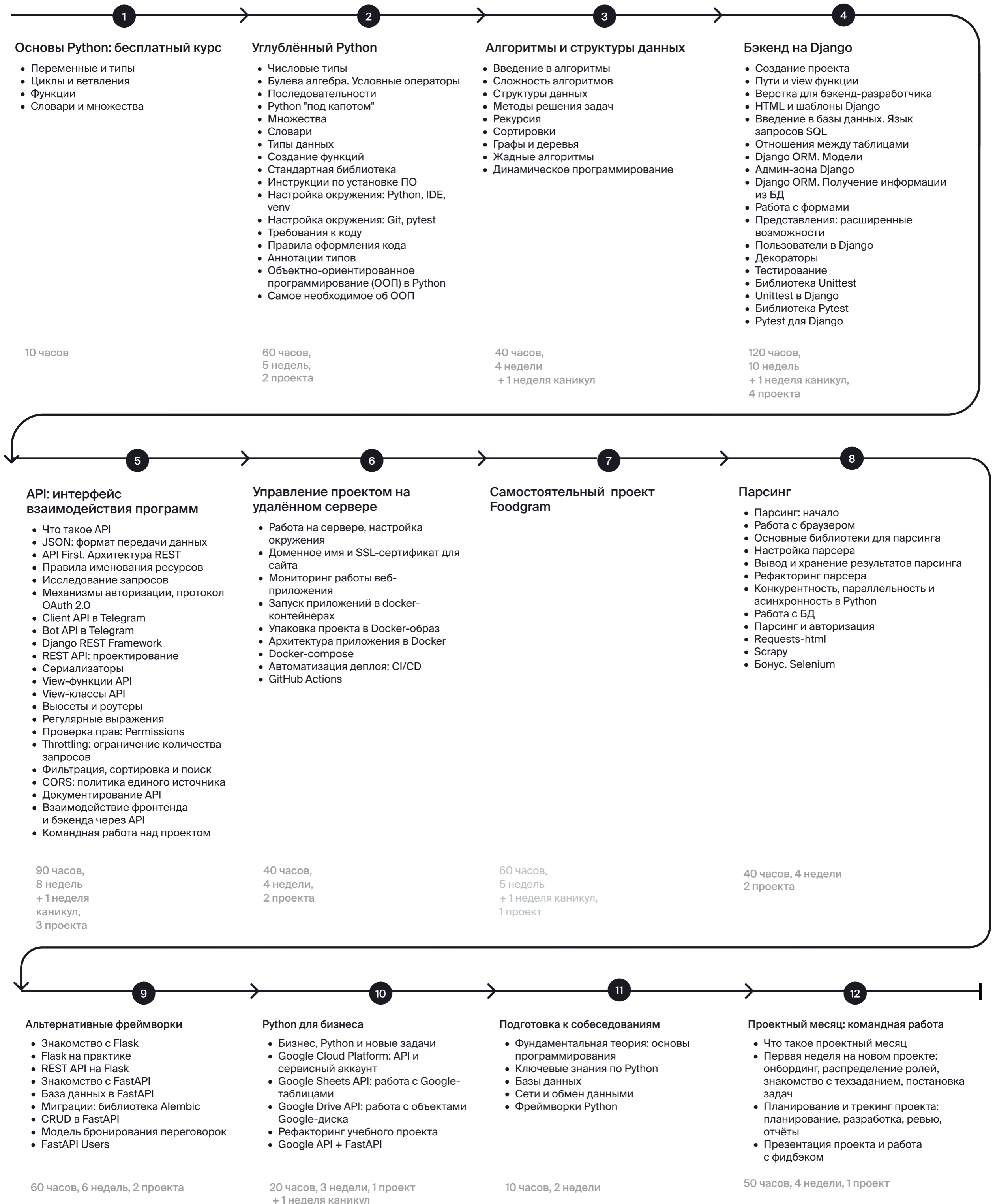


# Python-разработчик плюс



# Модуль 1 Основы Python: бесплатный курс

10 часов

Во вводном курсе вы почувствуете себя в роли бэкенд-разработчика на Python. Это поможет вам понять, хотите ли вы развиваться в этом направлении. Вы узнаете, что такое бэкенд и какие задачи он решает, разберётесь, как программы взаимодействуют с пользователями и с другими программами.

