

Python-разработчик

Базовый тариф

10 месяцев

продолжительность курса

13 проектов

в портфолио

8 НЕДЕЛЬ | 96 ЧАСОВ

01

Программирование на Python

1. Введение в Python
2. Управление логикой и качество кода
3. ООП: основы и проектирование
4. ООП: расширенные возможности

4 НЕДЕЛИ | 48 ЧАСОВ

02

Базы данных и фреймворк Flask

5. Введение в базы данных
6. Введение во Flask

8 НЕДЕЛЬ | 96 ЧАСОВ

03

Фреймворк Django

7. Введение в Django
8. Базы данных и ORM
9. Логика приложения и пользователи
10. Тестирование Django-приложений

8 НЕДЕЛЬ | 96 ЧАСОВ

04

API — интерфейс взаимодействия программ

11. Основы API
12. API на Flask
13. Django REST Framework: базовый API
14. Django REST Framework: продвинутый API

2 НЕДЕЛИ | 24 ЧАСА

05

Вайбкодинг

15. Вайбкодинг

4 НЕДЕЛИ | 96 ЧАСОВ

06

Управление проектом на удалённом сервере

16. Деплой и управление приложением на сервере
17. Контейнеризация и автоматизация деплоя

2 НЕДЕЛИ | 24 ЧАСА

07

Парное программирование

18. Парное программирование

2 НЕДЕЛИ | 24 ЧАСА

08

Асинхронный Python

19. Асинхронный Python и Flask

2 НЕДЕЛИ | 24 ЧАСА



Дипломный проект

Дипломный проект для базовой части

Python-разработчик

Расширенный тариф

14 месяцев

продолжительность курса

19 проектов

в портфолио

8 НЕДЕЛЬ | 96 ЧАСОВ

09

Парсинг данных на Python

- 20. Основы парсинга
- 21. Парсинг сайтов

4 НЕДЕЛИ | 48 ЧАСОВ

10

FastAPI

- 22. FastAPI: основы
- 23. FastAPI: продвинутый проект

4 НЕДЕЛИ | 48 ЧАСОВ



Дипломный проект

Дипломный проект
для расширенной части

4 НЕДЕЛИ | 48 ЧАСОВ

11

Обязательная практика

- 24. Обязательная практика: командный проект

2 НЕДЕЛИ | 24 ЧАСА

12

Работа над pet-проектом

- 25. Pet-проект: от идеи до реализации

Python-разработчик

В Практикуме есть три курса по Python-разработке с нуля:

- базовый 10-месячный курс «Python-разработчик»,
- расширенный 14-месячный «Python-разработчик расширенный»,
- и интенсивный 22-недельный курс «Python-разработчик буткемп».

У каждого курса разная длительность, количество теории и практики, а также стоимость. Ниже — сравнение их программ, чтобы вам было проще сориентироваться, какой из курсов вам больше подходит

Сравнение курсов

	Python-разработчик	Python-разработчик расширенный	Python-разработчик буткемп
Время обучения	10 месяцев	14 месяцев	22 недели
Интенсивность обучения	12 часов в неделю	12 часов в неделю	от 30 часов в неделю
Состав программы	Программирование на Python, API, Управление проектом на удалённом сервере, Вайбкодинг, Парное программирование, Дипломный проект	То же, что и в базовом курсе + Парсинг, Фреймворки Flask и FastAPI, Работа над собственным проектом (pet-проект)	Основы Python, Углублённый Python, API, Алгоритмы и структуры данных, Управление проектом на удалённом сервере, Дипломный проект
Помощь в трудоустройстве	✓	✓	✓
Диплом о дополнительном образовании	✓	✓	✓
Сопровождение: куратор, наставник, ревьюер	✓	✓	✓
Практика работы в команде	✓	✓	✓
Обязательная практика на основе реальной бизнес-задачи	✗	✓	✗

Сравнение курсов Python-разработчик

Проект, основанный на реальной бизнес-задаче

Проект позволяет получить опыт работы в команде и добавить кейс в резюме. Это обязательная часть учебной программы курса «Python-разработчик расширенный»

Почему программа Python-разработчик расширенный длиннее?

