



从入门到实战·微课视频

基础知识→核心技术→高级应用→项目实战

Python 项目案例开发 从入门到实战

爬虫、游戏和机器学习

微课版

◎ 郑秋生 夏敏捷 主编

- 零基础入门到实战
20个不同类型的完整项目案例
- 全程语音讲解
600分钟高品质配套教学视频
- 配套资源丰富
提供完整的源码和教学课件

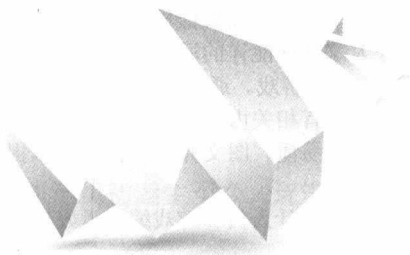
600分钟
视频讲解

清华大学出版社





从入门到实战·微课视频



Python 项目案例开发从入门到实战

爬虫、游戏和机器学习

◎ 郑秋生 夏敏捷 主编
宋宝卫 李 娟 副主编

清华大学出版社
北京

内 容 简 介

本书以 Python 3.5 为编程环境，从基本的程序设计思想入手，逐步展开 Python 语言教学，是一本面向广大编程学习者的程序设计类图书。本书以案例带动知识点的讲解，将 Python 知识点分解到各个不同的案例，每个案例各有侧重点，同时展示实际项目的设计思想和设计理念，使读者可以举一反三。

本书案例具有实用性，例如校园网搜索引擎、小小翻译器、抓取百度图片这些爬虫案例略加修改可以应用到实际项目中；还有通过微信通信协议开发微信机器人、机器学习的文本分类、基于卷积神经网络的手写体识别等案例；另外是一些大家耳熟能详的游戏案例，例如连连看、推箱子、中国象棋、网络五子棋、两人麻将、人物拼图和飞机大战等游戏。通过本书，读者将掌握 Python 编程技术和技巧，学会面向对象的设计方法，了解程序设计的所有相关内容。本书不仅为读者列出了完整的代码，同时对所有的源代码都进行了非常详细的解释，通俗易懂、图文并茂。扫描每章提供的二维码可观看知识点的视频讲解。

本书适用于 Python 语言学习者、程序设计人员和游戏编程爱好者。

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。

版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

图书在版编目（CIP）数据

Python 项目案例开发从入门到实战：爬虫、游戏和机器学习 / 郑秋生，夏敏捷主编. —北京：清华大学出版社，2019

（从入门到实战·微课视频）

ISBN 978-7-302-45970-5

I. ①P… II. ①郑… ②夏… III. ①软件工具-程序设计 IV. ①TP311.561

中国版本图书馆 CIP 数据核字（2018）第 195688 号

策划编辑：魏江江

责任编辑：王冰飞

封面设计：刘 键

责任校对：李建庄

责任印制：丛怀宇

出版发行：清华大学出版社

网 址：<http://www.tup.com.cn>, <http://www.wqbook.com>

地 址：北京清华大学学研大厦 A 座 邮 编：100084

社总机：010-62770175 邮 购：010-62786544

投稿与读者服务：010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈：010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者：北京密云胶印厂

经 销：全国新华书店

开 本：185mm×260mm 印 张：25.75 字 数：627 千字

版 次：2019 年 1 月第 1 版 印 次：2019 年 1 月第 1 次印刷

印 数：1~2000

定 价：79.80 元

产品编号：079718-01

前言

Python 语言从 20 世纪 90 年代初诞生至今, 逐渐被广泛应用于处理系统管理任务和科学计算, 是最受欢迎的程序设计语言之一。

学习编程是工程专业学生学习的重要部分, 除了直接应用外, 学习编程是了解计算机科学本质的方法。计算机科学对现代社会产生了毋庸置疑的影响。Python 是新兴的程序设计语言, 是一种解释型、面向对象、动态数据类型的高级程序设计语言。由于 Python 语言简洁、易读并且可扩展, 在国外用 Python 做科学计算的研究机构日益增多, 最近几年其社会需求逐渐增加, 许多国内高校纷纷开设 Python 程序设计课程。本书编者长期从事程序设计语言的教学与应用开发, 了解在学习编程的时候什么样的书能够提高 Python 开发能力, 以最少的时间投入得到最快的实际应用。

