

财政部规划教材

财经商贸大类岗课赛证
融合新形态教材

DASHUJU JISHU ZAI CAIWU
ZHONG DE YINGYONG

大数据技术 在财务中的应用

(Python版) 郭黎 吕文吉 何慧玲 ● 主编



落实立德树人根本任务
融入课程思政内容

+



遵循岗课赛证融合原则
理实一体综合育人

+



模块化知识够用为基点
实现新形态一体化

● 校企双元开发

● 符合岗位要求

● 项目任务教学

● 配备微课视频



中国财经出版传媒集团
中国财政经济出版社

财政部规划教材
财经商贸大类岗课赛证融合新形态教材

大数据技术在财务中的应用(Python版)
郭黎 吕文吉 何慧玲 主编



中国财经出版传媒集团
中国财政经济出版社

财政部规划教材

财经商贸大类岗课赛证融合新形态教材

经济法基础（第四版）

统计学基础

会计基础

出纳业务操作

财政金融基础

企业管理基础

大数据技术应用基础

财务机器人应用与开发

企业财务会计

※ 财务会计（第三版）

智能化成本核算与管理

智慧化税费申报与管理

会计信息系统应用

企业内部控制

★ 管理会计实务（第二版）

大数据技术在财务中的应用(Python版)

★ 智能财务分析与可视化

财务大数据分析可视化

会计综合实训（实操版）（第三版）

财务报表分析（第二版）

Excel在会计中的应用（第三版）

※ 审计基础与实务（第二版）

行业会计比较

财经应用文写作

经济应用数学

★ 经济学基础（第三版）

市场调查与预测

管理学基础

加※的为“十四五”职业教育国家规划教材

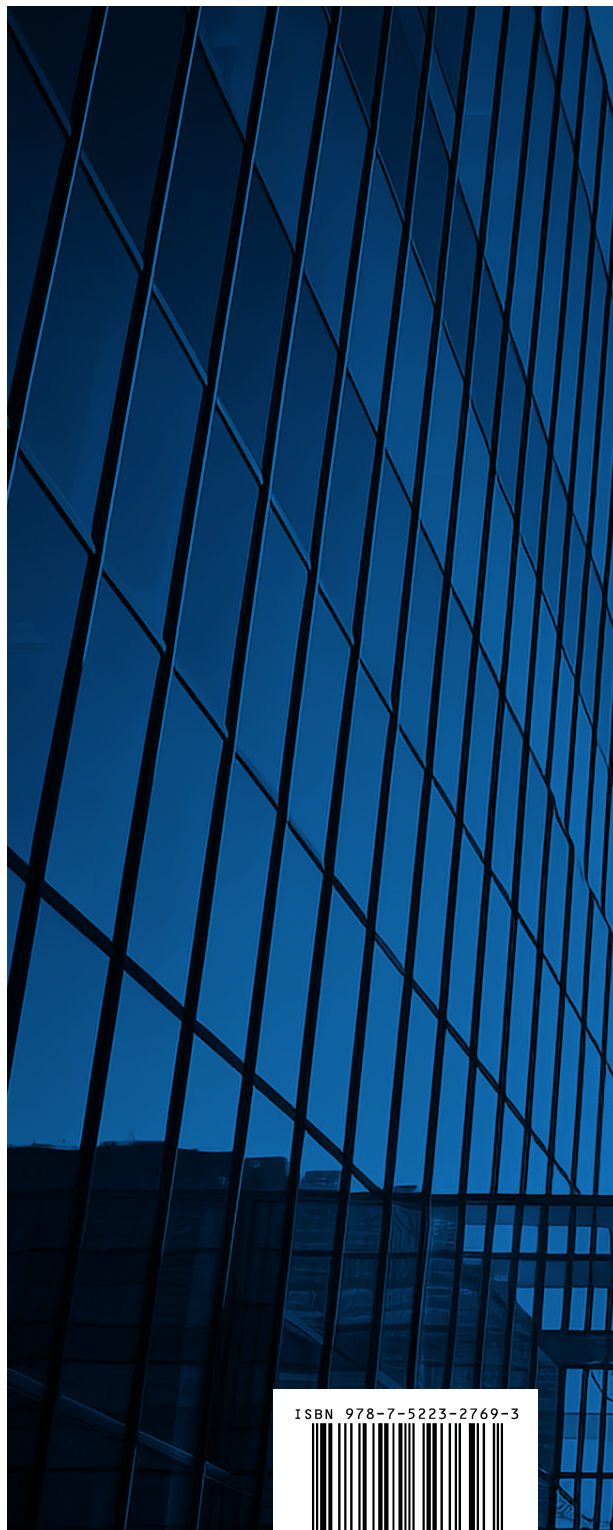
加★的为“十四五”职业教育省级规划教材



本社微信公众号
(请用微信“扫一扫”)



本社天猫旗舰店
(请用手机天猫APP“扫一扫”)



ISBN 978-7-5223-2769-3



9 787522 327693 >

定价：45.00元

财政部规划教材

财经商贸大类 岗课赛证融合新形态教材

大数据技术 在财务中的应用

(Python版)

主 编 郭 黎 吕文吉 何慧玲
副主编 杨博伟 卜鸣越 周 姝
潘 妮 宁靖华 张鲲鹏
邹 芳

DASHUJU JISHU ZAI CAIWU
ZHONG DE YINGYONG



中国财经出版传媒集团
中国财政经济出版社

图书在版编目(CIP)数据

大数据技术在财务中的应用: Python版 / 郭黎, 吕文吉, 何慧玲主编. — 北京: 中国财政经济出版社, 2024.1

ISBN 978-7-5223-2769-3

I. ①大… II. ①郭… ②吕… ③何… III. ①财务管理—数据处理—高等职业教育—教材 IV. ①F275

中国国家版本馆CIP数据核字(2024)第013518号

策划编辑: 李昊民

责任校对: 张 凡

责任编辑: 张怡然

责任印制: 张 健

大数据技术在财务中的应用(Python 版)

DASHUJU JISHU ZAI CAIWU ZHONG DE YINGYONG

中国财政经济出版社出版

URL: <http://www.cfeph.cn>

E-mail: cfeph@cfemg.cn

(版权所有 翻印必究)

社址: 北京市海淀区阜成路甲28号 邮政编码: 100142

营销中心电话: 010-88191522

天猫网店: 中国财政经济出版社旗舰店

网址: <https://zgczjjcbs.tmall.com>

廊坊市佳艺印务有限公司印刷 各地新华书店经销

成品尺寸: 203mm × 260mm 16开 14印张 290 000字

2024年1月第1版 2024年1月河北第1次印刷

定价: 45.00元

ISBN 978-7-5223-2769-3

(图书出现印装问题, 本社负责调换, 电话: 010-88190548)

