



泰山科技学院

Taishan College of Science and Technology

《Python 程序设计》教 案

课程学时： 32（理论）+24（实验）

课程性质： 必修

授课对象： 专升本

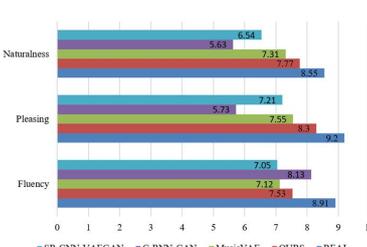
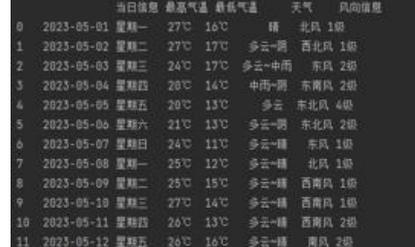
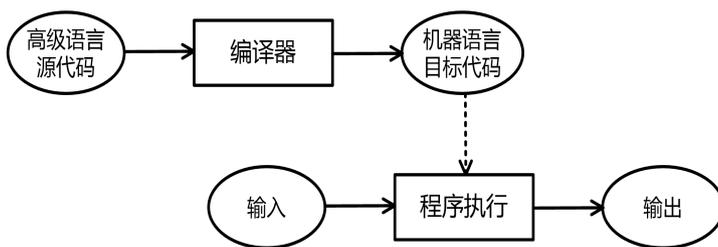
授课教师： 孙凤霄

开课单位： 大数据学院

《Python 程序设计》教 案

L1

授课时间	2022. 09. 19	课次	01
授课方式 (请打√)	理论课 <input checked="" type="checkbox"/> 讨论课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其他 <input type="checkbox"/>	课时 安排	2
授课题目 (教学章、节或主题): 第 1 章 程序设计基本方法 1.1 计算机的概念 1.2 程序设计语言 1.3 Python 语言概述 1.4 Python 语言开发环境配置 1.5 程序的基本编写方法 1.6 版本更迭			
教学目的与要求: (1)理解硬件和软件在计算机系统中的作用。 (2)了解程序设计语言的发展过程。 (3)理解 Python 语言的特点以及其重要性。 (4)掌握 Python 语言 Hello 程序的编写方法。 (5)掌握 Python 语言开发和运行环境的配置方法。 (6)理解编写程序的 IPO 方法。 (7)了解 Python 版本更迭过程和新旧版本的主要区别。			
教学重点及难点: 教学重点: 理解计算机的概念、程序设计语言的作用、Python 语言的特点、配置 Python 开发环境、掌握程序的基本编写方法、了解 Python 版本的更迭。 教学难点: 部分学生可能没有编程经验, 需要理解编程思维和 Python 的基本语法。			
作业、讨论题、思考题: 课后作业: 习题 1.2 、 1.7 讨论题: 为什么选择 Python 作为第一门编程语言? 思考题: Python 的应用领域有哪些?			
课后小结: 本章具体讲解了计算机的基本定义、计算机的功能性和可编程性、程序设计语言分类、编译和解释、Python 语言的历史和发展、配置 Python 开发环境等内容, 最后给出了 Python 版本的主要区别。			
下节课预习重点: 第 2 章 Python 程序实例解析-华氏度摄氏度温度转换			
参考文献: [1] 嵩天、礼欣、黄天羽. 《Python 语言程序设计基础(第 2 版)》. 高等教育出版社. 2017			

课时分配	教学内容	方法及手段
5M 5M	<p>一、问卷调查：对 python 了解程度？ 通过在线问卷的方法，简单了解学生们对这门语言的了解状况。</p> <p>二、课堂引入： ▲案例引入</p>	<p>问卷调查 案例引入</p>
10M	   <p>Python 实例展示：引入实际的 python 应用案例，通过图片的方式，启发学生的学习兴趣，让同学了解为什么要学习 python，学习好 python 后可以做什么？比如：科学计算、大数据分析、网络爬虫、深度学习、人工智能、游戏开发等等。</p> <p>三、编程语言简介：</p> <p>【1】TIOBE 排行榜 https://www.tiobe.com/tiobe-index/</p> <p>【2】多种编程语言“Hello World”程序对比 https://www.runoob.com/w3cnote/hello-world-programs-of-24-programming-language.html</p> <p>【3】python 语言 VS C 语言 编译和解释（举例：中国人/韩国人根据英文使用说明书操作机器） 编译：是将源代码（高级语言代码）转换成目标代码（机器语言代码）的过程。 执行编译的计算机程序称为编译器。</p>	<p>讲解</p>
10M		<p>讲解及演示</p>

解释：是将**源代码**逐条转换成**目标代码**，同时逐条运行目标代码的过程。执行解释的计算机程序称为**解释器**。

编译是一次性地翻译，一旦程序被编译，不再需要编译程序或者源代码。

- 对于相同源代码，编译所产生的目标代码执行速度更快。
- 目标代码不需要编译器就可以运行，在同类型操作系统上使用灵活。

解释则在每次程序运行时都需要解释器和源代码。

- 解释执行需要保留源代码，程序纠错和维护十分方便。
- 只要存在解释器，源代码可以在任何操作系统上运行，可移植性好

四、授课内容：

第一部分：Python快速入门（2周）

围绕两个具体实例，讲解Python基本语法元素，感性认识

第二部分：Python基础语法（10周）

从五个方面讲解基础语法全体系，提供10个示例，理性学习

第三部分：Python编程思维（4周）

从方法学角度开阔认识，提升整体编程能力，展望未来

第1章 程序设计基本方法

计算机的概念

- 计算机是根据指令操作数据的设备
- 计算机硬件所依赖的集成电路规模按照摩尔定律以指数方式增长
- 计算机运行速度也接近几何级数快速增加
- 计算机所能高效支撑的功能不断丰富发展
- 程序设计语言经历了从机器语言到高级语言的发展过程，朝着更接近自然语言的方向发展。

程序设计语言

计算思维是区别于以数学为代表的逻辑思维和以物理为代表的实证思维的第三种思维模式。

编程是一个从抽象问题到解决问题的过程。

■ 板书内容：

- 首先需要分析问题，抽象内容之间的交互关系
 - 设计利用计算机求解问题的确定性方法
 - 进而通过编写和调试代码解决问题
- 程序的基本编写方法-IPO 程序编写方法**
- 输入数据
 - 处理数据
 - 输出数据

Python 语言概述

Python 语言的发展 官网：<https://www.python.org/>

Python 语言创立 1990 年

10M	<p>2002 年, Python 2. x 2008 年, Python 3. x</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>Python 程序的特点</p> <ul style="list-style-type: none"> ■ Python 具有通用性。 ■ Python 语法简洁。 ■ Python 生态高产。 </div> <p>Python 解释器提供了几百个内置类和函数库, 世界各地程序员通过开源社区贡献了十几万个第三方函数库, 几乎覆盖了计算机技术的各个领域。</p> <p>例 5:</p> <pre>from datetime import datetime # 引用 datetime 库 now = datetime.now() # 获得当前日期和时间信息 print(now) a=now.strftime("%x") # 输出其中的日期部分 print(f"日期: {a}") b=now.strftime("%X") # 输出其中的时间部分 print("时间: {}".format(b))</pre> <ul style="list-style-type: none"> ■ 强制可读 <p>Python 通过强制缩进 (类似文章段落的首行空格) 来体现语句间的逻辑关系, 显著提高了程序的可读性, 进而增强了 Python 程序的可维护性。</p> <p>例:</p> <pre>def mean(numbers): sum = 0.0 for n in numbers: sum = sum + n return sum / len(numbers) nums = [0,1,2,3,4,5,6,7,8,9] print(mean(nums))</pre> <ul style="list-style-type: none"> ■ 支持中文 <p>Python 3. x 版本采用 Unicode 编码表达所有字符信息。Unicode 是一种国际通用表达字符的编码体系, 这使得 Python 程序可以直接支持英文、中文、法文、德文等各类自然语言字符, 在处理中文时更加灵活且高效。</p>	讲解及演示
10M	<p>Python 开发环境配置</p> <p>下载 Python 解释器并安装</p> <ul style="list-style-type: none"> ■ 到 Python 主页下载并安装 Python 基本开发和运行环境, 网址: www.python.org/downloads/ ■ 根据操作系统不同选择不同版本 ■ 下载相应的 Python 3.0 系列版本程序 <p>运行程序多种方法:</p>	讲解及演示

10M		操作演示
5M	<p>你猜：下列程序运行，键盘输入“郭靖”。输出结果？</p> <pre>name = input("输入姓名:") print("%s 同学，学好 Python，前途无量！"%name) print("%s 大侠，学好 Python，大展拳脚！"%name[0]) print("%s 哥哥，学好 Python，人见人爱！"%name[1:])</pre>	随堂练习
5M	<p>五、总结：本节学习内容并预习下节内容。</p> <p>学习资料：</p> <p>【1】教材自带 python123 网站： https://python123.io/python/songtian/5ed87dc83b5ca9663fd4761d 中国大学慕课： https://www.icourse163.org/course/BIT-268001?from=searchPage</p> <p>【2】网站菜鸟教程： https://www.runoob.com/python3/python3-tutorial.html</p> <p>【3】微信公众号“Python小屋”-董付国</p> <p>【4】比赛网站 洛谷：https://www.luogu.com.cn/</p> <p>课后练习巩固 预习第二章程序实例。</p>	总结并分享学习资源
5M	<p>★课程思政建设：</p> <p>在程序设计教学中，课程思政建设旨在通过学习编程语言深化学生的逻辑思维、创新能力和问题解决能力。通过编程实践，不仅能培养学生的耐心、解决问题的毅力，更能促进团队协作技能的发展。教育过程中，鼓励学生编写具有实际价值的程序，不仅有助于技术水平的提升，还能激发社会责任感。因此，课程内容不仅关注技术层面，更强调培养学生全面发展的能力，使其具备综合素养和社会责任感。</p>	思政建设

L2

授课时间	2022. 9. 20	课次	02
授课方式 (请打√)	理论课 <input checked="" type="checkbox"/> 讨论课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其他 <input type="checkbox"/>	课时 安排	2
授课题目 (教学章、节或主题): 第 2 章 Python 程序实例解析 2.1 实例程序 1: 温度转换 2.2 Python 程序语法元素分析 2.3 实例程序 2: Python 蟒蛇绘制 2.4 turtle 库语法元素分析			
教学目的与要求: (1)掌握解决计算问题的一般方法。 (2)掌握 Python 语言的基本语法, 包括缩进、变量、命名等。 (3)掌握 Python 语言绘制图形的一般方法。 (4)了解 Python 标准库的导入和使用。			
教学重点及难点: 重点: Python 程序的结构和语法元素, 使用 turtle 库进行绘图。 难点: 学生可能需要时间来理解 Python 的语法元素, 如缩进、变量和循环。使用 turtle 库进行绘图需要学生掌握库的语法和绘图概念。			
作业、讨论题、思考题: 课后作业: 习题 2.2 、2.4 讨论题: Python 程序中的缩进和代码块的概念, 以及其在代码结构中的作用。 思考题: 为什么程序的结构和语法元素对于编程非常重要?			
课后小结: 本章从实际问题入手, 以简单的温度转换程序为例, 逐行逐句地分析了 Python 语言的基本元素。通过讲解 Python 蟒蛇绘制实例, 介绍 Python 语言函数库 turtle 及其基本用法。			
下节课预习重点: 第 3 章 基本数据类型			
参考文献: [1] 嵩天、礼欣、黄天羽.《Python 语言程序设计基础(第 2 版)》. 高等教育出版社. 2017			

课时分配	教学内容	方法及手段
5M	<p>一、知识回顾： 回顾第一章基本概念。</p>	提问巩固
5M	<p>二、课堂引入： ▲案例引入 案例：绘制五角星 描述：假设你是一名美术老师，希望向学生展示如何使用计算机绘制五角星。你想用 Python 编写程序，让计算机自动绘制出完美的五角星。如何实现这个任务？ 引导学生思考如何手工绘制五角星。引出计算机编程的概念和作用，以及本章将介绍的内容。 讲解方法：展示手工绘制五角星的步骤，然后引入 Python 编程的方法，解释如何使用 Python 编写程序来自动绘制五角星。</p>	案例演示
5M	<p>三、授课内容： 2.1 实例程序 1：温度转换 ■板书内容：</p> <div style="border: 1px solid black; padding: 5px;"> <p>问题：如何用 Python 程序进行摄氏度和华氏度间转换？</p> <p>步骤 1：分析问题的计算部分 采用公式转换方式解决计算问题</p> <p>步骤 2：确定功能 输入：华氏或者摄氏温度值、温度标识 处理：温度转化算法 输出：华氏或者摄氏温度值、温度标识</p> <p>步骤 3：设计算法 根据华氏和摄氏温度定义，转换公式如下： $C = (F - 32) / 1.8$$F = C * 1.8 + 32$其中，C 表示摄氏温度，F 表示华氏温度</p> <p>步骤 4：编写程序</p> </div>	讲解及板书

课时分配	教学内容	方法及手段
10M	<pre data-bbox="261 324 766 627">#TempConvert.py TempStr = input("请输入带有符号的温度值:") if TempStr[-1] in ['F', 'f']: C = (eval(TempStr[0:-1]) - 32)/1.8 print("转换后的温度是{:.2f}C".format(C)) elif TempStr[-1] in ['C', 'c']: F = 1.8*eval(TempStr[0:-1]) + 32 print("转换后的温度是{:.2f}F".format(F)) else: print("输入格式错误")</pre> <div data-bbox="261 654 1182 996" style="border: 1px solid black; padding: 5px;"> <p>步骤 5: 调试、运行程序</p> <p>在系统命令行上运行如下命令执行程序:</p> <p>C:\> python TempConvert.py</p> <p>或者:</p> <p>使用 IDLE 打开上述文件, 按 F5 运行 (推荐)</p> <p>输入数值, 观察输出</p> </div> <p>★课程思政建设</p> <p>在本章的学习中, 不仅仅教授 Python 编程的基础知识, 还注重培养学生的问题解决能力。编程是一个充满挑战的过程, 学生需要通过逻辑思考和实践来解决各种问题。这个过程中, 他们将学会坚持、发散思维, 和持续寻找解决方案的毅力。这不仅是编程技能, 更是一种生活态度, 帮助他们面对各种挑战和问题都能够自信应对。</p>	思政建设
15M	<p>2.2 Python 程序语法元素分析</p> <p>■板书内容:</p> <div data-bbox="261 1473 1321 1989" style="border: 1px solid black; padding: 5px;"> <p>程序的格式框架</p> <p>Python 语言采用严格的“缩进”来表明程序的格式框架。缩进用来表示代码之间的包含和层次关系。</p> <p>推荐: 1 个缩进 建议 4 个空格</p> <p>命名与保留字</p> <p>常量: 程序中值不发生改变的元素。</p> <p>变量: 程序中值发生改变或者可以发生改变的元素。</p> <p>Python 语言允许采用大写字母、小写字母、数字、下划线(_)和汉字等字符及其组合给变量命名。</p> </div>	板书讲解及演示

课时分配	教学内容	方法及手段
10M	<p>但名字的首字符不能是数字，中间不能出现空格，长度没有限制。 标识符大小写敏感，stu 和 STU 是两个不同的名字。</p> <p>Python 3.x 保留字列表 (33 个)</p> <p>赋值语句</p> <p>Python 语言中，= 表示“赋值”，即将等号右侧的值计算后将结果值赋给左侧变量。</p> <p>例：a=3 sum=a+b ch= 'hello'</p> <p>同步赋值语句：同时给多个变量赋值。</p> <p><变量 1>, ..., <变量 N> = <表达式 1>, ..., <表达式 N></p> <p>a,b=3,4 a,b,sum=3,4,a+b</p> <p>input() 函数</p> <p>获得用户输入之前，input() 函数可以包含一些提示性文字。</p> <p><变量> = input(<提示性文字>)</p>	讲解及演示
20M	<p>字符串</p> <p>Python 语言中，字符串是用两个双引号“ ”或者单引号‘ ’括起来的一个或多个字符。</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p style="text-align: center;">反向递减序号</p> <p style="text-align: center;">-12 -11 -10 -9 -8 -7 -6 -5 -4 -3 -2 -1</p> <p style="text-align: center;">请输入带有符号的温度值：</p> <p style="text-align: center;">0 1 2 3 4 5 6 7 8 9 10 11</p> <p style="text-align: center;">正向递增序号</p> </div> <p>eval() 函数</p> <p>eval(<字符串>) 函数，能够以 Python 表达式的方式解析并执行字符串，将返回结果输出。</p> <p>输出函数</p> <p>print() 函数用来输出各种类型常量的值。 print() 函数可以输出各种类型变量的值。 print() 函数通过%来选择要输出的变量。 print() 函数通过 format() 函数选择输出的变量。</p> <p>分支语句</p> <p>分支语句是控制程序运行的一类重要语句，它的作用是根据判断条件选择程序执行路径。</p>	讲解及演示

课时分配	教学内容	方法及手段
10M	<p>循环语句 循环语句：控制程序运行，根据判断条件或计数条件确定一段程序的运行次数。</p> <p>def 定义函数 def 用于定义函数。 函数是一组代码的集合，用于表达一个功能，或者说，函数表示一组代码的归属，函数名称是这段代码的名字。 def 所定义的函数在程序中未经调用不能直接执行，需要通过函数名调用才能够执行。</p> <p>2.3 实例程序 2: Python 蟒蛇绘制</p> <pre data-bbox="284 891 762 1339"> 1 # e2.1DrawPython.py 2 import turtle 3 turtle.setup(650, 350, 200, 200) 4 turtle.penup() 5 turtle.fd(-250) 6 turtle.pendown() 7 turtle.pensize(1) 8 turtle.pencolor("purple") 9 turtle.seth(-40) 10 for i in range(4): 11 turtle.circle(40, 80) 12 turtle.circle(-40, 80) 13 turtle.circle(40, 80/2) 14 turtle.fd(40) 15 turtle.circle(16, 180) 16 turtle.fd(40 * 2/3) 17 turtle.done() </pre> <p>2.4 turtle 库语法元素分析</p> <pre data-bbox="256 1420 1058 2033"> # 启动窗口 turtle.setup(width, height, startx, starty) # 设置初始位置 turtle.penup() turtle.fd(50) # turtle.goto(x,y) #设置画笔 turtle.pensize(5) turtle.pencolor('red') # pencolor('#ff0000') # pencolor((255,0,0)) #准备画，笔落下 turtle.pendown() </pre>	讲解及演示

课时分配	教学内容	方法及手段
5M	<pre>#设置角度、前进距离 turtle.seth(0) # 绝对角度 #相对角度 left(90) right(90) turtle.fd(200) #暂停，保持画框不关闭 turtle.done()</pre> <p>四、总结： 学习完本课内容，程序的基本语法元素：程序的格式框架、缩进、注释、变量、命名、保留字、数据类型、赋值语句、函数。 基本输入输出函数：input()、eval()、print()。 源程序的书写风格。</p>	总结

L3

授课时间	2022. 9. 26	课次	03
授课方式 (请打√)	理论课 <input checked="" type="checkbox"/> 讨论课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其他 <input type="checkbox"/>		课时 安排
2			
授课题目 (教学章、节或主题): 第 3 章 基本数据类型 3.1 数字类型 3.2 数字类型的操作 3.3 math 库的使用			
教学目的与要求: (1)掌握 3 种数字类型的概念和使用。 (2)了解 3 种数字类型在计算机中的表示方法。 (3)运用 Python 的标准数学库进行数值计算。			
教学重点及难点: 重点: Python 中的数字类型和基本操作。如何使用 math 库进行数学计算。 难点: 理解数字类型的不同子类型, 如整数和浮点数。学生可能需要时间来适应数学库的使用。			
作业、讨论题、思考题: 课后作业: 习题 3.1 、 3.4 讨论题: 探讨使用 Python 编程解决数学问题的优势, 如计算圆的面积。 思考题: math 库如何扩展了 Python 的数学计算功能?			
课后小结: 本章介绍了计算机中常用的数字类型及操作, 包括 Python 内置的数值运算操作和数字类型转换函数等, 进一步介绍了常用的数学计算标准库 math 库。			
下节课预习重点: 第 3 章 基本数据类型			
参考文献: [1] 嵩天、礼欣、黄天羽.《Python 语言程序设计基础(第 2 版)》. 高等教育出版社. 2017			

课时分配	教学内容	方法及手段
10M	<p>一、知识回顾：</p> <p>回顾以简单的温度转换程序为例，逐行逐句地分析了 Python 语言的基本元素。通过讲解 Python 蟒蛇绘制实例，介绍 Python 语言函数库 turtle 及其基本用法。</p> <p>二、课堂引入：</p> <p>▲案例引入</p> <p>案例：计算圆的面积</p> <p>描述：假设你是一个园艺师，需要计算不同大小的花圃的面积，其中包括圆形花圃。你知道圆的面积计算公式，但如何用 Python 编写程序来自动计算呢？引导学生思考如何手工计算圆的面积。引出 Python 编程的概念和本章学习的内容。</p>	提问巩固
20M	<p>讲解方法：展示手工计算圆的面积的步骤，然后引入 Python 编程的方法，解释如何使用 Python 编写程序来自动计算圆的面积。</p> <p>三、授课内容：</p> <p>3.1 数字类型</p> <p>Python 语言包括三种数字类型</p> <div style="border: 1px solid black; padding: 5px;"> <p>■板书内容</p> <p>1) 整数类型</p> <p>a) pow(x, y) 函数：计算 x^y</p> <p>b) 示例</p> <p>i. 1010, 99, -217</p> <p>ii. 0x9a, -0X89 (0x, 0X 开头表示 16 进制数)</p> <p>iii. 0b010, -0B101 (0b, 0B 开头表示 2 进制数)</p> <p>iv. 0o123, -00456 (0o, 00 开头表示 8 进制数)</p> <p>2) 浮点数类型</p> <p>a) 带有小数点及小数的数字。</p> <p>b) Python 语言中浮点数的数值范围存在限制，小数精度也存在限制。这种限制与在不同计算机系统有关。</p> </div>	讲解及板书

课时分配	教学内容	方法及手段																				
10M	<p>c) 示例</p> <ul style="list-style-type: none"> i. 0.0, -77., -2.17 ii. 96e4, 4.3e-3, 9.6E5 (科学计数法) iii. 科学计数法使用字母“e”或者“E”作为幂的符号, 以10为基数。科学计数法含义如下: $\langle a \rangle e \langle b \rangle = a * 10^b$ <p>3) 复数类型</p> <ul style="list-style-type: none"> a) 与数学中的复数概念一致, $z = a + bj$, a是实数部分, b是虚数部分, a和b都是浮点类型, 虚数部分用j或者J标识。 b) 示例: $12.3+4j, \quad -5.6+7j$ <p>对于复数z, 可以用z.real获得实数部分, z.imag获得虚数部分。</p>	讲解																				
15M	<p>3.2 数字类型的操作</p> <p>三种类型存在一种逐渐“扩展”的关系: 整数 -> 浮点数 -> 复数 (整数是浮点数特例, 浮点数是复数特例)</p> <p>不同数字类型之间可以进行混合运算, 运算后生成结果为最宽类型。</p>	讲解及演示																				
10M	<table border="1" data-bbox="215 1357 1114 1899"> <thead> <tr> <th>操作符</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>$x + y$</td> <td>x与y之和</td> </tr> <tr> <td>$x - y$</td> <td>x与y之差</td> </tr> <tr> <td>$x * y$</td> <td>x与y之积</td> </tr> <tr> <td>x / y</td> <td>x与y之商</td> </tr> <tr> <td>$x // y$</td> <td>x与y之整数商, 即: 不大于x与y之商的最大整数</td> </tr> <tr> <td>$x \% y$</td> <td>x与y之商的余数, 也称为模运算</td> </tr> <tr> <td>$-x$</td> <td>x的负值, 即: $x*(-1)$</td> </tr> <tr> <td>$+x$</td> <td>x本身</td> </tr> <tr> <td>$x**y$</td> <td>x的y次幂, 即: x^y</td> </tr> </tbody> </table> <p>3.3 math库的使用</p> <p>首先使用保留字import引用该库</p>	操作符	描述	$x + y$	x与y之和	$x - y$	x与y之差	$x * y$	x与y之积	x / y	x与y之商	$x // y$	x与y之整数商, 即: 不大于x与y之商的最大整数	$x \% y$	x与y之商的余数, 也称为模运算	$-x$	x的负值, 即: $x*(-1)$	$+x$	x本身	$x**y$	x的y次幂, 即: x^y	讲解及演示
操作符	描述																					
$x + y$	x与y之和																					
$x - y$	x与y之差																					
$x * y$	x与y之积																					
x / y	x与y之商																					
$x // y$	x与y之整数商, 即: 不大于x与y之商的最大整数																					
$x \% y$	x与y之商的余数, 也称为模运算																					
$-x$	x的负值, 即: $x*(-1)$																					
$+x$	x本身																					
$x**y$	x的y次幂, 即: x^y																					

课时分配	教学内容	方法及手段															
15M	<p>第一种: <code>import math</code></p> <p>对 <code>math</code> 库中函数采用 <code>math.()</code> 形式使用</p> <p>第二种, <code>from math import <函数名></code></p> <p>对 <code>math</code> 库中函数可以直接采用 <code><函数名>()</code> 形式使用</code></p> <p><code>math</code> 库包括 4 个数学常数</p> <table border="1" data-bbox="210 683 852 891"> <thead> <tr> <th>常数</th> <th>数学表示</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td><code>math.pi</code></td> <td>π</td> <td>圆周率, 值为3.141592653589793</td> </tr> <tr> <td><code>math.e</code></td> <td>e</td> <td>自然对数, 值为2.718281828459045</td> </tr> <tr> <td><code>math.inf</code></td> <td>∞</td> <td>正无穷大, 负无穷大为<code>-math.inf</code></td> </tr> <tr> <td><code>math.nan</code></td> <td></td> <td>非浮点数标记, NaN (Not a Number)</td> </tr> </tbody> </table> <p><code>math</code> 库包括 16 个数值表示函数</p> <p><code>math</code> 库中包括 8 个幂对数函数</p> <p><code>math</code> 库包括六个“三角双曲函数”</p> <p><code>math</code> 库包括 4 个高等特殊函数</p>	常数	数学表示	描述	<code>math.pi</code>	π	圆周率, 值为3.141592653589793	<code>math.e</code>	e	自然对数, 值为2.718281828459045	<code>math.inf</code>	∞	正无穷大, 负无穷大为 <code>-math.inf</code>	<code>math.nan</code>		非浮点数标记, NaN (Not a Number)	讲解及演示
常数	数学表示	描述															
<code>math.pi</code>	π	圆周率, 值为3.141592653589793															
<code>math.e</code>	e	自然对数, 值为2.718281828459045															
<code>math.inf</code>	∞	正无穷大, 负无穷大为 <code>-math.inf</code>															
<code>math.nan</code>		非浮点数标记, NaN (Not a Number)															
10M	<p>四、总结:</p> <p>学习完本课内容, 能够掌握计算机中常用的数字类型及操作, 包括 Python 内置的数值运算操作和数字类型转换函数等, 进一步介绍了常用的数学计算标准库 <code>math</code> 库。</p>	总结															
5M	<p>★课程思政建设部分内容</p> <p>本章的教学旨在培养学生的数学思维和逻辑分析能力。数学不仅仅是一门学科, 还是一种思维方式, 可以帮助学生更好地理解 and 解决问题。通过编写程序解决数学问题, 学生将培养创新思维和解决实际问题的能力。</p>																

