



信息工程系

教

案

课程名称： Python 编程基础

教 师： 林晓红

总 学 时： 54

理论学时： 18

实训学时： 36

上课班级： 物联网应用技术（三二分段）251

授课学期： 2025-2026 第一学期

| | | | |
|------------------|--|-------------|-------------|
| <p>课题名称</p> | <p>第 1 章 开启 Python 学习之旅</p> | <p>计划学时</p> | <p>3 学时</p> |
| <p>内容分析</p> | <p>在方兴未艾的机器学习以及热门的大数据分析技术领域，Python 语言的热度可谓是如日中天。Python 语言因简洁的语法、出色的开发效率以及强大的功能，迅速在多个领域占据一席之地，成为最符合人类期待的编程语言。</p> | | |
| <p>教学目标及基本要求</p> | <ol style="list-style-type: none"> 1. 了解 Python 的特点、版本以及应用领域 2. 熟悉 Python3 的下载与安装 3. 了解 PyCharm 的安装及简单使用 4. 了解代码规范，掌握变量的意义 5. 掌握 Python 的基本输入输出 | | |
| <p>教学重点</p> | <ol style="list-style-type: none"> 1. Python 中变量的使用 2. Python 基本输入输出 | | |
| <p>教学难点</p> | <ol style="list-style-type: none"> 1. Python 基本输入输出 | | |
| <p>教学方式</p> | <p>讲授，案例，操练</p> | | |
| <p>课程思政</p> | <p>在文字未产生时，中国古代就有“上古结绳而治”的说法（《周易·系辞》），这是一种古老的计数和记事方法，说明中国在人类文明发展史上的先进、智慧和创新。在阿拉伯数字出现前，中国古代就发明了“算盘”计算工具，还传播到了世界其他地区，对人类的计算方式和数学发展产生了深远的影响。</p> | | |
| <p>教学过程</p> | <p style="text-align: center;">理论课（1 学时）</p> <p>学习内容：Python 概述，搭建 Python 环境，编码规范，变量，基本输入输出，实例 1：海洋单位距离的换算，实例 2：打印名片。</p> <p>一、构造情境，引出 Python 概述以及搭建 Python 环境</p> <p style="padding-left: 2em;">（1）教师向学生介绍什么是 Python。</p> <p style="padding-left: 2em;">Python 是一种面向对象的解释型计算机程序设计语言，他最初由荷兰人吉多·范罗苏姆创造，并于 1991 年首次发行。</p> | | |

(2) 为什么要学习 Python。

Python 语言因简洁的语法、出色的开发效率以及强大的功能，迅速在多个领域占据一席之地，成为最符合人类期待的编程语言。

二、重点知识的讲解

1. Python 概述

(1) 介绍 Python 的主要特点

Python 语言之所以能够迅速发展，受到程序员的青睐，与它具有的特点密不可分，Python 的特点可以归纳为简单易学、免费开源、可移植性、面向对象、丰富的库。

(2) 介绍 Python 的版本区别

Python 主要分为 Python2 与 Python3 两个版本，相较于 Python2 版本 Python3 经历了较大的变革，为了不带入过多的累赘，Python3 在设计之初没有考虑向下兼容，因此许多使用 Python2 设计的程序都无法在 Python3 上正常执行。其版本区别主要有：`print()`函数替代了 `print` 语句；Python3 默认使用 UTF-8 编码；除法运算的使用；异常的使用。

(3) 介绍 Python 应用领域

Python 作为一门功能强大的且简单易学的编程语言在实际开发中得打了广泛的使用，其主要应用领域包括：Web 开发、科学计算与数据分析、自动化运维、网络爬虫、游戏开发、人工智能。

(4) 介绍 Python3 的安装

首先在 Python 官方网站中 <http://www.python.org/>选择 windows 系统，然后根据教学需求选择对应的 Python 解释器版本。

(5) Python 自带 IDLE 的使用。

Python 在安装过程中默认自动安装了 IDLE，IDLE 是 Python 自带的集成开发环境，在 IDLE 中可通过交互式与文件式编写 Python 程序。

(6) 集成开发环境 PyCharm 的安装与使用。

PyCharm 是一款 Python 集成开发环境，PyCharm 包含智能提示、自动导入、智能代码编辑器等功能。

2. Python 程序

(1) 编写第一个 Python 程序

教师演示编写第一个 Python 程序——模拟手机充值,通过模拟手机充值程序带领学生了解 Python 的编写程序的方法。

(2) Python 中编程约定

良好的编程习惯不仅有良好的可读性,还有助于程序的调试与维护。在 Python 中编程约定包括:每个缩进使用 4 个空格、每行的最大长度不超过 79 个字符、合理使用代码注释、正确的命名规范。

(3) Python 变量的使用

Python 程序运行的过程中随时可能产生一些临时数据,应用程序会将这些数据保存在内存单元中,并使用不同的标识符来标识各个内存单元。这些具有不同标识、存储临时数据的内存单元称为变量,标识内存单元的符则为变量名(亦称标识符),内存单元中存储的数据就是变量的值。