本书内容:

第 1 章是 Python 基础知识, 主要讲解 Python 的基础语法和面向对象编程基础、图形界面设计、Python 文件的使用、Python 的第三方库等知识, 读者可以轻松掌握。

从第 2 章开始是实用项目案例开发, 综合应用前面所学的知识, 并且每章都有突出的新知识点, 例如侧重数据库应用的案例“智力问答测试”、应用爬虫技术开发的案例“校园网搜索引擎”、应用 itchat 开发的案例“微信机器人”、机器学习案例“基于朴素贝叶斯算法的文本分类”、深度学习案例“基于卷积神经网络的手写体识别”等, 还有经典的、大家耳熟能详的游戏案例, 例如连连看、推箱子、中国象棋、两人麻将、人物拼图、网络五子棋、飞机大战等。

本书特点:

(1) Python 程序设计涉及的范围非常广泛, 本书内容的编排并不求全、求深, 而是考虑零基础读者的接受能力, 语言的语法介绍以够用、实用为原则, 选择 Python 中必备、实用的知识进行讲解, 强化对程序思维能力的培养。

(2) 案例选取贴近生活, 有助于提高读者的学习兴趣。

(3) 书中每个案例均提供了详细的设计思路、关键技术分析以及具体的解决方案。

需要说明的是, 学习编程是一个实践的过程, 而不仅仅是看书、看资料, 亲自动手编写、调试程序才是至关重要的。通过实际的编程和积极的思考, 读者可以很快地掌握许多宝贵的编程经验, 这种编程经验对开发者来说尤其不可或缺。
















本书由郑秋生和夏敏捷(中原工学院)主持编写, 吴婷(中原工学院)编写第 6 章, 韩云飞(中原工学院)编写第 8 章, 周延萍编写第 9、10 章, 宋宝卫(郑州轻工业学院)编写第 11~15 章, 陈海蕊(中原工学院)编写第 16、17 章, 高艳霞(中原工学院)编写

第 18 章，李娟（中原工学院）编写第 19 章，郑秋生编写第 20 章，其余章节由夏敏捷编写。在本书的编写过程中，为确保内容的正确性，参阅了很多资料，并且得到了中原工学院的教材资助和资深 Python 程序员的支持，在此谨向他们表示衷心的感谢。本书的学习资源可以扫描封底课件二维码获取。由于编者水平有限，书中难免有不足之处，敬请广大读者批评指正，在此表示感谢。

编者
2018 年 9 月

目 录

第 1 章

Python 基础知识	1
1.1 Python 语言简介 	1
1.2 Python 语法基础	2
1.2.1 Python 数据类型 	2
1.2.2 序列数据结构 	4
1.2.3 Python 控制语句 	12
1.2.4 Python 函数与模块 	18
1.3 Python 面向对象设计	22
1.3.1 定义和使用类 	22
1.3.2 构造函数	23
1.3.3 析构函数	24
1.3.4 实例属性和类属性	24
1.3.5 私有成员与公有成员	25
1.3.6 方法	26
1.3.7 类的继承 	27
1.3.8 多态 	29
1.3.9 面向对象应用案例——扑克牌发牌程序 	31
1.4 Python 图形界面设计	34
1.4.1 创建 Windows 窗口 	35
1.4.2 几何布局管理器	35
1.4.3 Tkinter 组件	39
1.4.4 Tkinter 字体	49
1.4.5 Python 事件处理 	51
1.4.6 图形界面设计应用案例——开发猜数字游戏	55
1.5 Python 文件的使用 	57
1.5.1 打开/建立文件 	57
1.5.2 读取文本文件	59
1.5.3 写文本文件 	60
1.5.4 文件内移动 	62

