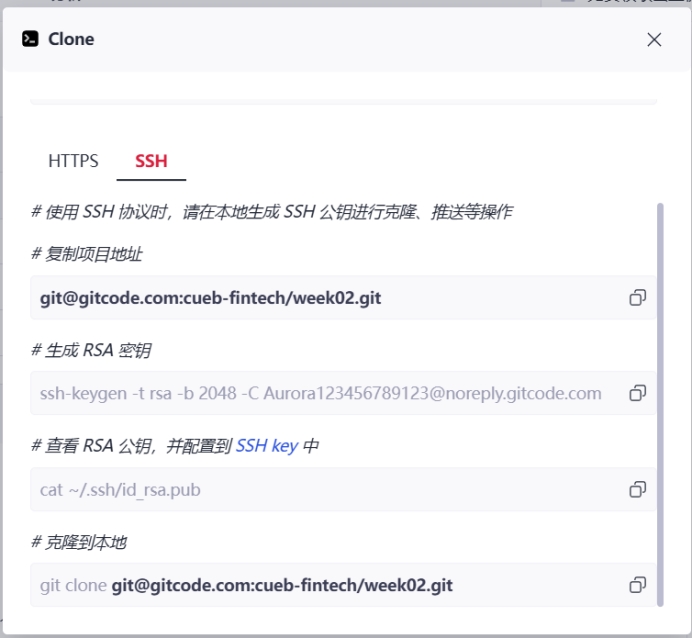
一、课程概述

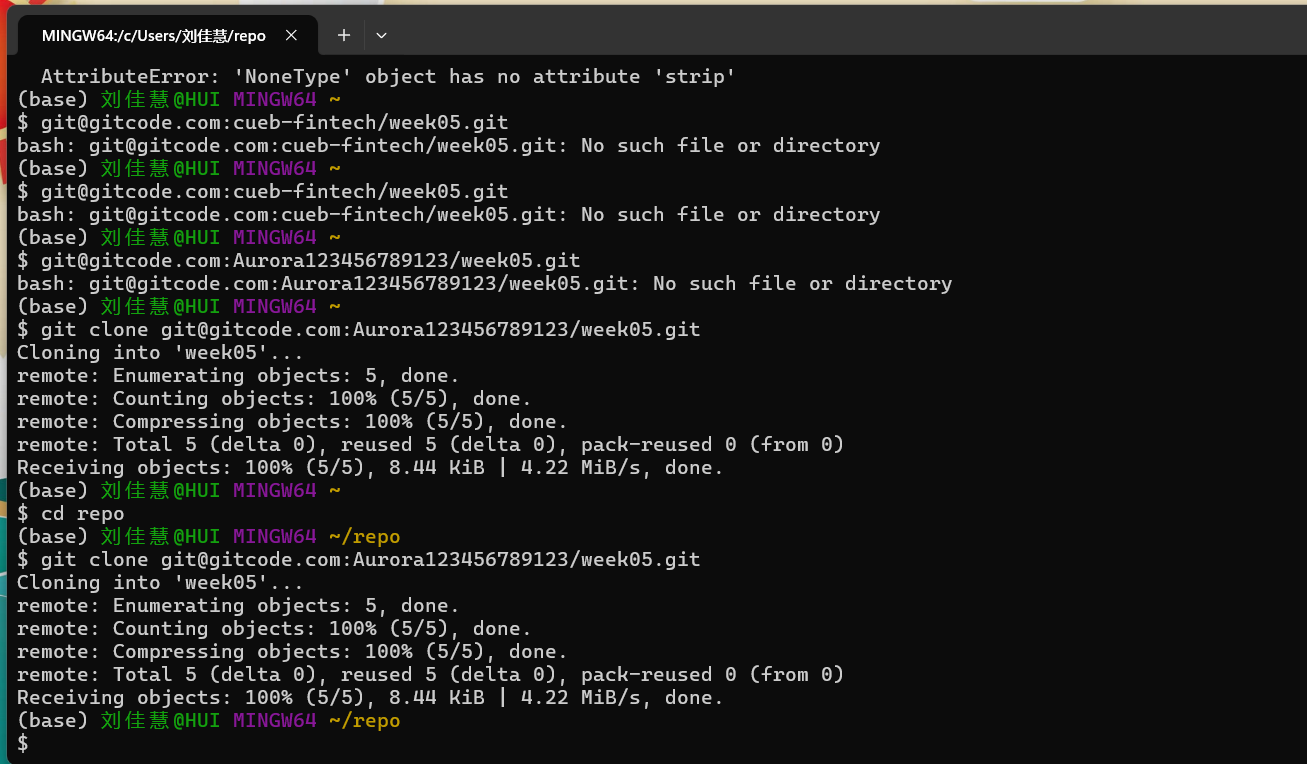
本次金融编程课程主要围绕Python编程的核心概念与实践应用展开，内容覆盖了对象类型、代码组织、抽象概念等关键知识点。课程设计充分考虑了不同水平学生的学习需求，鼓励学生持续学习并提交学习报告以巩固知识。

