

大数据创新人才培养系列

# Python

## 程序设计基础教程

慕课版

PYTHON PROGRAMMING  
TUTORIAL

◎ 薛景 陈景强 朱旻如 龚乐君 编著

学习 Python 的第一本书，零基础也能轻松掌握  
在线教学辅助平台提供多功能、一站式、全方位免费教学服务  
二维码扫描，随时观看 MOOC 视频，随时练习课后习题并即时判分  
随书附赠精美 PPT 课件、源代码等丰富资源



中国工信出版集团



人民邮电出版社  
POSTS & TELECOM PRESS

大数据创新人才培养系列

# Python

## 程序设计基础教程

慕课版

PYTHON PROGRAMMING  
TUTORIAL

© 薛景 陈景强 朱旻如 龚乐君 编著

人民邮电出版社

北京

## 图书在版编目 (C I P) 数据

Python程序设计基础教程：慕课版 / 薛景等编著

· 一 北京：人民邮电出版社，2018.11

(大数据创新人才培养系列)

ISBN 978-7-115-48810-7

I. ①P… II. ①薛… 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203141号

## 内 容 提 要

本书是 Python 语言程序设计的入门教程，针对没有程序设计基础的读者。全书共分为 11 章，主要内容包括：Python 语言基础、程序的流程控制、函数与模块、数据结构、异常处理和文件操作、面向对象编程、图形用户界面、数据分析与可视化等，最后通过几个配套实验，全面应用了 Python 语言中几乎所有的知识点，帮助读者通过模仿学会使用 Python 语言进行程序设计。

本书的配套资源包括：在线教学视频、在线教学辅助平台、电子课件和课后习题答案等。通过以上丰富且免费的配套资源，使教与学变得更加方便、简单。

本书可作为高等院校各专业程序设计课程的相关教材，也可作为编程爱好者自学 Python 语言的参考书。

◆ 编 著 薛 景 陈景强 朱旻如 龚乐君

责任编辑 李 召

责任印制 彭志环

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号

邮编 100164 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

固安县铭成印刷有限公司印刷

◆ 开本：787×1092 1/16

印张：11.5

2018 年 11 月第 1 版

字数：298 千字

2018 年 11 月河北第 1 次印刷

定价：39.80 元

读者服务热线：(010)81055256 印装质量热线：(010)81055316

反盗版热线：(010)81055315

Python 语言是目前主流的编程语言之一，具有广泛的应用场景，随着它的迅速普及，国内越来越多的本专科院校正在或准备开设该语言的相关课程。Python 语言功能强大且易于学习，使用它编写的程序可以在 Windows、macOS、Linux、iOS、Android 等软件操作平台上运行。它已经被越来越多的开发者、科研工作者、老师和学生接受。

本书遵循循序渐进的教学规律，从 Python 语言最基础的知识入手，从解决实际问题的需求出发，引伸出各章的内容，非常适合作为零基础的程序入门级教材使用。

全书共分为 11 章，内容覆盖程序设计的四大知识板块：基础知识、流程控制及结构化程序设计、数据的操作、面向对象程序设计。本书的主要特点如下。

第一，本书面向本专科零编程基础的非计算机专业学生。为了突出编程思想的培养，并没有介绍数据库、网络编程等专业性较强的内容，而是选取比较基础的语法、流程控制、数据操作等内容进行介绍，选取的内容具有针对性，且浅显易懂，特别适合编程的入门学习者。

第二，本书作为教材，在充分考虑课时和考试范围的基础上，注重趣味性和娱乐性。全书以一连串有趣的实例将知识点串联起来进行教学，使相对枯燥的编程学习变得有趣和生动，让学生在快乐的编程体验中学会编程。

第三，本书拥有国内领先的配套资源。读者可以结合本书在线学习 MOOC 课程 (<https://www.icourse163.org/>)，还可以免费使用本书编写组开发的在线教学辅助平台 (<https://c.njupt.edu.cn/>)，并下载电子课件 (PPT) 和课后习题答案。通过使用以上配套资源，教与学将变得更加方便、简单。

本书是编写组各位老师多年教学研究和经验的凝练、总结，更是课程组集体智慧的结晶。第 1 章~第 3 章、第 11 章由薛景编写，第 4 章、第 5 章由朱旻如编写，第 6 章由龚乐君编写，第 7 章~第 10 章由陈景强编写，付竟芝、叶水仙参与了课后习题和配套实验的编写。薛景负责最后的统稿工作。本书配套的教学视频由薛景、陈景强、付竟芝、朱旻如、张勤、吴敏共同制作完成。此外，南京邮电大学程序设计课程组的各位老师对本书提出了许多宝贵的建议，在此对他们的辛苦付出和支持表示衷心的感谢！

