

TURING 图灵程序设计丛书



Python Crash Course

A Hands-On, Project-Based Introduction to Programming

Python编程 从入门到实践

【美】Eric Matthes 著 袁国忠 译



 中国工信出版集团

 人民邮电出版社
POSTS & TELECOM PRESS

版权信息

书名: Python编程: 从入门到实践

作者: [美] Eric Matthes

译者: 袁国忠

ISBN: 978-7-115-42802-8

本书由北京图灵文化发展有限公司发行数字版。版权所有，侵权必究。

您购买的图灵电子书仅供您个人使用，未经授权，不得以任何方式复制和传播本书内容。

我们愿意相信读者具有这样的良知和觉悟，与我们共同保护知识产权。

如果购买者有侵权行为，我们可能对该用户实施包括但不限于关闭该帐号等维权措施，并可能追究法律责任。

图灵社区会员 田芳林 (tianfanglin@outlook.com) 专享 尊重版权

版权声明

前言

读者对象

本书内容

为何使用Python

致谢

第一部分 基础知识

第1章 起步

1.1 搭建编程环境

1.1.1 Python 2和Python 3

1.1.2 运行Python代码片段

1.1.3 Hello World程序

1.2 在不同操作系统中搭建Python编程环境

1.2.1 在Linux系统中搭建Python编程环境

1.2.2 在OS X系统中搭建Python编程环境

1.2.3 在Windows系统中搭建Python编程环境

1.3 解决安装问题

1.4 从终端运行Python程序

1.4.1 在Linux和OS X系统中从终端运行Python程序

1.4.2 在Windows系统中从终端运行Python程序

1.5 小结

第2章 变量和简单数据类型

2.1 运行hello_world.py时发生的情况

2.2 变量

2.2.1 变量的命名和使用

2.2.2 使用变量时避免命名错误

2.3 字符串

2.3.1 使用方法修改字符串的大小写

2.3.2 合并（拼接）字符串

2.3.3 使用制表符或换行符来添加空白

2.3.4 删除空白

2.3.5 使用字符串时避免语法错误

2.3.6 Python 2中的print语句

2.4 数字

2.4.1 整数

2.4.2 浮点数

2.4.3 使用函数str()避免类型错误

2.4.4 Python 2中的整数

2.5 注释

2.5.1 如何编写注释

2.5.2 该编写什么样的注释

2.6 Python之禅

2.7 小结

第3章 列表简介

3.1 列表是什么

3.1.1 访问列表元素

3.1.2 索引从0而不是1开始

3.1.3 使用列表中的各个值

3.2 修改、添加和删除元素

3.2.1 修改列表元素

3.2.2 在列表中添加元素

3.2.3 从列表中删除元素

3.3 组织列表

3.3.1 使用方法sort()对列表进行永久性排序

3.3.2 使用函数sorted()对列表进行临时排序

3.3.3 倒着打印列表

3.3.4 确定列表的长度

3.4 使用列表时避免索引错误

- 3.5 小结
- 第4章 操作列表
 - 4.1 遍历整个列表
 - 4.1.1 深入地研究循环
 - 4.1.2 在for循环中执行更多的操作
 - 4.1.3 在for循环结束后执行一些操作
 - 4.2 避免缩进错误
 - 4.2.1 忘记缩进
 - 4.2.2 忘记缩进额外的代码行
 - 4.2.3 不必要的缩进
 - 4.2.4 循环后不必要的缩进
 - 4.2.5 遗漏了冒号
 - 4.3 创建数值列表
 - 4.3.1 使用函数range()
 - 4.3.2 使用range()创建数字列表
 - 4.3.3 对数字列表执行简单的统计计算
 - 4.3.4 列表解析
 - 4.4 使用列表的一部分
 - 4.4.1 切片
 - 4.4.2 遍历切片
 - 4.4.3 复制列表
 - 4.5 元组
 - 4.5.1 定义元组
 - 4.5.2 遍历元组中的所有值
 - 4.5.3 修改元组变量
 - 4.6 设置代码格式
 - 4.6.1 格式设置指南
 - 4.6.2 缩进
 - 4.6.3 行长
 - 4.6.4 空行
 - 4.6.5 其他格式设置指南
 - 4.7 小结
- 第5章 if语句
 - 5.1 一个简单示例
 - 5.2 条件测试
 - 5.2.1 检查是否相等