1.5.5	文件的关闭	63
1.5.6	二进制文件的读/写	64
1.6	Python 的第三方库	66

第 2 章

序列应用——猜单词游戏 67

2.1	猜单词游戏功能介绍	67
2.2	程序设计的思路	67
2.3	关键技术——random 模块	68
2.4	程序设计的步骤	71

第 3 章

数据库应用——智力问答测试 73

3.1	智力问答测试功能介绍	73
3.2	程序设计的思路	73
3.3	关键技术	74
3.3.1	访问数据库的步骤	74
3.3.2	创建数据库和表	75
3.3.3	数据库的插入、更新和删除操作	76
3.3.4	数据库表的查询操作	77
3.3.5	数据库使用实例——学生通讯录	77
3.4	程序设计的步骤	80
3.4.1	生成试题库	80
3.4.2	读取试题信息	81
3.4.3	界面和逻辑设计	81

第 4 章

调用百度 API 应用——小小翻译器 83

4.1	小小翻译器功能介绍	83
4.2	程序设计的思路	83
4.3	关键技术	84
4.3.1	urllib 库简介	84
4.3.2	urllib 库的基本使用	84
4.4	程序设计的步骤	90
4.4.1	设计界面	90
4.4.2	使用百度翻译开放平台 API	90

第 5 章

爬虫应用——校园网搜索引擎 95

5.1	校园网搜索引擎功能分析	95
5.2	校园网搜索引擎系统设计	95
5.3	关键技术	98
5.3.1	正则表达式	98

5.3.2	中文分词	103
5.3.3	安装和使用 jieba	103
5.3.4	为 jieba 添加自定义词典	104
5.3.5	文本分类的关键词提取	105
5.3.6	deque	106
5.4	程序设计的步骤	107
5.4.1	信息采集模块——网络爬虫的实现	107
5.4.2	索引模块——建立倒排词表	111
5.4.3	网页排名和搜索模块	113

第 6 章

	爬虫应用——抓取百度图片	116
6.1	程序功能介绍	116
6.2	程序设计的思路	116
6.3	关键技术	117
6.3.1	图片文件下载到本地	117
6.3.2	爬取指定网页中的图片	117
6.3.3	BeautifulSoup 库概述	119
6.3.4	用 BeautifulSoup 库操作解析 HTML 文档树	121
6.3.5	requests 库的使用	125
6.4	程序设计的步骤	133
6.4.1	分析网页源代码和网页结构	133
6.4.2	设计代码	136

第 7 章

	itchat 应用——微信机器人	139
7.1	itchat 功能介绍	139
7.2	程序设计的思路	140
7.3	关键技术	140
7.3.1	安装 itchat	140
7.3.2	itchat 的登录微信	140
7.3.3	itchat 的消息类型	141
7.3.4	itchat 回复消息	143
7.3.5	itchat 获取账号	145
7.3.6	itchat 的一些简单应用	147
7.3.7	Python 调用图灵机器人 API 实现简单的人机交互	150
7.4	程序设计的步骤	152
7.5	开发消息同步机器人	153

第 8 章

	微信网页版协议应用——微信机器人	155
8.1	微信网页版机器人功能介绍	155

8.2	微信网页版机器人设计思路	155
8.2.1	分析微信网页版 API	155
8.2.2	API 汇总	158
8.2.3	其他说明	164
8.3	程序设计的步骤	166
8.3.1	微信网页版的运行流程	166
8.3.2	程序目录	167
8.3.3	微信网页版运行代码的实现	167
8.4	扩展功能	170
8.4.1	自动回复	170
8.4.2	群发消息、定时发送消息、好友状态检测	173
8.4.3	自动邀请好友加入群聊	175

