

Приднестровский государственный университет им. Т.Г. Шевченко
Инженерно-технический институт

**РАЗРАБОТКА АДАПТИВНОЙ ВЁРСТКИ ВЕБ-САЙТА
ПРИ ПОМОЩИ ТЕХНОЛОГИЙ HTML5 И CSS3**

Лабораторный практикум

Разработал:
ст. преподаватель
кафедры ИТиАУПП
Бричаг Д.В.

г. Тирасполь
2021 г.

Лабораторная работа №8

Разработка UI веб-сайта при помощи фреймворка Bootstrap 5

Цель работы: научиться разрабатывать вёрстку при помощи новых версий CSS-фреймворков. Научиться работать с документацией внешних фреймворков. Использовать готовые решения для реализации пользовательских интерфейсов.

Теоретическая справка

Сегодня дизайн веб-страниц стал ключом к привлечению аудитории. В гонке за пользовательское внимание на страницах появилось множество визуальных решений, призванных в первую очередь задержать пользователя на странице. По мере того, как дизайн становился всё сложнее, появлялись различные инструменты для реализации такого дизайна в кратчайшие сроки без потери качества исполнения.

С одним из таких инструментов мы познакомились в прошлой лабораторной работе – это Twitter Bootstrap. Помимо него, можно отметить сервис Google Fonts, который рассматривался в первой лабораторной работе. Он позволяет быстро и легко подключать уникальные, для вашего проекта, шрифты.

Кроме того, сам Twitter Bootstrap для создания стилей использует препроцессоры. В первых версиях это был LESS, начиная с четвёртой версии – SASS. О том, что такое препроцессоры CSS можно на странице документации MDN: https://developer.mozilla.org/ru/docs/Словарь/CSS_preprocessor

Это достаточно большая тема и в данном курсе лабораторного практикума мы не будем рассматривать работу с препроцессорами. К тому же это потребует установки дополнительного ПО для компиляции кода в обычный CSS.

Bootstrap

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

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

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

Почему Bootstrap так популярен? Это связано с тем, что он позволяет верстать сайты в несколько раз быстрее, чем на «чистом» CSS и JavaScript. А в нашем мире, время – это очень ценный ресурс. Ещё один его аспект – доступность. Она сводится к тому, что предоставляет возможность даже начинающему веб-разработчику (без глубоких знаний и достаточной практики) создавать достаточно качественные макеты.

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

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

Создание кнопки в Bootstrap 5:

```
<!-- Чтобы сделать ссылку в виде кнопки добавим к ней 2 класса: btn и btn-success -->  
<a href="#" class="btn btn-success">Ссылка, оформленная в виде кнопки</a>
```

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

А если, например, нужно создать вкладки. Для этого необходимо всего лишь скопировать готовый HTML фрагмент, и отредактировать его так, чтобы в нём остался код для определённого количества вкладок:

```

<!-- Bootstrap 5 -->
<nav>
  <div class="nav nav-tabs" id="tab" role="tablist">
    <!-- атрибут data-bs-target должен указывать на элемент, который необходимо показать -->
    <button class="nav-link active" id="tab-section-1" data-bs-toggle="tab" data-bs-target="#section-1" t
    <button class="nav-link" id="tab-section-2" data-bs-toggle="tab" data-bs-target="#section-2" type="bu
    <button class="nav-link" id="tab-section-3" data-bs-toggle="tab" data-bs-target="#section-3" type="bu
  </div>
</nav>
<div class="tab-content p-3" id="nav-tabContent">
  <!-- атрибут id определяет вкладку -->
  <div class="tab-pane fade show active" id="section-1" role="tabpanel" aria-labelledby="nav-section-1-ta
    ...
  </div>
  <div class="tab-pane" id="section-2" role="tabpanel" aria-labelledby="nav-section-2-tab">
    ...
  </div>
  <div class="tab-pane fade" id="section-3" role="tabpanel" aria-labelledby="nav-section-3-tab">
    ...
  </div>
</div>

```

Рис. 149 – Разметка блока с вкладками

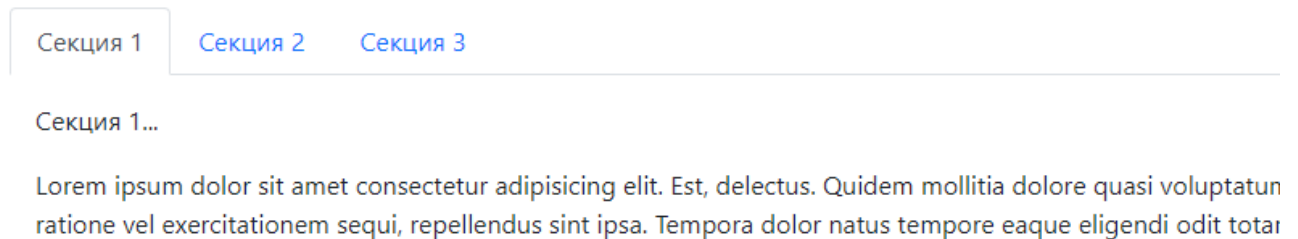


Рис. 150 – Блок с вкладками

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

Но, Bootstrap – это не просто набор готовых инструментов (HTML фрагментов, классов, компонентов и плагинов), а хорошо спроектированный фронтенд фреймворк, который довольно просто можно настроить под себя посредством редактирования Sass переменных и использования миксинов.

Из чего состоит Bootstrap

Bootstrap состоит из:

- инструментов для создания макета (обёрточных контейнеров, мощной системы сеток, гибких медиа-объектов, адаптивных утилитных классов);

- классов для стилизации базового контента: текста, изображений, кода, таблиц и `figure`;
- готовых компонентов: кнопок, форм, горизонтальных и вертикальных навигационных панелей, слайдеров, выпадающих списков, аккордеонов, модальных окон, всплывающих подсказок и др.;
- утилитных классов для решения традиционных задач наиболее часто возникающими перед веб-разработчиками: выравнивание текста, отображение и скрытие элементов, задания цвета, фона, `margin` и `padding` отступов, и т.д.

С чего начать изучение Bootstrap

Первое, что нам необходимо — это скачать фреймворк и подключить его к странице. Как это выполнить детально описано в практической части.

После того как мы это сделаем обычно возникает следующий вопрос: «А что дальше? Как использовать Bootstrap?». На самом деле тут нет ничего сложного, создавать страницы с использованием Bootstrap довольно просто.

Сначала необходимо изучить сетку и попрактиковаться на ней верстать макеты страниц и создавать разметку блоков. Изучить её можно <https://itchief.ru/bootstrap/grid-v3> (для тех, кто использует Bootstrap 3) или <https://itchief.ru/bootstrap/grid-v4> (для Bootstrap 4).

К сведению, сетка в «Bootstrap 3.x» построена на `float`, а в «4.x» и «5.x» — на `flexbox`.

После того как разберётесь с сеткой нужно изучить компоненты Bootstrap (как они верстаются и настраиваются) и научиться их вставлять в нужные места страницы.

Это минимальные задачи, которые перед собой необходимо поставить, чтобы научиться верстать сайты и веб-приложения на Bootstrap.

Ну если ваша задача не просто использовать Bootstrap, а что-то в нём изменять, переделывать, то без хороших знаний CSS тут конечно уже не обойтись.

