

# CSS - bootstrap

Фреймворки и библиотеки

# Фреймворк

Фреймворк - это набор инструментов, библиотек и шаблонов, предназначенных для упрощения и ускорения разработки программного обеспечения. Он представляет собой структуру, которая определяет основные принципы и правила для создания приложений определенного типа или для выполнения конкретной задачи.

# Аналогия из жизни

Хорошая аналогия к фреймворкам в программировании может быть сравнение с набором инструментов или оборудования для строительства дома.

Представьте, что вы строите дом. Вам нужны инструменты, материалы и план, чтобы начать работу. Фреймворт в программировании подобен этому набору инструментов и плану. Он предоставляет вам все необходимое для создания вашего программного "дома", включая основные компоненты, структуру и инструкции по сборке.

Так же, как строительный набор может содержать стандартные двери, окна, трубы и электропроводку, фреймворт может предоставлять стандартные компоненты, такие как кнопки, формы, таблицы и многое другое.

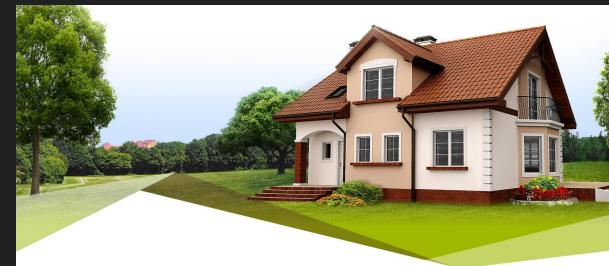
Как и в случае с набором для строительства, фреймворт в программировании помогает ускорить процесс разработки, обеспечивает единообразие и стандартизацию, а также предоставляет готовые решения для ряда распространенных задач.

Однако, подобно тому как набор инструментов для строительства может быть неидеальным для всех типов домов или индивидуальных предпочтений, фреймворт может иметь свои ограничения и недостатки в зависимости от конкретных потребностей проекта и предпочтений разработчика.

# Долго но зато сам (неповторимый)



# Быстро но по шаблону (нет индивидуальности)



# Плюсы фреймворков (и библиотек)

- Ускорение разработки: Фреймворки предоставляют готовые компоненты, шаблоны и инструменты, что позволяет разработчикам сосредоточиться на реализации бизнес-логики приложения, а не тратить время на написание базового кода с нуля.
- Единообразие и консистентность: Фреймворки определяют стандартные шаблоны и соглашения, что обеспечивает единообразие структуры и стиля кодирования в рамках проекта, а также упрощает совместную работу разработчиков.
- Готовая функциональность: Фреймворки предоставляют готовые решения для ряда распространенных задач, таких как работа с базами данных, обработка HTTP-запросов, аутентификация пользователей и многое другое, что упрощает разработку сложных приложений.
- Обновления и поддержка: Хорошие фреймворки обеспечивают активную поддержку и регулярные обновления, что помогает обеспечить безопасность, стабильность и совместимость с новыми технологиями и требованиями.
- Сообщество и экосистема: Фреймворки обычно имеют большие сообщества разработчиков, что обеспечивает доступ к дополнительным ресурсам, библиотекам, расширениям и решениям для решения различных задач.

# Минусы фреймворков (и библиотек)

- Избыточность: Некоторые фреймворки могут включать в себя больше функциональности, чем нужно для конкретного проекта, что может привести к избыточному объему кода и замедлению загрузки приложения.
- Ограничения в гибкости: Использование фреймворков может привести к ограничениям в гибкости и настраиваемости приложения, особенно если требуется реализация нестандартной функциональности или дизайна.
- Зависимость от фреймворка: Разработка приложения на базе фреймворка может создать зависимость от его архитектуры и API, что может затруднить переход на другой фреймворк или сделать приложение менее переносимым.
- Обучение и входной порог: Некоторые фреймворки могут иметь крутой кривой обучения и требовать времени для освоения их концепций и API, что может быть проблематично для новых разработчиков или при внедрении фреймворка в уже существующий проект.
- Производительность: Некоторые фреймворки могут иметь дополнительные накладные расходы в виде потребления ресурсов и увеличения времени загрузки страницы, особенно если они не используются эффективно или если проект не требует всей предоставляемой функциональности.

