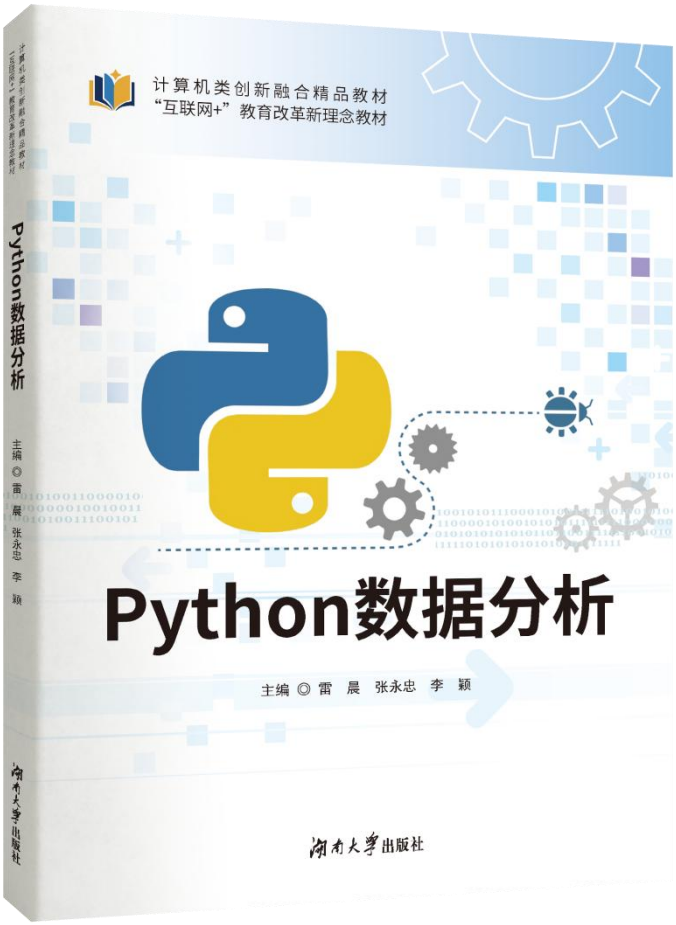
PYthon 数据分析



类目：计算机类
书名：PYthon 数据分析
主编：雷 晨 张永忠 李 颖
出版社：湖南大学出版社
开本：大 16 开
书号：978-7-5667-3375-7
使用层次：通用
出版时间：2024 年 1 月
定价：49.80 元
印刷方式：双色
是否有资源：是

责任编辑：黄　旺
封面设计：旗语书装

Python数据分析

ISBN 978-7-5667-1171-7
定价：49.80元

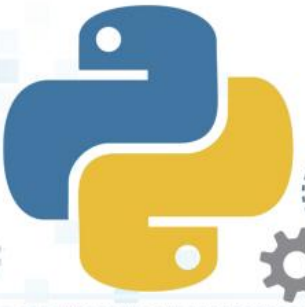
计算机类创新融合精品教材
“互联网+”教育改革新理念教材

Python数据分析

主编◎雷　晨　张永忠　李　颖

湖南大学出版社

计算机类创新融合精品教材
“互联网+”教育改革新理念教材

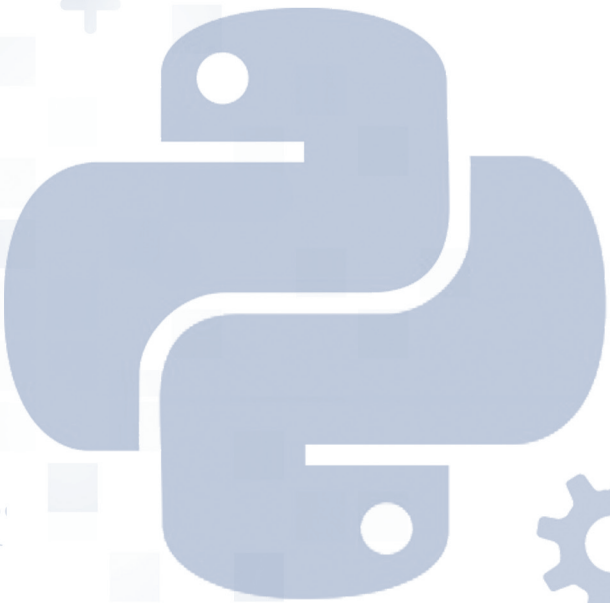


Python数据分析

主编◎雷　晨　张永忠　李　颖

湖南大学出版社

计算机类创新融合精品教材
“互联网+”教育改革新理念教材



Python 数据分析

主　编◎雷　晨　张永忠　李　颖
副主编◎凌　宁

湖南大学出版社
·长 沙·

图书在版编目（CIP）数据

Python 数据分析 / 雷晨 , 张永忠 , 李颖主编 . -- 长沙 : 湖南大学出版社 , 2024.1
ISBN 978-7-5667-3375-7

Ⅰ . ① P… Ⅱ . ①雷… ②张… ③李… Ⅲ . ①软件工具—程序设计 Ⅳ . ① TP311.561

中国国家版本馆 CIP 数据核字（2024）第 016902 号

Python 数据分析

Python SHUJU FENXI

主　　编：雷　晨　张永忠　李　颖
责任编辑：黄　旺
印　　装：涿州汇美亿浓印刷有限公司
开　　本：880 mm×1 230 mm　1/16　　**印　　张：**11　　**字　　数：**294 千字
版　　次：2024 年 1 月第 1 版　　**印　　次：**2024 年 1 月第 1 次印刷
书　　号：ISBN 978-7-5667-3375-7
定　　价：49.80 元

出 版 人：李文邦
出版发行：湖南大学出版社
社　　址：湖南・长沙・岳麓山　　**邮　　编：**410082
电　　话：0731-88822559（营销部）　88821174（编辑室）　88821006（出版部）
传　　真：0731-88822264（总编室）
网　　址：http://press.hnu.edu.cn
电子邮箱：xiaoshulianwenhua@163.com

前言 preface

随着人工智能、大数据时代的到来，Python 以其丰富的资源库、超强的可移植性和可扩展性成为数据科学与机器学习工具及语言的首选。如何学习利用 Python 进行大数据分析与挖掘，是广大初学者或者对数据挖掘技术感兴趣的读者非常关心的问题，也是高校众多专业学生需要学习和掌握的专业技能。本书介绍了 Python 的基础及数据分析和数据可视化的包、机器学习和深度学习等基本知识，希望帮助广大读者较好地掌握相关知识和技能，建立数据分析与挖掘的思维。

本书共有 8 个章节，包括数据分析基础、初识 Python、Python 基础与数据抓取、数据预处理、探索性数据分析、数据可视化包 MatpLotLib、数据库类型、数据挖掘工具，提供了丰富的学习内容，力求为读者打造一本“入门学习 + 应用 + 实践一体化”的 Python 数据分析图书。

本书在编写过程中参考和引用了一些专家、学者的研究成果和文献资料，同时也从媒体上查阅了相关内容，在此一并表示感谢。我们已经尽最大努力避免在文本和代码中出现错误，但是由于水平有限，编写时间仓促，书中难免出现一些疏漏和不足的地方，恳请读者批评指正，我们将在今后的修订过程中，进一步梳理与完善。

编　者

目录 contents

第 1 章
数据分析基础

学习目标

知识目标

- 理解数据分析的定义及重要性。
- 熟悉数据分析的基本流程。

能力目标

- 能够识别掌握数据分析常用工具的特点及用途。
- 能够对数据分析流程进行梳理。

素质目标

- 加强学生数据意识，提高数据思维。
- 提高学生学习、挖掘、从业数据分析的意愿。

思政目标

培养学生诚信守法品质，注重数据分析过程中的权利和隐私保护。

1.1 数据分析的概念

数据分析是利用数学、统计学理论相结合的科学统计分析方法，对 Excel 数据、数据库中的数据、收集的大量数据、网页抓取的数据进行分析，从中提取有价值的信息并形成结论进行展示的过程。

数据分析的本质，是通过总结数据的规律，解决业务问题，以帮助在实际工作中的管理者做出判断和决策。

数据分析主要包括如下三个方面内容：

（1）现状分析：分析已经发生了什么。

（2）原因分析：分析为什么会出现这种现状。

（3）预测分析：分析未来可能发生什么。

1.2 数据分析的重要性

Python

数据分析的重要性

大数据、人工智能时代的到来使数据分析无处不在。数据分析帮助人们做出判断，以便采取适当的措施，发现机遇、创造新的商业价值，以及发现企业自身的问题和预测企业的未来。

在实际工作中，无论从事哪种行业，从数据分析师、市场营销策划、销售运营、财务管理、客户服务、人力资源，到教育、金融等行业（如图 1-1 所示），数据分析都是基本功，它不单单是一个职位，而是职场必备技能，能够掌握这一项技能必然是职场的加分项。

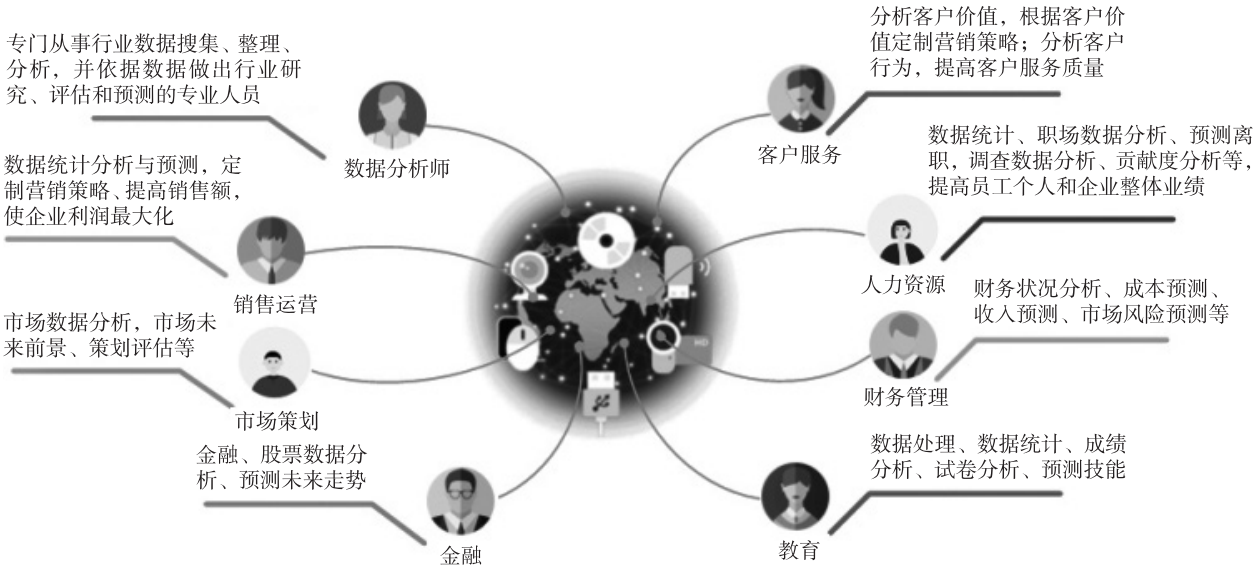


图 1-1　数据分析的行业需求

下面列举两个例子为大家展示合理运用数据分析的重要性。

情景 1：运营人员向管理者汇报工作，说明销量增长情况

（1）表达一：这个月比上个月销量好。

（2）表达二：11 月份的销量环比增长 69.8%，全网销量排名第一。

（3）表达三：近一年全国销量如图 1–2 所示，月平均销量 2834.5，整体呈上升趋势，6 月份环比增长 43.7%、7 月份环比增长 16.1%、9 月份环比增长 56.8%、11 月份环比增长 69.8%。虽然“618”大促的销量比 5 月份有所提高，但表现并不好，与双十一相比差很多，未来要加大“618”前后的宣传力度，做好预热和延续工作。

