



高等学校教材

Python 程序设计

主编 杨柏林 韩培友

副主编 陈远高 韩建平 卢勇 王勋

高等教育出版社

内容提要

本书作为程序设计语言教材，系统地阐述了 Python 语言的工作原理、程序设计技术、实现方法及其应用。全书共 10 章，主要内容包括环境与操作、数据与表达、输入与输出、控制与结构、模块与函数、对象与方法、库表与管理、用户与界面、图形与图像以及案例与发布。

本书利用 Python 3.6.5 介绍程序设计方法及其实现技术，以 SQL Server 2016 和 MySQL 等为 DBMS 介绍数据库的设计与实现方法，并提供完整程序。

本书内容丰富、深入浅出、通俗易懂、结构严谨、注重实用，同时提供取材新颖、实用的例题和习题，便于读者巩固所学知识。

本书适合作为高等院校计算机科学与技术、数据科学与大数据技术、信息安全、软件工程、网络工程、信息管理与信息系统、电子商务、物流、金融与财经等相关本科专业的程序设计语言教材，也可作为从事程序设计与应用开发的工程技术人员的参考书。

图书在版编目 (CIP) 数据

Python 程序设计 / 杨柏林, 韩培友主编. --北京:
高等教育出版社, 2019.2
ISBN 978-7-04-030845-7

I. ①P… II. ①杨… ②韩… 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 (2019) 第 003575 号

策划编辑 武林晓
插图绘制 于博

责任编辑 武林晓
责任校对 胡美萍

封面设计 张楠
责任印制 陈伟光

版式设计 马云

出版发行 高等教育出版社
社址 北京市西城区德外大街 4 号
邮政编码 100120
印刷 北京明月印务有限责任公司
开本 787mm × 1092mm 1/16
印张 17.75
字数 420 千字
购书热线 010-58581118
咨询电话 400-810-0598

网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>
网上订购 <http://www.hepmall.com.cn>
<http://www.hepmall.com>
<http://www.hepmall.cn>
版 次 2019 年 2 月第 1 版
印 次 2019 年 2 月第 1 次印刷
定 价 33.00 元

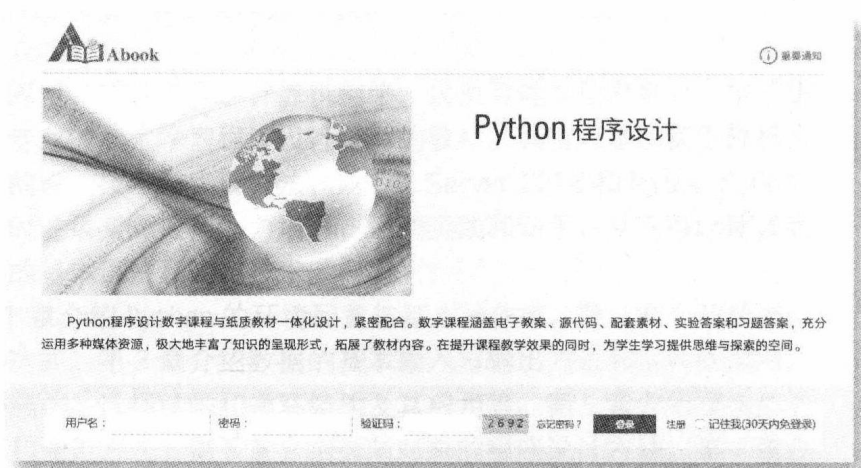
本书如有缺页、倒页、脱页等质量问题，请到所购图书销售部门联系调换
版权所有 侵权必究
物料号 30845-00

Python 程序设计

主 编
杨柏林 韩培友

副主编
陈远高 韩建平
卢 勇 王 勋

- 1 计算机访问<http://abook.hep.com.cn/1865451>，或手机扫描二维码、下载并安装 Abook 应用。
- 2 注册并登录，进入“我的课程”。
- 3 输入封底数字课程账号（20位密码，刮开涂层可见），或通过 Abook 应用扫描封底数字课程账号二维码，完成课程绑定。
- 4 单击“进入课程”按钮，开始本数字课程的学习。



课程绑定后一年为数字课程使用有效期。受硬件限制，部分内容无法在手机端显示，请按提示通过计算机访问学习。

如有使用问题，请发邮件至 abook@hep.com.cn。

电子教案

源代码

配套素材

实验答案

习题答案



扫描二维码
下载 Abook 应用

<http://abook.hep.com.cn/1865451>

前 言

Python 是一种简单易学、免费开源、功能强大的高级程序设计语言，使用 Python 高效的数据结构，可以简单有效地实现面向对象编程。Python 简洁的语法和对动态数据类型的支持，再加上解释性语言的本质，使得它在大多数平台上的许多领域都是一个理想的脚本语言，特别适用于快速的应用程序开发。同时具有高效、丰富和庞大的标准库和扩展库，而且已经成功广泛地应用到众多领域。

