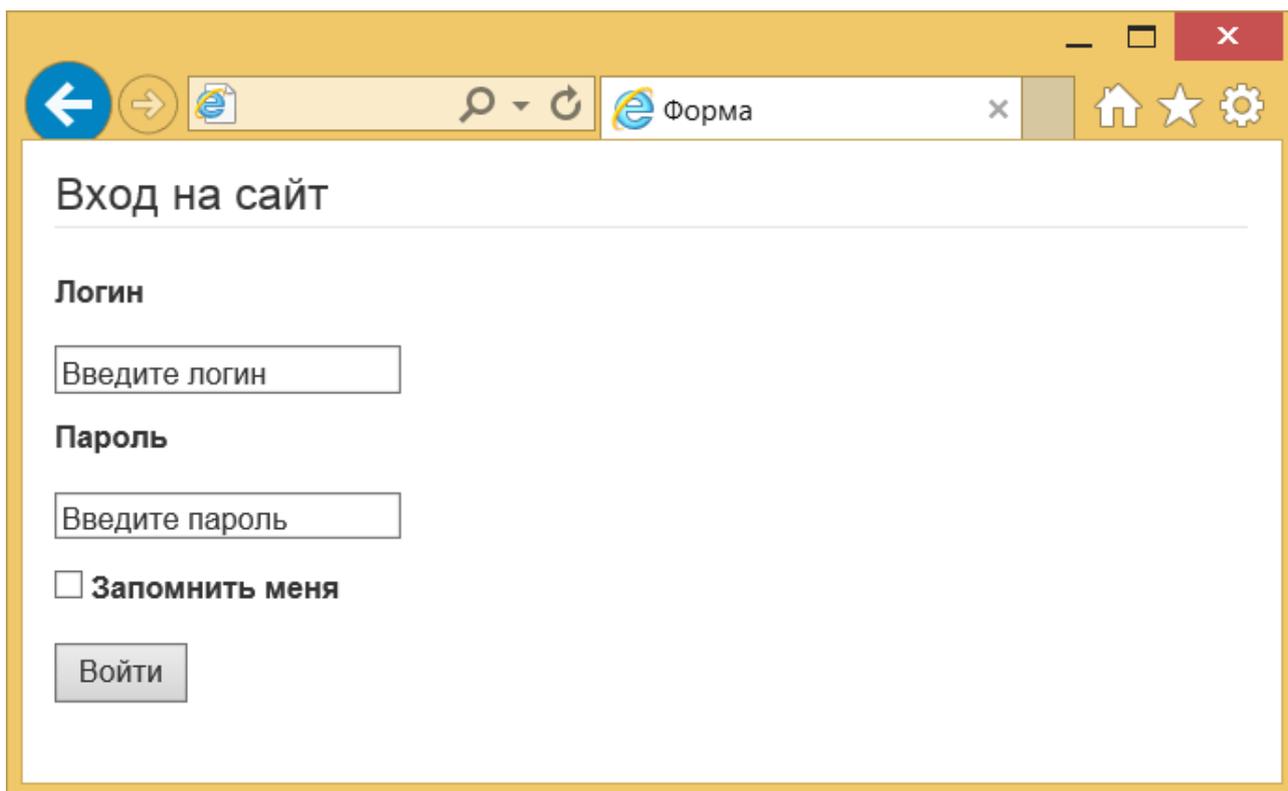


Рис. 1. Вид исходной формы

Для начала добавим к текстовому полю и полю с паролем класс `form-control`, это изменит стиль текстовых полей. Далее нужно сгруппировать заголовок и поле за ним, чтобы расстояние между ними было меньше обычного. Добавим `<label>` и `<input>` внутрь элемента `<div>` с классом `form-group`. Кроме того изменим чекбокс с текстом, добавив их внутрь `<div>` с классом `checkbox` и зададим другой вид кнопки. В примере 2 показаны изменения формы.

Пример 2. Применение классов

```
<form>
  <fieldset>
    <legend>Вход на сайт</legend>
    <div class="form-group">
      <label for="inputLogin">Логин</label>
      <input id="inputLogin" placeholder="Введите логин" class="form-control">
    </div>
    <div class="form-group">
      <label for="inputPassword">Пароль</label>
      <input type="password" id="inputPassword" placeholder="Введите пароль"
class="form-control">
    </div>
    <div class="checkbox">
      <label><input type="checkbox"> Запомнить меня</label>
    </div>
    <button type="submit" class="btn btn-default">Войти</button>
  </fieldset>
</form>
```

В результате форма выглядит иначе (рис. 2).

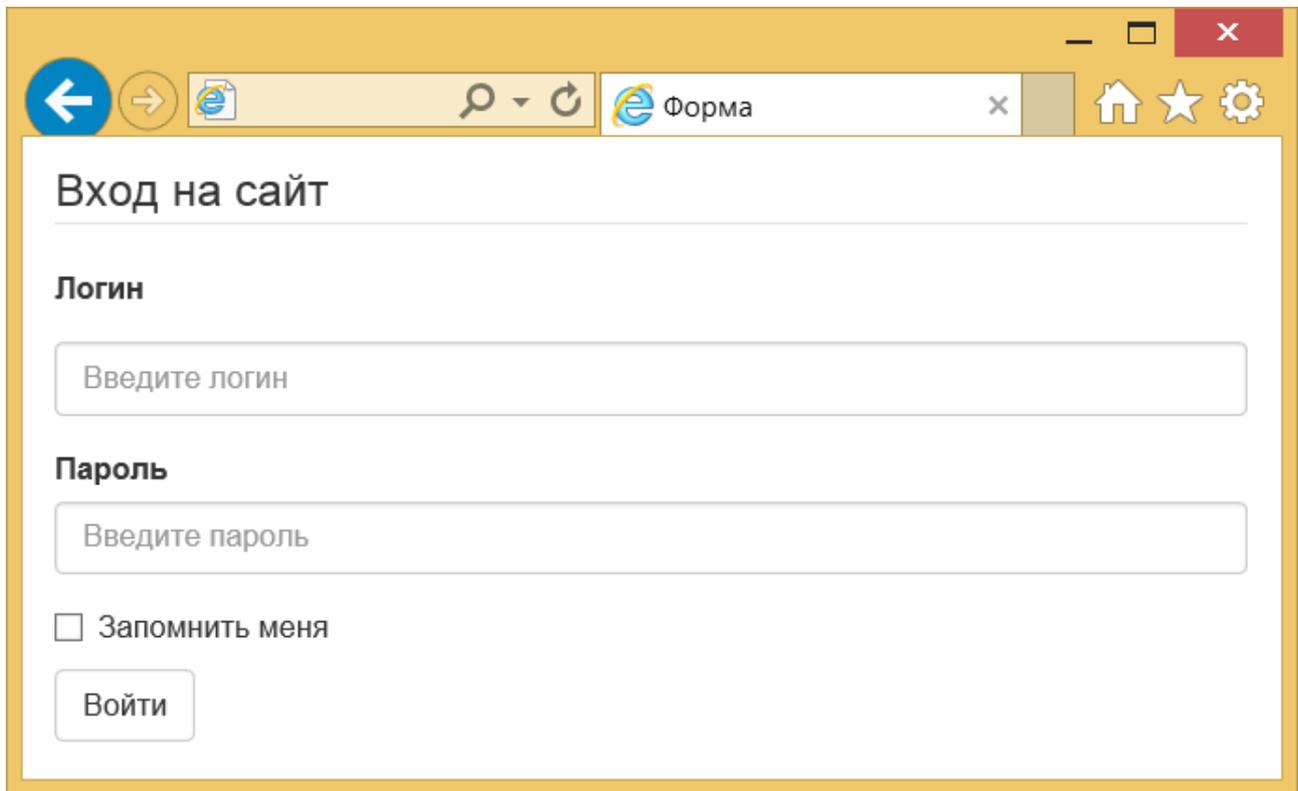
The image shows a browser window with a yellow title bar. The address bar contains the text 'Форма'. The main content area displays a login form with the title 'Вход на сайт'. Below the title, there are two input fields: one for 'Логин' (Login) with the placeholder text 'Введите логин', and one for 'Пароль' (Password) with the placeholder text 'Введите пароль'. Below these fields is a checkbox labeled 'Запомнить меня' (Remember me). At the bottom of the form is a button labeled 'Войти' (Login).

Рис. 2. Вид формы

Однострочная форма

Чтобы сделать форму компактнее, к элементу `<form>` достаточно добавить класс `form-inline`. Элементы формы после этого вытянутся в одну строку (рис. 3). Также из кода следует убрать `<fieldset>` и `<legend>`.

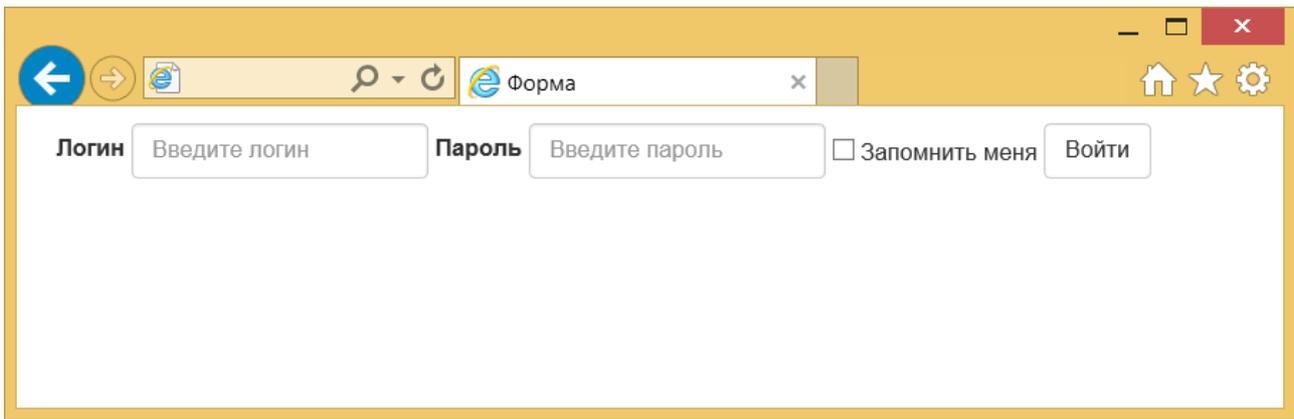


Рис. 3. Однострочная форма

В примере 3 показано создание такой формы.

Пример 3. Однострочная форма

```
<form class="form-inline">
  <div class="form-group">
    <label for="inputLogin">Логин</label>
    <input id="inputLogin" placeholder="Введите логин" class="form-control">
  </div>
  <div class="form-group">
    <label for="inputPassword">Пароль</label>
    <input type="password" id="inputPassword" placeholder="Введите пароль"
class="form-control">
  </div>
  <div class="checkbox">
    <label><input type="checkbox"> Запомнить меня</label>
  </div>
  <button type="submit" class="btn btn-default">Войти</button>
</form>
```

При уменьшении размера окна форма перестанет быть однострочной и примет исходный вид показанный на рис. 2.

Горизонтальная форма

Такой вид формы часто встречается на сайтах, когда поле и метка к нему располагаются на одной строке (рис. 4).

