

# Фулстек-разработчик расширенный

20 месяцев

продолжительность курса

15 проектов

в портфолио



HTML, CSS, JavaScript

Вводный модуль

31 час

HTML и CSS: лэйаут,  
доступность и интерактив

Спринт 1. Основы вёрстки  
Спринт 2. Доступность и подходы к организации стилей

6 недель

Дизайн, адаптивная вёрстка,  
декорирование

Спринт 3. Адаптивность и графика  
Спринт 4. Декорирование, подходы и инструменты

6 недель

Базовый JavaScript

Спринт 5. Основы JavaScript  
Спринт 6: Работы с событиями и формами в JS. Сборка кода и модульность

6 недель

Продвинутый JavaScript

Спринт 7: Продвинутая работа с данными, валидация, работа с API

3 недели

TypeScript и ООП

Спринт 8: TypeScript и ООП  
Спринт 9. Принципы разработки и архитектура ПО

6 недель

Интерфейсы с использованием  
React

Спринт 10: Основы React

3 недели

Роутинг и авторизация

Спринт 11: Роутинг и авторизация

3 недели

Тестирование  
React-приложений

Спринт 12: Тестирование React-приложений

3 недели

Проектный месяц

Спринт 13: Работа над крупным проектом

1 месяц

Основы Node.js, Express  
и MongoDB

Спринт 14: Основы Node.js, Express и MongoDB

3 недели

PostgreSQL и Nest.js

Спринт 15: Разработка бэкенда на Nest.js  
Спринт 16: Базы данных и PostgreSQL

5 недель

Деплой и автоматизация

Спринт 17: Тестирование бэкенда,  
деплой и автоматизация

3 недели

Обеспечение безопасности  
приложения

Спринт 18: Обеспечение безопасности приложения

3 недели

# Фулстек-разработчик расширенный

**20 месяцев**

продолжительность курса

**15 проектов**

в портфолио



Проектный месяц

Спринт 19: Работа над крупным проектом

2 месяца

Python  
для fullstack-разработчика

Спринт 20. Основы Python. Настройка рабочего места для работы с Python.  
Спринт 21. Объектно-ориентированное программирование в Python.

6 недель

Python frameworks

Спринт 22. Знакомство с Django.  
Django Rest Framework.  
Спринт 23. Асинхронный Python. Flask.  
Спринт 24. FastAPI: быстрый старт, быстрый API.

7 недель

Веб серверы и деплой  
Python приложений

Спринт 25. NGINX+GUNICORN  
UVICORN+Python+Docker

2 недели

# HTML, CSS, JavaScript.

## Вводный модуль

---

01

31 час

Вы узнаете, как устроены программа и обучение. Ознакомитесь с организационными деталями. Почувствуете себя в роли разработчика и поймёте, хотите ли развиваться в этом направлении. Изучите базовый синтаксис HTML и CSS. Научитесь размещать блоки на веб-странице, менять шрифт и цвета. Напишите простую программу на JavaScript и сделаете страницу интерактивной.

## Содержание

---

Темы

1. Как всё устроено
  2. Первый код
  3. Что такое HTML и CSS
  4. Базовые CSS-свойства
  5. Больше CSS
  6. JavaScript. Начало
  7. JavaScript. Приземление в реальность
  8. О том, что дальше
  9. Как устроено обучение
  10. Вёрстка — продолжение дизайна
  11. Готовимся писать код
  12. Файлы в проекте
-

6 недель,  
2 проекта

Создадите одностраничный сайт и напишете CSS для работающего приложения.

Научитесь работать со шрифтами в вебе. Разберётесь, как писать код, корректный для браузеров и поисковых систем. Изучите все способы организации каркаса вашей страницы. Научитесь настраивать страницы для поисковых роботов. Поработаете с видео-, аудио- и другими интерактивными элементами страниц. Поймёте, как и с какими событиями на странице может работать CSS. Разберётесь, какие элементы на странице может создавать язык стилей. Научитесь стилизовать элементы в разных состояниях и в разных ситуациях (например, в зависимости от порядка их появления). Разберётесь в основных принципах и инструментах создания доступных веб-страниц.