 - 5.2.2 检查是否相等时不考虑大小写
 - 5.2.3 检查是否不相等
 - 5.2.4 比较数字
 - 5.2.5 检查多个条件
 - 5.2.6 检查特定值是否包含在列表中
 - 5.2.7 检查特定值是否不包含在列表中
 - 5.2.8 布尔表达式
 - 5.3 if语句
 - 5.3.1 简单的if语句
 - 5.3.2 if-else语句
 - 5.3.3 if-elif-else结构
 - 5.3.4 使用多个elif代码块
 - 5.3.5 省略else代码块
 - 5.3.6 测试多个条件
 - 5.4 使用if语句处理列表
 - 5.4.1 检查特殊元素
 - 5.4.2 确定列表不是空的
 - 5.4.3 使用多个列表
 - 5.5 设置if语句的格式
 - 5.6 小结
- 第6章 字典
 - 6.1 一个简单的字典
 - 6.2 使用字典
 - 6.2.1 访问字典中的值
 - 6.2.2 添加键—值对
 - 6.2.3 先创建一个空字典
 - 6.2.4 修改字典中的值
 - 6.2.5 删除键—值对
 - 6.2.6 由类似对象组成的字典
 - 6.3 遍历字典
 - 6.3.1 遍历所有的键—值对
 - 6.3.2 遍历字典中的所有键
 - 6.3.3 按顺序遍历字典中的所有键
 - 6.3.4 遍历字典中的所有值
 - 6.4 嵌套
 - 6.4.1 字典列表
 - 6.4.2 在字典中存储列表
 - 6.4.3 在字典中存储字典
 - 6.5 小结
- 第7章 用户输入和while循环

- 7.1 函数input()的工作原理
 - 7.1.1 编写清晰的程序
 - 7.1.2 使用int()来获取数值输入
 - 7.1.3 求模运算符
 - 7.1.4 在Python 2.7中获取输入
- 7.2 while循环简介
 - 7.2.1 使用while循环
 - 7.2.2 让用户选择何时退出
 - 7.2.3 使用标志
 - 7.2.4 使用break退出循环
 - 7.2.5 在循环中使用continue
 - 7.2.6 避免无限循环
- 7.3 使用while循环来处理列表和字典
 - 7.3.1 在列表之间移动元素
 - 7.3.2 删除包含特定值的所有列表元素
 - 7.3.3 使用用户输入来填充字典
- 7.4 小结
- 第8章 函数
 - 8.1 定义函数
 - 8.1.1 向函数传递信息
 - 8.1.2 实参和形参
 - 8.2 传递实参
 - 8.2.1 位置实参
 - 8.2.2 关键字实参
 - 8.2.3 默认值
 - 8.2.4 等效的函数调用
 - 8.2.5 避免实参错误
 - 8.3 返回值
 - 8.3.1 返回简单值
 - 8.3.2 让实参变成可选的
 - 8.3.3 返回字典
 - 8.3.4 结合使用函数和while循环
 - 8.4 传递列表
 - 8.4.1 在函数中修改列表
 - 8.4.2 禁止函数修改列表
 - 8.5 传递任意数量的实参
 - 8.5.1 结合使用位置实参和任意数量实参
 - 8.5.2 使用任意数量的关键字实参
 - 8.6 将函数存储在模块中
 - 8.6.1 导入整个模块
 - 8.6.2 导入特定的函数
 - 8.6.3 使用as给函数指定别名
 - 8.6.4 使用as给模块指定别名
 - 8.6.5 导入模块中的所有函数
 - 8.7 函数编写指南
 - 8.8 小结
- 第9章 类
 - 9.1 创建和使用类
 - 9.1.1 创建Dog类
 - 9.1.2 根据类创建实例
 - 9.2 使用类和实例
 - 9.2.1 Car类
 - 9.2.2 给属性指定默认值
 - 9.2.3 修改属性的值
 - 9.3 继承
 - 9.3.1 子类的方法__init__()
 - 9.3.2 Python 2.7中的继承
 - 9.3.3 给子类定义属性和方法
 - 9.3.4 重写父类的方法
 - 9.3.5 将实例用作属性
 - 9.3.6 模拟实物
 - 9.4 导入类
 - 9.4.1 导入单个类
 - 9.4.2 在一个模块中存储多个类
 - 9.4.3 从一个模块中导入多个类
 - 9.4.4 导入整个模块
 - 9.4.5 导入模块中的所有类
 - 9.4.6 在一个模块中导入另一个模块
 - 9.4.7 自定义工作流程
 - 9.5 Python标准库
 - 9.6 类编码风格
 - 9.7 小结
- 第10章 文件和异常
 - 10.1 从文件中读取数据
 - 10.1.1 读取整个文件
 - 10.1.2 文件路径

