# Python Web开发 教案

周次：8

课次：1

章节内容：Flask框架基础

课时：2

## 单元教学目标

- 知识目标：掌握Flask框架的基本结构和核心概念，包括路由、视图函数、模板的使用方法   
- 技能目标：能够编写并运行一个简单的Flask应用，完成HTML模板的渲染与基本交互逻辑   
- 素养目标：养成良好的代码规范习惯，能通过调试工具排查常见错误，具备基础的软件工程思维

## 教学重点

• 掌握Flask框架的核心路由机制，理解如何通过@app.route装饰器将URL映射到视图函数   
• 熟练使用Jinja2模板引擎渲染动态内容，掌握变量传递、条件判断、循环渲染等基础语法   
• 理解请求对象（request）与响应对象（response）的结构，能通过request.method获取请求方法，通过render\_template渲染动态模板

## 教学难点

• 路由的装饰器使用与URL映射：学生可能难以理解如何通过装饰器（如@app.route）将URL路径与视图函数绑定，尤其是在处理动态URL（如带有参数的路径）时，容易混淆装饰器的参数传递逻辑。   
• 视图函数的参数传递与请求对象的处理：学生可能对如何将请求数据（如表单提交、JSON数据）传递给视图函数存在困惑，尤其是对request对象的属性（如request.form、request.json）的使用不熟悉，容易出现数据获取错误或逻辑错误。

## 教学活动

新课导入【10分钟】   
教师通过展示一个动态网页案例（如“实时天气查询”），提问：“如何用Python快速构建一个可交互的网页？”，引导学生思考。结合案例分析法，展示Flask框架的简化流程，激发学习兴趣。   
  
讲授新课【150分钟】   
1. \*\*讲授法\*\*（30分钟）：   
 - 解释Flask的核心概念（路由、视图函数、模板）；   
 - 通过演示导入，展示Flask快速搭建网页的代码片段（如`from flask import Flask`）。   
2. \*\*案例分析法\*\*（40分钟）：   
 - 分组讨论“如何用Flask创建一个‘欢迎页面’”，分析路由配置（`@app.route('/')`）和模板语法（`{{ name }}`）。   
3. \*\*练习法\*\*（30分钟）：   
 - 学生动手编写“Hello World”页面代码，教师巡视指导，确保理解路由与视图函数的关联。   
4. \*\*讨论法\*\*（20分钟）：   
 - 学生分享代码中遇到的问题（如模板语法错误），教师总结常见错误（如未正确导入模块）。   
  
巩固练习【30分钟】   
1. \*\*分层练习\*\*（15分钟）：   
 - 基础练习：编写简单路由（如`@app.route('/about')`）；   
 - 进阶练习：结合模板渲染动态内容（如`{{ current\_time }}`）。   
2. \*\*小组协作\*\*（15分钟）：   
 - 学生分组完成“用户登录页面”功能，使用Flask处理请求并返回响应。   
3. \*\*教师点评\*\*（10分钟）：   
 - 指出小组代码中的问题（如未正确传递参数），鼓励优化代码结构。   
  
归纳总结【20分钟】   
1. \*\*教师总结\*\*（10分钟）：   
 - 简要回顾Flask的核心功能（路由、视图、模板）及学习重点。   
2. \*\*学生回顾\*\*（10分钟）：   
 - 通过头脑风暴法，学生分享“Flask与Django的区别”“如何调试Flask代码”等收获，教师记录关键点。

## 教学资源

• 教学设备：   
 - 计算机（Windows/Linux/Mac）   
 - 屏幕投影仪（用于展示代码和演示）   
 - 白板或写板（用于板书讲解）   
 - 老师电脑（用于演示代码和操作）   
  
• 开发工具：   
 - Python解释器（建议安装最新版Python）   
 - 零代码编辑器（如VS Code、PyCharm）   
 - 浏览器（Chrome/Edge）用于调试和测试网页   
 - Postman（用于API测试）   
 - Git（用于版本控制）   
  
• 网络环境：   
 - 快速联网（确保能访问网络资源）   
 - 本地服务器（如Ubuntu/Windows Server）（可选，用于部署项目）   
  
• 教学辅助工具：   
 - 本地开发环境（如虚拟环境）   
 - 代码示例文件（如`app.py`、`templates/`目录）   
 - 项目模板（如Flask框架示例代码）   
  
