

2024-2025 学年初中信息技术（信息科技） 八年级下册滇人版（2016）教学设计合集

目录

一、第一单元 Python 程序设计

- 1.1 第 1 课 初识 Python
- 1.2 第 2 课 顺序结构程序设计
- 1.3 第 3 课 分支结构程序设计
- 1.4 第 4 课 多分支结构程序设计
- 1.5 第 5 课 for 循环结构程序设计
- 1.6 第 6 课 while 循环结构程序设计
- 1.7 第 7 课 循环应用与函数初识
- 1.8 第 8 课 斐波那契数列及图形设计
- 1.9 本单元复习与测试

二、第二单元 初识机器人

- 2.1 第 9 课 初识机器人
- 2.2 第 10 课 搭建第一个机器人
- 2.3 第 11 课 “绝壁求生”——避障机器人
- 2.4 第 12 课 “循迹夺宝”——循迹机器人
- 2.5 本单元复习与测试

第一单元 Python 程序设计第 1 课 初识 Python

科目		授课时间节次	--年—月—日（星期一）第一节
指导教师		授课班级、授课课时	
授课题目 (包括教材 及章节名称)	第一单元 Python 程序设计第 1 课 初识 Python		

教学内容	初中信息技术（信息科技）八年级下册滇人版（2016）第一单元 Python 程序设计第 1 课 初识 Python，主要包括以下内容： 1. Python 语言的简介：了解 Python 的历史、特点和应用领域。 2. Python 程序的基本结构：掌握 Python 程序的基本组成元素，如变量、表达式、语句等。 3. 安装 Python 环境：学会在计算机上安装 Python 编程环境，了解 Python 的开发工具。 4. 编写第一个 Python 程序：通过简单的示例，学习编写和运行 Python 程序的方法。 5. 基本输入输出函数：学习使用 <code>print()</code> 和 <code>input()</code> 函数进行简单的输入输出操作。
核 心 素 养 目 标	1. 提升信息意识，能够主动关注 Python 语言在现代信息技术中的应用。 2. 培养计算思维，通过编写简单的 Python 程序，锻炼逻辑思维与问题解决能力。 3. 增强信息处理能力，学会使用 Python 进行基本的数据输入输出操作，提高数据处理效率。 4. 发展创新精神，通过 Python 编程实践，激发创造力和创新意识。
教学难点与重点	1. 教学重点 - Python 环境的搭建：本节课的核心内容之一是教会学生如何在计算机上安装 Python 环境，包括 Python 解释器和开发工具（如 IDLE），这是编写和运行 Python 程序的基础。 - 基本语法结构：掌握 Python 的基本语法结构，包括变量定义、表达式、语句等，是编写 Python 程序的基础，例如： <pre> python # 变量定义与赋值 x = 5 y = "Hello, World!" # 表达式与输出 print(x + </pre>

	2) <pre>print(y) '''</pre> - 简单程序的编写与执行：通过编写简单的程序，如“Hello, World!”程序，让学生理解 Python 程序的执行过程和基本结构。 2. 教学难点 - 环境搭建的细节：学生在安装 Python 环境时可能会遇到各种问题，如操作系统兼容性、路径设置等，需要教师耐心指导，确保每个学生都能成功搭建环境。 - 语法规则的掌握：Python 的语法规则对学生来说可能比较抽象，如缩进的使用、变量命名规则等，这些规则的正确应用是编程的关键。例如，学生在编写程序时可能会忘记缩进或使用错误的变量名： <pre>```python # 错误的缩进示例 def greet(name): print("Hello, " + name) # 应为四个空格或一个制表符 print("Goodbye!") # 错误的变量命名示例 123name = "Alice" # 变量名不能以数字开头 '''</pre> - 程序调试与错误处理：学生在编写程序时可能会遇到错误，如语法错误、逻辑错误等，需要学会如何调试程序并找出错误原因。例如，理解并使用 Python 的错误提示信息来定位问题。
教学资源准备	1. 教材：确保每位学生都配备了《初中信息技术（信息科技）八年级下册滇人版（2016）》教材，以便于学生跟随课程进度学习。 2. 辅助材料：准备与 Python 程序设计相关的 PPT 演示文稿，以及在线编程环境（如 Repl.it）的访问权限，以便于直观展示和互动编程。 3. 实验器材：确保教室内的计算机设备正常运行，并安装有 Python 环境，以便学生进行编程实践。 4. 教室布置：将学生分成小组，每组一台或多台计算机，以便于学生进行小组讨论和编程练习。
教学流程	1. 导入新课（5 分钟） - 通过展示一些使用 Python 编写的有趣程序或应用（如简单的图形绘制、小游戏等），激发学生的兴趣。 - 简要介绍 Python 语言的特点，如简洁易读、功能强大等，引出本节课的主题“初识 Python”。 2. 新课讲授（15 分钟） - 介绍 Python 的历史和发展，让学生了解 Python 在现代信息技术中的应用。

