



21世纪高等学校计算机
应用技术系列教材

Python程序设计

(思政版·第2版)

微课视频版



◎ 王巍 王书序 郭小磊 编撰 刘小洋 编著

《教学实验》

《实训案例》

600
分钟讲解

《习题解答》

《思政元素》

《大纲解读》

《程序源码》

《知识导图》

《配套资源》

清华大学出版社

第5章

函数

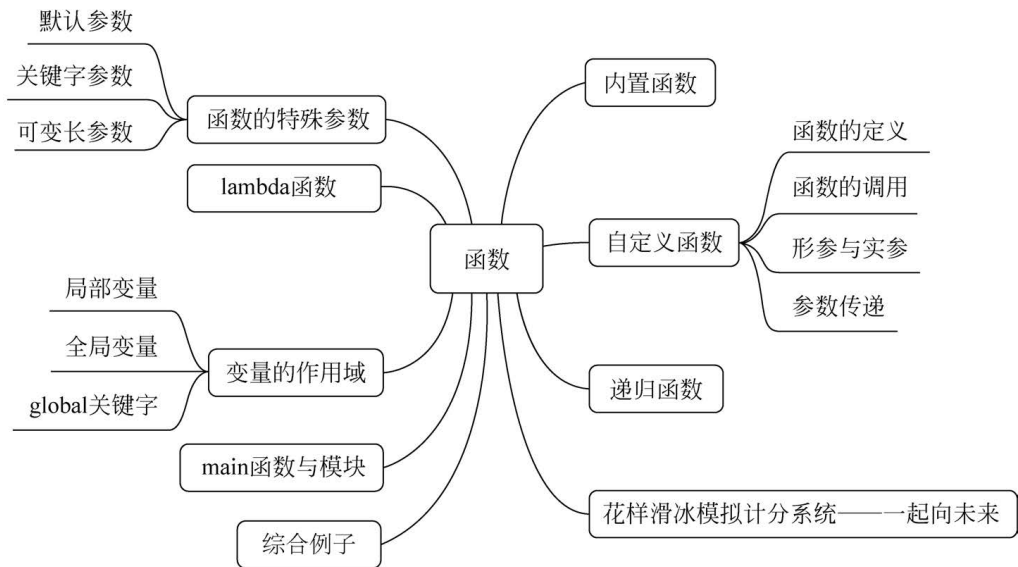
能力目标

【应知】 理解通过函数实现模块化编程思想。

【应会】 掌握函数定义和调用的方法,掌握函数参数传递机制,掌握默认参数、可变长参数和关键字参数,掌握局部变量和全局变量,掌握 lambda 函数和递归函数。

【难点】 函数的参数、lambda 函数和递归函数。

知识导图



在设计较复杂的程序时,一般采用自顶向下的方法,先将复杂问题划分为几个部分,再对各个部分进行细化,直到分解为能解决的子问题,每个子问题就变为独立的程序模块。每个模块是整个系统的一部分,完成一个单独的功能,这就是模块化编程。模块化编程的主要思想是将程序分解为逻辑上相对独立的模块,每个模块实现特定的功能,相互之间通过预先定义的接口进行交互。

为实现模块化,可使用函数封装模块的代码。函数可以将相关代码块组织在一起,通过定义函数接口与其他部分交互。函数有助于将大程序拆分为逻辑独立的模块,通过封装提高代码的复用性和可维护性,并提高程序的可读性。总之,使用函数可使代码更清晰、灵活、可管理,是模块化编程的基础。

本章主要内容包括 Python 内置函数和自定义函数,如何定义函数,如何调用函数,函

数的实参、形参,参数之间是如何传递的,函数的默认参数、可变长参数、关键字参数、局部变量和全局变量、lambda 函数、filter 函数、map 函数、递归函数、main 函数及模块。