В каждом уроке после небольшой теоретической части вы будете писать код на Python — с первого же урока. Работать вы будете в онлайн-тренажёре — нашей интерактивной среде.

Этот курс докажет главное: вы самостоятельно можете написать программный код, который будет работать и делать то, что задумано.

## Темы курса

- Знакомство с Python
- Циклы и ветвления
- Функции
- Словари и множества

# Модуль 2 Углублённый Python

60 часов,  
5 недель,  
2 проекта

**Языки:** Python 3.9

**Инструменты:** Git, GitHub, редактор кода Visual Studio Code, Flake8, Pytest, bash, виртуальное окружение

**Знания:** правила оформления кода, работа в виртуальном окружении, основы ООП, продвинутые возможности Python.

Вы детально изучите «грамматику» Python, научитесь применять встроенные инструменты и модули этого языка, познакомитесь с типами данных; выясните, как Python работает с оперативной памятью и узнаете, почему это знание важно на практике; изучите требования к коду и правила подготовки проектов.

В этом же блоке вы познакомитесь с основами объектно-ориентированного программирования, установите и настроите редактор кода, узнаете, как работать с системой версионирования кода.

В уроках вас ждут практические задания, а в конце каждого спринта — самостоятельный проект.

Финальный проект блока «Углублённый Python» — самостоятельное создание игры «Изгиб Питона».

Эта работа будет проверена ревьюерами.

## Спринт 1

- Числовые типы
- Булева алгебра. Условные операторы
- Последовательности
- Python "под капотом"
- Множества
- Словари
- Типы данных
- Создание функций
- Стандартная библиотека

**Проект:**  
приложение «Холодильник»

Система регистрации  
и отслеживания продуктов  
в холодильнике.

## Спринт 2

- Инструкции по установке ПО
- Настройка окружения: Python, IDE, venv
- Настройка окружения: Git, pytest
- Требования к коду
- Правила оформления кода
- Аннотации типов

**Проект:** игра «Изгиб Питона»

Реализация классической игры  
«Змейка» на Python.

## Спринт 3

- Объектно-ориентированное программирование (ООП) в Python
- Самое необходимое об ООП

# Модуль 3 Алгоритмы и структуры данных

40 часов,  
4 недели  
+ 1 неделя  
каникул

**Языки:** Python 3.9

**Инструменты:** Яндекс Контест

**Знания:** понимание применения алгоритмов при решении любых задач, поиск и реализация разных вариантов решения одних и тех же задач, оценка вариантов решения, понимание основных структур данных и применение их для решения конкретных задач.

Любую задачу можно решить медленно и неэффективно, а можно — быстро и экономно. Неэффективное программное решение может впустую занять все вычислительные ресурсы вычислительного устройства и замедлить его работу, а то и вовсе сломать его. Изучение алгоритмов поможет избежать подобных ошибок и ускорить работу программ.

## Спринт 1

- Введение в алгоритмы
- Сложность алгоритмов
- Структуры данных
- Методы решения задач

**Финальное задание спринта:**  
Задача «Служба доставки»

## Спринт 2

- Рекурсия
- Сортировки
- Графы и деревья
- Жадные алгоритмы
- Динамическое программирование

**Финальное задание спринта:**  
Шифрованные инструкции

Каникулы —————> 1 неделя

# Модуль 4 Бэкенд на Django

120 часов,  
10 недель  
+ 1 неделя  
каникул,  
4 проекта

**Языки:** Python 3.9, HTML, CSS

**Инструменты:** Django Web Framework, Bootstrap, Unittest, Pythonanywhere, Pytest, SQLite

**Знания:** основы HTML и вёрстки для бэкендера, создание основы проекта и добавление новых приложений в Django, применение MVC на практике, использование шаблонизатора Django, работа с Django ORM, тестирование проекта, деплой проекта в облако.

Главная тема курса — работа с веб-фреймворком Django, это программный «конструктор» для создания веб-проектов. Вы создадите полноценное веб-приложение — это будет платформа для блогов. Вы шаг за шагом создадите полноценный портал — с авторизацией, персональными лентами и комментариями. В процессе работы вы научитесь обрабатывать HTTP-запросы, подключите к приложению базу данных, создадите веб-страницы проекта. Любой код нуждается в проверке — вы научитесь писать автоматические тесты для своих программ. К концу курса вы допишете проект — и он станет ещё одной выполненной работой в вашем портфолио.