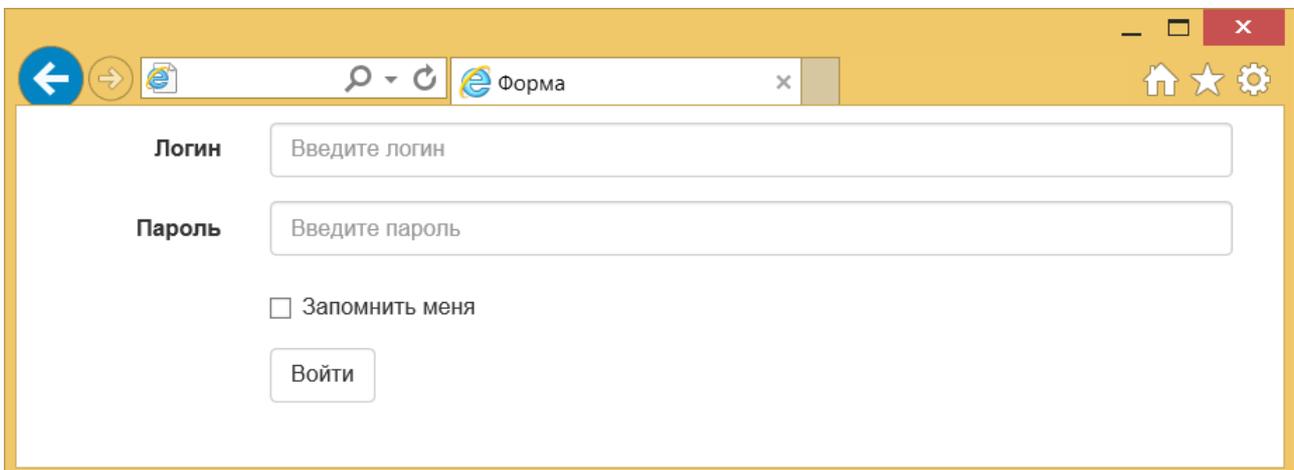


Рис. 4. Вид горизонтальной формы

Для создания формы такого вида требуется проделать несколько действий:

- к `<form>` добавить класс `form-horizontal`;
- к `<label>` добавить класс `control-label`, тогда метка поля будет выравниваться по правому краю и по центральной горизонтальной оси поля;
- обернуть `<label>` и `<input>` в `<div>` с классом `form-group`;
- использовать модульную сетку.

Мы уже имели дело с 12-колоночной модульной сеткой, применим эти навыки для управления дизайном формы. Метка будет занимать две колонки, а ширина самого поля — оставшиеся десять колонок. Вместо добавления дополнительного `<div>` добавим класс `col-sm-2` непосредственно к `<label>`, так немного сократится число элементов кода. Что касается чекбокса и кнопки, то добавим к ним класс `col-sm-offset-2`, он сдвинет поля прямо под текстовое поле (пример 4).

Пример 4. Горизонтальная форма

```
<form class="form-horizontal">
  <div class="form-group">
    <label for="inputLogin" class="col-sm-2 control-label">Логин</label>
    <div class="col-sm-10">
      <input id="inputLogin" placeholder="Введите логин" class="form-control">
    </div>
  </div>
  <div class="form-group">
    <label for="inputPassword" class="col-sm-2 control-label">Пароль</label>
    <div class="col-sm-10">
      <input type="password" id="inputPassword" placeholder="Введите пароль"
class="form-control">
    </div>
  </div>
  <div class="form-group">
    <div class="col-sm-10 col-sm-offset-2 checkbox">
      <label><input type="checkbox"> Запомнить меня</label>
    </div>
  </div>
  <div class="form-group">
    <div class="col-sm-10 col-sm-offset-2">
      <button type="submit" class="btn btn-default">Войти</button>
    </div>
  </div>
</form>
```

В классах мы использовали префикс `sm`, чтобы на смартфонах форма выглядела традиционно (рис. 2), а на больших экранах — как горизонтальная.

Компоненты

Компоненты Bootstrap — это коллекция инструментов применяемых в дизайне сайтов. Каждый компонент, как правило, включает в себя набор нескольких элементов HTML, стили для них и при необходимости скрипты, обеспечивающие взаимодействие с пользователем. Ниже описаны некоторые популярные компоненты Bootstrap.

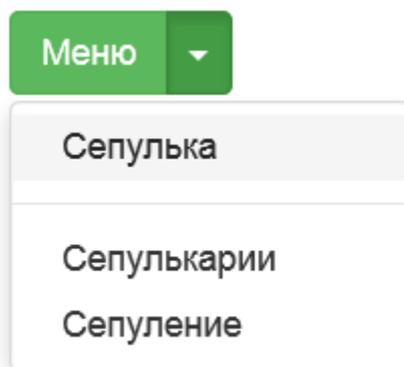
Группа кнопок

Объединяет несколько кнопок в одну группу, которая отображается как единое целое, без промежутков между кнопками.



Ниспадающее меню

Содержит кнопку, при нажатии на которую появляется список со ссылками.



Вкладки

На сайтах вкладки применяются там, где требуется компактно и наглядно скомпоновать информацию по блокам.



Хлебные крошки

На сайтах навигация в виде хлебных крошек применяется чтобы показать путь к текущему документу и его положение в иерархии сайта.

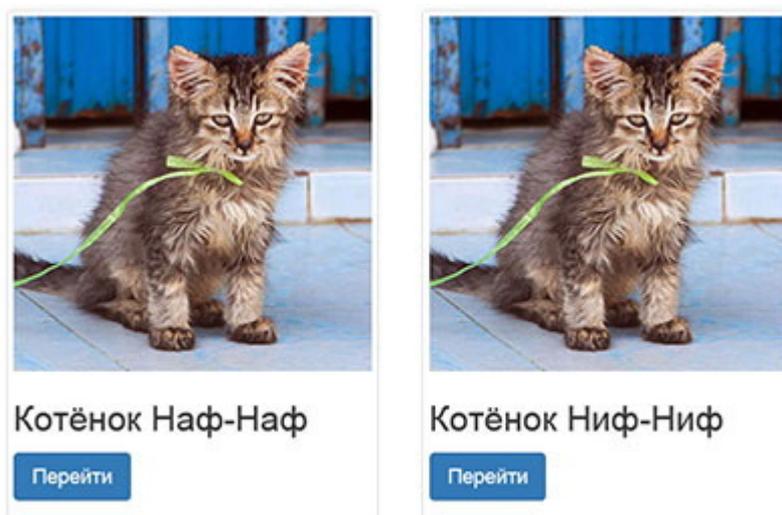
Дом / Лес / Опушка / Пряничный домик

Нумерация страниц

Для блогов общее количество постов разбивается на отдельные страницы, чтобы меньше загружать трафик и для упрощения восприятия информации. Переход между страницами делается с помощью специального пейджера с номерами всех страниц или ссылками на предыдущую и следующую страницу.

Эскизы

Эскиз — это уменьшенное изображение, одновременно являющееся ссылкой на страницу, либо на то же изображение увеличенное в размерах.



Сообщения

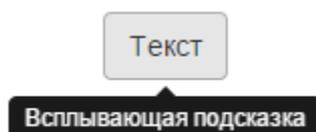
Чтобы дать пользователю знать о результатах его действий применяются информационные сообщения — для разных целей используются разные цвета.

Совет дня. Теперь вы можете закрыть это надоедливое сообщение, щёлкнув по крестику в углу.



Всплывающая подсказка

Выводит поясняющий текст при наведении курсора на элемент.



См. также

Описание всех компонентов Bootstrap

<http://getbootstrap.com/components>

Группа кнопок

Объединение кнопок в группы применяется для создания набора кнопок объединённых по какому-то принципу. Например, для кнопок редактирования можно выделить группу для выравнивания текста и группу для управления форматированием. Визуально такие кнопки располагаются в одну строку без пробелов между ними.

Для группирования кнопок надо поместить их в `<div>` с классом `btn-group`, как показано в примере 1.

Пример 1. Группирование кнопок

```
<div class="btn-group">
  <a class="btn btn-primary" href="#"><i class="glyphicon glyphicon-align-left"></i></a>
  <a class="btn btn-primary" href="#"><i class="glyphicon glyphicon-align-center"></i></a>
  <a class="btn btn-primary" href="#"><i class="glyphicon glyphicon-align-right"></i></a>
  <a class="btn btn-primary" href="#"><i class="glyphicon glyphicon-align-justify"></i></a>
</div>
<div class="btn-group">
  <a class="btn btn-primary" href="#"><i class="glyphicon glyphicon-font"></i></a>
  <a class="btn btn-primary" href="#"><i class="glyphicon glyphicon-bold"></i></a>
  <a class="btn btn-primary" href="#"><i class="glyphicon glyphicon-italic"></i></a>
</div>
```

Результат данного примера показан на рис. 1.

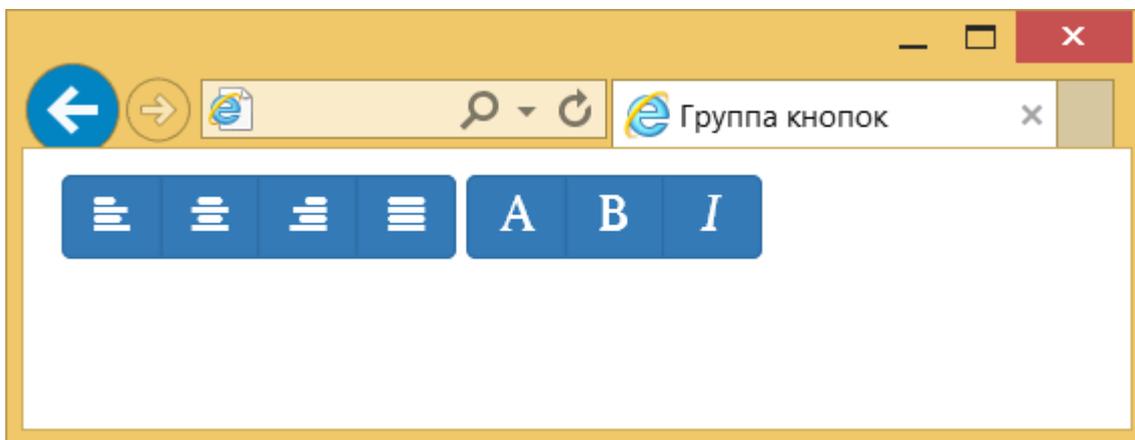


Рис. 1. Вид группы кнопок

Размер кнопок

Группе кнопок удобно сразу же задавать размер, в классах используя префиксы `xs`, `sm`, `md`, `lg`. Так, для уменьшения кнопок достаточно добавить класс `btn-group-sm` (пример 2).

Пример 2. Уменьшение кнопок

```
<div class="btn-group btn-group-sm">
  <a class="btn btn-info" href="#"><i class="glyphicon glyphicon-align-left"></i></a>
  <a class="btn btn-info" href="#"><i class="glyphicon glyphicon-align-center"></i></a>
```

```

<a class="btn btn-info" href="#"><i class="glyphicon glyphicon-align-right"></i></a>
<a class="btn btn-info" href="#"><i class="glyphicon glyphicon-align-justify"></i></a>
</div>
<div class="btn-group btn-group-sm">
  <a class="btn btn-info" href="#"><i class="glyphicon glyphicon-font"></i></a>
  <a class="btn btn-info" href="#"><i class="glyphicon glyphicon-bold"></i></a>
  <a class="btn btn-info" href="#"><i class="glyphicon glyphicon-italic"></i></a>
</div>

```

Результат данного примера показан на рис. 2.

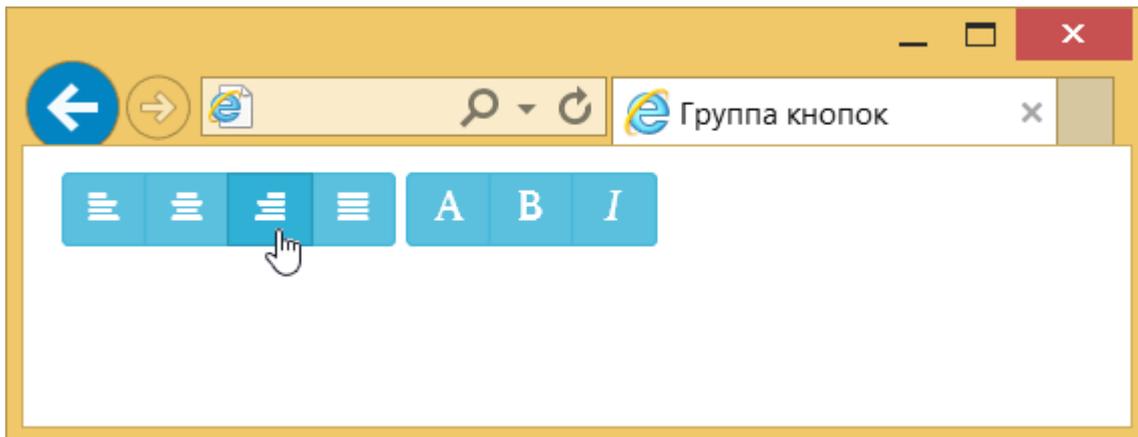


Рис. 2. Группа кнопок

Выравнивание по ширине

Группу кнопок можно выравнивать по ширине, тогда они будут подстраиваться под доступную им ширину и занимать её полностью (рис. 3). Подобное часто применяют для создания резинового меню.