1. 打开提交作业的仓库 创建Fork



1. 创建仓库后选择CLONE 克隆到本地 选最后一项 创建了week04文件



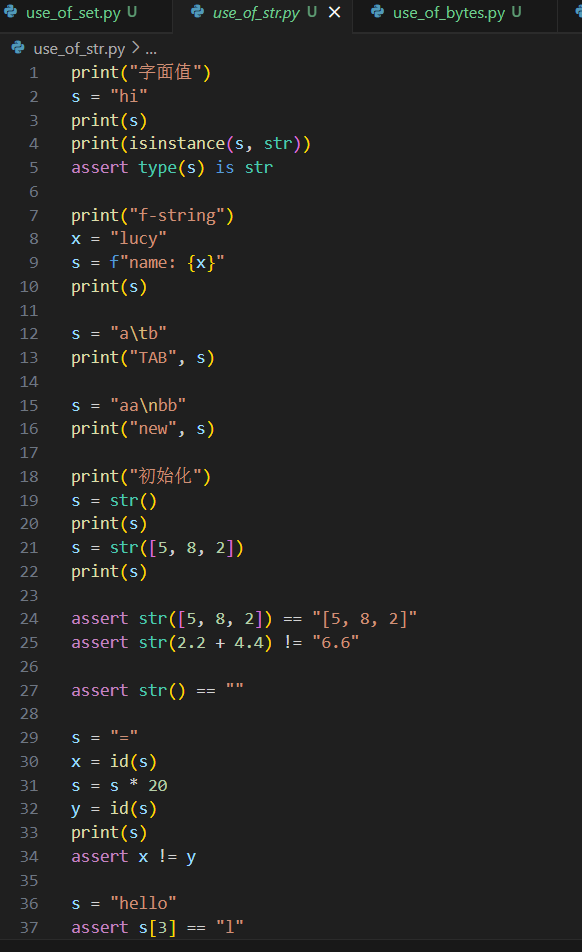


二、Python编程基础

1.对象类型与操作

讨论了Python中不同对象类型的特点，包括它们是如何被创建、使用和销毁的。

强调了理解对象概念的重要性，并通过实例演示了对象的操作。



2.代码组织与抽象

介绍了如何组织代码以提高可读性和可维护性，包括函数、模块和包的使用。