变量的定义格式为:变量名 = 数据。

(4) Python 的应用领域

Python 具有简单易学、类库丰富、通用灵活等特点,因此常用于 Web 开发、科学计算、游戏开发、自动化运维、多媒体应用、爬虫开发。

(5) Python 的基本输入输出函数

程序要实现人机交互功能,需能够向显示设备输出有关信息及提示,同时也要能够接收从键盘输入的数据。Python 提供了用于实现输入/输出功能的函数 `input()`和 `print()`。

(6) 实例

实例 1: 海洋单位距离的换算。

实例 2: 打印名片。

上机实验课 (2 学时)

上机练习主要针对本章中需要重点掌握的知识点,以及在程序中容易出错的内容进行练习,通过上机练习可以考察同学对知识点的掌握情况,对本章知识掌握程度。

形式: 独立完成

| | |
|-------|--|
| | <p>上机一：</p> <p>请按照以下要求开始上机课：</p> <p>要求如下：</p> <ul style="list-style-type: none">● 掌握 IDLE、PyCharm、Jupyter Notebook 的安装与使用● 掌握变量、print()、input()函数的使用● 实现实例 1 与实例 2 |
| 考题和习题 | 见教材第 1 章配套的习题 |
| 教学后记 | |

| | | | |
|------------------|---|-------------|--------------|
| <p>课题名称</p> | <p>第 2 章 数字类型与字符串</p> | <p>计划学时</p> | <p>12 学时</p> |
| <p>内容分析</p> | <p>数字类型和字符串是 Python 程序中基本的数据类型，其中数字类型分为整型、浮点型、复数类型、布尔类型，可通过运算符进行各种数学运算。本章将会对数字类型、字符串和运算符进行讲解，并通过实例带领大家掌握它们的使用方法。</p> | | |
| <p>教学目标及基本要求</p> | <ol style="list-style-type: none"> 1. 了解数字类型的表示方法 2. 掌握类型转换函数 3. 掌握字符串的格式化输出 4. 掌握字符串的常见操作 5. 掌握字符串的索引与切片 6. 熟练使用运算符，明确混合运算中运算符的优先级 | | |
| <p>教学重点</p> | <ol style="list-style-type: none"> 1. 字符串的格式化输出 2. 字符串的常见操作 | | |
| <p>教学难点</p> | <ol style="list-style-type: none"> 1. 字符串的格式化输出 2. 字符串的常见操作 3. 位运算符 4. 运算符优先级 | | |
| <p>教学方式</p> | <p>讲授，案例，操练</p> | | |
| <p>课程思政</p> | <p>在南北朝时期，祖冲之在没有任何先进科学仪器的情况下，将圆周率精确到小数点后七位，即 3.1415926，这一成就领先当时西方将近一千年左右，展示了中国古代数学家的高超智慧和计算能力。</p> <p>在编程过程中要追求精益求精的工匠精神、创新精神和探索欲望，敢于挑战自我、超越自我。</p> | | |

理论课（4 学时）

学习内容：数字类型的表示方法，实例 1，类型转换函数，实例 2；字符串的定义；字符串的格式化输出；字符串的常见操作；字符串的索引与切片；运算符的使用；对经典实例的讲解。

一、Python 中数字类型

(1) 通过提出需求引出 Python 中的数字类型

教师根据数学中的数字类型引出 Python 中的数字类型。

(2) 明确学习目标。

二、重点知识讲解

(1) Python 中的数字类型及其表示方法。

表示数字或数值的数据类型称为数字类型。Python 内置的数字类型有整型（int）、浮点型（float）、复数类型（complex）、布尔类型（bool）。

(2) 实例 1：根据身高体重计算 BMI 指数。

教师根据教材带领学生完成实例 1：根据身高体重计算 BMI 指数，讲解该实例中数字类型的使用。

(3) 类型转化函数。

Python 内置了一系列可实现强制类型转换的函数，保证用户在有需求的情况下，将目标数据转换为指定的类型。数字间进行转换的函数有 int()、float()、str()。

(4) 实例 2：模拟超市收银抹零行为。

教师根据教材带领学生完成实例 2：模拟超市收银抹零行为，讲解该实例中数字类型的使用。

(5) 字符串的定义。

字符串是一种用来表示文本的数据类型，它是由符号或者数值组成的一个连续序列，Python 中的字符串是不可变的，字符串一旦创建便不可修改。Python 支持使用单引号、双引号和三引号定义字符串，其中单引号和双引号通常用于定义单行字符串，三引号通常用于定义多行字符串，具体示例分别如下。