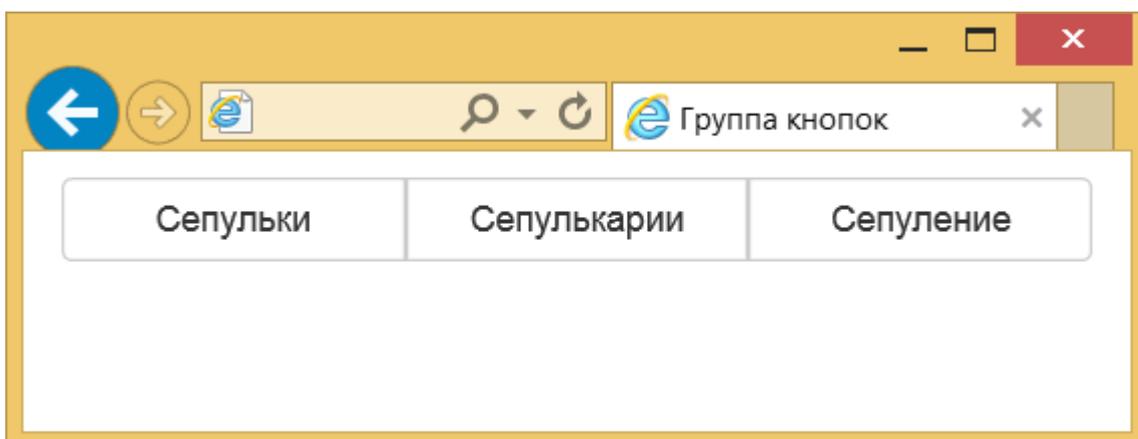


Рис. 3. Кнопки, занимающие всю ширину

Для выравнивания по ширине добавьте к <div> класс btn-group-justified как показано в примере 3.

Пример 3. Выравнивание кнопок по ширине

```

<div class="btn-group btn-group-justified">
  <a href="#" class="btn btn-default">Сепульки</a>
  <a href="#" class="btn btn-default">Сепулькарии</a>
  <a href="#" class="btn btn-default">Сепуление</a>

```

</div>

Ниспадающее меню

Такой тип меню прочно завоевал место в операционных системах и программах. На сайтах также применяется довольно часто из-за своей компактности и иерархической структуры — каждый пункт меню может содержать подменю, а тот в свою очередь ещё одно. Впрочем, из-за этого имеются некоторые проблемы связанные с удобством восприятия. По виду меню невозможно сразу понять сколько пунктов оно содержит, а вложенные подменю затрудняют навигацию. По этим причинам Microsoft в некоторых своих программах отказалась от ниспадающего меню в пользу ленточного интерфейса, как у Word. Но главное, есть инструмент, а уж пользоваться им и как решать вам, исходя из своих задач.

В Bootstrap само меню и кнопка для его вызова — это два разных элемента объединённых в один блок с классом `dropdown`. Внутри добавляем кнопку с атрибутом `data-toggle` и значением `dropdown`.

Традиционно возле текста указывается небольшая стрелка вниз — она означает, что пользователь имеет дело именно с ниспадающим меню, а не с обычной кнопкой или текстом. Для её создания просто вставьте `` после надписи на кнопке (пример 1).

Пример 1. Ниспадающее меню

```
<div class="dropdown">
  <button class="btn btn-default" id="dropdownMenu" type="button" data-
toggle="dropdown" aria-expanded="false">
  Меню <span class="caret"></span>
</button>
<ul class="dropdown-menu" role="menu" aria-labelledby="dropdownMenu">
  <li role="presentation"><a role="menuitem" href="#">Сепулька</a></li>
  <li role="presentation"><a role="menuitem" href="#">Сепулькарии</a></li>
  <li role="presentation"><a role="menuitem" href="#">Сепуление</a></li>
</ul>
</div>
```

Ниже кнопки мы добавляем список с пунктами меню и пишем класс `dropdown-menu`. Всё остальное берёт на себя Bootstrap, так что дополнительных действий от нас не требуется. В результате получим следующее меню (рис. 1).

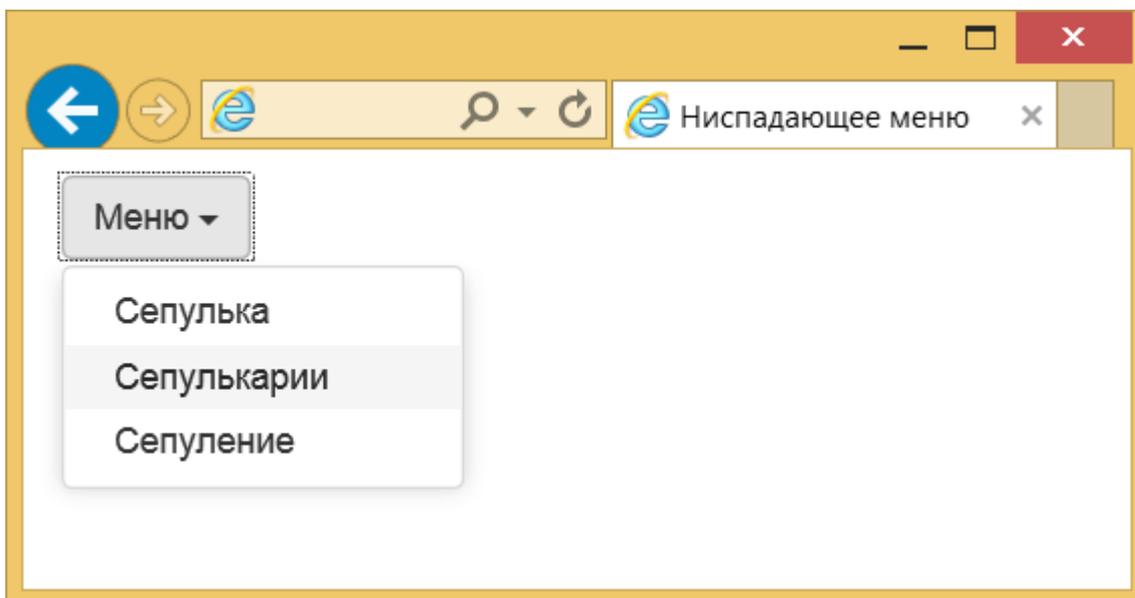


Рис. 1. Вид меню

Разделитель

Чтобы визуально отделить один или несколько пунктов меню от других применяется горизонтальный разделитель. Вместо обычной ссылки вставьте пустой элемент `` с классом `divider`, как показано в примере 2.

Пример 2. Разделитель

```
<div class="dropdown">
  <button class="btn btn-default" id="dropdownMenu" type="button" data-
toggle="dropdown" aria-expanded="false">
  Меню <span class="caret"></span>
</button>
<ul class="dropdown-menu" role="menu" aria-labelledby="dropdownMenu">
  <li role="presentation"><a role="menuitem" href="#">Сепулька</a></li>
  <li role="presentation" class="divider"></li>
  <li role="presentation"><a role="menuitem" href="#">Сепулькарии</a></li>
  <li role="presentation" class="divider"></li>
  <li role="presentation"><a role="menuitem" href="#">Сепуление</a></li>
</ul>
</div>
```

Результат данного примера показан на рис. 2.

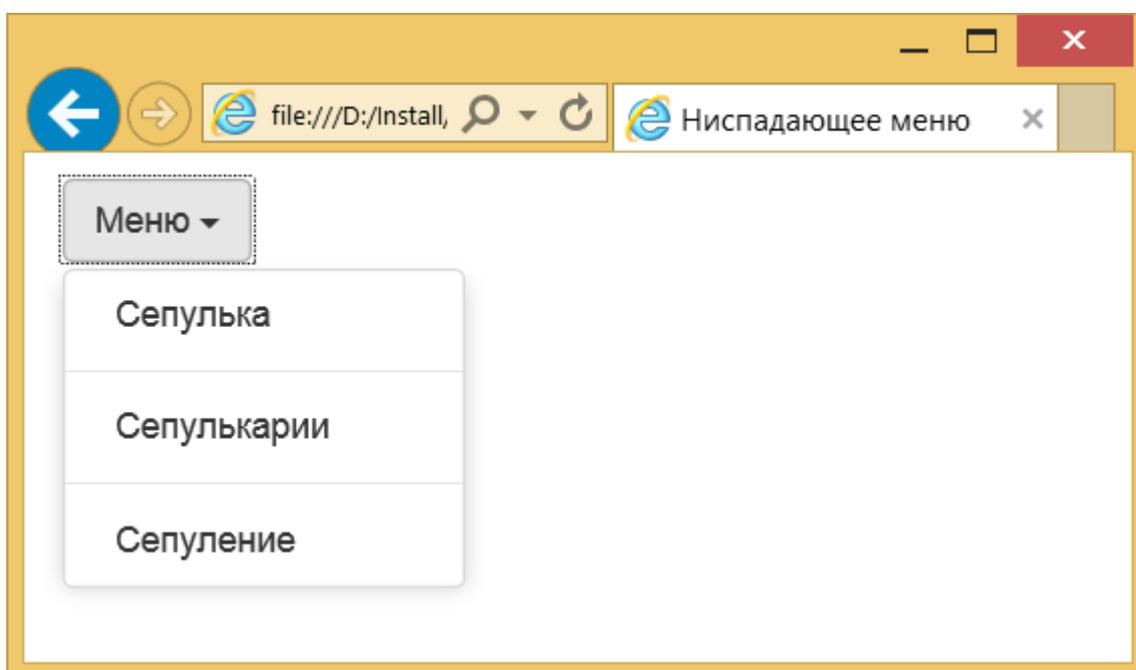


Рис. 2. Применение разделителя в меню

Блокирование пунктов меню

Если некоторые пункты меню требуется сделать неактивными, то к элементу `` следует добавить класс `disabled` (пример 3).

Пример 3. Блокированный пункт меню

```
<div class="dropdown">
  <button class="btn btn-default" id="dropdownMenu" type="button" data-
toggle="dropdown" aria-expanded="false">
  Меню <span class="caret"></span>
</button>
<ul class="dropdown-menu" role="menu" aria-labelledby="dropdownMenu">
  <li role="presentation" class="disabled"><a role="menuitem"
href="#">Сепулька</a></li>
```

```
<li role="presentation" class="divider"></li>
<li role="presentation"><a role="menuitem" href="#">Сепулькарии</a></li>
<li role="presentation"><a role="menuitem" href="#">Сепуление</a></li>
</ul>
</div>
```

Такой пункт меню будет виден, но он отображается другим цветом, на него нельзя щёлкнуть, а при наведении на пункт курсора мыши указатель меняется (рис. 3).

