

3.1 我的第 1 个 Windows 版 Python 程序

Python 的运行方法分为两种,本节将分别进行介绍。

1. 方法一

通过 cmd 命令进入 Command Prompt 模式,然后输入以下命令进入 Python 环境:

```
python
```

在 Python 开发环境中直接输入以下代码:

```
print("see you again, powenko")
```

结果如图 3-1 所示。

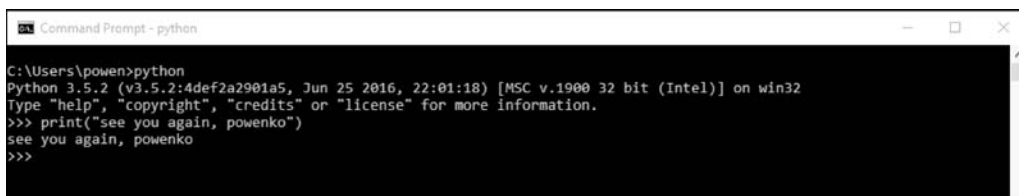


图 3-1 直接编写程序

可以发现,简单的程序可以逐行输入,如果程序很长或需要测试和修改,该方法就不再适用。

输入以下命令离开 Python 开发环境:

```
exit()
```

2. 方法二

首先通过文本编辑工具,将程序写在纯文本文档中,再让 Python 程序读入该文本并运行。Windows 平台的开发者,可以通过记事本等纯文本软件输入程序。

在纯文本软件中写入如图 3-2 所示的程序,并将其用 UTF-8 文档格式存储为名为 mycode.py 的纯文本文档。



图 3-2 Windows 的开发者可以通过记事本输入程序

在 cmd 模式下,移动路径至 mycode.py 所在的位置,并通过以下命令运行:

```
Python mycode.py
```

即可成功运行 Python 的程序,结果如图 3-3 所示。

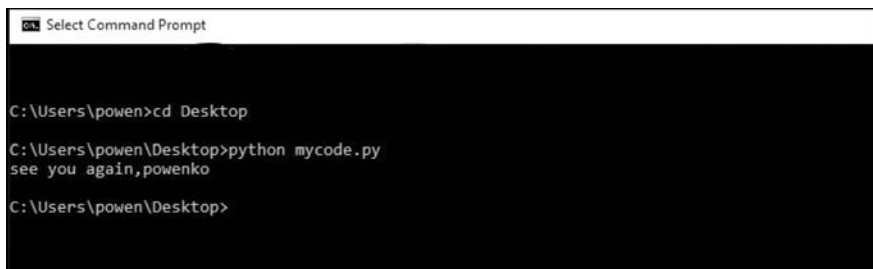


图 3-3 Windows 上的运行结果

微课视频: 1-Python-windows-Mycode.mp4。



3.2 我的第 1 个 Mac、Linux 和树莓派版 Python 程序

在 Mac、Linux 和树莓派系统上也可以打开 Terminal 文字模式,直接输入命令 \$ Python 或 \$ Python 3 进入 Python 开发环境。

输入以下命令：

```
print("see you again, powenko")
```

运行结果如图 3-4 所示。

```
pi@raspberrypi:~ $ python
Python 2.7.9 (default, Mar  8 2015, 00:52:26)
[GCC 4.9.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> print("see you again, powenko")
see you again, powenko
>>> exit()
```

图 3-4 在树莓派上运行 Python 程序

使用 Mac、Linux 或树莓派/LinkIt 系统的用户,可以通过文字编辑软件或内置的 nano 软件,先把要运行的程序写好,例如：

```
$ sudo nano mycode.py
```

运行结果如图 3-5 所示。

```
pi@raspberrypi:~/Desktop $ nano mycode.py
```

图 3-5 通过 nano 软件打开 mycode.py 的文档

在纯文本软件中输入以下代码(实例 1——mycode.py)：

```
print("see you again, powenko")
```

使用 nano 软件的 Mac 或树莓派开发者,可按 Ctrl+O 组合键或 Enter 键存储程序,按 Ctrl+X 组合键或 Enter 键离开文字编辑软件 nano,结果如图 3-6 所示。

```
GNU nano 2.2.6      File: mycode.py      Modified
print("see you again, powenko")
|

^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U UnCut Text ^T To Spell
```

图 3-6 用 nano 文字编辑软件编写程序

回到 Terminal 文字模式中,通过如图 3-7 所示的命令即可运行该程序。

```
$ Python mycode.py
```

```

pi@raspberrypi:~/Desktop $ nano mycode.py
pi@raspberrypi:~/Desktop $ ls
9x14Leds  day5  hello.py  pi4j-1.0.deb  ser3_LED0n0ff.py
al.jpg   day5B  ledon.py  pwm.py.save  ser4_LED0n0ff_F60.py
day3     day6  mycode.py  RPi.GPIO-0.6.2.tar.gz  ser.py
day4     gpio  mycv      ser2_write.py
pi@raspberrypi:~/Desktop $ python mycode.py
see you again, powenko
pi@raspberrypi:~/Desktop $

```

图 3-7 Terminal 文字模式下运行结果

微课视频：2-Python-RPi-Mycode.mp4。



3.3 开发和调试工具的下载和安装

Python 的开发调试工具较多,受欢迎的开发工具有 PyCharm、PyDev (Eclipse 的 Python 版本)、Thonny 和 Anaconda 的 Spyder。

本节将介绍如何安装和设置 PyCharm。PyCharm 有 Windows、Mac 和 Linux(包括树莓派)3 种版本。与其他开发调试工具相比,PyCharm 是安装和使用最简单的 IDE 开发工具。

下面介绍 PyCharm 的安装过程:

(1) 进入 PyCharm 官网。进入 PyCharm 的官方网站 <http://www.jetbrains.com/pycharm/>,单击 DOWNLOAD NOW 按钮进入下载页面,如图 3-8 所示。

(2) 下载 PyCharm 的最新版本。网页将自动判断当前操作系统,并切换到相应版本。单击 Community 选项下的 DOWNLOAD 即可下载最新版本的 PyCharm 开发工具,如图 3-9 所示。当前 Community(社区)版是免费使用的,而 Professional(专业)版则是收费的,有一个月的免费试用期。Professional 版提供更多用于 HTML、JS 和 SQL 的语法编辑功能。

(3) 运行安装。PyCharm 安装文档下载完成后,单击该文档或单击 Run 按钮即可安装,如图 3-10 所示。

(4) 安装设置。

① 在安装介绍页面上单击 Next 按钮,如图 3-11(a)所示。

② 在安装位置设置页面上,使用系统默认的位置,单击 Next 按钮进入下一步,如图 3.11(b)所示。

③ 选中 Create Desktop shortcut 下的 32-bit launcher 复选框创建桌面快捷方式,选中 Create associations 下的 .py 复选框,指定 PyCharm 软件为打开扩展名为 .py 的文档的工具,如图 3-11(c)所示。

