

В Django **переменные (placeholders)** и **теги** используются в шаблонах для динамического формирования содержимого страницы.

Если передать в контекст шаблона first\_name, то в шаблоне можно использовать:



**Теги** используются для выполнения логики прямо в шаблоне.

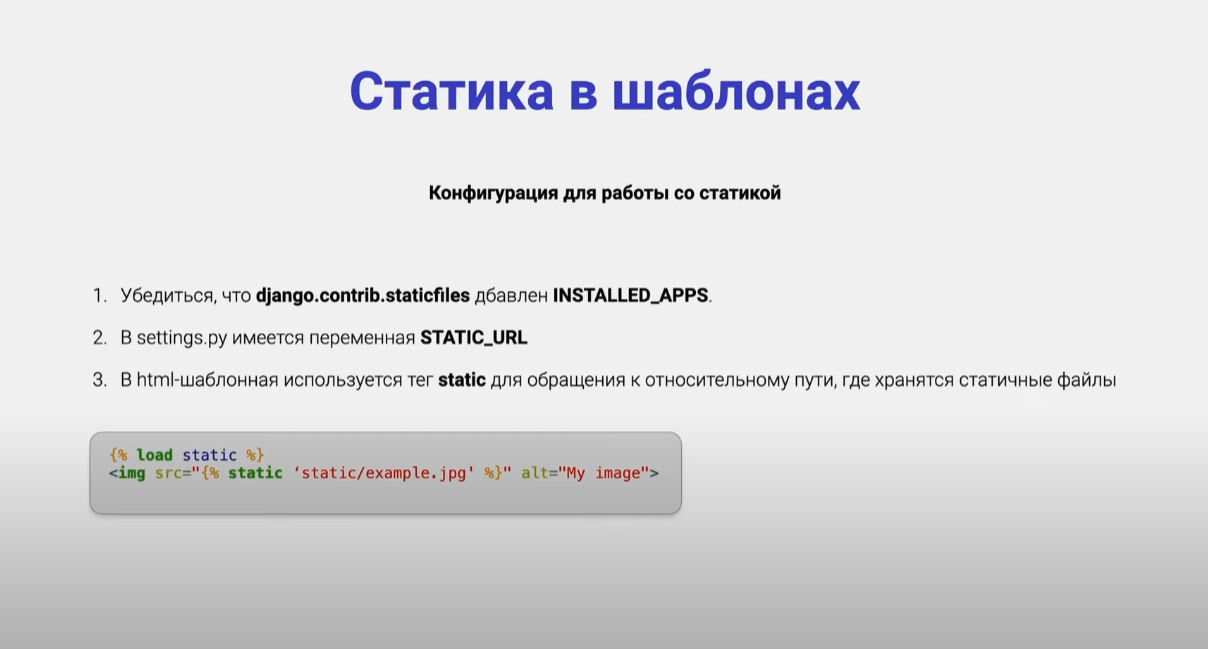
Обычно **переменные** передаются из views через контекст, а **теги** управляют их отображением.

-----------------------

Во время разработки файлы статического контента подгружаются напрямую из папки, указанной в **STATICFILES\_DIRS**.

Для продакшена и разработки пути к статике часто отличаются. Используя **STATIC\_URL**, легко настроить разные URL-адреса для каждой среды.

Django предоставляет специальный тег {% static %} для автоматического формирования URL для статических файлов.



Django сам подставит вместо тега **static** значение **STATIC\_URL** из **settings.py**, делая путь универсальным.

**STATIC\_URL:**

* Централизует управление путями к статическим файлам.
* Упрощает настройку URL для разработки и продакшена.
* Работает с тегом {% static %} для удобства написания шаблонов.

-----------------------

Динамические URL в шаблонах Django позволяют автоматически формировать ссылки на страницы, основываясь на именах маршрутов, заданных в **urls.py**.

1. Присваивается имя маршруту с помощью параметра **name** в файле **urls.py**.
2. В шаблоне используется тег **{% url ‘[имя маршрута]’ %}** для генерации ссылки.

----------------------

**DRY - Don’t repeat yourself**

В Django для избежания повторений в шаблонах используется механизм шаблонного наследования, который включает в себя такие конструкции, как **{% extends %}** и **{% block %}**.

1. Создаётся **базовый шаблон** с общей структурой сайта. Он содержит блоки ({% block %}), которые можно переопределять в дочерних шаблонах.
2. **Дочерний шаблон** наследует базовый с помощью {% extends %}. Он может переопределить блоки, заданные в базовом шаблоне.

----------------------

**ORM** (объектно-реляционное отображение) - механизм работы с БД через Python-код, без написания SQL-запросов.

**Django ORM** автоматически преобразует **модели** (классы, записанные в **models.py**, которые определяют структуру таблиц в БД) в записи БД.

