



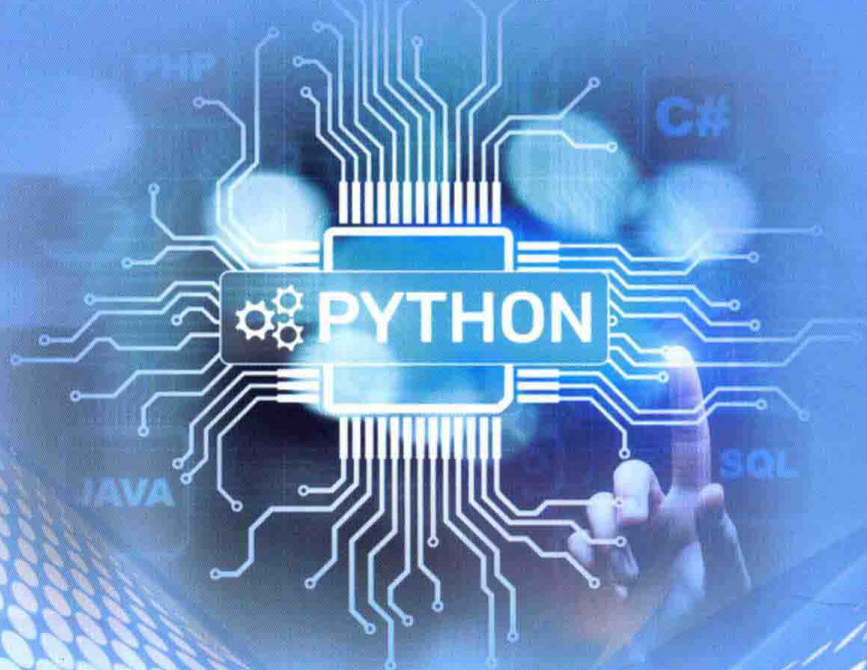
普通高等教育人工智能与大数据系列教材

Python程序

设计基础

Python Programming Basics

娄岩 主编



机械工业出版社
CHINA MACHINE PRESS

普通高等教育人工智能与大数据系列教材

Python 程序设计基础

主 编 娄 岩

副主编 胡仕坤 袁 磊 徐东雨

参 编 陈继超 靳瑞霞 李超科 朱云峰

王 维 熊 玮 徐 强 郭晨荣

王 魏

机械工业出版社

本书以实用为导向,从基础概念、语法结构、应用案例、开发技巧等方面深入浅出地介绍了 Python 的程序设计方法以及项目实施流程。

本书以实例方式进行讲解,每个实例程序都通过了验证。本书内容图文并茂,操作步骤完善,并附有具体脚本代码,易于掌握和学习。本书提供开放式的课程网站(<http://www.cmu.edu.cn/computer>)和相应的课件作为支持。

本书既可以作为全国普通高校各专业计算机公共基础课程的教材,又可以作为全国计算机等级考试的辅导教材,还可以作为专业技术人员的参考用书。

图书在版编目(CIP)数据

Python 程序设计基础/娄岩主编. —北京:机械工业出版社,2020.12
普通高等教育人工智能与大数据系列教材
ISBN 978-7-111-67048-3

I. ①P… II. ①娄… III. ①软件工具—程序设计—高等学校—教材
IV. ①TP311.561

中国版本图书馆 CIP 数据核字(2020)第 248163 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

策划编辑:刘琴琴 责任编辑:刘琴琴 侯颖

责任校对:王欣 封面设计:张静

责任印制:常天培

北京机工印刷厂印刷

2021 年 1 月第 1 版第 1 次印刷

184mm×260mm·15.75 印张·378 千字

标准书号:ISBN 978-7-111-67048-3

定价:43.00 元

电话服务

客服电话:010-88361066

010-88379833

010-68326294

封底无防伪标均为盗版

网络服务

机工官网:www.cmpbook.com

机工官博:weibo.com/cmp1952

金书网:www.golden-book.com

机工教育服务网:www.cmpedu.com

前 言

近年来，Python 发展迅速，已成为热门的程序设计语言之一，在科研、电子、数据分析、Web、金融、图像处理、人工智能等方面得到了广泛的应用。现在有越来越多的人投入到 Python 的应用与研究中。为适应新时代信息技术的发展，教育部考试中心决定自 2018 年 3 月起，在计算机二级考试中加入“Python 语言程序设计”科目。

为此，我们精心策划和编写了《Python 程序设计基础》这本面向实践、注重应用的教材，目的是使读者既能够结合实例学习 Python 程序设计方法，提高运用 Python3 编程并解决实际问题能力，又可以通过学习，为参加全国计算机等级考试做好准备。

