

1. Introduction to Django :

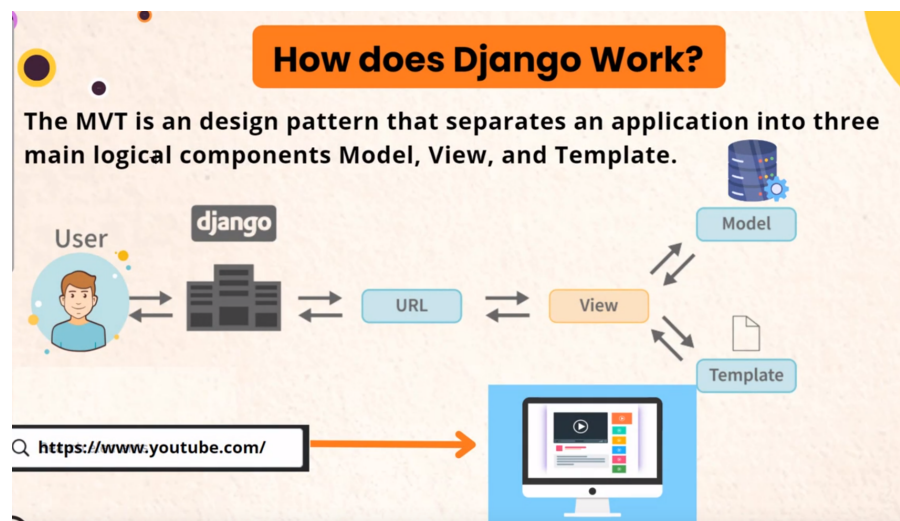
- a. **What is Django** → Web Application Framework written in Python based on MVT(Model View Template)
- b. **Why Django** → Batteries (built-in functionality) Included Framework.

Excellent Documentation and Scalability,
Top MNC & companies use Django. E.x: YouTube, Spotify, Instagram, Dropbox, Bitbucket etc.

It's also used for integrating Web Scraping, Machine Learning, Image Processing, Scientific Computing etc advance stuff.

- c. **Unique features** → Admin interface, ORM (Object Relational Mapping) URL Routing, Template System, Form Handling, Security Features Scalability

- d. **How Django works** →



MVT : Model, View, Template

Model → The data we want to present, like from a Database.

View → A request handler that returns the relevant template and content - based on the request from the user.

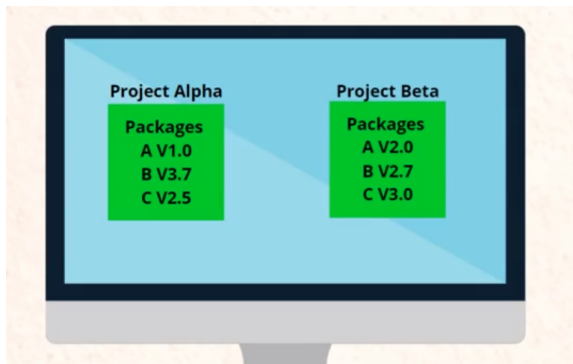
Logical works & Decision maker

Template → how data should be presented to the user. It may consists of HTML, CSS, JS mixed with Django template language.

2. Python and Vs Code Installation

3. Introduction to Virtual Environment : Global and Virtual environment.

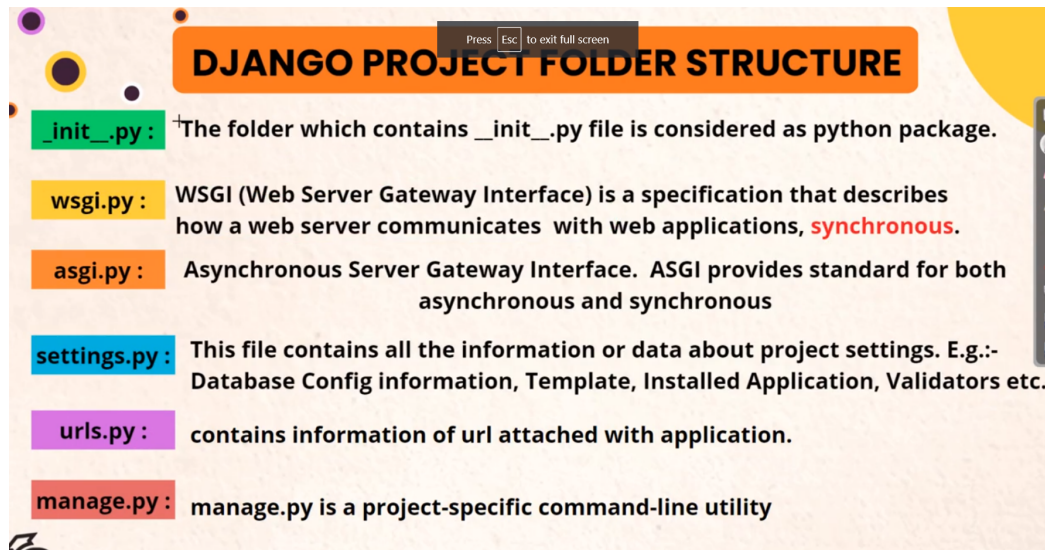
A virtual environment is a tool that helps to keep dependencies required by different projects, separate by creating isolated python virtual environment for them. Every time unique virtual environment for each project within the same system to avoid version clash or dependencies or re-install/uninstall :



4. Django Installation : With Globally & Virtual environment both.

5. Creating a Django Project :

6. Project Directory Structure :



The slide is titled "DJANGO PROJECT FOLDER STRUCTURE" in a large orange box at the top. Below the title, there is a list of files and their descriptions, each with a colored background for the file name:

- _init_.py :** The folder which contains _init_.py file is considered as python package.
- wsgi.py :** WSGI (Web Server Gateway Interface) is a specification that describes how a web server communicates with web applications, **synchronous**.
- asgi.py :** Asynchronous Server Gateway Interface. ASGI provides standard for both asynchronous and synchronous
- settings.py :** This file contains all the information or data about project settings. E.g.:- Database Config information, Template, Installed Application, Validators etc.
- urls.py :** contains information of url attached with application.
- manage.py :** manage.py is a project-specific command-line utility

A small overlay at the top center says "Press Esc to exit full screen".

A Project contains → Multiple applications.

Multiple Applications → A Project

7. Working with urls.py and views.py :

8. Creating An App :