L4

授课时间	2022. 9. 27	课次	04
授课方式 (请打√)	理论课 <input checked="" type="checkbox"/> 讨论课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其他 <input type="checkbox"/>	课时 安排	2
授课题目 (教学章、节或主题): 第 3 章 基本数据类型 3.4 字符串类型及操作 3.5 字符串类型的格式化			
教学目的与要求: (1)掌握字符串类型的概念和使用。 (2)掌握字符串类型的格式化操作方法和应用。			
教学重点及难点: 重点: 字符串数据类型的常见操作和方法。字符串格式化的应用。 难点: 学生可能需要时间来理解不同数据类型的特点和用途。字符串格式化要求学生理解占位符和格式规则。			
作业、讨论题、思考题: 课后作业: 习题 3.6 讨论题: 探讨如何使用字符串格式化在实际应用中创建友好的用户界面。 思考题: 字符串格式化如何提高程序的可读性和用户友好性?			
课后小结: 本章介绍了字符串类型及其操作和格式化方法, 并通过字符串格式化实现控制台风格的文本进度条。			
下节课预习重点: 第 4 章 程序的控制结构			
参考文献: [1] 嵩天、礼欣、黄天羽.《Python 语言程序设计基础(第 2 版)》. 高等教育出版社. 2017			

课时分配	教学内容	方法及手段
5M	<p>一、知识回顾：</p> <p>回顾数字类型及操作，包括 Python 内置的数值运算操作和数字类型转换函数等。</p>	提问巩固
5M	<p>二、课堂引入：</p> <p>▲案例引入：</p> <p>案例：文字游戏</p> <p>描述：假设你正在设计一个文字游戏，需要从玩家获取姓名和分数，然后输出一句鼓励的话，如“恭喜 [姓名]，你的得分是 [分数] 分！”你想用 Python 编写程序来实现这个功能。</p> <ul style="list-style-type: none"> • 引导学生思考如何手工完成这个任务。 • 引出基本数据类型和字符串操作的概念。 <p>讲解方法：展示手工获取信息和输出鼓励话语的步骤，然后引入 Python 编程的方法，解释如何使用基本数据类型和字符串操作来实现这个任务。</p>	案例引入
10M	<p>三、授课内容：</p> <p>3.4 字符串类型及操作</p> <p>■板书内容：</p> <div style="border: 1px solid black; padding: 5px;"> <p>1) 字符串是用双引号"或者单引号'括起来的一个或多个字符。</p> <p>使用单引号时，多引号可作为字符串一部分。</p> <p>使用双引号时，单引号可作为字符串一部分。</p> <p>使用三引号时，另两种可作为一部分。</p> <p>三引号可以表示单行或者多行字符串。</p> <p>2) 字符串可以保存在变量中，也可以单独存在。</p> <p>3) 可以用 type() 函数测试一个字符串的类型。</p> <p>4) Python 语言转义符： \</p> <p>5) 输出带有引号的字符串，可以使用转义符。</p> <p>6) 使用 \\ 输出带有转移符的字符串。</p> <p>7) 字符串是一个字符序列：字符串最左端位置标记为 0，依次增加。字符串中的编号叫做“索引”。</p> </div>	板书，讲解

课时分配	教学内容	方法及手段
15M	<p>a) 正向索引: 字符串索引从左侧 0 开始, 一个长度为 L 的字符串最后一个字符的位置是 L-1</p> <p>b) 反向索引: 使用负数, 最右侧索引值是-1, 从字符串右边末尾向左边进行。</p> <p>c) 单个索引: 辅助访问字符串中的特定位置。 格式为<string>[<索引>]</p> <pre>>>> greet="Hello John" >>> print(greet[2]) 1 >>> x=8 >>> print(greet[x-2]) J >>></pre> <p>d) 区间访问: 可以通过两个索引值确定一个位置范围, 返回这个范围的子串。 格式: <string> [<start> : <end>]</p> <ol style="list-style-type: none"> ① start 和 end 都是整数型数值 ② 从索引 start 开始直到索引 end 结束的子序列, 但不包括 end 位置。 ③ 正向序号、反向序号可以混用 ④ 缺省值为起始值、终止值 	讲解及演示
	<div style="display: flex; justify-content: space-around;"> <div data-bbox="295 1254 694 1444" style="border: 1px solid gray; padding: 5px;"> <pre>>>> a='大数据学院' >>> print(a[0:4:1]) 大数据学 >>> print(a[0:5:1]) 大数据学院 >>> print(a[0:5:2]) 大据院</pre> </div> <div data-bbox="742 1254 1077 1444" style="border: 1px solid gray; padding: 5px;"> <pre>>>> a='大数据学院' >>> print(a[5:0:-1]) 院学据数 >>> print(a[5::-1]) 院学据数大 >>> print(a[-1::-1]) 院学据数大 >>> print(a[-1::-2]) 院据大</pre> </div> </div> <p>思考: 判断是否是回文?</p> <div data-bbox="295 1512 678 1758" style="border: 1px solid gray; padding: 5px;"> <pre>>>> a='hello' >>> print(a[:]) # a[0:] # a[0::1] hello >>> >>> print(a[::-1]) #a[-1::-1] olleh >>> >>> print(a[:] == a[::-1]) #判断回文? False >>> >>> b='heeteeh' >>> print(b[:] == b[::-1]) #判断回文? True</pre> </div>	
10M	<p>3.5 字符串类型及其操作</p> <p>基本的操作符</p> <p>str [i] 索引</p> <p>str [N : M] 切片, 从 N 到 M 但不包括 M 的子串</p> <p>x in s x 是否是 s 的子串? 返回 TRUE / FALSE</p>	讲解及演示

课时分配	教学内容	方法及手段																
10M	<p>加法操作 (+) 将两个字符串连接成为一个新的字符串 乘法操作 (*) 生成一个由其本身重复连接而成的字符串</p> <p>字符串使用实例：输入一个月份数字，返回对应月份名称缩写 IPO 模式是： 输入：输入一个表示月份的数字 (1-12) 处理：利用字符串基本操作实现该功能 输出：输入数字对应月份名称的缩写</p> <p>内置处理函数</p>	讲解及演示																
10M	<p>len() 函数能返回一个字符串的长度 Python 3.0 以 Unicode 字符为计数单位，所以中文、英文字符都是 1 个长度单位。</p> <pre>>>> len("pine") 4 >>> len("祖国，您好！") 6</pre> <p>字符串内置处理函数-字符串类型的转换 大多数都可以通过 str() 函数转换为字符串形式 思考：判断数字是否是回文数字？ 1234? str(a)[:] == str(a)[::-1]</p>	讲解及演示																
10M	<p>内置处理方法</p> <p>面向对象过程中，类似 <a>. () 形式调用，处理某类的函数，称为“方法”。</p> <p>Python 解释器内部，将数据类型采用面向对象的方式实现，封装为一个类，字符串是一个类。 针对字符串处理函数，称为字符串的处理方法。 字符串类型包含 43 个内置方法。</p> <table border="1" data-bbox="268 1402 1051 1839"> <thead> <tr> <th>方法</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>str.lower()</td> <td>返回字符串str的副本，全部字符小写</td> </tr> <tr> <td>str.upper()</td> <td>返回字符串str的副本，全部字符大写</td> </tr> <tr> <td>str.split(sep=None, maxsplit=-1)</td> <td>返回一个列表，由str根据sep被分割的部分构成</td> </tr> <tr> <td>str.count(sub[,start[,end]])</td> <td>返回str[start: end]中sub子串出现的次数</td> </tr> <tr> <td>str.replace(old, new[, count])</td> <td>返回字符串str的副本，所有old子串被替换为new，如果count给出，则前count次old出现被替换</td> </tr> <tr> <td>str.center(width[, fillchar])</td> <td>字符串居中函数，详见函数定义</td> </tr> <tr> <td>str.format()</td> <td>返回字符串str的一种排版格式，3.6节将详细介绍</td> </tr> </tbody> </table>	方法	描述	str.lower()	返回字符串str的副本，全部字符小写	str.upper()	返回字符串str的副本，全部字符大写	str.split(sep=None, maxsplit=-1)	返回一个列表，由str根据sep被分割的部分构成	str.count(sub[,start[,end]])	返回str[start: end]中sub子串出现的次数	str.replace(old, new[, count])	返回字符串str的副本，所有old子串被替换为new，如果count给出，则前count次old出现被替换	str.center(width[, fillchar])	字符串居中函数，详见函数定义	str.format()	返回字符串str的一种排版格式，3.6节将详细介绍	讲解及演示
方法	描述																	
str.lower()	返回字符串str的副本，全部字符小写																	
str.upper()	返回字符串str的副本，全部字符大写																	
str.split(sep=None, maxsplit=-1)	返回一个列表，由str根据sep被分割的部分构成																	
str.count(sub[,start[,end]])	返回str[start: end]中sub子串出现的次数																	
str.replace(old, new[, count])	返回字符串str的副本，所有old子串被替换为new，如果count给出，则前count次old出现被替换																	
str.center(width[, fillchar])	字符串居中函数，详见函数定义																	
str.format()	返回字符串str的一种排版格式，3.6节将详细介绍																	
10M	<p>3.6 字符串类型的格式化</p> <p>引入案例：输出两个同学成绩？</p>	讲解及演示																

课时分配	教学内容	方法及手段
------	------	-------

```

1 a=80;b=90
2 print("a同学成绩%d分"%(a))
3 # print(" 字符串%d " % (变量) )
4 print("a同学成绩" + str(a)+ "分" )
5 # +运算符连接字符串
6 print("b同学成绩{1}分".format(a,b))
7 # print(" { } ".format(变量) )

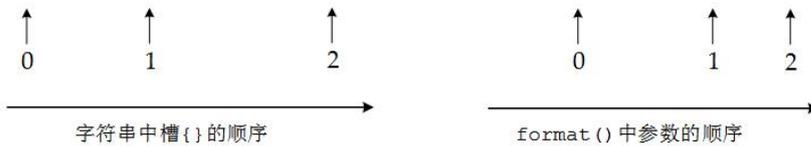
```

format() 方法的基本使用

字符串 format() 方法的基本使用格式是：

<模板字符串>.format(<逗号分隔的参数>)

"{ }：计算机{ }的CPU占用率为{ }%。".format("2016-12-31","PYTHON",10)



format() 方法中模板字符串的槽除了包括参数序号，还可以包括格式控制信息。此时，槽的内部样式如下：{<参数序号>:<格式控制标记>}

其中，格式控制标记用来控制参数显示时的格式。

	<填充>	<对齐>	<宽度>	,	<.精度>	<类型>
引导符号	用于填充的单个字符	< 左对齐 > 右对齐 ^ 居中对齐	槽的设定输出宽度	数字的千位分隔符 适用于整数和浮点数	浮点数小数部分的精度 或 字符串的最大输出长度	整数类型 b, c, d, o, x, X 浮点数类型 e, E, f, %

格式控制标记包括：<填充><对齐><宽度>,<.精度><类型>6 个字段，这些字段都是可选的，可以组合使用，这里按照使用方式逐一介绍。

```

>>>s = "等级考试"
>>>"{:25}".format(s) # 默认左对齐, 空格填充
'等级考试'
>>>">{:~25}".format(s) #居中对齐
'          等级考试          '
>>>">{:>25}".format(s) #右对齐
'          等级考试          '
>>>">{:~*25}".format(s) #居中对齐且填充*号
'*****等级考试*****'
>>>">{:+~25}".format(s) #居中对齐且填充+号
'+++++++等级考试+++++++'
>>>">{:+~25}".format(s) #居中对齐且填充汉字“十”
'+++++++等级考试+++++++'
>>>">{:~1}".format(s) #z指定宽度为1, 不足变量s的宽度
'等级考试'

```

随堂练习

15M

课时分配	教学内容	方法及手段
5M	<p>综合案例演示：</p> <p>恺撒密码</p> <p>文本进度条</p> <p>四、总结：</p> <p>学习完本课内容，能够掌握字符串类型及其操作和格式化方法，并通过字符串格式化实现控制台风格的文本进度条。</p>	总结
5M	<p>★课程思政建设：</p> <p>在本章的学习中，我们鼓励学生培养思考和表达的能力。文字游戏案例引入了互动性和创造性，鼓励学生在编程中发挥自己的创意。此外，通过教学，我们强调了编程的实用性，帮助学生理解编程技能如何应用于解决实际问题。</p>	思政建设

L5

授课时间	2022. 10. 04	课次	05
授课方式 (请打√)	理论课 <input checked="" type="checkbox"/> 讨论课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其他 <input type="checkbox"/>		课时 安排
2			
授课题目 (教学章、节或主题): 第 4 章 程序的控制结构 4.1 程序的基本结构 4.2 程序的分支结构 4.3 程序的循环结构			
教学目的与要求: (1)了解程序的基本结构并绘制流程图。 (2)掌握程序的分支结构。 (3)运用 if 语句实现分支结构。 (4)掌握程序的循环结构。 (5)运用 for 语句和 while 语句实现循环结构。			
教学重点及难点: 重点: Python 语言的指令流控制结构, 包括顺序结构、分支结构、循环结构。 难点: 学生可能需要时间来理解程序的控制结构和逻辑。在编写猜数字游戏时, 学生需要思考游戏规则和用户互动的细节。			
作业、讨论题、思考题: 课后作业: 习题 4.1 讨论题: 探讨在程序设计中如何更好地管理复杂性和提高代码的可维护性。 思考题: 为什么程序的控制结构对于编程非常重要?			
课后小结: 本章主要讲解程序的基本结构, 包括分支结构和循环结构, 介绍身体质量指数 BMI 的计算, 用实例说明分支结构的使用。			
下节课预习重点: 第 4 章 程序的控制结构 · 下			
参考文献: [1] 嵩天、礼欣、黄天羽.《Python 语言程序设计基础(第 2 版)》. 高等教育出版社. 2017			

课时分配	教学内容	方法及手段
5M	<p>一、知识回顾：</p> <p>回顾 C 语言的指令流控制结构，包括顺序结构、分支结构、循环结构。</p>	提问巩固
5M	<p>二、课堂引入：</p> <p>▲案例引入：</p> <p> 案例：猜数字游戏</p> <p> 描述：你正在设计一个猜数字游戏。游戏规则是给定确定的一个数字，玩家需要猜测这个数字。如果猜对了，游戏结束；如果猜错了，计算机会提示玩家数字是太大还是太小，玩家可以继续猜测。你希望用 Python 编写程序来实现这个游戏。</p> <ul style="list-style-type: none"> • 引导学生思考如何手工设计和运行这个游戏。 • 引出程序的控制结构和逻辑。 <p> 讲解方法：展示游戏的规则和玩法，然后引入 Python 编程的方法，解释如何使用程序的控制结构来实现猜数字游戏。</p>	案例引入
5M	<p>三、授课内容：</p> <p>4.1 程序的基本结构</p> <p>程序流程图用一系列图形、流程线和文字说明描述程序的基本操作和控制流程，它是程序分析和过程描述的最基本方式。</p> <p>顺序结构是程序的基础，但单一的顺序结构不可能解决所有问题。</p> <p>■ 板书内容：</p> <div data-bbox="252 1496 1316 2009" style="border: 1px solid black; padding: 10px;"> <p>程序由三种基本结构组成：</p> <ul style="list-style-type: none"> ■ 顺序结构 ■ 分支结构 ■ 循环结构 <p style="text-align: center;">二分枝结构 条件循环</p> </div>	讲解及板书

课时分配	教学内容	方法及手段
10M	<p>这些基本结构都有一个入口和一个出口。任何程序都由这三种基本结构组合而成。</p> <p>4.2 程序的分支结构</p> <p>问题：冬奥盛会引起全运健身潮，户外运动前查看软件提醒，PM2.5 值是多少？空气质量是否适合户外活动？</p> <p>思考：实现 PM 2.5 空气质量提醒，主要用分支结构还是循环结构实现？</p> <p>本节课内容：4.2 程序的分支结构</p> <p>4.2.1 单分支结构： if 语句</p> <p>4.2.2 二分支结构： if-else 语句</p> <p>4.2.3 多分支结构： if-elif-else 语句</p>	讲解及板书
10M	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <p>4.2.1 单分支结构: if 语句</p> <p>流程图展示及分析</p> <p>if 语句语法格式：</p> <pre style="margin-left: 20px;">if <条件>: 语句块 <if 后续语句></pre> </div> <p>重点解释说明：</p> <p>评估<条件>的结果值</p> <ul style="list-style-type: none"> ➤ 如为 True，则执行语句块，然后转向 if 后续语句。 ➤ 如为 False，语句块里的语句会被跳过。 <p>语句块</p> <ul style="list-style-type: none"> ➤ 是 if 条件满足后执行的一个或多个语句序列。 ➤ 语句块中语句通过与 if 所在行形成缩进，表达包含关系。 <p>判断条件</p> <ul style="list-style-type: none"> ➤ 形成判断条件最常见的方式是采用关系操作符。 ➤ Python 语言共有 6 个关系操作符。 <p>< <= >= > == !=</p> <p>用单分支语句实现两种结果的执行</p> <p>微实例 4.4：PM 2.5 空气质量提醒（1）</p> <p>优： 0<=PM<35 良： 35<=PM</p> <p>分析 输入：PM2.5 数值</p> <p> 处理：if 语句</p> <p> 输出：空气质量提醒</p> <p>代码</p> <pre>PM = eval (input("请输入 PM2.5 数值: ")) if 0<= PM < 35:</pre>	讲解及演示

课时分配	教学内容	方法及手段
10M	<pre>print("空气优, 快去户外运动!") if 35 <= PM : print("空气良, 适度户外活动! ")</pre> <p>除了单分支语句实现, 还可以用双分支语句, 实现的方法不是固定的。</p> <div style="border: 1px solid black; padding: 5px;"> <p>4.2.2 二分支结构: if-else 语句 流程图展示及分析 if 语句语法格式:</p> <pre> if <条件> : 语句块 1 else: 语句块 2 <if 后续语句></pre> </div> <p>重点解释说明: 评估<条件>的结果值 两种可能, True 或者 False, 分别形成执行路径 用二分支语句实现两种结果的执行 微实例 4.5: PM 2.5 空气质量提醒 (2) 优: 0<=PM<35 良: 35<=PM</p> <p>分析 输入: PM2.5 数值 处理: if-else 语句 输出: 空气质量提醒</p> <p>代码 PM = eval (input("请输入 PM2.5 数值: ")) if 0<= PM < 35: print("空气优, 快去户外运动!") else : print("空气良, 适度户外活动! ") </p>	讲解及演示
5M	<p>习主席提出了“绿水青山就是金山银山”。良好生态环境既是自然财富, 也是经济财富。部分地区空气质量重度污染, 重点关注。</p> <p>智能提示如何实现优、良、差三等级提醒:</p> <pre>如果 0<= PM < 35 : 提示"空气优质, 快去户外运动!" 如果 35 <= PM <75 : 提示"空气良好, 适度户外活动! " 如果 75<= PM : 提示"空气污染, 请小心!"</pre>	讲解及演示

课时分配	教学内容	方法及手段
10M	<p>4.2.3 多分支结构: if-elif-else 语句 流程图展示及分析、if 语句语法格式: if <条件 1>: 语句块 1 elif <条件 2>: 语句块 2 else: 语句块 N 重点解释说明: 评估<条件>的结果值 多条件判断, 若任何条件都不满足, 执行 else 语句</p>	讲解及演示
10M	<p>用多分支语句实现三种结果的执行 微实例 4.6: PM 2.5 空气质量提醒 (3) 优: $0 \leq PM < 35$ 良: $35 \leq PM < 75$ 差: $75 \leq PM$ 代码 PM = eval(input("请输入 PM2.5 数值: ")) if $0 \leq PM < 35$: print("空气优质, 快去户外运动!") elif $35 \leq PM < 75$: print("空气良好, 适度户外活动! ") else : print("空气污染, 请小心! ")</p> <p>4.3 程序的循环结构 问题: 循环依次输出 0-4?</p>	讲解及演示
	<p>遍历循环: 根据循环执行次数的确定性, 循环可以分为确定次数循环和非确定次数循环。确定次数循环指循环体对循环次数有明确的定义循环次数采用遍历结构中元素个数来体现 Python 通过保留字 for 实现“遍历循环”: for <循环变量> in <遍历结构>: <语句块></p> <p>遍历循环还有一种扩展模式, 使用方法如下:</p>	

课时分配	教学内容	方法及手段
10M	<pre>for <循环变量> in <遍历结构>: <语句块 1> else: <语句块 2></pre> <p>无限循环: while 语句</p> <p>无限循环:</p> <p>无限循环一直保持循环操作直到特定循环条件不被满足才结束,不需要提前知道确定循环次数。</p> <p>Python 通过保留字 while 实现无限循环,使用方法如下:</p> <pre>while <条件>: <语句块>语句块</pre> <p>无限循环也有一种使用保留字 else 的扩展模式:</p> <pre>while <条件>: <语句块 1> else: <语句块 2></pre>	讲解及演示
5M	<p>四、总结:</p> <p>学习完本课内容,能够:</p> <ol style="list-style-type: none"> 1. 深入理解分支结构,掌握单分支、二分支、多分支结构的语句及应用。 2. 总结编程解决问题的基本思路: 问题抽象=>分析 IPO (输入、处理、输出) =>确认程序结构=>选择相应的语句=>程序开发=>解决问题。 3. 解决问题的方法有多种,鼓励生活中多思考,并积极实践学以致用。 	总结
5M	<p>★课程思政建设:</p> <p>在本章的学习中,我们强调了解决问题的能力和思考的重要性。学生将学会如何使用程序控制结构来解决实际问题,培养逻辑思考和判断力。此外,编写猜数字游戏也鼓励学生创造性思考如何设计有趣的用户体验。</p>	思政建设

L6

授课时间	2022. 10. 11	课次	06
授课方式 (请打√)	理论课 <input checked="" type="checkbox"/> 讨论课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其他 <input type="checkbox"/>	课时 安排	2
授课题目 (教学章、节或主题): 第 4 章 程序的控制结构 4.4 random 库的使用 4.5 程序的异常处理			
教学目的与要求: (1)掌握随机库的使用方法。 (2)了解程序的异常处理及用法。			
教学重点及难点: 重点: 掌握 Python 语言标准库 random 的使用。 难点: 学生可能需要时间来理解如何使用 random 库和异常处理。在编写游戏时, 学生需要考虑用户输入的多样性和异常情况。			
作业、讨论题、思考题: 作业: 习题 4.2 讨论题: 讨论使用 random 库生成随机数的应用, 如何提高游戏的趣味性。 思考题: 为什么异常处理在编程中非常重要? 举例说明异常处理的应用场景。			
课后小结: 介绍了一个常用标准库 random 库, 利用它实现了蒙特卡罗方法求解 π 的过程。最后介绍了程序的异常处理操作。			
下节课预习重点: 第 5 章 函数和代码复用			
参考文献: [1] 嵩天、礼欣、黄天羽.《Python 语言程序设计基础(第 2 版)》. 高等教育出版社. 2017			

课时分配	教学内容	方法及手段
5M	<p>一、知识回顾：</p> <p>回顾程序的基本结构，包括分支结构和循环结构。</p>	提问巩固
5M	<p>二、课堂引入：</p>	案例引入
	<p>▲案例引入：</p> <p> 案例：猜数字游戏（升级版）</p> <p> 描述：继续猜数字游戏，但这次计算机将生成一个随机整数作为答案。玩家需要猜测这个数字，并计算机提供过大、过小的提示，以帮助玩家找到答案。这个升级版的游戏需要使用 random 库来生成随机数。</p> <ul style="list-style-type: none"> • 引导学生回顾之前的猜数字游戏，然后介绍新的游戏规则。 • 引出 random 库的概念和异常处理的需求。 <p> 讲解方法：展示游戏的规则和玩法，然后引入 Python 编程的方法，解释如何使用 random 库生成随机数，并为异常情况提供处理方法。</p>	
10M	<p>三、授课内容：</p> <p>循环保留字：break 和 continue</p> <p>循环结构有两个辅助保留字：break 和 continue，它们用来辅助控制循环执行。</p> <p>break 用来跳出最内层 for 或 while 循环，脱离该循环后程序从循环后代吗继续续执行。</p> <p>continue 用来结束当前当次循环，即跳出循环体中下面尚未执行的语句，但不跳出当前循环。</p> <p>对于 while 循环，继续求解循环条件。而对于 for 循环，程序流程接着遍历循环列表。</p>	讲解及演示
10M	<p>4.4 random 库的使用</p> <p>随机数在计算机应用中十分常见，Python 内置的 random 库主要用于产生各种分布的伪随机数序列。random 库采用梅森旋转算法（Mersenne twister）生成伪随机数序列，可用于除随机性要求更高的加解密算法外的大多数工程应用。</p>	讲解及演示

