

第 1 章 初识 Python

Python 语言是一种高级语言，是面向对象、解释性的程序设计语言，具有语法简洁、易于学习、功能强大、可扩展性强、跨平台等特点。本章的目标是了解 Python，能安装配置 Python 开发环境并掌握 pip、PyInstaller 工具的使用。



教学导航

| | |
|-------------|--|
| 学习目标 | <ol style="list-style-type: none">1. 了解 Python 发展历史及应用领域2. 掌握 Python 环境搭建3. 了解常用的 Python 开发环境4. 掌握 pip 包管理工具5. 掌握 PyInstaller 打包发布方法 |
| 教学重点 | IDLE 开发环境、pip 包管理工具、PyInstaller 打包发布方法 |
| 教学方式 | 案例教学法、分组讨论法、自主学习法、探究式训练法 |
| 课时建议 | 4 课时 |



1.1 认识 Python

本节介绍 Python 的发展历程、特点及主要应用领域，同时从数据处理分析这一角度给出了 Python 学习线路。

【学习目标】

- 小节目标**
1. 了解 Python 发展历史
 2. 了解 Python 的特点及应用领域
 3. 掌握 Python 的学习方法

1.1.1 Python 初识

1. Python 入门

对于大多数程序语言，第一个入门编程实例便是输出"Hello World！"，Python 3.0+版本已经把 `print` 作为一个内置函数，正确输出"Hello World！"代码极其简单，只需要一行：

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

2. Python 的发展

Python 是一种解释性、面向对象、动态数据类型的高级程序设计语言。

Python 是由 Guido van Rossum 在 20 世纪 80 年代末到 90 年代初，在荷兰国家数学和计算机科学研究所设计出来的。

Python 本身也是由诸多其他语言发展而来的，这包括 ABC、Modula-3、C、C++、Algol-68、SmallTalk、UNIX shell 和其他的脚本语言等。

像 Perl 语言一样，Python 源代码同样遵循 GPL（GNU General Public License）协议。

现在 Python 由一个核心开发团队维护，Guido van Rossum 在其中仍然占据着至关重要的地位，指导其进展。

1.1.2 Python 的特点及应用领域

1. Python 的特性

Python 是一个高层次地结合了解释性、编译性、互动性和面向对象的脚本语言。

Python 的设计具有很强的可读性，相比其他语言经常使用英文关键字，以及其他语言的一些标点符号，它具有比其他语言更有特色的语法结构。

- Python 是一种解释性语言，开发过程中没有编译这个环节，类似于 PHP 和 Perl 语言。对于 Python 而言，源代码不需要编译成二进制代码，可以直接从源代码运行程序。Python 解释器将源代码转换为字节码，然后把字节码转发到 Python 虚拟机（PVM）上执行。字节码是特定用于 Python 的一种表现形式，需要在 PVM 中进一步编译执行。

- Python 是交互式语言，可以在一个 Python 提示符后直接互动编程。

- Python 是面向对象语言。Python 支持面向对象的风格或代码封装在对象中的编程技术。

- Python 适合初学者。对初级程序员而言，是一种简单易懂的语言，它支持广泛的应用程序开发，从简单的文字处理到网页浏览器再到游戏。

Python 的功能特性如下：

- 其标准库提供了各种功能，旨在简化复杂应用程序的实现。

- 高级编程语言。使用较少的代码执行基本任务，语言简洁，只有少量的语法约束。

- 如果与 Java 相比，Python 的代码量是 Java 的 1/3~1/5；与 C++ 相比，它的代码量为 C++ 的 1/10~1/5。

- Python 语言通过强制缩进保证程序可读性。

- Python 语言具有丰富的数据结构（类型）。Python 语言在多数程序设计语言的基础上，增加了列表、字典、元组、集合等数据结构。

- Python 语言具有可移植性。

- Python 语言支持多种类型，支持面向过程和面向对象，还支持灵活的编程模式，如面向对象、命令式和函数式编程。

- 其内存管理由系统自动完成。

对于初学者来说，Python 比 Java、C++ 等传统静态语言更具实用性，更容易上手；对于有一定编程基础的开发者来说，学会了 Java、C++、C# 等语言，再反过来学习 Python 可以称得上小菜一碟了。

以上介绍的功能特性也是 Python 的优点。那么 Python 是否有缺点？答案必然是肯定的。它的缺点主要是执行速度不够快、语句只能单行显示、强制缩进等小问题，和它的优点相比，几乎可以忽略不计。

1.1.3 Python 的学习方法

1. Python 计算生态

Python 计算功能强大，得益于众多的第三方库。常用的第三方库有网络爬虫、数据分析、数据可视化、机器学习、Web 开发等。

2. Python 学习方法

Python 能进行桌面程序、网络网站、嵌入式等多种开发。不同的开发方向有不同的侧重点。用 Python 完成项目，编写的代码量更少，代码简短可读性强，团队协作开发时读别人的代码速度会非常快，使工作变得更加高效。因此，Python 应用范围广泛，学习方法也多种多样。本书以 Python 应用最广泛的数据采集、分析学习方法为例简要介绍 Python 的学习方法。

Python 从基础到数据处理、分析，再到机器学习、深度学习，学习方法一般如步骤 1 到步骤 6 所示。本教材只讲最基础的部分，即步骤 1 到步骤 4 前期基础这几部分，步骤 4 和步骤 5，学有余力的同学可以进行相应学习。步骤 6 仅提供参考，如果立志于大数据技术与人工智能，须重点学习步骤 6 所提的相关内容。

第 1 步：设置机器环境

设置机器环境，最简单的方法就是从 Continuum.io 上下载分发版 Anaconda。Anaconda 将应用 Python 编程可能会用到的大部分内容进行了打包。

第 2 步：学习 Python 语言的基础知识

了解 Python 语言的基础知识、库和数据结构。能轻松地利用 Python 写一些小脚本，同时也能理解 Python 中的类和对象。

重点学习内容：列表 List、元组 Tuple、字典 Dictionary、集合 Set、列表推导式、字典推导式。

第 3 步：学习 Python 语言中的文件处理、数据库操作、第三方库的使用。

了解了以上几个方面，就可以进行数据采集，编写爬虫程序进行网页数据采集。数据采集是数据处理分析的基础。有人把数据采集当作大数据技术的一部分，但在更多的分类中，

把数据采集归并于人工智能范畴。在数据采集区，需要处理文本数据，其中数据预处理中涉及各个处理步骤会是不小的挑战。

第 4 步：学习 Python 中的科学库——NumPy, Scipy, Matplotlib

从这步开始，学习旅程将会变得有趣了。下边是对各个库的简介，可以进行一些常用的操作：

(1) 根据 NumPy 教程进行完整的练习，特别要练习数组 `arrays` 的应用，这将会为后边的学习旅程打好基础。

(2) 接下来学习 Scipy 教程。看完 Scipy 介绍和基础知识后，可以根据自己的需要学习剩余的内容。

(3) Matplotlib 是一个 Python 的 2D 绘图库，通过 Matplotlib，开发者可以仅编写几行代码，便可以生成直方图、功率谱、条形图、错误图、散点图等。

第 5 步：学习 Pandas，进行数据可视化

Pandas 为 Python 提供 DataFrame 功能（类似于 R）。这也是在数据分析领域需要花比较多的时间练习的地方。Pandas 会成为所有中等规模数据分析的最有效的工具。Pandas 也是数据预处理的主要工具。

第 6 步：了解 Scikit-learn 等 Python 库和机器学习的内容

Scikit-learn 是机器学习领域最有用的 Python 库。需要学习机器学习的基本知识，了解回归、决策树、整体模型等监督算法及聚类等非监督算法。

完成以上步骤，再勤加练习，你就已经完成了 Python 数据分析整个学习旅程，已经学会了需要的所有技能。如果想更进一步，可以再进行深度学习。



1.2 搭建环境

Python 可应用于多平台，包括 Windows、Linux 和 Mac OS X。通过终端窗口输入“python”命令来查看本地是否已经安装 Python。

Python 可以在下述所列举的系统 and 平台上运行。

- UNIX (Solaris, Linux, FreeBSD, AIX, HP/UX, SunOS, IRIX 等)
- Windows XP/ 7/ 10/2008
- Macintosh (Intel, PPC, 68K)
- OS/2
- DOS (多个 DOS 版本)
- PalmOS
- Nokia 移动手机
- Windows CE
- Acorn/RISC OS
- BeOS
- Amiga
- VMS/OpenVMS