5.1 内置函数

为方便用户使用,Python 提供了许多内置函数,如 print、abs、len、int、max 等。表 5.1 列出了常用的内置函数,内置函数是可以直接使用的函数。对于内置函数,需要掌握函数名、函数功能、函数参数和返回值。

表 5.1 常用的内置函数

函数类别	函数名	说 明	示 例	运行结果
输入输出	print(s)	输出字符串 s	>>> print("hello, Python")	hello, Python
	input()	获取用户输入内容	>>> name = input("enter your name: ")	
数值运算	abs(x)	返回参数 x 的绝对值	>>> abs(-3.4)	3.4
	divmod(a, b)	一个包含商和余数的元组: (a // b, a % b)	x, y = divmod(10, 3)	(3, 1)
	round(x[, n])	返回参数 x 的四舍五入值,参数 n 可选,表示保留 n 位小数,默认表示不保留小数位	>>> round(3.1415926) >>> round(3.1415926, 2)	3 3.14
	pow(a, b)	返回参数 a 的 b 次幂	>>> power(3, 2)	9
	sum()	对可迭代对象参数求和	>>> sum(1, 2, 3, 4) >>> sum([1, 2, 3, 4])	10 10
	min()	求最小值	>>> min(1, 2, 3, 4) >>> min([1, 2, 3, 4])	1 1
	max()	求最大值	>>> max(1, 2, 3, 4) >>> max([1, 2, 3, 4])	4 4
类型转换	int(a)	将参数 a 转换为整数并返回	>>> int(3.5)	3
	float(a)	根据参数 a 返回其对应的浮点数	>>> float(3)	3.0
	str(a)	返回参数 a 的字符串形式	>> str(35)	'35'
数据结构	list(iterable)	接受一个可迭代的对象作为输入参数,将该对象中的元素转换为列表,再返回这个列表	>>> list((1, 2))	[1, 2]
	tuple(iterable)	接受一个可迭代的对象作为输入参数,将该对象中的元素转换为元组,再返回该元组	>>> tuple([1, 2])	(1, 2)
	dict(iterable)	通过迭代 iterable 对象,将键值对重新组装为字典	>>> dict([('a', 1), ('b', 2)]) >> dict(a=1, b=2)	{'a': 1, 'b': 2} {'a': 1, 'b': 2}
	set(iterable)	将可迭代对象转换为集合,元素不重复	>>> set([1, 2, 2, 3])	{1, 2, 3}

续表

函数类别	函数名	说 明	示 例	运行结果
序列运算	range(start, stop[, step])	返回一个 [start, stop) 步长为 step 的列表, start 和 step 的默认值分别为 0 和 1	>>> range(5) >>> range(1,5) >>> range(1,5,2)	[0,1,2,3,4] [1,2,3,4] [1,3]
	len(s)	返回对象 s 中元素的个数	>>> len([1,2,3,"h"])	4
	sorted(iterable, key = None, reverse=False)	对可迭代序列进行排序, 返回排序后的序列	>>> sorted([5,2,3]) >>> sorted([5,2,3], reverse=True)	[2,3,5] [5,3,2]
	filter(function, iterable)	对 iterable 对象中的每个元素调用 function 函数进行判断, 将函数返回 True 的元素过滤出来, 生成一个新的迭代器返回	>>> list(filter(lambda x: x%2 == 0, [1,2,3,4]))	[2,4]
	zip(*iterables)	将可迭代对象中对应的元素打包为一个元组, 再返回由这些元组组成的 zip 对象	>>> n=[1,2] >>> s=["a","b"] >>> list(zip(n,s))	[(1, 'a'), (2, 'b')]
其他	map(function, iterable, ...)	将 function 函数依次作用到 iterable 可迭代对象中的每个元素, 并将结果作为新的 map 对象返回	>>> n=[1,2] >>> list(map(lambda x: x*3,n))	[3,6]
	id(object)	返回参数 object 的内存地址	y=id(6)	
	type(object)	返回参数 object 的数据类型	type(4)	< class 'int'>