讲解了抽象概念，包括它如何帮助我们简化复杂问题和提高代码的可重用性。

3.环境搭建与数据操作

详细介绍了如何在计算机上创建和安装Python环境（康达环境）。

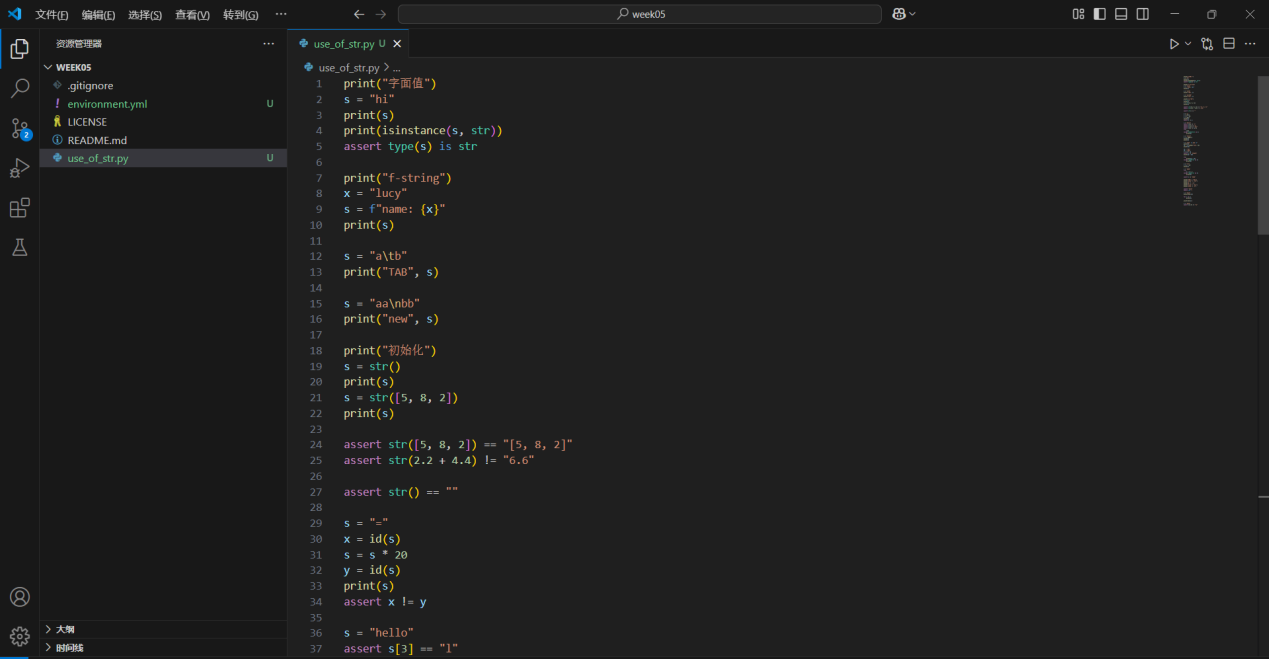
讲解了如何从文件中读取数据，包括解析和数据转换的方法。

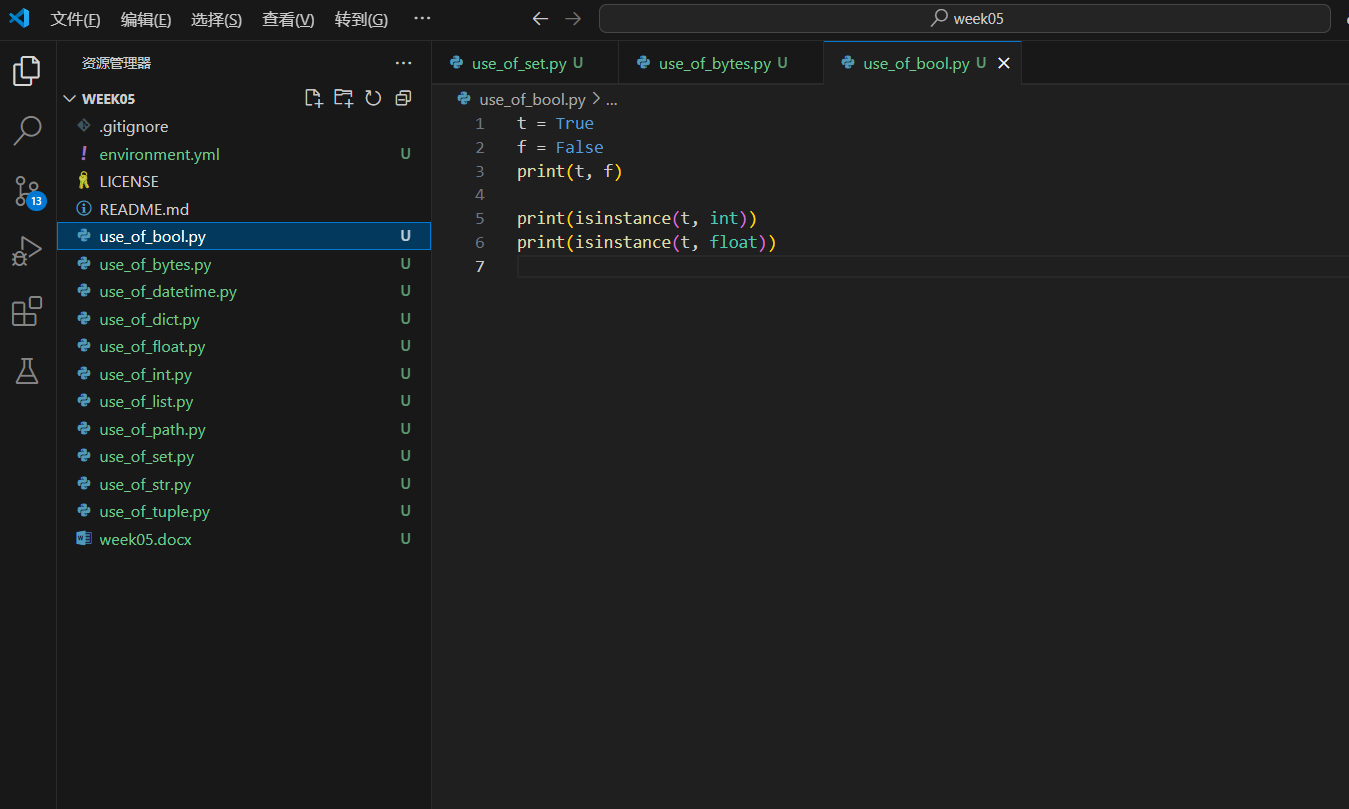
演示了如何将处理后的数据写入磁盘，包括不同格式的数据文件。

4.调试器PDB的使用

介绍了调试器PDB的基本使用，包括设置断点、单步执行和检查对象状态。