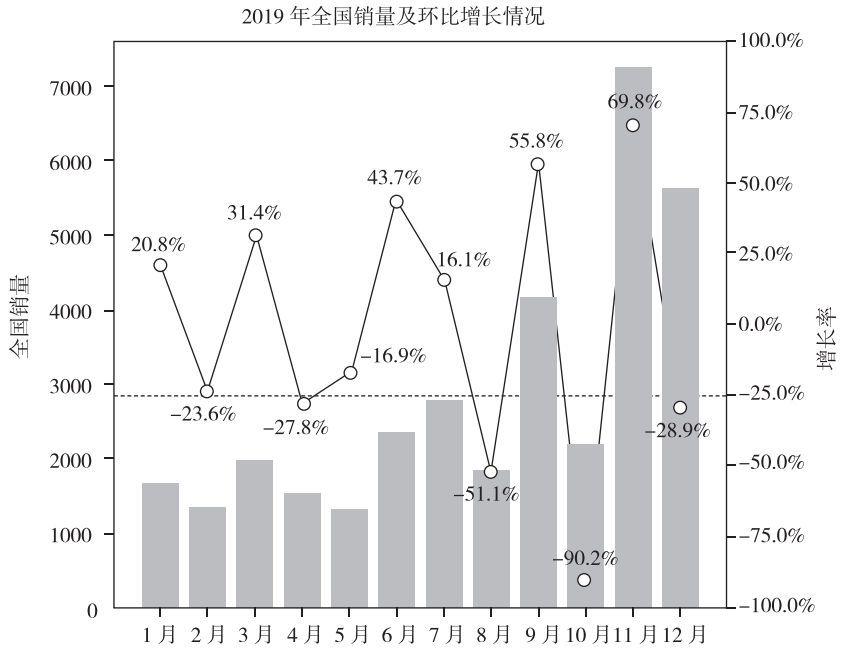


图 1–2　销量及环比增长情况

如果您是管理者，更青睐于哪一种表达方式呢？

其实，管理者需要的是真正简单、清晰的分析，以及接下来的决策方向。根据运营给出的解决方案，他可以预见公司未来的发展，从而解决真正的问题，提高平台的业务量。

情景 2：啤酒和纸尿裤的故事

为什么沃尔玛会将看似毫不相干的啤酒和纸尿裤（如图 1–3 所示）摆在一起销售，而啤酒和纸尿裤的销量却双双增长呢？

图 1–3　啤酒和纸尿裤

因为沃尔玛很好地运用了数据分析，发现了“纸尿裤”和“啤酒”的潜在联系。原来，美国的太太们常叮嘱她们的丈夫下班后为小孩买纸尿裤，而丈夫们在购买纸尿裤的同时又会随手带回两瓶啤酒。

而这一消费行为导致了这两件商品经常被同时购买。所以，沃尔玛索性就将它们摆放在一起，既方便顾客，又提高了产品销量。

还有很多通过数据分析而获得成功的例子。比如在营销领域，对客户的分群数据进行统计、分类，判断客户的购买趋势，对产品数据进行统计及预测销量，同时还可以发现销量薄弱点进行改善；在金融领域预测股价及其波动。

数据分析是如此重要，未来如果不懂数据分析，也将会与很多热门职位失之交臂。

1.3 数据分析的基本流程

数据分析的基本流程如图 1-4 所示。

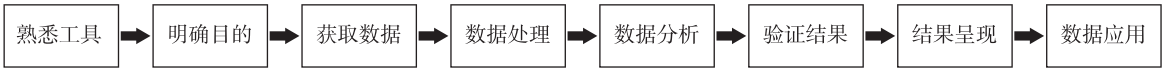


图 1-4　数据分析的基本流程图

1.3.1 熟悉工具

掌握一款数据分析工具至关重要，它能够帮助你快速解决问题，从而提高工作效率。常用的数据分析工具有 Excel、SPSS、R 语言、Python 语言，而本书采用的是 Python 语言。

1.3.2 明确目的

“如果给我 1 个小时解答一道决定我生死的问题，我会花 55 分钟来弄清楚这道题到底是在问什么。一旦清楚了它到底在问什么，剩下的 5 分钟足够回答这个问题。”——爱因斯坦。

在数据分析方面，首先要花一些时间搞清楚为什么要做数据分析、分析什么、想要达到什么效果。例如，为了评估产品改版后的效果相比之前是否有所提升，或通过数据分析找到产品迭代的方向等。

只有明确了分析目的，才能够找到适合的分析方法，才能够有效地进行数据处理、数据分析和预测等后续工作，最终得到结论并应用到实际中。

1.3.3 获取数据

数据的来源有很多，像我们熟悉的 Excel 数据、数据库中的数据、网站数据以及公开的数据集等。

获取数据之前首先要知道需要什么时间段的数据，哪个表中的数据，以及如何获得，比如是下载、复制还是爬取等。

数据处理

1.3.4 数据处理

数据处理是从大量、杂乱无章、难以理解、缺失的数据中，抽取并推导出对解决问题有价值、有意义的数据。数据处理主要包括数据规约、数据清洗、数据加工等，具体流程如图 1-5 所示。

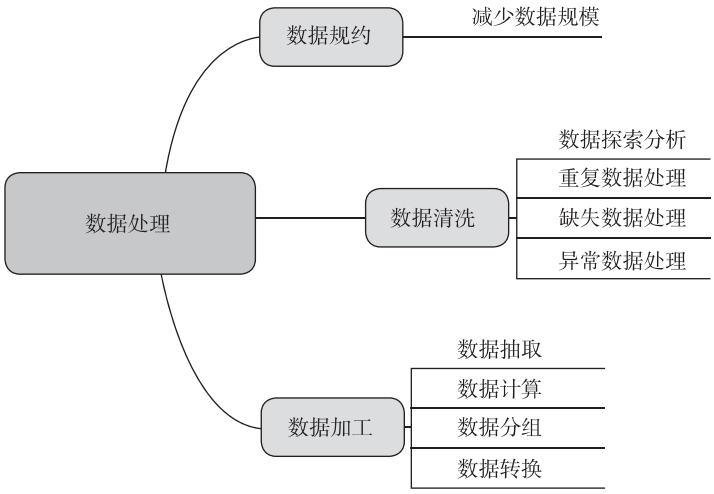


图 1–5　数据处理流程图

下面分别进行介绍：

（1）数据规约

在接近或保持原始数据完整性的同时将数据集规模减小，以提高数据处理的速度。例如，一个 Excel 表中包含近三年的几十万条数据，由于只分析近一年的数据，所以要一年的数据即可，这样做的目的就是减小数据规模，提高数据处理速度。

（2）数据清洗

在获取到原始数据后，可能其中的很多数据都不符合数据分析的要求，那么就需要按照如下步骤进行处理：

①数据探索分析：分析数据的规律，通过一定的方法统计数据，通过统计结果判断数据是否存在缺失、异常等情况。例如，通过最小值判断数量、金额是否包含缺失数据，如果最小值为 0，那么这部分数据就是缺失数据，以及通过判断数据是否存在空值来判断数据是否缺失。

②重复数据处理：对于重复的数据删除即可。

③缺失数据处理：对于缺失的数据，如果比例高于 30%，则可以选择放弃这个指标，删除即可；如果低于 30%，则可以将这部分的缺失数据进行填充，以 0 或均值填充。

④异常数据处理：异常数据需要对具体业务进行具体分析和处理，对于不符合常理的数据可进行删除。例如，性别男或女，如果数据中存在其他值，以及年龄超出了正常年龄范围，那么这些都属于异常数据。

（3）数据加工

数据加工包括数据抽取、数据计算、数据分组和数据转换：

①数据抽取：指选取数据中的部分内容。

②数据计算：进行各种算术和逻辑运算，以便得到进一步的信息。

③数据分组：按照有关信息进行有效的分组。

④数据转换：指数据标准化处理，以适应数据分析算法的需要，常用的有 z–score 标准化、“最小、最大标准化”和“按小数定标标准化”等。经过上述标准化处理后，数据中的各个指标值将会处在同一个数量级别上，以便更好地对数据进行综合测评和分析。

1.3.5 数据分析

在数据分析过程中，选择适合的分析方法和工具很重要，所选择的分析方法应兼具准确性、可操作性、可理解性和可应用性。但对于业务人员（如产品经理或运营）来说，在数据分析中最重要的是数据分析思维。

1.3.6 验证结果

通过数据分析会得到一些结果，但是这些结果只是数据的主观结果的体现，有些时候不一定完全准确，所以必须要进行验证。

例如，数据分析结果显示某产品点击率非常高，但实际下载量平平，对于这种情况先不要轻易定论这个产品受欢迎，而需要进一步验证，找到真正影响点击率的原因，这样才能更好地决策。

1.3.7 结果呈现

现如今，企业越来越重视数据分析为业务决策带来的有效应用，而可视化则是数据分析结果呈现的重要步骤。可视化是以图表方式呈现数据分析结果的，这样的结果会更清晰、直观，容易理解。

1.3.8 数据应用

数据分析的结果并不仅仅是把数据呈现出来，更应该关注的是通过分析这些数据之后可以做什么，如何将数据分析结果应用到实际业务当中去。

数据分析结果的应用是数据产生实际价值的直接体现，而这个过程需要具有数据沟通能力、业务推动能力和项目工作能力。如果得到了数据分析结果后并不知道做什么，那么这个数据分析就是失败的。

1.4 数据分析的常用工具

数据分析的常用工具

选择合适的数据分析工具尤为重要，下面介绍两种常用的数据分析工具——Excel 工具和 Python 语言。

1.4.1 Excel 工具

Excel 具备多种强大功能，例如创建表格、数据透视表和 VBA 等，Excel 的系统如此庞大，确保了大家可以根据自己的需求分析数据。

但是在当今的大数据、人工智能时代，在数据量很大的情况下 Excel 已经无法胜任，不仅处理起来很麻烦，而且处理速度也会变慢。而从数据分析的层面，Excel 也只是停留在描述性分析的阶段，例如对比分析、趋势分析、结构分析等。

1.4.2 Python 语言

虽然 Excel 已尽最大努力考虑到数据分析的大多数应用场景，但由于它是定制软件，很多东西都

固化了，不能自由修改。而 Python 语言则非常的强大和灵活，可以编写代码来执行所需的任何操作，从专业和方便的角度来看，它比 Excel 更加强大。另外，Python 还可以实现 Excel 难以实现的应用场景，具体内容如下：

（1）专业的统计分析

例如，正态分布、使用算法对聚类进行分类和回归分析等。这种分析就像使用数据做实验一样，它可以帮助我们回答以下问题。

数据的分布是正态分布、三角分布还是其他类型的分布？离散情况如何？它是否在我们想要达到的统计可控范围内？不同参数对结果的影响是多少？

（2）预测分析

例如，我们打算预测消费者的行为，预测他会在我们的商店停留多长时间，他会花多少钱，还可以找出他的个人信用情况，并根据他的在线消费记录确定贷款金额，或者根据他在网页上的浏览历史来推送不同的商品。

Python 作为数据分析工具，具有以下优势：

① Python 语言简单易学、数据处理方便高效，对于初学者来说更加容易上手。

② Python 的第三方扩展库不断更新，可用范围越来越广。

③在科学计算、数据分析、数学建模和数据挖掘方面占据越来越重要的地位。

④可以和其他语言进行对接，兼容性稳定。

本章小结

通过本章的学习，能够使读者对数据分析有基本的认识，了解什么是数据分析、数据分析的重要性，以及数据分析的基本流程和常用工具。

章 节 练 习

一、填空题

1. 数据分析是利用_____、_____理论相结合的_____分析方法，对 Excel 数据、数据库中的数据、收集的大量数据、网页抓取的数据进行分析，从中提取_____并形成结论进行展示的过程。

2. 数据分析主要包括：_____、_____、_____。

3. 数据分析帮助人们做出_____，以便采取适当的_____，发现机遇，创造新的商业价值，以及发现企业自身的问题和_____。

二、简答题

1. 简述一下数据分析的基本流程。

2. 谈一谈你所了解的数据分析工具。

第 2 章
初识 Python

学习目标

知识目标

- 了解 Python 的定义和特点。
- 掌握 Python 的应用场景。

能力目标

- 能够独立搭建 Python 的运行环境。
- 能够独立安装 PyCharm 开发环境。

素质目标

- 增强对程序编写的认知。
- 提高计算机应用素质。

思政目标

启发学生良好的编程思维，增强代码规范意识。

2.1 Python 概述

Python 是一种高级编程语言，由 Guido van Rossum 于 1989 年底发明，并在 1991 年发布了第一个公开版。Python 的设计哲学强调代码的可读性和简洁性，其语法清晰明了，它已经成为许多领域的首选语言，如数据科学、人工智能、网络编程、Web 开发、游戏开发等。