- 10.1.3 逐行读取
- 10.1.4 创建一个包含文件各行内容的列表
- 10.1.5 使用文件的内容
- 10.1.6 包含一百万位的大型文件
- 10.1.7 圆周率值中包含你的生日吗
- 10.2 写入文件
 - 10.2.1 写入空文件
 - 10.2.2 写入多行
 - 10.2.3 附加到文件
- 10.3 异常
 - 10.3.1 处理ZeroDivisionError异常
 - 10.3.2 使用try-except代码块
 - 10.3.3 使用异常避免崩溃
 - 10.3.4 else代码块
 - 10.3.5 处理FileNotFoundError异常
 - 10.3.6 分析文本
 - 10.3.7 使用多个文件
 - 10.3.8 失败时一声不吭
 - 10.3.9 决定报告哪些错误
- 10.4 存储数据
 - 10.4.1 使用json.dump()和json.load()
 - 10.4.2 保存和读取用户生成的数据
 - 10.4.3 重构
- 10.5 小结
- 第 11 章 测试代码
 - 11.1 测试函数
 - 11.1.1 单元测试和测试用例
 - 11.1.2 可通过的测试
 - 11.1.3 不能通过的测试
 - 11.1.4 测试未通过时怎么办
 - 11.1.5 添加新测试
 - 11.2 测试类
 - 11.2.1 各种断言方法
 - 11.2.2 一个要测试的类
 - 11.2.3 测试AnonymousSurvey类
 - 11.2.4 方法setUp()
 - 11.3 小结
- 第二部分 项目
- 项目1 外星人入侵
- 第 12 章 武装飞船
 - 12.1 规划项目
 - 12.2 安装Pygame
 - 12.2.1 使用pip安装Python包
 - 12.2.2 在Linux系统中安装Pygame
 - 12.2.3 在OS X系统中安装Pygame
 - 12.2.4 在Windows系统中安装Pygame
 - 12.3 开始游戏项目
 - 12.3.1 创建Pygame窗口以及响应用户输入
 - 12.3.2 设置背景色
 - 12.3.3 创建设置类
 - 12.4 添加飞船图像
 - 12.4.1 创建Ship类
 - 12.4.2 在屏幕上绘制飞船
 - 12.5 重构：模块game_functions
 - 12.5.1 函数check_events()
 - 12.5.2 函数update_screen()
 - 12.6 驾驶飞船
 - 12.6.1 响应按键
 - 12.6.2 允许不断移动
 - 12.6.3 左右移动
 - 12.6.4 调整飞船的速度
 - 12.6.5 限制飞船的活动范围
 - 12.6.6 重构check_events()
 - 12.7 简单回顾
 - 12.7.1 alien_invasion.py
 - 12.7.2 settings.py
 - 12.7.3 game_functions.py
 - 12.7.4 ship.py
 - 12.8 射击
 - 12.8.1 添加子弹设置
 - 12.8.2 创建Bullet类
 - 12.8.3 将子弹存储到编组中
 - 12.8.4 开火
 - 12.8.5 删除已消失的子弹
 - 12.8.6 限制子弹数量
 - 12.8.7 创建函数update_bullets()