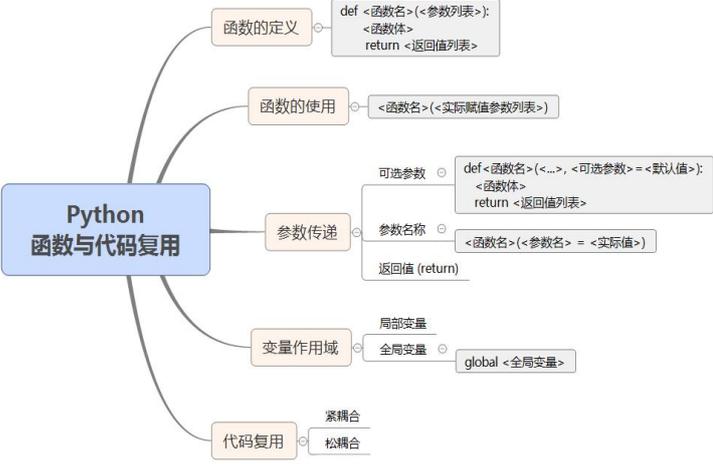
课时分配	教学内容	方法及手段																				
10M	<p>使用 random 库主要目的是生成随机数，因此，读者只需要查阅该库的随机数生成函数，找到符合使用场景的函数使用即可。这个库提供了不同类型的随机数函数，所有函数都是基于最基本的 random.random() 函数扩展而来。</p> <table border="1" data-bbox="261 495 1114 896"> <thead> <tr> <th>函数</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>seed(a=None)</td> <td>初始化随机数种子，默认值为当前系统时间</td> </tr> <tr> <td>random()</td> <td>生成一个[0.0, 1.0)之间的随机小数</td> </tr> <tr> <td>randint(a, b)</td> <td>生成一个[a,b]之间的整数</td> </tr> <tr> <td>getrandbits(k)</td> <td>生成一个k比特长度的随机整数</td> </tr> <tr> <td>randrange(start, stop[, step])</td> <td>生成一个[start, stop)之间以step为步数的随机整数</td> </tr> <tr> <td>uniform(a, b)</td> <td>生成一个[a, b]之间的随机小数</td> </tr> <tr> <td>choice(seq)</td> <td>从序列类型(例如：列表)中随机返回一个元素</td> </tr> <tr> <td>shuffle(seq)</td> <td>将序列类型中元素随机排列，返回打乱后的序列</td> </tr> <tr> <td>sample(pop, k)</td> <td>从pop类型中随机选取k个元素，以列表类型返回</td> </tr> </tbody> </table> <p>对 random 库的引用方法与 math 库一样，采用下面两种方式实现：</p> <pre>import random 或 from random import *</pre> <p>生成随机数之前可以通过 seed() 函数指定随机数种子，随机种子一般是一个整数，只要种子相同，每次生成的随机数序列也相同。这种情况便于测试和同步数据。</p> <pre>>>>seed(125) # 随机种子赋值 125 >>>"{}. {}. {}".format(randint(1, 10), randint(1, 10), randint(1, 10)) '4. 4. 10' >>>"{}. {}. {}".format(randint(1, 10), randint(1, 10), randint(1, 10)) '5. 10. 3' >>>seed(125) # 再次给随机种子赋值 125 >>>"{}. {}. {}".format(randint(1, 10), randint(1, 10), randint(1, 10)) '4. 4. 10'</pre>	函数	描述	seed(a=None)	初始化随机数种子，默认值为当前系统时间	random()	生成一个[0.0, 1.0)之间的随机小数	randint(a, b)	生成一个[a,b]之间的整数	getrandbits(k)	生成一个k比特长度的随机整数	randrange(start, stop[, step])	生成一个[start, stop)之间以step为步数的随机整数	uniform(a, b)	生成一个[a, b]之间的随机小数	choice(seq)	从序列类型(例如：列表)中随机返回一个元素	shuffle(seq)	将序列类型中元素随机排列，返回打乱后的序列	sample(pop, k)	从pop类型中随机选取k个元素，以列表类型返回	讲解及演示
函数	描述																					
seed(a=None)	初始化随机数种子，默认值为当前系统时间																					
random()	生成一个[0.0, 1.0)之间的随机小数																					
randint(a, b)	生成一个[a,b]之间的整数																					
getrandbits(k)	生成一个k比特长度的随机整数																					
randrange(start, stop[, step])	生成一个[start, stop)之间以step为步数的随机整数																					
uniform(a, b)	生成一个[a, b]之间的随机小数																					
choice(seq)	从序列类型(例如：列表)中随机返回一个元素																					
shuffle(seq)	将序列类型中元素随机排列，返回打乱后的序列																					
sample(pop, k)	从pop类型中随机选取k个元素，以列表类型返回																					
15M	<h4>4.5 程序的异常处理</h4> <pre>1 num = eval(input("请输入一个整数: ")) 2 print(num**2)</pre> <p>思考：当用户输入的不是数字呢？</p>	讲解及演示																				

课时分配	教学内容	方法及手段
15M	<p>Python 解释器返回了异常信息，同时程序退出。</p> <p>Python 异常信息中最重要的部分是异常类型，它表明了发生异常的原因，也是程序处理异常的依据。</p> <p>■ 板书内容：</p> <div style="border: 1px solid black; padding: 5px;"> <p>Python 使用 try-except 语句实现异常处理，基本的语法格式如下：</p> <pre>try: <语句块 1> except <异常类型>: <语句块 2></pre> <p>异常的高级用法</p> <p>try-except 语句可以支持多个 except 语句，语法格式如下：</p> <pre>try: <语句块 1> except <异常类型 1>: <语句块 2> ... except <异常类型 N>: <语句块 N+1> except: <语句块 N+2></pre> </div>	<p>板书讲解及演示</p>
10M	<p>最后一个 except 语句没有指定任何类型，表示它对应的语句块可以处理所有其他异常。这个过程与 if-elif-else 语句类似，是分支结构的一种表达方式。</p> <p>实例解析：猜数字游戏</p> <p>编写一个“猜数字游戏”的程序：</p> <p>在 1 到 100 之间指定一个数，然后请用户循环猜测这个数字。</p> <p>对于每个答案只回复“猜大了”或“猜小了”。</p>	<p>讲解及演示</p>

课时分配	教学内容	方法及手段
5M	<p>直到猜测准确为止，输出用户的猜测次数。</p> <p>(1) 完成基本猜数游戏</p> <p>(2) 猜的数字随机产生</p> <p>(3) 异常输入情况做处理</p> <p>四、总结：</p> <p>学习完本课内容，能够掌握常用标准库 random 库，利用它实现了蒙特卡罗方法求解 π 的过程。最后介绍了程序的异常处理操作。</p>	总结
5M	<p>★课程思政建设：</p> <p>在本章的学习中，我们强调了解决问题的能力和思考的重要性。学生将学会如何使用程序控制结构来解决实际问题，培养逻辑思考和判断力。此外，编写猜数字游戏也鼓励学生创造性思考如何设计有趣的用户体验。</p>	思政建设

L7

授课时间	2022. 10. 18	课次	07
授课方式 (请打√)	理论课 <input checked="" type="checkbox"/> 讨论课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其他 <input type="checkbox"/>		课时 安排
2			
<p>授课题目（教学章、节或主题）： 第 5 章 函数和代码复用 5.1 函数的基本使用 5.2 函数的参数传递 5.3 lambda 匿名函数。</p>			
<p>教学目的与要求： (1)掌握函数的定义和调用方法。 (2)理解函数的参数传递过程以及变量的作用范围。 (3)了解 lambda 函数。</p>			
<p>教学重点及难点： 重点：函数的参数传递过程以及变量的作用范围。 难点：需要理解形参和实参的概念与区别，局部变量的作用范围。</p>			
<p>作业、讨论题、思考题： 作业：习题 5.2 讨论题：讨论为什么要编写函数解决问题，函数可以带来什么便捷。 思考题：如何能够更高效地定义函数，包括函数名、参数、返回值的设置。</p>			
<p>课后小结： 本章主要介绍了函数及代码复用问题，包括函数的定义、lambda 函数使用，还介绍了函数参数、返回值等特殊情况的处理。</p>			
<p>下节课预习重点： 第 5 章 函数和代码复用 · 下</p>			
<p>参考文献： [1] 嵩天、礼欣、黄天羽.《Python 语言程序设计基础(第 2 版)》. 高等教育出版社. 2017</p>			

课时分配	教学内容	方法及手段
5M	<p>一、知识回顾：</p> <p>回顾其他语言中函数的含义</p> 	提问巩固
10M	<p>二、课堂引入：</p> <p>▲案例引入：</p> <pre data-bbox="268 1115 1088 1646"> 1 def happy(): 2 print("Happy birthday to you!") 3 def happyB(name): 4 happy() 5 happy() 6 print("Happy birthday, dear {}".format(name)) 7 happy() 8 happyB("Mike") 9 print() 10 happyB("Lily") </pre> <div data-bbox="710 1400 1088 1646" style="border: 1px solid gray; padding: 5px;"> <pre> Happy birthday to you! Happy birthday to you! Happy birthday, dear Mike! Happy birthday to you! Happy birthday to you! Happy birthday to you! Happy birthday, dear Lily! Happy birthday to you! </pre> </div> <p>案例：生日歌。</p> <p>描述：你正在设计一个生日歌生成器程序，该程序可以接受一个姓名，然后生成个性化的生日歌歌词。歌词将包括姓名，例如：“Happy Birthday to [姓名]”这个生成器需要使用函数来组织代码，并在歌词中插入姓名和。这个案例将引入函数的概念和如何将参数传递给函数。</p> <ul style="list-style-type: none"> • 引导学生思考如何手工编写个性化的生日歌。 • 引出函数的概念，如何定义函数并将参数传递给函数。 	案例引入

课时分配	教学内容	方法及手段
10M	<p>讲解方法：展示生成生日歌的需求和工作原理，然后引入 Python 编程的方法，解释如何定义函数来生成歌词，并如何传递姓名和年龄参数给函数。</p> <p>三、授课内容：</p> <p>5.1 函数的基本使用</p> <p>函数是一段具有特定功能的、可重用的语句组，用函数名来表示并通过函数名进行完成功能调用。</p> <p>函数也可以看作是一段具有名字的子程序，可以在需要的地方调用执行，不需要在每个执行地方重复编写这些语句。每次使用函数可以提供不同的参数作为输入，以实现对不同数据的处理；函数执行后，还可以反馈相应的处理结果。</p> <p>■板书内容：</p> <div style="border: 1px solid black; padding: 5px;"> <p>函数的定义</p> <p>Python 定义一个函数使用 def 保留字，语法形式如下：</p> <pre>def <函数名>(<参数列表>): <函数体> return <返回值列表></pre> <p>函数调用的过程</p> <p>程序调用一个函数需要执行以下四个步骤：</p> <ol style="list-style-type: none"> (1) 调用程序在调用处暂停执行； (2) 在调用时将实参复制给函数的形参； (3) 执行函数体语句； (4) 函数调用结束给出返回值，程序回到调用前的暂停处继续执行。 <p>lambda 函数</p> <p>Python 的有 33 个保留字，其中一个是 lambda，该保留字用于定义一种特殊的函数——匿名函数，又称 lambda 函数。</p> <pre>1 # 定义两数求和的函数 2 def sum(n , m): 3 return n + m 4</pre> </div>	<p>方法 手段</p> <p>板书、讲解及演示</p>
10M	<p>lambda 函数</p> <p>Python 的有 33 个保留字，其中一个是 lambda，该保留字用于定义一种特殊的函数——匿名函数，又称 lambda 函数。</p> <pre>1 # 定义两数求和的函数 2 def sum(n , m): 3 return n + m 4</pre>	<p>讲解及演示</p>

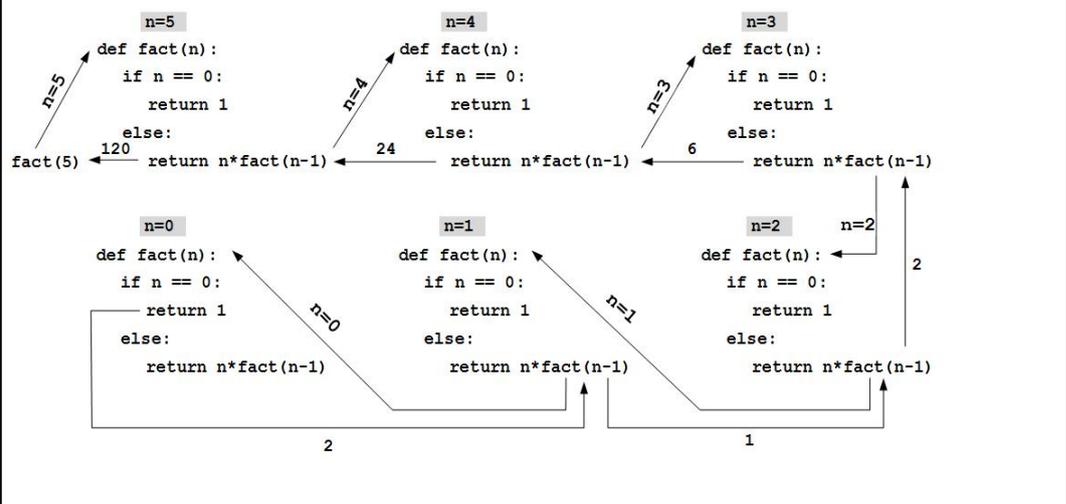
课时分配	教学内容	方法及手段
10M	<p>匿名函数并非没有名字，而是将函数名作为函数结果返回，如下： <函数名> = lambda <参数列表>: <表达式> lambda 函数与正常函数一样，等价于下面形式： def <函数名>(<参数列表>): return <表达式></p> <p>简单说，lambda 函数用于定义简单的、能够在一行内表示的函数，返回一个函数类型。</p> <p>5.2 函数的参数传递 可选参数和可变数量参数</p> <div style="border: 1px solid black; padding: 5px;"> <p>在定义函数时，有些参数可以存在默认值。</p> <ul style="list-style-type: none"> ■ 函数定义时，参数可以指定默认值。 ■ 函数被调用时，如果没有传入对应的参数值，则使用函数定义时的默认值替代。 ■ 可选参数一般都放置非可选参数的后面。 ■ 函数定义时的语法形式如下： <pre>def <函数名>(<非可选参数列表>, <可选参数> = <默认值>): <函数体> return <返回值列表></pre> </div> <p>在函数定义时，可以设计可变数量参数，通过参数前增加星号 (*) 实现 思考：计算两数的和？计算多数的和？</p> <pre>>>>def func(s, *num): print(type(num)) for i in num: s += i return s</pre>	演示及讲解
10M	<p>参数的位置和名称传递</p> <p>Python 提供了按照形参名称输入实参的方式，调用如下： result = func(x2=4, y2=5, z2=6, x1=1, y1=2, z1=3)</p>	演示及讲解

课时分配	教学内容	方法及手段
10M	<p>由于调用函数时指定了参数名称，所以参数之间的顺序可以任意调整。</p> <pre data-bbox="316 371 699 618"> >>>def fun(x, y ,z): return x-y//z #可以按照位置一一对应 x=5 y=4 z=2 >>>fun(5,4,2) 3 此外，Python提供了按照形参名称输入实参的方式 >>>fun(x=5, y=4, z=2) 3 </pre> <p>变量的返回值</p> <p>return 语句用来退出函数并将程序返回到函数被调用的位置继续执行。</p> <p>return 语句同时可以将 0 个、1 个或多个函数运算完的结果返回给函数被调用处的变量。</p> <p>函数可以没有 return，此时函数并不返回值，如微实例 5.1 的 happy() 函数。</p> <p>函数也可以用 return 返回多个值，多个值以元组类型保存。</p>	讲解及演示
15M	<p>函数对变量的作用</p> <p>一个程序中的变量包括两类：全局变量和局部变量。</p> <p>全局变量指在函数之外定义的变量，一般没有缩进，在程序执行全过程有效。</p> <p>局部变量指在函数内部使用的变量，仅在函数内部有效，当函数退出时变量将不存在。</p> <p>Python 函数对变量的作用遵守如下原则：</p> <p>简单数据类型变量无论是否与全局变量重名，仅在函数内部创建和使用，函数退出后变量被释放。</p>	
5M	<p>四、总结：</p> <p>学习完本课内容，能够掌握函数及代码复用问题，包括函数的定义、lambda 函数使用。</p>	总结
5M	<p>★课程思政建设：</p> <p>在本章的学习中，我们鼓励学生思考代码的可维护性和可重用性。函数的使用可以使代码更具结构性，降低代码冗余，提高代码的可读性和可维护性。这种编程思维也有助于培养学生的组织和规划能力。</p>	思政建设

L8

授课时间	2022. 10. 25	课次	08
授课方式 (请打√)	理论课 <input checked="" type="checkbox"/> 讨论课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其他 <input type="checkbox"/>	课时 安排	2
授课题目(教学章、节或主题): 第5章 函数和代码复用 5.4 代码复用和模块化 5.5 函数的递归 5.6 Python 内置函数			
教学目的与要求: (1)理解代码复用、基于函数的模块化设计 (2)理解函数递归的定义和使用方法。			
教学重点及难点: 重点: 代码复用和模块化编程的概念。函数的递归使用。Python 内置函数的用法。 难点: 函数的递归使用可能需要深入理解递归的原理。			
作业、讨论题、思考题: 课后作业: 习题 5.6 讨论题: 讨论代码复用和模块化编程的优点, 如何提高代码的可维护性。 思考题: 递归函数的原理是什么? 在哪些情况下递归函数比迭代更有优势?			
课后小结: 本章主要介绍了函数递归以及参数的位置和名称传递等内容, 讲解了如何使用函数递归绘制复杂精美的科赫曲线。			
下节课预习重点: 第6章 组合数据类型			
参考文献: [1] 嵩天、礼欣、黄天羽.《Python 语言程序设计基础(第2版)》. 高等教育出版社. 2017			

课时分配	教学内容	方法及手段
5M	<p>一、知识回顾： 回顾函数基本概念及基本使用方法</p>	提问巩固
5M	<p>二、课堂引入： ▲案例引入： 案例：绘制分形树 描述：假设你想使用 Python 绘制分形树，这是一种树的图形，树干逐渐分叉成更小的树干，形成一个美丽的自相似结构。这需要递归函数的使用，以不断分叉树干并调整角度和长度。 • 引导学生思考如何手工绘制分形树，以及如何将这个过程转化为可重复使用的代码。 • 引出代码复用和模块化的概念以及函数的递归使用。 讲解方法：展示绘制分形树的步骤，然后引入 Python 编程的方法，解释如何创建和使用模块来实现这个分形树，以及如何使用递归函数。</p>	案例引入
10M	<p>三、授课内容： 5.4 代码复用和模块化 函数是程序的一种基本抽象方式，它将一系列代码组织起来通过命名供其他程序使用。函数封装的直接好处是代码复用，任何其他代码只要输入参数即可调用函数，从而避免相同功能代码在被调用处重复编写。代码复用产生了另一个好处，当更新函数功能时，所有被调用处的功能都被更新。 ■板书内容：</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>模块化设计一般有两个基本要求： 紧耦合：尽可能合理划分功能块，功能块内部耦合紧密； 松耦合：模块间关系尽可能简单，功能块之间耦合度低。 使用函数只是模块化设计的必要非充分条件，根据计算需求合理划分函数十分重要。一般来说，完成特定功能或被经常复用的一组语句应该采用函数来封装，并尽可能减少函数间参数和返回值的数量。</p> </div>	板书及讲解

课时分配	教学内容	方法及手段
10M	<p>5.5 函数的递归</p> <p>函数作为一种代码封装，可以被其他程序调用，当然，也可以被函数内部代码调用。这种函数定义中调用函数自身的方式称为递归。就像一个人站在装满镜子的房间中，看到的影像就是递归的结果。递归在数学和计算机应用上非常强大，能够非常简洁的解决重要问题。</p> <p>递归的案例</p> <p>数学上有个经典的递归例子叫阶乘，阶乘通常定义为：</p> $n! = n(n-1)(n-2)...(1)$ <p>这个关系给出了另一种方式表达阶乘的方式：</p> $n! = \begin{cases} 1 & n = 0 \\ n(n-1)! & \text{otherwise} \end{cases}$ <p>阶乘的例子揭示了递归的 2 个关键特征：</p> <ol style="list-style-type: none"> (1) 存在一个或多个基例，基例不需要再次递归，它是确定的表达式； (2) 所有递归链要以一个或多个基例结尾。 <p>微实例 5.21：阶乘的计算。</p> <p>根据用户输入的整数 n，计算并输出 n 的阶乘值。</p>	讲解及演示
10M		板书、讲解及演示
10M	<p>微实例 5.32：字符串反转。</p> <p>对于用户输入的字符串 s，输出反转后的字符串。</p> <p>这个问题的基本思想是把字符串看作一个递归对象。</p>	讲解及演示

课时分配	教学内容	方法及手段																																																																						
10M	<p>观察这个函数的工作过程。s[0]是首字符，s[1:]是剩余字符串，将它们反向连接，可以得到反转字符串。执行这个程序，结果如下</p> <pre>def reverse(s): if len(s) == 1: return s else: return reverse(s[1:]) + s[0] print(reverse('ABC'))</pre> <p>5.6 Python 内置函数</p> <p>Python 解释器提供了 68 个内置函数，其中，前 36 个已经将结果，需要掌握。</p> <table border="1" data-bbox="209 864 1118 1335"> <tbody> <tr><td>abs()</td><td>id()</td><td>round()</td><td>compile()</td><td>locals()</td></tr> <tr><td>all()</td><td>input()</td><td>set()</td><td>dir()</td><td>map()</td></tr> <tr><td>any()</td><td>int()</td><td>sorted()</td><td>exec()</td><td>memoryview()</td></tr> <tr><td>ascii()</td><td>len()</td><td>str()</td><td>enumerate()</td><td>next()</td></tr> <tr><td>bin()</td><td>list()</td><td>tuple()</td><td>filter()</td><td>object()</td></tr> <tr><td>bool()</td><td>max()</td><td>type()</td><td>format()</td><td>property()</td></tr> <tr><td>chr()</td><td>min()</td><td>zip()</td><td>frozenset()</td><td>repr()</td></tr> <tr><td>complex()</td><td>oct()</td><td></td><td>getattr()</td><td>setattr()</td></tr> <tr><td>dict()</td><td>open()</td><td></td><td>globals()</td><td>slice()</td></tr> <tr><td>divmod()</td><td>ord()</td><td>bytes()</td><td>hasattr()</td><td>staticmethod()</td></tr> <tr><td>eval()</td><td>pow()</td><td>delattr()</td><td>help()</td><td>sum()</td></tr> <tr><td>float()</td><td>print()</td><td>bytearray()</td><td>isinstance()</td><td>super()</td></tr> <tr><td>hash()</td><td>range()</td><td>callable()</td><td>issubclass()</td><td>vars()</td></tr> <tr><td>hex()</td><td>reversed()</td><td>classmethod()</td><td>iter()</td><td><u>__import()</u></td></tr> </tbody> </table>	abs()	id()	round()	compile()	locals()	all()	input()	set()	dir()	map()	any()	int()	sorted()	exec()	memoryview()	ascii()	len()	str()	enumerate()	next()	bin()	list()	tuple()	filter()	object()	bool()	max()	type()	format()	property()	chr()	min()	zip()	frozenset()	repr()	complex()	oct()		getattr()	setattr()	dict()	open()		globals()	slice()	divmod()	ord()	bytes()	hasattr()	staticmethod()	eval()	pow()	delattr()	help()	sum()	float()	print()	bytearray()	isinstance()	super()	hash()	range()	callable()	issubclass()	vars()	hex()	reversed()	classmethod()	iter()	<u>__import()</u>	讲解及演示
abs()	id()	round()	compile()	locals()																																																																				
all()	input()	set()	dir()	map()																																																																				
any()	int()	sorted()	exec()	memoryview()																																																																				
ascii()	len()	str()	enumerate()	next()																																																																				
bin()	list()	tuple()	filter()	object()																																																																				
bool()	max()	type()	format()	property()																																																																				
chr()	min()	zip()	frozenset()	repr()																																																																				
complex()	oct()		getattr()	setattr()																																																																				
dict()	open()		globals()	slice()																																																																				
divmod()	ord()	bytes()	hasattr()	staticmethod()																																																																				
eval()	pow()	delattr()	help()	sum()																																																																				
float()	print()	bytearray()	isinstance()	super()																																																																				
hash()	range()	callable()	issubclass()	vars()																																																																				
hex()	reversed()	classmethod()	iter()	<u>__import()</u>																																																																				
10M	<p>5.3 datetime 库的使用</p> <p>以不同格式显示日期和时间是程序中最常用到的功能。Python 提供了一个处理时间的标准函数库 datetime，它提供了一系列由简单到复杂的时间处理方法。datetime 库可以从系统中获得时间，并以用户选择的格式输出。</p> <p>datetime 库概述</p> <p>datetime 库以类的方式提供多种日期和时间表达方式：</p> <p>datetime.date：日期表示类，可以表示年、月、日等</p> <p>datetime.time：时间表示类，可以表示小时、分钟、秒、毫秒等</p> <p>datetime.datetime：日期和时间表示的类，功能覆盖 date 和 time 类</p> <p>datetime.timedelta：时间间隔有关的类</p>	讲解及演示																																																																						

课时分配	教学内容	方法及手段																																							
10M	<p>datetime.tzinfo: 与时区有关的信息表示类</p> <p>datetime 库解析</p> <p>使用 datetime.now() 获得当前日期和时间对象，使用方法如下：</p> <pre>datetime.now()</pre> <p>作用：返回一个 datetime 类型，表示当前的日期和时间，精确到微秒。</p> <p>使用 datetime.utcnow() 获得当前日期和时间对应的 UTC（世界标准时间）时间对象，使用方法如下：</p> <pre>datetime.utcnow()</pre> <p>作用：返回 datetime 类型，表示当前日期和时间的 UTC 表示，精确到微秒。</p> <p>调用 datetime() 函数直接创建一个 datetime 对象</p> <p>datetime 对象有 3 个常用的时间格式化方法</p> <table border="1" data-bbox="296 958 954 1281"> <thead> <tr> <th>格式化字符串</th> <th>日期/时间</th> <th>值范围和实例</th> </tr> </thead> <tbody> <tr> <td>%Y</td> <td>年份</td> <td>0001-9999，例如：1900</td> </tr> <tr> <td>%m</td> <td>月份</td> <td>01-12，例如：10</td> </tr> <tr> <td>%B</td> <td>月名</td> <td>January-December，例如：April</td> </tr> <tr> <td>%b</td> <td>月名缩写</td> <td>Jan-Dec，例如：Apr</td> </tr> <tr> <td>%d</td> <td>日期</td> <td>01 ~ 31，例如：25</td> </tr> <tr> <td>%A</td> <td>星期</td> <td>Monday-Sunday，例如：Wednesday</td> </tr> <tr> <td>%a</td> <td>星期缩写</td> <td>Mon-Sun，例如：Wed</td> </tr> <tr> <td>%H</td> <td>小时（24h制）</td> <td>00 ~ 23，例如：12</td> </tr> <tr> <td>%I</td> <td>小时（12h制）</td> <td>01 ~ 12，例如：7</td> </tr> <tr> <td>%p</td> <td>上下午</td> <td>AM, PM，例如：PM</td> </tr> <tr> <td>%M</td> <td>分钟</td> <td>00 ~ 59，例如：26</td> </tr> <tr> <td>%S</td> <td>秒</td> <td>00 ~ 59，例如：26</td> </tr> </tbody> </table> <p>strftime() 格式化字符串的数字左侧会自动补零，上述格式也可以与 print() 的格式化函数一起使用。</p>	格式化字符串	日期/时间	值范围和实例	%Y	年份	0001-9999，例如：1900	%m	月份	01-12，例如：10	%B	月名	January-December，例如：April	%b	月名缩写	Jan-Dec，例如：Apr	%d	日期	01 ~ 31，例如：25	%A	星期	Monday-Sunday，例如：Wednesday	%a	星期缩写	Mon-Sun，例如：Wed	%H	小时（24h制）	00 ~ 23，例如：12	%I	小时（12h制）	01 ~ 12，例如：7	%p	上下午	AM, PM，例如：PM	%M	分钟	00 ~ 59，例如：26	%S	秒	00 ~ 59，例如：26	讲解及演示
格式化字符串	日期/时间	值范围和实例																																							
%Y	年份	0001-9999，例如：1900																																							
%m	月份	01-12，例如：10																																							
%B	月名	January-December，例如：April																																							
%b	月名缩写	Jan-Dec，例如：Apr																																							
%d	日期	01 ~ 31，例如：25																																							
%A	星期	Monday-Sunday，例如：Wednesday																																							
%a	星期缩写	Mon-Sun，例如：Wed																																							
%H	小时（24h制）	00 ~ 23，例如：12																																							
%I	小时（12h制）	01 ~ 12，例如：7																																							
%p	上下午	AM, PM，例如：PM																																							
%M	分钟	00 ~ 59，例如：26																																							
%S	秒	00 ~ 59，例如：26																																							
5M	<p>四、总结：</p> <p>学习完本课内容，能够掌握函数递归以及参数的位置和名称传递等内容，讲解了如何使用函数递归绘制复杂精美的科赫曲线。</p>	总结																																							
5M	<p>★课程思政建设：</p> <p>在本章的学习中，我们注重培养学生的系统思维和创新的能力。模块化编程和函数的递归使用有助于学生更好地组织和规划代码，同时激发他们的创造性思维。此外，学生也将学会如何使用 Python 内置函数来解决各种问题。</p>	思政建设																																							