④ 选择“开始”菜单上的文件夹,单击 Install 按钮继续,如图 3-11(d)所示。



图 3-8 PyCharm 官方网站

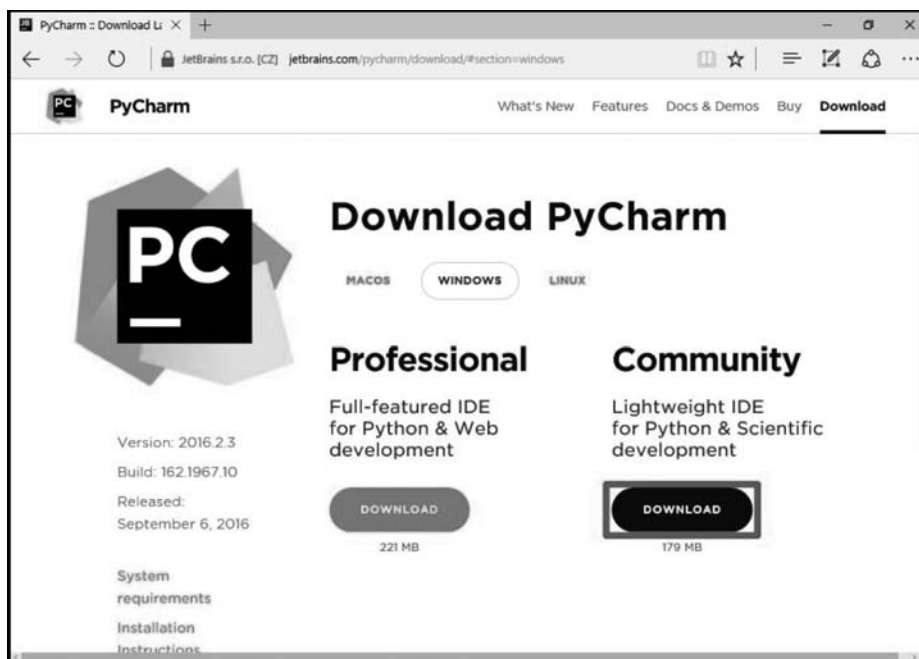


图 3-9 下载最新版本的 PyCharm

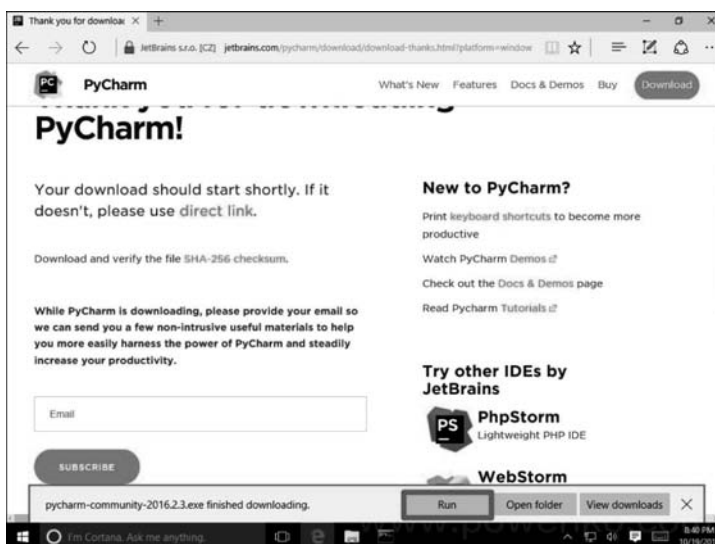


图 3-10 运行安装

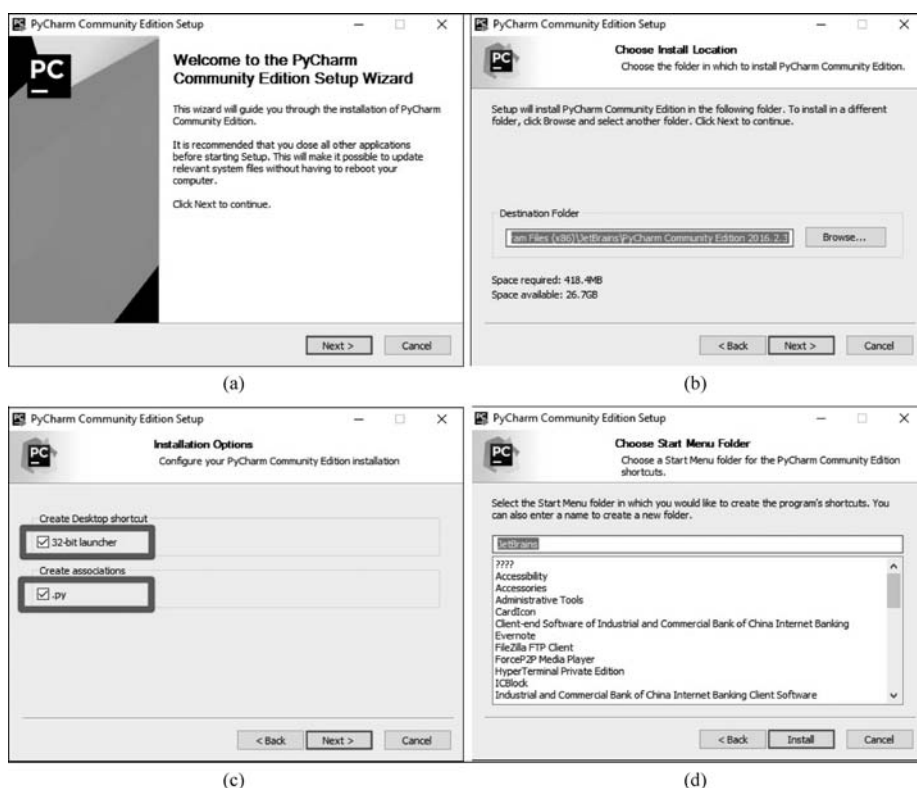


图 3-11 安装软件便会进行安装的动作

(5) 安装完成。出现如图 3-12 所示的界面后,整个 PyCharm 便安装完成了。



图 3-12 PyCharm 安装完成

注意：使用 Mac、树莓派和 Linux 系统的用户也可使用 PyCharm 软件开发和测试 Python,安装和设置方法与在 Windows 环境类似。

微课视频：3-Python-PyCharm-debugtools. mp4。



3.4 打开 PyCharm

打开 PyCharm 的方法如下：

(1) 双击桌面上的 PyCharm 图标,如图 3-13 所示,打开 PyCharm 软件。

(2) 首次打开时,PyCharm 将询问是否需要导入旧版的 PyCharm 设置,如图 3-14 所示。因为是首次使用,请选中 I do not have a previous version of PyCharm or I do not want to import my settings 单选按钮。

