第 1 章

Python 语言概述

CHAPTER 1

本章要点

- Python 概述
- Python 的版本和开发环境
- 程序设计基本方法
- 模块、包与库
- 使用帮助
- Python 模块的 __name__ 属性

Python 是一种广泛使用的解释型、高级和通用的编程语言，不仅提供了高效的高级数据结构，还能简单有效地进行面向对象编程。Python 以其动态系统类型解释执行的特性，为开发者提供了快速开发跨平台应用程序的能力，并因此成为软件开发、数据分析、人工智能等领域的首选工具。

1.1 Python 概述

Python 是一种跨平台的计算机程序设计语言，是 ABC 语言的替代品，属于面向对象的动态类型语言，最初被设计用于编写自动化脚本，随着版本的不断更新和语言新功能的添加，越来越多被用于独立、大型项目的开发。Python 是人工智能首选的编程语言。

1.1.1 Python 的产生和发展

Python 是一种支持面向对象的解释型计算机程序设计语言，由荷兰人 Guido Van Rossum 于 1989 年发明，第一个公开版本发布于 1991 年。Python 是一种开源编程语言，源代码和解释器 CPython 遵循 GPL（general public license）协议。Python 语法简洁清晰，特色之一是强制使用空白符（blank）作为语句缩进。

Python 名字的由来：1989 年圣诞节期间，在阿姆斯特丹的 Guido 为了打发圣诞节的无趣时光，决心开发一个新的脚本解释程序作为 ABC 语言的一种继承。之所以选中 Python（中文含义为“大蟒蛇”）作为该编程语言的名字，是因为他是 Monty Python 喜剧团体的爱好者。

ABC 语言是由 Guido 参加设计的一种教学语言，在他看来，ABC 这种语言非常优美和强大，是专门为非专业程序员设计的。但是 ABC 语言并没有成功，究其原因，Guido 认为是其非开放性造成的。Guido 决心在 Python 中避免这一错误。同时，他还想实现在 ABC 中计划实现但未曾实现的东西。就这样，Python 在 Guido 手中诞生了。1991 年，第一个 Python 编译器（同时也是解释器）CPathon 诞生，它是用 C 语言编写的。Python 的第一个版本已经具有了类（class）、函数（function）、异常处理（exception handling）、包括列表（list）和字典（dictionary）在内的核心数据类型，以及模块（module）为基础的拓展系统。

目前，Python 已经成为最受欢迎的程序设计语言之一。2011 年 1 月，它被 TIOBE 编程语言排行榜评为 2010 年度语言。自 2004 年以来，Python 的使用率呈快速增长趋势。

1.1.2 Python 语言的特点

Python 的设计哲学是“优雅”“明确”“简单”，它的语法清楚、干净、易读、易维护，编程简单直接，更适合初学编程者，让初学者专注于编程逻辑，而不是困惑于晦涩的语法细节。对于想快速就职的读者而言，学习 Python 无疑是一条捷径。

一般来说，Python 语言具有如下特点。

1. 简单易学

Python 是一种代表简单主义思想的语言，用 Python 编写的程序就像英语段落一样流畅。此外，使用 Python 还可以编写伪代码，这使得在开发程序时只需专注于解决问题，而不用深究语言本

身的语法。

2. 免费、开源

Python 是免费、开源的，人们可以自由地发布软件副本、阅读和修改源代码、提取部分功能用于其他软件等。Python 如此优秀，主要得益于其开源的特点，这使它可以由一群优秀的 Python 爱好者创造并时常改进完善。

3. 可移植性

Python 程序能够被移植到许多平台上，无须修改便可以在众多平台上运行，这些平台包括 Linux、Windows、FreeBSD、Macintosh、Windows CE、Pocket PC、Symbian，Android 等。