本书是面向高等学校计算机、软件工程、计算机网络、信息管理与信息系统、电子商务、物流金融与财经等相关专业的学生学习程序设计编写的教材。其特点是以实例分析为主，结合实际用例阐述程序的设计技术与实现方法，以 SQL Server 2016 和 MySQL 为 DBMS 介绍数据库的实现技术；同时注重实际应用能力的培养，强调实践和动手，从而可以使读者系统、全面地学习程序设计的基本语法和应用技术。

全书由 10 章组成，第 1 章介绍 Python 的环境配置与基本操作等。第 2 章介绍数据的基本类型、组合类型及其表达式。第 3 章介绍数据的基本输入与输出方法和文件的读写。第 4 章介绍程序设计的顺序结构、选择结构和循环结构及其应用等。第 5 章介绍模块的导入与引用方法、函数的定义和调用方法。第 6 章介绍面向对象的程序设计方法。第 7 章介绍数据库的创建、使用和管理。第 8 章介绍图形用户界面 GUI 的设计技术和实现方法。第 9 章介绍图形的绘制方法和图像的基本处理方法。第 10 章介绍应用案例的设计和实现方法及其发布。第 11 章通过 Python 与 Robot 的简易无缝连接，利用五个实用实验，设计和实现 Python 对 Robot 的灵活控制，从而熟练掌握 Python 的程序设计思想。

本书为新形态教材，配套资源丰富，包括电子教案、源代码、配套素材、实验答案和习题答案，可在高等教育出版社数字课程网站下载。

本书是作者多年从事程序设计和数据库技术教学和科研的经验总结。本书的编写人员主要包括浙江工商大学计算机与信息工程学院杨柏林教授、浙江工商大学计算机与信息工程学院韩培友副教授、浙江财经大学信息管理与工程学院陈远高教授、杭州电子科技大学计算机学院韩建平教授、中央民族大学信息工程学院卢勇和浙江工商大学计算机与信息工程学院王勋教授等。

全书由杨柏林教授负责整体结构设计、内容安排和全部审校工作，并最后审定。

鉴于作者水平有限，疏漏与不妥之处在所难免，敬请同行专家与广大读者不吝指正。

编 者

2018 年 12 月 杭州

郑重声明

高等教育出版社依法对本书享有专有出版权。任何未经许可的复制、销售行为均违反《中华人民共和国著作权法》，其行为人将承担相应的民事责任和行政责任；构成犯罪的，将被依法追究刑事责任。为了维护市场秩序，保护读者的合法权益，避免读者误用盗版书造成不良后果，我社将配合行政执法部门和司法机关对违法犯罪的单位和个人进行严厉打击。社会各界人士如发现上述侵权行为，希望及时举报，本社将奖励举报有功人员。

反盗版举报电话 (010) 58581999 58582371 58582488

反盗版举报传真 (010) 82086060

反盗版举报邮箱 dd@hep.com.cn

通信地址 北京市西城区德外大街4号

高等教育出版社法律事务与版权管理部

邮政编码 100120

防伪查询说明

用户购书后刮开封底防伪涂层，利用手机微信等软件扫描二维码，会跳转至防伪查询网页，获得所购图书详细信息。也可将防伪二维码下的20位密码按从左到右、从上到下的顺序发送短信至106695881280，免费查询所购图书真伪。

反盗版短信举报

编辑短信“JB，图书名称，出版社，购买地点”发送至10669588128

防伪客服电话

(010) 58582300

目 录

第 1 章 环境与操作 // 1

1.1 环境	1
1.1.1 Python 的安装	1
1.1.2 Python 数据库组件的安装	4
1.2 操作	7
1.2.1 执行方式	7
1.2.2 基本语法	9
1.3 环境与操作实验	13
实验 1.1 命令执行方式实验	14
实验 1.2 程序执行方式实验	14
习题	15

第 2 章 数据与表达 // 16

2.1 常量与变量	16
2.1.1 常量	16
2.1.2 变量	19
2.2 系统函数	21
2.2.1 数值函数	21
2.2.2 日期时间函数	24
2.2.3 字符串函数	25
2.2.4 其他函数	27
2.3 表达式	28
2.3.1 算术表达式	28
2.3.2 关系表达式	29
2.3.3 逻辑表达式	30
2.3.4 字符串表达式	31
2.4 列表	33
2.4.1 创建列表	33