# Bootstrap

Bootstrap - это фреймворк для разработки веб-интерфейсов, то есть это библиотека CSS и JavaScript, разработанная для создания адаптивных и стильных веб-интерфейсов. Она предоставляет готовые компоненты, сетки и стили, которые позволяют быстро и легко создавать профессионально выглядящие веб-страницы.

Bootstrap был создан в Twitter в 2011 году Марком Отто и Джейкобом Торнтоном как внутренний инструмент для упрощения разработки интерфейсов и обеспечения их единообразия на всех платформах. Они обнаружили, что в процессе разработки они часто переиспользовали один и тот же CSS и JavaScript код, и решили выделить его в отдельный набор инструментов, который позже стал известен как Bootstrap.

# Плюсы bootstrap

- Быстрая разработка: Bootstrap предоставляет готовые компоненты и стили, что позволяет разработчикам быстро создавать веб-страницы без необходимости писать код с нуля.
- Адаптивность: Bootstrap поставляется с адаптивной сеткой, что обеспечивает правильное отображение веб-страниц на различных устройствах и экранах.
- Кросс-браузерность: Bootstrap учитывает различия между браузерами и автоматически применяет соответствующие стили, обеспечивая одинаковый внешний вид на разных платформах.
- Расширяемость: Благодаря модульной структуре, Bootstrap позволяет разработчикам легко настраивать и расширять его функциональность.
- Большое сообщество: У Bootstrap огромное сообщество пользователей, что обеспечивает поддержку, обмен опытом и доступ к большому количеству дополнительных ресурсов.

# Минусы bootstrap

- Общий внешний вид: Использование стандартных компонентов Bootstrap может привести к тому, что ваши веб-страницы будут выглядеть слишком стандартизированно и похоже на другие сайты, использующие Bootstrap.
- Избыточность: В проектах, где требуется только часть функциональности Bootstrap, могут быть лишние CSS и JavaScript файлы, что замедляет загрузку страницы.
- Сложность настройки: Настройка Bootstrap для своих нужд может потребовать времени и экспертизы в CSS и JavaScript.

# Bootstrap решает множество проблем

- Унификация интерфейса: Позволяет создавать единообразные интерфейсы для различных проектов.
- Адаптивность: Обеспечивает правильное отображение сайтов на различных устройствах.
- Ускорение разработки: Сокращает время на создание основного дизайна и функциональности.
- Кросс-браузерность: Обеспечивает совместимость с различными браузерами.
- Экономия ресурсов: Позволяет разработчикам сосредоточиться на реализации бизнес-логики, а не на создании базовых компонентов интерфейса.

Bootstrap использовался во многих крупных проектах и компаниях, включая Twitter, Airbnb, Spotify, LinkedIn и многие другие. Он стал одним из наиболее популярных инструментов для фронтенд-разработки благодаря своей простоте использования, гибкости и масштабируемости.

# Основные принципы Bootstrap

- Адаптивность: Bootstrap предлагает адаптивную сетку, которая автоматически реагирует на изменения размера экрана, обеспечивая правильное отображение веб-страниц на различных устройствах и разрешениях.
- Сетка (Grid): Сетка Bootstrap состоит из контейнеров (container), строк (row) и колонок (column). Она позволяет разработчикам создавать гибкие и масштабируемые макеты, распределяя содержимое по разным столбцам и строки в зависимости от нужд дизайна.
- Готовые компоненты: Bootstrap предоставляет готовые компоненты интерфейса, такие как кнопки, формы, навигационные панели, модальные окна, карусели и многое другое. Эти компоненты позволяют быстро и легко добавлять функциональность на веб-страницу без необходимости писать код с нуля.
- Типография и стилизация: Bootstrap предоставляет набор стилей для стандартных HTML-элементов, таких как заголовки, параграфы, списки и т. д. Это помогает обеспечить единообразный внешний вид текста на веб-странице.
- Мобильное приоритетное разработка: Bootstrap разработан с учетом принципа "mobile-first", что означает, что дизайн и разработка ведутся сначала для мобильных устройств, а затем адаптируются для больших экранов. Это обеспечивает оптимальный пользовательский опыт на всех устройствах.
- Кастомизация и расширяемость: Bootstrap предоставляет возможности для кастомизации стилей и компонентов с помощью переменных, тем и настройки. Разработчики могут легко настраивать внешний вид и поведение компонентов Bootstrap под конкретные потребности проекта.
- Поддержка браузеров: Bootstrap обеспечивает кросс-браузерную совместимость, что означает, что веб-страницы, созданные с использованием Bootstrap, должны корректно отображаться во всех основных браузерах, включая Chrome, Firefox, Safari, Edge и другие.