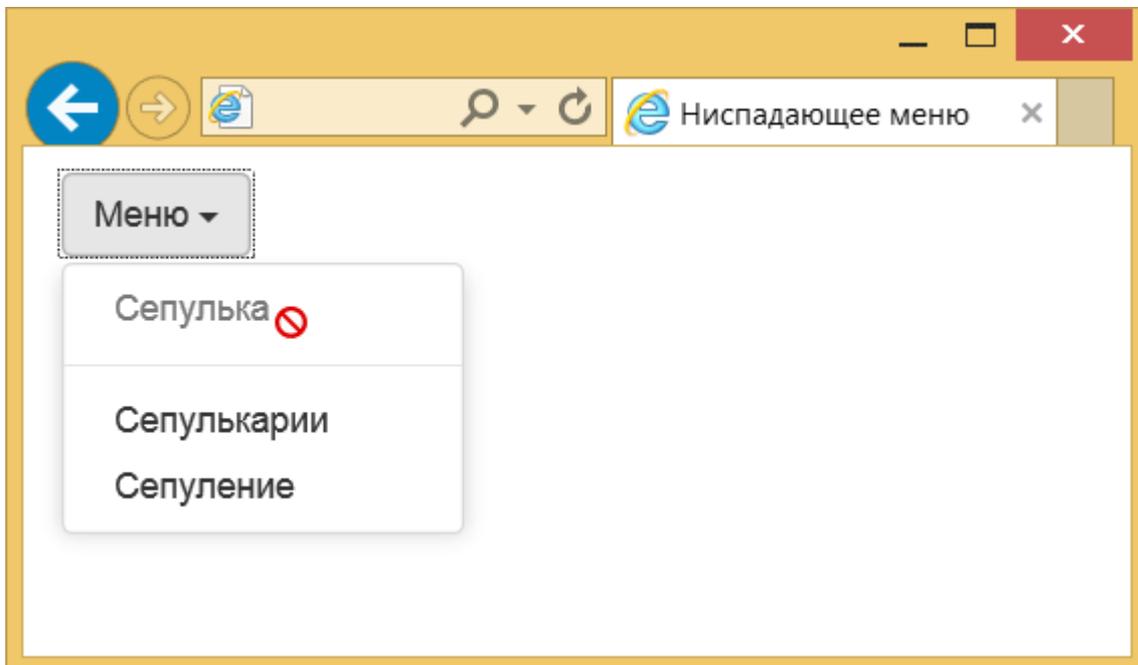


Рис. 3. Блокированный пункт меню

Открытое меню

Пользователю можно сразу показать открытое меню, тогда ему не придётся лишний раз щёлкать по нему чтобы посмотреть, что за ним скрывается. Просто добавьте класс `open` к родительскому `<div>`.

```
<div class="dropdown open">
```

Вид кнопок

Кроме добавления традиционных кнопок допустимо вставить кнопку, у которой стрелка от текста визуально отделена (рис. 4).

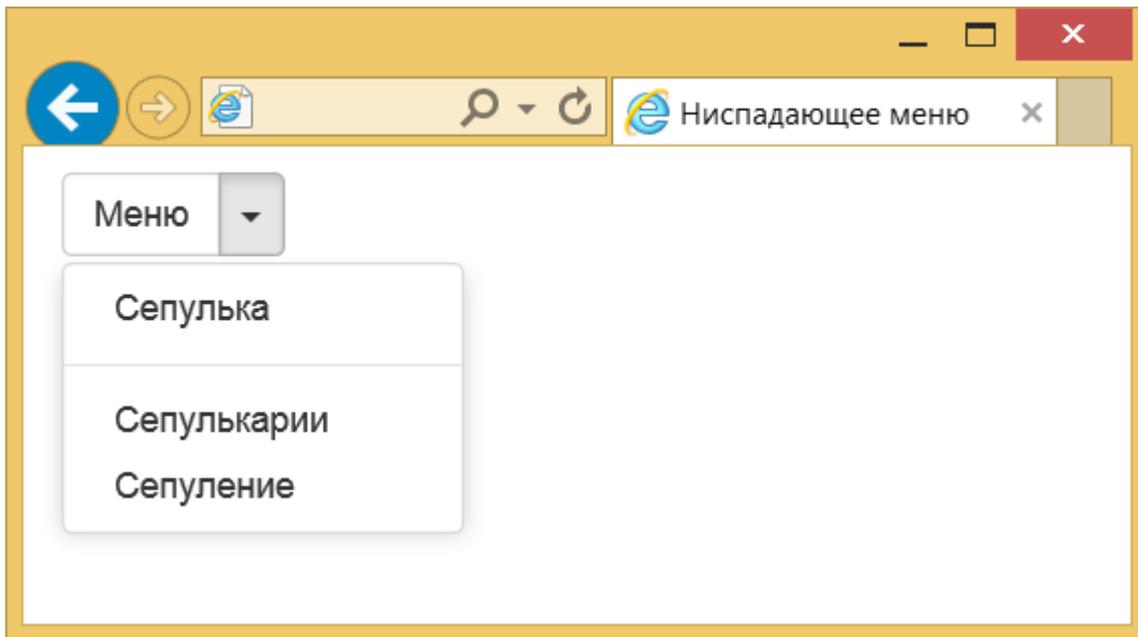


Рис. 4. Вид кнопки со стрелкой

Для создания такого элемента сделаем две кнопки — одна с текстом, другая со стрелкой и объединим их в группу, добавив класс btn-group (пример 4).

Пример 4. Кнопка со стрелкой

```
<div class="dropdown btn-group">
  <button class="btn btn-default" id="dropdownMenu" type="button">Меню</button>
  <button type="button" class="btn btn-default dropdown-toggle" data-
toggle="dropdown" aria-expanded="false">
  <span class="caret"></span>
</button>
  <ul class="dropdown-menu" role="menu" aria-labelledby="dropdownMenu">
  <li role="presentation"><a role="menuitem" href="#">Сепулька</a></li>
  <li role="presentation" class="divider"></li>
  <li role="presentation"><a role="menuitem" href="#">Сепулькарии</a></li>
  <li role="presentation"><a role="menuitem" href="#">Сепуление</a></li>
</ul>
</div>
```

Поскольку мы теперь имеем дело с двумя кнопками, возможны разные эксперименты. Например, можно задать разный цвет у кнопки с текстом и стрелкой или поменять их местами.

```
<button type="button" class="btn btn-success dropdown-toggle" data-
toggle="dropdown" aria-expanded="false">
  <span class="caret"></span>
</button>
<button class="btn btn-default" id="dropdownMenu" type="button">Меню</button>
```

В данном случае белая стрелка на зелёном фоне будет слева от текста «Меню».

Вкладки

Вкладки как элемент интерфейса пришли из делопроизводства. В обычные деловые папки вставляют небольшие кусочки картона с надписями, которые позволяют быстро сориентироваться в содержании и перейти к нужному разделу. На сайтах вкладки применяются там, где требуется компактно и наглядно скомпоновать информацию по блокам. Переход по вкладкам происходит без перезагрузки страницы, что ускоряет работу и упрощает пользователю жизнь.

Код вкладок состоит из двух частей — это отображение самих вкладок, а также информации с ними связанных.

Вначале создаём список `` с классами `nav` и `nav-tabs`. Важны оба класса, иначе мы не получим желаемый вид. Каждая вкладка формируется элементом ``, для текущей открытой вкладки требуется добавить класс `active`. Текст на вкладках делается с помощью ссылок, для каждой из них следует добавить атрибут `data-toggle` со значением `tab`. Сама ссылка указывает на элемент с идентификатором, который должен отображаться при открытии вкладки. Вот что в итоге получится.

```
<ul class="nav nav-tabs" role="tablist">
  <li role="presentation" class="active"><a href="#tab1" role="tab" data-
toggle="tab">Сепульки</a></li>
  <li role="presentation"><a href="#tab2" role="tab" data-
toggle="tab">Сепулькарии</a></li>
  <li role="presentation"><a href="#tab3" role="tab" data-
toggle="tab">Сепуление</a></li>
</ul>
```

Даже с этим куцым кодом вкладки должны уже работать и визуально переключаться.

Переходим к выводимому тексту. Оборачиваем всё элементом `<div>` с классом `tab-content`, внутри для каждой вкладки делаем отдельный `<div>` с классом `tab-pane`. Плюс добавляем уникальный идентификатор, связывающий текст с самой вкладкой. Последний штрих — для текущей вкладки указываем класс `active`.

```
<div class="tab-content">
  <div role="tabpanel" class="tab-pane active" id="tab1">
    Важный элемент цивилизации ардритов с планеты Энтеропия.
  </div>
  <div role="tabpanel" class="tab-pane" id="tab2">
    Устройства для сепуления.
  </div>
  <div role="tabpanel" class="tab-pane" id="tab3">
    Занятие ардритов с планеты Энтеропия.
  </div>
</div>
```

Соединяем вместе два фрагмента и получаем итоговый код для создания вкладок.

Пример 1. Создание вкладок

```
<ul class="nav nav-tabs" role="tablist">
  <li role="presentation" class="active"><a href="#tab1" role="tab" data-
toggle="tab">Сепульки</a></li>
  <li role="presentation"><a href="#tab2" role="tab" data-
toggle="tab">Сепулькарии</a></li>
  <li role="presentation"><a href="#tab3" role="tab" data-
toggle="tab">Сепуление</a></li>
</ul>
<div class="tab-content">
  <div role="tabpanel" class="tab-pane active" id="tab1">
    Важный элемент цивилизации ардритов с планеты Энтеропия.
  </div>
  <div role="tabpanel" class="tab-pane" id="tab2">
    Устройства для сепуления.
  </div>
```

```
</div>
<div role="tabpanel" class="tab-pane" id="tab3">
  Занятие ардритов с планеты Энтеропия.
</div>
</div>
```

Результат данного примера показан на рис. 1.

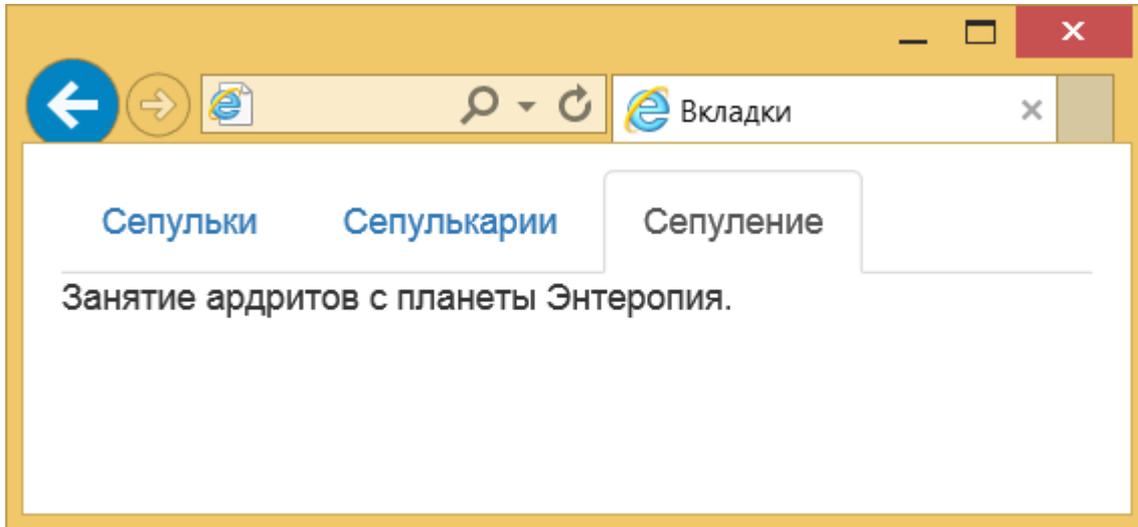


Рис. 1. Вид вкладок

Плавный переход

Переход между вкладками происходит мгновенно, но можно сделать это с небольшим эффектом, если добавить класс `fade` к блоку с текстом. Тогда текст будет плавно исчезать и появляться. Для активной вкладки обязательно надо включить ещё один класс `in`, без него возникнут ошибки.

```
<div class="tab-content">
  <div role="tabpanel" class="tab-pane active in fade" id="tab1">
    Важный элемент цивилизации ардритов с планеты Энтеропия.
  </div>
  <div role="tabpanel" class="tab-pane fade" id="tab2">
    Устройства для сепуления.
  </div>
  <div role="tabpanel" class="tab-pane fade" id="tab3">
    Занятие ардритов с планеты Энтеропия.
  </div>
</div>
```

Вкладки на всю ширину

Размер вкладки зависит от длины текста в ней, но можно насильно изменить их ширину, чтобы вкладки занимали всё доступную ширину. Просто добавьте класс `nav-justified` к элементу ``.

```
<ul class="nav nav-tabs nav-justified" role="tablist">
```

Вот как выглядят подобные вкладки (рис. 2).

