

# Инструкция для разработчиков по верстке проекта

**Оглавление**

[**Инструкция для разработчиков по верстке проекта** 1](#_Toc117077862)

[Кнопки 2](#_Toc117077863)

[Синяя кнопка 2](#_Toc117077864)

[Неактивная синяя кнопка 3](#_Toc117077865)

[Кнопка с svg иконкой внутри 3](#_Toc117077866)

[Формы 4](#_Toc117077867)

[Элементы форм 4](#_Toc117077868)

[Стандартное поле ввода (text, tel, email и подобные) 4](#_Toc117077869)

[Выпадающий список (стилизованный select) 4](#_Toc117077870)

[Поля внесения даты 5](#_Toc117077871)

[Поле ввода пароля 6](#_Toc117077872)

[Стилизованный чекбокс 7](#_Toc117077873)

[Блок колонок 8](#_Toc117077874)

# Вступление

В данной инструкции представлены базовые элементы вертски Sechnov ЛК, которые часто встречаются разных страницах. Сделано это, что бы вам было удобнее работать с версткой и при необходимости вы могли переиспользовать существующие элементы без правки стилей. Если же все таки понадобится привить стили, то убедительная просьба править именно в файла .scss, а не в .css, что бы не допустить захламления стилей и невозможности, со временем, их поддержки.

Товарищи! Поддержите стилистическую гигиену! Используй sass! Для этого лучше всего сделать форк проекта с гитхаба. Далее склонировать себе на машину. Далее, если у вас уже стоит Gulp, то командой npm install установить плагины используемые в данной сборке gulp. Потом запускаете gulp правите. В папке scss стили разнесены по разным файлам для удобства. Назначения файлов можно понять из названия. Дополню, что в файле elements.scss собраны стили элементов, у которых нет своих собственных медиа-запросов. Если же у элемента есть собственные меди-запросы, то в папке \elements у этого элемента будет собственный файл стилей.

Что касается JavaScript. Если в ходе интеграции верстки js код будет разбираться по разным шаблонам компонентов битрикса, то форки можно не делать, а вносить в будущие изменения в файлах script.js шаблона компонента. Но если же все стили будут хранится в том порядке, который существует в верстке сейчас, то скрипты лучше править и писать так же при помощи gulp (делаем фоки и пулреквесты).

Так же по внесению изменений в стили и скрипты можно писать мне, а измененные данные через время забирать из репозитория этого проекта на гитхабе.

У меня все. Успехов в труде.

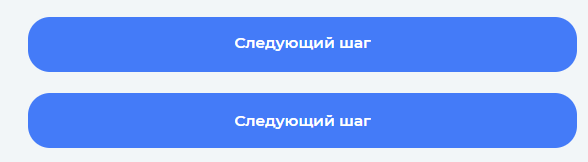
# Кнопки

Для кнопки применяется базовый класс .btn. Но к нему обязательно нужно применят модификатор, что бы у кнопки был фон (по умолчанию она прозрачная). Класс кнопки можно указывать как тегу <button>, так и тегу <a>

## Синяя кнопка

<a href="#" class="btn btn--blue ">Кнопка</a>

<button  class="btn btn--blue" >Кнопка</button>

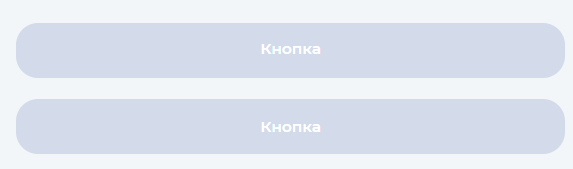


## Неактивная синяя кнопка

Для тега <button> устанавливается атрибут disabled="disabled", а для тега <a> устанавливается класс-модификатор btn—disabled

<a href="#" class="btn btn--blue disabled">Кнопка</a>

<button  class="btn btn--blue" disabled="disabled">Кнопка</button>



## Кнопка с svg иконкой внутри

Для того, что бы svg иконка нормально отображалась в кнопке нужно использовать класс-модификатор btn--with-elem

Пример:

