

教学共同体视阈下Python编程的多模态教材 创新设计与应用

钱学明

无锡科技职业学院物联网与人工智能学院, 江苏 无锡

收稿日期: 2025年5月13日; 录用日期: 2025年6月13日; 发布日期: 2025年6月20日

摘要

文章基于教学共同体视阈, 提出并设计了一种多模态Python编程教材, 旨在通过数字化、互动式和融媒化的教材形态提升教学效果。该教材整合活页、数字与融媒三种形式, 增强了教材的灵活性与适应性, 能够支持学生根据个人学习需求调整学习进度。通过搭建互动式在线学习平台并结合学习数据反馈系统, 教材实现了个性化学习路径推荐, 促进学生自主学习, 并帮助教师进行精准教学。同时, 教材注重学习共同体的构建, 鼓励学生在平台内进行代码分享、讨论与项目展示, 推动知识共享与合作学习。

关键词

教学共同体, 多模态教材, 活页式教材, 融媒教育, 互动学习平台, 个性化学习

Innovative Design and Application of a Multimodal Python Programming Textbook from the Perspective of the Teaching Community

Xueming Qian

School of Internet of Things and Artificial Intelligence, Wuxi Vocational College of Science and Technology,
Wuxi Jiangsu

Received: May 13th, 2025; accepted: Jun. 13th, 2025; published: Jun. 20th, 2025

Abstract

This paper presents the design of a multimodal Python programming textbook from the perspective of the teaching community, aiming to enhance teaching effectiveness through digital, interactive,

and multimedia-integrated formats. The textbook combines three modes: loose-leaf, digital, and multimedia, which significantly improve its flexibility and adaptability, allowing students to tailor their learning progress based on individual needs. By establishing an interactive online learning platform and incorporating a learning data feedback system, the textbook enables personalized learning pathways, promotes autonomous learning, and facilitates targeted teaching by instructors. Furthermore, the textbook emphasizes the creation of a learning community, encouraging students to engage in code sharing, discussions, and project presentations on the platform, thereby fostering knowledge exchange and collaborative learning.

Keywords

Teaching Community, Multimodal Textbook, Loose-Leaf Format, Multimedia Education, Interactive Learning Platform, Personalized Learning

Copyright © 2025 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

当前，全球职业教育正处于快速发展阶段，各国普遍将其视为推动产业升级、提升劳动力素质的重要途径。在我国，高职教育一直备受关注，已发展为培养高素质技术技能人才的关键引擎。为进一步推动高职教育的高质量发展，国家提出了“职业教育提质培优行动计划”，明确其在满足产业需求和服务社会经济发展中的重要作用。这一政策为高职教育的改革与发展提供了有力保障，同时为优化课程体系和培养技术技能人才指明了方向。

2019年1月国务院颁布《国家职业教育改革实施方案》，明确提出实施“三教”改革[1][2]，推动教材编写、教学方法和师资队伍的全面创新与提升。与此同时，职业教育改革中逐步形成的“五金”建设体系，包括“金专业、金课程、金教材、金师资、金基地”，成为破解高职教育发展难题的有效路径。这一体系以优化教学资源、提升课程质量为核心，旨在构建高水平的职业教育体系，培养能够服务于新兴产业需求的高素质技术技能人才。在这一改革框架下，课程内容和教学资源的创新成为重中之重。

Python作为人工智能技术的核心编程语言，凭借其广泛的应用场景和良好的生态，已成为人工智能、物联网、云计算和大数据等专业的核心课程之一[3][4]，为学生进入前沿技术领域提供了必备的编程能力和数据处理技能。Python课程的质量离不开优质教材的支撑，而高质量的Python教材不仅是教学内容的核心保障，更是实现理论与实践深度融合的关键载体。随着产业对复合型技术技能人才需求的日益增长，开发一套既符合高职教育需求又突出实践性的Python教材，已成为提升课程质量的核心任务。

