

# Full Stack Python

[All topics](#) | [Blog](#) | [Supporter's Edition](#) | [@fullstackpython](#) | [Facebook](#) | [What's new?](#)

## Flask

Flask (source code) is a Python web framework built with a small core and easy-to-extend philosophy.



### Why is Flask a good web framework choice?

Flask is considered more Pythonic than the Django web framework because in common situations the equivalent Flask web application is more explicit. Flask is also easy to get started with as a beginner because there is little boilerplate code for getting a simple app up and running.

For example, here is a valid "Hello, world!" web application with Flask:

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello, World!'

if __name__ == '__main__':
    app.run()
```

The above code shows "Hello, World!" on localhost port 5000 in a web browser when run with the `python app.py` command and the Flask library installed.

The equivalent "Hello, World!" web application using the Django web framework would involve significantly more boilerplate code.

Flask was also written several years after Django and therefore learned from the Python community's reactions as the framework evolved. Jökull Sólberg wrote a great piece articulating to this effect in his [experience switching between Flask and Django](#).

Flask is an implementation of the web frameworks concept. Learn how these parts fit together in the [web development chapter](#) or [view all topics](#).

## How does Flask relate to the Pallets Projects?

Flask was originally designed and developed by Armin Ronacher as an [April Fool's Day joke in 2010](#). Despite the origin as a joke, the Flask framework became wildly popular as an alternative to Django projects with their monolithic structure and dependencies.

Flask's success created a lot of additional work in issue tickets and pull requests. Armin eventually created [The Pallets Projects](#) collection of open source code libraries after he had been managing Flask under his own GitHub account for several years. The Pallets Project now serves as the community-driven organization that handles Flask and other related Python libraries such as [Lektor](#), [Jinja](#) and several others.

## Flask tutorials

The "Hello, World!" code for Flask is just seven lines of code but learning how to build full-featured web applications with any framework takes a lot of work. These resources listed below are the best up-to-date tutorials and references for getting started.

- The Flask mega tutorial by [Miguel Grinberg](#) is a perfect starting resource for using this web framework. Each post focuses on a single topic and builds on previous posts. The series includes 18 parts: [#1 Hello World](#), [#2 Templates](#), [#3 Web Forms](#), [#4 Database](#), [#5 User Logins](#), [#6 Profile Page and Avatars](#), [#7 Unit Testing](#), [#8 Followers, Contacts, and Friends](#), [#9 Pagination](#), [#10 Full Text Search](#), [#11 Email Support](#), [#12 Facelift](#), [#13 Dates and Times](#), [#14 I18n and L10n](#), [#15 Ajax](#), [#16 Debugging, Testing and Profiling](#), [#17 Deployment on Linux](#) and [#18 Deployment on the Heroku Cloud](#). Miguel also wrote and recorded numerous [Flask Web Development](#) content including a great book and video book that are excellent resources worth the price, especially to support his continuous revisions to the content.
- Armin Ronacher, the creator of Flask, presented the technical talk [Flask for Fun and Profit](#) at PyBay 2016 where he discusses using the framework to build web apps and APIs.
- [Explore Flask](#) is a public domain book that was previously backed on Kickstarter and cost money for about a year before being open sourced. The book explains

best practices and patterns for building Flask apps.

- [Learn to Build Web Applications with Flask and Docker](#) is a video course by [Nick Janetakis](#) that shows how to build a Software-as-a-Service (SaaS) application that is open sourced which uses Flask for the web framework and Docker for the local development environment.
- [Flask by Example: Part 1](#) shows the basic first steps for setting up a Flask project. [Part 2](#) explains how to use PostgreSQL, SQLAlchemy and Alembic. [Part 3](#) describes text processing with BeautifulSoup and NLTK. [Part 4](#) shows how to build a task queue with Flask and Redis.
- The blog post series "Things which aren't magic" covers how Flask's ubiquitous `@app.route` decorator works under the covers. There are two parts in the series, [part 1](#) and [part 2](#).