2.4.2 编辑列表	34
------------	----

2.4.3 使用列表	36
------------	----

2.5 元组	38
--------	----

2.5.1 创建元组	38
------------	----

2.5.2 编辑元组	39
------------	----

2.5.3 使用元组	39
------------	----

2.6 集合	41
--------	----

2.6.1 创建集合	41
------------	----

2.6.2 编辑集合	42
------------	----

2.6.3 使用集合	44
------------	----

2.7 字典	46
--------	----

2.7.1 创建字典	46
------------	----

2.7.2 编辑字典	48
------------	----

2.7.3 使用字典	49
------------	----

2.8 数据与表达实验	52
-------------	----

实验 2.1 基本数据类型实验	52
-----------------	----

实验 2.2 表达式实验	53
--------------	----

实验 2.3 组合数据类型实验	53
-----------------	----

习题	55
----	----

第 3 章 输入与输出 // 56

3.1 数据输入	56
----------	----

3.2 数据输出	57
----------	----

3.2.1 默认格式输出	58
--------------	----

3.2.2 特定格式输出	58
--------------	----

3.2.3 format()函数输出	60
--------------------	----

3.2.4 format()方法输出	61
--------------------	----

3.3 文件读写	62
----------	----

3.3.1 写入文件	62
------------	----

3.3.2 读取文件	64
------------	----

3.3.3 文件管理	66
3.4 输入与输出实验	67
实验 3.1 基本输入与输出实验	67
实验 3.2 文件读写实验	68
习题	68

第 4 章 控制与结构 // 69

4.1 顺序结构	69
4.2 选择结构	70
4.2.1 单分支 if	70
4.2.2 二分支 if	71
4.2.3 多分支 if	74
4.2.4 if 嵌套	77
4.3 循环结构	79
4.3.1 while 循环	79
4.3.2 continue、break 和 pass	80
4.3.3 for 循环	81
4.3.4 zip、map 和 filter	83
4.3.5 选择循环混合嵌套	85
4.4 异常处理	89
4.4.1 捕捉异常	89
4.4.2 异常处理	93
4.5 控制与结构实验	97
实验 4.1 程序设计实验	97
实验 4.2 异常处理实验	99
习题	99

第 5 章 模块与函数 // 101

5.1 函数	101
5.1.1 函数定义与调用	101
5.1.2 参数传递	105
5.1.3 局部变量和全程变量	111
5.2 模块	116
5.2.1 创建模块	116
5.2.2 使用模块	118
5.3 模块与函数实验	120
实验 5.1 函数实验	120

实验 5.2 局部变量与全局变量	121
实验	121
实验 5.3 模块实验	124
习题	124

第 6 章 对象与方法 // 125

6.1 OOP 的特性	125
6.2 类和对象	127
6.2.1 类和对象的创建	128
6.2.2 类和对象的使用	130
6.3 事件与方法	132
6.3.1 事件和方法	132
6.3.2 构造方法和析构方法	135
6.4 客户管理范例	136
6.5 对象与方法实验	141
实验 6.1 对象与方法实验	141
实验 6.2 OOP 应用实验	141
习题	141

第 7 章 库表与管理 // 142

7.1 数据库概述	142
7.2 访问 SQLite 数据库	146
7.2.1 数据库的连接与创建	146
7.2.2 创建表	146
7.2.3 编辑表记录	147
7.2.4 查询记录	149
7.2.5 客户管理 SQLite 实现	150
7.3 访问 SQL Server 数据库	154
7.3.1 SQL Server 系统安装与配置	154
7.3.2 数据库的连接与创建	157
7.3.3 创建表	157
7.3.4 编辑表记录	159
7.3.5 查询记录	159
7.3.6 记录集操作	160
7.3.7 客户管理 SQL Server 实现	164

7.4 MySQL 数据库	169
7.5 库表与管理实验	172
实验 7.1 Python 访问 SQL Server	
实验	172
实验 7.2 CSV 访问 Excel 实验	175
实验 7.3 xlrd 访问 Excel 实验	175
习题	176

第 8 章 用户与界面 // 177

8.1 图形用户界面概述	177
8.1.1 GUI 设计	177
8.1.2 GUI 窗口 Tk	178
8.2 对话窗口	181
8.3 常用控件	187
8.3.1 标签 Label	187
8.3.2 消息 Message	190
8.3.3 标签框架 LabelFrame	191
8.3.4 框架 Frame	192
8.3.5 按钮 Button	193
8.3.6 单选按钮 Radiobutton	194
8.3.7 复选按钮 Checkbutton	195
8.3.8 单行输入框 Entry	197
8.3.9 多行文本框 Text	198
8.3.10 列表框 Listbox	199
8.3.11 组合框 Combobox	200
8.3.12 滚动条 Scrollbar	201
8.3.13 刻度条 Scale	203
8.3.14 微调框 Spinbox	204
8.3.15 画布 Canvas	205
8.3.16 面板 PanedWindow	206
8.3.17 选项菜单 OptionMenu	207
8.3.18 菜单系统 Menu	208
8.4 控件布局	210
8.5 事件处理	215
8.6 用户与界面实验	218
实验 8.1 基本控件实验	219
实验 8.2 事件处理实验	223

习题	226
----	-----

第 9 章 图形与图像 // 227

9.1 图形	227
9.1.1 画布 Canvas	227
9.1.2 线 create_line	228
9.1.3 矩形 create_rectangle	230
9.1.4 多边形 create_polygon	231
9.1.5 圆弧 create_arc	232
9.1.6 椭圆 create_oval	233
9.1.7 文本 create_text	234
9.1.8 画布显示图像	
create_image/bitmap	235
9.2 图像	237
9.2.1 图像概述	237
9.2.2 图像读写与显示	238
9.2.3 图像裁剪与复制	239
9.2.4 图像放缩与旋转	240
9.2.5 像素读写与图文水印	242
9.2.6 图像滤波与增强	244
9.3 图形与图像实验	246
实验 9.1 图形实验	246
实验 9.2 图像实验	247
习题	248

