

Реализация Front-end

№ урока: 4 **Курс:** Django Essential

Средства обучения: Pycharm Community, Atom, GIT, аккаунт в GitHub, Python 3.6 и выше, Django проект, Docker образ.

Обзор, цель и назначение урока

Научиться работать с Django templates и добавить в проект front-end. Рассмотреть фреймворк Bootstrap как базовый front-end для проекта. Научиться использовать правильные типы view в зависимости от задачи. И последние - понять предназначение статик файлов.

Изучив материал данного занятия, учащийся сможет:

- Настраивать структуру front-end части.
- Создавать шаблон.
- Создавать свои теги.
- Подключать шаблон.
- Подключать к проекту фреймворк Bootstrap.
- Подключать статик файлы для front-end.
- Использовать различные типы view.

Содержание урока

1. Создание index.html
2. Создание структуры в index.html
3. Создание других шаблонов для приложения
4. Подключение всех шаблонов к index.html
5. Добавления html и Django тегов в шаблоны
6. Подключение Bootstrap: локально и с помощью ссылки
7. Создание структуры статик файлов
8. Добавление css, js, jquery и других дополнительных библиотек
9. Рассмотрение типов view и способы их применения

Резюме

- Index.html - "лицо" вашего сайта.
- Шаблоны Django позволяют вставлять Python в HTML, так что можно создавать динамические веб-сайты быстрее и проще.
- Переменные, фильтры, теги и комментарии - 4 основные конструкции в шаблонах.
- Основа основ: head, page, footer.
- Для реализации наследования и подключения других шаблонов используем ключевые теги extend и block.
- Для создания своего тега используем декоратор @register.
- Не забываем, что много уже реализовано, поэтому старайтесь использовать существующий код, не выдумывайте велосипед.
- Bootstrap - один из наиболее популярных HTML и CSS фреймворков для разработки веб-приложений.
- Скачиваем Bootstrap с официального сайта <https://getbootstrap.com/>.
- Используем статик файлы и папки.

- Структура: static, css, js, html
- Подключаем Bootstrap с помощью <link>.
- Подключаем js, jquery по аналогии.
- Категории view: AUTH VIEWS, GENERIC BASE, GENERIC DATES, GENERIC LIST, GENERIC EDIT, GENERIC DETAIL.
- Рассмотрим class ListView.
 - Атрибуты
 - Методы
 - Использование

Закрепление материала

- Зачем нужен index.html?
- Назовите 4 типа конструкций в Django templates.
- Какой принцип работы Django templates?
- Какая структура статик файлов?
- Какие есть классы для реализации view?

Дополнительное задание

Задание

Подключить к проекту Bootstrap, создать структуру шаблонов и создать свой тег.

Самостоятельная деятельность учащегося

Задание 1

Выучить и понять все преимущества и недостатки от шаблонов.

Задание 2

Реализовать в одном из шаблонов следующий код: создать цикл, который бежит по экземплярам модели и создает таблицу с парными id (2, 4, 6, 8, 10 ... n). Добавить условие - если элемент, кратный 10 (10, 20, 30 ... n) - выделять его красным. Подключить к шаблону html и css.

Задание 3

Подключить к проекту пакет Chart.js не используя ссылок в index.html, а используя статик файлы.

Рекомендуемые ресурсы

Django документация:

- <https://docs.djangoproject.com/en/2.2/ref/templates/language/#variables>

Описание view: <https://ccbv.co.uk/projects/Django/2.2/django.views.generic.list/ListView/>

Chart.js библиотека: <https://www.chartjs.org/>

Bootstrap: <https://getbootstrap.com/>

jQuery: <https://jquery.com/>