目前，Python编程的教材主要有两类。一类是按照Python的语法结构，依据传统方式编撰的“字典式”教材，内容全面，但行文枯燥，让初学者难以卒读。第二类是根据案例构建教学单元的“项目化”教材[5]，与生产实践紧密结合。近年来，教材中会通过引入二维码，将微课与教材融为一体[6][7]。此外，活页装帧的“活页式”教材[8]-[10]，也开始陆续出现。然而，这些教材无论从内容编排，还是教学使用，仍然遵循着老师教、学生学这样的传统模式。显然，这无法满足数字时代高素质技术技能人才培养。如何将融媒体赋能教材，人工智能技术引入教学实践，是亟待解决的重要问题。

2. 教材创新设计的总体定位

Python编程教材的创新设计将以“一本泛在教材撬动一个教学共同体”为总体定位，旨在构建一个

高度协作、知识共享的教学与学习环境。该创新设计聚焦于教材与课堂、师生互动、线上与线下资源整合，通过教材将教学与学生共同体连接，实现师生共融、知识共建的教学生态。

在此定位下，Python 编程教材的使用必然会形成教学共同体，并表现为一种泛在形态。

2.1. 基于联通主义的教学共同体

联通主义作为一种现代学习理论，强调学习是通过信息网络中的节点相互连接、共享知识的过程。与传统知识传递不同，联通主义关注个体与知识网络的动态交互及自我增值，认为学习的本质是将信息、资源、节点联系起来。这一理念在高职教育中适用于技术学习中多变、分散的知识点整合，通过形成动态网络，实现知识的灵活传递与共享。

基于联通主义理念，Python 教材的设计目标是构建一个教与学互通共融的教材体系，即教学共同体。在这个共同体中，教师担任知识整合的主导者，将分散的知识点系统化，构建完整的知识网络，并引导知识在学生之间流动。而学生则作为学习的主体，不仅吸收知识，还通过群体互动分享、传递知识。在这种网络式的学习环境下，教材成为连接师生的桥梁，形成一个不断进化的动态知识生态，促进学生的全面成长。

2.2. 基于混合式教学的泛在教材

“泛在教材”在教师的视角下具有线下和线上的双重形态。在线下，教材采用活页式装订，方便教师根据课程需求灵活调整教学内容。在线上，教材存在于数字化平台中，提供交互式学习资源，包括图片、漫画、视频和动画等多媒体内容，支持学生自主学习和个性化选择。同时，线上线下内容通过融媒体的形式相互支持，确保学生的学习过程无缝衔接。

从学生的立场出发，“泛在教材”不仅是课堂学习的工具，也是个性化的知识载体。学生可在课堂笔记中撰写属于自己的活页式手稿，通过社交平台(如微信、微博、小红书、知乎、QQ 空间等)分享 Python 学习心得，包括代码解说、项目案例讨论和实践经验。在项目实践中，学生将教材内容与项目案例知识融会贯通，实现从理论到实践的闭环学习体验。

通过“一本泛在教材撬动一个教学共同体”，Python 教材将不再局限于知识的静态传递，而是推动了动态的、师生共融的知识生态的建立，实现知识的普适化和教学效能的最大化。

3. 多维度导向下的教材内容设计

教材内容的选取遵循多维度导向原则，旨在通过准确的基础概念教学、递进式的实践项目、优雅的 Pythonic 编码风格以及课程思政的融入，实现知识传授、技能培养与价值引领的有机统一。教材以“剪刀石头布”游戏开发为主线，构建全书目录，设计从基础到高级的教学内容，通过精心设计的教学活动融入传统文化与思政元素，打造系统化、实践导向的 Python 教学体系，摒弃传统教材按数据类型或编程范式划分章节的结构。

3.1. 夯实基础，正本清源

Python 编程教学的核心在于帮助学生准确理解语言的动态特性和设计哲学。相较于传统教材按数据类型逐一讲解的模式，本教材从数据与对象的关系入手，强调 Python 中一切皆对象的理念，并融入缓存池机制以揭示语言底层设计。例如，在讲解整数对象时，教材通过 `id()` 函数展示小整数缓存池的工作原理，阐释为何 `a = 42`、`b = 42` 会导致 `id(a) == id(b)`，从而帮助学生理解 Python 内存管理的独特性。这种方式加深了对数据本质的认知，避免了机械记忆的弊端。