第 10 章 案例与发布 // 249

10.1 应用程序设计	249
10.2 四则运算测试	250
10.3 简易计算器	253
10.4 客户管理	257
10.5 打包与发布	265
10.5.1 安装 PyInstaller	265
10.5.2 生成 exe 执行文件	265
10.6 案例与发布实验	266
实验 10.1 案例实验	266
实验 10.2 发布实验	267

习题	267	实验 11.2 Robot 功能测试	269
第 11 章 Python 与 Robot	// 268	11.3 Robot 动作设计	269
11.1 Python 与 Robot 实验环境	268	实验 11.3 Robot 简单动作设计	269
实验 11.1 Robot 实验环境	268	实验 11.4 Robot 综合动作设计	269
11.2 Robot 功能测试	269	实验 11.5 Robot 自主作品设计	270
		参考文献	// 271

1

第 1 章 环境与操作

Python 是一种简单易学、免费开源、功能强大的高级程序设计语言，使用 Python 高效的数据结构，可以简单有效地实现面向对象编程。Python 简洁的语法和对动态数据类型的支持，再加上解释性语言的本质，使得它在大多数平台上的许多领域都是一个理想的脚本语言，特别适用于快速的应用程序开发。同时具有高效、丰富和庞大的标准库和扩展库，而且已经成功广泛地应用到众多领域。

1.1 环境

首先按照地址 <https://www.python.org/downloads/>，下载 python-3.6.5.exe（32 位）和 python-3.6.5-amd64.exe（64 位）。

然后按照地址 <https://sourceforge.net/projects/pywin32/files/pywin32/>，下载数据库组件 pywin32-221.win32-py3.6.exe（32 位）和 pywin32-221.Win - amd 64-py3.6.exe（64 位）。

1.1.1 Python 的安装

在 Windows 10 下，安装 Python 3.6.5 的方法如下。

(1) 运行安装文件，如图 1.1 所示。

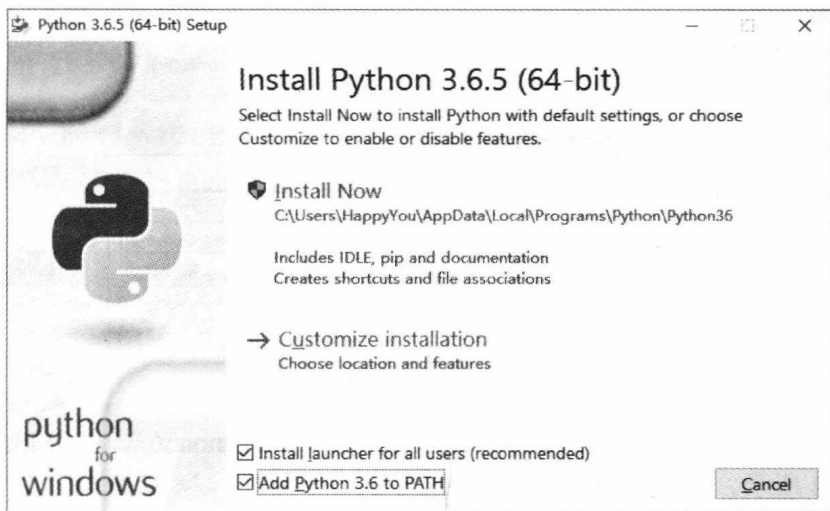


图 1.1 Python 3.6.5（64-bit）Setup 窗口

(2) 勾选 Install launcher for all users(recommended)和 Add Python 3.6 to PATH 复选框, 单击 Customize installation 选项, 打开界面如图 1.2 所示。

