

Создание web–приложений с использованием Python

Версия 1.0.0

Продолжительность курса – 140 пар (70 дней)

Цель курса

Обучить слушателя разработке веб-приложений с использованием Python.

По окончании курса слушатель будет:

- Разбираться в тонкостях построения веб-приложений с использованием Python
- Понимать особенности реализации механизмов ООП в Python
- Обрабатывать и анализировать данные форм
- Использовать стандартные возможности Python
- Применять регулярные выражения
- Понимать принципы функционального программирования
- Сохранять данные пользователя в файлах cookies
- Работать с механизмом сессий
- Понимать принципы сетевого взаимодействия
- Взаимодействовать с источниками данных
- Внедрять AJAX в веб-приложения
- Уметь пользоваться системой контроля версий
- Понимать основы командного взаимодействия
- Применять паттерны проектирования
- Использовать юнит-тестирование
- Понимать и разбираться в тонкостях паттерна MVC
- Создавать веб-проекты с использованием Python и паттерна MVC
- Использовать Flask/Bottle
- Разрабатывать web-приложения с помощью фреймворка Django

По окончании данного курса студент сдаёт практическое задание и теоретический экзамен по материалам курса. Для допуска к экзамену, должны быть сданы все домашние и практические задания. Практическое задание должно охватывать максимум материала из различных разделов курса.

Модуль 1 (4 пары)

Введение в web-программирование на Python

1. Введение в web-программирование и принципы работы web-приложений
 - a. Основы технологии клиент-сервер
 - b. Обзор протокола HTTP, знакомство со структурой Request и Response
 - c. Обоснование и истоки возникновения web-программирования. Отличия серверного web-программирования от клиентского. Цели, задачи, направление развития, краткая история.
 - d. Принципы и этапы загрузки web-страницы.
2. Обзор языков программирования
 - a. Знакомство основными парадигмами программирования

- b. Обзор современных языков программирования
- c. Понятие алгоритма
- d. Знакомство с языком Python, сферы применения
- 3. Введение в Python. Интерпретатор Python и его окружение
 - a. Введение в Python
 - b. Понятие интерпретатора и порядок установки
 - c. Знакомство со средами программирования
 - i. Стандартный пакет программирования IDLE и Python Shell
 - ii. IDE PyCharm, Spyder, Visual Studio
 - iii. Atom
- 4. Типы данных, переменные и синтаксические конструкции
 - a. Тип и значения
 - b. Переменные как объект в языке Python
 - c. Имена переменных и зарезервированные слова
 - d. Инструкции
 - e. Операторы и операнды
 - f. Приоритеты операторов
 - g. Операции над переменными
 - h. Порядок выполнения программы
 - i. Ввод/вывод
 - j. Преобразование типов
 - k. Ошибки синтаксические и логические, работа с ними

Модуль 2 (14 пар)

Операторы ветвлений, циклы, исключения

- 1. Условные инструкции и их синтаксис
 - a. Понятие «блока» выполнения
 - b. Логические выражения и операторы
 - c. Операторы ветвления if ... else
 - d. Вложенные конструкции
- 2. Понятие исключений
 - a. Типы исключений
 - b. Перехват исключений
 - c. Особенности работы с try ... except
- 3. Циклы
 - a. Понятие итерации
 - b. Цикл while
 - c. Бесконечные циклы
 - d. Управляющие операторы continue, break и else
 - e. Цикл for
 - f. Локальные и глобальные переменные

Модуль 3 (8 пар)

Строки, списки

- 1. Строки
 - a. Кодировка ASCII, Unicode, UTF-8, Byte-code
 - b. Строка – неизменяемая последовательность символов
 - c. Методы строк
 - d. Особенности работы со строками

- e. Срез строки
 - f. Экранированные последовательности
 - g. «Сырые строки»
 - h. Форматированный вывод
 - i. Модуль string
 - j. Байты и кодировки
 - k. Регулярные выражения, модуль re
2. Списки
- a. Понятие классического массива
 - b. Понятие коллекции объектов
 - c. Ссылочный тип данных list
 - d. Создание списков
 - e. Генераторы списков
 - f. Работа со списками
 - g. Методы списков
 - h. Оператор принадлежности in
 - i. Особенности списков, ссылки и клонирование
 - j. Поиск элемента
 - k. Матрицы

Модуль 4 (8 пар)