(6) 字符串的格式化输出。

字符串格式化输出的方式有 3 种分别为：占位符%、format()方法和 f-strings

三种方式实现格式化输出。

(7) 字符串的常见操作。

在 Python 中提供多种字符串操作，常见的操作有使用“+”进行字符串拼接、使用 `replace()` 方法替换、使用 `split()` 方法分隔字符串、使用 `strip()` 方法去除两侧空格。

(8) 字符串的索引与切片。

在程序的开发过程中，可能需要对一组字符串中的某些字符进行特定的操作，Python 中通过字符串的索引与切片功能可以提取字符串中的特定字符或子串。

字符串的正向索引从 0 开始计算，反向索引从 -1 开始计数。

字符串的切片语法格式为：字符串[起始:结束:步长]。

(9) 实例 3：文本进度条。

教师根据教学资源实现文本进度条实例，并向学生讲解其实现方式。

(10) 实例 4：敏感词替换。

教师根据教学资源实现敏感词替换实例，并向学生讲解其实现方式。

(11) 算术运算符的使用。

Python 中的算术运算符包括 +、-、*、/、//、% 和 **，这些运算符都是双目运算符，一个运算符可以和两个操作数组成一个表达式。

(12) 比较运算符的使用。

Python 中的比较运算符有：==、!=、>、<、>=、<=，比较运算符同样是双目运算符，它与两个操作数构成一个表达式。

(13) 赋值运算符的使用。

赋值运算符的功能是：将一个表达式或对象赋给一个左值，其中左值必须是一个可修改的值，不能为一个常量。“=”是基本的赋值运算符，此外“=”可与算术运算符组合成复合赋值运算符。Python 中的复合赋值运算符有：+=、-=、*=、/=、//=、%=、**=。

(14) 逻辑运算符的使用。

Python 支持逻辑运算，但 Python 逻辑运算符的功能与其它语言有所不同。Python 中分别使用“or”、“and”、“not”这三个关键字作为逻辑运算“或”、“与”、“非”的运算符，其中 or 与 and 为双目运算符，not 为单目运算符。

| | |
|--------------|---|
| | <p>(15) 位运算符的使用。</p> <p>程序中的所有数据在计算机内存中都以二进制形式存储，位运算即以二进制位为单位进行的运算。Python 的位运算主要包括按位左移、按位右移、按位与、按位或、按位异或、按位取反这 6 种。</p> <p>(16) 运算符优先级。</p> <p>Python 支持使用多个不同的运算符连接简单表达式，实现相对复杂的功能，为了避免含有多个运算符的表达式出现歧义，Python 为每种运算符都设定了优先级（具体详见表 2-7）。</p> <p>(17) 经典实例讲解</p> <p>了解实例判断水仙花数、找出最大数、计算三角形面积、下载操作模拟的实现过程。</p> <p style="text-align: center;">上机实验课（8 学时）</p> <p>上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对本章知识掌握程度。</p> <p>形式：独立完成</p> <p>请按照以下要求开始上机课：</p> <p>要求如下：</p> <ul style="list-style-type: none"> ● 练习 2.1-2.2 小节中的教材代码 ● 实现实例 1——实例 4 ● 练习 2.3 小节中教材代码 ● 实现 2.4 小节中实例 5——实例 8 代码 |
| <p>考题和习题</p> | <p>见教材第 2 章配套的习题</p> |

| | |
|------------------|--|
| 教 学 后 记 | |
|------------------|--|

| | | | |
|------------------|---|-------------|--------------|
| <p>课题名称</p> | <p>第 3 章 流程控制</p> | <p>计划学时</p> | <p>15 学时</p> |
| <p>内容分析</p> | <p>程序中的语句默认自上而下顺序执行。流程控制意指在程序执行时，通过一些特定的指令更改程序中语句的执行顺序，使程序产生跳跃、回溯等现象。</p> | | |
| <p>教学目标及基本要求</p> | <ol style="list-style-type: none"> 1. 掌握 if 语句的多种格式 2. 熟练使用 if 语句的嵌套 3. 掌握 for 循环与 while 循环的使用 4. 熟悉 for 循环与 while 循环嵌套 5. 掌握 break 与 continue 语句的使用 | | |
| <p>教学重点</p> | <ol style="list-style-type: none"> 1. if 语句的格式 2. for 循环 3. while 循环 | | |
| <p>教学难点</p> | <ol style="list-style-type: none"> 1. if 语句嵌套 2. while 循环嵌套 3. for 循环嵌套 | | |
| <p>教学方式</p> | <p>讲授，案例，操练</p> | | |
| <p>课程思政</p> | <p>程序执行是顺序的、不可跳跃的，顺序结构需要严格遵守语法规则和逻辑顺序；选择结构中的条件判断需要严格遵守语法规则和逻辑规则；任何错误或遗漏都可能导致程序无法正常运行。循环结构中的重复执行过程象征着坚持不懈的毅力。引导学生认识到在社会生活中也需要遵守各种规则和制度，如法律法规、校规校纪等，增强学生的规则意识和法律意识；强调逻辑思维能力和批判性思维能力，增强学生的社会责任感和道德感，能够坚守正确的价值观和道德标准。强调在面对困难和挑战时，要像程序中的循环结构一样，要有严谨的态度，并持之以恒、不断尝试，坚持不懈，直到达到目标。</p> | | |
| <p>教学过程</p> | <p style="text-align: center;">理论课（5 学时）</p> <p>教学内容： if 语句，实例 1-实例 3；if 语句的嵌套，实例 4：模拟乘客进站流程，实例 5：快递计算系统；for 循环，实例 6：数据加密；实例 7：逢七拍手游戏，while 循环，实例 8：登录系统账号检测；while 循环嵌套，for 循环嵌套，实例 9：九九乘法表，break 语句，</p> | | |