本书以实例方式进行讲解，每个实例程序都通过了验证，操作步骤完善，并附有具体的脚本代码。本书图文并茂，知识难易程度循序渐进，具有很强的可读性和可操作性。读者在学习过程中可以按图索骥，在较短的时间内掌握使用 Python 进行开发的技巧。本书兼顾了不同专业、不同层次读者的需要，以提高读者自主学习和运用知识的能力为目标，强化了学习过程中实践能力的培养，为 Python 初学者提供全面与翔实的参考资料，使其易于掌握 Python 程序设计方法、项目开发流程和步骤。

在此向所有对本书的编写给予帮助和指导的朋友们表示衷心的感谢！由于编者水平有限，书中难免存在疏漏之处，恳请广大读者批评指正。

娄 岩

目 录

前 言

第 1 章 Python 概述	1
导学	1
1.1 Python 简介	1
1.1.1 Python 的概念	1
1.1.2 Python 的发展历史	1
1.1.3 Python3	2
1.1.4 Python 的特点	2
1.1.5 Python 的应用领域	3
1.2 Python 的安装	3
1.3 第一个 Python 程序	6
1.4 搭建 Python 集成开发环境	7
1.5 Python 的输入/输出	15
1.5.1 Python 的输入	15
1.5.2 Python 的输出	16
本章小结	16
第 2 章 基础语法	17
导学	17
2.1 Python 变量的赋值	17
2.2 Python 的标准数据类型	18
2.2.1 数字	18
2.2.2 字符串	18
2.2.3 列表	21
2.2.4 元组	23
2.2.5 字典	24
2.2.6 集合	25
2.3 Python 的基本运算符	26
2.3.1 算术运算符	26
2.3.2 字符串运算符	27
2.3.3 比较运算符	27

2.3.4	赋值运算符	28
2.3.5	逻辑运算符	29
2.3.6	成员运算符	29
2.3.7	常用运算符的优先级	30
	本章小结	30
第3章	程序的控制结构	31
	导学	31
3.1	顺序结构	31
3.2	分支结构	32
3.2.1	if 语句(单分支结构)	32
3.2.2	if···else 语句(双分支结构)	33
3.2.3	if···elif···else 语句(多分支结构)	35
3.2.4	pass 语句	37
3.2.5	try···except 语句	37
3.3	循环结构	38
3.3.1	while 语句(while 循环)	38
3.3.2	for 语句(for 循环)	39
3.3.3	循环嵌套	40
3.3.4	break 语句和 continue 语句	42
	本章小结	43
第4章	函数、模块和文件	44
	导学	44
4.1	函数	44
4.1.1	系统函数	44
4.1.2	函数的定义	53
4.1.3	函数的参数和返回值	54
4.1.4	变量的作用域	55
4.1.5	递归调用	55
4.2	模块	56
4.3	文件	57
4.3.1	文件的打开和关闭	57
4.3.2	文件的读操作	58
4.3.3	文件的写操作	60
4.3.4	文件的指针定位	62
	本章小结	62
第5章	面向对象程序设计	63
	导学	63

5.1 面向对象程序设计基础	63
5.1.1 面向对象程序设计的基本概念	63
5.1.2 面向对象程序设计的三个基本特性	64
5.2 定义和访问类与对象	65
5.2.1 定义类和对象	65
5.2.2 利用构造函数定义类	66
5.2.3 定义和访问实例属性与类属性	67
5.3 类的继承与多态	68
5.3.1 类的继承	68
5.3.2 类的多继承	70
5.3.3 类的多态	71
本章小结	72
第6章 图形界面设计	73
导学	73
6.1 Python 图形开发库	73
6.1.1 开发平台	73
6.1.2 创建 Windows 窗口	74
6.2 Tkinter 常用组件	75
6.2.1 Tkinter 组件	75
6.2.2 标签	78
6.2.3 按钮	79
6.2.4 文本框	81
6.2.5 列表框	83
6.2.6 单选按钮和复选框	85
6.2.7 消息窗口	88
6.2.8 对话框	89
6.2.9 菜单	90
6.3 窗口内图形绘制	92
6.3.1 Canvas 画布组件	92
6.3.2 绘制图形对象	93
6.4 Python 事件处理	95
6.4.1 事件类型	95
6.4.2 事件处理函数	97
6.4.3 事件绑定	97
本章小结	99
第7章 数据库应用	100
导学	100

