

## Раздел сайта «Программа курса»

*Продолжение лабораторной работы  
ЛР\_Bootstrap\_3\_Сетка\_2x2.docx  
предыдущего занятия.*

На этом занятии мы будем верстать раздел сайта «ПРОГРАММА КУРСА». Здесь используется элемент Bootstrap «**аккордеон**».

«Аккордеон» позволяет скрывать и показывать различные элементы на сайте. Особенность его в том, что одновременно может быть открыт **только один элемент**, а все остальные тогда скрываются.

Мы посмотрим как сделать аккордеон на Bootstrap, как настроить и стилизовать его, как добавить новые элементы в «аккордеон».

Вид раздела «Программа курса» приведен на рис. 1.

ПРОГРАММА КУРСА		
Модуль 1 (HTML)	Модуль 2 (CSS)	Модуль 3 (Практика)
Введение	CSS селекторы	Верстка меню сайта
Установка браузера и редактора кода	Работа с изображениями	Домашнее задание
Обзор функций редактора	Границы	Верстка фрагмента сайта
Первые HTML теги	Размеры элементов	Домашнее задание
Как задавать путь к файлу	<i>Разберемся, как изменять размеры элементов на странице с помощью CSS.</i>	Верстка фрагмента сайта
Создание списков	Работа с шрифтом	Домашнее задание
Создание таблиц	Блочная модель документа	Photoshop для верстальщика
Работа с текстом	Псевдоклассы и псевдоэлементы	<i>В этом уроке мы рассмотрим графический редактор Photoshop и научимся основным приемам, которые могут пригодиться верстальщику.</i>
<i>В этом видео уроке мы узнаем, как работать с текстом.</i>	CSS Flexbox	Верстка фрагмента сайта
Создание формы	Адаптивная версия сайта	Домашнее задание
Универсальные атрибуты	Домашнее задание	Финальный проект-задание
Цвет RGB и спецсимволы		Бонусы

Рис. 1. Вид раздела сайта «Программа курса»

Видно, что в разделе *три столбца*. Такую верстку мы уже делать умеем с помощью сетки Bootstrap (см., например, раздел «О курсе»).

В каждом столбце использован элемент «аккордеон». Мы видим перечень пунктов. При разворачивании названия каждого пункта появляется более подробный текст. «Аккордеон» подразумевает, что если вы открываете какой-то элемент (пункт), то все остальные элементы скрываются, **нельзя открыть сразу несколько элементов.**

Мы сейчас сделаем три таких независимых «аккордеона» для каждого столбца.

Перейдем на страницу документации Bootstrap в раздел «Компоненты» по ссылке (см. рис. 2):

<https://getbootstrap.com/docs/5.0/components/accordion/>

Search docs...

Ctrl + /

> Getting started

> Customize

> Layout

> Content

> Forms

Components

Accordion

Alerts

Badge

Breadcrumb

Buttons

Button group

Card

Carousel

Close button

Collapse

Dropdowns

List group

Modal

Navs & tabs

Navbar

Offcanvas

Pagination

Popovers

Progress

Scrollspy

Spinners

Toasts

Tooltips

> Helpers

> Utilities

> Extend

> About

## Flush

Add `.accordion-flush` to remove the default `background-color`, some borders, and some rounded corners to render accordions edge-to-edge with their parent container.

