

巍巍交大 百年书香
www.jiaodapress.com.cn
bookinfo@sjtu.edu.cn



策划编辑 刘建
责任编辑 李健 陈祖英
封面设计 黄燕美

Python

程序设计实用教程

Python CHENGXU SHEJI
SHIYONG JIAOCHENG

Python 程序设计实用教程

主编 罗少甫 谢娜娜

Python

程序设计实用教程

Python CHENGXU SHEJI
SHIYONG JIAOCHENG

主编 罗少甫 谢娜娜

免费提供

精品教学资料包

服务热线: 400-615-1233
www.xinsijaocai.com



扫描二维码
关注上海交通大学出版社
官方微信

ISBN 978-7-313-22344-9



9 787313 223449 >

定价: 45.00元

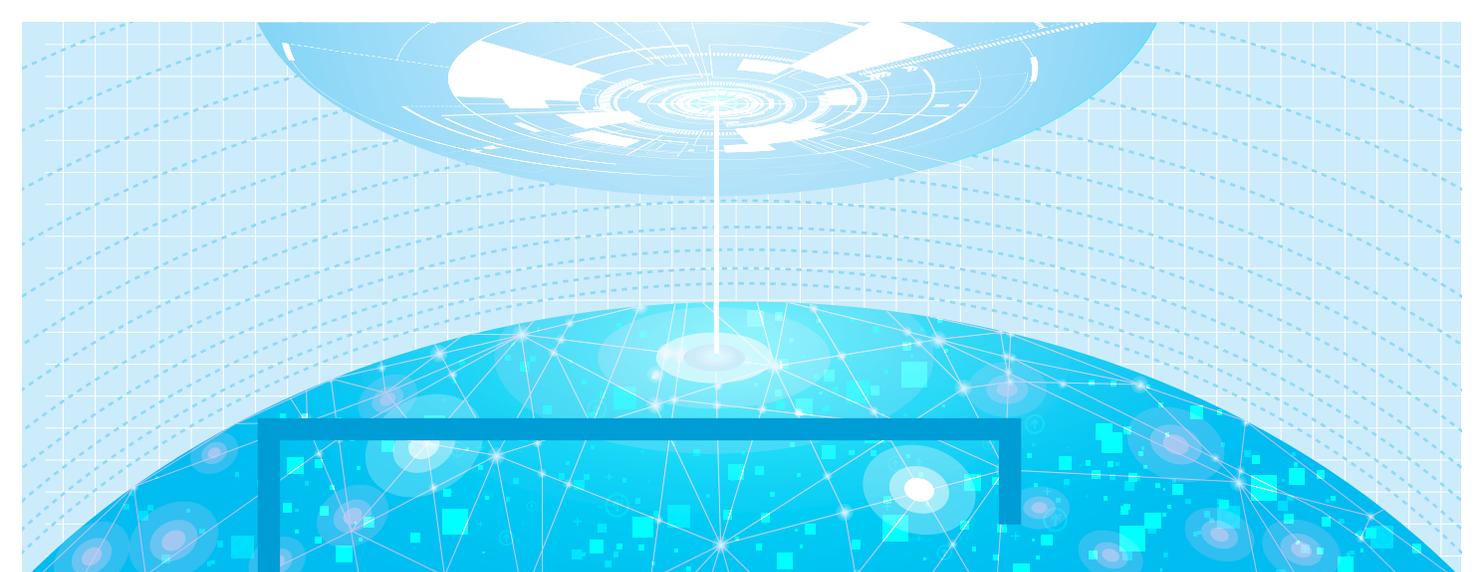


上海交通大学出版社

X-A



上海交通大学出版社
SHANGHAI JIAO TONG UNIVERSITY PRESS



Python

程序设计实用教程

Python CHENGXU SHEJI
SHIYONG JIAOCHENG

主 编 罗少甫 谢娜娜
副主编 董 明 赵 波



上海交通大学出版社
SHANGHAI JIAO TONG UNIVERSITY PRESS

内容提要

本书系统介绍了 Python 程序设计语言和程序设计的基本方法,按照由浅入深、由易到难、由理论到实践的原则进行编写。本书主要内容包括 Python 概述、PyCharm 开发环境、Python 输入与输出、Python 语言基础、字符串与正则表达式、Python 函数应用、Python 面向对象编程、Python 操作数据库、网络爬虫、pygame 游戏框架等。