- 12.8.8 创建函数fire_bullet()
- 12.9 小结
- 第 13 章 外星人
 - 13.1 回顾项目
 - 13.2 创建第一个外星人
 - 13.2.1 创建Alien类
 - 13.2.2 创建Alien实例
 - 13.2.3 让外星人出现在屏幕上
 - 13.3 创建一群外星人
 - 13.3.1 确定一行可容纳多少个外星人
 - 13.3.2 创建多行外星人
 - 13.3.3 创建外星人群
 - 13.3.4 重构create_fleet()
 - 13.3.5 添加行
 - 13.4 让外星人群移动
 - 13.4.1 向右移动外星人
 - 13.4.2 创建表示外星人移动方向的设置
 - 13.4.3 检查外星人是否撞到了屏幕边缘
 - 13.4.4 向下移动外星人群并改变移动方向
 - 13.5 射杀外星人
 - 13.5.1 检测子弹与外星人的碰撞
 - 13.5.2 为测试创建大子弹
 - 13.5.3 生成新的外星人群
 - 13.5.4 提高子弹的速度
 - 13.5.5 重构update_bullets()
 - 13.6 结束游戏
 - 13.6.1 检测外星人和飞船碰撞
 - 13.6.2 响应外星人和飞船碰撞
 - 13.6.3 有外星人到达屏幕底端
 - 13.6.4 游戏结束
 - 13.7 确定应运行游戏的哪些部分
 - 13.8 小结
- 记分
 - 14.1 添加Play按钮
 - 14.1.1 创建Button类
 - 14.1.2 在屏幕上绘制按钮
 - 14.1.3 开始游戏
 - 14.1.4 重置游戏
 - 14.1.5 将Play按钮切换到非活动状态
 - 14.1.6 隐藏光标
 - 14.2 提高等级
 - 14.2.1 修改速度设置
 - 14.2.2 重置速度
 - 14.3 记分
 - 14.3.1 显示得分
 - 14.3.2 创建记分牌
 - 14.3.3 在外星人被消灭时更新得分
 - 14.3.4 将消灭的每个外星人的点数都计入得分
 - 14.3.5 提高点数
 - 14.3.6 将得分圆整
 - 14.3.7 最高得分
 - 14.3.8 显示等级
 - 14.3.9 显示余下的飞船数
 - 14.4 小结
- 项目2 数据可视化
- 第 15 章 生成数据
 - 15.1 安装matplotlib
 - 15.1.1 在Linux系统中安装matplotlib
 - 15.1.2 在OS X系统中安装matplotlib
 - 15.1.3 在Windows系统中安装matplotlib
 - 15.1.4 测试matplotlib
 - 15.1.5 matplotlib画廊
 - 15.2 绘制简单的折线图
 - 15.2.1 修改标签文字和线条粗细
 - 15.2.2 校正图形
 - 15.2.3 使用scatter()绘制散点图并设置其样式
 - 15.2.4 使用scatter()绘制一系列点
 - 15.2.5 自动计算数据
 - 15.2.6 删除数据点的轮廓
 - 15.2.7 自定义颜色
 - 15.2.8 使用颜色映射
 - 15.2.9 自动保存图表
 - 15.3 随机漫步
 - 15.3.1 创建RandomWalk()类
 - 15.3.2 选择方向
 - 15.3.3 绘制随机漫步图