第 9 章

图像处理——生成二维码和验证码

9.1	二维码介绍	178
9.2	二维码生成和解析关键技术	179
9.2.1	qrcode 库的使用	179
9.2.2	PIL 库的使用	182
9.3	二维码生成和解析程序设计的步骤	184
9.3.1	生成带有图标的二维码	184
9.3.2	Python 解析二维码图片	186
9.4	用 Python 生成验证码图片	186

第 10 章

益智游戏——连连看游戏

10.1	连连看游戏介绍	189
10.2	程序设计的思路	190
10.3	关键技术	200
10.3.1	图形绘制——Tinker 的 Canvas 组件	200
10.3.2	Canvas 上的图形对象	200
10.4	程序设计的步骤	210

第 11 章

益智游戏——推箱子游戏

11.1	推箱子游戏介绍	215
11.2	程序设计的思路	216
11.3	关键技术	217
11.4	程序设计的步骤	218

第 12 章

娱乐游戏——两人麻将游戏

12.1	麻将游戏介绍	224
------	--------	-----

12.1.1	麻将术语	224
12.1.2	牌数	224
12.2	两人麻将游戏设计的思路	225
12.2.1	素材图片	225
12.2.2	游戏的逻辑实现	226
12.2.3	碰/吃牌的判断	226
12.2.4	和牌算法	227
12.2.5	实现计算机智能出牌	231
12.3	关键技术	233
12.3.1	声音的播放	233
12.3.2	返回对应位置的组件	233
12.3.3	对保存麻将牌的列表排序	234
12.4	两人麻将游戏设计的步骤	235
12.4.1	设计麻将牌类	235
12.4.2	设计游戏主程序	237

第 13 章

网络编程案例——基于 TCP 的在线聊天程序..... 247

13.1	基于 TCP 的在线聊天程序简介	247
13.2	关键技术	247
13.2.1	互联网 TCP/IP 协议	247
13.2.2	IP 协议和端口	248
13.2.3	TCP 协议和 UDP 协议	249
13.2.4	Socket	249
13.2.5	多线程编程	254
13.3	在线聊天程序设计的步骤	256
13.3.1	在线聊天程序的服务器端	256
13.3.2	在线聊天程序的客户端	259

第 14 章

网络通信案例——基于 UDP 的网络五子棋 游戏..... 263

14.1	网络五子棋游戏简介	263
14.2	五子棋游戏的设计思想	264
14.3	关键技术	267
14.3.1	UDP 编程	267
14.3.2	自定义网络五子棋游戏的通信协议	269
14.4	网络五子棋游戏程序设计的步骤	271
14.4.1	服务器端程序设计的步骤	271
14.4.2	客户端程序设计的步骤	276

第 15 章

益智游戏——中国象棋 281

- 15.1 中国象棋介绍 281
- 15.2 关键技术 282
- 15.3 中国象棋的设计思路 284
- 15.4 中国象棋实现的步骤 287

第 16 章

娱乐游戏——人物拼图游戏 297

- 16.1 人物拼图游戏介绍 297
- 16.2 程序设计的思路 298
- 16.3 关键技术 298
 - 16.3.1 复制和粘贴图像区域 298
 - 16.3.2 调整尺寸和旋转 298
 - 16.3.3 转换成灰度图像 299
 - 16.3.4 对像素进行操作 300
- 16.4 程序设计的步骤 300
 - 16.4.1 Python 处理图片切割 300
 - 16.4.2 游戏的逻辑实现 302

