

思维可视化在高中 Python 编程教学中的创新应用

孙雨清

长春市九台区实验高中

摘要：Python 编程涉及的领域极为宽泛，且通用性极强，其不仅是普通高中信息技术课程内容的重要组成部分，而且是增强学生信息素养的有力支撑。但是，由于 Python 编程语言在呈现形式上较为抽象，且对于学生的基本技能、意识、素质等要求很高，以致以思维可视化为导向，去优化、改进、创新高中 Python 编程教学策略，为学生在更为直观、具体、形象、丰富的认知视域内掌握一些关键的语法结构、规则等为奠基，则自然更利于学生编程水平的提高。因此，教师在探究思维可视化应用于高中 Python 编程教学的有效途径时，应以学生为主体，在灵活选用辅助性教学工具、精心设计丰富性教学活动、科学建构多元化学习项目等基础上，就教学重点、难点等以可视化途径呈现至学生视域，对学生进行全方位、深层次启发与引导，来化解教学难度、增强教学实效，为学生信息技术核心素养的塑造而奠基铺路。

关键词：思维可视化；高中 Python 编程教学；创新；应用

【DOI】 10.12252/j.issn.2096-627X.2024.12.143

引言

在新课标思想的支撑、驱动下，实现高中信息技术教学的改革与创新，为学生在多维时空、多维度获得信息技术核心素养培植而提供保障，已成为信息技术课改实践领域广受师生关注的焦点话题。但是，受诸多因素影响，导致很多教师在开展高中 Python 编程教学实践中，依然以简单操作、浏览信息等为侧重点，且对于一些抽象、深奥教学内容的讲解、分析、呈现等途径创新不足。这不仅不利于学生理解、运用 Python 编程语言的能力提升，而且会严重制约课改理念的贯彻。而将思维可视化迁移至高中信息技术 Python 编程教学的各环节、全过程，为学生在可视化实践、体验、探究中去学习 Python 编程语言、增强 Python 编程技能而助力，则势必会为全新教学模式的打造而注入新鲜血液。可以说，思维可视化既是帮助学生深入、高效、精准理解复杂的 Python 编程概念、逻辑、规则的坚实保证，更是创新高中信息技术教学策略的必然选择。基于此，教师应以深入运用思维可视化思想为引领，就高中信息技术教学手段、途径等进行全面创新。且通过对可视化工具的利用、平台的搭建、活动的建构等，为学生在可视化体验、探究、实践中高质量、全方位、深层次掌握 Python 编程规则、技能、素养而提供支持。以通过思维可视化的驱动作用发挥，去推动学生信息技术核心素养的发展，以及信息技术课程教学改革创新。

一、思维可视化概述

思维可视化，既是一种教改思想，又是一种育人理念，其着重强调在教师的适度帮扶、引领下，让学生就自己的思维轨迹、思考过程、探究流程等，以可视化、直观化、

形象化方式予以展示、反馈、呈现，来促使学生思维的活化、理解的深化。这一思维训练方式与教学设计导向，更利于学生对一些抽象概念、逻辑规则等内化，也是促使学生探究意识发展的坚实保证。同时，对教师而言，可借助思维可视化，精准研判、了解、把握不同学生的思维现状，进而借助针对性指导，去推动学生思维短板的弥补、思维优势的最大限度提升。尤其在高中信息技术 Python 编程教学中，实现思维可视化，又是创新教学模式，塑造学生 Python 编程素养的坚实保证。

二、高中 Python 编程的特点剖析

由于 Python 是一种更为高级，且通用性更强的编程语言，以致其本身特点也十分显著，同时对高中学生信息技术核心素养的塑造，也具有积极而重要的影响。具体而言，其特点主要体现在语法更简单、可读性更强、多领域支持、开放源代码等层面。

（一）语法更简单

与其他编程语言相比，Python 语法更加简单明了，且不需学生掌握更为复杂的语法规则。这自然更利于初步接触编程的学生编程技能的增强，以及对一些基本的编程概念的理解。

（二）可读性更强

Python 代码采用了更为简单、易懂的语法结构，其对于学生掌握不同代码的运行程序具有积极意义，同时更利于学生结合需要就不同代码进行反复调试、精准运用。

（三）多领域支持

在 Web 开发、科学计算、数据分析等领域，Python 均具有显著的兼容性。同时，因 Python 对各种类型的程

序均有着极强的支持性，以致涵盖领域也极为宽泛，且在不同应用场景内也体现出了更强的功能与优势。

（四）开放源代码

基于 Python 的开源性特点，以致其拥有极为庞大的开发者社区，且为应用者提供了更为丰富的第三方库与工具，而学生则可借助资源共享的方式，在深层次自学、探究中，不断增强自己的 Python 编程水平。

三、思维可视化在高中 Python 编程教学中的创新应用作用与意义

将思维可视化迁移至高中 Python 编程教学的各环节、全过程，自然更利于传统教学指导措施的革新、改进、优化，而且对学生信息技术核心素养的塑造，也会产生积极而重要的现实意义。

