Frameworks

Python Programming language has many applications when it comes to implementation. Web development being one of the applications, there is a pressing need to understand which framework will serve your purpose in the best way possible.

What Are Frameworks In Python?

A framework is a collection of modules or packages which helps in writing web applications. While working on frameworks in python we don't have to worry about the low level details such as protocols, sockets or thread management.

Frameworks automate the common implementation of common solutions which gives the flexibility to the users to focus on the application logic instead of the basic routine processes.

Frameworks make the life of web developers easier by giving them a structure for app development. They provide common patterns in a web application that are fast, reliable and easily maintainable.

Lets take a look at a few operations involved in a web application using a web framework:

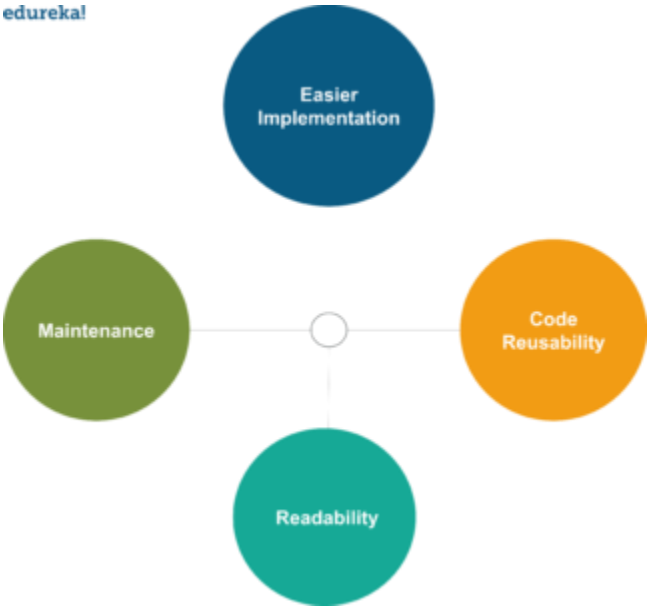
- **Url Routing** – Routing is the mechanism of mapping the URL directly to the code that creates the web page.
- **Input form handling and validation** – Suppose you have a form which takes some input, the idea is to validate the data and then save it.
- **Output formats with template engine** – A template engine allows the developers to generate desired content types like HTML, XML, JSON.
- **Database connection** – Database connection configuration and persistent data manipulation through an ORM.
- **Web security** – Frameworks give web security against cross-site request forgery aka CSRF, sql injection, cross-site scripting and other common malicious attacks.
- **Session storage and retrieval** – Data stored in the session storage gets cleared when the page session ends.

Advantages Of Frameworks

1. Open-source
2. Good documentation
3. Efficient
4. Secure
5. Integration

Why Use A Framework?

Frameworks make it easier to reuse the code for common HTTP operations. They structure the projects in a way so that the other developers with the knowledge of the framework can easily maintain and build the application.



Library vs Framework

Library	Framework
Less complex	More complex
When you call a method from a library, you are in control.	The control is inverted, the frameworks calls you.
A library performs specific operations.	A framework contains the basic flow, the rest is build by the user.

The key advantage of using a framework instead of a library is the flexibility. They are extensible and provides us with the necessary tools to extend its features.

When you have a library, you have to learn each functionality to perform certain operations. But with frameworks it becomes relatively easy due to the structured control of the flow. We just have to direct our operations using a certain operation using the functionalities already existing in the framework.

Although there are a lot of frameworks available in the market for web development, below are the top 5 frameworks in python.

Top 5 Frameworks In Python

Depending upon the sort of functionalities and key features they provide to the user, these are top 5 frameworks in python, both micro-frameworks and full-stack frameworks.

- Django
- Web2Py
- Flask
- Bottle
- CherryPy

Difference between a micro-framework and a full-stack framework?

Micro-framework	Full-stack framework
simple and easy to use	Complex and does the heavy lifting
url routing is RESTful often	Need not be RESTful
A good choice for small applications	Can be used to make any applications
Use WSGI and work through HTTP request/response.	Provide libraries, template engines, database management etc.

Django

Django is a free and open-source full-stack python framework, it includes all the necessary features by default.

It follows the DRY principle, which says don't repeat yourselves. Django uses its ORM mappers to map objects to database tables. An ORM or object relational mapper is a code library which helps you manipulate the data from a database using the object-oriented paradigm.

The main databases that django works on are PostgreSQL, MySQL, SQLite, Oracle. It can also work with other databases using the third party drivers.

Some of the exemplary features of django web frameworks are following:

- Authentication
- URL routing
- Template engine
- ORM
- Database Schema migrations

Django also follows MVC-MVT architecture,

MVC-MVT architecture:

MVT is slightly different from MVC, Although Django takes care of the controller part which is the code that controls the interactions between the model and the view. And the template is HTML file mixed with Django template language.

Developer provides the model, view and the template. User then maps it to the url and then the rest is done by django to serve it to the user.

Web2Py

Web2Py is open source, scalable and a full-stack framework . It does not support python 3 and comes with its own web based IDE which also includes a separate code editor, debugger and one click deployment.

Following are the features of Web2Py framework:

- It does not have any prerequisites for installation and configuration
- It has the ability to run on different platforms. Example- windows, mac, linux etc.
- Comes with an ability to read multiple protocols
- Web2Py provides data security against vulnerabilities like cross site scripting, sql injection and other malicious attacks.
- It has an error tracking mechanism through an error logging and ticketing system.
- Also has role based access control
- There is backward compatibility which ensures user oriented advancement without the need to lose any ties with earlier versions.

Flask

Flask is a micro-framework. It is lightweight and its modular design makes it easily adaptable to developer's needs. It has a number of out of the box features listed below:

- Built-in development server
- A fast debugger
- Integrated support for unit testing
- RESTful request dispatching
- Jinja2 templating
- Secure cookies support
- Unicode-based
- WSGI compliance
- Ability to plug any ORM
- HTTP request handling