计算机科学导论课程中的 Visual Studio Code：经验报告

Jialiang Tan
jtan02@wm.edu
威廉与玛丽学院 弗吉尼亚州，美国

Yu Chen
ychen39@wm.edu 威廉与玛丽学院 弗吉尼亚州，美国

Shuyin Jiao
sjiao2@ncsu.edu 北卡罗来纳州立大学 北卡罗来纳州，美国

arXiv:2303.10174v1 [cs.HC] 10 Mar 2023

摘要

在计算机科学导论课程中引入集成开发环境（IDE）初级（CS1）编程课程至关重要。然而，教师很难找到一款既适合初学者又功能强大的 IDE。本文报告了在一门 CS1 编程课程中使用 Visual Studio Code（VS Code）的经验。我们介绍了选择 VS Code 的动机，以及如何向学生介绍它。我们编制了带有分层索引的全面指南，以帮助具有不同编程背景的学生。我们对学生使用 VS Code 的编程体验进行了实验评估，并验证了 VS Code 配合指南是 CS1 编程课程的一种很有前景的解决方案。

ACM参考格式：
Jialiang Tan、Yu Chen 和 Shuyin Jiao。2023年。Visual Studio Code 在导论计算机科学课程中的应用：一份经验报告。载于*ACM*会议论文集（*Conference'17*）。ACM，纽约，美国，8页。
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 引言

集成开发环境（IDE）在学习编程语言的过程中发挥着重要作用。集成开发环境为程序员提供了丰富的开发能力。它们能够理解语言语法，并提供构建自动化、代码检查、测试和调试等功能，从而加快并简化编码过程。借助集成开发环境，学生能够更高效地进行编程、测试和调试。学生还可以养成更好的编码习惯，并降低学习新语言的难度。因此，越来越多的教师开始在 Python 入门课程中使用集成开发环境，例如 Atom [3]，Jupyter Notebook [2, 24, 36], 等，这显著改善了学生的编程体验。

然而，并非所有集成开发环境都适合入门级 Python 课程。教师很难选择一款令人满意的集成开发环境。一方面，专业的集成开发环境 [11, 12, 39]提供集成的编程环境和许多强大的功能，但对教育的支持有限。要正确使用它们，可能需要具备较高的专业知识并投入大量时间。对于那些刚开始学习编程的学生来说，在学习一门新编程语言的同时，还要充分利用集成开发环境并不容易，因为他们需要花费大量时间进行软件安装和环境配置。另一方面，面向教育的代码编辑器易于安装和操作，但在专业软件开发周期中很少使用。学生从大学迈向业界时会面临很大的差距 [33, 34], ，这会对他们未来的职业生涯产生一些负面影响。

因此，Python 入门课程亟需采用合适的集成开发环境，它不仅要提供足够的教学相关功能，还应是专业软件工程师普遍使用的主流工具。我们希望采用的集成开发环境具备以下四项特点：1）支持跨平台，学生能够在不同操作系统和硬件上以一致的方式安装和使用；2）易于使用，学生可以通过简洁明了的界面高效编写代码；3）功能丰富，例如支持代码编译、项目组织和多语言支持等；4）在专业软件工程师中广受欢迎，这意味着该集成开发环境在业界得到广泛使用，能够拉近入门课程与未来职业之间的距离。

因此，我们选定 Microsoft Visual Studio Code（VS Code）作为理想的集成开发环境，它具备上述所有四项特点。此外，VS Code 已应用于我们院系的许多高级课程中，例如操作系统、编译器构造、计算机网络等。然而，VS Code 在 CS1 课程中并未受到足够关注。此外，根据我们对 20 个计算机科学系的调查，没有一个系将 VS Code 指定为 Python 入门课程的默认集成开发环境。而且，目前也没有面向教育工作者和学生的 VS Code 综合指南。学生的背景各不相同，例如来自不同专业、学习进度不一，或熟悉不同的操作系统，因此，提供全面的指南对于减轻学生和教育工作者双方的负担十分必要。

本文首次报告了在入门级（CS1）Python课程中使用 VS Code 的经验。我们收集了用于 Python教育的插件（即 VS Code扩展）以及各种用户体验，为学生和教育工作者编写了首份全面的 VS Code 指南。该指南高度模块化。具有编程经验的学生可以直接阅读需要学习的章节，跳过已掌握的内容；编程初学者可以按照指南逐步学习；遇到错误的学生可以针对性地了解错误产生的原因及解决方法。借助这份指南，学生可以在课堂上快速掌握编程环境所需的知识。我们已将 VS Code 和这份指南整合到本院系的入门级 Python课程中。我们通过一项调查评估这些工作的成效

允许免费制作本作品的全部或部分内容的数字或纸质副本，用于个人或课堂用途，前提是副本不得用于营利或商业利益，不得为此复制或分发，且副本首页须附上本声明和完整引文。本作品中由 ACM 以外的其他方拥有版权的部分，必须尊重其版权。注明出处后允许摘录。若要以其他方式复制、再版、发布到服务器或分发至列表，须事先获得明确许可和/或支付费用。请向 permissions@acm.org 申请许可。
Conference'17，*2017*年7月，美国华盛顿哥伦比亚特区 © 2023年 美国计算机协会。
ACM国际标准书号 978-x-xxxx-xxxx-x/YY/MM...$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