# Установка Bootstrap

- Через Content Delivery Network (CDN)
- Через загрузку файлов (с оф. ресурса bootstrap)
- Через пакетный менеджер npm (нужен пакетный менеджер NPM)

# Через Content Delivery Network (CDN)

Внутри тега <head> добавьте ссылки на файлы Bootstrap CSS

```
<link href="https://stackpath.bootstrapcdn.com/bootstrap/5.1.3/css/bootstrap.min.css"  
rel="stylesheet">
```

# Через загрузку файлов

1. Перейдите на официальный сайт Bootstrap (<https://getbootstrap.com/>) и нажмите на кнопку "Download" для загрузки архива с файлами Bootstrap.
2. Распакуйте скачанный архив и скопируйте файлы bootstrap.min.css в папку вашего проекта.
3. Внутри тега <head> вашего HTML-документа добавьте следующий код для подключения файлов:

```
<link href="путь_к_файлу/bootstrap.min.css" rel="stylesheet">
```

# Через пакетный менеджер npm

1. Откройте терминал или командную строку и перейдите в директорию вашего проекта.
2. Выполните следующую команду для установки Bootstrap через npm:  
*npm install bootstrap*
3. После установки Bootstrap добавьте ссылки на файлы CSS и JavaScript в ваш HTML-документ

# Как работать с bootstrap

Когда мы работаем с Bootstrap, мы просто используем готовые классы, чтобы придать стили нашим элементам на веб-странице.

- Добавление классов к HTML: Мы добавляем специальные классы Bootstrap к нашему HTML-коду. Например, если мы хотим создать красивую кнопку, мы просто добавляем класс "btn btn-primary" к нашему `<button>` элементу.
- Использование готовых компонентов: Bootstrap предоставляет нам готовые компоненты, такие как кнопки, формы, навигационные панели и многое другое. Мы можем использовать эти компоненты, просто добавляя классы к соответствующим элементам в нашем HTML.
- Гибкость и адаптивность: Bootstrap также предоставляет гибкую сетку, которая помогает нам организовать элементы на странице. Мы можем использовать классы сетки, чтобы определить, как наши элементы будут выглядеть на различных устройствах, таких как компьютеры, планшеты и мобильные телефоны.
- Простое расширение и кастомизация: Если нам нужно что-то изменить или добавить, мы можем легко настроить Bootstrap под наши нужды. Мы можем изменить цвета, шрифты, размеры и многое другое, просто переопределяя некоторые стили или добавляя собственные классы.

# Основы Bootstrap

- Сетка
- Типография и стили
- Компоненты

# Сетка

GRID - оф. документация - [ТЫК](#)

документация на русском языке - [ТЫК](#)

Контейнер - это основной элемент сетки Bootstrap, который содержит все другие элементы сетки. Он помогает ограничить ширину содержимого и центрировать его на странице.

Строка - это горизонтальная группа столбцов в сетке Bootstrap. Она используется для разделения содержимого на логические секции.

Столбец - это вертикальный блок, который используется внутри строки для размещения содержимого. Столбцы распределяются внутри строки на 12 колонок.

