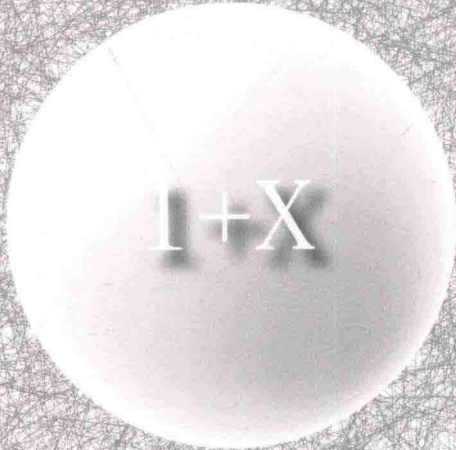


大学计算机基础教育规划教材

Python程序设计习题解析

周元哲 编著



1+X

清华大学出版社



大学计算机基础教育规划教材

Python程序设计习题解析

周元哲 编著

1+X

清华大学出版社
北 京

内 容 简 介

本书与《Python 程序设计基础》(周元哲编著)配套,针对该教材各章节中的内容,介绍每章的知识点和重点、难点知识,给出了每章的课后习题答案,并精心设计和安排了相应的习题解析。另外,本书新增 4 章内容:网络爬虫、软件测试框架、Web 开发框架、游戏开发。

本书可作为高等院校各专业程序设计和软件竞赛培训的辅导教材,也可作为程序员、编程爱好者的自学辅助用书和各类培训班的教学辅助用书。

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 程序设计习题解析/周元哲编著. —北京:清华大学出版社,2017

(大学计算机基础教育规划教材)

ISBN 978-7-302-46649-9

I. ①P… II. ①周… III. ①软件工具—程序设计—高等学校—题解 IV. ①TP311.561-44

中国版本图书馆 CIP 数据核字(2017)第 032050 号

责任编辑:张 民 战晓雷

封面设计:常雪影

责任校对:胡伟民

责任印制:刘海龙

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

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者:保定市中华美凯印刷有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:10.5 字 数:260 千字

版 次:2017 年 5 月第 1 版 印 次:2017 年 5 月第 1 次印刷

印 数:1~2000

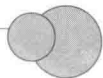
定 价:29.00 元

产品编号:073840-01



前言

Python 程序设计习题解析



Python 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,被列入 LAMP (Linux、Apache、MySQL 以及 Pthon/Perl/PHP),Python 语言将代码书写缩进作为语法规定,使得初学者养成良好的编码习惯。Python 具有序列、列表、元组和字典等数据结构,便于实现各种算法。作为开源语言,Python 外挂各种库,在大数据、科学计算等方面,功能堪比 Matlab。因此,作为面向对象的程序设计语言,Python 具有简单、现代、类型安全、性能优良,效率高等特点,是计算机等信息学科学生学习程序设计语言的首选之一。

本书与《Python 程序设计基础》(周元哲编著)相配套,针对《Python 程序设计基础》各章(Python 编程概述、数据类型和表达式、顺序与选择结构、循环结构、序列与字典、数据结构与算法、函数与模块、面向对象程序设计基础、文件、用户界面设计、绘图、数据库应用、网络编程和异常处理)中的内容,介绍每章的知识重点,给出每章的课后习题答案,并精心设计和安排了相应的习题与解答。另外,本书新增 4 章内容,包括网络爬虫、软件测试框架、Web 开发框架、游戏开发。

计算机学院李晓戈、邓万宇等阅读了部分手稿。西北工业大学软件与微电子学院郑炜对本教材的写作大纲、写作风格等提出了很多宝贵的意见。衷心感谢上述各位的支持和帮助。本书在写作过程中参阅了大量中英文专著、教材、论文、报告及网上的资料,限于篇幅,未能一一列出。在此,向有关作者一并表示敬意和衷心的感谢。

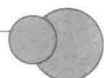
本书内容精练,文字简洁,结构合理,实训题目经典实用,综合性强,明确定位于面向初、中级读者,由入门起步,侧重提高。特别适合作为高等院校相关专业 Python 程序设计的教材或教学参考书,也可以供从事计算机应用开发的各类技术人员参考。