- QNX
- VxWorks
- Psion

Python 同样可以移植到 Java 和 .NET 虚拟机上。

【学习目标】

- 小节目标
1. 了解 Python 下载
 2. 了解并掌握 Python IDLE 开发环境的安装与使用
 3. 了解 PyCharm 开发环境
 4. 了解 Anaconda3 环境的安装与使用

1.2.1 Python 下载

Python 最新源码、二进制文档、新闻资讯等可以在 Python 的官网查看到。

Python 官网：<http://www.python.org/>

在链接中下载 Python 的文档，下载 HTML、PDF 和 PostScript 等格式的文档。

Python 文档下载地址：www.python.org/doc/

打开 Python 官网，如图 1-2-1 所示。

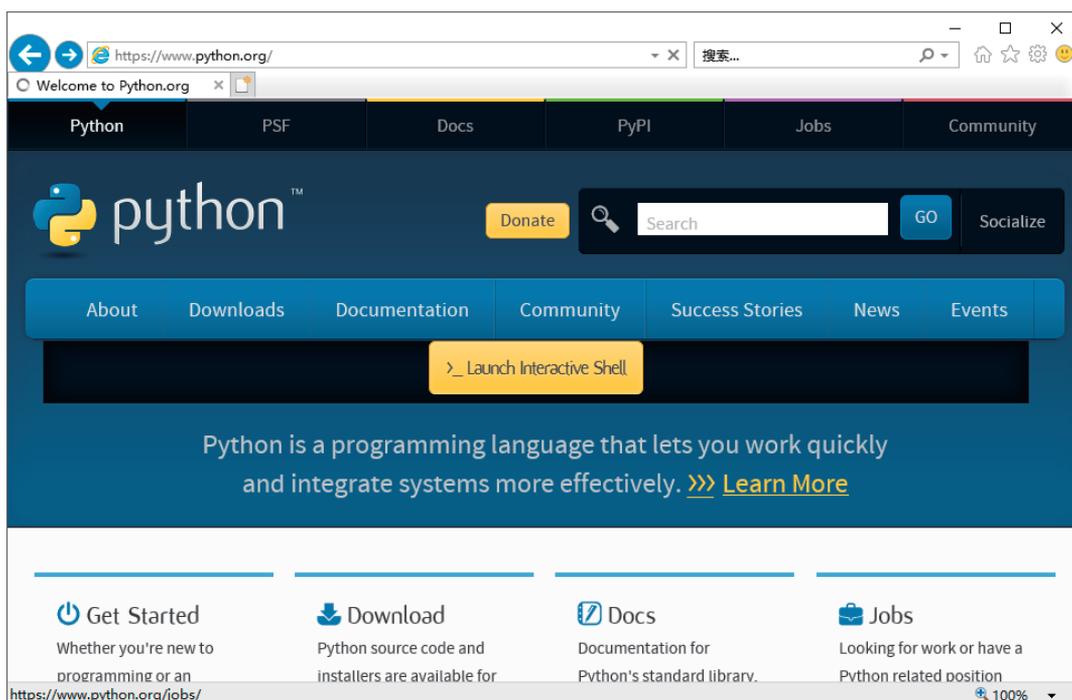


图 1-2-1 Python 官网主页

单击“Downloads”，弹出下载页面，如图 1-2-2 所示。

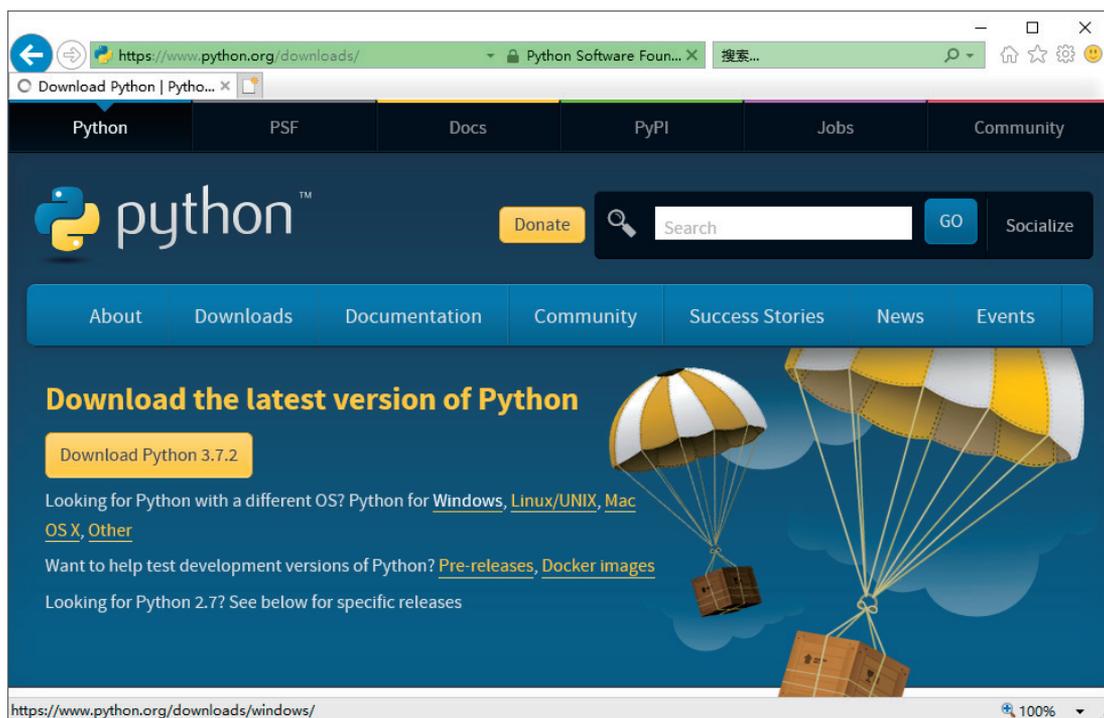


图 1-2-2 Python 下载页

单击图中的“Download Python 3.7.2”按钮，在弹出的页面中，滚动翻页找到具体版本，如图 1-2-3 所示。

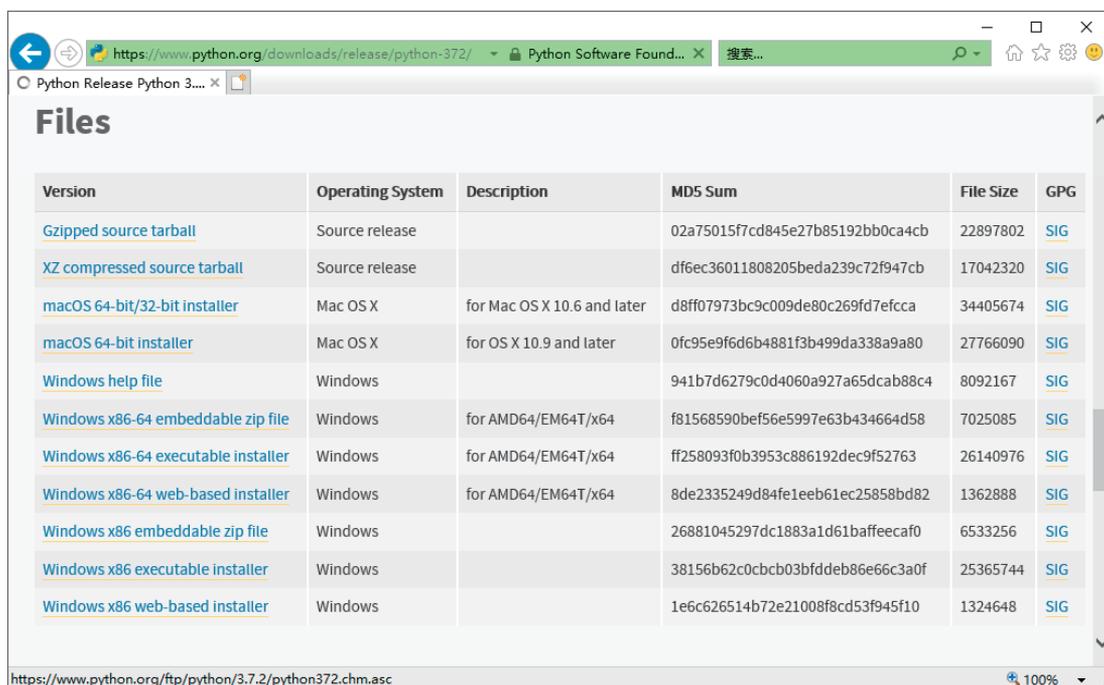


图 1-2-3 Python 具体版本页

本教材所用软件操作系统是 Windows 10，64 位操作系统，选择“Windows x86-64 executable installer”。如果是 32 位操作系统，选择“Windows x86 executable installer”，选择后按提示安装。此下载文件安装后的环境为教材所用的 IDLE 开发环境。建议下载最新版本。