。调查结果显示，学生高度认可 VS Code 及这份指南。具有一定业界实习经验的学生认为，相较于其他面向教育的集成开发环境，使用 VS Code 学习 Python 更有用，因为 VS Code 在业界得到了广泛应用。

贡献。 我们做出了以下四项贡献。

- 我们指出，为入门课程选择合适的集成开发环境至关重要。• 我们确定 Visual Studio Code 是理想的集成开发环境。我们从四个方面对其进行考察，并分析其在入门级 Python 课程中的实用性。
- 我们提出了首个全面的指导方案[1]，并采用分层索引，以指导背景各异的学生。
- 我们报告了在一门入门级编程课程中使用 VS Code 的经验，并表明 VS Code 是适用于入门级 Python 课程的理想集成开发环境。我们还验证了该指导方案的价值和必要性。

结构。 第2节回顾背景和相关工作。第3节介绍我们对 VS Code 的研究。第4节描述我们为学生提供的指导和支持。第5节展示一些实验性研究。第6节提供讨论。第7节呈现结论和未来工作。

2 背景与相关工作

2.1 Python 教育

Python 已成为大学中最重要的编程语言，位居最常教授的编程语言之首，尤其是在计算机科学入门课程中；这主要归因于三个事实：*Python* 对初学者友好，它采用简化语法，注重使用自然语言，而大多数编程语言规则复杂，因此非常适合初学者学习和练习 [9, 13, 23, 29]。*Python* 是一种强大的工具，它应用于机器学习和大数据等诸多领域，在科学计算和数据分析中尤其有效且不可或缺 [1, 5, 6, 19, 20, 22, 31, 35] 。

背景各异的学生使用 Python 来实现不同目的，例如分析金融市场、模拟生物学中的蛋白质结构。攻读计算机科学或相关专业正规教育的学生，在整个职业生涯中继续使用 Python 的可能性极高。*Python* 很受欢迎，根据 PYPL指数 [25] 和 Stack Overflow 开发者调查 2021 [30], ，截至2022年6月，Python 是全球最受欢迎且增长最快的编程语言（10.4%）。

2.2 教学方法

编程语言在计算机教育中发挥着重要作用。许多系统性研究探讨了入门级编程课程的教学 [14, 21, 27, 28]。

据我们观察，近年来已有大量研究探讨编程环境如何提升学生的编程体验，尤其是针对 Java 和 Python 的编程环境，因为这两种编程语言是大学中教授得最广泛的入门级编程语言。编程环境的选择通常由教师决定。教学工具和智能辅导系统（ITS） [8, 18, 32, 37, 38] 是专为编程初学者设计的。此外，一些研究报告了使用业界级编程环境的情况，例如 Jupyter 笔记本 [2, 4, 10, 36] 或 Eclipse 扩展 [26], ；这些研究表明，优秀的集成开发环境能够提升学生的参与度和表现。

2.3 教育领域中的集成开发环境

集成开发环境（集成开发环境）是指将软件开发项目所需的所有工具整合在一起的软件应用程序，包括编辑器、编译器和调试器。它们汇集了软件开发的不同环节，显著提升了程序员的工作效率。

市场上存在多种编程环境。大致可分为两类：纯文本/代码编辑器和集成开发环境。文本/代码编辑器无需复杂的安装和配置，有些编辑器会预先安装，例如 NotePad 或 TextEdit，但功能有限，且与编程并无直接关联。另一方面，功能齐全的集成开发环境将各种功能整合在一起，提供开发者构建和测试所需的全部工具；但集成开发环境需要更多的磁盘空间/内存或更快的处理器，用户可能还会遇到安装、配置方面的麻烦，甚至需要承担费用（某些集成开发环境的许可证价格昂贵）。不过，两者之间并没有明确的界限。用户可以通过安装插件/扩展，将文本/代码编辑器改造成集成开发环境。在安装/配置所花费的时间与功能的强大程度之间，始终需要权衡。

在编程课程中引入集成开发环境非常重要，尤其是在入门课程中。一款优秀的集成开发环境能从两个方面帮助学生。首先，它能帮助学生编写正确的代码。根据我们以往的教学经验，编程经验较少或没有编程经验的学生会反复遇到一些常见错误。其次，它能帮助学生养成良好的编码习惯。初学者更关注正确性，因此往往会写出质量较低的代码，包括格式杂乱、变量名缺乏意义、函数过长等。许多集成开发环境都支持代码格式化、变量名建议和函数长度警告等功能，帮助学生编写整洁规范的代码，体现符合业界标准的专业素养。

3 Visual Studio Code 调查 ION

Visual Studio（VS）Code 是由微软开发和支持的一款轻量级集成开发环境，可免费用于个人或商业用途。VS Code 的核心特性是支持扩展，用户可以为自己的安装添加语言、调试器和工具，以更好地满足后续开发需求。除了微软发布的标准扩展外，VS Code 扩展 Marketplace [17] 上还有许多由第三方组织和个人开发者提供的扩展。我们从可访问性、易用性、功能性和受欢迎程度四个方面分析它在 Python 入门课程中的实用性。

