

Python для продвинутых специалистов (Django)



**Тема №1 Введение в Django.
Инструментальные средства для
разработки web приложений.**

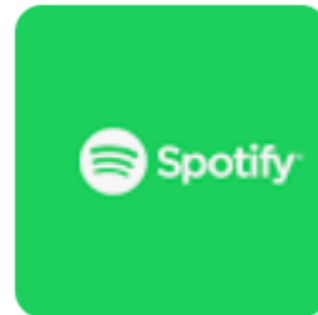
**Преподаватель: Панченко Игорь
Валентинович**

Общие сведения о Django

Что такое Django?

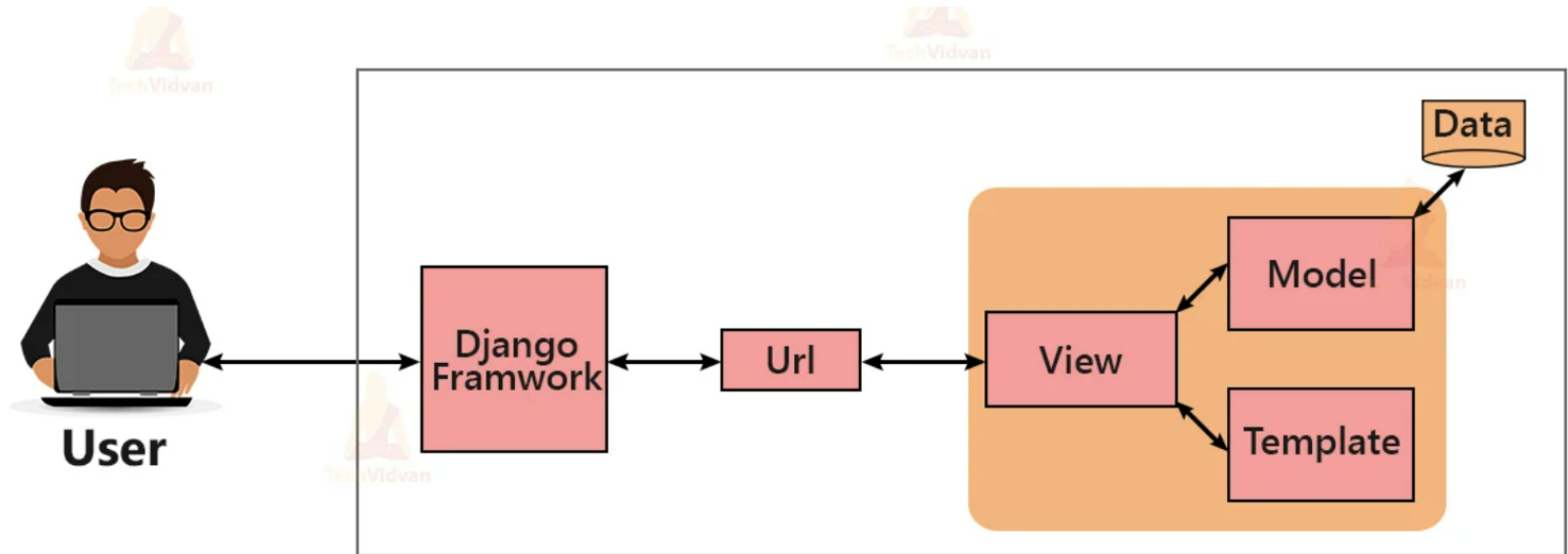
Django – Это фреймворк с открытым исходным кодом.

Django позволяет создавать высоконагруженные приложения с дружественными интерфейсами и обеспечивать безопасность данных.



Архитектура MVT

Django приложения используют архитектуру MODEL-VIEW-TEMPLATE.



Model

Модель – это компонент Django-приложения, который связывает между собой пользовательский интерфейс и базу данных.

Обычно, в этом компоненте реализуется логика предметной области приложения

Чаще всего на практике применяются реляционные базы данных: Sqlite3, MySQL или PostgreSQL

View

По-русски этот компонент называется **отображение** или **представление**. Обычно, этот компонент содержит логику пользовательского интерфейса.

