

 高等院校软件应用系列教材

# Python

## 程序设计实践

主 编 马亚军 刘振东 孔令信

主 审 甘利杰



重庆大学出版社

## 内容提要

本书是《Python 程序设计》的配套实践教材,同时也可以与其他 Python 程序设计教材配合使用。全书共分为两部分。第一部分为实验指导,结合主教材设计了 19 个实验,每个实验都和主教材的教学内容契合。实验内容既包含与 Python 语法规则相关的内容,也包含许多实际问题的程序设计,从而增强读者的学习兴趣,提高读者分析问题和解决问题的能力。读者在学习过程中可以根据实际情况从每个实验中选择部分内容作为上机练习。第二部分为配套练习题,包含 8 个部分的习题内容,基本覆盖了主教材的主要章节和知识点,可以帮助读者很好地巩固主教材各章知识要点。

### 图书在版编目(CIP)数据

Python 程序设计实践/马亚军,刘振东,孔令信主  
编.--重庆:重庆大学出版社,2021.3  
高等院校软件应用系列教材  
ISBN 978-7-5689-2603-4

I.①P... II.①马... ②刘... ③孔... III.①软件工  
具—程序设计—高等学校—教材 IV.①TP311.561

中国版本图书馆 CIP 数据核字(2021)第 040877 号

## Python 程序设计实践

Python CHENGXU SHEJI SHIJIAN

主 编 马亚军 刘振东 孔令信  
副主编 杨方捷 田凌燕 谢克武 韦贇圣  
主 审 甘利杰

策划编辑:鲁 黎

责任编辑:文 鹏 版式设计:鲁 黎  
责任校对:谢 芳 责任印制:张 策

\*

重庆大学出版社出版发行

出版人:饶帮华

社址:重庆市沙坪坝区大学城西路 21 号

邮编:401331

电话:(023) 88617190 88617185(中小学)

传真:(023) 88617186 88617166

网址: <http://www.cqup.com.cn>

邮箱: [fxk@cqup.com.cn](mailto:fxk@cqup.com.cn) (营销中心)

全国新华书店经销

重庆升光电力印务有限公司印刷

\*

开本:787mm×1092mm 1/16 印张:11 字数:257 千

2021 年 3 月第 1 版 2021 年 3 月第 1 次印刷

印数:1—3 500

ISBN 978-7-5689-2603-4 定价:38.00 元

---

本书如有印刷、装订等质量问题,本社负责调换  
版权所有,请勿擅自翻印和用本书  
制作各类出版物及配套用书,违者必究

## 前 言

上机实践是学习 Python 程序设计的重要环节,读者只有通过上机编程实践,才能熟练掌握 Python 的基础语法知识,充分理解程序设计的基本思想和方法,同时能更好地培养自学能力,学以致用。本书是《Python 程序设计》的配套实践教材,同时也可以与其他 Python 程序设计教材配套使用。

本实践教程共分为两部分。第一部分为实验指导,结合主教材设计了 19 个实验,每个实验都和主教材的教学内容配合。实验内容既包含与 Python 语法规则相关的内容,也包含许多实际问题的程序设计,从而增强读者的学习兴趣,提高读者分析问题和解决问题的能力。实验 1 主要练习 Python 开发环境的搭建和配置,以及熟悉 Python 编程规范。实验 2~7 主要涉及 Python 程序设计的基本语法练习,以及程序的基本算法和程序的控制流程。实验 8~13 主要涉及 Python 基本数据结构:列表、元组、字典、集合以及字符串的基本操作练习。实验 14 通过实践掌握函数的自定义方法,实验 15 涉及面向对象编程,实践类与对象的基本操作。实验 16、17 为正则表达式和网络爬虫实战练习。实验 18 是数据分析的综合案例,实验 19 主要练习 Python 标准库 turtle 绘制图形的方法。读者在学习过程中可以根据实际情况从每个实验中选择部分内容作为上机练习。第二部分为配套的练习题,包含了 8 部分习题内容,基本覆盖了主教材的主要章节和知识点,可以帮助读者很好地巩固对主教材各章知识要点的掌握。

本书由马亚军、刘振东、孔令信任主编,杨方捷、田凌燕、谢克武、韦贇圣任副主编;甘利杰任主审。其中,甘利杰负责本书的总体策划和后期初审工作,马亚军负责实验 1、2、3、4、5、19 的编写以及第二部分练习题的整理。刘振东负责实验 6、7、15 的编写。杨方捷负责实验 8、9、10、11 的编写。田凌燕负责实验 12、13、14 的编写。谢克武负责实验 16、17 的编写,韦贇圣负责实验 18 的编写。孔令信负责第二部分练习题的编写工作。马亚军负责全书统稿。

由于编写时间紧迫及编者水平所限,书中难免存在疏漏之处,敬请读者批评指正。

编 者  
2020 年 7 月

# 目 录

## 第 1 部分 实验指导

实验 1	Python 环境实验 .....	2
实验 2	程序基础与算法 .....	14
实验 3	运算符和表达式练习 .....	18
实验 4	顺序结构 .....	27
实验 5	分支结构 .....	32
实验 6	循环结构实验 .....	36
实验 7	多重循环及辅助语句实验 .....	40
实验 8	序列实验 .....	45
实验 9	字符串实验 .....	50
实验 10	列表实验 .....	55
实验 11	元组实验 .....	61
实验 12	字典练习 .....	65
实验 13	集合练习 .....	71
实验 14	函数练习 .....	75
实验 15	类与对象的基本操作实验 .....	78
实验 16	正则表达式 .....	83
实验 17	网络爬虫 .....	88
实验 18	数据分析 .....	94
实验 19	Turtle 库实验 .....	99