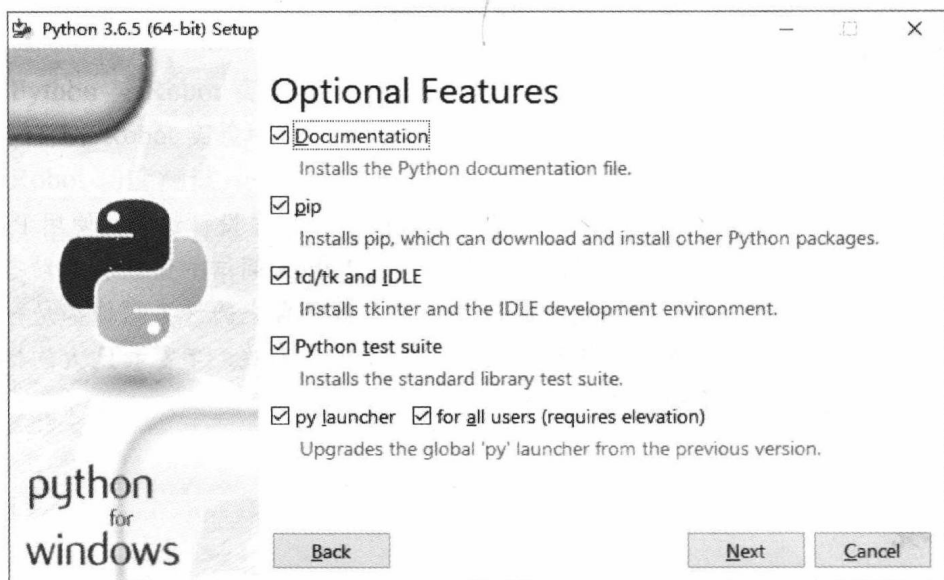


图 1.2 Option Features 界面

(3) 勾选所有复选框, 单击 Next 按钮, 打开界面如图 1.3 所示。

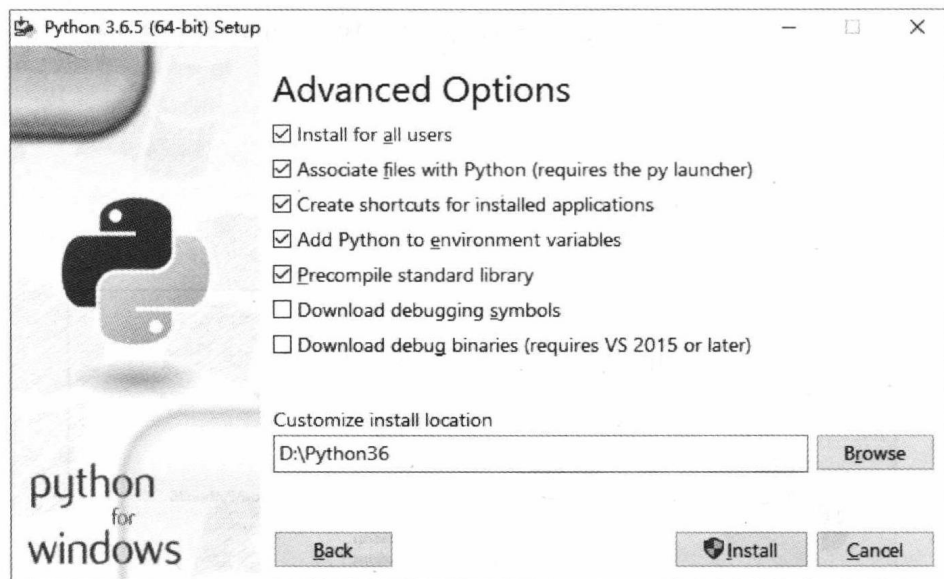


图 1.3 Advanced Options 界面

(4) 勾选前 5 个复选框, 选择或输入安装路径 “D: \Python36”, 单击 Install 按钮, 打开界面如图 1.4 所示。

(5) 安装完成, 如图 1.5 所示。单击 Close 按钮完成安装。

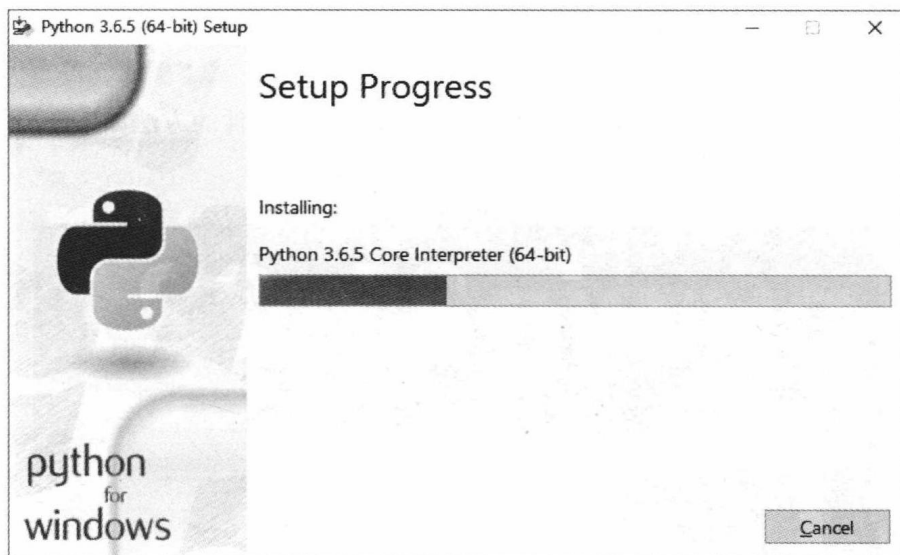


图 1.4 Setup Progress 界面



图 1.5 Setup was successful 界面

(6) 启动 Python 3.6 IDLE。选择“开始”→“所有程序”→ Python 3.6 → IDLE (Python 3.6 64-bit) 命令，打开 Python 3.6.5 Shell 窗口如图 1.6 所示。