Некоторые вещи, конечно, можно настроить через изменения SCSS переменных и применения миксинов, а другие – нет. И это касается любых фреймворков.

Преимущества и недостатки фреймворка Bootstrap

Преимущества Bootstrap при его использовании для frontend разработки сайтов и интерфейсов административных панелей:

- высокая скорость создания качественной адаптивной вёрстки даже начинающими веб-разработчиками (достигается это благодаря использованию готовых классов и компонентов, созданных профессионалами);
- кроссбраузерность и кроссплатформенность (корректное отображение и работа сайта во всех поддерживаемых этим фреймворком браузерах и операционных системах);
- наличие большого количество готовых хорошо продуманных компонентов, протестированных огромным сообществом веб-разработчиков на различных устройствах;
- возможность настройки под свой проект, достигается это посредством изменения SCSS переменных и использования миксинов (можно изменить количество колонок, цвета, радиус скруглений, отступы между колонками и т.д.);
- низкий порог вхождения; для работы с фреймворком не обязательно иметь «глубокие» знания по HTML, CSS, JavaScript и jQuery (достаточно знать только основы этих технологий);
- однородность дизайна и его согласованность между различными компонентами (в Bootstrap все компоненты выполнены в едином стиле);
- наличие огромного количества сообществ и обучающих материалов; при желании это поможет не только хорошо разобраться в фреймворке, но и найти ответы практически на любые возникающие у вас вопросы.

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

Bootstrap, конечно, хороший инструмент, но на нём верстают далеко не всё. Его, например не имеет смысла использовать для:

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

Bootstrap, как и большинство подобных фреймворков, имеет недостатки. Среди них можно отметить следующие:

- более большой размер конечных css и js-файлов проекта, чем они получились, если бы мы всё это создавали самостоятельно (это связано с тем, что стили фреймворка и его js-код содержат универсальный код (на все случаи жизни), а по факту для конкретного проекта из всего этого может потребоваться только часть);
- сложность использования Bootstrap для вёрстки сайтов с уникальным дизайном, т.к. разработка в этом случае будет сопровождаться значительным переписыванием его кода и простой настройкой Bootstrap переменных тут уже не обойтись.

Первый недостаток можно немного «сгладить», если выполнить самостоятельную сборку проекта из исходных кодов и включить в неё только те компоненты, которые нам нужны. В этом случае код этого фреймворка будет значительно меньше.

Какую версию Bootstrap выбирать для нового проекта

Для разработки большинства веб-проектов лучше использовать последнюю версию Bootstrap в выбранной ветке.

На текущий момент:

- в v5 - это 5.1.3;
- в v4 - это 4.6.1;
- в v3 - это 3.4.1.

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

Что нового в Bootstrap 5:

- нет зависимости от библиотеки jQuery, все плагины переписаны на чистом JavaScript;
- библиотека Popper обновлена до v2.x с v1.x;
- не поддерживаются Internet Explorer 10 и 11, Microsoft Edge <16, Firefox <60, Safari <10, iOS Safari <10, Chrome <60 и Android <6;
- добавлен новый брекпойнт в сетку (xxl для 1400px и выше);
- новые классы .g-*, .gx-* и .gy-* для указания горизонтальных и вертикальных отступов между ячейками;
- добавлен новый компонент offcanvas;
- удален компонент Media, его реализацию предлагают выполнять с помощью утилитных классов Bootstrap;
- улучшен раздел по формам в документации, и добавлена возможность создания в input плавающего label.

Третью версию в основном имеет смысл использовать, если нужна поддержка «старых» браузеров (IE8 и IE9).

Основные отличия Bootstrap 4 от Bootstrap 3:

- не поддерживает IE8, IE9 и iOS6 (v4 теперь только IE10+ и iOS7+);
- сетка и другие компоненты построены на флексах, а не на float;
- исходный код стилей написан на Sass, а не на Less;
- размеры, указываются в rem и em, а не в px;
- размер шрифта увеличен с 14px до 16px;

- увеличено количество брекпойнтов в сетки (в v4 - 5, в v3 - 4);
- удалены компоненты Panels, Thumbnails и Wells, вместо них добавлен Cards;
- удалены иконочный шрифт Glyphicons и jQuery плагин Affix;
- обновлены почти все компоненты;
- работа всплывающих подсказок (Tooltips) и блоков с контентом (Popovers) построена на библиотеке Popper JS.

Компоненты Bootstrap 3 в IE8 и некоторых других версиях отображаются без градиентов, теней и закруглённых углов. Это связано с тем, что в них нет поддержки используемых для этих целей CSS3 свойств.

SVG

SVG — XML язык разметки на основе векторной графики. По сути это текстовый файл, который является открытым веб-стандартом для описания двухмерных векторных изображений без потери качества при масштабировании. Также поддерживает интерактивность и анимацию. Разрабатывается с 1999 года и используется для того, чтобы описать с помощью языка программирования графические элементы сайта. Название образовано от акронима Scalable Vector Graphics. То есть тип таких изображений векторный. Обычно для SVG не используется сжатие, потому что здесь информация хранится как набор координат, но может быть сжат без потерь. Основные характеристики SVG документа:

1. размер SVG-файла не зависит от размера или глубины цвета. На вес влияет сложность точек и линий, с которых состоит изображение;
2. во время масштабирования не теряет информацию, поэтому заменяет собой несколько вариантов растровых файлов для разных разрешений экрана;
3. подходит для печати и индексации поисковыми системами;
4. может быть встроен в html-код сайта, а значит, кешируется вместе с ним и загружается быстрее, чем изображения в .png или .jpeg.

Преимущества и недостатки формата .svg

Плюсы:

- масштабируемость без изменения качества — один и тот же логотип будет хорошо выглядеть и на экране смартфона, и на большом экране Retina;
- простота создания: программисты пишут иконки с помощью xml-кода, веб-дизайнеры рисуют более сложные элементы в редакторах векторной графики Adobe Illustrator, Corel Draw, Sketch или Macromedia Freehand;
- малый размер: объекты .svg занимают гораздо меньше места, чем их близнецы, созданные как объекты растровой графики;
- гибкость: при помощи CSS можно изменить параметр графики на сайте, например, цвет фона или позицию логотипа на странице, а при помощи javascript задать анимацию элементов. Также можно отредактировать SVG-файл в графическом редакторе;
- четкая прорисовка деталей и лучший результат печати.

Минусы:

- не поддерживается старыми браузерами — Internet Explorer 8 и старше;
- WordPress воспринимает .svg как вредоносный код, поэтому блокирует отображение. Обойти блокировку можно с помощью плагинов;
- подходит для создания примитивных объектов, которые могут быть описаны простыми фигурами или их частями: окружность, прямая, эллипс, прямоугольник и т.д.