## Спринт 1. Основы вёрстки

Темы

1. Семантика и глобальные атрибуты
2. Шрифты и типографика
3. Флексбокс-вёрстка
4. Позиционирование элементов
5. Grid Layout
6. Кодстайл
7. Основы Bash и Git

## Спринт 2. Доступность и подходы к организации стилей

Темы

1. Настройка страницы и мета
2. Внешний встраиваемый контент: видео, iframe, API
3. Дополнение блочной модели
4. Псевдоклассы и псевдоэлементы
5. Доступность
6. Разметка форм
7. Чудные селекторы
8. Стилизация форм
9. Продвинутый Git и Bash

### Каникулы

1 неделя

Время отдохнуть  
или доработать проекты

6 недель,  
2 проекта

Сделаете адаптивный сайт с разными цветовыми темами, а затем самостоятельно соберёте веб-приложение.

Научитесь комплексу приёмов, которые позволят корректно отображать страницы на разных устройствах и с разными настройками. Подготовите графику для работы с разными экранами. Научитесь организовывать код для создания цветовой темы сайта. Разберёте инструменты декорирования ваших страниц. Градиенты, тени, фильтры, режимы смешивания — всё это здесь. Научитесь использовать векторную графику на страницах и пользоваться интерактивными инструментами, которые встроены в HTML. Вишенка на торте спринта — анимации страниц.

## Спринт 3. Адаптивность и графика

|      |   |
|------|---|
| Темы | <ol style="list-style-type: none"><li>1. Подходы</li><li>2. Растровая графика</li><li>3. CSS-переменные</li><li>4. Единицы измерения и функции</li><li>5. Grid Layout, часть 2</li><li>6. Разработка интерфейса для разных устройств</li><li>7. Кроссбраузерность и вендорные префиксы</li><li>8. Подходы к написанию вёрстки</li><li>9. Git для взрослых</li></ol> |
|------|---|

## Спринт 4. Декорирование, подходы и инструменты

|      |  |
|------|--|
| Темы | <ol style="list-style-type: none"><li>1. Использование SVG</li><li>2. Переходы и 2D трансформации</li><li>3. Анимации</li><li>4. Вариативные шрифты</li><li>5. Декорирование</li><li>6. 3D трансформации</li><li>7. Современные интерактивные элементы</li></ol> |
|------|--|

### Каникулы

1 неделя

Время отдохнуть  
или доработать проекты

6 недель,  
1 проект

Создайте интерактивную веб-страницу: сначала сверстаете основу, а затем шаг за шагом добавите JavaScript-логику.

Научитесь работать с пользовательскими действиями — кликами, формами, вводом данных. Пропишите поведение элементов страницы, разделите код на модули и соберёте проект с помощью современного инструмента Vite.

## Спринт 5. Основы JavaScript

Темы

1. От разметки к программированию
2. Переменные и типы данных
3. Функции: простые концепции
4. Основы работы с данными
5. Дебаггинг

---

## Спринт 6: Работы с событиями и формами в JS. Сборка кода и модульность

Темы

1. Что делать, когда не знаешь, что делать
  2. Знакомство с DOM
  3. Создание и изменение элементов страницы
  4. Работа с событиями
  5. Работа с элементами форм и кнопками
  6. Модули в JS
  7. Сборка кода на Vite
-

3 недели,  
1 проект

Научитесь делать запросы к серверу, обрабатывать ответы и строить взаимодействие интерфейса с реальными данными.

Создадите формы с валидацией, подключите пагинацию, реализуете логику, при которой пользователь получает обратную связь от сервера и видит результат своих действий на странице.

В финале соберёте полноценную таблицу, которая умеет загружать данные с сервера, обрабатывать ввод и обновляться без перезагрузки.