7.1 结构化查询语言	100
7.1.1 SQL 基本语句	100
7.1.2 SQL 查询语句	101
7.2 SQLite3 数据库基础	103
7.3 Python 的 SQLite3 数据库编程	104
7.3.1 访问数据库的方法	104
7.3.2 数据库应用实例	106
7.4 Python 访问主流数据库与存储文本数据	114
7.4.1 Python 访问主流数据库	114
7.4.2 Python 存储文本数据	117
本章小结	118
第 8 章 网络与多线程	119
导学	119
8.1 网络编程理论基础	119
8.1.1 软件体系结构	119
8.1.2 IP	120
8.1.3 端口	121
8.1.4 HTTP	121
8.1.5 TCP 和 UDP	122
8.1.6 socket	122
8.2 TCP 编程实现	126
8.2.1 利用 TCP 客户端获取网页内容	126
8.2.2 TCP 服务器端编程	129
8.3 UDP 编程实现	132
8.3.1 UDP 单客户端编程	132
8.3.2 UDP 多客户端编程	133
8.4 多线程编程	135
8.4.1 进程和线程的原理	135
8.4.2 创建线程的两种方法	136
8.4.3 线程同步机制	141
8.5 网络与多线程编程综合应用实例	143
本章小结	146
第 9 章 Python 网络爬虫	147
导学	147
9.1 网络爬虫原理	147
9.2 requests 爬取库	148
9.2.1 requests 爬取库的安装	148

9.2.2	requests 爬取库的使用方法	149
9.2.3	访问异常处理	154
9.2.4	requests 爬取库的应用	154
9.3	BeautifulSoup 解析库	158
9.3.1	BeautifulSoup 解析库的安装	159
9.3.2	BeautifulSoup 解析库的使用方法	159
9.3.3	BeautifulSoup 解析库的应用	167
	本章小结	180
第 10 章	图像操作与处理	181
	导学	181
10.1	Pillow 的安装	181
10.2	PIL 的基本概念	182
10.3	PIL 的常用模块	184
10.3.1	Image 模块	185
10.3.2	ImageChops 模块	185
10.3.3	ImageDraw 模块	188
10.3.4	ImageEnhance 模块	190
10.3.5	ImageFilter 模块	193
10.3.6	ImageFont 模块	195
10.4	PIL 对图像的基本操作	195
10.4.1	图像格式的转换	196
10.4.2	创建缩略图	197
10.4.3	图像的复制和粘贴	198
10.4.4	几何变换	198
10.5	PIL 对图像处理的综合实例	199
	本章小结	202
第 11 章	科学计算与数据可视化	203
	导学	203
11.1	科学计算	203
11.1.1	NumPy 数组	203
11.1.2	NumPy 数组的算术运算操作	206
11.1.3	NumPy 数组的形状操作	207
11.1.4	NumPy 的图像数组操作	208
11.2	数据可视化	211
11.2.1	使用 pyplot 模块的绘图方法	212
11.2.2	使用 pyplot 模块绘制直方图、条形图、散点图及饼状图等	217
	本章小结	224

第 12 章 机器学习	225
导学	225
12.1 机器学习的概念与操作流程	225
12.1.1 机器学习的概念	225
12.1.2 机器学习的操作流程	225
12.1.3 机器学习库 sklearn 的安装	226
12.2 算法与应用	226
12.2.1 线性回归	226
12.2.2 K-means 聚类	228
12.2.3 逻辑回归算法	229
本章小结	231
第 13 章 Python 与 XML	232
导学	232
13.1 XML 概述	232
13.2 Python 使用 SAX 解析 XML	233
13.2.1 SAX 简介	233
13.2.2 使用 SAX 解析 XML 文件的方法	233
13.2.3 使用 SAX 解析 XML 文件的实例	234
13.3 Python 使用 DOM 解析 XML	235
13.3.1 DOM 简介	235
13.3.2 使用 DOM 解析 XML 文件	236
13.3.3 使用 DOM 添加节点	237
13.3.4 使用 DOM 修改、删除节点	239
本章小结	241
参考文献	242

导 学

Python 是一种跨平台、开源的解释型高级语言。其作为动态编程语言，更适合初学者。通过引用外部库，Python 可快速、准确地实现多种实用功能，尤其在人工智能和大数据方面的应用，更优于目前市面上流行的其他高级程序语言。Python 模块化的设计理念使其具有更好的开放性和扩展性。通过学习 Python，不但能够帮助初学者掌握程序设计方法，增强逻辑思维能力，还可使其快速地开发出具有实用功能的应用程序。

了解：Python 的发展历史和主要应用领域。

掌握：Python 的相关概念以及其特点，Python 以及其集成开发环境的搭建、工作方式及安装方法，Python 输入和输出的主要方法。

Python 作为一种高扩展性的语言，有功能丰富的标准库，应用领域十分广泛，几乎可以在任何场合应用 Python，如网站、爬虫及机器人控制，是目前主流的程序设计语言之一。

