

# Django

ВВЕДЕНИЕ

# Django - фреймворк

- Что такое фреймворк?

# Что такое фреймворк?

## Ответ

The image shows a Google Translate interface. On the left, the source text 'framework' is entered in the English input field. The source language is set to 'английский' (English) and the target language is 'русский' (Russian). Below the input field, the phonetic transcription 'ˈfrām ˌwɜrk' is visible. On the right, the translated text 'фреймворк' is shown in the Russian output field. The target language is set to 'русский' (Russian). Below the output field, the phonetic transcription 'freymvork' is visible. A blue 'Перевести' (Translate) button is located between the two input fields. The interface also includes a swap button, a 'Предложить исправление' (Suggest correction) button, and various utility icons like a star, copy, and share.

английский русский немецкий Определить язык

↔ русский английский украинский **Перевести**

framework ×

9/5000

фреймворк ✓

☆ 📄 🔊 🔄

✎ Предложить исправление

ˈfrām ˌwɜrk

freymvork

# Фреймворк это



# Фреймворк Django

## Meet Django

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

1. Высокоуровневый Python-фреймворк для создания веб-приложений
2. OpenSource
3. Строится из одного или нескольких приложений
4. MVC'
5. DRY
6. Собственный ORM
7. Собственная система шаблонов
8. Кеширование
9. Батарейки включены

# Сайт [djangoproject.com](https://djangoproject.com)

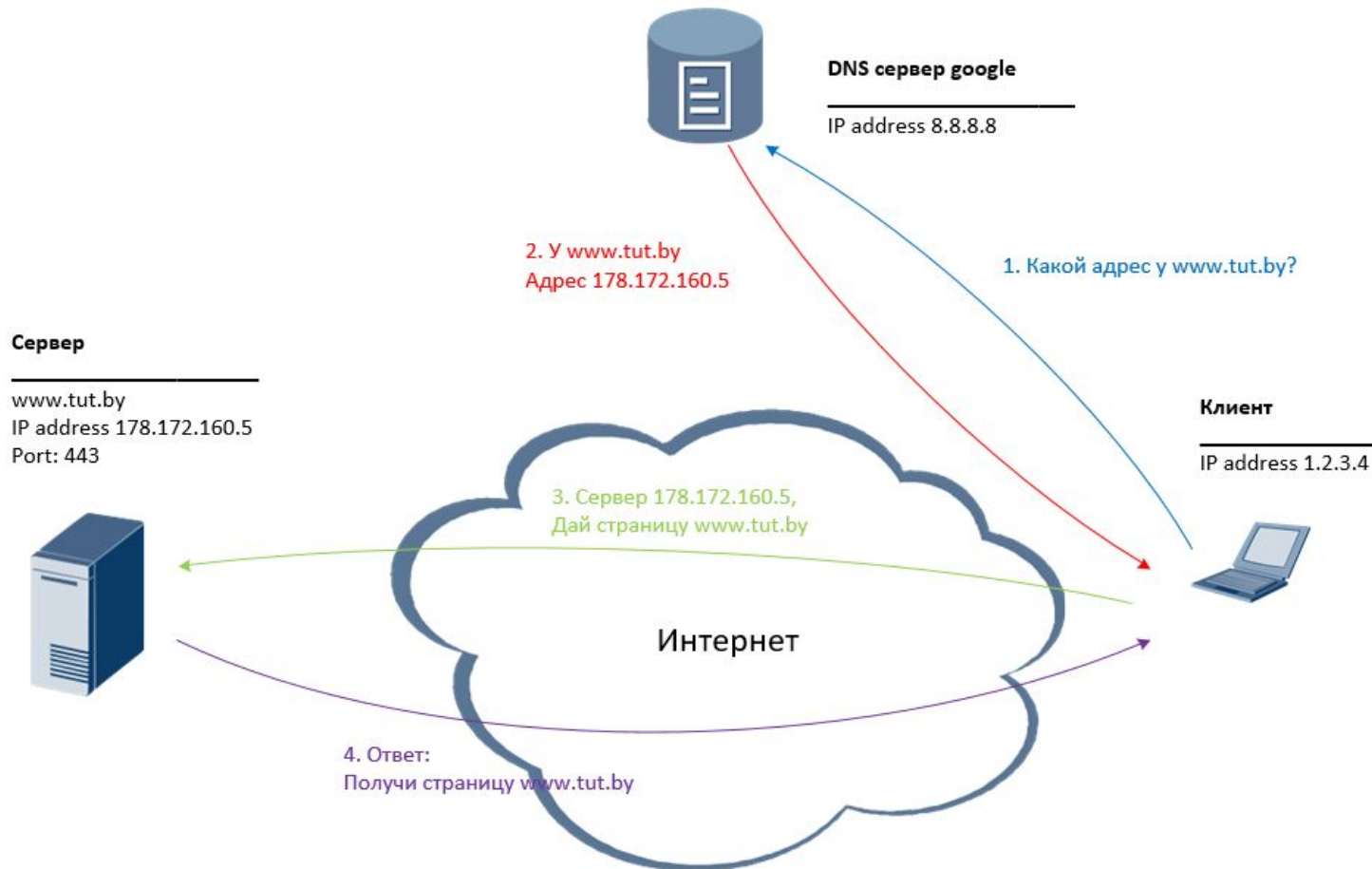
Основные разделы:

- Документация (туториалы, ари-гайды...)
- Страница загрузки фреймворка – карта релизов
- Гугл ищет лучше... :)

# Основные понятия клиент-серверной работы в веб (ака Ресторан – Гость)

- Клиент – инициирует соединение
- Клиент – формирует запрос
- Клиент – получает ответ
  
- Сервер – ожидает соединения клиентов
- Сервер – принимает соединение клиентов
- Сервер – принимает запрос
- Сервер – готовит ответ
- Сервер – отправляет ответ клиенту

# Основные понятия клиент-серверной работы в веб – Реальный пример



## Запрос

General  
Request URL: https://www.tut.by/  
Request Method: GET  
Status Code: 200  
Remote Address: 178.172.160.5:443  
Referrer Policy: no-referrer-when-downgrade

## Ответ

```
<!DOCTYPE html>
<html class="no-js desktop lang-rus" xmlns:fb="http://ogp.me/ns/fb#" lang="ru">
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-dns-prefetch-control" content="on" />
  <link rel="preconnect" href="https://img.tyt.by/" />
  <link rel="dns-prefetch" href="https://img.tyt.by/" />
  <link rel="dns-prefetch" href="https://news.tut.by/" />
  <link rel="dns-prefetch" href="https://kupi.tut.by/" />
  <link rel="dns-prefetch" href="https://img.tam.by/" />
  <link rel="dns-prefetch" href="https://reklama.tut.by/" />
  <link rel="dns-prefetch" href="https://bugaga.tut.by/" />
  <link rel="dns-prefetch" href="https://preved.tut.by/" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />
  <meta name="viewport" content="width=device-width" />
  <title>Белорусский портал TUT.BY. Новости Беларуси и мира</title>
  <meta name="referrer" content="always" />
  <link rel="shortcut icon" href="https://www.tut.by/favicon.ico" />
  <link rel="stylesheet" type="text/css" href="/css/by5/v5.2.4-r25021.css?QTYzMT0hRTYmNR
  <meta name="description" content="Последние новости Беларуси и зарубежья. Быстрый поиск
  <meta name="google-site-verification" content="f-6GnRRAR7oTzx-aPcTQ546aRHeu2ynaXw64h6PI
  <meta name="yandex-verification" content="72b4676d68aa83ff" />
```



# Создание проекта

- Проект – совокупность приложений
- Установка фреймворка
  - `pip install Django`
  - `python -m django --version`
- Создание проекта
  - `django-admin startproject mysite`

```
mysite/  
  manage.py  
  mysite/  
    __init__.py  
    settings.py  
    urls.py  
    wsgi.py
```