5.2 自定义函数

5.2.1 函数定义

当内置函数无法满足需求时, 需要自己创建函数, 称为自定义函数, 其语法格式为:

```
def func_name([arguments]):           # 函数定义头部, 指定函数名和参数
    statement(s)                       # 函数体语句, 实现功能
    [return expression]               # 返回值表达式
```

函数由两部分组成: 函数头和函数体。

函数头: 包括 def 关键词、函数名称、参数列表。函数头以 def 开头, 后跟函数名称和圆括号()。圆括号中是参数列表, 多个参数之间用英文逗号隔开, 可以没有参数。圆括号后面是一个冒号。

函数体: 函数体包含函数执行的语句, 使用缩进表示函数体代码块。函数体内部可使用 return 语句返回值。如果没有 return 语句, 函数将不返回值。一个函数可有多返回值。

函数定义涉及函数的名称、参数、返回值及函数体代码, 是创建一个函数的必要元素。通过函数定义, 构造一个可被重复调用的代码块, 用于实现特定的功能。

【实例 5.1】 函数定义示例。



```
1 def greet():          # 定义一个输出欢迎信息的函数. 函数名为 greet, 没有参数
2     print("Welcome to Python world!")
3 def c_f(c):           # 定义一个将摄氏度向华氏度转换的函数, 函数名为 c_f, 有一个参数 c
4     f = 32 + c * 1.8
5     return f
```

该程序定义了两个函数。

greet 函数：无参数，打印一条问候语，没有返回值。

c_f 函数：有 1 个参数 c，实现摄氏度到华氏度的转换，通过 return 返回计算结果。

运行上述程序，发现该程序没有任何输出，因为程序中只进行了函数定义，未调用函数。定义仅表示创建了函数，但未实际执行。要实现函数功能，必须在其他地方调用预先定义的函数。

5.2.2 函数调用

函数调用是使用预先定义的函数完成某项任务的过程，函数调用会引发程序控制流从主程序转移到被调用函数的函数体，执行其中的语句，再返回主程序调用点并带回返回值。

函数调用的格式如下：

```
func_name(par1, par2, ...)
```

其中：

func_name 为函数名，调用的函数名必须与定义的函数名完全一致。

函数调用中函数名后圆括号内的 par1, par2, ... 为函数实参，即从主程序向该函数传递的参数值。需要注意的是，函数调用时的参数个数和参数顺序必须与函数定义时一致。另外，即使该函数没有参数，调用时也要书写一对空的括号。

【实例 5.2】 调用【实例 5.1】中定义的函数示例。

```
1 greet()                # 调用函数 greet, 该函数没有返回值
2 f_temperature = c_f(30) # 调用函数 c_f, 将 30 的值传递给参数 c
3 print("30 摄氏度对应的华氏度为:", f_temperature)
4 print("20 摄氏度对应的华氏度为:", c_f(20)) # 调用函数 c_f, 将 20 的值传递给参数 c
```

运行结果如下：

```
Welcome to Python world!
30 摄氏度对应的华氏度为: 86.0
20 摄氏度对应的华氏度为: 68.0
```

实例 5.2 调用了实例 5.1 中定义的两个函数。

(1) 调用 greet 函数，该函数无返回值，直接调用即可执行其内部语句。

(2) 调用 c_f 函数，该函数有返回值。可将返回值赋给变量，如 f_temperature=c_f(30)，也可直接将函数调用作为表达式的一部分，如 print(c_f(20))。

所以对于无返回值的函数，直接调用即可。对于有返回值的函数，可通过赋值运算获得返回值，也可直接使用函数调用结果。

通过调用可重复使用函数内部的代码逻辑。正确调用函数是应用函数的关键。

数函数、判断猜测的数字是否正确的函数,以及让用户猜测多次的函数。这3个函数相互依赖,首先产生随机数 newNumber,这是后面猜数字的基础,因为必须先有猜测的对象,才能让用户猜测。其次 guessNumber 让用户输入自己的猜测并判断是否正确,而目标是让用户猜测多次,因此再创建一个函数 guessTime(number, times),该函数调用 times 次 guessNumber,从而实现用户最多可猜测 times 次,如果猜对,则退出循环。