- 15.3.4 模拟多次随机漫步
- 15.3.5 设置随机漫步图的样式
- 15.3.6 给点着色
- 15.3.7 重新绘制起点和终点
- 15.3.8 隐藏坐标轴
- 15.3.9 增加点数
- 15.3.10 调整尺寸以适合屏幕
- 15.4 使用Pygal模拟掷骰子
 - 15.4.1 安装Pygal
 - 15.4.2 Pygal画廊
 - 15.4.3 创建Die类
 - 15.4.4 掷骰子
 - 15.4.5 分析结果
 - 15.4.6 绘制直方图
 - 15.4.7 同时掷两个骰子
 - 15.4.8 同时掷两个面数不同的骰子
- 15.5 小结
- 第 16 章 下载数据
 - 16.1 CSV文件格式
 - 16.1.1 分析CSV文件头
 - 16.1.2 打印文件头及其位置
 - 16.1.3 提取并读取数据
 - 16.1.4 绘制气温图表
 - 16.1.5 模块datetime
 - 16.1.6 在图表中添加日期
 - 16.1.7 涵盖更长的时间
 - 16.1.8 再绘制一个数据系列
 - 16.1.9 给图表区域着色
 - 16.1.10 错误检查
 - 16.2 制作世界人口地图: JSON格式
 - 16.2.1 下载世界人口数据
 - 16.2.2 提取相关的数据
 - 16.2.3 将字符串转换为数字值
 - 16.2.4 获取两个字母的国别码
 - 16.2.5 制作世界地图
 - 16.2.6 在世界地图上呈现数字数据
 - 16.2.7 绘制完整的世界人口地图
 - 16.2.8 根据人口数量将国家分组
 - 16.2.9 使用Pygal设置世界地图的样式
 - 16.2.10 加亮颜色主题
 - 16.3 小结
- 第 17 章 使用API
 - 17.1 使用Web API
 - 17.1.1 Git和GitHub
 - 17.1.2 使用API调用请求数据
 - 17.1.3 安装requests
 - 17.1.4 处理API响应
 - 17.1.5 处理响应字典
 - 17.1.6 概述最受欢迎的仓库
 - 17.1.7 监视API的速率限制
 - 17.2 使用Pygal可视化仓库
 - 17.2.1 改进Pygal图表
 - 17.2.2 添加自定义工具提示
 - 17.2.3 根据数据绘图
 - 17.2.4 在图表中添加可单击的链接
 - 17.3 Hacker News API
 - 17.4 小结
- 项目3 Web应用程序
- 第 18 章 Django入门
 - 18.1 建立项目
 - 18.1.1 制定规范
 - 18.1.2 建立虚拟环境
 - 18.1.3 安装virtualenv
 - 18.1.4 激活虚拟环境
 - 18.1.5 安装Django
 - 18.1.6 在Django中创建项目
 - 18.1.7 创建数据库
 - 18.1.8 查看项目
 - 18.2 创建应用程序
 - 18.2.1 定义模型
 - 18.2.2 激活模型
 - 18.2.3 Django管理网站
 - 18.2.4 定义模型Entry
 - 18.2.5 迁移模型Entry
 - 18.2.6 向管理网站注册Entry
 - 18.2.7 Django shell

- 18.3 创建网页：学习笔记主页
 - 18.3.1 映射URL
 - 18.3.2 编写视图
 - 18.3.3 编写模板
- 18.4 创建其他网页
 - 18.4.1 模板继承
 - 18.4.2 显示所有主题的页面
 - 18.4.3 显示特定主题的页面
- 18.5 小结
- 第 19 章 用户账户
 - 19.1 让用户能够输入数据
 - 19.1.1 添加新主题
 - 19.1.2 添加新条目
 - 19.1.3 编辑条目
 - 19.2 创建用户账户
 - 19.2.1 应用程序users
 - 19.2.2 登录页面
 - 19.2.3 注销
 - 19.2.4 注册页面
 - 19.3 让用户拥有自己的数据
 - 19.3.1 使用@login_required限制访问
 - 19.3.2 将数据关联到用户
 - 19.3.3 只允许用户访问自己的主题
 - 19.3.4 保护用户的主题
 - 19.3.5 保护页面edit_entry
 - 19.3.6 将新主题关联到当前用户
 - 19.4 小结
- 第 20 章 设置应用程序的样式并对其进行部署
 - 20.1 设置项目“学习笔记”的样式
 - 20.1.1 应用程序django-bootstrap3
 - 20.1.2 使用Bootstrap来设置项目“学习笔记”的样式
 - 20.1.3 修改base.html
 - 20.1.4 使用jumbotron设置主页的样式
 - 20.1.5 设置登录页面的样式
 - 20.1.6 设置new_topic页面的样式
 - 20.1.7 设置topics页面的样式
 - 20.1.8 设置topic页面中条目的样式
 - 20.2 部署“学习笔记”
 - 20.2.1 建立Heroku账户
 - 20.2.2 安装Heroku Toolbelt
 - 20.2.3 安装必要的包
 - 20.2.4 创建包含包列表的文件requirements.txt
 - 20.2.5 指定Python版本
 - 20.2.6 为部署到Heroku而修改settings.py
 - 20.2.7 创建启动进程的Procfile
 - 20.2.8 为部署到Heroku而修改wsgi.py
 - 20.2.9 创建用于存储静态文件的目录
 - 20.2.10 在本地使用gunicorn服务器
 - 20.2.11 使用Git跟踪项目文件
 - 20.2.12 推送到Heroku
 - 20.2.13 在Heroku上建立数据库
 - 20.2.14 改进Heroku部署
 - 20.2.15 确保项目的安全
 - 20.2.16 提交并推送修改
 - 20.2.17 创建自定义错误页面
 - 20.2.18 继续开发
 - 20.2.19 设置SECRET_KEY
 - 20.2.20 将项目从Heroku删除
 - 20.3 小结
- 附录 A 安装Python
 - A.1 在Linux系统中安装Python
 - A.1.1 确定已安装的版本
 - A.1.2 在Linux系统中安装Python 3
 - A.2 在OS X系统中安装Python
 - A.2.1 确定已安装的版本
 - A.2.2 使用Homebrew来安装Python 3
 - A.3 在Windows系统中安装Python
 - A.3.1 在Windows系统中安装Python 3
 - A.3.2 查找Python解释器
 - A.3.3 将Python添加到环境变量Path中
 - A.4 Python关键字和内置函数
 - A.4.1 Python关键字
 - A.4.2 Python内置函数
- 附录 B 文本编辑器
 - B.1 Geany
 - B.1.1 在Linux系统中安装Geany