4. 面向对象

Python 既支持面向过程编程，也支持面向对象编程。在面向过程的语言中，程序是由封装了可重用代码的函数构成的。在面向对象的语言中，程序是由数据和功能组合而成的对象构建起来的。与其他主要编程语言，如 C++ 和 Java 相比，Python 可以以一种非常强大且简单的方式实现面向对象编程。

5. 丰富的库

Python 标准库非常庞大，可以用于处理各种工作，包括正则表达式、线程、数据库、网页浏览器、单元测试、图形用户界面（Graphical User Interface，GUI）等。除这些标准库外，Python 中还提供了许多高质量的库，包括 wxPython、Twisted 和 Python 图像库等。

1.1.3　Python 语言的应用领域

Python 作为一种功能强大且通用的编程语言广受好评，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，因此 Python 的应用领域也越来越广泛。Python 应用的场景包括如下领域。

1. Web 开发

尽管今天 PHP 依然是 Web 开发的流行语言，但 Python 上升势头明显。随着 Python 的 Web 开发框架逐渐成熟，如 Django 和 Flask，读者可以快速地开发功能强大的 Web 应用。Django 是 Python Web 开发的首选框架，无论是建大型网站，开发 OA 还是 Web API，Django 都可以轻松胜任。

2. 网络爬虫

有了 Python，用几行代码就可以编写一个爬虫程序。爬虫程序从网络上获取有用的数据或信息，可以节省大量人工时间。能够编写网络爬虫程序的编程语言不少，Python 是其中的主流之一，Python 自带的 urllib 库、第三方的 requests 库和 Scrappy 框架使开发爬虫程序变得非常容易。

3. 计算与数据分析

随着 NumPy、SciPy、Matplotlib 等众多 Python 库的开发和完善，Python 越来越适合科学计算和数据分析。它不仅支持各种数学运算，还可以绘制高质量的 2D 和 3D 图像。与科学计算领

域流行的商业软件 MATLAB 相比，Python 采用的脚本语言的应用范围更广泛，可以处理更多类型的文件和数据。

4. 人工智能

Python 在人工智能大范围领域内，如机器学习、神经网络、深度学习等方面都是主流的编程语言，得到了广泛的支持和应用。最流行的神经网络框架，如 Facebook 的 PyTorch 和 Google 的 TensorFlow 都采用了 Python 语言。

5. 自动化运维

Python 是运维工程师首选的编程语言。在很多操作系统中，Python 是标准的系统组件。大多数 Linux 发行版和 mac OS X 都集成了 Python，可以在终端下直接运行 Python。Python 标准库包含了多个调用操作系统功能的库。通过 pywin32 这个第三方软件包，Python 能够访问 Windows 的 COM 服务及其他 Windows API。使用 IronPython，Python 程序能够直接调用“.NetFramework”。一般来说，Python 编写的系统管理脚本在可读性、性能、代码可重用性、扩展性等方面都优于普通的 shell 脚本。

6. 云计算

Python 的强大之处在于模块化和灵活性，而构建云计算平台 IaaS 服务的 OpenStack 就是采用 Python 开发的，云计算的其他服务也都是建立在 IaaS 服务之上的。

7. 网络编程

Python 提供了丰富的模块支持 socket 编程，能方便快速地开发分布式应用程序，很多大规模软件开发计划，例如，Zope、Mnet、BitTorrent 和 Google 都在广泛地使用。

8. 游戏开发

很多游戏使用 C++ 编写图形显示等高性能模块，而使用 Python 或者 Lua 编写游戏的逻辑、服务器。相较于 Python，Lua 的功能更简单、体积更小，但 Python 支持更多的特性和数据类型。Python 的 PyGame 库也可用于直接开发一些简单游戏。

Python 是一门对新手友好、功能强大、高效灵活的编程语言，无论是想进入数据分析、人工智能、网站开发这些领域，还是希望掌握第一门编程语言，都可以用 Python 来开启未来无限的可能。