1.1 Python 简介

1.1.1 Python 的概念

Python 是一种高层次的，结合了解释性、编译性、互动性和面向对象的程序设计语言。Python 的设计具有很强的可读性，具有比其他语言更有特色的语法结构。

1) Python 是一种解释型语言。它开发过程中没有了编译这个环节，类似于 PHP 或者 Perl 语言。

2) Python 是一种交互式语言。它可以在命令提示符中，直接互动执行程序。

3) Python 是一种面向对象语言。Python 支持面向对象的或代码封装在对象中的编程技术。

4) Python 是一种被广泛应用的语言。Python 支持广泛的应用程序开发，从简单的文字处理到浏览器，再到网络爬虫、机器学习等。

1.1.2 Python 的发展历史

Python 是 Guido van Rossum 于 1989 年圣诞节在荷兰国家数学与计算机科学研究中心。

开发而成的，Python 这个名字来自于他最钟爱的电视剧 *Monty Python's Flying Circus*。Python 本身也是由诸多其他语言发展而来的，包括 C、C++、Unix shell 和其他的脚本语言等。

自 2004 年以后，Python 的使用率呈线性增长。2019 年 1 月，Python 被 TIOBE 编程语言排行榜评为“年度编程语言”。

1.1.3 Python3

Python 3.0 版本常被称为 Python 3000，或简称 Py3k。相对于 Python 的早期版本，这是一次较大的升级。Python 3.0 在设计的时候没有考虑向下相容，即许多针对 Python 早期版本设计的程序都无法在 Python 3.0 上正常执行。为了照顾现有程序，Python 2.6 作为一个过渡版本，基本使用了 Python 2.x 的语法和库，同时考虑了向 Python 3.0 的迁移，允许使用部分 Python 3.0 的语法与函数。新的 Python 程序建议使用 Python 3.0 版本的语法。大多数第三方库都正在努力地相容于 Python3，即使无法立即使用 Python3，也建议编写相容 Python3 的程序。本书中所有 Python 代码均采用 Python3 编写。

1.1.4 Python 的特点

1. Python 的优点

Python 作为目前应用广泛的程序设计语言，具有如下一些优点：

- 1) 易于学习。Python 有相对较少的关键字和一个明确定义的语法，结构简单，学习起来更加容易。
- 2) 易于阅读。Python 代码定义得更清晰。
- 3) 易于维护。Python 的成功在于它的源代码容易维护。
- 4) 广泛的标准库。Python 的最大优势之一是有丰富的库，是跨平台的，在 UNIX、Windows 和 Macintosh 上的兼容性很好。
- 5) 互动模式。有互动模式的支持，可以终端输入执行代码并获得结果的语言，互动测试和调试代码片段。
- 6) 可移植性。基于其开发源代码的特性，Python 已经被移植到许多平台。
- 7) 可扩展性。Python 程序中可调用 C 或者 C++ 程序。
- 8) 数据库。Python 提供所有主要的商业数据库的接口，如 Mysql、MongoDB 等。
- 9) 可嵌入性。可以将 Python 嵌入 C 或者 C++ 程序，让程序获得“脚本化”的能力。

2. Python 的缺点

虽然 Python 具有诸多优点，但同样存在缺点：

1) 运行速度慢。与 C 程序相比，Python 的运行速度非常慢。因为 Python 是解释型语言，代码在执行时需要逐行翻译成 CPU 能理解的机器码，这个翻译过程非常耗时，所以运行速度很慢；而 C 程序是运行已被编译成 CPU 能执行的机器码，所以运行速度非常快。

2) 代码不能加密。发布 Python 程序，实际上就是发布源代码，这一点跟 C 语言不同。C 语言不用发布源代码，只需要把编译后的机器码发布出去，要从机器码反推出 C 代码是不可能的。所以，凡是编译型语言，都没有这个问题，而解释型的语言，则必须把源码发布出去。

1.1.5 Python 的应用领域

Python 作为一种功能强大且通用的编程语言而广受好评，它具有非常清晰的语法特点，适用于多种操作系统，目前在国际上非常流行，应用越来越广泛。Python 具有优秀的扩展性，在诸多领域，如网络爬虫、人工智能、科学计算、Web 开发、系统运维、大数据、云计算、金融、图形界面和网站方面都具有良好的应用性。

1.2 Python 的安装

Python 具有良好的兼容性，可安装在 UNIX、Windows 和 Macintosh 等主流操作系统平台上。本节以 Windows 为例，详细介绍 Python 的安装过程。