第 17 章

基于 Pygame 的游戏设计 306

- 17.1 Pygame 基础知识 306
 - 17.1.1 安装 Pygame 库 306
 - 17.1.2 Pygame 的模块 306
- 17.2 Pygame 的使用 309
 - 17.2.1 Pygame 开发游戏的主要流程 309
 - 17.2.2 Pygame 的图像/图形绘制 311
 - 17.2.3 Pygame 的键盘和鼠标事件的处理 314
 - 17.2.4 Pygame 的字体使用 319
 - 17.2.5 Pygame 的声音播放 320
 - 17.2.6 Pygame 的精灵使用 321
- 17.3 基于 Pygame 设计贪吃蛇游戏 326
- 17.4 基于 Pygame 设计飞机大战游戏 333
 - 17.4.1 游戏角色 333
 - 17.4.2 游戏界面显示 336
 - 17.4.3 游戏的逻辑实现 338

第 18 章

机器学习案例——基于朴素贝叶斯算法的
文本分类 343

- 18.1 文本分类功能介绍 343

18.2	程序设计的思路	343
18.3	关键技术	344
18.3.1	贝叶斯算法的理论基础	344
18.3.2	朴素贝叶斯分类	346
18.3.3	使用 Python 进行文本分类	348
18.4	程序设计的步骤	348
18.4.1	收集训练数据	348
18.4.2	准备数据	349
18.4.3	分析数据	349
18.4.4	训练算法	350
18.4.5	测试算法并改进	353
18.4.6	使用算法进行文本分类	354
18.5	使用朴素贝叶斯分类算法过滤垃圾邮件	355
18.5.1	收集训练数据	355
18.5.2	将文本文件解析为词向量	356
18.5.3	使用朴素贝叶斯算法进行邮件分类	357
18.5.4	改进算法	359
18.6	使用 Scikit-Learn 库进行文本分类	360
18.6.1	文本分类常用的类和函数	360
18.6.2	案例实现	363

第 19 章

	深度学习案例——基于卷积神经网络的手写体识别	366
19.1	手写体识别案例需求	366
19.2	深度学习的概念及关键技术	366
19.2.1	神经网络模型	366
19.2.2	深度学习之卷积神经网络	367
19.3	Python 深度学习库——Keras	372
19.3.1	Keras 的安装	372
19.3.2	Keras 的网络层	372
19.3.3	用 Keras 构建神经网络	375
19.4	程序设计的思路	376
19.5	程序设计的步骤	377
19.5.1	MNIST 数据集	377
19.5.2	手写体识别案例实现	378
19.5.3	预测自己手写图像	382

第 20 章

词云实战——爬取豆瓣影评生成词云 383

20.1 功能介绍  383

20.2 程序设计的思路 384

20.3 关键技术  385

20.3.1 安装 WordCloud 385

20.3.2 使用 WordCloud 385

20.4 程序设计的步骤  389

参考文献

..... 397

第 1 章

Python 基础知识

Python 是一门跨平台、开源、免费的解释型高级动态编程语言，Python 作为动态语言更适合初学编程者。Python 可以让初学者把精力集中在编程对象和思维方法上，而不用担心语法、类型等外在因素。Python 易于学习，拥有大量的库，可以高效地开发各种应用程序。

1.1 Python 语言简介

Python 由吉多范罗·苏姆（Guido van Rossum）于 1989 年底发明，被广泛应用于处理系统管理任务和科学计算，是最受欢迎的程序设计语言之一。2011 年 1 月，它被 TIOBE 编程语言排行榜评为 2010 年度语言。自从 2004 年以后，Python 的使用率呈线性增长，TIOBE 最近公布的 2018 年 9 月编程语言指数排行榜中，Python 超越 C++，首次排名处于第三位（前两位是 Java 和 C）。2017 年 7 月，根据 IEEE Spectrum 发布的研究报告显示，Python 已经成为世界上最受欢迎的语言。