continue 语句，实例 10：猜数游戏。

一、创设情境，导入 if 语句格式

(1) 通过提出需求引出 if 语句的使用场景。

教师可根据生活事例引出 if 语句的使用场景，例如：用户登录的时候需要判断用户的用户名与密码是否全部正确，进而决定用户是否能够成功登录。类似这样的需求功能，都可以使用 if 语句实现。

二、重点知识的讲解

3.1 if 语句

(1) if 语句的使用。

if 语句是最简单的判断语句，它由三部分组成，分别是 if 关键字、条件表达式以及代码块。if 语句根据条件表达式的判断结果选择是否执行相应的代码块，其格式如下：

if 条件表达式:

 代码块

上述格式中，if 关键字可以理解为“如果”，当条件表达式的值为 True 时，则执行代码块。

(2) if-else 语句的使用。

if-else 语句产生两个分支，可根据条件表达式的判断结果选择执行哪一个分支。if-else 语句格式如下：

if 条件表达式:

 代码块 1

else:

 代码块 2

上述格式中，如果 if 条件表达式结果为 True，执行代码块 1；如果条件表达式结果为 False，则执行代码块 2。

(3) if-elif-else 语句的使用。

if-else 语句可以处理两种情况，如果程序需要处理多种情况，那么可以使用 if-elif-else 语句，其语句格式如下：

if 条件表达式 1:

 代码块 1

elif 条件表达式 2:

 代码块 2

elif 条件表达式 n-1:

 代码块 n-1

else:

 代码块 n

上述格式中，if 之后可以有任意数量的 elif 语句，如果条件表达式 1 的结果为 True，那么执行代码块 1，如果条件表达式 2 的结果为 True，那么执行代码块 2，以此类推，如果 else 前面的条件表达式结果都为 False，那么执行代码块 n。

(4) 完成教材中实例 1-实例 3。

教师根据教材资源完成实例 1：判断 4 位回文数、实例 2：奖金发放、实例 3：根据身高体重计算某人的 BMI 值，并给学生讲解其实现过程。

3.2 if 语句的嵌套

(1) 什么是 if 语句嵌套？

if 语句嵌套指的是 if 语句内部包含 if 语句，其格式如下：

if 条件表达式 1:

 代码块 1

 if 条件表达式 2:

 代码块 2

上述 if 语句嵌套的格式中，先判断外层 if 语句中条件表达式 1 的结果是否为 True，如果结果为 True，那么执行代码块 1，再判断内层 if 的条件表达式 2 的结果是否为 True，如果条件表达式 2 的结果为 True，那么执行代码块 2。

(2) 讲解实例 4：模拟乘客进站流程。

教师根据教材资源实现实例 4：模拟乘客进站流程，并向学生讲解其实现过程。

(3) 讲解实例 5：快递计算系统。

教师根据教材资源实现实例 5：快递计算系统，并向学生讲解其实现过程。

3.3 循环语句

(1) Python 中 for 循环的使用。

for 循环可以对可迭代对象进行遍历，for 循环的语法格式如下：

for 临时变量 in 可迭代对象:

 执行语句 1

 执行语句 2

每执行一次循环，临时变量都会被赋值为可迭代对象的当前元素，提供给执行语句使用。

(2) 讲解实例 6：数据加密。

教师根据教材资源实现 6：数据加密，并向学生讲解其实现过程。

(3) 讲解实例 7：逢七拍手案例。

教师根据教材资源实现实例 7：逢七拍手，并向学生讲解其实现过程。

(4) 讲述 while 循环的使用。

while 循环是一个条件循环语句，当条件满足时重复执行代码块，直到条件不满足为止，其语法格式如下：

while 条件表达式:

 代码块

上述格式中，首先判断条件表达式的结果是否为 True，如果条件表达式的结果为 True，那么执行 while 循环中的代码块，然后再次判断条件表达式的结果是否为 True，如果条件表达式的结果为 True，那么再次执行 while 循环中的代码块。每次执行完代码块都需要重新判断条件表达式的结果，直到条件表达式的结果为 False 时结束循环，不再执行 while 循环中的代码块。