- B.1.2 在Windows系统中安装Geany
 - B.1.3 在Geany中运行Python程序
 - B.1.4 定制Geany的设置
 - B.2 Sublime Text
 - B.2.1 在OS X系统中安装Sublime Text
 - B.2.2 在Linux系统中安装Sublime Text
 - B.2.3 在Windows系统中安装Sublime Text
 - B.2.4 在Sublime Text中运行Python程序
 - B.2.5 配置Sublime Text
 - B.2.6 定制Sublime Text的设置
 - B.3 IDLE
 - B.3.1 在Linux系统中安装IDLE
 - B.3.2 在OS X系统中安装IDLE
 - B.3.3 在Windows系统中安装IDLE
 - B.3.4 定制IDLE的设置
 - B.4 Emacs和vim
- 附录 C 寻求帮助
- C.1 第一步
 - C.1.1 再试试
 - C.1.2 歇一会儿
 - C.1.3 参考本书的在线资源
 - C.2 在线搜索
 - C.2.1 Stack Overflow
 - C.2.2 Python官方文档
 - C.2.3 官方库文档
 - C.2.4 r/learnpython
 - C.2.5 博客
 - C.3 IRC
 - C.3.1 创建IRC账户
 - C.3.2 加入频道
 - C.3.3 IRC文化
- 附录 D 使用Git进行版本控制
- D.1 安装Git
 - D.1.1 在Linux系统中安装Git
 - D.1.2 在OS X系统中安装Git
 - D.1.3 在Windows系统中安装Git
 - D.1.4 配置Git
 - D.2 创建项目
 - D.3 忽略文件
 - D.4 初始化仓库
 - D.5 检查状态
 - D.6 将文件加入到仓库中
 - D.7 执行提交
 - D.8 查看提交历史
 - D.9 第二次提交
 - D.10 撤销修改
 - D.11 检出以前的提交
 - D.12 删除仓库

后记

版权声明

Copyright © 2016 by Eric Matthes. *Python Crash Course: A Hands-On, Project-Based Introduction to Programming*, ISBN 978-1-59327-603-4, published by No Starch Press. Simplified Chinese-language edition copyright © 2016 by Posts and Telecom Press. All rights reserved.