2.1.1 Python 简介

Python 简介

Python 最初的设计哲学是作为一种可读性强、可维护性强的语言，具有简洁而清晰的语法，易于学习。Python 的开发一直以来都秉持着优雅、明确、简单的哲学，这也成为 Python 的一个特色。Python 采用缩进来代替代码块括号，这种方式使得代码更加易读，但同时也需要开发者遵循一定的代码规范。

1991 年，Python 的第一个版本 (Python 0.9.0) 发布，之后 Python 的发展越来越迅速。1994 年，Python 1.0 版本正式发布，引入了异常处理、函数式编程等特性。1999 年 ,Python 2.0 版本发布，添加了垃圾回收机制、列表推导式、Unicode 支持等特性。2008 年 ,Python 3.0 版本发布，这是 Python 发展历史上最重要的一次升级，Python 3.0 中对语言的一些不合理设计进行了重构，增加了一些新特性，但同时也与 Python 2.x 不完全兼容。

Python 是一种解释型语言，这意味着 Python 代码不需要编译成机器码就可以直接执行。Python 的解释器可以在各种操作系统上运行，并且可以轻松地与其他语言集成。Python 是一种面向对象的语言，它支持类、对象和继承等面向对象的编程范式，也可以通过函数式编程、过程式编程等其他编程范式来编写代码。

Python 具有强大的标准库，其中包含许多可用于各种用途的模块和函数，例如网络编程、GUI 开发、图像处理、数据分析等。此外，Python 社区拥有庞大的第三方库和工具生态系统，可以帮助开发人员更加高效地编写代码。

在近年来，Python 变得越来越流行，尤其是在数据科学和人工智能领域。Python 具有许多优点，如易于学习、高度可读性、强大的标准库等，因此它成为了一种非常受欢迎的编程语言。Python 被广泛应用于科学计算、数据处理、机器学习、人工智能、网络编程、自动化脚本等领域，同时也是许多大型公司和互联网公司的主要编程语言之一。

2.1.2 Python 的特点

（1）简单易学

Python 的语法简单明了，非常容易学习和上手。Python 往往只需要几行代码就能解决其他编程语言需要很多行代码才能解决的问题。下面是一个 Python 程序的例子，它将两个数相加并输出结果：

```
a=3
b=5
print(a + b)
```

可以看到，这个程序只有 3 行代码，其中第一行定义了变量 a，第二行定义了变量 b，第三行将变量 a 和变量 b 相加并输出结果。相比之下，如果使用 JAVA 语言实现同样的功能，则需要编写类似下面的代码。

```
public class AddNumbers {
    public static void main(String[] args) {
        int a=3;
        int b=5;
        System.out.println(a + b);
    }
}
```

（2）面向对象

Python 是一种面向对象的编程语言，这意味着它支持面向对象的编程风格和特性。

Python 中一切皆为对象，包括数值、字符串、函数、类等。面向对象编程是一种常用的编程思想，可以让代码更加简洁、灵活和易于维护。下面是一个使用 Python 实现的简单类的例子：

```
class Student:
    def __init__(self, name, age):
        self.name=name
        self.age=age
    def print_info(self):
        print("Name: ", self.name)
        print("Age: ", self.age)
s=Student("Tom", 20)
s.print_info()
```

这个程序定义了一个名为 Student 的类，该类包含两个属性 name 和 age 以及一个方法 print_info。通过创建一个 Student 对象并调用 print_info 方法，我们可以输出这个对象的属性值。

（3）动态类型

Python 是一门动态类型的编程语言，不需要显式声明变量的类型，Python 会根据变量的值自动推导其类型。这使得 Python 非常灵活，可以快速迭代开发，同时也降低了代码的复杂性。

```
#Python 可以自动推导变量的类型
x=5      # 整型
y=3.14   # 浮点型
z="Hello"  # 字符串型
# 可以改变变量的类型
x="World" # 现在 x 是字符串型
```

（4）跨平台

Python 可以运行在多个操作系统上，包括 Windows、Mac OS X、Linux、Unix 等。这意味着开发人员可以在不同的平台上编写和运行 Python 代码，而不需要担心代码的兼容性和可移植性。

```
# 在 Windows 上运行的 Python 程序
print("Hello，Windows!")
# 在 Mac OS X 上运行的 Python 程序
print("Hello，Mac!")
# 在 Linux 上运行的 Python 程序
print("Hello，Linux!")
```

（5）大量的标准库

拥有大量的标准库是 Python 的一大特点，它包含了各种各样的模块，用于完成许多常见的任务。标准库中的模块可以直接导入并在代码中使用，从而避免了重复造轮子的情况。下面是一些常见的标准库模块：

① os 模块：提供了访问操作系统功能的接口，如创建目录、删除文件等。

② datetime 模块：用于处理日期和时间。

③ random 模块：用于生成伪随机数。

④ re 模块：提供了正则表达式操作功能，用于对文本进行匹配和搜索。

⑤ json 模块：用于处理 JSON 格式的数据。

除了标准库，Python 还有许多第三方库和模块，可以用于实现各种功能，如数据分析、图像处理、网络编程等。这些库可以通过 pip 工具进行安装。例如：

① numpy 库：用于数值计算和科学计算。

② matplotlib 库：用于绘制数据可视化图表。

③ requests 库：用于发送 HTTP 请求和获取数据。

④ pandas 库：用于数据处理和分析。

总的来说，Python 的标准库和第三方库使得开发人员能够快速实现各种功能，极大地提高了开发效率。

2.1.3 Python 的应用场景

（1）数据科学和人工智能

Python 是一种在数据科学和人工智能领域中最受欢迎的编程语言之一。许多 Python 库和框架可以帮助数据科学家和人工智能研究人员快速高效地处理数据，并使用机器学习和深度学习算法来训练模型。以下是 Python 在数据科学和人工智能领域中的应用：

①数据分析：Pandas、NumPy、SciPy。

②机器学习：scikit-learn、TensorFlow、Keras、PyTorch。

③自然语言处理：NLTK、SpaCy、Gensim。

（2）网络开发

Python 在网络开发领域中也有很好的应用。Python 提供了许多库和框架，可以帮助开发人员快速构建网络应用。以下是 Python 在网络开发领域中的应用：

① Web 开发：Django、Flask、Bottle。

②网络爬虫：Scrapy、BeautifulSoup。

③网络编程：Socket。

④网络服务器：Twisted、Tornado、Gunicorn。

2.2 搭建 Python 运行环境

2.2.1 安装 Python

（1）下载 Python 安装程序

首先，我们需要到 Python 官网 (https://www.python.org/downloads/) 下载 Python 的安装程序。打开官网后，我们可以看到如图 2–1 所示的界面。

在这里，我们可以选择下载最新版本的 Python 或者其他版本的 Python。一般来说，我们建议下载最新版本的 Python，因为最新版本的 Python 包含了最新的特性和修复了已知的问题。选择下载最新版本的 Python，我们需要点击页面中间的“Download Python”按钮，如图 2–1 所示。

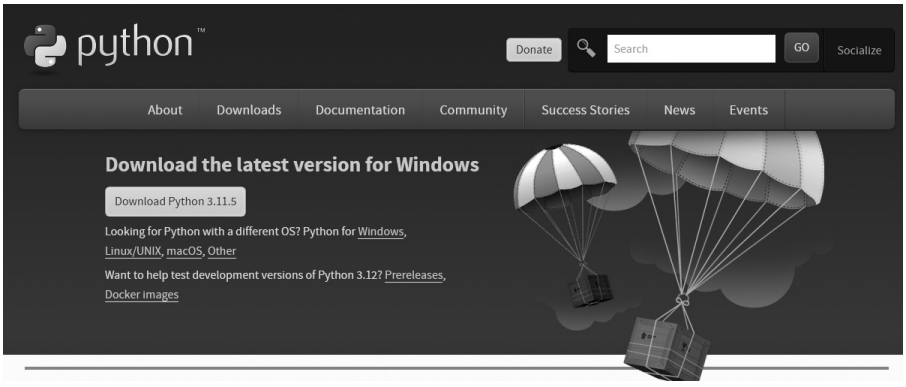


图 2–1　Python 下载界面

（2）运行 Python 安装程序

下载完成后，我们需要运行 Python 安装程序。双击刚刚下载的 Python 安装程序，会出现如图 2–2 所示的界面。在这个界面中，我们需要勾选“Add python.exe to PATH”选项，这个选项的作用是将 Python 添加到系统环境变量中，方便我们在命令行中使用 Python。勾选完成后，点击“Customize installation”按钮，从而查看更具体的安装选项。

图 2–2　Python 安装向导步骤

（3）选择 Python 安装选项

在“Customize installation”界面中，我们需要选择 Python 的安装选项。Python 的安装选项中有“for all users (requires admin privileges)”，如图 2-3 所示。如果你是单独使用这台电脑，并且是这台电脑上唯一的用户，建议去掉“□”中的“√”。如果你是和其他人共用这台电脑，或者你想让所有用户都能使用 Python，建议勾选“□”中的“√”。在选择完 Python 的安装选项后，我们需要点击“Next”按钮。

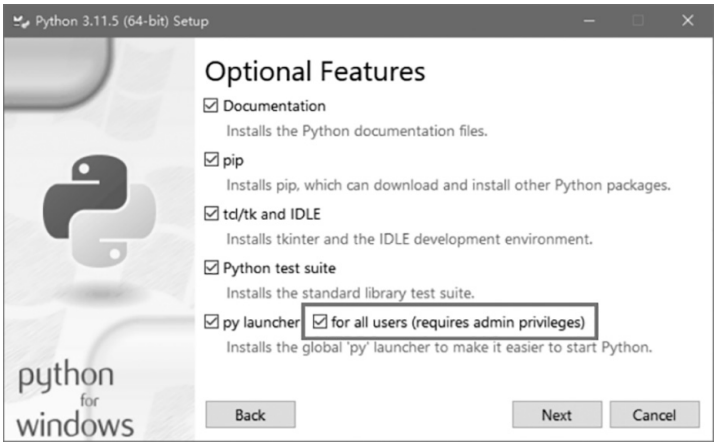


图 2-3　设置“安装选项”对话框

（4）选择 Python 安装目录

在“Customize installation”界面的下一步中，如图 2-4 所示，我们需要选择 Python 的安装目录。Python 的默认安装目录是“C:\Python3x”，其中“3x”表示 Python 的版本号。

如果你想将 Python 安装到其他目录，可以在这里进行设置。设置完毕后，点击“Install”按钮。

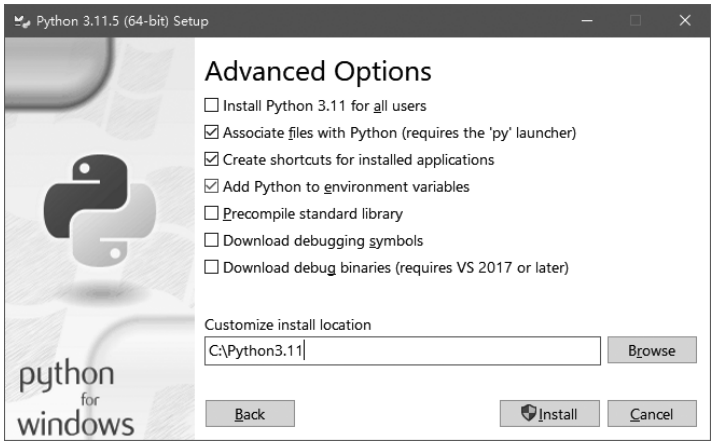


图 2-4　设置“高级选项”对话框

（5）等待 Python 安装完成

在点击“Install”按钮后，Python 的安装程序会开始安装 Python。安装过程可能需要几分钟的时间，具体时间取决于您的计算机配置和网络速度。在安装过程中，您可以看到如图 2-5 所示的程序安装进度。

