# Python网络编程 教案

周次：6

课次：2

章节内容：HTTP请求处理

课时：2

## 单元教学目标

- 知识目标：能够区分GET和POST方法，并列举常见的HTTP状态码（200、404、500等）   
- 技能目标：能够使用requests库发送GET和POST请求，并解析响应数据，完成简单网页爬取或表单提交任务   
- 素养目标：能够遵循编码规范编写HTTP请求相关代码，具备团队协作能力并能规范提交项目成果

## 教学重点

• 发送HTTP请求的基本方法：掌握GET和POST请求的构造方式，理解请求头（headers）和请求体（body）的设置逻辑   
• 处理HTTP响应数据：熟练掌握响应状态码的解析（如200成功/404错误），以及响应内容的提取与解析（如JSON数据的解析）   
• 异常处理机制：掌握网络请求中的常见异常（如ConnectionError、Timeout）的捕获与日志记录，确保程序稳定性

## 教学难点

• HTTP方法的区别与应用场景：学生可能混淆GET和POST请求的用途，如GET用于获取数据、POST用于提交数据，且无法在GET中携带敏感信息，导致实际编码时出现错误。   
• URL构造与参数处理：学生容易在拼接URL或传递参数时出错，如未正确编码（如`urllib.parse.quote`）或遗漏参数拼接逻辑，导致请求失败。   
  
（注：难点需结合具体教学内容和学生常见错误，上述分析基于Python网络编程中HTTP请求处理的核心知识点。）