5.9 花样滑冰模拟计分系统——一起向未来

5.9.1 思政导入

第24届冬季奥林匹克运动会简称2022年北京冬奥会,是由中国举办的国际性奥林匹克赛事,于2022年2月4日开幕,2月20日闭幕。此届冬奥会共设滑雪、滑冰、冰球、冰壶、雪车、雪橇和冬季两项7个大项,高山滑雪、自由式滑雪、单板滑雪、跳台滑雪、越野滑雪、北欧两项、短道速滑、速度滑冰、花样滑冰、冰壶、冰球、雪车、钢架雪车、雪橇和冬季两项15个分项及109个小项。

中国体育代表团总人数为387人,其中运动员176人,教练员、领队、科学医护人员等运动队工作人员164人,团部工作人员47人,创中国体育代表团历届冬奥会参赛规模之最。中国体育代表团完成了北京冬奥会全部7个大项、15个分项的“全项目参赛”任务,共获得9金4银2铜,位列奖牌榜第三,金牌数和奖牌数均创历史新高。其中,隋文静和韩聪以总成绩239.88分获得花样滑冰双人滑比赛金牌,实现了花滑双人滑全满贯的壮举。

花样滑冰起源于18世纪的英国,后相继在德国、美国、加拿大等欧美国家迅速开展。与其他竞技运动不同,花样滑冰是一项艺术与运动结合的体育项目,除了要掌握冰上技术,对运动员的艺术表现力有极高的要求。在音乐伴奏下,运动员在冰面上滑出各种图案、表演各种技巧和舞蹈动作,裁判员根据动作评分,决定名次。冬奥会花样滑冰包括4个项目:男子单人滑、女子单人滑、双人滑和冰舞,比赛均在室内进行。它要求在60米×30米的冰场上,运动员以40千米/时的速度完成各种高难度动作,同时还要用自己的艺术表演诠释背景音乐,感染裁判和观众。此项运动涵盖体育、艺术、音乐、舞蹈、服装设计、化妆……因此对运动员和教练技术以外的要求也非常高。

花样滑冰是比赛规则最复杂、评分难度最高的体育项目之一,评委需在高速运动且变化繁杂的动作中依据动作的类型、难度系数、完成情况、标准程度等给出精准的技术分,通过AI技术辅助评分难度也可见一斑。2022年1月21日,花样滑冰AI辅助评分系统1.0发布,这套辅助系统是根据中国花样滑冰运动员使用需求、场景应用需求打造的AI+虚拟现实解决方案,运用计算机视觉技术算法与深度学习,以对运动员的整体运动轨迹进行实时追踪,根据专业评分标准,对视频数据的人体骨骼、形体动作进行捕捉识别,从而实现稳定性可视化的比赛评判。

5.9.2 案例任务

由于花样滑冰的评分规则比较复杂,可对其进行简化,简化后的规则如下:花滑总分数

由技术水平分和节目内容分两部分构成。每位评委分别为每位选手打出技术水平分和节目内容分。裁判组对每位选手的技术水平分和节目内容分去掉最高分和最低分,再将其平均分相加,即得到该选手的综合分。

请编写程序模拟评委打分、裁判组汇总分数、显示分数。

5.9.3 案例分析与实现

首先比赛选手名单可用列表 `skater_lst` 存储,假定有 3 位选手,按照列表中选手的顺序依次进行表演,表演完毕后,评委(假定有 5 位)给出该选手的技术水平分 `element_score` 和节目内容分 `component_score`。定义一个字典 `player_score`,其键名为选手姓名,键值为列表类型,将每位评委打出的两部分分数组成一个元组,作为列表元素。打分完毕后,该字典有 3 个元素,每个元素的键名为选手姓名,键值是一个包含 5 个元素的列表,每个元素是一个形如“(`element_score`,`component_score`)”的元组。