# 前言

北京：人民邮电出版社，2019。  
（北京：人民邮电出版社，2019。）  
ISBN 978-7-113-25111-1

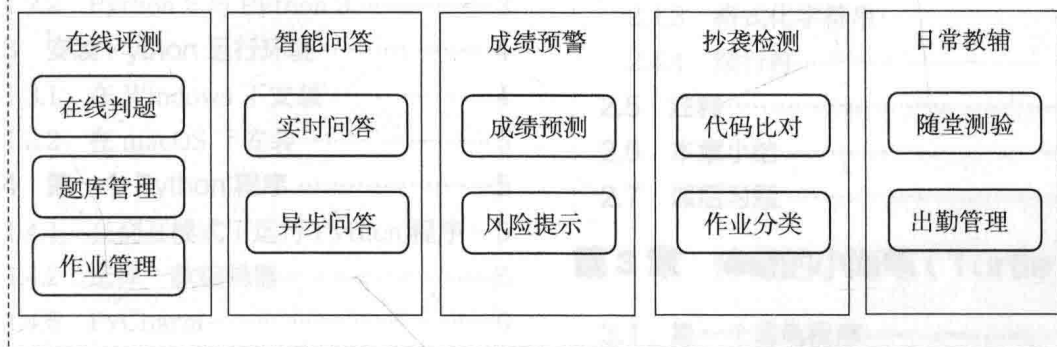
由于编写组水平有限，书中疏漏及不足之处在所难免。如有问题或发现错误，烦请直接与编写组联系，不胜感激！电子邮箱：xuejing@njupt.edu.cn。

本书编写组  
2018年4月

# 在线教学辅助平台使用说明

本书配套建设了全国领先的在线教学辅助平台，为教师教学和学生学习 Python 程序设计课程提供一站式、全方位免费服务，网址：<https://c.njupt.edu.cn/>。

在线教学辅助平台



(1) 本书为教师提供拥有班级管理、出勤管理、题库管理、师生在线交流等强大的功能平台，具体使用手册见本书附录 B。

(2) 本书为读者提供了丰富的配套资源，包括电子课件、源代码、教学视频、在线编程练习，具体使用手册见本书附录 C。

(3) 如果您在使用过程中有任何疑问，请加入 QQ 群：217681442 进行咨询。

注意：所有教学资源 and 平台使用仅限买本书的读者学习使用，不得以其他任何方式传播。

# 目 录 CONTENTS

## 第 1 章 编程前的准备工作 ..... 1

- 1.1 关于编程 ..... 1
- 1.2 关于 Python ..... 2
  - 1.2.1 Python 语言的特点 ..... 2
  - 1.2.2 Python 2 与 Python 3 ..... 3
- 1.3 安装 Python 运行环境 ..... 4
  - 1.3.1 在 Windows 下安装 ..... 4
  - 1.3.2 在 macOS 下安装 ..... 5
- 1.4 第一个 Python 程序 ..... 5
  - 1.4.1 在交互模式下运行 Python 程序 ..... 5
  - 1.4.2 选择一款编辑器 ..... 6
  - 1.4.3 PyCharm ..... 6
  - 1.4.4 以文件模式运行 Python 程序 ..... 10
  - 1.4.5 Python 之禅 ..... 11
- 1.5 本章小结 ..... 12
- 1.6 课后习题 ..... 12

## 第 2 章 Python 语言基础 ..... 14

- 2.1 常量和简单数据类型 ..... 14
  - 2.1.1 数字 ..... 14
  - 2.1.2 True 和 False ..... 15
  - 2.1.3 字符串 ..... 15
  - 2.1.4 数据类型的转换 ..... 16
- 2.2 变量与赋值语句 ..... 17
  - 2.2.1 标识符命名 ..... 17
  - 2.2.2 对象 ..... 17
  - 2.2.3 赋值语句 ..... 17
  - 2.2.4 案例：使用变量与常量 ..... 18
- 2.3 运算符与表达式 ..... 18
  - 2.3.1 运算符 ..... 18
  - 2.3.2 运算符的优先级 ..... 21
  - 2.3.3 改变运算顺序 ..... 22

- 2.3.4 表达式 ..... 22
- 2.3.5 eval() 函数 ..... 22
- 2.4 输入与输出 ..... 23
  - 2.4.1 input() 函数 ..... 23
  - 2.4.2 print() 函数 ..... 23
  - 2.4.3 格式化字符串 ..... 24
  - 2.4.4 续行符 ..... 25
- 2.5 注释 ..... 25
- 2.6 本章小结 ..... 26
- 2.7 课后习题 ..... 26

## 第 3 章 神奇的小海龟 (Turtle) / 29

- 3.1 第一个海龟程序 ..... 29
- 3.2 绘制正多边形 ..... 30
  - 3.2.1 重复、重复、再重复 ..... 30
  - 3.2.2 使用循环化简程序 ..... 30
  - 3.2.3 最重要的格式控制——缩进 ..... 31
- 3.3 绘制美丽的五角星 ..... 31
  - 3.3.1 向左转，向右转 ..... 31
  - 3.3.2 为五角星上色 ..... 32
- 3.4 大星星和小星星 ..... 33
  - 3.4.1 函数的定义与调用 ..... 33
  - 3.4.2 去吧，小海龟 ..... 34
  - 3.4.3 函数的参数 ..... 35
- 3.5 更多关于海龟的函数 ..... 36
- 3.6 本章小结 ..... 37
- 3.7 课后习题 ..... 37

