

Разработка реального проекта на языке Python. Продвинутый уровень

Код: SCRIPT-007

Длительность: 30 ч.

Описание:

Python - мощный динамический язык программирования, который используется в разнообразных прикладных областях, включая телекоммуникации. Python часто сравнивают с Tcl, Perl, Ruby, Scheme или Java. На курсе вы получите продвинутые навыки программирования на языке Python, приняв участие в разработке полноценного back-end программного продукта. Вы рассмотрите темы взаимодействия с файловой системой, объектно-ориентированного программирования, криптографии, асинхронного программирования, разработки серверного программного обеспечения, взаимодействия с базами данных, многопоточности, пакетной сборки.

Цели:

В рамках курса вы:

- на практике рассмотрите парадигму объектно-ориентированного программирования
- рассмотрите проблему переноса программного кода с Python 2 на Python 3
- исследуете концепцию асинхронного программирования в Python
- исследуете возможности языка Python для работы с базами данных
- напишите многопоточный загрузчик файлов
- соберете свой собственный устанавливаемый Python-пакет
- создадите свой собственный программный продукт

После прохождения курса вы начнете:

- использовать Python для взаимодействия с файловой системой
- грамотно поддерживать код
- писать автоматизированные тесты на базе фреймворка Pytest
- использовать Python для решения криптографических задач
- разрабатывать веб-сервер
- управлять доступом к ресурсам своего приложения

Разбираемые темы:

1. Вступление. Краткое введение в курс (теория + практика).

1. Теория
 - Основные особенности языка Python
 - Системы контроля версий. Git

- Модель Gitflow
- Виртуальные окружения
- 2. Практика
 - Развертывание виртуальных окружений

2. Файловый сервер. Описание проекта (теория).

1. Теория
 - Описание проекта и требования к нему
 - Архитектура проекта

3. Первый прототип. Работа с файловой системой (теория + практика).

1. Теория
 - Работа с файловой системой, контекстный менеджер
 - Модули и импорты
 - Переменные окружения
 - Архитектура первого прототипа
2. Практика
 - Разработка первого прототипа файлового сервера

4. Тестирование и сопровождение (теория + практика).

1. Теория
 - Основные принципы сопровождения кода на Python
 - Стандарты оформления и документирования кода
 - Тестирование кода. Фреймворк Pytest
 - Логирование
2. Практика
 - Поддержка Docstrings
 - Разработка тестов с использованием фреймворка Pytest
 - Добавление логирования

5. Защита ресурсов. Криптография. Объектно-ориентированное программирование (теория + практика).

1. Теория
 - Объектно-ориентированное программирование
 - Статический класс, singleton, метаклассы
 - Инкапсуляция в Python
 - Криптография в Python, цифровые подписи
 - Хэш-функции
 - Симметричное и асимметричное шифрование, гибридное шифрование
2. Практика
 - Внедрение ООП в проект
 - Реализация статических классов и singleton
 - Разработка функционала цифровых подписей файлов
 - Шифрование файлов

6. Перенос кода с Python 2 на Python 3 (теория + практика).

1. Теория
 - Различия между Python 2 и 3
 - Пакет future, алгоритм переноса кода с Python 2 на Python 3
2. Практика
 - Перенос кода с Python 2 на Python 3

7. Разработка асинхронного веб-сервера (теория + практика).

1. Теория
 - GIL
 - Асинхронное программирование в Python, asyncio
 - Корутины и awaitable-объекты, event loop
 - Сервер aiohttp
2. Практика
 - Написание собственных корутин
 - Разработка веб-сервера aiohttp
 - Тестирование разработанного веб-сервера

8. Авторизация. Аутентификация. Работа с базой данных (теория + практика).

1. Теория
 - Структура базы данных
 - СУБД PostgreSQL
 - Взаимодействие с базой данных в Python
 - Построение SQL-запросов в Python
2. Практика
 - Поддержка учетных записей пользователей
 - Аутентификация пользователей и авторизация запросов
 - Взаимодействие приложения с базой данных

9. Сборка Python-пакетов (теория + практика).

1. Теория
 - Пакеты Python, структура пакета
 - Сборка и установка пакета
2. Практика
 - Сборка и установка собственного Python пакета/li>

ДОПОЛНИТЕЛЬНАЯ ЧАСТЬ (при должном уровне подготовки аудитории)

10. Реализация ролевой модели доступа (теория + практика).

1. Теория
 - Ролевая модель доступа
 - ORM, SQLAlchemy
2. Практика
 - Реализация ролевой модели ограничения доступа
 - Внедрение ORM

11. Многопоточный загрузчик файлов (теория + практика).

1. Теория
 - Потоки, демоны, очереди
2. Практика
 - Разработка многопоточного загрузчика файлов

Целевая аудитория:

Курс предназначен для разработчиков, системных администраторов и инженеров QA, которые хотят научиться разрабатывать, автоматизировать и тестировать приложения и системы, используя один из самых мощных языков программирования, доступных сегодня.

Участник тренинга должен в обязательном порядке пройти тренинги [SCRIPT-002](#) и [SCRIPT-003](#). Необходимо знать следующие базовые темы:

- Структуры данных (списки, кортежи, словари)
- Функции и работу с параметрами
- Основы объектно-ориентированного подхода, включая наследование, инкапсуляцию и полиморфизм. Понятие исключения и механизм обработки исключений
- Лямбда-функции, замыкание
- Декораторы
- Магические методы `__init__()`, `__new__()`,
- Понятие итератора
- Оператор `yield`

Предварительная подготовка - общее:

- Средние общие знания в области программирования. Средние знания скриптовых языков. Понимание принципов парадигмы объектно-ориентированного программирования
- Базовые знания реляционных баз данных
- Понимание принципа работы протокола HTTP
- Прохождение курсов [SCRIPT-002](#), [SCRIPT-003](#)
- Способность понимать логические потоки кода.

Рекомендуемые дополнительные материалы, источники:

<https://docs.python.org/3/tutorial/index.html>
<https://docs.python.org/2/library/os.html> <https://docs.pytest.org/en/latest/>
<https://docs.python.org/3/library/hashlib.html>
<https://pycryptodome.readthedocs.io/en/latest/>
<https://python-future.org/index.html>
<https://www.python.org/dev/peps/pep-3156/>
<https://www.python.org/dev/peps/pep-0492/>
<https://aiohttp.readthedocs.io/en/stable/web.html>



<https://docs.sqlalchemy.org/en/13/>

<https://docs.python.org/3/library/threading.html>

<https://klen.github.io/create-python-packages.html>