1.2 Python 的版本和开发环境

目前广泛使用的是 Python 3 版本。对于每个 Python 工程师来说，必须使用集成开发环境这个开发工具；对于不同的读者来说，选择合适的编辑器环境，对 Python 编程效率的影响是非常大的。因此，选择合适的 Python 集成开发环境工具十分重要。

1.2.1 Python 语言的版本

Python 1.0 版本发布于 1994 年，是由 Guido Van Rossum 创建并发布的。Python 1.0 版本相较

于现在的版本，功能有限，缺乏很多现在常用的特性。

Python 2.0 版本发布于 2000 年，增加了很多新特性，如支持 Unicode、改进丰富了标准库、引入了新的运算符等。这些新功能使 Python 2.0 版本更加强大，更易于使用。

Python 3.0 版本发布于 2008 年，进行了重大的版本升级，修复了 Python 2.x 版本中的一些设计缺陷，并且不兼容 Python 2.x 版本。相较于 Python 2.0 版本，Python 3.0 版本新增改进字符串处理、改进整数除法、改进异常处理，新增语言特性，以及更好的 Unicode 支持等，这些新功能使 Python 3.0 版本更加强大且更易于使用。

1.2.2　Python 的下载和安装

Python 官方网站中可以下载 Python 解释器，构建 Python 的开发环境。下面以 Windows 系统为例演示 Python 的下载与安装过程，具体操作步骤如下。

（1）访问 Python 官网 https://www.Python.org/ 选项，如图 1-1 所示。

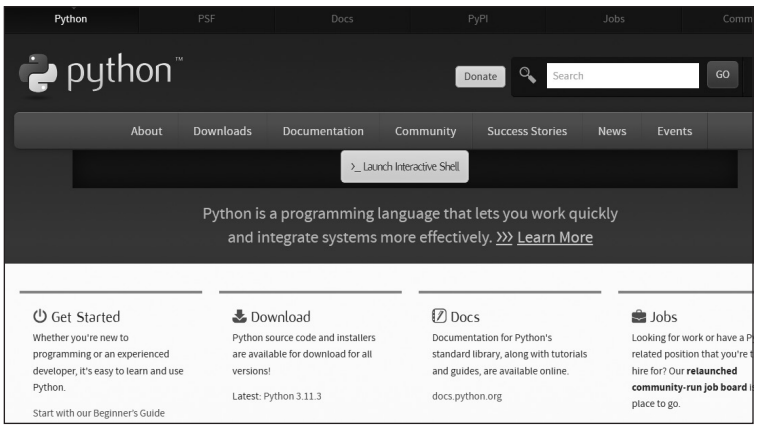

图 1-1　Python 官网首页

（2）选择 Downloads → Windows，跳转到 Python 下载页面，如图 1-2 所示。下载页面有很多版本的安装包，可以根据自己的需求下载相应的版本。这里选择 Python3.8.8 版本的 64 位离线安装包，单击 Windows installer（64-bit）超链接即可下载。