## 第 4 章 程序的流程控制 ..... 40

- 4.1 顺序结构 ..... 40
- 4.2 分支结构 ..... 41
  - 4.2.1 if...else 语句 ..... 41
  - 4.2.2 elif 语句 ..... 42

4.3 循环结构 .....	43	6.2.2 访问列表中的数据 .....	72
4.3.1 while 语句 .....	43	6.2.3 列表赋值 .....	72
4.3.2 for 语句 .....	43	6.2.4 删除列表中的元素 .....	73
4.3.3 嵌套循环 .....	44	6.2.5 列表数据的操作方法 .....	73
4.3.4 循环中的 else 语句 .....	45	6.2.6 常用列表函数 .....	75
4.4 流程中转 break 语句和 continue 语句 .....	45	<b>6.3 字符串 .....</b>	<b>75</b>
4.5 综合案例 .....	47	6.3.1 字符串的表示 .....	75
4.6 本章小结 .....	49	6.3.2 字符串的截取 .....	75
4.7 课后习题 .....	49	6.3.3 连接字符串 .....	75
<b>第 5 章 函数与模块 .....</b>	<b>54</b>	6.3.4 格式化字符串 .....	76
5.1 函数的定义与调用 .....	54	6.3.5 字符串的操作方法 .....	77
5.1.1 文档字符串 .....	55	6.3.6 其他操作 .....	77
5.1.2 函数调用 .....	55	<b>6.4 字典 .....</b>	<b>78</b>
5.1.3 函数的返回值 .....	56	6.4.1 字典的创建 .....	78
5.1.4 匿名函数 .....	57	6.4.2 访问字典中的数据 .....	78
5.2 函数的参数传递 .....	57	6.4.3 修改字典中的数据 .....	78
5.2.1 默认参数与关键字参数 .....	58	6.4.4 字典的操作方法 .....	79
5.2.2 不定长参数 .....	59	6.4.5 常用的字典函数 .....	80
5.3 变量的作用域 .....	59	6.4.6 嵌套字典 .....	81
5.4 函数的递归 .....	61	<b>6.5 集合 .....</b>	<b>81</b>
5.5 模块化程序设计 .....	62	6.5.1 创建集合 .....	81
5.5.1 模块及其引用 .....	63	6.5.2 集合数据的添加与删除 .....	81
5.5.2 包 .....	65	6.5.3 集合的数学运算 .....	81
5.6 内置函数 .....	66	<b>6.6 本章小结 .....</b>	<b>82</b>
5.7 本章小结 .....	67	<b>6.7 课后习题 .....</b>	<b>83</b>
5.8 课后习题 .....	68	<b>第 7 章 异常处理和文件操作 ...</b>	<b>85</b>
<b>第 6 章 数据结构 .....</b>	<b>70</b>	7.1 异常处理 .....	85
6.1 元组 .....	70	7.1.1 try...except 语句 .....	86
6.1.1 创建元组 .....	70	7.1.2 finally 语句 .....	87
6.1.2 访问元组中的数据 .....	71	7.2 断言 .....	87
6.1.3 元组的连接 .....	71	7.3 文件操作 .....	88
6.1.4 删除元组 .....	71	7.3.1 写文件操作 .....	89
6.1.5 常用元组函数 .....	72	7.3.2 读文件操作 .....	90
6.2 列表 .....	72	7.3.3 with 语句 .....	91
6.2.1 创建列表 .....	72	7.4 本章小结 .....	92
		7.5 课后习题 .....	92



**第 8 章 面向对象编程 ..... 94****8.1 类和对象 ..... 94**

8.1.1 Person 类的定义与实例化 ..... 95

8.1.2 Person 类的完整定义 ..... 96

8.1.3 对象属性的默认值设置 ..... 97

8.1.4 对象属性的添加、修改和删除 ..... 98

8.1.5 私有属性和私有方法 ..... 98

8.1.6 类属性 ..... 99

**8.2 类的继承 ..... 100**

8.2.1 一个简单的继承例子 ..... 100

8.2.2 子类方法对父类方法的覆盖 ..... 101

8.2.3 在子类方法中调用父类的  
同名方法 ..... 102**8.3 本章小结 ..... 103****8.4 课后习题 ..... 103****第 9 章 图形用户界面 ..... 105****9.1 Tkinter 简介 ..... 105**