## Спринт 7: Продвинутая работа с данными, валидация, работа с API

Темы

1. Асинхронность
2. Работа с API: запросы к серверу
3. Флоу пользовательского действия с подтверждением сервера
4. Пагинация
5. Регулярные выражения
6. Валидация формы

### Каникулы

1 неделя

Время отдохнуть  
или доработать проекты

6 недель,  
1 сквозной проект

Разработаете с нуля полноценное клиентское веб-приложение WEB-ларёк.

Сначала перепишите привычный JavaScript-код с использованием TypeScript, структурируете его с помощью классов и интерфейсов, а затем добавьте архитектуру и паттерны проектирования и подготовите окружение для масштабируемой разработки.

В результате вы получите хорошо организованное приложение с чистым кодом, строгой типизацией и продуманной структурой.

## Спринт 8: TypeScript и ООП

Темы

1. Введение в TypeScript
2. Типизация кода
3. Введение в ООП
4. Принципы ООП
5. Основы TypeScript

## Спринт 9: ООП в интерфейсах: простые (и не только) концепции

Темы

1. Архитектура приложения
2. Расширенный TypeScript и ООП
3. Паттерны проектирования
4. Работа с окружением

## Каникулы

1 неделя

Время отдохнуть  
или доработать проекты

3 недели,  
1 проект

Реализуете страницу блога с настраиваемым  
через CSS-переменные дизайном.

Изучите библиотеку React и её экосистему.  
Научитесь создавать и структурировать проекты на React,  
создавать компоненты и управлять потоками данных.  
Освойте функциональный подход к разработке на React  
и научитесь писать эффективный производительный код  
с помощью хуков.

## Спринт 10: Основы React

Темы

1. Концепция SPA и фреймворки
2. Настройка и сборка проекта
3. Основы React
4. Состояния и эффекты
5. Библиотеки компонент. Работа в изоляции. Storybook

# Роутинг и авторизация

# 08

3 недели,  
0,5 проекта

Реализуете глобальное состояние, роутинг и авторизацию.

Научитесь ускорять работу React-приложений, проводить  
их тестирование, выводить их в продакшен и релизить  
на собственном удалённом сервере.

## Спринт 11: Роутинг и авторизация

Темы

1. Продвинутый React
2. Глобальное состояние
3. Библиотека Redux
4. React Router
5. Механизм авторизации

3 недели,  
0,5 проекта

Покроете своё приложение юнит-тестами, а также интеграционными тестами с помощью библиотек Jest и Cypress.

Научитесь ускорять работу React-приложений, проводить их тестирование, выводить их в продакшен и релизить на собственном удалённом сервере.

## Спринт 12: Тестирование React-приложений

Темы

1. Оптимизация (ускорение работы приложения)
2. Тестирование React-приложений: Jest, Cypress
3. Подготовка проекта к продакшену

### Каникулы

1 неделя

Время отдохнуть  
или доработать проекты

## Спринт 13: Проектный месяц

1 месяц,  
1 проект

Работа над крупным проектом. В команде с менеджерами проектов, тимлидами, системными аналитиками, дизайнерами и фулстек-разработчиками подготовите фронтенд приложения.

### Каникулы

1 неделя

Время отдохнуть  
или доработать проекты

# Основы Node.js, Express и MongoDB 10

---

3 недели,  
1 проект

Разработаете бэкенд для ранее созданного фронтенда интернет-магазина. Реализуете его на TypeScript с использованием Express и хранением данных в MongoDB.

Познакомитесь со средой выполнения серверного кода на JavaScript. Освоите сторонние библиотеки и научитесь добавлять их в проект с помощью менеджера NPM. Научитесь работать с фреймворком Express. Воспользуетесь гибкой производительной noSQL базой данных MongoDB.