# Типография и стили

типоврафия на русском - [ТЫК](#)

типоврафия оф. документация - [ТЫК](#)

работа с цветом на русском - [ТЫК](#)

работа с цветом оф. документация - [ТЫК](#)

утилиты для верстки (отступы) на русском - [ТЫК](#)

утилиты для верстки (отступы) оф. документация - [ТЫК](#)

# Компоненты

кнопки - [на русском](#) | [оф.документация](#)

модальное окно - [на русском](#) | [оф.документация](#)

карусель (слайдер баннеров) - [на русском](#) | [оф.документация](#)

аккордеон - на русском (нету) | [оф.документация](#)

dropdown - [на русском](#) | [оф.документация](#)

# Продвинутые возможности Bootstrap

- Адаптивный дизайн
- Пользовательские стили и темы
- Инструменты и ресурсы

# Адаптивный дизайн

breakpoints - аналогия с media - ограничение по размеру экрана - [оф. документация | на русском](#)

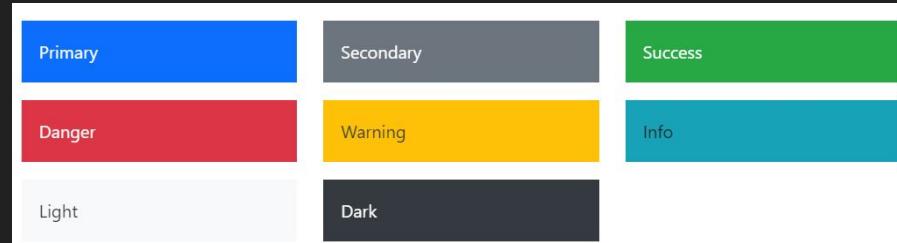
- Использование классов сетки: Bootstrap предоставляет классы для разделения содержимого на колонки. Например, col-12, col-sm-6, col-md-4, col-lg-3 и т. д., где число указывает количество колонок, которые элемент займет на сетке, а префикс определяет размер экрана, на котором этот класс будет действовать. Эти классы позволяют создавать макеты, которые адаптируются под разные размеры экранов. Например, вы можете создать макет, в котором элементы расположены в 2 колонки на больших экранах, а на маленьких - в 1 колонку.
- Использование адаптивных классов: Bootstrap предоставляет ряд классов для контроля отображения элементов на различных размерах экрана. Например, d-none, d-sm-block, d-md-inline, d-lg-flex и т. д., где префикс указывает на размер экрана, на котором этот класс будет применен. Эти классы позволяют скрывать, показывать или изменять отображение элементов в зависимости от размера экрана.
- Использование адаптивных типографических классов: Bootstrap предоставляет также классы для создания адаптивного текста и содержимого. Например, text-center, text-sm-left, text-lg-right, которые управляют выравниванием текста на различных размерах экрана.
- Использование медиа-запросов: Возможность создавать собственные медиа-запросы и комбинировать их с классами Bootstrap для более точного контроля над адаптивным поведением элементов.
- Использование адаптивных изображений: Bootstrap предоставляет классы для создания адаптивных изображений, таких как img-fluid, которые автоматически подстраиваются под размеры контейнера.

# Пользовательские стили и темы

Для реализации смены темы требуется JS код - [ТЫК](#)

bootstrap можно настраивать через css переменные - [официальная документация | на русском](#)

настройка цветов темы - [официальная документация | на русском](#)



Мы используем подмножество всех цветов, чтобы создать меньшую цветовую палитру для создания цветовых схем, также доступных в виде переменных Sass и карты Sass в scss/\_variables.scss файле Bootstrap.

```
$theme-colors: (  
    "primary": $primary,  
    "secondary": $secondary,  
    "success": $success,  
    "info": $info,  
    "warning": $warning,  
    "danger": $danger,  
    "light": $light,  
    "dark": $dark  
);
```

# Концепция изучения UI фреймворков и библиотек

- документация предоставляет необходимую и достаточную информацию о функционале фреймворка/библиотеки
- практика закрепляет знания
- все ui фреймворки и библиотеки работают +- по одному и тому-же принципу

# bootstrap + react

Документация - [ТЫК](#)

Концепция - тот же bootstrap но в данном случае мы используем не классы, а компоненты так как react подразумевает под собой компонентный подход разработки UI

Примеры - [ТЫК](#)

# Ресурсы

- bootstrap - [ТЫК](#)
- bootstrap (v5) документация на русском - [ТЫК](#)
- bootstrap (v4) документация на русском - [ТЫК](#)
- bootstrap для react - [ТЫК](#)
- примеры bootstrap - [ТЫК](#)