本书可作为 Python 程序设计课程的教材,也可供相关技术人员参考。

图书在版编目(CIP)数据

Python 程序设计实用教程/罗少甫,谢娜娜主编. —
上海:上海交通大学出版社,2019(2026 重印)
ISBN 978-7-313-22344-9

I. ①P… II. ①罗… ②谢… III. ①软件工具—程序设计—高等职业教育—教材 IV. ①TP311.561

中国版本图书馆 CIP 数据核字(2019)第 255769 号

Python 程序设计实用教程

Python CHENGXU SHEJI SHIYONG JIAOCHENG

主 编:罗少甫 谢娜娜

出版发行:上海交通大学出版社

邮政编码:200030

印 制:大厂回族自治县聚鑫印刷有限责任公司

开 本:787 mm×1 092 mm 1/16

字 数:282 千字

版 次:2019 年 12 月第 1 版

书 号:ISBN 978-7-313-22344-9

定 价:45.00 元

地 址:上海市番禺路 951 号

电 话:021-64071208

经 销:全国新华书店

印 张:15

印 次:2026 年 4 月第 7 次印刷

版权所有 侵权必究

告读者:如您发现本书有印装质量问题请与印刷厂质量科联系

联系电话:0316-8836866

前言

PREFACE

党的二十大报告指出,教育、科技、人才是全面建设社会主义现代化国家的基础性、战略性支撑,要坚持教育优先发展、科技自立自强、人才引领驱动,加快建设教育强国、科技强国、人才强国,坚持为党育人、为国育才,全面提高人才自主培养质量。为了适应信息社会发展的需求,培养学生的计算思维能力,不少专业都开设了计算机语言程序设计课程。

Python 是一种结合了解释性、编译性、互动性和面向对象的脚本语言,拥有高效的数据结构和简单而有效的面向对象的程序设计方法。Python 优雅的语法和动态类型及解释执行特性,使它在大多数计算机平台和许多应用领域中成为脚本编写和快速应用开发的理想编程语言。

近年来,随着云计算、大数据、人工智能等技术的不断发展,Python 语言越来越受到重视,应用也越来越广泛。为了适应我国目前相关专业及课程教学改革的需要,我们特别组织具有丰富教学经验的一线教师编写了本书。

本书推荐学时安排见下表。

模 块	内 容	学 时
1	Python 概述	2
2	PyCharm 开发环境	2
3	Python 输入与输出	4
4	Python 语言基础	8
5	字符串与正则表达式	6
6	Python 函数应用	8
7	Python 面向对象编程	4
8	Python 操作数据库	8
9	网络爬虫	8
10	pygame 游戏框架	8
	总计	58

本书由罗少甫、谢娜娜任主编,由董明、赵波任副主编。具体编写分工如下:模块 1、模块 3 和模块 7 由赵波编写,模块 2、模块 9 和模块 10 由罗少甫编写,模块 4 和模块 5 由董明编写,模块 6 和模块 8 由谢娜娜编写。全书由罗少甫统稿。本书在编写过程中得到了诸多同仁的大力支持和帮助,尤其得到了我校计算机工程系黄诚老师和管理工程系杨光老师的鼎力相助,在此向他们表示衷心的感谢。