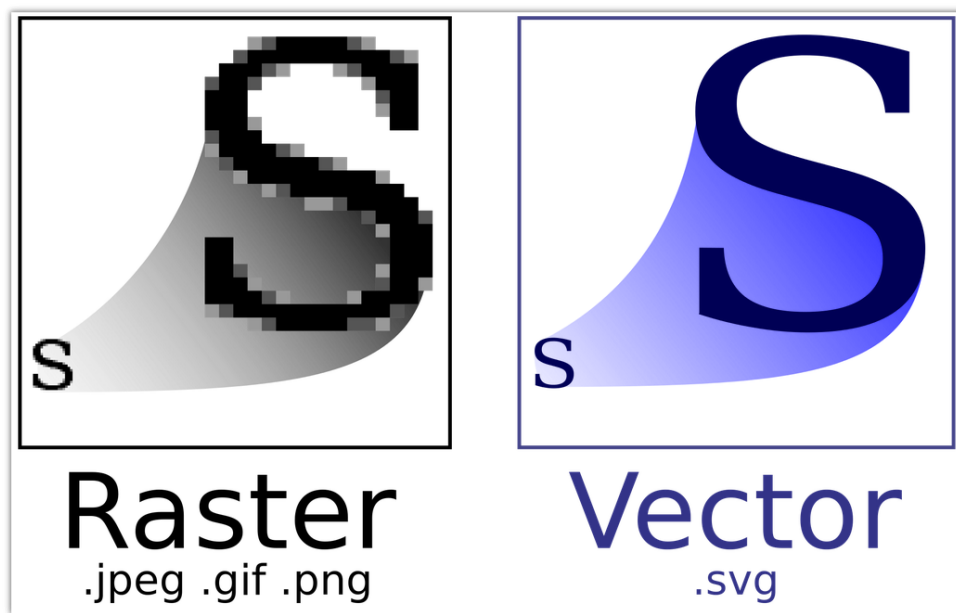


Рис. 151 – Растровая и векторная графика при увеличении

Практическая часть

Начнём практическую часть с создания новой страницы со слайдерами. Здесь мы создадим два слайдера. Сначала стандартными средствами HTML/CSS/JavaScript, а затем при помощи Bootstrap, чтобы наглядно увидеть разницу двух подходов разработки.

Вы можете создать страницу с названием по вашему желанию. Здесь, для примера, будет создана страница `gallery.html`, на которую можно попасть по ссылке «галерея» в меню.

Соответственно будут поправлены ссылки в меню заголовка и подвала всех трёх предыдущих страниц (главной, контактов и блога).

Также будет скопирована основная разметка с обложкой. После чего под заголовком добавим слайдер с возможностью перелистывать картинки по порядку как на рисунке ниже:

```
36     <main>
37         <section id="top">
38             
39         </section>
40         <section class="slider-section">
41             <h1 class="main-title">Наши работы</h1>
42             <div class="delimiter"></div>
43             <p class="main-title-description">Перечень изображений в слайдере</p>
44             <div class="slider-container">
45                 <div class="slider-arrow left"></div>
46                 <div class="slider-arrow right"></div>
47                 <div class="slider-image"></div>
48             </div>
49         </section>
50     </main>
```

Рис. 152 – Подключение внешней библиотеки

Однако стилей будет достаточно много. Также всю «магию» перелистывания организует JavaScript в связке с jQuery. Посмотрите, какие нам потребуются стили для десктопной версии:

	slider.js	gallery.css	blog.js	blog.html
1		<code>.slider-section {</code>		
2		<code>margin: 50px 0;</code>		
3		<code>}</code>		
4		<code>.slider-container {</code>		
5		<code>max-width: 1200px;</code>		
6		<code>height: 375px;</code>		
7		<code>margin: 50px auto;</code>		
8		<code>position: relative;</code>		
9		<code>}</code>		
10		<code>.slider-image {</code>		
11		<code>height: 100%;</code>		
12		<code>width: 100%;</code>		
13		<code>background-image: url('../img/blog/blog00.jpg');</code>		
14		<code>background-position: center center;</code>		
15		<code>background-size: cover;</code>		
16		<code>}</code>		
17		<code>.slider-arrow {</code>		
18		<code>position: absolute;</code>		
19		<code>top: 0;</code>		
20		<code>bottom: 0;</code>		
21		<code>cursor: pointer;</code>		
22		<code>width: 30%;</code>		
23		<code>opacity: 0;</code>		
24		<code>transition: opacity 350ms ease-in-out;</code>		
25		<code>}</code>		
26		<code>.slider-arrow.left {</code>		
27		<code>left: 0;</code>		
28		<code>background: linear-gradient(90deg, rgba(0,18,36,1) 0%, rgba(0,212,255,0) 100%);</code>		
29		<code>}</code>		
30		<code>.slider-arrow.right {</code>		
31		<code>right: 0;</code>		
32		<code>background: linear-gradient(-90deg, rgba(0,18,36,1) 0%, rgba(0,212,255,0) 100%);</code>		
33		<code>}</code>		
34		<code>.slider-container:hover .slider-arrow {</code>		
35		<code>opacity: 1;</code>		
36		<code>}</code>		

Рис. 153 – Таблица стилей слайдера

Откройте страницу и взгляните на результат. Можете, как и раньше изменить стили, например увеличить или уменьшить размер.



Наши работы

Перечень изображений в слайдере



Рис. 154 – Внешний вид слайдера

По наведению на картинку появляются «тени» управления. Можно вместо них использовать картинки со стрелками или индикатор количества изображений в слайдере. Здесь, для примера, я использовал боковые «тени» затемнения, как частый пример в сети. Тем более, что их реализация требует наименьшего количества кода. Это то, что нам нужно, для создания простого слайдера. Более сложные слайдеры обычно создаются при помощи готовых библиотек или заготовок.

Перейдём в head страницы gallery.html и добавим путь к скрипту slider.js. Соответственно в проект в каталог js добавим и соответствующий файл.

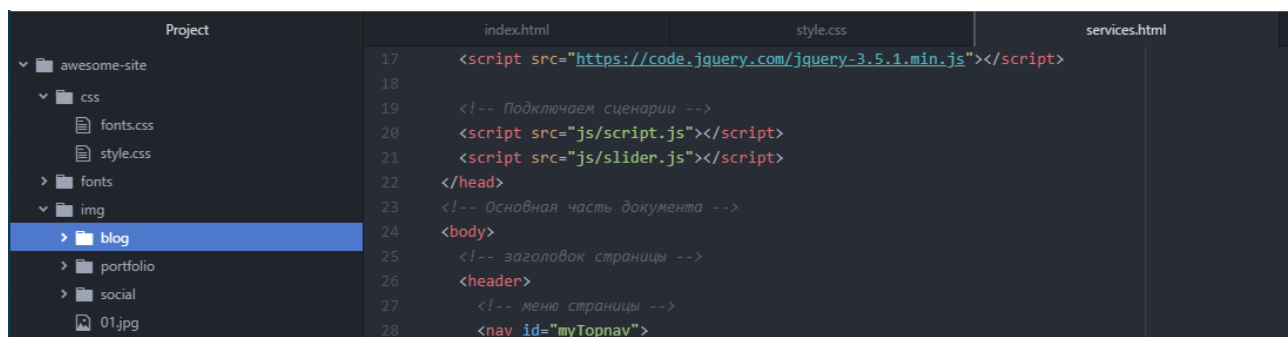


Рис. 155 – Подключение нового js файла

В нём напишем код как на рисунке 156, создаём функцию ожидания загрузки документа, и в ней объявляем переменные. Отдельной переменной был создан массив с перечислением путей картинок. В примере взяты картинки от блогов, вы вольны использовать любые картинки, просто укажите их пути в массиве `slides`.



```
slider.js  •  blog.css  gallery.css
1  $(document).ready(() => {
2    let currentSlide = 0;
3    let isBusy = false;
4
5  });
6
7  const slides = [
8    'img/blog/blog00.jpg',
9    'img/blog/blog01.jpg',
10   'img/blog/blog02.jpg',
11   'img/blog/blog03.jpg',
12   'img/blog/blog04.jpg',
13   'img/blog/blog05.jpg',
14   'img/blog/blog06.jpg',
15   'img/blog/blog07.jpg',
16  ]
```