## Спринт 14: Основы Node.js, Express и MongoDB

---

Темы

1. Введение в бэкенд-разработку
  2. Node.js. Экосистема NPM
  3. Основы Express
  4. Обработка пользовательского ввода
  5. Принципы построения API REST
  6. Введение в базы данных. MongoDB
  7. Работа с БД и авторизация. JWT
-

[5 недель],  
[0,5 проекта]

Разработаете бэкенд с модульным API и выделенным слоем данных для одновременной поддержки двух баз данных — MongoDB и PostgreSQL. А ещё вы сможете сами выбрать, с каким фреймворком работать — Express или Nest.js.

Освоите фреймворк Nest.js: основные компоненты, авторизация и логирование, концепции внедрения зависимостей. Познакомитесь с реляционными базами данных и языком SQL. Подключите базу данных PostgreSQL к проекту с использованием библиотеки TypeORM.

## Спринт 15: Разработка бэкенда на Nest.js

---

|      |  |
|------|--|
| Темы | <ol style="list-style-type: none"><li>1. Введение в разработку бэкенда на Nest.js</li><li>2. Основы разработки бэкенда на Nest.js</li><li>3. Рутинные задачи</li><li>4. Passport.js. Авторизация через соцсети</li></ol> |
|------|--|

---

## Спринт 16: Базы данных и PostgreSQL

---

|      |   |
|------|---|
| Темы | <ol style="list-style-type: none"><li>1. Введение в реляционные БД и SQL</li><li>2. Продолжение работы с реляционными БД и SQL</li><li>3. Связи в реляционных БД</li><li>4. PostgreSQL на практике</li><li>5. Продвинутая практика с PostgreSQL</li><li>6. Моделирование данных</li></ol> |
|------|---|

---

3 недели,  
0,5 проекта

Задеплоите веб-сервис на сервере с использованием Docker-контейнеров.

Научитесь создавать удалённый сервер, запускать на нём приложения и автоматизировать доставку изменений кода. Создадите собственный Docker-контейнер, опубликуете его в хранилище Docker Hub и запустите на вашем сервере. Познакомитесь с Docker Compose, который позволяет управлять группой контейнеров как единым приложением.

## Спринт 17: Тестирование бэкенда, деплой и автоматизация

Темы

1. Тестирование бэкенда
  2. Linux на практике
  3. Деплой приложения на сервер
  4. Введение в мир контейнеров и Docker
  5. Погружение в мир контейнеров и Docker
  6. Хранение и распространение Docker-образов. DockerHub
  7. Запуск приложения из нескольких контейнеров
-

# Обеспечение безопасности приложения

13

3 недели,  
1 проект

Поработаете с рефакторингом готового кода, у которого отсутствует валидация данных и есть ошибки безопасности.

Узнаете основные виды угроз для веб-приложений: SQL-инъекции, XSS, CSRF. Освоите инструменты и технологии, которые используются для обеспечения безопасности приложений. Разберёте основы сетевых протоколов HTTPS и SSL/TLS. Поймёте, как безопасно работать с базами данных.

## Спринт 18: Обеспечение безопасности приложения

Темы

1. Основы безопасности веб-приложений
2. Межсайтовый скриптинг и инъекции
3. Безопасность авторизации и межсайтовая подделка запросов
4. Защита инфраструктуры

### Каникулы

1 неделя

Время отдохнуть  
или доработать проекты

## Спринт 19: Проектный месяц

2 месяца,  
1 проект

Работа над крупным проектом. В команде с менеджерами проектов, тимлидами, системными аналитиками, дизайнерами и фулстек-разработчиками подготовите бэкенд приложения.

6 недель,  
1 проект

Познакомьтесь с базовым синтаксисом Python и сравните его с JavaScript. Настройте своё локальное рабочее место и среду разработки под Python.

Узнаете, как работать с классами на языке Python. Сравните с изученным в TypeScript. Выполните интересные практические задания и закрепите знания объектно-ориентированного программирования на итоговом проекте.

