

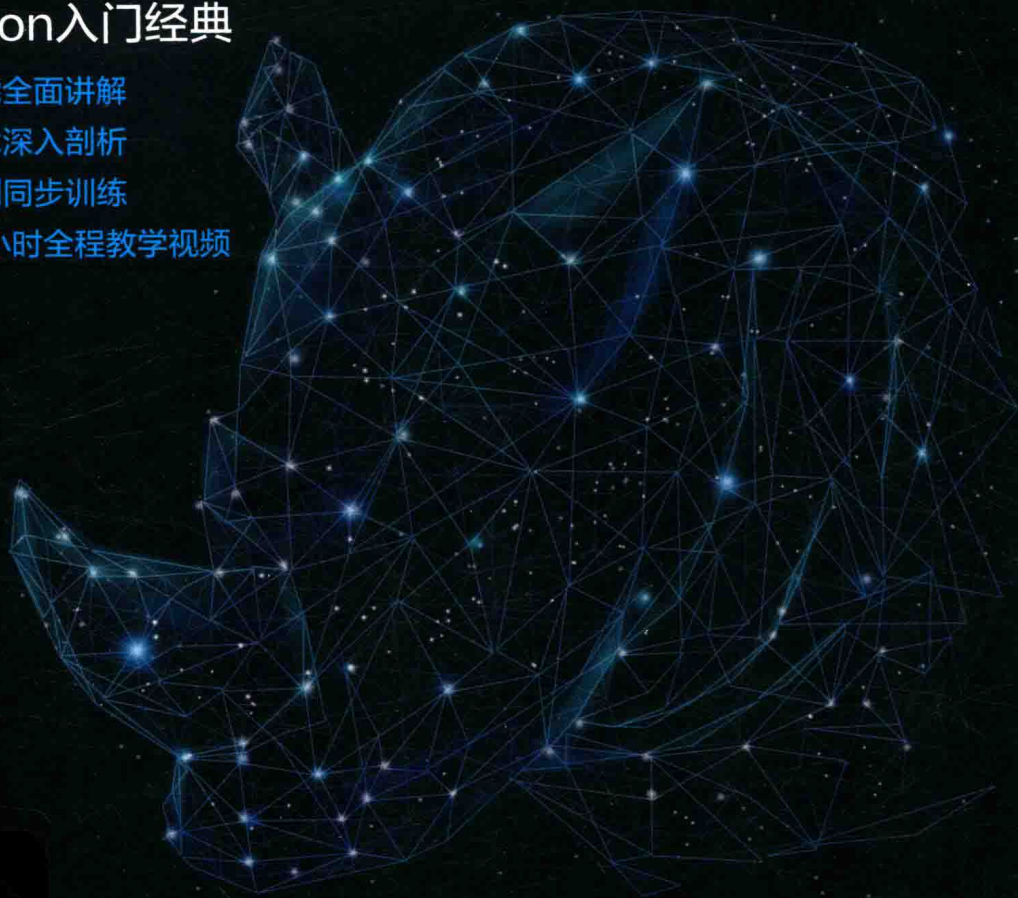
# Python 3.x

## 基础教程

史卫亚◎编著

Python入门经典

- ✓ 功能全面讲解
- ✓ 技术深入剖析
- ✓ 案例同步训练
- ✓ 32小时全程教学视频



北京大学出版社  
PEKING UNIVERSITY PRESS

# Python 3.x

## 基础教程

史卫亚◎编著



北京大学出版社  
PEKING UNIVERSITY PRESS

## 内 容 提 要

本书以零基础讲解为宗旨，旨在帮助读者掌握 Python 语言的基础知识，以及如何使用 Python 语言实现编程，了解其开发技巧，并通过实战案例熟悉开发过程及问题的解决方法。

全书共分 13 章，大致分为 4 部分：第 1~4 章介绍 Python 简介及环境搭建、Python 基础和面向对象的编程知识；第 5~7 章介绍读写文件、自带电池模块及系统编程的知识；第 8~11 章介绍网络编程、收发电子邮件、图形用户界面和 Web 开发；第 12、13 章通过两个综合案例的讲解，将全书各个知识点融会贯通，加深读者对所学知识的理解。

本书专为没有任何 Python 基础的初学者和爱好者打造，无论你是否从事计算机相关专业，是否有过 Python 项目经验，或是否想要转行从事计算机相关专业，均可通过本书快速掌握 Python 的基本知识和开发技巧。