Строки, списки

1. Функции и модули
 - a. Что такое функция?
 - b. Цели и задачи функции
 - c. Встроенные функции
 - d. Математические функции и случайные числа
 - e. Синтаксис объявления функций
 - f. Аргументы и возвращаемые значения
2. Расширенные приёмы по работе с функциями
 - a. Распаковка и упаковка аргументов
 - b. Аргументы по умолчанию, аргументы-ключи
 - c. Область видимости, правило LEGB
 - d. Локальные и глобальные переменные в функциях
 - e. Функции как объекты первого класса
 - f. Рекурсия
3. Функциональное программирование
 - a. Что такое функциональное программирование?
 - b. Анонимные функции lambda
 - c. Модуль functools
 - d. Функции map(), reduce(), filter(), zip()
 - e. Функции высших порядков
 - f. Замыкание
4. Замыкание
5. Карринг
6. Декораторы

Модуль 5 (6 пар)

Сортировка, поиск

1. Сортировка

- a. Оптимальность
 - b. Сортировка пузырьком
 - c. Сортировка слиянием
 - d. Сортировка Шелла
 - e. Пирамидальная сортировка
 - f. Быстрая сортировка
2. Поиск
- a. Линейный поиск
 - b. Бинарный поиск

Модуль 6 (5 пар)

Кортежи, множества, словари

1. Кортежи
 - a. Коллекции неизменяемых объектов
 - b. Применение и особенности кортежа
2. Множества
 - a. Математическое понятие множеств
 - b. Тип данных set(), frozenset()
 - c. Операции над множествами
 - d. Применение множеств
3. Словари
 - a. Ассоциативные массивы
 - b. Хеш-таблицы
 - c. Создание словаря
 - d. Методы словаря
 - e. Понятие разреженной матрицы
4. Практические примеры использования

Модуль 7 (5 пар)

Файлы

1. Файлы
 - a. Файловая система, особенности реализации форматов
 - b. Работа с файлами
 - i. Открытие
 - ii. Заккрытие
 - iii. Чтение
 - iv. Запись
2. Менеджер контекста
3. Типы файлов, текстовые и бинарные
4. Практические примеры использования

Модуль 8 (4 пары)

Упаковка данных

1. Сериализация и десериализация
 - a. Что такое сериализация?
 - b. Что такое десериализация
 - c. Цели и задачи сериализации и десериализации
 - d. Практические примеры использования
2. Модуль pickle
3. Модуль json
4. Сторонние модули сериализации

Модуль 9 (4 пары)

Системы контроля версий

1. Что такое контроль версий?
2. Зачем нужен контроль версий
3. Обзор систем контроля версий
 1. CVS
 2. SVN
 3. Git
 4. Другие системы контроля версий
4. Git
 1. Что такое Git?
 2. Цели и задачи Git?
 3. Основные термины
 1. Репозиторий
 2. Коммит
 3. Ветка
 4. Рабочий каталог
 4. Операции с Git
 1. Установка
 2. Создание репозитория
 3. Добавление файла в репозиторий
 4. Запись коммита в репозиторий
 5. Получение текущего состояния рабочего каталога
 6. Отображение веток
 7. Операции с накопительным буфером
 8. git remote
 9. git push
 10. git pull
 11. Другие операции
 5. Использование внешних сервисов (github)

Модуль 10 (14 пар)

ООП

1. Введение в ООП
 - a. Понятие ООП
 - b. Инкапсуляция
 - c. Наследование
 - d. Полиморфизм
 - e. Особенности реализации ООП в Python, «утиная типизация»
2. Типы данных, определяемые пользователем
 - a. Экземпляр класса
 - b. Классы и объекты
 - c. Поля (свойства), методы класса
3. Наследование и инкапсуляция
 - a. Общедоступный, внутренний и приватный метод
 - b. Magic-методы, конструкторы
 - c. Статические методы и методы класса
 - d. Множественное наследование и MRO (порядок разрешения методов)
4. Полиморфизм
 - a. Перегрузка операторов
 - b. Реализация магических методов

5. Создание и управление поведением экземпляров класса
 - a. Функторы
 - b. Декораторы
 - c. Управляемые атрибуты
 - d. Свойства
 - e. Дескрипторы
6. Метаклассы
 - a. Модель метаклассов
 - b. Метод конструктор `__new__()`
 - c. Протокол инструкции `class`

