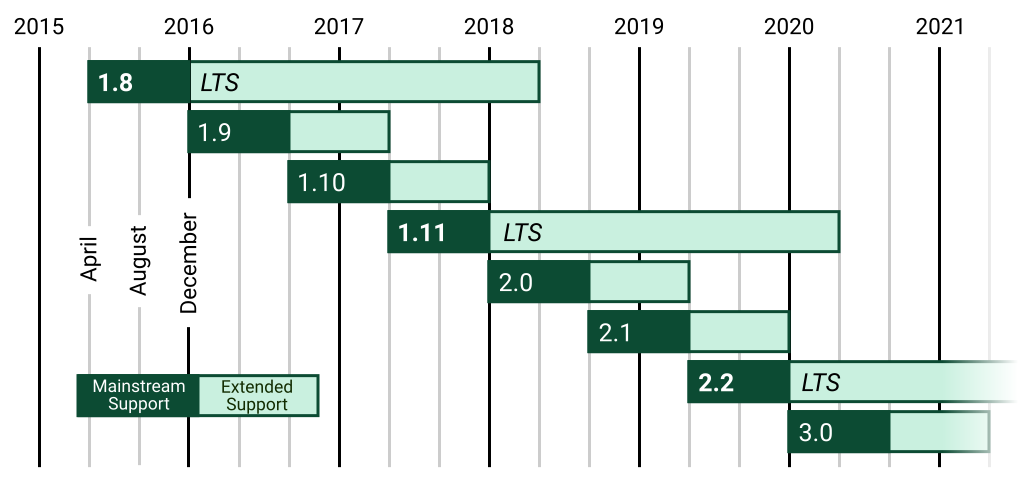
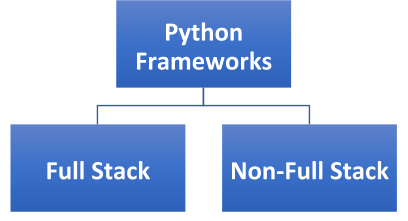
Certain feature releases will be designated as **long-term support (LTS) releases**. These releases will get security and data loss fixes applied for a guaranteed period of time, typically three years.



Python provides support for a wide range of frameworks. Generally, there are two types of Python framework used while developing applications.

https://stackify.com/wp-content/uploads/2018/08/word-image-83.png

Full-Stack Frameworks

The **full-stack frameworks** provide complete support to developers, including necessary elements such as form validation, form generators, and template layouts. Some of the common **full-stack frameworks** are:

1. [Django](https://www.djangoproject.com/)

Django, developed by Django Software Foundation, is a full-stack **Python web framework**. It is an open source and free-to-use framework, released officially in July 2005. It helps developers to create complex code and applications in an easier way, and requires much less time compared to other frameworks. It follows the principle of DRY (don’t repeat yourself) and a model-view-template architectural pattern.

It is widely popular among developers as it has a vast collection of libraries written in the Python language. It emphasizes efficiency, the reusability of components, and less code. Some of the main features of Django are URL routing, object-relational mapper (ORM), authentication mechanism, template engine, and database schema migrations.

Django implements ORM to map its instances to database tables. It provides support for multiple databases such as PostgreSQL, MySQL, SQLite, and Oracle. Hence, it becomes easier for developers to transfer the code from one database to another. In addition, it also provides support for web servers. Because of its amazing features, Django is widely used by most of the renowned companies such as *Instagram, Pinterest, Disqus, Mozilla, The Washington Times, and Bitbucket*.

2. [Web2py](http://www.web2py.com/)

Web2py, developed by *Massimo De Pierro,*is a cross-platform web application framework written in Python programming language. It is an open source and free-to-use **Python web framework**, released in September 2007. It enables users to create dynamic web content in Python. The Web2py framework comes with a code editor, debugger, and deployment tool with which you can develop and debug code, as well as test and maintain applications. It incorporates a ticketing system, which issues a ticket to the user whenever an error occurs. This ticket helps the user to track the status of the error.

Some of the main features of the Web2py Python framework are:

* Cross-platform framework that provides support for Windows, Unix/Linux, Mac, Google App Engine, and many other platforms.
* No additional installation and configuration.
* Built-in components to handle HTTP requests, HTTP responses, cookies, and sessions as well.
* Ability to read multiple protocols.
* Security to data against all possible threats such as cross-site scripting, injection flaws, and execution of infected files.
* Follows model-view-controller (MVC) pattern.
* Support for role-based access control and internationalization.
* Allows users to embed jQuery for Ajax and UI effects.

3. [TurboGears](http://www.turbogears.org/" \t "_blank)

TurboGears, developed by KevinDangoor and Mark Ramm, is a full-stack web application framework. It is a data-driven, open source and free-to-use **Python web framework**. With the help of components such as *WebOb, SQLAlchemy, Genshi, and Repoze*, you can easily develop applications that require database connectivity much faster as compared to other existing frameworks.