本书的 Python 版本为 2.7.3,所有程序都在 PyCharm 编辑器中进行了调试和运行。由于作者水平有限,时间紧迫,本书难免有疏漏之处,恳请广大读者批评指正。

作者的电子信箱是 zhouyuanzhe@163.com。

作者

2016 年 11 月



第 1 章 Python 编程概述	1
1.1 本章要求	1
1.2 本章知识重点	1
1.2.1 Python 的特点	1
1.2.2 Python 解释器	2
1.2.3 Python 编辑器	4
1.3 课后习题答案	7
第 2 章 数据类型和表达式	10
2.1 本章要求	10
2.2 本章知识重点	10
2.2.1 数据类型	10
2.2.2 变量	11
2.2.3 运算符	11
2.3 课后习题答案	13
2.4 习题与解答	14
2.4.1 习题	14
2.4.2 习题参考答案	14
第 3 章 顺序与选择结构	15
3.1 本章要求	15
3.2 本章知识重点	15
3.2.1 3 种基本逻辑结构	15
3.2.2 Python 程序设计流程	15
3.2.3 代码书写缩进	16
3.2.4 输入与输出	16
3.2.5 顺序结构	16
3.2.6 选择结构	16
3.3 课后习题答案	18

3.4 习题与解答	20
3.4.1 习题	20
3.4.2 习题参考答案	20
第4章 循环结构	23
4.1 本章要求	23
4.2 本章知识重点	23
4.2.1 构造循环结构	23
4.2.2 while 语句	24
4.2.3 for 语句	24
4.2.4 循环嵌套	24
4.3 课后习题答案	26
4.4 习题与解答	30
4.4.1 习题	30
4.4.2 习题参考答案	30
第5章 序列与字典	35
5.1 本章要求	35
5.2 本章知识重点	35
5.2.1 序列	35
5.2.2 列表	35
5.2.3 元组	36
5.2.4 字符串	37
5.2.5 字典	37
5.2.6 JSON	38
5.3 课后习题答案	39
5.4 习题与解答	42
5.4.1 习题	42
5.4.2 习题参考答案	42
第6章 数据结构与算法	44
6.1 本章要求	44
6.2 本章知识重点	44
6.2.1 数据结构	44
6.2.2 查找和排序	46
6.3 课后习题答案	48
6.4 习题与解答	51

6.4.1	习题	51
6.4.2	习题参考答案	51
第7章	函数与模块	53
7.1	本章要求	53
7.2	本章知识重点	53
7.2.1	函数的概念	53
7.2.2	函数声明和调用	53
7.2.3	函数的参数	54
7.2.4	递归函数	56
7.2.5	变量作用域	57
7.3	课后习题答案	57
7.4	习题与解答	59
7.4.1	习题	59
7.4.2	习题参考答案	59
第8章	面向对象程序设计基础	63
8.1	本章要求	63
8.2	本章知识重点	63
8.2.1	对象三大特性	63
8.2.2	类与对象	64
8.2.3	继承性	64
8.3	课后习题答案	64
8.4	习题与解答	66
8.4.1	习题	66
8.4.2	习题参考答案	66
第9章	文件	69
9.1	本章要求	69
9.2	本章知识重点	69
9.2.1	字符编码	69
9.2.2	文件分类	70
9.2.3	文件读写操作	70
9.3	课后习题答案	71
9.4	习题与解答	72
9.4.1	习题	72
9.4.2	习题参考答案	72

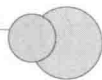
第 10 章 用户界面设计	74
10.1 本章要求	74
10.2 本章知识重点	74
10.2.1 界面设计原则	74
10.2.2 wxPython 开发流程	74
10.2.3 事件处理	75
10.3 课后习题答案	77
10.4 习题与解答	78
10.4.1 习题	78
10.4.2 习题参考答案	79
第 11 章 绘图与科学计算	84
11.1 本章要求	84
11.2 本章知识重点	84
11.2.1 NumPy	84
11.2.2 Matplotlib	85
11.2.3 scipy	85
11.2.4 pandas	87
第 12 章 数据库应用	90
12.1 本章要求	90
12.2 本章知识重点	90
12.2.1 关系型数据库	90
12.2.2 Python 连接数据库	91
12.2.3 Python 操作数据库	92
12.3 课后习题答案	92
第 13 章 网络编程	95
13.1 本章要求	95
13.2 本章知识重点	95
13.2.1 TCP/IP 四层模型	95
13.2.2 IP 地址和端口号	96
13.2.3 Socket 编程	96
13.3 课后习题答案	98
第 14 章 异常处理	100
14.1 本章要求	100