## 第 2 部分 练习题

习题 1	初识 Python .....	110
习题 2	Python 语言基础 .....	113
习题 3	程序流程控制 .....	121
习题 4	Python 的组合数据结构 .....	131
习题 5	字典与集合 .....	140
习题 6	函数 .....	155
习题 7	面向对象程序设计 .....	163
习题 8	文件 .....	167
参考文献	.....	169

# 第 1 部分

---

# 实验指导

# 实验 1 Python 环境实验

## 【实验目的】

- 1.掌握在 Windows 系统中搭建 Python 开发环境的方法。
- 2.掌握运行 Python 编程环境的基本操作。
- 3.熟悉程序编写规范。

## 【实验内容】

- 1.Python 开发环境的下载与配置。
- 2.熟悉编程环境。
- 3.熟悉文件的新建、保存以及上传、下载。
- 4.编写第一个简单程序。

## 【实验步骤】

### 1.Python 开发环境的下载与配置

Python 开发环境比较多,可以根据自己的习惯进行选择,Python 官网提供了 IDLE 开发环境,除此之外还有 PyCharm、wingIDE、Eric、PythonWin、Eclipse+PyDew。本书选用了目前教学中使用较多的 Anaconda3。

#### (1) Anaconda3 下载

Anaconda3 是一个用于科学计算的 Python 发行版,支持 Linux、Mac、Windows,包含了众多流行的科学计算、数据分析的 Python 包,但不建议用于 Anaconda3 基于 web 的开发,下载网址:<https://www.anaconda.com/>。如果下载速度过慢,可以使用清华大学开源软件镜像,下载列表链接为:<https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/>,使用帮助文档链接为:<https://mirrors.tuna.tsinghua.edu.cn/help/anaconda/>。

