大数据
教育丛书

Python

基础案例教程

主　编　强　彦　郭志强
副主编　赵涓涓

西安电子科技大学出版社
http://www.xduph.com

大数据
教育丛书

Python

基础案例教程

主　编　强　彦　郭志强
副主编　赵涓涓

西安电子科技大学出版社
http://www.xduph.com

内容简介

本书是针对零基础读者编写的一本入门书，以案例驱动的方式讲解知识点，包含了 Python 的入门介绍、流程控制、序列、函数、对象、异常和文件等主要内容，最后给出一个大的项目案例，详细介绍了相关知识，给出了实现步骤和对应的代码。

本书既可作为计算机、软件工程、大数据等相关专业大学本科生、研究生的教材，也可作为各种 Python 语言编程实践班的培训教材，亦可供广大程序开发人员参考。

图书在版编目(CIP)数据

Python 基础案例教程/强彦，郭志强主编. —西安：西安电子科技大学出版社，2019.6
ISBN 978-7-5606-5388-4

Ⅰ. ① P… Ⅱ. ① 强… ②郭… Ⅲ. ① 软件工具—程序设计—教材
Ⅳ. ① TP311.561

中国版本图书馆 CIP 数据核字(2019)第 115494 号

策划编辑　万晶晶
责任编辑　马晓娟
出版发行　西安电子科技大学出版社(西安市太白南路 2 号)
电　　话　(029)88242885　88201467　　邮　　编　710071
网　　址　www. xduph. com　　电子邮箱　xdupfxb001@163. com
经　　销　新华书店
印刷单位　咸阳华盛印务有限责任公司
版　　次　2019 年 6 月第 1 版　2019 年 6 月第 1 次印刷
开　　本　787 毫米×960 毫米　1/16　印张　12.75
字　　数　257 千字
印　　数　1～8000 册
定　　价　36.00 元
ISBN 978-7-5606-5388-4/TP
XDUP 5690001-1
＊＊＊如有印装问题可调换＊＊＊

前　言

编程语言对于程序员来说意味着什么?

当一个程序员解释什么是编程的时候，通常会说:“编程是告诉计算机做什么。”现实中，编程语言是程序员表达和交流思想的工具，观众是其他的程序员而不是计算机。在程序中所表达的思想会传达到终端用户，尽管这些终端用户可能不会编程甚至从来没读过代码，但是确实是用户受益。

想象下 Google 和 Facebook 这样无比成功的公司取得的难以置信的成就。他们核心的理念是“关于计算机能够为人类做些什么”，而这些核心的理念是用程序语言来表达的。最适合表达这个理念的程序语言是最受欢迎并且简单实用的，而 Python 就是当前能够满足这些需求的，且最容易表达程序员思想的高级程序语言。

Python 是一门计算机程序设计语言，从特点来看，它是一种“面向对象”的语言，同时也是一门“解释型”语言。我们知道，计算机的程序设计语言有很多，有最经典的语言 C，有面向对象的编程语言 C++、Java、C#，以及解释型语言 JavaScrpit、shell、perl 等，还有适用于数据计算的 R 语言和简便易行的 GO 语言。Python 语言能够从众多编程语言中脱颖而出，是因为它高度结合了解释性、编译性、互动性和面向对象等特性，而且具有很强的可读性，简单易学。

首先，Python 语言是一门解释型语言，它的语法更接近于人类的语言。因为它通过解释器逐行解释并执行程序，所以和 C 语言等编译型语言相比，较为占用 CPU、内存等硬件资源，执行效率和执行速度都无法媲美编译型语言。但是 Python 语言拥有强大、巨量的库，而且对 C 类语言有较强的黏合性，通过 Python 可以直接执行 C、C++、Java 等语言开发的程序，从而弥补了其性能上的不足。

其次，Python 是极少数编程语言中既简单又功能强大的编程语言。它专注于如何解决问题，而非拘泥于语法与结构。它自由开放，可以跨平台运行，且拥有巨量的库来帮助编程员更快地实现程序功能。它拥有良好的扩展性，可以结合 C、Java 等其他语言，实现特定的功能。

正如 Python 官方的解读:Python 是一款易于学习且功能强大的编程语言。它具有高效率的数据结构，能够简单又有效地实现面向对象编程。Python 简洁的语法与动态输入特性，加之其解释性语言的本质，使得它成为一种在多种领域