### 图书在版编目 (CIP) 数据

Python 3.x 基础教程 / 史卫亚编著. — 北京 : 北京大学出版社, 2019.6  
ISBN 978-7-301-30450-1

I. ①P… II. ①史… III. ①软件工具 — 程序设计 — 教材 IV. ①TP311.561

中国版本图书馆 CIP 数据核字 (2019) 第 074346 号

- |       |  |
|-------|--|
| 书 名   | Python 3.x 基础教程<br>PYTHON 3.X JICHU JIAOCHENG                                    |
| 著作责任者 | 史卫亚 编著   |
| 责任编辑  | 吴晓月 刘沈君  |
| 标准书号  | ISBN 978-7-301-30450-1   |
| 出版发行  | 北京大学出版社  |
| 地 址   | 北京市海淀区成府路 205 号 100871   |
| 网 址   | <a href="http://www.pup.cn">http://www.pup.cn</a> 新浪微博 : @北京大学出版社                |
| 电子信箱  | pup7@pup.cn  |
| 电 话   | 邮购部 010-62752015 发行部 010-62750672 编辑部 010-62570390                               |
| 印 刷 者 | 三河市北燕印装有限公司  |
| 经 销 者 | 新华书店<br>787 毫米 × 1092 毫米 16 开本 29 印张 654 千字<br>2019 年 6 月第 1 版 2019 年 6 月第 1 次印刷 |
| 印 数   | 1—4000 册   |
| 定 价   | 69.00 元  |

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

**版权所有，侵权必究**

举报电话：010-62752024 电子信箱：fd@pup.pku.edu.cn

图书如有印装质量问题，请与出版部联系，电话：010-62756370

# 前言

## PREFACE

本书是专为零基础读者打造的一本编程学习用书，我们的目的是当读者系统地学习完本书的内容之后，可以骄傲地宣布：“我已经是一个 Python 专家了！”

## 为什么要写这本书

2017 年 7 月 20 日，国务院印发《新一代人工智能发展规划》，提出了面向 2030 年我国新一代人工智能发展的指导思想、战略目标、重点任务和保障措施，部署构筑我国人工智能发展的先发优势，加快建设创新型国家和世界科技强国。随着人工智能（Artificial Intelligence, AI）时代的到来，如果想要从事 AI 相关的工作，最好要熟悉并掌握 Python 开发工具。

Python 作为一种功能强大且通用的编程语言，正在得到越来越多的应用。因此一本简单易学的 Python 基础书，将使读者学习 Python 语言事半功倍。综观当前编程图书市场，这本书的优势在于立足于基础，以编程思维方法为主线，将理论知识和实际应用结合，并通过两个综合的实际案例，贯穿全书所学知识，目标是让初学者能够快速地熟悉 Python 的基本概念和编程方法，同时会使用 Python 语言解决实际的应用问题。

本书通俗易懂，是学习 Python 基础知识的入门书籍，特别适合对 Python 还不熟悉的编程者，同时对想使用 Python 语言解决一些实际问题的初学者也非常适合。

## 读者对象

- 没有任何 Python 项目经验的初学者。
- 没有 Python 基础，想学习 Python 的零基础爱好者。
- 初、中级程序员和了解简单的编程语言的人。
- 正在进行毕业设计的学生。
- 高等院校及培训学校的教师和学生。



## 本书特色

### 1. 零基础入门

无论您是否从事计算机相关行业，是否接触过 Python，是否使用 Python 开发过项目，都能通过本书开启学习之旅。

### 2. 学习与巩固相结合

书中配有课堂范例和上机实战，课堂范例便于读者学习并梳理知识点，上机实战则可以让学生随时自我检测，巩固所学知识，真正做到知识点与范例相结合。

### 3. 超多实用、专业的范例

本书结合实际工作中的范例，逐一讲解 Python 的各种知识和开发技巧。最后，通过上机实战来帮助读者巩固本章知识，轻松掌握项目开发经验。

### 4. 视频教程，轻松学习

本书赠送 32 小时全程教学视频，详细地讲解了范例和项目开发的过程和关键点，帮助读者更加轻松地掌握本书的知识。

## 本书的编写思路

在第 1 章中介绍了 Python 的起源和环境的搭建，初学者应首先学习这一章，因为后续章节的代码和案例都需要在 Python 环境中运行。

初级编程主要包含 Python 基础、面向对象基础知识和面向对象高级知识，这 3 章是培养读者的 Python 编程思维，如果读者以前具有其他高级语言的编程经验，那么这一部分将学习得更为轻松，因为许多语言都是相通的，读者只需领会 Python 语言的特点即可。

在中级编程部分，介绍了读写文件、自带电池模块及系统编程的知识，这些内容建立在初级编程的基础上，可以让读者更加理解为什么 Python 越来越受欢迎，使用起来为什么更加简单方便。

在高级编程部分，重点介绍了如何编程实现图形用户界面，如何实现网络编程，并且给出了收发电子邮件和 Web 开发的应用实现方法。在掌握这些内容之后，通过两个具体的应用案例——飞机大战和每日生鲜，将全书所学的知识点融会贯通，提高读者解决实际问题的能力。