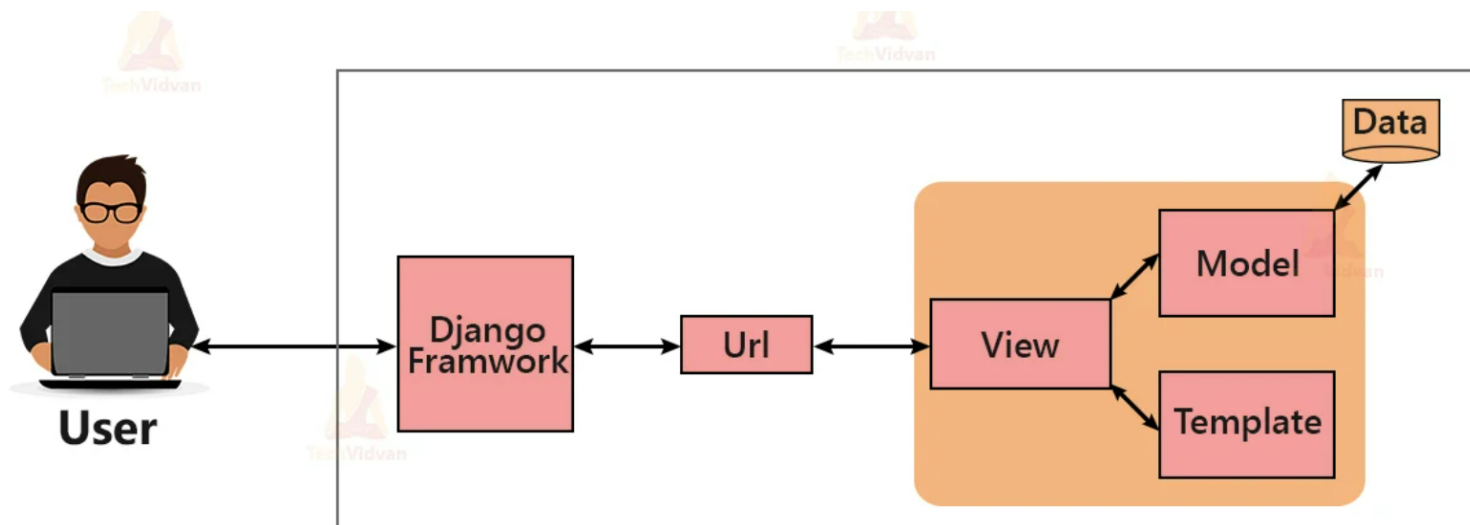
Эта часть приложения обрабатывает запросы, получает данные, вводимые пользователем на страницах веб сайта, обращается к моделям из БД и отправляет ответ в шаблоны сайта.

Template

Шаблон – это инструмент, позволяющий динамически создавать веб-страницы, изменяя их содержимое в зависимости от изменений в БД сайта. Для работы с шаблонами в Django есть специальный инструмент – «шаблонизатор», который позволяет использовать вывод данных, циклы и условия (и многое другое) прямо в HTML-коде страниц.

Пример обработки запроса

1. Пользователь отправляет URL-запрос на сайт.
2. Django отслеживает URL-ссылку
3. URL –ссылка вызывает соответствующий View
4. View запрашивает данные из БД через модели
5. View передает данные в шаблон, который будет отображен в браузере пользователя





Пример работы шаблонизатора


Один и тот же HTML-шаблон используется для отображения информации о разных товарах интернет магазина

Welcome to 9 Бит!

STORE NEWSLETTER FAQ ENG Российский рубль


 All Categories

 ADMIN PROFILE LOGOUT

 MY CART: 304500.00 RUB

CATEGORIES HOME CONTACT PAGE ABOUT US ADMIN PANEL

Home / Computers / Desktop PC / HP ProDesk 600



HP ProDesk 600

Availability: Out of stock

63000.00 RUB ~~63100 RUB~~

QTY: [ADD TO CART](#)

No reviews yet!
Be the first one who review this item!

Availability: In Stock

Характеристики ПК HP ProDesk 400 G7 MT [11M76EA]

Заводские данные

Гарантия продавца / производителя

12 мес.

Общие параметры

Тип

Welcome to 9 Бит!