可访问性。VS Code 支持跨平台，可在多种操作系统上运行，例如 macOS、Windows 和 Linux，并能在大多数可用硬件上提供一致的用户界面。VS Code 的下载体积很小（< 200 MB），磁盘占用也很小（< 500

[1]本文一经录用，VS Code 指南便会公开。

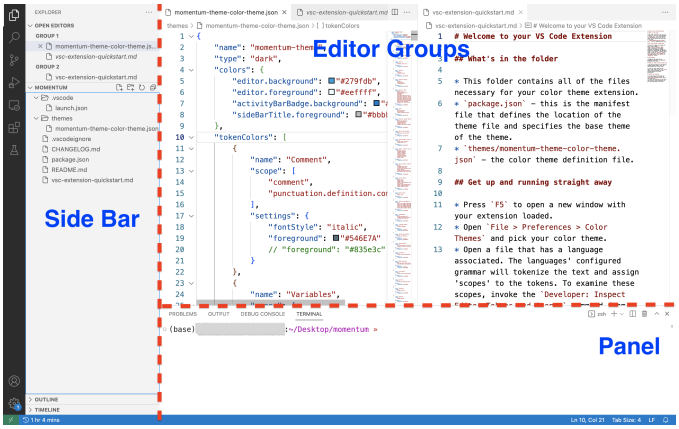

图 1：向学生介绍的 VS Code 用户界面基本布局 [7]。编辑器是用于编辑文件的区域，学生可以同时打开多个编辑器，并以垂直或水平方式并排显示。编辑器下方的面板用于显示输出或调试信息、错误和警告，也可用作集成终端。侧边栏包含不同视图，例如资源管理器或扩展市场，可帮助学生处理项目或下载扩展。

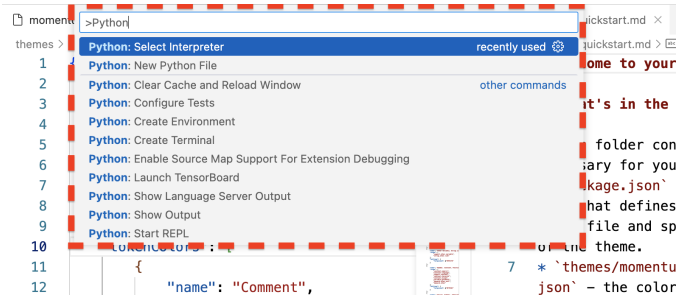

图 2：红框标出了 VS Code 的命令面板。学生可以通过命令面板使用 VS Code 和已安装扩展的所有功能。

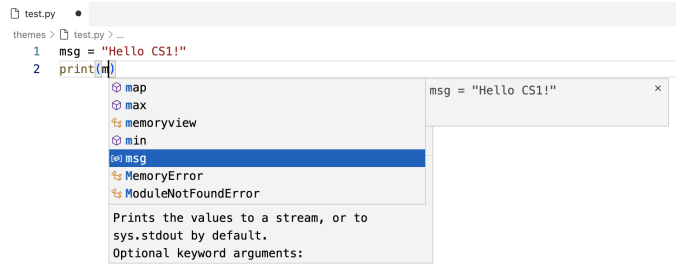

图 3：VS Code 支持代码补全的示例，IntelliSense 建议在 `msg` 变量的`print` 函数中使用该变量。

MB），它轻巧便捷，能完美适配使用不同设备的每位学生。

作为支持多种编程语言的集成开发环境，几乎所有主流编程语言都有对应的扩展支持，因此在不同编程语言之间切换十分轻松。虽然我们的课程面向 Python 初学者，但学生在后续的编程课程中仍可继续使用 VS Code 编写代码。

易于使用。VS Code 的用户界面紧凑而简洁。图 1 展示了一个示例。中央是编辑器，学生可以在其中编写作业代码。编辑器下方是面板，可切换显示调试控制台或内置终端等不同面板；终端始终从特定工作区的根目录启动。左侧是侧边栏，其中包含不同视图；图 1 展示了资源管理器视图，以便更好地帮助用户定位文件。将代码和调试功能集中在一个窗口中，可以有效提高编程效率。

除了简洁的用户界面外，VS Code 还提供了许多功能来提升开发体验。其中一项能简化编程的重要功能是工作区配置设置。VS Code 工作区是当前项目的根文件夹，其中的配置设置仅适用于该工作区，不适用于其他工作区。常见的情况是，学生需要在不同项目中使用不同的设置（例如解释器版本、依赖库、编程语言）。根据以往的经验，学生可能会对配置设置感到困惑，尤其是在开始新项目时设置或重置环境。使用 VS Code 工作区会让工作更加轻松，因为它允许在当前工作区的上下文中配置设置，并始终覆盖全局用户设置。