（一）有助于教学模式的创新与优化

受传统思想观念的束缚与制约，很多教师在开展 Python 编程教学中，普遍采用简单程序操作、各类信息反馈等层面。这不仅会严重影响学生创新意识的发展，且不利于学生基本技能的培植。而通过思维可视化的驱动，可为教师精准了解学生思维现状、认知情况而提供参考，以确保教师在教学指导中灵活选用多种方式、途径等，从多层面对学生进行启发与引导，使学生的课堂主体地位得以充分凸显，来保障预设教学目标的达成，以及传统教学模式的改革与创新，进而最大限度提升 Python 编程教学的实效性、针对性。

（二）有助于学生素养的培植与塑造

高中学生 Python 编程素养的提升与增强，本身是一个循序渐进的过程。而在可视化工具、平台等驱动下，学生自主展示、探究、实践、体验的机会也会显著增多。这自然会为学生直观、形象理解学习重点、突破认知困惑而奠基，同时又是保障学生创新意识形成、认知潜能开掘的必然选择。此外，学生在借助可视化手段展示自己思维过程的基础上，其思维活跃度、理解深刻性必然会进一步发展，这又是确保学生 Python 编程素养得以全面塑造的关键所在。

四、思维可视化在高中 Python 编程教学中的创新应用措施

在高中信息技术 Python 编程教学思维可视化创新策略、途径、手段等探究过程中，教师应从深入分析学生认知现状、发展需求出发，通过对可视化平台的搭建、可视化工具的借用、可视化探究的实施、可视化图形的运用等，从多维时空、纵深领域对学生进行诱导与帮扶，来增强教学内容的形象化、丰富性、吸引力，为传统教学模式的创新，以及学生核心素养的培植而拓宽渠道。

（一）借助可视化平台激发学生兴趣，创新教学指导方式

可视化平台的搭建，不仅为传统教学模式的革新提供了支持与铺垫，而且对学生展示、操作拓宽了渠道。因此，教师在开展高中信息技术 Python 编程教学创新实践时，应以实现思维可视化为导向，在充分借用诸如 Scratch、Code.org 等教学平台的基础上，为学生提供图形化展示、直观化编程环境，让学生在操作的基础上，就自己思考、分析过程中的思维轨迹进行呈现，去优化、创新教学模式，以最大限度激发学生的课堂热情。同时，在可视化平台的驱使下，学生可针对具体教学内容，以及所需突破的难点、困境等，利用拖拽、组合等方式，就一些编程代码进行深入理解，这也是适度降低教学难度的必然选择。例如，在进行广东教育出版社的高中信息技术教材一些重点编程概念解析中，可利用电子白板或其他可视化平台，就对应的 Python 编程任务进行直观演示，以帮助学生深入、系统了解对应编程任务中涉及的重点与关键。接着，指引学生通过动手操作、结合平台功能特点等，就自己思考的思路、操作的过程等，在平台内进行展示、分享，且借助教师的针对性点拨、点评，帮助学生突破思维局限、认知困境，进而真正达到实现思维可视化的目的。最后，还可利用平台所提供的教程、示例代码等资源，为学生自主学习提供支持，使抽象的编程概念得以形象化呈现，以确保学生兴趣的激发，以及教学指导方式的创新。

（二）依托可视化工具激活学生思维，创新教学实施途径

高中信息技术教学中，给代码加注释是培育学生 Python 编程素养最直接、最有效的途径之一。而且，学生在加注释的过程中，其思维过程也会直观、明确反馈。因此，教师应在充分借助一系列可视化工具的基础上，针对不同代码所涵盖的信息，以给代码加注释的方式，指导学生将代码的含义、逻辑等，以语言文字的形式表达出来，去加深学生对不同代码内涵的理解，进而实现思维可视化。同时，应依托辅助性可视化工具，让学生就 Python 编程过程中所确立的重要决策、所开展的深入思考、所探寻的实现思路等信息，用注释或图文等记录下来，为学生代码的维护、阅读等能力增强而奠基。特别是针对一些 Python 编程基础比较薄弱的学生，更应以充分运用可视化工具为铺垫，从给代码加注释视角切入，引领学生系统、精准了解对应代码的实现方式、思路等，来推动学生 Python 编程技能的提升，进而实现思维可视化。此外，针对不同学生给代码加注释中所反馈的问题、所出现的缺漏等，也可利用可视化工具进行纠错与指正，

为学生开展个性化思考、自主性探究而助力,以最大限度发挥可视化工具的驱动作用,来保障学生思维深度、广度的增强,以及思维能力的提升。例如,在为4个列表 [1, 2, 3], ['a', 'b', 'c'], ['h', 'e', 'y'] 和 [4, 5, 6] 创建一个新列表时,可通过给代码加注释的方式,就新列表创建过程中涉及的思路进行直观化呈现,据此建立新列表 [1, 'a', 'h', 4], [2, 'b', 'e', 5], [3, 'c', 'y', 6], 为学生精准反馈自己的思维过程而提供参考。