与绝大多数平台上都能进行脚本编写与应用，并快速进行开发工作的理想语言。

本书是一本案例驱动的编程实用指南，以案例需求的方式引导读者一步一步学习编程，从简单的输出一直到完整项目的实现，让初学者从基础的编程技术入手，最终体验到软件开发的基本过程。本书的一大特色是以实例为依据，介绍了很多基于 Python 的实战技术。本书以 Python 语言的实际应用为目标，系统地训练在开发应用系统的软件工程中，安装、设计、开发和调优各个环节的相关技术及开发方法。本书从技术角度阐述开发 Python 语言系统的基本要求，并以程序开发为导向，从系统设计开发的各个技术层面设计案例，展示 Python 语言编程实战的全过程。

本书将理论与实践充分结合，发挥高校和企业的各自优势，配套了线上教学及学习平台，以案例驱动教学为核心，由案例引出知识点，简单直观地让初学者了解各知识点，单点突破、快速上手。

注：书中的“扫码做练习”如果在手机上运行不流畅，建议在电脑上做练习，网址为 http：//r. alphacoding. cn/pyjc。电脑的建议配置如下：

（1）PC 或 Mac 电脑；

（2）最新版 Chrome 浏览器，或最新版 360 极速浏览器（IE 10 或以下无法支持）。

本书既可作为大学本科生、研究生相关课程的教材，也可作为各种 Python 语言编程实践班的培训教材，同时还可供广大程序开发人员参考。

本书共分为 8 章。其中第 1 章、第 2 章由太原理工大学强彦编写，第 3 章、第 4 章由山西维信致远科技有限公司郭志强编写，第 5 章、第 6 章由太原科技大学蔡星娟编写，第 7 章、第 8 章由太原理工大学赵涓涓编写，山西维信致远科技有限公司田璟霞、黄杰、岳迎春、王新娇、董东杰、潘瑞峰参与了本书的实践设计和案例开发工作，全书由强彦审阅。本书的出版得到了山西维信致远科技有限公司的大力支持和帮助，在此致以衷心的感谢。

由于编者水平有限，不当之处在所难免，恳请读者及同仁赐教指正。

编　者

2019 年 4 月

目　录

第 1 章

解读 Python 世界

1.1 什么是 Python

Python(大蟒蛇)是一种编程语言。编程语言(Programming Language)是用来定义计算机程序的形式语言，它是一种被标准化的交流技巧，用来向计算机发出指令。好比中国人和美国人沟通可能需要英语，人类和计算机沟通就需要编程语言。编程语言分为机器语言、汇编语言以及高级语言，一般我们将机器语言、汇编语言这样的偏向底层设计的语言统称为低级语言。称它们为低级语言，并不是说它们的功能少，而是相对于高级语言来说，它们太难理解、沟通，程序员的学习成本及编码难度较大。而对高级语言，程序员更加容易接受其思想，语法也更容易让人理解。其实对计算机来说，无论是高级语言还是低级语言，计算机能读懂的只有机器语言，也就是“0”“1”序列。要想让计算机“痛快地干活”，我们往往需要将现实中想要实现的功能由程序员翻译成高级语言，再由计算机负责将高级语言翻译成低级语言，将低级语言翻译成机器语言，当然这个流程是由计算机完成的，程序员只要理解需求并通过高级语言实现需求即可。

高级语言的代表有 Python、Java、PHP、C#、C++等。低级语言的代表有汇编语言。

编程语言其实也可以按另一个维度划分，即可以分为编译型语言和解释型语言。

编译型语言：通过编译器把源程序的每一条语句都编译成机器语言，并保存成二进制文件，计算机直接以机器语言来运行此程序，运行速度很快。

解释型语言：在执行程序时，通过解释器将程序语句一条一条地解释成机器语言后由计算机来执行，在执行流程上是边解释边执行，其特点是运行速度相对编译型语言来说较慢。

当然还有其他的划分方式，比如划分为静态语言和动态语言，强类型定义语言和弱类型定义语言等，这里不再赘述。

Python 是一门解释型、面向对象、带有动态语义的高级程序设计语言。

现在，全世界差不多有 600 多种编程语言，但流行的编程语言也就 20 几种。如果你听说过 TIOBE 排行榜，就能知道编程语言的大致流行程度。近日，TIOBE 排行榜官方正式宣布，时隔 8 年后：Python 再一次赢得了“年度编程语言”的称号！现在学习 Python 已成为一种潮流。