教材进一步弱化传统“变量”概念，将变量名视为指向对象的标签，突出动态绑定的灵活性。在“初

章：“打印游戏宣言”中，通过案例“游戏标题动态生成”引导学生探索对象与标签的关系。学生需完成以下任务：1) 使用 `id()` 观察字符串对象的缓存效果；2) 通过赋值操作分析字符串的不可变性；3) 使用 `is` 和 `==` 比较标题字符串的标识与值；4) 综合编写一个动态生成游戏标题的程序，输出格式化欢迎信息。这一案例通过递进任务帮助学生把握 Python 的动态特性，纠正部分教材将变量误解为需“声明类型”的静态语言思维，为后续游戏开发任务奠定基础。

3.2. 项目驱动，实践融知

在高职教育中，实践项目的引入能有效增强学生的动手能力。教材以“剪刀石头布”游戏开发为主线，精设 9 个项目，从基础的字符串处理逐步过渡到面向对象编程和图形界面开发。每个项目设计四个由浅入深的子任务。同时，通过案例化教学阐释 Python 核心技能，确保理论与实践的深度融合。教材摒弃传统章节划分，采用游戏开发的递进逻辑，引导学生从简单的输出任务逐步掌握复杂的系统设计，培养解决实际问题的能力。“剪刀石头布”框架设计如表 1 所示。

Table 1. Chapter structure, core content, and task design of the “Rock-Paper-Scissors” framework
表 1. “剪刀石头布”框架的章节结构、核心内容及任务设计

项目	任务	核心内容
初章：打印游戏宣言	<ol style="list-style-type: none"> 1) 输出静态游戏标题 2) 使用 f-string 格式化规则说明 3) 添加动态欢迎语 4) 宣言界面 	字符串操作与基本输出
交互：玩家输入系统	<ol style="list-style-type: none"> 1) 接收玩家选择(如“石头”) 2) 验证输入合法性 3) 处理大小写规范化 4) 构建输入模块 	输入处理与字符串的格式化
裁决：单局胜负引擎	<ol style="list-style-type: none"> 1) 比较玩家与电脑选择 2) 使用条件语句判断胜负 3) 返回结果字符串 4) 完成单局逻辑 	逻辑控制与条件语句
轮回：多回合竞技场	<ol style="list-style-type: none"> 1) 实现单局循环 2) 统计胜负次数 3) 判断最终胜者 4) 完成五局三胜制游戏 	循环结构与计数
记忆：战绩数据中心	<ol style="list-style-type: none"> 1) 记录每局结果 2) 使用列表存储历史数据 3) 计算胜率 4) 构建战绩统计系统 	列表与字典操作
重构：模块化游戏引擎	<ol style="list-style-type: none"> 1) 封装输入逻辑为函数 2) 分离裁决逻辑 3) 优化代码结构 4) 完成模块化重构 	函数与模块化设计
化身：角色工厂系统	<ol style="list-style-type: none"> 1) 定义玩家类型 2) 实现电脑角色策略 3) 添加角色属性 4) 构建角色系统 	面向对象编程

续表

进化：游戏规则扩展包	<ol style="list-style-type: none"> 1) 修改裁决逻辑 2) 更新输入验证 3) 测试新规则 4) 实现扩展模式 	规则扩展与测试
终局：可视化战争	<ol style="list-style-type: none"> 1) 设计按钮界面 2) 显示实时结果 3) 添加动画效果 4) 完成 GUI 游戏 	图形界面开发(tkinter)

3.3. 代码精炼，风格优雅

Python 以简洁优雅的代码风格著称，但许多入门教材沿用繁琐的“C 语言”风格，增加了初学者的认知负担。本教材项目实践中强化 Pythonic 风格，注重列表推导式、字典推导式、lambda 函数与高阶函数的应用，引导学生养成简洁高效的编码习惯，通过编程案例对比传统写法与 Pythonic 写法，突出语言的设计意图。