此外，VS Code 支持远程开发。远程 *SSH* 扩展允许打开运行着 SSH 服务器的任意远程计算机、虚拟机或容器上的远程文件夹。另一个深受许多开发者喜爱的实用功能是命令面板。图 2 展示了一个示例，用户可以通过它访问 VS Code 的所有功能，包括已安装扩展提供的命令。

功能。 除了源代码编辑器的基本功能外，VS Code 还提供扩展，以增强其功能。Marketplace 中有超过 30,000 个扩展，用户可以挑选心仪的扩展并自定义安装，从而改善编程体验。例如，在我们的 Python 课程中，我们建议学生安装微软开发的 Python 扩展，它对 Python 语言提供了强大的支持，具备 IntelliSense、 [15], 代码格式化、调试、变量资源管理器等强大功能。输入时会弹出 IntelliSense 建议，它会根据语言语义和已编写的源代码提供智能代码补全。图 3 展示了一个示例，其中 IntelliSense 建议在 print 函数中使用变量 msg。

受欢迎程度。 VS Code 在实际开发者群体中广泛使用。在 Stack Overflow 2021 年开发者调查 [30], 中，VS Code 是最受欢迎的开发者环境工具，超过 80,000 名受访者中有 71.06% 表示自己在使用它。由于用户数量庞大，错误和问题能够及时修复和解决。VS Code 更新频繁，从 2020 年 6 月到 2021 年 5 月，VS Code 开发团队总共发布了 22 个版本，几乎每个月都会更新。

受欢迎程度高是我们在 Python 课程中引入 VS Code 的原因之一。学生在切换集成开发环境时会遇到困难；使用 VS Code，他们就能在不同的编程语言课程中，以及未来的职业生涯中持续使用它。

4 指导与支持

为了加快不同背景学生学习 VS Code 的进程，我们在 Python 课程中提供了全面的指导。指导[2] 分为若干独立的教程，以下小节将详细介绍各个教程。

VS Code 的下载与安装。 在本教程中，我们会引导学生访问下载页面，并介绍如何根据操作系统选择正确的版本。安装完成后，我们会立即介绍工作区，因为它是操作项目的基本单位。我们会举例说明如何启动 VS Code 项目，以及如何整理项目中的文件。

Python 的下载与安装。 根据以往的教学经验，我们发现 Python 版本方面主要存在三个问题：1. 不了解 Python2 和 Python3 之间的区别；2. 安装了错误的版本；3. 在同一台设备上同时安装了 Python2 和 Python3，却不知道如何在它们之间切换。

在教程中，我们要求学生使用 Python3，并强调 Python2 和 Python3 之间的区别。安装前，学生需要检查是否已预先安装任何版本的 Python。如果没有安装 Python，或安装的是 Python2，我们会引导他们前往最新的 Python3 下载链接。如果他们已安装任何版本的 Python3，则无需再次安装。对于安装了多个版本 Python 的学生，我们会在下一篇教程中提供有关如何切换 Python 版本的详细指南。

设置 Python 环境。接下来，学生在 VS Code 中安装 Python 扩展并设置 Python 环境。根据我们以往的教学经验，学生总是对编程环境设置感到困惑，因此我们会特别说明在哪里以及如何配置或更改设置。

随后，我们会引导学生前往扩展市场，查找并安装由微软开发的 Python 扩展。Python 扩展适用于任何操作系统和 Python 版本，Jupyter 扩展也包含在 Python 扩展的安装包中。Python 扩展基于当前选定的 Python 解释器提供代码补全和 IntelliSense；我们通过配图提供逐步指导，说明如何从命令面板中选择 Python 解释器。我们还会指导学生如何更改编程语言模式，以便希望在后续课程中使用其他编程语言的学生进行操作。

运行 Python 示例。设置好 Python 环境后，VS Code 就成为真正的 Python 集成开发环境。在本教程中，我们将指导学生从头编写并运行他们的第一个简单 Python 程序，并说明 Python 扩展如何改进编程。我们要求学生为示例程序生成预期输出，以确保他们使用相同的 Python 环境设置。

[2]出于匿名审稿的考虑，论文中未展示指导材料的图片；论文录用后，这些指导材料将公开。

安装并运行 Jupyter 扩展。我们使用 VS Code 的 Jupyter 扩展运行课堂编程练习，以提高学生的参与度 [2, 24], 。通过 Jupyter 扩展，学生可以像打开和运行 Python 文件一样，在 VS Code 中打开并运行 ipynb 文件。Jupyter Notebook 扩展为语言内核提供基本的笔记本支持，并允许将任何 Python 环境用作 Jupyter 内核。在本教程中，我们将指导学生安装 Jupyter 扩展、创建新的 Jupyter 文件，以及管理和运行代码单元。

安装并使用科学计算包。 Python 包在解决科学计算、数据可视化、数据处理及许多其他领域的复杂问题时非常高效。在本教程中，我们指导学生安装并导入 Python 库，例如 NumPy、Pandas、Matplotlib 和 SciPy。随后，学生需要学习并使用基本 API，并提供了示例代码。要顺利完成本教程，学生需要使用 Matplotlib 函数绘制特定图形。

