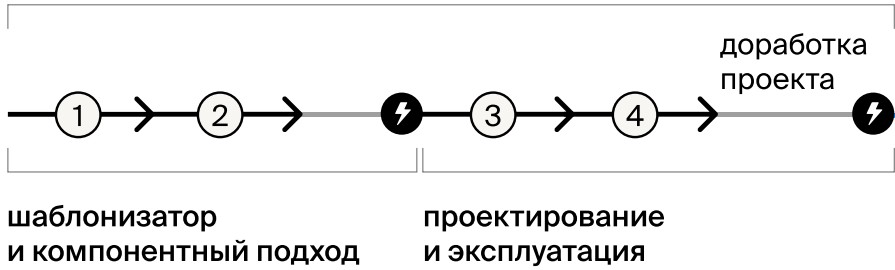


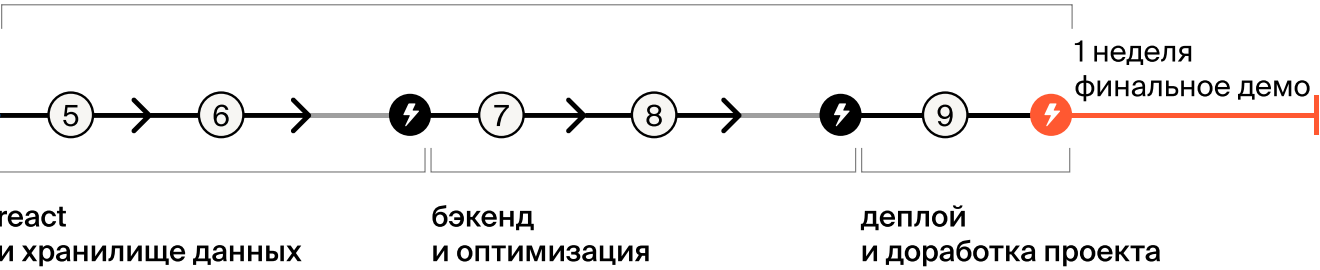
# Карта курса

## Мидл-фронтенд разработчик

модуль I самостоятельный проект



модуль II командный проект



два курса для самостоятельного прохождения:

JS

алгоритмы

1 номер спринта

1 неделя каникул: знакомство и подготовка к командному модулю

неделя закрытия хвостов с жёстким дедлайном в конце, до него нужно получить зачёт по всем работам двух спринтов

завершение программы

### 0 курс

20 часов

### Вступительное тестирование

Вы потренируетесь решать задачи на базовые принципы JavaScript, алгоритмы и работу с Canvas. Познакомитесь с форматом и структурой программы. Узнаете, какие вас ожидают проекты на платном обучении. Спойлер: самостоятельный и командный проекты, код которых будет проверять команда профессиональных ревьюеров.

Каков путь?

Этап 1. Базовый JavaScript

Этап 2. Новый ранг: падаван

Мидл разработчик.

Приземление в реальность

Билет в новый мир

### 1 курс

11 недель

Параллельное  
прохождение

### Самостоятельный-параллельный курс: Повторить всё

Вас ждёт повторение важных концепций в JavaScript.

Познакомитесь ближе с DOM-деревом и управлением содержимым DOM-узлов.

Глубже изучите асинхронность: начнёте с `setTimeout` и `setInterval`, продолжите с `Event loop` и нулевой задержкой, «приправите» знания чейнингом и `Callback`.

JavaScript, Глава 1

Переменные, типы данных, функции, массивы, объекты

DOM API

BOM и DOM, работа с узлами, свойства элементов, эффективность методов DOM

Асинхронность. Начало

Синхронный код, стек вызовов, очередь событий, `setTimeout`, `setInterval`, вызов `setTimeout` из `setTimeout`, нулевая задержка `Event loop`.

JavaScript, Глава 2

Типы данных II, ES6+, замыкания, ООП, прототипы, регулярные выражения

Асинхронность. Тёмный рыцарь

`Callback` и чейнинг, `promises`, `async`, `await`

# I Модуль.

## Самостоятельный проект «Веб-мессенджер»

В течение первого модуля вы выполните полноценное SPA: веб-мессенджер. Никаких фреймворков и библиотек по клиентской части, только JavaScript, TypeScript. Ещё до знакомства с React и другими библиотеками вы начнёте понимать, как они устроены. В работе над первым проектом вы реализуете часть функций таких библиотек, как, например, React и React Router.

### 2 курс

5 недель

2 вебинара

## Шаблонизатор и компонентный подход

Вы погрузитесь в работу браузера: компонентный подход, отображение, HTTP и REST больше не будут загадкой. Изучите алгоритм работы популярных шаблонизаторов и напишете собственную реализацию. С помощью знаний о компонентах раскроете всю магию фреймворков. Начнёте изучать архитектуру, паттерны разработки, дизайн CSS и его методологии.. Здесь начнётся ваша дружба с TypeScript.