1.2.2 Python IDLE 开发环境

Python 提供了交互式命令行操作环境，可以一边输入程序一边运行程序。开发时使用的是 Python 中的 IDLE。IDLE 是开发 Python 程序的基本 IDE（集成开发环境），具备基本的 IDE 的功能，初学者可以利用它方便地创建、运行、测试 Python 程序。

安装了基本的 Python 程序，就可以启动 IDLE 的交互式解释器工具 Python3.7.2 shell。从“开始”菜单→“所有程序”→“Python 3.6”→“IDLE（Python GUI）”来启动 IDLE。

IDLE 启动后的初始窗口如图 1-2-4 所示。

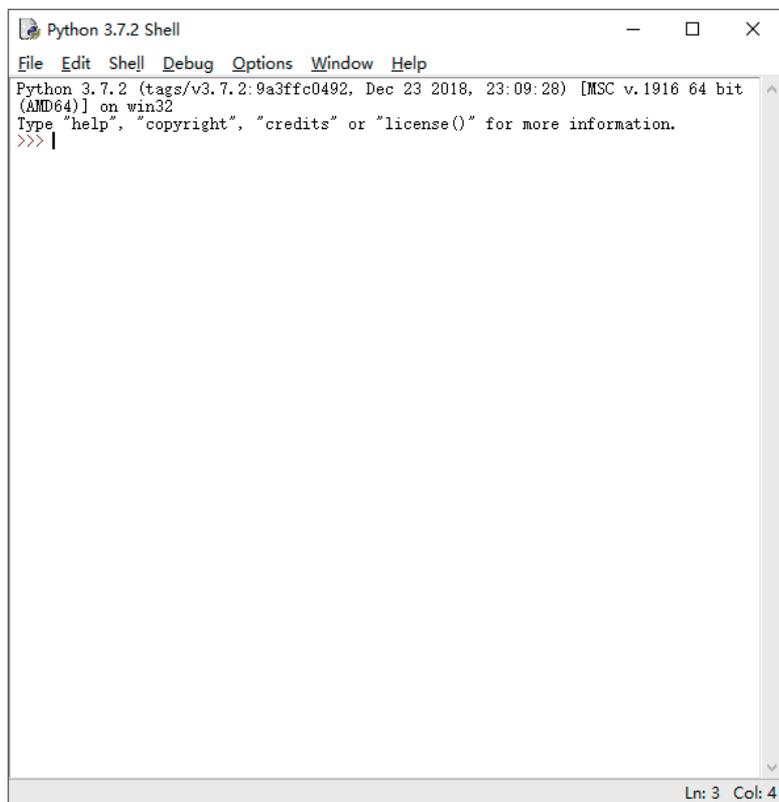


图 1-2-4 Python IDLE 启动窗口

IDLE 使用不同的颜色来表示关键字、常量、字符串等，用户可以很方便地进行区分。IDLE 常用操作介绍如下。

1. Python 命令

在 IDLE 交互式窗口中，使用“>>>”作为操作提示符，用户在其后直接输入 Python 命令，按“确定”键执行。

IDLE 支持自动补全功能，在对象变量名后输入“.”号时，自动显示该对象可用的属性和方法下拉列表。输入属性或方法名称的前几个字符串，列表会自动筛选。用户同样可以使用上下方向键从列表中选择相应的属性或方法，按回车键或空格键（在属性或方法名后会添加一个空格）可完成输入。

2. 查找历史记录

“>>>”提示符用来输入相关 Python 命令，如果想查找之前执行过的 Python 命令，可以按 Alt+P 组合键进行查找。也可以按 Alt+N 组合键进行查找。利用菜单“Edit”中的剪切、复制、粘贴等可以自制历史命令。

3. 创建程序

在 IDLE 交互式窗口中，选择菜单“File”下的“New File”命令（“File\New File”）或者按 Ctrl+N 组合键，打开 IDLE 编辑器编写 Python 程序，如图 1-2-5 所示。

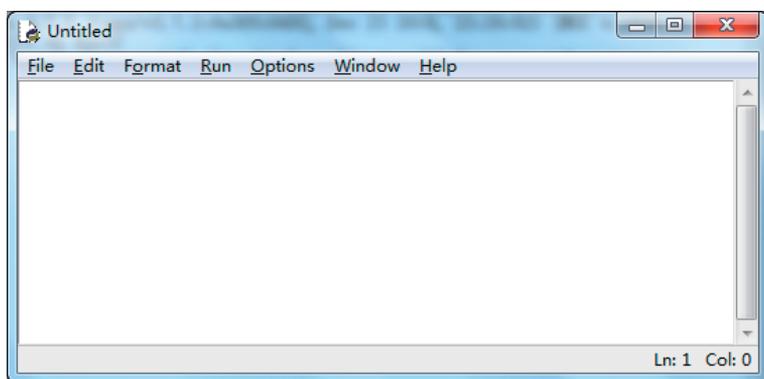


图 1-2-5 IDLE 编辑器

在 IDLE 编辑器中编写完成后，可以运行程序。首先保存文件，然后选择菜单“Run”中的“Run Module”命令或按键盘 F5 键运行。程序运行结果直接显示在 IDLE 交互式解释器窗口中。

4. 打开程序文件

在 IDLE 交互式解释器或编辑器中，选择菜单“File”下的“Open”命令可打开已有的 Python 程序。Python 程序文件扩展名主要有“.py”和“.pyw”两种，其中后者常用于 GUI 程序。

注意，结合全国计算机等级考试二级 Python 语言程序设计考试大纲要求，本教材使用开发环境主要是 IDLE 交互式解释器和编辑器，其中，代码前所带的“>>>”为 IDLE 交互式解释器的输入提示符。在交互式代码中，中间穿插有前面没有“>>>”的数据行，那是程序运行结果。代码块（所有行前不带“>>>”提示符的）的开发输入环境为 IDLE 编辑器。所有代码均能在 1.2.3 节中的 PyCharm 开发环境和 1.2.4 节中的 Anaconda3 开发环境下直接运行。

1.2.3 PyCharm 开发环境

PyCharm 是由 JetBrains 打造的一款 Python IDE。

PyCharm 具备一般 Python IDE 的功能，如调试、语法高亮、项目管理、代码跳转、智能

提示、自动完成、单元测试、版本控制等。

另外,PyCharm 还提供了一些很好的功能用于 Django 开发,同时支持 Google App Engine,更酷的是,PyCharm 支持 IronPython。

PyCharm 官方下载地址: <http://www.jetbrains.com/pycharm/download/>。进入 PyCharm 的下载页面,可以根据不同的平台下载不同版本的 PyCharm,并且每个平台均可以选择下载 Professional 和 Community 两个版本。两个版本的特点如下。

1. Professional 版本

- (1) 提供 Python IDE 所有的功能,支持 Web 开发。
- (2) 支持 Django、Flask、Google App 引擎、Pyramid 和 web2py。
- (3) 支持 JavaScript、CoffeeScript、TypeScript、CSS 和 Cython 等。
- (4) 支持远程开发、Python 分析器、数据库和 SQL 语句。

2. Community 版本

- (1) 它是轻量级的 Python IDE,只支持 Python 开发。
- (2) 免费、开源、集成 Apache2 的许可证。
- (3) 智能编辑器、调试器、支持重构和错误检查,集成 VCS 版本控制。