Рис. 156 – Подключение нового js файла

Затем создаём обработчик кликов по боковым тням – переключателям. По клику выполняем функцию и передаём параметр `e`. Этот параметр – объект события клик мыши. В нём содержится информация о событии: какая клавиша кликнула, по чему, связанные объекты, время клика и тому подобное. В частности **`e.currentTarget`** – это сама тень, по которой был произведён щелчок мыши. Передавая этот объект в функцию **jQuery** – создаётся **jQuery** объект, которым можно манипулировать всеми методами библиотеки. Поэтому создаём переменную, в которую передаём этот элемент.

```

1  $(document).ready(() => {
2      let currentSlide = 0;
3      let isBusy = false;
4
5      $('.slider-arrow').on('click', (e) => {
6          const that = $(e.currentTarget);
7      });
8
9  });

```

Рис. 157 – Создание обработчика события «клик»

После этого создаём логику переключения слайдов. Для начала устанавливаем условие, если слайдер не занят «перелистыванием». Это необходимо чтобы предотвратить «слипание» кликов, когда слайд ещё не перелистнулся.

В функции слайдера на блоки с затемнением устанавливается обработчик кликов, который по левому щелчку мыши вызовет анонимную функцию.

Далее проверяем, если это правая тень – наращиваем страницу, если нет – уменьшаем. При этом заботимся о крайних случаях: если новая страница больше количества слайдов – возвращаем в положение 0. И наоборот, если меньше 0, то устанавливаем значение последнего слайда.

```

9      if (!isBusy) {
10
11          if (that.hasClass('right')) {
12              currentSlide += 1;
13              if (currentSlide > slidesCount) currentSlide = 0;
14          } else {
15              currentSlide -= 1;
16              if (currentSlide < 0) currentSlide = slidesCount;
17          }
18      }
19  });

```

Рис. 158 – Логика переключения номера слайда

И наконец добавляем логику анимации перелистывания. Устанавливаем флаг `isBusy` в логическое состояние `true` и убираем лишь тогда, когда завершится анимация. А самой картинке мы сначала задаём прозрачность 0 (абсолютно прозрачный), затем меняем картинку, а после устанавливаем прозрачность 1 (не прозрачный).

```
5   $('.slider-arrow').on('click', (e) => {
6     const that = $(e.currentTarget);
7     const slidesCount = slides.length - 1;
8
9     if (!isBusy) {
10
11       if (that.hasClass('right')) {
12         currentSlide += 1;
13         if (currentSlide > slidesCount) currentSlide = 0;
14       } else {
15         currentSlide -= 1;
16         if (currentSlide < 0) currentSlide = slidesCount;
17       }
18       isBusy = true;
19       $('.slider-image').animate({'opacity': 0}, 350, () => {
20         $('.slider-image').css('background-image', 'url(' + slides[currentSlide] + ')');
21         $('.slider-image').animate({'opacity': 1}, 350, () => isBusy = false);
22       });
23     }
24   });
25 });
```

Рис. 159 – Переключение и анимация

В теле функции происходит чтение длины массива изображений, для дальнейшей логики. В строке номер 7 – проверка не занят ли слайдер переключением слайдов. В слайдере задействована анимация переключения изображений: метод **animate()** из библиотеки **jQuery** принимает несколько параметров, первым – какое CSS свойство нужно изменить, вторым – с какой задержкой, и третьим – `callBack` функцию, которая будет выполнена по окончании анимации.

Таким образом, строка 19 выполнится не сразу, а только когда картинка будет скрыта. В этот момент JS изменит картинку на другую и запустит новую анимацию – чтобы изображение появилось. По окончании анимации в `callBack`

изменяется состояние переменной `isBusy` на `false`. Теперь слайдер готов обрабатывать новый клик.

После этого слайдер должен плавно переключать картинки по клику на «тени». Таких реализаций можно придумать бесконечное множество. Абсолютно правильного варианта нет, и всё ограничивается лишь фантазией разработчика. Однако, каждый раз придумывать новую логику для слайдера может быть слишком затратным. Поэтому давайте рассмотрим использование готового решения из библиотеки.

Для этого воспользуемся уже упомянутой в теоретической справке библиотекой Bootstrap.

Библиотеку Bootstrap подключаем, как и раньше внешние зависимости: либо ссылками на официальный репозиторий, либо добавлением в проект.

Ссылки можно найти по адресу <https://v5.getbootstrap.com/>



Рис. 160 – Ссылки для подключения фреймворка

Для CSS стилей:

```
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/5.0.0-alpha2/css/bootstrap.min.css" integrity="sha384-DhY6onE6f3zzKbjUPRc2hOzGAdEf4/Dz+WJwBvEYL/lkkIsI3ihufq9hk9K41VoK" crossorigin="anonymous">
```

Для JavaScript

```
<script src="https://stackpath.bootstrapcdn.com/bootstrap/5.0.0-alpha2/js/bootstrap.bundle.min.js" integrity="sha384-BOsAfwezjNJHrJ8cZidOg56tcQWfp6y72vEJ8xQ9w6Quywb24iOsW913URv1IS4GD" crossorigin="anonymous"></script>
```

Но добавим не для всех страниц, а для текущей новой страницы gallery.html. В шапку этой страницы подключаем **стили** Bootstrap, а после футера, перед закрывающим тегом </body> — **скрипты** (это рекомендация самой библиотеки).

```
1 <!DOCTYPE html>
2 <html lang="ru" dir="ltr">
3   <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>My very first page</title>
7     <link rel="stylesheet" type="text/css" href="css/fonts.css">
8     <link rel="stylesheet" type="text/css" href="css/style.css">
9     <link rel="stylesheet" type="text/css" href="css/gallery.css">
10
11     <link href="https://fonts.googleapis.com/css?family=Roboto:300,400,500&display=swap&subset=
12     <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/5.0.0-alpha2/css/
13     <!-- подключаем сценарии -->
14     <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
15     <script src="js/script.js"></script>
16     <script src="js/slider.js"></script>
17   </head>
18   <!-- Основная часть документа -->
19   <body>
20     <!-- заголовок страницы -->
21   >   <header>=
37     <!-- основной контент сайта -->
38   >   <main>=
52     </main>
53     <!-- подвал страницы -->
54   >   <footer>=
97     <script src="https://stackpath.bootstrapcdn.com/bootstrap/5.0.0-alpha2/js/bootstrap.bundle.
98   </body>
99 </html>
```

Рис. 161 – Подключение библиотеки к странице gallery.html

Теперь создадим ещё одну секцию с заголовком и новым слайдером, но уже при помощи bootstrap, чтобы почувствовать скорость разработки с помощью этой библиотеки.