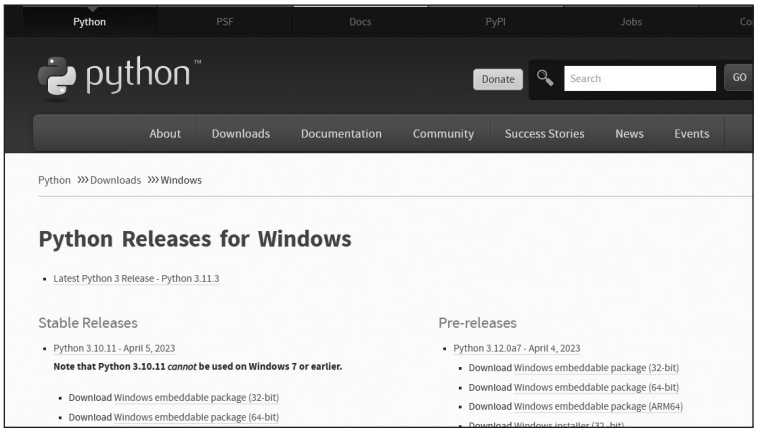

图 1-2　选择 Windows 平台的 Python 版本页面

（3）下载成功后，双击安装包，打开如图 1-3 所示的安装界面，开始安装。在 Python 3.8.8 安装界面中提供默认安装与自定义安装两种方式。对于安装界面下方的 Add Python 3.8 to PATH 复选框，若勾选此复选框，安装完成后 Python 解释器路径将被自动添加到环境变量中；若不勾选此复选框，则在使用 Python 之前需要手动将 Python 解释器路径添加到环境变量。

图 1-3　Python 的安装方式选择界面

（4）这里采用自定义安装方式，可以根据用户需求有选择地进行安装。在图 1-3 所示界面选择 Customize installation 选项，进入 Optional Features（可选功能）界面，如图 1-4 所示。