这里建议下载 Professional 版本。

PyCharm 安装方式简单,直接按安装界面默认设置进行操作即可。安装后的 PyCharm 操作效果图如图 1-2-6 所示。

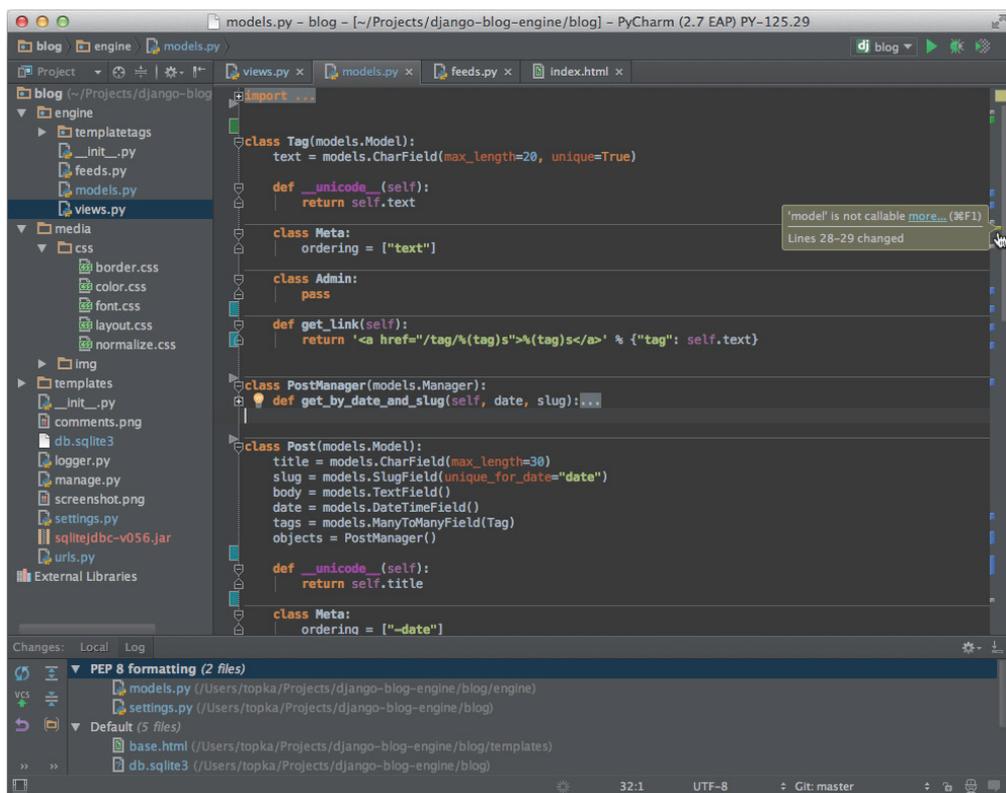


图 1-2-6 PyCharm 效果图

1.2.4 Anaconda3 开发环境

在众多 Python 开发环境中，Anaconda3 因为集成安装大量扩展库，得到了很多 Python 学习者和开发人员尤其是科研人员的喜爱。为此，这里简单介绍 Anaconda3 开发环境的使用。

这里以 Windows 操作系统为例，首先打开网址 <https://www.anaconda.com/>，选择“Download”，再选择“Windows”，选择 Python 3.7 version 或 Python 2.7 version，建议选择 Python 3.7 version，如图 1-2-7 所示。

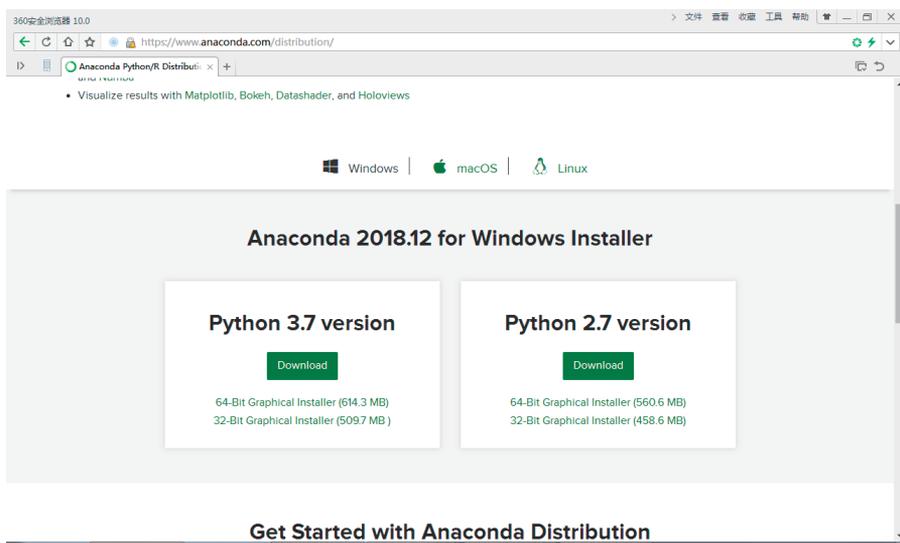


图 1-2-7 Anaconda3 下载页面

安装较为简单，基本都是单击“Next”按钮，为了避免不必要的麻烦，最后采用默认安装路径，具体安装过程为：下载最新版本的安装包，然后双击安装文件就可以安装了，如图 1-2-8 所示。