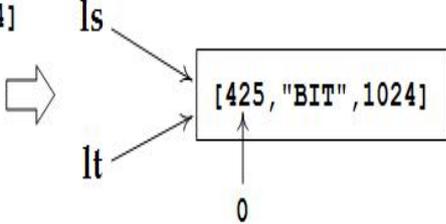
L9

授课时间	2022. 11. 01	课次	09
授课方式 (请打√)	理论课 <input checked="" type="checkbox"/> 讨论课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其他 <input type="checkbox"/>	课时 安排	2
<p>授课题目 (教学章、节或主题):</p> <p>第 6 章 组合数据类型</p> <p>6.1 组合数据类型概述</p> <p>6.2 列表类型及操作</p>			
<p>教学目的与要求:</p> <p>(1) 了解 3 类基本组合数据类型。</p> <p>(2) 理解列表概念并掌握 Python 中列表的使用。</p> <p>(3) 运用列表管理采集的信息, 构建数据结构。</p>			
<p>教学重点及难点:</p> <p>重点: 组合数据类型的概念, 包括列表、元组和字典。列表类型的创建、操作、索引和切片。使用列表处理数据的方法。</p> <p>难点: 学生可能需要时间来理解组合数据类型的概念和用法。索引和切片的使用可能需要一些实践。</p>			
<p>作业、讨论题、思考题:</p> <p>作业: 习题 6.1。</p> <p>讨论题: 讨论组合数据类型的作用和用途, 如何提高数据管理和处理的效率。</p> <p>思考题: 为什么组合数据类型对于数据管理和处理非常重要? 举例说明组合数据类型如何提高数据操作的灵活性。</p>			
<p>课后小结:</p> <p>本章主要介绍了组合数据类型中元组、数组、列表等类型及基本操作。</p>			
<p>下节课预习重点:</p> <p>第 6 章 组合数据类型 · 下</p>			
<p>参考文献:</p> <p>[1] 嵩天、礼欣、黄天羽.《Python 语言程序设计基础(第 2 版)》. 高等教育出版社. 2017</p>			

课时分配	教学内容	方法及手段
5M	<p>一、知识回顾：</p> <p>回顾数值、字符串基本类型的使用，思考编程计算班级同学的平均成绩？</p>	提问巩固
5M	<p>二、课堂引入：</p> <p>▲案例引入：</p> <p> 案例：学生成绩管理系统</p> <p> 描述：你是一位老师，需要管理学生的考试成绩。你希望使用 Python 编写程序来存储学生成绩并执行不同的操作，如添加、查找、修改和删除成绩。这需要使用组合数据类型来存储学生数据。</p> <ul style="list-style-type: none"> • 引导学生思考如何手工管理学生成绩，以及如何将这个过程转化为可重复使用的代码。 • 引出组合数据类型的概念和列表类型的用途。 <p> 讲解方法：展示学生成绩管理系统的需求，然后引入 Python 编程的方法，解释如何使用列表类型来存储学生成绩，执行各种操作，并进行数据处理。</p>	案例引入
10M	<p>三、授课内容：</p> <p>6.1 组合数据类型概述</p> <p>计算机不仅对单个变量表示的数据进行处理，更多情况，计算机需要对一组数据进行批量处理。一些例子包括：</p> <p>给定一组单词 {python, data, function, list, loop}，计算并输出每个单词的长度；</p> <p>给定一个学院学生信息，统计一下男女生比例；</p> <p>一次实验产生了很多组数据，对这些大量数据进行分析；</p> <p>组合数据类型能够将多个同类型或不同类型的数据组织起来，通过单一表示使数据操作更有序更容易。根据数据之间的关系，组合数据类型可以分为三类：</p> <p>■板书内容：</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>序列类型、集合类型和映射类型。</p> <p>序列类型是一个元素向量，元素之间存在先后关系，通过序号访问，元素之</p> </div>	讲解及板书

课时分配	教学内容	方法及手段
10M	<p>间不排他。</p> <p>集合类型是一个元素集合，元素之间无序，相同元素在集合中唯一存在。</p> <p>映射类型是“键-值”数据项的组合，每个元素是一个键值对，表示为 (key, value)。</p> <div style="text-align: center;"> <pre> graph LR A[组合数据类型] --- B[序列类型] A --- C[映射类型] B --- D["字符串(str) (3.5节)"] B --- E["元组(tuple) (6.1节)"] B --- F["列表(list) (6.2节)"] C --- G["字典(map) (6.4节)"] H[集合类型] --- I["集合(set) (6.1节)"] </pre> </div> <p>Python 语言中有很多数据类型都是序列类型，其中比较重要的是：str（字符串）、tuple（元组）和 list（列表）。</p> <p>元组是包含 0 个或多个数据项的不可变序列类型。元组生成后是固定的，其中任何数据项不能替换或删除。</p> <p>列表则是一个可以修改数据项的序列类型，使用也最灵活。</p> <p>元组（tuple）是序列类型中比较特殊的类型，因为它一旦创建就不能被修改。元组类型在表达固定数据项、函数多返回值、多变量同步赋值、循环遍历等情况下十分有用。Python 中元组采用逗号和圆括号（可选）来表示。</p> <p>集合类型与数学中集合的概念一致，即包含 0 个或多个数据项的无序组合。集合中元素不可重复，元素类型只能是固定数据类型，例如：整数、浮点数、字符串、元组等，列表、字典和集合类型本身都是可变数据类型，不能作为集合的元素出现。</p>	板书
15M	<p>序列类型常见操作</p> <p>序列类型支持成员关系操作符（in）、长度计算函数（len()）、分片（[]），元素本身也可以是序列类型。</p> <p>序列类型有 12 个通用的操作符和函数：</p>	讲解及演示

课时分配	教学内容	方法及手段																										
15M	<table border="1"> <thead> <tr> <th data-bbox="209 320 443 371">操作符</th> <th data-bbox="443 320 1225 371">描述</th> </tr> </thead> <tbody> <tr> <td data-bbox="209 371 443 423">x in s</td> <td data-bbox="443 371 1225 423">如果x是s的元素，返回True，否则返回False</td> </tr> <tr> <td data-bbox="209 423 443 474">x not in s</td> <td data-bbox="443 423 1225 474">如果x不是s的元素，返回True，否则返回False</td> </tr> <tr> <td data-bbox="209 474 443 526">s + t</td> <td data-bbox="443 474 1225 526">连接s和t</td> </tr> <tr> <td data-bbox="209 526 443 577">s * n 或 n * s</td> <td data-bbox="443 526 1225 577">将序列s复制n次</td> </tr> <tr> <td data-bbox="209 577 443 629">s[i]</td> <td data-bbox="443 577 1225 629">索引，返回序列的第i个元素</td> </tr> <tr> <td data-bbox="209 629 443 680">s[i: j]</td> <td data-bbox="443 629 1225 680">切片，返回包含序列s第i到j个元素的子序列（不包含第j个元素）</td> </tr> <tr> <td data-bbox="209 680 443 732">s[i: j: k]</td> <td data-bbox="443 680 1225 732">步骤切片，返回包含序列s第i到j个元素以j为步数的子序列</td> </tr> <tr> <td data-bbox="209 732 443 784">len(s)</td> <td data-bbox="443 732 1225 784">序列s的元素个数（长度）</td> </tr> <tr> <td data-bbox="209 784 443 835">min(s)</td> <td data-bbox="443 784 1225 835">序列s中的最小元素</td> </tr> <tr> <td data-bbox="209 835 443 887">max(s)</td> <td data-bbox="443 835 1225 887">序列s中的最大元素</td> </tr> <tr> <td data-bbox="209 887 443 938">s.index(x, i, j)]</td> <td data-bbox="443 887 1225 938">序列s中从i开始到j位置中第一次出现元素x的位置</td> </tr> <tr> <td data-bbox="209 938 443 990">s.count(x)</td> <td data-bbox="443 938 1225 990">序列s中出现x的总次数</td> </tr> </tbody> </table>	操作符	描述	x in s	如果x是s的元素，返回True，否则返回False	x not in s	如果x不是s的元素，返回True，否则返回False	s + t	连接s和t	s * n 或 n * s	将序列s复制n次	s[i]	索引，返回序列的第i个元素	s[i: j]	切片，返回包含序列s第i到j个元素的子序列（不包含第j个元素）	s[i: j: k]	步骤切片，返回包含序列s第i到j个元素以j为步数的子序列	len(s)	序列s的元素个数（长度）	min(s)	序列s中的最小元素	max(s)	序列s中的最大元素	s.index(x, i, j)]	序列s中从i开始到j位置中第一次出现元素x的位置	s.count(x)	序列s中出现x的总次数	讲解及演示
	操作符	描述																										
	x in s	如果x是s的元素，返回True，否则返回False																										
	x not in s	如果x不是s的元素，返回True，否则返回False																										
	s + t	连接s和t																										
	s * n 或 n * s	将序列s复制n次																										
	s[i]	索引，返回序列的第i个元素																										
	s[i: j]	切片，返回包含序列s第i到j个元素的子序列（不包含第j个元素）																										
	s[i: j: k]	步骤切片，返回包含序列s第i到j个元素以j为步数的子序列																										
	len(s)	序列s的元素个数（长度）																										
	min(s)	序列s中的最小元素																										
	max(s)	序列s中的最大元素																										
	s.index(x, i, j)]	序列s中从i开始到j位置中第一次出现元素x的位置																										
	s.count(x)	序列s中出现x的总次数																										
	<p>序列类型—字符串 str 基本操作举例</p> <p>序列类型—元组 tuple 基本操作举例</p> <p>序列类型—列表 list 基本操作举例</p>																											
<p>列表类型和操作</p>																												
<p>列表类型继承序列类型特点，有一些通用的操作函数</p>																												
<p>列表类型存在一些操作方法，使用语法形式是：</p>																												
<table border="1"> <thead> <tr> <th data-bbox="209 1458 539 1509">方法</th> <th data-bbox="539 1458 1270 1509">描述</th> </tr> </thead> <tbody> <tr> <td data-bbox="209 1509 539 1561">ls.append(x)</td> <td data-bbox="539 1509 1270 1561">在列表ls最后增加一个元素x</td> </tr> <tr> <td data-bbox="209 1561 539 1612">ls.insert(i, x)</td> <td data-bbox="539 1561 1270 1612">在列表ls第i位置增加元素x</td> </tr> <tr> <td data-bbox="209 1612 539 1664">ls.clear()</td> <td data-bbox="539 1612 1270 1664">删除ls中所有元素</td> </tr> <tr> <td data-bbox="209 1664 539 1715">ls.pop(i)</td> <td data-bbox="539 1664 1270 1715">将列表ls中第i项元素取出并删除该元素</td> </tr> <tr> <td data-bbox="209 1715 539 1767">ls.remove(x)</td> <td data-bbox="539 1715 1270 1767">将列表中出现的第一个元素x删除</td> </tr> <tr> <td data-bbox="209 1767 539 1818">ls.reverse()</td> <td data-bbox="539 1767 1270 1818">列表ls中元素反转</td> </tr> <tr> <td data-bbox="209 1818 539 1870">ls.copy()</td> <td data-bbox="539 1818 1270 1870">生成一个新列表，复制ls中所有元素</td> </tr> </tbody> </table>	方法	描述	ls.append(x)	在列表ls最后增加一个元素x	ls.insert(i, x)	在列表ls第i位置增加元素x	ls.clear()	删除ls中所有元素	ls.pop(i)	将列表ls中第i项元素取出并删除该元素	ls.remove(x)	将列表中出现的第一个元素x删除	ls.reverse()	列表ls中元素反转	ls.copy()	生成一个新列表，复制ls中所有元素												
方法	描述																											
ls.append(x)	在列表ls最后增加一个元素x																											
ls.insert(i, x)	在列表ls第i位置增加元素x																											
ls.clear()	删除ls中所有元素																											
ls.pop(i)	将列表ls中第i项元素取出并删除该元素																											
ls.remove(x)	将列表中出现的第一个元素x删除																											
ls.reverse()	列表ls中元素反转																											
ls.copy()	生成一个新列表，复制ls中所有元素																											
<p>特殊-引用</p> <p>与整数和字符串不同，列表要处理一组数据。</p> <p>因此，列表必须通过显式的数据赋值才能生成，简单将一个列表赋值给</p>																												

课时分配	教学内容	方法及手段
10M	<p>另一个列表不会生成新的列表对象。</p> <pre data-bbox="268 405 624 607"> >>>ls = [425, "BIT", 1024] >>>lt = ls >>>ls[0] = 0 >>>lt </pre>  <p>修改列表元素值</p> <p>当使用一个列表改变另一个列表值时，Python 不要求两个列表长度一样，但遵循“多增少减”的原则，例子如下：</p> <pre data-bbox="268 853 1118 1182"> >>>vlist[1:3]=[“bit”，“new”，123] # 2 个 = 3 个 >>>vlist [0 , 'bit', 'new', 123, 'python', 4] >>>vlist[1:3]=["fewer"] #2 个 = 1 个 >>>vlist [0, 'fewer', 123, 'python', 4] </pre> <p>遍历</p> <p>字符串、元组、列表可以通过 for...in 语句对其元素进行遍历，基本语法结构如下：</p> <pre data-bbox="268 1384 826 1473"> for <任意变量名> in <序列名>: 语句块 </pre> <p>列表是一个十分灵活的数据结构，它具有处理任意长度、混合类型的能力，并提供了丰富的基础操作符和方法。当程序需要使用组合数据类型管理批量数据时，请尽量使用列表类型。</p> <p>列表类型的操作练习 P162</p>	讲解及练习
10M	<p>集合类型</p> <p>由于集合是无序组合，它没有索引和位置的概念，不能分片，集合中元素可以动态增加或删除。集合用大括号（{}）表示，可以用赋值语句生成一个集合。</p>	讲解及演示

L10

授课时间	2022. 11. 08	课次	10
授课方式 (请打√)	理论课 <input checked="" type="checkbox"/> 讨论课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其他 <input type="checkbox"/>	课时 安排	2
授课题目 (教学章、节或主题): 第 6 章 组合数据类型 6.3 字典类型及操作 6.4 jieba 库的使用 6.5 Python 之禅			
教学目的与要求: (3)理解字典概念并掌握 Python 中字典的使用。 (4)运用字典处理复杂的数据信息。 (5)运用组合数据类型进行文本词频统计。			
教学重点及难点: 重点: 字典数据类型的概念, 包括创建、操作、索引和遍历。jieba 库的基本使用和中文文本分词。 难点: Python 之禅的核心原则可能需要深入思考和讨论。			
作业、讨论题、思考题: 作业: 习题 6.4。 讨论题: 讨论字典数据类型的作用和用途, 如何用于数据存储和检索。 思考题: 为什么字典数据类型对于存储和检索数据非常有用?			
课后小结: 本章主要介绍了组合数据类型中字典等类型及基本操作, 讲解了如何使用 jieba 词库对中文文档进行分词并进一步统计文档词频, 最后利用 Python 之禅的例子介绍了编写好代码的基本原则。			
下节课预习重点: 第 7 章 文件和数据格式化			
参考文献: [1] 嵩天、礼欣、黄天羽.《Python 语言程序设计基础(第 2 版)》. 高等教育出版社. 2017			

课时分配	教学内容	方法及手段
5M	<p>一、知识回顾：</p> <p>回顾组合数据类型概念及常见类型含义</p>	提问巩固
5M	<p>二、课堂引入：</p> <p>▲案例引入：</p> <p>案例：中文文本分析</p> <p>描述：假设你需要分析一段中文文本，以了解其中的关键词和主题。你希望使用 Python 编写程序来将文本分成词汇，并统计每个词汇的出现次数。这需要使⽤字典数据类型和 jieba 库来实现。</p> <ul style="list-style-type: none"> • 引导学生思考如何手工进行文本分析，以及如何将这个过程转化为可重复使用的代码。 • 引出字典数据类型、jieba 库和 Python 之禅的概念。 <p>讲解方法：展示中文文本分析的需求，然后引入 Python 编程的方法，解释如何使用字典来存储词汇出现次数，以及如何使用 jieba 库进行中文文本分词。</p>	案例引入
10M	<p>三、授课内容：</p> <p>6.3 字典类型及操作</p> <p>字典类型的基本概念</p> <p>通过任意键信息查找一组数据中值信息的过程叫映射,Python 语言中通过字典实现映射。Python 语言中的字典可以通过大括号({})建立,建立模式如下:</p> <p>{<键 1>:<值 1>, <键 2>:<值 2>, … , <键 n>:<值 n>}</p> <p>其中,键和值通过冒号连接,不同键值对通过逗号隔开。</p> <p>字典最主要的用法是查找与特定键相对应的值,这通过索引符号来实现。</p> <p>一般来说,字典中键值对的访问模式如下,采用中括号格式:</p>	讲解
10M	<p style="padding-left: 40px;"><值> = <字典变量>[<键>]</p> <p>字典中对某个键值的修改可以通过中括号的访问和赋值实现:</p> <p style="padding-left: 40px;">通过中括号可以增加新的元素</p> <p style="padding-left: 40px;">直接使用大括号({})可以创建一个空的字典,并通过中括号([])向其增加元素。</p>	讲解及板书

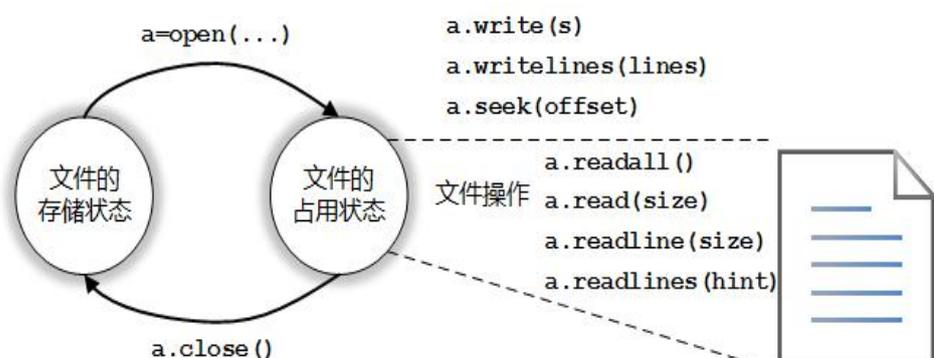
课时分配	教学内容	方法及手段																
5M	<h3>6.4 jieba 库的使用</h3> <p>jieba 是 Python 中一个重要的第三方中文分词函数库</p> <p>jieba 库是第三方库，不是安装包自带，需要通过 pip 指令安装</p> <pre style="border: 1px solid green; padding: 5px;">:\>pip install jieba # 或者 pip3 install jieba</pre> <table border="1" data-bbox="212 651 994 981"> <thead> <tr> <th>函数</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>jieba.cut(s)</td> <td>精确模式，返回一个可迭代的数据类型</td> </tr> <tr> <td>jieba.cut(s, cut_all=True)</td> <td>全模式，输出文本s中所有可能单词</td> </tr> <tr> <td>jieba.cut_for_search(s)</td> <td>搜索引擎模式，适合搜索引擎建立索引的分词结果</td> </tr> <tr> <td>jieba.lcut(s)</td> <td>精确模式，返回一个列表类型，建议使用</td> </tr> <tr> <td>jieba.lcut(s, cut_all=True)</td> <td>全模式，返回一个列表类型，建议使用</td> </tr> <tr> <td>jieba.lcut_for_search(s)</td> <td>搜索引擎模式，返回一个列表类型，建议使用</td> </tr> <tr> <td>jieba.add_word(w)</td> <td>向分词词典中增加新词w</td> </tr> </tbody> </table>	函数	描述	jieba.cut(s)	精确模式，返回一个可迭代的数据类型	jieba.cut(s, cut_all=True)	全模式，输出文本s中所有可能单词	jieba.cut_for_search(s)	搜索引擎模式，适合搜索引擎建立索引的分词结果	jieba.lcut(s)	精确模式，返回一个列表类型，建议使用	jieba.lcut(s, cut_all=True)	全模式，返回一个列表类型，建议使用	jieba.lcut_for_search(s)	搜索引擎模式，返回一个列表类型，建议使用	jieba.add_word(w)	向分词词典中增加新词w	讲解及演示
函数	描述																	
jieba.cut(s)	精确模式，返回一个可迭代的数据类型																	
jieba.cut(s, cut_all=True)	全模式，输出文本s中所有可能单词																	
jieba.cut_for_search(s)	搜索引擎模式，适合搜索引擎建立索引的分词结果																	
jieba.lcut(s)	精确模式，返回一个列表类型，建议使用																	
jieba.lcut(s, cut_all=True)	全模式，返回一个列表类型，建议使用																	
jieba.lcut_for_search(s)	搜索引擎模式，返回一个列表类型，建议使用																	
jieba.add_word(w)	向分词词典中增加新词w																	
10M	<h3>6.5 Python 之禅</h3> <p>Python 编译器以函数库的形式内置了一个有趣的文件，被称为“Python 之禅”（The Zen of Python）。当调用如下一行语句后，会出现一段有趣的运行结果。</p> <pre>>>>import this</pre> <table border="1" data-bbox="217 1303 1134 1877"> <thead> <tr> <th>Python之禅 作者: Tim Peters</th> <th>译者心得</th> </tr> </thead> <tbody> <tr> <td> <ul style="list-style-type: none"> • 优美胜于丑陋 • 明了胜于隐晦 • 简洁胜于复杂 • 复杂胜于凌乱 • 扁平胜于嵌套 • 间隔胜于紧凑 • 可读性很重要 • 即便假借特例的实用性之名，也不要违背上述规则 • 除非你确定需要，任何错误都应该有应对 • 当存在多种可能，不要尝试去猜测 • 只要你不是Guido，对于问题尽量找一种，最好是唯一明显的解决方案 • 做也许好过不做，但不假思索就动手还不如不做 • 如果你无法向人描述你的实现方案，那肯定不是一个好方案 • 如果实现方案容易解释，可能是个好方案 • 命名空间是绝妙的理念，要多运用 </td> <td> <p>以编写优美代码为目标，不多解释 优美代码应该清晰明了，规范统一 优美代码应该逻辑简洁，避免复杂逻辑 如果必须采用复杂逻辑，接口关系也要清晰 优美代码应该是扁平的，避免太多层次嵌套 优美代码间隔要适当，每行代码解决适度问题 优美代码必须是可读且易读的 上述规则是至高无上的</p> <p>捕获异常，不让程序留有因错误退出的可能</p> <p>不要试图给出多种方案，找到一种实现它，几乎所有人没有Guido那么牛 编程之前要有思考</p> <p>能说清楚的往往才是对的</p> <p>适合复杂程序编程</p> </td> </tr> </tbody> </table> <p>什么样的程序是好的？如何编写漂亮的代码？这都是学习编程一段时间最经常提出的问题，却最难回答。程序设计语言如同自然语言，好的代码</p>	Python之禅 作者: Tim Peters	译者心得	<ul style="list-style-type: none"> • 优美胜于丑陋 • 明了胜于隐晦 • 简洁胜于复杂 • 复杂胜于凌乱 • 扁平胜于嵌套 • 间隔胜于紧凑 • 可读性很重要 • 即便假借特例的实用性之名，也不要违背上述规则 • 除非你确定需要，任何错误都应该有应对 • 当存在多种可能，不要尝试去猜测 • 只要你不是Guido，对于问题尽量找一种，最好是唯一明显的解决方案 • 做也许好过不做，但不假思索就动手还不如不做 • 如果你无法向人描述你的实现方案，那肯定不是一个好方案 • 如果实现方案容易解释，可能是个好方案 • 命名空间是绝妙的理念，要多运用 	<p>以编写优美代码为目标，不多解释 优美代码应该清晰明了，规范统一 优美代码应该逻辑简洁，避免复杂逻辑 如果必须采用复杂逻辑，接口关系也要清晰 优美代码应该是扁平的，避免太多层次嵌套 优美代码间隔要适当，每行代码解决适度问题 优美代码必须是可读且易读的 上述规则是至高无上的</p> <p>捕获异常，不让程序留有因错误退出的可能</p> <p>不要试图给出多种方案，找到一种实现它，几乎所有人没有Guido那么牛 编程之前要有思考</p> <p>能说清楚的往往才是对的</p> <p>适合复杂程序编程</p>	演示												
Python之禅 作者: Tim Peters	译者心得																	
<ul style="list-style-type: none"> • 优美胜于丑陋 • 明了胜于隐晦 • 简洁胜于复杂 • 复杂胜于凌乱 • 扁平胜于嵌套 • 间隔胜于紧凑 • 可读性很重要 • 即便假借特例的实用性之名，也不要违背上述规则 • 除非你确定需要，任何错误都应该有应对 • 当存在多种可能，不要尝试去猜测 • 只要你不是Guido，对于问题尽量找一种，最好是唯一明显的解决方案 • 做也许好过不做，但不假思索就动手还不如不做 • 如果你无法向人描述你的实现方案，那肯定不是一个好方案 • 如果实现方案容易解释，可能是个好方案 • 命名空间是绝妙的理念，要多运用 	<p>以编写优美代码为目标，不多解释 优美代码应该清晰明了，规范统一 优美代码应该逻辑简洁，避免复杂逻辑 如果必须采用复杂逻辑，接口关系也要清晰 优美代码应该是扁平的，避免太多层次嵌套 优美代码间隔要适当，每行代码解决适度问题 优美代码必须是可读且易读的 上述规则是至高无上的</p> <p>捕获异常，不让程序留有因错误退出的可能</p> <p>不要试图给出多种方案，找到一种实现它，几乎所有人没有Guido那么牛 编程之前要有思考</p> <p>能说清楚的往往才是对的</p> <p>适合复杂程序编程</p>																	