Модуль 11 (5 пар)

Структуры данных

1. Связанные списки
 - a. Что такое список?
 - b. Односвязный и двусвязный список
 - c. Практические примеры использования
2. Стек
 - a. Что такое стек?
 - b. Принцип LIFO
 - c. Практические примеры использования
3. Очередь
 - a. Что такое очередь?
 - b. Виды очередей
 - c. Практические примеры использования
4. Деревья
 - a. Что такое дерево?
 - b. Виды деревьев
 - c. Практические примеры использования

Модуль 12 (4 пары)

Паттерны проектирования

1. Что такое паттерны проектирования
2. Причины возникновения паттернов проектирования
3. Понятие паттерна проектирования
4. Принципы применения паттернов проектирования
5. Принципы выбора паттернов проектирования
6. Принципы разделения паттернов на категории
7. Введение в UML
 - a. Диаграмма классов
 - b. Диаграмма объектов
 - c. Диаграмма взаимодействия
8. Использование UML при анализе паттернов проектирования
 - a. Диаграмма классов
 - b. Диаграмма объектов
 - c. Диаграмма взаимодействия
9. Порождающие паттерны
 - a. Что такое порождающий паттерн?
 - b. Цели и задачи порождающих паттернов
 - c. Обзор порождающих паттернов
 - d. Разбор порождающих паттернов
 - i. Abstract Factory

1. Цель паттерна
 2. Причины возникновения паттерна
 3. Структура паттерна
 4. Результаты использования паттерна
 5. Практический пример использования паттерна
 - ii. Builder
 1. Цель паттерна
 2. Причины возникновения паттерна
 3. Структура паттерна
 4. Результаты использования паттерна
 5. Практический пример использования паттерна
 - iii. Factory Method
 1. Цель паттерна
 2. Причины возникновения паттерна
 3. Структура паттерна
 4. Результаты использования паттерна
 5. Практический пример использования паттерна
 - iv. Prototype
 1. Цель паттерна
 2. Причины возникновения паттерна
 3. Структура паттерна
 4. Результаты использования паттерна
 5. Практический пример использования паттерна
 - v. Singleton
 1. Цель паттерна
 2. Причины возникновения паттерна
 3. Структура паттерна
 4. Результаты использования паттерна
 5. Практический пример использования паттерна
10. Структурные паттерны
 - a. Что такое структурный паттерн?
 - b. Цели и задачи структурных паттернов
 - c. Обзор структурных паттернов
 - d. Разбор структурных паттернов
 - i. Adapter
 - ii. Composite
 - iii. Facade
 - iv. Proxy
 - v. Другие структурные паттерны
11. Паттерны поведения
 - a. Что такое паттерны поведения?
 - b. Цели и задачи паттернов поведения
 - c. Обзор паттернов поведения
 - d. Разбор паттернов поведения
 - i. Command
 - ii. Iterator
 - iii. Observer
 - iv. Strategy
 - v. Другие структурные паттерны

Модуль 13 (2 пары)

Паттерн MVC

1. Что такое паттерн MVC?
2. Цели и задачи паттерна Model-View-Controller
3. Model
 - a. Что такое Model?
 - b. Цели и задачи Model
4. View
 - a. Что такое View?
 - b. Цели и задачи View
5. Controller
 - a. Что такое Controller?
 - b. Цели и задачи Controller
6. Примеры использования паттерна MVC

Модуль 14 (2 пары)

Принципы проектирования классов SOLID

1. Обзор проблем, встречающихся при проектировании и разработке классов
2. Принципы проектирования классов SOLID
 - a. Принцип единственности ответственности (The Single Responsibility Principle)
 - b. Принцип открытости/закрытости (The Open Closed Principle)
 - c. Принцип подстановки Барбары Лисков (The Liskov Substitution Principle)
 - d. Принцип разделения интерфейса (The Interface Segregation Principle)
 - e. Принцип инверсии зависимостей (The Dependency Inversion Principle)
3. Примеры использования принципов SOLID

Модуль 15 (2 пары)

Модульное тестирование

1. Что такое модульное тестирование?
2. Цели и задачи модульного тестирования
3. Необходимость модульного тестирования
4. Обзор инструментов для модульного тестирования
5. Инструмент для модульного тестирования Python приложений

Модуль 16 (5 пар)