在“记忆：战绩数据中心”阶段，教材通过案例“筛选胜利记录”展示列表推导式的优势。传统写法为“`wins = []; for record in records: if record['result'] == 'win': wins.append(record)`”，而 Pythonic 写法为“`wins = [record for record in records if record['result'] == 'win']`”。学生逐步实践以下任务：1) 使用循环筛选记录；2) 改用列表推导式；3) 扩展为多条件筛选；4) 优化为完整胜利查询函数。类似地，在“进化：游戏规则扩展包”中，字典推导式通过案例“统计选择频率”展开，对比传统写法“`freq = {}; for choice in choices: freq[choice] = freq.get(choice, 0) + 1`”与“`freq = {choice: choices.count(choice) for choice in set(choices)}`”，突出简洁性。任务包括：1) 使用循环统计频率；2) 改用字典推导式；3) 处理选择规范化；4) 完成频率统计模块。

对于 lambda 函数与高阶函数，教材在“化身：角色工厂系统”中通过案例“按角色胜率排序”展示 `sorted(players, key=lambda x: x.win_rate)` 的优雅性，对比传统循环排序的复杂性。学生可以：1) 实现简单排序；2) 使用 lambda 函数按多字段排序；3) 结合 `filter()` 筛选高胜率角色；4) 完成综合排序系统。这些案例通过递进任务融入“剪刀石头布”框架，引导学生掌握 Pythonic 写法，避免冗长代码，培养优雅编码习惯。

3.4. 思政入课，润物无声

在高职教育中，课程思政的融入是提升学生综合素养的重要途径。教材内容的选择不仅要围绕编程技能展开，还要在其中自然融入人文关怀与社会责任。教学设计以编程任务为载体，围绕学生熟悉的技术实践，巧妙嵌入人文关怀与价值引领，通过技术与文化的双重体验，培养学生的文化自信和社会责任感，避免生硬的理念灌输。

例如，开展“革命文献分析”教学活动，学生编写程序处理革命文献，提取高频词汇(如“奋斗”、“团结”)并统计出现频率，任务从基础的字符串操作到正则表达式的应用，与“记忆：战绩数据中心”阶段的数据处理内容结合。教学设计通过技术任务引导学生接触革命历史，如分析文献中体现的牺牲与团结精神，教师在课堂讨论中进一步引导学生反思这些精神在当代的价值。这种设计使学生在掌握编程技能的同时，潜移默化地接受红色文化的熏陶，增强对革命传统的认同感。

通过文化的浸润与价值的引导，逐步形成文化自信与社会责任感，实现了课程思政与编程教育的有机统一，达到润物无声的教育效果。

4. 多模态需求下的教材形态优化

4.1. 活页形态

在传统教材的编排中，内容的顺序和组织形式是固定的，往往难以兼顾不同层次学生的学习进度和节奏。为了应对这一问题，我们采用活页形式，以模块化的内容安排实现灵活、可调的学习路径。每个知识单元独立成篇，学生可以根据教学进度自由选择 and 组合学习内容，从而增强学习的自主性。同时，为学生在课后整理笔记、复习巩固提供了便利。活页形态也使教师能够根据教学内容的需要增删调整章节，提供个性化教学资源，为不同学习水平的学生提供合适的挑战和帮助。

此外，活页教材的设计有利于知识点的更新迭代。随着 Python 语言和编程技术的发展，教材内容可以根据需要不断扩展和补充新内容，或将过时部分移除，确保教材始终保持实用性和前沿性。这样的活页设计既提高了教学资源的利用率，也让教材的使用寿命更长，能够更好地支持高职教育需求。