强调了调试技能在编程中的重要性，以及如何使用调试器来检查和解决问题。



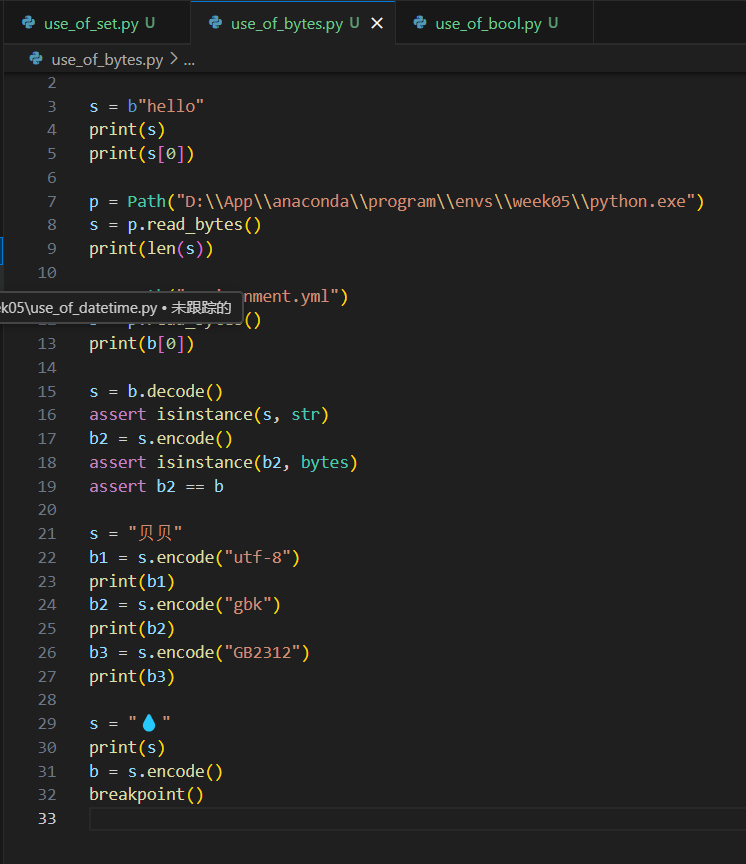


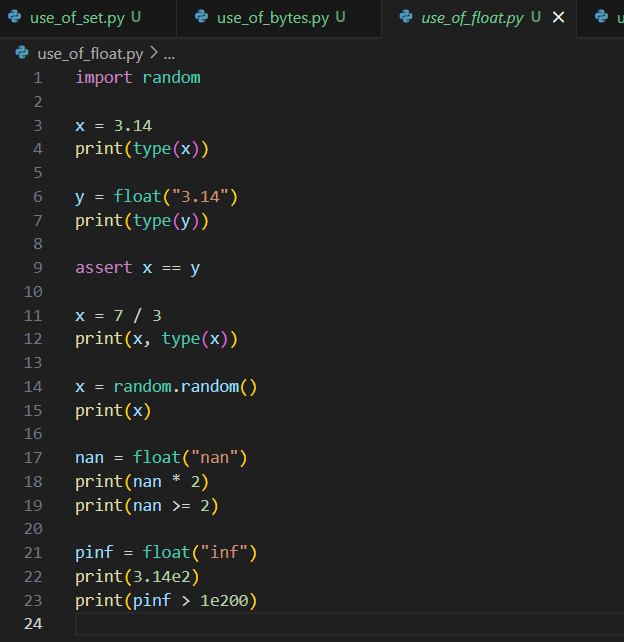
三、数据结构

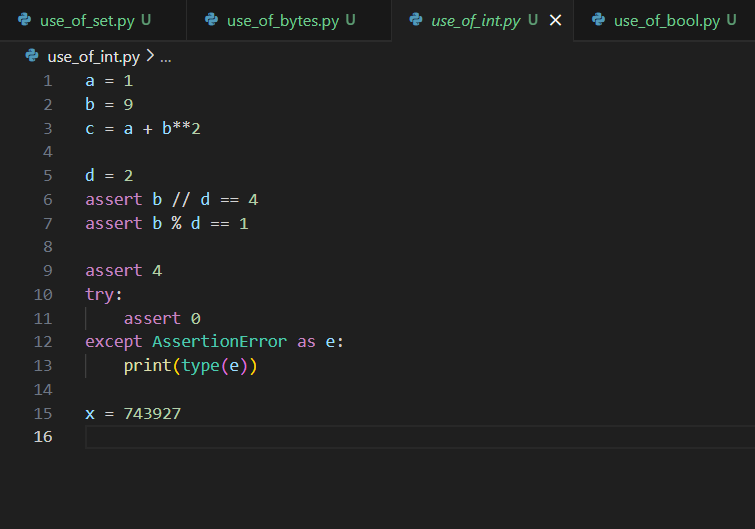
1.整数概念与用法

讨论了整数的概念，包括Python中整数的表示和操作。

强调了在处理较大整数时需要注意的存储优化和代码调整。