(3) 将弹出 JetBrains 版权声明,单击 Accept 按钮接受,如图 3-15 所示。



图 3-13 打开 PyCharm 开发软件

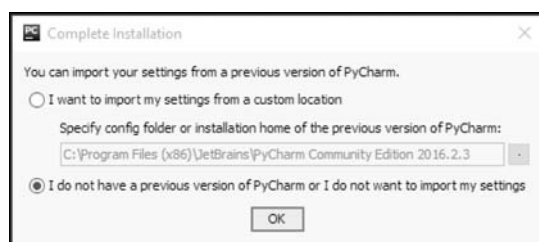


图 3-14 导入项目

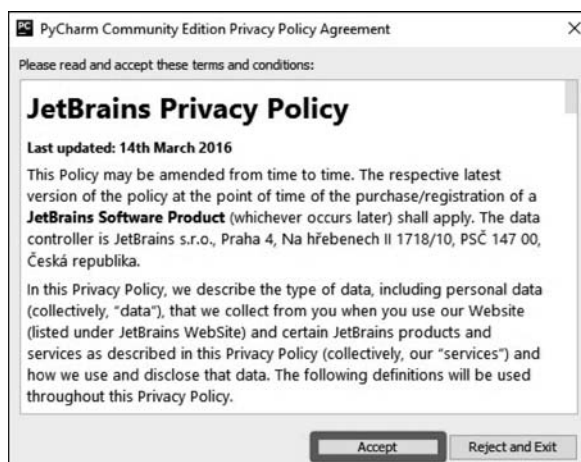


图 3-15 单击 Accept 按钮

(4) PyCharm 将询问外观的设置,使用默认值并单击 OK 按钮即可,如图 3-16 所示。

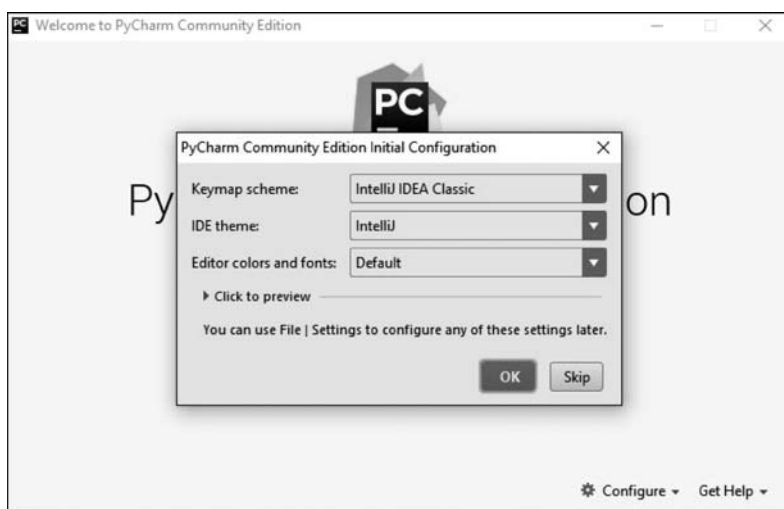


图 3-16 单击 OK 按钮

3.5 用 PyCharm 创建项目

用 PyCharm 创建项目的步骤如下：

(1) 在 PyCharm 主菜单中选择 Create New Project 选项,如图 3-17 所示。



图 3-17 选择 Create New Project 选项创建新的项目

(2) 系统会询问存放项目的路径及 Interpreter 运行 Python 程序的路径,推荐将路径指向 Python 3.6.6 的路径,否则会多创建一份 Python 程序,给后续通过 pip 安装第三方函数库带来麻烦。位置一般为 C:\Users\名字\AppData\local\Program\Python\Python 版本号\Python.exe。

(3) 设置完成后,单击 Create 按钮创建新的项目,如图 3-18 所示。



图 3-18 为项目设置路径

(4) 进入 PyCharm 主系统,如图 3-19 所示,需要创建 Python 脚本进行开发。单击左侧的项目名称,在弹出菜单中选择 New→Python File 命令。

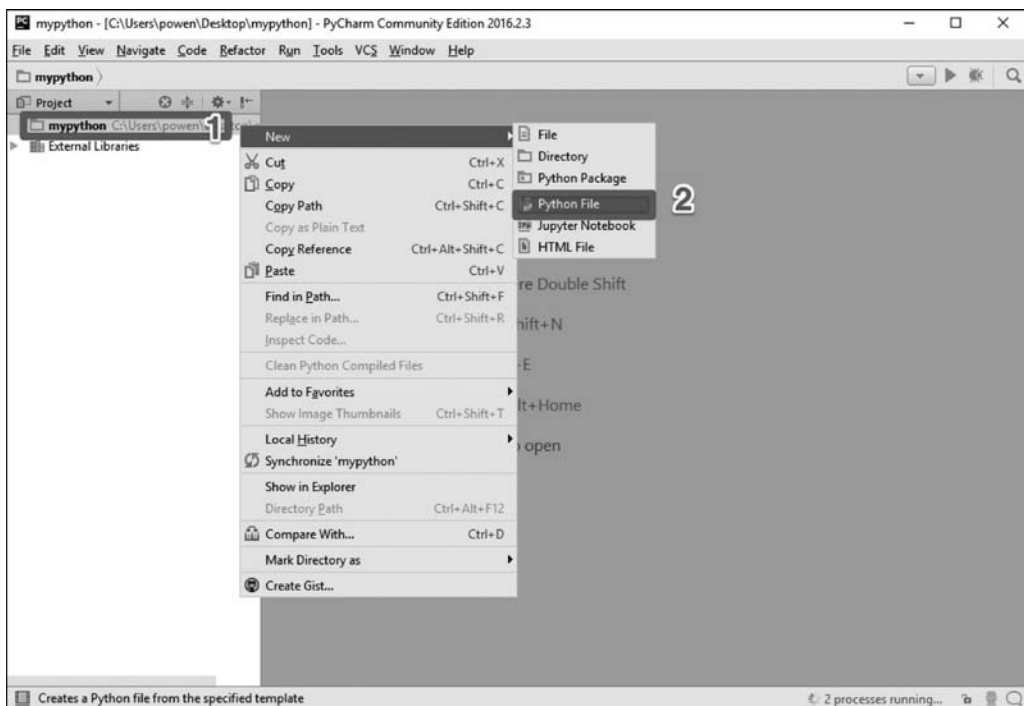


图 3-19 在项目中选择 New→Python File

(5) 设置 Python 脚本的名称,如图 3-20 所示,这里使用的是 mypython.py。

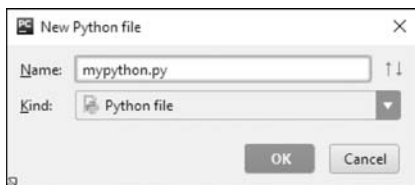


图 3-20 设置 Python 脚本名称

(6) 在新增的 mypython.py 文档中编写以下程序(实例 2——mypython.py):

```
1. print("powenko")
2. print("I love Python")
```