课时分配	教学内容	方法及手段
5M	<p>就像文学作品，不仅达意，更要优美。那什么是“好”？什么是“优美”？领悟编程代码优美的过程类似参禅，除了不断练习，也需要理解一些原则。</p> <p>除了 Python 之禅所表达的 Python 设计理念，该程序还有另一段魅力： 密文：A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 原文：N O P Q R S T U V W X Y Z A B C D E F G H I J K L M 密文：a b c d e f g h i j k l m n o p q r s t u v w x y z 原文：n o p q r s t u v w x y z a b c d e f g h i j k l m</p> <p>这个算法可以看作是凯撒密码的一种扩展，相比凯撒密码，采用循环移动 13 个位置，加密和解密可以用同一个程序。</p> <p>四、总结： 学习完本课内容，能够掌握组合数据类型中字典等类型及基本操作，讲解了如何使用 jieba 词库对中文文档进行分词并进一步统计文档词频，最后利用 Python 之禅的例子介绍了编写好代码的基本原则。</p>	总结
5M	<p>★课程建设：</p> <p>在学习中文文本分析和 Python 之禅时，鼓励学生理解信息处理和价值观的关系。信息分析不仅帮助我们获取有用的知识，还需要考虑信息的来源、准确性和隐私。此外，Python 之禅的核心原则教导学生编写更好的代码，强调代码的可读性和可维护性，以更好地服务社会 and 他人。这有助于培养学生的社会责任感和价值观。</p>	思政建设

L11:

授课时间	2022. 11. 15	课次	15
授课方式 (请打√)	理论课 <input checked="" type="checkbox"/> 讨论课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其他 <input type="checkbox"/>	课时 安排	2
授课题目 (教学章、节或主题): 第 7 章 文件和数据格式化 7.1 文件的使用 7.2 PIL 库的使用			
教学目的与要求: (1)掌握文件的读写方法以及打开和关闭等基本操作。 (2)理解数据组织的维度及其特点。 (3)运用 PIL 库进行基本的图像处理。			
教学重点及难点: 重点: 文件的操作, 包括文件的打开、读写和关闭。PIL 库的基本用法, 包括图像的加载、处理和保存。 难点: 学生需要多练习来理解文件的操作方法, 尤其是文件的打开和关闭。 PIL 库的使用可能需要一些实践和图像处理的知识。			
作业、讨论题、思考题: 作业: 课后习题 7.1 讨论题: 讨论文件的重要性和用途, 如何使用文件进行数据存储和共享。 思考题: 为什么文件操作对于数据存储和共享非常重要?			
课后小结: 本章主要介绍了文件的输入输出操作框架, 介绍了 PIL 库并使用 PL 库演示了字符画绘制实例。			
下节课预习重点: 第 7 章 文件和数据格式化			
参考文献: [1] 嵩天、礼欣、黄天羽. 《Python 语言程序设计基础(第 2 版)》. 高等教育出版社. 2017			

课时分配	教学内容	方法及手段
5M	<p>一、知识回顾：</p> <p>回顾组合数据类型中字典等类型及基本操作，jieba 词库对中文文档进行分词。</p>	提问巩固
5M	<p>二、课堂引入：</p> <p>▲讨论引入：</p> <p>回顾处理小说的案例，数据源读入，中文文本分词，并进行词频分析需要用到的知识点内容。</p> <p>本章重点掌握：</p> <p>文件的使用：文件打开、关闭</p> <p>文件内容读、写</p> <p>一维数据的处理：表示、存储和处理</p> <p>二维数据的处理：表示、存储和处理</p> <p>采用 CSV 格式对一二维数据文件的读写</p>	讨论发言
5M	<p>■板书内容：</p> <div data-bbox="245 1182 1316 1805" data-label="Diagram"> </div>	讲解板书
10M	<p>三、授课内容：</p> <p>7.1 文件的使用</p> <p>■ 文本文件一般由单一特定编码的字符组成，如 UTF-8 编码，内容容易统一展示和阅读。</p>	讲解及演示

课时分配	教学内容	方法及手段
15M	<p> ■ 二进制文件直接由比特 0 和比特 1 组成，文件内部数据的组织格式与文件用途有关。二进制是信息按照非字符但特定格式形成的文件，例如，png 格式的图片文件、avi 格式的视频文件。 </p> <p> ■ 二进制文件和文本文件最主要的区别在于是否有统一的字符编码。 </p> <p> ■ 无论文件创建为文本文件或者二进制文件，都可以用“文本文件方式”和“二进制文件方式”打开，但打开后的操作不同。 </p> <p>微实例 7.1：理解文本文件和二进制文件的区别。</p> <div style="display: flex; justify-content: space-around;"> <pre style="background-color: #f0f0f0; padding: 5px;"> 1 f = open("a.txt", "rt") 2 print(f.readline()) 3 f.close() </pre> <pre style="background-color: #f0f0f0; padding: 5px;"> 1 f = open("a.txt", "rb") 2 print(f.readline()) 3 f.close() </pre> </div> <p style="font-size: small; margin-top: 5px;"> 全国计算机等级考试 b'\xc8\xab\xb9\xfa\xbc\xc6\x \xb5\xc8\xbc\xb6\xbf\xbc\xc </p> <div style="border: 1px solid black; padding: 10px; margin-top: 10px;"> <p>文件的打开关闭</p> <p>Python 对文本文件和二进制文件采用统一的操作步骤，即“打开-操作-关闭”。</p> <div style="display: flex; align-items: center; justify-content: center;">  </div> <p>Python 通过解释器内置的 open() 函数打开一个文件，并实现该文件与一个程序变量的关联，open() 函数格式如下：</p> <p><变量名> = open(<文件名>, <打开模式>)</p> <ul style="list-style-type: none"> ① 第一个参数file: 需要打开的文件名。 <ul style="list-style-type: none"> • 如果文件在项目路径下，用相对路径; • 如果文件不在项目路径下，需要写文件的绝对路径 ② 第二个参数mode: 文件的操作方式。 <ul style="list-style-type: none"> • 主要有 w, r, a, wb, rb。 ③ 第三个参数encoding: 文件的编码方式。 </div>	讲解及演示

课时分配	教学内容	方法及手段																										
10M	<p>open() 函数提供 7 种基本的打开模式：</p> <table border="1" data-bbox="261 389 1157 763"> <thead> <tr> <th>打开模式</th> <th>含义</th> </tr> </thead> <tbody> <tr> <td>'r'</td> <td>只读模式，如果文件不存在，返回异常FileNotFoundError，默认值</td> </tr> <tr> <td>'w'</td> <td>覆盖写模式，文件不存在则创建，存在则完全覆盖源文件</td> </tr> <tr> <td>'x'</td> <td>创建写模式，文件不存在则创建，存在则返回异常FileExistsError</td> </tr> <tr> <td>'a'</td> <td>追加写模式，文件不存在则创建，存在则在原文件最后追加内容</td> </tr> <tr> <td>'b'</td> <td>二进制文件模式</td> </tr> <tr> <td>'t'</td> <td>文本文件模式，默认值</td> </tr> <tr> <td>'+'</td> <td>与r/w/x/a一同使用，在原功能基础上增加同时读写功能</td> </tr> </tbody> </table> <p>根据打开方式不同可以对文件进行相应的读写操作，Python 提供 4 个常用的文件内容读取方法：</p> <table border="1" data-bbox="261 913 1181 1223"> <thead> <tr> <th>方法</th> <th>含义</th> </tr> </thead> <tbody> <tr> <td><file>.readall()</td> <td>读入整个文件内容，返回一个字符串或字节流*</td> </tr> <tr> <td><file>.read(size=-1)</td> <td>从文件中读入整个文件内容，如果给出参数，读入前size长度的字符串或字节流</td> </tr> <tr> <td><file>.readline(size = -1)</td> <td>从文件中读入一行内容，如果给出参数，读入该行前size长度的字符串或字节流</td> </tr> <tr> <td><file>.readlines(hint=-1)</td> <td>从文件中读入所有行，以每行为元素形成一个列表，如果给出参数，读入hint行</td> </tr> </tbody> </table> <p><file>.readall() 最新版本中取消。</p> <p>微实例 7.2：文本文件逐行打印</p> <p>如果程序需要逐行处理文件内容，建议采用上述代码格式：</p> <pre>fo = open(fname, "r") for line in fo: # 处理一行数据 fo.close()</pre>	打开模式	含义	'r'	只读模式，如果文件不存在，返回异常FileNotFoundError，默认值	'w'	覆盖写模式，文件不存在则创建，存在则完全覆盖源文件	'x'	创建写模式，文件不存在则创建，存在则返回异常FileExistsError	'a'	追加写模式，文件不存在则创建，存在则在原文件最后追加内容	'b'	二进制文件模式	't'	文本文件模式，默认值	'+'	与r/w/x/a一同使用，在原功能基础上增加同时读写功能	方法	含义	<file>.readall()	读入整个文件内容，返回一个字符串或字节流*	<file>.read(size=-1)	从文件中读入整个文件内容，如果给出参数，读入前size长度的字符串或字节流	<file>.readline(size = -1)	从文件中读入一行内容，如果给出参数，读入该行前size长度的字符串或字节流	<file>.readlines(hint=-1)	从文件中读入所有行，以每行为元素形成一个列表，如果给出参数，读入hint行	讲解及演示
打开模式	含义																											
'r'	只读模式，如果文件不存在，返回异常FileNotFoundError，默认值																											
'w'	覆盖写模式，文件不存在则创建，存在则完全覆盖源文件																											
'x'	创建写模式，文件不存在则创建，存在则返回异常FileExistsError																											
'a'	追加写模式，文件不存在则创建，存在则在原文件最后追加内容																											
'b'	二进制文件模式																											
't'	文本文件模式，默认值																											
'+'	与r/w/x/a一同使用，在原功能基础上增加同时读写功能																											
方法	含义																											
<file>.readall()	读入整个文件内容，返回一个字符串或字节流*																											
<file>.read(size=-1)	从文件中读入整个文件内容，如果给出参数，读入前size长度的字符串或字节流																											
<file>.readline(size = -1)	从文件中读入一行内容，如果给出参数，读入该行前size长度的字符串或字节流																											
<file>.readlines(hint=-1)	从文件中读入所有行，以每行为元素形成一个列表，如果给出参数，读入hint行																											
10M	<p>Python 提供 3 个与文件内容写入有关的方法，如表所示。</p> <table border="1" data-bbox="261 1823 1088 2020"> <thead> <tr> <th>方法</th> <th>含义</th> </tr> </thead> <tbody> <tr> <td><file>.write(s)</td> <td>向文件写入一个字符串或字节流</td> </tr> <tr> <td><file>.writelines(lines)</td> <td>将一个元素为字符串的列表写入文件</td> </tr> <tr> <td><file>.seek(offset)</td> <td>改变当前文件操作指针的位置，offset的值： 0：文件开头； 1:当前位置； 2:文件结尾</td> </tr> </tbody> </table>	方法	含义	<file>.write(s)	向文件写入一个字符串或字节流	<file>.writelines(lines)	将一个元素为字符串的列表写入文件	<file>.seek(offset)	改变当前文件操作指针的位置，offset的值： 0：文件开头； 1:当前位置； 2:文件结尾	讲解及演示																		
方法	含义																											
<file>.write(s)	向文件写入一个字符串或字节流																											
<file>.writelines(lines)	将一个元素为字符串的列表写入文件																											
<file>.seek(offset)	改变当前文件操作指针的位置，offset的值： 0：文件开头； 1:当前位置； 2:文件结尾																											

课时分配	教学内容	方法及手段
10M	<p>文件打开后，对文件的读写有一个读取指针，当从文件中读入内容后，读取指针将向前进，再次读取的内容将从指针的新位置开始。f.seek()方法能够移动读取指针的位置。</p> <pre data-bbox="256 499 858 891"> >>>ff = open("a.txt", "r", encoding="utf-8") >>>s = ff.read() >>>print(s) hello world 中国！北京！ >>>ff.seek(0) # 将读取指针重置到文件开头 >>>ls = ff.readlines() >>>print(ls) ['hello world\n' , '中国！北京！'] >>>ff.close() </pre> <p>文件使用结束后要用 close() 方法关闭，释放文件的使用授权，语法形式如下：</p> <p><变量名>.close()</p> <p>案例学习：人名统计</p> <pre data-bbox="288 1216 1042 1597"> # 文件读入全部人名内容 ff =open('数据/name.txt', 'r', encoding='utf-8') s = ff.read() s = s.split() name = set(s) n = len(name) print(n) </pre>	讲解及演示
10M	<p>7.2 PIL 库的使用</p> <p>PIL (Python Image Library) 库是 Python 语言的第三方库，需要通过 pip 工具安装。</p> <pre data-bbox="288 1776 1267 1823"> :>>pip install pillow # 或者 pip3 install pillow </pre> <p>PIL 库支持图像存储、显示和处理，它能够处理几乎所有图片格式，可以完成对图像的缩放、剪裁、叠加以及向图像添加线条、图像和文字等操作。</p> <p>PIL 库可以完成图像归档和图像处理两方面功能需求：</p>	讲解及演示

L12:

授课时间	2022. 11. 22	课次	15
授课方式 (请打√)	理论课 <input checked="" type="checkbox"/> 讨论课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其他 <input type="checkbox"/>	课时 安排	2
授课题目 (教学章、节或主题): 第 7 章 文件和数据格式化 7.3 一二维数据的格式化和处理 7.4 高维数据的格式化 7.5 CSV 和 JSON 格式转换			
教学目的与要求: (1)掌握一二维数据的存储格式和读写方法。 (2)了解 JSON 库进行数据的维度转换。 (3)了解高维数据的存储格式和读写方法。			
教学重点及难点: 重点: 一维和二维数据的格式化和处理方法。高维数据的格式化和处理。CSV 和 JSON 格式的特点和使用。 难点: 高维数据的管理和访问方法。数据格式转换可能需要一些实践。			
作业、讨论题、思考题: 作业: 课后习题 7.2。 讨论题: 讨论数据格式化和处理的重要性,如何提高数据的可读性和可分析性。 思考题: 为什么数据格式化和处理对于数据分析和应用非常关键?			
课后小结: 通过这次课的学习,介绍了数据的维度概念和多维数据的格式化处理方法,演示了 JSON 和 CSV 格式相互转化以及 CSV 的 HTML 格式展示等方法。			
下节课预习重点: 第 8 章 程序设计方法			
参考文献: [1] 嵩天、礼欣、黄天羽.《Python 语言程序设计基础(第 2 版)》. 高等教育出版社. 2017			

课时分配	教学内容	方法及手段																								
5M	<p>一、知识回顾：</p> <p>(1) 掌握文件的读写方法以及打开和关闭等基本操作。</p> <p>(2) 理解数据组织的维度及其特点。</p> <p>(3) 运用 PIL 库进行基本的图像处理。</p>	提问巩固																								
5M	<p>二、课堂引入：</p> <p>▲案例引入：</p> <p>案例：学生成绩统计与分析</p> <p>描述：你是一名老师，需要统计学生成绩并进行分析，以更好地了解班级的表现。你希望使用 Python 编写程序，自动计算平均分、最高分、最低分等统计信息。此外，你还想将数据保存到文件中，以备后续分析。这需要格式化和处理数据以及使用 CSV 和 JSON 格式。</p> <ul style="list-style-type: none"> 引导学生思考如何手工统计学生成绩，以及如何将这个过程转化为可重复使用的代码。 引出数据格式化和文件格式转换的概念。 <p>讲解方法：展示学生成绩统计与分析的需求，然后引入 Python 编程的方法，解释如何格式化和处理数据，以及如何使用 CSV 和 JSON 格式保存和读取数据。</p>	案例引入																								
15M	<p>三、授课内容：</p> <p>7.3 一二维数据的格式化和处理</p> <p>数据组织的维度</p> <p>一维数据由对等关系的有序或无序数据构成，采用线性方式组织，对应于数学中的数组和集合等概念。</p> <div data-bbox="261 1435 1015 1525" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>北京、上海、天津、重庆</p> </div> <p>二维数据，也称表格数据，由关联关系数据构成，采用表格方式组织，对应于数学中的矩阵，常见的表格都属于二维数据。</p> <table border="1" data-bbox="341 1671 1295 2018" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th>城市</th> <th>环比</th> <th>同比</th> <th>定基</th> </tr> </thead> <tbody> <tr> <td>北京</td> <td>101.5</td> <td>120.7</td> <td>121.4</td> </tr> <tr> <td>上海</td> <td>101.2</td> <td>127.3</td> <td>127.8</td> </tr> <tr> <td>广州</td> <td>101.3</td> <td>119.4</td> <td>120.0</td> </tr> <tr> <td>深圳</td> <td>102.0</td> <td>140.9</td> <td>145.5</td> </tr> <tr> <td>沈阳</td> <td>100.1</td> <td>101.4</td> <td>101.6</td> </tr> </tbody> </table>	城市	环比	同比	定基	北京	101.5	120.7	121.4	上海	101.2	127.3	127.8	广州	101.3	119.4	120.0	深圳	102.0	140.9	145.5	沈阳	100.1	101.4	101.6	讲解
城市	环比	同比	定基																							
北京	101.5	120.7	121.4																							
上海	101.2	127.3	127.8																							
广州	101.3	119.4	120.0																							
深圳	102.0	140.9	145.5																							
沈阳	100.1	101.4	101.6																							

课时分配	教学内容	方法及手段
10M	<p>高维数据由键值对类型的数据构成，采用对象方式组织，属于整合度更好的数据组织方式。高维数据在网络系统中十分常用，HTML、XML、JSON等都是高维数据组织的语法结构。</p> <pre> "本书作者" : [{ "姓氏" : "嵩", "名字" : "天", "单位" : "北京理工大学" }, { "姓氏" : "礼", "名字" : "欣", "单位" : "北京理工大学" }, { "姓氏" : "黄", "名字" : "天羽", "单位" : "北京理工大学" }] </pre> <p>■ 板书内容：</p>	
15M	<p>一二维数据的存储格式</p> <p>逗号分割数值的存储格式叫做 CSV 格式（Comma-Separated Values，即逗号分隔值），它是一种通用的、相对简单的文件格式，在商业和科学上广泛应用，尤其应用在程序之间转移表格数据。</p> <p>该格式的应用有一些基本规则，如下：</p> <ol style="list-style-type: none"> (1) 纯文本格式，通过单一编码表示字符； (2) 以行为单位，开头不留空行，行之间没有空行； (3) 每行表示一个一维数据，多行表示二维数据； (4) 以逗号分隔每列数据，列数据为空也要保留逗号； (5) 可以包含或不包含列名，包含时列名放置在文件第一行。 <p>一维数据</p> <p>表示形式：列表</p> <p>存储形式：空格、逗号、换行、特殊符号</p>	<p>板书、讲解及演示</p> <p>讲解及演示</p>

课时分配	教学内容	方法及手段
10M	<p>常见转换操作：</p> <p>列表 -> 写入 CSV 存储</p> <p>读取 CSV -> 列表 处理</p> <p>一维数据的表示</p> <p>一维数据是最简单的数据组织类型，由于是线性结构，在 Python 语言中主要采用列表形式表示。</p> <p>例如：中国的直辖市数据可以采用一个列表变量表示。</p> <p>一维数据的存储</p> <p>一维数据是最简单的数据组织类型，有多种存储格式，常用特殊字符分隔：</p> <p>(1) 用一个或多个空格分隔，例如：</p> <p>中国 美国 日本 德国 法国 英国 意大利</p> <p>(2) 用逗号分隔，例如：</p> <p>中国,美国,日本,德国,法国,英国,意大利</p> <p>(3) 用其他符号或符号组合分隔，建议采用不出现在数据中的特殊符号</p> <p>中国；美国；日本；德国；法国；英国；意大利</p> <p>逗号分割的存储格式叫做 CSV 格式 (Comma-Separated Values, 即逗号分隔值)。</p> <p>它是一种通用的、相对简单的文件格式，在商业和科学上广泛应用。</p> <p>CSV 格式存储的文件一般采用.csv 为扩展名，可以通过文本编辑工具如记事本或 Excel 表格工具打开。</p> <p>一维数据保存成 CSV 格式后，各元素采用逗号分隔，形成一行。</p> <p>一维列表 -> 存储为 CSV 文件</p> <p>列表对象输出为 CSV 格式文件方法如下，采用字符串的 join() 方法最为方便。</p> <pre>ls = ['北京', '上海', '天津', '重庆'] f = open("city.csv", "w") f.write("," . join(ls) + "\n") f.close()</pre> <p>读取 CSV -> 一维列表</p>	讲解及演示

课时分配	教学内容	方法及手段
10M	<p>对一维数据进行处理首先需要从 CSV 格式文件读入一维数据，并将其表示为列表对象。</p> <pre>f = open("city.csv", "r") ls = f.read().strip('\n').split(",") f.close() print(ls)</pre> <p>■ 板书内容：</p> <p>二维数据的处理</p> <ul style="list-style-type: none"> ■ 二维数据由多条一维数据构成，可以看成是一维数据的组合形式。 ■ 因此，二维数据可以采用二维列表来表示。 ■ 列表的每个元素对应二维数据的一行，这个元素本身也是列表类型，其内部各元素对应这行中的各列值。 <p>二维数据 表示形式：二维列表 存储形式：CSV 文件的每一行是一维数据 常见转换操作： 二维列表 -> 写入 CSV 存储 CSV 文件的每一行是一维数据。 采用遍历循环和字符串的 join() 方法相结合。</p>	讲解及演示
10M	<p>二维数据的处理</p> <ul style="list-style-type: none"> ■ 二维数据处理等同于二维列表的操作，一般需要借助循环遍历实现对每个数据的处理，格式如下： <pre>for row in ls : for item in row : <对第 row 行第 item 列元素进行处理></pre> <p>7.4 高维数据的格式化</p> <p>与一维二维数据不同，高维数据能展示数据间更为复杂的组织关系。为了保持灵活性，表示高维数据不采用任何结构形式，仅采用最基本的二元关系，</p>	讲解及演示

L13:

授课时间	2022. 11. 29	课次	15	
授课方式 (请打√)	理论课 <input checked="" type="checkbox"/> 讨论课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其他 <input type="checkbox"/>		课时 安排	2
授课题目 (教学章、节或主题): 第 8 章 程序设计方法 8.1 计算思维 8.2 实例体育竞技分析 8.3 自顶向下和自底向上 8.4 pyinstaller 库的使用 8.5 计算生态和模块编程 8.6 Python 第三方库的安装				
教学目的与要求: (1)了解计算思维的概念。 (2)掌握自顶向下的设计方法。 (3)掌握自底向上的执行过程。 (4)了解计算生态和模块编程思想。 (5)掌握 Python 第三方库的安装方法。 (6)掌握 Python 源文件的打包方法。				
教学重点及难点: 重点: 计算思维的概念和应用。自顶向下和自底向上的程序设计方法。 难点: 理解计算思维的应用和程序设计方法。使用 pyinstaller 库打包程序和安装第三方库可能需要一些技术细节。				
作业、讨论题、思考题: 作业: 课后习题 8.4。 讨论题: 探讨自顶向下和自底向上的程序设计方法, 如何提高程序设计的效率和可维护性。 思考题: 为什么计算思维对于解决问题和分析数据非常重要?				
课后小结: 本章阐述了计算思维的概念, 以体育竞技分析为例介绍了自顶向下的设计方法自底向上的测试方法。进一步阐述了利用 Python 第三方库编程的模块编程思想和算生态的理解和运用。				
下节课预习重点: 第 9 章 科学计算和可视化				
参考文献: [1] 嵩天、礼欣、黄天羽. 《Python 语言程序设计基础(第 2 版)》. 高等教育出版社. 2017				

课时分配	教学内容	方法及手段
5M	<p>一、知识回顾： 软件开发过程，如何提高效率？</p>	提问巩固
5M	<p>二、课堂引入： ▲案例引入： 案例：体育竞技分析 描述：假设你是一名体育分析师，负责分析篮球比赛数据以预测比赛结果。你需要编写一个 Python 程序，将比赛数据输入到程序中，分析球员的表现，并提供比赛结果的预测。</p> <ul style="list-style-type: none"> • 引导学生思考如何使用计算思维来解决这个问题。 • 引出计算思维的概念和应用。 • 讲解方法：展示分析篮球比赛数据的步骤，然后引入 Python 编程的方法，解释如何使用计算思维来分析数据和预测结果，以及如何使用程序设计方法解决这个问题。 	案例引入
15M	<p>三、授课内容： 8.1 计算思维 计算思维是人类科学思维活动的重要组成部分。人类在认识世界、改造世界过程中表现出三种基本的思维特征： 以实验和验证为特征的实证思维，以物理学科为代表； 以推理和演绎为特征的逻辑思维，以数学学科为代表； 以设计和构造为特征的计算思维，以计算机学科为代表。 计算思维的本质是抽象（Abstraction）和自动化（Automation）。</p> <p>■板书内容： 8.2 实例体育竞技分析</p> <div style="border: 1px solid black; padding: 5px;"> <p>模拟（simulation）是抽象原系统某些行为特征并用另一系统来表示这些特征的过程，通常用于设计初期的模型验证。</p> <p>仿真（emulation）则更进一步，需要模仿系统真实能做的事情，接收同样的数据，获得同样的结果，只不过实现的“过程”不同。仿真一般用于处理兼容性问题或在资源有限条件下实现系统原型。</p> </div>	讲解及板书