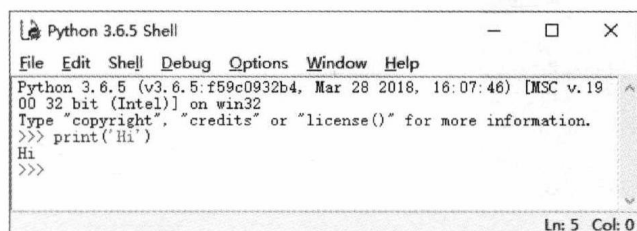


图 1.6 Python 3.6.5 Shell 窗口

提示：Python 3.x 与 Python 2.x 在用法上存在一定的差别，且不兼容。本书使用前者。

1.1.2 Python 数据库组件的安装

在 Windows 10 和 Python 3.6.5 下，安装 Python 数据库组件的方法如下。

(1) 运行安装文件，如图 1.7 所示。

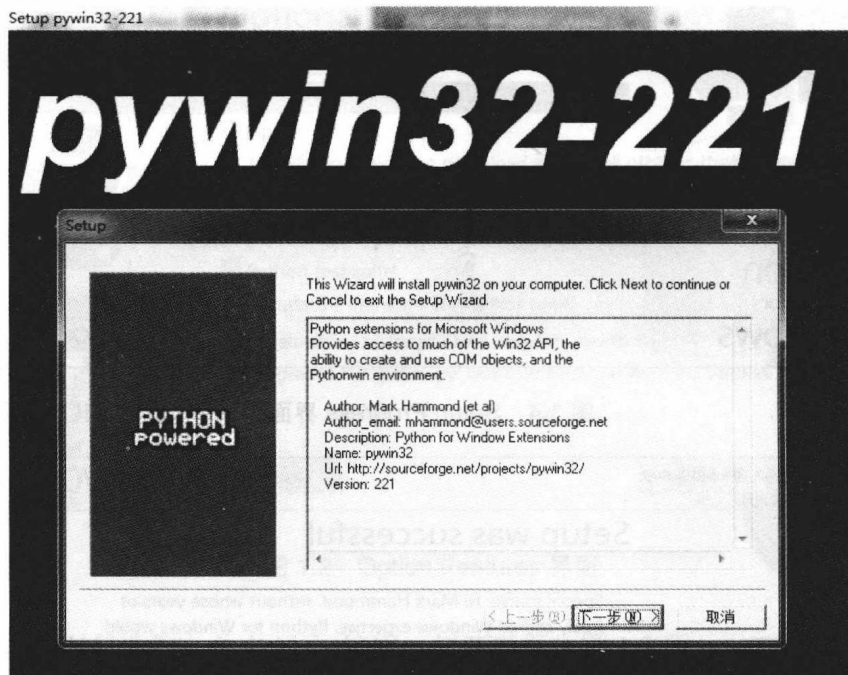


图 1.7 Setup 界面

(2) 单击“下一步”按钮，打开界面如图 1.8 所示。

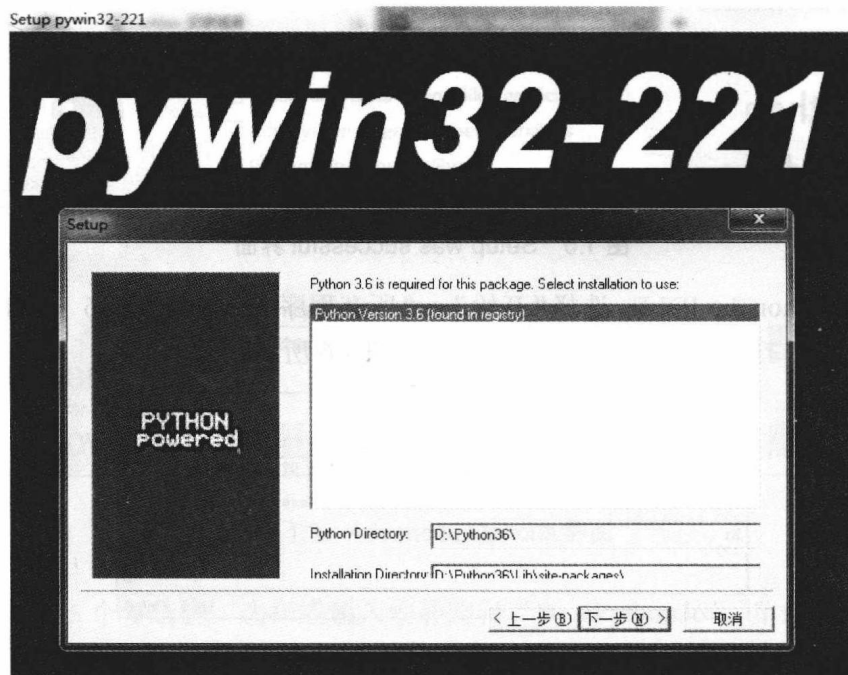


图 1.8 Set Path 界面

(3) 选择或输入安装路径“D:\Python36”，单击“下一步”按钮，打开界面如图 1.9 所示。

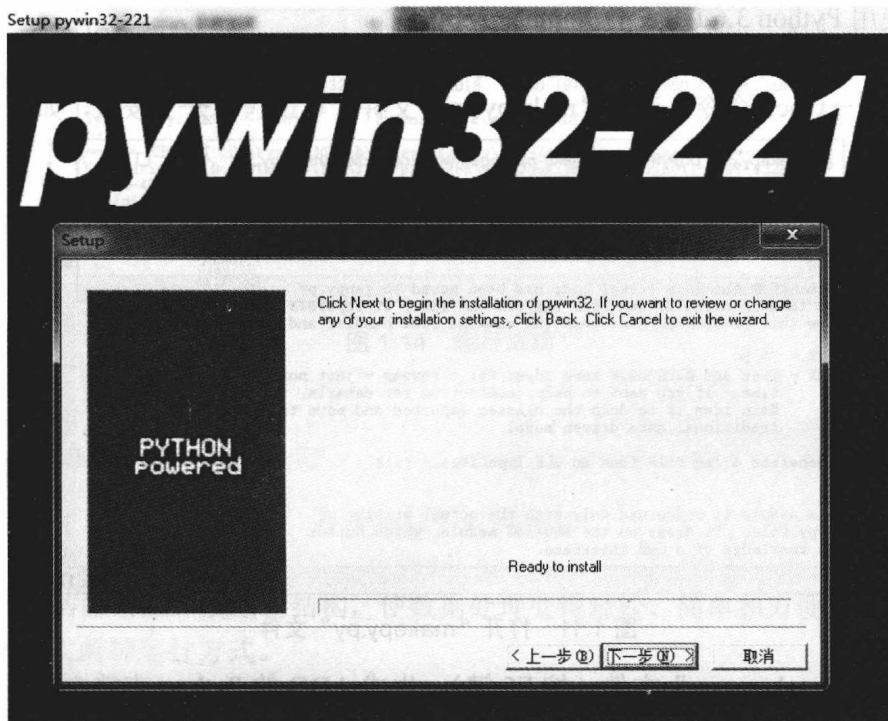


