**ПРОГРАММА ОБУЧЕНИЯ**

**КУРС «ВЕБ-ПРОГРАММИРОВАНИЕ PYTHON/DJANGO»**

**(длительность одного занятия – 3 часа)**

|  |  |  |
| --- | --- | --- |
| **Номер занятия** | **Тема** | **Основные вопросы для изучения** |
| 1 | Вводное занятие по языку программирования Python | 1. Экскурс в историю языка Python, основные сферы применения, особенности языка. 2. Установка интерпретатора Python и среды разработки PyCharm. 3. Написание первой программы на Python. 4. Переменные и объекты. 5. Основные типы данных. 6. Функции print(), input(), type(), dir(), help(). 7. Написание комментариев. |
| 2 | Числа, булевый тип данных, None | 1. Типы данных integer и float. 2. Операции с числами. 3. Операторы в Python. 4. Встроенные математические функции. 5. Знакомство с модулем Math. 6. Типы данных bool и None. 7. Преобразование типов. |
| 3 | Строки | 1. Операции со строками, срезы. 2. Методы строк. 3. Индексация. 4. Форматированный вывод строк. 5. Функции len(), ord(), char(). 6. UNICODE, ASCII. |
| 4 | Списки | 1. Операции со списками, срезы. 2. Методы списков. 3. Индексация. 4. Вложенные списки. 5. Поверхностное и глубокое копирование. 6. Функции zip(), enumerate(). |
| 5 | Кортежи, множества | 1. Операции с кортежами, методы кортежа. 2. Операции с множествами, методы множеств. 3. Тип данных frozenset. 4. Основные отличия между списками, кортежами, множествами, их использование. |
| 6 | Словари | 1. Операции со словарями. 2. Методы словарей. 3. Использование словарей. 4. Хэш-таблица, хэшируемый объект. |
| 7 | Оператор ветвления if, циклы for и while | 1. Оператор ветвления if. 2. Тернарное выражение if/else. 3. Циклы for и while. 4. Операторы break, continue и pass. 5. Конструкция else цикла. 6. Обработка исключений, конструкция try/except/finally |
| 8 | Функции | 1. Объявление и вызов функции. 2. Аргументы и параметры функции, способы передачи аргументов, использование \*args, \*\*kwargs. 3. Оператор return. 4. Вложенные функции. 5. Области видимости. 6. Анонимная функция lambda. 7. Функции map(), reduce(), filter(). |
| 9 |
| 10 | Файлы | 1. Работа с файлами (открытие, чтение, запись в файл). 2. Кодировки. 3. Знакомство с JSON. 4. Менеджеры контекста. |
| 11 | Система контроля версий Git | 1. Использование Git в работе. 2. Установка Git. 3. Основные команды Git. 4. Создание аккаунта на GitHub. |
| 12 |
| 13 | Контрольная работа, решение задач по пройденным темам. | |
| 14 | Объектно-ориентированное программирование (ООП) | 1. Основные принципы ООП: абстракция, инкапсуляция, полиморфизм, наследование. 2. Класс как основной инструмент ООП. 3. Создание класса, атрибуты и методы класса. 4. Создание экземпляров класса. |
| 15 | 1. Конструктор класса метод \_\_init\_\_. 2. Магические методы. 3. Методы экземпляра, методы класса, статические методы. 4. Доступ к атрибутам класса. 5. Геттеры, сеттеры, делиттеры, свойство property |
| 16 | 1. Иерархии наследования. 2. Порядок разрешения методов (MRO). 3. Алгоритм С3-линеаризации. 4. Переопределение методов через super() 5. Перегрузка операторов. 6. Принципы SOLID. |
| 17 | Система управления базами данных (СУБД) | 1. Понятие СУБД, реляционные и нереляционные базы данных. 2. Установка PostgreSQL. 3. Создание базы данных. 4. Транзакции в базу данных. |