## Спринт 1

- Создание проекта
- Пути и view функции
- Верстка для бэкенд-разработчика
- HTML и шаблоны Django

**Проект: Блогикум. Часть 1.**  
Развёртывание и настройка веб-приложения, которое станет платформой для блогов.

## Спринт 2

- Введение в базы данных. Язык запросов SQL
- Отношения между таблицами

## Спринт 3

- Django ORM. Модели
- Админ-зона Django
- Django ORM. Получение информации из БД

**Проект: Блогикум. Часть 2.**  
Настройка панели администратора и работа с публикациями

## Спринт 4

- Работа с формами
- Представления: расширенные возможности

## Спринт 5

- Пользователи в Django

**Проект: Блогикум. Часть 3.**  
Подключение возможности самостоятельной регистрации пользователей, публикации постов через форму, а также их комментирование.

## Спринт 6

- Тестирование
- Библиотека Unittest
- Unittest в Django
- Библиотека Pytest
- Pytest для Django

**Проект: Vice Versa**  
Тестирование проектов YaNote и YaNews на unittest и pytest



# Модуль 5 API: интерфейс взаимодействия программ

90 часов,  
8 недель  
+ 1 неделя  
каникул,  
3 проекта

**Языки:** Python 3.9, RegExр

**Инструменты:** Django REST Framework, React, Telegram API, JWT, Postman, requests

**Знания:** настройка взаимодействия Python-приложения с внешними API-сервисами, создание собственного API-сервиса на базе проекта Django, создание Telegram-ботов, подключение SPA к бэкенду на Django через API, основы групповой разработки ПО.

В этой теме вы разберётесь, как веб-проекты взаимодействуют друг с другом, запрашивая и передавая информацию: например, каким образом ваша программа на Python может общаться с серверами Telegram для управления ботом, а тот в свою очередь может получать информацию от других сервисов в интернете. Вы узнаете, что такое API, создадите свой собственный REST API сервис, научитесь настраивать взаимодействие между фронтендом и бэкендом, а также напишете бота, который взаимодействует с популярными веб-сервисами. Кроме того, вы узнаете, как может быть организована авторизация в разных проектах.

## Спринт 1

- Что такое API
- JSON: формат передачи данных
- API First. Архитектура REST
- Правила именования ресурсов
- Исследование запросов
- Механизмы авторизации, протокол OAuth 2.0
- Client API в Telegram
- Bot API в Telegram

### Проект: Бот-ассистент

Создание Telegram-бота, оповещающего о статусе домашнего задания

## Спринт 2

- Django REST Framework
- REST API: проектирование
- Сериализаторы
- View-функции API
- View-классы API
- Вьюсеты и роутеры
- Регулярные выражения

### Проект: YouTube. Часть 1.

CRUD API для сервиса YouTube

## Спринт 3

- Проверка прав: Permissions
- Throttling: ограничение количества запросов
- Фильтрация, сортировка и поиск
- CORS: политика единого источника
- Документирование API
- Взаимодействие фронтенда и бэкенда через API

### Проект: YouTube. Часть 2.

Полноценный API для сервиса YouTube. Создание REST API

## Спринт 4

- Командная работа над проектом

### Итоговый проект модуля: YaMDB

Создание REST API для сервиса YaMDB — базы отзывов о фильмах, книгах и музыке. Проект выполняется в команде с другими студентами

## Модуль 6 Управление проектом на удалённом сервере

40 часов,  
4 недели,  
2 проекта

**Языки:** Python 3.9, YAML

**Инструменты:** Docker, Nginx, PostgreSQL, Gunicorn, GitHub Actions

**Знания:** создание образов и запуск контейнеров Docker, создание и деплой мультиконтейнерных приложений, основы DevOps, CI&CD.

Работа с облачными сервисами, настройка серверов, тестирование и обновление кода — всё это рутинные операции, с которыми сталкивается любой программист. Вы самостоятельно настроите сервер в облаке, узнаете, как автоматизировать процессы тестирования и обновления кода, развернёте свой проект в docker-контейнерах на сервере.