本社质量投诉电话: 010-88190744

打击盗版举报热线: 010-88191661 QQ: 2242791300

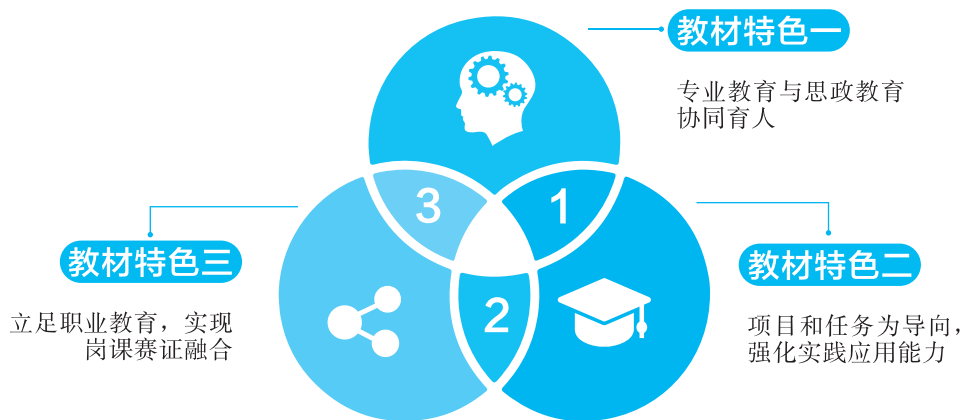
前言

随着经济的发展和信息技术的进步，大数据时代已经到来。大数据技术直接影响着企业的经营管理方式，企业对财务岗位职能的要求越来越高。2021年年底，财政部印发了《会计信息化发展规划（2021—2025年）》，强调会计工作实现数字化转型、发挥其服务单位价值创造功能是我国会计信息化发展的重要任务。同年，教育部印发《职业教育专业目录（2021年）》，对财务会计类专业进行了全面更名。这标志着财务会计类专业的升级与数字化改造势在必行。培养既具备财务专业知识，又掌握大数据分析处理技术的新型高端技术技能型财务人才，成为高职财务会计类专业新的人才培养目标。

Python 是大数据技术中最常用的编程语言之一，具有易学易用、灵活性强等特点，并且有着丰富的第三方库支持，如 Pandas、NumPy、Matplotlib 等，可以快速处理和分析大量数据。这些优势使 Python 成为处理海量数据和进行机器学习任务的首选语言之一。

本书紧扣行业发展趋势和社会发展对人才的需求，着重阐明了大数据技术（Python 技术）的基本理论、基本方法、基本技能以及这些基本知识的运用，以培养学生敏锐的思维能力、分析能力和实际应用能力为编写原则，以简单的大数据分析方法体系作为基本框架，借助直观明了的语言代码和最新的会计信息资料，用通俗易懂的语言讲述了大数据技术在财务中的基本理论和方法，增强了本书的实用性和时效性。其中，项目一至项目四讲解 Python 的基础知识，项目五至项目八以财务大数据分析工作过程为轴线分别对财务数据分析、财务数据可视化、上市公司财务数据分析等内容加以阐述。将知识传授、能力培养与价值塑造融为一体，遵循学生的学习规律，通过学练结合的方式，引导学生从易到难、循序渐进地进行学习。

本书的特色主要体现在以下三点：



教材特色一 专业教育与思政教育协同育人

结合财会实务案例和德技并修栏目，引导学生正确认识财务工作，有效地将奉献精神、工匠精神、创新精神等融入课堂教育之中，使学生能够在具体的教学实践中，显性与隐性结合地感受思政教育内容，以此实现对学生思想政治观念的引导与培养，促进其能力和素质的共同发育，为“培养什么样的人”这个基本教育问题提供解决思路。

教材特色二 项目和任务为导向，强化实践应用能力

本书以学生为主体，从学生的认知规律出发设计教学流程。明确学习目标，通过“知识储备任务”讲解 Python 技术应用知识点；通过“技能训练任务”引入财务案例，帮助学生灵活运用理论知识解决财务问题；通过“拓展学习”拓宽学生视野、丰富学生认知；通过“技能提升”帮助学生巩固强化知识，提升应用能力。

教材特色三 立足职业教育，实现岗课赛证融合

职业教育应以国家需要、市场需求为出发点，培养高素质技术技能人才，提升学生技术水平和就业创业能力。本书服务于“1+X”证书制度改革，创新融合技能等级证书，引入“1+X”证书真题，配以丰富案例，突出考点知识，强化学生理解，满足考证需求。同时本书增加了业务场景、业务实训等内容，使学生快速适应财税岗位技能任职需要，为走向工作岗位打下坚实基础。

本书配有课件、教案、微课视频、代码文件等资源。

本书由武汉软件工程职业学院郭黎老师和咸宁职业技术学院吕文吉、何慧玲老师担任主编。咸宁职业技术学院杨博伟、周姝老师，武汉软件工程职业学院卜鸣越、潘妮老师，湖南交通职业技术学院宁靖华老师，湖南保险职业学院张鲲鹏老师，巴中职业技术学院邹芳老师参与编写。广州福思特科技有限公司在本书编写过程中给予了技术支持，并协助开发财务案例、编写代码。

本书可作为高等职业院校财会类、经贸类、管理类及其他相关专业的教学用书，也可作为各类成人院校及企业职工的培训教材，还可以作为社会从业人员学习 Python 语言和从事企业财务工作的参考用书。教材中前七个项目中涉及的公司名称、财务数据均为虚拟，仅用于课堂学习。

本书在编写的过程中，参考借鉴了一些专家学者的专著和教材。在此，我们向这些专著、教材的作者表示衷心的感谢！由于编者理论知识和实践能力有限，书中难免存在疏漏之处，敬请广大读者批评指正，以便进一步丰富和完善。