- [How to Structure Large Flask Applications](#) covers a subject that comes up quickly once you begin adding significant functionality to your Flask application.
- [Flask Blueprint templates](#) shows a way of structuring your `__init__.py` file with blueprints for expanding projects into many files and modules.
- If you're not sure why `DEBUG` should be set to `False` in a production deployment, be sure to read this article on [how Patreon got hacked](#).
- [Developing a Single Page App with Flask and Vue.js](#) step-by-step walkthrough of how to set up a basic CRUD app with Vue and Flask.

## Intermediate to advanced Flask resources

Once you move past the beginner tutorials and have created a few Flask projects you will want to learn how to use Flask extensions, [deploy](#) your code and integrate [web APIs](#) to build more extensive functionality. The following tutorials will guide you through more advanced topics and provide solid learning materials, especially when combined with the example real-world projects listed in the next section.

- [Microservices with Docker, Flask, and React](#) is an awesome course for beyond-the-basics work with Flask. There are a couple of free chapters and the rest of the course is well worth paying for to learn a bunch of valuable tools such as [Docker](#), [React](#) and [microservices architectures](#).
- [Visualize your trip with Flask and Mapbox](#) along with the [open source flask\\_mapbox](#) GitHub repository provides a fantastic example visualization of a trip to Iceland with Flask as the backend web framework.
- [Microservices with Flask, Docker, and React](#) teaches how to spin up a reproducible Flask development environment with [Docker](#). It shows how to

[deploy it to an Amazon EC2 instance then scale the services on Amazon EC2 Container Service \(ECS\).](#)

- [Build a Video Chat Application with Python, JavaScript and Twilio](#)  
[Programmable Video](#) shows how to use [Twilio Programmable Video](#) to build cross-platform (web, iOS and Android) video into Flask applications.
- [Why and how to handle exceptions in Python Flask](#) has some great example code and reasons why you should code defensively by anticipating and handling the unhappy path exceptions in your Flask applications. The examples are relevant to any web framework you will use and are easy to copy and paste to test in your own applications.
- [The Flask Extensions Registry](#) is a curated list of the best packages that extend Flask. It's the first location to look through when you're wondering how to do something that's not in the core framework.
- [How I Structure My Flask Application](#) walks through how this developer organizes the components and architecture for his Flask applications.
- [Adding phone calling to your web application](#) is a killer Flask tutorial with all the code needed to create a web app that can dial phones and receive inbound calls.
- If you're looking for a fun tutorial with Flask and WebSockets, check out my blog post on creating [Choose Your Own Adventure Presentations with Reveal.js, Python and WebSockets](#). Follow up that tutorial by [building an admin interface in part 1, part 2 and part 3](#) that'll show you how to use forms and SQLAlchemy. There is also a companion open source [GitHub repository](#) for the app with [tags](#) for each step in the blog posts.
- [One line of code cut our Flask page load times by 60%](#) is an important note about optimizing Flask template cache size to dramatically increase performance in some cases.
- [Unit Testing Your Twilio App Using Python's Flask and Nose](#) covers integrating the Twilio API into a Flask application and how to test that functionality with [nose](#).
- [The Flask documentation](#) has some quick examples for how to deploy Flask with standalone WSGI containers.
- [Serverless Python Web Applications With AWS Lambda and Flask](#) is a spectacular post that walks through how to run Flask applications on AWS Lambda's [serverless offering](#). The tutorial has instructions on how to include [application dependencies](#) and handle your [deployment workflow](#).

- [Visualize your trip with Flask and Mapbox](#) uses geographic GeoJSON data and presents it in a Flask application that uses [Mapbox](#).
- [Handling Email Confirmation in Flask](#) is a great walkthrough for a common use case of ensuring an email address matches with the user's login information.