4.1 指导材料的改进

为更好地指导学生，我们对指导材料进行了如下优化：

- 我们分别为 macOS 用户和 Windows 用户提供两个版本的指导材料。根据我们的经验，为所有操作系统提供单一版本的指导材料并不够；在入门课程中，操作系统之间的细微差异也很重要。
- 我们提供三种获取指导材料的方式：下载 PDF、访问网站和观看视频。传统方式是将指导材料以 PDF 格式上传至课程学习管理系统，但 PDF 格式的呈现形式有限，因此我们还提供了网站版指导材料。由于 COVID-19，学生可能无法获得充足的面对面教学，因此我们还提供视频指导，以提高学生的参与度。
- 我们持续收集学生反馈，并定期改进和更新指导内容。

4.2 分层索引

我们会详细介绍教程，学生按照教程逐步操作，便可轻松配置编程环境。然而，学生对教程的偏好因其编程背景而异。扁平结构的教程无法满足所有学生的需求。偏好视频教程的学生可能不会从头开始跟随视频的节奏。常见的情况是先观看口头演示，遇到问题时再参考视频。如果教程采用扁平结构，就很难根据文本定位到相应的视频时间戳。此外，具备一定编程知识的学生需要更简洁的信息，例如，他们已安装 Python 和 VS Code，但仍需要了解如何在 VS Code 中配置 Python 环境；扁平结构的教程无法提供这类指导。对于已经熟悉环境配置的学生而言，他们很难从扁平结构的教程中提取关键信息。

我们为教程设计了分层索引，以满足多样化的需求。分层索引包含三个层级：1）高层：我们在每个教程开头提供摘要，列出所有关键信息，例如软件名称、下载

链接、软件版本等。2）中层：我们将每个教程分成若干部分，并通过副标题进行概述。我们列出副标题，并链接到教程中的相应位置。学生无需费力即可找到所需信息。3）低层：我们建立视频与口头演示之间的对应关系，即将副标题中每个步骤的口头演示链接到相应的视频时间戳。因此，遇到问题时，学生可以在口头演示和视频演示之间切换。

5 评估与学生反馈

本节评估以下两个目标：

- 验证 VS Code 是否适用于入门级 Python 编程课程
- 验证 VS Code 指导功能的价值与必要性

5.1 课程介绍

2022年秋季，我们在一门 CS1 Python 编程课程中引入了 VS Code 和 VS Code 指南，该课程共有 141 名学生。本 CS1 课程仅面向非计算机专业学生，重点培养基本编程技能并应用于工程领域。学生学习如何通过编写 Python 程序解决问题，具体内容包括：根据规格说明开发 Python 程序；文档编写与编码风格；数据类型、控制结构和数据结构的使用；抽象与验证。

本课程采用翻转课堂形式。每周，学生首先通过观看视频并在线完成自测测验，学习本周主题的概念知识。随后，他们参加一节 100 分钟的课堂练习课和一节 165 分钟的实验练习课，参与主动学习活动。课后，学生完成每周的作业和项目任务，以巩固本周所学的知识与技能。学生必须通过安装在 VS Code 中的 Jupyter Notebook 扩展运行课堂编程练习，并独立使用 VS Code 完成编程任务（作业、实验练习和项目）。

5.2 VS Code 与 VS Code 指南评估

我们设计了一份包含 21 个问题的调查，在学期末收集学生对 VS Code 和 VS Code 指南的反馈。调查首先了解学生的背景，然后请学生根据整个学期的编程体验，对 VS Code 和 VS Code 指南进行评分。我们总共收集到 42 份有效答卷。

首先，我们收集学生的背景信息。作为一门典型的 CS1 课程，79% 的学生是大一或大二学生，86% 的学生此前没有上过任何编程课程，74% 的学生编程体验不足一年。

接下来，请学生评价整个学期中 VS Code 和 VS Code 指南的表现。他们从以下四个方面对 VS Code 进行评分，满意度范围为 1 到 5，其中 1 表示非常不满意，5 表示非常满意。

- 视觉吸引力：评价 VS Code 的用户界面和编辑器布局。
- 扩展生态系统：评价搜索、安装和卸载扩展的方式。
- 调试体验：评价 VS Code 的内置调试器，看看它是否有助于加快学生的编辑、编译和调试循环，以及推荐的调试器扩展是否有帮助。
- 编辑体验：评价 VS Code 的整体编辑体验。如果基本编辑功能（即键盘快捷键和命令面板）实用且对初学者友好。还请评价 IntelliSense，以及代码补全、参数信息和内容辅助等代码编辑功能是否有帮助。

表 1：42 名学生对四个方面的满意度平均得分：视觉吸引力、扩展生态系统、调试体验和编辑体验。满意度分值介于 1 到 5 之间，其中 1 表示非常不满意，5 表示非常满意。

方面	视觉吸引力	扩展	调试	编辑
平均评分	4.17	3.81	4.02	4.05

根据调查结果，61% 的学生过去没有使用集成开发环境或编码平台的经验，39% 的学生使用过其他集成开发环境，例如 Atom、Spyder、PyCharm/CLion/IntelliJ 等。如表 1 所示，视觉吸引力、扩展生态系统、调试体验和编辑体验的平均评分分别为 4.17、3.81、4.02 和 4.05。