Some of the main features of TurboGears are:

* Support for multiple databases.
* Follows an MVC pattern.
* Support for web servers such as Pylons.
* Numerous libraries.
* WSGI (Web Server Gateway Interface) components. For example, it uses ToscaWidgets, which enable developers to embed any complex widget in their application.

4. [CubicWeb](https://www.cubicweb.org/" \t "_blank)

CubicWeb, developed by Logilab, is an open source, semantic, and free-to-use **Python web framework**. This framework is based on the data model. You are required to define the data model in order to get a functional application. It uses the *cube* in place of using separate views and models. Multiple cubes are joined together to create an instance with the help of some configuration files, a web server, and a database.

Some of the main features of CubicWeb are:

* Multiple databases, security workflows, and reusable components.
* Support for Web Ontology Language (OWL) and Resource Description Framework (RDF).
* Embeds Relational Query Language (RQL) in order to simplify the queries related to data.

5. [Giotto](https://pypi.org/project/giotto/)

Giotto is a **Python framework** that is based on the MVC (Model View Controller) pattern. It separates Model, View, and Controller elements in order to ensure that the web designers, web developers, and system administrators can perform their functions independently and effectively.

Apart from this, it also incorporates controller modules, which enables users to create applications on top of the web, irc, or command line.

6. [Pylon](https://pylonsproject.org/about-pylons-framework.html)

Pylon, developed in December 2010, is a lightweight **Python web framework**. It places emphasis on the rapid development of applications. It is developed with some of the best ideas taken from languages such as Ruby, Python, and Perl. Hence, it provides a highly flexible structure for web development.

Note: The Pylon framework is in maintenance mode, after being merged with Pyramid to make Pylons project in order to develop Python-based web technologies.

Non-full-stack Frameworks

The **non-full-stack frameworks** do not provide additional functionalities and features to the users. Developers need to add a lot of code and other things manually. Some commonly used **Python frameworks** are:

7. [Bottle](https://bottlepy.org/docs/dev/)

Bottle, developed by Marcel Hellkemp, is a **microframework**. It is an easy-to-use lightweight framework generally used to build small web applications. It creates a single source file of every project or application. It has no other dependency than **Python Standard Library**.

Some of the basic features of the Bottle framework are:

* Built-in HTTP server.
* Adapters for third-party template engines and WSGI/HTTP servers.
* Allows users to access form data, file uploads, cookies, and other HTTP-related metadata in a much simpler way.
* Provides request-dispatching routes having URL-parameter support.
* Support for plugins of different databases.

8. [CherryPy](https://cherrypy.org/" \t "_blank)

CherryPy is an open source object-oriented **Python framework**. Remi Delon is known as the founder of the CherryPy project. **The CherryPy framework** is widely implemented by developers to create Python web applications. It has its own multi-threaded web server.

You can create applications using CherryPy that will run on any Python-supporting operating systems such as Windows, Linux/Unix, and macOS.

Some of the common features of CherryPy are:

* Contains an HTTP/1.1-compliant, WSGI threaded-pooled web server. It provides support for different web servers as well, for example, Apache and IIS.
* Allows you to run several HTTP servers simultaneously.
* Contains some tools for events such as caching, encoding, authorization, etc. by default.
* Support for profiling, testing, and coverage by default.
* Built-in plugin system.

9. [Flask](http://flask.pocoo.org/)

Flask, developed by Armin Ronacher, is a powerful **Python web application framework**. It is generally termed a microframework because it does not have the following elements:

* Specific tools and libraries
* No database abstraction layer
* No form validation

The functionalities provided by the above-mentioned elements are now provided by third-party libraries. It depends on the *Werkzeug WSGI toolkit* and *Jinja2template*. Some of the common features of the Flask framework are:

* Built-in development server and debugger also.
* Support for unit testing.
* Incorporates RESTful request dispatching.
* Establishes secure client-side sessions.
* Compatible with Google App Engine.

10.[Sanic](https://www.fullstackpython.com/sanic.html)

Sanic is an easy, open source, and simple **Python framework**. This framework is similar to Flask in function but it is much faster comparatively. It was specially designed for quick HTTP responses with the help of asynchronous request handlers.

A remarkable record was made during a benchmark test performed using the Sanic framework. It was recorded that Sanic framework has the potential to process 33,342 requests in a second. This statistic is enough to display how fast Sanic is.

11. [Tornado](http://www.tornadoweb.org/en/stable/)

Tornado, developed by Ben Darnell and Bret Taylor, is a **Python web application framework**. Initially, it was developed for a company named *FriendFeed*, which was later taken over by Facebook in 2009. Tornado is an open source framework and is generally known for its high performance. It uses non-blocking network I/O with the ability to handle more than 10,000 connections at a single time.

Some of the main features of the Tornado framework are:

* Support for user authentication by default.
* Provides high-quality output.
* Non-blocking HTTP client.
* Allows you to implement third-party authentication and authorization schemes, such as Google OpenID/OAuth, Facebook login, Yahoo BBAuth, and Twitter OAuth.