

Python для разработчика: учимся писать профессиональный код (PYTHON_PRO)

ID PG-PYTHON_PRO Цена 98 500,- руб. Длительность 5 дней

Кому следует посетить

Для разработчиков, которые уверенно владеют базовым Python и хотят прокачать навыки: писать более оптимизированный код, работать с большими наборами данных, тестировать проекты и использовать асинхронность.

Предварительные требования

Программирование на Python в рамках курса [Программирование на языке Python. Базовый уровень \(PYTHON_BASIC\)](#)

Цели курса

Что даёт курс

- Глубокое понимание устройства Python и его возможностей.
- Владение современными инструментами: аннотации типов, документация, тестирование.
- Навык работы с большими объёмами данных и оптимизацией памяти.
- Понимание кеширования и хеширования для ускорения приложений.
- Знание, как писать асинхронные и многопроцессные программы.

Итог курса

- Участники владеют современными практиками Python-разработки.
- Умеют писать читаемый код с аннотациями и документацией.
- Освоили тестирование и могут проверять качество проектов.
- Знают, как работать с большими данными, использовать кеширование и хеширование.
- Умеют применять асинхронность и многопроцессность в реальных задачах.

Программа курса

Современные инструменты Python

- Аннотации типов и их польза для больших проектов.
- Докстринги и документация по коду.
- Лучшие практики оформления кода.

Функциональный Python и структуры данных

- Декораторы и принципы повторного использования кода.
- Лямбда-функции, map(), filter(), reduce().
- Генераторы списков и словарей.
- Продвинутое сортирование данных
- Стек, очередь и их практическое применение.

Работа с данными и оптимизация

- Обработка больших наборов данных: подходы и ограничения.
- Введение в polars и обработку больших массивов данных
- Использование встроенных и внешних библиотек (например, itertools).
- Кеширование: зачем и где использовать (functools.lru_cache).
- Хеширование: принципы, применение, встроенные структуры (dict, set).

Тестирование и надежность кода

- Основы модульного тестирования: unittest, pytest.
- Организация тестов в проекте.

Асинхронность и многопроцессность

- Синхронные vs асинхронные задачи.
- Основы async/await и asyncio.
- Многопроцессность и multiprocessing.
- Обзор библиотек для асинхронной работы и распределённых вычислений.