1.2 Python 的由来

Python 是著名的“龟叔”Van Rossum Guido 于 1989 年圣诞节期间，在阿萨姆特丹为了打发圣诞节的无趣，开发的一个新的脚本解释程序，它是 ABC 语言的一种继承。之所以选中 Python(大蟒蛇的意思)作为该编程语言的名字，是因为“龟叔”是一个名叫 Monty Python 的喜剧团体的成员。Python 的第一个发行版公开于 1991 年。Python 也是一款纯粹的自由软件，源代码和解释器 CPython 都遵循了 GPL(GNU General Public License)协议。

Python 可以应用于众多领域，如数据分析、组件集成、网络服务、图像处理、数值计算和科学计算等领域。目前业内几乎所有大中型互联网企业都在使用 Python，如 Youtube、Dropbox、BT、Quora、豆瓣、知乎、Google、Yahoo!、Facebook、NASA、百度、腾讯、汽车之家、美团等。互联网公司普遍使用Python来做的事一般有自动化运维、自动化测试、大数据分析、爬虫、Web 等。

下面我们一起看一下 Python 之父 Van Rossum Guido 的编程理念。

Beautiful is better than ugly. 优美胜于丑陋。

Explicit is better than implicit. 明了胜于晦涩。

Simple is better than complex. 简单胜过复杂。

Complex is better than complicated. 复杂胜过凌乱。

Flat is better than nested. 扁平胜于嵌套。

Sparse is better than dense. 间隔胜于紧凑。

Readability counts. 可读性很重要。

Special cases aren't special enough to break the rules. 即使假借特例的实用性之名，也不应违背规则。

Errors should never pass silently. 错误不应该被无声地忽略。

In the face of ambiguity, refuse the temptation to guess. 当存在多种可能时，不要尝试去猜测。

Now is better than never. 现在做总比不做好。

If the implementation is hard to explain, it's a bad idea. 如果这个实现不容易解释，那么它肯定是个坏主意。

If the implementation is easy to explain, it may be a good idea. 如果这个实现容易解释，那么它很可能是个好主意。

Namespaces are one honking great idea—let's do more of those! 命名空间是一种绝妙的理念，应当多加利用。

1.3 Python 的版本区别

Python 是一门很优雅的语言，无论是选择 Python 2 还是 Python 3，都能够做出一些令人兴奋的软件项目。很多人被它的这一特点吸引过来。

Python 的 3.0 版本常被称为 Python 3000，简称 Python 3。相对于 Python 的早期版本，此版本有一个较大的升级。为了不带入过多的累赘，Python 3 在设计的时候没有考虑向下兼容。

Python 2 在 2020 年就不再被支持了，所以如果没有特殊要求，建议直接学 Python 3，现在网上针对 Python 3 的资源也很丰富。

虽然有几个关键的区别，比如 print、整数除法、对 Unicode 的支持等，但是通过做一些调整，从 Python 2 跨越到 Python 3 并不是太困难，并且在 Python 2.7 上可以轻松地运行 Python 3 的代码。重要的是，随着越来越多的开发人员和团队的注意力集中在 Python 3 上，这种语言将变得更加精细，并与程序员不断变化的需求相一致。相较而言，对 Python 2.7 的支持将会越来越少。

1.4 Python 的应用

随着大数据、人工智能等技术的迅速发展，Python 作为一门基础语言逐渐受到了人们的追捧。Python 到底能做什么？其实能用到 Python 的地方非常多，从比较前沿的数据挖掘、科学计算、网络爬虫、图像处理、人工智能到传统的 Web 开发、游戏开发，Python 都可以胜任。或许正是因为 Python 如此强大的应

用场景，现在有很多的小伙伴都加入到了学习 Python 的队伍中来。不仅仅是程序员、大学生，连小学的课程中都有了 Python 的影子，也许在不久的将来，Python真的会成为人人都必须要懂的语言。

1.5 Python 的种类

Python 是一门解释器语言，要运行代码，就必须通过解释器执行。Python 有多种解释器，分别基于不同语言开发，每个解释器有不同的特点，但都能正常运行 Python 代码。以下是常用的五种 Python 解释器。