结果如图 3-21 所示。

(7) 运行程序时需要为项目指定主程序。单击窗口左侧的 mypython.py,在弹出菜单中选择 Run 'mypython.py'命令,如图 3-22 所示。

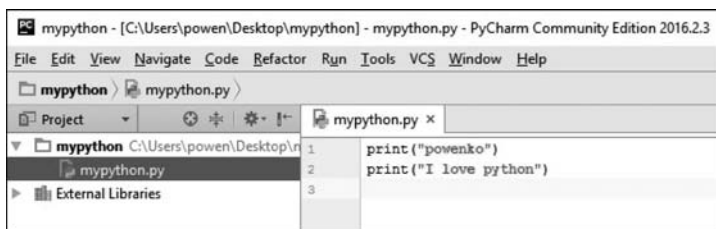


图 3-21 在 mypython.py 文档中编写程序

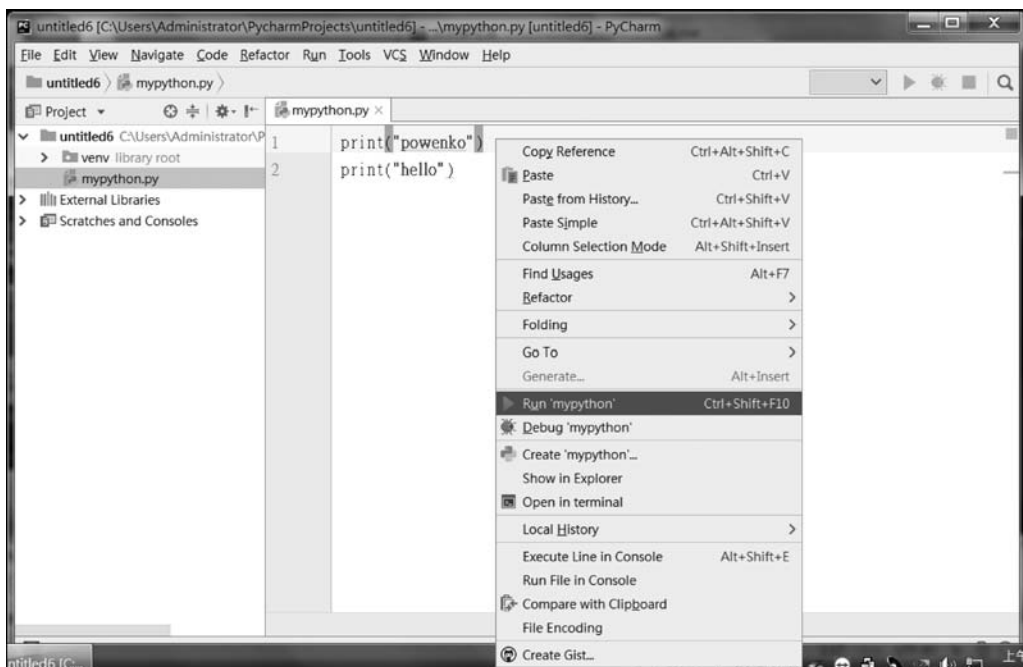


图 3-22 选择 Run 'mypython.py' 命令运行程序

(8) 运行编译的结果将出现在最下方的 Console(控制台)窗口,如有错误,错误提示将出现在同一位置,如图 3-23 所示。

下面简单介绍使用书中配套代码的方法。

只要打开一个项目,找到书中配套的 xxx.py 代码文档,通过鼠标拖放到 PyCharm 的程序区中,即可打开代码程序并运行和调试。该方法相当方便,无须新增项目并设置。本书所有代码均可按照该方法进行开发和测试。

微课视频: 4-Python-PyCharm-tutorial.mp4。





图 3-23 运行结果

通过以下方法,可以确认运行的 Python.exe 的位置。该步骤很重要,因为 PyCharm 在新增项目时,会将整套 Python 软件复制到项目中,后续安装第三方函数库 pip 时,常出现找不到相对应的函数库的问题。

(1) 选择 File→Settings 命令,如图 3-24 所示。

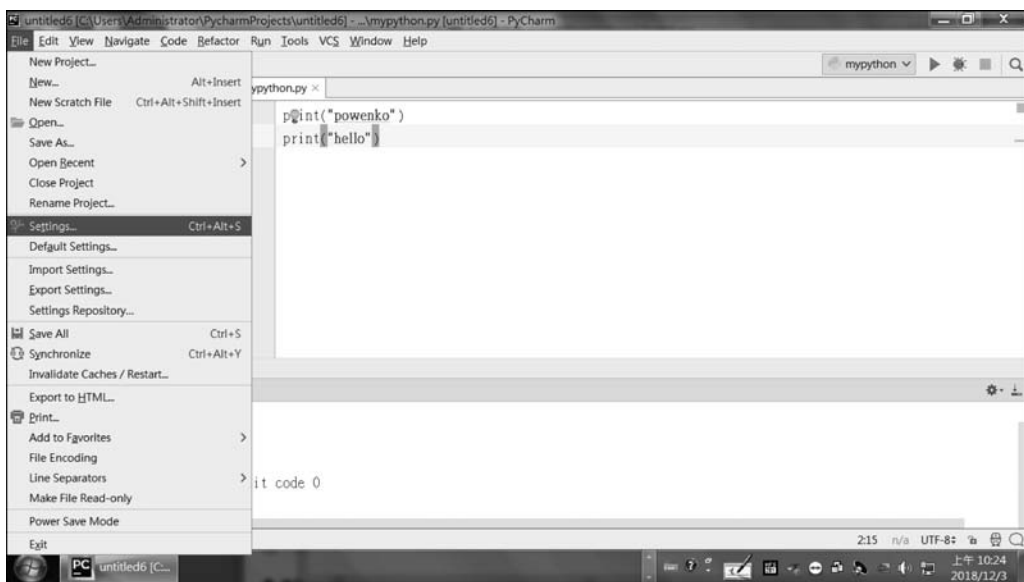


图 3-24 选择 File→Settings

(2) 在 Settings 窗口,打开左侧的 Project:untitled6 下拉选项。选择 Project Interpreter 选项,在右侧的 Project Interpreter 下拉列表框中选择刚刚安装的 Python 3.6.6,这样可以确认此项目用的是同一个 Python,然后单击 OK 按钮,如图 3-25 所示。如果找不到,请重启计算机,或选择 Show All 选项自行指定 Python.exe。

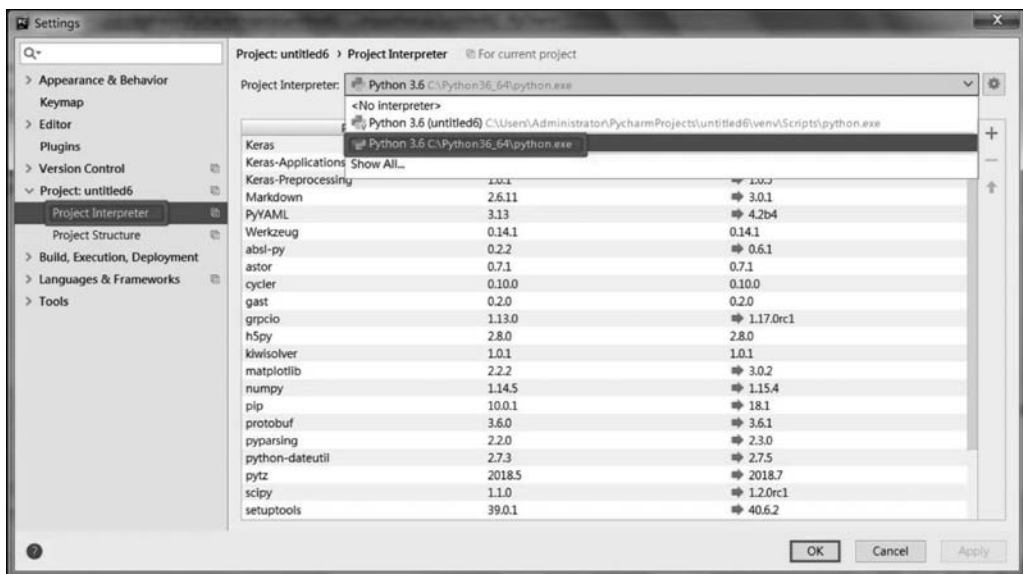


图 3-25 选择 Python 3.6.6 版本

3.6 调试项目

PyCharm 最方便的是调试功能。在程序的数字后方单击,出现如图 3-26 所示的圆圈后,可指定程序断点,重复此步骤可删除断点。

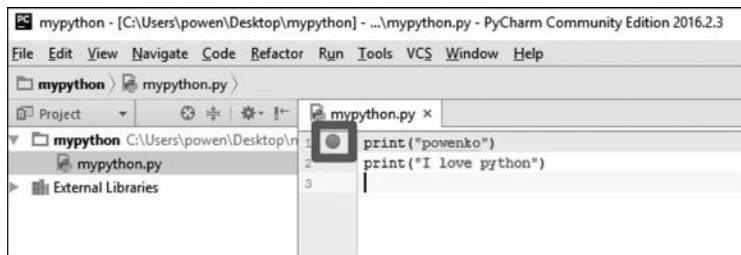


图 3-26 指定程序断点

进入调试模式的步骤如下:

- (1) 在程序的空白处右击。
- (2) 在弹出的菜单中选择 Debug 'mypython'命令,如图 3-27 所示。

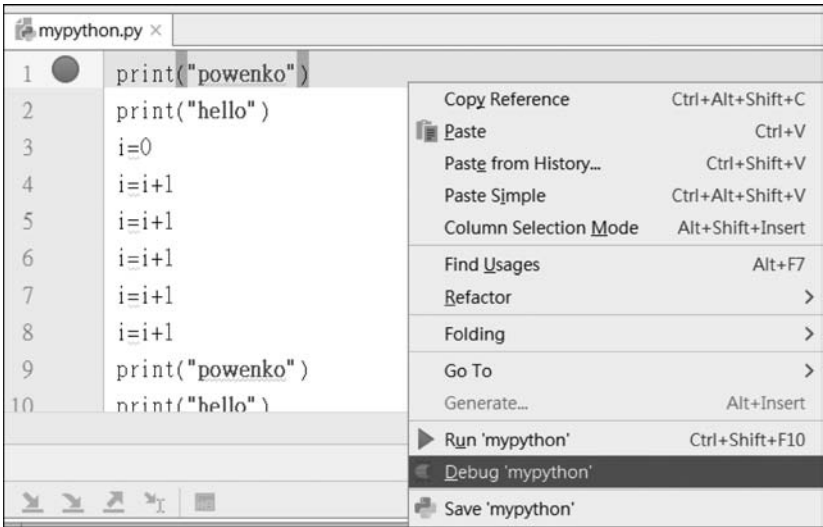


图 3-27 进入调试模式

(3) 只要程序中有指定断点,程序就会停在该位置。此时 PyCharm 软件会进入调试界面,如图 3-28 所示。

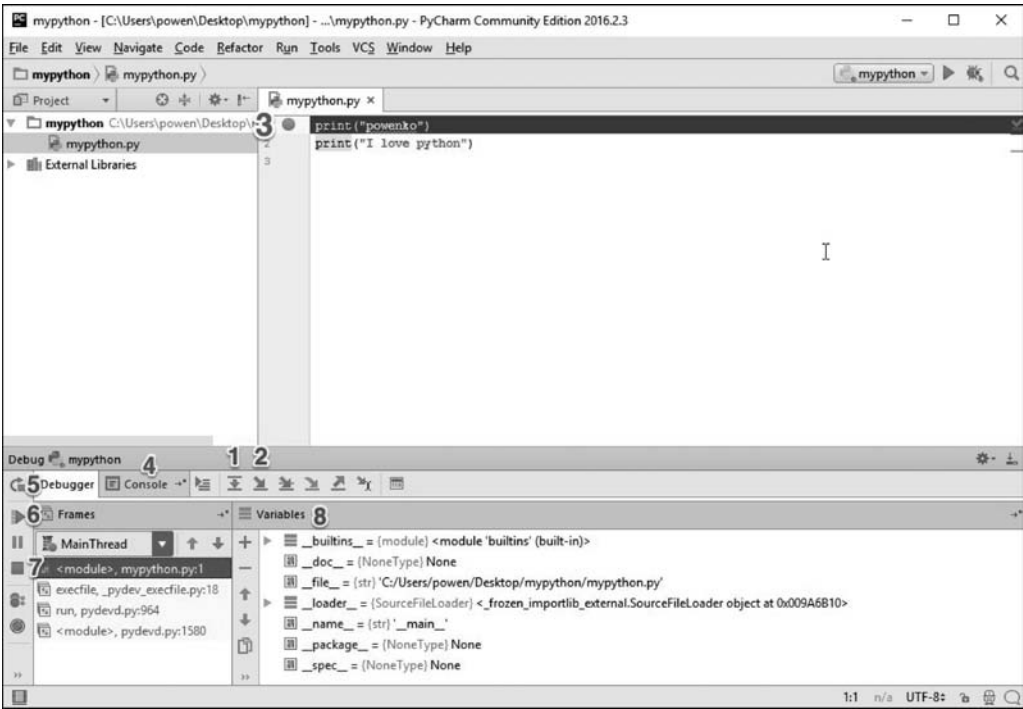


图 3-28 调试模式

在调试模式时,开发者较为常用的功能有:一次运行一行,进入函数内,用蓝底白字表示当前程序运行的位置,切换到 Console 查看程序输出结果,重新运行程序,继续运行直到遇到下一个断点或运行完成,停止调试模式,查看当前程序的变量状态。

PyCharm 安装完毕后,可以安装其他 Packages(包)函数库。

Python 的强大之处就是有大量第三方函数库可以使用。下面介绍下载和安装相关的 Packages 函数库的步骤:

(1) Windows 用户在 PyCharm 中选择 File→Settings 命令; Mac 或 Linux 用户则在 PyCharm 中选择 PyCharm→Preferences→Project→Project Interpreter 命令。

(2) 如图 3-29 所示,在打开的 Settings 窗口中,打开右侧的 Project:untitled6 选项,选择 Project Interpreter 选项。

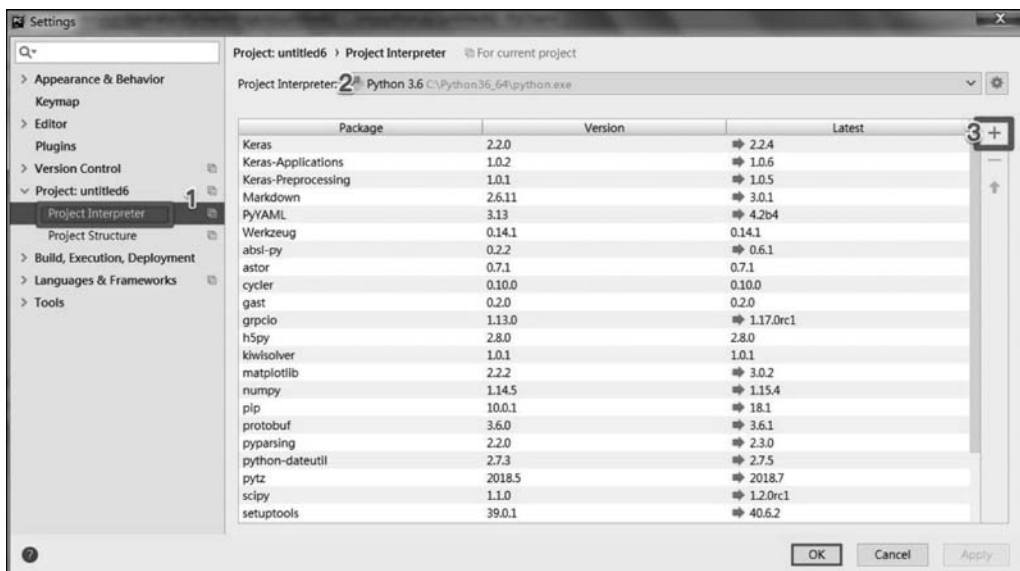


图 3-29 添加和安装其他 Python 函数库

(3) 确认要安装的其他 Packages 函数库位置。建议与 Python 3.6 的路径相同。

(4) 单击+按钮即可添加和安装其他 Python 函数库。

(5) 推荐先选择 pip 的项目包,然后单击↑按钮,将 pip 包升级至最新版本。

(6) 接着查找或选择输入的函数库名称,如 numpy;单击要安装的函数库,如 numpy;单击 Install Package 按钮下载并安装,如图 3-30 所示;出现 Package xx installed successfully 提示即表示安装成功。