1) 进入 Python 官网下载页面(下载网址是 <https://www.python.org/downloads/>)，如图 1-1 所示。

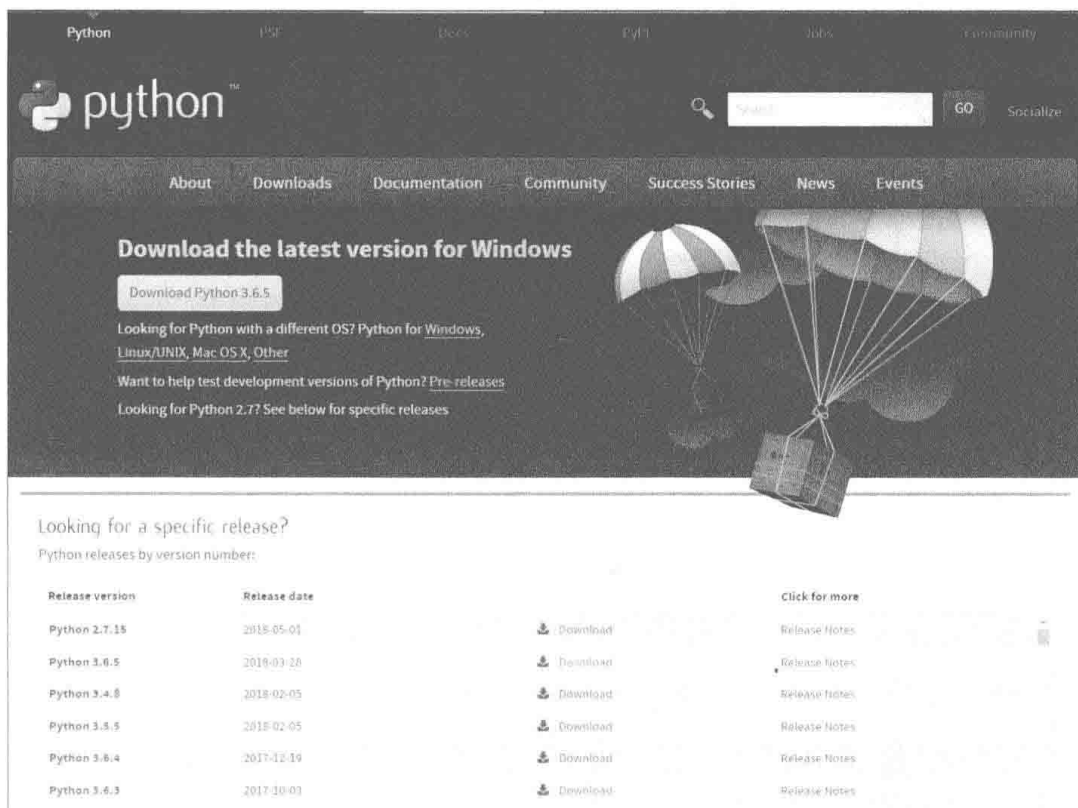


图 1-1 Python 官网下载页面

2) 单击“Download Python 3.6.5”按钮，进入下载页面。网站会根据操作系统下载合适的版本，如图 1-2 所示。

3) 双击安装程序进行安装。在安装开始界面选中“Add Python 3.6 to PATH”复选框，将 Python 添加到环境变量，如图 1-3 所示。

4) 可选择“Install Now”进行默认安装，也可选择“Customize installation”进行个性化安装，这里以个性化安装为例进行安装。单击“Customize installation”命令，进入个性化安装界面。在该界面可选择需要安装的模块，如图 1-4 所示。