9.1.1 第一个 Tkinter 窗口 ..... 105

9.1.2 在窗口中加入组件 ..... 106

9.1.3 为按钮设置动作事件 ..... 107

9.1.4 坐标管理器 ..... 108

**9.2 Tkinter 组件及其属性 ..... 109**

9.2.1 Label 组件和 Entry 组件 ..... 110

9.2.2 Listbox 组件 ..... 111

9.2.3 Canvas 组件 ..... 112

**9.3 案例分析：简单计算器 ..... 113**

9.3.1 实现计算器界面 ..... 114

9.3.2 实现数字按钮的点击功能 ..... 115

9.3.3 实现小数点按钮的功能 ..... 115

9.3.4 实现运算按钮的功能 ..... 116

**9.4 本章小结 ..... 117****9.5 课后习题 ..... 117****第 10 章 数据分析与可视化 ... 119****10.1 数值计算库 numpy ..... 119**

10.1.1 创建 numpy 数组 ..... 119

10.1.2 数组与数值的算术运算 ..... 120

10.1.3 数组与数组的算术运算 ..... 121

10.1.4 数组的关系运算 ..... 121

10.1.5 分段函数 ..... 122

10.1.6 数组元素访问 ..... 122

10.1.7 数组切片操作 ..... 123

10.1.8 改变数组形状 ..... 123

10.1.9 二维数组转置 ..... 124

10.1.10 向量内积 ..... 124

10.1.11 数组的函数运算 ..... 125

10.1.12 对数组的不同维度元素进行  
计算 ..... 125

10.1.13 广播 ..... 126

10.1.14 计算数组中元素的出现  
次数 ..... 127

10.1.15 矩阵运算 ..... 127

**10.2 科学计算扩展库 scipy ..... 128**

10.2.1 常数模块 constants ..... 128

10.2.2 特殊函数模块 special ..... 129

10.2.3 多项式计算与符号计算 ..... 129

**10.3 数值计算可视化库 matplotlib ..... 131**

10.3.1 绘制正弦曲线 ..... 131

10.3.2 绘制散点图 ..... 132

10.3.3 绘制饼图 ..... 134

10.3.4 绘制带有中文标签和图例  
的图 ..... 135

10.3.5 绘制带有公式的图 ..... 135

10.3.6 绘制三维参数曲线 ..... 136

10.3.7 绘制三维图形 ..... 137

**10.4 本章小结 ..... 139****10.5 课后习题 ..... 139****第 11 章 学生成绩管理系统的  
设计与实现 ..... 141**

11.1 系统概述 ..... 141

11.2 数据类型的定义 ..... 142

11.3 为学生类型定制的基本操作 ..... 143

11.4 用文本文件实现数据的永久保存 ..... 146

11.5 用两级菜单四层函数实现系统 ..... 148

11.6 课后习题 ..... 152

**附录 A 配套实验 ..... 153**

实验一 使用 Turtle 库绘制七巧板 ..... 153

实验二 程序的流程控制 ..... 155

实验三 函数的定义和调用 ..... 156

实验四 数据结构及文件读写应用 ..... 159

实验五 GUI 程序设计 ..... 162

**附录 B 在线教学辅助平台教师使用手册 ..... 164**

**附录 C 配套电子资源使用手册 ..... 169**

# 第1章 编程前的准备工作

## 学习目标

- 了解 Python 语言的特点。
- 掌握安装 Python 3.x 运行环境的方法。
- 掌握在交互模式下运行 Python 语句的方法。
- 掌握建立、保存、打开、编辑以及运行 Python 程序

文件的方法。



本章电子课件

程序设计（编程）是一项非常热门的计算机应用能力，Python 语言使学习这项新技能变得非常容易，在深入学习编程之前，你需要先了解它的基础知识，并为今后的学习做准备。本章将讨论如何搭建 Python 语言的开发环境，以及如何编辑和运行一个简单的 Python 程序。

## 1.1 关于编程

众所周知，计算机在人们的工作和生活中发挥了巨大的作用，它可以帮助人们完成非常复杂的计算工作，处理海量的数据，分析数据之间的关系，最后还能以图形化的方式将处理结果展现在人们的面前。那么，计算机是如何完成这些工作的呢？

其实，计算机并不是天生就具备这些超强的能力，它只不过是按照人们预先设置好的程序（**Program**）一步一步地完成自己的工作，而程序就是一组告诉计算机应该如何正确工作的指令集合。因为计算机的计算速度特别快，所以使用计算机可以大大提高人们的工作效率。

简单地说，程序设计也就是编程（**Programming**），是让计算机按照程序员（**Programmer**）给出的指令去做一些它能够胜任的工作，如解一个方程、绘制一幅图像、获取一张网页上的数据等。如果你直接对着计算机说中文，计算机是不能理解你所说的内容的，所以你需要使用计算机能够理解的语言和它交流。计算机能够理解的语言，称为程序设计语言（**Programming**）。