STORE NEWSLETTER FAQ ENG Российский рубль

 All Categories

 ADMIN PROFILE LOGOUT

 MY CART: 304500.00 RUB

CATEGORIES HOME CONTACT PAGE ABOUT US ADMIN PANEL

Home / Smartphones / Apple iPhone 12



Apple iPhone 12

Choose configuration

QTY: [ADD TO CART](#)

★★★★☆

4.4 Average, 5 Review(s)

Availability: In Stock

Brand	Apple
Model Name	iPhone 12 Mini
Wireless Carrier	Unlocked for All Carriers
Operating System	iOS 14
Cellular Technology	5G
Memory Storage Capacity	64 GB
Connectivity Technology	Wi-Fi
Color	Black
Screen Size	5.4 Inches
Wireless network technology	GSM, Wi-Fi, CDMA

About this item

This phone is unlocked and compatible with any carrier of choice on GSM and CDMA networks (e.g. AT&T, T-Mobile, Sprint, Verizon, US Cellular, Cricket, Metro, Tracfone, Mint Mobile, etc.).

Функциональные возможности Django

- **Forms** – возможность создавать формы для организации взаимодействия пользователя с моделями БД
- **Authentication/Authorization** – инструменты для регистрации, аутентификации и манипуляции аккаунтами пользователей
- **Site Administration** – Django имеет встроенный гибкий интерфейс панели администратора сайта
- Встроенный **ORM** - механизм для работы с моделями БД как с классами и объектами Python
- **URL routing** – система менеджмента адресов и ссылок
- **Template engine** – HTML шаблонизатор
- **Internationalization** – инструменты для создания многоязычных приложений
- **Возможность подключения внешних инструментов**

Первое приложение на Django

Начало работы

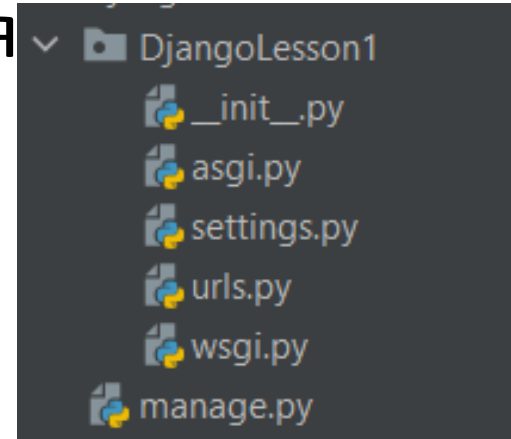
- `pip install django`
- Создаём джанго – проект:

`django-admin startproject projectName`

Что произошло?

manage.py – скрипт для управления проектом. При его запуске создаётся «приложение».

Сайт может состоять из нескольких частей (блог, вики, FAQ и т.д.), философия Django заключается в том, чтобы разделить их на отдельные приложения



Что произошло?

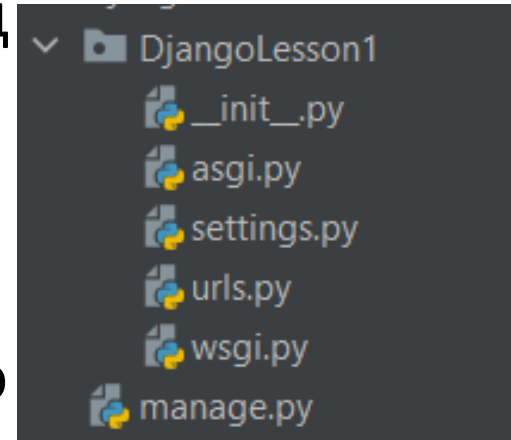
DjangoLesson1 – папка с приложением под именем DjangoLesson1.

init.py – скрипт, выполняемый при запуске проекта

asgi.py и **wsgi.py** – файлы для корректного подключения к серверу (как правило, код внутри них не изменяется)

settings.py – файл с настройками проекта

urls.py – файл для отслеживания URL адресов. В нём привязываются приложения к ссылкам



Settings.py

BASE_DIR – путь к проекту

SECRET_KEY – секретный ключ, необходимый для доступа к сайту

DEBUG - режим отладки в браузере вкл/выкл

ALLOWED_HOST – список разрешенных хостов (адресов), где можно опубликовать сайт

INSTALLED_APPS – список встроенных в проект приложений. Список будет расширяться

MIDDLEWARE – библиотеки, используемые в нашем проекте (промежуточное ПО)

Settings.py

ROOT_URLCONF – основной URL – файл проекта

TEMPLATES – шаблоны нашего проекта

WSGI_APPLICATION – WSGI приложение проекта
(Web-Server Gateway Interface)

DATABASES – база данных приложения

Далее идут валидаторы, настройки языка,
часовой пояс, адрес папки static и т.д.