(5) 讲解实例 8：登录系统账号检测。

教师根据教材资源实现实例 8：登录系统账号检测，并向学生讲解其实现过程。

3.4 循环嵌套

(1) while 循环嵌套的使用。

在编写代码时，可能需要对一段代码执行多次，这时可以使用循环语句，假

设需要多次执行这个循环语句，那么可以将循环语句放在循环语句中，实现循环嵌套。

while 循环中可以嵌套 while 循环，其格式如下：

while 条件表达式 1:

 代码块 1

 while 条件表达式 2:

 代码块 2

(2) for 循环嵌套的使用。

for 循环也可以嵌套使用，其语法格式如下：

for 临时变量 in 可迭代对象:

 代码块 1

 for 临时变量 in 可迭代对象:

 代码块 2

for 循环嵌套语句与 while 循环嵌套语句大同小异，都是先执行外层循环，再执行内层循环，每执行一次外层循环都要执行一遍内层循环。

(3) 讲解实例 9：九九乘法表。

教师根据教材资源实现实例 9：九九乘法表，并向学生讲解其实现过程。

(4) break 语句的使用。

break 语句用于跳出离它最近一级的循环，能够用于 for 循环和 while 循环中，通常与 if 语句结合使用，放在 if 语句代码中，其格式如下：

for 临时变量 in 可迭代对象:

 执行语句

 if 条件表达式:

 代码块

 break

(5) continue 语句的使用。

continue 语句用于跳出当前循环，继续执行下一次循环。当执行到 continue 语句时，程序会忽略当前循环中的剩余代码，重新开始执行下一次循环。

(6) 讲解实例 10：猜数游戏。

教师根据教材资源实现实例 10：猜数游戏，并向学生讲解其实现过程。

上机实验课（10 学时）

上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对代码的熟练程度。

上机一：（练习教材中 if 语句）

形式：独立完成

要求：

- (1) 要求学生能够使用 if 语句。
- (2) 要求学生能够自己实现实例 1-实例 3 程序。

上机二：（练习教材中 if 语句的嵌套）

形式：独立完成

要求：

- (1) 要求学生能够使用 if 语句的嵌套。
- (2) 要求学生能够自己实现实例 4-实例 5 程序。

上机三：（练习 for、while 循环语句）

形式：独立完成

要求：

- (1) 要求学生能够使用 for、while 循环语句。
- (2) 要求学生能够自己实现实例 6-实例 8 程序。

上机四：（练习 for 循环嵌套、while 循环嵌套、break 语句、continue 语句）

形式：独立完成

要求：

- (1) 要求学生能够使用 for 循环嵌套、while 循环嵌套、break 语句、continue 语句。
- (2) 要求学生能够自己实现实例 9-实例 10 程序。

| | |
|------------------|---------------|
| 思考题 和习题 | 见教材第 3 章配套的习题 |
| 教 学 后 记 | |

| | | | |
|-----------|--|------|------|
| 课题名称 | 第 4 章 列表与元组 | 计划学时 | 9 学时 |
| 内容分析 | 列表和元组是 Python 内置的两种重要的数据类型，他们都是序列类型，可以存放任何类型的数据，并且支持索引、切片、遍历等一系列操作。 | | |
| 教学目标及基本要求 | <ol style="list-style-type: none"> 1. 掌握列表的创建与访问列表元素的方式 2. 掌握列表的遍历和排序 3. 掌握添加、删除、修改列表元素的方式 4. 熟悉嵌套列表的使用 5. 掌握元组的创建与访问元组元素的方式 | | |
| 教学重点 | <ol style="list-style-type: none"> 1. 访问列表元素 2. 列表的遍历 3. 添加列表元素 4. 删除列表元素 5. 修改列表元素 | | |
| 教学难点 | <ol style="list-style-type: none"> 1. 列表的遍历 2. 列表的排序 3. 嵌套列表的创建与元素的访问 | | |
| 教学方式 | 讲授，案例，操练 | | |
| 课程思政 | <p>我国自古以来是一个多民族国家。中华人民共和国成立后，经中央人民政府调查与统计正式确认的民族共有 56 个。汉族是我国的主体民族，约占全国人口总数的 91.11%，其他还有 55 个民族，约占 8.89%。汉族和 55 个少数民族共同组成伟大的中华民族。利用列表来存储我国 56 个民族，并进行一系列的操作。</p> <p>利用列表实现社会主义 24 字核心价值观的输入和输出。</p> | | |
| 教学过程 | <p style="text-align: center;">理论课（3 学时）</p> <p>学习内容：列表的创建方式、访问列表元素、实例 1：刮刮乐、列表的遍历；列表排序、实例 2：商品价格区间设置与排序、添加删除和修改列表元素、实例 3：好友系统；嵌套列表的创建与元素访问、实例 4：随机分配办公室、元组的创建方式、访问元组元素、实例 5：中文数字对照表。</p> | | |