请耐心等待，直到安装完成。完成后，您将看到一个如图 2-6 所示的安装完成对话框。单击“关闭”按钮以关闭对话框。至此，Python 的安装已经完成。

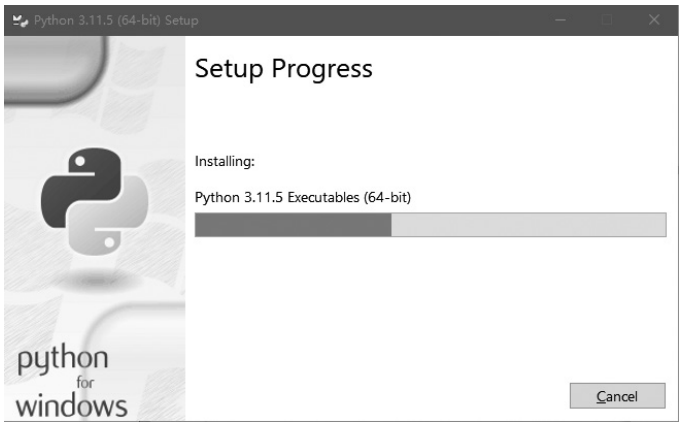


图 2–5　开始安装

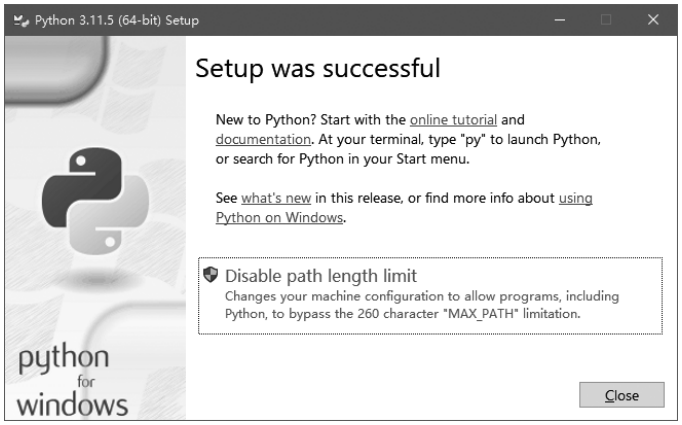


图 2–6　“安装完成”对话框

2.2.2 启动 Python

Python 启动 Python

Python 安装完成后，你需要启动 Python 来开始编写代码。在本章节中，我们将介绍两种方法来启动 Python: 使用 IDLE 集成开发环境和使用 Windows 命令行。

（1）使用 IDLE 集成开发环境

IDLE 是 Python 自带的一个简单的集成开发环境 (IDE)，它可以帮助你编写和运行 Python 代码。下面是使用 IDLE 启动 Python 的步骤。

①如图 2–7 所示，在 Python 安装目录下，找到 Lib\idlelib\idle.bat 文件，双击该文件启动 IDLE。

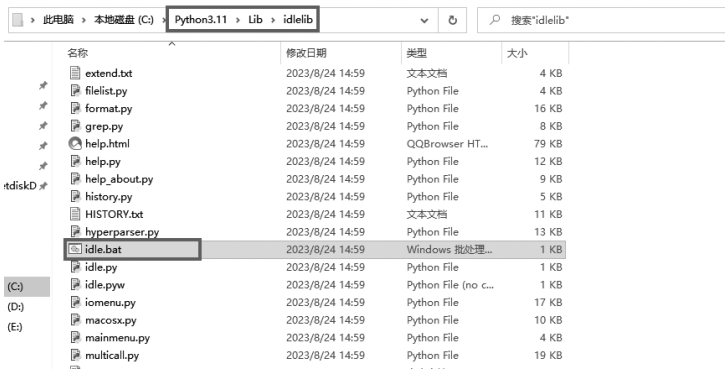


图 2–7　IDLE 文件夹内容

除了在文件夹中启动 idle.bat 文件之外，还可以在 Windows 桌面单击“开始”菜单，在出现的搜索框中输入“IDLE”来启动 IDEL。启动 IDEL 之后，可以看到如图 2-8 所示的 IDLE 主界面。

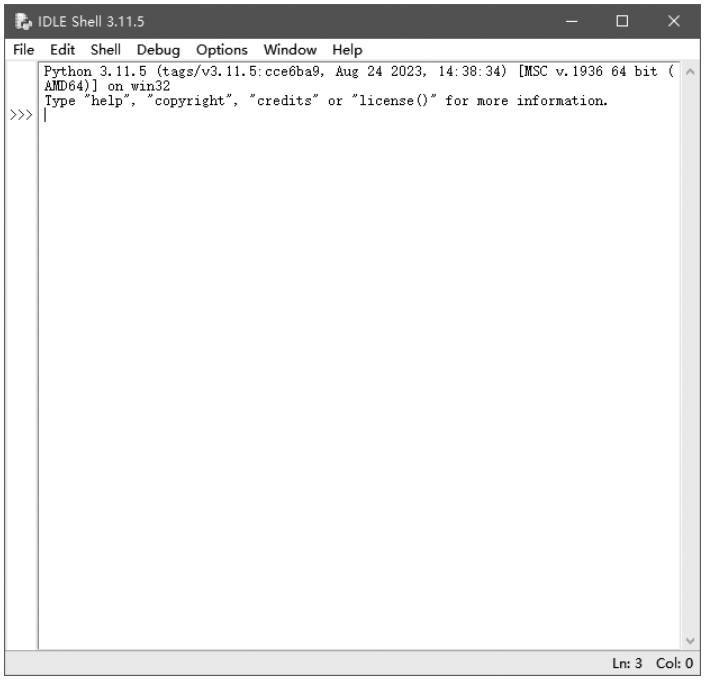


图 2-8　IDLE 主界面

②在 IDLE 界面中，选择“File”菜单，然后选择“New File”，如图 2-9 所示。

③在新建的编辑器窗口中输入你的 Python 代码，如图 2-10 所示，在此处请输入 print（'hello world'）。

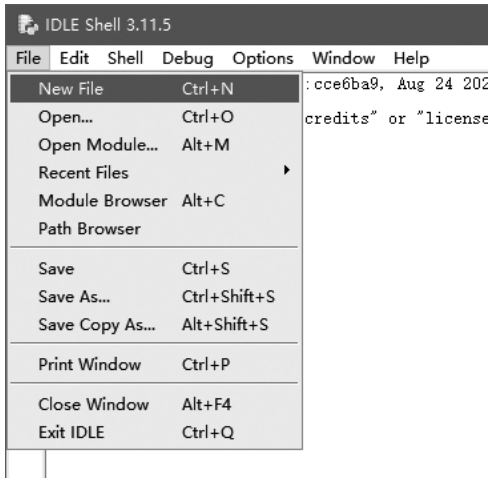


图 2-9　IDLE 的“File”菜单

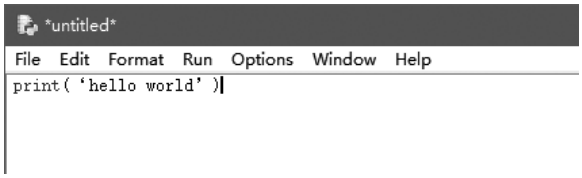


图 2-10　编写代码之后的 IDLE 窗口

④按下“F5”键或者选择“Run”菜单中的“Run Module”选项来运行你的代码。在此过程中，可能会弹出如图 2–11 所示的窗口，提示用户保存文件。在此，我们将文件保存为 demo.py。

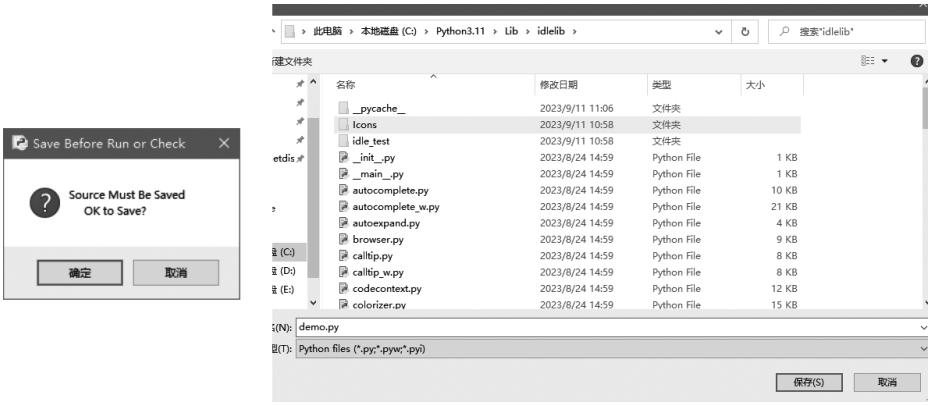


图 2–11　文件保存

⑤运行结果会在 IDLE 的 Shell 界面中显示，如图 2–12 所示。

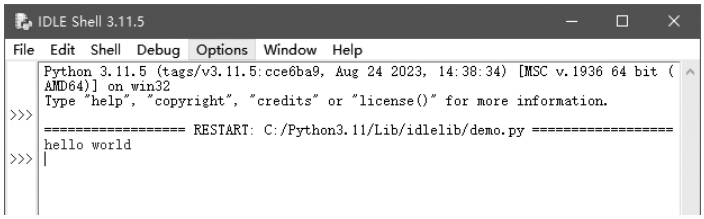


图 2–12　程序运行结果

（2）使用 Windows 命令行

你也可以在 Windows 命令行上执行 Python 代码。下面是使用 Windows 命令行启动 Python 的步骤。

①打开 Windows 命令行，可以使用“Windows 键 +R”组合键打开运行对话框，输入“cmd”并按下“Enter”键，如图 2–13 所示。

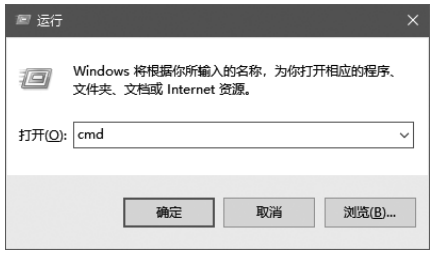


图 2–13　运行“cmd”命令

②输入“python”并按下“Enter”键，这会启动 Python 解释器，如图 2–14 所示。

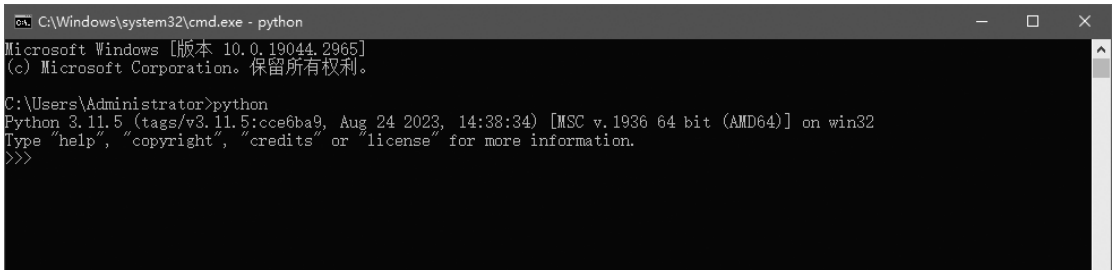


图 2–14　Python 解释器窗口

③在 Python 解释器中，输入你的 Python 代码，在此输入 print（'hello world'），如图 2-15 所示。

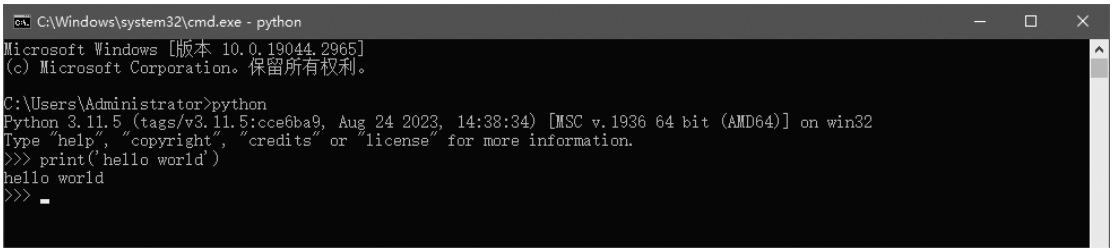


图 2-15　在 Python 解释器中编写代码

④如图 2-16 所示，可按下“Ctrl+Z”组合键或者输入“exit()”退出 Python 解释器。

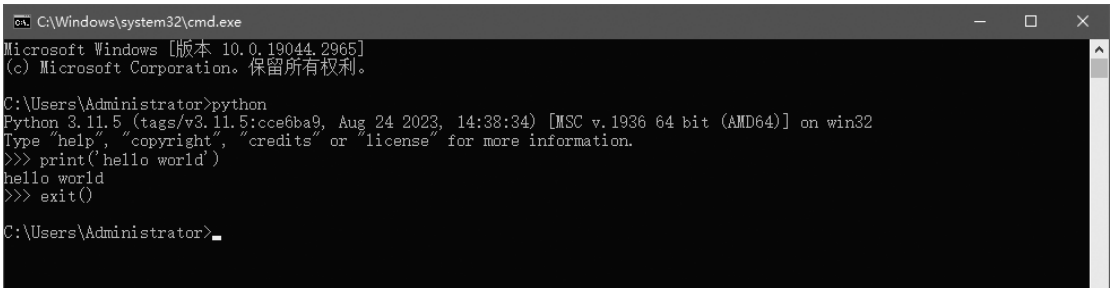


图 2-16　退出 Python 解释器

注意：如果你的 Python 安装目录不在系统的环境变量中，你需要在命令行中输入完整的 Python 安装目录路径，例如“cd C:\Python39\python.exe”。