关于编程

**Language**），本书的核心内容就是教你使用一门程序设计语言——Python，使用这门语言，计算机就可以帮助你把工作做得更快、更好。

## 1.2 关于 Python

Python 是一种极少数能兼具简单与功能强大两个特点的编程语言。你将惊异地发现这门编程语言是如此简单，它专注于如何解决问题，而非拘泥于语法与结构。

Python 的官方网站是这样描述这门语言的。

Python 是一款易于学习且功能强大的开放源代码的编程语言。它可以快速帮助人们完成各种编程任务，并且能够把用其他语言制作的各种模块很轻松地联结在一起。使用 Python 编写的程序可以在绝大多数平台上顺利运行。

### 1.2.1 Python 语言的特点

选择 Python 语言作为程序设计的入门语言，其主要原因是相比于其他计算机编程语言，它具有以下特点。

(1) **简单 (Simple)**。Python 是一门语法简单且风格简约的语言。阅读一份优秀的 Python 代码就如同在阅读英语文章一样，尽管这门英语也会有严格的语法格式！Python 这种接近自然语言的书写特质正是它的一大优势，能够让你专注于解决问题的方案，而不是语言本身。

(2) **易于学习 (Easy to Learn)**。正如你即将看到的，Python 是一门非常容易入门的语言，它有一套简单的语法体系，这大大降低了学习计算机编程的门槛。

(3) **自由且开放 (Free and Open Source)**。Python 是 FLOSS（自由/开放源代码软件）的成员之一。简单来说，可以自由地分发这一软件的拷贝，阅读它的源代码，并对其做出改动，或是将它的一部分运用于一款新的自由程序中。FLOSS 基于一个可以分享知识的社区理念而创建。这正是 Python 能如此优秀的一大原因——它由一群希望看到 Python 能变得更好的社区成员创造，并持续改进至今。

(4) **高级语言 (High-level Language)**。就像其他的计算机高级语言一样，在用 Python 编写程序时，你不必考虑诸如程序应当如何使用 CPU 或者内存等具体实现细节。

(5) **跨平台性 (Portable)**。由于其开放源码的特性，Python 已经被移植到其他诸多软件操作平台（如 Windows、mac OS、Linux、iOS、Android 等）中。如果小心地避开了所有系统依赖型的特性，不必做出任何改动，所有的 Python 程序就可以在其中任何一个平台上工作。

(6) **解释执行 (Interpreted)**。使用诸如 C 或 C++ 等编译执行类语言编写程序时，需要将这些语言的源代码通过编译程序（Compiler）转换成计算机使用的语言（如由 0 与 1 构成的二进制码），当运行这些程序时，链接程序或载入程序将会从硬盘中将程序复制到内存中并将其运行。

然而，作为解释执行类的 Python 语言，不需要将其编译成二进制码，只需要直接从源代

码运行该程序。在程序内部，Python 会将源代码转换为字节码（**Bytecodes**）的中间形式，然后再转换成计算机使用的语言，并运行它。实际上，这一流程使得 Python 更加易于使用，你不必再担心该如何编译程序，或如何保证适当的库被正确地链接并加载等步骤。这种运行方式使得 Python 程序更加易于迁移，只需要将 Python 程序复制到另一台计算机便可让它立即开始工作！

(7) 面向对象 (**Object Oriented**)。Python 同时支持面向过程编程与面向对象编程。在面向过程 (**Procedure-oriented**) 的编程语言中，程序仅仅是由带有可重用特性的子程序与函数构建起来的。在面向对象的编程语言中，程序是由结合了数据与功能的对象构建起来的。与 C++ 或 Java 这些大型的面向对象语言相比，Python 用特别的、功能强大又简单的方式来实现面向对象编程。

(8) 可扩展性 (**Extensible**)。如果你需要代码的某一重要部分能够快速地运行，或希望算法的某些部分不被公开，可以在 C 或 C++ 语言中编写这些程序，然后再将其运用于 Python 程序中，Python 可以完美地与这些使用其他语言编写的程序一起工作。

(9) 可嵌入性 (**Embeddable**)。可以在 C 或 C++ 程序中嵌入 Python 程序，从而向程序用户提供编写脚本 (**Scripting**) 的功能。

(10) 丰富的库 (**Extensive Libraries**)。实际上 Python 标准库的规模非常庞大。它能够帮助用户完成诸多事情，包括正则表达式、文档生成、单元测试、多线程、数据库、网页浏览器、CGI、FTP、邮件、XML、XML-RPC、HTML、WAV 文件、密码系统、GUI (图形用户界面)，以及其他系统依赖型的活动。只需记住，只要安装了 Python，这些功能便随时可用。

除了标准库以外，你还可以在 Python 库索引 (<https://pypi.python.org/pypi>) 中发掘许多其他高质量的库。

Python 实在是一门令人心生激动且功能强大的语言。它恰当地结合了性能与功能，使得编写 Python 程序是如此简易又充满了乐趣。

## 1.2.2 Python 2 与 Python 3

如果你对 Python 2 与 Python 3 之间的区别不感兴趣，那么可以略过本小节。但务必注意你正在使用的版本，因为不同版本的 Python 在程序语法上并不兼容，即遵循 Python 2 语法书写的源程序也无法顺利地在 Python 3 的运行环境中运行，反之亦然。本书是以 Python 3 为默认运行环境撰写的。