由于编者水平有限,书中存在的不足之处,恳请读者批评指正。

编 者

目录

CONTENTS

模块 1 Python 概述	1
1.1 初识 Python	1
1.1.1 Python 简介	1
1.1.2 Python 的版本	1
1.1.3 Python 的应用领域	2
1.2 搭建 Python 开发环境	3
1.2.1 安装 Python	3
1.2.2 执行 Python 脚本文件	4
1.3 使用 IDLE 编写第一个 Python 程序	5
1.4 实训	6
1.4.1 输出“Nothing in the world is difficult if put your heart into it”	6
1.4.2 输出“好好学习,天天向上”	7
模块 2 PyCharm 开发环境	8
2.1 PyCharm 的下载与安装	8
2.1.1 PyCharm 的下载	8
2.1.2 PyCharm 的安装	10
2.2 测试 PyCharm 开发环境	11
2.2.1 运行 PyCharm	11
2.2.2 创建工程目录	12
2.2.3 编写“好好学习,天天向上”程序	13
2.3 PyCharm 常用操作	15
2.3.1 在同一个工程下创建多个程序文件	15
2.3.2 运行程序的方法	15
2.3.3 保存程序	16
2.3.4 关闭和删除程序	16
2.3.5 打开最近的程序	17

2.3.6	断点调试	17
2.4	PyCharm 开发环境的组成	19
2.4.1	菜单栏	20
2.4.2	工具栏	21
2.4.3	工具窗口	21
2.4.4	调试设置区	22
2.4.5	代码编辑区	22
2.4.6	代码检查区	22
2.4.7	运行控制台	22
2.4.8	状态条	23
2.5	PyCharm 开发环境配置	23
2.5.1	作者信息设置	23
2.5.2	设置 Python 解析器	24
2.5.3	设置开发环境字体	25
2.6	实训	26
2.6.1	输出“Never say die”	26
2.6.2	输出小卖部标签	28
模块 3	Python 输入与输出	30
3.1	基本的输入与输出	30
3.1.1	使用 print() 函数	30
3.1.2	使用 input() 函数	31
3.2	注释	32
3.2.1	单行注释	32
3.2.2	多行注释	32
3.2.3	中文编码声明注释	32
3.3	代码缩进	33
3.4	规范	33
3.4.1	编码规范	33
3.4.2	命名规范	34
3.5	实训	34
3.5.1	输出唐诗《静夜思》	34
3.5.2	输出大写字母的 ASCII 状态值	35
模块 4	Python 语言基础	36
4.1	Python 数据类型	36
4.1.1	简单数据类型	36

4.1.2	变量与常量	37
4.2	字符串和编码	38
4.2.1	字符编码概述	38
4.2.2	Python 的字符串	39
4.3	Python 运算符及优先级	41
4.3.1	算术运算符	41
4.3.2	比较运算符	42
4.3.3	赋值运算符	43
4.3.4	逻辑运算符	44
4.3.5	成员运算符	45
4.3.6	身份运算符	46
4.3.7	运算符优先级	47
4.4	控制语句	47
4.4.1	条件控制语句	47
4.4.2	循环控制语句	49
4.4.3	pass 语句	53
4.5	合理使用 list、tuple、dict 和 set	53
4.5.1	list 列表的运用	53
4.5.2	tuple 元组的运用	55
4.5.3	dict 字典的运用	56
4.5.4	set 的运用	57
4.6	实训:统计一段文字中数字、字母、符号、空格的个数	59
模块 5	字符串与正则表达式	61
5.1	字符串常用操作	61
5.1.1	拼接字符串	61
5.1.2	计算字符串长度	62
5.1.3	截取字符串	63
5.1.4	分割字符串	63
5.1.5	检索字符串	64
5.1.6	大小写字母转换	64
5.1.7	字符串清洗	64
5.1.8	格式化字符串	65
5.1.9	替换字符串	65
5.2	正则表达式	66
5.2.1	正则表达式的定义方式	66
5.2.2	re 模块	67

5.2.3	切分字符串	70
5.2.4	贪婪匹配和非贪婪匹配	71
5.2.5	编译	71
5.3	实训	72
5.3.1	模拟砍价	72
5.3.2	判断是否买到假货	73
模块 6	Python 函数应用	74
6.1	函数的相关运用	74
6.1.1	调用函数	74
6.1.2	数据类型转换	76
6.1.3	定义函数	76
6.1.4	函数的参数	79
6.1.5	递归函数	84
6.2	常用的高级特征	85
6.2.1	切片	86
6.2.2	迭代	88
6.2.3	列表生成式	90
6.2.4	生成器	91
6.2.5	迭代器	93
6.3	高阶函数	94
6.3.1	map()函数	94
6.3.2	reduce()函数	95
6.3.3	filter()函数	96
6.3.4	sorted()函数	97
6.4	返回函数	98
6.4.1	函数作为返回值	99
6.4.2	闭包	100
6.5	匿名函数	100
6.6	装饰器	102
6.6.1	函数对象与函数名称	102
6.6.2	装饰器运用方式	103
6.6.3	处理装饰器的函数名称	105
6.7	偏函数	106
6.7.1	偏函数简介	106
6.7.2	偏函数的应用	107