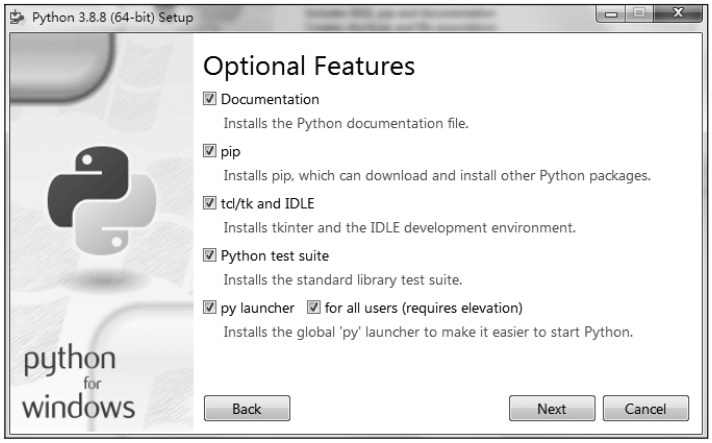

图 1-4　自定义安装的 Optional Features 界面

图 1-4 默认勾选了所有功能，这些功能的相关介绍如下。

Documentation：Python 帮助文档，其目的是帮助开发者查看 API 及相关说明。

pip：Python 包管理工具，该工具提供了对 Python 包的查找、下载、安装、卸载功能。

tcl/tk and IDLE：tcl/tk 是 Python 的标准图形用户界面接口，IDLE 是 Python 自带的简洁集成开发环境。

Python test suite：Python 标准库测试套件。

py launcher：安装 py launcher 后可以通过全局命令 py 更方便地启动 Python。

for all users：适用于所有用户。

（5）保持默认配置，单击 Next（下一步）按钮进入 Advanced Options（高级选项）界面，用

户在该界面依然可以根据自身需求，选择是否勾选相应功能对应的复选框，以及设置 Python 安装路径，具体如图 1-5 所示。

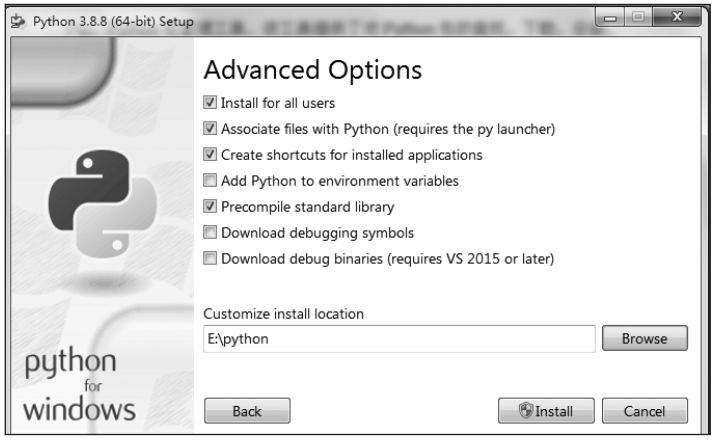

图 1-5　Python 安装的 Advanced Options 界面

（6）选好 Python 的安装路径后，单击 Install（安装）按钮开始安装，安装过程如图 1-6 所示。自此，Python3.8.8 安装完成，此时，打开 Windows 的“开始”菜单，在“所有程序”中会显示如图 1-7 所示的内容，其中“IDLE（Python 3.8）”为 Python 自带的编程工具。下面使用 Python 的 IDLE 程序检测 Python 3.8.8 是否安装成功。

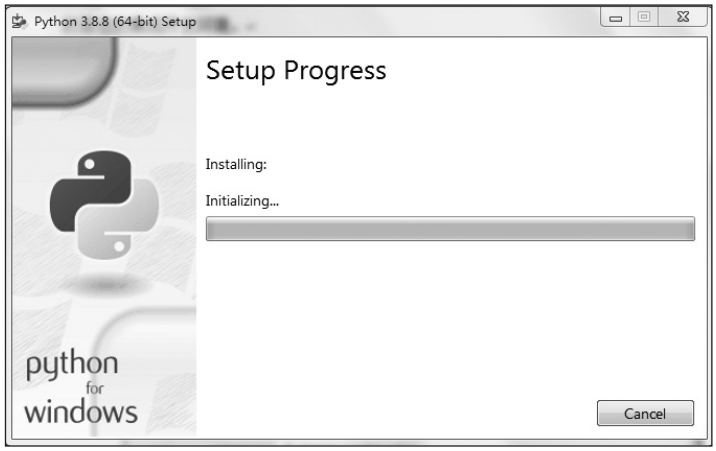

图 1-6　Python 安装过程界面

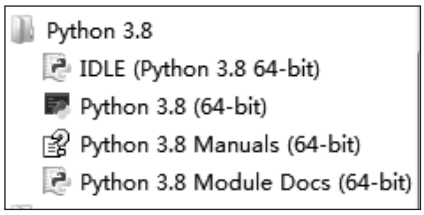

图 1-7　安装成功后的程序

在 Windows 系统中打开 IDLE Shell（IDLE 主窗口），如图 1-8 所示，在 IDLE Shell 中输入 print("hello python") 按 Enter 键后显示 hello python 的信息，表明安装成功。IDLE 主窗口的打开

方式会在 1.2.3 节详细说明。

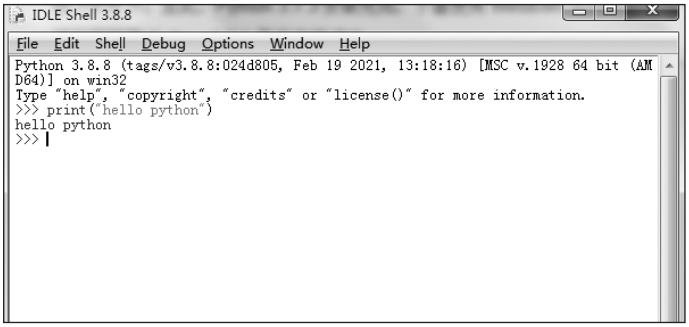

图 1-8　Python 安装成功后的显示内容界面

1.2.3　Python 语言的集成开发环境

Python 的集成开发环境一般包括代码编辑器、编译器、调试器和图形用户界面等工具。Pydev、Eclipse、VIM 等多种集成开发环境都可以用于 Python 程序的开发，本节介绍常用的 3 种集成开发环境，Python 自带的 IDLE、Spyder 和 PyCharm。

1. IDLE

IDLE（Integrated Development and Learning Environment，集成开发和学习环境）是 Python 自带的、默认的、入门级代码编写工具，包含交互式和文件式两种方式。在交互式中可以编写一行或者多行语句并且立即看到结果。在文件式中可以像其他文本工具类集成开发环境（Integrated Development Environment，IDE）一样编写代码。当安装好 Python 后，IDLE 就会自动安装，不需要另外安装，IDLE 的基本功能包括语法加亮、段落缩进、基本文本编辑、Tab 键控制、程序调试等。