	讲解 Python 环境搭建的步骤，包括 Python 解释器和开发工具的安装，以及如何打开和配置开发环境。 - 通过示例代码，展示 Python 的基本语法结构，如变量定义、表达式、语句，以及基本的输入输出函数。 <pre>```python # 变量定义与输出 x = 10 print("x 的值是:", x) # 输入函数 name = input("请输入你的名字:") print("Hello, " + name + "!") ```</pre> 3. 实践活动（15 分钟） - 指导学生在计算机上安装 Python 环境，并打开 IDLE 或其他开发工具。 - 让学生尝试编写并运行一个简单的打印“Hello, World!”的 Python 程序。 - 引导学生使用 input() 和 print() 函数编写一个简单的交互式程序，如询问用户姓名并打印问候语。 4. 学生小组讨论（10 分钟） - 讨论 Python 环境安装过程中遇到的问题及解决方法。 - 分析示例程序中的语法错误，如缺少缩进、错误的变量命名等，并讨论如何避免这些错误。 - 探讨 Python 编程在实际生活中的应用，如数据分析、网站开发等，举例说明 Python 如何帮助解决实际问题。 5. 总结回顾（5 分钟） - 复习本节课的主要内容，包括 Python 环境的搭建、基本语法结构、输入输出函数。 - 强调 Python 编程中的注意事项，如缩进的正确使用、变量命名规则等。 - 鼓励学生在课后继续探索 Python 编程，尝试编写更有趣的程序，并预告下一节课的内容。
教学资源拓展	1. 拓展资源 - Python 语言的广泛应用：介绍 Python 在科学计算、数据分析、人工智能、网络编程、自动化脚本等领域的应用，以及相关的库和工具，如 NumPy、Pandas、TensorFlow、Django 等。 - Python 编程实践项目：提供一些适合初学者的 Python 编程项目，如制作简单的计算器、数据可视化、网页爬虫等，让学生在实践中学学习。 - Python 编程规范与最佳实践：介绍 Python 的 PEP 8 编码规范，以及编写可读性和可维护性强的代码的最佳实践。 2. 拓展建议

	阅读 Python 官方文档：建议学生阅读 Python 的官方文档，以获取最权威的语言特性和库的使用方法。 - 参与在线编程社区：鼓励学生加入如 Stack Overflow、GitHub、CSDN 等在线编程社区，与其他 Python 开发者交流问题解决方案。 - 完成在线编程挑战：推荐学生参与如 LeetCode、Codeforces 等在线编程挑战平台，通过解决算法问题来提高编程能力。 - 阅读 Python 编程书籍：推荐学生阅读一些经典的 Python 编程书籍，如《Python 编程：从入门到实践》、《流畅的 Python》等，以加深对 Python 的理解。 - 观看教学视频：建议学生观看 MOOC 平台上的 Python 编程课程，如 Coursera、edX 上的相关课程，以获得不同视角的教学内容。 - 实施个人项目：鼓励学生根据自己的兴趣和需求，设计并实施个人项目，将所学知识应用于实际问题中，如开发一个小游戏、编写数据分析报告等。 - 参与编程竞赛：鼓励学生参加学校或地区举办的编程竞赛，如 ACM ICPC、NOIP 等，通过竞赛锻炼编程能力和解决问题的能力。 - 定期复习与总结：建议学生定期复习所学内容，并总结学习经验，形成自己的编程笔记，加深对知识点的理解和记忆。
课堂	1. 课堂评价 - 提问：在教学过程中，通过提问的方式检查学生对 Python 环境的搭建、基本语法结构和输入输出函数的理解程度。例如，教师可以询问学生如何安装 Python 环境，如何定义一个变量，以及如何使用 print() 和 input() 函数。 - 示例问题：“谁能告诉我如何在计算机上安装 Python 环境？”“如何定义一个整型变量并打印它的值？” - 观察：教师在学生实践环节观察学生的操作，了解他们是否能够独立完成环境搭建和程序编写。同时，注意观察学生是否能够正确使用缩进和遵循变量命名规则。 - 测试：在课程结束时，进行一次小测验，测试学生对本节课内容的掌握情况。测试可以包括选择题、填空题和编程题，如编写一个简单的交互式程序。 - 示例测试题：“以下哪个选项是正确的 Python 变量命名？A. 1name B. name1 C. name_1 D. name\$” 2. 作业评价 - 批改：认真批改学生的作业，检查程序是否能够正确运行，代码是否符合 Python 的语法规则和编程规范。 - 点评：在作业批改后，对学生的作业进行点评，指出作业中的优点和需要改进的地方。对于共性问题，可以在课堂上集中讲解，帮助学生理解和纠正。 - 反馈：及时将作业评价结果反馈给学生，鼓励他们根据反馈进行自我调整和改进。对于表现优秀的学生，可以给予表扬和鼓励，激发他们的学习热情。 - 鼓励：在评价过程中，注重鼓励学生，特别是对于遇到困难的学生，要给予耐心指导和支持，帮助他们克服学习中的障碍。 3. 评价标准

	- 正确性：学生的代码是否能够正确运行，输出结果是否符合预期。 -
--	---

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。
。如要下载或阅读全文，请访问：

<https://d.book118.com/368110115017006135>