

# 农学背景下 python 实验教学方法探析

况江

贵州师范学院

DOI:10.12238/mef.v8i8.12364

**[摘要]** 针对没有编程语言基础的学生在学习python语言时,本文提出了一种“实用性引导+兴趣激发+情景式项目实战”的模式,以帮助学生深入理解python编程语言,激发他们的学习兴趣,并通过情景式的项目实践系统的引导学生主动掌握python语言基础的核心知识点以及数据分析的各项技术要领。

**[关键词]** python; 实用性引导; 兴趣激发; 情景式项目实战; 模式

中图分类号: G422 文献标识码: A

## Methods of python experiment in the context of agriculture

Jiang Kuang

Guizhou Education University

**[Abstract]** Regarding students without a foundation in programming languages when learning Python, this article presents a model of "practical guidance + interest stimulation + scenario-based project practice" to assist students in profoundly comprehending the Python programming language, stimulating their learning enthusiasm, and systematically guiding them to proactively grasp the core knowledge points of the Python language foundation and the various technical essentials of data analysis through scenario-based project practices.

**[Key words]** Python; Practical Guidance; Interest Stimulation; Scenario-Based Project Practice; Model

## 引言

随着社会的发展与进步,农业资源与环境专业对学生的要求不断提高。这不仅包括夯实理论知识、具备实验动手能力和良好的科学素养,还强调了对实验数据分析能力的培养。因此,掌握一门编程语言以更好地处理实验数据并拓展学习能力已成为一种趋势<sup>[1]</sup>。Python是一门跨平台、开源且免费的解释型高级动态编程语言,是目前市场上最受欢迎的主流程序语言之一。本文将探讨在农业资源与环境专业中如何快速掌握python语言基础与数据分析在教学中应用方法。

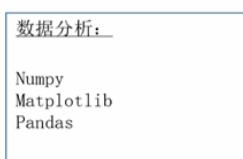
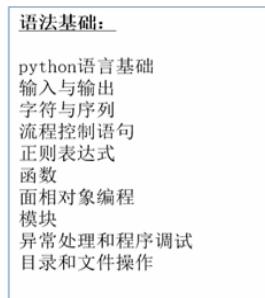


图1 “python语言基础与数据分析”课程主要内容

## 1 “python语言基础与数据分析”课程的主要内容

Python具有易于学习、易于阅读、可扩展可移植以及广泛的标准库和第三方库等特点。此外,python结构丰富且应用广泛,相较于其他编程语言,对没有编程基础的学生更加友好。因此,在“python语言基础与数据分析”课程的教学中,以学生为中心,将课程分为基础语法和数据分析两个阶段<sup>[1]</sup>。如图1所示。

## 2 “python语言基础与数据分析”课程教学中遇到的主要困难

每年在开课前对本专业的学生进行问卷调研,摸清学生的计算机基础、编程语言等真实情况,了解学生的基础,便于更好的开展教学工作。问卷部分内容如图2:

课程结束后,收集学生的反馈信息,不断改进教学方法,部分内容如图3:

尽管Python对于初学者非常友好,但是通过分析近三年的农业资源与环境专业学生的学情与信息反馈数据,总结得出以下教学实践过程中遇到的问题<sup>[2][3]</sup>。

### 2.1 python课程学时有限

\* 1. 您是否了解Python语言?

A 非常了解  
 B 了解一些  
 C 不太了解  
 D 完全不知道

\* 2. Python语言可以应用在哪些领域? 【多选题】

A 数据分析  
 B 人工智能  
 C 网络开发  
 D 游戏开发  
 E 科学计算

\* 3. 您认为学习Python语言的难度如何?

A 很简单  
 B 比较简单  
 C 有一定难度  
 D 非常困难

\* 4. 您通过哪种途径了解到Python语言的? 【多选题】

A 学校课程  
 B 在线教程  
 C 书籍  
 D 朋友推荐  
 E 其他

\* 1. 您在学习Python之前的编程经验如何? 学过哪些计算机基础课程

A 没有编程经验  
 B 有一些基础  
 C 有较丰富的经验  
 D 选项1  
 具体学过哪些计算机课程

\* 2. 您认为学习Python的难度如何?

A 非常简单  
 B 较简单  
 C 一般  
 D 较困难  
 E 非常困难

\* 3. 您在学习Python过程中最喜欢的部分是什么?