В программу «Python-разработчик расширенный» входит три модуля, которых нет в курсе «Python-разработчик»:

- Парсинг
- FastAPI
- Работа над собственным проектом (pet-проект)

Содержание модулей

Модуль «Парсинг»

В теме «Парсинг» вы узнаете, как автоматически собирать данные с сайтов и сохранять полученную информацию в файлы и в базы данных. Поработаете с библиотеками Beautiful Soup, requests-html, изучите ORM SQLAlchemy, узнаете, как парсить данные, если необходима авторизация и познакомитесь с Selenium — инструментом для автоматизации действий браузера.

Модуль «FastAPI»

Этот блок посвящён фреймворку FastAPI. После изучения Django знакомство с ними пойдёт легче: если знаешь один фреймворк — разобраться в другом гораздо легче. Знание этого фреймворка может оказаться плюсом при поиске работы: работодатель сэкономит время и ресурсы на включение новичка в команду. FastAPI — современный, быстрый и асинхронный. Назначение этого фреймворка — веб-приложения и API-сервисы.

Работа над собственным проектом (pet-проект)

Вы сможете поработать над собственным проектом для портфолио, а также получить обратную связь на него. Вы самостоятельно спроектируете и реализуете backend-приложение: выберете идею, спланируете архитектуру и реализуете ключевые функции.

Если pet-проекта пока нет в ваших планах, вы сможете прорешать тестовые задания и лучше подготовиться к собеседованиям.

После изучения этих модулей в вашем арсенале появится больше профессиональных инструментов; это даст вам дополнительные бонусы на собеседованиях и поможет эффективнее решать реальные рабочие задачи.

Сравнение курсов Python-разработчик

В чём особенность программы «Python-разработчик буткемп»?

Программа Python-разработчик буткемп предназначена для тех, кто может себе позволить учиться в интенсивном режиме и тратить на обучение 5–7 часов в сутки (время, сопоставимое с обучением в вузе или с работой фулл-тайм).

Режим жёсткий, но и выгода очевидна: уже через четыре месяца студенты получают знания, достаточные для трудоустройства джун-разработчиком. Тот же объём знаний и навыков можно получить в более щадящем режиме на курсе Python-разработчик, но это потребует девяти месяцев обучения вместо четырёх.

Интенсивность обучения влияет не только на студентов, но и на команду сопровождения: наставники и кураторы в программе Python-разработчик буткемп работают более активно. Например, по сравнению с программой Python-разработчик сокращено время ответов на вопросы студентов.

12 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.

С первого же урока после небольшой теоретической части вы будете писать код на Python. 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.

Содержание модуля

Темы

1. Знакомство с Python
 2. Циклы и ветвления
 3. Функции
 4. Словари и множества
-

96 часов

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

Вы познакомитесь с основами объектно-ориентированного программирования, установите и настройте редактор кода, узнаете, как работать с системой версионирования кода.

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

Языки

Python 3.14

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.

Содержание модуля

01. Введение в Python	<ol style="list-style-type: none">1. Установка Python и VS Code.2. Настройка рабочего окружения3. Знакомство с Python4. Модули в python5. Ввод/вывод (input/print)6. Простые типы данных7. Система контроля и управления версиями	Проект Персональный цифровой фитнес-трекер
02. Управление логикой и качество кода	<ol style="list-style-type: none">1. Условные операторы: ветвления2. Импортируемые типы данных3. Коллекции, словари и множества4. Итерации и циклы5. Функции 2.06. Отладка программ7. Требования к коду	Проект: приложение «Холодильник» Система регистрации и отслеживания продуктов в холодильнике.
03. ООП: основы и проектирование	<ol style="list-style-type: none">1. Объекты и классы2. Знакомство с ООП3. Документирование кода4. Расширенные возможности Python5. Система контроля и управления версиями: продолжение	Проект Проект: система управления транспортными средствами

Содержание модуля

04. ООП: расширенные возможности

1. Знакомство с Pygame
2. Работа с файлами
3. Аннотации
4. Знакомство с Pytest

Проект: игра «Изгиб Питона»
Реализация классической игры «Змейка» на Python

Неделя каникул



Базы данных и фреймворк Flask

02

48 часов

Вы познакомитесь с реляционными базами данных и научитесь работать с SQL. Затем создадите первое веб-приложение на Flask: разберётесь с маршрутами, шаблонами и взаимодействием с базой данных через ORM.

Языки

Python 3.14

Инструменты

SQL, SQLite, Flask

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.