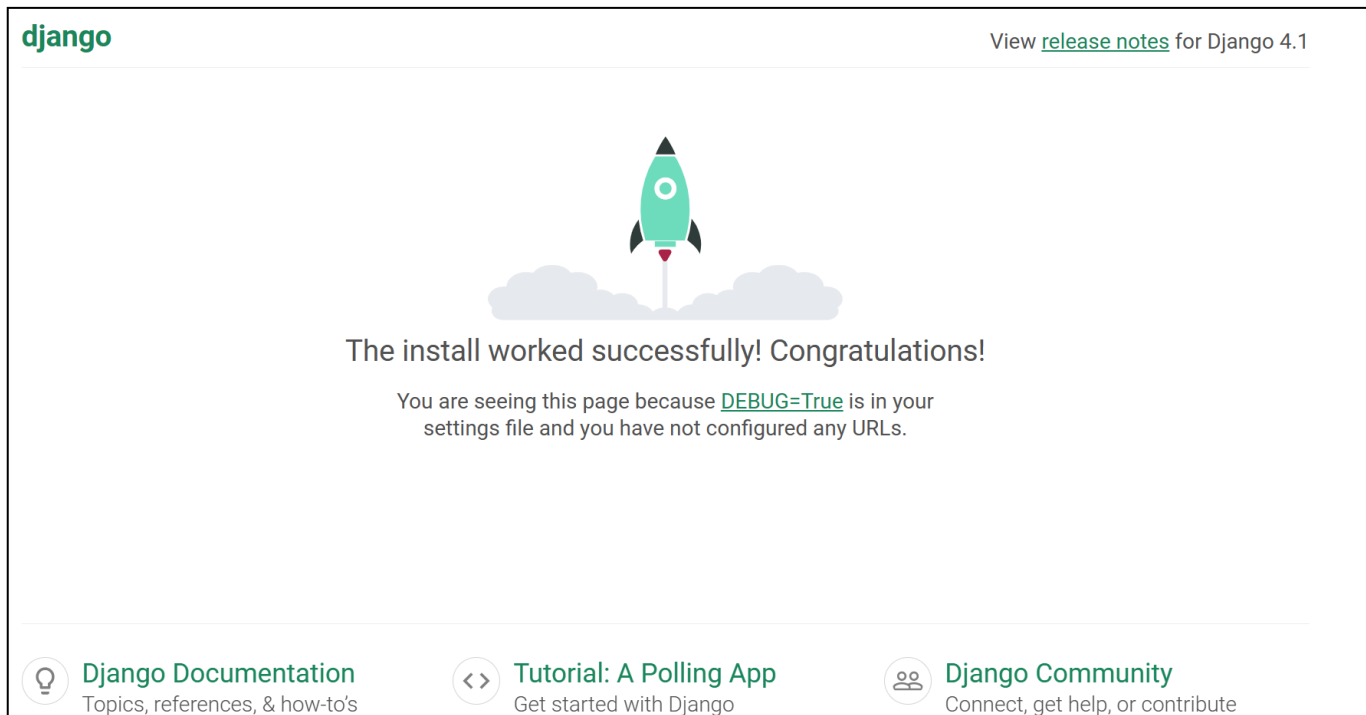
Запуск сервера

- Запускаем сервер:

python projectName/manage.py runserver

Адрес: 127.0.0.1:8000

ctrl+c - для откл.



Django - приложения

Приложения в Django отвечают за различные модули вашего веб-сайта. Каждое приложение имеет 3 обязательных файла:

- файл для работы с базой данных;
- шаблоны различных страниц или блоков, что будут отображены на сайте;
- файл проверяющий адрес ссылки и открывающий нужный **HTML** шаблон.

Создание приложения

Создадим приложение `main` – которое будет отвечать за основную часть сайта (страницы основная, контакты, о нас и т.д.)