Создадим внутри тега <main> под предыдущим слайдером следующую вёрстку:

```

52 <section class="slider-section">
53 <h1 class="main-title">Наши работы</h1>
54 <div class="delimiter"></div>
55 <p class="main-title-description">Перечень изображений в слайдере</p>
56 <div id="carouselExampleIndicators" class="carousel slide" data-ride="carousel">
57
58 <div class="carousel-inner">
59 <div class="carousel-item active">
60 
61 </div>
62 <div class="carousel-item">
63 
64 </div>
65 <div class="carousel-item">
66 
67 </div>
68 </div>
69
70 </div>
71 </section>

```

Рис. 162 – Создание разметки слайдера

Сохраним, откроем и видим, что на нашей странице добился блок с картинкой. К тому же, по истечению нескольких секунд картинки меняются с плавным эффектом.

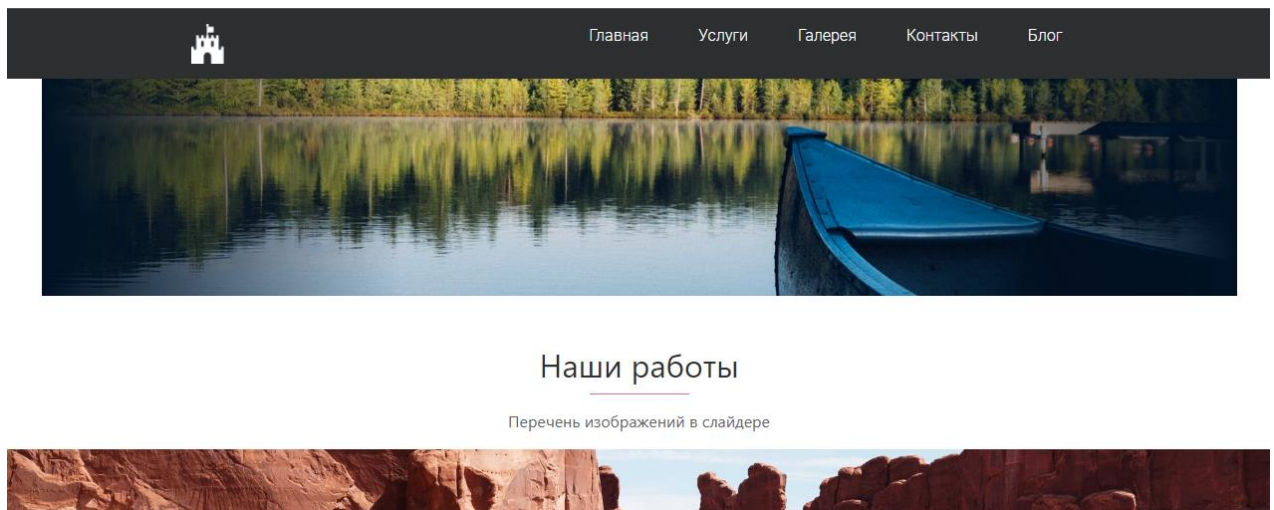


Рис. 163 – Внешний вид слайдера

Мы не написали ни единой строчки кода на CSS и JavaScript, но уже получили рабочий слайдер. Такими же лёгкими действиями мы можем добавить «**Стрелочки**» переключения и подсказки статуса слайдера. Просто добавив такой код:

```

58 <div id="carouselExampleIndicators" class="carousel slide" data-ride="carousel">
59   <ol class="carousel-indicators">
60     <li data-target="#carouselExampleIndicators" data-slide-to="0" class="active"></li>
61     <li data-target="#carouselExampleIndicators" data-slide-to="1"></li>
62     <li data-target="#carouselExampleIndicators" data-slide-to="2"></li>
63   </ol>
64   <div class="carousel-inner">
65     <div class="carousel-item active">
66       
67     </div>
68     <div class="carousel-item">
69       
70     </div>
71     <div class="carousel-item">
72       
73     </div>
74   </div>
75   <a class="carousel-control-prev" href="#carouselExampleIndicators" role="button" data-slide="prev">
76     <span class="carousel-control-prev-icon" aria-hidden="true"></span>
77     <span class="visually-hidden">Previous</span>
78   </a>
79   <a class="carousel-control-next" href="#carouselExampleIndicators" role="button" data-slide="next">
80     <span class="carousel-control-next-icon" aria-hidden="true"></span>
81     <span class="visually-hidden">Next</span>
82   </a>
83 </div>

```

Рис. 164 – Добавление инструментов управления слайдером

Переключимся к нашей странице, и увидим кнопки стрелочки и указатель текущего активного слайдера внизу картинки, который тоже можно использовать для переключения слайдов.

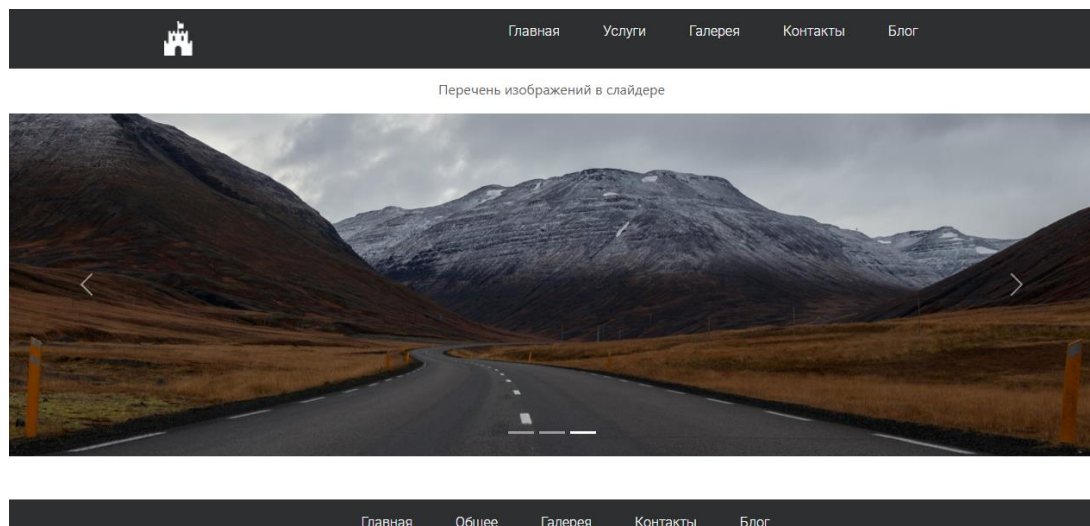


Рис. 165 – Доработанный слайдер с управлением переключением

Всё работает «из коробки» – в библиотеке уже прописаны стили и классы для всех элементов, а слайдер является шаблонной заготовкой.

Добавим только два стиля для контейнера слайдера, чтобы он принял такой же размер, как предыдущий.

```
37 .carousel {  
38   max-width: 1200px;  
39   margin: 50px auto;  
40 }
```

Рис. 166 – Ограничение максимальной ширины слайдера

Обратите внимание на названия классов, в них есть подсказки за что они отвечают. Например, класс `d-block` у картинок имеет свойство `display: block`, а `w-100` – `width: 100%`.

Как узнать какие классы за что отвечают? Для этого вы можете воспользоваться документацией Bootstrap. Там описаны все классы, свойства и параметры скриптов для настройки тех или иных компонентов.