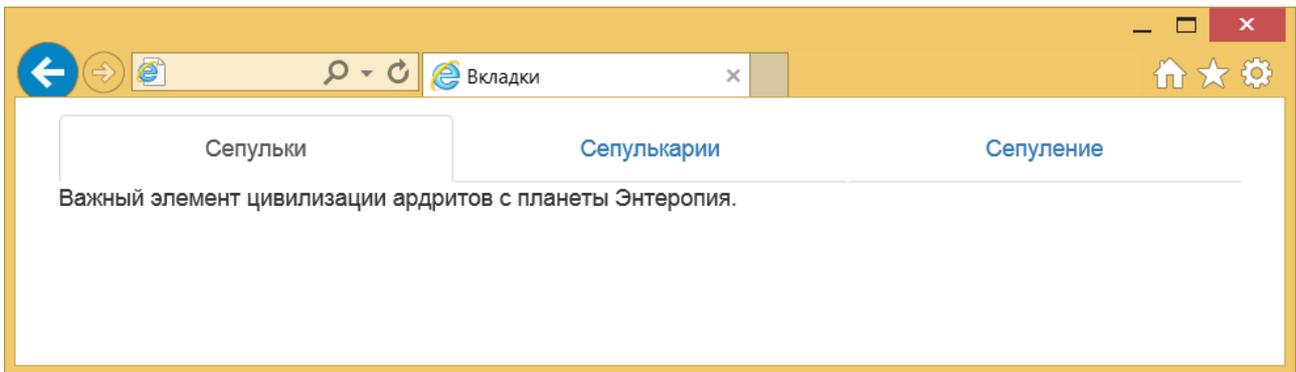


Рис. 2. Вкладки на всю ширину

При уменьшении размера окна такие вкладки меняют свой вид и выстраиваются не по горизонтали, а друг под другом (рис. 3).

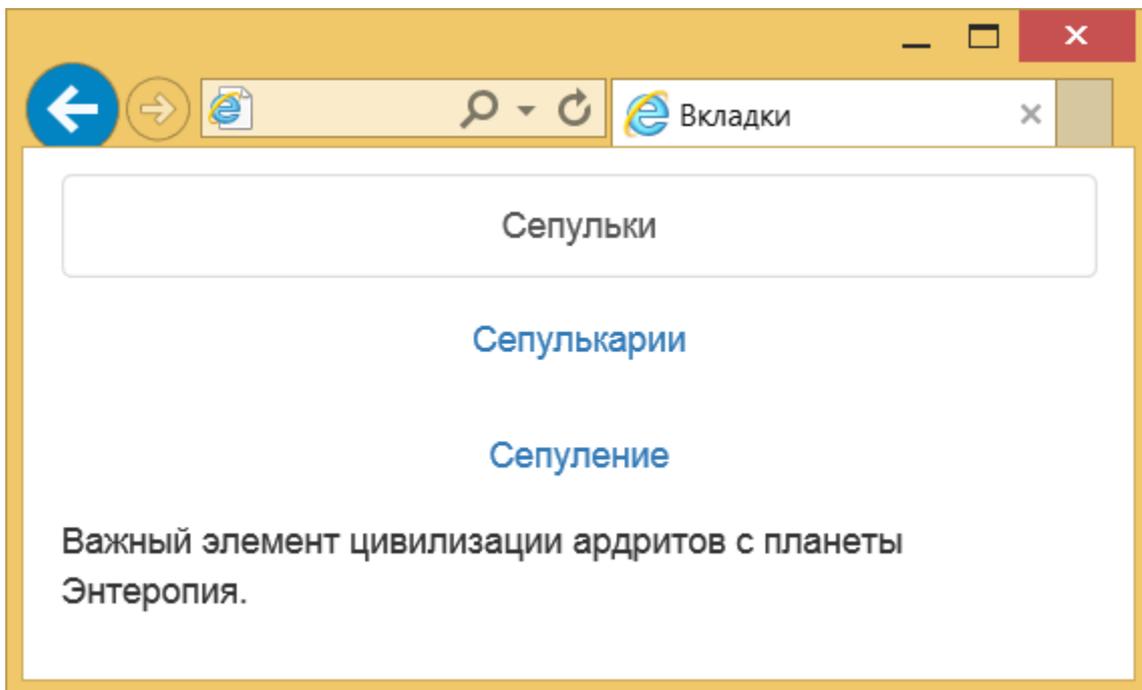


Рис. 3. Вид вкладок при уменьшении размера окна

Вид вкладок

Традиционно вкладки имеют вид унаследованный из жизни — выступающие из папки кусочки картона с надписью. Заменяв класс `nav-tabs` на `nav-pills` получим вкладки в виде пиллюль (рис. 4).

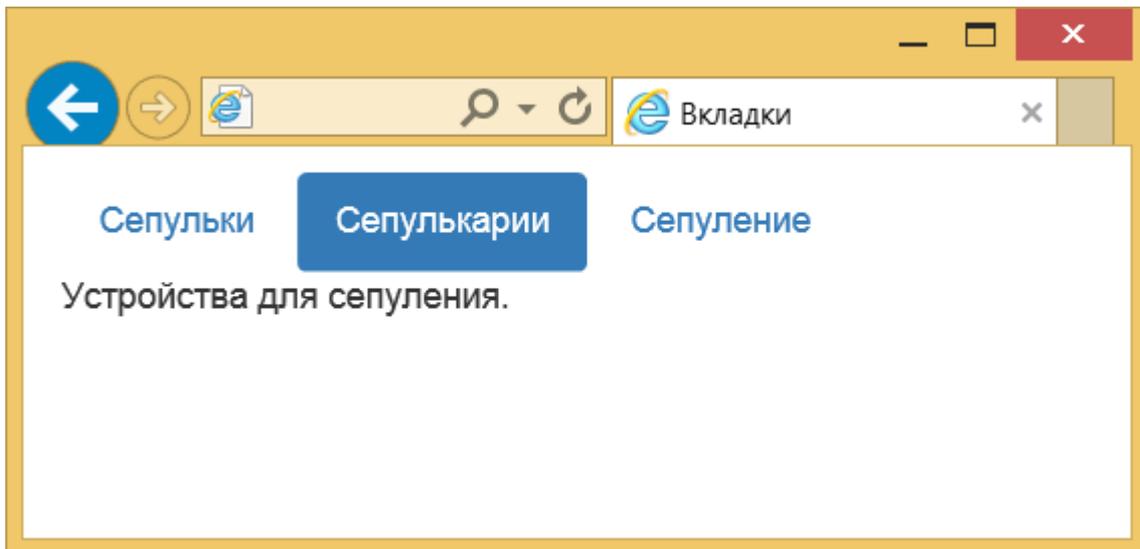


Рис. 4. Изменение вида вкладок

Код будет таким.

```
<ul class="nav nav-pills" role="tablist">
```

Вкладки также можно расположить вертикально, для этого предназначен класс `nav-stacked`.

```
<ul class="nav nav-pills nav-stacked" role="tablist">
```

Вид таких вкладок продемонстрирован на рис. 5.

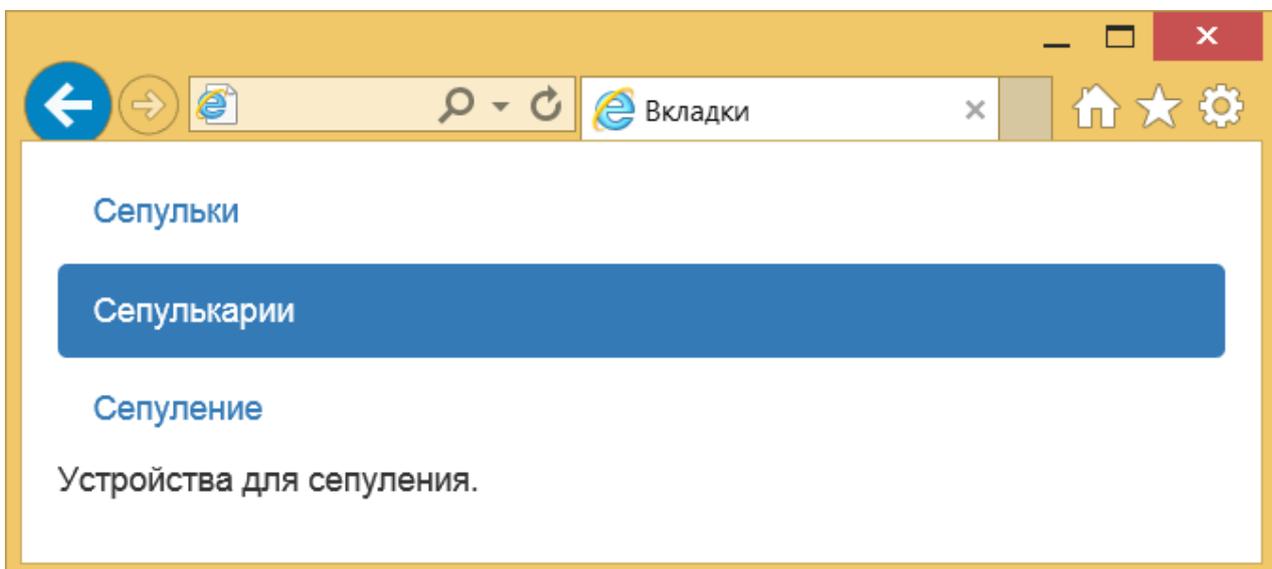


Рис. 5. Вертикальное расположение вкладок

Хлебные крошки

Название «хлебные крошки» пошло из детских сказок. Помните историю про мальчика, который по пути рассыпал хлебные крошки и по ним сумел найти обратную дорогу домой. Впрочем, в одной из версий крошки склевали птицы и это дало толчок новым приключениям героя. На сайтах навигация в виде хлебных крошек применяется чтобы показать путь к текущему документу и его положение в иерархии сайта (рис. 1).

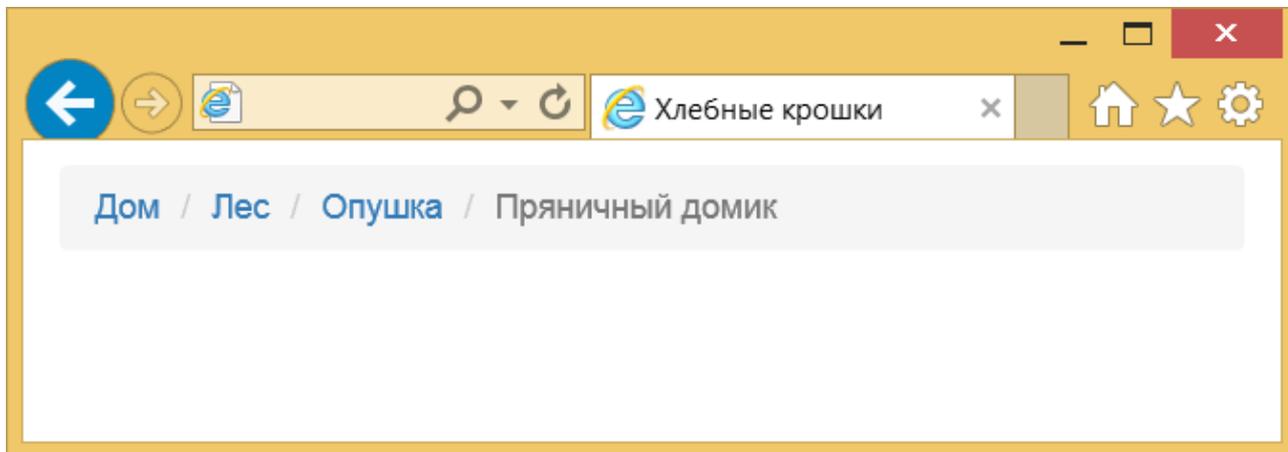


Рис. 1. Вид хлебных крошек

Серый текст показывает название текущего документа, а ссылки ведут на предыдущие документы, при этом первая ссылка, как правило, всегда указывает на главную страницу сайта.