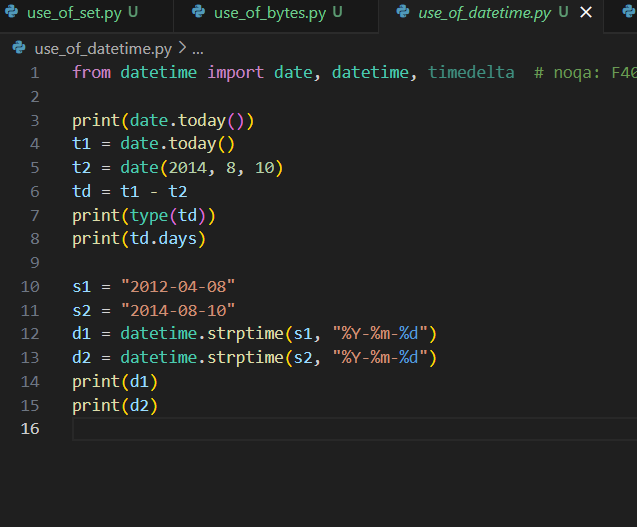


2.元组、列表、字典与集合

讨论了元组和列表的区别，以及它们在不同场景下的适用性。

介绍了不可修改的元组和字典的特性，以及它们在数据结构中的优势。

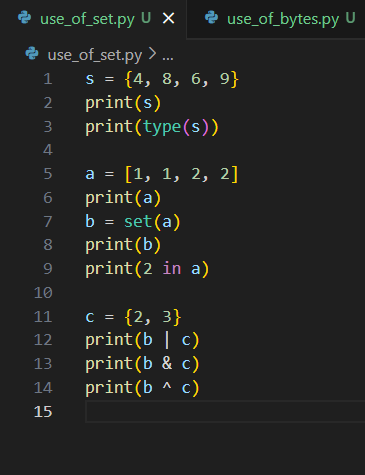
讲解了集合的概念和运算，包括并集、交集和对称差等。

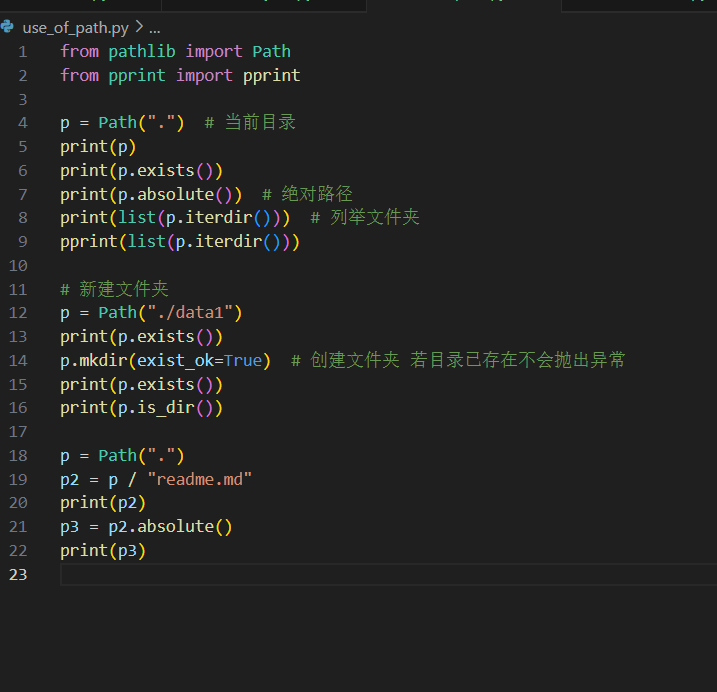


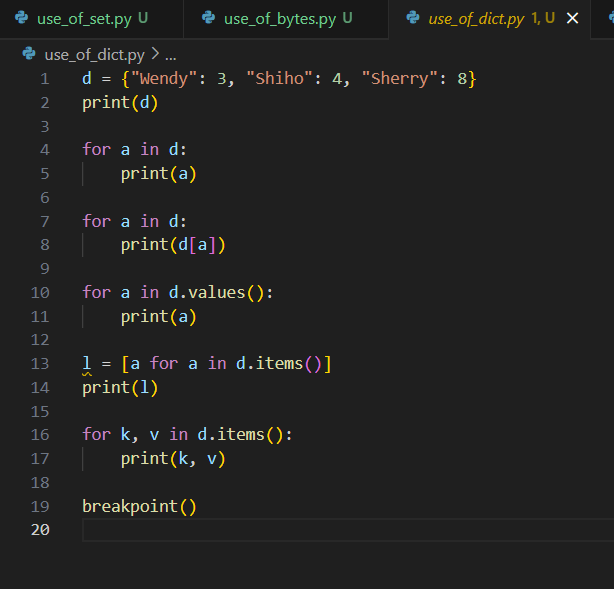
四、文件处理

介绍了文件后缀、路径和属性的处理方法，以及如何使用Python进行文件操作。

讲解了EDR（可能是特定的文件处理工具或库）的使用。







五、日期时间处理

介绍了日期时间的使用方法，包括`datetime`模块的引入和`date`、`time`类的使用。

讲解了日期时间的格式化编码和解析，以及如何在Python中处理日期时间数据。

六、金融分析应用

强调了金融学院学生应该以正确的态度使用Python进行金融分析，而不是试图重新发明轮子。

讨论了Python在金融分析中的应用，包括数据处理、模型构建和结果展示。