Для каждой версии библиотеки есть соответствующая версия документации. Для текущей редакции, соответственно, документация 5 по адресу:

<https://getbootstrap.com/docs/5.1/getting-started/introduction/>

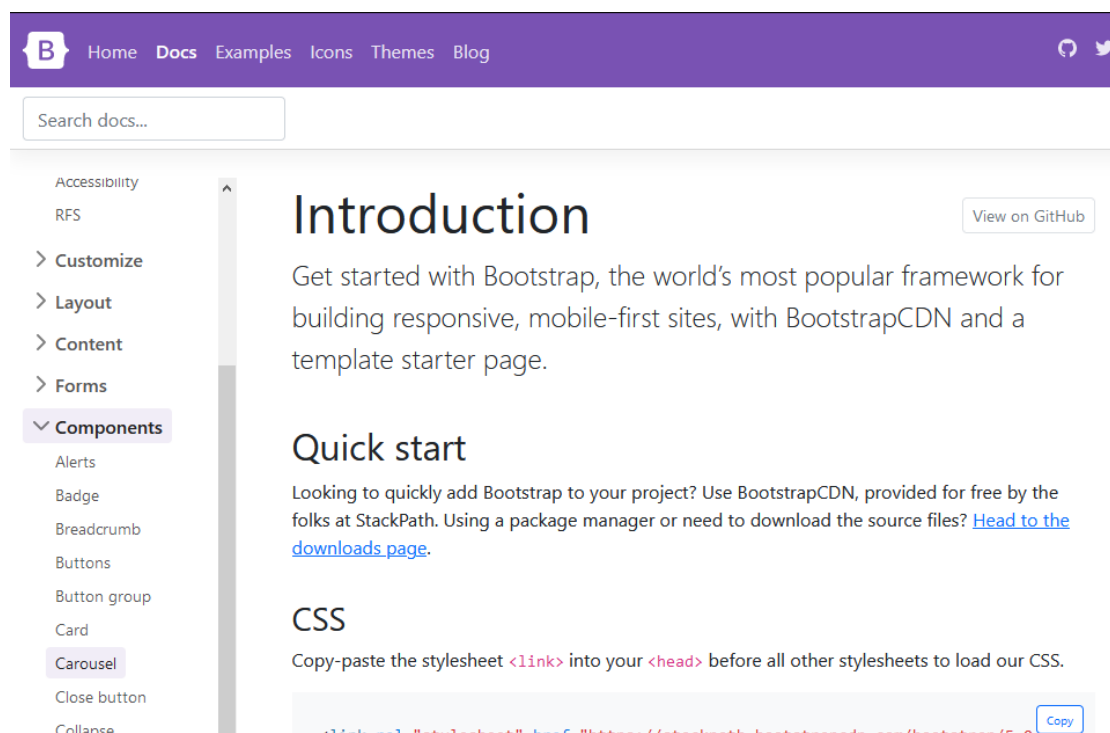


Рис. 167 – Навигация по документации библиотеки

По правде говоря, тут можно найти и тот самый слайдер, который мы только что создали. Перейдите в левом меню в раздел Components, пункт Carousel. Здесь описаны шаги для создания такого слайдера. А также указаны опции, которые можно добавить, например стрелочки, подсказки или текст. Прокрутите страницу до параграфа «With captions» - с подсказками. Здесь описано, что нужно добавить, чтобы появился блок с текстом у слайдера. Живой пример и необходимый HTML код.

With captions

Add captions to your slides easily with the `.carousel-caption` element within any `.carousel-item`. They can be easily hidden on smaller viewports, as shown below, with optional [display utilities](#). We hide them initially with `.d-none` and bring them back on medium-sized devices with `.d-md-block`.



Рис. 168 – Типовой слайдер в документации

Воспользуйтесь примером из документации и добавьте самостоятельно подсказки к вашему слайдеру. Текст подсказок не имеет значения.

Здесь же в меню вы можете найти другие компоненты Bootstrap такие как кнопки, выпадающие меню и формы. Кроме того, на официальном сайте можно использовать и некоторые готовые решения.

Откройте страницу «**Examples**», найти её можно по ссылке из верхнего меню или по адресу <https://getbootstrap.com/docs/5.1/examples/>

Здесь представлены целые блоки популярных пользовательских интерфейсов: меню заголовка, меню подвала, заголовки и обложки страницы, страницы блога, авторизации и администрирования. В начале страницы есть кнопка скачать для загрузки исходного кода всех этих элементов.

Однако, если необходим только один элемент, мы можем поступить проще и скопировать его прямо со страницы. Так как и наша страница, и страница примера используют одну и ту же библиотеку, то и отображаться такой блок будет идентично.

Откройте раздел «**Pricing**», нажмите правой клавишей мыши на блок с ценами и выберите в контекстном меню пункт «Просмотреть код», если вы пользуетесь браузером Google Chrome.

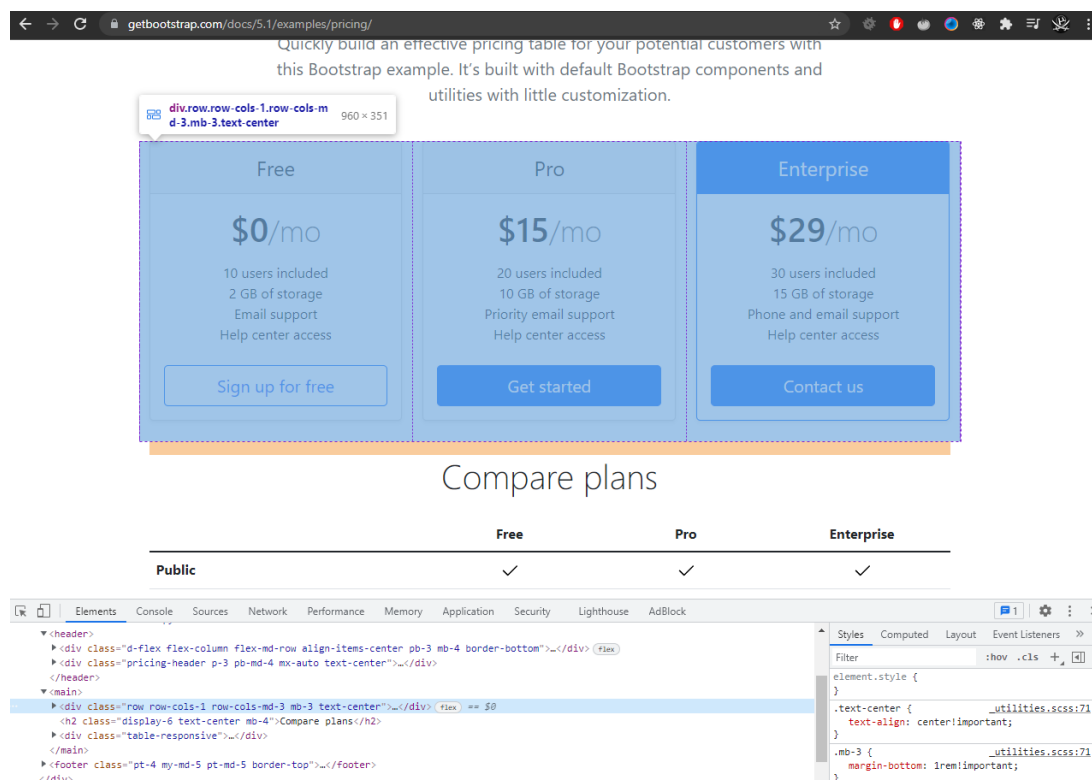


Рис. 169 – Фрагмент блока цен на странице примеров

В консоли разработчика выделите блок внутри тега main и скопируйте в новую секцию на странице gallery.html

```
71     </section>
72     <section class="container">
73         <div class="row row-cols-1 row-cols-md-3 mb-3 text-center">
74             <div class="col">
75                 <div class="card mb-4 rounded-3 shadow-sm">
76                     <div class="card-header py-3">
77                         <h4 class="my-0 fw-normal">Free</h4>
78                     </div>
79                     <div class="card-body">
80                         <h1 class="card-title pricing-card-title">$0<small class="text-mute
81                         <ul class="list-unstyled mt-3 mb-4">
82                             <li>10 users included</li>
83                             <li>2 GB of storage</li>
84                             <li>Email support</li>
85                             <li>Help center access</li>
86                         </ul>
87                         <button type="button" class="w-100 btn btn-lg btn-outline-primary">
88                     </div>
89                 </div>
90             </div>
91             <div class="col">
92                 <div class="card mb-4 rounded-3 shadow-sm">
```