| 18 | 1. Таблицы, поля таблиц. 2. Связи между таблицами. 3. Создание/редактирование/удаление таблиц. 4. Добавление данных в таблицы, выборка данных. |
| 19 | Контрольная работа, решение задач по пройденным темам. | |
| 20 | Основы web. Введение в Django | 1. Передача данных по сети. 2. HTTP-протокол. 3. Установка фрэймворка Django. 4. Паттерн MVC(MTV). 5. Создание и настройка проекта. 6. Виртуальное окружение, установка зависимостей приложения. |
| 21 |
| 22 | Модели Django, миграции | 1. Object-Relational Mapping(ORM). 2. Создание моделей, поля моделей. 3. Методы моделей. 4. Связи между моделями. 5. Валидация моделей. 6. Миграции. 7. Подключение PostgreSQL. |
| 23 |
| 24 | Маршрутизация, контроллеры | 1. Обработка URL-запросов. 2. Маршрутизация. 3. Контроллеры-функции. 4. Контроллеры-классы. 5. Миксины. |
| 25 |
| 26 | Шаблоны | 1. Рендеринг шаблонов. 2. Знакомство с HTML, CSS, JavaScript. 3. Вывод объектов в шаблон. 4. Директивы и теги шаблонизатора. 5. Статические файлы. 6. Пагинация |
| 27 | QuerySet | 1. Запросы в базу данных. 2. Запись, чтение, удаление, изменение данных таблицы (CRUD). 3. Доступ к данным связанных таблиц. 4. QuerySet, методы QuerySet. 5. Применение field lookups. 6. Агрегатные функции. 7. Q() и F() объекты. |
| 28 |
| 29 | Формы | 1. Работа с формами. 2. Формы, связанные с моделями. 3. Формы, не связанные с моделями. 4. Поля формы. 5. Валидация формы и полей формы. |
| 30 |
| 31 | Административная панель Django | 1. Управление админкой Django. 2. Создание суперпользователя. 3. Регистрация моделей. 4. Заполнение таблиц базы данных через админку. 5. Класс редактора модели. |
| 32 | Сигналы | 1. Сигналы Django. 2. Отправка электронных писем. |
| 33 | Разграничение доступа | 1. Разграничение доступа. 2. Аутентификация, авторизация, идентификация пользователя. 3. Сессии и cookie. 4. Токены. |
| 34 | Итераторы, генераторы, декораторы | 1. Понятие стека и очереди. 2. Рекурсия. 3. Замыкания. 4. Декораторы. 5. Итераторы, итерируемые объекты. 6. Генераторы, оператор yield. |
| 35 |
| 36 | Контрольная работа | |
| 37 | Django Rest Framework (DRF) | 1. Установка Django Rest Framework. 2. Введение в DRF. 3. Принципы REST. 4. Основы построения веб-API. 5. Формат данных JSON. |
| 38 | Сериализаторы, модуль rest\_framework.generics | 1. Написание сериализаторов. 2. Классы Serializer и ModelSerializer. 3. Описание полей сериализатора. 4. Работа с Postman. 5. Знакомство с модулем rest\_framework.generics. |
| 39 |
| 40 | Celery | 1. Асинхронное программирование. 2. Подключение Celery. 3. Знакомство с Redis и RabbitMQ. 4. Фоновые задачи в Celery. |
| 41 |
| 42 | Docker | 1. Установка и знакомство с Docker. 2. Основы контейниризации. 3. Написание Dockerfile. 4. Основные команды. 5. Запуск веб-приложения на Django в контейнере Docker. 6. Тома. |
| 43 |
| 44 | Docker Compose | 1. Установка и знакомство с Docker Сompose. 2. Основные команды. 3. Написание docker-compose.yml. 4. Запуск многоконтейнерного приложения. |
| 45 | Теоретический экзамен | |
| 46 | Защита проекта | |

\*последовательность прохождения тем может корректироваться преподавателем.