14.2	本章知识重点	100
14.2.1	错误类型	100
14.2.2	异常处理	100
14.2.3	PyCharm 调试功能	101
14.3	课后习题答案	104
14.4	习题与解答	104
14.4.1	习题	104
14.4.2	习题参考答案	105
第 15 章	网络爬虫	107
15.1	本章要求	107
15.2	本章知识重点	107
15.2.1	网络爬虫简介	107
15.2.2	正则表达式	107
15.2.3	Python re 模块	108
15.2.4	从网页上抓取特定信息	113
15.2.5	保存贴吧网页的小爬虫	114
第 16 章	软件测试框架	115
16.1	本章要求	115
16.2	本章知识重点	115
16.2.1	Python 与软件测试	115
16.2.2	用 PyUnit 进行单元测试	117
16.2.3	用 pywinauto 进行 GUI 测试	118
16.2.4	用 Selenium 进行 Web 测试	120
16.2.5	用 Pylot 进行性能测试	123
16.3	习题与解答	127
16.3.1	习题	127
16.3.2	习题参考答案	127
第 17 章	Web 开发框架	131
17.1	本章要求	131
17.2	本章知识重点	131
17.2.1	MVC 设计模式	131
17.2.2	web2py 框架	132
17.2.3	Django 框架	137

第 18 章 游戏开发	145
18.1 本章要求	145
18.2 本章知识重点	145
18.2.1 游戏简介	145
18.2.2 pygame 简介	145
18.2.3 pygame 模块	147
参考文献	154

第1章

Python编程概述



1.1 本章要求

- 了解计算机基础知识。
- 了解 Python 语言及其特点。
- 了解 Python 语言版本和开发环境。
- 掌握 Python 解释器的下载和安装。
- 掌握 Python 各类编辑器,特别是 Pycharm。
- 了解 Python 与其他语言的关系。

1.2 本章知识重点

1.2.1 Python 的特点

Python 是一种简单易学、功能强大的编程语言,它有高效率的高层数据结构,能够简单而有效地实现面向对象编程。Python 具有如下一些特点。

1. 简单易学

Python 语法简捷而清晰,易于快速上手学习,在学习过程中便于专注程序本身的逻辑和算法,探究程序执行的过程。

2. 免费开源

Python 是开源软件,可以自由地阅读它的源代码。

3. 解释型语言

Python 作为解释型语言,其源代码通过解释器转换成字节码的中间形式,由虚拟机在不同计算机上运行。

4. 面向对象

Python 是面向对象的语言,函数、模块、字符串都是对象,并且完全支持继承、重载、

派生、多重继承。

5. 丰富的库

Python 称为胶水语言,能够轻松地与其他语言(特别是 C 或 C++)结合在一起,具有丰富的 API 和标准库。

1.2.2 Python 解释器

下面介绍 Python 解释器在 Linux 和 Windows 下的安装步骤。

1. 在 Ubuntu 下安装 Python

Ubuntu(乌班图)是一个以桌面应用为主的 Linux 操作系统,内置 Python,如图 1.1 所示。

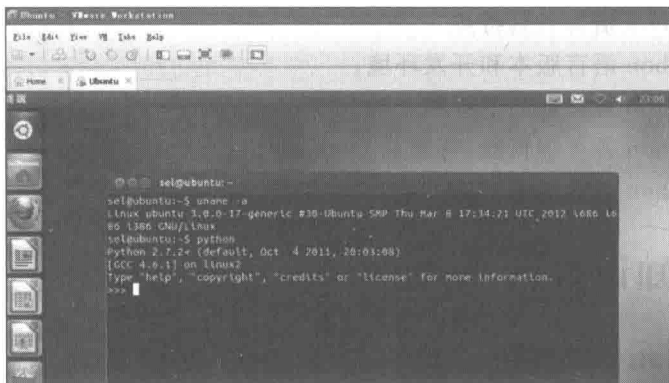


图 1.1 Ubuntu 下内置 Python

2. 在 Windows 下安装 Python

在 Windows 下安装 Python,一般具有如下步骤。

步骤一:下载 Python 2.7.3 安装包进行安装。

在浏览器中输入 <http://www.python.org>,在下载页 <https://www.python.org/download/releases/2.7.3/>中找到 Windows x86 MSI Installer (2.7.3) (sig)进行下载,如图 1.2 所示。