只要正确理解并学习了其中一个版本的 Python，你就可以很容易地理解与另一版本的区别，并能快速学习如何使用。在学习过程中，真正困难的是学习如何编程以及理解 Python 语言的基础部分。这便是本书希望讨论的关键目标，而一旦你达成了该目标，便可以根据实际情况，决定是使用 Python 2 还是 Python 3。

要想了解有关 Python 2 和 Python 3 之间区别的更多细节，可自行在网上查询学习。

## 1.3 安装 Python 运行环境

### 1.3.1 在 Windows 下安装

访问 Python 的官方网站并下载最新版本的 Python 安装程序，请根据 Windows 版本（32 位或者 64 位）选择对应的安装文件，如图 1-1 所示。其安装过程与 Windows 平台其他软件的安装过程并没有太大的差别。



在 Windows 下安装

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		2d0fc9f3a5940707590e07f03ecb08b9	22540566	SIG
XZ compressed source tarball	Source release		692b4fc3a2ba0d54d1495d4ead5b0b5c	16872064	SIG
Mac OS X 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	6dd08e7027d2a1b3a2c02cfacbe611ef	27511848	SIG
Windows help file	Windows		69082441d723060fb333cda8815105e	7986690	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64, not Itanium processors	708496ebbe9a730d19d5d288afd216f1	6926999	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64, not Itanium processors	ad69fdacde90f2ce8286c279b11ca188	31392272	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64, not Itanium processors	a055a1a0e938e74c712a1c495261ae6c	1312520	SIG
Windows x86 embeddable zip file	Windows		8dff09a1b19b7a7dcb915765328484cf	6320763	SIG
Windows x86 executable installer	Windows		3773db079c173bd6d8a631896c72a88f	30453192	SIG
Windows x86 web-based installer	Windows		f58f019335f39e0b45a0ae68027888d7	1287064	SIG

图 1-1 根据 Windows 版本选择对应的安装程序进行下载

注意：请务必在安装界面中确认勾选了 Add Python 3.7 to PATH 选项，如图 1-2 所示。

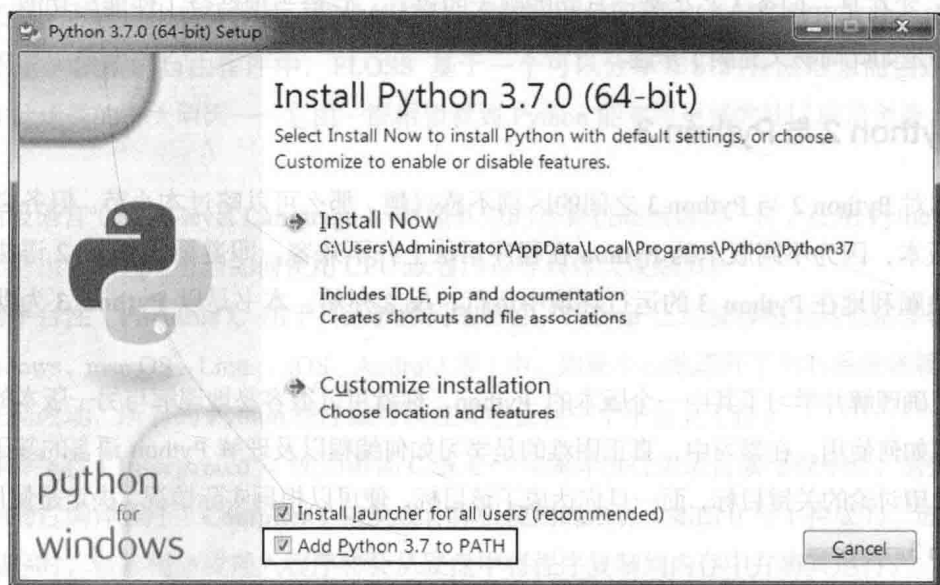


图 1-2 安装时请勾选 Add Python 3.7 to PATH 选项

## 1.3.2 在 macOS 下安装

对于 macOS 用户，既可以使用 Homebrew 并通过命令 `brew install python 3` 进行安装，也可以在官方网站下载对应的 macOS 版本安装程序进行安装，如图 1-3 所示。