6.8 使用模块	108
6.8.1 模块简介	109
6.8.2 import 语句	109
6.8.3 from-import 语句	110
6.8.4 Python 的常用内建模块	111
6.8.5 Python 中的包	116
6.8.6 作用域	118
6.9 实训:安装第三方模块	120
模块 7 Python 面向对象编程	123
7.1 面向对象的基础编程	123
7.1.1 类和实例	125
7.1.2 数据封装	126
7.1.3 访问权限	127
7.1.4 继承和多态	130
7.1.5 获取对象信息	132
7.1.6 实例属性与类属性	134
7.2 面向对象的高级编程	135
7.2.1 使用_slots_	135
7.2.2 使用@property	137
7.2.3 多重继承	139
7.2.4 定制类	142
7.2.5 使用枚举类	142
7.2.6 使用元素	143
7.3 实训:实现交通工具分类	145
模块 8 Python 操作数据库	147
8.1 数据库编程接口	147
8.1.1 连接对象	148
8.1.2 游标对象	149
8.2 使用 SQLite	150
8.2.1 创建数据库文件	150
8.2.2 操作 SQLite	151
8.3 使用 MySQL	153
8.3.1 下载并安装 MySQL	153
8.3.2 安装 PyMySQL	157
8.3.3 连接数据库	158

8.3.4	创建数据库表	159
8.3.5	操作 MySQL 数据库	160
8.4	实训:记录用户访问日志	166
模块 9	网络爬虫	171
9.1	初识网络爬虫	171
9.1.1	网络爬虫介绍	171
9.1.2	网络爬虫的基本工作原理	172
9.2	网络爬虫的常用技术	173
9.2.1	Requests 类库介绍	173
9.2.2	网页爬取的通用代码框架	177
9.2.3	HTTP 与 Requests 类库的方法	178
9.2.4	Requests 类库的 request 方法解析	180
9.2.5	认识 Beautiful Soup 类库	181
9.2.6	Beautiful Soup 搜索文档树	187
9.2.7	解决爬取数据过程中的常见问题	189
9.3	实训:网络小说下载	192
模块 10	pygame 游戏框架	199
10.1	初识 pygame	199
10.1.1	安装 pygame	199
10.1.2	pygame 的常用模块	200
10.2	pygame 的基本应用	202
10.3	实训:开发飞机大战游戏	209
参考文献	230

Python 概述



学习前请思考

- (1)什么是 Python?
- (2)Python 可以用来做什么?
- (3)应选择 Python 的哪个版本进行学习?
- (4)如何搭建简单的 Python 开发环境?
- (5)如何编写、运行一个简单的 Python 程序?

1.1 初识 Python

1.1.1 Python 简介

Python 是一种面向对象的解释型计算机程序设计语言,由荷兰人 Guido van Rossum 于 1989 年发明,第一个公开发行版发行于 1991 年。Python 是纯粹的自由软件,源代码和解释器 CPython 遵循 GPL(GNU General Public License)协议。Python 语法简洁清晰,特色之一是强制用空白符(white space)作为语句缩进。Python 具有丰富和强大的库。它常被称为“胶水语言”,能够把用其他语言(尤其是 C/C++)制作的各种模块很轻松地连接在一起。常见的一种应用情形是使用 Python 快速生成程序的原型(有时甚至是程序的最终界面),然后对其中有特别要求的部分,用更合适的语言改写,如 3D 游戏中的图形渲染模块,性能要求特别高,就可以用 C/C++重写,而后封装为 Python 可以调用的扩展类库。需要注意的是,在使用扩展类库时可能要考虑平台问题,某些可能不支持跨平台的实现。