扫码获取代码文件

编者

岗位能力获得清单

职能岗位	岗位职责	职业技能		职业素养
财务数据分析师	协助企业完成数据分析平台模型搭建	计算机应用能力	1. 熟练使用 Excel、Python 等数据分析工具，熟悉编程语言 2. 熟练使用可视化工具	具备正确的技能观，具备工匠精神
	完成日常数据整理和分析工作；收集核算数据，制作公司管理报表	数据采集能力	1. 熟悉数据采集的原理与方法，能运用爬虫的原理获取网络数据 2. 有较强的逻辑分析能力，具有数据敏感性，能够从繁杂的财务数据中，采集到有效的财务数据	养成遵守社会规则的习惯，尊重他人的劳动成果，尊重知识产权
	基于对海量运行数据的分析，制定多维度获取相关数据的策略；通过对数据和业务的挖掘，持续给出优化、升级方案	数据加工能力	1. 能够通过定义函数实现代码的反复利用，从而提高工作效率 2. 能够运用大数据技术熟练地读写和保存文件 3. 能够对企业生产、采购、财务、销售等方面产生的数据信息，进行增减、筛选、运算、加工，获取有效的财务数据 4. 熟悉财务工作流程，能够运用数据分析工具设计和优化财务流程，节省时间和财务成本	严谨细致、精益求精，遵循编程规范，促进团队协作和提高开发效率，降低沟通成本
	进行经营分析，帮助业务部门优化经营管理流程，提高经营效率	数据可视化能力	1. 熟悉主流数据可视化工具，能够绘制基本的数据分析图表，如折线图、饼图、条形图等 2. 能够从繁杂的财务数据中，选择恰当的可视化方式进行呈现，提炼出可视化报告，支持管理者决策	致知于行，合作共赢，充分将信息技术与财务分析工作结合起来，将财务数据化繁为简，创造更大价值
	做好信息保护工作，了解信息安全标准，确保企业数据在采集、存储、处理和传输等方面的保密性、完整性和可用性	数据分析能力	1. 熟悉财务分析知识，能够对采集到的数据进行处理和分析，从中提取有价值的信息 2. 将财务数据转化为可视化方式进行呈现，开展财务分析，支持管理者决策	能够与其他职能部门进行有效沟通和协调，将财务分析结果与业务部门对接，实现业财融合

001 项目一 大数据与财务认知

002 知识储备任务

002 任务一 认知大数据技术

004 任务二 认知大数据技术在财务中的应用

006 任务三 认知 Python

007 技能训练任务 搭建 Python 大数据开发环境

032 项目二 财务数据的录入与存储

033 知识储备任务

033 任务一 输入与输出财务数据

037 任务二 设置财务数据变量

038 任务三 明确数据类型和运算符

052 技能训练任务 员工信息管理

057 项目三 财务数据的选择

058 知识储备任务

058 任务一 认知流程控制——分支条件

062 任务二 认知流程控制——循环语句

066 技能训练任务 固定资产折旧计算

074 项目四 财务数据的调用

075 知识储备任务

075 任务一 认知内置函数

087 任务二 认知自定义函数

092 技能训练任务 低值易耗品的管理

098 项目五 财务数据分析——Pandas 应用

099 知识储备任务

099 任务一 认知 Pandas 数据结构

103 任务二 读写文件操作

109 任务三 选取和筛选数据

116 任务四 合并数据

122 技能训练任务 职工薪酬计算

133 项目六 财务数据分析——NumPy 应用

134 知识储备任务

134 任务一 认知 NumPy 数组对象

138 任务二 NumPy 数组操作

148 任务三 NumPy 常用函数

156 技能训练任务 股票价格分析

172 项目七 财务数据可视化——Matplotlib 应用

173 知识储备任务

173 任务一 绘制基础图形

181 任务二 绘制子图

185 技能训练任务 淘宝数字化选品

194 项目八 上市公司财务数据分析

195 知识储备任务

195 任务一 认知数据采集

197 任务二 采集上市公司财务数据

202 技能训练任务 东风汽车公司财务分析

216 参考文献

项目二

财务数据的录入与存储



学习目标

知识目标	技能目标	素养目标
1. 掌握标准输出语句的基本用法	1. 能够使用 <code>print</code> 语句进行财务数据输出	1. 树立正确的技能观，培养工匠精神，实现从基础代码到复杂编程的积累和能力提升
2. 掌握标准输入语句的基本用法	2. 能够使用 <code>input</code> 语句进行财务数据输入	
3. 掌握变量的赋值与命名规则	3. 能够使用变量对财务数据进行命名和赋值	2. 认知编程规范的重要性，养成遵守社会规则的习惯，学会降低沟通成本的方法
4. 掌握数据类型和运算符的基本知识	4. 能够创建、修改常见数据类型的对象，并结合运算符使用内置函数进行运算	



知识框架图

知识储备任务

任务一 输入与输出财务数据

- 输出
- 输入
- 注释

任务二 设置财务数据变量

- 变量与赋值
- Python 变量的特点
- 变量的命名规则

任务三 明确数据类型和运算符

- 数据类型
- 运算符

技能训练任务

- 员工信息管理

【职场箴言】

没有数据，就没有竞争力。

——托马斯·瑞德曼 (Thomas Redman)

自主预习单

一、学习指南
1. 课题名称 财务数据的录入与存储
2. 学习方法建议 演示法、课堂讨论法、自主探究法
3. 课堂学习形式预告 本项目采取“自主预习—情景导入—教学认知—实操演练—课后巩固提升”的基本教学模式
二、预习任务
本项目内容属于 Python 技术应用的基础知识，较容易掌握。具体预习任务为： 1. 了解输入与输出函数的作用 2. 理解财务数据变量的含义 3. 了解数据类型
三、困惑与建议

知识储备任务

任务一 输入与输出财务数据

一、输出

在 Python 中，使用内置的 print() 函数可以输出运行结果。括号中的输出内容可以是任何想在屏幕上显示的信息，包括数字、字符串及其组合等，也可以是包含运算符的表达式计算结果。

其语法格式如下：

```
print( 输出内容 )
```

当有多个输出内容时，需用英文半角符号“,”进行分隔。若输出的内容为字符串，需用英文半角单引号(')、双引号(" ")或三引号('' ''')('"' '"')进行标记。

【练一练 2-1】

1. 输出字符串

运行代码：

```
print("Hello World!")
```

运行结果如下:

```
Hello World!
```

2. 输出数值

运行代码:

```
print(100)
```

运行结果如下:

```
100
```

3. 输出含运算符的表达式

运行代码:

```
print(20+15)
```

运行结果如下:

```
35
```

4. 混合输出

运行代码:

```
print("应收账款为人民币", 30000, "元")
```

运行结果如下:

```
应收账款为人民币 30000 元
```

二、输入

1. 单一值输入

Python 提供了内置函数 `input()`，用于接收用户从键盘输入的信息。在括号内可以输入提示文字，告知用户需要做什么。

其语法格式如下:

```
input('输入提示信息')
```

【练一练 2-2】

运行代码:

```
input('请输入你的专业名称:')
```

运行结果如下:

```
请输入你的专业名称:|
```

用户在光标“|”处输入专业名称“大数据与会计”后，系统会输出以下内容:

```
大数据与会计
```

无论用户输入的是什么内容，`input()` 函数都是以字符串类型返回结果。在财务工作中，需要对各类数据进行处理，此时可以使用函数 `int()` 将输入的值转换成数字。