课时分配	教学内容	方法及手段
15M	<p>该问题的 IPO 模式如下： 输入：两个球员（球员 A 和 B）的能力概率，模拟比赛的场次； 处理：模拟比赛过程； 输出：球员 A 和 B 分别赢得球赛的概率</p> <p>8.3 自顶向下和自底向上</p> <p>程序需要采用自顶向下设计方法，采用自底向上的执行过程。</p> <p>自顶向下的设计方法</p> <p>以一个总问题开始，试图把它表达为很多小问题组成的解决方案。再用同样的技术依次攻破每个小问题，最终问题变得非常小，以至于可以很容易解决。然后只需把所有的碎片组合起来，就可以得到一个程序。</p> <p>顶层设计</p> <p>自顶向下设计中最重要的是顶层设计。</p> <p>以体育竞技分析为例，可以从问题的 IPO 描述开始。大多数程序都可以简单将 IPO 描述直接用到程序结构设计中，体育竞技分析从用户得到模拟参数模拟比赛，最后输出结果。</p> <p>第 n 层设计</p> <p>每层设计中，参数和返回值如何设计是重点，其他细节部分可以暂时忽略。确定事件的重要特征而忽略其它细节过程称为抽象。</p> <p>自顶向下设计的第二阶段是实现或进一步抽象第 2 层函数。</p> <div style="border: 1px solid black; padding: 5px;"> <p>设计过程总结</p> <p>步骤 1：将算法表达为一系列小问题；</p> <p>步骤 2：为每个小问题设计接口；</p> <p>步骤 3：通过将算法表达为接口关联的多个小问题来细化算法；</p> <p>步骤 4：为每个小问题重复上述过程。</p> </div> <p>自底向上执行</p> <p>开展测试的更好办法也是将程序分成小部分逐个测试</p> <p>可以从 gameOver() 函数开始测试。Python 解释器提供 import 保留字辅助开展单元测试，语法格式如下：</p> <pre>import <源文件名称></pre>	讲解及板书

课时分配	教学内容	方法及手段
10M	<p>通过继续进行这样的单元测试可以检测程序中的每个函数。独立检验每个函数更容易发现错误。通过模块化设计可以分解问题使编写复杂程序成为可能，通过单元测试方法分解问题使运行和调试复杂程序成为可能。自顶向下和自底向上贯穿程序设计和执行的整个过程。</p> <p>8.4 pyinstaller 库的使用</p> <p>pyinstaller 是将 Python 语言脚本 (.py 文件) 打包成 Windows、Linux、Mac OS X 等操作系统下可执行文件的第三方库。</p> <p>PyInstaller 是一个十分有用的第三方库，它能够在 Windows、Linux、Mac OS X 等操作系统下将 Python 源文件打包，通过对源文件打包，Python 程序可以在没有安装 Python 的环境中运行，也可以作为一个独立文件方便传递和管理。</p>	讲解及演示
10M	<p>8.5 计算生态和模块编程</p> <p>Python 语言有 9 万多个第三方库，形成了庞大的计算生态，模块编程思想是 Python 语言最大的价值。</p> <p>第三方库</p> <p>Python 内置的库称为标准库，其他库称为第三方库。</p> <p>在计算生态思想指导下，编写程序的起点不再是探究每个具体算法的逻辑功能和设计，而是尽可能利用第三方库进行代码复用，探究运用库的系统方法。</p> <p>模块编程</p> <p>这种像搭积木一样的编程方式，称为“模块编程”。</p> <p>每个模块可能是标准库、第三方库、用户编写的其他程序或对程序运行有帮助的资源等。</p> <p>模块编程与模块化编程不同，模块化编程主张采用自顶向下设计思想，主要开展耦合度低的单一程序设计与开发，而模块编程主张利用开源代码和第三方库作为程序的部分或全部模块，搭积木一样编写程序。</p>	讲解
10M	<p>8.6 Python 第三方库的安装</p> <p>Python 第三方库依照安装方式灵活性和难易程度有三个方法，建议读者依次使用，能够将第三方库安装成功，这三个方法是：pip 工具安装、自定</p>	讲解及演示

课时分配	教学内容	方法及手段
10M	<p>义安装和文件安装。</p> <ul style="list-style-type: none"> ■ 最常用且最高效的 Python 第三方库安装方式是采用 pip 工具安装。 ■ pip 是 Python 官方提供并维护的在线第三方库安装工具。 ■ 对于同时安装 Python 2 和 Python 3 环境的系统，建议采用 pip3 命令专门为 Python 3 版本安装第三方库。 ■ pip 支持安装 (install)、下载 (download)、卸载 (uninstall)、列表 (list)、查看 (list)、查找 (search) 等一系列安装和维护子命令。 ■ 安装一个库的命令格式如下，例如： <pre>pip install <拟安装库名></pre> <p>自定义安装</p> <p>自定义安装指按照第三方库提供的步骤和方式安装。第三方库都有主页用于维护库的代码和文档。</p> <p>文件安装</p> <p>由于 Python 某些第三方库仅提供源代码，通过 pip 下载文件后无法在 Windows 系统编译安装，会导致第三方库安装失败。在 Windows 平台下遇到的无法安装第三方库的问题大多属于这类。</p> <p>为了解决这类第三方库安装问题，美国加州大学尔湾分校提供了一个页面，帮助 Python 用户获得 Windows 可直接安装的第三方库文件，链接地址如下：http://www.lfd.uci.edu/~gohlke/pythonlibs/。</p>	讲解及演示
5M	<p>四、总结：</p> <p>学习完本课内容，能够掌握计算思维的概念，自顶向下的设计方法自底向上的测试方法。利用 Python 第三方库编程的模块编程思想和算生态的理解和运用。</p>	总结
5M	<p>★课程思政建设：</p> <p>在本章的学习中，我们强调计算思维的培养和应用。计算思维是一种解决问题的能力，有助于学生分析复杂情境并提供解决方案。此外，本章还涉及自顶向下和自底向上的程序设计方法，培养学生的系统性思维和规划能力。</p>	思政建设

L14:

授课时间	2022. 12. 06	课次	15	
授课方式 (请打√)	理论课 <input checked="" type="checkbox"/> 讨论课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其他 <input type="checkbox"/>		课时 安排	2
授课题目 (教学章、节或主题): 第 9 章 科学计算和可视化 9.1 问题概述 9.2 numpy 库的使用 9.4 matplotlib 库的使用				
教学目的与要求: (1) 了解科学计算的基本概念。(2) 了解数据可视化的概念。 (3) 运用科学计算库进行矩阵分析和数值运算。 (4) 运用数据绘图库进行坐标系绘制。				
教学重点及难点: 重点: 运用科学计算库进行矩阵分析和数值运算。 难点: 使用 numpy 库进行数据处理和计算。使用 matplotlib 库进行数据可视化。				
作业、讨论题、思考题: 作业: 课后习题 9.2。 讨论题: 讨论科学计算在不同领域的应用, 如天文学、生物学等。 思考题: 如何选择合适的可视化方法, 以便更好地传达数据的含义和结果?				
课后小结: 本章以科学计算和可视化为中心, 本节重点掌握工具库 numpy 在科学计算方面的强大功能、工具库 matplotlib 在科学计算方面的强大功能。				
下节课预习重点: 第 9 章 科学计算和可视化				
参考文献: [1] 嵩天、礼欣、黄天羽. 《Python 语言程序设计基础(第 2 版)》. 高等教育出版社. 2017				

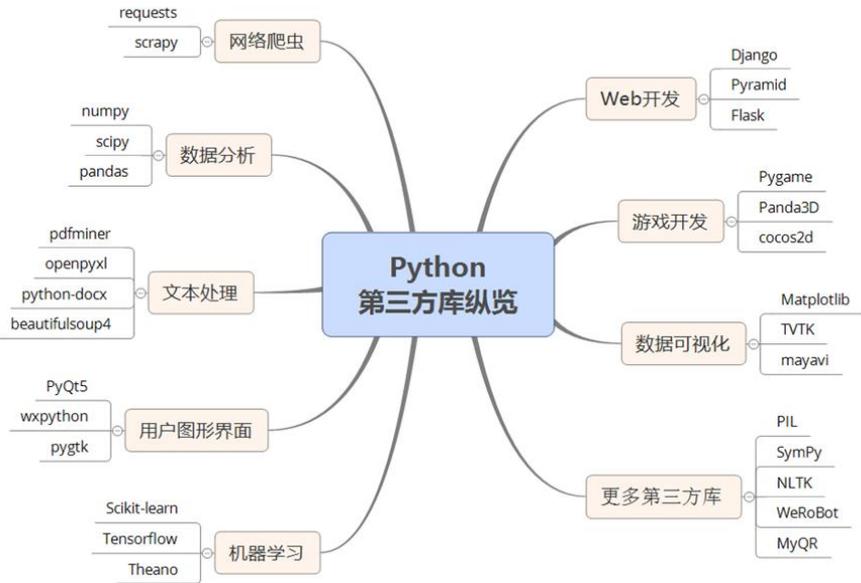
课时分配	教学内容	方法及手段
5M	<p>一、知识回顾：</p> <p>回顾自顶向下的程序设计思想,Python 第三方库编程的模块编程思想和算生态。</p>	提问巩固
5M	<p>二、课堂引入：</p> <p>▲讨论引入：</p> <p>讨论：探索宇宙中的谜题</p> <p>描述：在这个引入讨论中，引导学生思考宇宙中的一些谜题，如黑洞、宇宙起源、星系演化等。提出问题：科学家如何使用计算和可视化来解决这些谜题？</p> <p>讲解方法： 引入本章内容，说明计算和可视化在解决科学问题中的关键作用，以激发学生的兴趣。</p>	案例引入
10M	<p>三、授课内容：</p> <p>9.1 问题概述</p> <p>科学计算需要采用矩阵运算库 numpy 和绘制库 matplotlib</p> <p>科学计算是解决科学和工程中的数学问题利用计算机进行的数值计算，它不仅是科学家在运算自然规律时所使用的计算工具，更是普通人提升专业化程度的必要手段。</p> <p>Python 语言为开展人人都能使用的科学计算提供了有力支持。</p> <p>开展基本的科学计算需要两个步骤：组织数据和展示数据；</p> <p>组织数据是运算的基础，也是将客观世界数字化的必要手段；</p> <p>展示数据是体现运算结果的重要方式，也是展示结论的有力武器。</p>	讲解
10M	<p>9.2 numpy 库的使用</p> <p>numpy 是用于处理含有同种元素的多维数组运算的第三方库。</p> <p>Python 标准库中提供了一个 array 类型，用于保存数组类型数据，然而这个类型不支持多维数据，处理函数也不够丰富，不适合用于做数值运算。</p> <p>因此,Python 语言的第三方库 numpy 得到了迅速发展,至今,numpy 已经成为了科学计算事实上的标准库。</p> <p>numpy 库处理的最基础数据类型是由同种元素构成的多维数组</p>	讲解及演示

课时分配	教学内容	方法及手段																
10M	<p>(ndarray)，简称“数组”。</p> <p>数组中所有元素的类型必须相同，数组中元素可以用整数索引，序号从0开始。ndarray 类型的维度(dimensions)叫做轴(axes)，轴的个数叫做秩(rank)。一维数组的秩为1，二维数组的秩为2，二维数组相当于由两个一维数组构成。</p> <p>由于 numpy 库中函数较多且命名容易与常用命名混淆，建议采用如下方式引用 numpy 库：</p> <p>■ 板书内容：</p> <pre>>>>import numpy as np</pre> <table border="1" data-bbox="300 857 1201 1272"> <thead> <tr> <th>函数</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>np.array([x,y,z], dtype=int)</td> <td>从 Python 列表和元组创建数组</td> </tr> <tr> <td>np.arange(x,y,i)</td> <td>创建一个由 x 到 y，以 i 为步长的数组</td> </tr> <tr> <td>np.linspace(x,y,n)</td> <td>创建一个由 x 到 y，等分成 n 个元素的数组</td> </tr> <tr> <td>np.indices((m,n))</td> <td>创建一个 m 行 n 列的矩阵</td> </tr> <tr> <td>np.random.rand(m,n)</td> <td>创建一个 m 行 n 列的随机数组</td> </tr> <tr> <td>np.ones((m,n),dtype)</td> <td>创建一个 m 行 n 列全 1 的数组，dtype 是数据类型</td> </tr> <tr> <td>np.empty((m,n),dtype)</td> <td>创建一个 m 行 n 列全 0 的数组，dtype 是数据类型</td> </tr> </tbody> </table> <p>ndarray 类的常用属性</p> <p>ndarray 类的形态操作方法</p> <p>ndarray 类的索引和切片方法</p> <p>numpy 库的算术运算函数</p> <p>numpy 库的比较运算函数</p> <p>numpy 库的其他运算函数</p> <p>numpy 库还包括三角运算函数、傅里叶变换、随机和概率分布、基本数值统计、位运算、矩阵运算等非常丰富的功能，读者在使用时可以到官方网站查询。</p>	函数	描述	np.array([x,y,z], dtype=int)	从 Python 列表和元组创建数组	np.arange(x,y,i)	创建一个由 x 到 y，以 i 为步长的数组	np.linspace(x,y,n)	创建一个由 x 到 y，等分成 n 个元素的数组	np.indices((m,n))	创建一个 m 行 n 列的矩阵	np.random.rand(m,n)	创建一个 m 行 n 列的随机数组	np.ones((m,n),dtype)	创建一个 m 行 n 列全 1 的数组，dtype 是数据类型	np.empty((m,n),dtype)	创建一个 m 行 n 列全 0 的数组，dtype 是数据类型	<p>板书、讲解及演示</p>
函数	描述																	
np.array([x,y,z], dtype=int)	从 Python 列表和元组创建数组																	
np.arange(x,y,i)	创建一个由 x 到 y，以 i 为步长的数组																	
np.linspace(x,y,n)	创建一个由 x 到 y，等分成 n 个元素的数组																	
np.indices((m,n))	创建一个 m 行 n 列的矩阵																	
np.random.rand(m,n)	创建一个 m 行 n 列的随机数组																	
np.ones((m,n),dtype)	创建一个 m 行 n 列全 1 的数组，dtype 是数据类型																	
np.empty((m,n),dtype)	创建一个 m 行 n 列全 0 的数组，dtype 是数据类型																	
10M	<p>9.3 实例：图像的主成分分析</p> <p>这是一个使用 numpy 和 PIL 库提取图像特征形成手绘效果的实例。</p>	<p>讲解及演示</p>																

课时分配	教学内容	方法及手段
15M	<p>第 7 章使用 PIL 库获取了图像的轮廓，虽然提取了轮廓，但这个轮廓缺少立体感，视觉效果不够丰满</p> <p>图像的手绘效果</p> <ul style="list-style-type: none"> ■ 为了实现手绘风格，即黑白轮廓描绘，首先需要读取原图像的明暗变化，即灰度值。从直观视觉感受上定义，图像灰度值显著变化的地方就是梯度，它描述了图像灰度变化的强度。 ■ 通常可以使用梯度计算来提取图像轮廓，numpy 中提供了直接获取灰度图像梯度的函数 <code>gradient()</code>，传入图像数组表示即可返回代表 <code>x</code> 和 <code>y</code> 各自方向上梯度变化的二维元组。 ■ 具体来说，为了更好体现立体感，增加一个 <code>z</code> 方向梯度值，并给 <code>x</code> 和 <code>y</code> 方向梯度值赋权值 <code>depth</code>。这种坐标空间变化相当于给物体加上一个虚拟光源，根据灰度值大小模拟各部分相对于人视角的远近程度，使画面显得有“深度”。 <p>9.4 matplotlib 库的使用</p>	讲解及演示
15M	<div style="border: 1px solid black; padding: 5px;"> <p>matplotlib 是提供数据绘图功能的第三方库，其 <code>pyplot</code> 子库主要用于实现各种数据展示图形的绘制。</p> <p><code>matplotlib.pyplot</code> 是 <code>matplotlib</code> 的子库，引用方式如下：</p> <pre>>>>import matplotlib.pyplot as plt</pre> <p>上述语句与 <code>import matplotlib.pyplot</code> 一致，<code>as</code> 保留字与 <code>import</code> 一起使用能够改变后续代码中库的命名空间，有助于提高代码可读性。简单说，在后续程序中，<code>plt</code> 将代替 <code>matplotlib.pyplot</code>。</p> </div> <p>matplotlib.pyplot 库解析</p> <p><code>plt</code> 子库提供了一批操作和绘图函数，每个函数代表对图像进行的一个操作，比如创建绘图区域、添加标注或者修改坐标轴等。</p> <p>这些函数采用 <code>plt.()</code> 形式调用，其中 <code></code> 是具体函数名称。</p> <ul style="list-style-type: none"> ■ <code>plt</code> 库的绘图区域函数使用 <code>figure()</code> 函数创建一个全局绘图区域，并且使它成为当前的绘图对象，<code>figsize</code> 参数可以指定绘图区域的宽度和高度，单位为英寸。鉴于 <code>figure()</code> 函数参数较多，这里采用指定参数名称的方式输入参数。 	讲解及演示

L15:

授课时间	2022. 12. 13	课次	15	
授课方式 (请打√)	理论课 <input checked="" type="checkbox"/> 讨论课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其他 <input type="checkbox"/>		课时 安排	2
授课题目 (教学章、节或主题): 第 9 章 科学计算和可视化 9.5 实例: 科学坐标系绘制 PYTHON 计算生态				
教学目的与要求: (1) 运用数据绘图库进行坐标系绘制。 (2) 运用数据绘图库进行雷达图绘制。				
教学重点及难点: 重点: Python 计算生态的使用。 难点: 了解和选择合适的 Python 计算生态工具和库可能需要一定的技术背景。				
作业、讨论题、思考题: 作业: 课后习题 9.2。 讨论题: 讨论科学坐标系图的作用和好处, 如何帮助科学研究。 思考题: 如何设计有效的科学坐标系图, 以便更好地传达数据和结果?				
课后小结: 本节重点掌握 Python 计算生态强大功能。不限于以下领域: 网络爬虫、数据分析、文本处理、数据可视化、用户图形界面、机器学习、Web 开发、游戏开发等				
下节课预习重点: 第 10 章 网络爬虫和自动化				
参考文献: [1] 嵩天、礼欣、黄天羽.《Python 语言程序设计基础(第 2 版)》. 高等教育出版社. 2017				

课时分配	教学内容	方法及手段
5M	<p>一、知识回顾： 科学计算和可视化为中心，工具库 numpy 在科学计算方面的强大功能。</p>	提问巩固
10M	<p>二、课堂引入： ▲讨论引入： 讨论：科学坐标系绘制的实际应用。 描述：在这个讨论中，引导学生思考科学坐标系在不同领域中的实际应用，如天文学中的星图、物理学中的图表、地理学中的地图等。讨论如何使用 Python 进行科学坐标系的绘制和数据可视化。 教学讨论方法： 小组讨论。将学生分成小组，要求每个小组选择一个特定领域的实际应用，讨论如何使用科学坐标系绘制图表或地图，并分享他们的见解。</p>	讨论发言
5M	<p>★课程思政建设： 培养学生的科学思维和数据分析能力。学生将在本章学会如何使用计算工具和可视化来解决实际问题，同时强调科学家的创新思维和负责任的科学研究。</p>	思政建设
15M	<p>三、授课内容： 广泛的 Python 计算生态不限于以下领域：网络爬虫、数据分析、文本处理、数据可视化、用户图形界面、机器学习、Web 开发、游戏开发等。</p> 	讲解及演示

课时分配	教学内容	方法及手段
15M	<p>9.5 实例：科学坐标系绘制</p> <p>使用 matplotlib.pyplot 子库绘制科学坐标图并适当标注。</p> <p>采用坐标系绘制和展示数据趋势是经常使用的功能，掌握绘制专业的科学坐标系将会为生活和工作提供更高效的支持。</p> <p>科学坐标图绘制有 4 个要素：坐标轴、数据曲线、标题和图注。这个实例以阻尼衰减曲线绘制来具体阐述科学坐标系的绘制方法。</p> <p>为了便于阅读，程序设计了 Draw()、Shadow() 和 XY_Axis() 函数，分别用于绘制曲线、设置阴影和修改坐标轴。第 13 行 annotate() 函数配合箭头在曲线绘图界面添加动态注释，箭头的线条和尖端都有多种样式和参数，可以通过 arrowprops 和 connectionstyle 选择，具体请参考官方文档。</p> <p>plt.savefig() 函数能够将产生的坐标图保存为文件。</p>	讲解及演示
15M	<p>拓展：科学计算可视化</p> <p>可视化技术与科学计算相结合形成了可视化技术的一个重要分支——科学计算可视化(Visualization in Scientific Computing)。科学计算可视化将科学数据如测量获得的数值、图像或是计算产生的数字信息等以直观的、图形图像方式展示。通过直观展示，宇宙空间有了颜色、物理现象更为直观，感性理解和理性求证相辅相成共同促进科学计算的深入发展。</p>	讲解及演示
15M	<p>多级雷达图绘制：</p> <p>DOTA 人物能力值雷达图 霍兰德人格分析雷达图</p> <p>综合案例展示</p> <p>(同学推荐)</p>	案例代码展示
10M	<p>四、总结：</p> <p>学习完本课内容，能够体会 Python 语言在各领域方面的强大功能。将掌握如何使用 matplotlib 库创建科学坐标系图，并了解 Python 计算生态中的相关工具和库。他们将能够更好地呈现科学数据和研究结果，同时了解 Python 计算生态的多样性和应用。教案设计强调实际问题的引入，以激发学生的兴趣，同时强调思政建设内容，培养科学思维和数据分析能力。</p>	总结

L16:

授课时间	2022. 12. 20	课次	16
授课方式 (请打√)	理论课 <input checked="" type="checkbox"/> 讨论课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其他 <input type="checkbox"/>	课时 安排	2
授课题目 (教学章、节或主题): 第 10 章 网络爬虫和自动化 10.1 问题概述 10.2 request 库的使用 10.3 beautifulsoup 库的使用 10.4 实例: 中国大学排名爬虫 10.5 实例: 搜索关键词自动提交			
教学目的与要求: (1)掌握网络爬虫的基本方法。(2)运用 requests 库编写基本 URL 访问过程。 (3)运用 beautifulsoup4 库解析和处理 HTML。 (4)掌握向搜索引擎自动提交关键词并获取返回结果的方法。			
教学重点及难点: 重点: requests 库和 beautifulsoup4 库的使用。 难点: 处理网页数据和自动提交关键词可能具有一定的复杂性。			
作业、讨论题、思考题: 作业: 课后习题 10.1 讨论题: 探讨信息获取和处理的伦理和法律问题, 如数据隐私和知识产权。 思考题: 为什么网络爬虫和自动化在信息时代如此重要? 举例说明它们在不同领域的应用。			
课后小结: 本章主要介绍了设计并实现网络爬虫的基本方法, 结合 requests 和 beautifulsoup4 两个库的使用, 讲述了如何处理 HTTP 协议以及解析网页 HTML 和 XML 页面信息的方法。			
下节课预习重点: 预习后续课程内容: 数据采集与网络爬虫、数据可视化、大数据分析等。			
参考文献: [1] 嵩天、礼欣、黄天羽. 《Python 语言程序设计基础(第 2 版)》. 高等教育出版社. 2017			

课时分配	教学内容	方法及手段
5M	<p>一、知识回顾：</p> <p>回顾 python 库广泛应用领域</p>	提问巩固
5M	<p>二、课堂引入：</p> <p>▲讨论引入：</p> <p> 讨论：信息获取与自动化</p> <p> 描述：引导学生讨论信息获取的不同途径和方法，如手动查找、搜索引擎、自动化爬虫等。提出问题：在现代社会中，信息获取和处理的重要性如何增加？</p> <p> 教学讨论方法： 让学生分享信息获取的经验和方法，然后引入本章内容，强调网络爬虫和自动化在信息获取和处理中的作用。</p>	讨论发言
10M	<p>三、授课内容：</p> <p>10.1 问题概述</p> <p> 网络爬虫应用一般分为两个步骤：（1）通过网络连接获取网页内容（2）对获得的网页内容进行处理。</p> <p> 这两个步骤分别使用不同的函数库：requests 和 beautifulsoup4。</p> <p> 采用 pip 指令安装 requests 库，如果在 Python 2 和 Python 3 并存的系统中，采用 pip3 指令：</p> <p> :\>pip install requests # 或者 pip3 install requests。</p> <p> 采用 pip 或 pip3 指令安装 beautifulsoup4 库，注意，不要安装 beautifulsoup 库，后者由于年久失修，已经不再维护了</p> <p> :\>pip install beautifulsoup4</p> <p> # 或者 pip3 install beautifulsoup4</p> <p> 网页爬虫</p> <p> 使用 Python 语言实现网络爬虫和信息提交是非常简单的事情，代码行数很少，也无须知道网络通信等方面知识，非常适合非专业读者使用。然而，肆意的爬取网络数据并不是文明现象，通过程序自动提交内容争取竞争性资源也不公平。</p> <p> Robots 排除协议（Robots Exclusion Protocol），也被称为爬虫协议，</p>	讲解及演示

课时分配	教学内容	方法及手段										
10M	<p>它是网站管理者表达是否希望爬虫自动获取网络信息意愿的方法。管理者可以在网站根目录放置一个 robots.txt 文件，并在文件中列出哪些链接不允许爬虫爬取。绝大部分成熟的搜索引擎爬虫都会遵循这个协议，建议个人也能按照互联网规范要求合理使用爬虫技术。</p> <p>10.2 request 库的使用</p> <p>requests 库是一个简洁且简单的处理 HTTP 请求的第三方库。</p> <p>requests 的最大优点是程序编写过程更接近正常 URL 访问过程。</p> <p>这个库建立在 Python 语言的 urllib3 库基础上，类似这种在其他函数库之上再封装功能提供更友好函数的方式在 Python 语言中十分常见。在 Python 的生态圈里，任何人都有通过技术创新或体验创新发表意见和展示才华的机会。</p> <p>request 库支持非常丰富的链接访问功能，包括：国际域名和 URL 获取、HTTP 长连接和连接缓存、HTTP 会话和 Cookie 保持、浏览器使用风格的 SSL 验证、基本的摘要认证、有效的键值对 Cookie 记录、自动解压缩、自动内容解码、文件分块上传、HTTP(S)代理功能、连接超时处理、流数据下载等。</p> <p>get() 是获取网页最常用的方式，在调用 requests.get() 函数后，返回的网页内容会保存为一个 Response 对象，其中，get() 函数的参数 url 必须链接采用 HTTP 或 HTTPS 方式访问。</p>	讲解及演示										
10M	<p>■ 板书内容：</p> <p>网页请求函数</p> <p>和浏览器的交互过程一样，requests.get() 代表请求过程，它返回的 Response 对象代表响应。返回内容作为一个对象更便于操作，Response 对象的属性如下表所示，需要采用 <a>. 形式使用。</p> <p>Response 对象的属性</p> <table border="1" data-bbox="288 1731 1206 1989"> <thead> <tr> <th>属性</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>status_code</td> <td>HTTP 请求的返回状态，整数，200 表示连接成功，404 表示失败</td> </tr> <tr> <td>text</td> <td>HTTP 响应内容的字符串形式，即，也是 url 对应的页面内容</td> </tr> <tr> <td>encoding</td> <td>HTTP 响应内容的编码方式</td> </tr> <tr> <td>content</td> <td>HTTP 响应内容的二进制形式</td> </tr> </tbody> </table>	属性	描述	status_code	HTTP 请求的返回状态，整数，200 表示连接成功，404 表示失败	text	HTTP 响应内容的字符串形式，即，也是 url 对应的页面内容	encoding	HTTP 响应内容的编码方式	content	HTTP 响应内容的二进制形式	板书、讲解及演示
属性	描述											
status_code	HTTP 请求的返回状态，整数，200 表示连接成功，404 表示失败											
text	HTTP 响应内容的字符串形式，即，也是 url 对应的页面内容											
encoding	HTTP 响应内容的编码方式											
content	HTTP 响应内容的二进制形式											