步骤二:在 Windows 环境变量中添加 Python。

将 Python 的安装目录添加到 Windows 下的 path 变量中,如图 1.3 所示。

步骤三:测试 Python 安装是否成功。

在 Windows 下使用 cmd 打开命令行窗口,输入 python 命令测试 Python 安装是否成功,图 1.4 表示安装成功。

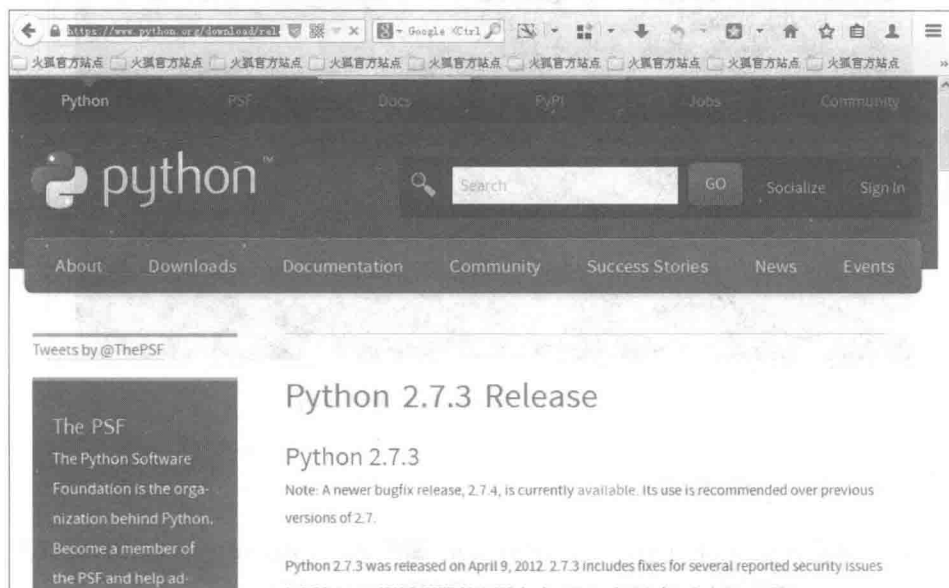


图 1.2 下载 Python 2.7.3

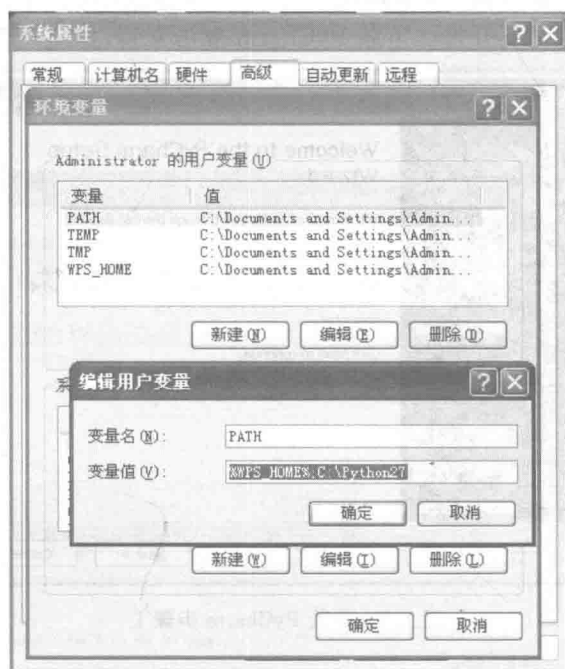


图 1.3 设置环境变量

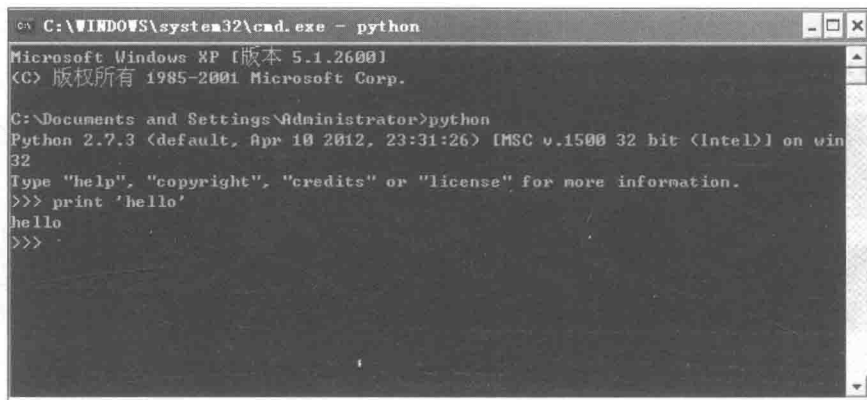


图 1.4 测试 Python 安装是否成功

1.2.3 Python 编辑器

Python 编辑器众多,有 Python 自带的 IDE 编辑器、Notepad++、UliPad 以及 Vim 和 emacs 等。Linux 下的 Eclipse with PyDev 和 Windows 下 Python 编辑器 PyCharm 功能较为强大,本书重点介绍 PyCharm。PyCharm 具有一整套可以帮助用户在使用 Python 语言开发时提高其效率的工具,比如调试、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。此外,PyCharm 提供了一些高级功能,支持 Django 框架下的专业 Web 开发。下载 PyCharm 双击安装,如图 1.5 所示。