## 如何阅读本书

本书主要介绍 Python 语言的基本概念和编程思维，分为 4 部分，分别是初级编程、中级编程、高级编程和应用案例，为了使读者能更好地学习本书，下面为读者梳理了每一章所讲的知识。

首先带领读者对 Python 有一个初步的了解,第 1 章主要介绍 Python 的起源和环境的搭建,并通过一个简单的 Python 程序,让读者对 Python 编程有一个初步的体验。

接下来为了使读者能够掌握 Python 相关知识,第 2 章介绍了 Python 编程过程中所使用的知识,主要包含变量、注释、各种运算符、判断语句、循环语句、列表和元组、字典、无序集合、字符串、函数的基本概念及使用方法,并通过简单的实例让读者知道如何在编程过程中使用这些知识。

面向对象设计是一种软件设计方法,是一种工程化规范。第 3 章介绍面向对象的编程思想,以及 Python 语言中如何实现面向对象,并通过简单的实例让读者加深对面向对象编程的理解。

在学习完面向对象的基础知识后,第 4 章继续介绍面向对象编程过程中的高级知识,内容涉及设计模式、元类、动态语言、生成器、迭代器、闭包、装饰器、内建和异常等概念。通过对这些概念的理解及使用方法的学习,读者可以更加容易地实现代码的编写和调试。

在平时编写程序的过程中,经常涉及对计算机上某个目录中的文件进行读写的问题,经常会进行创建文件和目录、删除文件和目录、重命名等操作,在第 5 章中介绍了文件读写的知识。

Python 流行的一个原因就是因为它第三方模块数量巨大,程序员编写代码不必从零开始“造轮子”,许多要用的功能在 Python 中都已经写好并封装成库了,模块就是具有某些功能的 Python 文件,第 6 章介绍了系统内置的模块和常见的第三方模块,最后通过实战案例演示了如何使用模块。

为了使指令更高效地执行,CPU 通过流水线方式或以几乎并行执行指令的工作方法来提高指令的执行速度。第 7 章重点介绍了提高计算机效率的两个重要概念——进程和线程。

随着网络的普及,人们的日常生活越来越离不开网络,如通过网络交流、网络购物等。如何通过编程在网络中各个节点之间进行信息的传递,是网络编程的重点内容,因此第 8 章就介绍了网络通信的基本原理,并重点介绍在 Python 中如何实现网络编程。

在某些应用中,可能会需要管理员给所有用户群发电子邮件,或者类似的应用。第 9 章首先介绍了网络中电子邮件的收发过程,然后介绍在 Python 中如何使用代码编程实现电子邮件的收发,并通过范例,解决一些实际问题。

现在大多数应用软件都是图形用户界面,操作方便。第 10 章介绍了 Python 的标准 GUI 工具包,以及如何实现 GUI 编程,主要包括图形库安装、主要组件、事件处理、布局和对话框等内容。通过这一章的学习,读者可以轻松编制图形窗口。

Web 应用程序是基于浏览器/服务器(Browser/Server, B/S)的一种应用程序。第 11 章首先介绍了 Web 应用程序的开发方法,主要包含 HTTP 协议和一些前端开发语言或方法,如 HTML、CSS、JavaScript、jQuery 等,然后介绍了如何简化服务器端程序开发。