<a href="#" class="btn btn--blue btn--with-elem " title="Выход">

    <span>sobzina@gmail.com</span>

    <svg width="20" height="20" viewBox="0 0 20 20" fill="none" xmlns="http://www.w3.org/2000/svg">

        <path d="M8.59375 13.2812L11.875 10L8.59375 6.71875" stroke="white" stroke-width="2" stroke-linecap="round" stroke-linejoin="round"/>

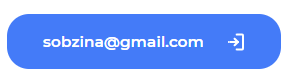
        <path d="M3.12268 10L11.8727 10" stroke="white" stroke-width="2" stroke-linecap="round" stroke-linejoin="round"/>

        <path d="M10.625 2.5L16.25 2.5C16.4158 2.5 16.5747 2.56585 16.6919 2.68306C16.8092 2.80027 16.875 2.95924 16.875 3.125L16.875 16.875C16.875 17.0408 16.8092 17.1997 16.6919 17.3169C16.5747 17.4342 16.4158 17.5 16.25 17.5L10.625 17.5" stroke="white" stroke-width="2" stroke-linecap="round" stroke-linejoin="round"/>

    </svg>

</a>

Выглядит вот так



**Обратить внимание!!!** Текст в кнопке обернут в тег span.

# Формы

## Элементы форм

### Стандартное поле ввода (text, tel, email и подобные)

Пример кода:

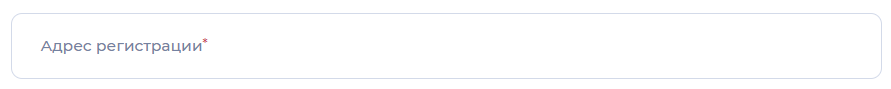
<label class="form-elem">

    <input type="text" class="form-elem\_\_input form-elem\_\_input--validation" name="address-registration" placeholder="Адрес регистрации" autocomplete="off">

    <span>Адрес регистрации<sup>\*</sup></span>

</label>

Внешний вид:



Если поле не является обязательным, то убрать тег <sup>\*</sup>. Если поле используется в большой форме с табом, то убрать еще и класс form-elem\_\_input--validation, т.к. он отвечает за проверку полей скриптом на заполненность.

### Выпадающий список (стилизованный select)

Пример кода:

<label class="form-elem">

    <select class="form\_\_category-select form\_\_category-select--no-search

form-elem\_\_input--validation" name="contact-face-relative" id="cazwt127" required>

        <option value="">This is a placeholder</option>

        <option value="f">Отец</option>

        <option value="m">Мать</option>

        <option value="gf">Дедушка</option>

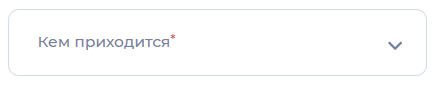
        <option value="gm">Бабушка</option>

    </select>

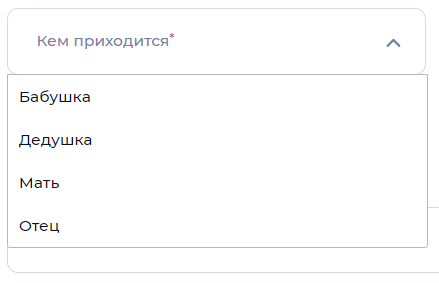
    <span>Кем приходится<sup>\*</sup></span>

</label>

Пример внешнего вида:



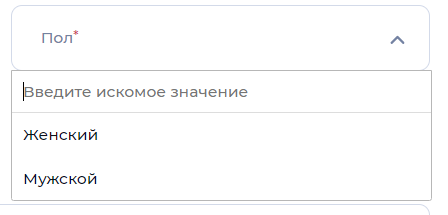
и открытый:



Если select не является обязательным, то убрать тег <sup>\*</sup>. Если поле используется в большой форме с табом, то убрать еще и класс form-elem\_\_input--validation, т.к. он отвечает за проверку полей скриптом на заполненность.

Атрибут id на данный момент ни на что не влияет. Я его поставил на всякий случай. Если не пригодится, можно смело убирать.