2.2.3 执行 Python 代码文件

如果要编写一个功能比较强大的程序时，可以先把代码写到一个文件中，然后再通过 IDLE 或 Python 命令行来运行该文件。执行 Python 文件需要先创建一个 Python 文件，可在电脑上的任意位置创建，例如“C:\Python”。在“C:\Python”右键点击，选择“新建”-“文本文档”，并将其命名为“test.py”。注意，文件后缀名一定要改为“.py”，否则无法识别为 Python 文件。打开“test.py”文件，使用记事本等文本编辑器输入下列代码：

```
print("Hello，World")
```

代码作用是输出“Hello，World”这个字符串。保存“test.py”文件并关闭文本编辑器。

（1）用 IDLE 执行 Python 文件

①在 Windows 开始菜单中搜索“IDLE”，然后点击打开，如图 2-17 所示。

②在 IDLE 菜单栏中，选择“File”-“Open”，打开你创建的“test.py”文件，如图 2-18 所示。

图 2–17　查找 IDEL 应用

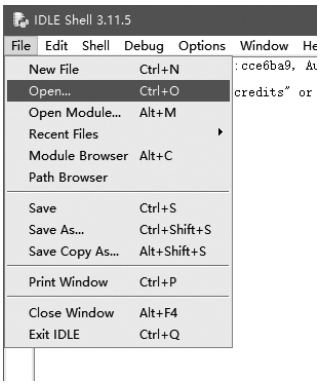


图 2–18　打开文件窗口

③在“test.py”文件的编辑窗口中，按下“F5”键或选择“Run”–“Run Module”，如图 2–19 所示。

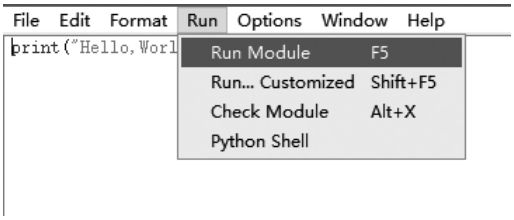


图 2–19　执行 Python 程序

④如果一切正常，你将在 IDLE Shell 窗口中看到如图 2–20 所示的输出结果。

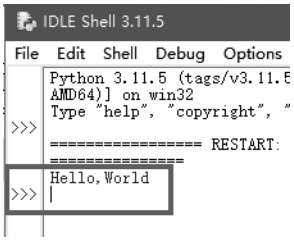


图 2–20　运行结果

如果出现任何错误信息，请检查代码是否正确，并尝试重新执行。

（2）用 Windows 命令行执行 Python 文件

①打开 Windows 命令提示符。如图 2–21 所示，在 Windows 运行菜单中搜索“cmd”，然后点击确定。

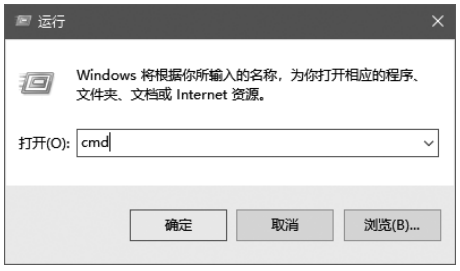


图 2–21　运行“cmd”命令

②在命令提示符中，使用“cd”命令进入“test.py”所在的目录。例如，如果“test.py”文件在桌面上，请输入以下命令“cd C:\Python”，然后按回车键，如图 2–22 所示。

图 2-22　进入文件所在目录

③输入以下命令执行“test.py”文件“python test.py”。

如果 Python 环境变量已正确配置，你将在命令提示符窗口中看到输出结果“Hello, World”，如图 2-23 所示。如果出现任何错误信息，请检查代码是否正确，并尝试重新执行。

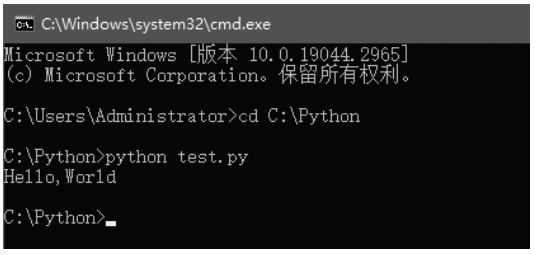


图 2-23　文件执行结果

至此，你已经成功地在 IDLE 和 Windows 命令行中执行了一个简单的 Python 文件。你可以尝试修改代码，然后重新执行，看看会有什么不同的输出结果。

2.3 PyCharm 集成开发环境

IDE 全称 Integrated Development Environment（即集成开发环境）是一种软件应用程序，用于编写、测试和调试软件程序。IDE 集成了多种工具和功能，包括代码编辑器、编译器、调试器、自动代码补全、版本控制等，能让开发人员更加高效地编写程序。

PyCharm 是一款专为 Python 开发人员设计的 IDE，由 JetBrains 公司开发。它提供了丰富的功能和工具，支持 Python 的多个版本，并且可以与其他工具集成使用。以下是 PyCharm 的一些主要功能：

①代码编辑器：提供自动代码补全、代码格式化、语法高亮等功能，可以帮助开发人员更快更准确地编写代码。

②调试器：可以帮助开发人员快速定位和解决程序中的错误。

③版本控制：支持与 Git、SVN 等版本控制工具集成，可以更好地管理和协作开发。

④智能重构：可以帮助开发人员轻松修改和优化代码结构。

⑤ Jupyter Notebook 集成：可以在 PyCharm 中直接运行 Jupyter Notebook。

⑥支持多种 Python 版本：支持 Python2 和 Python3，并提供了丰富的库和框架支持，如 Django、Flask、PyQt、Pandas 等。

2.3.1 安装 PyCharm

安装 PyCharm

JetBrains 公司的官网提供了支持 Windows、Linux 和 Mac OS 平台的 PyCharm 版本，开发者可以根据需要选择不同版本下载。针对不同平台的 PyCharm 版本的安装过程大同小异，本节以 Windows 平台上的安装为例进行介绍。

（1）步骤 1：下载 PyCharm 安装程序

首先，我们需要下载 PyCharm 的安装程序。请访问以下网址：https://www.jetbrains.com/pycharm/download/，选择 PyCharm Community 版本，并单击“Download”按钮，如图 2–24 所示 。

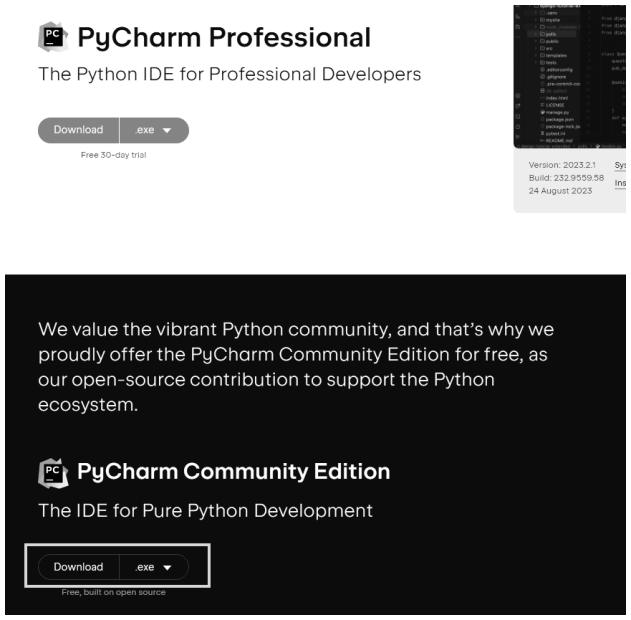


图 2–24　选择 PyCharm 版本

（2）步骤 2：运行安装程序

下载完成后，双击下载的安装程序以运行安装向导。在第一个页面上，您将看到 PyCharm 的欢迎屏幕。请单击“Next”继续，如图 2–25 所示。

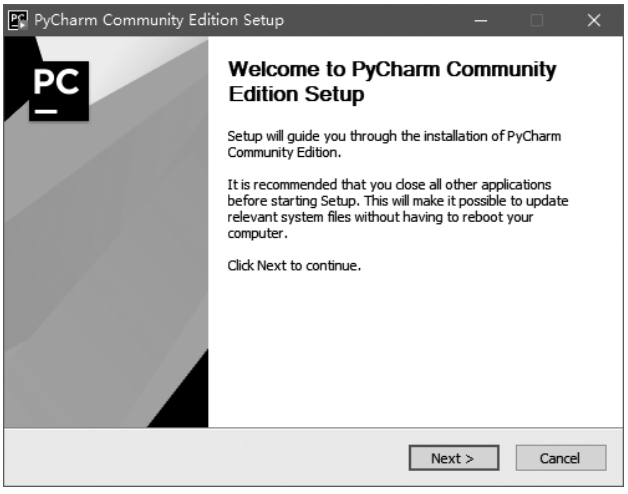


图 2–25　安装界面

（3）步骤 3：选择安装位置

如图 2-26 所示，您需要选择要安装 PyCharm 的位置。我们建议将其保留在默认位置，并单击“Next”按钮。

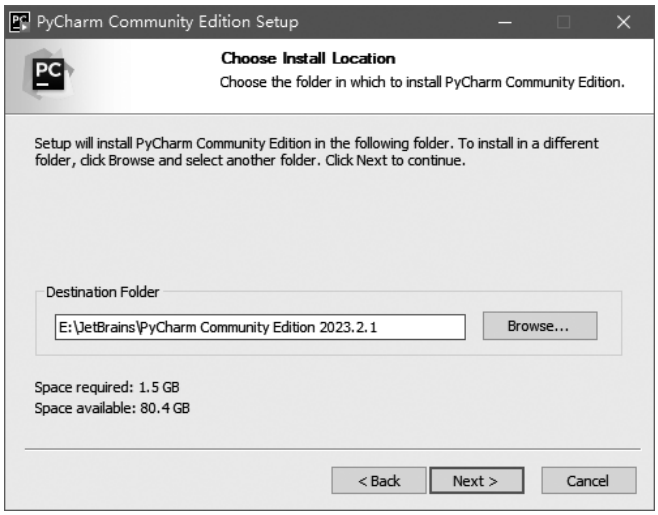


图 2-26　设置安装路径

（4）步骤 4：选择创建快捷方式

如图 2-27 所示，您需要选择是否要为 PyCharm 创建快捷方式。如果您想在桌面上创建一个快捷方式，请选中“Create desktop shortcut”选项。否则，请取消选中该选项，并单击“Next”按钮。

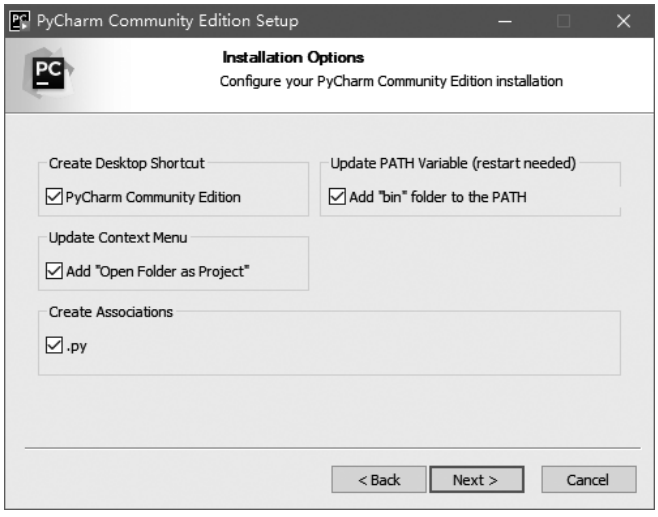


图 2-27　设置安装选项

（5）步骤 5：等待安装完成

图 2-28 所示，您可以设置开始菜单的目录。如果没有特殊的需要，建议使用默认名 JetBrains。至此，您已经完成了所有必要的配置选项。请单击“Install”按钮，开始安装 PyCharm。这可能需要几分钟的时间，具体取决于您的计算机速度和安装选项。