本节将以 Windows 7 系统安装的 IDLE 为例，详细介绍如何使用 IDLE 开发 Python 程序。

单击系统的“开始”菜单，选择“所有程序”→ Python 3.8 → IDLE（Python 3.8 64-bit）选项（笔者电脑中安装的版本），即可打开 IDLE 主窗口，如图 1-9 所示。

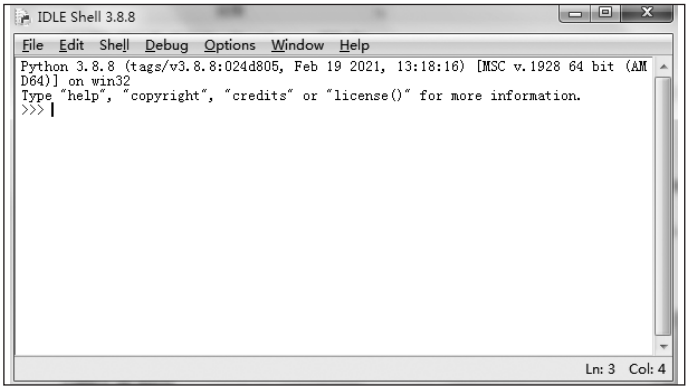

图 1-9　IDLE 主窗口

使用 IDLE 进行程序编写的具体步骤如下。

（1）在 IDLE 主窗口的菜单栏上，选择 File（文件）→ New File（新建文件）选项，打开一个新窗口，在该窗口中可以直接编写 Python 代码。

（2）在输入一行代码后按 Enter 键，将自动换到下一行，等待继续输入，如图 1-10 所示。

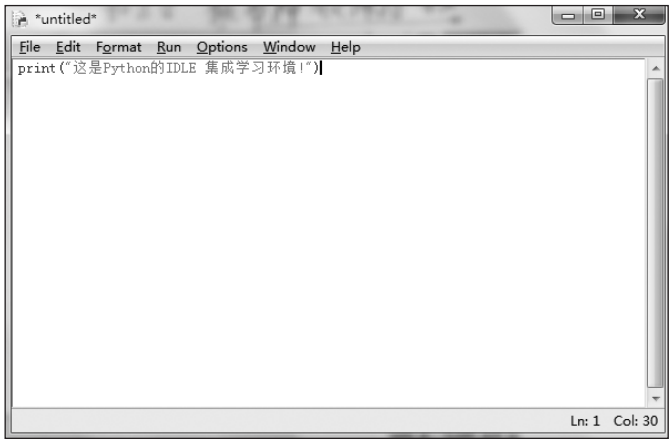

图 1-10　IDLE 的代码编辑界面

（3）按 Ctrl+S 快捷键保存文件，在弹出的“另存为”对话框中将文件名设置为 demo.py（.py 是 Python 文件的扩展名），单击“保存”按钮。在菜单栏中选择 Run → Run Module 选项（也可以直接按 F5 键），运行程序，运行结果如图 1-11 所示。