### Спринт 1

#### Хостинг для кода и проекта

Удалённый репозиторий, деплой в Netlify, деплой стабильной ветки, автодеплой, хостинг проектов.

#### Клиент-серверное взаимодействие I

HTTP, REST

#### NodeJS

Ограничения JavaScript в браузере. Node.js, Express

#### Модули

Зачем нужны модули, что было до ES6, импорты ES6, заикливание, поддержка браузеров, импорты в Node.js

#### Шаблонизация

Шаблонизаторы, реализации, пишем свой шаблонизатор

#### Отображение

Блочная модель, основы управления блоками, схлопывание отступов, арифметика блока, инлайн блоки, позиционирование, CSS сегодня

#### Архитектура CSS

Дизайн языка, методологии, CSS через JS, препроцессоры, хранение файлов в проекте

#### Рубрика мастерим сами

полезные утилиты

Last, first, range, rangeRight

#### UI / UX

Как разрабатывать прототип для проекта. В темах UI и UX нужно ориентироваться не только дизайнерам, но и веб-разработчикам. Расскажем, из чего состоит процесс разработки прототипов и что нужно знать, чтобы сделать макет

#### Работа над проектом первого модуля

### Спринт 2

#### Компонентный подход

SPA

#### Браузерные события

События в браузере, подписки, распространение событий

#### Клиент-серверное взаимодействие I

HTTP, REST

#### TypeScript I

Проблемы со стандартами, отсутствие строгой типизации, TypeScript, типы данных, вывод и приведение, ООП, обобщение типов, типизация JS-кода

#### Паттерны

Архитектура в общем смысле, методологии, декомпозиция, MV\*

#### Компонента

Представление, Event Bus, триггеры, блок, страница

#### API

XMLHttpRequest, Fetch API, реализация fetch

#### CodeStyle

Статический анализатор кода, прикручиваем ESLint, правила в ESLint, добавляем проверку код-стайл при деплое

#### Работа над проектом первого модуля



Неделя доработки проектов

## 3 курс

11 недель

Параллельное  
прохождение

# Самостоятельный-параллельный курс: Алгоритмы

Курс посвящён алгоритмам и структурам данных. Вы узнаете, как они помогают создавать производительные программы, писать меньше кода и не тратить время на изобретение велосипеда.

### Введение в алгоритмы

Основные понятия: передача данных и рекурсия, оценка сложности алгоритма: асимптотический анализ, основные функции сложности, введение в структуры данных

### Сортировка I

Пузырьковая сортировка, сортировка вставками, связный список

### Сортировка II

Стек и очередь, быстрая сортировка, сортировка слиянием

### Хеш-таблицы

Хеш-таблицы, методы разрешения коллизий и операции с хеш-таблицей

### Деревья и графы

Деревья. Операции с бинарным деревом поиска и бинарной кучей. Графы. Структуры данных: повторение

### Сжатие

Сжатие данных. Алгоритм Хаффмана

### Поиск

Алгоритмы поиска. Бинарный поиск. Поиск в ширину и в глубину

### Трудоустройство

Решаем задачи с собеседований

## 4 курс

5 недель

2 вебинара

# Проектирование и эксплуатация

Здесь вы познакомитесь со смежными областями: тестированием и безопасностью.

Узнаете, как централизовать логику переходов по страницам и сделать большой шаг к полноценному SPA. Вы научитесь применять протокол связи WebSocket, погрузитесь в работу с Docker.

К окончанию этого курса вы завершите свой первый проект!

### Спринт 3

#### Роутинг

Переходы, History API, роутер

#### Как работать с запросами

Работа с запросами, обработка и хранение

#### Тесты

Зачем нужны тесты, TDD / BDD, как описывать баги правильно, как писать тест-кейсы, пишем тесты на JS / TS. Chai и Mocha

#### Рубрика мастерим сами полезные утилиты

Trim, merge, set, isEqual, CloneDeep, queryString

#### Клиент-серверное взаимодействие II

HTTP 2/0, WebSockets, gRPC

#### API чатов

CORS-запросы, использование «ручек» чата

#### WebSocket

Подключаем real-time сообщения

#### Безопасность веб-приложений

ИБ, угрозы, риски, примеры уязвимостей, инъекции (SQL, command), Crlf, HTTP header splitting, HTTP parameter contamination, path traversal, Race conditions, криптография, DoS, DDoS, SSRF, SOP, CORS, cookie, CSRF, XSS, CSP, clickjacking