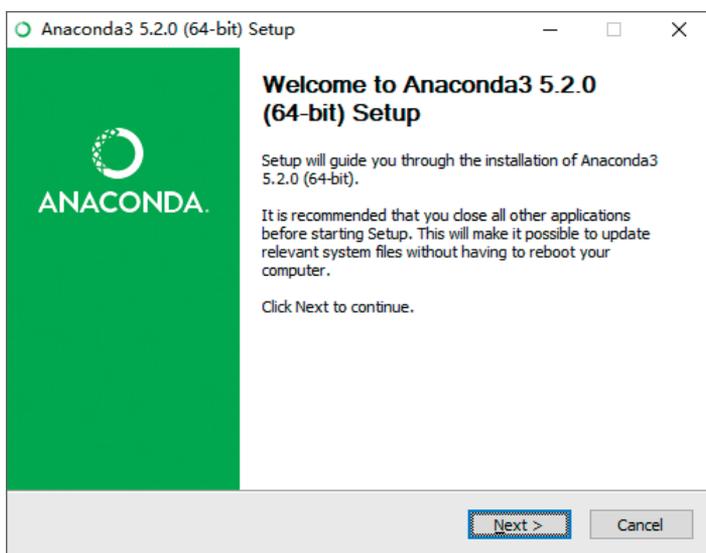


图 1-2-8 Anaconda3 安装开始界面

单击“Next”按钮，弹出如图 1-2-9 所示界面。

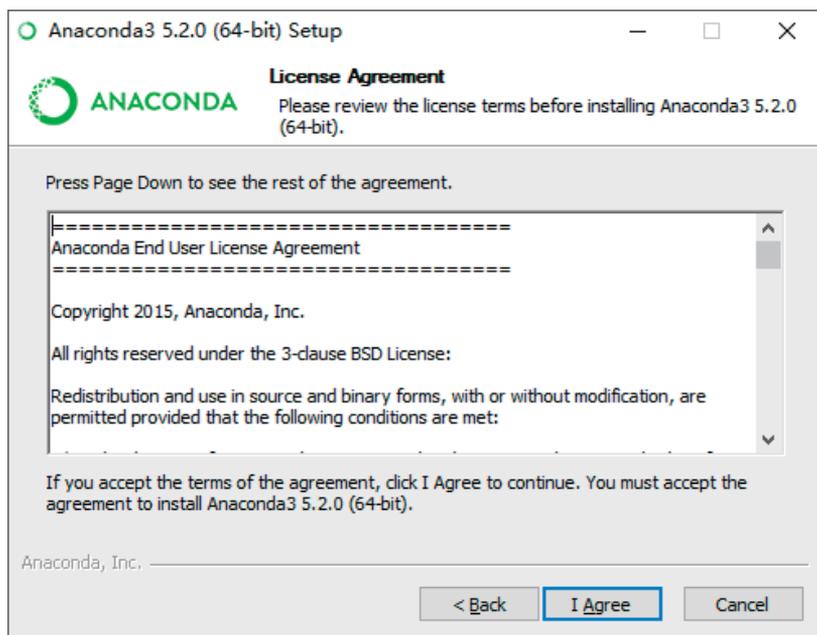


图 1-2-9 Anaconda3 安装协议界面

选择“I Agree”后，在弹出的以下界面中，可以根据自己的情况选择“Just Me”或“All Users”，如图 1-2-10 所示。

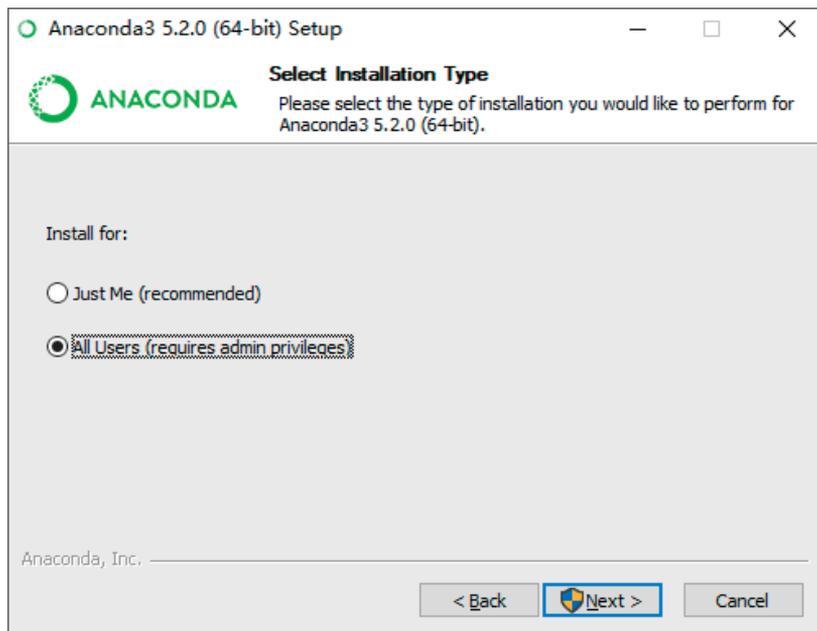


图 1-2-10 Anaconda3 安装类型界面

选择安装目录，再单击“Next”按钮继续，如图 1-2-11 所示。

在弹出的以下页面中，如果是第一次安装可以将两个选项都选上，将安装路径写入环境

变量。如果不是第一次安装，默认即可，如图 1-2-12 所示。

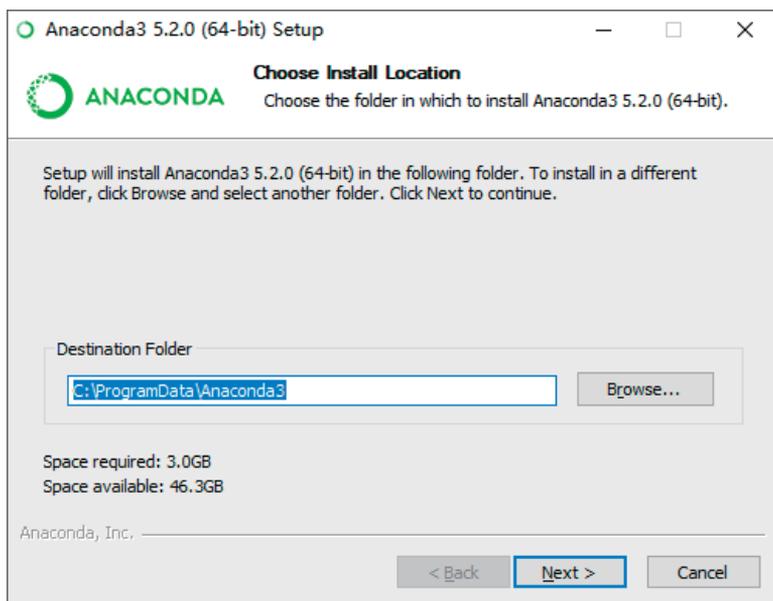


图 1-2-11 Anaconda3 安装目录界面

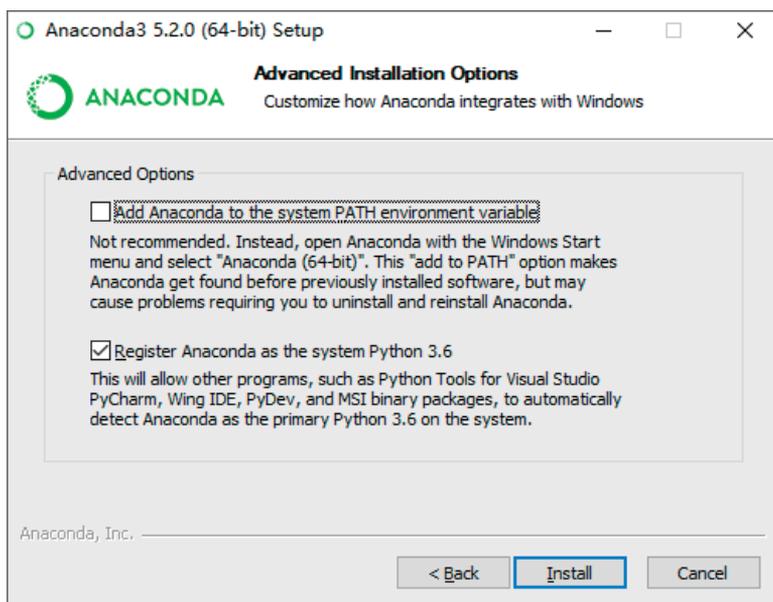


图 1-2-12 Anaconda3 安装选择环境变量界面

单击“Install”按钮然后等待安装，进度完成表示安装结束。

安装完成后，打开 Windows 的命令行窗口：按 Win+R 键打开窗口，输入 cmd。打开 Windows 的命令提示符，输入 conda list 就可以查询现在安装了哪些库，建议安装常用的 NumPy、Scipy 等，如图 1-2-13 所示。

如果还有什么包没有安装上，可以运行 conda install *** 来进行安装（***为需要的包的名称）。如果某个包版本不是最新的，运行 conda update *** 就可以更新了。

```

C:\WINDOWS\system32\cmd.exe
pyflakes 1.1.0 py35_0
pygments 2.1.1 py35_0
pyopenssl 0.15.1 py35_2
pyparsing 2.0.3 py35_0
pyqt 4.11.4 py35_5
pyreadline 2.1 py35_0
pytables 3.2.2 np110py35_2
pytest 2.8.5 py35_0
python 3.5.1 4
python-dateutil 2.5.1 py35_0
pytz 2016.2 py35_0
pywin32 220 py35_1
pyyaml 3.11 py35_3
pyzmq 15.2.0 py35_0
qt 4.8.7 [vc14]
qtawesome 0.3.2 py35_0
qtconsole 4.2.0 py35_1
qtpy 1.0 py35_0
requests 2.9.1 py35_0
rope 0.9.4 py35_1
rope-py3k 0.9.4.post1 <pip>
scikit-image 0.12.3 np110py35_0
scikit-learn 0.17.1 np110py35_0
scipy 0.17.0 np110py35_0
setuptools 20.3 py35_0
simplegeneric 0.8.1 py35_0
singledispatch 3.4.0.3 py35_0
sip 4.16.9 py35_2
six 1.10.0 py35_0
snowballstemmer 1.2.1 py35_0
sockjs-tornado 1.0.1 py35_0
sphinx 1.3.5 py35_0
sphinx-rtd-theme 0.1.9 <pip>
sphinx_rtd_theme 0.1.9 py35_0
spyder 2.3.8 py35_1
sqlalchemy 1.0.12 py35_0
statsmodels 0.6.1 np110py35_0
sympy 1.0 py35_0
tables 3.2.2 <pip>
tk 8.5.18 [vc14]
toolz 0.7.4 py35_0
tornado 4.3 py35_0
traitlets 4.2.1 py35_0
unicodcsv 0.14.1 py35_0
vs2015_runtime 14.00.23026.0 0
werkzeug 0.11.4 py35_0
wheel 0.29.0 py35_0
xlrd 0.9.4 py35_0
xlsxwriter 0.8.4 py35_0
xlwings 0.7.0 py35_0
xlwt 1.0.0 py35_0

```

图 1-2-13 查询 Anaconda 库界面

各个包都安装好以后，就可以进行后续操作了！数据分析最常用的程序叫 Jupyter，它是一个交互式的笔记本，能快速创建程序，支持实时代码、可视化和 Markdown 语言。

安装之后，有 IPython、Jupyter Notebook 和 Spyder 三个 Python 开发环境可用，本教材中，多行代码使用 Spyder 运行，单行交互式代码使用 IPython 或 Jupyter Notebook 运行，基于行业的通用性，介绍后面两个。

单击系统“开始”菜单，选择“所有程序”，再选择“Anaconda3”，展现如图 1-2-14 所示项目。在此选项中，包括 IPython、Jupyter Notebook 和 Spyder 等 Python 开发环境。