图 1.9 Ready to install 界面

(4) 单击“下一步”按钮，打开界面如图 1.10 所示。

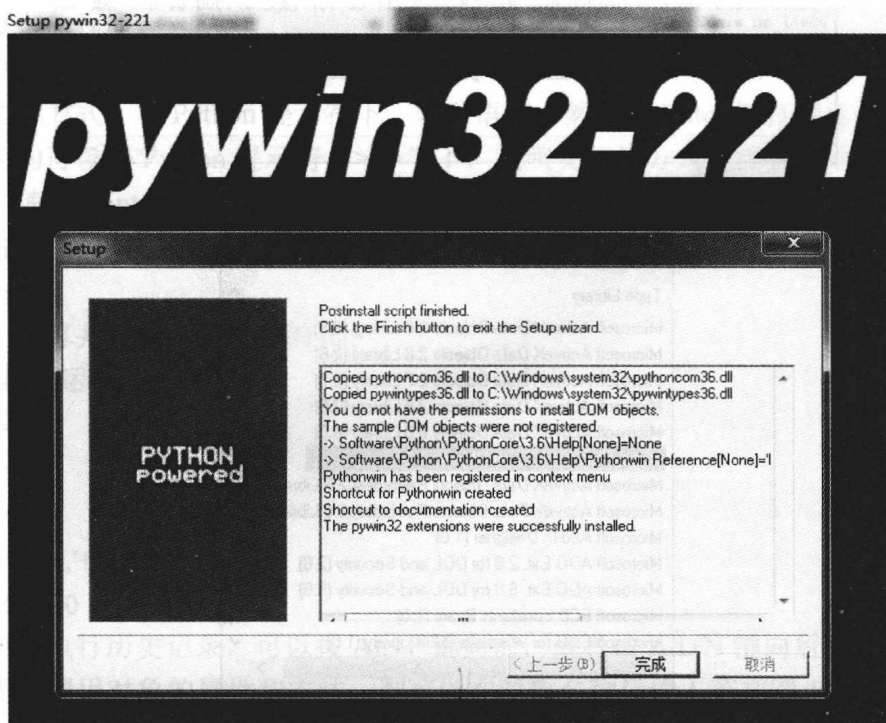


图 1.10 Setup Successful 界面

- (5) 单击“完成”按钮，完成安装。
- (6) 启动 Python 3.6 IDLE，如图 1.6 所示。
- (7) 使用 Python 3.6 IDLE 打开 makepy.py。

文件路径：D:\Python36\Lib\site-packages\win32com\client。

选择 File→Open 命令，选择“makepy.py”文件，单击“打开”按钮，如图 1.11 所示。

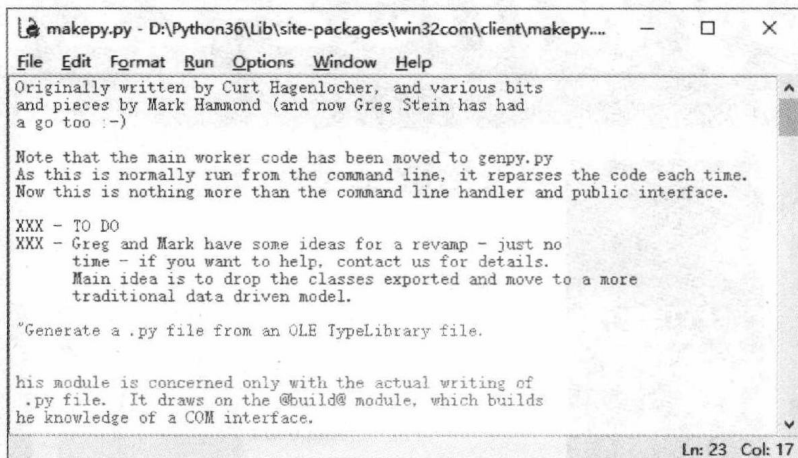


图 1.11 打开“makepy.py”文件

- (8) 运行“makepy.py”文件（按 F5 键），生成 ADO 的 Python 支持文件。
- 选择 Run→Run Module 命令，如图 1.12 和图 1.13 所示。