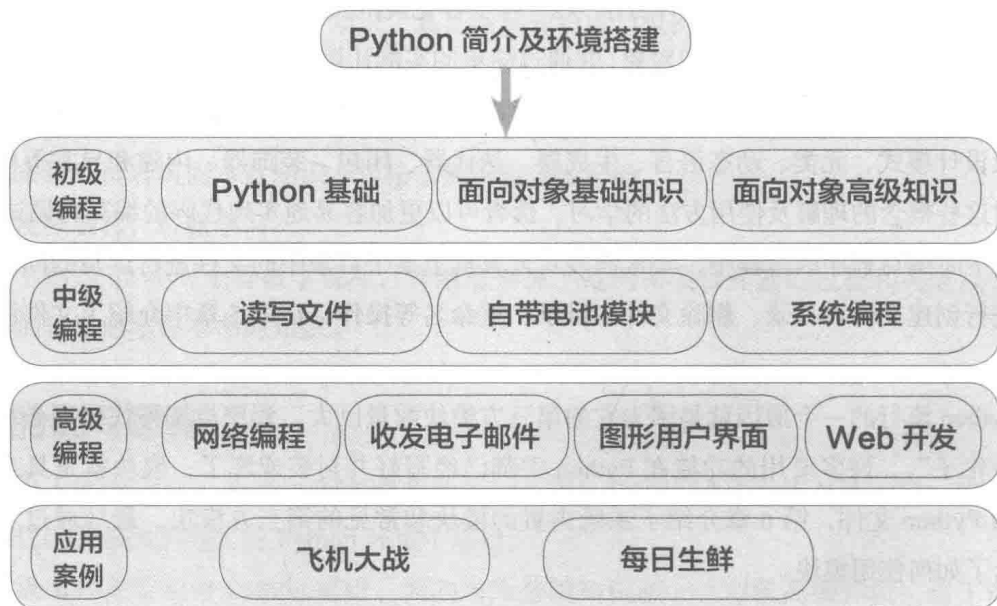


最后通过课堂范例与上机实战巩固学习的内容。

第 12 章和第 13 章通过两个典型案例进行综合讲解，让读者以编程的思维明白如何将所学知识应用到真实的项目开发中。

## Python 最佳学习路线

本书总结了作者多年的教学实践经验，为读者设计了最佳的学习路线。



## 配套资源

扫描下方二维码，即可下载本书配套素材与视频资源。



本书中源码的路径，如“03\代码\3.2类和对象\demo01.py”，其中 03 表示第三章，“代码”为目录名称，“3.2类和对象”也是目录名称，“demo01.py”为具体的源代码文件名称。

## 作者团队

本书由龙马高新教育策划，河南工业大学史卫亚老师任主编，河南工业大学王社伟老师任副主编，其中第1~8章、第12章由史卫亚老师编写，第9~11章和第13章由王社伟老师编写。在编写过程中，编者竭尽所能地为读者呈现最好、最全的实用功能，但仍难免有疏漏之处，敬请广大读者指正。读者若在学习过程中有疑问或任何建议，可以通过以下方式联系我们。

投稿邮箱：[pup7@pup.cn](mailto:pup7@pup.cn)

读者信箱：[2751801073@qq.com](mailto:2751801073@qq.com)



## 第1章 Python 简介及环境搭建..... 1

|   |    |
|---|----|
| 1.1 了解 Python.....  | 2  |
| 1.2 搭建 Python 的编程环境.....  | 3  |
| 1.2.1 Python 环境简介.....  | 3  |
| 1.2.2 Windows 下 Python 环境的搭建.....   | 4  |
| 1.2.3 Windows 环境变量的配置.....  | 6  |
| 1.2.4 Linux 下的配置.....   | 7  |
| 1.3 编写第一个 Python 程序.....  | 8  |
| 1.4 常用的 IDE 工具.....   | 10 |
|  课堂范例..... | 14 |
|  上机实战..... | 16 |

## 第2章 Python 基础..... 17

|                    |    |
|--------------------|----|
| 2.1 变量.....        | 18 |
| 2.1.1 变量的定义.....   | 18 |
| 2.1.2 变量的类型.....   | 19 |
| 2.1.3 标识符.....     | 22 |
| 2.1.4 关键字.....     | 22 |
| 2.1.5 输入和输出.....   | 23 |
| 2.1.6 数据类型转换.....  | 25 |
| 2.1.7 == 与 is..... | 26 |

## 2.2 编写注释.....27

|                 |    |
|-----------------|----|
| 2.2.1 单行注释..... | 27 |
| 2.2.2 多行注释..... | 28 |
| 2.2.3 中文注释..... | 28 |
| 2.2.4 平台注释..... | 28 |

## 2.3 运算符.....29

|                    |    |
|--------------------|----|
| 2.3.1 算术运算符.....   | 29 |
| 2.3.2 比较运算符.....   | 30 |
| 2.3.3 赋值运算符.....   | 31 |
| 2.3.4 位运算符.....    | 33 |
| 2.3.5 逻辑运算符.....   | 35 |
| 2.3.6 成员运算符.....   | 36 |
| 2.3.7 身份运算符.....   | 36 |
| 2.3.8 运算符的优先级..... | 37 |

## 2.4 判断语句.....38

|                         |    |
|-------------------------|----|
| 2.4.1 if 语句.....        | 38 |
| 2.4.2 if...else 语句..... | 39 |
| 2.4.3 if 嵌套.....        | 40 |
| 2.4.4 elif 语句.....      | 41 |

## 2.5 循环语句.....42

|                             |    |
|-----------------------------|----|
| 2.5.1 while 循环.....         | 42 |
| 2.5.2 嵌套 while 循环.....      | 42 |
| 2.5.3 for 循环.....           | 43 |
| 2.5.4 嵌套 for 循环.....        | 44 |
| 2.5.5 break 和 continue..... | 44 |