### Спринт 1

- Работа на сервере, настройка окружения
- Доменное имя и SSL-сертификат для сайта
- Мониторинг работы веб-приложения

Проект: Kittygram. Часть 1.  
Классический деплой проекта на удалённый сервер.

### Спринт 2

- Запуск приложений в docker-контейнерах
- Упаковка проекта в Docker-образ
- Архитектура приложения в Docker
- Docker-compose
- Автоматизация деплоя: CI/CD
- GitHub Actions

Проект: Kittygram. Часть 2.  
Деплой проекта в контейнерах, автоматизация деплоя с применением CI/CD.

## Модуль 7 Самостоятельный проект Foodgram

60 часов,  
5 недель  
+ 1 неделя каникул,  
1 проект

Вы самостоятельно напишете веб-приложение Foodgram: сайт, на котором пользователи будут публиковать собственные рецепты, добавлять чужие рецепты в избранное и подписываться на публикации других авторов. Сервис «Список покупок» позволит пользователям создавать список продуктов, которые нужно купить для приготовления выбранных блюд.

Каникулы —————> 1 неделя

## Модуль 8 Парсинг

40 часов,  
4 недели,  
2 проекта

**Языки:** Python 3.9

**Инструменты:** requests, requests-cache, requests-html, bs4, re, tqdm, argparse, prettytable, csv, SQLAlchemy, Scrapy, Selenium, threading, multiprocessing, AsyncIO

**Знания:** парсинг информации с помощью библиотеки Beautiful Soup, парсинг страницы с необходимостью авторизации и динамическим содержимым, асинхронный код и использование Scrapy для создания асинхронных парсеров. использование Scrapy для создания асинхронных парсеров.

Вы познакомитесь с парсингом — искусством собирать информацию с сайтов, структурировать её и выводить в удобном формате. Узнаете, что такое «инструменты разработчика в браузере», научитесь исследовать HTML-разметку сайта и поработаете регулярными выражениями. Первое самостоятельное задание курса по парсингу — собрать данные с сайта, структурировать их и сохранить в удобном для обработки формате. Ещё одна тема курса — фреймворк Scrapy, инструмент для создания асинхронных парсеров.

### Спринт 1

- Парсинг: начало
- Работа с браузером
- Основные библиотеки для парсинга
- Настройка парсера
- Вывод и хранение результатов парсинга
- Рефакторинг парсера

#### Проект: Парсинг PEP

Сбор информации о статусах и типах документов в PEP (Python Enhancement Proposals).

### Спринт 2

- Конкурентность, параллельность и асинхронность в Python
- Работа с БД
- Парсинг и авторизация
- Requests-html
- Scrapy
- Бонус. Selenium

#### Проект: Асинхронный парсер PEP

Этот парсер будет собирать и сохранять данные о документах PEP

## Модуль 9 Альтернативные фреймворки

60 часов  
6 недели  
2 проекта

**Языки:** Python 3.9

**Инструменты:** Flask, SQLite, Flask-SQLAlchemy, Flask-WTF, Flask-Migrate, FastAPI, Pydantic, Uvicorn, SQLAlchemy, Alembic, fastapi-users

**Знания:** создание типичного приложения на Flask и FastAPI, основы работы с асинхронным фреймворком.

На основе фреймворка Flask можно реализовать практически любой проект: от простого одностраничного сайта до серьёзного проекта с авторизацией, аутентификацией и другими возможностями. Flask подходит для задач, которые подразумевают гибкость в выборе компонентов. Разработчик сам принимает решение, что ему пригодится в работе. Фреймворк FastAPI предназначен для быстрой (от англ. fast) разработки API-приложений и основан на использовании стандартной аннотации типов Python. В этом спринте вы реализуете два проекта на FastAPI, они дадут вам представление о том, как работает этот фреймворк.