Содержание модуля

05. Введение в базы данных

1. Работа с базами данных
2. Типы данных и их преобразования
3. Фильтрация и агрегация
4. Группировка и сортировка
5. Категоризация значений
6. Работа с таблицами
7. Работа с датой и временем

Финальное задание спринта
Проект: подключение баз данных

06. Введение во Flask

1. Введение в Flask
2. Работа с базами данных через ORM
3. Шаблоны
4. Маршрутизация и формы

Проект
Flask-приложение с БД



96 часов

Вы освоите архитектуру Django-приложений: от первых маршрутов и шаблонов до полноценной логики с базой данных, пользователями и тестированием. Научитесь разрабатывать и поддерживать функциональные веб-приложения на Django. Реализуете небольшую фичу с помощью готовых ML-инструментов.

Языки

Python 3.14, HTML, CSS

Инструменты

Django Web Framework, Bootstrap, Unittest, Pythonanywhere, Pytest

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.

Содержание модуля

07. Введение в Django	<ol style="list-style-type: none">1. Создание проекта2. Пути и view-функции3. Вёрстка для бэкендера4. HTML и шаблоны Django	Проект: Блогикум. Часть 1. Развёртывание и настройка веб-приложения, которое станет платформой для блогов.
08. Базы данных и ORM	<ol style="list-style-type: none">1. Основы Django ORM2. Миграции и админзона3. Основы запросов4. Сложные запросы и оптимизация	Проект: Блогикум. Часть 2. Настройка панели администратора и работа с публикациями
09. Логика приложения и пользователи	<ol style="list-style-type: none">1. Основы работы с формами2. Продвинутая работа с формами и данными3. Представления на основе классов4. Аутентификация, авторизация и работа с пользователями	Проект: Блогикум. Часть 3. Подключение возможности самостоятельной регистрации пользователей, публикации и комментирования постов, а также их комментирование.
10. Тестирование Django-приложений	<ol style="list-style-type: none">1. Тестирование2. Библиотека Unittest3. Unittest в Django4. Pytest для Django	Проект: Vice Versa Тестирование проектов на Pytest

Неделя каникул



96 часов

Вы научитесь работать с внешними API и создавать собственные REST-сервисы. Освоите HTTP, JSON и принципы REST. Реализуете полноценный backend-сервис на Django REST Framework с авторизацией, правами доступа и фильтрацией.

Языки

Python 3.14

Инструменты

Django REST Framework, React, JWT, Postman, requests

Знания

Настройка взаимодействия Python-приложения с внешними API-сервисами, создание собственного API-сервиса на базе проекта Django, подключение SPA к бэкенду на Django через API, основы групповой разработки ПО.

Содержание модуля

11. Основы API	<ol style="list-style-type: none">1. Что такое API2. JSON: формат передачи данных3. API First. Архитектура REST4. Правила именования ресурсов5. Исследование запросов6. Механизмы авторизации, протокол OAuth 2.0	Проект: Бот-ассистент Создание бота, оповещающего о статусе домашнего задания
12. API на Flask	<ol style="list-style-type: none">1. CRUD-операции во Flask2. Валидация и обработка ошибок3. Внешние интеграции API	Проект CRUD-API на Flask
13. Django REST Framework: базовый API	<ol style="list-style-type: none">1. Django REST Framework2. REST API: проектирование3. Сериализаторы4. View-функции API5. View-классы API6. Вьюсеты и роутеры7. Регулярные выражения	Проект CRUD для Yatube, часть 1
14. Django REST Framework: продвинутый API	<ol style="list-style-type: none">1. Проверка прав: Permissions2. Throttling: ограничение количества запросов3. Фильтрация, сортировка и поиск4. CORS и политика единого источника5. Документирование API6. Взаимодействие фронтенда и бэкенда через API	Проект CRUD для Yatube, часть 2

24 часа
1 проект

Вы научитесь использовать современные инструменты и подходы для ускорения разработки. Примените их на практике.

Содержание модуля

15. Подходы и инструменты вайбкодинга

Практика

Написание SQL-запросов в рамках задачи по подготовке данных

Неделя каникул



Управление проектом на удалённом сервере

06

48 часов

Вы научитесь разворачивать веб-приложения на сервере и обеспечивать их стабильную работу. Освоите классический деплой, работу с Linux, а затем контейнеризацию и автоматический деплой с помощью Docker и CI/CD.

