

DJANGO THE PYTHON WEB FRAMEWORK

¹Prof. B Nithya Ramesh, ²Aashay R Amballi, ³Vivekananda Mahanta

Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC

Department of Master of Computer Applications, NHCE, Bangalore, India

Abstract: This article gives you an overview of why to choose Django over any other framework. Django is a high-level Python framework by using it the development speed will be faster and cleaner. Django is built by the much more experienced people, so it takes care of the web development process in easier way. With takes care of much of the hassle of web development, so we can focus on writing our app without needing to reinvent the wheel. Django is also open source and free and it follows the principle of “Don’t Repeat Yourself”.

Keyword: Framework, Python, Web Development, Model-View-Template.

I. INTRODUCTION

We’ve all made sandwiches using bread at some point of time in life. Frameworks in web development are like the bread, they are the base on which sandwiches, here web applications, web services and software are built. Just like you have umpteen options in making a sandwich regarding choice of bread, fillings used, flavours and so on. We get number of options when it comes to web application development frameworks – including the correct framework, specific language, correct libraries and so on.

There are many advantages of python such as it charts in the recent years over other programming languages and widely used by programmers. Python has undergone a drastic change since its release 25 years ago as many add-on features are introduced. Many of the software development companies prefer language because of its versatile features and clean and neat programming codes. Python has many characteristic features like it is Interactive, Interpreted, Modular, Dynamic, Object oriented, Portable and the list goes so on.

There is much to be said about the advantages of Python. It has been the most popular development language in the recent years and it continues to be favourite among many skilled developers. But let’s narrow our focus to why Django framework in specific.

II. MODULE SPECIFICATION

A. Reasons for “Why Django is better than any other framework!”

1. Benefit to the customer:

In case something requires you to change your development team mid-way through your project , Django allows you to find a new team that can comprehend the projects architecture with very little effort. This can save a lot of time and money

2. Benefit to the developer:

Even if the project is developed by one team from start to finish, using Django makes the development process remarkably quickly, from idea to release through to launch and production. With transparent, clean code, development can be both efficient and effective. And we all know that time = money.

This is why Django is considered “the web framework for perfectionists with deadlines.” It allows any developer to quickly grasp the project’s structure and gives them the tools they need to implement any idea and develop it properly, and in record time.

3. Django is Fast:

This has been designed in a way to help the developers make an application as fast as possible. From idea, production to release, Django helps in making it both cost effective and efficient. Thus it becomes an ideal solution for developers having a primary focus on deadlines.

4. Django is Secure:

When you are doing it in Django, it is ensured that developers don’t commit any mistakes related to security. Some of the common mistakes include SQL injection, cross-site request forgery, clickjacking and cross-site scripting. To manage effectively usernames and passwords, the user authentication system is the key.

5. Django is flexible:

Content management, scientific computing platforms, and even big organizations, all these aspects are very efficiently managed by the use of Django.

B. Contents:

1. The very first step:

Before we use Django, Python should be installed, and if we have no background knowledge on Python then Python is the one which we should learn first. Eventually being a python Web framework Django requires Python.

2. Install Django:

We’ll get three easy options to install Django

- Installing an official release. This is the best approach for most users.
- Installing a version Django provided by our operating system distribution.
- Install the latest development version. This option is for enthusiasts who want the latest-and-greatest features and aren’t afraid of running brand new code. You might encounter new bugs in the development version, but reporting them helps the development of Django. Also, releases of third-party packages are less likely to be compatible with the development version than with the latest stable release.

3. The Model-View-Controller design pattern:

- MVC has been around as a concept for a long time, but has seen exponential growth since the advent of the Internet because it is the best way to design client-server applications. All of the best web frameworks are built around the MVC concept. At the risk of starting a flame war, I contest that if you are not using MVC to design web apps, you are doing it wrong. As a concept, the MVC design pattern is really simple to understand:
- The **model(M)** is the representation of our data. Where it is not an actual data, but an interface to the data. The model will help us in pulling the data from the database.
- The **view(V)** is going to be the front end (what we see in the screen). It is the presentation layer of our model.
- The **controller(C)** is which will control the flow between model and view. Controller is basically the backend logic (Business Logic) which we write.
- Django also follows the MVC pattern similarly, but it follows its own logic while implementing. Because the “C” is handled by the framework itself and most excitement in Django happens in the models, templates and views. As I mentioned it will it’s own logic while implementing, which means Django will follow **MTV** pattern where,
- **M stands for “Model”** which is normally the data accessing layer. This layer contains about data only. how to validate it, which behaviours it has, and the relationships between the data.
- **T stands for “Template”** which is front end. It is the presentation layer.

- **V stands for “View”** which will control the flow between model and template. View is layer where we write our business logic.

4. Django Forms:

Django Forms will come handy when we want to publish the data in our Web application. In HTML, a form is a collection of elements inside `<form>...</form>` that allow a visitor to do things like enter text, select options, manipulate objects or controls, and so on, and then send that information back to the server.

Using the Django form, it will takes care of a number of common form related tasks such as:

- I. Display an HTML form with automatically generated form widgets.
- II. Check submitted data against a set of validation rules.
- III. Redisplay a form in the case of validation errors.
- IV. Convert submitted form data to the relevant Python data types.

5. Django admin interface:

One of the most best and powerful parts of Django is the automatic admin interface. It reads metadata from models to provide a quick, model-centric interface where trusted users can manage content on your site. The admin's recommended use is limited to an organization's internal management tool. It's not intended for building your entire front end around.

The admin has many hooks for customization, but beware of trying to use those hooks exclusively. If you need to provide a more process-centric interface that abstracts away the implementation details of database tables and fields, then it's probably time to write your own views.