学生对 VS Code 指南的总体满意度为 4.2。在 VS Code 指南中的详细教程帮助下，74% 的学生认为 VS Code 安装和使用起来相当容易，76% 的学生认为 VS Code 是一款适合编写代码的集成开发环境。有 13 名学生表示，他们在整个学期使用 VS Code 指南的过程中遇到了问题或困难；我们收集了这些问题，并在 VS Code 指南中进行了修复或更新。

5.3 班级平均成绩

我们还比较了2022年秋季与2022年春季的班级平均成绩，结果如表2所示。2022年秋季的班级平均总分为85.10%，所有编程作业的班级平均成绩为89.31%；而2022年春季这两项数据分别为82.86%和88.37%。

表2：2022年春季和秋季学期编程作业与总分的班级平均成绩。

班级平均成绩	编程作业	总分
2022年春季	88.37%	82.86%
2022年秋季	89.31%	85.10%

尽管表2的比较显示，2022年秋季的班级平均成绩高于2022年春季，但我们不能简单地将平均成绩的提升归因于 VS Code 或 VS Code 指南。2022年春季，我们采用传统课堂，要求学生使用集成于 Anaconda 中的 Spyder；而在2022年秋季，我们采用翻转课堂并使用 VS Code。不过，根据学生的体验，我们可以得出结论：VS Code 和 VS Code 指南都具有积极作用，有助于完成编程作业。

5.4 学生评论

在学生调查结束时，我们邀请学生对整个学期使用 VS Code 以及 VS Code 指南的体验发表评论。一名学生这样描述他/她的总体体验：

> 使用 VS Code，无需新建窗口，我就能同时查看 Python 文件和文本文件。界面在视觉上很吸引人。VS Code 教程简单明了，说明也很容易理解。在我看来，这就是一份教程真正需要做到的全部。

另一位学生尤其喜欢 VS Code 的用户界面：

> 我喜欢 VS Code 的易用性，整体界面通俗易懂，让我们能够更专注于学习和应用 Python概念。

一些学生更喜欢 VS Code 的深色模式，一位学生表示，这能提升他/她的编程体验：

> 我喜欢 VS Code 用颜色区分不同的代码类型和函数，让它们更加醒目。我也喜欢默认的黑色背景，这让代码更容易阅读。90% 的时候，都很容易在代码中找到错误。总的来说，这份教程很容易理解。

我们观察到，相比亮色模式，许多学生更喜欢深色模式。这可能是因为深色模式或深色主题能让语法高亮更加醒目。在浅色背景下，代码文本大多使用较深的颜色；而在深色背景下，代码文本的颜色则更加丰富。此外，与教学型集成开发环境相比，在 VS Code 中更换主题相当容易。学生们对 VS Code 中的主题扩展赞不绝口，一位学生表示：

> 我喜欢 VS Code 安装扩展的方式，尤其喜欢 Marketplace 中的主题扩展。每当我厌倦了代码，就会更换主题，这样它就变成了一个全新的软件！

一些学生喜欢 VS Code 指南，简单地这样说道：

> 我很满意这些步骤写得如此清晰，而且大多数步骤都配有相应的屏幕截图。

我们还发现，学生喜欢 VS Code 指南中的分层索引，一位学生这样评论道：

> 内容很容易跟着做，链接也嵌入在各个步骤中。此外，每个步骤都配有图片和屏幕截图来提供指导，这有助于我提前准备
> 好这份材料，以便上课前使用 .

根据反馈，我们收集到了许多关于 VS Code 和 VS Code 指南的建设性建议。一些学生抱怨终端中的命令过于复杂：

> 终端指令复杂且难以理解。

的确，我们可能会在课程开始时涉及大量终端指令，但教师认为这些指令/命令是本课程所必需的。为了解决这个问题，我们将命令学习内容分配到不同的实验课中，并为每条指令添加详细说明。

另一位学生认为 VS Code 指南很有用，但建议我们可以在课堂上使用更多辅助扩展：

> 教程内容详尽，直截了当。但我认为 VS Code 缺少协作功能，尤其是在实验课上。只需花大约 1 分钟安装一个扩展，你和实验搭档就可以协作处理同一个 Python 文件。我和实验搭档在课程的实验环节进行调试和编程练习时，就使用了这个扩展。我觉得通过“共享”文档进行协作会很有帮助，因为没有它的话，通常只有一个人在独自编写代码。

Live Share 扩展使学生能够共享屏幕，并与同学/助教/教师协作编写同一份代码，而无需同步代码，也不必配置相同的开发工具、设置或环境 [16]。我们正在评估此功能，经过慎重考虑后将向学生介绍。