A 语法基础  
 B 数据结构  
 C 面向对象编程  
 D 网络编程  
 E 数据分析

\* 4. 您在学习Python时遇到的最大挑战是什么?

A 理解语法  
 B 调试代码  
 C 找到学习资源  
 D 完成项目

图3 信息反馈收集问卷截图

通过学前学情调查获悉,90%以上的学生没有编程语言基础,从未了解和学习过编程语言,没有相应的基础。农业资源与环境专业顺应时代的发展增设“python语言基础与数据分析”课程,并设置48学时,但是对于初学者而言,不仅需要学习语言基础语法,还要掌握相应的数据分析方法,学习理论知识、实践动手练习和编写代码需要较多的时间,因此48课时的设置显得明显不足。一方面,编程不仅需要学习理论知识,更需要大量的实践操作;另一方面,大部分农业资源与环境专业的学生未曾接触过任何编程,因此计算机相关知识几乎为零。因此,应根据农业资源与环境专业的需求,从大量的python基础知识中选择核心知识点,重点讲解。并结合课堂学习和课后查阅资料等方式来培养学生自主学习拓展能力,通过课堂学习-拓展自学-提出问题-课堂解决问题的方法,提高学生主动学习积极性。在实际教学过程中,与学生交流发现,他们面临着密集的课程安排问题,使得分配给python上的时间极其有限。同时,自律自学的学生也比较少,导致他们在主动性上不高,从而影响整体学习效果。

## 2. 对编程语言存在莫名的畏难情绪

农业资源与环境专业通常吸引偏文科背景的同学,这使得许多人对计算机编程产生主观上的畏难心理。他们普遍认为编程很难理解,同时觉得术语繁杂且逻辑复杂,而这类技能似乎只适合计算机或者数学专业。而且,大部分同学是第一次接触此类内容。“python语言基础与数据分析”这一课程设立于第三学期,此时学生经历了一年的大学生活,自主学习能力尚未得到充分锻炼,自律性相对薄弱,加之缺乏任何编程经验,他们常常自我设限并缺乏信心。此外,即便现在可以通过各种渠道获取关于python的信息,但很多人依旧停留在简单复制粘贴理论知识层面<sup>[4]</sup>,没有真正通过敲代码,读代码及调试运行来提升自己的纠错能力<sup>[5]</sup>。由于网络上有关python的信息繁杂无序,导致学生无法系统化地进行有效思考,也不能高效利用这些信息。因此,需要教师深入思考如何更好地引导并激发学生兴趣,以克服这种畏难情绪。

## 2.3 对编程语言的代码编写意愿不强

对于大多数同学而言,初次接触编程语言时,往往会面临一些挑战。一方面,心理上可能会对编写代码存在障碍,担心写错或不知道如何下手;另一方面,缺乏兴趣和成就感也是常见的问题。在整个学习过程中,如果无法看到自己所学知识的实际应用,或者没有明显的进步和直观的成果,就容易感到缺乏兴趣,失去主动学习和探索的动力。例如,许多学生在初次编写“Hello World”程序时,可能会因为简单的输出结果而感到困惑,不知道这样的练习如何帮助自己提升编程能力。此外,课程内容如果过于理论化,缺乏实际项目的练习,也容易让学生感到枯燥。因此,找到合适的学习方法和实践机会非常重要,比如参与开源

项目或者开发简单的应用程序,都能有效提升学生学习兴趣和自信心。

#### 2.4 编程逻辑思维未建立

对于初学Python的同学而言,他们尚未形成编程的基本逻辑思维<sup>[6]</sup>。这种逻辑思维的培养不仅仅是掌握语法知识,更需要在实际问题中运用和思考。然而,现有的多数教程侧重于语法的讲解,却鲜少有内容清晰地帮助学生建立起编程的逻辑思维。实际上,逻辑思维的构建需要经过大量的实践和尝试,通过不断的练习和摸索,学生才能逐步形成编程的逻辑思维。在这个过程中,学生应该多参与项目实践,从简单的编程任务开始,逐渐提升难度,从而在实践中锻炼和巩固自己的逻辑思维能力。此外,寻求有经验的导师或者参与编程社区讨论,也能有效加速逻辑思维的形成。只有通过综合运用多种方法,学生才能在编程之路上打下坚实的基础。