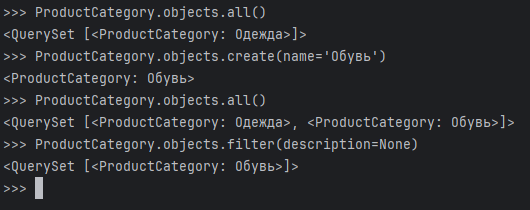
**Миграции** — это файлы, которые фиксируют изменения в моделях и применяют их к БД.

----------------------

**QuerySet** - набор запросов, который представляет собой набор объектов из БД.

Методы QuerySet:

* **create()** - создаёт объект из БД
* **get()** -возвращает объект из БД
* **all()** - возвращает список всех объектов из БД
* **filter()** - возвращает список объектов по признаку



----------------------

В Django работа с медиафайлами (например, изображениями и видео) основывается на двух ключевых переменных в settings.py:

**MEDIA\_ROOT** – это путь на сервере, куда будут загружаться файлы.

*[ MEDIA\_ROOT = BASE\_DIR / "media" ]*

**MEDIA\_URL** – это URL, по которому можно будет получить доступ к загруженным файлам.

*MEDIA\_URL = "/media/"*

Например, если **MEDIA\_URL** = "/media/", а файл загружен в **media/uploads/image.jpg,** он будет доступен по **<http://localhost:8000/media/uploads/image.jpg.>**

----------------------

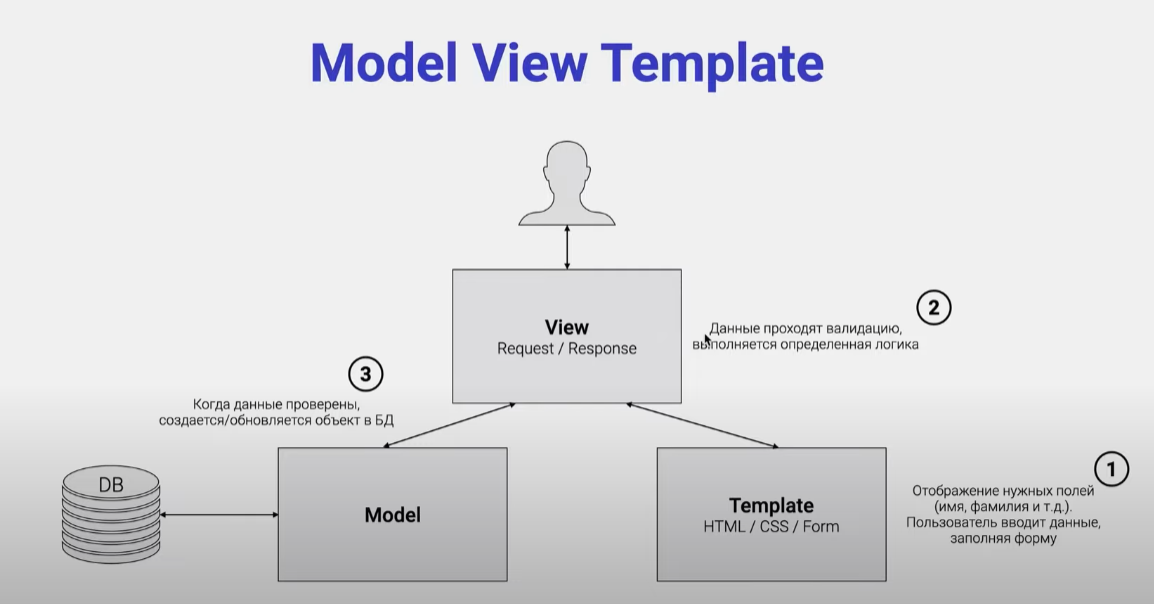
**Fixtures** – это файлы с данными для заполнения базы данных (обычно в формате JSON). Они используются для резервного копирования, тестирования или загрузки предустановленных данных.

**dumpdata** создаёт фикстуру, сохраняя данные из БД в файл.

**loaddata** считывает данные из фикстуры и загружает их обратно в БД.

Используется для начального наполнения БД, резервного копирования или тестов.

----------------------



----------------------

**URL Dispatcher:**

В Django маршруты можно организовать по уровням, где главный urls.py (на уровне проекта) передаёт управление urls.py приложениям с помощью include(). Это похоже на наследование, так как глобальный urls.py задаёт общие правила, а каждый urls.py в приложении управляет своими маршрутами.

Смысл и преимущества такого подхода:

✅ Поддержка модульности – каждый модуль (приложение) управляет своими маршрутами.  
✅ Гибкость – легко добавлять и изменять маршруты без изменения главного urls.py.  
✅ Чистая архитектура – URL-адреса организованы по логическим блокам.

----------------------