图 1.5 安装 PyCharm 步骤 1

单击 Next 按钮,如图 1.6 所示。

安装结束,运行 PyCharm,如图 1.7 所示。

下一步,可以选择免费试用 30 天,如图 1.8 所示。

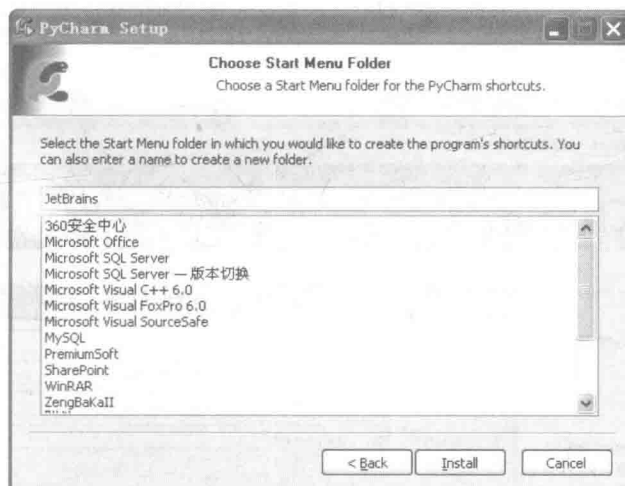


图 1.6 安装 PyCharm 步骤 2

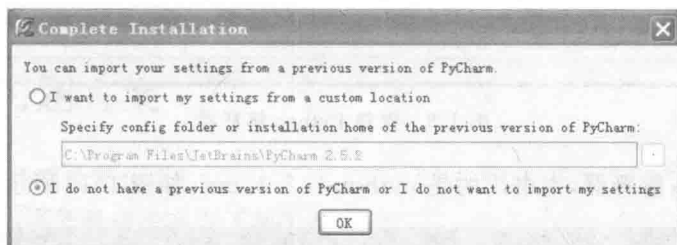


图 1.7 运行 PyCharm

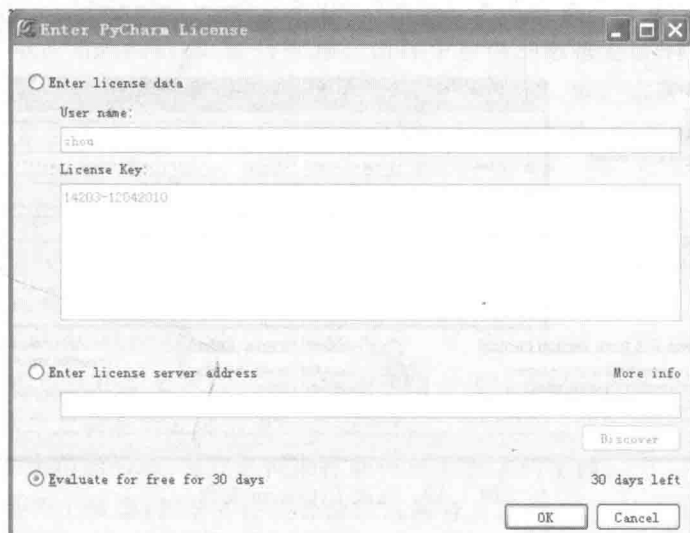


图 1.8 激活 PyCharm

单击 Create New Project, 输入项目名、路径, 选择 Python 解释器。如果没有 Python 解释器, 对话框如图 1.9 所示。

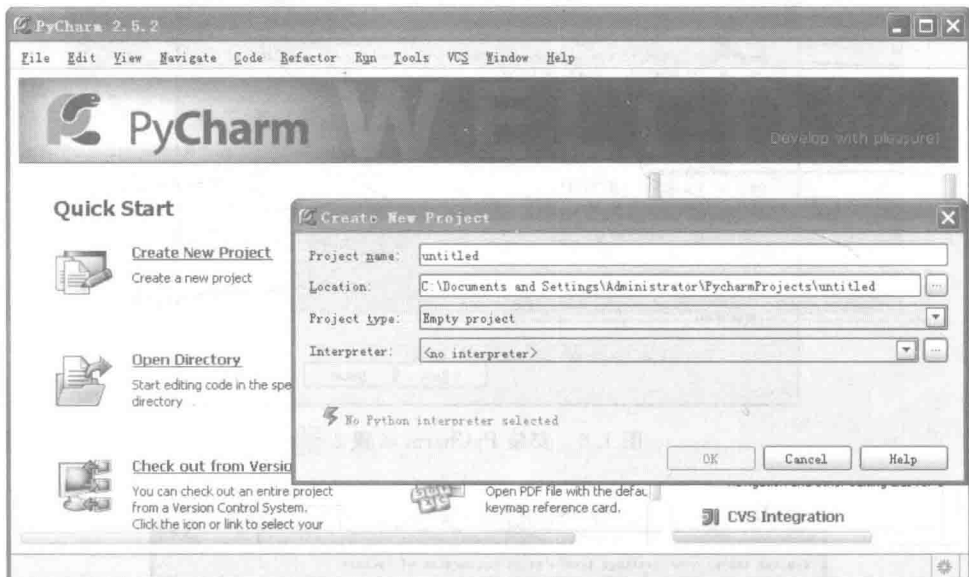


图 1.9 选择 Python 解释器