4.2. 数字形态

在 AI+时代，数字化教材成为实现无纸化和信息资源共享的重要载体。我们可以采用数字形态，将内容以电子资源的形式呈现，为学生提供了更便捷的学习途径。通过数字形态，学生可以随时随地访问教材内容，支持碎片化学习，从而适应当代学生的学习习惯。同时，教材提供的电子版不仅包含核心内容，还整合了视频、音频、交互式代码等多媒体资源，帮助学生更直观地理解和掌握编程知识。

数字形态的教材还能够很好地支持在线测试和互动反馈。教材在每个章节后都设置了练习题，学生完成练习后可以即时获得答案和详细解析，有助于巩固所学知识。在一些实践性强的章节，教材还设计了互动代码区，学生可以在其中直接运行和调试代码，获得即时反馈，提高编程能力。此外，数字教材提供的搜索功能也便于学生查找特定内容，提升学习效率。

数字形态还允许学生个性化标记和笔记管理。每个页面都设有电子标注功能，学生可以随时对难点和重点内容进行标记，并记录下个人学习笔记。这样在复习时可以快速回顾，避免遗漏知识盲区。这种基于学生个性化需求的数字形态教材，帮助学生在有效管理知识，提升记忆效果，从而更加高效地掌握 Python 编程技能。

为了便于课堂内外的延展学习，数字化教材还可以集成班级学习群组功能。学生可以通过教材平台分享学习心得、发布代码片段和提出问题，从而实现在线学习交流。此外，教师可以在平台上发布课堂练习答案、解答疑难问题，以及共享补充资料，为学生提供全方位的学习支持。学生在学习过程中形成的个人理解和总结也可以发布到群组中，促进学习共同体成员间的交流与互助，真正实现知识的共建共享。

4.3. 融媒形态

随着融媒体的“高歌猛进”，教材形态逐渐多元化，传统的纸质内容不再局限于静态的图文信息，而是借助多种媒介形式，打破时空限制，延展教材内容的表现力和传播力。我们引入融媒形态，通过图文、视频、动画、互动测试等多媒体形式展现编程知识。比如，在讲解 Python 的面向对象编程时，可以利用动画展示类与对象的关系，帮助学生更直观地理解抽象概念。

融媒形态的教材在数字平台上整合了可交互的练习题和代码运行环境，学生可以边看边学，边学边练。在讲解复杂算法时，教材中的视频与动画演示结合代码调试功能，帮助学生更清楚地理解算法的运作流程，增强学习效果。此外，教材还支持学生分享学习成果，例如学生可以将自己编写的 Python 程序或作业分享到融媒平台，教师也可及时给予反馈与指导，从而形成闭环式学习。

融媒形态中的短视频和动态案例分析模块也为学生自主学习提供了强有力的支持。学生可以通过观看教材中的短视频快速掌握 Python 基础知识，而案例分析模块则帮助学生理解实际应用中的关键点与难点。视频的灵活性让学生在碎片化时间里有效学习，提高了学习的连续性和效率。特别是融媒教材中嵌入的短视频为课堂内外的教学一体化提供了新的支持，使学生能够随时回顾课堂内容，强化所学。

教材融媒形态中的知识点推送和学习打卡功能可以有效提升学习积极性。教材平台会根据课程进度每日推送相关知识点和小测验，学生完成打卡后能收到即时反馈，从而形成持续的学习动力。同时，平台中的学习群组和社区讨论区为学生提供了展示学习成果的渠道，有助于构建积极的学习氛围。学生通过融媒教材的互动功能进行讨论、分享与交流，有助于增强学习共同体的凝聚力，实现知识的双向流动。

融媒形态不仅是教学内容的延展，也是教学资源与学生学习生活的深度融合。这种多样化的教学模式通过线上线下联动，打造出一个全方位的学习生态，真正实现“一本泛在教材撬动一个教学共同体”的目标。