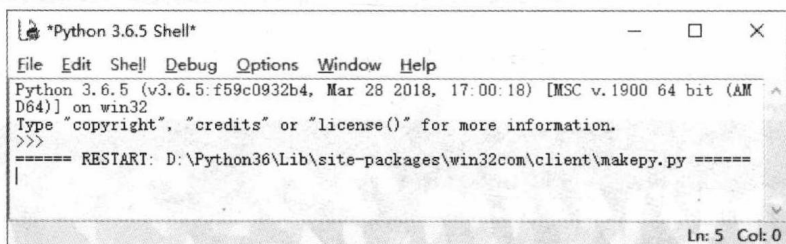


图 1.12 运行“makepy.py”文件



图 1.13 Select Library 对话框

(9) 在图 1.13 中, 选择 Microsoft ActiveX Data Objects 6.1 Library(6.1)选项, 单击 OK 按钮, 运行成功, 如图 1.14 所示。

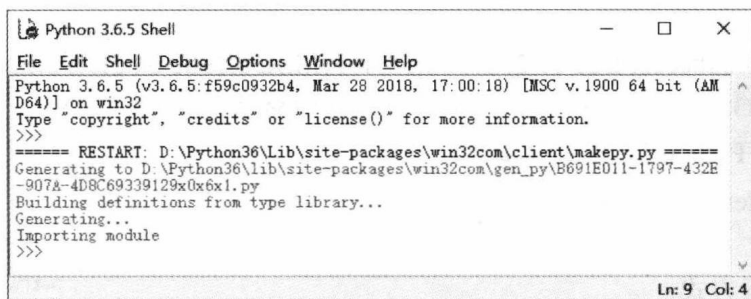


图 1.14 运行成功

1.2 操作

Python 简洁的语法和清晰的结构, 使数据处理变得轻松、简单和方便, 同时支持命令方式和程序方式两种运行方式。

1.2.1 执行方式

在 Python 环境下, 不但可以按照命令方式直接执行 Python 语句, 而且可以按照程序方式设计复杂的程序。

1. 命令执行方式

命令执行方式: 在 Python 提示符下, 输入语句, 然后按 Enter 键直接运行该语句, 即每一个语句均可以在 Python 提示符 “>>>” 下, 按照交互的方式直接独立地执行。

例如: 使用 print 函数输出 “Hi, Python!”。

```
>>>print('Hi, Python!')
```

```
Hi, Python!
```

再如: 计算并输出两个整数的和。

```
>>> #加法运算
```

```
>>> x=10
```

```
>>> y=20
```

```
>>> z=x+y
```

```
>>> print(x,'+',y, '=',z)
```

```
10 + 20 = 30
```

如果需要执行历史记录, 可以按 Alt+P 键向前, 或者按 Alt+N 键向后查找。

如果需要调用对象的属性和方法, 则可以在对象名称后输入英文圆点 “.”, 然后在弹出的列表中选择相应的属性或者方法, 并按 Enter 键。该方法也可用于程序的编辑。

例如：查看当前所用 Python 版本，可以使用以下命令。

```
>>> import sys    #导入系统模块 sys
>>> sys.version
'3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:54:40) [MSC v.1900 64 bit (AMD64)]'
```

提示：在 DOS 提示符（C: \>）下，可以使用命令：python -V。

又如：输出 Python 3.x 的默认编码。

```
>>> sys.getdefaultencoding()
'utf-8'
```

2. 程序执行方式

显然，命令执行方式给简单的数据处理带来了方便，但是，对于复杂的数据处理，还需要设计相应的程序，并按照程序方式快速高效地实现预定的任务。

编辑保存程序：选择 File→New File 命令，或按 Ctrl+N 键，输入语句/编辑程序，选择 Save 命令，或按 Ctrl+S 键，输入文件名 (*.py)，单击“保存”按钮。

运行程序：选择 Run → Run Module 命令，或按 F5 键。

例 1.1 分析如下猜数游戏。随机产生一个两位整数，任意输入一个整数，如果错误，则提示大小，如果正确，游戏结束。

```
#Pro0101.py
'''
猜数游戏程序
游戏虽然好玩，切记不要上瘾
适度游戏益脑，沉迷游戏伤身
合理安排时间，注意自我保护
'''
import random    #导入 random 模块
i=int(input('请输入两位整数:'))
j=random.randint(10, 99)
k=1
while i!=j:
    if i<j:
        print('你输入的数字小了，继续加油！')
    else:
        print('你输入的数字大了，继续努力！')
    i=int(input('请输入两位整数:'))
    k=k+1
else:
    print('你输入的数字正确，你好聪明！')
    print('你一共猜了',k,'次！')
```