总而言之，学生对 VS Code 和 VS Code 指南的整体体验积极且令人鼓舞，这表明 VS Code 及其指南对 CS1 课程具有价值和前景。

6 讨论

关于教育相关问题的讨论。 VS Code 具备一些高级功能，例如代码自动补全和函数签名提示，这些功能在入门级编程课程中的应用引发了一些争论。有人认为，辅助功能会让人盲目依赖集成开发环境，而不是真正理解代码。在实践中，我们的教学计划涵盖了所有必要的编程知识和概念，而自动补全和函数签名提示等功能则有助于学生更快、更高效地学习。此外，如有需要，也可以在 VS Code 中自定义这些功能或将其完全禁用。

关于一些局限性的讨论。我们可以看到本研究存在一些局限性。我们只评估了一门 Python 编程入门课程，原因是我们的院系将 Python 作为编程入门语言。不过，VS Code 支持多种编程语言，例如 Matlab、Java 和 C/C++。将我们的指南调整为适用于使用其他编程语言的课程并不困难。我们将与院系的教师合作，报告更多使用 VS Code 的经验。

7 结论与未来工作

本文介绍了在一所大型工科大学的 Python 编程入门（CS1）课程中引入 Visual Studio Code 的经验。本文探讨了 VS Code 是否适合作为 CS1 编程课程的集成开发环境。为了更好地帮助学生，我们为具有不同编程背景的学生编写了全面的 VS Code 指南。我们对学生进行了评估，验证了 VS Code 的实用性，并检验了 VS Code 指南的质量。我们根据收集到的反馈定期更新和完善指南。目前，我们正在更多使用 Python 以外编程语言的 CS1 编程课程中使用 VS Code 和我们的 VS Code 指南。

我们未来的工作分为两个方面。首先，我们将继续探索 Marketplace 上有助于教育的实用 VS Code扩展。其次，我们将开发实用的 VS Code扩展，进一步支持教育相关活动。具体而言，

我们将开发能够让学生在完成编程作业时更积极参与的扩展。

参考文献

[1] Martín Abadi、Ashish Agarwal、Paul Barham、Eugene Brevdo、Zhifeng Chen、Craig Citro、Greg S. Corrado、Andy Davis、Jeffrey Dean、Matthieu Devin、Sanjay Ghemawat、Ian Goodfellow、Andrew Harp、Geoffrey Irving、Michael Isard、Yangqing Jia、Rafal Jozefowicz、Lukasz Kaiser、Manjunath Kudlur、Josh Levenberg、Dandelion Mané、Rajat Monga、Sherry Moore、Derek Murray、Chris Olah、Mike Schuster、Jonathon Shlens、Benoit Steiner、Ilya Sutskever、Kunal Talwar、Paul Tucker、Vincent Vanhoucke、Vijay Vasudevan、Fernanda Viégas、Oriol Vinyals、Pete Warden、Martin Wattenberg、Martin Wicke、Yuan Yu 和 Xiaoqiang Zheng。2015。TensorFlow：异构系统上的大规模机器学习。https://www.tensorflow.org/ 软件可从 tensorflow.org 获取。[2] Abdulmalek Al-Gahmi、Yong Zhang 和 Hugo Valle。2022。课堂中的 Jupyter：一份经验报告。载于第 53 届 ACM计算机科学教育技术研讨会论文集，第 1 卷。425–431。[3] Atom。2022。Atom 主页。https://atom.io/。[4] Lorena A Barba、Lecia J Barker、Douglas S Blank、Jed Brown、Allen B Downey、Timothy George、Lindsey J Heagy、Kyle T Mandli、Jason K Moore、David Lippert 等。2019。使用 Jupyter 进行教与学。访问地址：https://jupyter4edu.github.io/jupyter-edu-book（2019）。[5] Yu Chen 和 Zhenming Liu。2020。关于检查点的高效构造。载于第 37 届 2020 年国际机器学习会议（ICML），2020 年 7 月 13–18 日。[6] Yu Chen、Ivy B Peng、Zhen Peng、Xu Liu 和 Bin Ren。2020。Atmem：异构内存上图应用的数据自适应放置。载于第 18 届 ACM/IEEE国际代码生成与优化研讨会论文集。293–304。[7] Visual Studio Code。2022。用户界面。https://code.visualstudio.com/docs/getstarted/userinterface。[8] Tyne Crow、Andrew Luxton-Reilly 和 Burkhard Wuensche。2018。面向编程教育的智能辅导系统：系统性综述。载于第 20 届澳大拉西亚计算教育会议论文集。53–62。[9] Vineesh Cutting 和 Nehemiah Stephen。2021。关于将 Python 作为初学者首选编程语言的综述。（2021）。[10] Roland DePratti。2019。在大数据编程课程中使用 Jupyter Notebook。《大学计算科学杂志》34，6（2019），157–159。[11] Eclipse基金会。2022。Eclipse 主页。https://www.eclipse.org/ide/。[12] JetBrains。2022。软件开发人员和团队的必备工具。https://www.jetbrains.com/。[13] Selina Khoirom、Moirangthem Sonia、Borishphia Laikhuram、Jaeson Laishram 和 Tekcham Davidson Singh。2020。面向初学者的 Python 与 Java 比较分析。《国际工程与技术研究期刊》7，8（2020），4384–4407。[14] Rodrigo Pessoa Medeiros、Geber Lisboa Ramalho 和 Taciana Pontual Falcão。2018。高等教育中编程入门教学与学习的系统性文献综述。IEEE教育汇刊 62，2（2018），77–90。[15] 微软。2022。IntelliSense。https://code.visualstudio.com/docs/editor/intellisense。[16] 微软。2022。什么是 Visual Studio Live Share。https://learn.microsoft.com/en-us/visualstudio/liveshare/。[17] Mircosoft。2022。VS Code Marketplace。https://marketplace.visualstudio.com/。[18] Samy S Abu Naser。2008。为学习 C++ 编程的学生开发智能辅导系统。《信息技术期刊》，Scialert 7，7（2008），1055–1060。[19] Travis E Oliphant。2006。《NumPy 指南》。第 1 卷。Trelgol Publishing 美国。[20] Adam Paszke、Sam Gross、Soumith Chintala、Gregory Chanan、Edward Yang、Zachary DeVito、Zeming Lin、Alban Desmaison、Luca Antiga 和 Adam Lerer。2017。PyTorch 中的自动微分。（2017）。[21] Arnold Pears、Stephen Seidman、Lauri Malmi、Linda Mannila、Elizabeth Adams、Jens Bennedsen、Marie Devlin 和 James Paterson。2007。关于编程入门教学的文献调查。ITiCSE 创新与计算机科学教育技术工作组报告（2007），204–223。[22] F. Pedregosa、G. Varoquaux、A. Gramfort、V. Michel、B. Thirion、O. Grisel、M. Blondel、P. Prettenhofer、R. Weiss、V. Dubourg、J. Vanderplas、A. Passos、D. Cournapeau、M. Brucher、M. Perrot 和 E. Duchesnay。2011。Scikit-learn：Python 中的机器学习。《机器学习研究期刊》12（2011），2825–2830。[23] Jean-Philippe Pellet、Amaury Dame 和 Gabriel Parriaux。2019。编程语言对初学者有多友好？基于 Java 和 Python 示例的简要分析。（2019）。[24] Fernando Perez 和 Brian E Granger。2015。Project Jupyter：以计算叙事为引擎的协作式数据科学。访问于 2015 年 9 月 11 日，207 2015 108( )，。[25] PYPL。2022。编程语言受欢迎程度。http://pypl.github.io/PYPL.html。[26] Charles Reis 和 Robert Cartwright。2004。让专业集成开发环境适用于课堂。载于第 35 届 SIGCSE 计算机科学教育技术研讨会论文集。156