Рис. 170 – Фрагмент кода блока цен внутри новой секции

Секции, которой мы добавили вёрстку из примера, зададим класс container. Это основной класс разметки, который задаёт максимальную ширину элементов на странице. Что-то похожее мы создавали раньше. Также мы можем добавить сюда и наш стандартный заголовок.


```

72     <section class="container">
73         <h1 class="main-title">Ценовая политика</h1>
74         <div class="delimiter"></div>
75         <p class="main-title-description">Перечень основных цен на планы</p>
76         <div class="row row-cols-1 row-cols-md-3 mb-3 text-center">
77             <div class="col">
78                 <div class="card mb-4 rounded-3 shadow-sm">
79                     <div class="card-header py-3">
80                         <h4 class="my-0 fw-normal">Free</h4>
81                     </div>
82                     <div class="card-body">
83                         <h1 class="card-title pricing-card-title">$0<small class="text-mu
84                         <ul class="list-unstyled mt-3 mb-4">
85                             <li>10 users included</li>
86                             <li>2 GB of storage</li>
87                             <li>Email support</li>

```

Рис. 171 – Фрагмент блока цен с заголовком секции

В результате вся вёрстка сохранила свой изначальный вид из примера.

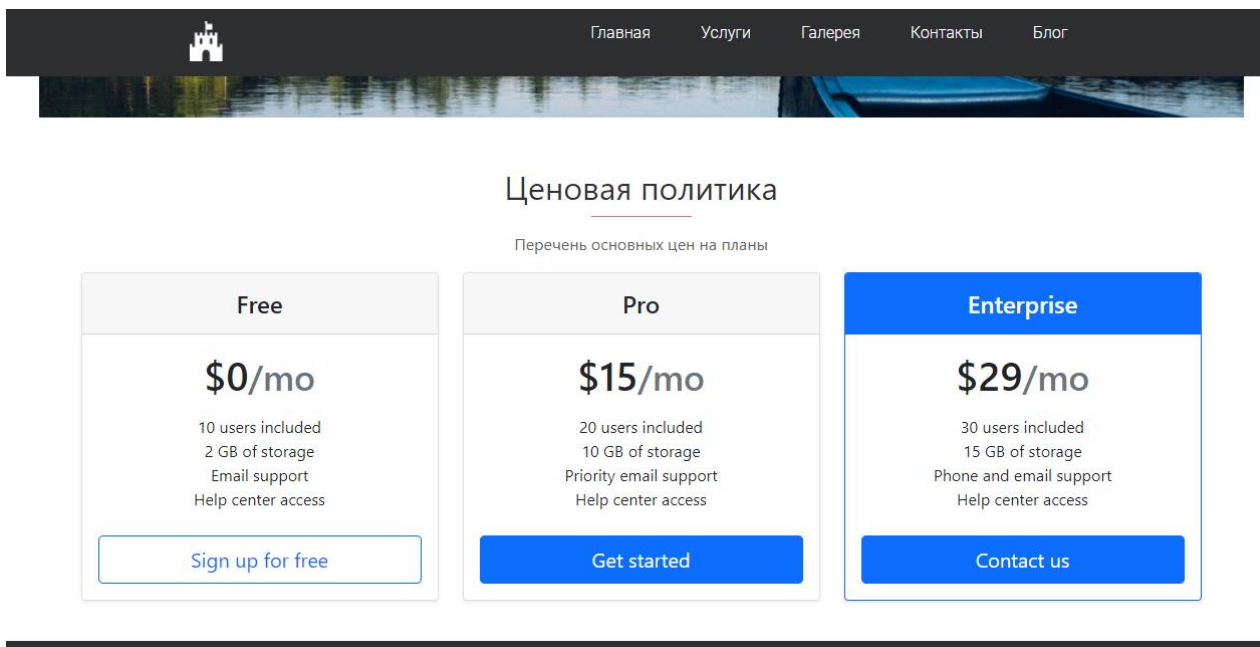


Рис. 172 – Внешний вид блока на странице

Кроме того, у этой страницы есть полная поддержка адаптивности. Взгляните на мобильную версию – здесь блоки «автоматически» выстроились в колонку. На самом деле никакой магии нет, просто эти стили уже прописаны заранее.

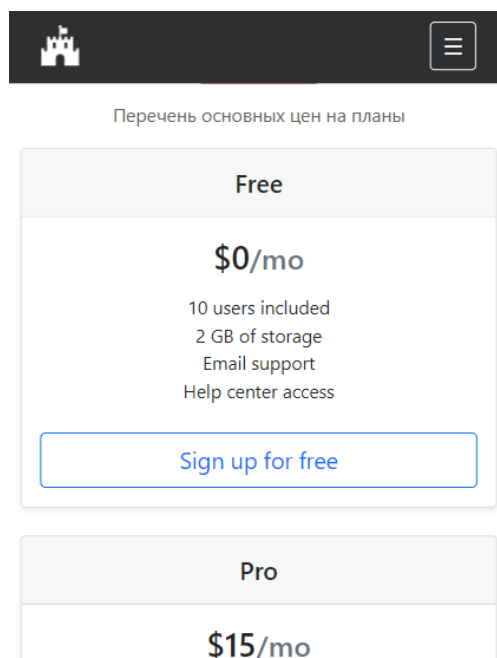


Рис. 173 – Внешний вид блока на мобильном разрешении

Но нами не было добавлено ни одной строки кода в CSS. В этом заключается сила разработки с использованием библиотек.

Закономерным вопросом будет как кастомизировать такие элементы под дизайн или требования заказчика? Часть стилей мы можем использовать всё из той же библиотеки и документации. Например: чтобы добавить больше отступы между заголовком и блоком с ценами можно добавить стиль **my-5** вместо **mb-3** к корневому блоку цен как на рисунке 174.

```

73     <h1 class="main-title">Ценовая политика</h1>
74     <div class="delimiter"></div>
75     <p class="main-title-description">Перечень основных цен на планы</p>
76     <div class="row row-cols-1 row-cols-md-3 my-5 text-center">

```

Рис. 174 – Изменение классов из блока-примера

Что это значит? Класс **mb-3** – устанавливает стиль `margin-bottom` равный **1rem**. Единица **rem** задаёт размер относительно размера шрифта элемента **<html>**.