**ВНИМАНИЕ!!!** Если выпадающему списку нужно поле поиска, то удалить у тега <select> класс form\_\_category-select--no-search и будет вот так:





### Поля внесения даты

Пример кода:

<label class="form-elem form-elem--date">

    <input type="text" name="date-of-birth" class="form-elem\_\_input flatpickr flatpickr-input form-elem\_\_input--validation" placeholder="Дата публикации" data-id="datetime">

    <span>Дата рождения<sup>\*</sup></span>

    <svg width="20" height="20" viewBox="0 0 20 20" fill="none" xmlns="http://www.w3.org/2000/svg">

        <path d="M16.25 3.125H3.75C3.40482 3.125 3.125 3.40482 3.125 3.75V16.25C3.125 16.5952 3.40482 16.875 3.75 16.875H16.25C16.5952 16.875 16.875 16.5952 16.875 16.25V3.75C16.875 3.40482 16.5952 3.125 16.25 3.125Z" stroke="#7482A2" stroke-width="1.5" stroke-linecap="round" stroke-linejoin="round"/>

        <path d="M13.75 1.875V4.375" stroke="#7482A2" stroke-width="1.5" stroke-linecap="round" stroke-linejoin="round"/>

        <path d="M6.25 1.875V4.375" stroke="#7482A2" stroke-width="1.5" stroke-linecap="round" stroke-linejoin="round"/>

        <path d="M3.125 6.875H16.875" stroke="#7482A2" stroke-width="1.5" stroke-linecap="round" stroke-linejoin="round"/>

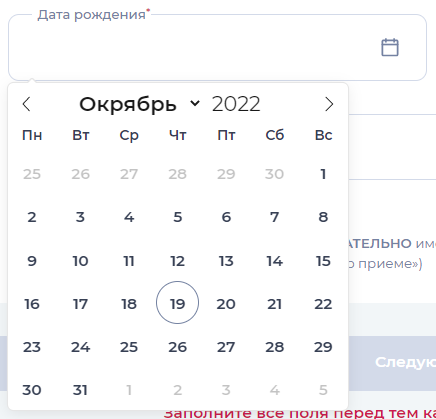
    </svg>

</label>

Пример внешнего вида:



и открытый:



Если поле не является обязательным, то убрать тег <sup>\*</sup>. Если поле используется в большой форме с табом, то убрать еще и класс form-elem\_\_input--validation, т.к. он отвечает за проверку полей скриптом на заполненность.

### Поле ввода пароля

Пример кода:

<label class="form-elem form-elem--password">

    <input type="password" class="form-elem\_\_input" name="password" placeholder="Пароль">

    <span>Пароль<sup>\*</sup></span>

    <svg width="21" height="21" viewBox="0 0 21 21" fill="none" xmlns="http://www.w3.org/2000/svg">

        <path class="eye-open" d="M10.5 4.59314C3.9375 4.59314 1.3125 10.5 1.3125 10.5C1.3125 10.5 3.9375 16.4056 10.5 16.4056C17.0625 16.4056 19.6875 10.5 19.6875 10.5C19.6875 10.5 17.0625 4.59314 10.5 4.59314Z" stroke="#7482A2" stroke-width="1.5" stroke-linecap="round" stroke-linejoin="round"/>

        <path class="eye-open" d="M10.5 13.7812C12.3122 13.7812 13.7812 12.3122 13.7812 10.5C13.7812 8.68782 12.3122 7.21875 10.5 7.21875C8.68782 7.21875 7.21875 8.68782 7.21875 10.5C7.21875 12.3122 8.68782 13.7812 10.5 13.7812Z" stroke="#7482A2" stroke-width="1.5" stroke-linecap="round" stroke-linejoin="round"/>

        <path class="eye-close" d="M16.5006 10.443L18.3717 13.6839" stroke="#7482A2" stroke-width="1.5" stroke-linecap="round" stroke-linejoin="round"/>

        <path class="eye-close" d="M12.6477 12.2443L13.2313 15.5537" stroke="#7482A2" stroke-width="1.5" stroke-linecap="round" stroke-linejoin="round"/>