(三) 通过可视化探究深化学生理解,创新教学实践模式

Python 编程中涉及的很多变量、数据结构等,普遍呈现形式极为抽象。而注重对直观图形、辅助资源的利用,为学生系统理解 Python 编程中各类数据的组织形式而奠基,则又是帮助学生精准把握对应变量、数据结构变化过程的有力支撑。因此,教师应在引导学生开展可视化探究的同时,从观察、分析数据的变化处着力,去预判不同数据操作的结果、效果等,来实现思维可视化。而且,可通过对代码执行过程的研判、考量,帮助学生清晰、直观认识变量、数据结构在不同语句和函数间的传递和变化,来增强学生思维的严密性。比如,可引导学生在观察变量的赋值、数据结构的变动等基础上,去把握变量的作用,以及数据结构的操作、使用等,为实现思维可视化而奠基。而学生开展可视化探究中所进行的思考、感知、理解等,又是促使其进行思维转化,且不断矫正 Python 编程中各类谬误的关键所在。此外,通过可视化探究的驱动,可切实丰富学生的 Python 编程体验,这自然更利于教学模式的创新,以及实践路径的拓展,进而为高中信息技术课程教学中 Python 编程指导策略的改革注入新鲜血液,以最大限度凸显学生的课堂主体地位,使学生的实践操作技能得以深度培育。例如,在不同编程变量、数据的分析中,可设置针对性探究任务,去帮助学生实现思维可视化,来保障学生在实践探究中获得编程技能的增强。

(四) 利用可视化图形丰富学生体验,创新教学开展过程

执行路径的可视化,既是确保学生思维、理解能力增强的关键,更是培植学生信息技术核心素养,且创新教学策略的必然选择。因此,教师应以借助图形化展示为支撑,就程序执行的路径、流程等,与一些直观、形象的图形深度衔接起来,去深化对学生的 Python 编程素养塑造。而且,对于程序中各个语句、操作等,也可转化为可视化图形,用图文结合的方式对学生进行启发与诱导,让学生在图形的作用下,去了解程序执行路径、

变量变化等,为实现程序执行过程可视化与学生编程思维可视化而搭建平台。同时,教师更多为学生提供一些自主展示、反馈、分享的机会,倡导学生在呈现思维过程的基础上,运用一些可视化软件,在动态模拟、仿真实操中理解程序的控制流程,使控制流程中包含的条件语句、循环语句、函数调用等被学生深层次、全方位内化。而在学生遇到程序错误或逻辑问题时,则可通过可视化展示程序执行路径、变量变化等,让学生在踊跃尝试、自主调试中去解决问题,为实现思维可视化而拓宽渠道。此外,可结合学生对重点知识、关键内容的掌握程度,多设计一些互动性项目、任务等,为学生在可视化图形的作用下完善认知体系而奠基,来创新 Python 编程教学策略,使学生的认知体验更为丰富。又如,可采用指导学生绘制思维导图的方式,去实现思维可视化,来优化教学设计,强化学生核心素养塑造。

结语

总之,将思维可视化迁移至高中信息技术 Python 编程的全过程、各环节,为学生在可视化视域内深层次、高质量掌握 Python 编程技巧、方法,提高 Python 编程技能、素养而拓宽渠道,则不仅是创新传统教学模式的必然选择,更是促使学生信息技术核心素养塑造的坚实保证。同时,在思维可视化的作用下,很多看似抽象、深奥的教学内容在呈现形式上自会更为形象、丰富、直观、具体,这不仅是推动学生创新意识、编程技能等发展的关键所在,而且对教学目标的达成、学习难点的突破等,也会产生积极而重要的影响。因此,教师在应用思维可视化落实高中信息技术 Python 编程教学创新实践时,应从深入剖析课程标准、教材内容等切入,在充分兼顾学生身心特点、接受能力等基础上,去建构可视化教学活动、项目、任务等,据此对学生帮扶与点拨。让学生在可视化学习体验中,逐步完善认知体系、不断增强核心素养,为高中信息技术 Python 编程教学创新注入新鲜血液。

参考文献

- [1] 陈艺. 基于计算思维培养的高中信息技术教学方式探究——以 Python 语言程序设计教学为例 [J]. 教学管理与教育研究, 2023, 8(24): 109-111.
- [2] 马静. 高中信息技术教学活动中编程 Python 语言应用的探索 [J]. 求知导刊, 2023, (34): 35-37.
- [3] 杨俊凯. 高中信息技术 Python 编程教学微项目学习研究 [J]. 学周刊, 2023, (32): 97-99.
- [4] 李文娟. Python 数据可视化的优势——以《三国演义》为例 [J]. 中国教育技术装备, 2023, (19): 61-65.