Как правило, браузеры ставят этому элементу некоторый «разумный» (reasonable) размер по умолчанию, который в нашем случае равен **16px**. Таким образом класс **mb-3** устанавливает **margin-bottom: 16px**.

Как об этом можно узнать? Подсмотрев в консоли разработчика, какие стили применились к элементу, и какой класс их задал.

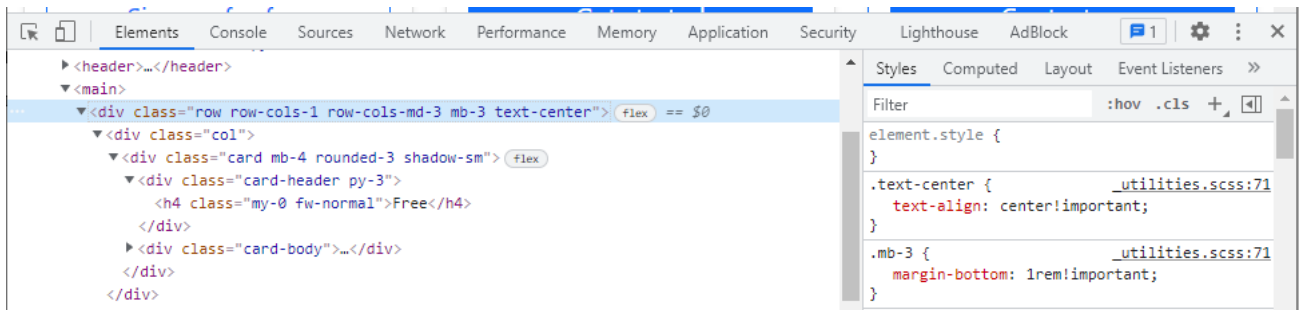


Рис. 175 – Поиск применённых стилей к HTML элементам

О других классах и за что они отвечают можно узнать из документации. Например, для внешних отступов (margin) и внутренних полей(padding) мы можем найти правила в разделе **Spacing**:



Рис. 176 – Пример использования шаблонных класс

Согласно документации для того, чтобы задать внешние поля сверху и снизу мы можем использовать классы **mt-*** и **mb-*** или же класс **my-***, как краткую версию, задающую сразу оба стиля. Таким образом установив **my-5** мы задали блоку отступы сверху и снизу равные **3rem = 48px**.

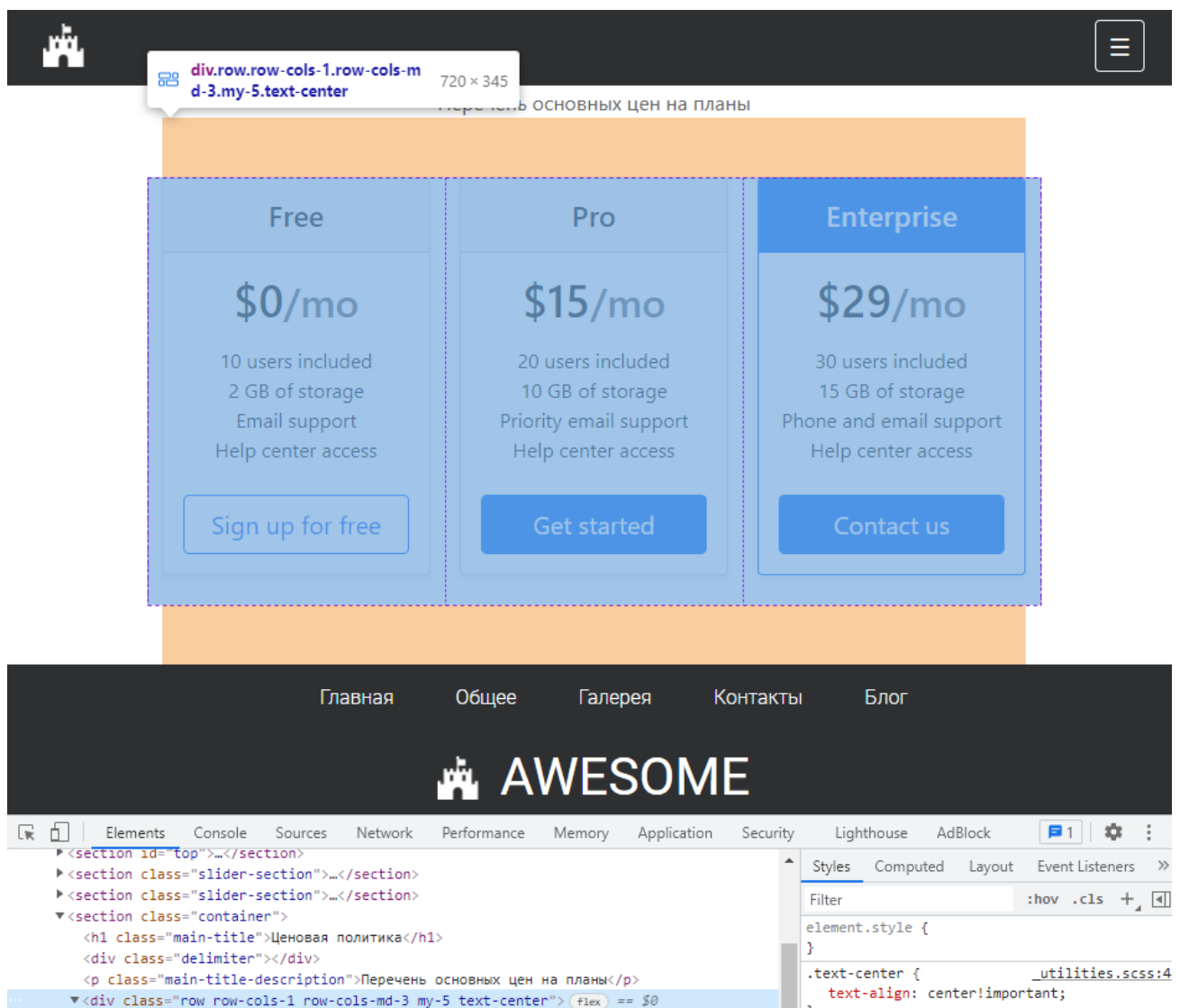


Рис. 177 – Типовой слайдер в документации

Но что делать, если этих классов недостаточно, если заказчику нужен другой размер, который не предусмотрен библиотекой?

Во-первых, самое важное, не изменять стили самой библиотеки. Они носят глобальный характер и будут менять все элементы, которые тот или иной класс используют. Кроме того, отступив от их правил, вы больше не сможете сверяться с документацией.

Во-вторых, для этого используется классический CSS. Вы создаёте рядом свои классы, которым задаёте те стили, которые нужны. А для того, чтобы они имели большее значение, чем значения библиотеки — используются правила каскадов CSS.

Задание на контроль

1. Повторить пример из практической части.
2. Создать работающий слайдер при помощи jQuery.
3. Подключить библиотеку Bootstrap 5 к проекту и добавить слайдер, используя возможности библиотеки.
4. Добавить блок цен из страницы примеров документации библиотеки Bootstrap.
5. Добавить описание слайдам и увеличить отступ блока цен инструментами библиотеки.
6. Лабораторная работа считается защищенной, если студент показал опубликованный в сети проект, а его вёрстка соответствует заданному шаблону. Необходимо продемонстрировать проект в git-репозитории и правильно ответить на все заданные контрольные вопросы.