一、创设情境，引出列表

列表是 Python 中最灵活的有序序列，它可以存储任意类型的元素，开发人员可以对列表中的元素进行添加、删除、修改等操作。

二、知识讲解

(1) 列表的创建方式。

Python 中创建列表可以使用“[]”或 list()函数创建，

例如：

list_one = [] 使用“[]”创建列表

list_two = list() 使用 list()函数创建列表

(2) 如何访问列表元素。

列表中的元素可以通过索引或切片的方式访问，例如：

a.使用索引方式访问列表元素

```
list_01 = [1,2,3,4]      list_01[1]
```

b.使用切片方式访问列表元素

```
list_01[1:1:1]
```

(3) 实例 1:刮刮乐。

(4) 列表的遍历方式。

Python 中列表的遍历主要通过 for 循环来实现，例如：for i in 列表:pass

(5) 列表的排序。

列表的排序是将元素按照某种规定进行排列。列表中常用的排序方法有 sort()、reverse()、sorted()。sort()方法能够对列表元素排序，其语法格式为 sort(key=None,reverse=False)；sorted()方法用于将列表元素升序排列，该方法的返回值是升序排列后的新列表；reverse()方法用于将列表中的元素倒序排列，即把原列表中的元素从右至左依次排列存放。

(6) 实例 2:商品价格区间设置与排序。

(7) 添加列表元素的方法。

向列表中添加元素的常用方法有 append()、extend()和 insert()，append()方法用于在列表末尾添加新的元素；extend()方法用于在列表末尾一次性添加另一个

序列中的所有元素；`insert()`方法用于将元素插入列表的指定位置。

(8) 删除列表元素的方法。

删除列表元素的常用方式有 `del` 语句、`remove()`、`pop()`，`del` 语句用于删除列表中的指定位置的元素；`remove()`方法用于移除列表中的某个元素，若列表中有多个匹配结果，只会移除匹配到的第一个元素；`pop()`用于移除列表中的某个元素，如果不指定具体元素，那么移除列表中的最后一个元素。

(9) 修改列表的方法。

修改列表中的元素就是通过索引获取元素并对该元素重新赋值。

(10) 实例 3: 好友管理系统。

(11) 嵌套列表的创建与元素的访问。

列表可以存储任何元素，当然也可以存储列表，如果列表存储的元素也是列表，则称为嵌套列表。

(12) 实例 4: 随机分配办公室。

(13) 元组的创建方式。

元组的创建方式与列表的创建方式相似，元组可以通过圆括号“()”或内置的 `tuple()`函数快速创建，例如：

`tuple_01 = ()` 使用“()”创建元组

`tuple_02 = tuple()` 使用 `tuple()`函数创建元组

(14) 访问元组元素。

可以通过索引或切片的方式来访问元组中的元素，例如：

`tuple_test = (1,2,3,4)`

`tuple_test[0]` 使用索引访问元素

`tuple_test[1:2]` 使用切片访问元素

(15) 实例 5: 中文数字对照表。

上机实验课（6 学时）

上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对本章知识掌

| | |
|------------------|--|
| | <p>握程度。</p> <p>形式：独立完成</p> <p>请按照以下要求开始上机课：</p> <ul style="list-style-type: none">(1) 要求学生能够熟练掌握教材中示例代码。(2) 要求学生能够自己实现实例 1~实例 5 程序。 |
| 思考题和习题 | 见教材第 4 章配套的习题 |
| 教 学 后 记 | |

| | | | |
|------------------|--|-------------|-------------|
| <p>课题名称</p> | <p>第 5 章 字典和集合</p> | <p>计划学时</p> | <p>6 学时</p> |
| <p>内容分析</p> | <p>Python 中的组合类型包括序列类型、集合类型和映射类型，其中序列类型主要包括字符串、元素和列表；集合类型是一个无序组合，它的概念和数学中的集合类似；映射类型是“键-值”数据项的组合，主要以字典体现。</p> | | |
| <p>教学目标及基本要求</p> | <ol style="list-style-type: none"> 1. 掌握字典的创建和访问元素的方式 2. 掌握字典的基本操作 3. 掌握集合的创建和常见操作 4. 了解集合操作符的使用 | | |
| <p>教学重点</p> | <ol style="list-style-type: none"> 1. 通过“键”访问字典 2. 字典元素的添加和修改 3. 字典元素的删除 4. 字典元素的查询 | | |
| <p>教学难点</p> | <ol style="list-style-type: none"> 1. 字典元素的添加和修改 2. 字典元素的删除 3. 字典元素的查询 4. 集合类型的操作符 | | |
| <p>教学方式</p> | <p>讲授，案例，操练</p> | | |
| <p>课程思政</p> | <p>中国幅员辽阔，陆地面积大约为 960 万平方公里，共有 34 个省级行政区，包括 23 个省、5 个自治区、4 个直辖市、2 个特别行政区，每个省级行政区都有一个省名和一个省会城市。利用字典结构来处理省份和对应的省会城市的关系。</p> | | |
| <p>教学过程</p> | <p style="text-align: center;">理论课（2 学时）</p> <p>学习内容：字典的创建方式、通过“键”访问字典、实例 1：单词识别；字典元素的添加和修改、字典元素的删除、字典元素的查询、实例 2：手机通讯录；集合的创建方式、集合元素的添加删除和清空、集合类型的操作符、实例 3：生词本。</p> <p>一、创设情境，引出字典</p> <p>某些数据是以“键-值”形式存在，在 Python 中该如何保存这些数据呢？</p> | | |