1.1.2 Python 的版本

Python 有两个版本:一个是 2. x 版,另一个是 3. x 版。这两个版本是不兼容的。尽管

2014 年 Python 的创始人宣布将 Python 2.7 支持时间延长到 2020 年,并且使用 Python 2.x 的用户还是占很大一部分,但现在 Python 是朝着 3.x 版本进化的,在进化过程中,大量的针对 2.x 版本的代码要修改后才能运行,所以对于初学者来说,采用 3.5 或之后的版本比较合适。Python 语言解释器是一个轻量级的小尺寸软件(文件大小为 25~30 MB),可以在 Python 语言主网站上下载,下载网址为 <https://www.python.org/downloads/>。

注意

Python 提供了不同的操作系统版本,包括 Windows 操作系统版本、Linux 操作系统版本、Mac 操作系统版本及其他操作系统版本。本书以 Windows 操作系统版本为例进行介绍。

1.1.3 Python 的应用领域

Python 的应用领域主要有常规软件开发、科学计算、自动化运维、云计算、Web 开发、网络爬虫、数据分析和人工智能。

1. 常规软件开发

Python 支持函数式编程和面向对象编程(OOP),能够承担任何种类软件的开发工作,因此,常规的软件开发、脚本编写、网络编程等都属于其标配能力。

2. 科学计算

随着 NumPy、SciPy、Matplotlib 等众多程序库的开发,Python 越来越适合进行科学计算、绘制高质量的 2D 和 3D 图像。与科学计算领域最流行的商业软件 MATLAB 相比,Python 是一门通用的程序设计语言,比 MATLAB 所采用的脚本语言的应用范围更广泛,有更多程序库的支持。虽然 MATLAB 中的许多高级功能和 toolbox 目前还是无法替代的,但是在日常的科研开发之中仍然有很多的工作是可以用 Python 完成的。

3. 自动化运维

Python 支持自动化运维,是运维工程师首选的编程语言。Python 在自动化运维方面已经深入人心,如 SaltStack 和 Ansible 都是大名鼎鼎的自动化平台。

4. 云计算

开源云计算解决方案 OpenStack 就是基于 Python 开发的。

5. Web 开发

基于 Python 的 Web 开发框架有很多,如耳熟能详的 Django,还有 Tornado、Flask 等。其中,Python+Django 架构应用范围非常广,开发速度非常快,学习门槛也很低,能够帮助人们快速搭建可用的 Web 服务。

6. 网络爬虫

网络爬虫也称网络蜘蛛,是大数据行业获取数据的核心工具。能够编写网络爬虫的编程语言有很多,但 Python 绝对是其中的主流之一,其 Scrapy 爬虫框架应用非常广泛。

7. 数据分析

在大量数据的基础上,结合科学计算、机器学习等技术,对数据进行清洗、去重、规格化和针对性的分析是大数据行业的基石。Python 是数据分析的主流语言之一。

8. 人工智能

Python 在人工智能大范畴领域内的机器学习、神经网络、深度学习等方面都是主流的编程语言,得到广泛的支持和应用。

1.2 搭建 Python 开发环境

1.2.1 安装 Python

这里以 3.6.5 版本为例进行介绍,具体安装过程如下:

(1) 双击所下载的 Python 安装程序,弹出如图 1-1 所示的界面,选中“Add Python 3.6 to PATH”复选框。

(2) 在图 1-1 中有两个安装选项:一个是根据软件默认安装设置进行安装;另一个是根据自定义设置进行安装。这里单击“Customize installation”按钮自定义安装,随即进入“Optional Features”界面,如图 1-2 所示。



图 1-1 安装启动界面

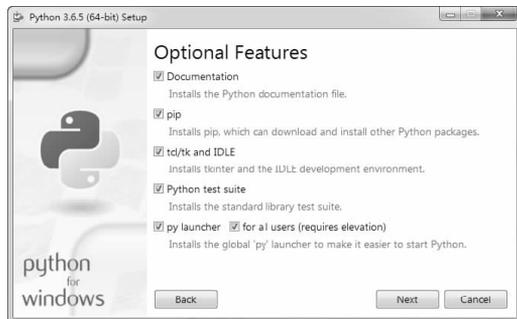


图 1-2 “Optional Features”界面

(3) 在“Optional Features”界面中选中要安装的选项对应的复选框,这里选中所有的复选框,然后单击 Next 按钮,弹出“Advanced Options”界面,如图 1-3 所示。

(4) 在“Advanced Options”界面中根据需要设置安装路径和相关选项,然后单击 Install 按钮,程序开始安装,如图 1-4 所示。

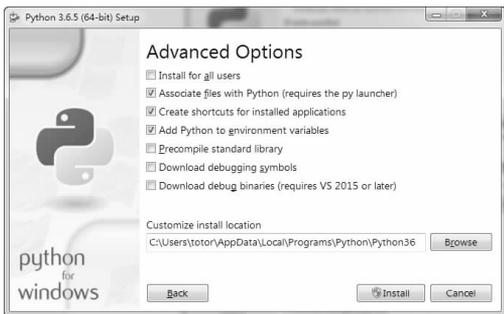


图 1-3 “Advanced Options”界面

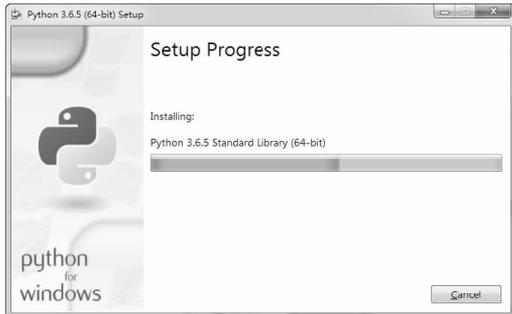


图 1-4 “Setup Progress”界面

(5) 安装完成,弹出“Setup was successful”界面(见图 1-5),单击 Close 按钮完成安装。



图 1-5 “Setup was successful”界面

Python 安装包将在系统中安装一批与 Python 开发和运行相关的程序,其中最重要的两个是 Python 命令行和 Python 集成开发环境 IDLE。

1.2.2 执行 Python 脚本文件

运行 Python 程序有两种方式:交互式和文件式。交互式指 Python 解释器即时响应用户输入的每条代码,给出输出结果。文件式也称批量式,指用户将 Python 程序写在一个或多个文件中,然后启动 Python 解释器批量执行文件中的代码。下面分别以这两种方式来编写并运行“Hello World”程序。

1. 交互式启动和运行方法

交互式有两种启动和运行方法。

(1)启动 Windows 操作系统命令行工具(<Windows 系统安装目录>\system32\cmd.exe),在控制台中输入“python”并按 Enter 键,在命令提示符>>>后输入如下程序代码:

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

按 Enter 键后显示输出结果“Hello World”,如图 1-6 所示。在>>>提示符后输入 exit()或者 quit()并按 Enter 键可以退出 Python 运行环境。

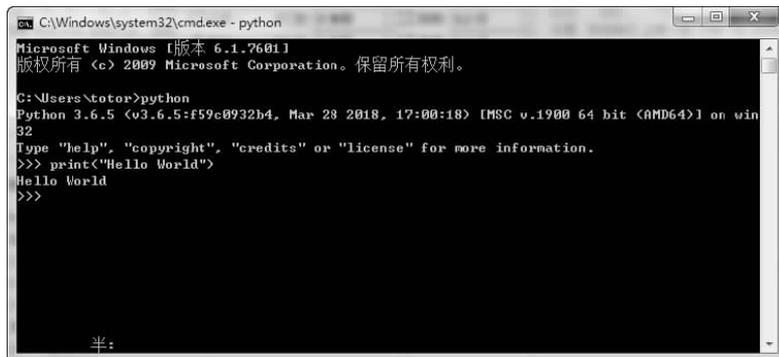


图 1-6 通过命令行启动交互式 Python 运行环境

(2)通过调用安装的 IDLE 来启动 Python 运行环境。IDLE 是 Python 软件包自带的集成开发环境,可以在 Windows 操作系统的“开始”菜单中搜索“IDLE”,找到 IDLE 的快捷方式。图 1-7 展示了 IDLE 环境中运行“Hello World”程序的效果。