### 3 “python语言基础与数据分析”课程的“实用性引导+兴趣激发+情景式项目实战”教学方法

针对学生存在的“四大问题”,始终秉持OBE理念,“以学生为中心、以产出为导向”的原则,不断优化教学方式,致力于寻找更优的解决方法。课程经历了从纯代码显示到自主设计图形界面展示对话的转变,逐步实现了从代码计算到可视化展示及图形界面操作的过渡。在这一过程中,课程团队不断探索和优化教学方法,确保学生能够更好地掌握Python编程技能。具体而言,课程首先通过简化代码显示,使学生能够快速上手。这种简化的方式帮助学生迅速建立起对编程基础的理解和信心,避免了初学者常常遇到的挫败感。随着学习的深入,学生被引导自主设计图形界面,将代码转化为更加直观的对话展示。这一过程不仅促使学生思考如何将复杂的问题简单化,还培养了他们的创新思维和解决问题的能力。这种逐步过渡的方式,不仅激发了学生的学习兴趣<sup>[7]</sup>,还增强了他们的实践能力,使他们在面对实际编程任务时更加游刃有余。通过如图4所示的实用性引导,点燃学生的学习兴趣,增强他们对Python的理解。学生能够深入了解学习Python后的广泛应用,包括Python编程语言的功能及其在专业领域的帮助,从而激发他们对Python练习的浓厚兴趣。这种方法促使学生变被动为主动,不仅完成布置的练习,还会主动探索课堂上未能涉及的Python实用功能,拓展知识面和提高实用性,最终达到深度练习的效果。通过这种方式,学生不仅掌握了基础技能,还能将Python应用于实际项目中,提升解决问题的能力。

逐步地,学生会形成一种主动学习的模式。由于Python的开源特性,它的更新速度比一般编程语言更快。在学生养成主动学习习惯后,他们能够更好地发现Python广泛的实用性及趣味性,从而提高自我兴趣,主动学习应用Python。这时,Python的实用性不再仅仅是引导,而是有助于学生思维建立和日常学习的得

力助手。最终,形成一个从实用性引导到深度练习,再到自主学习的良性循环,使学生更好地掌握Python编程语言。这种循环不仅提升了学生的技术水平,还培养了他们的创新思维和自主学习能力,为未来的职业发展奠定坚实的基础。



图4 循环促进

在Python编程课程的教学过程中,我们首先注重实用性的引导。许多课程内容偏重理论,学生常常感到困惑,不知如何实际应用。通过强调Python编程语言的实用性,我们让学生切实感受到Python在日常生活和学习中的广泛用途,特别是那些能够立即上手操作的功能。例如,学生可以使用Python编写简单的脚本,自动处理课业中的重复性任务,如数据整理和文件管理。这不仅提高了他们的学习效率,还使他们对编程产生了浓厚的兴趣。

其次,在实用性引导的基础上,我们加强Python编程语言的深度练习。在夯实学生基础知识的同时,不断扩展新的知识点。通过项目实战练习,学生从理解项目需求开始,逐步思考如何实现,进一步分析项目结构,从而理清项目的构思和实现流程。最终,通过编写代码来完成项目要求。在这个过程中,学生不断修正错误,逐渐建立起编程思维,克服对编程语言的畏难情绪,并在一个个项目的成功实现中建立起自信和成就感。例如,在一个学生项目中,他们开发了一个简单的网站,能够展示和处理数据,这不仅巩固了他们的编程技能,还让他们看到了实际成果。

最后,我们鼓励学生自主探索新知识。随着编程思维的建立和自信心的提升,学生对Python的兴趣愈加浓厚,主动学习的意愿也更强。通过项目的练习,学生能够举一反三,逐步拓展Python的应用范围,最终达到我们的课程目标<sup>[8]</sup>。例如,一些学生开始探索数据科学领域,利用Python进行数据分析,甚至参与了一些实际的数据分析项目,这极大地提高了他们的综合能力。

通过这些方法,我们不仅让学生掌握了Python编程的基础知识和技能,还培养了他们的自主学习能力和创新意识,使他们在未来的学习和职业生涯中具备更强的竞争力。

在“Python语言基础与数据分析”的授课过程中,根据课堂情况及课后反馈,我们经历了四个阶段:第一阶段从理论讲解到模仿例题进行程序设计及处理;第二阶段则侧重于大量实验练习及作业完成,再到具体的数据分析;第三阶段涵盖了基本概念讲解、实验练习以及项目应用;第四阶段则采用实用性引导+兴趣激发+项目实战模式,以更有效地激励学生兴趣并提升其综合解析能力。