裁判组对每位选手进行分数计算,用一个 `compute_score` 函数实现。该函数的参数为字典 `play_score`,功能为:对于该字典中的每个键名(即选手姓名),按照规则计算其最终技术水平分、最终节目内容分和总分,并将 3 部分内容合在一起构建一个元组,附加到该选手对应键值(列表类型)的末尾。

显示分数环节用 `show` 函数实现,参数为字典 `play_score`。其功能为根据总分排序并输出选手的名次、姓名、总分、技术水平分和节目内容分。

源代码如下:

```
1  import random
2  import time
3  skater_lst = ["金博洋","羽生结弦","陈巍"]  # 定义列表,保存选手姓名信息
4  player_score = {}
5  num_judge = 5                                # 评委人数
6
7  def compute_score(player_score):
8      '''
9      player_score:{player_name:[(element_score1,component_score1),...]}
10     '''
11     for name in player_score.keys():
12         score_list = player_score[name]
13         for i in range(len(score_list)): # len(player_score[name])为评委个数
14
15             # 将 element_score 汇总起来
16             element_scores = [score_list[j][0] for j in range(len(score_list))]
17             component_scores = [score_list[j][1] for j in range(len(score_list))]
18             # 去掉最高分、最低分,求均值
19             max_element = max(element_scores)
20             max_component = max(component_scores)
21             min_element = min(element_scores)
22             min_component = min(component_scores)
23             final_element = (sum(element_scores) - max_element - min_element)/(len(element_
24 scores) - 2)
25             final_component = (sum(component_scores) - max_component - min_component)/(len
26 (component_scores) - 2)
```

```

27         final_score = final_element + final_component
28         tmp = (round(final_score, 2), round(final_element, 2), round(final_component, 2))
29         player_score[name].append(tmp)
30
31     def show(player_score):
32         '''
33         player_score:{player_name:[(element_score1,component_score1),...(final_score,
34         final_element,final_component)]}
35         '''
36         # 将 player_score 中的姓名、总分、技术分、内容分存为列表,排序
37         score_lst = []
38         for name in player_score.keys():
39             score = player_score[name][-1]
40             tmp = [name]
41             tmp.extend(score)
42             score_lst.append(tmp)
43         score_lst.sort(key = lambda x:x[1], reverse = True)
44         print(" 名次   选手姓名   总 分   技术分   内容分 ")
45         for i in range(len(score_lst)):
46             print(f"{i+1:3} ", end = '')
47
48             for j in range(len(score_lst[i])):
49                 print("{:^9}".format(score_lst[i][j]), end = "")
50             print()
51
52     def main():
53         print(" *** 欢迎使用花样滑冰模拟计分系统 *** ")
54         for i in range(len(skater_lst)):
55             print(f"欢迎欣赏{skater_lst[i]}的花样滑冰")
56             player = skater_lst[i]
57             player_score[player] = []
58             # 每个评委为该选手打分
59             for j in range(num_judge):
60                 print(f"请第{j+1}位评委为{player}打分")
61                 # element_score = eval(input("请输入技术分:"))
62                 # component_score = eval(input("请输入内容分:"))
63                 element_score = round(random.uniform(50, 100), 2) # 为便于测试,由系统随机
64                                                         # 生成分数
65                 component_score = round(random.uniform(50, 100), 2)
66                 tmp = (element_score, component_score)
67                 player_score[player].append(tmp)
68             print("现在是裁判组汇总分数时间")
69             compute_score(player_score)           # 计算每个选手的总分
70             show(player_score)                     # 按照从高到低的顺序输出选手的得分信息
71     if __name__ == "__main__":
72         main()