Python 支持命令式编程、函数式编程，完全支持面向对象程序设计，语法简洁清晰，并且拥有大量的几乎支持所有领域应用开发的成熟扩展库。

Python 为用户提供了非常完善的基础代码库，覆盖了网络、文件、GUI、数据库、文本等大量内容，用 Python 开发，许多功能不必从零编写，直接使用现成的即可。除了内置的库外，Python 还有大量的第三方库，也就是别人开发的，大家可以直接使用的库。当然，也可以自己开发代码通过很好地封装，作为第三方库给别人使用。Python 就像胶水一样，可以把用多种不同语言编写的程序融合到一起实现无缝拼接，更好地发挥不同语言和工具的优势，满足不同应用领域的需求。所以，Python 程序看上去简单易懂，初学者学 Python，不但入门容易，而且将来深入下去，可以编写那些非常复杂的程序。

Python 同时支持伪编译，将 Python 源程序转换为字节码来优化程序和提高运行速度，可以在没有安装 Python 解释器和相关依赖包的平台上运行。

Python 语言的应用领域主要如下。

(1) Web 开发: Python 语言支持网站开发, 比较流行的开发框架有 web2py、Django 等。许多大型网站就是用 Python 开发的, 例如 YouTube、Instagram 等。很多大公司, 例如 Google、Yahoo 等, 甚至 NASA (美国航空航天局) 都大量地使用 Python。

(2) 网络编程: Python 语言提供了 socket 模块, 对 Socket 接口进行了两次封装, 支持 Socket 接口的访问; 还提供了 urllib、httplib、scrapy 等大量模块, 用于对网页内容进行读取和处理, 并可以结合多线程编程以及其他有关模块快速开发网页爬虫之类的应用程序; 可以使用 Python 语言编写 CGI 程序, 也可以把 Python 程序嵌入到网页中运行。

(3) 科学计算与数据可视化: Python 中用于科学计算与数据可视化的模块很多, 例如 NumPy、SciPy、Matplotlib、Traits、TVTK、Mayavi、VPython、OpenCV 等, 涉及的应用领域包括数值计算、符号计算、二维图表、三维数据可视化、三维动画演示、图像处理以及界面设计等。

(4) 数据库应用: Python 数据库模块有很多, 例如可以通过内置的 sqlite3 模块访问 SQLite 数据库; 使用 pywin32 模块访问 Access 数据库; 使用 pymysql 模块访问 MySQL 数据库; 使用 pywin32 和 pymssql 模块访问 SQL Server 数据库。

(5) 多媒体开发: PyMedia 模块可以对 WAV、MP3、AVI 等多媒体格式文件进行编码、解码和播放; PyOpenGL 模块封装了 OpenGL 应用程序编程接口, 通过该模块可以在 Python 程序中集成二维或三维图形; PIL (Python Imaging Library, Python 图形库) 为 Python 提供了强大的图像处理功能, 并提供广泛的图像文件格式支持。

(6) 电子游戏应用: Pygame 就是用来开发电子游戏软件的 Python 模块, 使用 Pygame 模块可以在 Python 程序中创建功能丰富的游戏和多媒体程序。

Python 有大量的第三方库, 可以说需要什么应用就能找到什么 Python 库。



视频讲解

1.2 Python 语法基础

1.2.1 Python 数据类型

计算机程序理所当然地可以处理各种数值。计算机能处理的远不止数值, 还可以处理文本、图形、音频、视频、网页等各种各样的数据, 不同的数据需要定义不同的数据类型。

① 数值类型

Python 数值类型用于存储数值, Python 支持以下数值类型。

- 整型 (int): 通常被称为整型或整数, 是正或负整数, 不带小数点。在 Python3 中只有一种整数类型 (int), 没有 Python2 中的 long。
- 浮点型 (float): 浮点型由整数部分与小数部分组成, 浮点型也可以使用科学记数法



视频讲解

表示 (2.78E2 就是 $2.78 \times 10^2 = 278$)。

- 复数 (complex): 复数由实数部分和虚数部分构成, 可以用 `a+bj` 或者 `complex(a,b)` 表示, 复数的虚部以字母 `j` 或 `J` 结尾, 例如 `2+3j`。

