

JavaScript - финал

Что прошли;

Что стоит пройти;

Где практиковаться;

Вопросы по на собеседованиях;

Проектная работа.

Что прошли

Базовый синтаксис JavaScript (переменные, условия, функции, типы данных)

Взаимодействие браузера и JavaScript-а (DOM, события и тд)

Прототипное наследование (основа JavaScript-а)

Продвинутая работа с функциями
(рекурсия, замыкание, области видимости и паттерны)

ООП (классы и методологии)

Асинхронность (событийный цикл, промисы, webWorker-ы и serviceWorker-ы)

Что стоит пройти

Более углубленные знания паттернов и методологий

Спецификация ECMAScript-а

Более глубокое понимание ООП (в частности SOLID)

Регулярные выражения

Алгоритмы

Уязвимости в браузере

Серверный JavaScript (node.js)

Изучение современных фреймворков и библиотек

Работа с сервером (работа js с базой данных и с файловой системой)

Где практиковаться

LeetCode - [ТЫК](#)

CodeWars - [ТЫК](#)

Тестовые задания - можно получать при прохождении собеседований

Пет Проекты - на основе бесплатных сервисов реализовывать свои проекты

Open Source проекты - много библиотек в которые можно внести свой вклад или же вообще написать свой open source проект

Решение математических задач (алгоритмических)

ПЕТ ПРОЕКТЫ

Базовое изучение технологии - реализация приложений COUNTER и TODO на разных фреймворках

Сравнительный анализ (на основе типовых задач) - например решение вычислений чисел Фибоначчи с разными подходами и сравнение какой подход более оптимальный по параметрам скорость и удобство

Пет проекты решающие собственные повседневные задачи - например приложение отслеживания трат

Пет проекты на основе open source API - пример приложения погоды

OPEN SOURCE API

[Weather API](#) - бесплатный back-end для получения прогноза погоды

[Fake Store](#) - бесплатный back-end реализующий онлайн магазин

[JSONPlaceholder](#) - бесплатный сервис для создания фейковых запросов

[New API](#) - бесплатный back-end с предоставлением новостей

[Vercel](#) - можно развернуть свой почти полноценный back-end со своей архитектурой и своей базой данных

Вопросы по на собеседованиях (пример)

Назвать 7 «примитивных» типов данных?

Чем отличаются let, const и var?

Как работают условия в JS (if, ?, switch case)

Что такое опциональная цепочка .?

Какие есть области видимости в JS?

Что такое цикл и что такое рекурсия, в чем разница?

Что такое функция и что такое function expression и function declaration?

Что такое стрелочная функция и какие особенности данной функции?

Что такое this в JavaScript и назвать особенности?

Что такое call, apply, bind и зачем нужны?

Что такое замыкание и как работает?

Что такое Set и Map в чем разница?

Назвать основные методы массива?

Что такое преобразование типов и чем отличается == от ===

Что такое итерируемый объект в JS?

Что такое DOM, что такое async и defer?

Что такое событие в JS (браузерный), как работает событие (всплытие и погружение))

Что такое делегирование событий?

Что такое cookies, local storage, session storage?

Что такое прототипное наследование?

Как работают class в JavaScript?

Что такое ООП и назвать 4 основных принципа ООП?

Что такое событийный цикл и как он работает?

Что такое promise?

Что такое асинхронные функции?

Что такое генератор

Как получать данные с back-end-a?

Что такое web-worker и service-worker?

Что такое модули в JS и зачем они нужны?

Проектная работа - онлайн магазин

На основе Fake Store реализовать свой онлайн магазин

Использовать UI библиотеку
(либо написать свою либо использовать bootstrap)

Использовать модули (export/import) для разделения логики проекта

Использовать JSDoc или readme.md для реализации документации к проекту

Разместить исходник проекта на удаленном репозитории в GITHUB

Развернуть проект на vercel или github pages

Ресурсы

Лучший учебник по JavaScript-у - [ТыК](#)

Лучшая документация по JavaScript-у - [ТыК](#)

Спецификация по ECMAScript-у - [ТыК](#)

CodeWars - [ТыК](#)

LeetCode - [ТыК](#)

RoadMap по изучению JavaScript-а - [ТыК](#)

Хороший учебник по паттернам программирования - [ТыК](#)