Accordion Item #1

Accordion Item #2

Accordion Item #3

Copy

```
<div class="accordion accordion-flush" id="accordionFlushExample">
  <div class="accordion-item">
    <h2 class="accordion-header" id="flush-headingOne">
      <button class="accordion-button collapsed" type="button" data-bs-toggle="collapse" data-bs-target="#flush-collapseOne" aria-expanded="false" aria-controls="flush-collapseOne">Accordion Item #1</button>
    </h2>
    <div id="flush-collapseOne" class="accordion-collapse collapse" aria-labelledby="flush-headingOne" data-bs-parent="#accordionFlushExample">
      <div class="accordion-body">Placeholder content for this accordion, which is intended to demonstrate how collapsible content is styled with .flush.</div>
    </div>
  </div>
  <div class="accordion-item">
    <h2 class="accordion-header" id="flush-headingTwo">
      <button class="accordion-button collapsed" type="button" data-bs-toggle="collapse" data-bs-target="#flush-collapseTwo" aria-expanded="false" aria-controls="flush-collapseTwo">Accordion Item #2</button>
    </h2>
    <div id="flush-collapseTwo" class="accordion-collapse collapse" aria-labelledby="flush-headingTwo" data-bs-parent="#accordionFlushExample">
      <div class="accordion-body">Placeholder content for this accordion, which is intended to demonstrate how collapsible content is styled with .flush.</div>
    </div>
  </div>
  <div class="accordion-item">
    <h2 class="accordion-header" id="flush-headingThree">
      <button class="accordion-button collapsed" type="button" data-bs-toggle="collapse" data-bs-target="#flush-collapseThree" aria-expanded="false" aria-controls="flush-collapseThree">Accordion Item #3</button>
    </h2>
    <div id="flush-collapseThree" class="accordion-collapse collapse" aria-labelledby="flush-headingThree" data-bs-parent="#accordionFlushExample">
      <div class="accordion-body">Placeholder content for this accordion, which is intended to demonstrate how collapsible content is styled with .flush.</div>
    </div>
  </div>
</div>
```

Рис. 2. Фрагмент кода документации.

Здесь есть готовый фрагмент кода, который мы можем скопировать и использовать.

Перейдем в редактор кода.

Создадим в файле **index.html** нашего проекта следующую новую секцию с классом **shedule**. То есть создадим пока следующую структуру для заголовка этой секции:

```
<section class="shedule">

    <div class="container">
        <div class="row">
            <div class="col-12">
                <h2 class="text-center text-uppercase color2 mb-3">
                    Программа курса
                </h2>
            </div>
        </div>
    </div>

</section>
```

Заменим здесь **color2** на **color1**. Получаем в результате заголовок нашей секции (Рис. 3):



Рис. 3. Вид заголовка **<h2>**.

Теперь допишем в стилевом файле **style.css** для только что созданного класса **.shedule** для цвета фона :

```
.about, .shedule {
    background-color: #eee;
}
```

Цвет фона в проекте чередуется с **белого** на **серый** по секциям. Эта секция будет **серой** (Рис. 4).



Рис. 4. Вид секции с классом `.shedule` после указания цвета фона.

Теперь создаем в контейнере `.container` следующую строку `.row` из трех столбцов, то есть допишем код (см. код в красной рамке):

```
<!-- ПРОГРАММА КУРСА -->
<section class="shedule">
  <div class="container">
    <div class="row">
      <div class="col-12">
        <h2 class="text-center text-uppercase color1 mb-3">Программа
курса</h2>
      </div>
    </div>
    <div class="row">
      <div class="col-md-4 col-sm-12">
        <h4 class="text-center fw-bold mb-3 mt-3">Модуль 1 <br>(HTML)</h4>
      </div>
    </div>
  </div>
</section>
```

Здесь мы указали `.col-md-4` для *средних* экранов и `.col-sm-12` для *маленьких экранов*, то есть для маленьких экранов вместо трех столбцов будем иметь только один столбец.

Заголовок `<h4>` внутри столбца – это название модуля первого столбца: «Модуль 1 `<br>` (HTML)». Для форматирования заголовка использованы классы Bootstrap:

**text-center**,

**fw-bold** для того чтобы установить жирный текст,

**mb-3** – небольшой отступ *снизу* и

**mt-3** – отступ сверху для того чтобы этот элемент отделялся.

И, как вы видите, не написав ни одного css-свойства, просто указывая классы Bootstrap, мы придали тексту нужный нам внешний вид.