图 1-2 下载 Python 安装程序

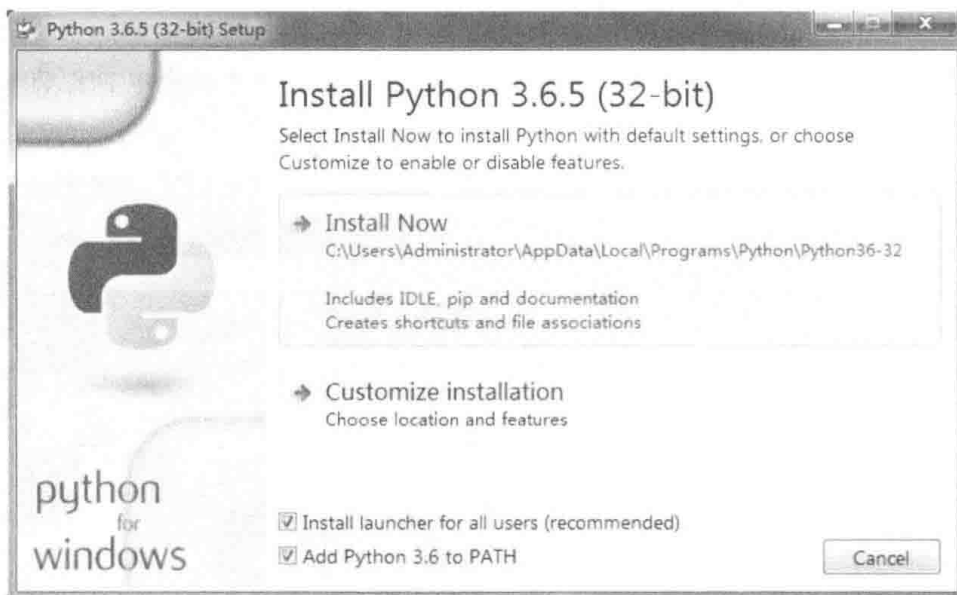


图 1-3 将 Python 添加到环境变量

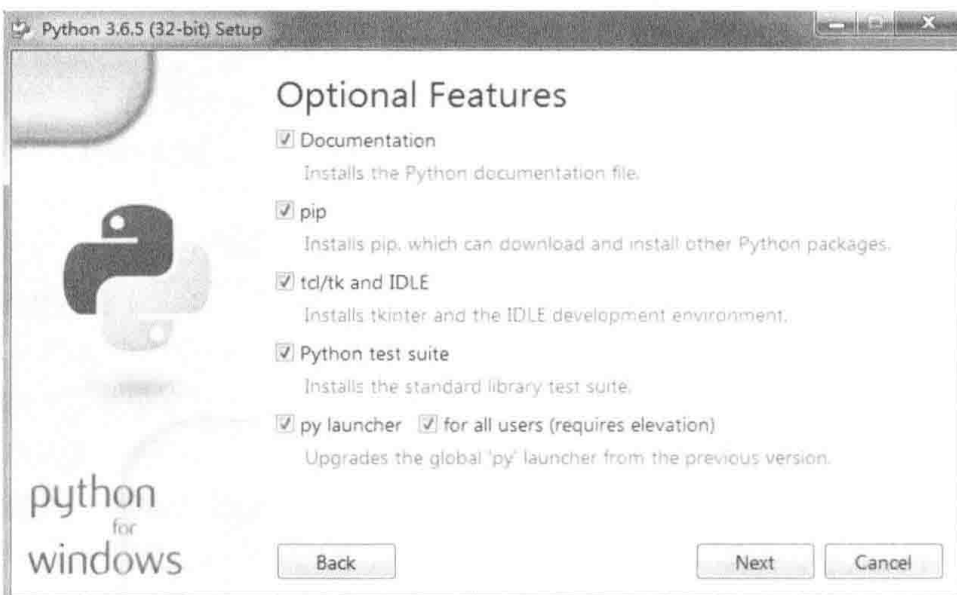


图 1-4 个性化安装界面

5) 单击“Next”按钮继续安装，进入“Advance Options”界面。在该界面中可进行选择安装目录等设置，如图 1-5 所示。

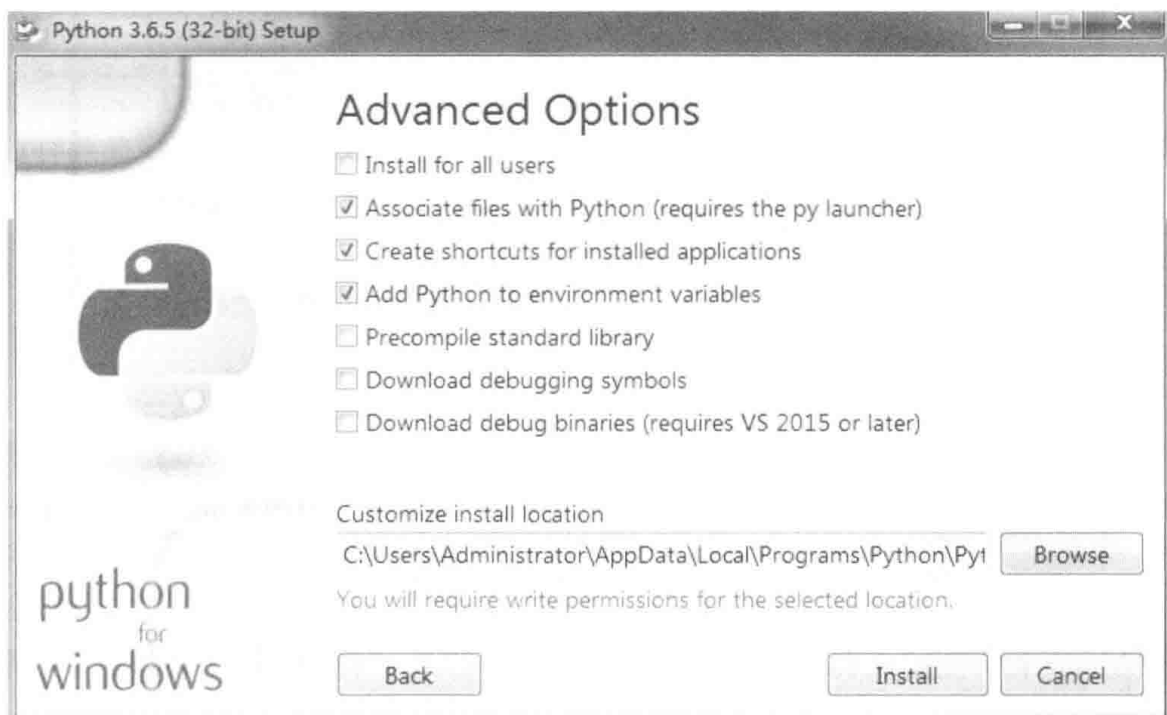


图 1-5 高级设置界面

6) 单击“Install”按钮进行安装，如图 1-6 所示。

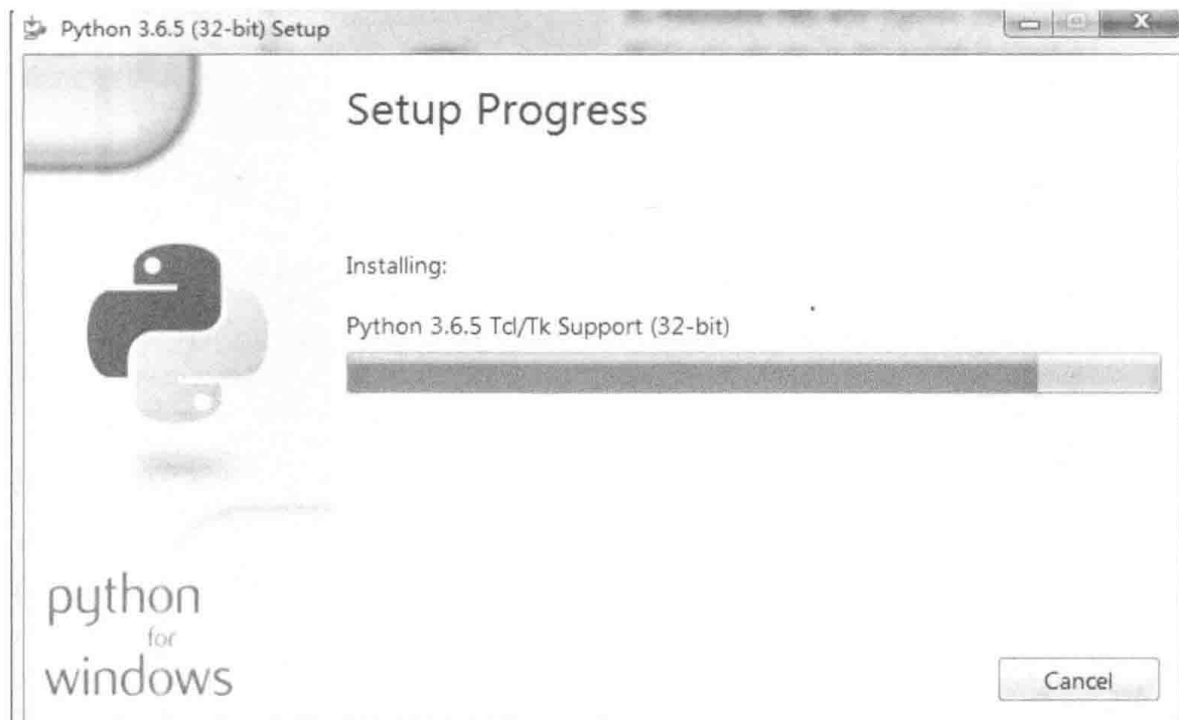


图 1-6 Python 安装界面

7) 安装完成后会自动弹出安装成功说明界面，如图 1-7 所示。

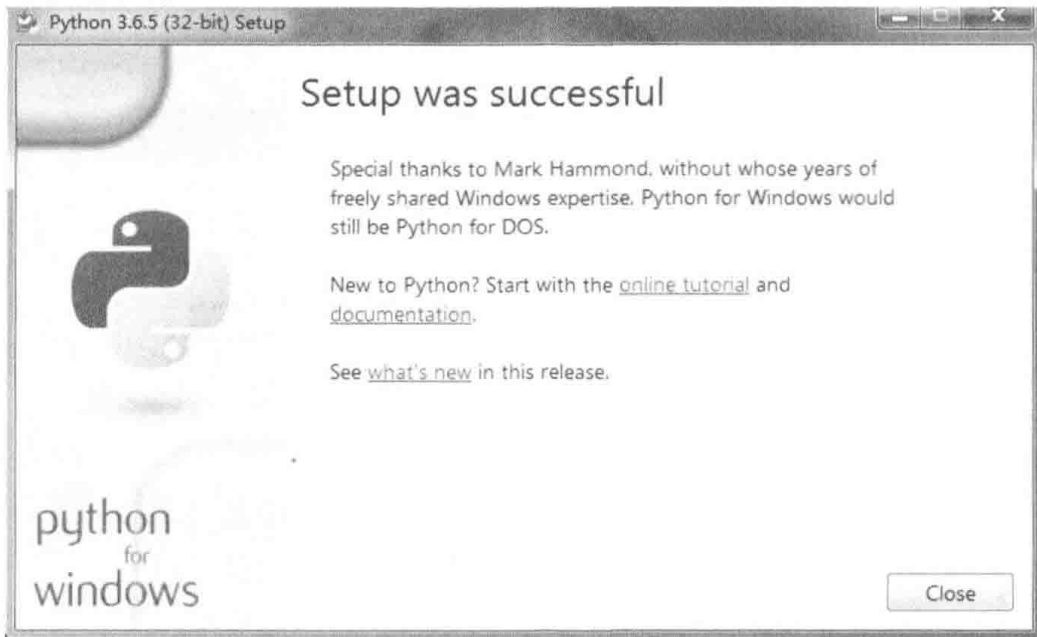


图 1-7 Python 安装成功说明界面

