

Полная программа

Vue.js разработчик

Быстрый JS-фреймворк для эффективной frontend-разработки

Длительность курса: 114 часов

Модуль 1. Основы Vue

Тема 1

Введение в курс, современная веб-разработка

Цель занятия

использовать основные инструменты современной разработки; инцидентировать проект vite; рассмотреть основы работы с проектом vite; вспомнить основы js необходимые для работы с vue.

Краткое содержание

знакомство; освоим основные javascript необходимые для работы с vue; структура курса, типы заданий, итоговый проект; используемые технологии в современном фронтенде; структура проекта, поиск нужной информации; основы работы с vite при и настройкой базового проекта; примеры работы с vue.

Тема 2

Vue, введение

Цель занятия

объяснить преимущества Vue; применять инструменты для быстрого развертывания проекта Vue; создавать и настраивать новый проект с vite; объяснить отличия Vue от React и других фреймворков; различия между options и composition API.

Краткое содержание

почему правильно выбрать Vue для фронтенд разработки; инструменты и ресурсы для разработчика; options и composition API; как установить необходимое, как начать новый Vue проект; vite starter, Vue developer tools, Vetur (VSC extension).

Домашние задания

Создание проекта с Vite

Цель
разворачивать проект с нуля, разобраться какие какие есть настройки в vite.config.js; кастомизация настроек базового проекта.

Тема 3

Шаблоны

Цель занятия

использовать встроенные директивы шаблонов Vue; различать v-if и v-show; написать переключения и списки в шаблонах; переключать class и style и другие атрибуты у элементов в шаблонах vue.

Краткое содержание

существующий синтаксис шаблонизатора; встроенные директивы и свойства - v-if, v-for, v-show, v-bind, v-for.

Домашние задания

Создание Vue-приложения

Цель
освоить работу с основными директивами Vue (v-for, v-if, v-show, v-bind, v-on), научиться динамически управлять отображением данных и событиями в шаблоне.

Тема 4

Компоненты, props и события

Цель занятия

создавать компоненты Vue, применять SFC; создавать связанные компоненты; организовывать обмен данными между компонентами; объяснить принципы разделения приложений на компоненты; использовать события в Vue для передачи данных.

Краткое содержание

почему компонентный подход популярен в современном фронтенде; как писать компоненты на Vue; как организовывать взаимоотношения данных и событий между ними; пример работы с props при передаче адреса картинки.

Домашние задания

Создание главной страницы / базовые компоненты

Цель
загружать данные; передавать данные в компоненты; работать с директивами.

Тема 5

Формы, слоты, директивы

Цель занятия

писать форму для получения данных из шаблона; объяснить возможности расширения v-model; применять SFC подход.

Краткое содержание

применение SFC подхода; организация приложения из многих взаимодействующих компонентов; кастомные директивы, v-model; директивы; props.

Домашние задания

Формы и работа с API

Цель
создавать формы и работать с данными в приложении.

Модуль 2. Уверенная разработка на Vue

Тема 1

Vue router

Цель занятия

использовать роутинг в Vue; написать конфигурацию роутера; использовать хуки роутера в компонентах.

Краткое содержание

что такое URL; как с ним работать JS; как добавить навигацию в SPA; использование роутера для создания многостраничных приложений; популярные хуки для выполнения функций до или после переключения - vue-router.

Домашние задания

Реализовать рутинг в приложении

Цель
научиться работать с Vue Router, передавать параметры, использовать navigation guards.

Тема 2

Pinia работа с данными

Цель занятия

различать преимущества центрального управления стейтом; добавлять State Management приложению Vue; писать гетер, экшен; писать данные в State из компонентов.

Краткое содержание

популярный стейт менеджер для Vue - Pinia; почему централизованное управление стейтом популярно в современном фронтенде; паттерны и используемые методы.

Домашние задания

Добавление стейта в приложение - Pinia

Цель
использовать pinia знать что такое state manager и как с ним работать.