5. 驱动学习共同体的教材生态构建

5.1. 互动式在线学习社区

互动式在线学习平台不仅是教材资源的存储和分发空间，更是一个动态的学习社区，通过多样化的功能设计和人工智能驱动的支持，促进师生间、学生间的互动交流。这一教材平台不仅配备了问答区和讨论区，便于学生在遇到编程难题时随时提问和讨论，利用 AI 还能进一步优化学习体验。人工智能技术能够智能分析学生的问题，推荐相关解答资源或直接给出提示性答案，提升了自助式、协作式学习的便捷性与效率。AI 的辅助帮助学生更快找到解决方案，在自我探究过程中增加参与感与归属感，激励他们在学习中更主动地思考与探索。

学习社区中设有代码共享和作业展示功能，学生可以上传自己的 Python 代码，接受教师和同学的点评与反馈。通过 AI 的介入，平台能够自动分析和检测学生提交代码的正确性、效率和风格，并提供改进建议，帮助学生从多个角度完善代码。这不仅拓宽了学生的编程思路，也提高了代码质量，让学生在分享和改进中逐步形成 Pythonic 的编程习惯。教师还可利用教材平台组织编程挑战赛和小组项目，借助 AI 进行学生成绩的智能分析和个性化的反馈，进一步激发创造力。例如，在讲授 Python 面向对象设计时，教师可以组织一次小型代码竞赛，由 AI 系统自动评估代码的创新性和优化程度，为学生提供即时反馈，并最终评选出优胜者。

教材平台中的“动态消息”功能也为学习社区提供了及时的更新和激励。无论是发布学习进度、更新资源，还是展示学生的学习成果，AI 都可以为师生推荐与当前学习内容相关的最新资源，自动提醒学生尚未掌握的知识点或薄弱环节。教师还可以通过“动态消息”发布教学笔记或重点提示，学生则能分享个人学习心得、推荐参考资源，甚至上传项目实践中的编程体验。这种信息流不仅增强了学习共同体的活力，还推动了知识的双向互动与深入理解，使整个学习过程更加高效和富有成效。

5.2. 数字资源库

多模态教材不仅是知识的载体，更是教学共同体的资源核心，且在云上资源库中引入了人工智能技术，以提升学习体验的智能化和个性化。为确保学生在不同学习场景下均能得到有效支持，教材配套的云上资源库涵盖基础知识点、典型案例和各类项目实践资源，按照知识模块分类整理了编程练习、案例代码和实战项目，使学生能够灵活选择适合自己进度和兴趣的内容。人工智能在此通过个性化推荐和智能导航功能，辅助学生找到最佳学习路径，满足各类学习需求，并帮助教师灵活挑选适配的教学案例。

案例库精选实践性和代表性的 Python 项目，以帮助学生在理论学习中同步获得实践经验。例如，在数据分析模块中，案例库提供了真实的电商销售数据分析项目，学生通过 Python 实现数据清洗、分析和可视化模块，从而掌握数据分析的核心技能。AI 工具还可以对学生在项目中提交的代码进行自动分析和反馈，指出潜在问题和优化建议，为学生提供深度的实践指导。这种项目驱动的学习方式不仅增强了学生的编程技能，更帮助他们获得职业所需的初步项目实战经验。

此外，云上资源库提供了丰富的学习辅助材料，包括教学视频、逐步讲解的代码示例、Python 开发工具包和使用指南。借助 AI 的辅助功能，学生可以在资源库中随时获取相关内容的推荐，并通过可视化工具深入理解复杂概念。例如，对于递归算法等难点内容，资源库配备了动态演示代码和动画，帮助学生通过多视角理解算法原理。在 AI 驱动的资源支持下，学生不仅能自我推进学习，还能更高效地完成编程任务。

云上资源库与在线学习平台无缝衔接，学生可直接通过教材链接访问相应资源，获得即时的学习支持。通过这种云端化、灵活化的资源整合，学习内容不再受限于纸质教材，使学生的学习体验更加流畅和高效，教师的教学效率也因此大幅提升。