【微课】
input 函数

其语法格式如下:

```
int(input('输入提示信息'))
```

【练一练 2-3】

运行代码:

```
num=int(input('请输入你的专业人数:'))
```

运行结果如下:

```
请输入你的专业人数:|
```

用户在光标“|”处输入专业人数“1000”后,运行代码:

```
print(type(num))
```

运行结果如下:

```
<class 'int'>
```

【业务场景 2-1】

张晓明为企业采购一批办公用 A4 纸, A4 纸的售价是 20 元/包, 他购买了 10 包。请编程计算张晓明需要支付的金额。

参考代码:

```
price=20
num=int(input("请输入购买的数量:"))
total_price=price*num
print(total_price)
```

运行代码:

```
请输入购买的数量:|
```

用户在光标处输入购买的数量“10”后,运行结果如下:

```
200
```

2. 多值输入

input() 函数与 split() 函数联合使用, 可以实现一次输入多个值。split() 函数默认以空格分隔多个值, 如需其他符号进行分隔, 可在括号内输入相应符号。

【练一练 2-4】

运行代码:

```
a,b=input('请输入你们的名字以空格分隔:').split()
print(a)
print(b)
```

运行结果如下:

```
请输入你们的名字以空格分隔:|
```

用户在光标 “|” 处输入两个人的名字 “张三 李四” 后, 运行结果如下:

```
张三  
李四
```

三、注释

为了提高程序的可读性, 要为程序添加注释。注释用来解释说明程序某些代码的作用和功能, 同时也能方便他人阅读。

1. 单行注释

Python 中的单行注释以 # 开头, 直到换行处结束, 此部分内容都作为注释的内容, 当 Python 解释器碰到 # 后, 会自动跳过这段代码, 继续向下执行。

其语法格式如下:

```
# 单行注释内容
```

【练一练 2-5】

运行代码:

```
print ('大数据与会计') # 输出字符串
```

运行结果如下:

```
大数据与会计
```

2. 多行注释

多行注释可以一次性注释程序中的多行内容 (包含一行)。

多行注释采用三单引号 (""" """) 或三双引号 (""" """) 将多行注释包裹起来。

其语法格式如下:

```
'''  
用三引号  
表示多行  
注释  
'''
```

多行注释通常用于 Python 文件、模块、类或函数等添加版权、功能描述等信息。

任务二

设置财务数据变量

一、变量与赋值

变量（variable）就是在程序执行过程中，其值是可以变化的量，可以将变量看作是一个容器，用来存放数据。

赋值是指将数据存放给变量的过程，Python 中用“=”作为赋值运算符。

其语法格式如下：

变量名 = 值

【练一练 2-6】

运行代码：

```
a=100      # 将数值 100 赋值给变量 a
print(a)    # 输出变量 a
```

运行结果如下：

```
100
```

还可以利用变量接收用户输入的数据，然后输出。

运行代码：

```
name=input('请输入你的姓名:')
age=input('请输入你的年龄:')
print(name, age)
```

运行结果如下：

```
请输入你的姓名：张明
请输入你的年龄：19
张明 19
```

二、Python 变量的特点

（1）Python 中的变量无须定义即可直接使用，无须事先声明变量名及其类型，给变量名直接赋值即可创建各种类型的对象变量。

（2）Python 解释器会根据赋值或运算来自动推算变量的数据类型。

（3）变量的类型可根据值随时变化。

三、变量的命名规则

（1）变量名区分大小写，例如 Python 不能写成 python。

- (2) 变量可采用字母、下划线、数字、汉字等字符及其组合进行命名，且第一个字符不能是数字。
- (3) 变量名不能是 Python 关键字。Python 关键字指的是 Python 本身“已经在使用”的名字。常用 Python 关键字见表 2-1。

表 2-1 常用 Python 关键字

True	elif	is
False	else	not
None	except	or
and	finally	pass
as	for	return
assert	from	try
break	global	while
continue	if	with
def	import	
del	in	

任务三 明确数据类型和运算符

一、数据类型

变量可以存储许多不同类型的数据，数据类型是反映数据形态的一种属性，是 Python 中的基本概念。Python 的内置数据类型包括数字、布尔值、字符串、列表、字典等，可以供财务数据处理时直接使用。

(一) 数字 (Number)

在 Python 中，数字类型的变量可以表示任意大的数值，可以对应数学的数字概念来理解。数字类型主要有整数、浮点数和复数。

1. 整数 (Int)

整数，也被称为整型，包括正整数、负整数和零，不带小数点。整数类型主要包括十进制整数、二进制整数、八进制整数、十六进制整数。整数的分类及用法见表 2-2。

表 2-2 整数的分类及用法

整数类型	前缀符号	合法数字	示例 (以整数 18 为例)
十进制整数		0、1、2、3、4、5、6、7、8、9	18
二进制整数	0b	0、1	0b10010
八进制整数	0o	0、1、2、3、4、5、6、7	0o22
十六进制整数	0x	0、1、2、3、4、5、6、7、8、9、a、b、c、d、e、f	0x12

 【练一练 2-7】

运行代码:

```
var1=1
print("var1 的数值:", var1)
print("var1 的数据类型:", type(var1))

var2=-9999999999
print("var2 的数值:", var2)
print("var2 的数据类型:", type(var2))

var3=int(18)
print("var3 的数值:", var3)

var4=int("0b10010", 2)
print("var4 的数值:", var4)

var5=int("0o22", 8)
print("var5 的数值:", var5)

var6=int("0x12", 16)
print("var6 的数值:", var6)
```

运行结果如下:

```
var1 的数值:1
var1 的数据类型:<class'int'>
var2 的数值:-9999999999
var2 的数据类型:<class'int'>
var3 的数值:18
var4 的数值:18
var5 的数值:18
var6 的数值:18
```

 【提示】

内置的 `type()` 函数可以用来查询变量所对应的对象类型。

2. 浮点数 (Float)

浮点数由整数部分与小数部分组成, 可以对应数学中的小数来理解, 也是做数值计算中常用的类型, Python 能容纳极小和极大的浮点数。Python 中的写法与数学中一样, 如 3.1415926、-6.5 等。当数值很大或很小时, 浮点数就会以科学计数法形式呈现, 如 `3e-10`, 等价于 3×10^{-10} 。此处, e 是科学