### 3.1 python实用性引导

由于农业资源与环境专业的大多数同学过去鲜有接触过编程,因此往往不知道它能实现哪些功能。例如,一些简单数学运算,如图形面积体积计算、多元方程求解等都是python的一些基本应用。目前,许多同学使用Office三大办公软件完成日常作业或工作,但对于这些软件却仅停留在最基本操作层面。而实际上,在办公自动化领域,python具有显著优势。在早期教学和实验环节,通过展示python在办公自动化中的强大功能,引起了同学们对此工具关注和兴趣。例如,在第一堂课前,可以简要介绍一些办公自动化实例,让大家直观了解python能够实现什么样功能,比如创建一个2024年包含12个月份表格的Excel文件,仅需如图5中所示几行代码即可迅速生成所需表格,运行的结果如图6显示。从而提高工作效率。这种直观体验能够有效增强他们对pythons的认知,并促使进一步探索<sup>[9]</sup>。

```
File Edit Format Run Options Window Help
import xlwt

x, y = 2023, 0
while x<2024:
    E_1 = xlwt.Workbook()
    x+=1
    while y<12:
        y+=1
        E_2 = E_1.add_sheet('%d 年%d 月'%(x,y))
```

图5 创建表格的代码截图

代码运行的结果为:

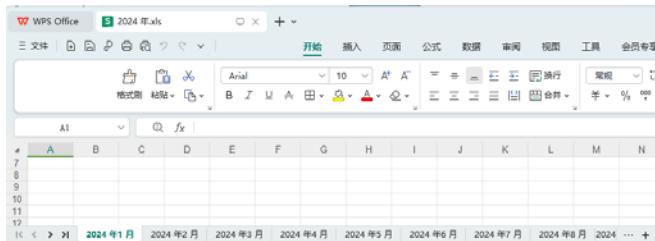


图6 代码运行结果图

通过几行代码快速生成我们日常工作所需表格,提高了办公效率。这种高效便捷的方法有助于提升学生对Python语言的主观认知,并增强其学习兴趣。此外,一些能让学生直观体验到

Python实用性的实例,如PDF文件拆分、合并,以及Excel表格合并等,将在每次课前进行演示和简要讲解。例如,上述实例将在后续章节中详细解释图5中的代码,使同学们理解并掌握while循环语句。

### 3.2 激发学生学习兴趣

通过平时与学生交流及观察发现,当代大学生对手机依赖性较强,且许多同学热衷于玩游戏,他们往往是出色玩家。在学习Python编程过程中,如果加入一些简单游戏代码编写,将有助于激发他们设计小游戏以提高学习兴趣。这不仅能帮助同学们练习编程技能,还能锻炼他们逻辑思维能力。更重要的是,通过提升成就感和参与感,使得同学们对编程产生更浓厚的兴趣,有利于自主学习及深入探索。当教学进入控制语句章节时,可以与同学共同开发一个简单猜拳游戏如图7,同时控制猜拳次数。该程序运行结果见图8,此举旨在帮助同学掌握if和while语句,并分析代码以锻炼逻辑思维能力。通过小游戏来带动学习热情,实现从被动到主动转变<sup>[10]</sup>。

```
File Edit Format Run Options Window Help
import random
i=0
while i<5:
    print("欢迎您的到来")
    My=int(input("请选择: 1. 剪刀, 2. 锤子, 3. 布"))
    if My>0:
        root = random.randint(1, 3)
        while My>0:
            if My==root:
                print("平局, 高手!")
                root = random.randint(1, 3)
            My=int(input())
            elif (My == 1 and root == 2):
                print("锤子!")
                print("我赢了!")
            else:
                print("你赢了!")
    i+=1
```

图7 代码截图

游戏运行结果:

```
===== RESTART: D:\python-310\test\game.py =====
欢迎您的到来
请选择: 1. 剪刀, 2. 锤子, 3. 布
你赢了, 太强了!
1
你赢了, 太强了!
3
你赢了, 太强了!
1
锤子!
我赢了!
```