#### (2) Anaconda3 安装教程

第一步:双击 Anaconda3-2020.07-Windows-x86\_64 安装包。

第二步:在如图 1-1 所示对话框中单击“Next”。

第三步:在如图 1-2 所示对话框中单击“I Agree”。

第四步:在如图 1-3 所示对话框中选择默认项,单击“Next”。

第五步:安装路径为默认,但也可更改安装路径,在如图 1-4 所示对话框中继续单击“Next”。

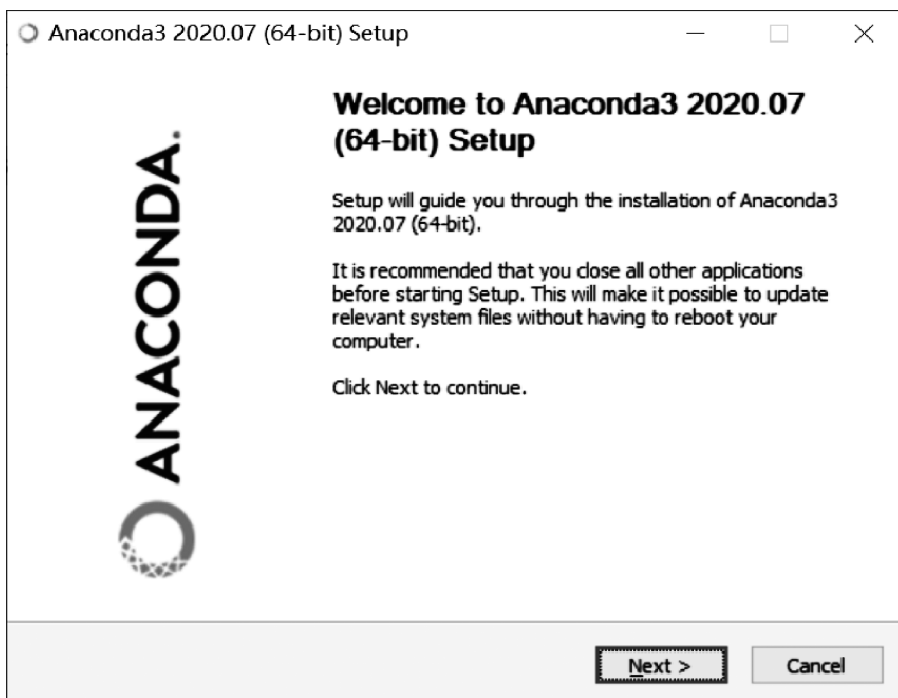


图 1-1 初始安装界面 1

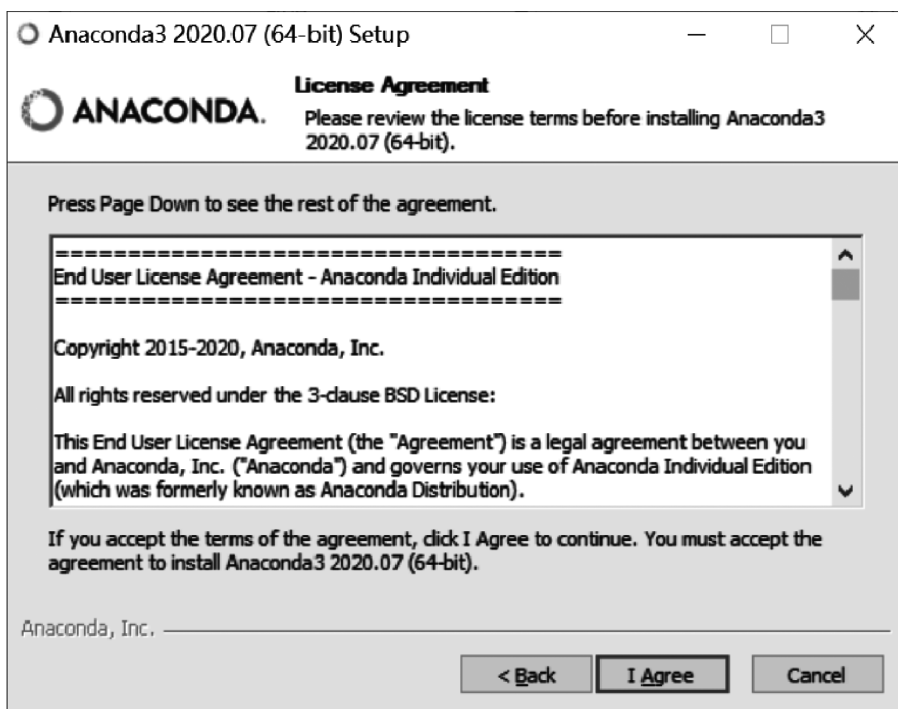


图 1-2 初始安装界面 2

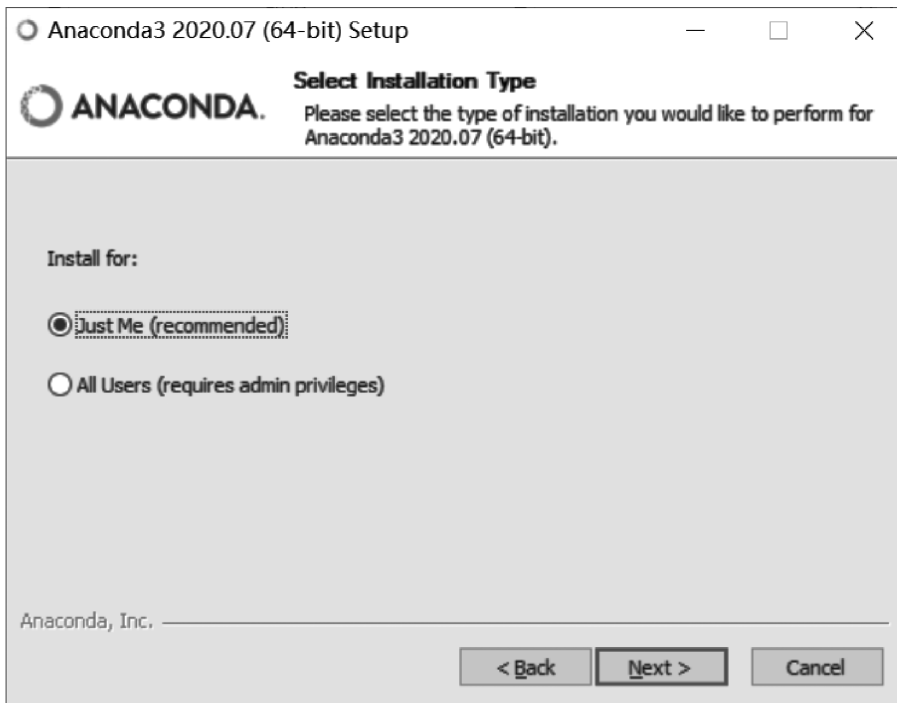


图 1-3 初始安装界面 3

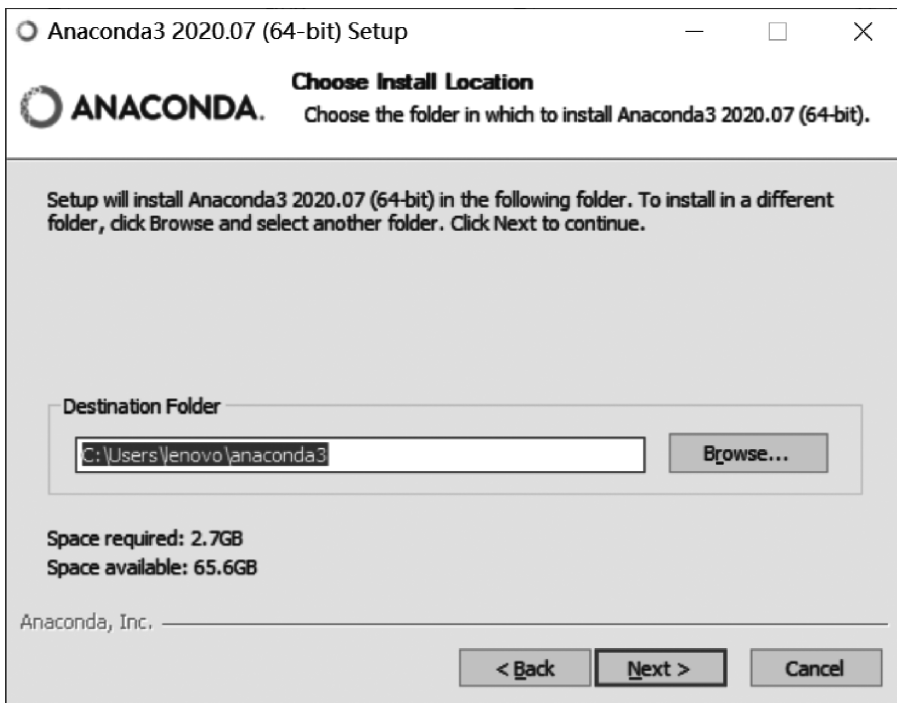


图 1-4 初始安装界面 4

第六步:在图 1-5 所示对话框中勾选“Add Anaconda to……”选项,单击“Install”。