计数法的固定符号，用来分割底数部分和指数部分，也可以用 E 替代。

内置的 `isinstance()` 函数可以用来判断变量所对应的对象类型。

【练一练 2-8】

运行代码：

```
x1=-3.14
print(isinstance(x1,float))

x2=1.0
print(isinstance(x2,float))

x3=23E4
print(isinstance(x3,float))

x4=0
print(isinstance(x4,float))
```

运行结果如下：

```
True
True
True
False
```

注意：

- (1) 书写浮点数时必须包含一个小数点，否则会被 Python 当作整数处理。
- (2) 以科学计数法形式表示的数值都是浮点数，如 23E4 等价于 230 000，但 23E4 是浮点数。
- (3) 浮点数不精确存储，能保证 15 位的准确性，超过 15 位会产生误差。

3. 复数 (Complex)

复数是数学中的概念，由实数部分 (real) 和虚数部分 (imag) 构成。Python 中复数的虚部以 “j” 或者 “J” 作为后缀，具体格式为 “a+bj”，也可以用 `complex(a,b)` 表示，复数的实部 a 和虚部 b 都是浮点数。

【练一练 2-9】

运行代码：

```
x=2+3j
y=1.2+3.4j
print(x.real)
print(y.imag)
print(isinstance(x+y,complex))
z=complex(1,2)
print(z)
```

运行结果如下：

```
2.0
3.4
True
(1+2j)
```

(二) 布尔值 (Boolean)

布尔值是用来进行是非判断的数据类型。布尔值比较特殊，它有且仅有两个值 True 和 False。而其他常见的数据类型的值是由用户自定的、无固定值，如数字、字符串。布尔值经常用来进行条件判断，若判断为真则执行特定条件中的代码块，若为假则执行相反的指定条件内的代码块或不执行任何内容。

1. 布尔真值 (True)

数据对象如果具有某种内容，则几乎所有值都将认定为 True。任何非零数字均为 True，任何非空的字符串、列表、元组、集合和字典都为 True。

2. 布尔假值 (False)

可以认定为假的值除了 False 本身外，还包括数字 0、空字符串、空列表、空元组、空集合和空字典和计算结果为 None 的值。常见的布尔假值类型见表 2-3。

表 2-3 常见的布尔假值类型

认定为假的值	含义	数据类型
False	布尔假值	Boolean
0	整数 0	Int
" "	空字符串	String
[]	空列表	List
()	空元组	Tuple
{ }	空字典	Dictionary
None	空值	NoneType

【练一练 2-10】

运行代码：

```
print(6>7)
print(6==7)
print(6<7)
print(0)
print(bool(""))
print(bool(" "))
print(bool("小明是会计人员！"))
```

```
print(True and False)
print(False or True)
print(True<False)
print(True*3)
print(False -1)
```

运行结果如下:

```
False
False
True
0
False
True
True
False
True
False
3
-1
```

注意:

布尔值 True 进行数值运算时识别为 1, False 识别为 0。

【提示】

bool(x) 函数用以判断 x 是否为真, 并返回 True 或 False。

(三) 字符串 (String)

若干个字符的集合就是一个字符串, 可以对应文本来理解。Python 中的字符串一般使用单引号、双引号、三引号进行界定, 字符串的内容可以包含数字、字母、标点、特殊符号、中文等。Python 字符串中的单引号、双引号、三引号没有任何区别, 一律原样输出, 而且它们之间可以相互嵌套, 用来表示复杂字符串。

下面都是合法的字符串:

- "123456"
- "123abc"
- "I'm an accountant"
- "https://www.Python.org"
- "有借必有贷, 借贷必相等"

1. 字符串的索引

Python 的索引是双向索引, 包括正索引和负索引, 分别使用正向递增序号和反向递减序号, 需要注意的是, 正索引值从 0 开始计数。以字符串 “Accountant” 的索引为例, 索引方向见图 2-1。

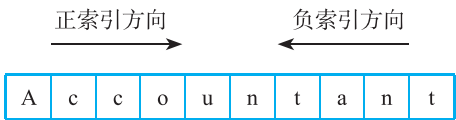


图 2-1 字符串的索引方向

正索引从字符串左边第一个字符开始，第一个字符的索引为 0，第二个字符的索引为 1，以此类推；负索引从字符串右边第一个字符开始，第一个字符的索引为 -1，第二个字符的索引为 -2，以此类推。以字符串“Accountant”的索引为例，索引号见表 2-4。

表 2-4 字符串的索引号

字符串	A	c	c	o	u	n	t	a	n	t
正索引号	0	1	2	3	4	5	6	7	8	9
负索引号	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

【练一练 2-11】

运行代码：

```
s="你好，会计！"
print(s[0])
print(s[2])
print(s[-3])
```

运行结果如下：

```
你
，
会
```

2. 字符串的运算

在字符串运算上，Python 语言提供了连接运算符，用于连接多个字符串；重复运算符，用于重复输出字符串内容；访问运算符，用于获取字符串中的单个字符和任意子串；成员运算符，用于判定字符或子串是否出现在字符串中。常见的操作符类型见表 2-5。

表 2-5 常见的操作符类型

操作符类型	符号表示	含义
连接运算符	+	把两个字符串合并成一个新的字符串，与数值运算的“+”运算符有不同的意义
重复运算符	*	多次重复输出同一字符串，重复次数由运算符后面的操作数指定
访问运算符	[]	获取字符串序列中的单个字符
	[:]	获取字符串序列中的一部分
成员运算符	in	判断一个字符或者一个子串是否在当前字符串中
	not in	判断一个字符或者一个子串是否不在当前字符串中

【练一练 2-12】

运行代码：

```
a="我要成为"
b="一名合格的会计"
print(a+b)
print(b*3)
print(a[2])
print(b[1:5])
print("会计" in a)
print("会计" not in a)
```

运行结果如下：

```
我要成为一名合格的会计
一名合格的会计一名合格的会计一名合格的会计
成
名合格的
False
True
```

3. 字符串的格式化输出

Python 中字符串的格式化输出是将其他类型对象按格式转化为字符串，或者嵌入其他字符串中进行输出。常见的格式字符类型及其含义见表 2-6。

表 2-6 常见的格式字符类型及其含义

格式字符类型	含义
%s	格式化成字符串
%c	格式化成字符及其 ASCII 码
%d	格式化成整数
%o	格式化成八进制整数
%x	格式化成十六进制整数
%e、%E	用科学计数法格式化浮点数（基底分别为 e、E）
%f	格式化成浮点数字，可指定小数点后的精度