No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

本书中文简体字版由No Starch Press授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

谨以此书献给我的父亲，以及儿子Ever。感谢父亲抽出时间来回答我提出的每个编程问题，而儿子Ever也开始向我提问了。

前言

如何学习编写第一个程序，每个程序员都有不同的故事。我还是个孩子时就开始学习编程了，当时我父亲在计算时代的先锋之一——数字设备公司（Digital Equipment Corporation）工作。我使用一台简陋的计算机编写了第一个程序，这台计算机是父亲在家里的地下室组装而成的，它没有机箱，裸露的主板与键盘相连，显示器是裸露的阴极射线管。我编写的这个程序是一款简单的猜数字游戏，其输出类似于下面这样：

```
I'm thinking of a number! Try to guess the number I'm thinking of: 25
Too low! Guess again: 50
Too high! Guess again: 42
That's it! Would you like to play again? (yes/no) no
Thanks for playing!
```

看到家人玩着我编写的游戏，而且它完全按我预期的方式运行，我心里不知有多满足。此情此景我永远都忘不了。

儿童时期的这种体验一直影响我至今。现在，每当我通过编写程序解决了一个问题时，心里都会感到非常满足。相比于孩提时期，我现在编写的软件满足了更大的需求，但通过编写程序获得的满足感几乎与从前一样。

读者对象

本书旨在让你尽快学会Python，以便能够编写能正确运行的程序——游戏、数据可视化和Web应用程序，同时掌握让你终身受益的基本编程知识。本书适合任何年龄的读者阅读，它不要求你有任何Python编程经验，甚至不要求你有编程经验。如果你想快速掌握基本的编程知识以便专注于开发感兴趣的项目，并想通过解决有意义的问题来检查你对新学概念的理解程度，那么本书就是为你编写的。本书还可供初中和高中教师用来通过开发项目向学生介绍编程。

本书内容

本书旨在让你成为优秀的程序员，具体地说，是优秀的Python程序员。通过阅读本书，你将迅速掌握编程概念，打下坚实的基础，并养成良好的习惯。阅读本书后，你就可以开始学习Python高级技术，并能够更轻松地了解其他编程语言。

在本书的第一部分，你将学习编写Python程序时需要熟悉的基本编程概念，你刚接触几乎任何编程语言时都需要学习这些概念。你将学习各种数据以及在程序中将数据存储在列表和字典中的方式。你将学习如何创建数据集合以及如何高效地遍历这些集合。你将学习使用while和if语句来检查条件，并在条件满足时执行代码的一部分，而在条件不满足时执行代码的另一部分——这可为自动完成处理提供极大的帮助。

你将学习获取用户输入，让程序能够与用户交互，并在用户没停止输入时保持运行状态。你将探索如何编写函数来让程序的各个部分可重用，这样你编写执行特定任务的代码后，想使用它多少次都可以。然后，你将学习使用类来扩展这种概念以实现更复杂的行为，从而让非常简单的程序也能处理各种不同情形。你将学习编写妥善处理常见错误的程序。学习这些基本概念后，你就能编写一些简短的程序来解决一些明确的问题。最后，你将向中级编程迈出第一步，学习如何为代码编写测试，以便在进一步改进程序时不用担心可能引入bug。第一部分介绍的知识让你能够开发更大、更复杂的项目。

在第二部分，你将利用在第一部分学到的知识来开发三个项目。你可以根据自己的情况，以最合适的顺序完成这些项目；你也可以选择只完成其中的某些项目。在第一个项目（第12~14章）中，你将创建一个类似于《太空入侵者》的射击游戏。这个游戏名为《外星人入侵》，它包含多个难度不断增加的等级。完成这个项目后，你就能够自己动手开发2D游戏了。

第二个项目（第15~17章）介绍数据可视化。数据科学家的目标是通过各种可视化技术来搞懂海量信息。你将使用通过代码生成的数据集、已经从网络下载下来的数据集以及程序自动下载的数据集。完成这个项目后，你将能够编写能对大型数据集进行筛选的程序，并以可视化方式将筛选出来的数据呈现出来。

在第三个项目（第18~20章）中，你将创建一个名为“学习日记”的小型Web应用程序。这个项目能够让用户将学到的与特定主题相关的概念记录下来。你将能够分别记录不同的主题，还可以让其他人建立账户并开始记录自己的学习日记。你还将学习如何部署这个项目，让任何人都能够通过网络访问它，而不管他身处何方。