        <path class="eye-close" d="M8.345 12.2426L7.76135 15.5526" stroke="#7482A2" stroke-width="1.5" stroke-linecap="round" stroke-linejoin="round"/>

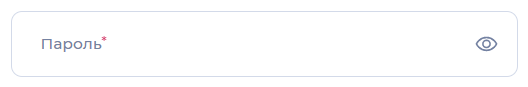
        <path class="eye-close" d="M4.49598 10.4403L2.61584 13.6968" stroke="#7482A2" stroke-width="1.5" stroke-linecap="round" stroke-linejoin="round"/>

        <path class="eye-close" d="M2.625 8.60303C4.00415 10.3101 6.53248 12.4687 10.5 12.4687C14.4676 12.4687 16.9959 10.3102 18.3751 8.60305" stroke="#7482A2" stroke-width="1.5" stroke-linecap="round" stroke-linejoin="round"/>

    </svg>

</label>

Пример внешнего вида:







Если поле не является обязательным, то убрать тег <sup>\*</sup>.

### Стилизованный чекбокс не выделяющийся при клике по подписи

Пример кода:

<div class="form-elem-checkbox clearfix">

    <label class="form-elem-checkbox\_\_container fleft">

        <input type="checkbox" name="approval">

        <svg width="10" height="10" viewBox="0 0 10 10" fill="none" xmlns="http://www.w3.org/2000/svg">

            <path d="M8.4375 2.81262L4.0625 7.18743L1.875 5.00012" stroke="#1BCCAC" stroke-width="2" stroke-linecap="round" stroke-linejoin="round"/>

        </svg>

    </label>

    <span class="form-elem-checkbox\_\_caption db rL hid">

        Запомнить меня на этом компьютере

    </span>

</div>

Пример внешнего вида:

### Стилизованный чекбокс выделяющийся при клике по подписи

Пример кода:

<label class="form-elem-checkbox-full">

    <input type="checkbox" name="name234">

    <svg width="24" height="24" viewBox="0 0 24 24" fill="none" xmlns="http://www.w3.org/2000/svg">

        <rect width="24" height="24" rx="6" fill="#F2F6F8"/>

        <path d="M16.125 9.37524L10.875 14.625L8.25 12.0002" stroke="#1BCCAC" stroke-width="2" stroke-linecap="round" stroke-linejoin="round"/>

    </svg>

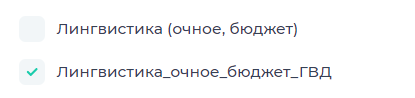
    <span class="form-elem-checkbox-full\_\_caption">

        Лингвистика (очное, бюджет)

    </span>

</label>

Пример внешнего вида:



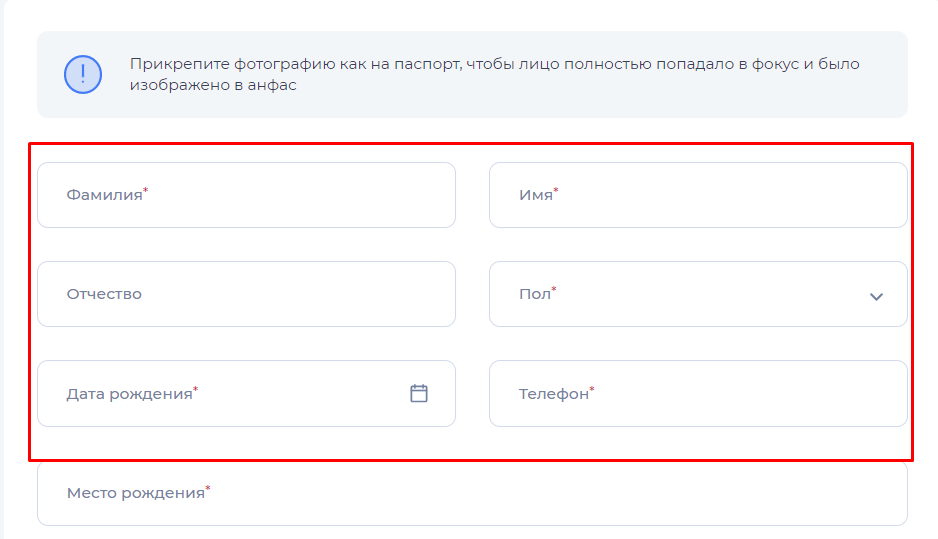
## Блок колонок

Блок колонок, исходя из названия, служит для развивания внутреннего содержимого на колонки. Количество колонок определяет класс-модификатор. Элементы добавляются внутрь как угодно. Ни каких оберток не нужно.