5.3. 学习数据与反馈系统

在教学共同体的建设中，学习数据与反馈系统为实现个性化教学和精准学习支持提供了核心支撑，同时引入人工智能技术以优化数据分析和反馈机制。该系统通过实时跟踪和反馈学生的学习数据，使教师能更全面地了解每位学生的学习进展、知识掌握情况及薄弱环节，从而有针对性地调整教学策略和进度。系统记录学生的各类学习行为，包括课程学习时间、章节完成情况、测试成绩和代码提交记录。AI 技术则进一步分析这些数据，生成全面的学习表现评估报告，助力教师精准掌控班级整体学习情况。

学习反馈系统不仅收集和分析数据，还向学生提供个性化的学习状态反馈。每完成一个章节，系统会自动生成学习报告，涵盖进度、完成率、测试成绩、代码准确性等多个维度的反馈。AI 技术根据学生的薄弱环节推荐相关资源，如补充练习和详细讲解视频，帮助学生巩固概念并提升编程能力。例如，在学习循环结构后，若某学生在相关练习中的错误率较高，系统会智能推送循环相关的补充练习和讲解资源，从而为学生提供深层次的概念理解支持。

此外，系统通过智能学习建议功能，根据学生的学习轨迹和表现自动生成学习路径和资源推荐。对于进展较快的学生，系统会推送更具挑战性的项目，鼓励他们在编程技能上取得进一步进展；而对于进度稍慢的学生，系统则建议复习基础内容，并推荐适合的练习资源。这样的个性化推荐机制使学生能够根据自身需求合理安排学习任务，显著提升自主学习的效率和效果。

通过实时的数据反馈和 AI 驱动的智能分析，教学共同体得以实现高度个性化和动态化，促进了师生间的频繁互动与有效协同。学生能够根据系统反馈优化学习方式，教师也能在教学过程中精准进行学业辅导，最终实现科学、高效的学习路径。这一数据驱动的教学模式不仅深化了教学效果，还构建了一个智能协同的学习生态系统。

6. 特色与创新

6.1. 特色

厘清概念误区，强化编程思维。面对当前 Python 编程中基础概念模糊化、教学内容片面化的问题，教材设计中聚焦编程核心概念的准确性和系统性。通过针对常见谬误的深度剖析，教材帮助学生厘清变量的概念、变量作用域、函数参数传递等关键概念，避免学生在初学阶段形成错误认知。此外，利用可视化工具和逐步分解的代码示例，将复杂的抽象概念转化为直观的学习内容，注重编程中的核心逻辑与

思想,通过案例、项目强化学生的理解能力和问题解决思维,从根本上引导学生从“学会编程”到“用编程思考”的转变。

践行 Pythonic 理念,提升代码质量。教材设计中,编程实践部分全面融入“Pythonic”风格,旨在培养学生简洁、优雅和高效的编程习惯。通过精选案例,教材展示了 Python 编程的核心哲学,例如“明确优于晦涩,简单优于复杂”。学生通过对比优化代码的过程,深入理解如何写出清晰易读、逻辑合理的代码。此外,教材还提供丰富的代码优化示例,直观展示常见编程误区和最佳实践,引导学生逐步掌握高质量代码开发的能力。Pythonic 理念的贯穿不仅提升了学生的编码水平,也为其未来的职业发展奠定了良好的技术基础。

融入课程思政,拓展教育价值。积极践行“课程思政”的要求,将知识传授与价值引领有机结合。在教材内容选取上,教材优先选择与社会热点相关的项目案例,如红色文化、可持续发展、人工智能伦理与数据安全等,通过编程任务引导学生关注科技与社会发展的关系。同时,在案例分析和项目设计中渗透团队合作、责任意识和职业道德等教育元素,使学生在在学习技能的同时受到思想启迪和人格塑造。教材通过内容和形式的双重创新,将编程学习转化为技能培养与思政教育融合的过程,充分体现职业教育的德技并重特质。

6.2. 创新

新教材形态创新,突破传统局限。教材以“活页、数字、融媒”三位一体的形态设计,创新性地突破了传统教材的单一模式。数字形态通过云端平台扩展学习场景,融媒体形态则以多模态呈现方式弥补传统教材的局限性,真正实现学习时间、地点、方式的自由化。这种形态突破,不仅提升了教材的功能性,也为现代职业教育提供了教材设计的新思路。