- [Static websites with Flask](#) shows how to use Flask with [Frozen-Flask](#) to generate a static website from a backend data source.
- [Running Flask on Docker Swarm](#) details how to run a Flask app on Docker Swarm.
- [Running Flask on Kubernetes](#) step-by-step walkthrough of how to deploy a Flask-based microservice (along with Postgres and Vue.js) to a Kubernetes cluster.
- [Dynamic Secret Generation with Vault and Flask](#) looks at how to use Hashicorp's Vault and Consul to create dynamic Postgres credentials for a Flask web app.

## Open source Flask example projects

Flask's lack of standard boilerplate via a commandline interface for setting up your project structure is a double edged sword. When you get started with Flask you will have to figure out how to scale the files and modules for the code in your application. The following open source projects range from simple to complex and can give you ideas about how to working on your codebase.

- [Skylines](#) is an open source flight tracking web application built with Flask. You can check out a [running version of the application](#).
- [Flask JSONDash](#) is a Flask blueprint that creates JavaScript Object Notiation (JSON) [APIs for data dashboards](#).
- [Microblog](#) is the companion open source project that goes along with Miguel Grinberg's O'Reilly Flask book.
- [Flaskr TDD](#) takes the official Flask tutorial and adds test driven development and JQuery to the project.
- Charles Leifer (author of [Peewee](#) and [Pony ORM](#)) built a [note-taking app](#) along with the [source code in Gists](#).
- [Reddit Job Search](#) uses the [Reddit API](#) for a jobs data set and presents them via a Flask web app.
- [Bean Counter](#) is an open source Flask app for tracking coffee.
- [FlaskBB](#) is a Flask app for a discussion forum.

- psdash is an app built with Flask and psutils to display information about the computer it is running on.

## Flask project templates

Flask's wide array of extension libraries comes at the cost of having a more complicated project setup. The following project templates provide a starter base that you can either use for your own applications or just learn various ways to structure your code.

- Use the Flask App Engine Template for getting set up on Google App Engine with Flask.
- Flask Foundation is a starting point for new Flask projects. There's also a companion website for the project that explains what extensions the base project includes.
- Cookiecutter Flask is a project template for use with Cookiecutter.
- Flask-Boilerplate provides another starting project with sign up, log in and password reset.
- The company Sunscrapers provides this Flask boilerplate project with SQLAlchemy, py.test and Celery baked into the Flask project structure.
- flask-webpack-cookiecutter combines a Flask framework project structure with Webpack, a module bundler frequently used in the JavaScript world.

## Open source code for learning Flask

There are many open source projects that rely on Flask to operate. One of the best ways to learn how to use this framework is to read how other projects use it in real-world code. This section lists these code examples by class and method in Flask.

### Flask: Extensions, Plug-ins and Related Libraries & Example Projects and Code

flask.app BadRequest, Flask, Headers, ImmutableDict

flask.blueprints Blueprint

flask.cli AppGroup, DispatchingApp, FlaskGroup, ScriptInfo, pass\_script\_info, with\_appcontext

flask.ctx after\_this\_request, has\_app\_context, has\_request\_context

flask.globals current\_app, g, request, session

flask.helpers flash, get\_root\_path, make\_response, safe\_join, send\_file, url\_for

flask.json JSONEncoder, jsonify

flask.sessions BadSignature, SessionInterface, SessionMixin

flask.signals Namespace, got\_request\_exception

flask.templating render\_template, render\_template\_string

flask.views MethodView, View, http\_method\_funcs

What web development topic do you want to learn next?

I've built a Python web app, now how do I deploy it?

What other Python web frameworks exist?

How can I version and store my source code?

## Full Stack Python

Full Stack Python is an open book that explains concepts in plain language and provides helpful resources for those topics.

Updates via Twitter & Facebook.

## Chapters

1. Introduction

2. Development Environments

3. Data

4. Web Development

» Flask

5. Deployment

6. DevOps

Changelog

What Full Stack Means

About the Author

Future Directions

Page Statuses

...or view the full table of contents.

---

Matt Makai 2012-2022