Version	Operating System	Description	MD5 Sum	File Size	PGP
Gzipped source tarball	Source release		2d0fc9f3a5940707590e07f03ecb08b9	22540566	SIG
XZ compressed source tarball	Source release		692b4fc3a2ba0d54d1495d4ead5b0b5c	16872064	SIG
Mac OS X 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	6dd08e7027d2a1b3a2c02cfacbe611ef	27511848	SIG
Windows help file	Windows		69082441d723060fb333dca8815105e	7986690	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64, not Itanium processors	708496bbe9a730d19d5d288afd216f1	6926999	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64, not Itanium processors	ad69fdacde90f2ce8286c279b11ca188	31392272	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64, not Itanium processors	a055a1a0e938e74c712a1c495261ae6c	1312520	SIG
Windows x86 embeddable zip file	Windows		8dff09a1b19b7a7dcb915765328484cf	6320763	SIG
Windows x86 executable installer	Windows		3773db079c173bd6d8a631896c72a88f	30453192	SIG
Windows x86 web-based installer	Windows		f58f019335f39e0b45a0ae68027888d7	1287064	SIG

图 1-3 下载对应的 macOS 版本安装程序安装 Python 3

验证安装是否成功，可以按 `Command+Space` 组合键（以启动 Spotlight 搜索），输入 `Terminal` 并按下 `Enter` 键启动终端程序。现在，试着运行 `Python 3` 来确保其没有任何错误。

## 1.4 第一个 Python 程序

下面将介绍如何在 Python 中运行一个传统的“Hello World”程序，包括如何编写、保存与运行 Python 程序。

通过 Python 运行程序有两种方法：使用交互式解释器提示符和直接运行一个源代码文件。下面介绍如何使用这两种方法。

### 1.4.1 在交互模式下运行 Python 程序

在操作系统中打开终端（Terminal）程序（在 Windows 操作系统中被称为命令提示符），然后输入 `python 3` 并按 `Enter` 键来打开 Python 提示符工具（Python Prompt）。

启动 Python 后，会出现“`>>>`”。这个被称作 Python 解释器提示符（Python Interpreter Prompt）。

在 Python 解释器提示符后输入以下语句。

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

在输入完成后按 `Enter`（回车）键，会看到屏幕上打印出 `Hello World!` 字样。

用户会注意到，Python 立即输出了一行结果！刚才输入的便是一句独立的 Python 语句。使



在交互模式下运行 Python 程序

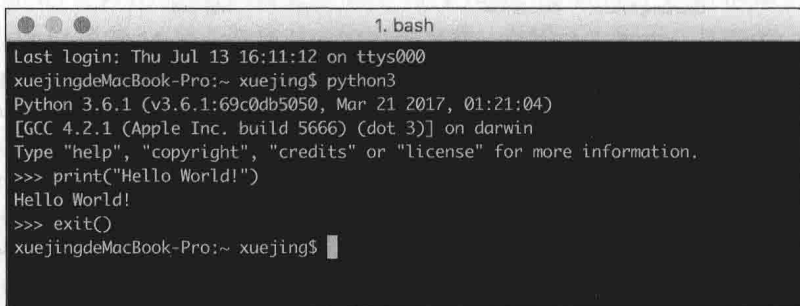
用 `print` 命令可以打印用户提供的信息。这里提供了文本 `Hello World!`，然后它便被迅速打印到屏幕上。

如果要退出提示符，只需在解释器提示符后输入：

```
exit()
```

注意：`exit` 后要包含一对括号`()`，并按 `Enter` 键来退出解释器提示符。

在 `macOS` 的终端程序中，上述操作的执行过程如图 1-4 所示。

A terminal window titled "1. bash" showing the execution of Python. The prompt is "xuejingdeMacBook-Pro:~ xuejing\$". The user enters "python3", which outputs "Python 3.6.1 (v3.6.1:69c0db5050, Mar 21 2017, 01:21:04) [GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin". The user then enters ">>> print('Hello World!')", which outputs "Hello World!". Finally, the user enters ">>> exit()", which outputs "xuejingdeMacBook-Pro:~ xuejing\$".

```
1. bash
Last login: Thu Jul 13 16:11:12 on ttys000
xuejingdeMacBook-Pro:~ xuejing$ python3
Python 3.6.1 (v3.6.1:69c0db5050, Mar 21 2017, 01:21:04)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello World!")
Hello World!
>>> exit()
xuejingdeMacBook-Pro:~ xuejing$
```

图 1-4 在 Python 解释器提示符中执行 `print("Hello World!")`

## 1.4.2 选择一款编辑器

当我们希望运行某些程序时，总不能每次都在解释器提示符中输入希望运行的程序。因此需要将它们保存为文件，以后便可以多次运行这些程序。

创建 Python 源代码文件，需要一款能够提供输入并存储代码的编辑器软件。一款优秀的编辑器能够使编写源代码的工作变得轻松许多。故而选择一款编辑器至关重要。要像挑选想要购买的汽车一样挑选编辑器。一款优秀的编辑器能够帮助用户更轻松地编写 Python 程序，使用户的编程之旅更加舒适，并帮助找到一条更加安全、快速的道路到达目的地。