Пример блока колонок с двумя колонками:

<div class="columns-box columns-box--two-col"></div>

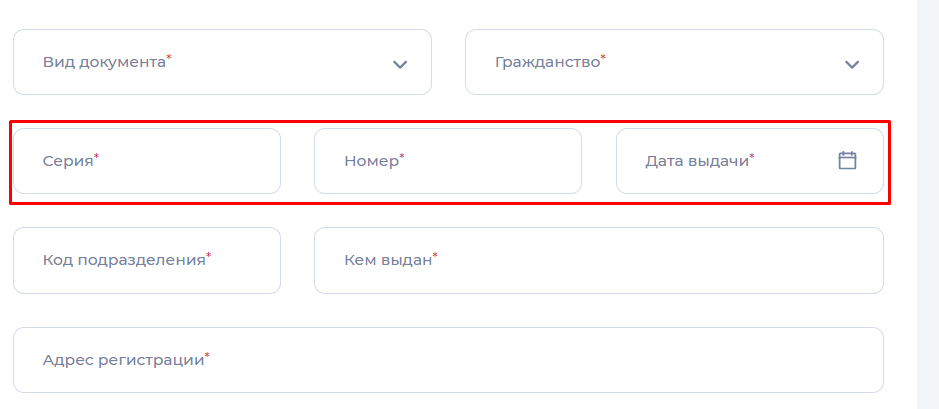
Пример внешнего вида



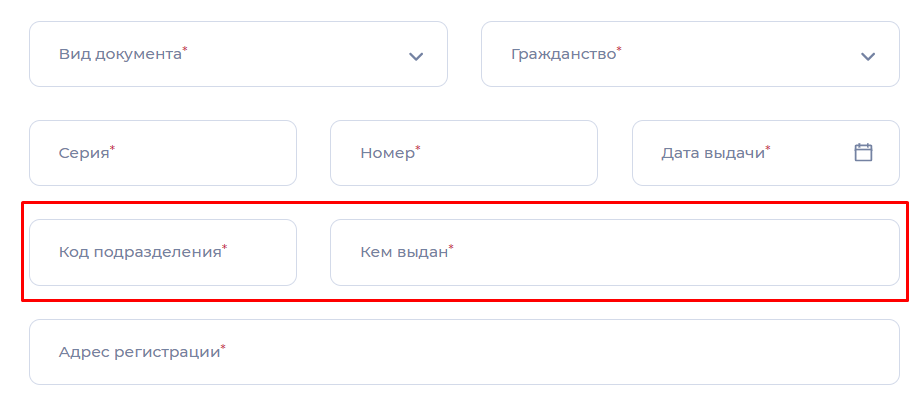
Пример блока колонок с тремя колонками:

<div class="columns-box columns-box--trhee-col"></div>

Пример внешнего вида:

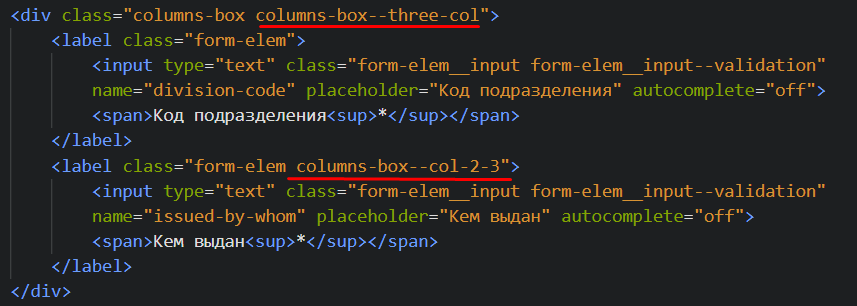


Так же можно в трехколоночный блок разместить два элемента и сделать так, чтобы один из элементов тянулся на две колонки. Выглядит вот так:



Делается очень просто. Если хотим, что бы тянулся левый элемент, то добавляем ему класс **columns-box--col-1-2.** Аналогично и для правого, только класс вот такой: **columns-box--col-2-3**

Пример:



# Модальные окна

Код пустого модального окна:

<div class="modal modal-text-2">

    <button class="cross-btn close-modal"></button>

        <h2 class="h5">Какие бывают стили текста, примеры</h2>

        <!-- тут размещять свой контент -->

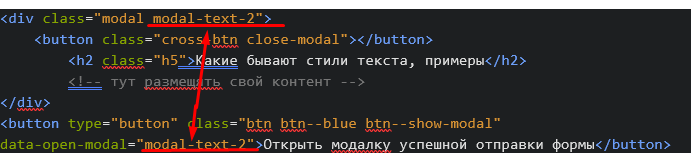
</div>

Код кнопки или ссылки запускающей модалку

<button type="button" class="btn btn--blue btn--show-modal"

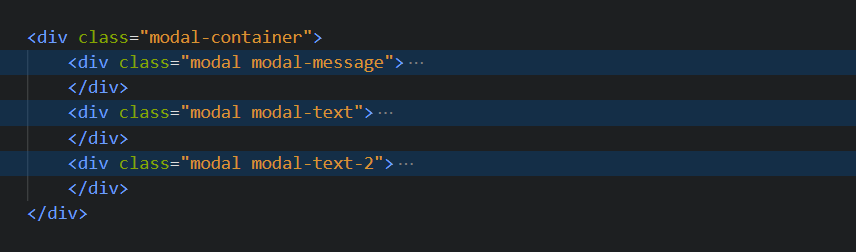
data-open-modal="modal-text-2">Открыть модалку успешной отправки формы</button>

Связка между кнопкой и модалкой происходит по следующей схеме – у кнопки имеется дата-атрибут data-open-modal в котором прописывается класс модалки, которую нужно открыть.

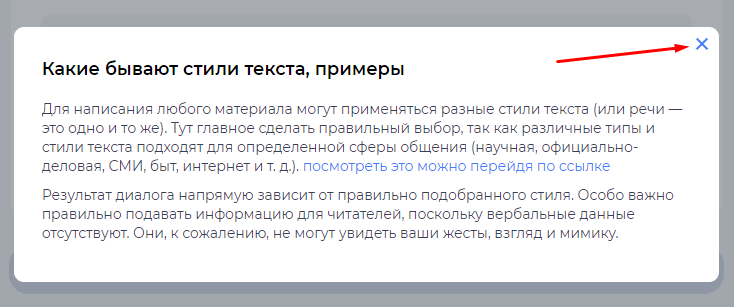


Все модалки размещаются в специальном контейнере имеющим класс modal-container

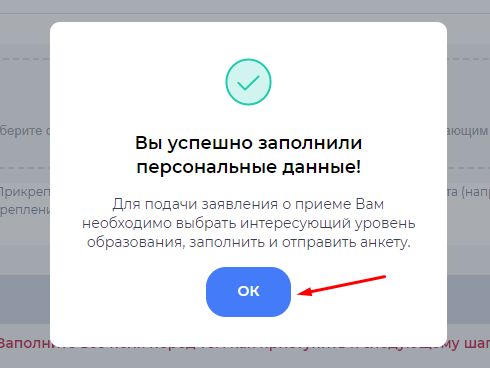
Пример:



Закрытие модалки производится по клику по «серой зоне» (все что вне модалки) или по кнопке имеющей класс close-modal. По умолчанию это крестик:



Так же вышеупомянутый класс может быть присвоен другому элементу, который будет закрывать модалку. Пример:



В данном примере модалка закрывается по клику на кнопку OK

В JS модалка открывается функцией showModal(“класс или id модалки”).

Если модалка открывается по кнопке, как было описано выше, то открытие производится только по классу.