另一种安装第三方函数库的方法是通过 pip 命令,将在第 3.9 节详细介绍。

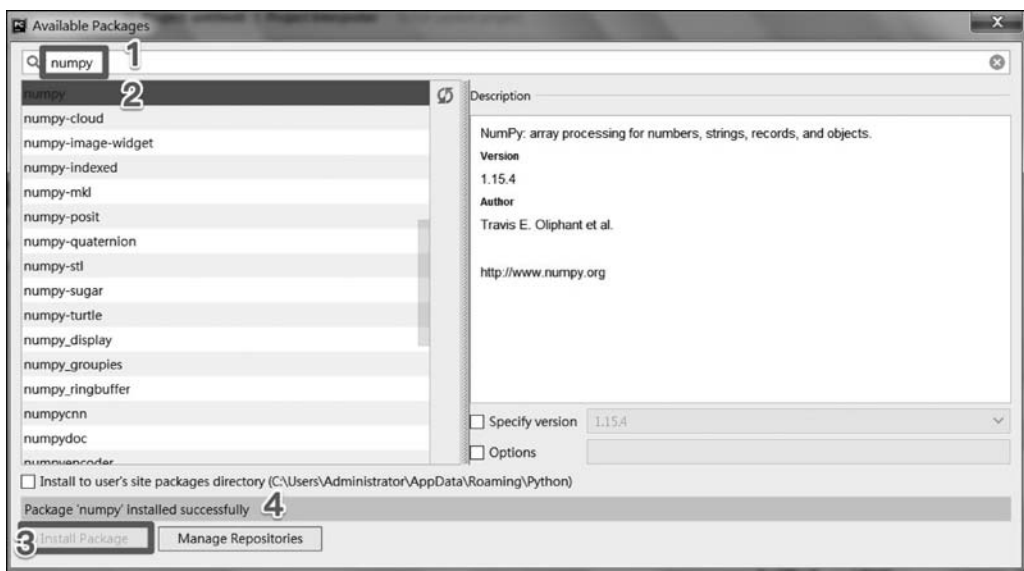


图 3-30 添加和安装其他 Python 的函数库

3.7 安装 Anaconda

Anaconda 也是常用的 Python 开发工具,其特点如下:

- (1) 包含了众多流行的科学、数学、工程及数据分析的 Python 包。
- (2) 完全开源和免费。
- (3) 额外的加速和优化是收费的,但学术用途可以申请免费许可。
- (4) Linux、Windows 及 Mac 全平台支持。
- (5) 支持 Python 2.6、Python 2.7、Python 3.3 及 Python 3.4,可自由切换。
- (6) 内置 Spyder 编译器。
- (7) 自带 Jupyter Notebook 环境,即网页版的 Python。

下载及安装 Anaconda 的步骤如下:

(1) 输入网址 <https://www.continuum.io/downloads>,在打开的网页中选择操作系统,如图 3-31 所示,并选择 Python 的版本。如果没有特殊情况,请选择 Python 3.6。

(2) 单击 Download 按钮即可下载 Anaconda 安装包,下载的安装包如图 3-32 所示。

(3) 下载之后,按图 3-33 所示进行安装。安装过程中选中 Add Anaconda to the system PATH environment variable 复选框(见图 3-33(f)),可以避免因为路径的关系找不到 Anaconda 开发工具。另外,安装完毕后系统将询问是否要安装微软的 VS(Visual Studio)Code,因为本书是用来学习 Python 的,所以单击 Skip 按钮跳过(见图 3-33(h)),即可完成安装。