Параллельное, многопоточное и сетевое программирование

1. Параллельное и многопоточное программирование
 - a. Создание потоков
 - b. Синхронизация потоков
 - c. Очереди задач
 - d. GIL и особенности реализации многопоточности в Python
 - e. Процессы и передача данных между процессами
2. Сетевое программирование
 - a. Протокол HTTP/HTTPS
 - b. Модель OSI, tcp/udp
 - c. Клиент-серверная модель
 - d. Реализация эхо-сервера простейшего, многопоточного и асинхронного
 - e. Apache и Nginx

Модуль 17 (10 пар)

Введение в работу с базами данных

1. Введение в теорию баз данных
 1. История и этапы развития
 2. Понятия база данных и система управления базами данных
 3. Сравнение существующих моделей баз данных
 1. Файловая модель
 2. Сетевая модель
 3. Иерархическая модель
 4. Реляционная модель
 5. Объектно-ориентированная модель
 4. Понятие реляционной модели баз данных
 5. Двенадцать правил Кодда
 6. СУБД MySQL
 1. Что такое MySQL?
 2. История развития MySQL
 3. Версии MySQL
 4. Установка MySQL
 7. Таблицы
 1. Первичный ключ
 2. Значение по умолчанию
 3. Уникальность
 8. Типы данных
 9. Индексы
 10. Запросы
 1. Введение в язык структурированных запросов SQL
 2. Язык SQL. Стандарты языка SQL
 3. Понятия DDL, DML, DCL
2. Запросы SELECT, INSERT, UPDATE, DELETE
 1. Оператор SELECT
 1. Предложение SELECT
 2. Предложение FROM
 3. Предложение WHERE
 4. Предложение ORDER BY
 2. Ключевые слова IN, BETWEEN, LIKE
 3. Оператор INSERT
 4. Оператор UPDATE
 5. Оператор DELETE
3. Многотабличные базы данных
 1. Аномалии взаимодействия с однотабличной базой данных
 1. Аномалии обновления
 2. Аномалии вставки
 3. Аномалии обновления
 4. Аномалии удаления
 2. Принципы создания многотабличной базы данных
 1. Причины создания многотабличной базы данных
 2. Внешний ключ
 3. Связи. Типы связей
 4. Целостность данных
4. Нормализация
 1. Необходимость нормализации
 2. Понятие нормальной формы

3. Первая нормальная форма
4. Вторая нормальная форма
5. Третья нормальная форма
6. Нормальная Форма Бойса-Кодда
5. Многотабличные запросы
 1. Принципы создания многотабличного запроса
 2. Декартовое произведение
6. Функции агрегирования
 1. Функции агрегирования
 1. Функция COUNT
 2. Функция AVG
 3. Функция SUM
 4. Функция MIN
 5. Функция MAX
7. Понятие группировки. Ключевое слово GROUP BY
8. Ключевое слово HAVING. Сравнительный анализ HAVING и WHERE
9. Подзапросы
 1. Необходимость создания и использования подзапросов
 2. Сравнение подзапросов и многотабличных запросов
 3. Принцип работы подзапросов
10. Операторы для использования в подзапросах, объединения
 1. Операторы для использования в подзапросах
 1. Оператор EXISTS
 2. Операторы ANY/SOME
 3. Оператор ALL
 2. Объединение результатов запроса
 1. Принципы объединения
 2. Ключевое слово UNION
 3. Ключевое слово UNION ALL
 3. Объединения JOIN
 1. Понятие inner join
 2. Понятие left join
 3. Понятие right join
 4. Понятие full join
11. План выполнения запроса
12. Оптимизация запросов
13. Понятие транзакции. Использование транзакций.
14. Представления.
 1. Создание представлений
 2. Модификация представлений
 3. Удаление представлений
 4. Изменения данных через представления
15. Хранимые процедуры.
16. Триггеры.