尽管字符串的格式化输出过程有时会用到非常复杂的表达式，但最基本的用法是将一个值嵌入一个有格式字符的字符串中。

【练一练 2-13】

运行代码:

```
str1="%s"% 56
print(str1)
str2="%d, %c"% (56, 56)
print(str2)
str3="My name is %s"% ("Xiaoming")
print(str3)
str4="我的职业是%s, 工作经验 %d 年"% ("会计", 12)
print(str4)
```

以上示例中, 字符串 str1 中的第一个 “%” 是格式字符的标志, 与字母 “s” 组合成了格式字符 “%s”, 表示需要将进行格式化的对象 “56” 转化为字符串的格式输出。str1 中的第二个 “%” 是格式字符串 “%s” 与需要进行格式化的对象 “56” 之间的间隔标志。这里, 格式字符除了起到确定字符串的格式外, 还起到了占位作用, 以便间隔标志 “%” 后的被格式化对象嵌入到字符串中进行输出。

运行结果如下:

```
56
56, 8
My name is Xiaoming
我的职业是会计, 工作经验 12 年
```

(四) 列表 (List)

列表是有序数据的集合。定义的语法是使用中括号括起来, 以逗号分隔数据。列表中的元素可以是任意类型的, 包括数字、字符串、列表等。同一个列表可以包含不同类型的数据, 列表中的元素也可以重复出现。

创建一个列表, 只要把逗号分隔的不同的数据项使用中括号括起来即可。与字符串的索引一样, 列表索引从 0 开始。列表可以进行截取、组合等。

【练一练 2-14】

运行代码:

```
list1=[1, 2, 3, 4, 2]
print(list1)
list2=["记账凭证", "财务账簿", "财务报表"]
print(list2[2])
list3=["小明", 25, [175, 60], "财务管理"]
print(list3[1])
list4=[["预算会计", "财务会计"], ["财务会计", "管理会计"]]
print(list4[0])
```


运行结果如下:

```
[1, 2, 3, 4, 2]
财务报表
25
['预算会计', '财务会计']
```

1. 列表元素的增加和删除

列表元素的增加可以使用 `append` 函数、`extend` 函数和 `insert` 函数。`append` 函数是从列表的尾部加入单个新的元素，`extend` 函数是从列表的尾部加入多个新的元素，`insert` 函数是在指定的索引位置插入元素。

【练一练 2-15】

运行代码:

```
list=["资产", "负债"]
print(list)
list.append("收入")
print(list)
list.extend(["费用", "利润"])
print(list)
list.insert(2, "所有者权益")
print(list)
```

运行结果如下:

```
['资产', '负债']
['资产', '负债', '收入']
['资产', '负债', '收入', '费用', '利润']
['资产', '负债', '所有者权益', '收入', '费用', '利润']
```

列表元素的删除可以使用 `del` 函数，`del` 函数可以删除列表中指定索引位置的元素，也可以删除整个列表。

【练一练 2-16】

运行代码:

```
list=["资产", "负债", "所有者权益", "收入", "费用", "利润"]
print(list)
del list[3]
print(list)
del(list)
print(list)
```

运行结果如下：

```
['资产', '负债', '所有者权益', '收入', '费用', '利润']
['资产', '负债', '所有者权益', '费用', '利润']
<class 'list'>
```

2. 列表元素的切片操作

列表元素的切片操作是根据需要从原始列表中提取列表的一部分的过程。Python 中符合序列的有序序列都支持切片，如列表、字符串、元组。

其语法格式如下：

```
list[start_index:end_index:step]
```

其中 list 为需要进行切片操作的列表，start_index、end_index、step 是切片操作的参数。

(1) start_index 表示切片起始位置的索引，包括该索引位置本身，缺省时默认为 0。

(2) end_index 表示切片截止位置的索引，截取时索引为 end 的元素不会被截取到新列表中，缺省时默认为整个列表长度。

(3) step 表示切片的步长，其大小表示间隔截取元素时的间隔长度；其正负表示截取的方向。step 为正数时从左向右截取，为负数时从右向左截取，缺省时默认为 1。

【练一练 2-17】

运行代码：

```
list1=["短期借款","应付票据","应付账款","预收款项","应付职工薪酬","应交税费","应
付利息","应付股利"]
list2=list1[2:6:1]
print(list2)
list3=list1[1:8:-2]
print(list3)
list4=list1[:7:3]
print(list4)
list5=list1[1::2]
print(list5)
list6=list1[0:5]
print(list6)
list7=list1[::-3]
print(list7)
list8=list1[::]
print(list8)
```



【微课】
列表元素的切片
操作

运行结果如下:

```
['应付账款', '预收款项', '应付职工薪酬', '应交税费']  
[]  
['短期借款', '预收款项', '应付利息']  
['应付票据', '预收款项', '应交税费', '应付股利']  
['短期借款', '应付票据', '应付账款', '预收款项', '应付职工薪酬']  
['短期借款', '预收款项', '应付利息']  
['短期借款', '应付票据', '应付账款', '预收款项', '应付职工薪酬', '应交税费', '应付利息', '应付股利']
```

3. 列表元素的排序

列表元素可以使用 `sort` 函数配合参数 `reverse` 进行升序排序或者降序排序, 也可使用 `reverse` 函数进行元素逆序排序。具体用法如下。

- (1) `list.sort()` 中缺省了参数 `reverse`, 默认 `reverse=False`, 表示将列表 `list` 中的元素进行升序排序;
- (2) `list.sort(reverse=True)` 表示将列表 `list` 中的元素进行降序排序;
- (3) `reverse` 作为函数使用时, 表示列表中的元素逆序排序, 即新生成的列表中元素为原列表中元素逆序排列。

【练一练 2-18】

运行代码:

```
list=[30, 70, 50, 40, 10, 20]  
print(list)  
list.sort()  
print(list)  
list=[30, 70, 50, 40, 10, 20]  
list.sort(reverse=True)  
print(list)  
list=[30, 70, 50, 40, 10, 20]  
list.reverse()  
print(list)
```

运行结果如下:

```
[30, 70, 50, 40, 10, 20]  
[10, 20, 30, 40, 50, 70]  
[70, 50, 40, 30, 20, 10]  
[20, 10, 40, 50, 70, 30]
```

(五) 字典 (Dictionary)

在前面介绍的数据类型中, 一个元素只能表示一个信息, 为了存储成对出现的关联信息, 需

要引入新的数据类型——字典。字典是由多个键值对构成的无序集合，所以字典的元素就是键值对（key:value）。键（key）可以理解为元素的索引，值（value）是元素的数值，键和值之间是影射关系。形式上，字典中每个元素都是一对键值对，键和值之间用冒号连接，键值对之间用逗号隔开，所有的键值对都放在花括号中。