Получим такой результат (см. рис. 5):

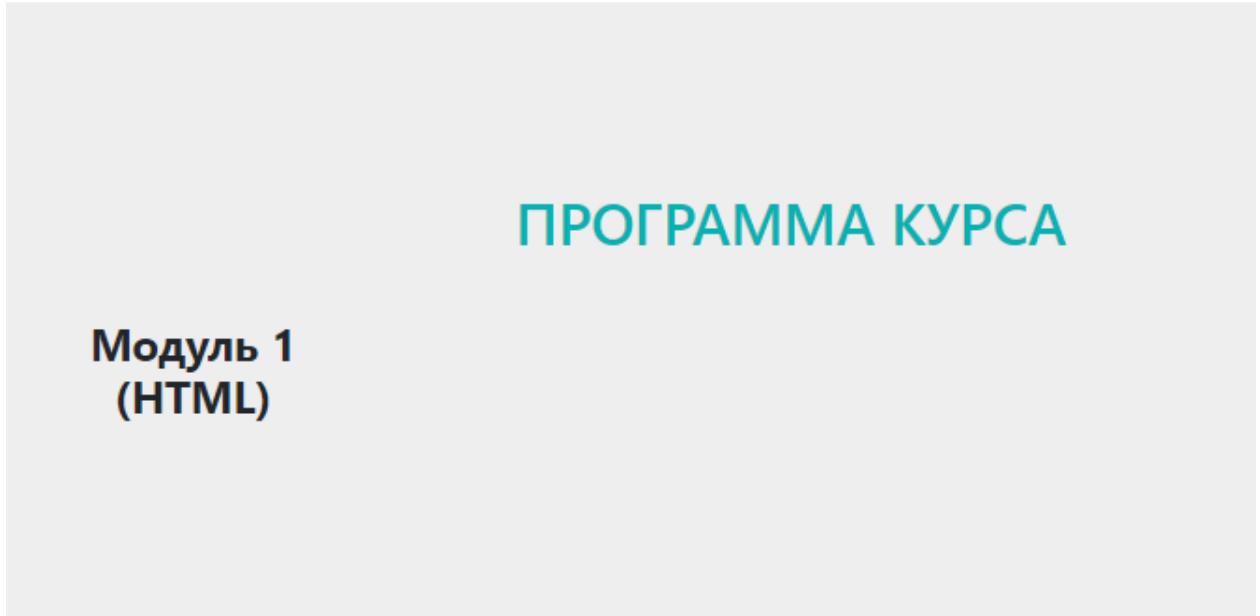


Рис. 5. Заголовок первого столбца.

Теперь начинаем создавать «аккордеон».

Копируем код «аккордеона» из документации Bootstrap, раздел Поток (**Flush**):

<https://getbootstrap.ru/docs/5.3/components/accordion/>

**ПОТОК**

Добавьте `.accordion-flush`, чтобы удалить некоторые границы и закругленные углы, чтобы аккордеоны отображались от края до края с их родительским контейнером.

Элемент аккордеона #1

Placeholder content for this accordion, which is intended to demonstrate the `.accordion-flush` class. This is the first item's accordion body.

Элемент аккордеона #2

Элемент аккордеона #3

HTML

```
<div class="accordion accordion-flush" id="accordionFlushExample">
  <div class="accordion-item">
    <h2 class="accordion-header">
      <button class="accordion-button collapsed" type="button" data-bs-toggle="collapse" data-bs-parent="#accordionFlushExample">Элемент аккордеона #1</button>
    </h2>
    <div id="flush-collapseOne" class="accordion-collapse collapse" data-bs-parent="#accordionFlushExample">
      <div class="accordion-body">Placeholder content for this accordion, which is intended to demonstrate the flush class. This is the first item's accordion body.</div>
    </div>
  </div>
  <div class="accordion-item">
    <h2 class="accordion-header">
      <button class="accordion-button collapsed" type="button" data-bs-toggle="collapse" data-bs-parent="#accordionFlushExample">Элемент аккордеона #2</button>
    </h2>
    <div id="flush-collapseTwo" class="accordion-collapse collapse" data-bs-parent="#accordionFlushExample">
      <div class="accordion-body">Placeholder content for this accordion, which is intended to demonstrate the flush class. This is the second item's accordion body. It is not supposed to overlap with the first accordion's body.</div>
    </div>
  </div>
  <div class="accordion-item">
    <h2 class="accordion-header">
      <button class="accordion-button collapsed" type="button" data-bs-toggle="collapse" data-bs-parent="#accordionFlushExample">Элемент аккордеона #3</button>
    </h2>
    <div id="flush-collapseThree" class="accordion-collapse collapse" data-bs-parent="#accordionFlushExample">
      <div class="accordion-body">Placeholder content for this accordion, which is intended to demonstrate the flush class. This is the third item's accordion body. It is not supposed to overlap with the first two accordions' bodies.</div>
    </div>
  </div>
</div>
```

Вставляем в наш код после тега `<h4>`. И смотрим что у нас получилось (см. рис. 6).