#### Работа над проектом первого модуля

### Спринт 4

#### TypeScript II

Объединение типов, ключевые слова TS, Utils TS (readonly-freeze, omit, pick), ESNext и фишки 3.7+ версии

#### Webpack

Проблема без сборщика, как собрать index.js, модули и TypeScript. Как применять плагины / loaders. Пишем свой loader, watch, запуск с Express

#### Эксплуатация приложения

Аудит пакетов, обновление, версионирование, lock-files, код-ревью, .editorconfig, ESLint эксплуатация, конфиги, правила, пресеты, игнорирование правил, автоматическое исправление, ESLint + TS, Скрипты (makefile, npm, scripts, gulp), precommit

#### Docker

Разворачивание сервиса (share machine, virtualization, Docker), Dockerfile, команды докера, nginx для фронта, деплой docker-image

#### Размещаем статику

CDN, что это и зачем, пробуем surge, кеширование

#### Рубрика мастерим сами полезные утилиты

Omit, classnames, take, unzip

#### Работа над проектом первого модуля



Неделя доработки проектов

# Каникулы. Командная работа

Ура, каникулы! Но не всё так просто, — есть небольшая домашка и материал для ознакомления. Главная задача на каникулы: настроить взаимодействие внутри вашей команды и познакомиться с будущими «коллегами».

Вы продумаете функционал вашего проекта, подготовите макеты и дизайн игры. Узнаете больше о командном взаимодействии и цикле работы над продуктом: да, тот самый Agile и декомпозирование задач. Здесь вам откроются секреты кросс-ревью.

Командное взаимодействие	Командный Git	Кросс-ревью
Декомпозиция задач	Консоль vs GUI. Rebase и Merge	Зачем и как правильно?

# II Модуль.

## Командный проект «Веб-игра»

Изучив нативные возможности JavaScript, во втором модуле вы перейдёте к высокоуровневым инструментам. В течение первого модуля ваша команда выполнит полноценную браузерную игру, похожую на: Flappy Bird, Doom 2D, покер, дурака, монополию и подобные карточные игры

### 6 курс

5 недель

1 вебинар

1 демо

5 созвонов

с ментором

## Привет, React. Здравствуй, хранилище данных

Вы начнёте свою работу с React и его окружением. Снова скажете «привет» верному старому другу, чьё имя TypeScript. Узнаете, что такое Canvas API.

Узнаете, как хранить данные и тестировать при помощи Jest. Разберётесь в чем особенность безопасности в React.

После этого курса ваш проект приобретёт вид готового продукта

### Спринт 5

#### Разработка современных приложений

Как разрабатывали раньше, компонентный подход и реактивное программирование, VDOM, Shadow DOM, жизненный цикл компоненты, шаблонизаторы, HTML templates, иные веб-компоненты, JSX

#### ReactJS

Как устроен react.js, как пользоваться, простая компонента. CRA — не панацея

#### Компоненты в React

Fc / class components, render в DOM, props, state, lifecycle, events, fragments. Композиция компонент, обработка ошибок, propTypes

#### Router

React router, пример создания страниц с разными роутами, примеры редиректов, примеры использования history, подстраницы.

#### API

Чем делать запросы? Как делать запросы в React? Как обработать запросы и куда их класть? Как прокидывать данные между компонентами?

#### Webpack, React

Настройка, ладеры, плагины

#### Refs

Что это и как пользоваться, HTML и Ref

#### Canvas API

Как рисовать, best practice, движение, React.js + Canvas

#### Игровые механики

Паттерны при разработке игр и их применение, таймеры, request animation frame

#### TypeScript в React

Как использовать и все тонкости. Типы компоненты и пропсов

#### Работа над командным проектом

### Спринт 6

#### Ещё больше React

Render Props, контексты, «хуки» и «хоки». Пишем свой «хук»

#### Базы данных

(SQL, NoSQL, MongoDB, WebSQL, Redis и другое). Зачем фронтендеру БД. SQL, NoSQL, их отличия, пример проектов на NoSQL+ JS и SQL + JS

#### Хранение данных

Проблема хранения и обработки данных. Redux и его схема, подключение Redux к проекту, его сущности. MobX, stateless

#### Тестирование

Jest + React. Jest + React + Redux

#### Service workers

Что это и для чего? Как подключить?

#### Безопасность в react

SQL injection, XSS, CSP, CORS

#### Полезное WEB API

Fullscreen API, Canvas, performance (time), GeoAPI, Notifications API

#### Работа над командным проектом



Неделя доработки проектов



7 курс

5 недель

1 вебинар

1 демо

5 созвонов с ментором

Спринт 7

# Проектирование и эксплуатация

Вы настроите в игре Server Side Rendering, разберётесь в чём его отличие от Client Side Rendering. Научитесь контролировать память. Кроме того, вы погрузитесь в работу серверной части приложения, допишете часть «ручек» с помощью своего дополнительного API. Создадите телеграм-бота, сделаете темизацию.