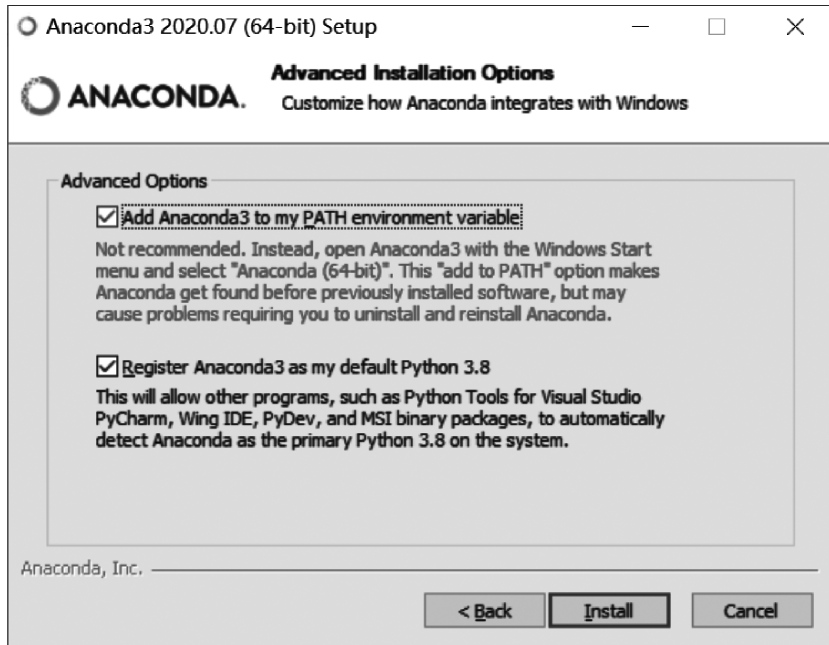


图 1-5 初始安装界面 5

第七步:等待安装,可能需要几分钟。

第八步:单击“Next”。

第九步:单击“Next”后继续点击图 1-6 所示对话框中的“Finish”,安装完成。

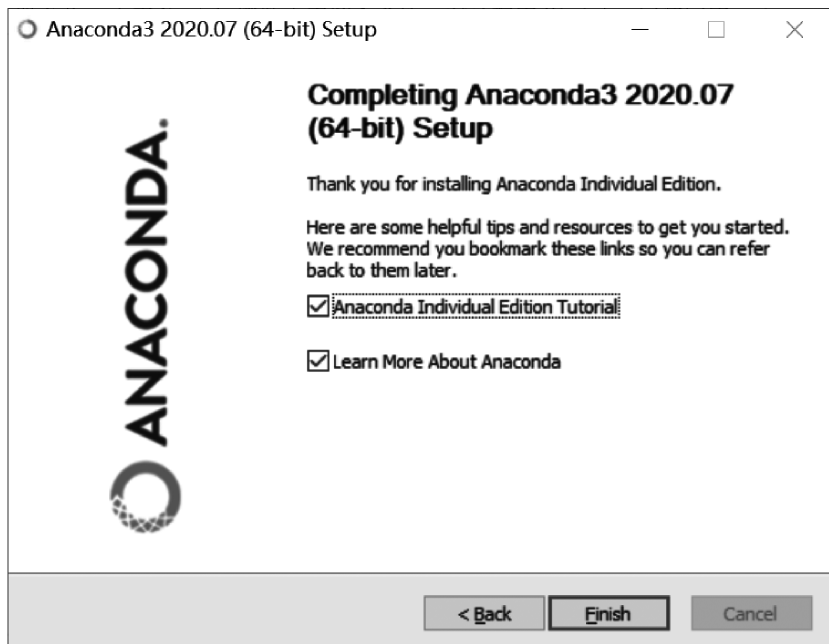


图 1-6 初始安装界面 6

## 2. 熟悉 Python 开发环境

Anaconda3 常用的开发环境是 Jupyter Notebook 和 Spyder, 如图 1-7 所示。

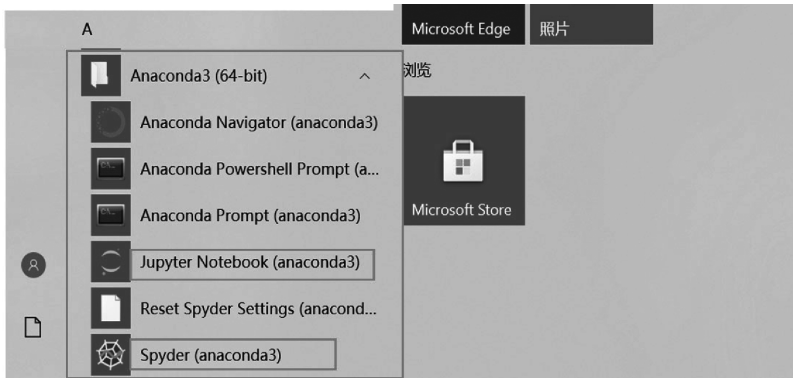


图 1-7 Anaconda3 菜单

### (1) Jupyter Notebook

#### 1) Jupyter Notebook 的启动

第一步: 在开始菜单中找到 Anaconda3 中的 Jupyter Notebook 点击运行, 出现图 1-8 所示服务器运行界面, 需要等几秒时间。

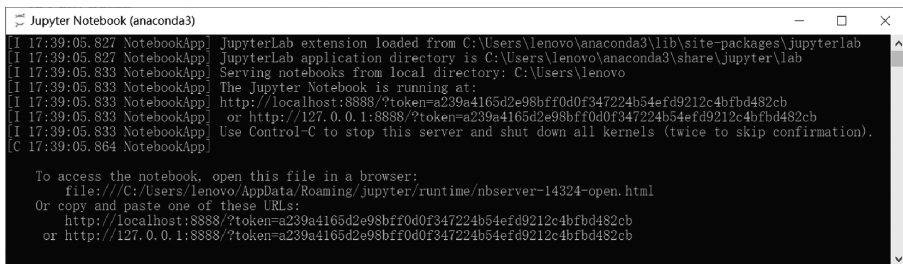


图 1-8 Jupyter Notebook 服务器运行界面

