

为期30天的 Flask 学习计划，每天的内容逐步递进，适合初学者稳步提升 Flask 开发技能。这个计划假设你每天有 1-2 小时的学习时间。

第1周：基础入门

目标：理解 Flask 的基础知识，并创建第一个 Flask 应用。

Day 1: Flask 介绍与安装

- 理解 Flask 是什么，以及它的特点。
- 安装 Flask，熟悉开发环境设置。
- 创建一个简单的 "Hello World" 应用。

Day 2: 路由与视图函数

- 学习什么是路由，如何通过 `@app.route()` 定义。
- 创建多个不同的视图函数，返回简单的 HTML 响应。

Day 3: 动态路由与 HTTP 方法

- 实现动态路由，理解如何处理 URL 中的参数。
- 学习不同的 HTTP 方法 (GET、POST)，实现基本的 POST 表单提交。

Day 4: 模板基础

- 学习如何使用 Jinja2 模板引擎。
- 创建简单的 HTML 模板并向其传递数据。

Day 5: 模板继承与静态文件

- 学习模板继承，实现布局模板。
- 学习如何加载静态文件，如 CSS 和 JavaScript。

Day 6: 表单处理 (GET/POST)

- 使用 HTML 表单提交数据。
- 使用 `request` 对象处理用户输入。

Day 7: 小项目总结

- 创建一个包含多个页面的小项目 (如一个简单的博客或待办事项应用)，复习本周学习的内容。

第2周：数据库与用户认证

目标：掌握数据库的基本操作和用户认证机制。

Day 8: Flask-SQLAlchemy 安装与设置

- 安装并配置 Flask-SQLAlchemy。

- 创建数据库和表，定义数据库模型。

Day 9: 数据库操作 (CRUD)

- 学习如何在数据库中创建、读取、更新和删除记录 (CRUD 操作) 。

Day 10: 用户注册与登录表单

- 创建用户注册和登录表单，使用 Flask-WTF 进行表单验证。

Day 11: Flask-Login 用户认证

- 学习如何使用 Flask-Login 进行用户认证。
- 实现用户登录、登出和访问保护。

Day 12: 会话管理与 Cookies

- 学习如何使用 Flask 管理用户会话和 Cookies。

Day 13: 权限管理

- 实现用户角色和权限控制，保护特定路由。

Day 14: 项目总结

- 创建一个带有用户注册、登录和权限控制的项目 (如一个简单的博客系统) 。

第3周：API 开发与错误处理

目标：学习如何开发 API，并处理常见错误。

Day 15: RESTful API 基础

- 理解 RESTful API 的概念。
- 创建一个基本的 API 端点，使用 JSON 格式返回数据。

Day 16: Flask-RESTful 扩展

- 安装并使用 Flask-RESTful 创建更复杂的 API。

Day 17: 表单数据与 API 请求处理

- 学习如何通过 API 接收表单数据，处理不同的 HTTP 方法。

Day 18: API 认证与授权

- 实现 API 的用户认证，确保 API 的安全性。

Day 19: 错误处理 (404/500)

- 学习如何处理应用中的错误。
- 创建自定义的 404 和 500 错误页面。

Day 20: 日志记录与调试

- 学习如何使用 Flask 的日志功能。
- 启用 Flask 调试工具，调试应用中的问题。

Day 21: 项目总结

- 创建一个 RESTful API 项目（如任务管理系统），复习这一周的知识。

第4周：部署与优化

目标：学习如何部署和优化 Flask 应用，了解一些常见的 Flask 扩展。

Day 22: Flask 部署基础

- 学习如何使用 Gunicorn 和 Nginx 部署 Flask 应用。
- 部署 Flask 应用到本地或云平台。

Day 23: 部署到 Heroku

- 创建 Heroku 账户，学习如何将 Flask 应用部署到 Heroku。
- 理解如何配置 `Procfile` 和环境变量。

Day 24: Docker 容器化部署

- 学习如何使用 Docker 将 Flask 应用容器化。
- 构建 Dockerfile 并运行 Flask 应用。

Day 25: 使用 Flask-Mail 发送邮件

- 安装并配置 Flask-Mail 扩展，学习如何发送邮件。

Day 26: 使用 Flask-Migrate 进行数据库迁移

- 学习如何使用 Flask-Migrate 管理数据库模式的变更。

Day 27: API 测试与性能优化

- 使用 Postman 测试 RESTful API 接口。
- 学习如何优化 Flask 应用的性能（如缓存管理和查询优化）。

Day 28: Flask 安全基础

- 学习 CSRF 保护、XSS 防护和 SQL 注入防范。

Day 29: 项目总结与优化

- 创建一个完整的项目，结合用户认证、API 和数据库操作。
- 复习并优化项目的代码和性能。

Day 30: 项目部署与展示

- 部署最终项目并展示你的成果。
- 撰写简短的项目总结，整理学习中的难点和收获。

通过这个 30 天的计划，你可以从基础逐步学习到部署一个完整的 Flask 应用，掌握 web 开发的核心技能。