# Python模块 教案

周次：5

课次：1

章节内容：模块导入与使用

课时：2

## 单元教学目标

- 知识目标：能够准确区分模块（module）与库（library）的概念，掌握模块导入的基本语法结构   
- 技能目标：能够使用Python内置模块（如math、random）并正确执行其功能，完成简单模块功能调用   
- 素养目标：养成规范使用模块代码的习惯，注重代码可读性与模块化组织，培养严谨的职业素养

## 教学重点

• 模块导入的基本语法：掌握`import`语句的使用，理解模块文件的保存格式（如`module.py`）   
• 模块的使用方式：熟练掌握通过`module\_name.function()`调用模块中的函数，或`module\_name.variable`访问模块中的变量   
• 模块的常见错误处理：识别模块未找到错误（如`ModuleNotFoundError`）及导入冲突问题（如`NameError`）

## 教学难点

• 模块导入方式的区分：学生可能混淆`import`和`from...import`两种方式，例如误将`from math import sqrt`理解为直接使用`sqrt`而未正确导入模块，导致后续使用时出现错误。   
• 模块的正确导入和使用：学生可能忽略模块需先导入（如忘记`import math`），或误将模块名直接用于函数调用（如`math.sqrt`而非`import math`），导致运行时错误。

## 教学活动

新课导入【10分钟】   
教师用趣味故事导入：讲述程序员小李因误将模块名写成“module”而非“Module”导致程序无法运行的案例。提问：“为什么这段代码无法运行？如何解决？”引导学生思考模块导入的必要性，激发学习兴趣。采用故事导入法与提问法，营造情境氛围。   
  
讲授新课【20分钟】   
1. \*\*讲授法\*\*：讲解Python模块的基本概念（模块文件、模块名、导入语句）。   
2. \*\*案例分析法\*\*：展示代码片段：   
 ```python   
 import math   
 print(math.sqrt(25))   
 ```   
 说明模块导入的步骤（安装模块、导入模块、使用功能）。   
3. \*\*练习法\*\*：分组完成“模块安装与使用”练习，教师巡回指导，学生互评。   
4. \*\*讨论法\*\*：小组讨论“为何需使用`from module import func`而非`import func`？”并派代表发言。   
5. \*\*示范法\*\*：教师演示如何安装第三方模块（如`pip install requests`），强调版本控制的重要性。   
  
巩固练习【15分钟】   
1. \*\*分层练习\*\*：   
 - \*\*基础题\*\*：编写代码导入`datetime`模块并输出当前时间。   
 - \*\*进阶题\*\*：用`from datetime import datetime`获取特定时间并格式化输出。   
2. \*\*分组合作\*\*：学生分组完成“模块冲突处理”任务（如同时导入`math`和`random`模块，需区分功能）。   
3. \*\*实践操作\*\*：利用Jupyter Notebook完成“模块导入与使用”实验，教师实时监控操作流程。   
  
归纳总结【5分钟】   
教师引导学生回顾核心知识点：模块的安装、导入、使用方法，强调“模块名大小写敏感”“`import`与`from`的区别”等重点。学生用思维导图整理所学内容，教师简要总结并布置课后任务（如调研Python第三方模块）。

## 教学资源

</think>  
  
• \*\*教学设备和工具\*\*   
- 计算机（带Python环境）   
- 白板或投影仪（用于演示）   
- Python开发环境（如PyCharm、VS Code、Jupyter Notebook）   
- 屏幕共享工具（如Screen Sharing）   
- 项目管理工具（如Git）（可选）   
- 书籍或参考资料（如《Python编程：从入门到实践》）   
  
• \*\*推荐的参考资料或网站\*\*   
- 《Python编程：从入门到实践》——适合初学者   
- Python官方文档：https://docs.python.org/3/   
- Python教程网站：https://www.python.org/   
- Codecademy Python课程（在线学习）   
- LeetCode（练习编程）   
- GitHub（查看开源项目，学习模块实践）   
- 《Python核心编程》——深入学习模块与包   
- Stack Overflow（提问与解决问题）

## 教学反思

• 教学效果：学生是否能够正确理解模块的基本概念（如模块定义、导入语法）？是否能独立完成模块的创建与使用？是否存在学生在实际操作中遇到的典型问题（如导入错误、模块未定义等）？   
• 学生反馈：课堂互动是否充分？学生是否对模块的组织结构（如包的使用）存在困惑？是否有学生提出关于模块版本兼容性、跨平台部署等问题？   
• 改进建议：增加模块实际应用场景的案例（如使用第三方库解决真实问题）；分步骤教学（先讲导入语法，再讲模块函数使用）；通过代码示例强化错误处理逻辑（如`import error`与`from ... import ...`的区别）；提供分层练习（基础题+进阶题）以满足不同学习需求。

## 教学评价

• 过程性评价   
• 课堂参与度：观察学生是否积极回答问题、参与讨论，记录发言频率和质量，评分标准：优秀（≥3次主动发言）/良好（1-2次）/合格（≤1次）   
• 作业完成情况：检查学生是否正确使用模块（如import语句、函数调用、变量赋值），评分标准：优秀（无错误）/良好（1-2处错误）/合格（≥3处错误）   
• 小组合作表现：评估小组任务分工是否合理、沟通是否有效，记录成员贡献度，评分标准：优秀（分工明确、协作高效）/良好（分工合理、沟通顺畅）/合格（存在协作障碍）   
  
• 结果性评价   
• 作业评分：根据代码规范性、模块使用准确性（如是否正确导入模块、是否调用函数），评分标准：优秀（90+分）/良好（70-89分）/合格（60-69分）   
• 项目成果：评估学生是否完成指定模块（如实现一个功能模块并正确导入其他模块），评分标准：优秀（功能完整、逻辑清晰）/良好（功能基本完成）/合格（存在明显缺陷）   
• 测试评估：通过选择题或编程题检验知识掌握程度，评分标准：优秀（100分）/良好（80-99分）/合格（60-79分）   
  
• 评价方式   
• 过程性评价：课堂观察记录表、小组任务分工表、作业提交截图   
• 结果性评价：作业评分表、项目展示评分表、测试题答案分析表   
  
• 评价标准   
• 过程性评价：总分100分，占比60%（课堂参与度30% + 作业完成度30% + 小组合作20%）   
• 结果性评价：总分100分，占比40%（作业评分40% + 项目成果40% + 测试评分20%）