

A Noob's guide to Django

- By Alisha Mohanty

What we will cover?

- > What is Django?
- > Django's Features
- > Architecture
- > Installation Process

continued..

- > How to create a Project
- > Project Layout and File Structure
- > Pros and Cons
- > Resources to continue learning.

What is Django?

What is Django?

- Django is basically a **high-level Python** web application framework that enables the rapid development of web applications.
- It is a backend framework used to **resolve problems of connectivity** with databases, other server problems, **SEO solutions**, etc so that a web developer need **not write the same code for the similar modules**.

What is Django?

- Helps for **easier import of applications** according to need and thus it **concentrate more on the unique application** of website rather than dealing with all these backend problems.

History of Django

History of Django

- Django was created by two web developers **Adrian Holovaty** and **Simon Willison** working at the Lawrence Journal-World Newspaper in **2003**.
- It was released publicly as a BSD license in July **2005**.
- The framework was named after guitarist **Django Reinhardt**.

Prerequisites for Learning Django

Prerequisites for Learning Django

- > Some prior knowledge of Python
- > Basic idea of Command Line Interfaces
(Powershell for Windows / Terminal for macOS/Linux)

P.S Learning Django only requires a basic knowledge of these technologies, therefore a fresher in Django can learn the Django framework easily.

Why learn Django?

Why learn Django?

1. Django has evolved over time
2. Open-Source Technology
3. One of the World's Best Community Software
4. Django has lots of free made apps

Features of Django

Features of Django

1. Written in Python

Python is easy to read, powerful, and used in implementation of scientific computation.

1. Excellent Documentation

If we compare Django with other open source technologies, it offers the best documentation in the market.

Features of Django

3. Highly Scalable

Django is written by those experienced programmers from scratch without using [any python library existing](#) other than what the developers created themselves making it scalable.

4. Provides Rapid Development

No prior expert backend knowledge needed to make a fully functional website. We will also not create separate server files to design the database and connect the same while also making another file for transferring data to and from the server.

Features of Django

5. Thoroughly Tested

Withstand the dynamic changes happening in the industry.

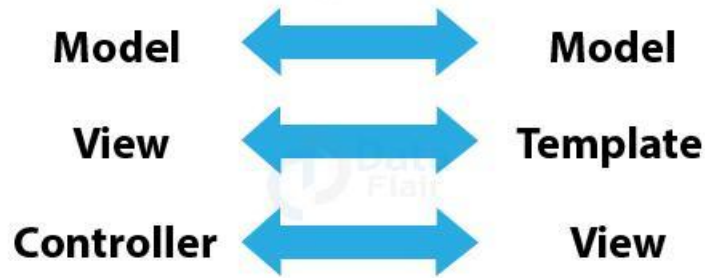
Django has been in the industry for more than a decade and is still a popular technology which is beating frameworks like Laravel(PHP) in their own game.