# Запуск сервера разработки

```
Performing system checks...
```

```
System check identified no issues (0 silenced).
```

```
You have unapplied migrations; your app may not work properly until they are applied.
```

```
Run 'python manage.py migrate' to apply them.
```

```
August 16, 2018 - 15:50:53
```

```
Django version 2.1, using settings 'mysite.settings'
```

```
Starting development server at http://127.0.0.1:8000/
```

```
Quit the server with CONTROL-C.
```

# Создание приложения

- `python manage.py startapp polls`

```
polls/  
  __init__.py  
  admin.py  
  apps.py  
  migrations/  
    __init__.py  
  models.py  
  tests.py  
  views.py
```

# Концепция MVC'

- Django использует видоизмененный MVC шаблон при проектировании приложений – MVT
- Model -> Model
- View -> Template
- Controller – View

где:

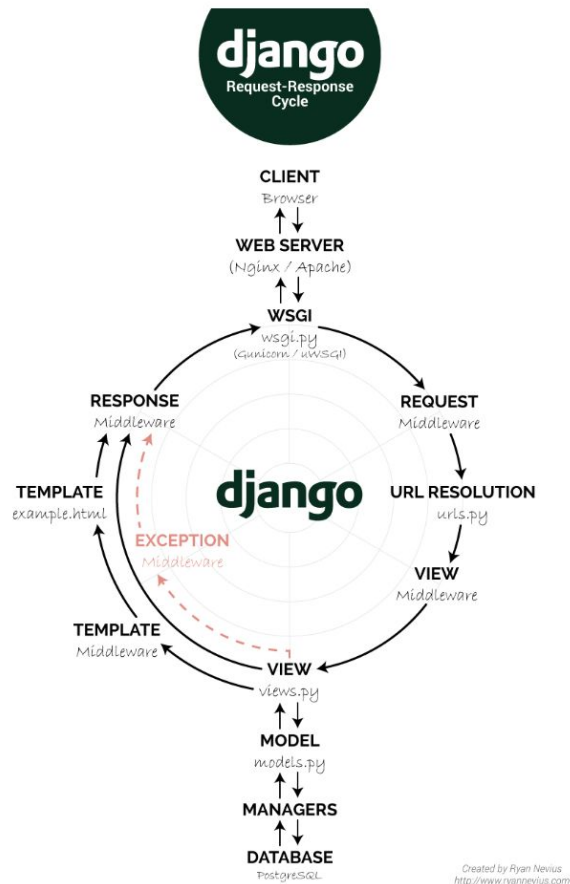
**Model** – описывает и предоставляет данные и реагирует на команды контроллера, изменяя своё состояние

**View (Template)** - отвечает за отображение данных модели пользователю, реагируя на изменения модели

**Controller(View)** - интерпретирует действия пользователя, оповещая модель о необходимости изменений

# Назначение view и url файлов

- На примере простого “Hello World”



# Как обрабатывается запрос?

## How Django processes a request¶

(<https://docs.djangoproject.com/en/2.1/topics/http/urls/#how-django-processes-a-request>)

When a user requests a page from your Django-powered site, this is the algorithm the system follows to determine which Python code to execute:

Django determines the root URLconf module to use. Ordinarily, this is the value of the `ROOT_URLCONF` setting, but if the incoming `HttpRequest` object has a `urlconf` attribute (set by middleware), its value will be used in place of the **`ROOT_URLCONF` setting**.

Django loads that Python module and looks for the variable `urlpatterns`. This should be a Python list of `django.urls.path()` and/or `django.urls.re_path()` instances.

Django runs through each URL pattern, in order, and stops at the first one that matches the requested URL.

Once one of the URL patterns matches, Django imports and calls the given view, which is a simple Python function (or a class-based view). The view gets passed the following arguments:

An instance of `HttpRequest`.

If the matched URL pattern returned no named groups, then the matches from the regular expression are provided as positional arguments.

The keyword arguments are made up of any named parts matched by the path expression, overridden by any arguments specified in the optional `kwargs` argument to `django.urls.path()` or `django.urls.re_path()`.

If no URL pattern matches, or if an exception is raised during any point in this process, Django invokes an appropriate error-handling view. See Error handling below.