Для создания хлебных крошек используется список `` с классом `breadcrumb`, элементы `` формируют сами «крошки». Текущий пункт ссылкой не является и к нему добавляется класс `active` (пример 1).

Пример 1. Хлебные крошки

```
<ul class="breadcrumb">
  <li><a href="#">Дом</a></li>
  <li><a href="#">Лес</a></li>
  <li><a href="#">Опушка</a></li>
  <li class="active">Пряничный домик</li>
</ul>
```

Нумерация страниц

Для блогов общее количество постов разбивается на отдельные страницы, чтобы меньше загружать трафик и для упрощения восприятия информации. Навигация между страницами происходит одним из двух способов:

1. выводятся номера всех страниц со стрелками, ведущими на предыдущую и следующую страницу (рис. 1);

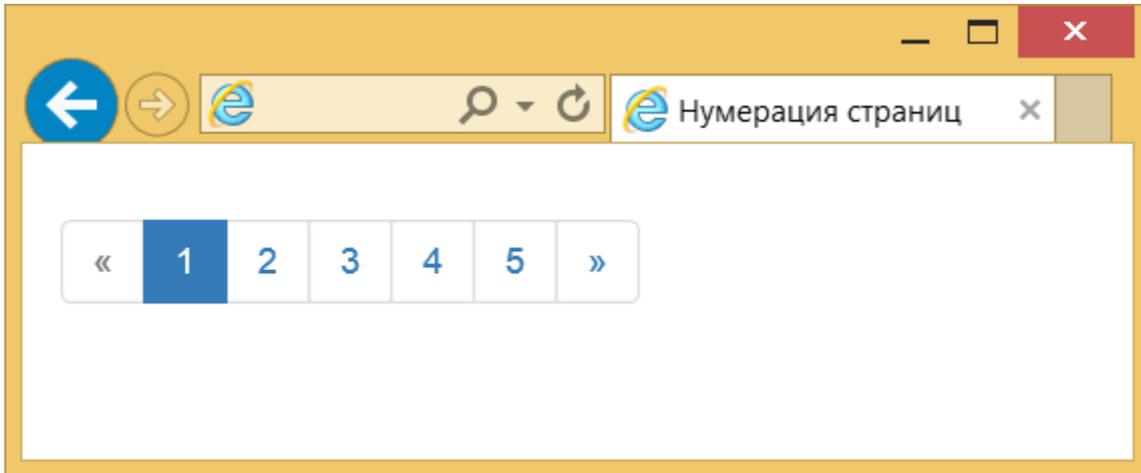


Рис. 1. Нумерация всех страниц

2. выводятся только ссылки на предыдущую и следующую страницу (рис. 2).

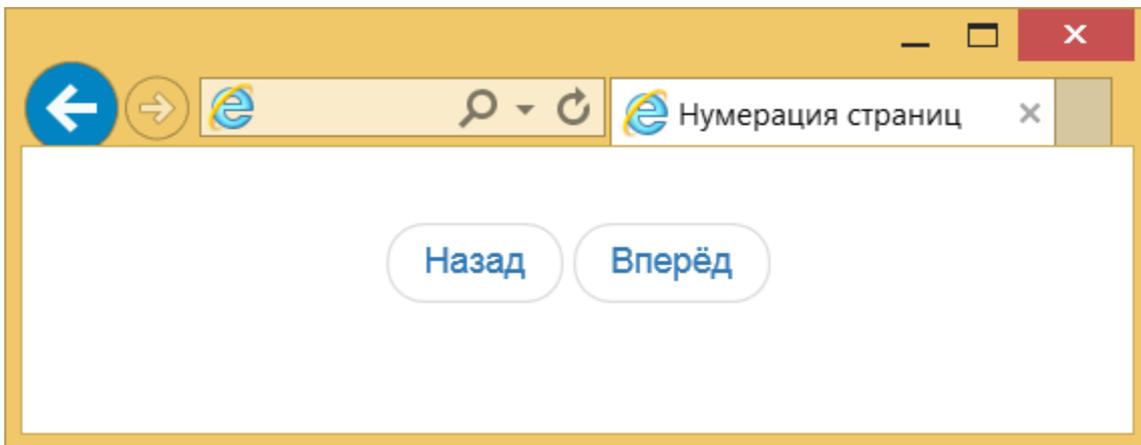


Рис. 2. Ссылки на предыдущую и следующую страницу

Все страницы

Нумерация делается с помощью элемента `` с классом `pagination`. Номера добавляются через элементы ``, при этом мы сами настраиваем их количество. Например, если нам нужны ссылки для перехода на первую и последнюю страницу или ссылки ведущие на предыдущую и следующую страницу (пример 1).

Пример 1. Нумерация страниц

```
<ul class="pagination">
  <li class="disabled"><a href="#">&laquo;</a></li>
  <li class="active"><a href="#">1</a></li>
  <li ><a href="#">2</a></li>
  <li><a href="#">3</a></li>
```

```
<li><a href="#">4</a></li>
<li><a href="#">5</a></li>
<li><a href="#">&raquo;</a></li>
</ul>
```

Текущий пункт навигации выделяется через класс `active` добавляемый к ``. Если требуется заблокировать какой-то пункт, то добавьте к нему класс `disabled`. Блокирование нужно, чтобы не вводить пользователя в заблуждение. Так, если он находится на первой странице, то переход на предыдущую страницу невозможен, потому как её просто в данный момент нет. При наведении курсора мыши на такой пункт указатель меняет свой вид.

Для уменьшения или увеличения размера ссылок используйте соответственно классы `pagination-sm` и `pagination-lg`, добавив их к ``. Сам класс `pagination` никуда не уходит при этом. Вот как будет выглядеть код для большего размера.

```
<ul class="pagination pagination-lg">
```

Предыдущая и следующая страница

Второй тип навигации по страницам содержит только ссылки на предыдущую и следующую страницу. Соответственно, упрощается код и вид нумерации, поскольку в списке будет всего два пункта. При этом для `` используем класс `pager`, как показано в примере 2.

Пример 2. Переход по страницам

```
<ul class="pager">
  <li><a href="#">Назад</a></li>
  <li><a href="#">Вперёд</a></li>
</ul>
```

Ссылки выводятся достаточно близко друг от друга и чтобы их разнести и выровнять по левому и правому краям добавьте к ним классы `previous` и `next` (пример 3).

Пример 3. Выравнивание ссылок

```
<ul class="pager">
  <li class="previous"><a href="#">Сюда</a></li>
  <li class="next"><a href="#">Туда</a></li>
</ul>
```

Результат данного примера показан на рис. 3.

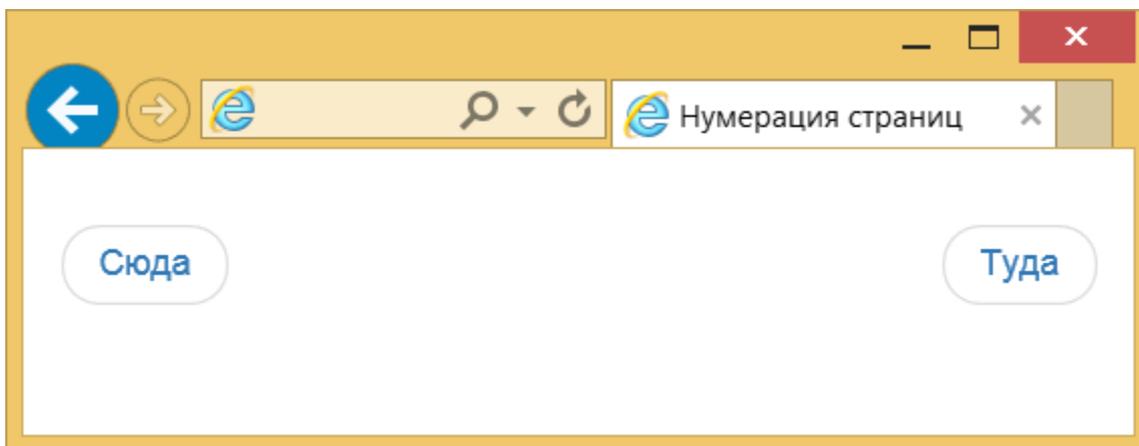


Рис. 3. Выравнивание ссылок

Эскизы

Эскиз — это уменьшенное изображение, одновременно являющееся ссылкой на страницу, либо на то же изображение увеличенное в размерах. Часто используют синоним «ноготки» (от англ. thumbnail — ноготь) или кальку «тамбы».

В отличие от обычных изображений эскизы вписываются в модульную сетку и размеры картинок подстраиваются под ширину макета. Поэтому вначале мы должны сделать сетку, а уже в колонки вставить ссылки с фотографиями. Пусть будет четыре колонки для средних экранов и две колонки для маленьких экранов. В каждую колонку добавляем элемент `<a>` с классом `thumbnail` и элемент `` (пример 1).

Пример 1. Создание эскизов

```
<div class="row">
  <div class="col-xs-6 col-md-3">
    <a href="#" class="thumbnail"></a>
  </div>
  <div class="col-xs-6 col-md-3">
    <a href="#" class="thumbnail"></a>
  </div>
  <div class="col-xs-6 col-md-3">
    <a href="#" class="thumbnail"></a>
  </div>
  <div class="col-xs-6 col-md-3">
    <a href="#" class="thumbnail"></a>
  </div>
</div>
```

Результат данного примера показан на рис. 1.

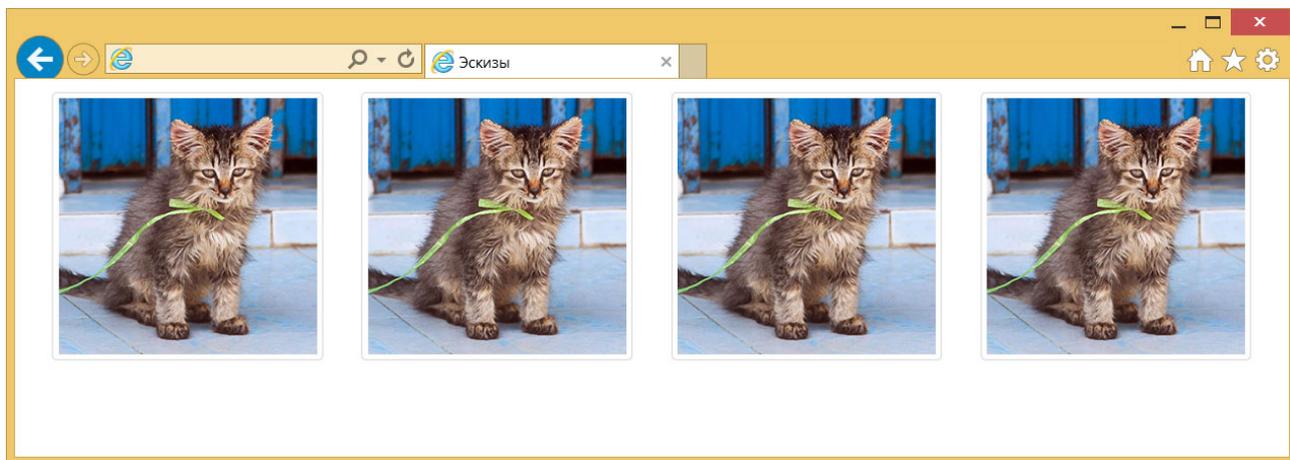


Рис. 1. Вид эскизов для среднего экрана

При уменьшении ширины окна браузера вид макета немного меняется (рис. 2).