第二步: 当出现如图 1-9 所示 Jupyter Notebook Web 界面时, 在该界面右上方找到 New 下的 Python3 点击运行。

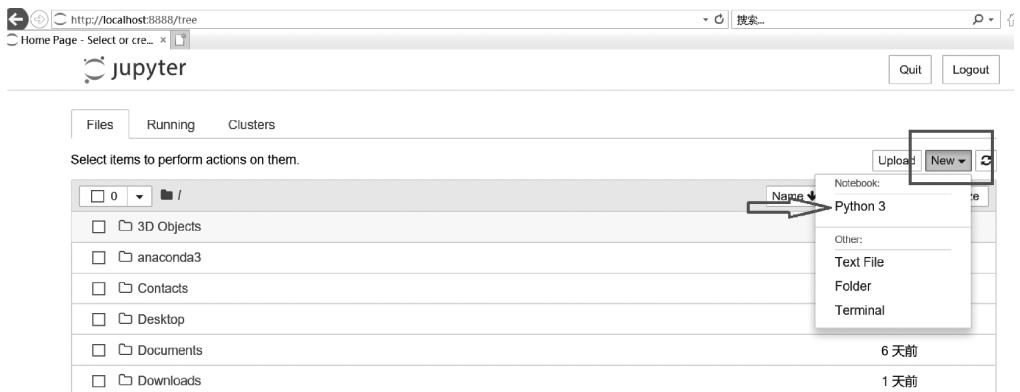


图 1-9 Jupyter Notebook Web 界面

第三步:开始编程之旅,在图 1-10 所示 Jupyter Notebook 交互式编程界面中 in[ ] 所在单元格的编辑框中输入 `print("hello python!")`,单击运行按钮,编辑框下方会显示程序的运行结果。

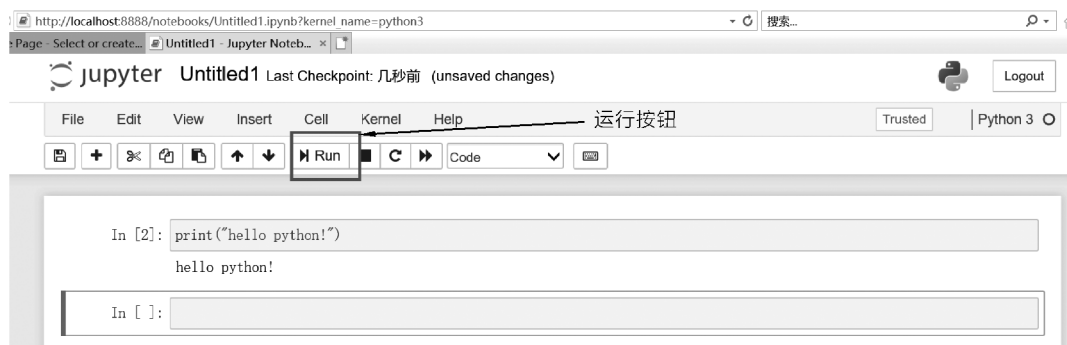


图 1-10 Jupyter Notebook 交互式编程界面

## 2) Jupyter Notebook 默认工作路径修改

第一步:程序菜单中找到 Anaconda3 列表中的 Jupyter Notebook (anaconda3) 选项,在右键菜单中选择“更多”子菜单中的“打开文件位置”选项,如图 1-11 所示;打开 Jupyter Notebook (anaconda3) 快捷方式所在文件夹,如图 1-12 所示。