|                             |    |                        |     |
|-----------------------------|----|------------------------|-----|
| <b>2.6 列表和元组</b> .....      | 45 | 2.10.3 函数的文档说明         | 80  |
| 2.6.1 列表                    | 45 | 2.10.4 函数的4种类型         | 81  |
| 2.6.2 元组                    | 53 | 2.10.5 函数的嵌套           | 82  |
| <b>2.7 字典</b> .....         | 54 | 2.10.6 函数的参数           | 82  |
| 2.7.1 字典的定义                 | 54 | 2.10.7 变量作用域           | 87  |
| 2.7.2 字典的新增和修改              | 55 | 2.10.8 递归函数            | 89  |
| 2.7.3 字典的查询                 | 55 | 2.10.9 匿名函数            | 90  |
| 2.7.4 字典的删除                 | 57 | 2.10.10 高阶函数           | 90  |
| 2.7.5 字典的判断                 | 58 | <b>2.11 Python之“禅”</b> | 91  |
| 2.7.6 字典的遍历                 | 58 | 课堂范例                   | 92  |
| 2.7.7 字典的其他功能               | 59 | 上机实战                   | 94  |
| 2.7.8 字典与列表对比               | 61 | <b>第3章 面向对象基础知识</b>    | 95  |
| <b>2.8 无序集合</b> .....       | 61 | <b>3.1 理解面向对象的思想</b>   | 96  |
| 2.8.1 定义无序集合                | 61 | <b>3.2 类和对象</b>        | 96  |
| 2.8.2 无序集合的新增               | 62 | 3.2.1 理解类和对象           | 96  |
| 2.8.3 无序集合的删除               | 62 | 3.2.2 定义类              | 97  |
| 2.8.4 无序集合的交、并、<br>差、集      | 63 | 3.2.3 创建对象             | 98  |
| 2.8.5 无序集合的判断               | 64 | 3.2.4 self             | 99  |
| 2.8.6 无序集合与列表、元组之间<br>的相互转换 | 64 | <b>3.3 魔法方法</b>        | 99  |
| <b>2.9 字符串</b> .....        | 65 | <b>3.4 公有和私有</b>       | 102 |
| 2.9.1 字符串介绍                 | 65 | <b>3.5 继承</b>          | 105 |
| 2.9.2 字符串编码                 | 66 | 3.5.1 单继承              | 105 |
| 2.9.3 字符串的定义及访问             | 66 | 3.5.2 多继承              | 106 |
| 2.9.4 字符串的格式化               | 67 | 3.5.3 重写               | 106 |
| 2.9.5 转义字符                  | 68 | <b>3.6 多态</b>          | 109 |
| 2.9.6 字符串运算符                | 69 | <b>3.7 属性和方法</b>       | 112 |
| 2.9.7 字符串的其他常见操作            | 70 | 3.7.1 属性               | 112 |
| <b>2.10 函数</b> .....        | 79 | 3.7.2 方法               | 114 |
| 2.10.1 函数的介绍                | 79 |                        |     |
| 2.10.2 函数的定义和调用             | 79 |                        |     |