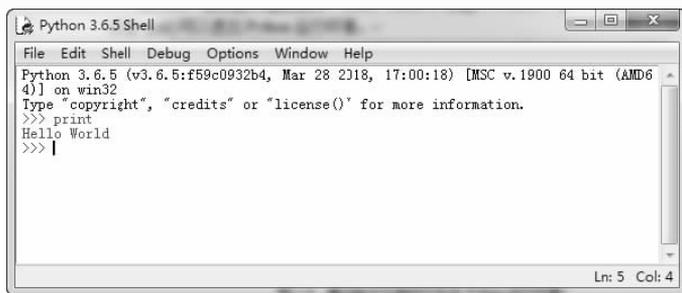


图 1-7 通过 IDLE 启动交互式 Python 运行环境

2. 文件式启动和运行方法

文件式也有两种运行方法,与交互式相对应。

(1)按照 Python 的语法格式编写代码,并保存为 .py 形式的文件(以“Hello World”程序为例,将代码保存成文件 hell.py)。Python 代码可以在任意编辑器中编写,对于百行以内规模的代码,建议使用 Python 安装包中的 IDLE 编辑器或者第三方开源记事本增强工具 Notepad++。然后,打开 Windows 的命令行(cmd.exe),进入 hello.py 文件所在目录,运行 Python 程序文件获得输出。

(2)打开 IDLE,按 Ctrl+N 组合键打开一个新窗口,或执行 File→New File 命令。这个新窗口不是交互模式,它是一个具备 Python 语法高亮辅助的编辑器,可以进行代码编辑。在其中输入 Python 代码。例如,输入“Hello World”程序并保存为 hello.py 文件。按快捷键 F5 或执行 Run→Run Module 命令,运行该文件。

3. 启动和运行方法推荐

交互式和文件式共有四种 Python 程序运行方法,其中,最常用且最重要的是采用 IDLE 的文件式方法。

IDLE 是一个简单有效的集成开发环境,无论是交互式还是文件式,都有助于快速编写和调试代码,它是小规模 Python 软件项目的主要编写工具。本书所有程序都可以通过 IDLE 编写并运行。行文方面,对于单行代码或通过观察输出结果讲解少量代码的情况,本书采用 IDLE 交互式(由 >>> 开头)进行描述;对于讲解整段的情况,采用 IDLE 文件式。

1.3 使用 IDLE 编写第一个 Python 程序

搭建 Python 开发环境之后,我们使用 IDLE 编写第一个 Python 程序。首先,写一个简单的“hello world”。

(1)新建一个空白的名为 helloworld 的 txt 文件,再将文件的后缀改为 .py,改了后缀之后的文件图标会变为 Python 程序的图标,如图 1-8 所示。



图 1-8 改后缀前后图标对比

(2) 在图标上右击,在弹出的快捷菜单中选择“Edit with IDLE”→“Edit with IDLE 3.6 (32-bit)”选项,使用 Python 自带的编辑器 IDLE 编写代码,如图 1-9 所示。

