

# Flask 基础

---

1. 一个最小的应用
2. 调试模式
3. 路由
4. 静态文件
5. 模板渲染
  1. 原理：结合控制器使用，渲染数据，模板代码和静态网页代码的混合
  2. 具体用法（课下自学）
    - [Jinja2模板设计者文档](#)
    - [Python模板引擎比较](#)
6. 访问请求数据
  1. Get方式
  2. Post方式
  3. 理解两种方式的区别：表象和原理
  4. 数据收集和处理方式
    - 静态表单页面+控制器+模板（向导式，多页面）
    - 控制器+模板（单页面）
7. 重定向和错误
  1. redirect
  2. 机制：内部处理、redirect、abort
8. 响应
9. 会话

# Flask实例

---

1. 基本功能
  1. 准备数据库
  2. 准备数据库访问库
  3. 显示数据
    - 控制器：控制器读取数据，并调用render\_template推送数据并渲染模板
    - 模板视图：渲染数据，需要掌握模板用法
  4. 添加数据
    1. 数据收集页面：两种方式，纯静态页面和控制器+模板
    2. 添加处理：采用控制器处理数据添加功能，成功后跳转至数据显示视图
    3. 功能合并
  5. 删除数据
    1. 采用控制器处理删除功能，成功后跳转至数据显示视图
    2. 功能合并：通过链接，采用控制器带参数或Get带参数方式
  6. 修改数据
    - 数据修改页面控制器+模板
    - 修改处理：采用控制器处理数据修改功能，成功后跳转至数据显示视图

- 功能合并：通过链接，采用控制器带参数或Get带参数方式

- 难点：

- 如何在修改视图显示待修改的原始数据

## 2. 改进1：

1. 数据库优化

2. 显示数据改进

3. 添加数据改进

1. 数据输入方式改为选择方式，实现采用控制器+模板

4. 修改数据改进

1. 数据输入方式改为选择方式

2. 难点：如何在修改视图显示待修改的原始数据（选择情况）

5. 增加数据显示页查询功能

## 3. 改进2：基于登录和会话实现页面的访问验证

1. 实现登录功能

2. 使用会话变量存储登录状态，并根据该会话变量验证是否合法访问