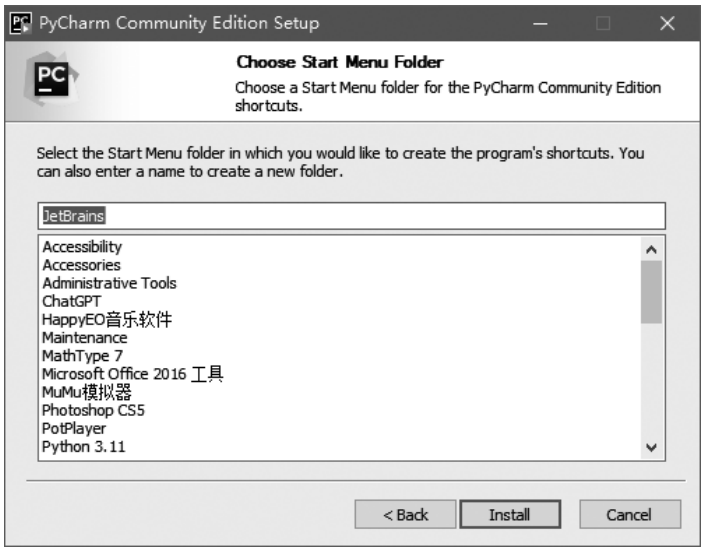


图 2-28　选择开始菜单的目录

2.3.2 在 PyCharm 创建 Python 项目

安装和配置完成后，就可以开始使用 PyCharm 编写 Python 代码了。在 PyCharm 中，可以创建新的 Python 项目，并在项目中创建 Python 文件，编辑和运行 Python 代码。同时，PyCharm 还提供了许多方便的工具和功能，如代码自动完成、调试器、测试工具等，可以大大提高编写 Python 代码的效率和质量。

①启动 PyCharm，进入如图 2-29 所示的主界面。

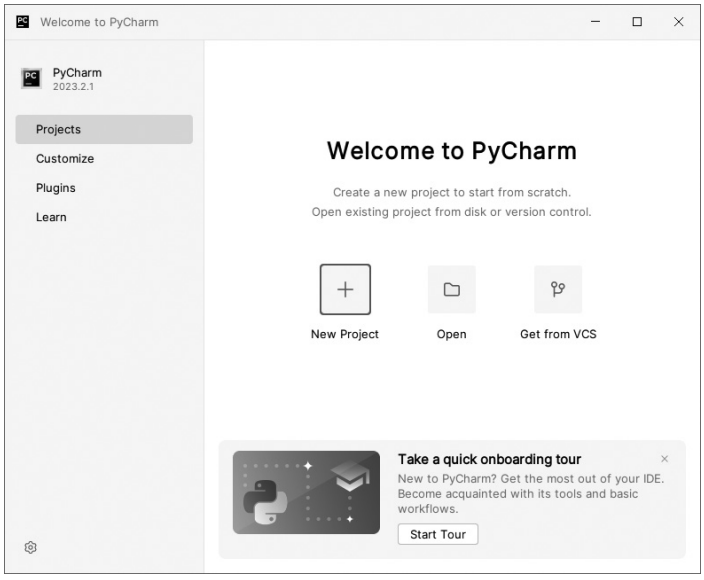


图 2-29　PyCharm 主界面

②点击在“projects”菜单项下，选择“New Project”以创建新的项目。之后，将看到如图 2-30 所示的新建项目对话框。在新建项目对话框中，可以设置项目名称、项目路径、解释器等信息。在此，建议使用已安装的 Python 解释器，点击“Create”完成创建。

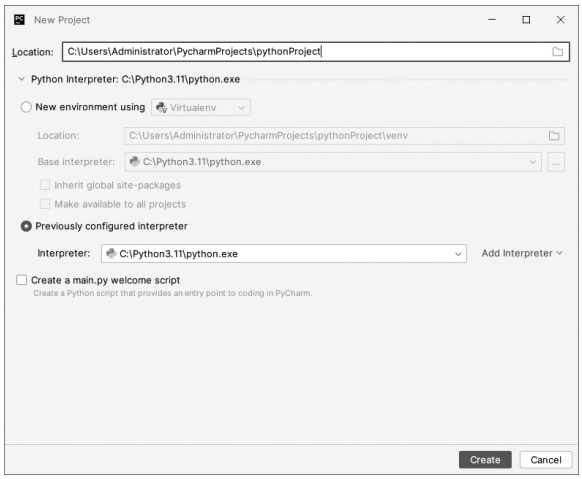


图 2-30　新建项目对话框

③在左边的“Project”窗口中，选中刚创建的项目，并右键单击鼠标，选择“New”->“Python File”，输入文件名称，如“Demo.py”。如图 2-31 所示。

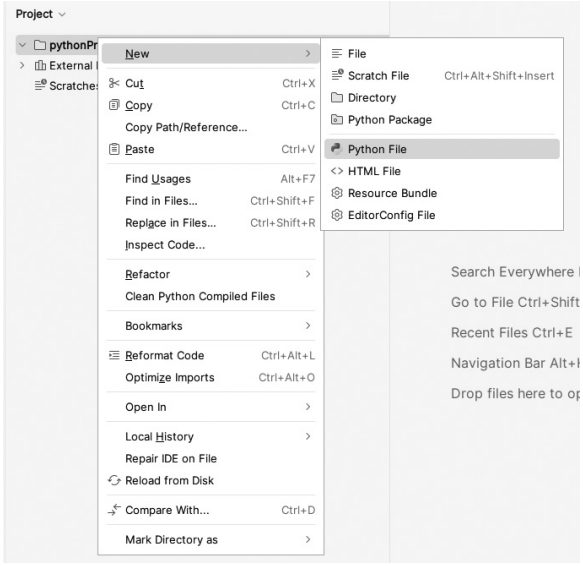


图 2-31　新建 Python 文件

④如图 2-32 所示，在新建的 Python 文件中，输入代码“print（'Hello World'）”。

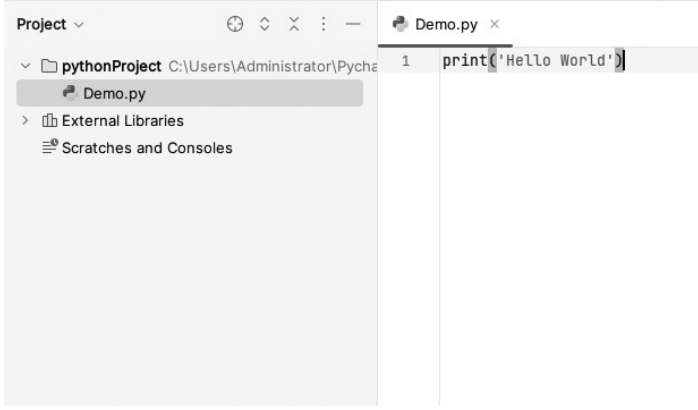


图 2-32　在 PyCharm 编写代码

⑤如图 2–33 所示，按下快捷键“Shift + F10”，或者在文件的编辑界面上右键单击鼠标，选择“Run 'Demo'”，即可运行代码。

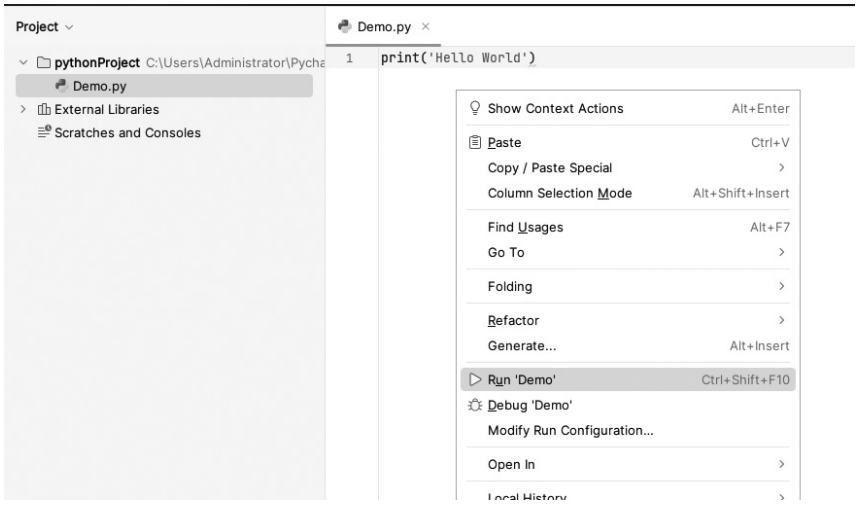


图 2–33　执行程序

⑥在下方的“Run”窗口中，可以看到如图 2–34 所示的运行结果。

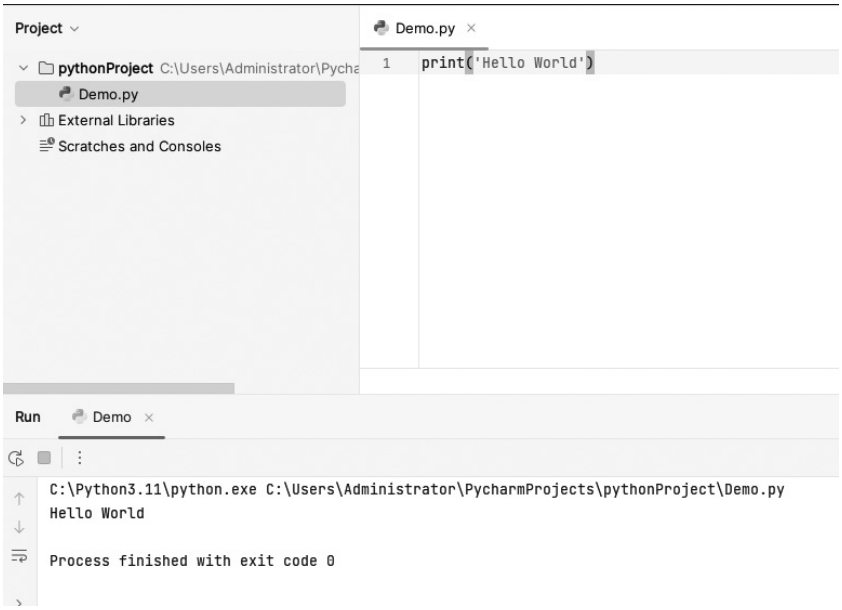


图 2–34　程序输出结果

这样，我们就成功地在 PyCharm 中创建了一个 Python 项目，编写并执行了一段 Python 代码。

2.3.3 在 pycharm 中管理第三方库

在 PyCharm 中可以通过包管理工具 pip 或 PyCharm 提供的可视化界面管理第三方库。以下分别介绍这两种方式的操作过程。

（1）使用 pip 管理第三方库

打开 PyCharm，打开你的项目，点击主界面下方的“Terminal”，或者使用快捷键 Alt+F12 打开终端，便可看到如图 2–35 所示的 Terminal 终端。

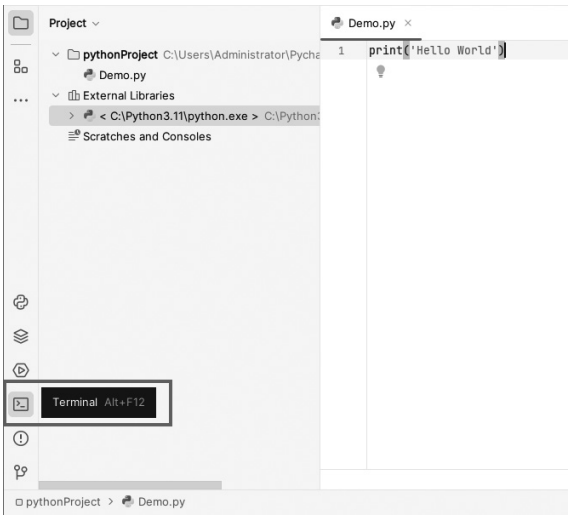


图 2-35　Terminal 终端

在终端中输入以下命令来安装一个示例库 requests：

```
pip install requests
```

安装过程中可以看到如图 2-36 所示的安装信息，如安装版本、安装路径等。安装完成后，可以在 PyCharm 的项目中看到 requests 库已被添加到项目中的 External Libraries 中（方框中代表安装版本）。