为何使用Python

继续使用Python，还是转而使用其他语言——也许是编程领域较新的语言？我每年都会考虑这个问题。可我依然专注于Python，其中的原因很多。Python是一种效率极高的语言：相比于众多其他的语言，使用Python编写时，程序包含的代码行更少。Python的语法也有助于创建整洁的代码：相比其他语言，使用Python编写的代码更容易阅读、调试和扩展。

大家将Python用于众多方面：编写游戏、创建Web应用程序、解决商业问题以及供各类有趣的公司开发内部工具。Python还在科学领域被大量用于学术研究和应用研究。

我依然使用Python的一个最重要的原因是，Python社区有形形色色充满激情的人。对程序员来说，社区非常重要，因为编程绝非孤独的修行。大多数程序员都需要向解决过类似问题的人寻求建议，经验最为丰富的程序员也不例外。需要有人帮助解决问题时，有一个联系紧密、互帮互助的社区至关重要，而对于像你一样将Python作为第一门语言来学习的人而言，Python社区无疑是坚强的后盾。

Python是一门杰出的语言，值得你去学习，咱们现在就开始吧！

致谢

要是没有No Starch Press出色的专业人士的帮助，本书根本不可能出版。Bill Pollock邀请我编写一本入门图书，因此这里要深深感谢他给予我这样的机会。Tyler Ortman在我编写本书的早期帮助我理清思路。Liz Chadwick和Leslie Shen仔细阅读了每一章，并提出了宝贵的意见，而Anne Marie Walker让本书的很多地方都更清晰。Riley Hoffman回答了我就图书出版过程提出的每个问题，并且耐心地将我的作品变成了漂亮的图书。

感谢技术审稿人Kenneth Love。我与Kenneth相识于一次PyCon大会，他对Python和Python社区充满热情，一直是我获取专业灵感的源泉。Kenneth不仅检查了本书介绍的知识是否正确，还抱着让初学编程者对Python语言和编程有扎实认识的目的进行了审阅。即便如此，倘若书中有任何不准确的地方，责任都完全由我承担。

感谢我的父亲，感谢他在我很小的时候就向我介绍编程，而且一点都不担心我破坏他的设备。感谢妻子Erin在我编写本书期间对我一如既往的鼓励和支持。还要感谢儿子Ever，他的好奇心每天都会给我带来灵感。

第一部分 基础知识

本书的第一部分介绍编写Python程序所需要熟悉的基本概念，其中很多都适用于所有编程语言，因此它们在你的整个程序员生涯中都很有用。

第1章介绍在计算机中安装Python，并运行第一个程序——它在屏幕上打印消息“Hello world!”。

第2章论述如何在变量中存储信息以及如何使用文本和数字。

第3章和第4章介绍列表。使用列表能够在一个变量中存储任意数量的信息，从而高效地处理数据：只需几行代码，你就能够处理数百、数千乃至数百万个值。

第5章讲解使用if语句来编写这样的代码：在特定条件满足时采取一种措施，而在该条件不满足时采取另一种措施。

第6章演示如何使用Python字典，将不同的信息关联起来。与列表一样，你也可以根据需要在字典中存储任意数量的信息。

第7章讲解如何从用户那里获取输入，让程序变成交互式的。你还将学习while循环，它不断地运行代码块，直到指定的条件不再满足为止。

第8章介绍编写函数。函数是执行特定任务的被命名的代码块，你可以根据需要随时运行它。

第9章介绍类，它让你能够模拟实物，如小狗、小猫、人、汽车、火箭等，让你的代码能够表示任何真实或抽象的东西。

第10章介绍如何使用文件，以及如何处理错误以免程序意外地崩溃。你需要在程序关闭前保存数据，并在程序再次运行时读取它们。你将学习Python异常，它们让你能够未雨绸缪，从而让程序妥善地处理错误。

第11章为代码编写测试，以核实程序是否像你期望的那样工作。这样，扩展程序时，你就不用担心引入新的bug。要想脱离初级程序员的阵容，跻身于中级程序员的行列，测试代码是你必须掌握的基本技能之一。

欢迎访问：电子书学习和下载网站 (<https://www.shgis.cn>)

文档名称：Python 编程：从入门到实践-2016-中文版.pdf

请登录 <https://shgis.cn/post/1774.html> 下载完整文档。

手机端请扫码查看：