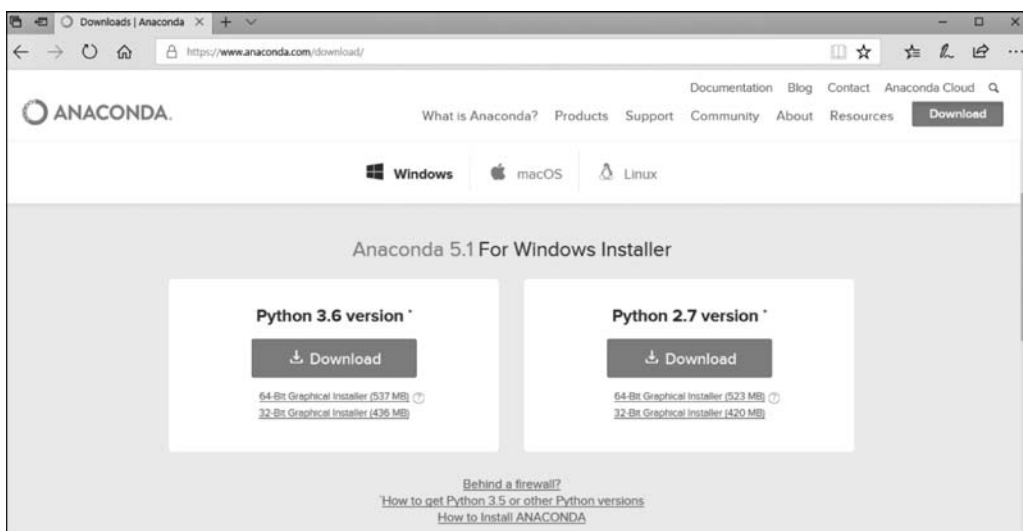


图 3-31 下载 Anaconda

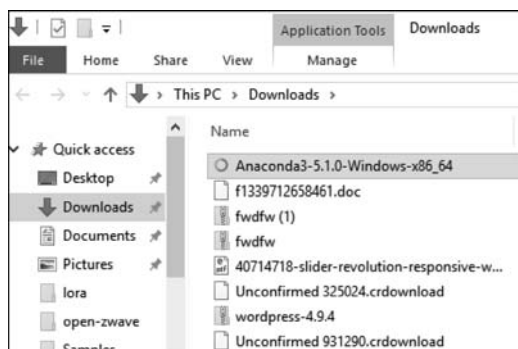


图 3-32 Anaconda 安装包

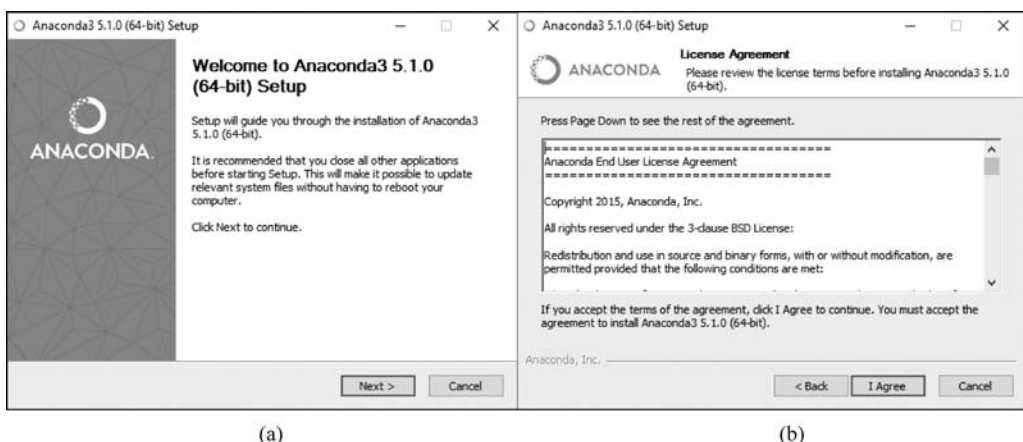


图 3-33 安装过程

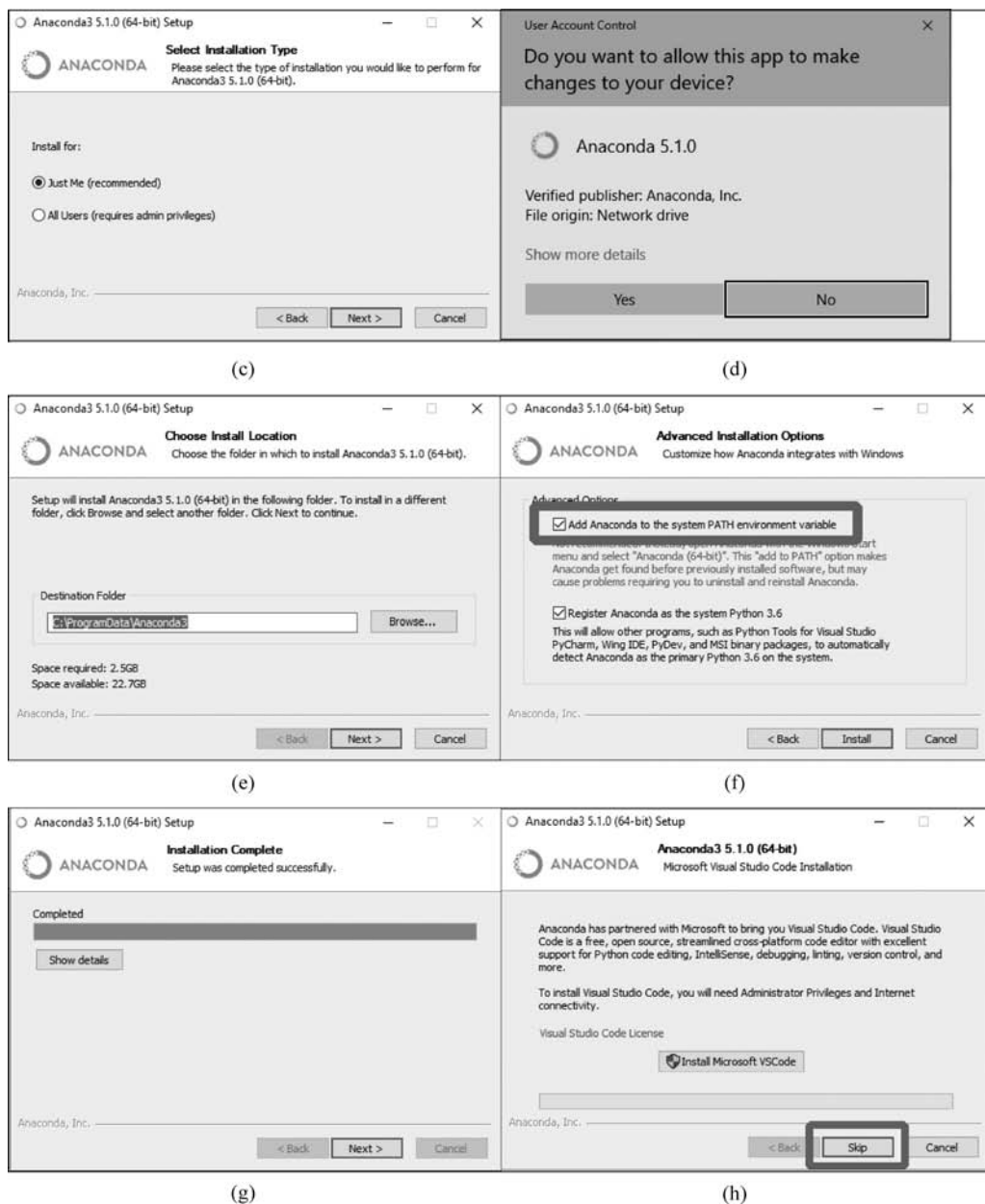


图 3-33 (续)