 课堂范例 ..... 116

 上机实战 ..... 120

**第4章 面向对象高级知识 ..... 121**

4.1 设计模式 ..... 122

    4.1.1 理解设计模式 ..... 122

    4.1.2 单例设计模式 ..... 122

    4.1.3 工厂设计模式 ..... 124

4.2 元类 ..... 128

4.3 动态语言 ..... 131

    4.3.1 给对象添加和删除属性 ..... 131

    4.3.2 给类添加和删除属性 ..... 132

    4.3.3 动态地给类增加方法 ..... 132

    4.3.4 魔法方法 `_slots_` ..... 134

4.4 生成器 ..... 134

4.5 迭代器 ..... 137

4.6 闭包 ..... 139

4.7 装饰器 ..... 140

4.8 属性 `property` ..... 144

    4.8.1 私有属性添加 `getter` 和 `setter` 方法 ..... 144

    4.8.2 使用 `property` 升级 `getter` 和 `setter` 方法 ..... 145

    4.8.3 使用 `@property` 代替 `getter` 和 `setter` 方法 ..... 145

4.9 内建 ..... 146

    4.9.1 内建属性 ..... 146

    4.9.2 内建函数 ..... 148

    4.9.3 内建类 ..... 149

4.10 异常 ..... 150

    4.10.1 理解异常 ..... 150

    4.10.2 捕获异常 ..... 151

    4.10.3 自定义异常 ..... 155

 课堂范例 ..... 156

 上机实战 ..... 158

**第5章 读写文件 ..... 159**

5.1 文件的打开和关闭 ..... 160

    5.1.1 打开文件 ..... 160

    5.1.2 关闭文件 ..... 161

5.2 文件的读写 ..... 161

    5.2.1 读文件 ..... 162

    5.2.2 写文件 ..... 163

5.3 操作文件和目录 ..... 164

5.4 内存中的读写 ..... 167

5.5 序列化 ..... 168

 课堂范例 ..... 173

 上机实战 ..... 174

**第6章 自带电池模块 ..... 175**

6.1 谈谈模块 ..... 176

    6.1.1 模块的介绍和使用 ..... 176

    6.1.2 包的介绍和使用 ..... 179

    6.1.3 发布和安装 ..... 182

6.2 再次探究模块 ..... 184

6.3 受人喜爱的内建模块 ..... 185

    6.3.1 `random` 模块 ..... 185

    6.3.2 `datetime` 模块 ..... 186



|            |                 |            |
|------------|-----------------|------------|
| 6.3.3      | collections 模块  | 188        |
| 6.3.4      | hashlib 模块      | 190        |
| 6.3.5      | hmac 模块         | 191        |
| 6.3.6      | base64 模块       | 192        |
| 6.3.7      | struct 模块       | 192        |
| 6.3.8      | itertools 模块    | 193        |
| 6.3.9      | contextlib 模块   | 194        |
| 6.3.10     | xml 模块          | 195        |
| 6.3.11     | html 模块         | 197        |
| 6.3.12     | urllib 模块       | 198        |
| 6.3.13     | enum 模块         | 200        |
| 6.3.14     | logging 模块      | 200        |
| 6.3.15     | re 模块           | 202        |
| <b>6.4</b> | <b>有趣的第三方模块</b> | <b>206</b> |
| 6.4.1      | PIL 模块          | 206        |
| 6.4.2      | PyMysql 模块      | 207        |
| 6.4.3      | pygame 模块       | 209        |
|            | 课堂范例            | 210        |
|            | 上机实战            | 214        |

**第7章 系统编程 ..... 215**

|            |                    |            |
|------------|--------------------|------------|
| <b>7.1</b> | <b>进程</b>          | <b>216</b> |
| 7.1.1      | 理解多任务和进程           | 216        |
| 7.1.2      | 使用 fork            | 217        |
| 7.1.3      | 多进程修改全局变量          | 218        |
| 7.1.4      | Multiprocessing 模块 | 219        |
| 7.1.5      | 进程池                | 224        |
| 7.1.6      | 进程间的通信             | 226        |
| <b>7.2</b> | <b>线程</b>          | <b>228</b> |
| 7.2.1      | 使用 threading 创建线程  | 228        |
| 7.2.2      | 进程和线程的区别           | 230        |
| 7.2.3      | 同步                 | 233        |

|       |                 |     |
|-------|-----------------|-----|
| 7.2.4 | ThreadLocal 的用法 | 238 |
| 7.2.5 | 异步              | 239 |
|       | 课堂范例            | 240 |
|       | 上机实战            | 242 |

**第8章 网络编程 ..... 243**

|            |               |            |
|------------|---------------|------------|
| <b>8.1</b> | <b>网络通信概述</b> | <b>244</b> |
| 8.1.1      | TCP/IP 协议     | 244        |
| 8.1.2      | IP 地址         | 244        |
| 8.1.3      | 端口号           | 244        |
| 8.1.4      | 子网掩码          | 245        |
| 8.1.5      | Socket        | 245        |
| <b>8.2</b> | <b>UDP 编程</b> | <b>246</b> |
| 8.2.1      | UDP 介绍        | 246        |
| 8.2.2      | 收发数据          | 247        |
| 8.2.3      | 通信过程          | 249        |
| 8.2.4      | UDP 广播        | 251        |
| <b>8.3</b> | <b>TCP 编程</b> | <b>252</b> |
| 8.3.1      | TCP 介绍        | 253        |
| 8.3.2      | TCP 客户端编程     | 253        |
| 8.3.3      | TCP 服务端编程     | 254        |
| 8.3.4      | TCP 三次握手      | 256        |
| 8.3.5      | TCP 四次挥手      | 258        |
| 8.3.6      | TCP10 种状态     | 259        |
| 8.3.7      | TCP 长连接和短连接   | 260        |
|            | 课堂范例          | 260        |
|            | 上机实战          | 262        |

**第9章 收发电子邮件 ..... 263**

|            |               |            |
|------------|---------------|------------|
| <b>9.1</b> | <b>电子邮件介绍</b> | <b>264</b> |
| 9.1.1      | 纸质邮件发送过程      | 264        |

