

# Представления

№ урока: 3 Курс: Django Starter

**Средства обучения:** Персональный компьютер с установленными:  
Python 3.8  
Django 3.0

## Обзор, цель и назначение урока

Целью данного урока является ознакомиться с основами представлений в Django. Представление отвечает за то, что именно отдается пользователю.

## Изучив материал данного занятия, учащийся сможет:

- Составлять представления на основе функций.
- Работать с объектами Джанго `HttpResponse`, `HttpRequest`.
- Составлять представления на основе классов.
- Понимать, как взаимодействуют между собой представления и модели.

## Содержание урока

1. Function Based Views
2. Взаимодействие представления и модели
3. Class Based Views

## Резюме

- **view (представление)** — это функции/методы в Python, которые отвечают за отображение. Чаще всего одно view соответствует одной странице.
- **Function Based Views** – это представления на основе функций.
- **Class Based Views** – представления на основе классов.
- Объект **HttpRequest** содержит в себе информацию о запросе клиента.
- Объект **HttpResponse** содержит информацию, используемую сервером для ответа клиенту.
- У **HttpResponse** есть 4 основных атрибута: **content**, **status**, **reason\_phrase** и **charset**.
- **ORM** (Object-Relation Mapping, Объектно-реляционное отображение) – это связь между объектами и базой данных. Представления используют **ORM** для доступа к БД. В Django это обычно **Django ORM**. С его помощью мы получаем или сохраняем данные в БД.
- Основные отличия представлений на основе классов от представлений на основе функций это объединение специфичных для HTTP методов (GET, POST, и т.д.) в один класс, а также возможность повторного использования кода с помощью примесей (миксинов) и множественного наследования.
- У представления на основе класса входной точкой является метод **as\_view**, который указывается как точка входа в класс при маршрутизации. В нём, с помощью метода **dispatch** определяется тип запроса и вызывается метод, соответствующий HTTP коду запроса. Поддерживаемые названия методов: **get**, **post**, **put**, **patch**, **delete**, **head**, **options**, **trace**

## Закрепление материала

- Что такое view (представление)?

- Что такое Function Based Views? Что такое Class Based Views? В чем основная разница между ними?
- Какой объект содержит в себе информацию о запросе клиента?
- Какой объект содержит информацию, используемую для ответа клиенту сервером?
- Что такое ORM?

### Дополнительное задание

#### Задание

Составить представления, которые будут использовать различные типы ответов: файл, json, html и текст,

Реализовать несколько Function и Class Based Views.

### Самостоятельная деятельность учащегося

#### Задание 1

Дан следующий список:

```
lets_do_it = [{'priority': 100, 'task': 'Составить список дел'}, {'priority': 150, 'task': 'Изучать Django'}, {'priority': 1, 'task': 'Подумать о смысле жизни'}]
```

Вывести его в представлении.

#### Задание 2

Составить структуру сайта с помощью представлений так, чтобы у вас была главная страница с приветствием и ссылками на другие страницы сайта:

Добро пожаловать во вселенную звездных войн!

Выберите страницу:

[Люк](#)

[Лея](#)

[Хан](#)

Страница Люка:

Люк Скайуокер — один из главных персонажей вселенной «Звёздных войн», джедай, сын сенатора с Набу Падме Амидалы Наберри и рыцаря-джедая Энакина Скайуокера

Страница Леи:

Лея Органа — дочь рыцаря-джедая Энакина Скайуокера и сенатора Падме Амидалы Наберри.

Страница Хана:

Хан. Соло — пилот космического корабля «Тысячелетний сокол», его бортмехаником и вторым пилотом является вуки по имени Чубакка.

#### Задание 3

Создать представление и свой HTTPResponse который бы отправлял пользователи файл с текстом «Вот ваш файл» и кодом 227.

### Рекомендуемые ресурсы

Официальный с документацией по Django:

