

Tutorial

Contents:

- [Project Layout](#)
- [Application Setup](#)
- [Define and Access the Database](#)
- [Blueprints and Views](#)
- [Templates](#)
- [Static Files](#)
- [Blog Blueprint](#)
- [Make the Project Installable](#)
- [Test Coverage](#)
- [Deploy to Production](#)
- [Keep Developing!](#)

This tutorial will walk you through creating a basic blog application called Flaskr. Users will be able to register, log in, create posts, and edit or delete their own posts. You will be able to package and install the application on other computers.



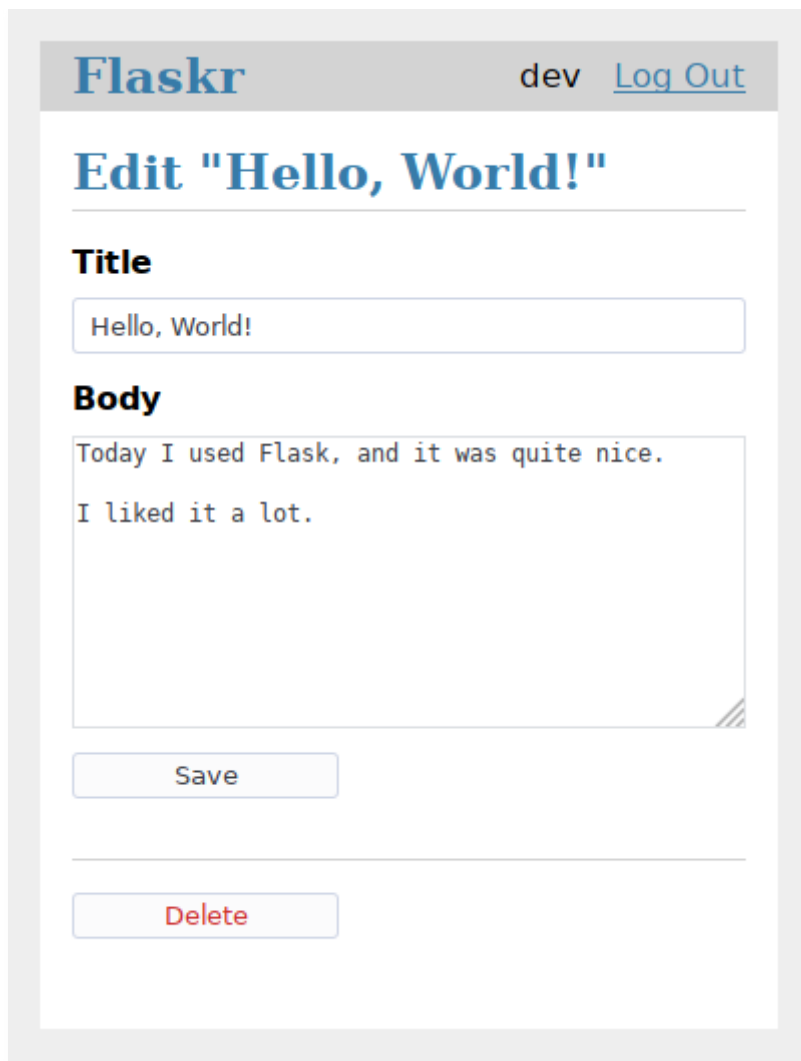
It's assumed that you're already familiar with Python. The [official tutorial](#) in the Python docs is a great way to learn or review first.

While it's designed to give a good starting point, the tutorial doesn't cover all of Flask's features. Check out the [Quickstart](#) for an overview of what Flask can do, then dive into the docs to find out more. The tutorial only uses what's provided by Flask and Python. In another project, you might decide to use [Extensions](#) or other libraries to make some tasks simpler.



The image shows a web form for logging in. At the top, there's a header with the word 'Flaskr' in a large blue font. To its right are two links: 'Register' and 'Log In', both in blue and underlined. Below the header, the main content area has a title 'Log In' in blue. Underneath the title, there are two labels: 'Username' and 'Password', both in bold black font. Each label is followed by a white text input field with a thin blue border. At the bottom of the form is a button labeled 'Log In' in black text on a light gray background.

Flask is flexible. It doesn't require you to use any particular project or code layout. However, when first starting, it's helpful to use a more structured approach. This means that the tutorial will require a bit of boilerplate up front, but it's done to avoid many common pitfalls that new developers encounter, and it creates a project that's easy to expand on. Once you become more comfortable with Flask, you can step out of this structure and take full advantage of Flask's flexibility.



The screenshot shows a web application interface for editing a post. At the top, there is a header bar with the word 'Flaskr' on the left, 'dev' in the center, and a 'Log Out' link on the right. Below the header, the main content area has a title 'Edit "Hello, World!"'. Underneath the title, there is a section labeled 'Title' with a text input field containing 'Hello, World!'. Below that is a section labeled 'Body' with a larger text area containing the text 'Today I used Flask, and it was quite nice. I liked it a lot.'. At the bottom of the form, there are two buttons: a 'Save' button and a 'Delete' button. The 'Delete' button is highlighted in red.

[The tutorial project is available as an example in the Flask repository](#), if you want to compare your project with the final product as you follow the tutorial.

Continue to [Project Layout](#).