个性化设计优化,增强学习效率。教材引入学习数据反馈与路径推荐系统,搭建以学生为中心的智能学习环境。系统通过大数据记录与分析学生学习情况,生成个性化报告并推荐优化方案,使学生能够有针对性地提高薄弱环节。同时,教师借助该系统精准开展教学管理,节省备课时间并优化教学效率。

共同体模式创新,促进协作发展。教材推动“教学共同体”的实践,将学生的自主学习与合作学习相结合。平台通过构建案例库和项目库,提供优质学习资源;同时,支持学生间的代码共享与问题协作,培养学生的问题解决能力和团队协作意识。这种共同体驱动的学习模式,为高职教育中的实践能力培养提供了创新途径,也充分体现了现代教育以学生为本的核心理念。

7. 总结

本文在教学共同体视阈下围绕“一本泛在教材撬动一个教学共同体”的核心理念,创新设计了 Python 编程的多模态教材。在教材内容上,重视基础知识的精准传达,通过 Pythonic 编程风格培养学生扎实的编程思维;在形态上,采用活页、数字和融媒的形式,为师生提供灵活、丰富的学习体验;在教学模式上,教材构建了互动式学习社区和数据反馈系统,支持个性化学习路径和资源推荐。所有这些设计无不体现出对现代高职教育特点的深刻理解,力图将教材打造为学生、教师和行业资源共融的学习生态。

当然,随着大语言模型的广泛应用,教材设计也面临一些挑战。大模型拥有强大的知识整合和自动化教学支持功能,能够根据学生的需求提供实时解答、代码建议等服务,但这种优势也可能削弱教材自身的部分功能,例如学生在编程问题上直接依赖大模型解答,可能影响其自主思考能力,带来过度依赖问题,是未来教材开发中需要思考的重要课题。

基金项目

本研究得到了全国高等院校计算机基础教育研究会计算机基础教育教学研究项目(编号:2021-AFCEC-

580)、江苏省科技副总项目(2023)、江苏高校“青蓝工程”项目(中青年学术带头人(2022)、优秀教学团队(2020))、无锡市科协软科学研究课题(编号: KX-24-B39)、2022年无锡科技职业学院教改研究课题(编号: JG2022103)的资助和支持。

参考文献

- [1] 秦华伟, 陈光. “双高计划”实施背景下“三教”改革[J]. 中国职业技术教育, 2019(33): 35-38.
- [2] 吴秀杰, 张蕴启. “双高计划”背景下高职“三教”改革的价值、问题与路径[J]. 教育与职业, 2021(9): 11-18.
- [3] 张焱. 人工智能时代高职物联网应用技术专业信息化教学应用的探讨[J]. 职业技术, 2019, 18(12): 27-30.
- [4] 唐哲卿, 孙希强. 人工智能背景下黑龙江省物联网专业人才培养改革研究[J]. 无线互联科技, 2021, 18(20): 126-127.
- [5] 史峰, 蔡效猛. “三教”改革背景下信息化资源区域建设研究[J]. 职教通讯, 2021(1): 59-65.
- [6] 王金凤. 新冠肺炎疫情下的高职院校线上教学[J]. 中国教育技术装备, 2021(20): 120-121.
- [7] 王子琪, 王亚轩. 新冠肺炎疫情防控背景下基于大数据平台的“轻直播式”线上教学模式的研究[J]. 科技资讯, 2021, 19(6): 11-13, 67.
- [8] 散晓燕. 1+X证书制度下高职新型活页式教材的特征、价值与设计[J]. 教育与职业, 2021(11): 93-97.
- [9] 陈高锋, 付建军. 活页式教材设计及应用探索与实践[J]. 陕西教育(高教), 2020(5): 26-27.
- [10] 姚建盛, 刘凯辉. 校企合作活页式教材开发研究与实践[J]. 汽车维护与修理, 2022(2): 32-34.