Тема 3

Тестирование Vue приложений. 1

Цель занятия

различать виды тестов; писать юнит-тесты для Vue приложений; использовать особенности библиотеки jest.

Краткое содержание

Виды тестирования: модульное, компонентное, энд-ту-энд (E2E). Основные библиотеки: Jest (модульное тестирование), Cypress (E2E тестирование), PhantomJS (устаревший headless-браузер для тестов). Примеры тестирования компонентов и логики Vue.js.

Тема 4

Тестирование Vue приложений. 2

Цель занятия

различать виды тестов; писать юнит-тесты для Vue приложений; использовать особенности библиотеки jest.

Краткое содержание

Готовый проект с настроенным запуском тестов. Виды тестирования: модульное (unit), компонентное, энд-ту-энд (E2E). Основные библиотеки: Jest (unit), Cypress (E2E). Примеры unit-тестов для Vue. Примеры e2e-тестов с локаторами.

Домашние задания

Добавляем тестирование - unit tests

Цель
освоить тестирование Vue-приложения; написать unit-тесты для методов и компонентов; протестировать добавление в корзину, проверку логина и загрузку данных, а также реализовать e2e-тестирование; эмулировать аутентификацию через localStorage.

Тема 5

QA сессия

Цель занятия

получить ответы на вопросы по курсу и ДЗ.

Краткое содержание

обсуждение вопросов и проблемных тем, ДЗ.

Тема 6

Система реактивности Vue3, MVVM

Цель занятия

объяснить что такое MVVM; объяснить особенности change detection в vue; написать свою систему реактивности по аналогии с системой vue3; объяснить различия в системах реактивности между vue 2 и vue 3

Краткое содержание

MVVM; разработка своих примитивов реактивности: ref, shallowRef, reactive, computed, watch, watchEffect.

Тема 7

Typescript

Цель занятия

знать особенности и основные преимущества typescript; добавить typescript к существующему проекту на Vue; создать компоненты и сервисы Vue приложение на typescript; создать интерфейсы, модули и .ts файлы.

Краткое содержание

преимущества использования TypeScript в Vue; создание проекта с TypeScript; использование vue-property-decorator для работы с классами в Vue.

Модуль 3. Продвинутое и практические решения

Тема 1

Vue3 работа с библиотеками

Цель занятия

знать функционал, который добавляют популярные библиотеки для работы с composition API; использовать свои кастомные хуки.

Краткое содержание

vueuse library, i18n, vuelidate, awesome-vue.

Тема 2

Интеграции Vue: GraphQL

Цель занятия

знать возможности GraphQL; различать работу через REST vs GraphQL; добавлять GraphQL к Vue приложению.

Краткое содержание

разница между REST и GraphQL; как работает GraphQL (запросы, мутации, подписки); установка Apollo Client; работа с запросами и мутациями.

Тема 3

Интеграция Vue с WebSockets

Цель занятия

знать что такое WebSockets; объяснить разницу между HTTP и WebSockets; подключать WebSockets в Vue-приложение.

Краткое содержание

принципы работы WebSockets; различия между HTTP-запросами и WebSockets; установка socket.io-client; подписка на события, отправка сообщений.

Домашние задания

Интеграция с GraphQL API и WebSockets

Цель
научиться подключаться к GraphQL API, получать и отображать данные, а также обновлять информацию в реальном времени с помощью WebSockets; управлять состоянием корзины.

Тема 4

Авторизация и аутентификация

Цель занятия

знать отличие авторизации от аутентификации; познакомиться с JWT; использовать различные техники авторизации и аутентификации на фронтенде.

Краткое содержание

теория аутентификации и авторизации; разбор различий между ними; изучение JWT; живой пример аутентификации и авторизации в проекте.

Домашние задания

Авторизация и аутентификация в приложении на JWT

Цель
научиться использовать авторизацию и аутентификацию в своих приложениях.