(1) 首先看一下 Jupyter Notebook，单击上面的菜单，自动打开一个网页，如图 1-2-15 所示。

单击右上角的“New”菜单，选择“Python3”，进入下面的界面，如图 1-2-16 所示。

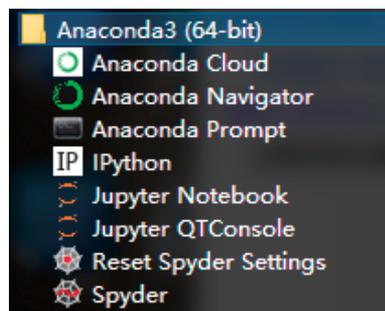


图 1-2-14 Anaconda3 列表界面

在每个 cell 中输入代码块，然后单击箭头所指处按钮“run cell”运行代码并查看输出结果，如图 1-2-17 所示。

(2) 单击“开始”菜单的“Spyder”，打开如图 1-2-18 所示的界面。

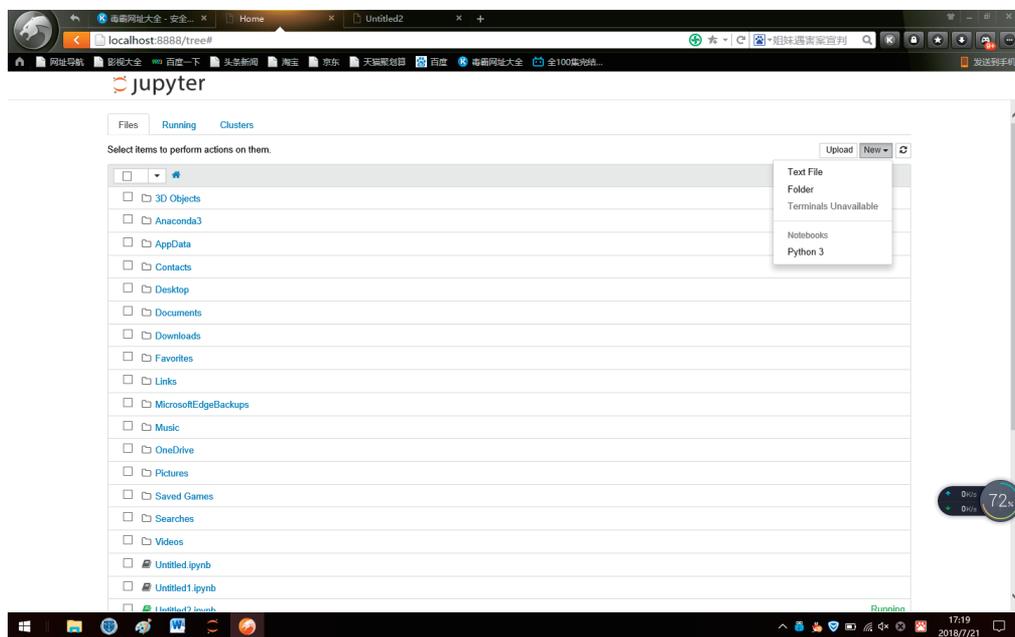


图 1-2-15 Jupyter Notebook 显示界面

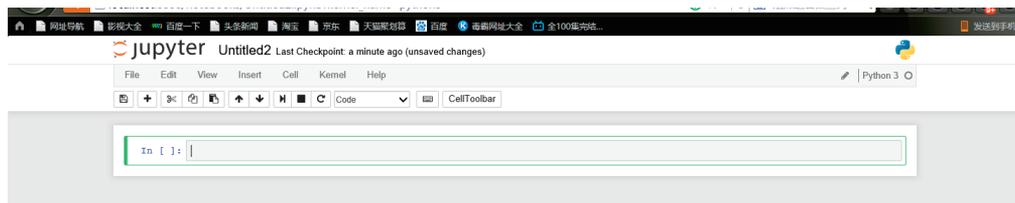


图 1-2-16 Jupyter Notebook 交互编程界面

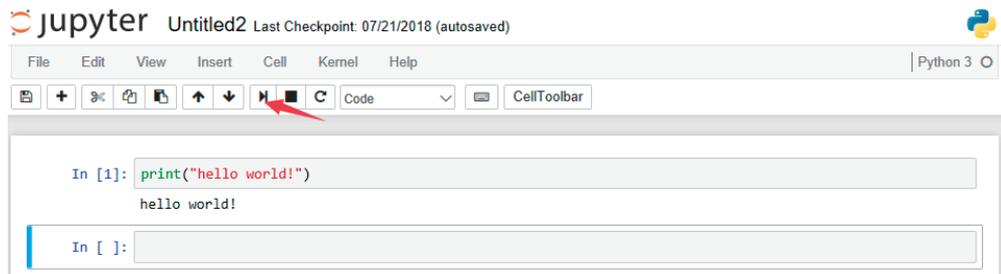


图 1-2-17 Jupyter Notebook 运行代码界面

在窗口中，1 处为程序窗口，可以编写完整的程序然后单击上方工具栏的“运行”按钮执行程序，程序运行结果会显示在右下角窗口的 Python 或 IPython 窗口中；2 处为 Python console 交互窗口；3 为 IPython console 交互式窗口。2 和 3 窗口中的操作类似于 Jupyter Notebook。

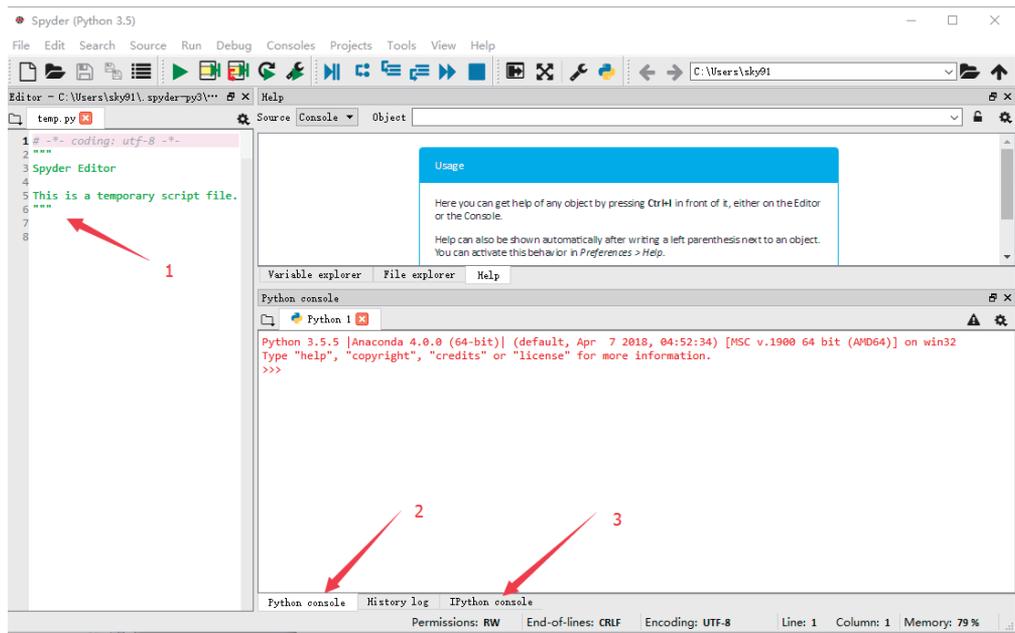


图 1-2-18 Spyder 显示界面

本教材所有的案例均可基于 Spyder 运行，如果是完整程序，可在图 1-2-17 中数字 1 所示程序窗口处输入运行，而交互式程序则在图 1-2-17 中数字 3 处所示 IPython console 交互窗口中运行。数字 2 处所示 Python console 交互窗口只能在 Anaconda3.5 之前的版本下加载，Anaconda3.5 后续版本只能加载运行 IPython console 交互窗口。