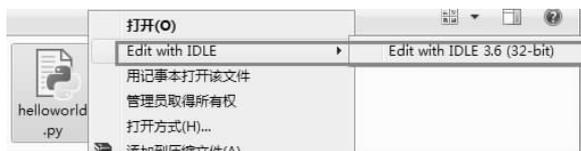


图 1-9 选择“Edit with IDLE”→“Edit with IDLE 3.6(32-bit)”选项

(3) 在编辑窗口中输入代码:print('hello world!'),如图 1-10 所示。

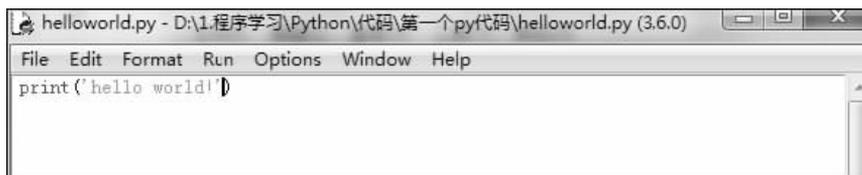


图 1-10 输入代码

(4) 执行 Run→Run Module 命令或按 F5 键运行代码,如图 1-11 所示。



图 1-11 运行代码

(5) 得到如图 1-12 所示的结果,第一个 helloworld 程序就完成了。

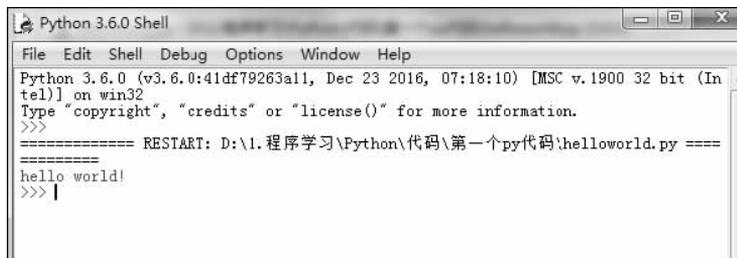


图 1-12 运行结果

1.4 实训

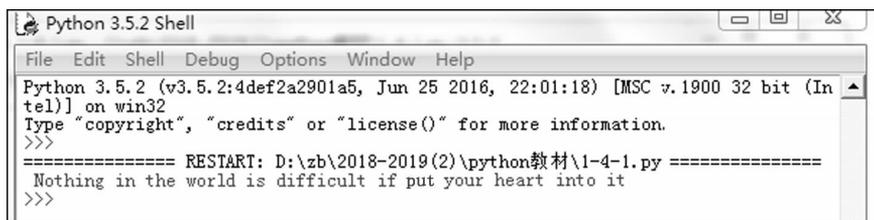
1.4.1 输出“Nothing in the world is difficult if put your heart into it”

【实训描述】

本实训要求在屏幕上输出“Nothing in the world is difficult if put your heart into it”。

【实训过程】

- (1)新建一个空白的名为“1-4-1”的 txt 文件,再将文件的后缀改为.py。
 - (2)右击文件图标,选择使用 Python 自带的编辑器 IDLE 来编写代码。
 - (3)在编辑窗口中输入代码:`print('Nothing in the world is difficult if put your heart into it')`。
 - (4)执行 Run→Run Module 命令或按 F5 键运行代码。
- 程序运行结果如图 1-13 所示。



```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\zb\2018-2019(2)\python教材\1-4-1.py =====
Nothing in the world is difficult if put your heart into it
>>>
```

图 1-13 程序运行结果

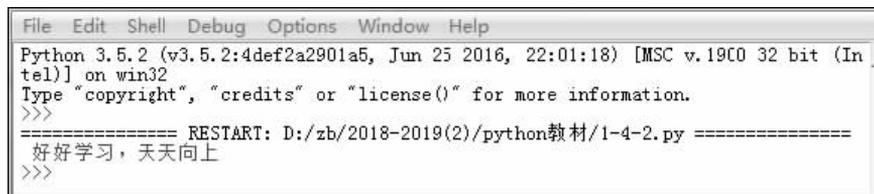
1.4.2 输出“好好学习,天天向上”

【实训描述】

本实训要求在屏幕上输出“好好学习,天天向上”。

【实训过程】

- (1)新建一个空白的名为“1-4-2”的 txt 文件,再将文件名的后缀改为.py。
 - (2)右击文件图标,选择使用 Python 自带的编辑器 IDLE 来编写代码。
 - (3)在编辑窗口中输入代码:`print('好好学习,天天向上')`。
 - (4)执行 Run→Run Module 命令或按 F5 键运行代码。
- 程序运行结果如图 1-14 所示。



```
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:/zb/2018-2019(2)/python教材/1-4-2.py =====
好好学习,天天向上
>>>
```

图 1-14 输出“好好学习,天天向上”运行结果