Тема 5

SSR с Nuxt

Цель занятия

объяснить преимущества SSR; настроить pre-rendering страниц frontend приложения Vue; создать с нуля Nuxt в приложении с Vue; добавить страницы и роутер в Nuxt приложении.

Краткое содержание

разница между SSR, SSG и CSR; устанавливаем и настраиваем Nuxt 3; создаем страницы и настраиваем маршрутизацию.

Домашние задания

Typescript

Цель
перевести приложение (GraphQL + WebSockets) на TypeScript, внедрить строгую типизацию; описать интерфейсы для данных, props, computed и методов, а также для получаемых сообщений из GraphQL и WebSocket, обеспечивая безопасность и предсказуемость кода.

Тема 6

Работа с компонентами и библиотеками в Nuxt

Цель занятия

использовать сторонние библиотеки компонентов в Nuxt; объяснить преимущества Vuetify, Quasar, Bulma, ElementUI.

Краткое содержание

разбираем работу с библиотеками компонентов в Nuxt; подключаем Vuetify и Quasar; стилизуем страницы с помощью UI-библиотеки.

Домашние задания

Nuxt 3-проект с SSR

Цель
освоить базовые возможности Nuxt 3, включая SSR, маршрутизацию и динамические страницы, а также научиться работать с UI-библиотеками для стилизации компонентов.

Тема 7

Выбор темы и организация проектной работы

Цель занятия

обсудить идеи студентов для финального проекта и возможные этапы реализации; предложить готовые направления для разработки итогового проекта.

Краткое содержание

будут получены ответы на вопросы по проекту, ДЗ и по курсу.

Домашние задания

План проекта

Цель
описать планируемый проект.

Тема 8

Анимация во Vue + Web Components

Цель занятия

добавить анимацию к спискам; добавить анимацию между рутами приложения; создать собственную анимацию; управлять анимацией из свойств CSS; знать популярные библиотеки готовых эффектов; ориентироваться в веб спецификациях Custom Elements и Shadow DOM.

Краткое содержание

встроенные решения для анимации в Vue; примеры и выполнение нескольких самостоятельных работ; transition tag, CSS animations, keyframes, bezier curve; Custom Elements; Standalone Elements; vue-CLI - сборка в виде веб-компонента

Тема 9

Организация кода и паттерны Vue

Цель занятия

структурировать файлы и код в приложении по мере роста кодовой базы: service pattern, fabric pattern, axios; объяснить provide/inject.

Краткое содержание

структурирование файлов и кода в масштабируемом приложении; паттерны: Service Pattern, Factory Pattern; работа с axios для запросов; использование тега <component>; механизм provide/inject для передачи данных между компонентами.

Тема 10

QA / Code review

Цель занятия

анализировать код проекта; проверять код коллег; поддерживать высокий уровень кодовой базы проекта.

Краткое содержание

жизненный цикл задачи; стратегии ветвления git; как и зачем проводить Code Review; на что обращать внимание.

Тема 11

Сборка и деплой проекта, cloud, CI/CD

Цель занятия

рассмотреть популярные технологии для работы с IT проектами - от планирования до деплоя.

Краткое содержание

Continues Integration, Continues Delivery, Continues Deployment; варианты запуска проекта в облаке; автоматический CI/CD pipeline в gitlab.

Модуль 4. Проектная работа

Тема 1

Консультация по проектам и домашним заданиям

Цель занятия

обсудить тему проектной работы; получить ответы на вопросы по проекту, ДЗ и по курсу.

Краткое содержание

требования к результату проекта и итоговой документации; вопросы по улучшению и оптимизации работы над проектом; затруднения при выполнении ДЗ; вопросы по программе.

Домашние задания

Проектная работа

Цель
научиться грамотно планировать проект; декомпозировать реализацию; выбирать технологии и архитектуру; оценивать сложность проекта.

Тема 2

Защита проектных работ

Цель занятия

защитить проект и получить рекомендации экспертов.

Краткое содержание

презентация проектов перед комиссией; вопросы и комментарии по проекту.