字典的主要特征有以下三种。

（1）无序性。与数据类型字符串、列表不同，字典中的元素是无序的，无法用数字索引访问，只能通过元素的键来访问。

（2）键的唯一性。字典不支持同一个键多次出现，否则仅保留最后一个键值对。元素的键唯一，值可以不唯一。

（3）键的不变性。字典中每个元素的键是不可变的，只能使用数字、字符串等，不能使用列表。

1. 字典元素的修改与添加

字典元素的修改与添加，可以直接访问字典元素的键来赋值。若访问的键存在，表示修改对应键值对的值；若不存在，表示增加新的键值对，也可以利用 update 函数来修改和添加字典元素。

【练一练 2-19】

运行代码：

```
dict1={"姓名":"小明","年龄":25,"岗位":"成本会计"}
print(dict1)
dict1["年龄"]=26
print(dict1)
dict1["性别"]="男"
print(dict1)
dict1.update({"爱好":"打篮球","岗位":"财务会计"})
print(dict1)
```

运行结果如下：

```
{'姓名': '小明', '年龄': 25, '岗位': '成本会计'}
{'姓名': '小明', '年龄': 26, '岗位': '成本会计'}
{'姓名': '小明', '年龄': 26, '岗位': '成本会计', '性别': '男'}
{'姓名': '小明', '年龄': 26, '岗位': '财务会计', '性别': '男', '爱好': '打篮球'}
```

2. 字典元素的删除

字典元素的删除方法很多，常见的有利用 pop 函数和 del 函数删除访问键位置的元素，或者使用 clear 函数删除字典中所有的元素。

【练一练 2-20】

运行代码：

```
dict1={"姓名":"小明","年龄":25,"岗位":"成本会计","性别":"男","爱好":"打篮球"}
print(dict1)
dict1.pop("年龄")
print(dict1)
del dict1["性别"]
print(dict1)
dict1.clear()
print(dict1)
```

运行结果如下：

```
{'姓名': '小明', '年龄': 25, '岗位': '成本会计', '性别': '男', '爱好': '打篮球'}
{'姓名': '小明', '岗位': '成本会计', '性别': '男', '爱好': '打篮球'}
{'姓名': '小明', '岗位': '成本会计', '爱好': '打篮球'}
{}
```

二、运算符

运算符就是用来操作 Python 各数据类型的符号。常见的运算符包括算术运算符、比较运算符、赋值运算符、逻辑运算符、成员运算符等，运算由变量和运算符构成，同时要考虑运算符的优先级。

1. 算术运算符

算术运算符主要是对两个对象进行算术计算的符号，其运算逻辑与数学的概念相似。常见的算数运算符见表 2-7。

表 2-7 常见的算术运算符

运算符	描述	示例（假设 a=12,b=5）
+	加法	a+b 的输出结果: 17
-	减法	a-b 的输出结果: 7
*	乘法	a*b 的输出结果: 60
/	除法	a/b 的输出结果: 2.4
**	幂	a**b 的输出结果: 248 832
//	取整商	a//b 的输出结果: 2
%	取余数	a%b 的输出结果: 2

2. 比较运算符

两个对象进行比较，其运算对象可以是数值也可以是字符串。常见的比较运算符见表 2-8。

表 2-8 常见的比较运算符

运算符	描述	示例（假设 a=12,b=5）
>	大于	a>b 的输出结果: True
>=	大于等于	a>=b 的输出结果: True
<	小于	a<b 的输出结果: False
<=	小于等于	a<=b 的输出结果: False
!=	不等于	a!=b 的输出结果: True
==	等于	a==b 的输出结果: False

3. 赋值运算符

赋值运算符是编程开发中最常用的运算符，即对一个对象进行赋值，将运算符右侧的值赋值给左侧的变量。常见的赋值运算符见表 2-9。

表 2-9 常见的赋值运算符

运算符	描述	示例
=	赋值	a=b, 将 b 的数值赋值给 a
+=	加法赋值	a+=b 等价于 a=a+b
-=	减法赋值	a-=b 等价于 a=a-b
=	乘法赋值	a=b 等价于 a=a*b
/=	除法赋值	a/=b 等价于 a=a/b
=	幂赋值	a=b 等价于 a=a**b
//=	取整商赋值	a//=b 等价于 a=a//b
%=	取余数赋值	a%=b 等价于 a=a%b

4. 逻辑运算符

用于逻辑预算的符号，一般用于判断两个变量的交并集，一般返回一个布尔值，其运算原理与数学中的逻辑运算相同。常见的逻辑运算符见表 2-10。

表 2-10 常见的逻辑运算符

运算符	描述	示例
and	逻辑与	True and False 的输出结果: False
or	逻辑或	True or False 的输出结果: True
not	逻辑非	not True 的输出结果: False

5. 成员运算符

用于判断两个对象是否存在包含关系，即一个对象中是否包含另外一个对象，其返回结果为布尔值。常见的成员运算符见表 2-11。

表 2-11 常见的成员运算符

运算符	描述	示例
in	判断对象是否在序列中	1 in [1,2,3] 的输出结果: True
not in	判断对象是否在序列中	1 not in [1,2,3] 的输出结果: False

6. 运算符的优先级

Python 运算符的优先级从高到低：算术运算符>比较运算符>赋值运算符>成员运算符>逻辑运算符。若有括号，括号的优先级最高。优先级高的运算符优先计算或处理，同级别的运算符从左往右按顺序计算。具体运算符的优先级见表 2-12。

表 2-12 运算符的优先级

运算符	所属类型
**	算术运算符
* / % //	算术运算符
+ -	算术运算符
< = < > > =	比较运算符
== !=	比较运算符
= %= /= //= -= += *= **=	赋值运算符
in not in	成员运算符
and or not	逻辑运算符

技能训练任务 员工信息管理

情景描述

创艺科技公司创始于 2015 年 4 月 26 日，是一家专注于将大数据技术应用于信息管理、财经行业领域的科技创新型公司。公司根据当前目标客户行业领域精细化管理的要求，利用大数据技术和对业务场景的理解，在数据采集、治理、建库等基础上开发各类实战化的大数据模型，并通过大数据技术向各个层级的客户提供业务支撑及数据服务。

伴随着信息化时代的发展，公司业务规模不断扩大，需要有新鲜血液加入公司。经人事部门层层筛选，招聘了若干名员工入职。现需将这些员工的相关信息（见表 2-13）进行登记管理，以便后续进行工资薪酬核算和社保登记。公司将这项任务交给了小刘后，他陷入了沉思，如何能够高效准确地录入员工信息？如何让系统自动检测相关信息有无填写错误？另外，在后续需要查看某位员工的相关资料时，如何调用与核对信息呢？

表 2-13 新进员工信息

部门	姓名	性别	身份证号	手机号	岗位
财务部	李兰	女	31020219990403××××	12901241300	出纳

续表

部门	姓名	性别	身份证号	手机号	岗位
市场部	刘晓	男	32120219970821××××	12371519566	业务员
.....					