图8 game运行截图

### 3.3 通过情景式项目实战提高学生python知识综合应用的能力

当同学们掌握了Python基础语法、函数、面向对象编程及模块等知识后,虽然能够完成一些简单程序,但尚未形成系统性的应用认识。因此,仅靠参与项目实践才能加深对整个知识体系归纳理解与综合运用,从而增强自信心。从最基本的一款计算器开始设计,该计算器限定为5次运算,包括加减乘除以及求余求

商等功能(见图9)。还可以进一步设计成为一个电子计算器的模型,实现可视化与人机交互。

```
i=0
while i<5: #应用while循环控制计算次数,
    a=float(input("第1个数字:"))
    b=float(input("第2个数字:"))
    c=input("计算方法:")
    if c=='+':
        result=a+b
    elif c=='-':
        result=a-b
    elif c=='*':
        result=a*b
    elif c=='**':
        result=a**b
    elif c=='/':
        result=a/b
```

图9 简单计算器代码

运行结果如图10:

```
=====
第1个数字: 12
第2个数字: 45
计算方法: +
57.0
第1次计算
第1个数字: 32
第2个数字: 89
计算方法: *
2848.0
第2次计算
第1个数字: 45
第2个数字: 9
计算方法: /
5.0
第3次计算
第1个数字: 24
第2个数字: 5
计算方法: %
4.0
第4次计算
第1个数字: 87
第2个数字: 7
计算方法: //
12.0
第5次计算
>>>
```

图10 简单计算器运行结果

此外,还会涉及通讯录设计,以及当前频繁操作点单系统设计如图11,然后对每日营业额进行分析及可视化处理。通过一系列项目不断练习<sup>[11]</sup>,让同学从身边熟悉场景出发,一步步培养逻辑思维与编程思维,从而更加全面地了解和掌握Python编程语言。

#### 4 结语

农业与资源环境专业开设python语言基础与数据分析这门课程,目的就是为了培养学生对编程语言学习兴趣,培养其数据分析能力、编程过程中解决问题的能力与拓展延伸探索的能力。这种“实用性引导+兴趣激发+情景式项目实战”的模式已在农业资源与环境专业中获得良好反馈,在这种模式下的教学使复杂抽象的编程语言更加贴近生活。同学们能够更积极地参与到

编码设计与数据分析之中,更具生活气息场景使得编程不再遥不可及,而是具体可感,同学们也因此更容易理解接受,并领略到Python语言的强大功能。从简单代码到较复杂的控制语句编程,再到面向对象编程,逐步掌握第三方库及可视化编程,最终达到开设这门课的培养目的。



图11 点单系统

#### 参考文献

- [1]胡艳华,崔亚楠.新工科背景下Python程序设计课程教学改革的研究与实践[J].信息与电脑(理论版),2024,36(21):224-226.
- [2]杨艳玲,彭红霞,彭显琪.循序渐进式教学方法在“Python程序设计”课程中的应用[J].无线互联科技,2024,21(03):95-99.
- [3]王小宁.基于OBE理念的高职Python程序设计课程教学范式改革与实践研究[J].电脑知识与技术,2024,20(24):169-171+174.
- [4]刘盈,谷建涛,闫海波,等.基于OBE理念的Python程序设计课程实践案例教学[J].计算机教育,2023,(01):21-27.
- [5]边小勇,李晨,吴志祥,等.新工科背景下计算机类实践教学创新培养与成绩溯源[J].计算机教育,2023,(06):7-11.
- [6]金传洋,王丽辉,罗晓玲,等.Python语言程序设计课程教学改革与实践[J].电脑知识与技术,2019,15(18):114-115.
- [7]吴纪磊.Python程序设计课程过程化教学改革实践[J].电脑知识与技术,2021,17(25):252-253+280.
- [8]王玉,范冬林,姜建武.Python程序设计课程教学改革与实践[J].科技风,2022,(34):95-97.

[9]庞丽英.Python程序设计课程混合式教学改革实践研究[J].大学,2024,(08):155–158.

[10]郭乐江,涂文婕,程敏,等.基于“一围绕三结合四融入”的Python程序设计基础课程教学改革与实践[J].计算机教育,2022,(06):153–157.

[11]侯晓志.Python程序设计课程教学方法改革实践探究

[J].电脑知识与技术,2024,20(09):129–131.

#### 作者简介:

况江(1984--),男,汉族,贵州省贵定县人,硕士研究生,初级实验师,目前主要研究方向地质勘探,地质灾害,python数据处理,python地理空间分析。