课时分配	教学内容	方法及手段
10M	<p>Response 对象的方法</p> <p>json()方法能够在 HTTP 响应内容中解析存在的 JSON 数据，这将带来解析 HTTP 的便利。</p> <p>raise_for_status()方法能在非成功响应后产生异常，即只要返回的请求状态 status_code 不是 200，这个方法会产生一个异常，用于 try...except 语句。使用异常处理语句可以避免设置一堆复杂的 if 语句，只需要在收到响应调用这个方法，就可以避开状态字 200 以外的各种意外情况。</p> <p>requests 会产生几种常用异常。当遇到网络问题时，如：DNS 查询失败、拒绝连接等，requests 会抛出 ConnectionError 异常；遇到无效 HTTP 响应时，requests 则会抛出 HTTPError 异常；若请求 url 超时，则抛出 Timeout 异常；若请求超过了设定的最大重定向次数，则会抛出一个 TooManyRedirects 异常。</p> <p>拓展：HTTP 的 GET 和 POST</p> <p>HTTP 协议定义了客户端与服务器交互的不同方法，最基本的方法是 GET 和 POST。顾名思义，GET 可以根据某链接获得内容，POST 用于发送内容。然而，GET 也可以向链接提交内容。</p> <p>1) GET 方式可以通过 URL 提交数据，待提交数据是 URL 的一部分；采用 POST 方式，待提交数据放置在 HTML HEADER 内；</p> <p>2) GET 方式提交的数据最多不超过 1024 字节，POST 没有对提交内容的长度限制。</p> <p>3) 安全性问题。如(1)所述，使用 GET 时参数会显示在 URL 中，而 POST 不会。所以，如果这些数据是非敏感数据，那么使用 GET；如果提交数据是敏感数据，建议采用 POST 方式。</p>	综合对比
10M	<p>10.3 beautifulsoup 库的使用</p> <p>beautifulsoup4 库是一个解析和处理 HTML 和 XML 的第三方库。</p> <p>使用 requests 库获取 HTML 页面并将其转换成字符串后，需要进一步解析 HTML 页面格式，提取有用信息，这需要处理 HTML 和 XML 的函数库。</p> <p>beautifulsoup4 库，也称为 Beautiful Soup 库或 bs4 库，用于解析和处理 HTML 和 XML。</p>	板书、讲解及演示

课时分配	教学内容	方法及手段
10M	<p>需要注意,它不是 BeautifulSoup 库。它的最大优点是能根据 HTML 和 XML 语法建立解析树,进而高效解析其中的内容。</p> <p>beautifulsoup4 库采用面向对象思想实现,简单说,它把每个页面当做一个对象,通过<a>.的方式调用对象的属性(即包含的内容),或者通过<a>.()的方式调用方法(即处理函数)。</p> <p>■ 板书内容:</p> <div style="border: 1px solid black; padding: 5px;"> <p>在使用 beautifulsoup4 库之前,需要进行引用,由于这个库的名字非常特殊且采用面向对象方式组织,可以用 from...import 方式从库中直接引用 BeautifulSoup 类,方法如下。</p> <pre>>>>from bs4 import BeautifulSoup</pre> <p>beautifulsoup4 库中最主要的是 BeautifulSoup 类,每个实例化的对象相当于一个页面。采用 from...import 导入库中类后,使用 BeautifulSoup() 创建一个 BeautifulSoup 对象。</p> <p>每一个 Tag 标签在 beautifulsoup4 库中也是一个对象,称为 Tag 对象。上例中, title 是一个标签对象。每个标签对象在 HTML 中都有类似的结构:</p> <pre>糯米</pre> </div> <p>拓展: 正则表达式</p> <ul style="list-style-type: none"> ■ 正则表达式是表达和操作字符串的一种逻辑表达,一般在计算机编译器中使用。Python 语言采用正则表达式辅助字符串查找。正则表达式是一种规则,只要字符串符合这个规则,就算作匹配。例如,通过 re.compile() 函数注册一个正则表达式'jquery',则所有包含表达式的字符串都与它匹配。除了字符串,正则表达式还可以通过*+{}等符号扩展功能。有兴趣的读者可以查阅资料了解 Python 中正则表达式函数库 re 的更多高级使用。 ■ 除了 find_all() 方法,BeautifulSoup 类还提供一个 find() 方法,它们的区别只是前者返回全部结果而后者返回找到的第一个结果,find_all() 函数由于可能返回更多结果,所以采用列表形式; find() 函数返回字符串形式。 ■ 处理网页需要对 HTML 有一定的理解,然而实现爬虫就不算复杂,这 	讲解及演示

课时分配	教学内容	方法及手段
10M	<p>里仅介绍 beautifulsoup4 库中与爬虫相关的一些属性和操作。beautifulsoup4 库是一个非常完备且活跃的 HTML 解析函数库,它还可以完成更多复杂操作。深入使用请阅读 beautifulsoup4 库官方网站上提供的使用文档。</p> <p>10.4 实例：中国大学排名爬虫</p> <p>采用了 requests 和 beautifulsoup4 函数库。大学排名爬虫的构建需要三个重要步骤：</p> <p>第一，从网络上获取网页内容；</p> <p>第二，分析网页内容并提取有用数据到恰当的数据结构中；</p> <p>第三，利用数据结构展示或进一步处理数据。</p> <p>由于大学排名是一个典型的二维数据，因此，采用二维列表存储该排名所涉及的表单数据。</p> <p>10.5 实例：搜索关键词自动提交</p> <p>本节以百度为例介绍搜索关键字自动提交并获得返回结果的方法。</p> <p>利用百度搜索提供的这个链接接口，可以通过 requests 的 get() 函数提交查询，响应结果为百度搜索结果。这个问题的 IPO 描述如下：</p> <p>输入：待查询关键字</p> <p>处理：自动获得百度搜索结果页面，并对页面内容解析处理</p> <p>输出：返回链接的标题列表</p>	案例展示
5M	<p>四、总结：</p> <p>学习完本课内容，通过以上教学内容，学生将了解网络爬虫的工作原理，掌握 requests 和 Beautiful Soup 库的使用，以及实际应用爬虫和自动化技术。他们将能够获取网络信息并处理数据，同时了解信息获取和处理的伦理和法律问题。</p>	总结
5M	<p>★课程思政建设：</p> <p>本章的学习侧重培养学生的信息获取和处理能力，以及自动化思维。学生将学会如何使用技术工具来获取有用的信息，并了解信息处理的道德和法律问题。教案设计强调实际问题的引入，以激发学生的兴趣，同时强调思政建设内容，培养信息获取和自动化思维。</p>	思政建设

s1:

授课时间	2022. 10. 03	课次	01
授课方式 (请打√)	理论课 <input type="checkbox"/> 讨论课 <input type="checkbox"/> 实验课 <input checked="" type="checkbox"/> 习题课 <input type="checkbox"/> 其他 <input type="checkbox"/>	课时 安排	2
授课题目 (教学章、节或主题): 实验 1 环境安装及使用			
教学目的与要求: (1)理解 Python 语言的特点以及其重要性。 (2)掌握 Python 语言程序的编写方法。 (3)掌握 Python 语言开发和运行环境的配置方法。 (4)理解编写程序的 IPO 方法。 (5)了解 Python 版本更迭过程和新旧版本的主要区别。			
教学重点及难点: 编写程序最基本的 IPO 方法, python 语言安装和运行过程。			
作业、讨论题、思考题: 习题 1.2 、 1.7			
课后小结: 通过这次课的学习, 熟练掌握配置 Python 开发环境的方法。			
下节课预习重点: Python 数值类型的操作。			
参考文献: [1] 嵩天、礼欣、黄天羽.《Python 语言程序设计基础(第 2 版)》. 高等教育出版社. 2017			

课时分配	教学内容	方法及手段
10M	<p>一、实验目的</p> <p>1.掌握 Python 语言的环境安装及配置的基本思路。</p> <p>2.掌握多种运行 Python 程序的操作方法。</p> <p>(1) 采用 Windows 操作系统命令行工具。</p> <p>(2) 采用 Python IDLE 交互式。</p> <p>(3) 采用 Python IDLE 文件式。</p> <p>(4) 采用 Pycharm 集成开发环境。</p> <p>(5) 采用 Anaconda 的 Jupyter 工具。</p> <p>3.掌握编写程序的 IPO 基本方法。</p> <p>4.掌握 input()函数的使用。</p> <p style="padding-left: 2em;"><变量> = input(<提示性文字>)</p> <p style="padding-left: 2em;">配合 eval () 函数使用</p> <p>5.掌握 print()函数的使用。</p> <p style="padding-left: 2em;">print(" 格式字符 " % (变量列表))</p> <p style="padding-left: 2em;">print(" {格式设置 } ".format(变量列表))</p>	讲解及演示
10M	<p>二、实验内容</p> <p>(1) 采用上述 5 种运行方法分别实现：屏幕打印“hello world!”。</p> <p>(2) 在 Pycharm 环境中实现：键盘输入两个整数，计算两数之和，输出结果。</p> <p>(3) 选择其中一种运行方法实现：圆面积的计算。(调用 math 库)</p>	讲解及演示
20M	<p>三、实验过程</p> <p style="text-align: center;">指导过程中查阅并记录成绩</p>	实验指导及同学演示
20M		

课时分配	教学内容	方法及手段
15M	<p>四、总结</p> <p>能够分析问题，进行算法设计，完成程序代码的编辑、运行，并测试程序无误。</p> <p>1.注意标点符号的中英文书写，特别是括号、冒号的使用。</p> <p>2.注意语句使用的灵活性。</p>	总结本次实验问题
15M	<p>五、思考拓展</p> <p>1.数值运算。描述：</p> <p>获得用户输入的一个字符串，格式如下：M OP N。</p> <p>根据 OP，输出 M OP N 的运算结果，统一保存小数点后 2 位。</p> <p>注意：M 和 OP、OP 和 N 之间可以存在多个空格，不考虑输入错误情况。</p> <p>输入示例：</p> <p style="padding-left: 40px;">2 + 3</p> <p>输出示例：</p> <p style="padding-left: 40px;">5.00</p> <p>程序源代码，以及运行结果截图如下：</p> <p>2.正方形绘制。源代码以及运行结果截图如下(图省略)：</p> <p>3.教材案例自行选择运行。</p>	讲解及演示

s2:

授课时间	2022. 10. 10	课次	02
授课方式 (请打√)	理论课 <input type="checkbox"/> 讨论课 <input type="checkbox"/> 实验课 <input checked="" type="checkbox"/> 习题课 <input type="checkbox"/> 其他 <input type="checkbox"/>	课时 安排	2
授课题目 (教学章、节或主题): 实验 2 数值类型操作			
教学目的与要求: (1) 掌握 3 种数字类型的概念和使用。 (2) 了解 3 种数字类型在计算机中的表示方法。 (3) 运用 Python 的标准数学库进行数值计算。			
教学重点及难点: Python 语言的基本数字数据类型, 包括整数、浮点类、复数类型的概念和使用。			
作业、讨论题、思考题: 习题 3.1 、 3.4			
课后小结: 通过这次课的学习, 熟练掌握计算机中常用的数字类型及操作, 包括 Python 内置的数值运算操作和数字类型转换函数等, 进一步介绍了常用的数学计算标准库 math 库方法。			
下节课预习重点: Python 字符类型的操作。			
参考文献: [1] 嵩天、礼欣、黄天羽. 《Python 语言程序设计基础(第 2 版)》. 高等教育出版社. 2017			

课时分配	教学内容	方法及手段
10M	<p>一、实验目的</p> <p>(1)掌握 3 种数字类型的概念和使用。</p> <p>(2)掌握内置数值运算操作符的使用。共 9 个 + - * / // % - + **。</p> <p>(3)运用 Python 内置的数值运算函数进行数值计算。abs() pow() max()等。</p> <p>(4)运用 Python 的标准数学库进行数值计算。math 库 4 个常数，44 个函数。</p>	讲解及演示
10M	<p>二、实验内容</p> <p>(1)思考各操作符的含义及运算优先级，并编程计算下列表达式的值。P69 3.5</p> <ol style="list-style-type: none"> 1. $30-3**2+8//3**2*10$ 2. $3*4**2/8\%5$ 3. $2**2**3$ 4. $(2.5+1.25j)*4j/2$ <p>(2)请利用数值运算符、内置函数、math 库等完成以下计算。</p> <ol style="list-style-type: none"> 1. 求 27 和 18 的最大公约数 2. 求 lne 的值 3. 求 -2.6 的绝对值 4. 对键盘输入任意小数，向下取整 <p>(3)天天向上的力量。</p> <p>一年 365 天，以第 1 天的能力值为基数，记为 1.0。</p> <p>当好好学习时能力值相比前一天提高 f，当没有学习时能力值相比前一天下降 f。</p> <p>每天努力和每天放任，一年下来的能力值分别是多少？</p> <p>(f 值由键盘输入，例如：0.01)</p> <p>(4)编写程序，计算并输出方程的实数根。</p> <p>求：输入一元二次方程 $ax^2+bx+c=0$ 的参数 a、b、c。</p>	讲解及演示

课时分配	教学内容	方法及手段
20M	<p>计算并输出方程的实数根（结果保留两位小数）。若方程没有实数根，输出“方程没有实数根！”。</p> <p>（例如：a b c 依次为 2,3,1）</p>	实验指导及同学演示
20M	<p>三、实验过程</p> <p>指导过程中查阅并记录成绩</p>	
15M	<p>四、总结</p> <p>能够分析问题，进行算法设计，完成程序代码的编辑、运行，并测试程序无误。</p> <p>1.注意标点符号的中英文书写，特别是括号、冒号的使用。</p> <p>2.注意语句使用的灵活性。</p>	总结本次实验问题
15M	<p>五、思考拓展</p> <p>1.上述（3），f 值为 0.01，如果一周工作 5 天，休息 2 天，一年下来的能力值是多少？</p> <p>2.上述（4），如果计算复数根呢？</p> <p>3.利用 help 帮助</p> <p>1) 打开 IDLE 交互式界面，输入 help('/') help('math.sqrt') help(math) help(cmath)</p> <p>2) 打开 IDLE 交互式界面，输入 help()进入交互式帮助，输入 math。回车退出</p> <p>3) 打开 IDLE 交互式界面，菜单 Help ->Python Docs F1</p>	讲解及演示

s3:

授课时间	2022. 10. 17	课次	03
授课方式 (请打√)	理论课 <input type="checkbox"/> 讨论课 <input type="checkbox"/> 实验课 <input checked="" type="checkbox"/> 习题课 <input type="checkbox"/> 其他 <input type="checkbox"/>	课时 安排	2
授课题目 (教学章、节或主题): 实验 3 字符串类型操作			
教学目的与要求: (1)掌握字符串类型的概念和使用。 (2)掌握字符串类型的格式化操作方法和应用。			
教学重点及难点: 掌握 Python 语言的字符串类型概念和使用。			
作业、讨论题、思考题: 习题 3.6			
课后小结: 通过这次课的学习, 熟练掌握字符串类型及其操作和格式化方法。			
下节课预习重点: Python 程序控制结构。			
参考文献: [1] 嵩天、礼欣、黄天羽. 《Python 语言程序设计基础(第 2 版)》. 高等教育出版社. 2017			

课时分配	教学内容	方法及手段
10M	<p>一、实验目的</p> <p>(1)掌握字符串类型的概念。</p> <p>(2)掌握字符串常见操作方法。</p> <p>基本的操作符 P80: str [i]索引 str [N : M]切片 +连接 *重复 in 是否含</p> <p>内置处理函数 P82: len() str() chr() ord()</p> <p>内置处理方法 P83: str.lower() str.center()</p> <p>(3)掌握字符串类型的格式化操作方法 str.format()和应用。</p>	讲解及演示
10M	<p>二、实验内容</p> <p>(1)已知</p> <pre>s1='python' s2="程序设计" s3="python 课程" s4="加油! "</pre> <pre>s5=202202</pre> <p>利用字符串操作方法输出下列结果:</p> <ol style="list-style-type: none"> python 程序设计 加油! 加油! 加油! p 课程 python 程序设计课程 判断 s3 中是否包含单词 'python' 判断 s1 是否是回文字符 判断 s5 是否是回文数字 字符串 s3 的长度 	讲解及演示

课时分配	教学内容	方法及手段
15M	<p>序无误。</p> <ol style="list-style-type: none"> 1.注意标点符号的中英文书写，特别是括号、冒号的使用。 2.注意语句使用的灵活性。 <p>五、思考拓展</p> <ol style="list-style-type: none"> 1.将电话号码大写输出？ 2.动态进度条。提示： time.sleep(0.5) 函数适当进行停顿 print 语句中取消换行输出，并将光标移动到行首。 补充 p81 转义字符 \t 水平制表 \n 换行 \r 回到行首 \a 蜂鸣(CMD) 3.小技巧 <ol style="list-style-type: none"> 1) 打开 IDLE 交互式界面，输入 help('/') help('math.sqrt') help(math) help(cmath) 2) pycharm 快捷键 CTRL + / : 快速注释、取消注释 	讲解及演示

s4:

授课时间	2022. 10. 24	课次	04
授课方式 (请打√)	理论课 <input type="checkbox"/> 讨论课 <input type="checkbox"/> 实验课 <input checked="" type="checkbox"/> 习题课 <input type="checkbox"/> 其他 <input type="checkbox"/>	课时 安排	2
授课题目 (教学章、节或主题): 实验 4 程序的控制结构			
教学目的与要求: (1) 运用 if 语句实现分支结构。 (2) 运用 for 语句和 while 语句实现循环结构。			
教学重点及难点: 灵活运用 python 语言的控制结构语句。			
作业、讨论题、思考题: 习题 4.1			
课后小结: 通过这次课的学习, 熟练掌握 Python 控制结构的实现方法。			
下节课预习重点: Python 控制结构的综合运用。			
参考文献: [1] 嵩天、礼欣、黄天羽. 《Python 语言程序设计基础(第 2 版)》. 高等教育出版社. 2017			

课时分配	教学内容	方法及手段
10M	<p>一、实验目的</p> <p>(1)运用 if 语句实现分支结构。</p> <p>(2)运用 for 语句和 while 语句实现循环结构。</p>	讲解及演示
10M	<p>二、实验内容</p> <p>(1)货币兑换。(提示:综合运用字符串切片、分支语句)</p> <p style="padding-left: 2em;">RMB6.78 = USD1</p> <p>(2)整数加减和。(提示 方法多种多样)</p> <p style="padding-left: 2em;">编写程序计算如下数列的值: 1-2+3-4...966</p> <p style="padding-left: 2em;">其中,所有数字为整数,从 1 开始递增,奇数为正,偶数为负。</p> <p>(3)三位水仙花数。(提示 用字符串存储结果,切片处理末尾逗号)</p> <p style="padding-left: 2em;">"水仙花数"是指一个三位整数,其各位数字的 3 次方和等于该数本身。</p> <p style="padding-left: 2em;">例如: ABC 是一个"3 位水仙花数",则: A 的 3 次方+B 的 3 次方+C 的 3 次方 = ABC。</p> <p style="padding-left: 2em;">请按照从小到大的顺序输出所有的 3 位水仙花数,请用"逗号"分隔输出结果。</p> <p>(4)用户的三次登录机会。(提示 与 and 或 or 非 not)</p> <p style="padding-left: 2em;">给用户三次输入用户名和密码的机会,要求如下:</p> <p style="padding-left: 4em;">1)如输入第一行输入用户名为 'Kate',第二行输入密码为 '666666',输出 '登录成功!',退出程序;</p> <p style="padding-left: 4em;">2)当一共有 3 次输入用户名或密码不正确输出"3 次用户名或者密码均有误!退出程序。"</p>	讲解及演示
20M	<p>三、实验过程</p> <p style="text-align: center; padding: 10px 0;">指导过程中查阅并记录成绩</p>	实验指导及同学演示
20M		实验指导及同学演示

课时分配	教学内容	方法及手段															
15M	<p>四、总结</p> <p>能够分析问题，进行算法设计，完成程序代码的编辑、运行，并测试程序无误。</p> <p>1.注意标点符号的中英文书写，特别是括号、冒号的使用。</p> <p>2.注意语句使用的灵活性。</p>	总结本次实验问题															
15M	<p>五、思考拓展</p> <p>1.身体质量指数 BMI。（提示 输入多个信息 a,b=eval(input("英文逗号分隔："))）</p> <p>定义：BMI = 体重 (kg) / 身高² (m²)</p> <table border="1" data-bbox="355 1003 612 1153"> <thead> <tr> <th>分类</th> <th>国际BMI值</th> <th>国内BMI值</th> </tr> </thead> <tbody> <tr> <td>偏瘦</td> <td><18.5</td> <td><18.5</td> </tr> <tr> <td>正常</td> <td>18.5 ~ 25</td> <td>18.5 ~ 24</td> </tr> <tr> <td>偏胖</td> <td>25 ~ 30</td> <td>24 ~ 28</td> </tr> <tr> <td>肥胖</td> <td>≥30</td> <td>≥28</td> </tr> </tbody> </table> <div data-bbox="671 1003 928 1160" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>输入示例1</p> <p>1.68,41</p> <p>输出示例1</p> <p>BMI数值为:14.53 BMI指标为:国际'偏瘦',国内'偏瘦'</p> </div> <p>要求如下：</p> <p>(1) 混合计算并给出国际和国内的 BMI 分类；</p> <p>(2) 使用 input()获得测试用例输入时，不要增加提示字符串。</p>	分类	国际BMI值	国内BMI值	偏瘦	<18.5	<18.5	正常	18.5 ~ 25	18.5 ~ 24	偏胖	25 ~ 30	24 ~ 28	肥胖	≥30	≥28	讲解及演示
分类	国际BMI值	国内BMI值															
偏瘦	<18.5	<18.5															
正常	18.5 ~ 25	18.5 ~ 24															
偏胖	25 ~ 30	24 ~ 28															
肥胖	≥30	≥28															

s5:

授课时间	2022. 10. 31	课次	05
授课方式 (请打√)	理论课 <input type="checkbox"/> 讨论课 <input type="checkbox"/> 实验课 <input checked="" type="checkbox"/> 习题课 <input type="checkbox"/> 其他 <input type="checkbox"/>	课时 安排	2
授课题目 (教学章、节或主题): 实验 5 控制结构的综合运用			
教学目的与要求: (1) 综合运用分支语句、循环语句。 (2) 掌握随机库的使用方法。 (3) 了解程序的异常处理及用法。			
教学重点及难点: 随机库的灵活使用方法。			
作业、讨论题、思考题: 习题 4.1			
课后小结: 通过这次课的学习, 熟练掌握控制结构的综合运用的方法。			
下节课预习重点: Python 函数的定义及调用。			
参考文献: [1] 嵩天、礼欣、黄天羽. 《Python 语言程序设计基础(第 2 版)》. 高等教育出版社. 2017			

课时分配	教学内容	方法及手段
10M	<p>一、实验目的</p> <p>(1)综合运用分支语句、循环语句。</p> <p>(2)掌握随机库的使用方法。 P112</p> <p>(3)了解程序的异常处理及用法。</p>	讲解及演示
10M	<p>二、实验内容</p> <p>(1)输入身份证号，输出性别。</p> <p>采用基本分支语句实现。(提示 身份证号倒数第二位，若偶数为女，若奇数为男)</p> <p>(2)四位玫瑰花数。</p> <p>四位玫瑰数是 4 位数的自幂数。自幂数是指一个 n 位数，它的每个位上的数字的 n 次幂之和等于它本身。ABCD 是一个"4 位玫瑰花数"，则：A 的 4 次方+B 的 4 次方+C 的 4 次方+D 的 4 次方= ABCD</p> <p>请输出所有四位玫瑰数，按照从小到大顺序，每个数字一行。</p> <p>(3)猜数游戏。</p> <ul style="list-style-type: none"> ➤ 在 1 到 100 之间指定一个数（思考用随机数），然后请用户循环猜测这个数字。 ➤ 对于每个答案只回复“猜大了”或“猜小了”。 ➤ 直到猜测准确为止，输出用户的猜测次数。 <p>(4)圆周率的计算。(提示 random 库)</p> <p>https://python123.io/student/courses/6318/groups/79296/problems/programmings/6884#pagetop</p> <p>采用蒙特卡罗方法，在一个正方形中撒点，根据在 1/4 圆内点的数量占总撒点数的比例计算圆周率值。</p>	讲解及演示
20M	<p>三、实验过程</p>	实验指导及同学演示

课时分配	教学内容	方法及手段
15M	<p style="text-align: center;">指导过程中查阅并记录成绩</p> <p>四、总结</p> <p>能够分析问题，进行算法设计，完成程序代码的编辑、运行，并测试程序无误。</p> <p>1.注意标点符号的中英文书写，特别是括号、冒号的使用。</p>	总结本次实验问题
15M	<p>2.注意语句使用的灵活性。</p> <p>五、思考拓展</p> <p>1.上述实验（1）采用 if - else 语句紧凑结构实现。 （提示 < 表达式 1 > if <条件> else < 表达式 2 >）。</p> <p>2.同心圆靶盘。（提示 for 语句 turtle 库） 画出包含 3 个同心圆，半径依次为 50、100、150 的靶盘。（可尝试填充颜色）</p> <p>3.如果上述实验（3）猜数字游戏，用 try...except 语句进行异常处理，如何完善代码？</p>	讲解及演示

s6:

授课时间	2022. 11. 07	课次	06
授课方式 (请打√)	理论课 <input type="checkbox"/> 讨论课 <input type="checkbox"/> 实验课 <input checked="" type="checkbox"/> 习题课 <input type="checkbox"/> 其他 <input type="checkbox"/>	课时 安排	2
授课题目 (教学章、节或主题): 实验 6 函数			
教学目的与要求: (1) 函数的定义和使用 (2) 函数的参数传递: 可选参数传递、参数名称传递、函数的返回值			
教学重点及难点: 函数的定义和灵活使用方法			
作业、讨论题、思考题: 习题 5.2			
课后小结: 通过这次课的学习, 熟练掌握函数定义和运用的方法。			
下节课预习重点: Python 组合数据类型的操作方法。			
参考文献: [1] 嵩天、礼欣、黄天羽. 《Python 语言程序设计基础(第 2 版)》. 高等教育出版社. 2017			