数据类型是不允许改变的, 这就意味着如果改变数值数据类型的值, 将重新分配内存空间。

② 字符串

字符串是 Python 中最常用的数据类型。用户可以使用引号来创建字符串。Python 不支持字符类型, 单字符在 Python 也是作为一个字符串使用。Python 使用单引号和双引号来表示字符串是一样的。

③ 布尔类型

Python 支持布尔类型的数据, 布尔类型只有 `True` 和 `False` 两种值, 但是布尔类型有以下几种运算。

(1) `and` (与) 运算: 只有两个布尔值都为 `True` 时计算结果才为 `True`。

```
True and True      #结果是 True
True and False     #结果是 False
False and True     #结果是 False
False and False    #结果是 False
```

(2) `or` (或) 运算: 只要有一个布尔值为 `True`, 计算结果就是 `True`。

```
True or True       #结果是 True
True or False      #结果是 True
False or True      #结果是 True
False or False     #结果是 False
```

(3) `not` (非) 运算: 把 `True` 变为 `False`, 或者把 `False` 变为 `True`。

```
not True           #结果是 False
not False          #结果是 True
```

布尔运算在计算机中用来做条件判断, 根据计算结果为 `True` 或者 `False`, 计算机可以自动执行不同的后续代码。

在 Python 中, 布尔类型还可以与其他数据类型做 `and`、`or` 和 `not` 运算, 这时下面的几种情况会被认为是 `False`: 为 0 的数字, 包括 `0`、`0.0`; 空字符串 `''`、`''''`; 表示空值的 `None`; 空集合, 包括空元组 `()`、空序列 `[]`、空字典 `{}`。其他的值都为 `True`。例如:

```
a='python'
print(a and True)  #结果是 True
b=''
print(b or False) #结果是 False
```

④ 空值

空值是 Python 中的一个特殊值, 用 `None` 表示。它不支持任何运算, 也没有任何内置函数方法。None 和任何其他数据类型比较永远返回 `False`。在 Python 中未指定返回值的函

数会自动返回 None。

1.2.2 序列数据结构

数据结构是计算机存储、组织数据的方式。序列是 Python 中最基本的数据结构。序列中的每个元素都分配一个数字，即它的位置或索引，第 1 个索引是 0，第 2 个索引是 1，依此类推。序列都可以进行的操作包括索引、截取（切片）、加、乘、成员检查。此外，Python 已经内置确定序列的长度以及确定最大和最小元素的方法。Python 内置序列类型最常见的是列表、元组和字符串。另外，Python 提供了字典和集合这样的数据结构，它们属于无顺序的数据集合体，不能通过位置索引来访问数据元素。

① 列表

列表 (List) 是最常用的 Python 数据类型，列表的数据项不需要具有相同的类型。列表类似其他语言的数组，但功能比数组强大得多。

创建一个列表，只要把逗号分隔的不同数据项使用方括号括起来即可，实例如下。

```
list1=['中国', '美国', 1997, 2000];  
list2=[1, 2, 3, 4, 5];  
list3=["a", "b", "c", "d"];
```

列表索引从 0 开始。列表可以进行截取（切片）、组合等。

1) 访问列表中的值

使用下标索引来访问列表中的值，同样可以使用方括号的形式截取字符，实例如下。

```
list1=['中国', '美国', 1997, 2000];  
list2=[1, 2, 3, 4, 5, 6, 7];  
print("list1[0]: ", list1[0])  
print("list2[1:5]: ", list2[1:5])
```

以上实例的输出结果：

```
list1[0]: 中国  
list2[1:5]: [2, 3, 4, 5, 6]
```

2) 更新列表

可以对列表的数据项进行修改或更新，实例如下。

```
list=['中国', 'chemistry', 1997, 2000];  
print("Value available at index 2 : ")  
print(list[2])  
list[2]=2001;  
print("New value available at index 2 : ")  
print(list[2])
```

以上实例的输出结果：

```
Value available at index 2 :  
1997
```



视频讲解