

LEARN DJANGO



Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Django makes it easier to build better web apps quickly and with less code.

Note – Django is a registered trademark of the Django Software Foundation, and is licensed under BSD License.

History of Django

2003 – Started by Adrian Holovaty and Simon Willison as an internal project at the Lawrence Journal-World newspaper.

2005 – Released July 2005 and named it Django, after the jazz guitarist Django Reinhardt.

2005 – Mature enough to handle several high-traffic sites.

Current – Django is now an open source project with contributors across the world.

Django – Design Philosophies

Django comes with the following design philosophies –

Loosely Coupled – Django aims to make each element of its stack independent of the others.

Less Coding – Less code so in turn a quick development.

Don't Repeat Yourself (DRY) – Everything should be developed only in exactly one place instead of repeating it again and again.

Fast Development – Django's philosophy is to do all it can to facilitate hyper-fast development.

Clean Design – Django strictly maintains a clean design throughout its own code and makes it easy to follow best web-development practices.

Advantages of Django

Here are few advantages of using Django which can be listed out here –

Object-Relational Mapping (ORM) Support – Django provides a bridge between the data model and the database engine, and supports a large set of database systems including MySQL, Oracle, Postgres, etc. Django also supports NoSQL database through Django-nonrel fork. For now, the only NoSQL databases supported are MongoDB and google app engine.

Multilingual Support – Django supports multilingual websites through its built-in internationalization system. So you can develop your website, which would support multiple languages.

Framework Support – Django has built-in support for Ajax, RSS, Caching and various other frameworks.

Administration GUI – Django provides a nice ready-to-use user interface for administrative activities.

Development Environment – Django comes with a lightweight web server to facilitate end-to-end application development and testing.

As you already know, Django is a Python web framework. And like most modern framework, Django supports the MVC pattern. First let's see what is the Model-View-Controller (MVC) pattern, and then we will look at Django's specificity for the Model-View-Template (MVT) pattern.

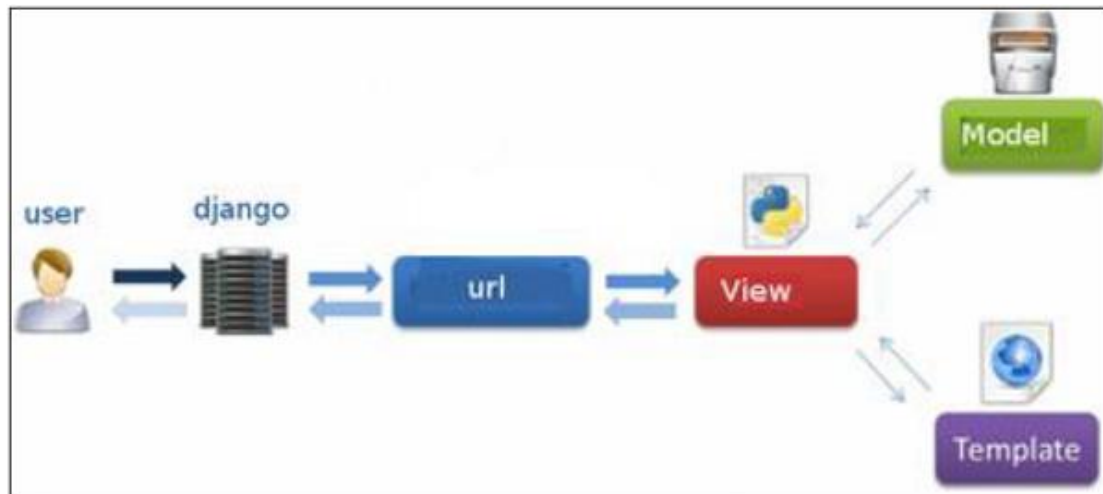
MVC Pattern

When talking about applications that provides UI (web or desktop), we usually talk about MVC architecture. And as the name suggests, MVC pattern is based on three components: Model, View, and Controller. Check our MVC tutorial here to know more.

DJANGO MVC - MVT Pattern

The Model-View-Template (MVT) is slightly different from MVC. In fact the main difference between the two patterns is that Django itself takes care of the Controller part (Software Code that controls the interactions between the Model and View), leaving us with the template. The template is a HTML file mixed with Django Template Language (DTL).

The following diagram illustrates how each of the components of the MVT pattern interacts with each other to serve a user request –



The developer provides the Model, the view and the template then just maps it to a URL and Django does the magic to serve it to the user.