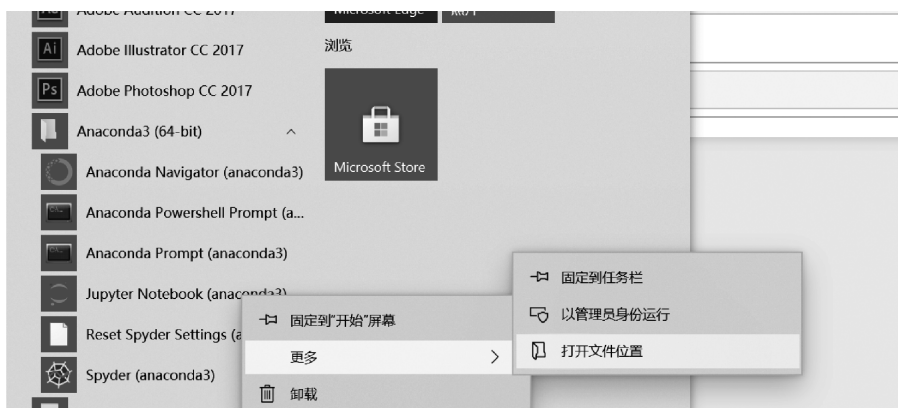


图 1-11 Jupyter Notebook(anaconda3) 打开文件位置菜单



图 1-12 Jupyter Notebook(anaconda3) 所在文件夹

第二步:在文件 Jupyter Notebook( anaconda3 )的右键菜单中选择“属性”选项,打开属性对话框。

第三步:将“属性”对话框中“目标”文本框内容中最后的文字“%USERPROFILE%/” (图 1-13(a)中选中部分)改为设定好的工作路径如:“D:\python3”,同时将起始位置中的内容也改为“D:\python3”,如图 1-13(b)所示。



图 1-13 属性对话框

第四步:重启 Jupyter Notebook,工作路径设定完成。

### 3) Jupyter Notebook 的基本操作

①新建文件:单击 Jupyter Notebook 界面右边的按钮“New”,选择下拉列表中的“python3”选项,创建一个新的页面文件,默认文件名为“Untitled”。

②文件重命名:单击 Jupyter 图标旁边的默认文件名“Untitled”,在弹出的“重命名”对话框中输入文件名“实验 1”,如图 1-14 所示。

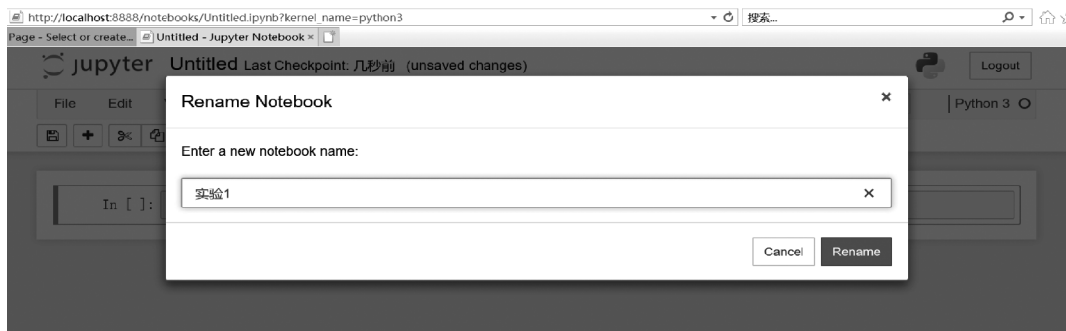


图 1-14 文件重命名

③Jupyter Notebook 界面功能介绍如图 1-15 所示。

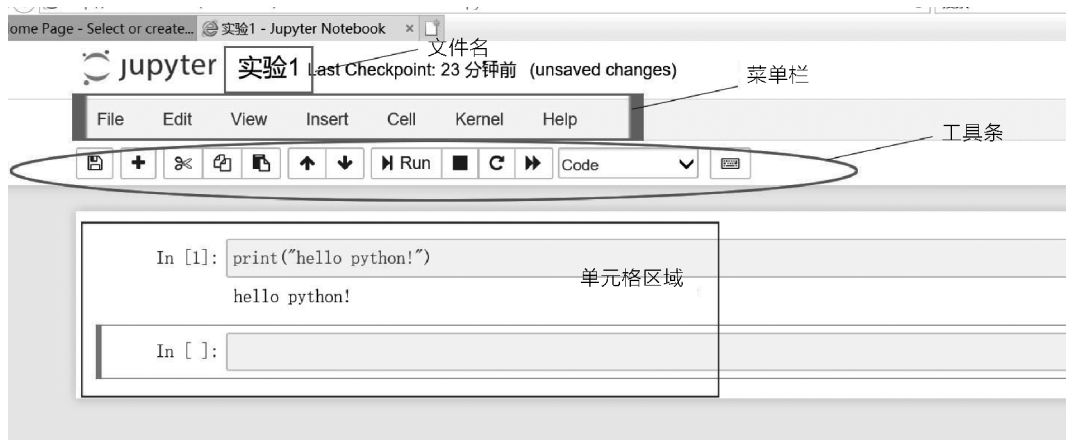


图 1-15 Jupyter Notebook 界面功能介绍

④Jupyter Notebook 单元格有两种模式:编辑模式和命令模式。

- 编辑模式由绿色单元格边框指示。当单元格处于编辑模式时,可以像普通文本编辑器一样键入单元格。按 Enter 键或使用鼠标单击单元格的编辑器区域可进入编辑模式。
- 命令模式由带有蓝色左边距的灰色单元格边框表示。