### Спринт 1

- Знакомство с Flask
- Flask на практике
- REST API на Flask

#### Проект: Сервис YaCut

Сервис укорачивания ссылок и API к нему

### Спринт 2

- Знакомство с FastAPI
- База данных в FastAPI
- Миграции: библиотека Alembic
- CRUD в FastAPI
- Модель бронирования переговоров
- FastAPI Users

#### Проект: Приложение QRKot

Проект для Благотворительного фонда поддержки котиков

## Модуль 10 Python для бизнеса

20 часов  
3 недели  
1 проект  
+ 1 неделя  
каникул

**Языки:** Python 3.9

**Инструменты:** SQLite, FastAPI, Pydantic, Uvicorn, SQLAlchemy, Alembic, fastapi-users, aiogoogle

**Знания:** управление сервисами через Google API, интеграция приложения на FastAPI с сервисами Google Sheets API и Google Drive API, разработка системы управления Google-таблицами.

Вы прокачаете навыки работы с документацией и узнаете, как использовать сервисы платформы Google. Это поможет вам понять общий принцип работы подобных сервисов и в будущем использовать этот навык для работы не только с Google, но и с другими внешними сервисами. Это большой фронт работ, на котором вы сможете найти себе применение.

### Спринт 1

- Бизнес, Python и новые задачи
- Google Cloud Platform: API и сервисный аккаунт
- Google Sheets API: работа с Google-таблицами
- Google Drive API: работа с объектами Google-диска
- Рефакторинг учебного проекта
- Google API + FastAPI

Проект: Отчёт в Google Sheets для приложения QRKot

Каникулы —————> 1 неделя

## Модуль 11 Подготовка к собеседованиям

10 часов,  
2 недели

Последний спринт посвящён основным вопросам и темам, которые затрагивают на собеседованиях: вы повторите пройденный материал и дополните его теорией, которой не было в курсе. В конце каждой темы — тесты. В этом спринте вы обзорно пройдёте от фундаментальных основ программирования до конкретных каверзных вопросов, относящихся к python и к веб-разработке.

### Спринт 1

- Фундаментальная теория: основы программирования
- Ключевые знания по Python
- Базы данных
- Сети и обмен данными
- Фреймворки Python

## Модуль 12 Проектный месяц: командная работа

50 часов,  
4 недели,  
1 проект

Знания из дополнительных уроков курса про мягкие навыки наконец-то пригодятся на практике. Вы примете участие в разработке реального проекта, и сделаете работу над ошибками по командному взаимодействию. Всё будет настоящее: сторонний заказчик (некоммерческая организация), техническое задание, команда разработчиков (вы и ваши однокурсники), тимлид, проджект, тестировщик и СТО. Будут ревью, дедлайны, отчёты, релиз — полное погружение в реальность. В течение всей программы вы изучали навыки и осваивали компетенции необходимые разработчикам, а проектный месяц полигон для их испытаний.

### Спринт 1

- Что такое проектный месяц
- Первая неделя на новом проекте: онбординг, распределение ролей, знакомство с техзаданием, постановка задач
- Планирование и трекинг проекта: планирование, разработка, ревью, отчёты
- Презентация проекта и работа с фидбэком

## Вебинары

Вебинары с наставником проводятся, в среднем, 1 раз в 2 недели



# Карьерный трек: Трудоустройство

30 часов,  
4 недели

Эта часть курса для тех, кто решил найти работу в сфере разработки. Карьерный трек организован так же, как и предыдущие темы: теория и практика. Но вместо навыков программирования вы изучите стратегии поиска работы и узнаете о тонкостях выбора компании-работодателя, а в качестве практики вы будете писать не код, а резюме и сопроводительное письмо. В течение курса вы подготовите портфолио к визиту работодателя и проведёте исследование рынка трудоустройства.

## Спринт 1

- Профессия и целеполагание
- Поиск работы
- Рынок труда
- Работа мечты
- Подготовка резюме
- Хорошее портфолио

## Спринт 2

- Неформальный поиск работы
- Сопроводительное письмо
- Коммуникация с HR
- Анализ собственных результатов

## Спринт 3

- Подготовка к собеседованию
- Soft- и hard-навыки
- Деньги
- Собеседование в Яндекс
- Выбор компании
- Офис, удалёнка или фриланс?
- Как не ошибиться при выборе компании
- Испытательный срок

## Акселерация

2–6 месяцев

Групповые и индивидуальные консультации по стратегии поиска работы. Обсуждение вакансий и тестовых заданий. Разбор сложных кейсов.