下面介绍 Spyder 的配置。

基本的配置都在 Tools→Preferences 里面，如图 1-2-19 所示。

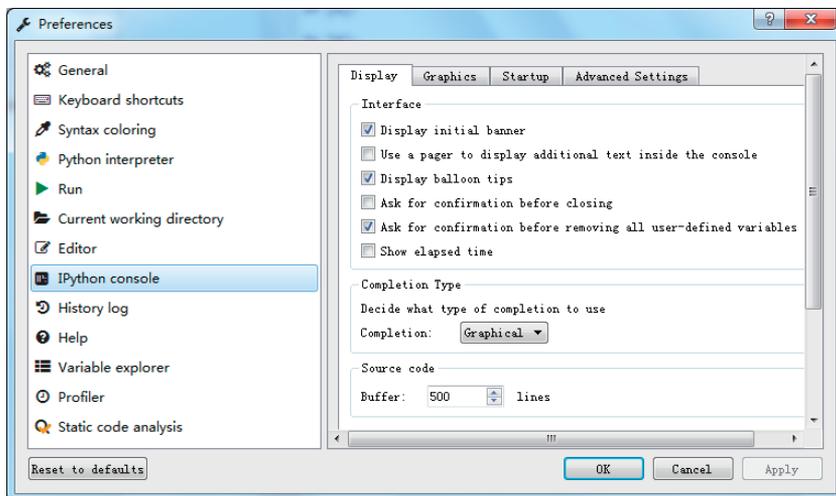


图 1-2-19 Spyder 配置界面

在配置界面可以进行 Spyder 的一些基本配置，如在 General 中可以调整字号和背景颜色，进行脚本编辑器的设置；在 Display 中可以设置背景、行号、高亮，在 Code Analysis 中可以设置代码提示等。



1.3 打包发布

Python 脚本在装有 Python 解释器、开发环境的计算机上可以运行，如果在没有安装这些的 Windows 系统下就需要打包发布运行。本节将讲解 pip 包管理工具及 PyInstaller 第三方库，讲解如何将脚本转换为 Windows 下的可执行程序。

【学习目标】

- 小节目标
1. 了解 Python 第三方库的概念
 2. 掌握 Python 第三方库的安装操作
 3. 了解并掌握 PyInstaller 第三方库的安装
 4. 掌握 Python 程序打包及发布

1.3.1 pip 包管理工具

pip 是一个现代的、通用的 Python 包管理工具，用于第三方库的获取和安装，提供了对 Python 包的查找、下载、安装、卸载的功能。Python 3.4 以上版本都自带 pip 工具。本节讲述 pip 工具的下载安装及使用。

第三方库是相对于 Python 的标准库的说法。第三方库都会在 Python 官方的 pypi.python.org 网站注册，要安装一个第三方库，必须先知道该库的名称，可以在官网上搜索，然后使用 pip 下载安装。

pip 官网：<https://pypi.org/project/pip/>

可以通过在命令提示符下输入“pip --version”命令来判断是否已安装 pip 工具，如图 1-3-1 所示。

```
C:\Users\Administrator>pip --version
pip 19.1.1 from c:\users\administrator\appdata\local\programs\python\python37-32
\lib\site-packages\pip (python 3.7)
```

图 1-3-1 命令提示符界面

1. 安装 pip

如果还未安装，则可以使用以下方法来安装。

(1) 安装 pip 之前，首先确认你的 Python 环境正常，然后通过官网下载 pip 安装包，如下载“pip-19.1.1.tar.gz”压缩包，如图 1-3-2 所示。

(2) 将下载的包解压，如解压到 D 盘，然后通过 Win+R 快捷方式打开命令提示符窗口，并在命令提示符中进入到 pip 文件目录下，如图 1-3-3、图 1-3-4 所示。

(3) 在 pip 目录下，输入命令“python setup.py install”进行 pip 模块的安装；安装完成后会有“Finished processing dependencies for pip==19.1.1”字样，如图 1-3-5 所示。

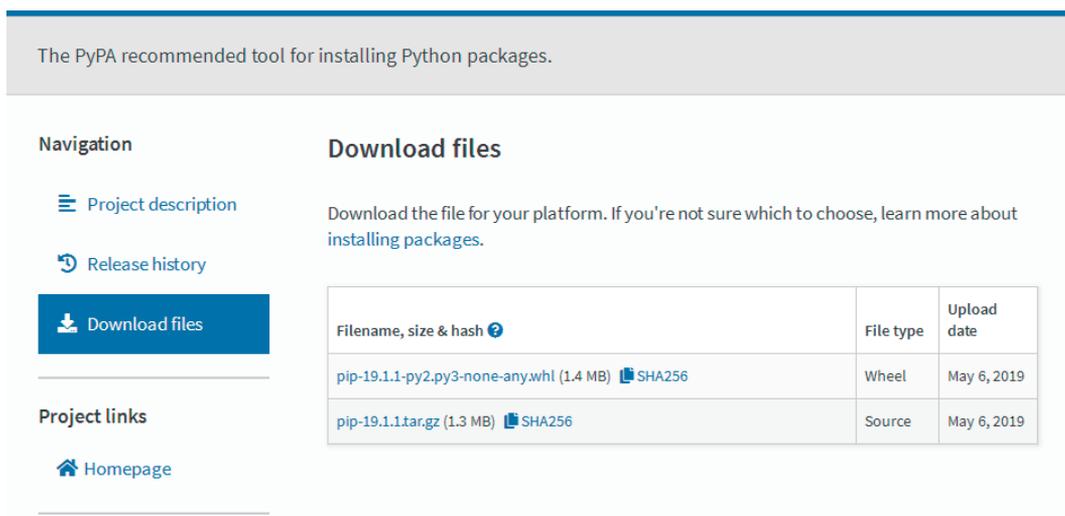


图 1-3-2 pip 官网主页下载页面

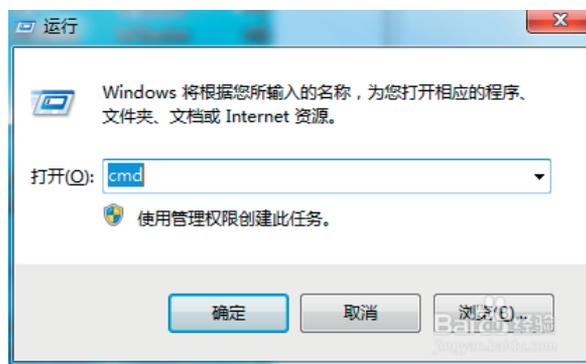


图 1-3-3 命令输入页面

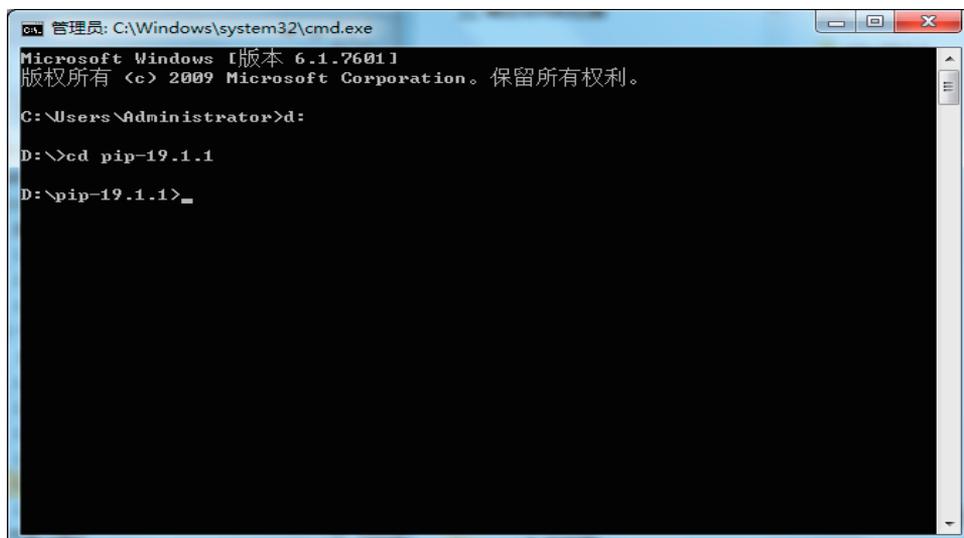


图 1-3-4 字符串命令页面

```

c:\ 管理员: C:\Windows\system32\cmd.exe

Installing pip-script.py script to C:\Users\Administrator\AppData\Local\Programs\Python\Python37-32\Scripts
Installing pip.exe script to C:\Users\Administrator\AppData\Local\Programs\Python\Python37-32\Scripts
Installing pip.exe.manifest script to C:\Users\Administrator\AppData\Local\Programs\Python\Python37-32\Scripts
Installing pip3-script.py script to C:\Users\Administrator\AppData\Local\Programs\Python\Python37-32\Scripts
Installing pip3.exe script to C:\Users\Administrator\AppData\Local\Programs\Python\Python37-32\Scripts
Installing pip3.exe.manifest script to C:\Users\Administrator\AppData\Local\Programs\Python\Python37-32\Scripts
Installing pip3.7-script.py script to C:\Users\Administrator\AppData\Local\Programs\Python\Python37-32\Scripts
Installing pip3.7.exe script to C:\Users\Administrator\AppData\Local\Programs\Python\Python37-32\Scripts
Installing pip3.7.exe.manifest script to C:\Users\Administrator\AppData\Local\Programs\Python\Python37-32\Scripts

Installed c:\users\administrator\appdata\local\programs\python\python37-32\lib\site-packages\pip-19.1.1-py3.7.egg
Processing dependencies for pip==19.1.1
Finished processing dependencies for pip==19.1.1

D:\pip-19.1.1>