1. CPython

当从 Python 官方网站下载并安装好 Python 后，就直接获得了一个官方版本的解释器：CPython，这个解释器是用 C 语言开发的，所以叫 CPython。在命令行下运行 Python，就是启动 CPython 解释器。CPython 先将源文件(.py)转换成字节码文件(.pyc)，然后在 Python 虚拟机上运行。CPython 是使用最广的 Python 解释器。

2. IPython

IPython 是基于 CPython 的一个交互式解释器，也就是说，IPython 只是在交互方式上有所增强，但是执行 Python 代码的功能和 CPython 是完全一样的，好比很多国产浏览器虽然外观不同，但内核其实是调用了 IE 一样。

3. PyPy

PyPy 是另一个 Python 解释器，它的目标是提高执行速度。PyPy 采用 JIT 技术，对 Python 代码进行动态编译，所以可以显著提高 Python 代码的执行速度。

4. Jython

Jython 是运行在 Java 平台上的 Python 解释器，可以直接把 Python 代码编译成 Java 字节码执行。

5. IronPython

IronPython 和 Jython 类似，只不过 IronPython 是运行在微软 .Net 平台上的 Python 解释器，可以直接把 Python 代码编译成 .Net 的字节码。

在 Python 的解释器中，使用最广泛的是 CPython。对于 Python 的编译，除了可以采用以上解释器进行编译外，技术高超的开发者还可以按照自己的需求自行编写 Python 解释器来执行 Python 代码。

第 2 章

揭开 Python 程序的面纱

2.1 计算机语言

2.1.1 什么是计算机语言

计算机就是一台用来计算的机器，而将人类的一些想法通过计算机来实现需要使用计算机能读懂的语言，这种语言就称为计算机语言(编程语言)。计算机语言其实和人类的语言没有本质的区别，不同点就是交流的主体不同。

2.1.2 计算机语言的发展

计算机语言的发展经历了三个阶段，分别是机器语言、汇编语言和高级语言。

1. 机器语言

机器语言是计算机能够识别并运行的二进制编码，是计算机能读懂的唯一语言。其他所有计算机语言都必须翻译成机器语言才能被计算机执行。机器语言的优点是执行效率高，缺点是编写起来太过复杂。我们可以把机器语言看做是计算机界原始时代交流的语言。

2. 符号(汇编)语言

汇编语言是“大神”级别的程序员经常使用的语言，可以把它看做计算机界古代交流的语言。汇编语言使用符号来代替机器码，编写程序时，不需要使用二

进制码，而是直接编写符号，编写完成后，需要将符号转换为机器码，然后由计算机执行。将符号转换为机器码的过程，称为汇编；将机器码转换为符号的过程，称为反汇编。汇编语言一般只适用于某些硬件，其兼容性比较差，编写难度大，但执行效率比较高。

3. 高级语言

高级语言的语法基本和现在的英语语法类似，你可以理解为已经进入了计算机界的现代社会，并且和硬件的关系没有那么紧密了。也就是说，我们通过高级语言开发的程序可以在不同的硬件系统中执行，而它的执行原理正如前面所述，最终是依靠计算机将高级语言翻译成机器语言来执行的。也正是因为计算机不能直接读懂高级语言，所以相比机器语言和汇编语言来说，高级语言的执行效率要低一点，但编程难度不大，这就是我们为什么要选择高级语言学习的原因。现在我们熟悉的编程语言基本都是高级语言，如 C、C++、C#、Java、JavaScript、Python 等。

2.1.3 计算机语言的分类

计算机只能识别二进制编码(机器码)，所以任何的语言在交由计算机执行时都必须要先转换为机器码，比如，语句“print('hello')”必须要转换为类似“1010101”这样的机器码。根据转换时机的不同，语言分成了两大类：编译型语言和解释型语言。

1. 编译型语言

编译型语言的典型代表是 C 语言。编译型语言会在代码执行前将代码编译为机器码，然后将机器码交由计算机执行，如：

源代码 A→编译→编译后生成可被识别的机器码 B→执行

编译型语言的特点是：

(1) 执行速度特别快；

(2) 跨平台性比较差。

2. 解释型语言

解释型语言的典型代表有 Python、JS、Java 等。解释型语言不会在执行前对代码进行编译，而是一边编译一边执行：

源代码 A→解释器→解释、执行

解释型语言的特点是：

(1) 执行速度比较慢；

(2) 跨平台性比较好。