流程设计

员工信息管理流程设计见图 2-2。



图 2-2 员工信息管理流程设计

任务实施



【微课】
技能训练任务——
员工信息管理

一、员工信息采集登记

1. 业务分析

新进员工信息包含部门、姓名、性别、身份证号、手机号、岗位六项信息，故可以用列表或字典的方式进行员工信息归集。

2. 编写代码

```
s1=["财务部","李兰","女","31020219990403××××","12901241300","出纳"]
s2=["市场部","刘晓","男","32120219970821××××","12371519566","业务员"]
```

二、系统检测核对信息

1. 业务分析

在录入员工信息时，特别容易在数字较多的身份证号、手机号等信息上填写错误，由于这类数字有固定的长度（如我国居民身份证号为 18 位、手机号为 11 位），故可以通过编写代码，检验字符长度，并根据填写是否符合要求给出提示。

2. 编写代码，检验身份证号的长度

```
lens=[len(s1[3]),len(s2[3])]
for i in lens:
    if i==18:
        print("身份证号长度正确")
    else:
        print("身份证号长度不正确，请重新输入")
```


运行结果如下：

```
身份证号长度正确  
身份证号长度正确
```

两名新员工的身份证号码长度均为 18 位，不存在多录或少录数字的情况。

【提示】

要对所有录入的员工身份证号码长度进行检验，这里运用到了 if...else... 分支结构和 for...in... 循环，该语法在项目三中有详细介绍。

3. 编写代码，检验手机号的长度

```
lens=[len(s1[4]),len(s2[4])]  
for i in lens:  
    if i==11:  
        print("手机号码长度正确")  
    else:  
        print("手机号码长度不正确，请重新输入")
```

执行代码后，运行结果如下：

```
手机号码长度正确  
手机号码长度正确
```

两名新员工的手机号码长度均为 11 位，不存在多录或少录数字的情况。

三、调用员工信息

1. 业务分析

当录入若干企业员工信息后，可以通过编写代码调用某名员工的具体信息，通过查看调用的员工信息，可以检查和核对该员工信息是否录入准确。

2. 编写代码，调用职工李兰的相关信息

```
s=[s1,s2]  
name=input("请输入员工姓名:")  
for a in s:  
    if name in a:  
        print(a)
```

运行以上代码后，在提示词后面输入“李兰”的名字，继续运行代码，运行结果如下：

```
请输入员工姓名：李兰  
['财务部', '李兰', '女', '31020219990403××××', '12901241300', '出纳']
```

德技并修

当今，大数据伴随着云计算、移动互联网的发展，正在对全球经济社会产生巨大的影响。随着大数据时代的到来和市场经济的不断发展，简单的数据分析已经不能满足会计信息使用者的需求，在大数据环境下，会计人员需要运用数据库对企业的财务数据和其他部门的数据进行综合分析，从不同的角度和深度挖掘可用信息，通过这些数据的隐藏信息了解企业的现实情况，并及时对企业的经营成果进行客观的评价。由此可以看出，大数据时代不再要求会计人员进行简单的机械性工作，而是要求其能够熟练运用各种数据模型并拥有一定的数据敏感度，熟练掌握计算机的应用程序并适当拥有一些自己编程建造数据模型的能力。那么，怎么提升会计人员的数据分析能力呢？

思考建议

Python 是大数据、云计算领域应用最广泛的编程语言，在网络爬虫、科学计算、可视化分析、人工智能等方面均有广泛的应用。初学 Python 语言时，需要培养严谨细致、精益求精的“工匠精神”，认真掌握相应代码的编写规则，养成良好的编写习惯，编写代码时要仔细检查每个变量名是否合理，代码缩进是否规范，确保上下代码段的对应关系，降低出错率，提高工作效率。

技能提升

理论夯实

单项选择题

- 根据命名原则，以下定义变量不合法的是（ ）。

A. and1=1+0
B. 1and=2
C. 总数 =2+1
D. and=4
- 以下为输入函数的是（ ）。

A. input()
B. sum()
C. print()
D. split()
- 执行以下语句：


```
y=5
x=y**2
y*=2
print(x==y)
```

 输出的结果为（ ）。

A. 25
B. 10
C. True
D. False

业务实训题

打开项目二业务实训代码文件，将文件中的全部内容复制到本地的 Python 代码编辑器中（或录入后附程序代码），请在下划线处编写代码，不修改其他代码，实现以下功能。

案例材料：新宁家具制造有限公司采用“基本工资+绩效工资”的方式发放销售部门的工资，张华的基本工资是5 000元，本月共销售150件产品，销售单价是2 200元，绩效工资依据销售额计算提成，提成率是0.05%。

要求：编辑代码输入张华的销售数量、销售单价、提成率，并计算其当月的工资。

程序代码：

```
# 请在下划线处使用一行代码或表达式替换，不要修改其他已给出的代码
# 定义基本工资
basic_salary=_____
# 输入本月的销售数量
quantity=int(input("_____"))
# 输入销售单价
price=float(input("请输入销售单价"))
# 输入提成率
rate=float(input("请输入销售提成率"))
# 计算本月的工资
salary=_____
# 打印输出
print("张华本月的工资是 %.2f 元"%(salary))
```

学习评价

本项目内容学习结束，请填写学习任务完成评价表（见表 2-14）。

表 2-14 学习任务完成评价表

评价范围	评价标准	自我评价 (五星制打分)	小组评价 (五星制打分)	教师评价 (五星制打分)
职业知识	掌握标准输出语句的基本用法			
	掌握标准输入语句的基本用法			
	掌握变量的赋值与命名规则			
	掌握数据类型和运算符的基本知识			
职业能力	能够使用 print 语句进行财务数据输出			
	能够使用 input 语句进行财务数据输入			
	能够使用变量对财务数据进行命名和赋值			
	能够创建、修改常见数据类型的对象，并结合运算符使用内置函数进行运算			
职业素养	具备正确的技能观、工匠精神			
	重视编程规范			