Python 中的字典提供以“键-值”形式保存数据。

二、知识讲解

(1) 列表的创建方式。

Python 中创建字典可以使用“{}”或 list()函数创建，

例如：

`dict_one = {}` 使用“{}”创建列表

`dict_two = dict()` 使用 dict()函数创建列表

(2) 通过“键”访问字典。

因为字典中的键是唯一的，所以可以通过键获取对应的值。例如：

```
color_dict = {'purple': '紫色', 'green': '绿色', 'black': '黑色'}
```

```
color_dict['purple']  获取键为 purple 对应的值“紫色”。
```

(3) 实例 1：单词识别。

(4) 字典元素的添加和修改。

字典支持使用 update()方法或通过指定的键添加元素或修改元素；可以通过字典中的键对元素进行修改。

(5) 字典元素的删除。

删除字典中元素的方法有 pop()、popitem()、clear()，其中 pop()方法可根据指定键值删除字典中的指定元素，若删除成功，该方法返回目标元素的值；popitem()方法可以随机删除字典中的元素；clear()方法用于清空字典中的元素。

(6) 字典元素的查询。

字典中 items()方法会返回一个 dict_items 对象，该对象支持迭代操作，通过 for 循环遍历 dict_items 对象中的数据并以 (key, value) 的形式显示；keys()方法可以查看字典中所有的键；values()方法返回字典中所有的值。

(7) 实例 2：手机通讯录。

(8) 什么是集合。

Python 中集合可分为可变集合与不可变集合，可变集合由 set()函数创建，集合中的元素可以动态地增加或删除；不可变集合由 frozenset()函数创建，集合中

| | |
|---------------|--|
| | <p>的元素不可改变。</p> <p>(9) 集合的创建方式。</p> <p> 可变集合的创建语法格式：<code>set([iterable])</code></p> <p> 不可变集合的创建语法格式：<code>frozenset([iterable])</code></p> <p> 其中，<code>iterable</code> 是一个可迭代对象，返回值是 <code>set</code> 或 <code>frozenset</code> 对象。</p> <p>(10) 集合元素的添加删除和清空。</p> <p> Python 中可变集合支持添加、删除和情况元素，可变集合的 <code>add()</code> 方法或 <code>update()</code> 方法都可以实现向集合中添加元素，不同的是，<code>add()</code> 方法只能添加一个元素，而 <code>update()</code> 方法可以添加多个元素；<code>remove()</code>、<code>discard()</code>、<code>pop()</code> 方法删除可变集合中的元素；<code>clear()</code> 方法可以清空集合中的元素。</p> <p>(11) 集合类型的操作符。</p> <p> Python 支持通过操作符 <code> </code>、<code>&</code>、<code>-</code>、<code>^</code> 对集合进行联合、取交集、差补和对称差分操作。</p> <p>(12) 实例 3:生词本。</p> <p style="text-align: center;">上机实验课（4 学时）</p> <p> 上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对代码的熟练程度。</p> <p> 上机一：（练习教材中示例代码以及实例代码）</p> <p> 形式：独立完成</p> <p> 要求：</p> <p> (1) 要求学生能够熟练掌握教材中示例代码。</p> <p> (2) 要求学生能够自己实现实例 1-实例 3 程序。</p> |
| <p>思考题和习题</p> | <p>见教材第 5 章配套的习题</p> |

| | |
|------------------|--|
| 教 学 后 记 | |
|------------------|--|

| 课题名称 | 第 6 章 函数 | 计划学时 | 6 学时 |
|-----------|---|------|------|
| 内容分析 | 函数被封装起来的、实现某种功能的一段代码，它可以被其他函数调用。当程序实现的功能较为复杂时，开发人员通常会提取其中的功能性代码模块化为一个函数，提高代码复用性、降低代码冗余、使程序结构更加清晰。 | | |
| 教学目标及基本要求 | <ol style="list-style-type: none">1. 掌握函数的定义与调用2. 掌握函数的参数传递方式3. 掌握局部变量和全局变量的使用4. 熟悉匿名函数与递归函数的使用5. 了解常用的内置函数 | | |
| 教学重点 | <ol style="list-style-type: none">1. 位置参数2. 关键字参数3. 默认参数4. 不定长参数 | | |
| 教学难点 | <ol style="list-style-type: none">1. 位置参数2. 关键字参数3. 默认参数4. 不定长参数5. 全局变量6. 匿名函数7. 递归函数 | | |
| 教学方式 | 讲授，案例，操练 | | |
| 课程思政 | 函数定义时需要遵循一定的语法规则，如函数名、参数列表、返回类型等都需要明确指定。在调用函数时，也需要按照规定的参数类型和数量传递实参。引导学生认识到在编程中需要严格遵守规则，任何违反规则的行为都可能导致程序错误。培养学生的规则意识、逻辑思维、团队协作和创新精神，保持高度的责任心和敬业精神。 | | |