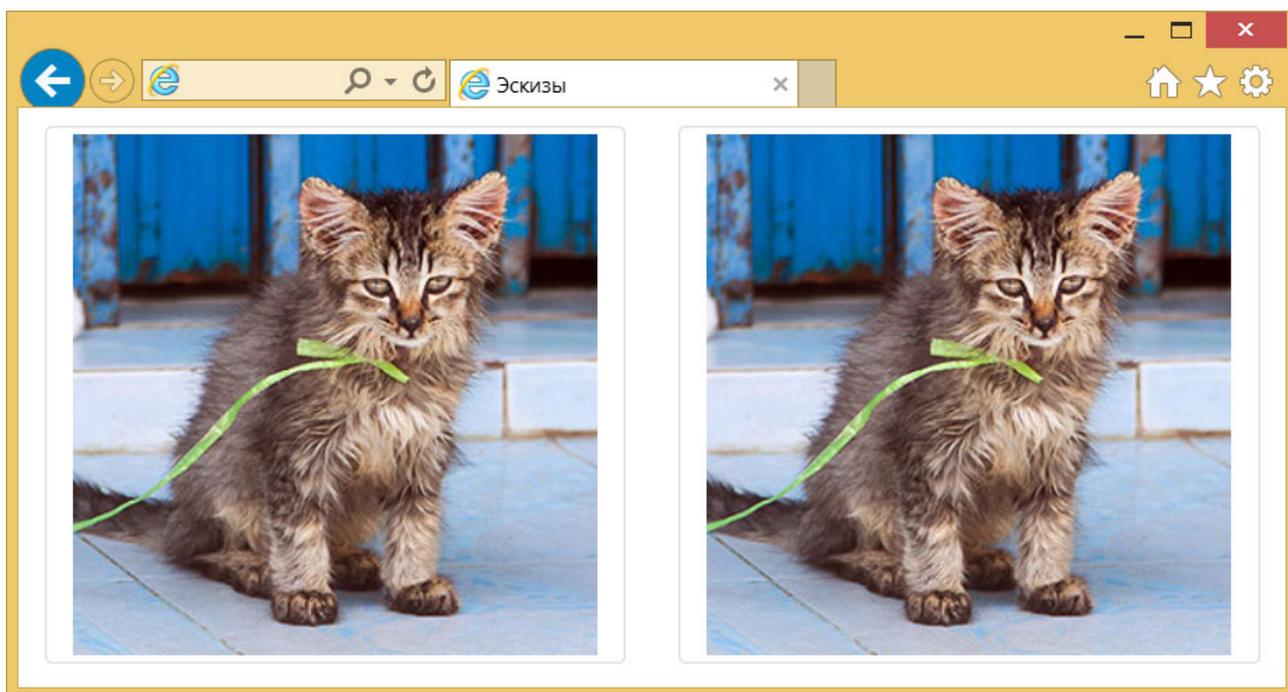


Рис. 2. Вид эскизов для маленького экрана

Внутри элемента с классом `thumbnail` разрешается добавлять не только изображения, но и поясняющий текст к нему. Это позволяет гибко настраивать вид в соответствии со своим дизайном (пример 2).

Пример 2. Эскизы с текстом и кнопкой

```
<div class="row">
  <div class="col-xs-6 col-md-3">
    <div class="thumbnail">
      <a href="#"></a>
      <h3>Котёнок Наф-Наф</h3>
      <p><a href="#" class="btn btn-primary" role="button">Перейти</a></p>
    </div>
  </div>
  <div class="col-xs-6 col-md-3">
    <div class="thumbnail">
      <a href="#"></a>
      <h3>Котёнок Ниф-Ниф</h3>
      <p><a href="#" class="btn btn-primary" role="button">Перейти</a></p>
    </div>
  </div>
  <div class="col-xs-6 col-md-3">
    <div class="thumbnail">
      <a href="#"></a>
      <h3>Котёнок Нуф-Нуф</h3>
      <p><a href="#" class="btn btn-primary" role="button">Перейти</a></p>
    </div>
  </div>
  <div class="col-xs-6 col-md-3">
    <div class="thumbnail">
      <a href="#"></a>
      <h3>Котёнок Няф-Няф</h3>
      <p><a href="#" class="btn btn-primary" role="button">Перейти</a></p>
    </div>
  </div>
</div>
```

В данном примере под каждым изображением добавлен заголовок и кнопка для перехода (рис. 3).

Фотография по прежнему выступает ссылкой.

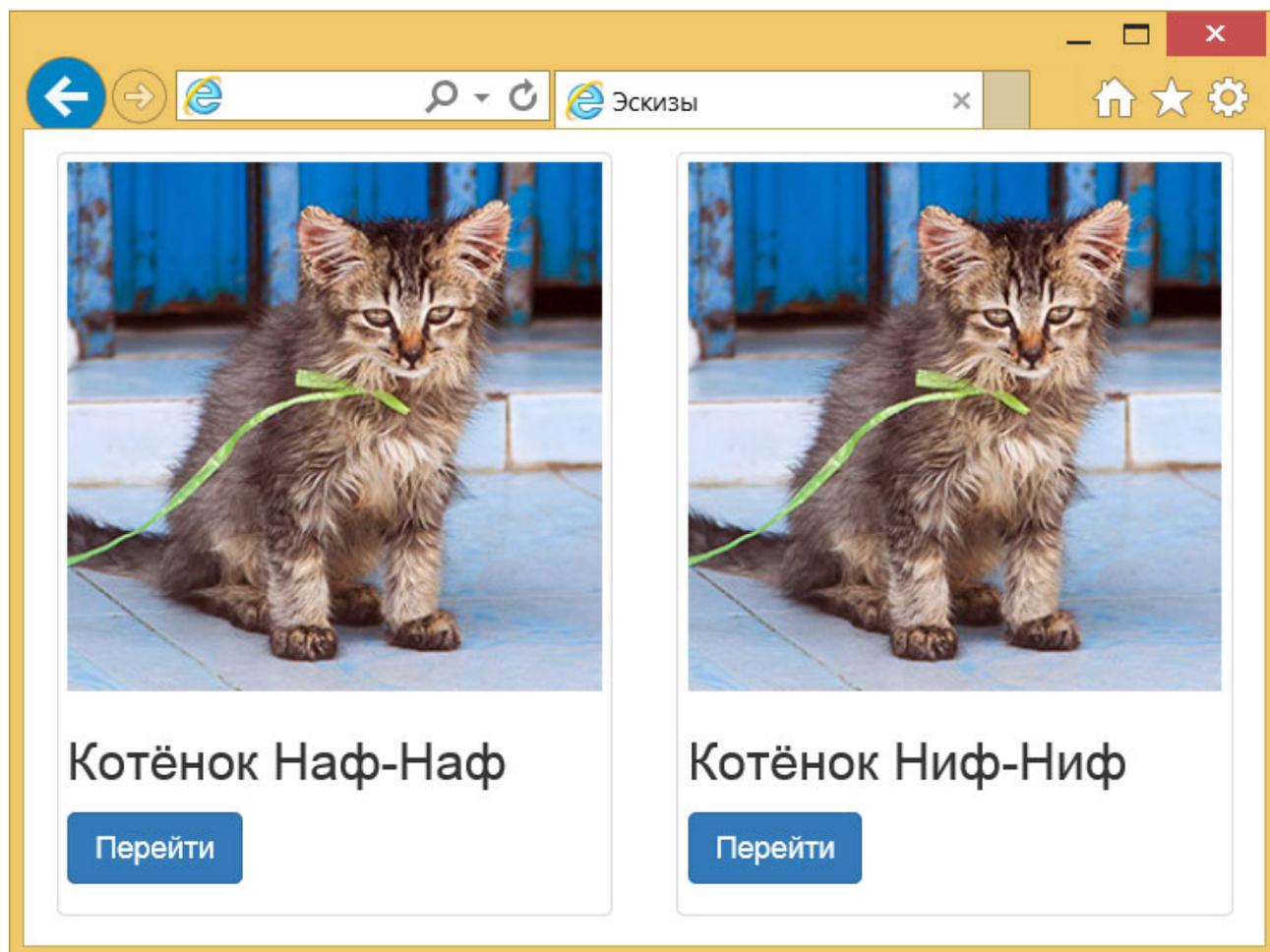


Рис. 3. Эскизы с заголовками

Сообщения

Чтобы дать пользователю знать о результатах его действий применяются информационные сообщения — для разных целей используются разные цвета. К элементу `<div>` добавьте класс `alert` и один из классов показанных в примере 1.

Пример 1. Виды сообщений

```
<div class="alert alert-success" role="alert">У нас всё хорошо</div>  
<div class="alert alert-info" role="alert">Всё в порядке, просто сообщаем об этом</div>  
<div class="alert alert-warning" role="alert">Кажется, что-то пошло не так</div>  
<div class="alert alert-danger" role="alert">Ой-ой, всё плохо!</div>
```

Результат данного примера показан на рис. 1.

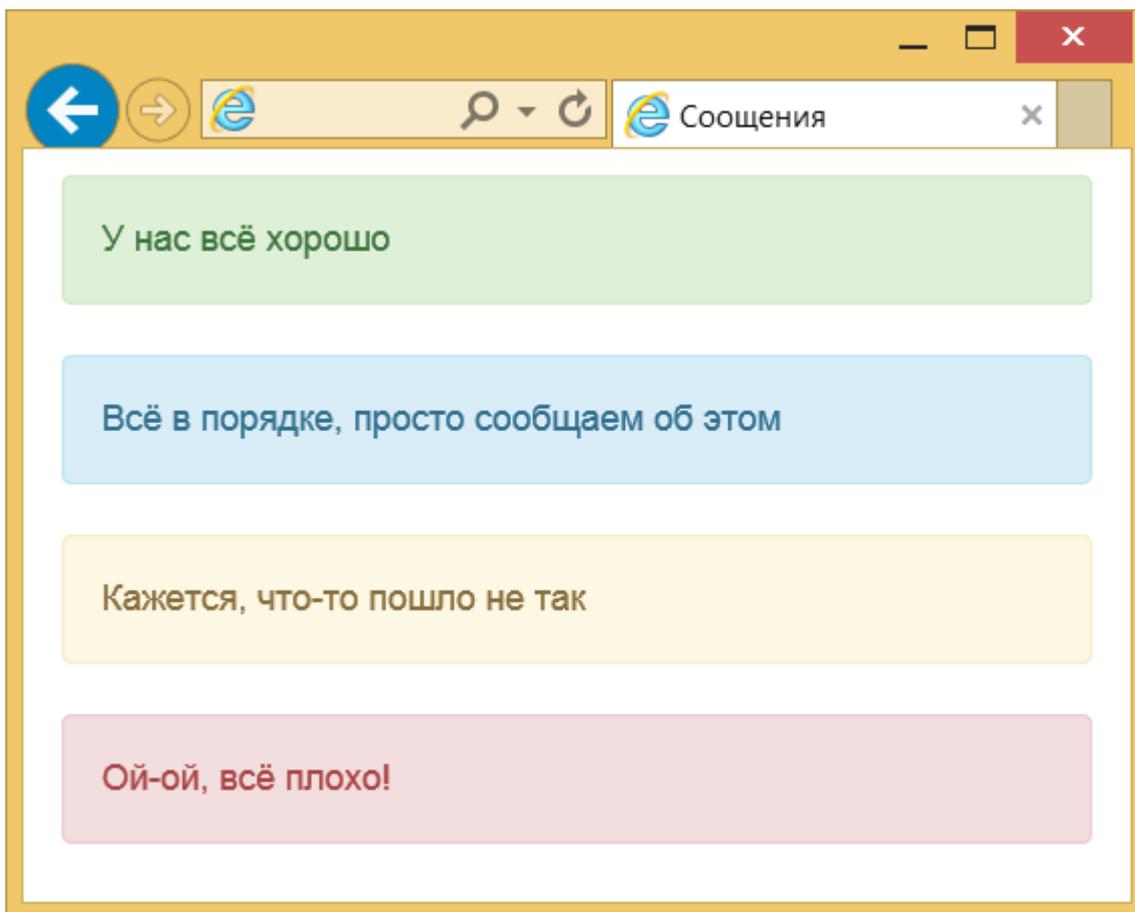


Рис. 1. Вид сообщений

Ссылки в сообщениях

Чтобы ссылки в сообщениях на цветном фоне выглядели гармонично к элементу `<a>` надо добавить класс `alert-link`, как показано в примере 2.

