

Бауманский учебный центр «Специалист»

Программа курса Python для DevOps и сетевых инженеров

Код курса: ПИТДЕВ

Тема	Ак. часов
Модуль 1. Основы Python для инфраструктурных задач <ul style="list-style-type: none">• Преимущества автоматизации: примеры из реальной практики.• Настройка среды: интерпретатор Python, менеджер пакетов, изолированные окружения.• Практика: Рабочее пространство. <i>Базовая настройка инструментов разработки.</i>• Практика: Первая программа. <i>Генерация сетевых диапазонов.</i>• Практика: Разработка утилиты для перечисления всех адресов в подсети. <i>Создание инструмента анализа подсетей.</i>• Введение в Docker. Базовые команды, использование docker-compose.• Практика: Взаимодействие Python с СУБД Postgres. <i>Интеграция с базой данных.</i>• Практика: VS Code с плагинами для Python, Docker, YAML. <i>Оптимизация среды разработки.</i>	4
Модуль 2. Структуры данных для работы с сетевыми объектами <ul style="list-style-type: none">• Числовые операции: конвертация IP-адресов между форматами.• Практика: Текстовая обработка. <i>Анализ выводов CLI.</i>• Коллекции: организация информации об интерфейсах и их состояниях.• Практика: Условные конструкции. <i>Фильтрация оборудования.</i>• Практика: Создание парсера для преобразования вывода "show interfaces" в структурированный формат. <i>Преобразование неструктурированных данных.</i>• Словари для облачных сетей: организация данных о VPC, security groups, load balancers.• Списки подсетей в облаке: анализ IP-пула, проверка пересечений CIDR-блоков.• Практика: преобразование облачных network policies в словарь Python для анализа. <i>Анализ политик безопасности.</i>• Словари для метрик: хранение данных о latency, packet loss, bandwidth.• Практика: Модуль os и pathlib. <i>Работа с файловой системой.</i>• Практика: Модуль subprocess для работы с Docker. <i>Автоматизация Docker.</i>• Практика: Обработка ошибок. <i>Создание устойчивых скриптов.</i>• Практика: Утилита для проверки доступности сервиса по URL с обработкой таймаутов и ошибок. <i>Мониторинг сервисов.</i>	6
Модуль 3. Алгоритмы для анализа инфраструктуры <ul style="list-style-type: none">• Алгоритмы поиска и фильтрации.• Практика: Линейный поиск. <i>Валидация IP-адресов.</i>• Бинарный поиск: быстрый поиск в отсортированных логах/configs.• Практика: Фильтрация по маске. <i>Селекция объектов.</i>• Сортировка: упорядочивание данных для отчетов (например, по потреблению CPU).• Практика: Поисковые алгоритмы. <i>Определение принадлежности к подсетям.</i>• Упорядочивание: систематизация физических адресов.• Практика: Анализ текста. <i>Парсинг логов.</i>• Практика: Утилита для статистики по сетевым инцидентам. <i>Анализ инцидентов.</i>• Практика: Функция диагностики доступности сетевого оборудования. <i>Проверка оборудования.</i>• Практика: Алгоритмы распределения нагрузки. <i>Имплементация балансировщиков.</i>• Round-robin для балансировки трафика.• Least connections для выбора наименее нагруженного сервера.	6

<p>Модуль 4. WEB API для управления облачной и сетевой инфраструктурой</p> <p>Тема 4.1: Основы HTTP и REST.</p> <ul style="list-style-type: none"> • Методы GET, POST, PUT, DELETE. • Коды ответа. Заголовки. <p>Тема 4.2: Библиотека requests</p> <ul style="list-style-type: none"> • Установка и настройка. • Выполнение запросов GET (получение данных), POST (создание), PUT(обновление). • Практика: Работа с заголовками, параметрами, телом запроса (JSON). <i>Формирование HTTP-запросов.</i> • Практика: Обработка ответов, проверка статус-кода. <i>Анализ API-ответов.</i> <p>Тема 4.3: Аутентификация в API</p> <ul style="list-style-type: none"> • Практика: Токены (Bearer Token). <i>Имплементация аутентификации.</i> • Basic Auth. • JWT Tokens. • Практика: Программа для получения статистики работающих интерфейсов через API. <i>Сбор метрик через API.</i> 	<p>6</p>
<p>Модуль 5. Объектно-ориентированный подход к управлению оборудованием</p> <ul style="list-style-type: none"> • Практика: Создание классов. <i>Проектирование модели сетевого узла.</i> • Практика: Функциональность объектов. <i>Методы подключения и опроса.</i> • Иерархия классов: специализированные типы устройств. • Практика: Защита данных. <i>Безопасное хранение аутентификационной информации.</i> • Практика: Реализация класса маршрутизатора. <i>Создание объекта сетевого устройства.</i> 	<p>6</p>
<p>Модуль 6. Параллельная обработка сетевых устройств</p> <ul style="list-style-type: none"> • Принципы параллелизма: одновременный опрос оборудования. • Библиотека asyncio: организация конкурентных подключений. • Практика: Асинхронные функции. <i>Параллельная диагностика и сбор данных.</i> • Практика: Управление процессами. <i>Обработка сбоев и таймаутов.</i> • Практика: Одновременная проверка состояния группы устройств. <i>Массовый опрос оборудования.</i> • Практика: Asyncio: параллельные HTTP-запросы к API. <i>Конкурентные API-вызовы.</i> • Практика: Асинхронная проверка доступности списка эндпоинтов сервиса. <i>Массовая проверка сервисов.</i> 	<p>6</p>
<p>Модуль 7. Полезные дополнительные модули для работы</p> <ul style="list-style-type: none"> • Библиотека ipaddress: для продвинутой работы с IP-сетями в облаке. • ruyaml + json: для работы с конфигами инфраструктуры. • pandas: для анализа больших объемов сетевых метрик. • Модуль Docker. Работа с контейнерами и образами. 	<p>6</p>
<p>Аудиторная нагрузка в классе с преподавателем</p>	<p>40 ак.ч. + 40 ак.ч. самостоятельно</p>

По окончании обучения на курсе проводится итоговая аттестация. Аттестация проводится в виде теста на последнем занятии или на основании оценок практических работ, выполняемых во время обучения на курсе.

Для частных лиц
+7 (495) 232-32-16

Для организаций
+7 (495) 780-48-44

E-mail
info@specialist.ru