Языки

Python 3.14

Инструменты

Docker, Nginx, PostgreSQL, Gunicorn, GitHub Actions

Знания

Создание образов и запуск контейнеров Docker, создание и деплой мультиконтейнерных приложений, основы DevOps, CI/CD.

Содержание модуля

16. Деплой и управление приложением на сервере

1. Подготовка сервера
2. Настройка окружения
3. Развёртывание Python-приложения
4. Веб-сервер и безопасность

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

Содержание модуля

17. Контейнеризация
и автоматизация деплоя

1. Основы Docker
2. Контейнеризация приложения
3. Оркестрация многоконтейнерных приложений
4. Основы автоматизации (CI/CD) с GitHub Actions
5. Создание пайплайна тестирования и деплоя

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

Парное программирование

07

24 часа

Совместно с другим участником курса вы напишете веб-приложение Foodgram — сайт, на котором пользователи смогут публиковать собственные рецепты, добавлять чужие рецепты в избранное и подписываться на публикации других авторов. А ещё — создавать списки продуктов, которые нужно купить для приготовления выбранных блюд.

Содержание модуля

18. Парное программирование

Неделя каникул



48 часов

Вы познакомитесь с асинхронным программированием на Python. Научитесь использовать его для интеграций с внешними сервисами и решения фоновых задач. Поймёте, как повышать отзывчивость и масштабируемость приложений. Поработаете с Celery/RabbitMQ.

Языки

Python 3.14

Инструменты

GIL, Flask, SQLAlchemy, ORM, SQLite, Dropbox, Jinja2, Celery, RabbitMQ

Знания

С помощью асинхронного подхода создадите сервис, который будет сокращать ссылки.

Содержание модуля

19. Асинхронный Python и Flask

1. Конкурентность, параллельность и асинхронность
2. Асинхронные интеграции: инструменты и паттерны
3. Работа с внешними API во Flask
4. Системы управления задачами: Celery и RabbitMQ

Проект

Сервис сокращения ссылок с асинхронными интеграциями

Дипломный проект для базовой части

24 часа

Разработаете итоговый проект курса, который объединяет ключевые технологии и подходы. Как итог — сформируете проект для портфолио и закрепите основные навыки backend-разработчика.

Расширенный тариф

48 часов

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 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.

Языки

Python 3.14

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.

Содержание модуля

20. Основы парсинга

1. Парсинг: начало
2. Работа с браузером
3. Основные библиотеки для парсинга
4. Настройка парсера
5. Вывод и хранение результатов парсинга
6. Рефакторинг парсера

Проект: Парсинг PEP

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

21. Парсинг сайтов

1. Работа с БД
2. Парсинг и авторизация
3. Requests-HTML
4. Scrapy
5. Бонус: Selenium

Проект: Асинхронный парсер PEP

Этот парсер будет собирать и сохранять данные о документах PEP

Расширенный тариф

48 часов

Фреймворк FastAPI предназначен для быстрой 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.

Языки

Python 3.14

Инструменты

FastAPI, Pydantic, Uvicorn, SQLAlchemy, Alembic, fastapi-users

Знания

Создание приложения на FastAPI.

Содержание модуля

22. FastAPI: основы	<ol style="list-style-type: none">1. Знакомство с FastAPI2. База данных в FastAPI3. Миграции: библиотека Alembic4. CRUD в FastAPI5. Модель бронирования переговоров6. FastAPI Users	Проект: Приложение QRKot Проект для Благотворительного фонда поддержки котиков
23. FastAPI: продвинутый проект	<ol style="list-style-type: none">1. Система аутентификации с FastAPI Users2. Связь данных с пользователем3. Разграничение доступа	Проект: Приложение QRKot Реализация авторизации

Неделя каникул



Проектный месяц: командная работа

11

Расширенный тариф

48 часов

Знания из дополнительных уроков курса про мягкие навыки наконец-то пригодятся на практике. Вы выполните проект, основанный на реальной бизнес-задаче, получите опыт работы в команде и добавите кейс в резюме.

Будут ревью, дедлайны, отчёты, релиз — полное погружение в реальность. В течение всей программы вы изучали навыки и осваивали компетенции, необходимые разработчикам, а проектный месяц позволит испытать их в реальности.

Вебинары

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

Содержание модуля

24. Обязательная практика:
командный проект

1. Проектный месяц
2. Финальная тема