根据图 1-15 中,在第一个单元格中输入 `print("hello python! ")`,注意观察在编辑过程中和程序运行过程中单元格的变化。

Jupyter Notebook 单元格有 4 种常用状态:Code, Markdown, Heading, Raw NBconvert。其中,最常用的是前两个,分别是 code 代码状态,Markdown 编写状态。

根据图 1-16 所示,将第 2 个单元格切换到 Markdown 状态输入内容:“#练习 1:一个简单的程序”。单击工具条中的 run(运行)按钮,观察结果。再将“#”的数量增加,观察运行结果有什么不同。(注意:#和后面内容之间要有空格)

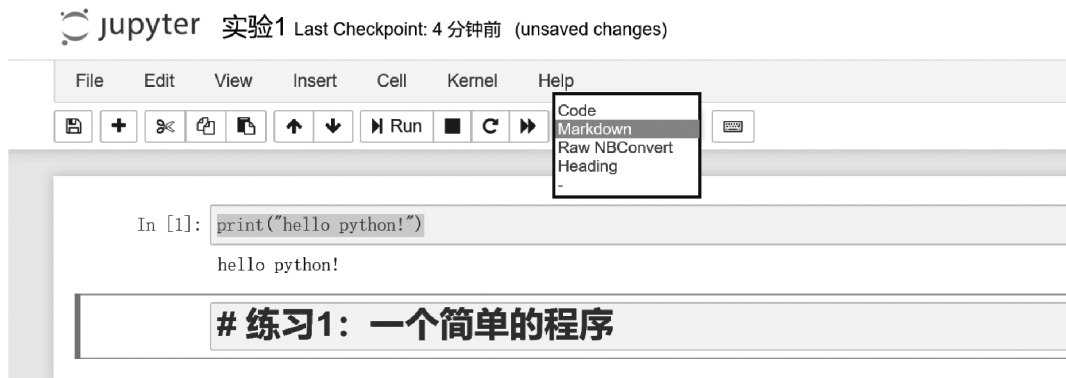


图 1-16 单元格 Markdown 状态

4) 常用快捷键

使用快捷方式可以节省程序员大量的时间并优化编程体验。表 1-1 中列出了一些快捷键,更多 Jupyter Notebook 内置的键盘快捷键,可以在“help”菜单栏下找到:“help”→“Help>Keyboard Shortcuts”。

表 1-1 Jupyter Notebook 常用快捷键

命令模式下(按 ESC 开启)			
Enter	转入编辑模式	K	选中上方单元
Shift+Enter	运行本单元,选中下个单元	Down	选中下方单元
Ctrl+Enter	运行本单元	J	选中下方单元
Alt+Enter	运行本单元,在其下插入新单元	A	在上方插入新单元
Y	单元转入 Code 代码状态	B	在下方插入新单元
M	单元转入 Markdown 状态	X	剪切选中的单元
R	单元转入 Raw 状态	C	复制选中的单元
1	设定 1 级标题	Shift+V	粘贴到上方单元
2	设定 2 级标题	V	粘贴到下方单元
3	设定 3 级标题	Shift+M	合并选中的单元
Up	选中上方单元	Ctrl+S	文件存盘
编辑模式 ( Enter 键启动)			
Tab	代码补全或缩进	Ctrl+Home	跳到单元开头
Shift+Tab	提示	Ctrl+Up	跳到单元开头
Ctrl+] ]	缩进	Ctrl+End	跳到单元末尾
Ctrl+[ [	解除缩进	Ctrl+Down	跳到单元末尾
Ctrl+A	全选	Ctrl+M	进入命令模式
Ctrl+Z	复原	Up	光标上移或转入上一单元
Ctrl+Shift+Z	再做	Down	光标下移或转入下一单元

## (2) Spyder

Anaconda3 中 Spyder 开发环境的功能介绍,如图 1-17 所示。

启动 Spyder 之后,可以使用主界面右下角控制台的交互模式,也可以使用主界面左侧的代码编辑区编写程序文件并直接运行。如果对界面风格不满意,可以通过“Tools”菜单的“Preferences”选项对应的对话框进行参数设置,如图 1-18 所示。