**Как делать продукты, чтобы они не «лагали» у пользователей**  
Управление памятью в JS, утечки памяти

**Работа с ошибками**  
Проблема, как можно обрабатывать ошибки, где хранить и как реагировать, улучшение взаимодействия с пользователем

**Server Side Rendering**  
Проблема-решение, примеры с React, сравнение преимуществ подходов

**Хранилища в WEB**  
LocalStorage, cookie, sessions, indexed DB, cache API, обмен данными между вкладками, NodeJS

**Другие способы авторизации**  
Классическая, OAuth, JWT, паттерны, кратко об остальном

**Полезное WEB API**  
Media API, FileSystem API, Gamepad API, Pointer Lock API, Web workers API, Ambient Light, WebGL, WebVR API

**Работа над командным проектом**

Спринт 8

**Бэкенд для фронтов: зачем и что?**  
Настройка load-балансера, S3 и highload-фронтенда

**Подготовка окружения**  
Подключаем PostgreSQL. Запускаем MongoDB

**Основы PostgreSQL**  
Работа с таблицами. Типы данных и их модификация. Подзапросы. JOIN. Индексы

**«Свадьба» PSQL & NodeJS**  
Подключаем ORM. Модели. CRUD. Связи. JOIN и транзакции

**Добавляем свою логику**  
Переключение на тёмную, разноцветную или праздничную тему оформления. Sequelize

**Полезное для продукта**  
Биллинг и Яндекс.Оплата. DaData. Метрики. Webhook деплоя на GitHub. Телеграм-бот

**Работа над командным проектом**

♥

Включенный хакатон

⚡

Неделя доработки проектов

8 курс

2 недели

1 финальное демо

1 созвон с ментором

Спринт 9

# Собираем, дорабатываем и выступаем

Последний курс, дошлифовка проекта и показ проектов! В этом спринте вы закончите последние правки, погрузитесь в nginx и настроите CI/CD, задеплоите проект. Программа обучения завершится финальным мероприятием, где вы покажете функциональность игры, поделитесь выбранными решениям и расскажите о процессах в вашей команде.

**Глубже в nginx**  
Настройка статики и прокси, логирования. Настройка Load Balancing, настройка кеширования и сжатия, HTTP 2/0.

**DevOps**  
Домены, как строятся, поддомены, уровни, DNS записи, SSL, wildcard, подготовка проекта к деплою на VPS, выкладка на VPS. Яндекс.Облако

**Настройка CI/CD в Github Actions**

**Доработка командного проекта**  
Как рассказать о проекте на финальном мероприятии?

**Конец?**

⚡

Завершение программы

# Трек программы трудоустройства

В карьерном треке вас продолжат сопровождать эксперты. Параллельно основной программе или после её завершения вы сможете пройти подготовку к получению job offer. Научитесь составлять резюме, писать сопроводительные письма, собирать портфолио и проходить собеседования. В финале – проделаете все шаги с нуля до трудоустройства или повышения на текущем месте работы. Мы поддержим.

## Трудоустройство. Новые возможности

### Стратегия профессионального развития

Вы погрузитесь в мир возможностей, познакомитесь с различными карьерными путями развития. Оцените все за и против трудоустройства. Блок будет включать в себя HR-консультацию по дальнейшему развитию. Кроме того, вы пройдёте тестовое скрининг интервью и получите советы по представлению себя, и рекомендации по подходу к ответам на типичные вопросы IT-рекрутера.

### Улучшаем артефакты: резюме, портфолио и сопроводительное письмо

Совместно с комментариями и рекомендациями HR-консультанта, вы сделаете качественное и цепляющее резюме, улучшите описание портфолио.

### Стратегия поиска и её нюансы.

### Неформальная сторона трудоустройства и нетворкинг

Узнаете о новых путях поиска работы. Что такое слабые связи и почему «мир тесен», когда никогда описывает IT-круг.

### Собеседования. Какие бывают и как проходить?

Блок включает в себя обязательное участие в тестовом собеседовании с экспертом. Продлится 1-1.5 часа, где вы получите рекомендации по развитию навыков, обратную связь по знаниям и понимаю процессов разработки.

### Собеседования. Копаем глубже. Готовимся тщательней

Вы пройдёте второе тестовое собеседование, более прицельное на JavaScript или алгоритмы. Продлится 1 час и пройдёт в серьёзной обстановке.

### Выбор. Как расставить приоритеты и выбрать оффер?

## Повышение. Новая ответственность

### Стратегия профессионального развития

### Как коммуницировать о повышении?