9.1.2 电子邮件的发送流程 .....264

9.1.3 邮件收发协议 .....265

9.1.4 邮箱开发设置 .....265

**9.2 发送邮件 .....267**

9.2.1 使用 SMTP 发送邮件 .....267

9.2.2 发送纯文本格式的邮件 ....267


9.2.3 发送 HTML 格式的邮件 ...272

9.2.4 发送带附件的邮件 .....275

**9.3 接收邮件 .....278**

9.3.1 下载邮件 .....278

9.3.2 解析邮件 .....283

 课堂范例 .....287

 上机实战 .....290

**第 10 章 图形用户界面 ..... 291**

**10.1 图形用户界面介绍 .....292**

**10.2 下载和安装 Tkinter .....293**

**10.3 丰富的组件 .....294**

10.3.1 Tk 组件及 GUI 程序  
构成 .....294

10.3.2 Label 组件 .....296

10.3.3 Button 组件 .....298

10.3.4 Checkbutton 组件 .....300

10.3.5 Radiobutton 组件 .....302

10.3.6 LabelFrame 组件 .....303

10.3.7 Entry 组件 .....305

10.3.8 Listbox 组件 .....308

10.3.9 Scrollbar 组件 .....310

10.3.10 Scale 组件 .....312

10.3.11 Text 组件 .....313

10.3.12 Canvas 组件 .....315

10.3.13 Menu 组件 .....317

10.3.14 Menubutton 组件 .....319

10.3.15 OptionMenu 组件 .....321

10.3.16 Message 组件 .....323

10.3.17 Spinbox 组件 .....324

10.3.18 PanedWindow 组件 .....326

10.3.19 Toplevel 组件 .....327

**10.4 事件的处理 .....329**

10.4.1 事件绑定 .....329

10.4.2 事件序列 .....331

10.4.3 Event 对象 .....332

**10.5 智能的布局 .....335**

10.5.1 pack 布局 .....335

10.5.2 grid 布局 .....337


10.5.3 place 布局 .....339

**10.6 有趣的对话框 .....342**

10.6.1 messagebox 消息对话框 ..342

10.6.2 filedialog 文件对话框 .....344

10.6.3 colorchooser 颜色选择  
对话框 .....346

 课堂范例 .....347

 上机实战 .....350

**第 11 章 Web 开发 ..... 351**

**11.1 HTTP 协议 .....352**

11.1.1 简介 .....352

11.1.2 分析 .....352

11.1.3 流程 .....354

11.1.4 格式 .....354

**11.2 前端简介 .....359**

11.2.1 HTML 简介 .....359




|             |                      |            |
|-------------|----------------------|------------|
| 11.2.2      | CSS 简介 .....         | 362        |
| 11.2.3      | JavaScript 简介 .....  | 365        |
| 11.2.4      | jQuery 简介 .....      | 367        |
| <b>11.3</b> | <b>WSGI 接口 .....</b> | <b>369</b> |
| 11.3.1      | 理解 WSGI .....        | 369        |
| 11.3.2      | 运行 WSGI 服务 .....     | 370        |
|             | 课堂范例 .....           | 372        |
|             | 上机实战 .....           | 377        |

## 第 12 章 飞机大战 ..... 379

|        |                 |     |
|--------|-----------------|-----|
| 12.1   | 认识 pygame ..... | 380 |
| 12.2   | 事件的处理 .....     | 383 |
| 12.2.1 | 理解事件 .....      | 383 |
| 12.2.2 | 捕捉事件 .....      | 383 |
| 12.3   | 有趣的功能 .....     | 385 |
| 12.3.1 | 显示模式 .....      | 385 |
| 12.3.2 | 调整图像大小 .....    | 387 |
| 12.3.3 | 转换图像透明度 .....   | 388 |
| 12.3.4 | 音效 .....        | 389 |
| 12.4   | 图形的绘制 .....     | 390 |
| 12.4.1 | 绘制线段 .....      | 391 |
| 12.4.2 | 绘制矩形 .....      | 392 |
| 12.4.3 | 绘制多边形 .....     | 393 |
| 12.4.4 | 绘制圆形 .....      | 393 |
| 12.4.5 | 绘制椭圆形 .....     | 394 |
| 12.4.6 | 绘制弧线 .....      | 395 |
| 12.5   | 碰撞检测 .....      | 395 |
| 12.5.1 | 动画精灵 .....      | 395 |
| 12.5.2 | 碰撞检测 .....      | 398 |
| 12.6   | 飞机大战 .....      | 399 |

## 第 13 章 每日生鲜 ..... 411

|        |                      |     |
|--------|----------------------|-----|
| 13.1   | Web 项目相关知识 .....     | 412 |
| 13.1.1 | 电商项目介绍 .....         | 412 |
| 13.1.2 | 网站的运行流程 .....        | 412 |
| 13.1.3 | Django 建立网站的优势 ..... | 413 |
| 13.1.4 | 搭建开发环境 .....         | 413 |
| 13.1.5 | 小结 .....             | 417 |
| 13.2   | 商品模块 .....           | 417 |
| 13.2.1 | 模型类设计 .....          | 417 |
| 13.2.2 | 首页 .....             | 421 |
| 13.2.3 | 列表页 .....            | 423 |
| 13.2.4 | 详情页 .....            | 425 |
| 13.2.5 | 搜索页 .....            | 427 |
| 13.3   | 用户模块 .....           | 428 |
| 13.3.1 | 模型类设计 .....          | 428 |
| 13.3.2 | 注册 .....             | 429 |
| 13.3.3 | 登录 .....             | 433 |
| 13.3.4 | 安全退出 .....           | 435 |
| 13.3.5 | 个人信息 .....           | 435 |
| 13.3.6 | 收货地址 .....           | 437 |
| 13.3.7 | 最近浏览 .....           | 438 |
| 13.4   | 购物车模块 .....          | 439 |
| 13.4.1 | 设计模型类 .....          | 439 |
| 13.4.2 | 购买商品 .....           | 439 |
| 13.4.3 | 修改 .....             | 441 |
| 13.4.4 | 删除 .....             | 443 |
| 13.5   | 订单模块 .....           | 444 |
| 13.5.1 | 设计模型类 .....          | 444 |
| 13.5.2 | 下单 .....             | 445 |
| 13.5.3 | 显示 .....             | 448 |
| 13.5.4 | 付款 .....             | 449 |