[27] Norsaremah Salleh、Emilia Mendes 和 John Grundy。2010。高等教育中用于计算机科学/软件工程教学的结对编程实证研究：系统性文献综述。*IEEE*软件工程汇刊37，4（2010），509–525。[28] Carsten Schulte 和 Jens Bennedsen。2006。教师在编程入门课程中教什么？载于计算教育研究第二届国际研讨会论文集。17–28。[29] KR Srinath。2017。Python——增长最快的编程语言。国际工程与技术研究期刊4，12（2017），354–357。[30] stackoverflow。2021。2021 年开发者调查。https://insights.stackoverflow.com/survey/2021#section-most-popular-technologies-integrated-development-environment。[31] Jialiang Tan、Yu Chen、Zhenming Liu、Bin Ren、Shuaiwen Leon Song、Xipeng Shen 和 Xu Liu。2021。促进 Python 与本机库之间的高效交互。载于第 *29* 届 *ACM* 欧洲软件工程会议与软件工程基础研讨会联合会议论文集。1117–1128。[32] Python Tutor。2021。学习 Python、JavaScript、C、C++ 和 Java。https://pythontutor.com/。[33] Sander Valstar、Sophia Krause-Levy、Alexandra Macedo、William G Griswold 和 Leo Porter。2020。教师对本科计算机科学教育目标的看法以及学术界与产业界的差距。载于第*51*届*ACM*计算机科学教育技术研讨会论文集。577–583。[34] Sander Valstar、Caroline Sih、Sophia Krause-Levy、Leo Porter 和 William G Griswold。2020年。关于计算机科学本科项目目标及培养学生进入业界就业的师资观点的定量研究。载于*2020*年*ACM*国际计算教育研究会议论文集。113–123。[35] Stefan Van Der Walt、S Chris Colbert 和 Gael Varoquaux。2011 年。NumPy 数组：一种高效数值计算的数据结构。《计算科学与工程》13, 2（2011），22。[36] Eric Van Dusen。2020年。用于数据科学教学的Jupyter。载于第*51*届*ACM*计算机科学教育技术研讨会论文集。1399–1399。[37] Kelsey Van Haaster 和 Dianne Hagan。2004年。使用BlueJ进行教学与学习：一项教学工具评估。《信息科学与信息技术议题》1（2004）。[38] Kurt VanLehn。2011年。人工辅导、智能辅导系统与其他辅导系统的相对有效性。《教育心理学家》46, 4（2011），197–221。[39] Vim。2022年。Vim编辑器。https://www.vim.org/。