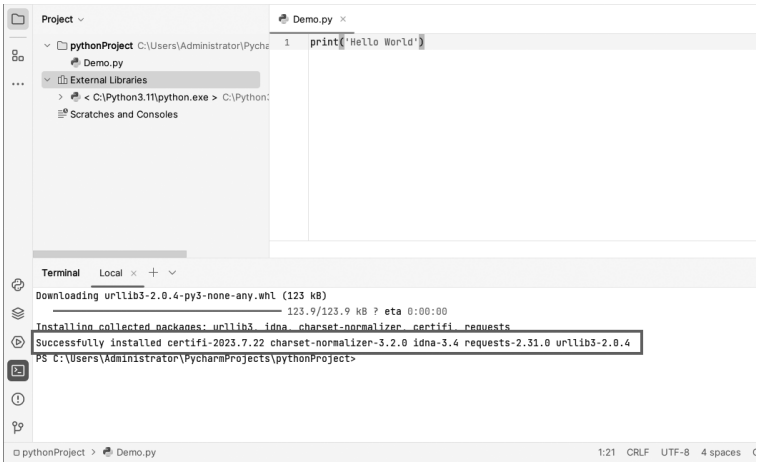


图 2-36　安装 requests 库

如果需要更新一个已安装的库，可以使用以下命令：

```
pip install --upgrade requests
```

执行完毕后，requests 库会被更新到最新版本。

如果需要卸载一个已安装的库，可以使用以下命令：

```
pip uninstall requests
```

执行完毕后，requests 库将从项目中移除。这样，就可以在 PyCharm 中轻松地管理第三方库了。

（2）通过可视化界面管理第三方库

PyCharm 提供了可视化界面来安装、卸载、更新和管理第三方库。

①在 PyCharm 中打开项目，例如打开之前创建的 pythonProject，如图 2–37 所示。

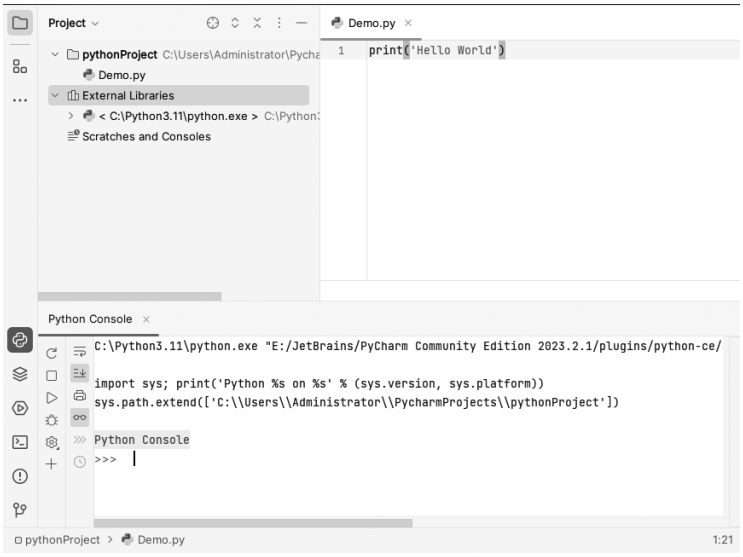


图 2–37　打开 pythonProject 项目

②在菜单栏中选择 File –> Settings，可看到如图 2–38 所示的设置窗口。

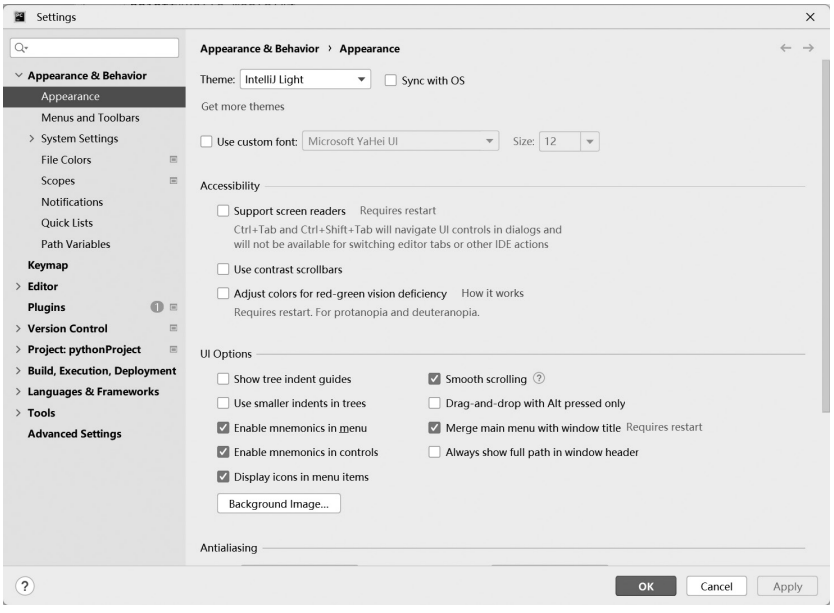


图 2–38　Setting（设置）窗口

③在设置窗口中，选择 Project: < 项目名 > –> Python Interpreter。其中，项目名将根据您选所打开的项目有所不同。如图 2–39 所示，您可以查看到 Python 解释器中已安装的库列表。

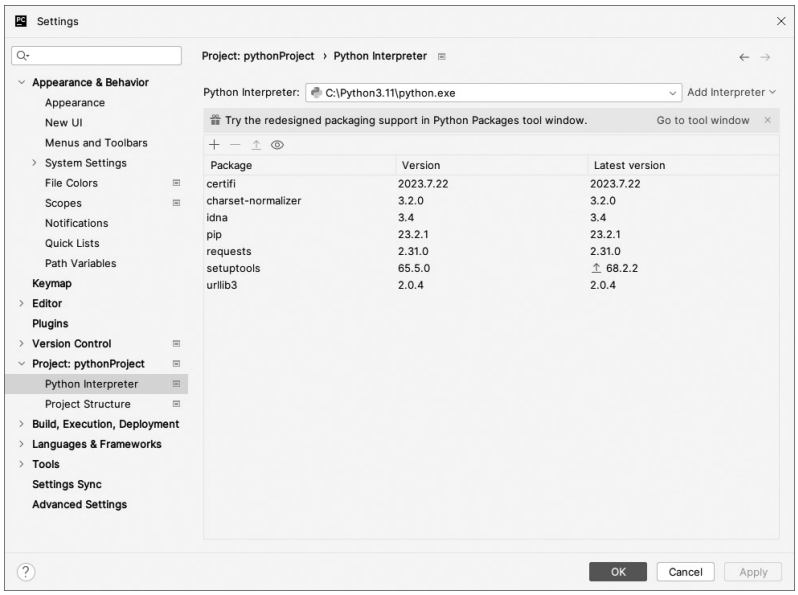


图 2–39　已安装的库列表

④要安装新的第三方库，点击右上角的“+”按钮，搜索你需要的库。这里以安装 numpy 库为例。在搜索框中输入 numpy，在搜索结果中选择 numpy 库，并点击“Install Package”，如图 2–40 所示。

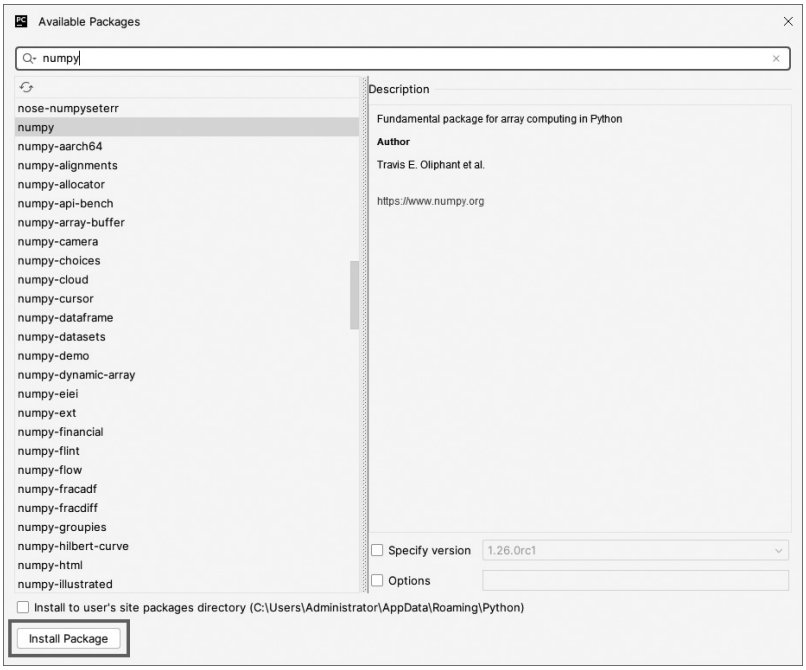


图 2–40　安装 numpy

⑤等待安装完成，你可以在 PyCharm 的控制台中看到安装的过程。

⑥安装完成后，如图 2–41 所示，您可以在 Python 解释器的已安装库列表中看到 numpy 库。

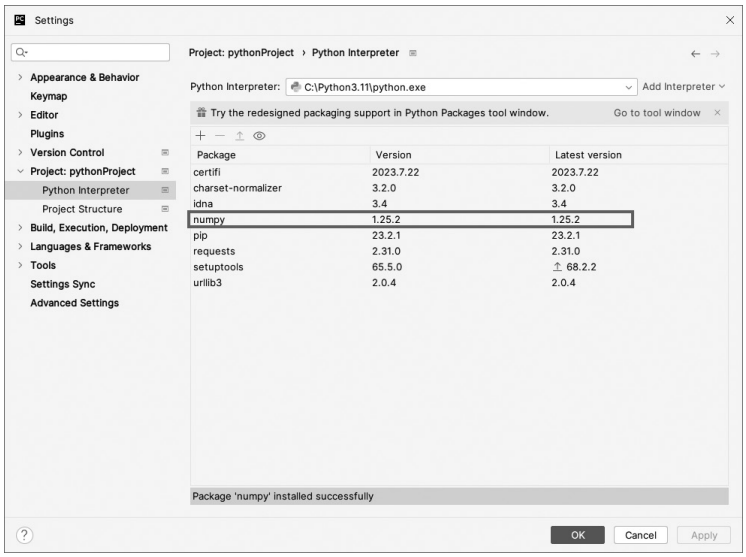


图 2–41　已安装 numpy 库

⑦如图 2–42 所示，当您需要卸载库时，选择要卸载的库并点击“–”按钮，并等待卸载完成。

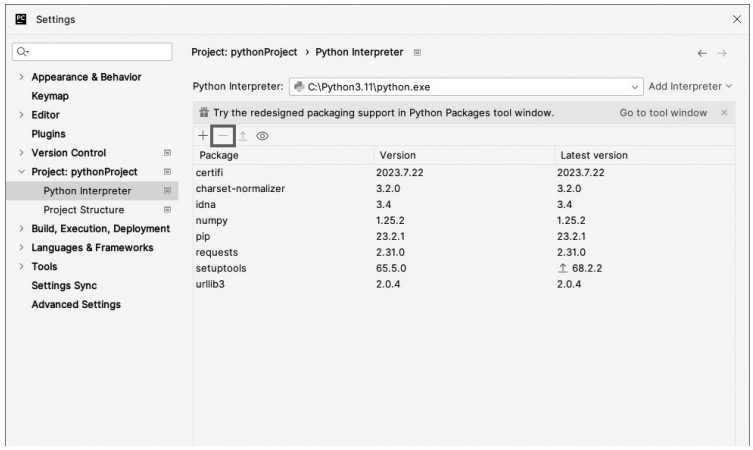


图 2–42　卸载库

⑧如图 2–43 所示，如果您需要更新已安装的库，选择要更新的库并点击右边的“Upgrade Package”，并等待更新完成。

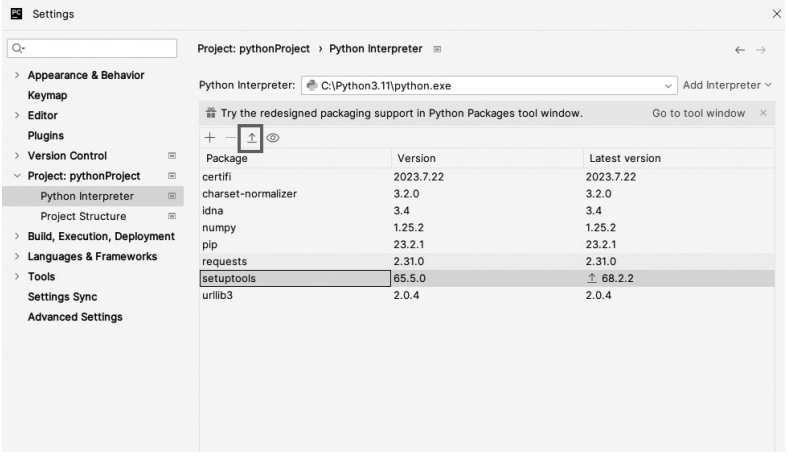


图 2–43　更新库

2.4 科学计算工具

IPython 是公认的现代科学计算中最重要的 Python 工具之一，它是一个加强版的 Python 交互式命令行工具，与系统自带的 Python 交互环境相比，IPython 主要具有以下特点：