对编辑器的最基本要求为语法高亮，这一功能通过标以不同的颜色来帮助区分 Python 程序中的不同部分，从而更好地阅读程序，并使它的运行模式更加形象化。

如果你对应该从哪里开始还没有概念，本书推荐使用 **PyCharm** 教育版软件，它在 `Windows`、`macOS X`、`GNU/Linux` 上都可以运行。在下一节你将了解到更多信息。

如果你使用的是 `Windows` 系统，不要用记事本——这是一个很糟糕的选择，因为它没有语法高亮功能，而且不支持文本缩进功能，之后你将会慢慢了解这一功能究竟有多重要。而一款好的编辑器能够自动帮助你完成这一工作。

如果你已是一名经验丰富的程序员，那一定在用 `Sublimes Text` 或 `Visual Studio Code` 了。无需多言，它们都是最强大的编辑器之一，用它们来编写 Python 程序自是受益颇多。

再此重申，你可以选择任意一款合适的编辑器——它能够让编写 Python 程序变得更加有趣且容易。同时，此刻你更应该专注于学习 Python 而不是编辑器的使用方法。

## 1.4.3 PyCharm

`PyCharm` 教育版是一款有助于编写 Python 程序的免费编辑器，读者可以在官网进行下载。

(1) 打开 `PyCharm` 时，会看见如图 1-5 所示的界面，单击 `Create New Project`。



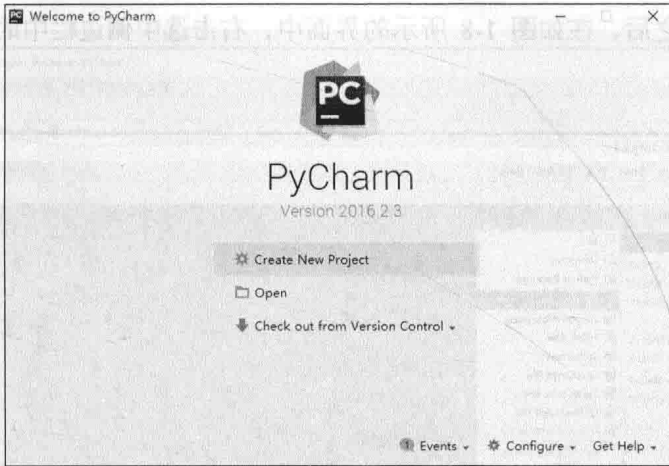


图 1-5 PyCharm 欢迎界面

在如图 1-6 所示的界面中，选择 Pure Python。

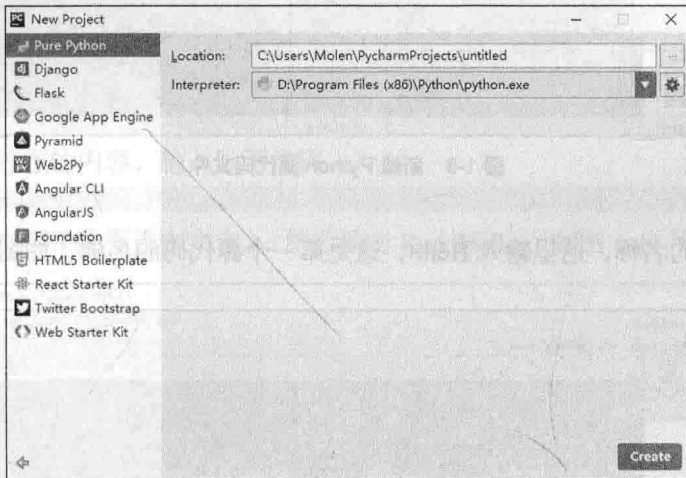


图 1-6 选择 Pure Python 项目

(2) 将项目路径位置中的 `untitled` 更改为 `helloworld`，如图 1-7 所示。

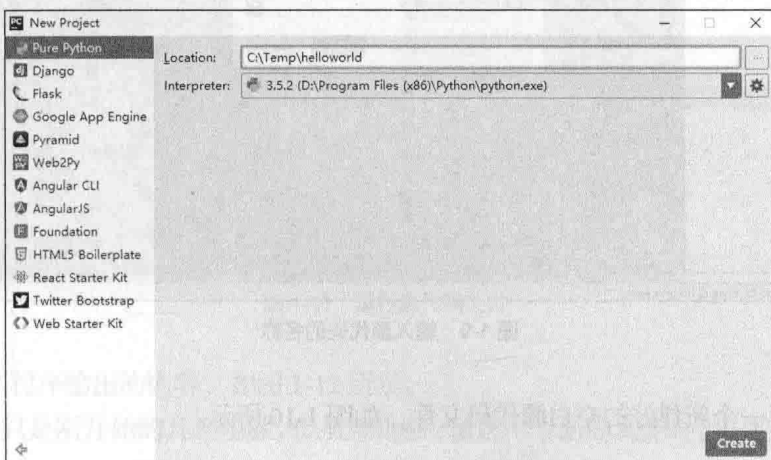


图 1-7 更改项目的保存路径