### 3. Python 编程风格

以下练习的代码都是基于 Jupyter Notebook 开发环境,但也同样适用于其他开发环境。

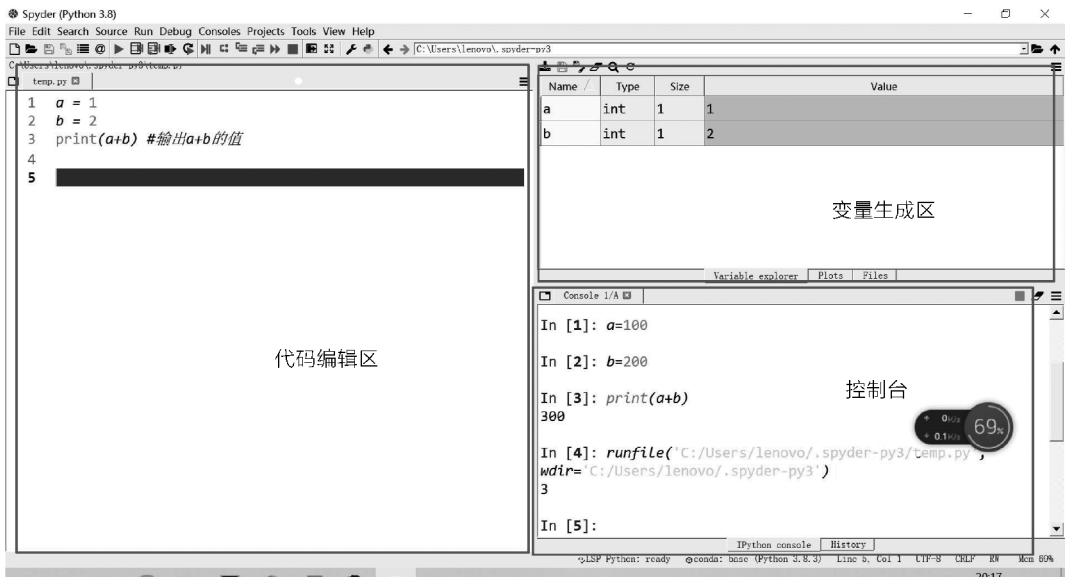


图 1-17 Spyder 开发环境界面功能介绍

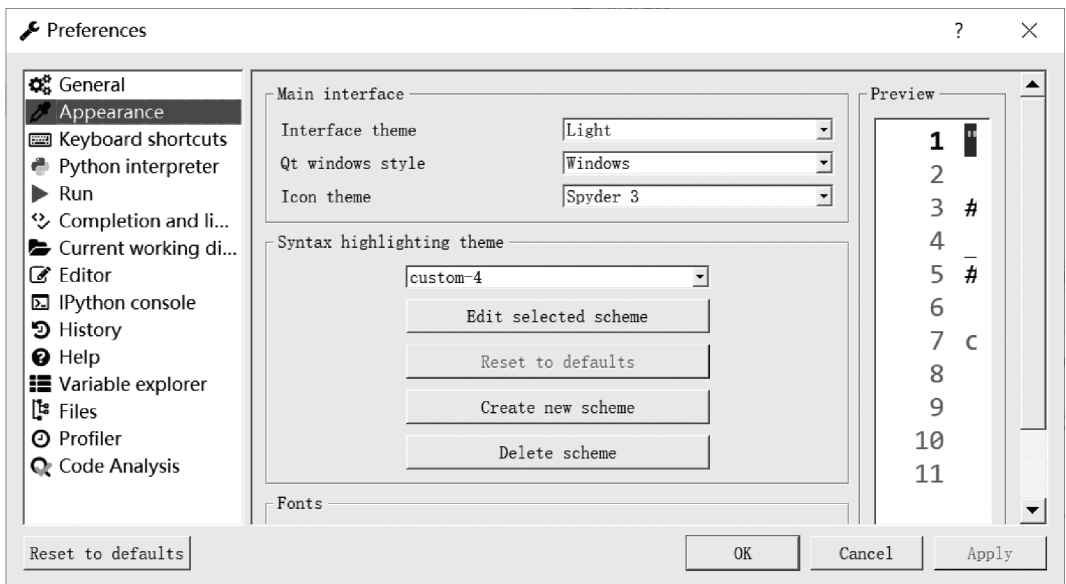


图 1-18 Preferences 对话框

(1) 以“#”开始的单行注释

[程序代码]

#单行注释练习

```
a = 1          #a 赋值为 1
b = 2          #b 赋值为 2
c = 3          #c 赋值为 3
print(a,b,c)   #输出 a,b,c 的值
```

[运行结果]

1 2 3

(2)以3对引号(单引号或双引号)开始、以3对引号结束的多行注释

[程序代码]

```
"""此处文字为注释内容
必要的注释会提高程序可读性
三对双引号示例"""
print("三对双引号注释")
```

[运行结果]

三对双引号注释

(3)语句缩进练习

Python 依靠语句块的缩进来体现语句之间的逻辑关系。书写代码时最好使用4个空格进行悬挂式缩进,并且同一级别语句块的缩进量必须相同。例如:

[程序代码]

```
#语句块缩进练习
m = 10           #m 赋值为 10
n = 20           #n 赋值为 20
if m > n:        #如果 m>n
    print(m)     #输出 m 的值
else:           #否则
    print(n)     #输出 n 的值
```

[运行结果]

20

(4)语句书写练习

1)一行一条语句

```
a = 1           #a 赋值为 1
b = 2           #b 赋值为 2
c = 3           #c 赋值为 3
print(a,b,c)   #输出 a,b,c 的值
```

2)可以一行多条语句

```
a = 1;b = 2;c = 3;print(a,b,c) #一行多条语句,用语句分隔符“;”分隔语句
```

3)可以一条语句多行

有时由于语句过长,一行放不下,可以在语句的外部加上一对圆括号,也可以使用续行符“\”(反斜杠)来实现分行书写功能。



[程序代码]

```
m=1+2+3+4+5+6+7\  
+8+9+10+11+12+\  
13+14+15+16  
print(m)
```

[程序代码]

```
m=(1+2+3+4+5+6+7  
+8+9+10+11+12+  
13+14+15+16)  
print(m)
```

[程序代码]

```
m=1+2+3+4+5+6+7+8+9+10+11+12+13+14+15+16  
print(m)
```

说明:以上3个程序的3种书写方法不同,其功能是相同的,运行结果都是136。