

Web-разработчик на Python

Курс предназначен для подготовки специалиста, который сможет не только решать рядовые задачи бекенд-разработки, но и сделать с нуля современную фронтенд часть.

Длительность курса: 188 академических часов

1 Модуль

- 1 **Технические детали курса; декомпозиция; чистые функции; цикломатическая сложность; читаемость; фичеризм; поддерживаемость; статический анализ кода.**

Домашние задания

- 1 Рефакторинг существующего кода.

Отрефакторить и оформить в виде библиотеки на Гитхабе следующий код:

<https://gist.github.com/VladimirFilonov/cf841d9dd2d11dad8c2dc7a3fef4ac3>

- 2 **Объекты и переменные; структуры данных под капотом основных типов; list vs tuple; профилирование памяти; как работает сборщик мусора.**

- 3 **Принципы ООП; mixins; class/instance variables; MRO; dict; slots; name mangling.**

Домашние задания

- 1 Добавление фич в скрипт для статического анализа кода.

<https://gist.github.com/VladimirFilonov/775853d680f4ea7f0ce5a75d094490f7>

4 IP, TCP, HTTP; [u]WSGI.

Домашние задания

- 1 Написать свой wsgi-совместимый веб-фреймворк.
-

5 werkzeug; Flask.

Домашние задания

- 1 Сделать демо интернет-магазина на Flask.
-

6 ORM – зачем нужно;
описание схемы данных;
сессии;
добавление/удаление/апдейт
данных

Домашние задания

- 1 Написать свою ORM для sqlite.
-

7 Django apps; Django ORM;
Django Class Based Views;
Django common tools.

Домашние задания

- 1 Сделать демо интернет-магазина на Django.
-

8 Семинар. Реализуем свой
WSGI-совместимый
фреймворк.

1 Django REST Framework. Django Configurations.

Домашние задания

- 1 Реализуем базовый бекенд для сайта с курсами.

Цель: У нас есть курс, у которого есть занятия, у курсов есть преподаватели. Есть пользователи, которые могут зарегистрироваться, залогиниться и записаться на курс. Сайт будет разделен на две составляющие - backend и frontend. Backend это REST-API на Django, который позволяет: - зарегистрироваться, можно без капчи и подтверждения email; - залогиниться; - посмотреть список курсов; - зайти внутрь одного курса, где посмотреть его описание и список уроков, прикреплённых к дате; - дополнительно можно задать возможность записаться на курс. Фронтенд пока не делаем - его будем прикручивать позже, на уроках по JS

2 API. GraphQL и его реализация в Python. GraphQL и Django.

Домашние задания

- 1 Добавляем сущности и фичи для API сайта с курсами.
-

3 Fabric. Развертывание проекта на сервере.

4 Софтскилз для разработчика

5 Знакомство с Front-end частью курса. Основы HTML, CSS, методологии вёрстки. Немного Bootstrap 4.

Домашние задания

- 1 Сделать вёрстку (HTML + CSS) главной страницы сайта для обучения.
-

6 Django m2m, select_related/prefetch_related, django debug toolbar, factory_boy, faker.

7 **Основы JS: типы данных, операторы, объекты, работа с DOM и браузером**

Домашние задания

- 1 Добавление поведения к главной странице, страница курсов
-

8 **Django ORM, оптимизация работы с БД**

- | | | |
|---|--|--|
| 1 | Продвинутый JS: ООП в JS, прототипирование, асинхронность. | Домашние задания
1 Страница расписания занятий |
| 2 | Азы работы с очередями задач. | Домашние задания
1 Добавляем асинхронную отправку электронной почты |
| 3 | ES6, NodeJS окружение, babel + webpack, транспайлинг, CSS препроцессоры | Домашние задания
1 Сборка UI с помощью webpack, форма логина и регистрации, проксирование вызовов на back-end |
| 4 | fetch axios \$.ajax для REST запросов, модульность | Домашние задания
1 Загрузка списка курсов с сервера, логин страница, REST API для работы |
| 5 | Пишем свой менеджер очередей. | |
| 6 | Code review | Домашние задания
1 Исправление недостатков в соответствии с code review |
| 7 | Введение в автотесты. | Домашние задания
1 Покрыть проект приёмочными тестами на 60% |

1	Основы React, JSX, компоненты React	Домашние задания 1 Главная страница на React
2	Введение в TDD.	
3	state и props, data-flow в React-компонентах	Домашние задания 1 Страница логина, регистрации на React
4	Введение в pytest.	Домашние задания 1 Переписать тесты на pytest и factory_boy
5	Жизненный цикл React-компонент	Домашние задания 1 Страница личного кабинета, отправка решения на сервер
6	InfluxDB, Grafana.	
7	Состояние приложения. Flux & Redux	Домашние задания 1 Блок успеваемости в личном кабинете, ведомость в кабинете преподавателя
8	Jenkins, Sentry.	Домашние задания 1 Настроить логирование ошибок в Sentry.

1 **Code review**

Домашние задания

- 1 Исправление недостатков в соответствии с code review
-

2 **python-telegram-bot**3 **Routing в React. SPA**

Домашние задания

- 1 Организация всего приложения в виде SPA.
-

4 **threading,
multiprocessing, queue.**5 **Тестирование JS
приложений**

Домашние задания

- 1 Написание unit-тестов для UI и back-end
-

6 **Введение в docker,
docker-compose.**

Домашние задания

- 1 Завернуть проект в докер-контейнеры
-

7 **Обзор Vue2**

Домашние задания

- 1 Landing page на Vue
-

8 **asynсio**9 **QnA**