## ПРОГРАММА КУРСА

### Модуль 1 (HTML)

- Элемент аккордеона #1
- Элемент аккордеона #2
- Элемент аккордеона #3

Рис. 6. Вид секции после вставки кода «аккордеона» из документации.

Появился вот такой «аккордеон».

Внесем определенные изменения в наш код.

Переходим в CSS файл, создаем здесь новый комментарий:

```
/*Программа курса*/
```

Изменим цвета для активного элемента «аккордеона», допишем в стилевом файле такой селектор:

```
.accordion-button:not(.collapsed) {  
    color: var(--bs-white);  
    background-color: var(--color2);  
}
```

То есть установили для выбранного элемента *белый цвет текста* (`var(--bs-white)`) и *цвет фона – var(--color2)*.

Надо установить также белый цвет для **стрелки** в конце названия элемента:



В стилевом файле `style.css` добавим селектор (скопировали его из кода страницы):

```
.accordion-button:not(.collapsed)::after {  
    background-image:  
url("data:image/svg+xml,%3csvg xmlns='http://www.w3.org/2000/svg' viewBox='0 0 16  
16' fill='white'%3e%3cpath fill-rule='evenodd' d='M1.646 4.646a.5.5 0 0 1 .708 0L8  
10.29315.646-5.647a.5.5 0 0 1 .708.7081-6 6a.5.5 0 0 1-.708 0L6-6a.5.5 0 0 1  
0-.708z '%3e%3c/svg%3e")  
;  
}
```

```
.accordion-button:not(.collapsed)::after {  
    background-image: url("data:image/svg+xml,%3csvg  
xmlns='http://www.w3.org/2000/svg' viewBox='0 0 16 16' fill='white'%3cpath  
fill-rule='evenodd' d='M1.646 4.646a.5.5 0 0 1 .708 0L8 10.29315.646-5.647a.5.5 0  
0 1 .708.7081-6 6a.5.5 0 0 1-.708 0L6-6a.5.5 0 0 1 0-.708z '%3e%3c/svg%3e");  
}
```

Здесь цвет стрелки уже заменили на белый:

```
fill='white'
```

Пока в нашем «аккордеоне» три элемента. Мы можем отделить их друг от друга.

Проанализируем код одного из элементов `<div class="accordion-item">`:

```
<div class="accordion accordion-flush" id="accordion1">  
    <div class="accordion-item">  
        <h2 class="accordion-header">  
            <button class="accordion-button" type="button" data-bs-  
            toggle="collapse" data-bs-target="#collapse1" aria-expanded="false" aria-  
            controls="collapse1">  
                Введение  
            </button>  
        </h2>  
        <div id="collapse1" class="accordion-collapse collapse show"  
        data-bs-parent="#accordion1">  
            <div class="accordion-body fst-italic">Краткий обзор того, что мы  
будем изучать в данном видео курсе.</div>  
        </div>  
    </div>
```

У каждого «аккордеона» должен быть свой **id**. Например, **id="accordion1"**. Этот же **id** должен использоваться в значении атрибута **data-bs-parent="#accordion1"** дочерних элементов.

Следует также обратить внимание и на то, что параметр **aria-controls="collapse1"**, который связывает кнопку **<button class="accordion-button">** с соответствующим текстом, должен совпадать с **id="collapse1"** внешнего **<div>**, содержащего выпадающий текст элемента, а также с параметром **data-bs-target="#collapse1"** кнопки.

Это справедливо для каждого элемента «аккордеона».