```

5.9.4 总结和启示

从2008年到2022年,奥林匹克两度携手中国。“冰丝带”盈盈飘动,“雪如意”雄踞山巅,“冰之帆”御风而行,“雪飞天”长袖善舞……

“成功举办北京冬奥会、冬残奥会,不仅可以增强我们实现中华民族伟大复兴的信心,而且有利于展示我们国家和民族致力于推动构建人类命运共同体,阳光、富强、开放的良好形象,增进各国人民对中国的了解和认识。”正如习近平总书记所言,2022年北京冬奥会不仅是一场体育盛会,更折射出中国推动构建人类命运共同体的价值追求,具有深远的世界意义。

武大靖、苏翊鸣、谷爱凌等运动健儿在冬奥赛场上奋力拼搏,勇创佳绩,可喜可贺!同时,人工智能也在冬奥会上绽放异彩,花样滑冰 AI 辅助评分系统的发布也让我们看到互联网、人工智能、大数据等技术在体育运动领域的飞速发展。

5.10 本章小结

本章主要介绍了函数的定义和调用、函数的参数、变量的作用域、lambda 函数、filter 函数、map 函数、递归函数及 main 函数与模块的使用。其中函数的参数是本章的难点之一,包括形参和实参,参数之间的传递、默认参数、可变长参数及关键字参数。通过具体实例的讲解,读者对其中的概念有更直观、更深刻的理解。最后通过几个综合例子锻炼读者的综合编程能力。

5.11 巩固训练

【训练 5.1】 给定一个正整数,编写程序计算有多少对质数的和等于输入的这个正整数,并输出结果。输入值小于 1000。

【训练 5.2】 编写函数 `change(str)`,其功能是对参数 `str` 进行大小写互换,即将字符串中的大写字母转为小写字母、小写字母转换为大写字母。

【训练 5.3】 编写函数 `digit(num,k)`,其功能为:求整数 `num` 第 `k` 位的值。

【训练 5.4】 编写递归函数 `fibonacci(n)`,其功能为:求第 `n` 个斐波那契数列的值,进而输出前 20 个斐波那契数列。

【训练 5.5】 编写一个函数 `calculate`,可接收任意个数,返回一个元组。元组的第一个值为所有参数的平均值,第二个值为小于平均值的个数。

【训练 5.6】 模拟轮盘抽奖游戏。轮盘分为三部分:一等奖、二等奖和三等奖。轮盘随机转动,如果范围在 $[0, 0.08)$ 之间,代表一等奖;如果范围在 $[0.08, 0.3)$ 之间,代表二等奖;如果范围在 $[0, 1.0)$ 之间,代表三等奖。

【训练 5.7】 有一段英文: What is a function in Python? In Python, function is a group of related statements that perform a specific task. Functions help break our program into smaller and modular chunks. As our program grows larger and larger, functions make it more organized and manageable. Furthermore, it avoids repetition and makes code reusable. A function definition consists of following components. Keyword `def` marks the start of function header. A function name to uniquely identify it. Function naming follows the same rules of writing identifiers in Python. Parameters (arguments)

through which we pass values to a function. They are optional. A colon (:) to mark the end of function header. Optional documentation string (docstring) to describe what the function does. One or more valid Python statements that make up the function body. Statements must have same indentation level (usually 4 spaces). An optional return statement to return a value from the function.

任务：1. 请统计该段英文有多少个单词，以及每个单词出现的次数。2. 如果不算 of、a、the 这 3 个单词，给出出现频率最高的 10 个单词，并给出其出现的次数。

【训练 5.8】 利用函数实现磅(lb)与千克(kg)的转换。用户可以输入千克，也可以输入磅，函数将根据用户的输入转换为磅或千克。

【训练 5.9】 一个数如果恰好等于其因子之和，该数就称为“完数”，例如 $6=1+2+3$ 。编程找出 1000 以内的所有完数。

【训练 5.10】 利用递归函数调用方式，将用户输入的字符串以相反的顺序输出。