В терминале пишем:

```
python projectName/manage.py startapp main
```

Зайдите в `projectName/settings.py` и добавьте к списку `INSTALLED_APPS` новое приложение:

```
'main'
```

Файлы приложения

main – папка с приложением под именем main :

init.py – скрипт при запуске

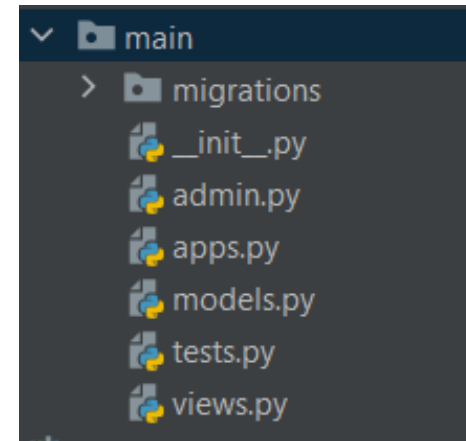
admin.py – настройки админ. Части

apps.py – настройки приложения main

models.py – модели приложения

tests.py – тесты для проверки

views.py - отображения



Добавление новых страниц приложения

Зайдите в **main/views.py** и запишите код:

```
from django.shortcuts import render
from django.http import HttpResponse

def index(request):
    return HttpResponse("<h1>Тестовый текст </h1>")
```

Создайте в папке **main** файл **urls.py**. И запишите туда код:

```
from django.urls import path
from . import views

urlpatterns = [
    path('', views.index)
]
```

Добавление новых страниц приложения

Зайдите в файл **projectName/urls.py** и добавьте туда `import include` и укажите путь к `main.urls`

```
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('main.urls'))
]
```

Добавление новых страниц приложения

Что произошло:

- 1) Мы привязали к главному файлу `urls.py` файл `main/urls.py`, при вызове страницы по корневому адресу (на главную страницу)
- 2) В файле `views` Мы описали отображение для главной страницы (`index`)
- 3) Итоговый путь запроса:

`ProjectName/urls.py` → `main/urls.py` → `views.py`

Задание

Добавьте 2 страницы: **about** и **contacts** с содержимым вида «страница про нас», «контакты»

Вывод целых веб-страниц

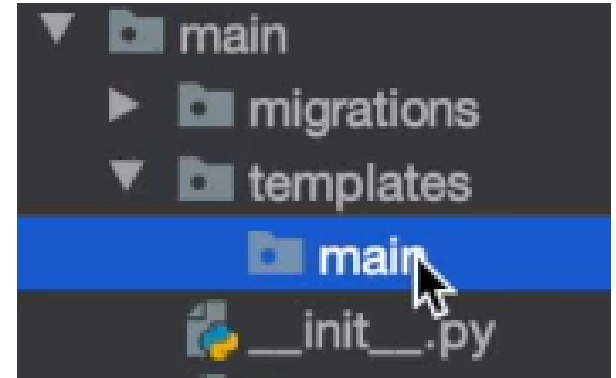
Для вывода больших фрагментов HTML- кода, нужно использовать функцию **render** Вместо **HttpResponse**.

Для импорта этой команды, не забудьте дописать код в Файле **Main/views.py**:

```
from django.shortcuts import render
```

Работа с шаблонами

Создайте папку **templates**.
внутри папки приложения.
(в ней будут храниться HTML)



Документация требует создать внутри папки **templates** еще одну папку с именем приложения

Если у вас будет много приложений, у все них будет папка **templates**. **Django** будет по итогу объединять все эти папки, и если в нескольких папках **templates** у разных приложений будет файл с именем, например, **index.html**, то Django заменит файлы с одинаковыми названиями на один. Сайт будет работать не корректно!

Работа с шаблонами

Создайте файл index.html



Напишите любой код в этом файле:

(!+Tab), измените **title** и **body**

Чтобы отобразить этот шаблон на сайте, нужно описать его в файле **views.py**.

Заменим HttpResponse на render:

```
def index(request):  
    return render(request, 'main/index.html')
```

Задача

Создайте несколько html-страниц с информацией о вас:

About – текстовое поле с кратким описанием

Contacts- небольшая таблица с данными

Content - текст + картинка

Добавьте их к views и протестируйте работоспособность страниц

**Материал для
самостоятельного изучения**

Материал для самостоятельного изучения

Преридер:

Что такое интернет, типы запросов

Постридер:

Типы архитектур MVT vs MVC