Итак, у всех дочерних элементов данного «аккордеона» параметр **data-bs-parent="#accordion1"** должен совпадать с **id="accordion1"**.

А параметр **id="collapse1"** элемента должен совпадать с параметрами кнопки:

```
data-bs-target="#collapse1"  
aria-controls="collapse1"
```

Здесь имя **collapse1** – это название, придуманное пользователем.

Тогда для следующих дочерних элементов имеет смысл задавать значение **id**, равным **id="collapse2"**, **id="collapse3"** и т.д.

Чтобы сделать первый элемент «аккордеона» развернутым по умолчанию, выполним следующие преобразования:

```
<div class="accordion-item">  
    <h2 class="accordion-header">  
        <button class="accordion-button collapsed" type="button" data-bs-  
        toggle="collapse" data-bs-target="#collapse1" aria-expanded="false" aria-  
        controls="collapse1">  
            Название элемента  
        </button>  
    </h2>  
    <div id="collapse1" class="accordion-collapse collapse show" data-  
    bs-parent="#accordion1">  
        <div class="accordion-body fst-italic">Содержание выпадающего  
текста элемента</div>  
    </div>  
</div>
```

То есть удалить класс **collapsed** и добавить класс **show**.

Осталось добавить элементы в созданный «аккордеон».

Приводим содержание текстовых элементов **первого** столбца в таблице:

Название элемента	Выпадающий текст
Введение	<i>Краткий обзор того, что мы будем изучать в данном видео курсе.</i>
Установка браузера и редактора кода	<i>В этом уроке мы установим браузер Google Chrome и редактор кода VS Code.</i>
Обзор функций редактора	<i>Рассмотрим функции редактора VS Code и узнаем как с ним работать.</i>
Первые HTML теги	<i>Выучим свои первые HTML теги, с помощью которых и создаются любые сайты.</i>
Как задавать путь к файлу	<i>Научимся правильно указывать путь к файлам изображений и другим страницам сайта.</i>
Создание списка	<i>Научимся создавать списки на своем сайте.</i>
Создание таблиц	<i>В этом уроке мы рассмотрим, как создавать таблицы на языке HTML.</i>
Работа с текстом	<i>В этом видео уроке мы узнаем, как работать с текстом.</i>
Создание формы	<i>Научимся создавать формы для ввода текста, флагки и кнопки.</i>
Универсальные атрибуты	<i>Рассмотрим универсальные атрибуты, которые можно добавлять любому HTML тегу.</i>
Цвет RGB и спецсимволы	<i>Научимся работать с RGB цветом и добавлять специальные символы на свой сайт.</i>

Выпадающий текст оформлен *курсивом*. За это отвечает класс текста **fst-italic**.

Для создания **второго** столбца пишем код:

```
<div class="col-md-4 col-sm-12">
    <h4 class="text-center fw-bold mb-3 mt-3">Модуль 2 <br>(CSS)</h4>
    <div class="accordion accordion-flush" id="accordion2">

        Вставить сюда элементы аккордеона

    </div>

</div>
```

Элементы «аккордеона» создаем по аналогии с первым «аккордеоном». «Аккордеоны» в двух других столбцах должны быть независимы от первого, поэтому у каждого из них должен быть свой идентификатор, например, **`id="accordion2"`** и **`id="accordion3"`**. Внести изменения в соответствующих местах кода. Надо проследить также, чтобы идентификаторы всех дочерних элементов тоже были разными.

Содержание текстовых элементов **второго** столбца:

Название элемента	Выпадающий текст
CSS селекторы	<i>Рассмотрим основные CSS селекторы, узнаем что это такое и зачем они нужны.</i>
Работа с изображениями	<i>Научимся работать с изображениями на языке CSS.</i>
Границы	<i>В этом уроке мы научимся создавать границы вокруг любого элемента на сайте и стилизовать их.</i>
Размеры элементов	<i>Разберемся, как изменять размеры элементов на странице с помощью CSS.</i>
Работа с шрифтом	<i>Научимся работать со шрифтами. Установим свой шрифт на сайт.</i>
Блочная модель документа	<i>В этом уроке мы поговорим про блочную модель документа и научимся пользоваться блочной моделью.</i>
Псевдоклассы и псевдоэлементы	<i>Псевдоклассы и псевдоэлементы на языке CSS позволяют расширять основные возможности языка, которые мы уже выучили.</i>
CSS Flexbox	<i>В этом уроке мы изучим базовые приемы Flexbox верстки и будем применять их в следующем модуле.</i>
Адаптивная версия сайта	<i>В этом уроке мы будем учиться создавать адаптивную сетку на CSS. Так же научимся работать с переменными и функцией calc. Узнаем, как создавать медиа-запросы, чтобы сделать адаптивную версию своего сайта для мобильных устройств. Научимся создавать макет с 12-колончатой сеткой в Photoshop, сравним нашу сетку с сеткой Bootstrap 5, а так же научимся её подключать и узнаем как проверять свой сайт на эмуляторе мобильных устройств прямо с браузера на ПК.</i>
Домашнее задание	<i>Задание для практики по теме 'создание адаптивной сетки на CSS'.</i>

## Содержание текстовых элементов **третьего** столбца.

Название элемента	Выпадающий текст
Верстка меню сайта	Урок, в котором мы будем верстать фрагмент современного сайта, используя все полученные ранее знания.
Домашнее задание	Домашнее задание для закрепления изученного материала в предыдущем уроке.
Верстке фрагмента сайта	Урок, в котором мы будем верстать фрагмент современного сайта, используя все полученные ранее знания.
Домашнее задание	Домашнее задание для закрепления изученного материала в предыдущем уроке.
Верстке фрагмента сайта	Урок, в котором мы будем верстать фрагмент современного сайта, используя все полученные ранее знания.
Домашнее задание	Домашнее задание для закрепления изученного материала в предыдущем уроке.
Photoshop для верстальщика	В этом уроке мы рассмотрим графический редактор Photoshop и научимся основным приемам, которые могут пригодиться верстальщику.
Верстке фрагмента сайта	Урок, в котором мы будем верстать фрагмент современного сайта, используя все полученные ранее знания.
Домашнее задание	Домашнее задание для закрепления изученного материала в предыдущем уроке.
Финальный проект-задание	Подведение итогов и финальный проект, который вы сможете выполнить самостоятельно и добавить эту работу себе в портфолио.
Бонусы	Вы получите все исходные коды к лекциям и домашним заданиям этого модуля, дополнительные материалами, которые будут вам полезны при разработке и ссылки на большое количество проверенных онлайн ресурсов, которые помогут вам при создании сайта.

Окончательный вид раздела «Программа курса» сайта приведен ниже на (см. рис. 7):

ПРОГРАММА КУРСА		
Модуль 1 (HTML)	Модуль 2 (CSS)	Модуль 3 (Практика)
<b>Введение</b>	<b>CSS-селекторы</b>	<b>Верстка меню сайта</b>
Краткий обзор того, что мы будем изучать в данном видео курсе.	Рассмотрим основные CSS селекторы, узнаем что это такое и зачем они нужны.	Урок, в котором мы будем верстать фрагмент современного сайта, используя все полученные ранее знания.
Установка браузера и редактора кода.	Работа с изображениями.	Домашнее задание.
Обзор функций редактора.	Границы	Верстка фрагмента сайта.
Первые HTML-теги.	Размеры элементов	Верстка фрагмента сайта.
Как задавать путь к файлу.	Работа со шрифтами	Верстка фрагмента сайта.
Создание списка.	Блочная модель документа.	Домашнее задание.
Создание таблиц.	Псевдоклассы и псевдоэлементы.	Photoshop для верстальщика.
Работа с текстом.	CSS FlexBox	Верстка фрагмента сайта.
Создание формы.	Адаптивная версия сайта.	Домашнее задание.
Универсальные атрибуты атрибуты.	Домашнее задание.	Финальный проект-задание.
Цвет RGB и спецсимволы.		Бонусы.

Рис. 7. Окончательный вид раздела «Программа курса» сайта

## Документация Bootstrap 5 по Аккордеону

На русском языке

<https://getbootstrap.ru/docs/5.3/components/accordion/>