8) 在 cmd 窗口测试 Python 是否安装成功。输入“python”命令，如果显示类似版本等信息，则说明安装成功，如图 1-8 所示。

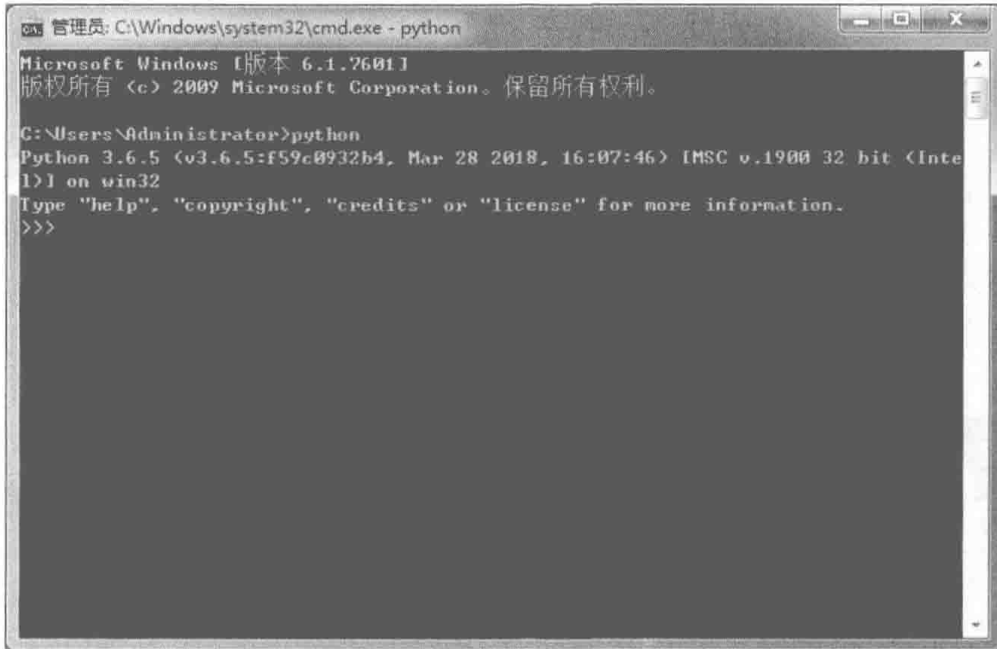


图 1-8 在 cmd 窗口测试 Python 是否安装成功

1.3 第一个 Python 程序

一般来说，第一个入门编程代码便是输出“Hello, World!”。下面就是使用 Python 编写的，实现输出“Hello, World!”功能的代码：

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

以上命令的输出结果如下:

```
Hello,World!
```

这样便编写了一个 Python 程序,成功输出了“Hello, World!”文本。在接下来的章节中,会详细讲解 Python 的语法,本节作为示例,不对代码进行详细说明。

1.4 搭建 Python 集成开发环境

在成功安装 Python 后,可在 cmd 窗口运行 Python 程序。但在 cmd 窗口运行程序存在效率较低,易出错等缺点。一般在开发 Python 程序时,会选择集成开发环境(Integrated Development Environment, IDE)。PyCharm 是一种使用广泛的 Python 集成开发环境,可帮助用户在使用 Python 语言开发时提高效率,如调试、语法高亮、Project 管理、代码跳转、智能提示、自动完成、单元测试、版本控制等。此外,该集成开发环境提供了一些高级功能,用于支持 Django 框架下的专业 Web 开发。本节以 PyCharm 为例,讲解 Python 集成开发环境的搭建。

1) 进入 PyCharm 官网(网址是 <http://www.jetbrains.com/pycharm/>),如图 1-9 所示。



图 1-9 PyCharm 官网

2) 单击“DOWNLOAD NOW”按钮进入下载版本选择页面,可选择相应的操作系统以及不同的版本进行下载,如图 1-10 所示。

3) Professional 版本具有 Python 开发的全部功能,这里以下载 Windows 操作系统的 Professional 版本为例。单击“Professional”下的“DOWNLOAD”按钮,弹出安装程序下载对话框,