微课视频：5_Anaconda-download-install.mp4。



3.8 使用 Anaconda

在开始菜单中选择 Anaconda3(64bit)→Spyder 命令,打开 Spyder 开发工具,如图 3-34 所示。

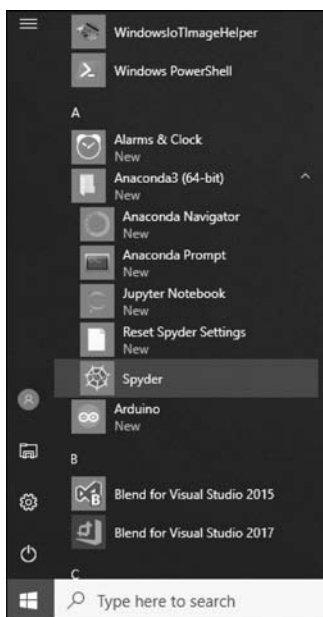


图 3-34 选择 Anaconda3(64bit)→Spyder 命令

如图 3-35 所示为 Anaconda 开发工具 Spyder 的界面。

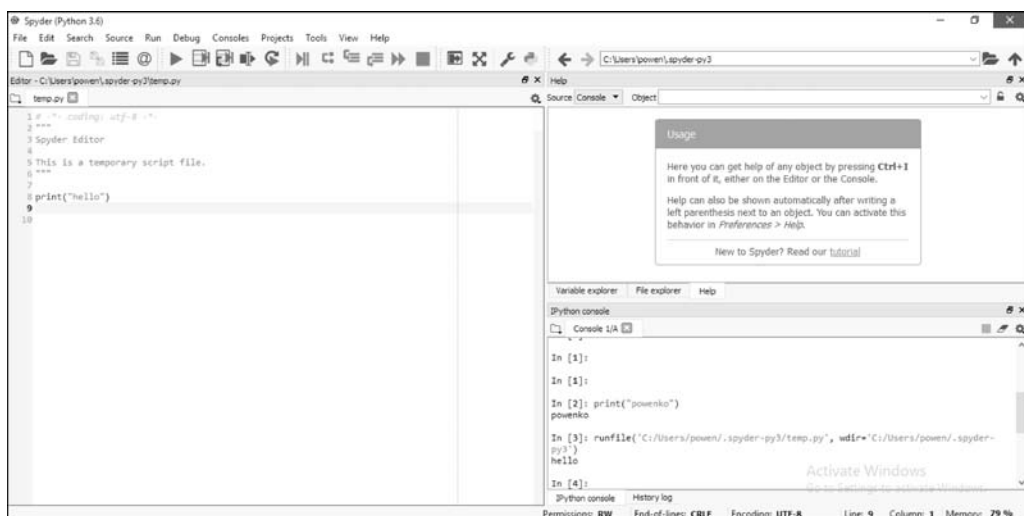


图 3-35 Anaconda 开发工具 Spyder 的界面

使用 Anaconda 后,其系统也将内置网页版的 Python 调试和开发系统 Jupyter,如图 3-36 所示。

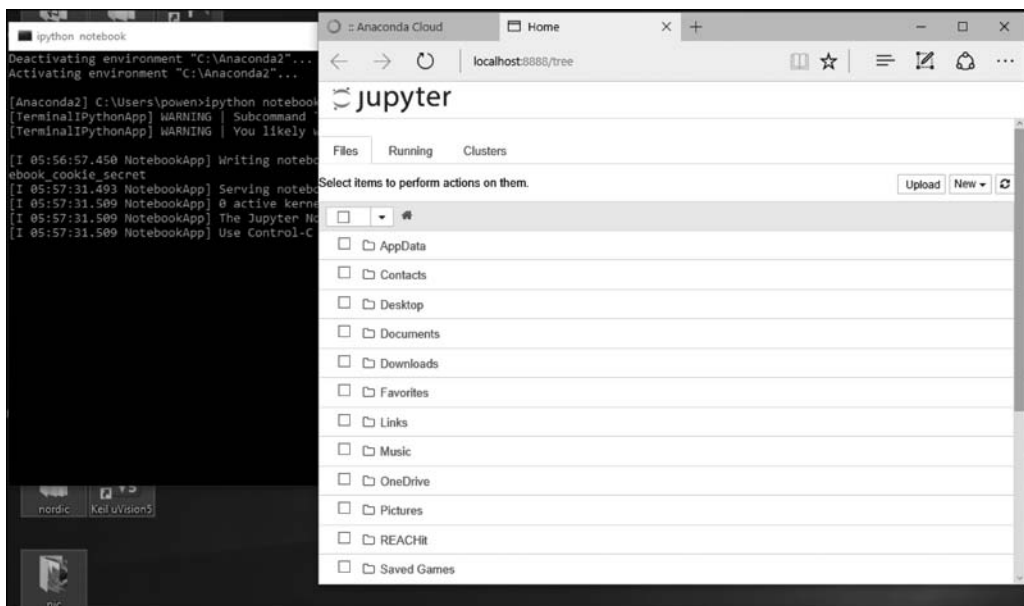


图 3-36 Jupyter 网页版的 Python 调试和开发工具

微课视频: 6_Anaconda-helloWorld. mp4。



3.9 pip 安装包

pip 是 Python 的包管理程序,通过 pip 可以轻松管理、下载和安装第三方的扩展包,而且程序员写程序也更轻松。Python 的扩展包以 .egg 为扩展名,是一个压缩文档。

新的 Python 版本中已经包含了 pip 安装工具,使用 Windows 的用户可以通过管理员权限运行 Command Prompt 命令,进入命令提示符窗口,如图 3-37 所示。

推荐使用 Python 3.X 的读者安装 pip3。pip3 主要针对 Python 3.X 的用户进行开发,与 pip 的命令相同。下面介绍 pip 常用的命令。

(1) 安装。在命令提示符窗口中输入下面的命令:

```
pip install '包名称'
```

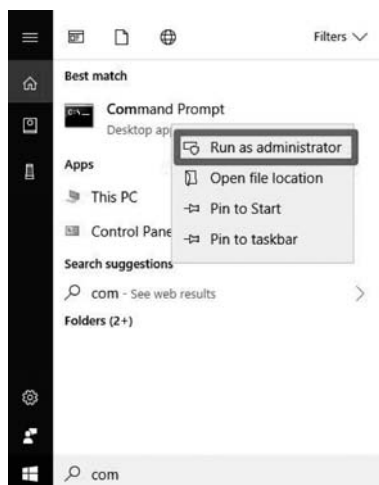


图 3-37 通过管理员权限运行 Command Prompt 命令

默认安装当前最新版本。如安装 numPy 包,命令和结果如图 3-38 所示。

```
C:\Users\Administrator\Desktop>pip install numpy
Collecting numpy
  Using cached https://files.pythonhosted.org/packages/51/70/7096a735b27359dbc0c
380b23b9c9bd05fea62233f95849c43a6b02c5f40/numpy-1.15.4-cp36-none-win_amd64.whl
Installing collected packages: numpy
Successfully installed numpy-1.15.4
```

图 3-38 安装包

也可以安装指定版本,命令为:

```
pip '[包名称] == [版本]'
```

以安装 virtualenv 并指定 1.6.3 版本为例,命令为:

```
pip install virtualenv == 1.6.3
```

也可以指定一个范例的版本,命令为:

```
pip install virtualenv >= 1.6.3
pip install virtualenv < 1.6.3
```

也可以指定一个网络链接来安装,命令为:

```
pip install install http://example.com/virtualenv-1.6.4.zip
pip install git+https://github.com/simplejson/simplejson.git
pip install svn+ssh://svn.zope.org/repos/main/zope.interface/trunk/
```

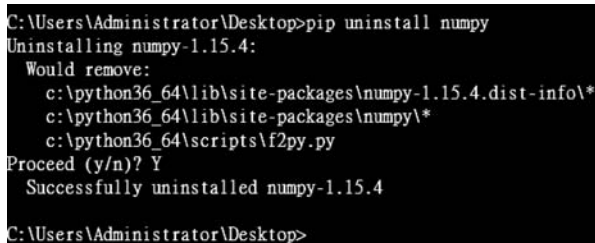
(2) 删除。pip 支持很多自动化清理,后续无须手动清理残留文档。删除命令为:

```
pip uninstall '包名称'
```

如只删除 NumPy, 命令为:

```
pip uninstall numpy
```

也可删除指定版本, 命令和结果如图 3-39 所示。



```
C:\Users\Administrator\Desktop>pip uninstall numpy
Uninstalling numpy-1.15.4:
Would remove:
  c:\python36_64\lib\site-packages\numpy-1.15.4.dist-info\*
  c:\python36_64\lib\site-packages\numpy\*
  c:\python36_64\scripts\fp2py.py
Proceed (y/n)? Y
Successfully uninstalled numpy-1.15.4
C:\Users\Administrator\Desktop>
```

图 3-39 删除包

(3) 升级 pip 安装软件。在命令提示符窗口中输入下列命令即可升级 pip 安装软件:

```
Python -m pip install --upgrad pip
```

(4) 用 pip 列出所有已安装的包的版本。在命令提示符窗口中输入下列命令即可利用 pip 列出所有已安装的包的版本:

```
pip numpy
```

(5) 用 pip 升级包。在命令提示符窗口中输入下列命令即可利用 pip 升级包:

```
pip install -U '包名称'
```

如要更新 NumPy 包, 命令为:

```
pip install -U numpy
```

(6) 列出已安装的包和版本。在命令提示符窗口中输入下列命令即可列出已安装的包和版本:

```
pip list
```

结果如图 3-40 所示。

(7) 利用 pip 查找可安装或管理的包。在命令提示符窗口中输入下列命令即可利用 pip 查找可安装或管理的包:

```
pip search [关键字]
```

```
C:\WINDOWS\system32>pip list
DEPRECATION: The default format will switch to columns in the future. You can avoid this
format=(legacy|columns) in your pip.conf under the [display_options] section.
alabaster (0.7.10)
anaconda-client (1.6.9)
anaconda-navigator (1.7.0)
anaconda-project (0.8.2)
asn1crypto (0.24.0)
astroid (1.6.1)
astropy (2.0.3)
attrs (17.4.0)
Babel (2.5.3)
backports.shutil-get-terminal-size (1.0.0)
beautifulsoup4 (4.6.0)
```

图 3-40 pip list 运行效果

(8) 列出使用范例。如需列出 pip 的使用范例,可在命令提示符窗口中输入以下命令:

```
pip help
```

微课视频: 7_pip-install-uninstall.mp4。



3.10 本书需要安装的第三方函数库列表

为了方便读者学习,下面列出本书需要安装的第三方函数库列表。

pip install Pillow	# 显示图片
pip install Pillow - PIL	# 显示图片
pip install PyInstaller	# 把 Python 程序打包成扩展名为 .exe 的可执行文件
pip install XlsxWriter	# Excel 函数库
pip install beautifulsoup4	# 爬虫等函数库
pip install MySQL - Python	# 数据库 Python 2.X
pip install pymysql	# 数据库 Python
pip install TensorFlow	# TensorFlow 类神经
pip install h5py	# TensorFlow 类神经的权重函数库
pip install jieba	# 中文语意处理
pip install lxml	# XML 处理
pip install matplotlib	# 画图表函数库
pip install opencv - Python	# OpenCV 函数库
pip install opencv - contrib - Python	# OpenCV 函数库
pip install pandas	# 窗体数据函数库
pip install pandas - datareader	# 窗体数据函数库
pip install requests	# 网络函数库
pip install scipy	# 机器学习函数库
pip install xlrd	# XML 处理函数库
pip install xlwt	# XML 处理函数库