Пример 2. Ссылки

```
<div class="alert alert-success" role="alert"><a href="#" class="alert-link">У нас всё хорошо</a></div>  
<div class="alert alert-info" role="alert"><a href="#" class="alert-link">Всё в порядке, просто сообщаем об этом</a></div>  
<div class="alert alert-warning" role="alert"><a href="#" class="alert-
```

```
link">Кажется, что-то пошло не так</a></div>  
<div class="alert alert-danger" role="alert"><a href="#" class="alert-link">Ой-  
ой, всё плохо!</a></div>
```

Результат данного примера показан на рис. 2.

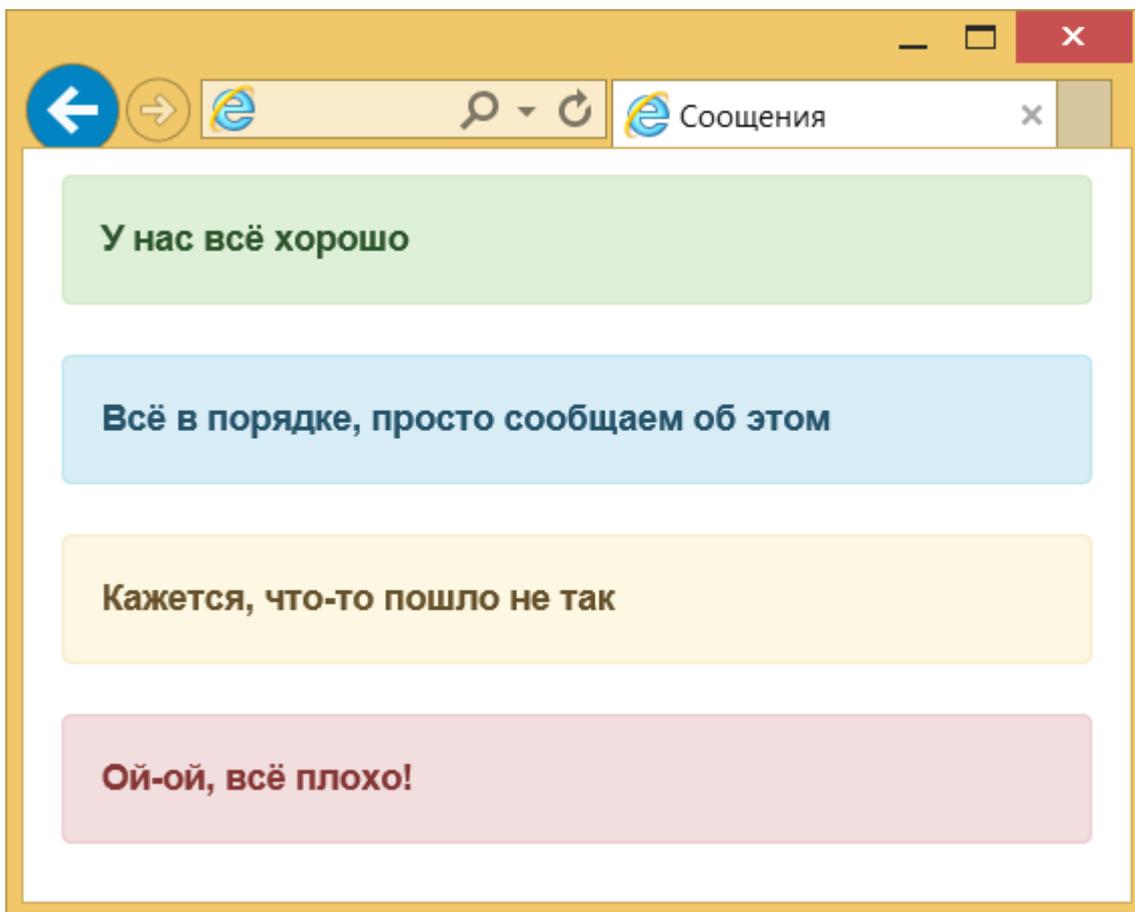


Рис. 2. Вид ссылок в сообщениях

Закрытие сообщений

Исходно пользователь не может убрать сообщения с экрана, чтобы добавить эту возможность вставьте в `<div>` с сообщением класс `alert-dismissible` и сделайте кнопку закрытия, как показано в примере 3.

Пример 3. Закрытие сообщения

```
<div class="alert alert-info alert-dismissible" role="alert">  
  <button type="button" class="close" data-dismiss="alert"><span>&times;</span></  
  button>
```

Совет дня. Теперь вы можете закрыть это надоедливое сообщение, щёлкнув по крестику в углу.

```
</div>
```

Результат данного примера показан на рис. 3.

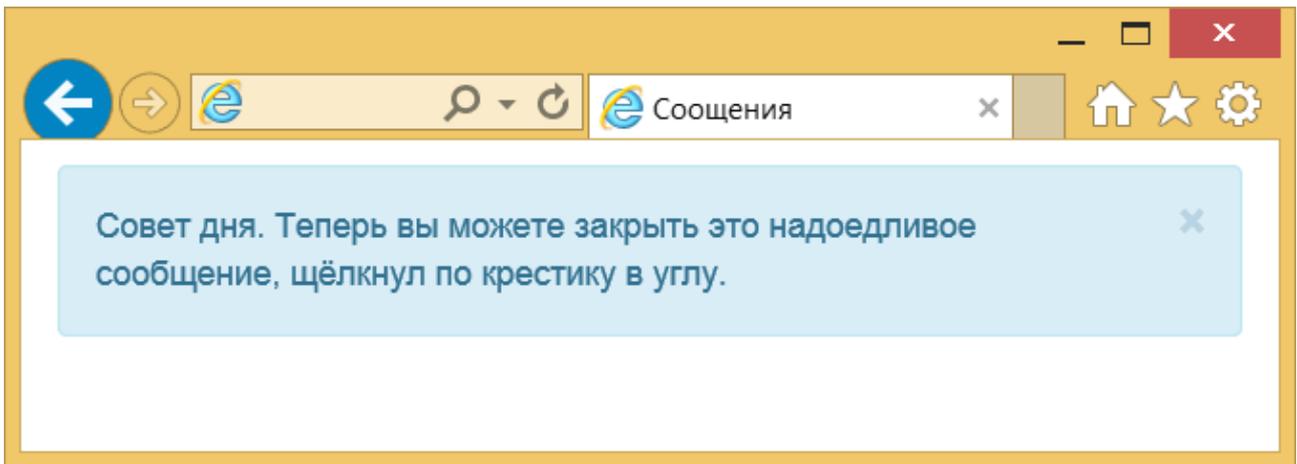


Рис. 3. Сообщение с кнопкой закрытия

После закрытия сообщения вернуть его можно лишь обновив страницу.

Всплывающая подсказка

Всплывающая подсказка может быть добавлена к любому элементу HTML с помощью универсального атрибута `title`. Тогда при наведении курсора на элемент появится текст, заданный внутри `title`. Оформление такого текста возложено на браузер и никак не меняется через стили. Поэтому часто применяют пользовательские подсказки сделанные через JavaScript. Их можно гибко настраивать по своему вкусу, добавляя возможности, которых лишена встроенная подсказка. Например, можно задавать положение подсказки относительно элемента.

Bootstrap переопределяет атрибут `title`, так что текст пишем в этом атрибуте, но также добавляем атрибут `data-toggle` со значением `tooltip`. Он будет сообщать скрипту, что следует изменить вид всплывающей подсказки. Также можно добавить и другие атрибуты управляющие видом, например `data-placement` указывает положение подсказки: `left` — слева; `right` — справа; `top` — сверху; `bottom` — снизу (пример 1). Учтите, что если для вывода подсказки недостаточно места, то она окажется за краем окна браузера.

Пример 1. Создание всплывающей подсказки

```
<p>Популярность языка JavaScript порождена его широкими возможностями по взаимодействию с элементами веб-страницы через <abbr title="Document Object Model, Объектная модель документа" data-toggle="tooltip" data-placement="bottom">DOM</abbr>.</p>
```

Перед закрывающим тегом `</body>` следует добавить также код скрипта, как показано ниже. Без этого наша подсказка работать не будет.

```
<script>
$(function () {
  $('[data-toggle="tooltip"]').tooltip()
})
</script>
```

Вид всплывающей подсказки показан на рис. 1.

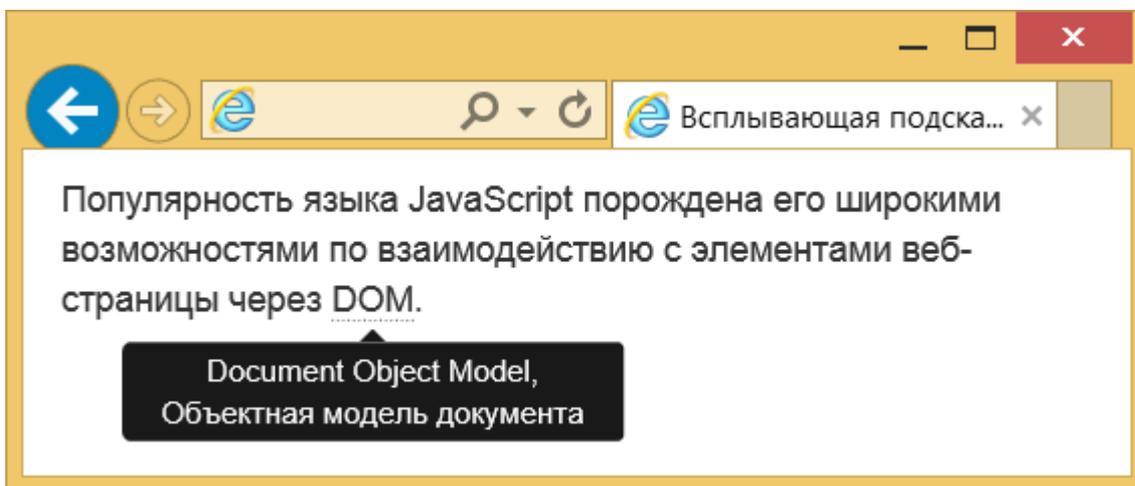


Рис. 1. Всплывающая подсказка внизу текста

Настройки анимации

По умолчанию всплывающая подсказка появляется и исчезает плавно. Вы можете настроить скорость анимации или отключить её вовсе. С этой целью применяется два подхода. В первом случае мы добавляем к элементу атрибуты `data-animation` (`true` — включает анимацию; `false` — отключает её) и `data-delay`, как показано в примере 2.

Пример 2. Настройка анимации

```
<p>Популярность языка JavaScript порождена его широкими возможностями по взаимодействию с элементами веб-страницы через <abbr title="Document Object Model, Объектная модель документа" data-toggle="tooltip" data-placement="bottom" data-animation="false" data-delay="{\"show\": 100, \"hide\": 1000}\">DOM</abbr>.</p>
```

Атрибут `data-delay` задаёт время задержки перед отображением и сокрытием текста в миллисекундах. Если значение одно, то и задержка будет одинакова. Но можно написать структуру как показано в данном примере и отдельно управлять временем задержки отображения и сокрытия.

Во втором подходе параметры анимации добавляются напрямую в скрипт.

```
$(function () {  
  $('[data-toggle="tooltip"]').tooltip({  
    animation: false,  
    delay: {"show": 100, "hide": 1000}  
  })  
})
```