```

图 1-3-5 pip 提示安装成功

(4) 待安装完成后，在命令提示符中输入“pip --help”，如果有相应的输出且没有报错，则表示 pip 成功安装，如图 1-3-6 所示。

```

c:\ 管理员: C:\Windows\system32\cmd.exe

D:\pip-19.1.1>pip --help

Usage:
  pip <command> [options]

Commands:
  install           Install packages.
  download          Download packages.
  uninstall         Uninstall packages.
  freeze           Output installed packages in requirements format.
  list             List installed packages.
  show             Show information about installed packages.
  check            Verify installed packages have compatible dependencies.
  config           Manage local and global configuration.
  search           Search PyPI for packages.
  wheel           Build wheels from your requirements.
  hash            Compute hashes of package archives.
  completion       A helper command used for command completion.
  help            Show help for commands.

General Options:
-h, --help          Show help.
--isolated          Run pip in an isolated mode, ignoring environment variables and user configuration.
-v, --verbose       Give more output. Option is additive, and can be used up to 3 times.
-U, --version       Show version and exit.
-q, --quiet         Give less output. Option is additive, and can be used up to 3 times (corresponding to WARNING, ERROR, and CRITICAL logging levels).
--log <path>       Path to a verbose appending log.
--proxy <proxy>    Specify a proxy in the form
                    user:passwd@lproxy.server:port.
--retries <retries> Maximum number of retries each connection should attempt (default 5 times).

```

图 1-3-6 pip 帮助查看界面

以上所述操作是下载 pip 压缩包进行安装的操作，也可下载“pip-19.1.1-py2.py3-none-any.whl”直接安装。同样，pip 命令也支持扩展名为.whl的文件直接安装 Python 扩展库。

(5) 部分 Python 版本默认安装了 pip，但是由于不是最新版本故需要升级，pip 的升级命令为“python-m install pip--upgrade pip”。

2. pip 常用命令

pip 常用命令包括获取帮助、升级与卸载、显示、第三方库下载等，如表 1-3-1 所示。

表 1-3-1 pip 常用命令

| 序号 | 命令 | 含义 |
|----|---|---|
| 1 | pip--version | 显示版本和路径 |
| 2 | pip--help | 获取帮助 |
| 3 | pip install-U pip | 升级 pip |
| 4 | pip install 库名 # 最新版本 pip install 库名==版本号 # 指定版本 pip install 库名>=版本号 # 最小版本 | 安装库 安装指定的库，通过使用==, >=, <=, >, < 来指定一个版本号 |
| 5 | pip install--upgrade 库名 | 升级库 升级指定的库和安装指定的库操作相同 |
| 6 | pip uninstall 库名 | 卸载库 |
| 7 | pip search SomePackage | 搜索库 |
| 8 | pip show | 显示安装库信息 |
| 9 | pip show-f SomePackage | 查看指定库的详细信息 |
| 10 | pip list | 列出已安装的库 |
| 11 | pip list-o | 查看可升级的库 |

1.3.2 PyInstaller

Python 脚本在装有 Python 解释器、开发环境的计算机上可以直接运行，如果在没有这些的 Windows 系统下就需要打包发布运行。本节将讲解如何将脚本转换为 Windows 下的可执行程序（后缀名为 exe）。PyInstaller 属于第三方库。第三方库的获取和安装方法基本都相同。PyInstaller 操作过程如下所示。

1. 安装 PyInstaller

(1) 首先计算机要连接互联网，打开命令提示符，输入“pip install pyinstaller”，并按回车键，如图 1-3-7 所示。



```
C:\Documents and Settings\Administrator>pip install pyinstaller
```

图 1-3-7 安装 PyInstaller

(2) 确定后开始从网络下载 PyInstaller，如图 1-3-8 所示。

(3) 如果安装完，则提示成功，如图 1-3-9 所示。

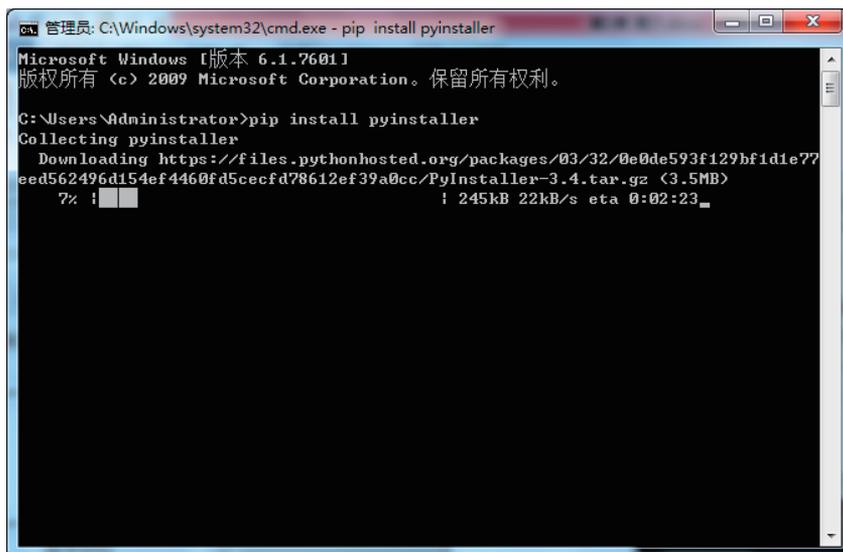


图 1-3-8 安装 PyInstaller 进程

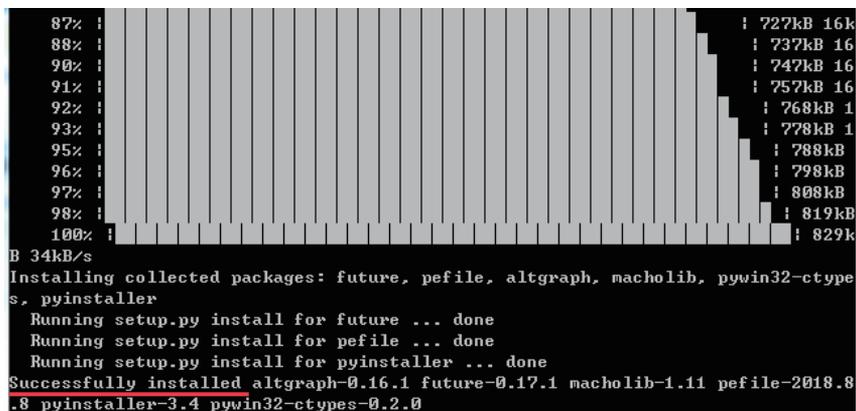


图 1-3-9 安装 PyInstaller 成功提示

注意：如果 pip 提示升级，如图 1-3-10 所示，则输入 “python -m pip install --upgrade pip” 或 “pip install -U pip” 进行 pip 升级。

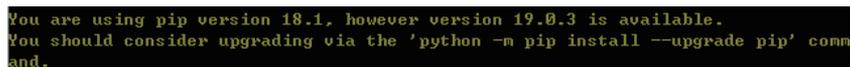


图 1-3-10 pip 升级提示

(4) 同样，pip 升级成功会提示升级成功，如图 1-3-11 所示。

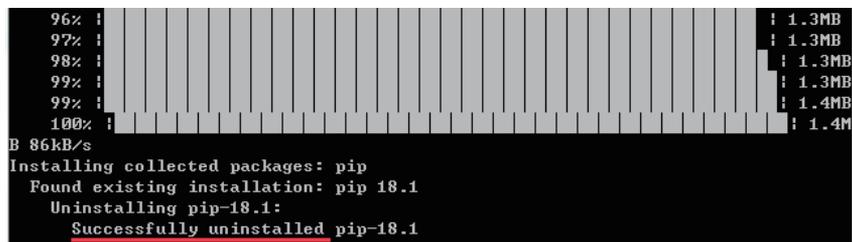


图 1-3-11 pip 升级成功提示