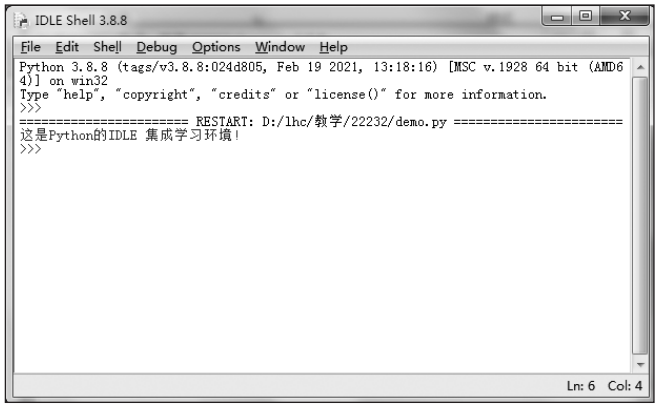

图 1-11　程序运行结果

2. Spyder

Spyder 是一个强大的交互式 Python 语言集成开发环境，提供高级的代码编辑、交互测试、调试等功能，支持 Windows、Linux 和 OS X 系统。安装 Anaconda[①] 后，Spyder 作为其中一个 IDE 自动安装。

Spyder 是 Python(x, y) 的作者为它开发的一个简单的集成开发环境。和其他 Python 开发环境相比，它最大的优点是可以很方便地观察和修改数组的值。

Spyder 的主界面如图 1-11 所示，Spyder 的界面由许多窗格构成，用户可以根据自己的喜好调整它们的位置和大小。多个窗格出现在一个区域时，将使用标签页的形式显示。在图 1-12 中，可以看到如下内容。

① Anaconda 是 Python 的一个发行版，其中内置了很多工具包，不用再单独安装。Anaconda 将 Python 和许多与科学计算相关的库捆绑在一起，形成了一个方便的科学计算环境，安装了 Ananconda 就相当于安装了 Python 和这些模块及库。

菜单栏（Menu bar）：包括 File | 文件、Edit | 编辑、Search | 查找、Source | 源代码、Run | 运行、Debug | 调试、Consoles | 控制台、Projects | 项目、Tools | 工具、View | 查看、Help | 帮助等菜单，通过菜单可以使用 Spyder 的各项功能。

工具栏（Tools bar）：通过单击图标可快速执行 Spyder 中最常用的操作，将鼠标悬停在某个图标上可以获取相应功能的说明。

路径窗口（Python path）：显示文件目前所处路径，通过其下拉菜单和后面的两个图标可以很方便地进行文件路径的切换。

代码编辑区（Editor）：编写 Python 代码的窗口，左边的行号区域显示代码所在行。

变量查看器（Variable explorer）：类似 MATLAB 的工作空间，可以方便地查看变量。

文件查看器（File explorer）：可以方便地查看当前文件夹下的文件。

帮助窗口（Help）：可以快速便捷地查看帮助文档。

控制台（IPython console）：类似 MATLAB 中的命令窗格，可以一行行地交互。

历史日志（History log）：按时间顺序记录输入到任何 Spyder 控制台的每个命令。

图 1-12　Spyder 主界面

Spyder 是 Python 用户中知名度很高的集成开发环境，具有如下特点。

（1）类 MATLAB 设计：Spyder 在设计上参考了 MATLAB 软件，Python 中的变量查看器模仿了 MATLAB 中的“工作空间（Workspace）”的功能，并且有类似 MATLAB 的 Python Path 管理对话框，对熟悉 MATLAB 的 Python 的初学者非常友好。

（2）资源丰富且查找便利：Spyder 具有变量自动完成、函数调用提示及随时随地访问文档帮助的功能，能够访问的资源及文档链接包括 Python、Matplotlib、NumPy、SciPy、Qt、IPython 等多种工具及工具包的使用手册。

（3）对初学者友好：Spyder 在其菜单栏的 Help（帮助）菜单中给用户提供了交互式的使用教程及快捷方式的备忘单，能够帮助用户快速直观地了解 Spyder 的用户界面及使用方式。

（4）工具丰富，功能强大：Spyder 中除了拥有一般 IDE 普遍具有的编辑器、调试器、用户图形界面等组件，还具有对象查看器、变量查看器、交互式命令窗口、历史命令窗口等组件，除此之外还拥有数组编辑及个性定制等多种功能。

3. PyCharm

PyCharm 是专业的 Python 集成开发环境，有两个版本。一个是免费的 Community（社区）版本，