---  
  
• 推荐参考资料：   
 - 书籍：《Flask Web Development》（作者：Greg Wilson）   
 - 官方文档：https://flask.palletsprojects.com/（权威教程）   
 - 在线教程：   
 - Coursera《Python Web Development with Flask》   
 - Udemy《Flask: The Web Framework for Python》   
 - YouTube视频（如“Flask Tutorial”系列）   
 - GitHub项目：   
 - Flask官方示例（https://github.com/flask/Flask）   
 - 学生项目示例（如https://github.com/Flask-quickstart）   
 - 练习网站：   
 - Exercism（Python Web开发练习）   
 - LeetCode（Web开发相关题目）   
 - Codecademy（交互式学习）   
  
• 其他资源：   
 - Flask社区（https://discuss.flask.palletsprojects.com/）   
 - Flask博客（https://blog.flask.palletsprojects.com/）   
 - 网站：https://www.flaskbook.com/（中文教程）

## 教学反思

• 教学效果：学生对Flask框架的核心概念（如路由、视图函数、模板渲染）掌握程度参差不齐，部分学生未能正确理解路由参数与URL路径的映射关系   
• 学生反馈：部分学生反映内容偏理论化，缺乏实际项目案例引导，导致学习兴趣不足；部分学生对Flask的扩展性（如插件、装饰器）理解模糊   
• 教学内容：课程未充分覆盖Flask的进阶特性（如蓝图管理、REST API接口开发），导致后续学习出现断层   
• 课堂互动：学生参与度较低，缺乏课堂讨论或实战演练环节，影响学习积极性   
• 评估方式：未通过项目实践检验学生对Flask框架的综合应用能力，缺乏分层评价标准   
• 技术难点：学生对Flask的调试工具（如Debug模式）和异常处理机制理解不足，影响学习效率   
• 课时分配：理论讲解时间占比过高，实践环节时间不足，导致学生无法及时巩固知识   
• 教学资源：未提供足够的学习材料（如代码示例、文档链接），影响自主学习效果   
• 个性化指导：未针对不同水平学生设计差异化教学方案，导致部分学习困难学生被忽略

## 教学评价

• 评价目标   
 - 通过过程性评价了解学生学习动态，结果性评价检验知识掌握与实践能力   
  
• 过程性评价标准与方式   
 1. \*\*课堂参与度\*\*（30%）   
 - 评分标准：积极回答问题（20分）、主动提问（15分）、小组协作贡献（5分）   
 - 评价方式：课堂观察记录+学生自评   
  
 2. \*\*作业完成情况\*\*（25%）   
 - 评分标准：按时提交（15分）、代码结构规范（10分）、注释完整性（5分）   
 - 评价方式：作业批改+学生提交记录   
  
 3. \*\*代码质量\*\*（20%）   
 - 评分标准：逻辑清晰（15分）、错误处理完善（10分）、代码注释规范（5分）   
 - 评价方式：代码审查+学生自评   
  
 4. \*\*学习态度\*\*（15%）   
 - 评分标准：主动提问（10分）、课后复习（5分）、学习反馈（5分）   
 - 评价方式：学习日志+教师访谈   
  
• 结果性评价标准与方式   
 1. \*\*项目成果\*\*（40%）   
 - 评分标准：项目完成度（30分）、代码规范性（20分）、文档完整性（10分）   
 - 评价方式：项目评审+代码提交记录   
  
 2. \*\*考试成绩\*\*（30%）   
 - 评分标准：理论知识掌握（20分）、实际应用能力（10分）、代码调试能力（10分）   
 - 评价方式：课堂测验+项目代码提交   
  
 3. \*\*课堂表现总结\*\*（10%）   
 - 评分标准：综合表现（10分）   
 - 评价方式：教师综合评分+学生自评   
  
• 评价工具与实施   
 - 评分表：包含各维度的量化评分项   
 - 观察记录：记录课堂行为与作业提交情况   
 - 项目评审：通过代码审查和文档评估确定最终评分   
 - 学生自评：结合学习日志和反思报告   
  
• 评价反馈机制   
 - 提供个性化反馈（如代码优化建议、学习改进方向）   
 - 建立成长档案（记录阶段性评价结果）   
 - 通过课堂讨论或项目汇报展示评价成果