选择 Python 解释器, 本书用的是 python-2.7.3.msi, 如图 1.10 所示。

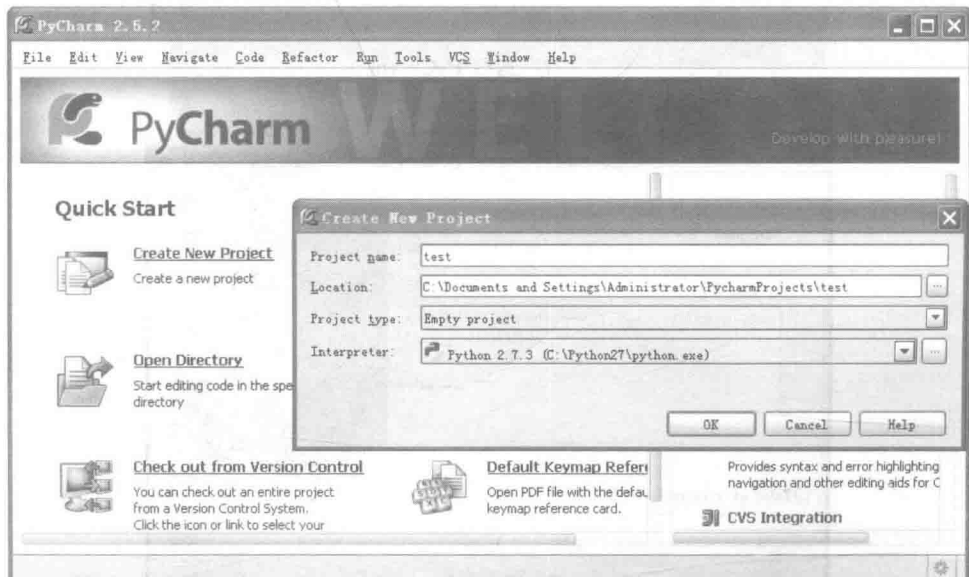


图 1.10 完成 PyCharm 安装

启动 PyCharm, 创建 Python 文件, 如图 1.11 所示。

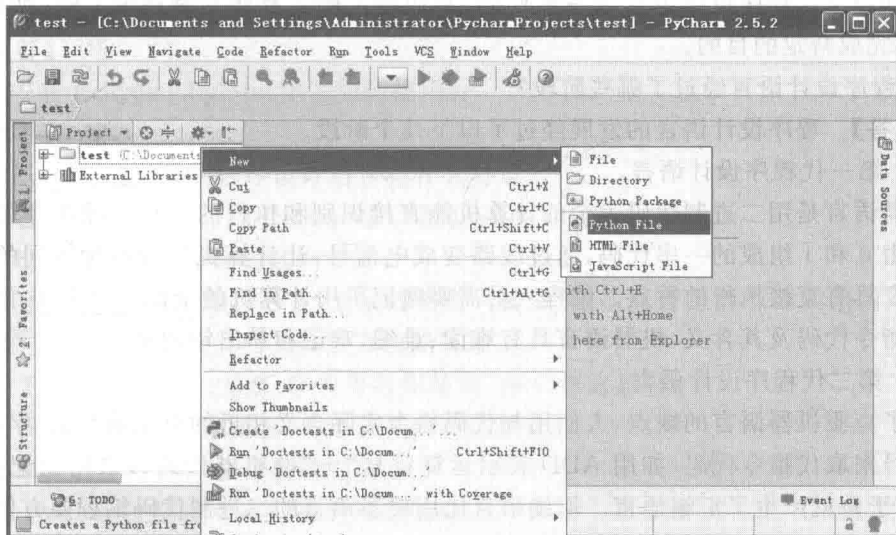


图 1.11 在 PyCharm 中创建 Python 文件

1.3 课后习题答案

1. 冯·诺依曼理论是什么？

【解答】 冯·诺依曼理论有以下两个要点：

(1) 计算机硬件设备由存储器、运算器、控制器、输入设备和输出设备 5 个部分组成。其中，运算器和控制器组成中央处理器单元(Center Process Unit, CPU)。中央处理单元用于执行指令，如算术操作、从别的设备写入或读出数据。存储器分为内存和外存。CPU 从内存中读取所需要的数据，进行处理。内存中存储的数据是临时的，当程序退出或者计算机关机时，数据将会丢失。如果需要永久地存储数据，需要用到外存，如硬盘、闪存等设备。键盘、鼠标等输入设备用于接收用户输入数据和指令，显示器通常作为输出设备。

(2) 存储程序思想。把计算过程描述为由许多命令按一定顺序组成的程序，然后把程序和数据一起输入计算机，计算机对已存入的程序和数据进行处理后输出结果。

2. 软件和程序是否一样？

【解答】 一般认为，软件包括以下一些内容：

- (1) 运行时，能够提供所要求的功能和性能的指令或计算机程序集合。
- (2) 程序能够令人满意地处理信息的数据结构。
- (3) 有描述程序功能需求、程序如何操作和使用所要求的文档。

软件和程序是两个概念，初学者往往会混淆这两者。其实，这发生在软件历史的第一阶段(20 世纪 50 年代初期至 60 年代中期)，由于软件的生产个体化，规模较小，功能单一，软件只有程序而无文档，形成了“软件等于程序”的错误观念。程序是为实现特定目标或解决特定问题而用计算机语言编写的命令序列的集合，通过使用与自然语言具有相似