6. Django ORMs:

ORM stands for Object-Relational mapper, it is library that automates the transfer of data stored in relational databases tables into objects that are more commonly used in application code.

Why are ORMs useful?

ORMs provide a high-level abstraction upon a relational database that allows a developer to write Python code instead of SQL to create, read, update and delete data and schemas in their database. Developers can use the programming language they are comfortable with to work with a database instead of writing SQL statements or stored procedures.

C. Scope of Python and Django framework:

The state of web-development is in a constantly changing and we can never surely say when a game changer technology would come and change the current trends entirely, so it would be wise to take a peek into any new technology that is gaining enough momentum and stay updated with the current trends.

Thousands of websites are currently using Django, from daily newspapers to social media and sharing sites to major foundations and non profits. Since Django was originally developed for use in the newsroom, it's no surprise that major daily publications like the Washington Post and The Guardian rely on it. Start ups like Eventbrite and Disqus have relied on Django to scale quickly, while social media behemoths like Instagram and Pinterest have used it to power their dynamic web apps.

But, having said that, python and Django are not going anywhere in the next 10–15 years. Let's talk facts:

Python:

- Python is one of the most used languages in past few years, with major tech giants are its users.
- It also has a huge open-source fanbase, currently ranking 5th in the github repository trends. Also, according to Google Trends, economically developed countries - US, China and South Korea - are its top-3 users, which further strengthens its position.

- Python has a large number of web-frameworks, including scalability-proven and time-tested frameworks like Django, Flask and Pyramid.

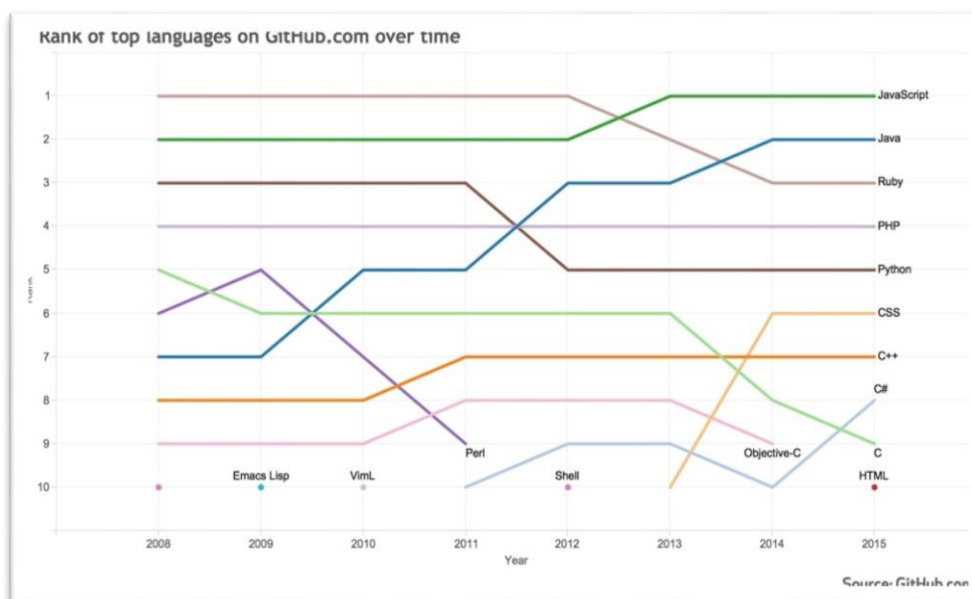


Figure 1.0

Django:

- Django is the most popular and extensive among all of the python frameworks. It is good for developing complex applications with many individual parts. Its major users include Pinterest, Instagram, Mozilla, The Washington Times, Disqus, Bitbucket, NASA and Nextdoor.
- The popularity of the framework is also on a constant rise. Check the trends here: [Django CSRF Usage Statistics](#)
- Surely, Django is a little slow in incorporating new trending feature like Job Queues (which can be fulfilled with 3rd party libraries like Celery), but the explicitness of the framework separates it from its competitors like RoR and Laravel, which rather prefers the automagic approach. Explicitness is enjoyed by many programmers who wants to know what is happening underneath the surface.
- Django also can connect to the large number of third party applications that exist, further accelerating an already quick development process. For example, django-allauth is the best way to provide a "social login" (e.g., Twitter, Facebook, GitHub, etc.) option to users. And the Django Rest Framework is the best way to write REST APIs to connect Django with payment applications like Stripe and others.

III. CONCLUSION

Python is a must-have in any programmer's repertoire. It's clean, consistent, and features principles that not only make it a pleasure to learn and use, but also makes transitioning to other languages much faster.

Where as Django is a high-level web framework which was created for quick web project development. It delivers transparent and high-quality code writing, making it important for developers, and equally important for customers. There are many advantages and disadvantages as well. Disadvantages such as it doesn't support real time web application. The aim of Django is to take the developers from project conception to the launch platform quickly, and it really helps in that. It also takes away the burden of security issues like cross-site request forgery, SQL injection, cross-site scripting, and clickjacking. Developers can manage user accounts and passwords through a reliable user authentication system.

Summing up, things are looking bright for both Python(in general) and Django(in specific), and we will be in a safe place for the next 10–15 years (at least), if we are investing our time in Python(and Django) presently.

REFERENCES

- [1] <https://docs.djangoproject.com>
- [2] <https://steelkiwi.com>
- [3] <https://hackernoon.com>
- [4] <https://www.cabotsolutions.com>
- [5] <https://tutorial.djangogirls.org>
- [6] <https://developer.mozilla.org>
- [7] <https://en.wikipedia.org>
- [8] <https://www.quora.com>
- [9] <http://www.codingdojo.com>
- [10] <https://data-flair.training>