①与 Shell 紧密关联，可以在 IPython 开发环境下直接执行 Shell 指令。

②它是可以直接进行绘图操作的 Web GUI 环境，在机器学习领域、探索数据模式、可视化数据、绘制学习曲线时，功能都非常强大。

③更强大的交互功能，包括内省、Tab 键自动完成、魔术命令等。

2.4.1 安装 IPython

安装 IPython 很简单，直接使用 pip 命令即可。在计算机的搜索框中输入 cmd，打开“命令提示符”窗口，输入“pip install ipython”命令，按“Enter”键开始安装 IPython，如图 2-44 所示。

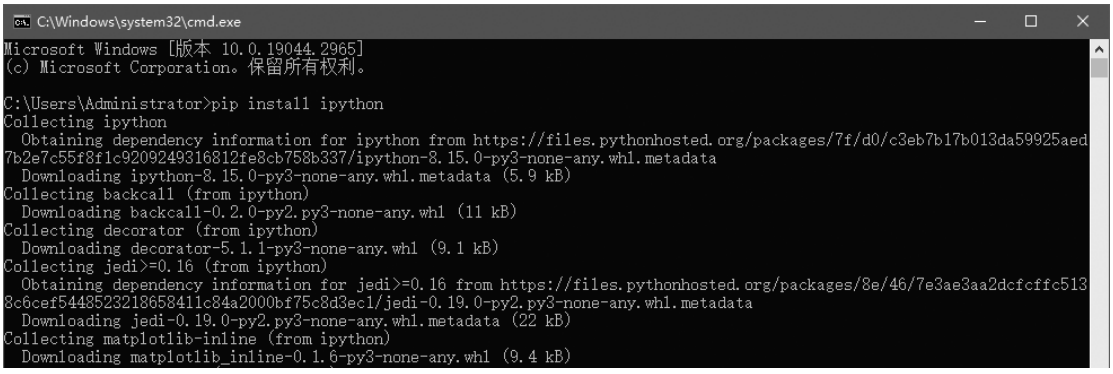


图 2-44　安装 IPython

2.4.2 运行 IPython

运行 IPython 非常简单，使用组合键“win + R”（win 键位置如图 2-45 所示），打开“运行”窗口，在此输入 ipython，如图 2-46 所示，单击“确定”按钮打开 IPython，如图 2-47 所示。

图 2-45　win 键

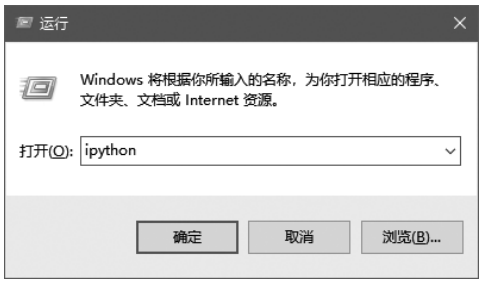


图 2-46　运行窗口

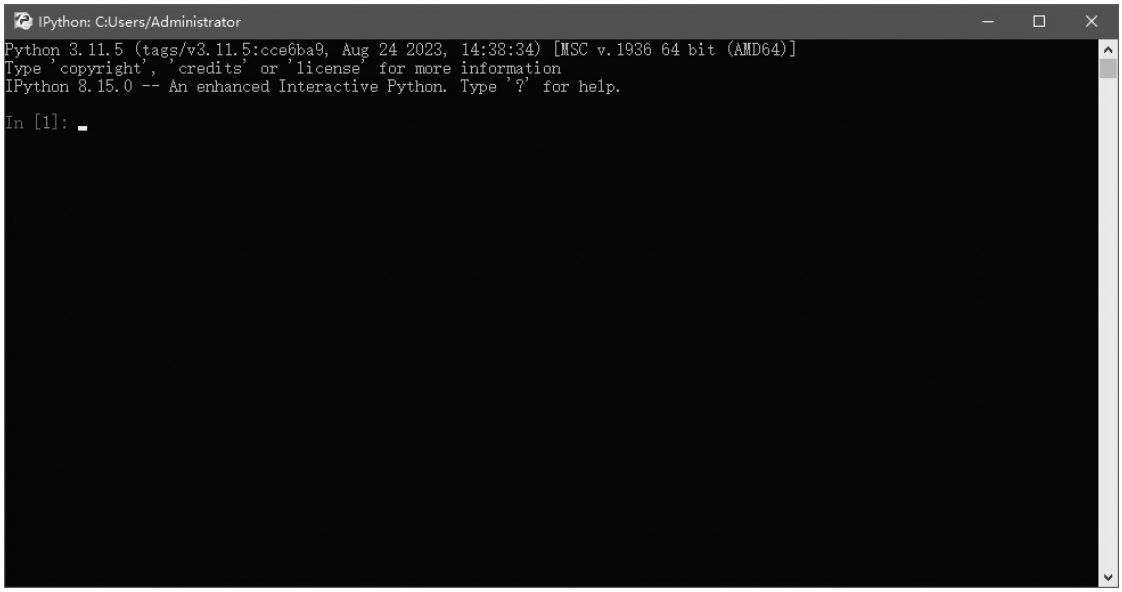


图 2-47　运行 IPython 界面

2.4.3 在 IPython 中编写“Hello World”

首先打开 IPython，使用组合键“win + R”打开“运行”窗口，在此输入 ipython，然后在命令提示符下输入代码“print（'Hello World'）”，按“Enter”键运行程序，结果如图 2-48 所示。

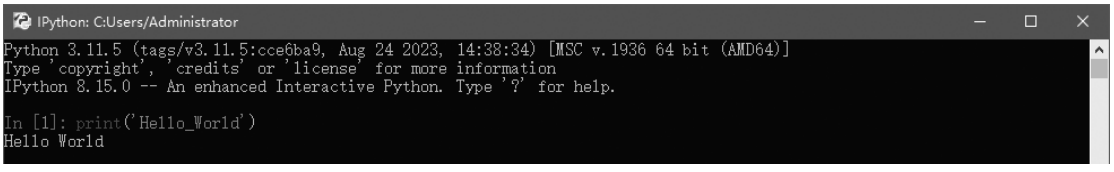


图 2-48　编写“Hello World”

2.4.4 Tab 键自动搜索

在 shell 中输入表达式时，只要按下 Tab 键，与当前输入内容相匹配的方法、函数、对象等就会被找出来。例如，通过 Pandas 模块获取 Excel 数据时，读者突然忘记用哪个方法了，此时按下“Tab”键，则相关方法都将被找出来，如图 2-49 所示。

```
In [11]: import pandas as pd

In [12]: pd.re_
read_clipboard() read_fwf()       read_json()      read_sas()       read_stata()
read_csv()       read_gbq()       read_msgpack()   read_sql()       read_table()
read_excel()     read_hdf()       read_parquet()   read_sql_query() reset_option
read_feather()   read_html()      read_pickle()    read_sql_table()
```

图 2-49　Tab 键自动搜索

2.4.5 内省（帮助）功能

在变量的前面或者后面加上一个问号“?”，就可以将有关该对象的一些通用信息显示出来，这就叫作对象的内省。例如，创建一个列表 a，然后输入“a?”，将输出列表 a 的相关信息，如类型、列表元素和长度等，如图 2-50 所示。

```
In [13]: a=[1,2,3]

In [14]: a?
Type:        list
String form: [1, 2, 3]
Length:      3
Docstring:
Built-in mutable sequence.

If no argument is given, the constructor creates a new empty list.
The argument must be an iterable if specified.
```

图 2-50　输出列表 a 的相关信息

如果使用两个问号“??”，则显示源代码。另外，还可以使用通配符字符串查找所有与该通配符字符串相匹配的名称。

2.4.6 IPython 常用的魔法命令

IPython 常用的魔法命令如下：

① %run：运行外部 Python 文件。在 IPython 环境中，所有文件都可以通过 %run 命令当作 Python 程序来运行，输入“%run *.py”即可（默认是当前目录）。例如，运行 demo.py 文件，效果如图 2-51 所示。

```
In [20]: %run demo.py
Hello World
```

图 2-51　运行 demo.py 文件

② %hist：历史命令。简单地使用上、下翻页键就可以查看所有的历史输入。

③ %timeit：用于快速测试代码的运行时间。

④ %debug：用于在程序异常点启动调试器，也可以使用 %pdb 命令激活 IPython 调试器。这样，每当异常抛出时，调试器就会自动运行。

⑤ %pylab：魔法命令。它可以使得 Numpy 和 Matplotlib 中的科学计算功能生效，这些功能被称为基于向量和矩阵的高效操作，具有交互可视化的特性。它能够让我们在控制台进行交互式计算和动态绘图。

⑥ %paste：用于直接粘贴一段代码，前提是先复制一段代码。%paste 的执行顺序是：先将代码打

印出来，然后再执行该段代码。

⑦ %lsmagic：用于获取更多的魔法命令。

2.4.7 直接执行 Shell 命令

在 IPython 环境中可以直接执行 Shell 命令，在 Shell 命令前加上叹号“!”即可。例如，测试百度网络连接（ping 百度，即 !ping baidu.com），效果如图 2–52 所示。

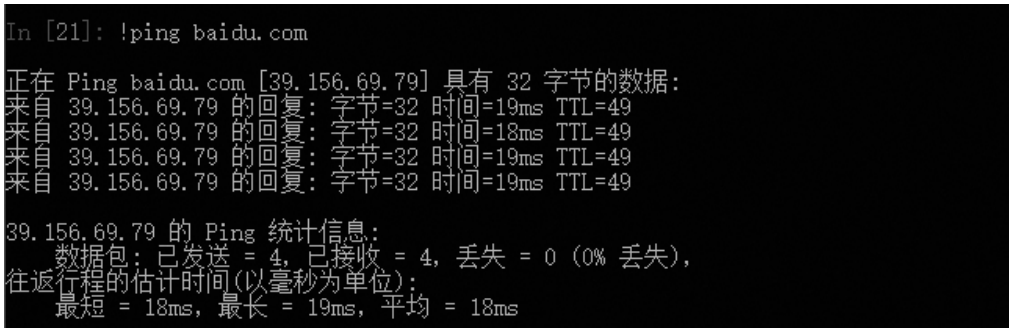


图 2–52　ping 百度效果

本章小结

本章介绍了多款 Python 开发工具，如 Python IDLE、集成开发环境 PyCharm、适合数据分析的标准环境和科学计算工具 IPython 等。但是，这里建议大家有选择性的学习，尤其对于初学者来说，学会使用 Python 自带的 IDLE 和集成开发环境 PyCharm 即可。由于本书采用的开发环境是 PyCharm，所以建议首先学习 PyCharm，对于其他开发工具初步了解就可以。

章 节 练 习

一、填空题

1.Python 是一种_____语言，由 Guido van Rossum 于_____年底发明，并在_____年发布了第一个公开版。

2.Python 最初的设计哲学是作为一种_____、_____的语言，具有简洁而清晰的语法，易于_____。

3.Python 是一种_____语言，这意味着 Python 代码不需要_____就可以直接执行。Python 的解释器可以在_____上运行，并且可以轻松地与其他语言集成。

二、简答题

1. 简述搭载 Python 运行环境的方法。

2. 简述 PyCharm 开发环境的安装方式。

第 3 章
Python 基础与数据抓取

学习目标

知识目标

- ◆掌握数据的结构及数据使用方法。
- ◆理解 Python 中的控制语句。

能力目标

- ◆掌握字符串的处理方法。
- ◆明确自定义函数的使用方法。

素质目标

- ◆提高逻辑思维与程序思维。
- ◆加强整体意识与框架意识。

思政目标

强调遵守数据法规，坚决抵制非法获取和使用数据。