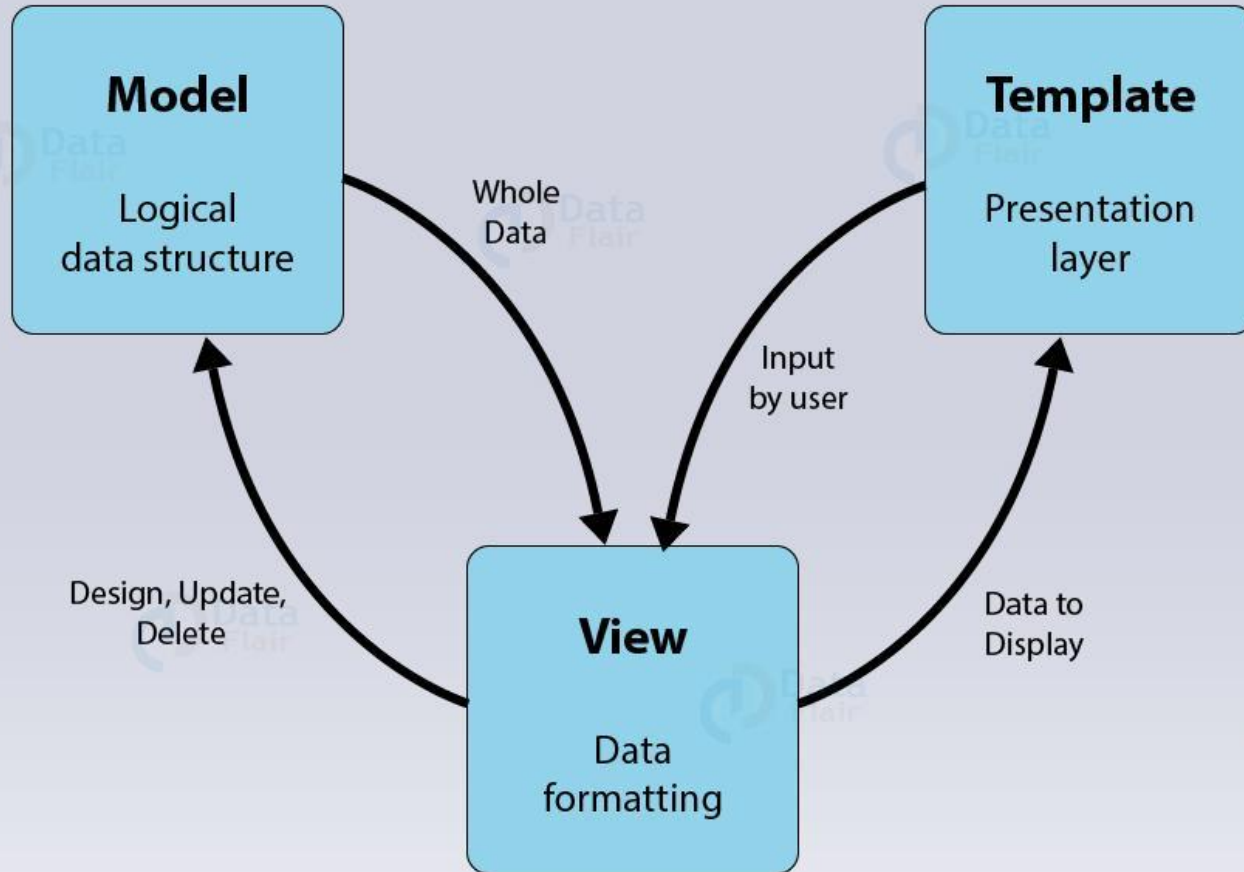
Architecture

Architecture

Follows **Model-Template-View** Architecture.

MTV instead of MVC in **django**





Model

- Model **contains the logical file structure** of the project and is the middleware & data handler between database and view.
- The Model provides a **definition of how the data** formats as coming from the view so, it stores in the database and vice-versa.
- It **retrieves information from the database** transfers to the view in the displayable format.

Template

- Template's main goal is to **keep everything that browser renders**.
- It **showcase model's data** that's coming from the server in different parts while integrating the same when the user interacts with the website.

View

- The View in this **MTV architecture** is **formatting the data** via the model.
- In turn, it **communicates to the database** and that data which transfer to the template for viewing.

Installation

Installation Process

1. Install **python** and **pip**

Make sure you have **python** and **pip** installed in your computer.

In order to check that in your windows shell write

```
python --version
```

```
pip --version
```



```
PS C:\Users\HP\documents\dataflair> python --version
```

```
Python 3.7.2
```

```
PS C:\Users\HP\documents\dataflair> pip --version
```

```
pip 19.0.3 from c:\users\hp\appdata\local\programs\python\python37\lib\site-packages\pip (python 3.7)
```

```
PS C:\Users\HP\documents\dataflair>
```

Installing Django

2. Install, create and start Virtual Environment

```
pip install virtualenv
```

```
virtualenv env
```

```
env\scripts\activate
```

Installing Django

— — —

3. **Install Django**

```
pip install django
```

— * —

How to create and run a Django Project

Create & Run Django Project

— — —

Create Django Project

```
django-admin startproject project-name
```

Run the Test-Server

```
python manage.py runserver
```

Directory Structure of Project

```
(myvenv) → my_first_project ls
manage.py my_first_project
(myvenv) → my_first_project cd my_first_project
(myvenv) → my_first_project ls
__init__.py settings.py urls.py wsgi.py
(myvenv) → my_first_project
```

After starting the server

File Edit View Search Terminal Help

(myvenv) → my_first_project python manage.py runserver

Watching for file changes with StatReloader

Performing system checks...

System check identified no issues (0 silenced).

You have 17 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.

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

July 23, 2019 - 13:51:41

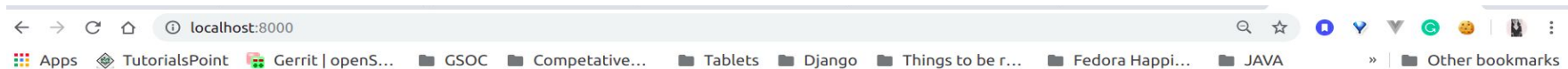
Django version 2.2.3, using settings 'my_first_project.settings'

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

Quit the server with CONTROL-C.

□

How it looks in your browser



django

View [release notes](#) for Django 2.2



The install worked successfully! Congratulations!

You are seeing this page because `DEBUG=True` is in your settings file and you have not configured any URLs.



Django Documentation
Topics, references, & how-to's



Tutorial: A Polling App
Get started with Django



Django Community
Connect, get help, or contribute

Project Layout and File Structure

django Project Layout & Files Structure



manage.py

- Command Line utility of Django and this will be used to deploy, debug and test the project.
- **runserver** is a command to start the test server provided by Django framework and that is also one of the advantages of Django over other frameworks.
- **makemigrations** is the command for integrating project with files or apps that have been added to it.
- **migrate** actually add those migrations to the whole project.

`__init__.py`

- Tells python interpreter that the project Directory is a package.

`settings.py`

- This file contains the installed applications and middleware information which are installed on this Django project.
- Main settings file of the Django project.

urls.py

- urls.py file contains the project level URL information.
- urls.py file contains the project level URL information.

wsgi.py

- wsgi is still important though if you want to deploy the applications on Apache servers or any other server because Django is still backend and you will need its support with different servers.

How to create a Django App

Inside root directory

```
$ django-admin startapp application-name
```

Ex: **django-admin startapp demo**

Sample Directory Photo

```
(myvenv) → my_first_project ls
db.sqlite3  demo  manage.py  my_first_project
(myvenv) → my_first_project cd demo
(myvenv) → demo ls
admin.py  apps.py  __init__.py  migrations  models.py  tests.py  views.py
(myvenv) → demo □
```

Pros & Cons

Pros

- Implemented in Python
- Better CDN Connectivity and Content Management
- Fast Processing
- Scalable

Cons

- Django is monolithic
- Not suitable for smaller projects
- Uses regular expression for URLs

Resources to continue learning about Django

1. <https://tutorial.djangogirls.org/en/>
2. <https://www.djangoproject.com/>
3. <https://www.fullstackpython.com/django.html>

— — —

Contact Me?

- **Twitter**(@freaky_mortal)
- alishamohanty1416@**@**gmail.com

Thank You