教
学
过
程

理论课（2 学时）

学习内容：函数的定义、函数的调用、实例 1：计算器；位置参数、关键字参数、默认参数、不定长参数；局部变量、全局变量、实例 2：学生信息管理系统；匿名函数、递归函数、实例 3：汉诺塔、实例 4：斐波那契数列、Python 常用内置函数。

一、创设情境，引出字典

函数是指被封装起来的、实现某种功能的一段代码，它可以被其他函数调用。

二、知识讲解

（5）函数的定义方式。

在 Python 中，使用关键字 def 定义函数，其语法格式为：

def 函数名([参数列表]):

 ["函数文档字符串"]

 函数体

 [return 语句]

函数以 def 关键字开头，其后跟函数名和圆括号；函数名用于标识函数的名称，遵循标识符的命名规则；参数列表用于接收传入函数中的数据，可以为空；冒号用于表示函数体的开始；函数文档字符串一对由三引号包含的字符串，是函数的说明信息，可以省略；函数体实现函数功能的具体代码；return 语句用于将函数的处理结果返回给函数调用者，若函数没有返回值，return 语句可以省略。

（6）函数的调用。

函数调用格式为：函数名([参数列表])，定义好的函数直到被程序调用后才会执行。

（7）实例 1：计算器。

（8）什么是函数的参数传递。

函数的参数传递是指将实际参数传递给形式参数的过程，根据不同的传递形式，函数的参数可分为位置参数、关键字参数、默认值参数、不定长参数。

(9) 什么是位置参数。

调用函数时，编译器会将函数的实际参数按照位置顺序依次传递给形式参数，即将第 1 个实际参数传递给第 1 个形式参数，将第 2 个实际参数传递给第 2 个形式参数，以此类推。

(10) 什么是关键字参数。

关键字参数传递通过“形式参数=实际参数”的格式将实际参数与形式参数相关联。

(11) 什么是默认参数。

定义函数时可以指定形式参数的默认值，调用函数时，若没有给带有默认值的形式参数传值，直接使用参数的默认值；若给带有默认值的形式参数传值，实际参数的值会覆盖默认值。

(12) 什么是不定长参数。

若要传入函数中的参数的个数不确定，可以使用不定长参数。不定长参数也称可变参数，此种参数接收参数的数量可以任意改变。

(13) 什么是变量作用域。

变量的作用域是指变量的作用范围。根据作用范围，Python 中的变量分为局部变量与全局变量。

(14) 局部变量。

局部变量指的是在函数内定义的变量，只在定义它的函数内生效。

(15) 全局变量。

全局变量指的是在函数外定义的变量，它在程序中任何位置都可以被访问。

(16) 实例 2：学生信息管理系统。

(17) 什么是匿名函数。

匿名函数是无需函数名标识的函数，它的函数体只能是单个表达式。Python 中使用关键字 `lambda` 定义匿名函数，其格式为 `lambda [arg1 [,arg2,.....argn]]:expression`，其中“`[arg1 [,arg2,.....argn]]`”表示匿名函数的参数，“`expression`”是一个表达式。

(18) 什么是递归函数。

| | |
|----------------------------|---|
| | <p>递归是一个函数过程在定义或说明中直接或间接调用自身的一种方法，它通常把一个大型的复杂问题层层转化为一个与原问题相似，但规模较小的问题进行求解。如果一个函数中调用函数本身，这个函数就是递归函数。</p> <p>(19) 实例 3: 汉诺塔。</p> <p>(20) 实例 4: 斐波那契数列。</p> <p>(21) Python 常用的内置函数。</p> <p>Python 内置了一些实现特定功能的函数，这些函数无需由 Python 使用者重新定义，可直接使用。如 abs()、len()、map()、help()、ord()、chr()等。</p> <p style="text-align: center;">上机实验课（4 学时）</p> <p>上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对代码的熟练程度。</p> <p>上机一：（练习教材 6.1-6.3 示例代码以及实例 1 至实例 2）</p> <p>形式：独立完成</p> <p>要求：</p> <p>(3) 要求学生能够熟练掌握教材中示例代码。</p> <p>(4) 要求学生能够自己实现实例 1~实例 4 程序。</p> <p>(5) 要求学生掌握 Python 常用内置函数的使用。</p> |
| <p>思考题和习题</p> | <p>见教材第 6 章配套的习题</p> |
| <p>教 学 后 记</p> | |