Модуль 18 (8 пар)

Использование баз данных в Python, библиотеки Numpy и Pandas

1. Использование баз данных
2. ORM системы
 - a. Теория и практика использования
 - b. SQLAlchemy

- с. PonyORM и другие
- 3. NoSQL базы данных
 - а. Основы NoSQL, теорема CAP
 - б. Основные типы NoSQL баз данных
 - с. СУБД Redis
 - д. СУБД MongoDB
- 4. Работа с базами данных, сериализация данных
 - а. Запись и чтение в формате XML DOM
 - б. StAX и SAX-парсеры

Модуль 19 (4 пары)

Работа в команде, управление программными проектами

1. Что такое управление программными проектами?
2. Причины возникновения дисциплины управление программными проектами.
3. Диаграммы Ганта.
4. Важные вопросы по управлению программными проектами
 1. Что такое проект и программный проект?
 2. Что такое жизненный цикл процесса разработки программного обеспечения?
 3. Что такое управление проектами?
 4. Что такое одиночная разработка?
 5. Что такое командная разработка?
 6. Анализ проблем одиночной и командной разработки программного обеспечения
5. Анализ терминов предметной области
 1. Процесс
 2. Проект
 3. Персонал
 4. Продукт
 5. Качество
6. Характеристики проекта
 1. Тип проекта
 2. Цель проекта
 3. Требования к качеству
 4. Требования к бюджету
 5. Требования по срокам завершения
7. Расходы, связанные с проектом
 1. Прямые
 2. Непрямые
8. Общий обзор моделей и методологий процесса разработки
 1. Фазы процесса
 - Определение требований
 - Проектирование
 - Конструирование (“реализация”, “кодирование”)
 - Интеграция
 - Тестирование и отладка (“верификация”)
 - Инсталляция
 - Поддержка
 2. Водопадная модель
 3. Спиральная модель
 4. Итеративная модель
 - Agile

- Scrum
 - XP
- 5. RUP
- 6. MSF
- 7. Анализ существующих моделей и методов
- 9. Подробнее о Scrum
 1. Что такое Scrum?
 2. Причины возникновения Scrum
 3. Роли в Scrum
 - Владелец продукта
 - Команда
 - Scrum мастер
 4. Бэклог продукта
 - Что такое бэклог продукта?
 - Как создавать бэклог?
 - Как оценивать задачи в бэклоге?
 - Что такое scrum-доска?
 - Примеры создания бэклога
 5. Спринт
 - Что такое спринт?
 - Планирование спринтов
 - Ежедневный скрам
 - Обзор спринта
 - Ретроспективное собрание
 - Практическое задание: Необходимо провести симуляцию работы команды по методологии Scrum. Например, это может быть так называемое скрам-лего. Подробно тут:
https://www.scrumalliance.org/system/resource_files/0000/3689/Scrum-Simulation-with-LEGO-Bricks-v2.0.pdf

Модуль 20 (24 пары)

Фреймворки

1. Классификация web-фреймворков
 - a. Web-фреймворк Flask
 - b. Механизм шаблонов и язык шаблонов Jinja2
 - c. Сессии и формы
 - d. WEB-фреймворк Bottle
2. Асинхронные web-приложения
 - a. Фреймворк Tornado
 - b. Библиотека Twisted
3. Django – фреймворк для создания web-приложений
 - a. Установка Django
 - b. Создание проекта Django
 - c. Структура Django-проекта (url-view-model-template)
 - d. Паттерн MVC/MVT
4. Модели и ORM
 - a. Модели и поля
 - b. Связи между таблицами, проектирование и реализация БД благодаря механизму ORM
 - c. Миграции
 - d. Менеджеры модели

- е. Административная часть
- 5. Работа с административной частью, настройка отображения
 - а. Настройка интерфейса администратора
 - б. Сортировка, фильтры, редактирование полей
 - с. Вывод данных
- 6. Язык шаблонов и создание web-форм
 - а. Шаблоны
 - б. Роутинг, представления
 - с. Синтаксис, логические конструкции
 - д. Контекст процессор
- 7. Создание форм и страниц
 - а. Добавление страниц
 - б. GET/POST запросы
 - с. Статическое содержимое и динамическое
 - д. Авторизация
 - е. Ограничения прав доступа
 - ф. Валидация
 - г. Ajax
- 8. Стандартные задачи
 - а. Встроенные class-based views
 - б. Использование форм с Django CBV
- 9. Стандартные задачи
 - а. Аутентификация
 - б. Пагинация
- 10. Погружение в Django
 - а. Middleware
 - б. Сигналы
 - с. Сообщения
 - д. Сессии
- 11. Продолжение
 - а. Древовидные структуры в Django
 - б. Миксины
 - с. Дебаггер Django
 - д. Логирование, отправка почты

Модуль 21 (2 пары)

Экзамен