# 第1章 Python简介及环境搭建

## 01

本章主要介绍 Python 的基本知识，与其他高级语言相比有哪些优缺点，以及它的主要应用。同时本章将搭建 Python 编程环境。这个编程环境是本书后面所介绍的实例的基础，也是学习 Python 的第一步，然后将通过一个简单的 Python 小程序，让读者有一个简单的编程体验。



## 1.1 了解 Python

本节首先了解 Python 的历史，然后介绍它有什么优缺点，介绍目前它的应用领域。

### 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（吉多·范罗苏姆）于 1989 年开发，第一个公开发行版发行于 1991 年。

面向对象是一种对现实世界理解和抽象的方法，是计算机编程技术发展到现在一定阶段后的产物。早期的计算机编程是基于面向过程的方法，随着计算机技术的不断发展，计算机被用于解决越来越复杂的问题，通过面向对象的方式，将现实世界的事物抽象成对象，现实世界中的关系抽象成类、继承，帮助人们实现对现实世界的抽象与数字建模。面向对象是一种程序设计范型，同时也是一种程序开发的方法。对象指的是类的集合。它将对象作为程序的基本单元，将程序和数据封装其中，以提高软件的重用性、灵活性和扩展性。

计算机不能直接理解高级语言，只能直接理解机器语言，所以必须要把高级语言翻译成机器语言，计算机才能执行高级语言编写的程序。翻译一般有两种方法：编译和解释。编译型语言写的程序在执行之前需要一个专门的编译过程，把程序编译成为机器语言的文件，以后要运行的话就不用重新翻译，直接使用编译的结果就行了。而解释型语言则不同，解释型语言的程序不需要编译，在运行程序时才翻译，每个语句都是执行的时候才翻译。这样解释型语言每执行一次就要翻译一次，解释是一句一句地翻译。

### 2 Python 的优点

- (1) 易于学习：Python 有相对较少的关键字，结构简单，语法明确。
- (2) 易于阅读：Python 代码定义清晰。
- (3) 易于维护：源代码容易维护。
- (4) 具有丰富的标准库：Python 最大的优势之一是具有丰富的库，这些库都是跨平台的，兼容性很好。
- (5) 可移植：基于其开放源代码的特性，Python 可以很方便地移植到许多平台上。
- (6) 可扩展：如果需要一段运行很快的关键代码，或者是想要编写一些不愿开放的算法，可以使用 C 或 C++ 完成那部分程序，然后从 Python 程序中调用。
- (7) 数据库：Python 提供所有主要的商业数据库的接口。
- (8) GUI 编程：Python 支持 GUI 创建和移植到许多系统中进行调用。
- (9) 可嵌入：可以将 Python 嵌入 C 或 C++ 程序中，让程序的用户获得“脚本化”的能力。



目前，许多大型网站就是用 Python 开发的，如 YouTube、Instagram，还有国内的豆瓣。很多大公司，包括 Google、Yahoo，甚至 NASA（美国航空航天局）等都大量地使用 Python。

### 3 Python 的缺点

- (1) 运行速度不快，如果对速度要求很高，可以使用 C 语言改写程序的关键部分。
- (2) 国内市场份额相对比较小。但随着时间推移，目前国内的许多软件公司已开始大规模使用 Python，尤其是在游戏和数据处理方面。
- (3) 中文资料相对匮乏。不过，随着 Python 的普及，目前已有一些优秀的教材被翻译成中文，但其中入门级教材较多，高级编程内容还是只能看英文版。

### 4 Python 应用领域

Python 作为一种功能强大且通用的编程语言，目前在国际上非常流行，正在得到越来越多的应用。图 1-1 中给出了一些经典的应用领域。



图 1-1 Python 的应用领域

## 1.2 搭建 Python 的编程环境

要想进行 Python 的学习和实验，首先需要在计算机中搭建 Python 的编程环境，本节将分别介绍在 Windows 和 Linux 系统中环境搭建及环境变量的配置过程。

### 1.2.1 Python 环境简介

Python 是跨平台的，它可以运行在 Windows、Mac 和各种 Linux/UNIX 系统中。在 Windows 中写 Python 程序，放入 Linux 中也是能够运行的。