课时分配	教学内容	方法及手段
10M	<p>一、实验目的</p> <p>(1) 函数的定义和使用</p> <p>(2) 函数的参数传递: 可选参数传递、参数名称传递、函数的返回值</p> <p>二、实验内容</p> <p>(1) 生日歌。(参数传递)</p>	讲解及演示
10M	<p>输入寿星姓名、演唱者姓名, 定义并调用函数 <code>happyb (name,singer)</code> 函数, 输出相应生日歌歌词。</p> <p>参考效果如下:</p> <pre> 祝你生日快乐 祝你生日快乐 寿星张三同学 祝你生日快乐 演唱者: 王五 </pre> <p>思考: 按照位置传参、按照名称传参的区别?</p> <p>(2) n 位随机密码。(形参 实参 默认参数值)</p> <p>获取用户输入整数 n 为长度, 产生 1 个长度为 n 位的整数随机密码。以 17 为随机数种子。</p> <p>提示: 产生密码采用 <code>random.randint(a,b)</code> 函数, 产生 [a,b] 之间任意整数。 <code>help('random.randint')</code>。</p> <p>(说明: 在...处补充完善代码)</p> <pre> import random def genpwd(length): # 定义函数 random.seed(17) # 确定随机数种子 ... # 产生并返回随机密码 </pre>	讲解及演示

课时分配	教学内容	方法及手段
20M	<p>对于用户输入的字符串 s，输出反转后的字符串。这个问题的基本思想是把字符串看作一个递归对象。</p> <p>(说明：在...处补充完善代码)</p> <pre>def rev(s): if ... : return s else: return rev(...) + s[0]</pre> <p><code>print(rev('ABC'))</code> # 函数调用</p>	及同学演示
15M	<h3>三、实验过程</h3> <p>指导过程中查阅并记录成绩</p>	总结本次实验问题
15M	<h3>四、总结</h3> <p>能够分析问题，进行算法设计，完成程序代码的编辑、运行，并测试程序无误。</p> <ol style="list-style-type: none"> 1.注意标点符号的中英文书写，特别是括号、冒号的使用。 2.注意语句使用的灵活性。 <h3>五、思考拓展</h3> <p>1.上述实验（1），生日歌最后添加日期。教材 P133~136</p> <p>参考代码如下</p> <pre># import datetime # today = datetime.datetime.now() from datetime import datetime 从库 导入 类 today = datetime.now() # now()方法：获取当前时间 #today = datetime(2023, 4, 26) # datetime ()方法：构造时间日期对象 print(today.strftime("%Y 年%m 月%d 日"), end=") # strftime()方法：日期格式设置</pre>	讲解及演示

课时分配	教学内容	方法及手段
	<pre>w = today.isoweekday() # isoweekday() 方法：根据日期计算星期 print("星期{}".format(w),end='!')</pre>	

s7:

授课时间	2022. 11. 14	课次	07
授课方式 (请打√)	理论课 <input type="checkbox"/> 讨论课 <input type="checkbox"/> 实验课 <input checked="" type="checkbox"/> 习题课 <input type="checkbox"/> 其他 <input type="checkbox"/>	课时 安排	2
授课题目 (教学章、节或主题): 实验 7 组合数据类型			
教学目的与要求: (1) 序列: 字符串、元组(,)、列表[,] 定义、索引、切片、操作函数、操作方法。 (2) 集合: 集合 {,} 定义、操作符、操作函数。 (3) 映射: 字典{:, :} 定义、索引、操作函数、操作方法。			
教学重点及难点: 组合数据类型的操作函数、方法。			
作业、讨论题、思考题: 习题 6.1			
课后小结: 通过这次课的学习, 熟练掌握配置 Python 组合数据类型的操作方法。			
下节课预习重点: Python 文件的打开、读写、关闭。			
参考文献: [1] 嵩天、礼欣、黄天羽. 《Python 语言程序设计基础(第 2 版)》. 高等教育出版社. 2017			

课时分配	教学内容	方法及手段
10M	<p>一、实验目的</p> <p>(1) 序列：字符串、元组(,)、列表[,] 定义、索引、切片、操作函数、操作方法</p> <p>(2) 集合：集合 {,} 定义、操作符、操作函数</p> <p>(3) 映射：字典 {:,} 定义、索引、操作函数、操作方法</p>	讲解及演示
10M	<p>二、实验内容</p> <p>(1)序列基础练习。</p> <p>字符串 <code>s = "hello world"</code></p> <p>姓名元组 <code>t = ("张三","李四","王五")</code></p> <p>成绩元组 <code>g = (80,100,90,90)</code></p> <p>课程列表 <code>c = ["C","JAVA","python","JAVA"]</code></p> <ol style="list-style-type: none"> 1. 遍历输出上述序列元素？ 2. 计算成绩总和？ 3. 查找成绩最高分？ 4. 成绩排序，从高到低？ 5. 学生姓名逆序输出？ 6. 末尾增加课程 "R" "PS" <p>(2)计算中位数。</p> <p>中位数是指将数据按大小顺序排列起来，形成一个数列，居于数列中间位置的那个数据为中位数，若有偶数个，取中间两个数的平均数作为中位数。</p> <p>(3)统计课程门数（不重复统计）。</p> <p>（提示：方法 <code>str.split()</code>将字符串转列表，函数 <code>set()</code>将列表转集合）</p>	讲解及演示

课时分配	教学内容	方法及手段
20M	<p>已知字符串 chose = "高等数学 java python 人工智能 高等数学 java 高等数学 python 人工智能 高等数学 java"。</p> <p>其中包含了大量重复课程，请统计字符串 chose 中共有几门课程（不重复统计）。</p> <p>思考：用 list()函数将字符串转列表，结果一致吗？</p> <p>(4) 验证身份证号真假函数 iden()。</p> <p>身份证号最后一位是校验位，计算规则如下：</p> <ul style="list-style-type: none"> ➤ 将前面的身份证号码 17 位数分别乘以不同的系数。从第一位到第十七位的系数分别为: (7,9,10,5,8,4,2,1,6,3,7,9,10,5,8,4,2)。 ➤ 将这 17 位数字和系数相乘的结果相加。 ➤ 用加出来和除以 11,余数只可能有 0~10 这 11 个数字。其分别对应的最后一位身份证的号码为('1','0','X','9','8','7','6','5','4','3','2')。 <p>根据身份证号前 17 位计算并判断最后一位是否造假？</p> <p>三、实验过程</p>	实验指导及同学演示
20M	<p>指导过程中查阅并记录成绩</p>	
15M	<p>四、总结</p> <p>能够分析问题，进行算法设计，完成程序代码的编辑、运行，并测试程序无误。</p> <ol style="list-style-type: none"> 1.注意标点符号的中英文书写，特别是括号、冒号的使用。 2.注意语句使用的灵活性。 	总结本次实验问题
15M	<p>五、思考拓展</p> <ol style="list-style-type: none"> 1.去重人名统计。 <p>https://python123.io/student/courses/6318/groups/79300/problems/programmings/16707#pagetop</p>	讲解及演示

s8:

授课时间	2022. 11. 21	课次	08
授课方式 (请打√)	理论课 <input type="checkbox"/> 讨论课 <input type="checkbox"/> 实验课 <input checked="" type="checkbox"/> 习题课 <input type="checkbox"/> 其他 <input type="checkbox"/>	课时 安排	2
授课题目 (教学章、节或主题): 实验 8 文件			
教学目的与要求: (1) 掌握文件的打开和关闭等基本操作。 (2) 掌握文件的读写方法。			
教学重点及难点: 文件的读和写			
作业、讨论题、思考题: 课后习题 7.2。			
课后小结: 通过这次课的学习, 熟练掌握配置 Python 文件的读写操作。			
下节课预习重点: Python 综合练习。			
参考文献: [1] 嵩天、礼欣、黄天羽. 《Python 语言程序设计基础(第 2 版)》. 高等教育出版社. 2017			

课时分配	教学内容	方法及手段
10M	<p>一、实验目的</p> <p>(1) 文件打开 <code><变量名> = open(<文件路径>, <打开模式>, <encoding=>)</code></p> <p>(2) 文件读 <code><file>.read(size)</code> <code><file>.readline(size)</code> <code><file>.readlines(hint)</code></p> <p>(3) 文件写 <code><file>.write(str)</code> <code><file>.writelines(list)</code></p> <p>(4) 文件关闭 <code><变量名>.close()</code></p> <p>二、实验内容</p> <p>(1) 唐诗一首。</p> <p>在代码路径下创建“画.txt”并写入以下内容。</p>	讲解及演示
10M	<pre> 画 王维 远看山有色， 近听水无声。 春去花还在， 人来鸟不惊。 </pre> <pre> ff = open(__?__, __?__, encoding='ansi') ff.seek(0) s = input() while s!='#': __?__ # 逐行写入内容 s = input() __?__ # 读并打印文件内容 __?__ # 关闭文件 </pre> <p>(2) 复制文件内容。</p>	讲解及演示

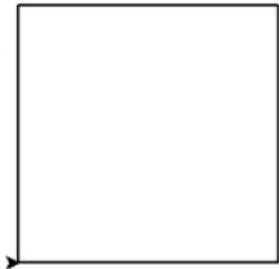
课时分配	教学内容	方法及手段
20M	<p>已知 windows 记事本中文件“画.txt”存有唐诗内容。 编写程序，打开“画.txt”并将其内容复制至"new.txt"。</p> <p>(3) 满意度调查问卷结果</p> <p>问卷调查结果在文本文件 result.txt 中保存，请读取该文件后统计问卷结果，将最终统计结果字典追加到该文本。参考步骤如下：</p> <pre>#打开文件 #创建空字典 count={} #内容统一小写 方法 upper() #根据空白字符分割为列表 方法 split() #统计关键字出现次数 方法 count(), 并更新字典值 #写入结果 #关闭文件</pre> <p>(4) 统计选课频次。</p> <p>已知学生选课信息在“数据文件\chose.txt”中，统计每门课程被选次数，并输出前三名热门课程及被选次数。</p> <p>补充读取并统计选课频次过程的代码，保存至字典 counts。</p> <p>输出前三名代码参考如下：</p> <pre>re = list(counts.items()) # 字典转为列表类型 re.sort(key = lambda x:x[1],reverse=True) # 根据单词出现的次数，由高到低排序 for k in range(3): # 打印排名前 3 的单词名称及次数 name,num = re[k] print(name,num)</pre>	实验指导及同学演示
20M	三、实验过程	

课时分配	教学内容	方法及手段
15M	<p style="text-align: center;">指导过程中查阅并记录成绩</p> <p>四、总结</p> <p>能够分析问题，进行算法设计，完成程序代码的编辑、运行，并测试程序无误。</p> <ol style="list-style-type: none"> 1.注意标点符号的中英文书写，特别是括号、冒号的使用。 2.注意语句使用的灵活性。 	总结本次实验问题
15M	<p>五、思考拓展</p> <ol style="list-style-type: none"> 1.去重人名统计 2. 哈姆雷特小说（英文）热词统计 教材 P173 <p>已知 hamlet.txt 存储哈姆雷特小说英文内容，请统计出现频次最高的十个热词。</p>	讲解及演示

s9:

授课时间	2022. 11. 28	课次	09
授课方式 (请打√)	理论课 <input type="checkbox"/> 讨论课 <input type="checkbox"/> 实验课 <input checked="" type="checkbox"/> 习题课 <input type="checkbox"/> 其他 <input type="checkbox"/>	课时 安排	2
授课题目 (教学章、节或主题): 实验 9 综合练习			
教学目的与要求: (1) 数值类型及操作运算 (2) 字符串类型处理 (3) 组合数据类型的处理 (4) 控制结构常见语句及使用 (5) 文件读写 (6) 第三方库的安装及使用			
教学重点及难点: 利用 IPO 方法, 综合运用 python 语言完成功能			
作业、讨论题、思考题: 课后习题 7.2			
课后小结: 通过这次课的学习, 熟练掌握 Python 开发的方法。			
下节课预习重点: Python 库的使用。			
参考文献: [1] 嵩天、礼欣、黄天羽. 《Python 语言程序设计基础(第 2 版)》. 高等教育出版社. 2017			

课时分配	教学内容	方法及手段
10M	<p>一、实验目的</p> <ul style="list-style-type: none"> (1) 数值类型及操作运算 (2) 字符串类型处理 (3) 组合数据类型的处理 (4) 控制结构常见语句及使用 (5) 文件读写 (6) 第三方库的安装及使用 <pre>pip install -i https://pypi.douban.com/simple/ jieba</pre> <p>(永久配置镜像源)</p> <p>二、实验内容</p> <p>(1)数学表达式。</p>	讲解及演示
10M	<p>仅使用 Python 基本语法，即不使用任何模块，编写 Python 程序计算下列数学表达式的结果并输出，小数点后保留 3 位。</p> $x = \sqrt{\frac{(3^4 + 5 \times 6^7)}{8}}$ <p>(2) 格式输出。</p> <p>0x4DC0 是一个十六进制数，它对应的 Unicode 编码是中国古老的《易经》六十四卦的第 1 卦，请输出第 51 卦（震卦）对应的 Unicode 编码的二进制、十进制、八进制和十六进制格式。</p> <p>(3) 正方形。</p> <p>使用 turtle 库的 turtle.fd() 函数和 turtle.seth() 函数绘制一个边长为 200 的正方形，效果如下图所示。请结合格式框架，补充横线处代码。</p>	讲解及演示

课时分配	教学内容	方法及手段
<p>20M</p> <p>20M</p>	<div style="display: flex; align-items: flex-start;">  <pre style="font-family: monospace;"> _____ turtle # 导入 turtle 库 d = 0 for i in _____: _____ # 前进 200 d = _____ turtle.seth(d) # 设置绝对角度 </pre> </div> <p>思考: 使用 <code>turtle.left(90)</code>, 如何修改代码?</p> <p>(4)统计类型数量。</p> <p>列表 <code>ls</code> 中存储了我国 39 所 985 高校所对应的学校类型, 请以这个列表为数据变量, 统计各类型的数量, 保存在字典 <code>d</code> 中并输出。</p> <pre style="font-family: monospace;"> ls = ["综合", "理工", "综合", "综合", "综合", "综合", "综合", "综合", "综合", "综合", \ "师范", "理工", "综合", "理工", "综合", "综合", "综合", "综合", "综合", \ "综合", "理工", \ "理工", "理工", "理工", "师范", "综合", "农林", "理工", "综合", "理工", \ "理工", \ "理工", "综合", "理工", "综合", "综合", "理工", "农林", "民族", "军事"] </pre> <p style="color: red;">补充: 符号 <code>\</code> 的作用: 一行代码分多行写。此外 <code>for</code> 循环条件用 <code>()</code> 可换行, 字符串用 <code>''' '''</code> 可换行。</p> <p>(5)文件分词。</p> <p>中国共产党第十九次全国代表大会报告中一句话, 存在记事本 <code>report.txt</code> 中。将其内容作为字符串变量 <code>s</code>, 编写 Python 程序, 分别用 Python 内置函数及 <code>jieba</code> 库中已有函数, 计算字符串 <code>s</code> 的字符个数、词语个数。</p> <h3>三、实验过程</h3>	<p>实验指导及同学演示</p>

课时分配	教学内容	方法及手段
15M	<p style="text-align: center;">指导过程中查阅并记录成绩</p> <p>四、总结</p> <p>能够分析问题，进行算法设计，完成程序代码的编辑、运行，并测试程序无误。</p> <ol style="list-style-type: none"> 1.注意标点符号的中英文书写，特别是括号、冒号的使用。 2.注意语句使用的灵活性。 	总结本次实验问题
15M	<p>五、思考拓展</p> <ol style="list-style-type: none"> 1. 三国演示小说（中文）热词统计 教材 P174 <p>已知三国演义.txt 存储三国演义小说中文内容，请统计出现频次最高的五个人名。</p>	讲解及演示

s10:

授课时间	2022. 12. 05	课次	10
授课方式 (请打√)	理论课 <input type="checkbox"/> 讨论课 <input type="checkbox"/> 实验课 <input checked="" type="checkbox"/> 习题课 <input type="checkbox"/> 其他 <input type="checkbox"/>	课时 安排	2
授课题目 (教学章、节或主题): 实验 10 库的使用			
教学目的与要求: (1) turtle 库 P305 (2)datetime 库 P304 (3) random 库 P304 (4) jieba 库 P304 (5) PIL 库 P306。			
教学重点及难点: 熟练应用 python 语言常见库			
作业、讨论题、思考题: 课后习题 8.4。			
课后小结: 通过这次课的学习, 熟练掌握配置 Python 库的使用方法。			
下节课预习重点: Python 期末测试。			
参考文献: [1] 嵩天、礼欣、黄天羽. 《Python 语言程序设计基础(第 2 版)》. 高等教育出版社. 2017			

课时分配	教学内容	方法及手段
10M	<p>一、实验目的</p> <ul style="list-style-type: none"> (1) 数值类型及操作运算 (2) 字符串类型处理 (3) 组合数据类型的处理 (4) 控制结构常见语句及使用 (5) 文件读写 (6) 第三方库的安装及使用 <pre style="background-color: #f0f0f0; padding: 5px;">pip install -i https://pypi.douban.com/simple/ jieba</pre> <p>二、实验内容</p> <p>(1)打印九九乘法表</p>	讲解及演示
10M	<pre style="background-color: #2e3436; color: #eeeeec; padding: 10px;">1*1=1 1*2=2 2*2=4 1*3=3 2*3=6 3*3=9 1*4=4 2*4=8 3*4=12 4*4=16 1*5=5 2*5=10 3*5=15 4*5=20 5*5=25 1*6=6 2*6=12 3*6=18 4*6=24 5*6=30 6*6=36 1*7=7 2*7=14 3*7=21 4*7=28 5*7=35 6*7=42 7*7=49 1*8=8 2*8=16 3*8=24 4*8=32 5*8=40 6*8=48 7*8=56 8*8=64 1*9=9 2*9=18 3*9=27 4*9=36 5*9=45 6*9=54 7*9=63 8*9=72 9*9=81</pre> <p>(2) 数字不同数之和</p> <p>获得用户输入的一个整数 N，输出 N 中所出现不同数字的和。</p> <p>例如：用户输入 123123123，其中所出现的不同数字为：1、2、3，这几个数字和为 6。</p> <p>(3) Hello World 的条件输出</p> <p>获得用户输入的一个整数，参考该整数值，打印输出"Hello World"，要求：</p> <p style="padding-left: 40px;">如果输入值是 0，直接输出"Hello World"</p> <p style="padding-left: 40px;">如果输入值大于 0，以两个字符一行方式输出"Hello World"（空格也是</p>	讲解及演示

课时分配	教学内容	方法及手段
20M	字符) 如果输入值小于 0, 以垂直方式输出"Hello World" 思考: 如果输入值大于 0, 以每个字符后面添加 1 个空格方式输出"Hello" (4) 100 以内素数之和 求 100 以内所有素数之和并输出。素数指从大于 1, 且仅能被 1 和自己整除的整数。 提示: 可以逐一判断 100 以内每个数是否为素数, 然后求和。 三、实验过程 指导过程中查阅并记录成绩	实验指导及同学演示
20M	四、总结 能够分析问题, 进行算法设计, 完成程序代码的编辑、运行, 并测试程序无误。 1.注意标点符号的中英文书写, 特别是括号、冒号的使用。 2.注意语句使用的灵活性。	总结本次实验问题
15M	五、思考拓展 1.数值运算 https://python123.io/student/courses/6318/groups/79292/problems/programmings/12353#pagetop 2.平方根格式化 https://python123.io/student/courses/6318/groups/79297/problems/programmings/13469#pagetop 3.连续质数计算 https://python123.io/student/courses/6318/groups/79302/problems/programmings/15293#pagetop	讲解及演示

s11:

授课时间	2022. 12. 12	课次	11
授课方式 (请打√)	理论课 <input type="checkbox"/> 讨论课 <input type="checkbox"/> 实验课 <input checked="" type="checkbox"/> 习题课 <input type="checkbox"/> 其他 <input type="checkbox"/>	课时 安排	2
授课题目 (教学章、节或主题): 实验 11 综合测试			
教学目的与要求: (1) 数值类型及操作运算。 (2) 字符串类型处理。 (3) 组合数据类型的处理。 (4) 控制结构常见语句及使用。 (5) 文件读写。 (6) 第三方库的安装及使用。			
教学重点及难点: python 语言综合运用。			
作业、讨论题、思考题: 习题 2.2			
课后小结: 通过这次课的学习, 熟练掌握配置 Python 开发的方法。			
下节课预习重点: Python 生态。			
参考文献: [1] 嵩天、礼欣、黄天羽. 《Python 语言程序设计基础(第 2 版)》. 高等教育出版社. 2017			

课时分配	教学内容	方法及手段
10M	<p>一、实验目的</p> <p>(1) 数值类型及操作运算</p> <p>(2) 字符串类型处理</p> <p>(3) 组合数据类型的处理</p> <p>(4) 控制结构常见语句及使用</p> <p>(5) 文件读写</p> <p>(6) 第三方库的安装及使用</p> <p><code>pip install -i https://pypi.douban.com/simple/ jieba</code></p>	讲解及演示
10M	<p>二、实验内容</p> <p>(1)打印九九乘法表</p> <p>(2) 数字不同数之和</p> <p>获得用户输入的一个整数 N，输出 N 中所出现不同数字的和。</p> <p>例如：用户输入 123123123，其中所出现的不同数字为：1、2、3，这几个数字和为 6。</p> <p>(3) Hello World 的条件输出</p> <p>获得用户输入的一个整数，参考该整数值，打印输出"Hello World"，要求：</p> <p>如果输入值是 0，直接输出"Hello World"</p> <p>如果输入值大于 0，以两个字符一行方式输出"Hello World"（空格也是字符）</p> <p>如果输入值小于 0，以垂直方式输出"Hello World"</p>	讲解及演示
20M	<p>思考：如果输入值大于 0，以每个字符后面添加 1 个空格方式输出"Hello"</p> <p>(4) 100 以内素数之和</p> <p>求 100 以内所有素数之和并输出。素数指从大于 1，且仅能被 1 和自己整除的整数。</p>	实验指导及同学演示
20M	<p>提示：可以逐一判断 100 以内每个数是否为素数，然后求和。</p>	同学演示

课时分配	教学内容	方法及手段						
15M	<p>三、实验过程</p> <p style="text-align: center;">指导过程中查阅并记录成绩</p> <p>四、总结</p> <p>能够分析问题，进行算法设计，完成程序代码的编辑、运行，并测试程序无误。</p> <p>1.注意标点符号的中英文书写，特别是括号、冒号的使用。</p> <p>2.注意语句使用的灵活性。</p>	总结本次实验问题						
15M	<p>五、思考拓展</p> <p>CSV 数据清洗。</p> <p>已知 CSV 格式文件 data.csv，提取数据进行如下格式转换：</p> <p>1) 按行进行倒序排列；</p> <p>2) 每行数据倒序排列；</p> <p>3) 使用分号 (;) 代替逗号 (,) 分割数据，无空格；</p> <p>按照上述要求转换后将数据输出。</p> <table border="1" data-bbox="341 1442 983 1630" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th style="width: 10%;"></th> <th style="width: 40%;">输入</th> <th style="width: 40%;">输出</th> </tr> </thead> <tbody> <tr> <td style="vertical-align: top;">示例 1</td> <td style="background-color: #f0f0f0;">(以下内容在文件中) 1,2,3 4,5,6 7,8,9</td> <td style="background-color: #f0f0f0;">9;8;7 6;5;4 3;2;1</td> </tr> </tbody> </table>		输入	输出	示例 1	(以下内容在文件中) 1,2,3 4,5,6 7,8,9	9;8;7 6;5;4 3;2;1	讲解及演示
	输入	输出						
示例 1	(以下内容在文件中) 1,2,3 4,5,6 7,8,9	9;8;7 6;5;4 3;2;1						

s12:

授课时间	2022. 12. 19	课次	12
授课方式 (请打√)	理论课 <input type="checkbox"/> 讨论课 <input type="checkbox"/> 实验课 <input checked="" type="checkbox"/> 习题课 <input type="checkbox"/> 其他 <input type="checkbox"/>	课时 安排	2
授课题目 (教学章、节或主题): 实验 12 Python 生态			
教学目的与要求: (1) numpy 库 (2)matplotlib 库 (3)wordcloud 库 (4)pygame 库			
教学重点及难点: 理解并掌握 python 语言第三方库的安装和使用过程			
作业、讨论题、思考题: 课后习题 9.2			
课后小结: 通过这次课的学习, 熟练掌握配置 Python 第三方库的安装和使用方法。			
下节课预习重点: 无			
参考文献: [1] 嵩天、礼欣、黄天羽. 《Python 语言程序设计基础(第 2 版)》. 高等教育出版社. 2017			

课时分配	教学内容	方法及手段
10M	<p>一、实验目的</p> <p>(1) numpy 库</p> <p>numpy 是 Python 的一种开源数值计算扩展第三方库，用于处理数据类型相同的多维数组。</p> <p>pandas 是基于 numpy 扩展的一个重要第三方库，pandas 提供两种最基本的数据类型：Series 和 DataFrame，分别代表一维数组和二维数组类型。</p> <p>(2)matplotlib 库</p> <p>数据可视化指根据数据特点将其展示为易于理解图形的过程。Python 语言在数据可视化方面具有较强的优势。使用这个库可以利用 Python 程序绘制超过 100 种数据可视化效果。</p> <p>(3)wordcloub 库</p> <p>词云图，也叫文字云，是对文本中出现频率较高的“关键词”予以视觉化的展现，词云图过滤掉大量的低频低质的文本信息，使得浏览者只要一眼扫过文本就可领略文本的主旨。在这种图中，词的形状代表了词频，其尺寸越大，代表出现频率越高。</p> <p>(4)pygame 库</p> <p>游戏开发是一个有趣的方向，在游戏逻辑和功能实现层面，Python 已经成为重要的支撑性语言。</p>	讲解及演示
10M	<p>二、实验内容</p> <p>(1)Jupyter-数据分析</p> <p>(2)matplotlib-画图</p> <p>(3)Wordcloud-红海行动 B 站弹幕</p> <p>(4)Pygame-超级玛丽</p>	讲解及演示
20M	<p>(5)PIL-趣味代码运动小人</p>	实验指导及同学演示
20M	<p>三、实验过程 (不需要提交结果)</p>	实验指导及同学演示

课时分配	教学内容	方法及手段
15M	<p style="text-align: center;">指导过程中查阅并记录成绩</p> <p>四、总结</p> <p>能够分析问题，进行算法设计，完成程序代码的编辑、运行，并测试程序无误。</p> <p>1.注意标点符号的中英文书写，特别是括号、冒号的使用。</p> <p>2.注意语句使用的灵活性。</p>	总结本次实验问题
15M	<p>五、思考拓展</p> <p>三国演义词云图：https://zhuanlan.zhihu.com/p/377144700</p>	讲解及演示