## Спринт 20. Основы Python. Настройка рабочего места для работы с Python

|      |   |
|------|---|
| Темы | <ol style="list-style-type: none"><li>1. Знакомство с Python</li><li>2. Простые типы данных</li><li>3. Ветвления</li><li>4. Коллекции</li><li>5. Импортируемые типы данных</li><li>6. Итерации и циклы</li><li>7. Настройка рабочего окружения</li><li>8. Контекстные менеджеры и работа с файлами</li><li>9. Система контроля и управления версий</li><li>10. Требования к коду</li><li>11. Отладка программ</li></ol> |
|------|---|

## Спринт 21. Объектно-ориентированное программирование в Python

|      |   |
|------|---|
| Темы | <ol style="list-style-type: none"><li>1. Объекты и классы</li><li>2. Принципы ООП в Python</li><li>3. Продвинутое техники</li><li>4. Аннотации и типизация</li><li>5. Паттерны (Python-way)</li></ol> |
|------|---|

### Каникулы

1 неделя

Время отдохнуть  
или доработать проекты

7 недель,  
1 проект

Изучите работу с самыми популярными и мощными фреймворками Django, Flask и FastAPI на языке Python. Познакомьтесь с принципами асинхронного программирования на Python и научитесь писать код, который работает быстрее и эффективнее, обрабатывая множество запросов одновременно.

## Спринт 22. Знакомство с Django. Django Rest Framework

|      |   |
|------|---|
| Темы | <ol style="list-style-type: none"><li>1. Бэкенд на Django</li><li>2. Пути и view-функции</li><li>3. HTML и шаблоны Django</li><li>4. Django ORM</li><li>5. Админ-зона Django</li><li>6. Django ORM. получение информации из БД</li><li>7. Django REST Framework</li></ol> |
|------|---|

## Спринт 23. Асинхронный Python. Flask

|      |  |
|------|--|
| Темы | <ol style="list-style-type: none"><li>1. Конкурентность, параллельность и асинхронность</li><li>2. Основы Flask</li><li>3. API на Flask</li><li>4. Асинхронный Flask</li></ol> |
|------|--|

## Спринт 24. FastAPI: быстрый старт, быстрый API

|      |   |
|------|---|
| Темы | <ol style="list-style-type: none"><li>1. Знакомство с FastAPI</li><li>2. База данных в FastAPI</li><li>3. Миграции: библиотека Alembic</li><li>4. CRUD в FastAPI</li><li>5. Модель бронирования переговоров</li></ol> |
|------|---|

2 недели

Вы освоите полный цикл развёртывания современных Python-приложений: от написания кода — до настройки серверов и контейнеров для надёжной работы в продакшене. Познакомитесь с NGINX — самым популярным веб-сервером, научитесь использовать Gunicorn и Uvicorn как продвинутое приложения-серверы для Python, вспомните, как работать с Docker, и развернёте собственное приложение в изолированной, защищённой среде.

## Спринт 25. NGINX+GUNICORN/UVICORN+Python+Docker

Темы

1. Погружение в задачу
2. Работа с сервером
3. Docker: приложение в коробке
4. Автоматизация. Деплой на сервер.

### Каникулы

1 неделя

Время отдохнуть  
или доработать проекты

# Параллельные модули

---

60 часов

## Софт-скилы для разработчиков

17

«Мягкие» навыки важны так же, как и «жёсткие». Поэтому параллельно с основной программой вы будете проходить софтовый трек — «курс в курсе», где вы освоите инструменты, которые помогут вам в учёбе и в будущей работе. Вы узнаете, как организовать учебный процесс, адаптироваться в новой среде и организовать командную работу. Будет и теория, и учебные задания, которые помогут вам сделать и обучение хардовым навыкам, и пребывание на будущей работе легче и осознаннее.

## Содержание

---

|      |  |
|------|--|
| Темы | <ol style="list-style-type: none"><li>1. Рефлексия</li><li>2. Целеполагание</li><li>3. Планирование и реализация</li><li>4. Работа в команде</li><li>5. Стрессоустойчивость</li><li>6. Получение и предоставление обратной связи</li></ol> |
|------|--|

---

